

Power BI from Rookie to Rock Star

Author: Reza Rad

Edition: 3, July 2017



RADACAD

Power BI from Rookie to Rock Star



PUBLISHED BY
RADACAD Systems Limited
<http://radacad.com>

89A Fancourt street,
Meadowbank, Auckland 1072
New Zealand

Copyright © 2017 by RADACAD

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Cover: Virgin Silberhorn Ebnefluh, Pixabay

About the book; Quick Intro from Author

In July 2015, after the first release of Power BI Desktop, I've been encouraged to publish a Power BI online book through a set of blog posts. The main reason to publish this book online, was that with the fast pace of updates for Power BI Desktop, it is impossible to publish a paperback book, because it will be out dated in few months. From that time till now, I've been writing blog posts (or sections) of this book almost weekly in RADACAD blog. So far, I have more than 60 sections wrote for this book. The book covers all aspects of Power BI; from data preparation, to modeling, and visualization. From novice to professional level, that's why I called it Power BI from Rookie to Rock Star.

You can start reading this book with no prerequisite. Each section can be read by itself, normally you don't need to follow specific order. However, there are some sections, that need an example previously built in another section. These sections have a prerequisite section mentioning this requirement.

After a year and half of writing online, I decided to release this book as a PDF version as well, for two reasons; First to help community members who are more comfortable with PDF books, or printed version of materials. Second; as a giveaway in my Power BI training courses. Feel free to print this book and keep it in your library, and enjoy. This book is FREE!

This book will be updated with newer editions (hopefully every month), so you can download the latest version of it anytime from my blog post here:

<http://www.radacad.com/online-book-power-bi-from-rookie-to-rockstar>

Because I've been writing these chapters and sections from mid-2015, there are some topics or images or sections out dated with new changes in Power BI. I will do my best to update any changes in next few editions. However, to keep you informed; There is a date at the beginning of each section under the header that mentioned the publish date of that section.

About Author

Reza Rad is the world well-known name in the Microsoft BI field. Reza Rad is invited speaker in world's best and biggest SQL Server and BI conferences such as PASS Summits, PASS Rallies, SQLBits, TechEds, Ignites, SQL Nexus, SQL Days, SQL Saturdays and so on. He is author of books on this topic, and he has more than 15 years' experience in the Microsoft BI technologies. More than 10 years of his experience focused on training and consulting to the largest BI teams in the world. He is also Microsoft Certified Trainer for years. He is Microsoft Data Platform MVP (Most Valuable Professional) focused on BI and Data Analysis, He has been awarded MVP from Microsoft because of his dedication and expertise in Microsoft BI technologies from 2011 till now. He is author of Power BI online book; from Rookie to Rock Star;

www.radacad.com/online-book-power-bi-from-rookie-to-rockstar

These days Reza run Power BI training, consulting, and mentoring in many cities and countries around the world (USA, Canada, Europe, Asia, Australia, and New Zealand). He trained more than 1000 students in just last few months for Power BI. Reza lives in Auckland, New Zealand, but you will probably see him speaking in a conference, or teaching a course near your city or country time by time.

If you are interested to be in touch with Reza, or learn about his upcoming courses, visit RADACAD events page;

<http://radacad.com/events>



Who Is This Book For?

BI Developers and Consultants who want to know how to develop solutions with this technology. BI Architects and Decision Makers who want to make their decision about using or not using Power BI in their BI applications. Business Analysts who want to have better tool for playing with the data and learn tricks of producing insights easier. The book titled "Power BI from Rookie to Rockstar", and that means it will cover wide range of readers. I'll start by writing 100 level and we will go deep into 400 level at some stage. So, if you don't know what Power BI is, or If you are familiar with Power BI but want to learn some deep technical topics about Power Query M language, then this book is for you.

Upcoming Training Courses

Reza runs Power BI training courses both online and in-person. RADACAD also runs Advanced Analytics with R, Power BI, Azure Machine Learning and SQL Server courses ran by Dr. Leila Etaati. Our courses run both online, and in-person in major cities and countries around the world.

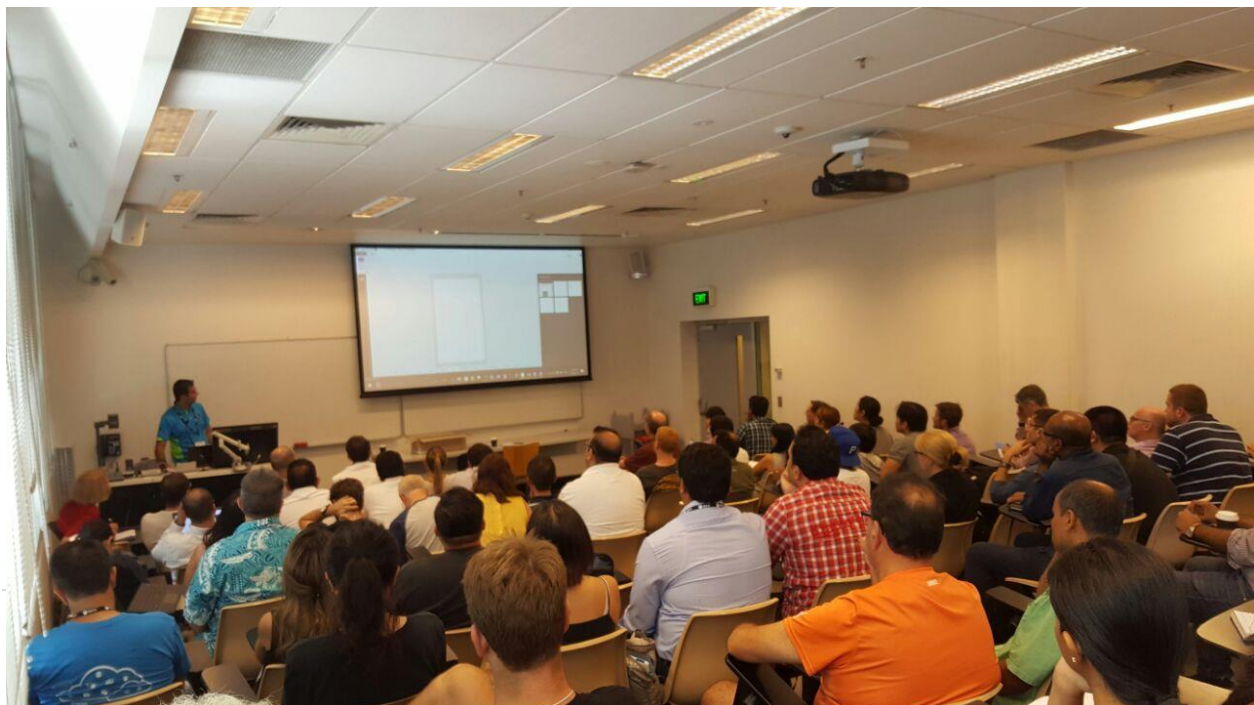
Check schedule of upcoming courses here:

<http://radacad.com/events>

<http://radacad.com/power-bi-training>

some of upcoming events in next few months:

31st of July 2017 – Power BI Live 5-days course – Europe
11th of August 2017 – Advanced Data Modeling and DAX – Colombo, Sri Lanka
14th of August 2017 – Power BI Live 3-days course – Bangalore, India
28th of August 2017 – Power BI Live 5-days course – US East
11th of September 2017 – Power BI Live 5-days course – Asia and Australia West
25th of September 2017 – Power BI Live 5-days course – US West
9th of October 2017 – Power BI Live 5-days course – Australia East
16th of October 2017 – Power BI Live 5-days course – Auckland, New Zealand
20th of November 2017 – Power BI Live 5-days course – Wellington, New Zealand



Heading Table of Content

About the book; Quick Intro from Author	3
About Author	4
Who Is This Book For?.....	5
Heading Table of Content	6
Detailed Table of Content	11
<u>Introduction to Power BI</u>	
Introduction to Power BI: What is Power BI?	29
Power BI Desktop; The First Experience	41
Power BI Website: You'll Need Just A Web Browser	61
<u>Getting Data</u>	
What Is Power Query? Introduction to Data Mash-up Engine of Power BI	78
Get Started with Power Query: Movies Data Mash-Up.....	90
Power BI Get Data from Excel: Everything You Need to Know	117
Power BI Get Data: From Azure SQL Database.....	133
Meetup Data Source for Power BI.....	158
Power BI and Spark on Azure HDInsight; Step by Step Guide	182
<u>Data Transformation with Power Query</u>	
Data Preparation; First and Foremost Important Task in Power BI.....	202
Warning! Misleading Power Query Filtering.....	211
Flawless Date Conversion in Power Query	218
Make Your Numeric Division Faultless in Power Query	230
Pivot and Unpivot with Power BI.....	238
Grouping in Power Query; Getting The Last Item in Each Group	246
Append vs. Merge in Power BI and Power Query	259
<u>Power Query Functions</u>	
Fetch Files and/or Folders with Filtering and Masking: Power Query	268
Power Query; Convert Time Stamp to Date Time	274
How to Change Joining Types in Power BI and Power Query.....	275
Power Query Library of Functions; Shared Keyword	281

[M: Power Query Formula Language](#)

Basics of M: Power Query Formula Language	293
Custom Functions Made Easy in Power BI Desktop	304
Day Number of Year, Power Query Custom Function	328
Power Query Function that Returns Multiple Values	337
Get List of Queries in Power BI	343
Exposing M Code and Query Metadata of Power BI (PBIX) File	352

[Power Query Use Cases](#)

Do You Need a Date Dimension?.....	361
Script to Generate Date Dimension with Power Query M - With Financial Columns	370
Be Fitbit BI Developer in Few Steps: First Step Get Data from CSV	375
Be Fitbit BI Developer in Few Steps: Step 2 Loop Through All CSV Files.....	389
Power Query Not for BI: Event Date and Time Scheduler - Part 1	397
Power Query Not for BI: Event Date and Time Scheduler - Part 2	414
Power Query Not for BI: Event Date and Time Scheduler - Part 3	423

[Data Modeling](#)

Sort by Column in Power BI	430
Relationship in Power BI with Multiple Columns	435

[DAX](#)

M or DAX? That is the Question!	442
Secret of Time Intelligence Functions in Power BI	448
Lost Customers DAX Calculation for Power BI	469
Scenarios of Using Calculated Tables in Power BI	481
Solving DAX Time Zone Issue in Power BI	493
Overwrite Interaction of Power BI with DAX.....	499
Previous Dynamic Period DAX Calculation	505
IF and Filter are Different! Be Careful (DAX)	512

[Data Visualization](#)

Control the Interaction in Power BI Report.....	519
Grouping and Binning; Step Towards Better Data Visualization.....	525
Custom Visuals in Power BI; Build Whatever You Want.....	535

Power Behind the Line Chart in Power BI; Analytics	548
Stacked Chart or Clustered? Which One is the Best?	560
Power BI Waterfall Chart: What's That All About?	568
Storytelling with Power BI Scatter Chart	576
KPIs and Power BI; Visualization Aspect.....	658
Filtering Slicer Resolved in Power BI	666
Be Fitbit BI Developer in Few Steps: Step 3 Visualization	675
<u>Map Visualization</u>	
How to Do Power BI Mapping With Latitude and Longitude Only	590
Power BI Says Hi to 3D Maps.....	595
Shape Map Better than the Filled Map.....	604
Power BI Filled Map; the Good, the Bad, and the Ugly	621
ArcGIS Map in Power BI; Simple, But Insightful.....	635
<u>Architecture</u>	
Step Beyond the 10GB Limitation of Power BI	691
One Data Set, Multiple Reports; Power BI Next Big Step Forward	783
<u>Power BI Service</u>	
Dashboard vs Report; Differences At a Glance - Power BI	702
Power BI Q&A; How to Ask Questions?	899
Dashboard Design for Mobile Power BI	911
Schedule Refresh Local Files on Power BI Web Site	731
<u>Sharing Content</u>	
Power BI Publish to Web - Questions Answered	718
Groups in Power BI	745
Content Pack; Sharing, Self-Service and Governance Together	758
Ultimate Sharing Strategy: Power BI Apps.....	769
<u>Security in Power BI</u>	
Row Level Security Configuration in Power BI Desktop.....	792
Row Level Security with SSAS Tabular Live Connection in Power BI.....	803
Dynamic Row Level Security with Power BI Made Simple	823

[Gateways](#)

Definitive Guide to Power BI Personal Gateway	835
Step by Step Walk through: On-Premises Live SQL Server Connection with Power BI Enterprise Gateway	845
Loop Through On-Premises Files with Power BI and Schedule it to Refresh Automatically	861

[Integration](#)

Power BI On-Premises; Hello World!	918
Power BI Story in PPT Slides with Comments	926
Bring the Power into Your Application; Power BI Embedded	938

[Real-time Dashboards](#)

Stream Analytics and Power BI Join Forces to Real-time Dashboard	954
Monitor Real-time Data with Power BI Dashboards	981

[Performance Tuning](#)

Performance Tip for Power BI; Enable Load Sucks Memory Up	992
Not Folding; the Black Hole of Power Query Performance	1009

[Licencing](#)

Power BI Premium. Is It For You or Not?	1022
---	------

[Power BI for Developers](#)

Integrate Power BI into Your Application: Part 1 - Register your App	1038
Integrate Power BI into Your Application: Part 2 - Authenticate	1049
Integrate Power BI into Your Application: Part 3 - Embed Content	1055
Integrate Power BI into Your Application: Part 4 - Refresh Data Set	1064
Integrate Power BI into Your Application: Part 5 – Data Source Management	1069

Power BI Training	1075
-------------------------	------

Detailed Table of Content

About the book; Quick Intro from Author	3
About Author	4
Who Is This Book For?.....	5
Heading Table of Content	6
Detailed Table of Content	11
Introduction to Power BI: What is Power BI?	29
So, what is Power BI?	30
A Quick Overview of Components.....	31
Power BI Pricing	39
Preparation	40
What to Expect in Next Section?	40
Power BI Desktop; The First Experience	41
Power BI Desktop	41
Prerequisites	42
Startup Screen.....	43
Get Data	44
Modelling Data	51
Visualization	55
Publish.....	59
Power BI Website: You'll Need Just A Web Browser	61
Power BI Website.....	62
Create Reports in Power BI Website	62
Build Dashboard	66
Sharing	68
Get Data	69
Power Q&A.....	71
Power BI Mobile App	73
Summary.....	77

What Is Power Query? Introduction to Data Mash-up Engine of Power BI	78
What Is Power Query?.....	79
How to Use Power Query?	81
What You Can Do With Power Query?	81
Get Data From Wide Range of Sources.....	81
Apply Transformation In a Development Editor	86
Load Data into Destination	87
What Are Features of Power Query Premium?.....	88
Get Started with Power Query: Movies Data Mash-Up.....	90
Let's Get Started	92
Query Editor	96
Use a Query as a Reference.....	100
Append Queries.....	101
Extract First Characters.....	102
Remove Columns.....	105
Split Column.....	106
Replace Values	109
Trim.....	111
Applied Steps	111
Final Merge.....	112
Summary.....	116
Power BI Get Data from Excel: Everything You Need to Know	117
Excel Data Source	117
Loading Excel Tables into Power Query	119
Loading Excel Sheets into Power Query.....	121
What If Your Excel Table Has Merged Cells?	124
Example Excel Data Source: Olympic Games.....	124
Summary.....	132
Power BI Get Data: From Azure SQL Database.....	133
Preparation	134

Get Data from Azure SQL Database.....	142
Schedule Data Refresh.....	149
Direct Connection to Azure SQL Database from Power BI Website.....	153
Summary.....	157
Meetup Data Source for Power BI.....	158
About Meetup.....	158
Get Data with Power BI.....	162
TimeStamp to Date Time.....	166
Visualization	168
Adding More Data.....	172
Power BI and Spark on Azure HDInsight; Step by Step Guide	182
A Bit of Introduction.....	183
Setup and Configure Spark on Azure HDInsight	184
Setup the Sample Table for Spark on Azure HDInsight	190
Connection from Power BI Desktop to Spark on Azure HDInsight	194
Connection from Power BI website to Spark on Azure HDInsight	199
Data Preparation; First and Foremost Important Task in Power BI.....	202
Why Data Preparation?	202
How to Design a Star Schema?.....	204
Design Tips	208
More to Come.....	210
Warning! Misleading Power Query Filtering.....	211
Filtering in Power Query	211
Basic Filtering	212
Misleading Behavior of Basic Filtering.....	213
Advanced Filtering: Correct Filtering	216
Summary.....	217
Flawless Date Conversion in Power Query.....	218
Different Formats of Date.....	219
Automatic Type Conversion.....	221

Date Conversion Issue	224
Date Conversion Using Locale	226
Summary	229
Make Your Numeric Division Faultless in Power Query	230
Sample Data Set.....	230
Simple Division Calculation.....	231
Error Output	232
Infinity	234
NaN	235
Null Check.....	235
Function to Check All Anomalies.....	236
Summary	237
Pivot and Unpivot with Power BI.....	238
Pivot: Turning Name,Value Rows to Columns	238
Unpivot; Turning Columns to Rows; Name, Values	243
Grouping in Power Query; Getting The Last Item in Each Group	246
Learning Objectives for this section.....	246
Scenario	247
Get Data	247
Group By Transformation	248
First and Last Item in each Group	252
Append vs. Merge in Power BI and Power Query	259
Why Combine Queries?	259
Append.....	260
Merge	262
Fetch Files and/or Folders with Filtering and Masking: Power Query	268
Fetch All Files in a Folder	268
Fetch All Files and Folders	270
Fetch Files and Folders with Masking	272
Power Query; Convert Time Stamp to Date Time	274

What is Timestamp	274
Power Query Convert Timestamp to Date Time	274
How to Change Joining Types in Power BI and Power Query	275
Power BI Desktop	276
Power Query	278
Change Through M	279
Power Query Library of Functions; Shared Keyword	281
#shared Keyword	283
Use the Result Set as a Table	287
Documentation of Function	290
Enumerators	292
Summary	292
Basics of M: Power Query Formula Language	293
What is M?	293
Syntax of M	294
End of the Line	296
Variable Names	297
Step by Step Coding	299
Literals	299
Function Call	300
Comments	301
A real-world example	302
Custom Functions Made Easy in Power BI Desktop	304
What is Custom Function?	304
Benefits of Custom Function	305
How to Create Custom Function?	306
Summary	327
Day Number of Year, Power Query Custom Function	328
Power Query Function that Returns Multiple Values	337
Return First and Last Dates of Month	338

Don't Limit Yourself.....	342
Get List of Queries in Power BI	343
Question: How to Fetch Name of All Queries into One Query?	343
Answer: Using #shared or #sections Keywords.....	344
Sample Scenario of Usage.....	349
Exposing M Code and Query Metadata of Power BI (PBIX) File	352
Structure of a PBIX File	352
DataMashup File: Everything You Need	353
M Script File.....	355
Metadata Information in XML Format.....	357
Summary.....	360
Do You Need a Date Dimension?.....	361
What is Date Dimension?	361
Why Date Dimension?	362
Public Holidays Insight.....	365
Extended Functions for Time Intelligence.....	366
Do I Need a Date Dimension?	368
How to Build a Date Dimension?	368
Script to Generate Date Dimension with Power Query M - With Financial Columns.....	370
Be Fitbit BI Developer in Few Steps: First Step Get Data from CSV.....	375
Be Fitbit BI Developer in Few Steps: Step 2 Loop Through All CSV Files.....	389
Looping Through CSV Files.....	389
Where We've been in Previous Step.....	390
Get Data From Folder.....	391
Custom Function.....	392
EACH Singleton Function	393
Combining, Appending, or UNION of Tables.....	394
Power Query Not for BI: Event Date and Time Scheduler - Part 1	397
Why?.....	397
What You Will Learn?	401

Get Data from Web	403
Apply Basic Transformations.....	406
Insert a Step.....	410
Append Queries.....	412
Next Steps.....	413
Power Query Not for BI: Event Date and Time Scheduler - Part 2	414
Group Rows	414
Concatenate Rows	416
Final Polish.....	421
Next Steps.....	421
Power Query Not for BI: Event Date and Time Scheduler - Part 3	423
Using Parameters.....	423
Summary	429
Sort by Column in Power BI	430
Sorting in Power BI	431
Sort By Column	432
More Cases to use Sort By Column.....	433
Hide the Column Used for Sorting.....	434
Relationship in Power BI with Multiple Columns	435
Defining the Problem.....	435
Workaround	438
M or DAX? That is the Question!	442
What is M?	442
What is DAX?.....	443
Example Usage of M	444
Example Usage of DAX?	444
Calculated Column Dilemma	445
Which one is best?	446
As a Power BI Developer Which Language Is Important to Learn?	447
Secret of Time Intelligence Functions in Power BI	448

Time Intelligence in Power Pivot	450
Time Intelligence Functions Issue in Power BI.....	457
Adding ALL(<Date Table>)	458
Import Date Table from Power Pivot	466
Lost Customers DAX Calculation for Power BI	469
The Model.....	470
Method.....	472
Scenarios of Using Calculated Tables in Power BI	481
Role Playing Dimension	482
In-Memory Structure, Less Refresh Time	486
DAX Table Functions	488
Top 100 Customers as a Calculated Table	488
Limitations	492
Solving DAX Time Zone Issue in Power BI	493
Defining the Problem	493
Method 1 – DAX Formula Manipulation	494
Method 2 – Power Query DateTimeZone Functions	495
Method 3 – Web Query with Power Query.....	496
Overwrite Interaction of Power BI with DAX.....	499
Filter Context in DAX.....	500
Total Sales Regardless of Filters	501
Total Sales Filterable Only by Date Selection	502
Measure Filterable with Multiple Selections	503
Summary.....	504
Previous Dynamic Period DAX Calculation	505
Current Period.....	506
Previous Period	508
Summary.....	511
IF and Filter are Different! Be Careful (DAX)	512
Brief of Functions.....	513

Conditional SUM.....	514
Which One to Use?.....	518
Control the Interaction in Power BI Report.....	519
What is Interactivity of Power BI Visuals?.....	519
Default Behavior	520
Changing the Interaction.....	521
Grouping and Binning; Step Towards Better Data Visualization.....	525
Banding or Binning.....	526
Grouping.....	530
Behind the Scene	534
Summary	534
Custom Visuals in Power BI; Build Whatever You Want.....	535
Synoptic Panel by SQL BI.....	536
Chiclet Slicer	544
Power Behind the Line Chart in Power BI; Analytics	548
Introduction to line types in the line chart	549
Building a Line Chart	549
Trend Line	554
Analytics Tab	555
Constant Line	556
Dynamic Reference Lines.....	558
Do Not Overuse	559
Stacked Chart or Clustered? Which One is the Best?	560
Simple Column Chart	561
Clustered Chart	562
Stacked Chart	563
100% Stacked Chart	564
Data Label is Your Asset	565
Your Story of Data	566
Be Careful of Too Many Items	567

Power BI Waterfall Chart: What's That All About?	568
What is Waterfall Chart?	569
When to use Waterfall Chart?	569
How to use this Chart?	569
Storytelling with Power BI Scatter Chart	576
Scatter Chart	577
Building Sample Report	577
Formatting	588
Summary	588
How to Do Power BI Mapping With Latitude and Longitude Only	590
Power BI Says Hi to 3D Maps	595
3D Visualization in Power BI Desktop	595
Power Map	600
Conclusion	602
A Question for You	603
Shape Map Better than the Filled Map	604
Maps in Power BI	605
Shape Map	605
How Shape Map Works?	608
Add a Custom Map	610
Where to Find Map?	611
Convert Shape File to JSON	612
Pros and Cons of Shape Map	617
Summary	619
Power BI Filled Map; the Good, the Bad, and the Ugly	621
The Good: How to Use Filled Map	622
The Ugly: When Geographic Structure is Different	626
The Bad: When Data Category is Useless	632
Summary	634
ArcGIS Map in Power BI; Simple, But Insightful	635

Maps in Power BI	636
ArcGIS Map Visual	636
ArcGIS Features	642
Adding Time Dimension	655
Summary	656
Download Demo Files	656
KPIs and Power BI; Visualization Aspect	658
What is KPI?	658
KPI Elements	659
Built-in KPI	659
Bullet Chart	662
Dial Gauge	664
Linear Gauge	665
Summary	665
Filtering Slicer Resolved in Power BI	666
Defining the Problem	666
Calculated Tables	667
Sample Scenario	668
Create Calculated Table for Filtered View	670
Expand the Possibilities	673
Limitations	673
Summary	674
Be Fitbit BI Developer in Few Steps: Step 3 Visualization	675
Adding Date Dimension	675
A Touch to Modelling	677
Let's Visualize	679
Step Beyond the 10GB Limitation of Power BI	691
Compressed Structure of Power BI	691
Live Connection	692
Live Connection to SQL Server On Premises	692

Live Connection to Azure	696
Limitations of Live Connection	696
Optimization at Data Source Level.....	700
Dashboard vs Report; Differences At a Glance - Power BI	702
Definition	703
Beyond the Definition	703
Unique Features of a Power BI Report	706
Unique Features of a Power BI Dashboard	710
Differences At a Glance.....	717
Power BI Publish to Web - Questions Answered	718
What is Publish to Web?	719
How to Do It?	719
Security Thoughts?	723
What You Share is for Everyone!	723
Difference with Sharing?.....	725
Power BI Embedded	730
Schedule Refresh Local Files on Power BI Web Site	731
Install and Configure Gateway.....	732
Deploy Power BI Solution.....	736
Schedule Refresh	739
Testing the Refresh.....	742
Summary.....	744
Groups in Power BI	745
Why Groups?	745
Creating Groups in Power BI	746
Publish Content to Group.....	750
Office 365 Synchronization	753
Power BI Pro	757
Content Pack; Sharing, Self-Service and Governance Together	758
Content Packs	758

How to Create Organizational Content Pack?	759
Consuming Content Packs	761
Advantages of Content Packs	764
Summary	768
Ultimate Sharing Strategy: Power BI Apps	769
Power BI Sharing Methods	770
Power BI App	770
My Thoughts on Power BI Apps	781
One Data Set, Multiple Reports; Power BI Next Big Step Forward	783
Get Data from Power BI Service	784
What This Change Means?	788
Limitations	790
Summary	791
Row Level Security Configuration in Power BI Desktop	792
What's Good About Row Level Security in Power BI Desktop?	792
Create Sample Report	793
Creating Roles	795
Testing Roles in Desktop	797
Power BI Service Configuration	799
Test Roles in Power BI Service	801
Re Publish Won't Hurt	802
Row Level Security with SSAS Tabular Live Connection in Power BI	803
Install and Configure Enterprise Gateway	804
Create Data Source	808
Effective User Name	811
UPN Mapping	812
Configure Row Level Security in SSAS Tabular	815
Create Sample Report	818
Test the Result	820
How does it work?	821

Summary	821
Dynamic Row Level Security with Power BI Made Simple	823
Sample Report	826
DAX USERNAME()	828
Assign users to Power BI Security	830
Share the Dashboard	832
Summary	834
Definitive Guide to Power BI Personal Gateway	835
What Is Power BI Personal Gateway?	836
How to Install Power BI Personal Gateway?	837
Do I Need More than One Instance of Gateway Installed?	838
How to Set Scheduled Refresh in Power BI Website?	838
Limitations of Power BI Personal Gateway	842
Myths and Misconceptions About Power BI Personal Gateway	843
Step by Step Walk through: On-Premises Live SQL Server Connection with Power BI Enterprise Gateway	845
Install and Configure Enterprise Gateway	846
Create Data Source	850
Building Sample Power BI Report	852
Test the Live Report	857
Live Refresh	858
Advantages of Live Connection	859
DirectQuery or Live SQL Server On-Premises Limitations	860
Loop Through On-Premises Files with Power BI and Schedule it to Refresh Automatically	861
Defining the Challenge	861
First Approach: Custom Function	862
Second Approach: Combine Files	888
Which Method is Best?	898
Power BI Q&A; How to Ask Questions?	899
Explore the Data	900

Explore a Dataset	900
Filter Values	901
Explore Related Datasets	902
Sorting Data	903
More Filters.....	904
Aggregation	907
Visualization Customization	908
Combination of some above.....	909
Your Turn	910
Dashboard Design for Mobile Power BI	911
Change Dashboard View	912
View Reports on Mobile Device	916
Mobile Reports with Power BI Gets Better and Better	916
Power Q&A for Mobile Dashboard?	916
Power BI On-Premises; Hello World!	918
Technical Preview of SQL Server Reporting Services.....	918
Installation	919
Building a Report.....	922
Publish to On-Premises	922
My Thoughts	924
Power BI Story in PPT Slides with Comments.....	926
Power BI Tiles	927
Adding Power BI Tiles to Slides	929
Storytelling with Power BI and Power Point	932
Images or Re-Loaded Tile Feature	933
Working with other Work Spaces.....	934
Reports in Tile	936
Summary.....	937
Bring the Power into Your Application; Power BI Embedded	938
What is Power BI Embedded?	938

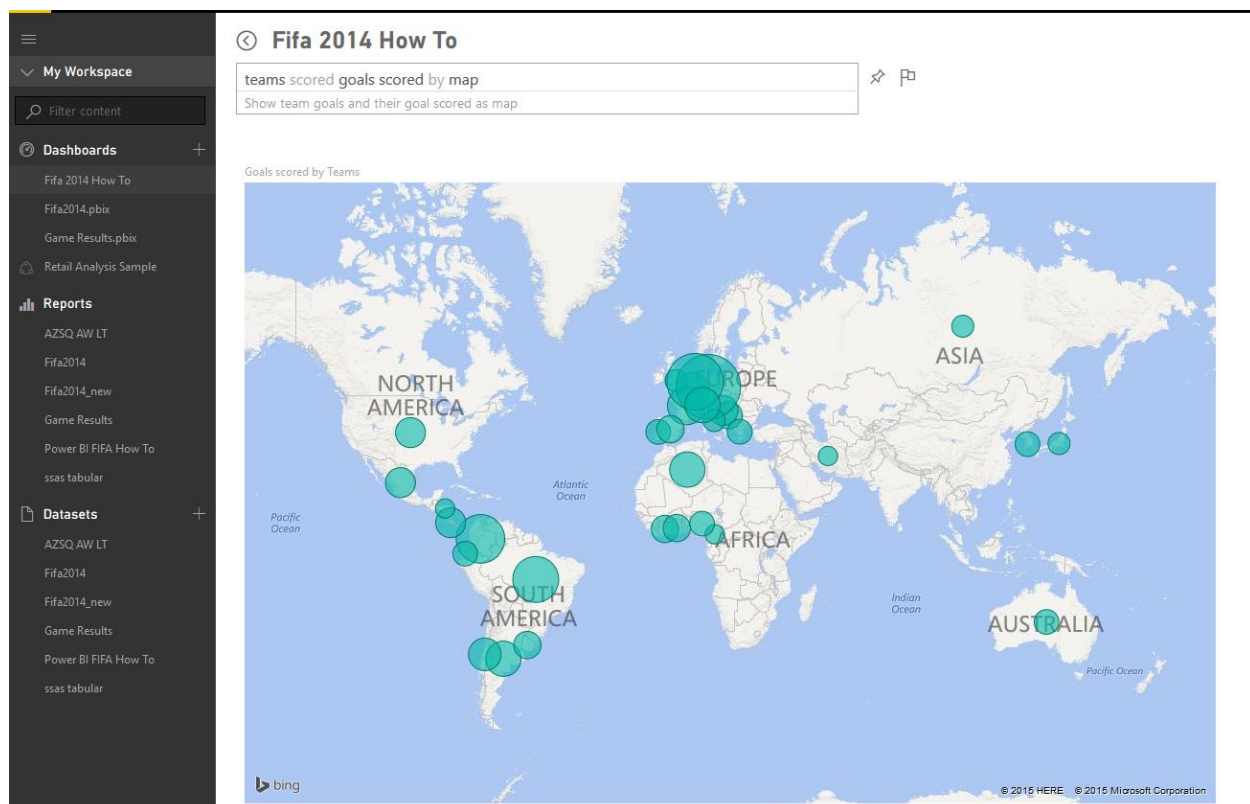
Key Advantages of Power BI Embedded	939
Getting Started with Power BI Embedded	939
Power BI Workspace Collection	942
Sample .NET Projects	944
Create Workspace	947
Publish Report.....	947
Update Connection String of the Report	948
Testing the Embedded Report.....	949
Security Thoughts?	951
Pricing	952
Limitations of Power BI Embedded	953
Stream Analytics and Power BI Join Forces to Real-time Dashboard	954
Architecture	955
Azure Event Hub	955
Azure Stream Analytics	960
Push Data Application.....	970
Power BI Real-time Dashboard.....	976
Think Further	980
Monitor Real-time Data with Power BI Dashboards.....	981
Real-Time Data, Instant Monitoring.....	981
Real-time Data Set.....	982
Create Sample Streaming Data Set.....	985
Sample Streaming Data API Application	987
Create Dashboard Tile	988
API Data Can Be Anything.....	991
Performance Tip for Power BI; Enable Load Sucks Memory Up	992
Load Mechanism for Power Query.....	992
Example Scenario.....	994
Build the Transformations	995
Default Behavior: Enable Load	1002

What about Hiding from Report?	1004
Disable Load to Save Memory	1005
Summary	1008
Not Folding; the Black Hole of Power Query Performance	1009
Query Folding.....	1010
Is Query Folding Good or Bad?	1011
Can I see the Native Query?	1012
So Why Not Query Folding?	1013
Example: Merge Columns.....	1013
How Do I know Which Transformations Folds?	1020
Power BI Premium. Is It For You or Not?	1022
Power BI Licensing Before Premium	1024
Why Premium?	1026
What is Power BI Premium?	1027
What Are Benefits of Power BI Premium?	1028
Power BI Report Server	1030
The Most Important Topic! Costs of Premium	1032
What is Break-Even Point?.....	1034
Summary	1036
Your Thoughts?.....	1037
Integrate Power BI into Your Application: Part 1 - Register your App.....	1038
Power BI Embedded – Previous Version.....	1039
Integrate Power BI content into your Application – New Way	1039
Power BI REST API	1039
Register Your Application.....	1040
Integrate Power BI into Your Application: Part 2 - Authenticate	1049
Step 2: Authenticate	1050
Get Access Token from C# Code.....	1051
Sample Application Structure	1052
Authentication Process	1052

Power BI Client Object	1053
Next Steps.....	1054
Integrate Power BI into Your Application: Part 3 - Embed Content.....	1055
Step 3: Embed Content.....	1056
Power BI Client Object	1056
Get List of Reports.....	1056
JAVA Script in the Web page	1058
Testing Application.....	1059
Getting List of Dashboards.....	1060
Embed Dashboard.....	1061
Getting Tiles	1062
Embedding Content Limitless.....	1063
Integrate Power BI into Your Application: Part 4 - Refresh Data Set	1064
Step 3: Get Data Set	1065
Refresh Data Set	1066
Refresh History.....	1067
Integrate Power BI into Your Application: Part 5 – Data Source Management	1069
Get List of Gateways.....	1069
Get List of Data Sources	1070
Data Source Management	1073
Summary.....	1073
Power BI Training.....	1075

Introduction to Power BI: What is Power BI?

Published Date: August 8, 2015



Power BI is not a new name in the BI market, components of Power BI has been in the market through different time periods. Some components such As [Power BI Desktop is such new that released as general availability at 24th of July](#). On the other hand [Power Pivot released at 2010](#) for the first time. Microsoft team worked through long period of time to build a big umbrella called Power BI, this big umbrella is not just a visualization tool such as Tableau, it is not just a self-service data analysis tool such as PivotTable and PivotChart in Excel, it is not just a cloud based tool for data analysis. Power BI is combination of all of those, and it is much more. With Power BI you can connect to many data sources (wide range of data sources supported, and more data sources add to the list every month). You can mash up the data as you want with a very powerful data mash up engine. You can model the data, build your star schema, or add measures and calculated columns with an In-Memory super fast engine. You can visualize data with great range of data visualization elements and customize it to tell the story behind

the data. You can publish your dashboard and visualization tool in cloud and share it to those who you want. You can work with On-premises as well as Azure/cloud based data sources. and believe me there are much more things that you can do with Power BI which you can't do with other products easily.

So, what is Power BI?

There are many definitions for this tool, here is my version of it simplified for everyone to understand;

Power BI is a cloud based data analysis, which can be used for reporting and data analysis from wide range of data source. Power BI is simple and user friendly enough that business analysts and power users can work with it and get benefits of it. On the other hand Power BI is powerful and mature enough that can be used in enterprise systems by BI developers for complex data mash-up and modelling scenarios.

Power BI made of 6 main components, these components released in the market separately, and they can be used even individually. Components of Power BI are:

- Power Query: Data mash up and transformation tool.
- Power Pivot: In-memory tabular data modelling tool
- Power View: Data visualization tool
- Power Map: 3D Geo-spatial data visualization tool
- Power Q&A: Natural language question and answering engine.
- Power BI Desktop: A powerful companion development tool for Power BI

There are many other parts for Power BI as well, such as;

- PowerBI.com Website; which Power BI data analysis can be shared through this website and hosted there as cloud service
- Power BI Mobile Apps; Power BI supported in Android, Apple, and Windows Phones.

Some of above components are strong and has been tested for very long time. Some of them however are new and under frequent regular updates. Power BI built easy graphical user interfaces to follow, so a business user simply could user Power Query or Power BI desktop to mash up the data without writing even a single line of code. It is on the other hand so powerful with power query formula language (M) and data analysis expression (DAX) that every developer can write complex codes for data mash up and calculated measures to respond challenging requirements. So if you've heard

somewhere that Power BI is a basic self-service data analysis tool for business analysts and cannot be used for large enterprises systems, I have to say this is totally wrong! I've been using Power BI technology myself in many large enterprise scale systems and applications, and I've seen usage of that in many case studies all around the world.

Power BI components can be used individually or in a combination. Power Query has an add-in for Excel 2010 and Excel 2013, and it is embedded in Excel 2016. The add-in for Power Query is available for free! for everyone to download and use it alongside with existing an Excel (as long as it is Excel 2010 or higher versions). Power Pivot has been as an add-in for Excel 2010, from Excel 2013 Power Pivot is embedded in Excel, this add-in is again free to use! Power View is an add-in for Excel 2013, and it is free for use again. Power Map is an add-in for Excel 2013, it is embedded in Excel 2016 as 3D maps. Power Q&A doesn't require any installation or add-in, it is just an engine for question and answering that works on top of models built in Power BI with other components.

Components above can be used in a combination. You can mash up the data with Power Query, and load the result set into Power Pivot model. You can use the model you've built in Power Pivot for data visualization in Power View or Power Map. There is fortunately a great development tool that combines three main components of Power BI. Power BI Desktop is the tool that gives you combined editor of Power Query, Power Pivot, and Power View. Power BI Desktop is available as stand-alone product that can be downloaded separately. With Power BI Desktop you will have all parts of the solution in one holistic view.

A Quick Overview of Components

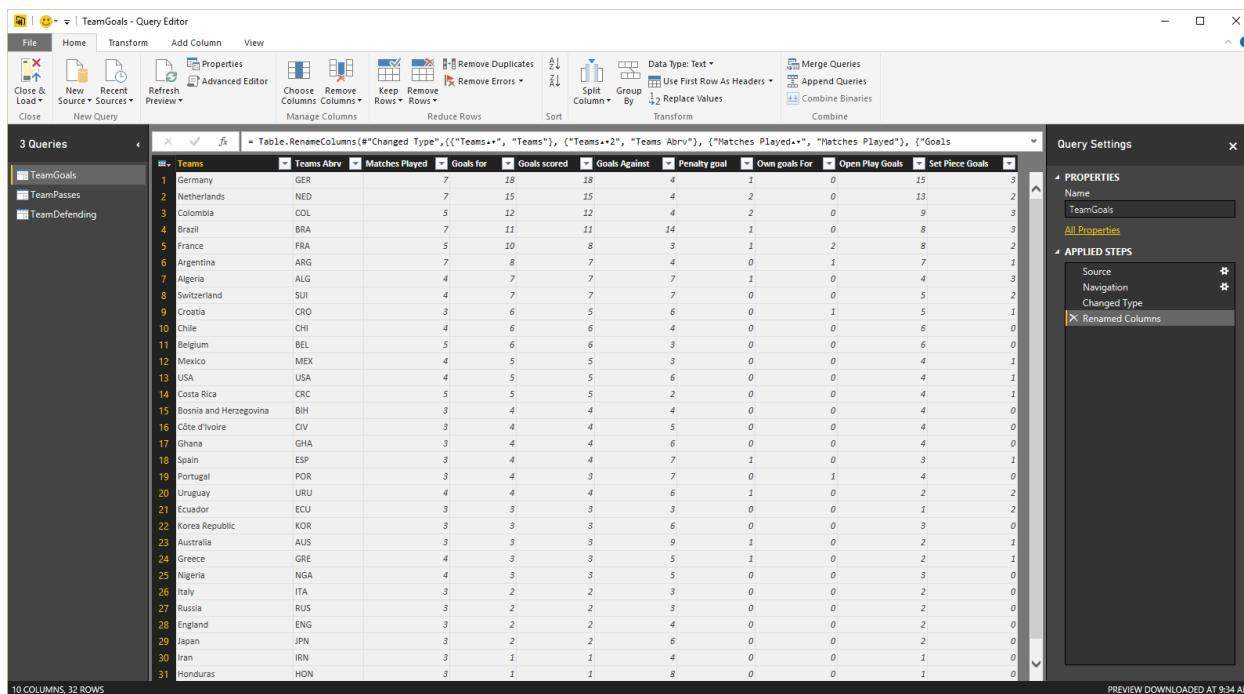
To give you an overall view of what you would expect to see in each component I've put few explanation for each component here. There will be detailed description for all components later on in future chapters.

Power Query

Power Query is data transformation and mash up engine. Power Query can be downloaded as an add-in for Excel or be used as part of Power BI Desktop. With Power Query you can extract data from many different data sources. You can read data from databases such as SQL Server, Oracle, MySQL, DB2, and many other databases. You can fetch data from files such as CSV, Text, Excel. You can even loop through a folder. You can use Microsoft Exchange, Outlook, Azure.... as source. You can connect to Facebook as source and many other applications. You can use online search or use a web address as the source to fetch the data from that web page. Power Query gives you a graphical

user interface to transform data as you need, adding columns, changing types, transformations for date and time, text, and many other operations are available. Power Query can load the result set into Excel or into Power Pivot model.

Power Query also uses a powerful formula language as code behind called M. M is much more powerful than the GUI built for it. There are many functionality in M that cannot be accessed through graphical user interface. I would write deeply about Power Query and M in future chapters so you can confidently write any code and apply complex transformations to the data easily. screenshot below is a view of Power Query editor and some of it's transformations.

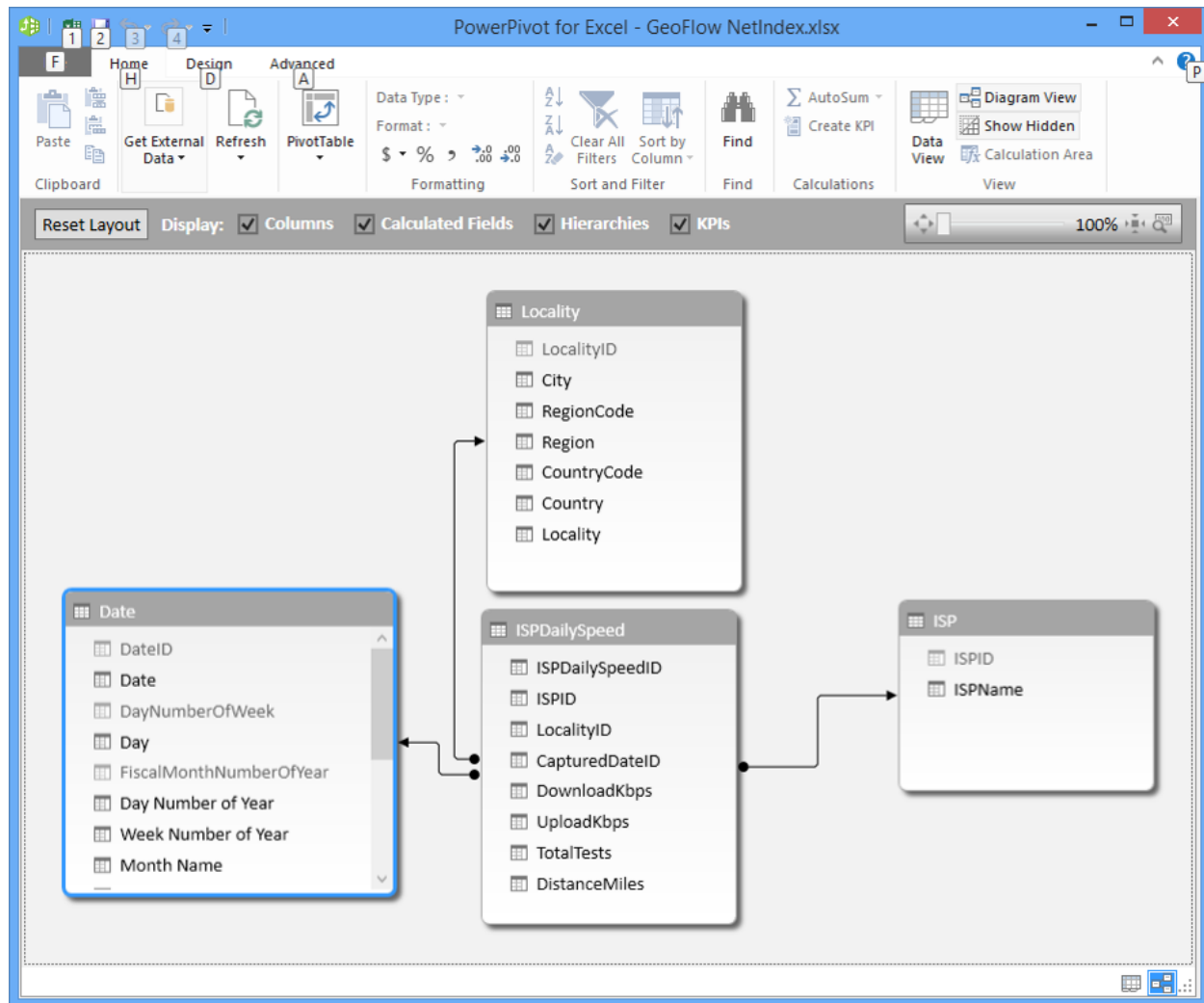


The screenshot shows the Power Query Editor interface. The main area displays a table with 10 columns and 32 rows of data. The columns are: #, Teams, Teams Abbr, Matches Played, Goals for, Goals scored, Goals Against, Penalty goal, Own goals For, Open Play Goals, and Set Piece Goals. The data lists various football teams and their performance metrics. The right sidebar shows the 'Query Settings' pane with 'TeamGoals' selected. The bottom status bar indicates '10 COLUMNS, 32 ROWS' and 'PREVIEW DOWNLOADED AT 9:34 AM'.

#	Teams	Teams Abbr	Matches Played	Goals for	Goals scored	Goals Against	Penalty goal	Own goals For	Open Play Goals	Set Piece Goals
1	Germany	GER	7	18	18	4	1	0	15	3
2	Netherlands	NED	7	15	15	4	2	0	13	2
3	Colombia	COL	5	12	12	4	2	0	9	3
4	Brazil	BRA	7	11	11	14	1	0	8	3
5	France	FRA	5	10	8	3	1	2	8	2
6	Argentina	ARG	7	8	7	4	0	1	7	1
7	Algeria	ALG	4	7	7	7	1	0	4	3
8	Switzerland	SUI	4	7	7	7	0	0	5	2
9	Croatia	CRO	3	6	5	6	0	1	5	1
10	Chile	CHI	4	6	6	4	0	0	6	0
11	Belgium	BEL	5	6	6	3	0	0	6	0
12	Mexico	MEX	4	5	5	3	0	0	4	1
13	USA	USA	4	5	5	6	0	0	4	1
14	Costa Rica	CRC	5	5	5	2	0	0	4	1
15	Bosnia and Herzegovina	BIH	3	4	4	4	0	0	4	0
16	Côte d'Ivoire	CIV	3	4	4	5	0	0	4	0
17	Ghana	GHA	3	4	4	6	0	0	4	0
18	Spain	ESP	3	4	4	7	1	0	3	1
19	Portugal	POR	3	4	3	7	0	1	4	0
20	Uruguay	URU	4	4	4	6	1	0	2	2
21	Ecuador	ECU	3	3	3	3	0	0	1	2
22	Korea Republic	KOR	3	3	3	6	0	0	3	0
23	Australia	AUS	3	3	3	9	1	0	2	1
24	Greece	GRE	4	3	3	5	1	0	2	1
25	Nigeria	NGA	4	3	3	5	0	0	3	0
26	Italy	ITA	3	2	2	3	0	0	2	0
27	Russia	RUS	3	2	2	3	0	0	2	0
28	England	ENG	3	2	2	4	0	0	2	0
29	Japan	JPN	3	2	2	6	0	0	2	0
30	Iran	IRN	3	1	1	4	0	0	1	0
31	Honduras	HON	3	1	1	8	0	0	1	0

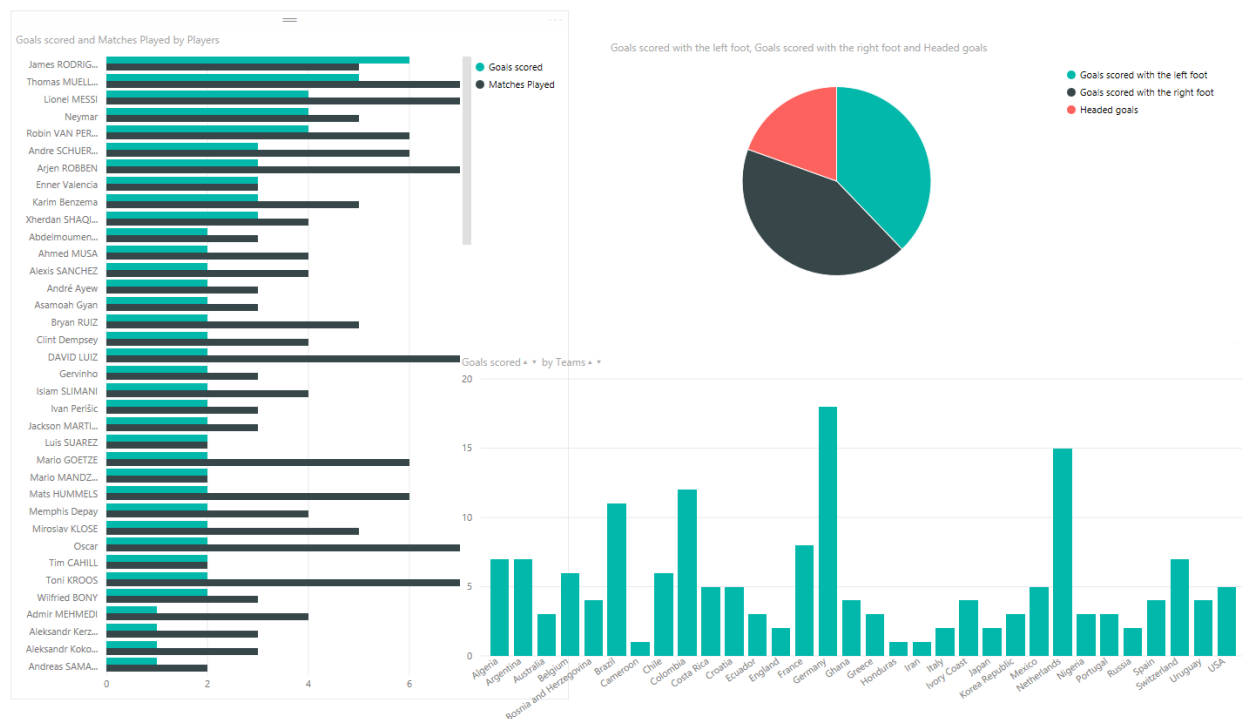
Power Pivot

Power Pivot is data modelling engine which works on xVelocity In-Memory based tabular engine. The In-Memory engine gives Power Pivot super fast response time and the modelling engine would provide you a great place to build your star schema, calculated measures and columns, build relationships through entities and so on. Power Pivot uses Data Analysis eXpression language (DAX) for building measures and calculated columns. DAX is a powerful functional language, and there are heaps of functions for that in the library. We will go through details of Power Pivot modelling and DAX in future chapters. Screenshot below shows the relationship diagram of Power Pivot



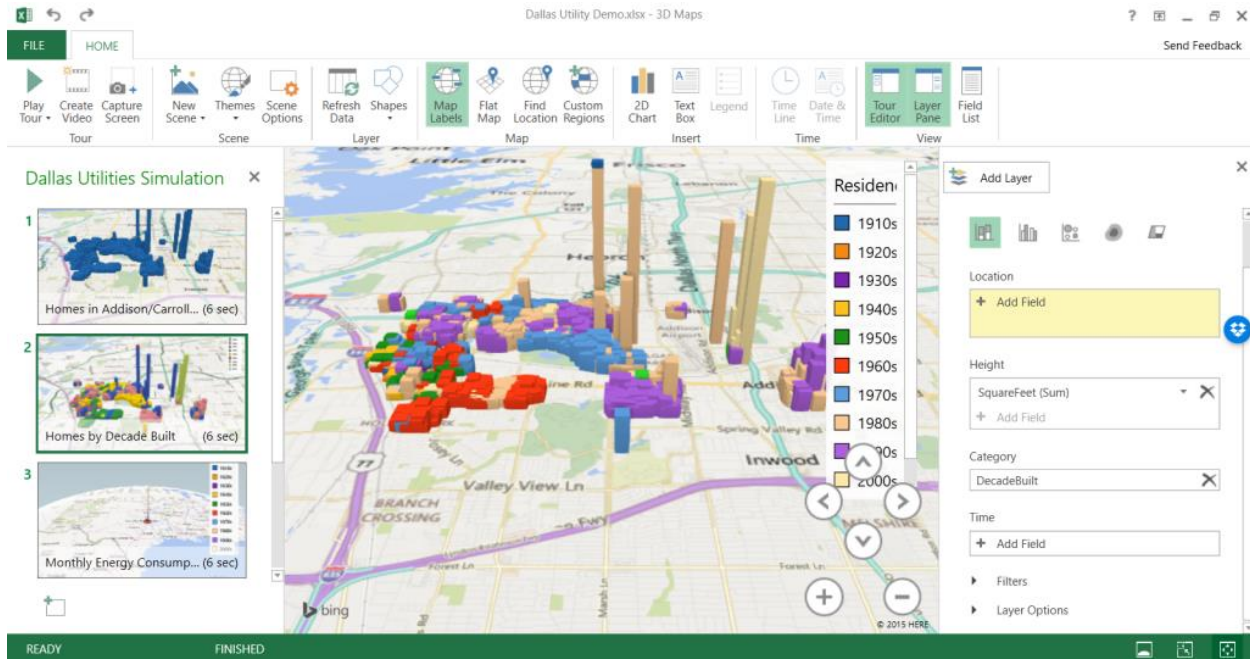
Power View

The main data visualization component of Power BI is Power View. Power View is an interactive data visualization that can connect to data sources and fetch the metadata to be used for data analysis. Power View has many charts for visualization in its list. Power View gives you ability to filter data for each data visualization element or for the entire report. You can use slicers for better slicing and dicing the data. Power View reports are interactive, user can highlight part of the data and different elements in Power View talk with each other. There are many configurations in Power View visualization that I will explain fully in future chapters.



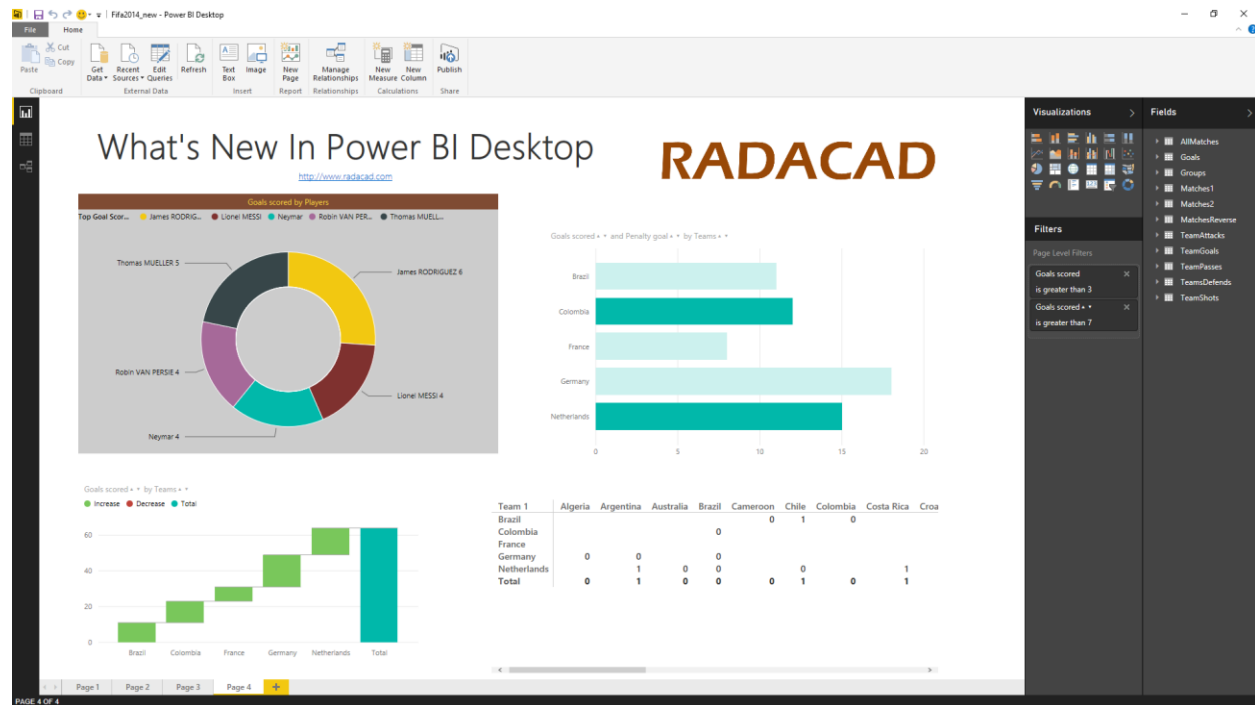
Power Map

Power Map is for visualizing Geo-spatial information in 3D mode. When visualization renders in 3D mode it will give you another dimension in the visualization. You can visualize a measure as height of a column in 3D, and another measure as heatmap view. You can highlight data based on the Geo-graphical location such as country, city, state, and street address. Power Map works with Bing maps to get best visualization based on Geo-graphical either latitude and longitude or country, state, city, and street address information. Power Map is an add-in for Excel 2013, and embedded in Excel 2016.



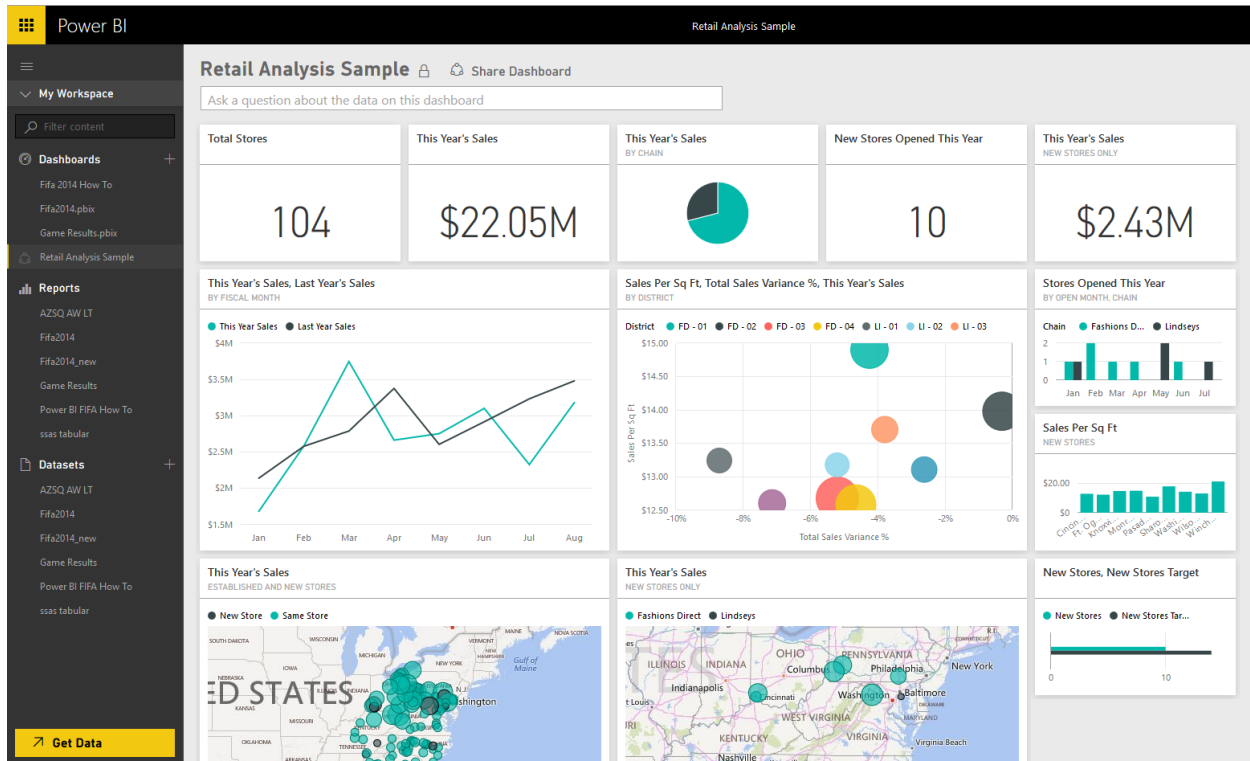
Power BI Desktop

Power BI Desktop is the newest component in Power BI suite. Power BI Desktop is a holistic development tool for Power Query, Power Pivot and Power View. With Power BI Desktop you will have everything under a same solution, and it is easier to develop BI and data analysis experience with that. Power BI Desktop updates frequently and regularly. This product has been in preview mode for a period of time with name of Power BI Designer. There are so much great things about Power BI Desktop that cannot fit in a small paragraph here, you'll read about this tool in future chapters. because of great features of this product I'll write the a section "Power BI Hello World" with a demo of this product. You can have a better view of [newest features of Power BI Desktop here in this blog post](#). screenshot below shows a view of this tool;



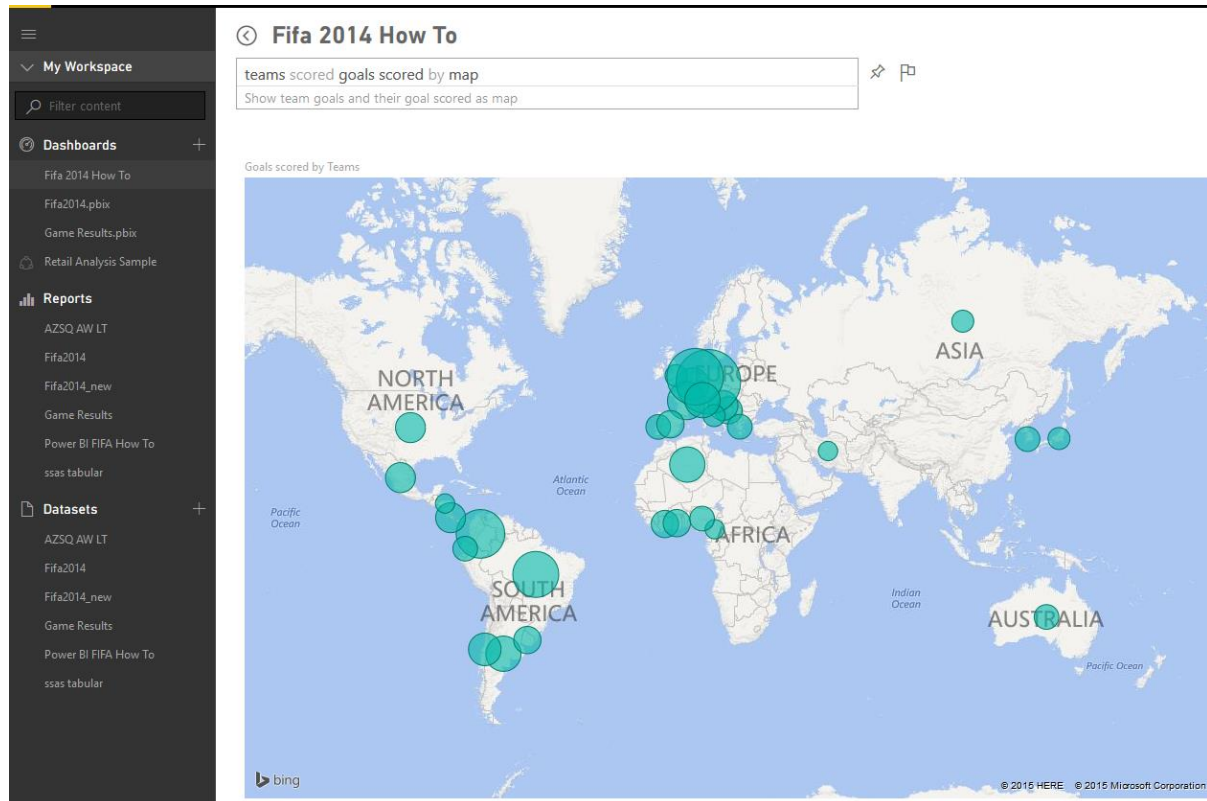
Power BI Website

Power BI solution can be published to PowerBI website. In Power BI website the data source can be scheduled to refresh (depends on the source and is it supporting for schedule data refresh or not). Dashboards can be created for the report, and it can be shared with others. Power BI website even gives you the ability to slice and dice the data online without requiring any other tools, just a simple web browser. You can built report and visualizations directly on Power BI site as well. screenshot below shows a view of Power BI site and dashboards built there;



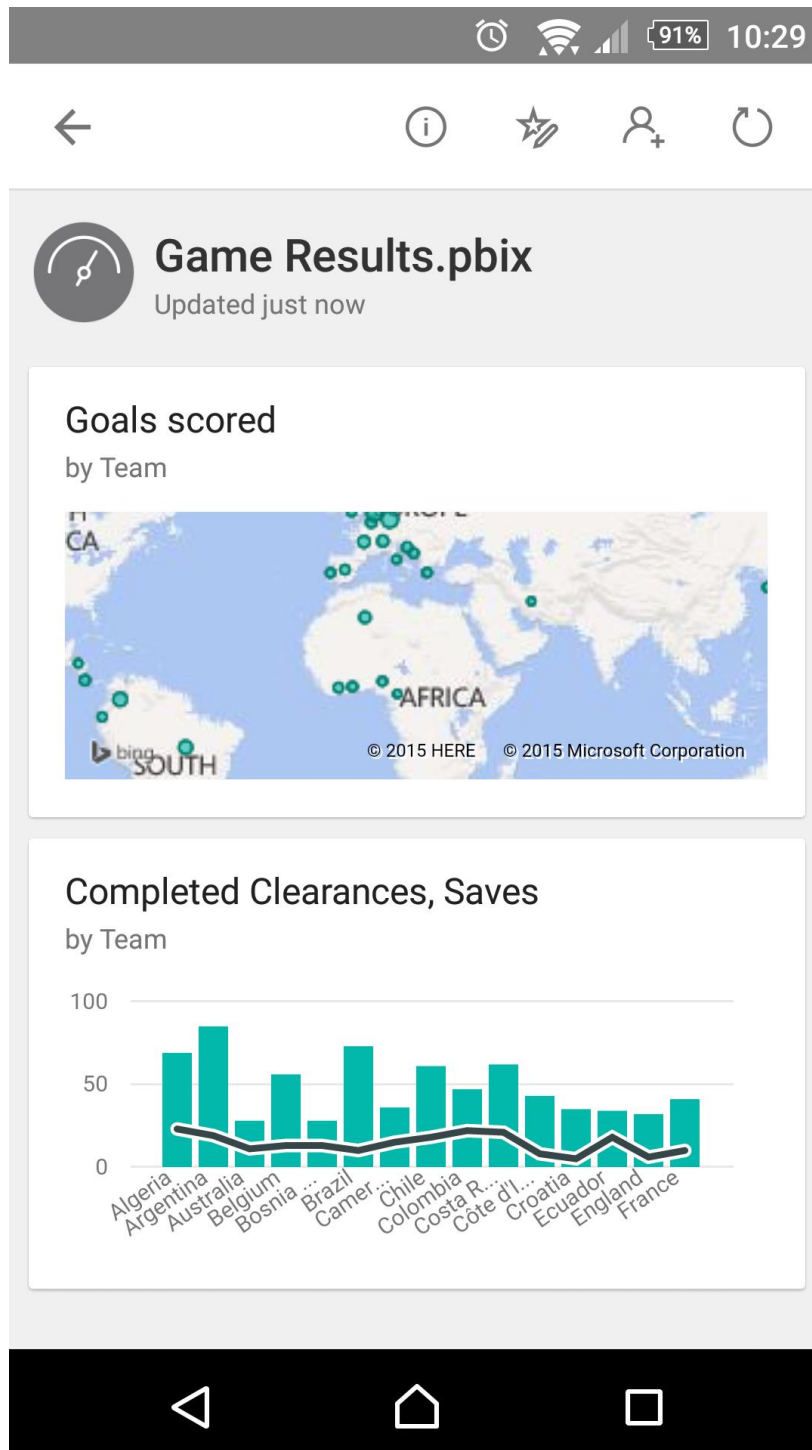
Power Q&A

Power Q&A is a natural language engine for questions and answers to your data model. Once you've built your data model and deployed that into Power BI website, then you or your users can ask questions and get answers easily. There are some tips and tricks about how to build your data model so it can answer questions in the best way which will be covered in future chapters. Power Q&A works with Power View for the data visualizations. So users can simply ask questions such as: Number of Customers by Country, and Power Q&A will answer their question in a map view with numbers as bubbles, Fantastic, isn't it?



Power BI Mobile Apps

There are mobile apps for three main mobile OS providers: Android, Apple, and Windows Phone. These apps give you an interactive view of dashboards and reports in the Power BI site; you can share them even from a mobile app. You can highlight part of the report, write a note on it, and share it with others.



Power BI Pricing

Power BI provide these premium services for free! You can create your account in PowerBI.com website just now for free. Many components of Power BI can be used individually for free as well. you can download and install Power BI Desktop, Power

Query add-in, Power Pivot add-in, Power View add-in, and Power Map add-in all for free! There are some features of these products that reserved for paid version however, such as Power BI Pro which gives you some more features of the product. If you want to learn more about pricing of the Power BI I encourage you to [read this page](#). However create your free account today and give it a try, it won't cost you anything except your precious time, which I'd say definitely worth it.

Preparation

To follow examples of this book download the latest version of Power BI Desktop from [here](#). Most of examples will be demoed through this product, however there might be some exceptions, which I will mention at the beginning of the specific section if you need another tool to download. for the data source for some demos I'll use [AdventureWorks database examples](#), it might be the database itself or the tabular model or other versions and shapes of that, I'll write more information about the requirement for running demo at the beginning of each demo. Files of demos will be shared, either *.pbix files (Power BI Desktop files) or Excel files for you as a reference. If you have any questions just use the comment section below each post.

If you want to have a clue about what to expect in this book [read the table of content here](#).

What to Expect in Next Section?

In summary, you've read about what Power BI is, and what are Power BI components. You've learned that Power BI is cloud based data analysis tool that can be used by data analysts, business analysts and power users because it is easy to use, however it is so much powerful that can be used to answer complex BI requirements. In next section I'll explain how to use Power BI Desktop for a Hello world example. You'll learn some great features of this product through a demo and you will have a better understanding of Power BI tools.

Power BI Desktop; The First Experience

Published Date: August 11, 2015



There are some reasons that I chose Power BI Desktop for the very first demo of [this book](#). Power BI Desktop has three components all in one (Power Query, Power Pivot, and Power View), it can upload directly to PowerBI website, and it is a [recently released product with many shining features](#). In this section I want to get you to start working with Power BI Desktop, you will get familiar with Power BI Desktop editor, and you'll see main panes of that. This would be the first experience with Power BI Desktop. If you want to know [What is Power BI read the first section of book here](#).

Power BI Desktop

Power BI Desktop is the new development editor released 24th of July 2015 as general availability for Power BI. Power BI Desktop can connect to many data sources, transform the data, load it into a data model with relationship design, and finally visualize it in a report style. Power BI Desktop apply all of these actions on a *.pbix file, and it can publish the file directly to Power BI website where users can see the report and interact with it from web browser.

Power BI Desktop as mentioned above is an editor for three components; Power Query, Power Pivot, and Power View. Power Query connects to data sources and mash up the data with a formula language, the result set of Power Query will be loaded into a tabular model which is Power Pivot. Power Pivot can set the relationship and allow you to create measures and calculated columns and set the data model as you want. Then Power View connects to the model and visualize the data with different charts and visualization elements.



Power BI Desktop has everything in one editor, and this makes it an easy to use tool, but don't be fool of this easy interface. Power BI Desktop uses three powerful components and two of those components give you extensive features (I mean Power Query and Power Pivot). You can solve very complex challenges with Power BI Desktop only because of it's underlying components. To be a professional and expert in Power BI, you have to be an expert in Power Query (M language), and Power Pivot (DAX expressions). I'll cover deep detailed discussion about these products in future chapters, but for now, let's look at Power BI Desktop with a very easy example.

Prerequisites

Power BI Desktop:

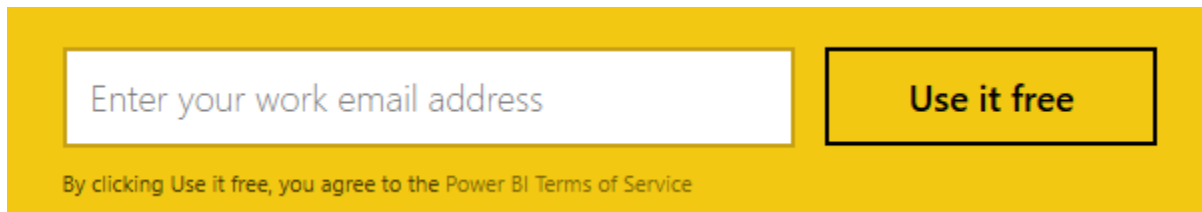
Download Power BI Desktop from [this link](#). You can also install 64 or 32 bit options if you go to advanced download options.

Sample Database:

For this example I used the old Pubs database (just to create something different than AdventureWorks samples). you can download it [here](#) and install Pubs database.

Power BI Account:

If you don't have a Power BI account, simply go to Power BI website, and enter your company mail address to get a free account. Note that you cannot use public email accounts such as gmail and yahoo here

A yellow rectangular form with a white input field on the left containing the placeholder text "Enter your work email address". To the right of the input field is a black-outlined button with the text "Use it free". Below the input field, in smaller text, it says "By clicking Use it free, you agree to the Power BI Terms of Service".

Enter your work email address

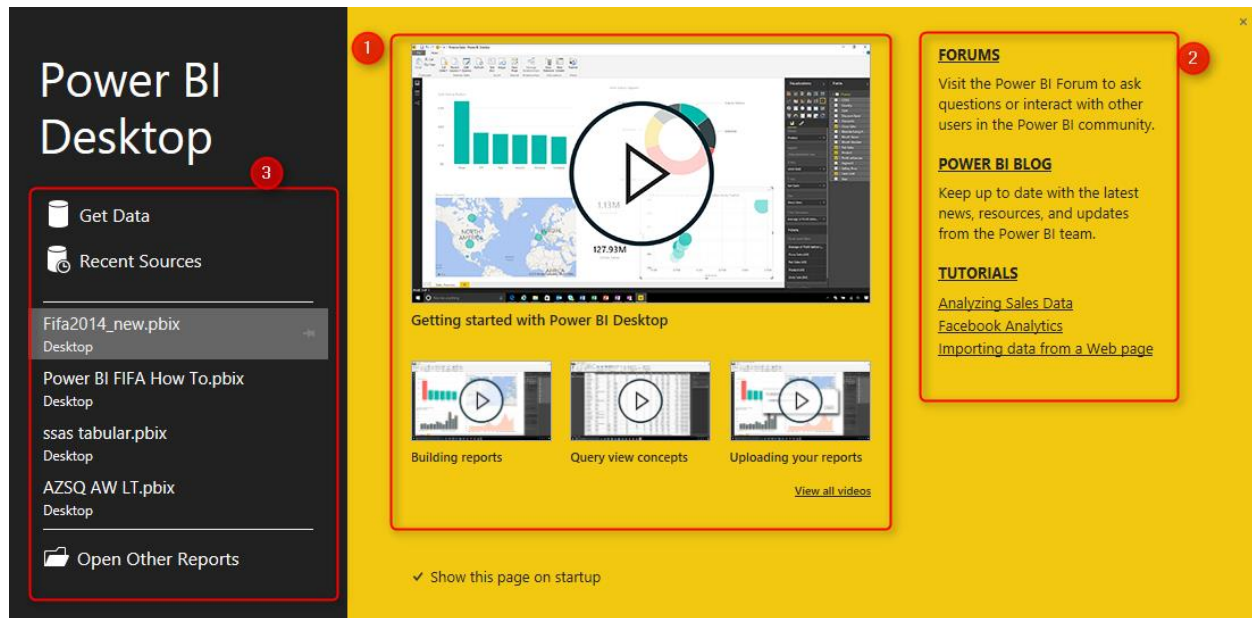
Use it free

By clicking Use it free, you agree to the Power BI Terms of Service

Startup Screen

Power BI Desktop like many other Microsoft editor applications has a startup screen. in the startup screen you have some options to go through as below:

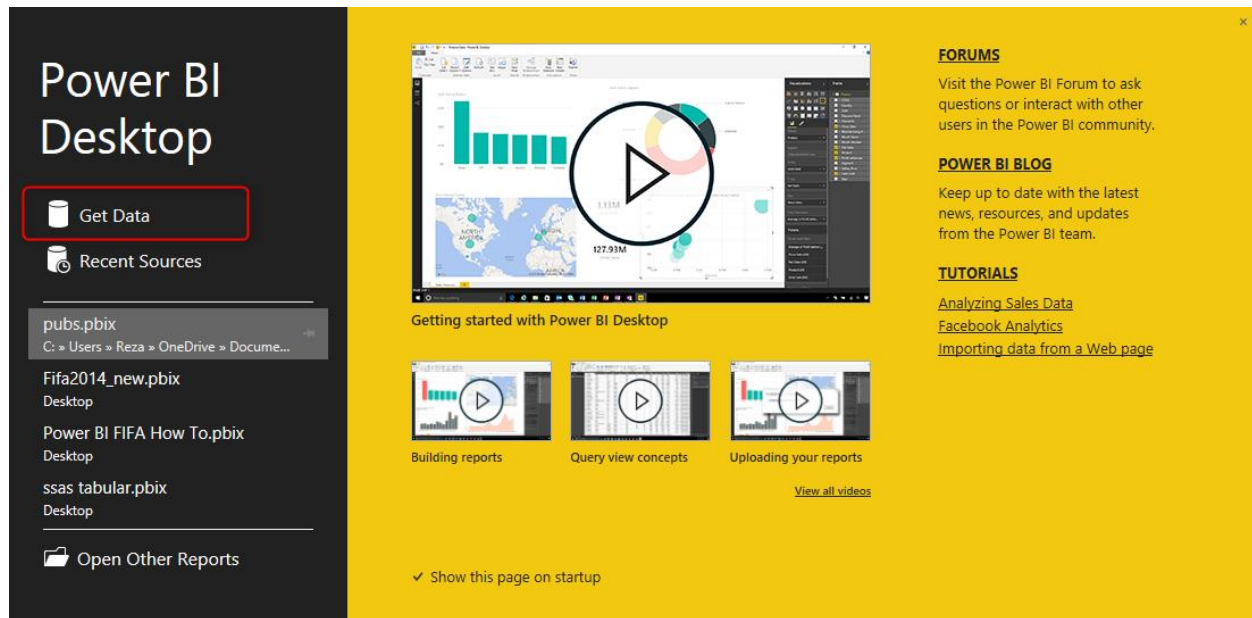
1. you can watch some Power BI Desktop tutorial videos.
2. You can also read latest news about Power BI or some tutorials of Power BI, or you can go to Power BI forums to ask question or search for some answers there.
3. You can open an existing report with Power BI, or start building a new report by getting data. You can see list of latest files that you've worked on as well.



Get Data

Let's start the work with getting data. Pubs is a database of some books published and sold in stores, information about titles, authors, stores and etc are available in this database. I want to keep this example easy so I create a very simple model. for every experience with Power BI Desktop you have to first get data from somewhere. wide range of data sources is supported from files to database, web search, and many other data sources. once you've connected to the data you can transform it (re-shape it) as you want. So in this part of the example you get data from Pubs database, and you'll apply very simple transformation to it.

Open Power BI Desktop. In the startup screen choose Get Data.



Get Data window will open. This window is the graphical interface to connect to many data sources. from the left options choose Database, and in the main section select SQL Server Database. (Remember we are connecting to SQL Server instance that has Pubs database installed)

Get Data

Search

All

File

Database

Azure

Other

Database

SQL Server Database

Access Database

SQL Server Analysis Services Database

Oracle Database

IBM DB2 Database

MySQL Database

PostgreSQL Database

Sybase Database

Teradata Database

Connect

Cancel

Next dialog box asks for server name and database name. Server name is necessary, but you can leave database name blank (blank will load all databases to choose from). You can also write a sql statement here if you want to. because my server for database is my local computer, so I just enter my computer name as below

SQL Server Database

Import data from a SQL Server database.

Server

reza-vaio

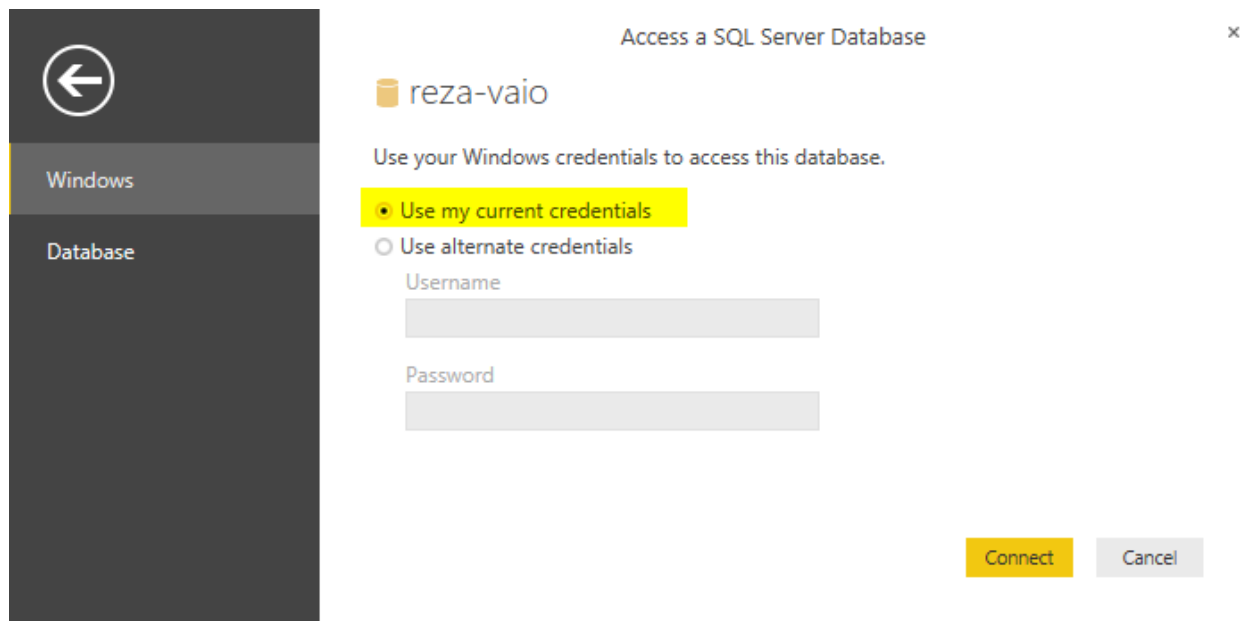
Database (optional)

SQL statement (optional)

OK

Cancel

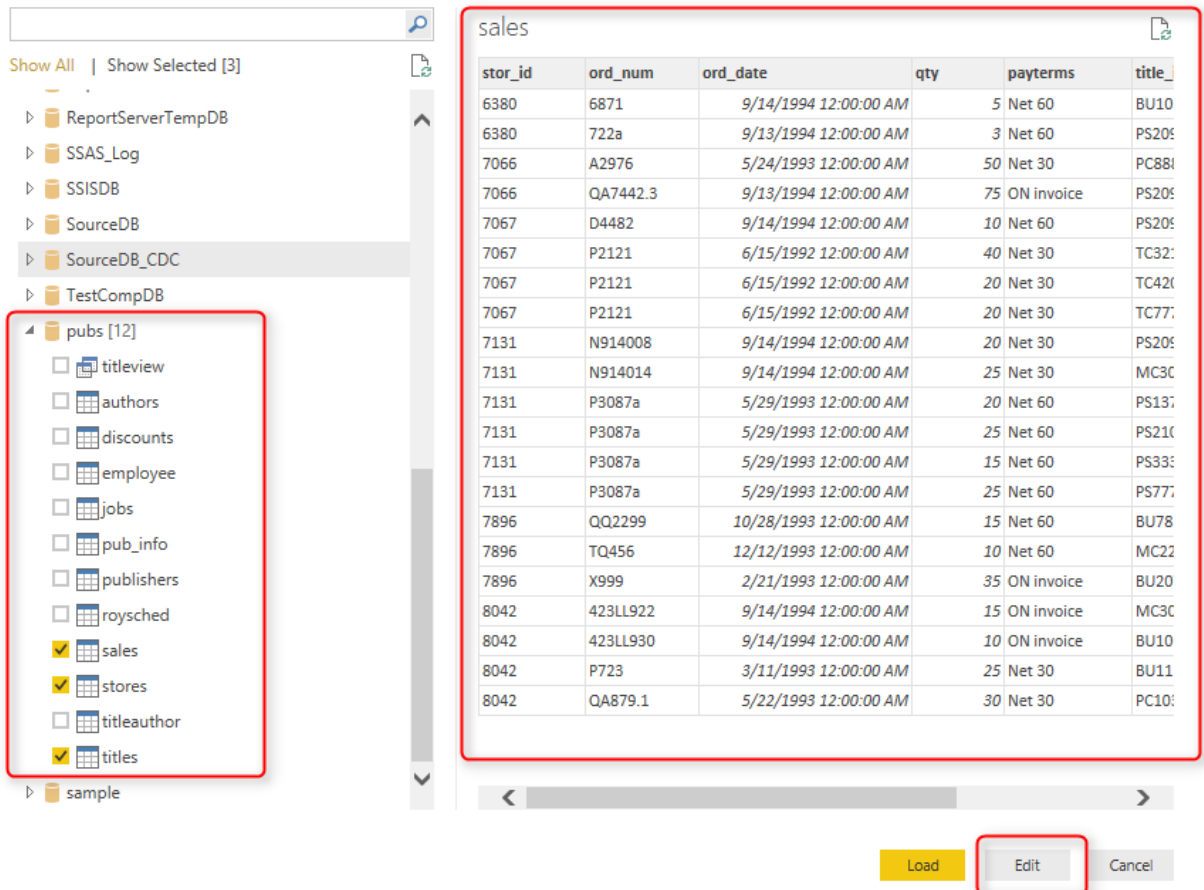
You can set up credential in the next window. I use my current credentials, but note that there are other options such as entering an alternate credential or using database (SQL Server authorization)



The dialog box is titled "Access a SQL Server Database". It features a sidebar on the left with a back arrow icon, a "Windows" button (highlighted in yellow), and a "Database" button. The main area displays the server name "reza-vaio" with a database icon. Below this, it says "Use your Windows credentials to access this database." and provides two radio button options: "Use my current credentials" (selected and highlighted in yellow) and "Use alternate credentials". Under the "Use alternate credentials" option, there are input fields for "Username" and "Password". At the bottom right, there are "Connect" and "Cancel" buttons.

you might get a message about encryption support, just click OK on that, we will cover that later. Because we've entered blank in the database name, now we see all databases, expand Pubs, and then select these tables: titles, stores, and sales.

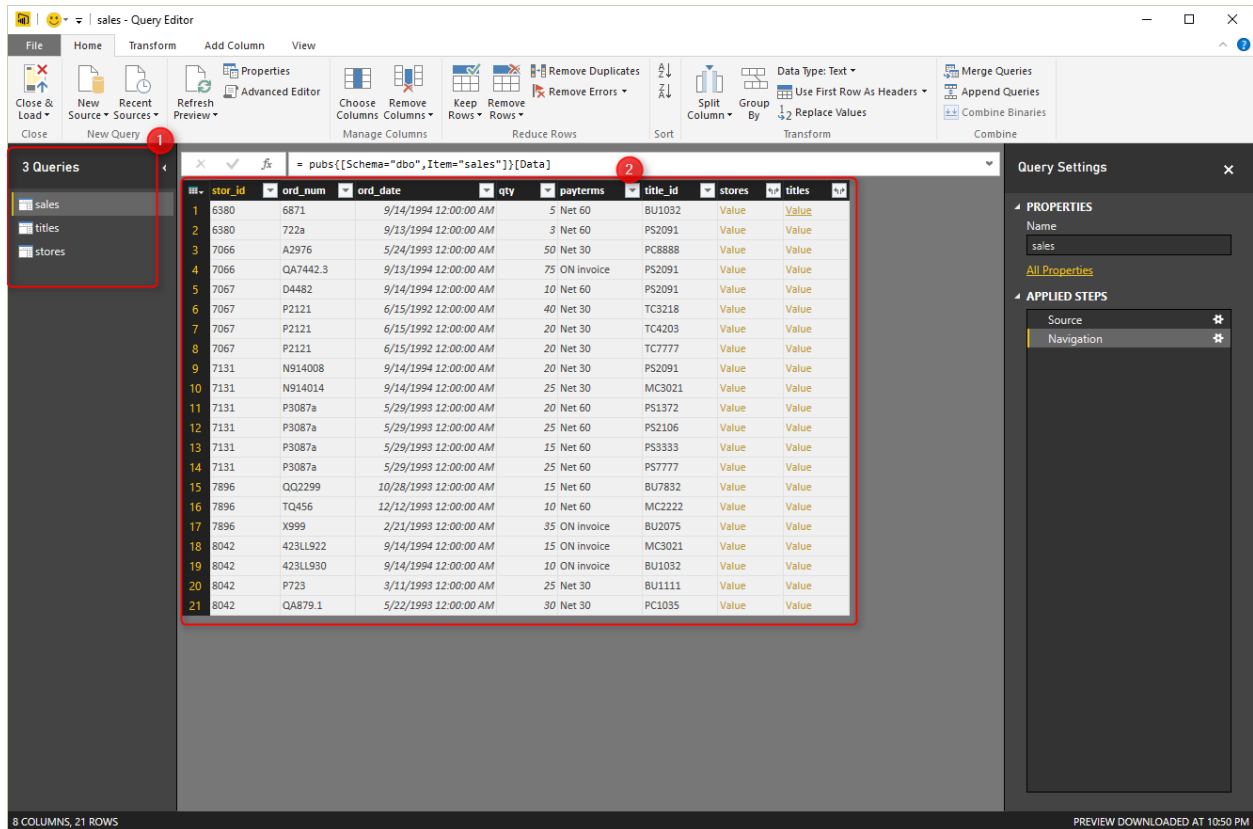
Navigator



The screenshot shows the Power BI Navigator window. On the left, the 'pubs' database is expanded, showing a list of tables. The 'sales' table is selected, indicated by a red box. The main area displays a sample data row from the 'sales' table, also highlighted with a red box. Below the table, there are 'Load', 'Edit', and 'Cancel' buttons. The 'Edit' button is highlighted with a red box.

stor_id	ord_num	ord_date	qty	payterms	title
6380	6871	9/14/1994 12:00:00 AM	5	Net 60	BU10
6380	722a	9/13/1994 12:00:00 AM	3	Net 60	PS20
7066	A2976	5/24/1993 12:00:00 AM	50	Net 30	PC88
7066	QA7442.3	9/13/1994 12:00:00 AM	75	ON invoice	PS20
7067	D4482	9/14/1994 12:00:00 AM	10	Net 60	PS20
7067	P2121	6/15/1992 12:00:00 AM	40	Net 30	TC32
7067	P2121	6/15/1992 12:00:00 AM	20	Net 30	TC42
7067	P2121	6/15/1992 12:00:00 AM	20	Net 30	TC77
7131	N914008	9/14/1994 12:00:00 AM	20	Net 30	PS20
7131	N914014	9/14/1994 12:00:00 AM	25	Net 30	MC30
7131	P3087a	5/29/1993 12:00:00 AM	20	Net 60	PS13
7131	P3087a	5/29/1993 12:00:00 AM	25	Net 60	PS21
7131	P3087a	5/29/1993 12:00:00 AM	15	Net 60	PS33
7131	P3087a	5/29/1993 12:00:00 AM	25	Net 60	PS77
7896	QQ2299	10/28/1993 12:00:00 AM	15	Net 60	BU78
7896	TQ456	12/12/1993 12:00:00 AM	10	Net 60	MC22
7896	X999	2/21/1993 12:00:00 AM	35	ON invoice	BU20
8042	423LL922	9/14/1994 12:00:00 AM	15	ON invoice	MC30
8042	423LL930	9/14/1994 12:00:00 AM	10	ON invoice	BU10
8042	P723	3/11/1993 12:00:00 AM	25	Net 30	BU11
8042	QA879.1	5/22/1993 12:00:00 AM	30	Net 30	PC10

As you see in above screenshot when you click on tables you'll see a sample data rows of that appearing in the main section. this helps you to have an understanding of what you are looking at. Click on Edit. The Query Editor window will open. This is the main Power Query Window that has all the tables we've selected.



3 Queries

- sales
- titles
- stores

Query Settings

PROPERTIES

Name: sales

APPLIED STEPS

Source

Navigation

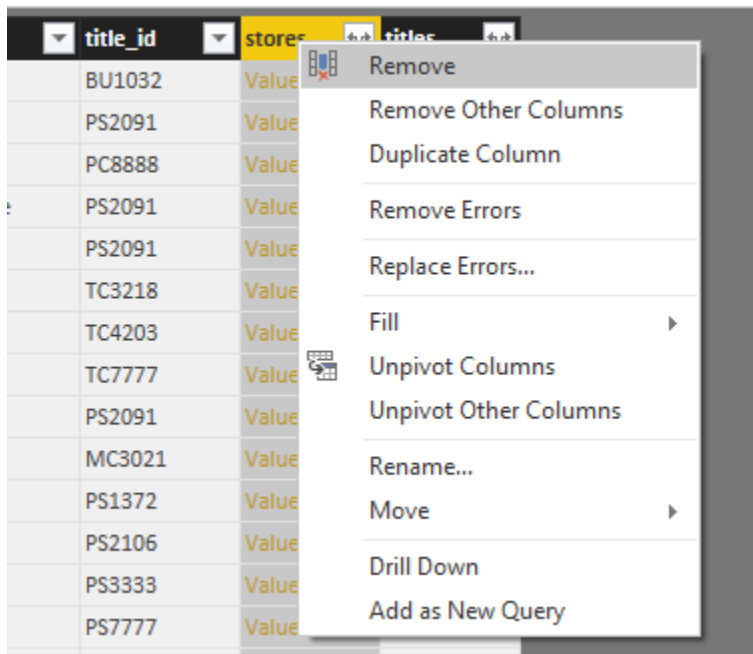
	stor_id	ord_num	ord_date	qty	payterms	title_id	stores	titles	mp
1	6380	6871	9/14/1994 12:00:00 AM	5	Net 60	BU1032	Value	Value	
2	6380	722a	9/13/1994 12:00:00 AM	3	Net 60	PS2091	Value	Value	
3	7066	A2976	5/24/1993 12:00:00 AM	50	Net 30	PC8888	Value	Value	
4	7066	QA7442.3	9/13/1994 12:00:00 AM	75	ON invoice	PS2091	Value	Value	
5	7067	D4482	9/14/1994 12:00:00 AM	10	Net 60	PS2091	Value	Value	
6	7067	P2121	6/15/1992 12:00:00 AM	40	Net 30	TC3218	Value	Value	
7	7067	P2121	6/15/1992 12:00:00 AM	20	Net 30	TC4203	Value	Value	
8	7067	P2121	6/15/1992 12:00:00 AM	20	Net 30	TC7777	Value	Value	
9	7131	N914008	9/14/1994 12:00:00 AM	20	Net 30	PS2091	Value	Value	
10	7131	N914014	9/14/1994 12:00:00 AM	25	Net 30	MC3021	Value	Value	
11	7131	P3087a	5/29/1993 12:00:00 AM	20	Net 60	PS1372	Value	Value	
12	7131	P3087a	5/29/1993 12:00:00 AM	25	Net 60	PS2106	Value	Value	
13	7131	P3087a	5/29/1993 12:00:00 AM	15	Net 60	PS3333	Value	Value	
14	7131	P3087a	5/29/1993 12:00:00 AM	25	Net 60	PS7777	Value	Value	
15	7896	QQ2299	10/28/1993 12:00:00 AM	15	Net 60	BU7832	Value	Value	
16	7896	TQ456	12/12/1993 12:00:00 AM	10	Net 60	MC2222	Value	Value	
17	7896	X999	2/21/1993 12:00:00 AM	35	ON invoice	BU2075	Value	Value	
18	8042	423LL922	9/14/1994 12:00:00 AM	15	ON invoice	MC3021	Value	Value	
19	8042	423LL930	9/14/1994 12:00:00 AM	10	ON invoice	BU1032	Value	Value	
20	8042	P723	3/11/1993 12:00:00 AM	25	Net 30	BU1111	Value	Value	
21	8042	QA879.1	5/22/1993 12:00:00 AM	30	Net 30	PC1035	Value	Value	

8 COLUMNS, 21 ROWS

PREVIEW DOWNLOADED AT 10:50 PM

In Query Editor window in the left pane you can see three tables that we've loaded (numbered as 1). Main pane (numbered as 2) shows selected the data set from table in the left hand side. Right hand side shows the steps applied on the data set. the menu on top contains some transformations that you can apply on the data set.

Select sales table in left pane, now in the main pane you'll see data columns of that table plus two hyperlinked columns (stores, and titles). Power Query is intelligent enough to understand relationship between tables and load them together. That is why stores and titles are here as hyperlinks, because Power Query joined them based on the FK-PK relationship in the database. Let's don't use that intelligence for now, so I'll remove these two columns by right click on them and then selecting Remove.



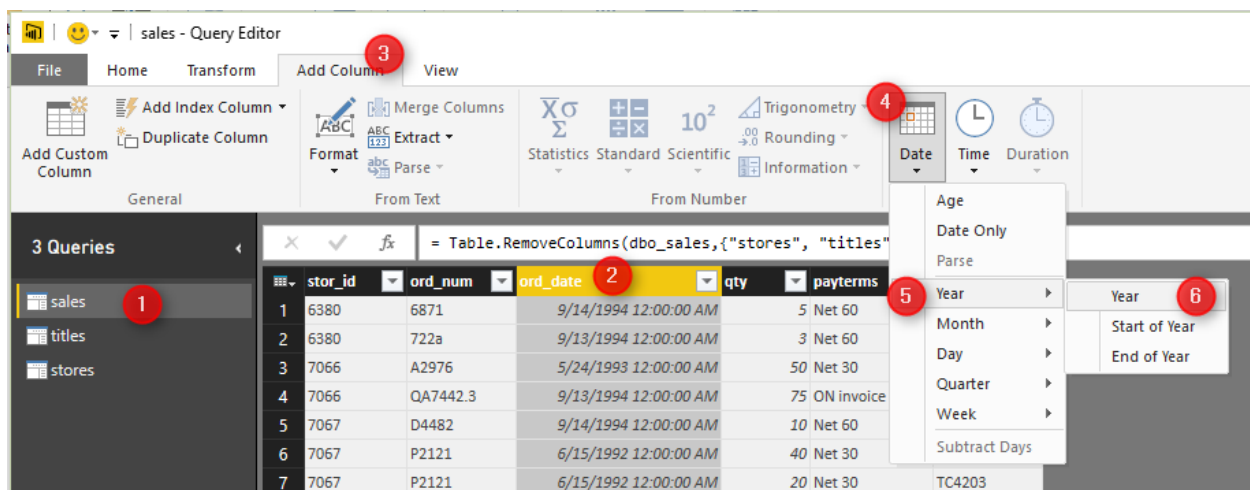
title_id	stores	titles
BU1032	Value	
PS2091	Value	
PC8888	Value	
PS2091	Value	
PS2091	Value	
TC3218	Value	
TC4203	Value	
TC7777	Value	
PS2091	Value	
MC3021	Value	
PS1372	Value	
PS2106	Value	
PS3333	Value	
PS7777	Value	

Context menu options:

- Remove
- Remove Other Columns
- Duplicate Column
- Remove Errors
- Replace Errors...
- Fill
- Unpivot Columns
- Unpivot Other Columns
- Rename...
- Move
- Drill Down
- Add as New Query

Do the same in "titles" table, and remove columns: publishers, roysched, sales, titleauthor. Also from "stores" table remove discounts and sales columns. Now let's make a very small change. I want to keep this example very simple, so regardless of best practices of having date dimension that have columns for year, months, week and etc, I just want to add a year column to the sales table. the year column that I want to add is the Order Year, and I can simply calculate that from the ord_date column which is a datetime value.

select sales query, then click on ord_date column, and then from the main menu click on Add Column. in the Add Column section under Date, choose Year.



The screenshot shows the Power BI Query Editor interface with the following steps highlighted by red circles:

- 1**: Select the 'sales' query in the left-hand pane.
- 2**: Click on the 'ord_date' column in the data table.
- 3**: Click on the 'Add Column' tab in the top ribbon.
- 4**: Click on the 'Date' button in the 'Add Column' section of the ribbon.
- 5**: Click on 'Year' in the dropdown menu that appears.
- 6**: Click on 'Year' in the sub-menu that appears.

The formula bar shows the following DAX formula:

```
Table.RemoveColumns(dbo_sales,{"stores", "titles"}
```

The data table shows the following columns: stor_id, ord_num, ord_date, qty, and payterms.

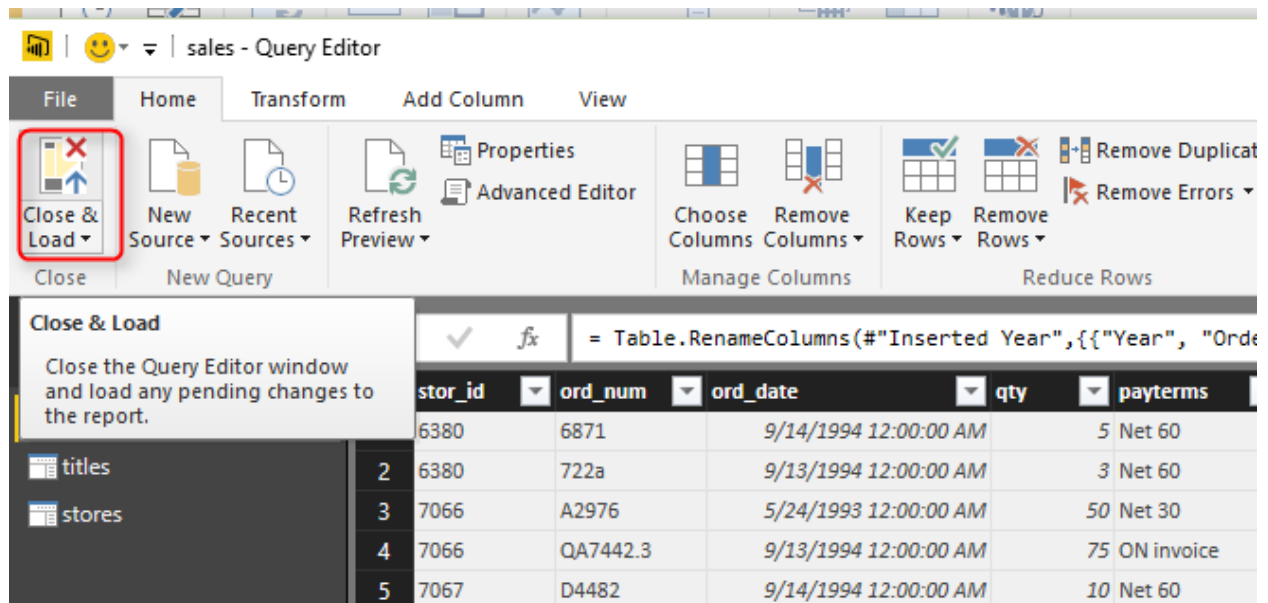
	stor_id	ord_num	ord_date	qty	payterms
1	6380	6871	9/14/1994 12:00:00 AM	5	Net 60
2	6380	722a	9/13/1994 12:00:00 AM	3	Net 60
3	7066	A2976	5/24/1993 12:00:00 AM	50	Net 30
4	7066	QA7442.3	9/13/1994 12:00:00 AM	75	ON invoice
5	7067	D4482	9/14/1994 12:00:00 AM	10	Net 60
6	7067	P2121	6/15/1992 12:00:00 AM	40	Net 30
7	7067	P2121	6/15/1992 12:00:00 AM	20	Net 30

A new column with name of Year will be added to the table, right click on the column and rename it to Order Year. Well done, you've done your first transformation with Power Query. It was so simple, wasn't it?

	stor_id	ord_num	ord_date	qty	payterms	title_id	Order Year
1	6380	6871	9/14/1994 12:00:00 AM	5	Net 60	BU1032	1994
2	6380	722a	9/13/1994 12:00:00 AM	3	Net 60	PS2091	1994
3	7066	A2976	5/24/1993 12:00:00 AM	50	Net 30	PC8888	1993
4	7066	QA7442.3	9/13/1994 12:00:00 AM	75	ON invoice	PS2091	1994
5	7067	D4482	9/14/1994 12:00:00 AM	10	Net 60	PS2091	1994
6	7067	P2121	6/15/1992 12:00:00 AM	40	Net 30	TC3218	1992
7	7067	P2121	6/15/1992 12:00:00 AM	20	Net 30	TC4203	1992
8	7067	P2121	6/15/1992 12:00:00 AM	20	Net 30	TC7777	1992

Modelling Data

Now let's load data into a model, in Query Editor window under Home tab, click on Close & Load button. This will load all the queries with their data into the model.



Close & Load

Close the Query Editor window and load any pending changes to the report.

titles

stores

2

3

4

5

stor_id

ord_num

ord_date

qty

payterms

6380

6871

9/14/1994 12:00:00 AM

5

Net 60

6380

722a

9/13/1994 12:00:00 AM

3

Net 60

7066

A2976

5/24/1993 12:00:00 AM

50

Net 30

7066

QA7442.3

9/13/1994 12:00:00 AM

75

ON invoice

7067

D4482

9/14/1994 12:00:00 AM

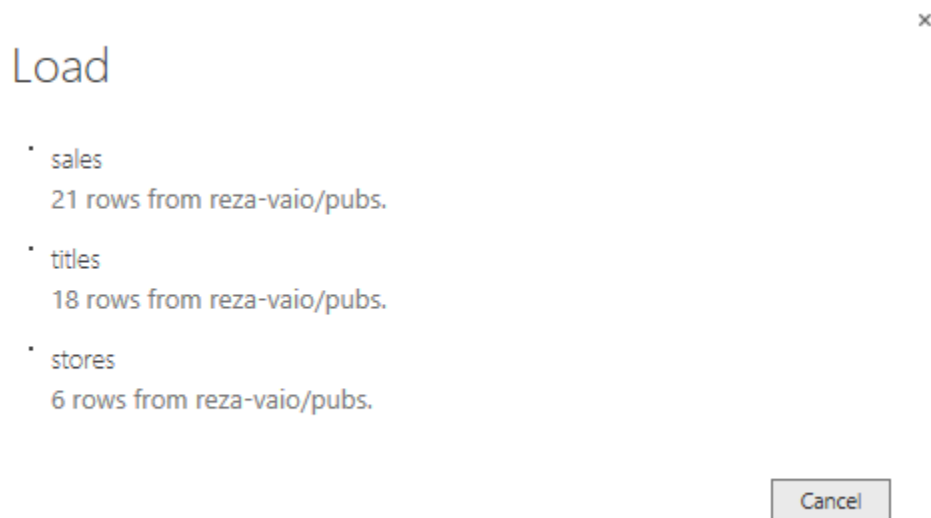
10

Net 60

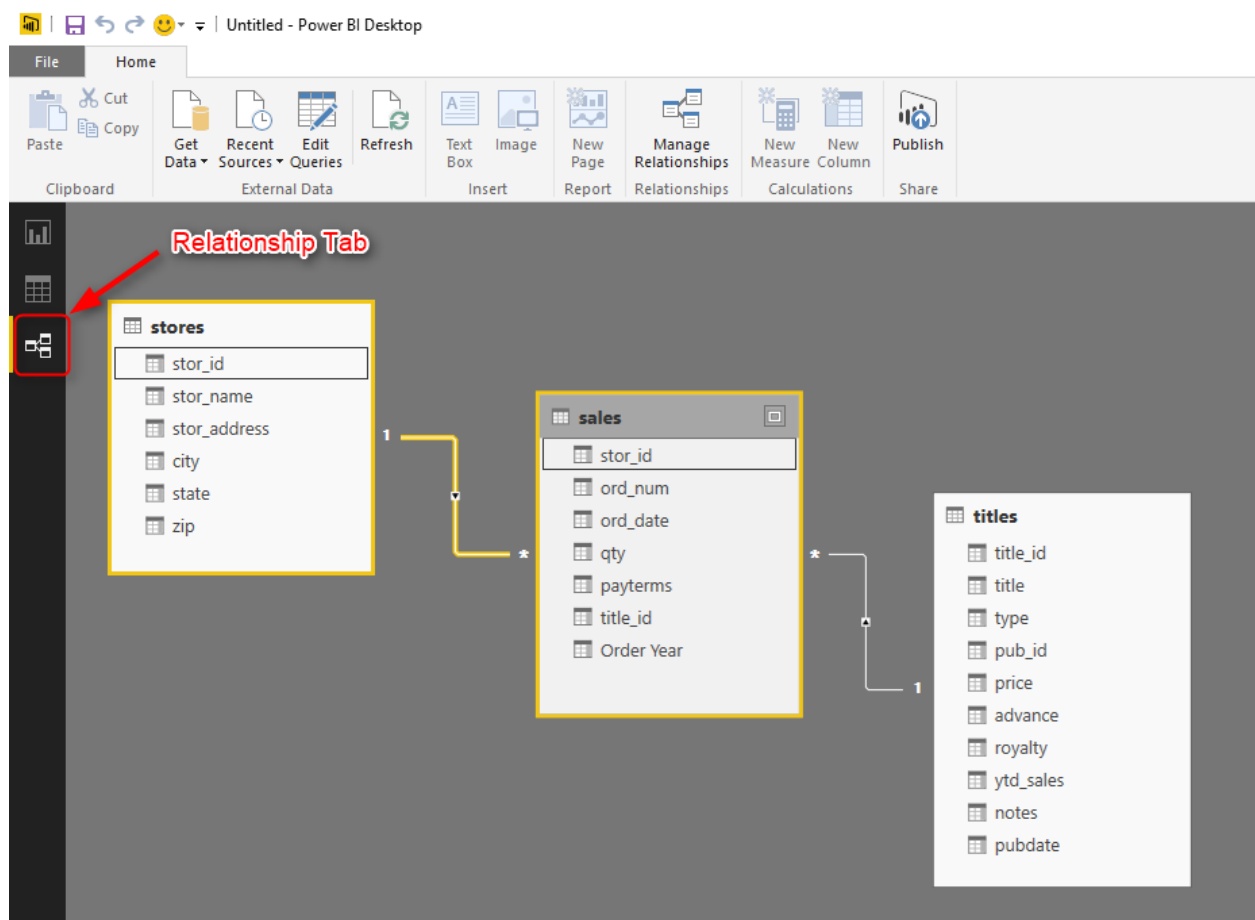
fx

= Table.RenameColumns(#"Inserted Year",{{"Year", "Order Year"}})

You can see in the screenshot below that data is loading into the model

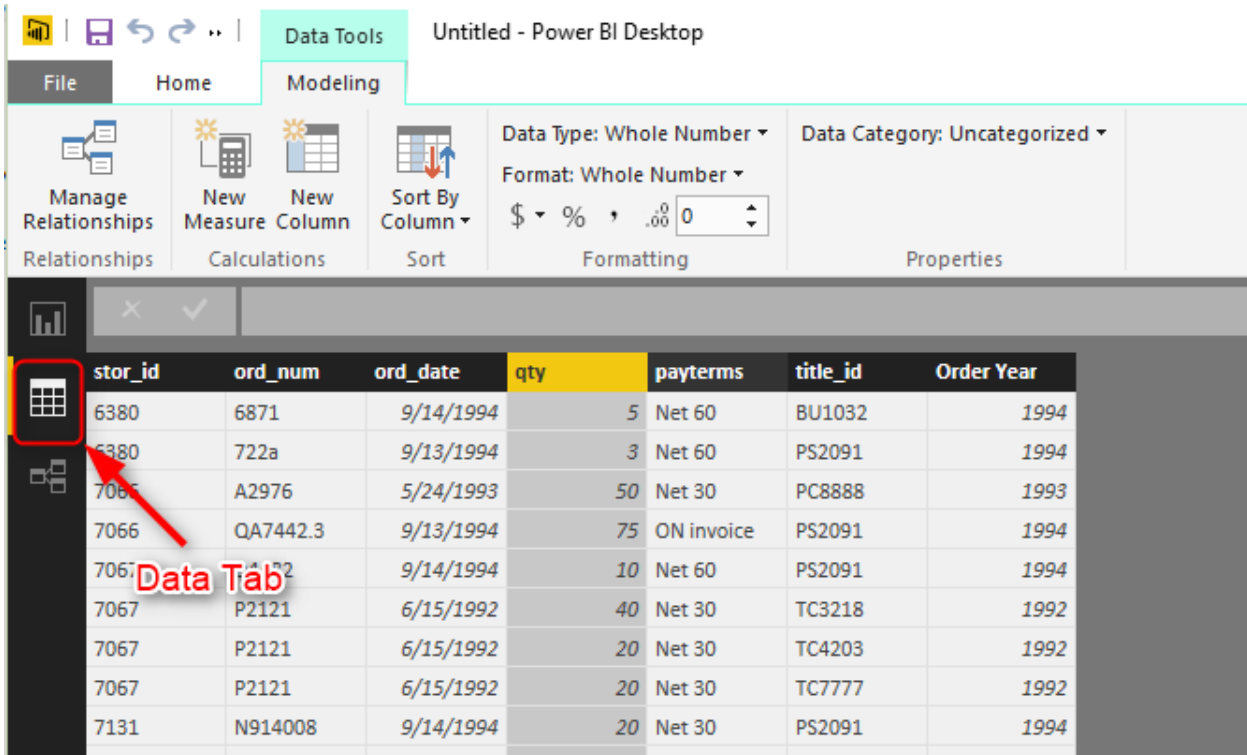


Now you will see the blank Power BI Desktop editor. From the left tabs, click on Relationship (last tab) as below to see the relationship diagram of your model



Now you see the diagram. Power Pivot intelligently created relationship between tables based on their relationship in the pubs database. If you double click on any of relationships you can modify it if you want to. Let's just leave it as is, because relationship is just fine. The change that we want to do in the modelling is just adding a sales amount to sales table. (Note that sales table has the quantity only, and price is in titles table). Sales amount would be quantity sold multiplied by price of the title. Let's do it in the model.

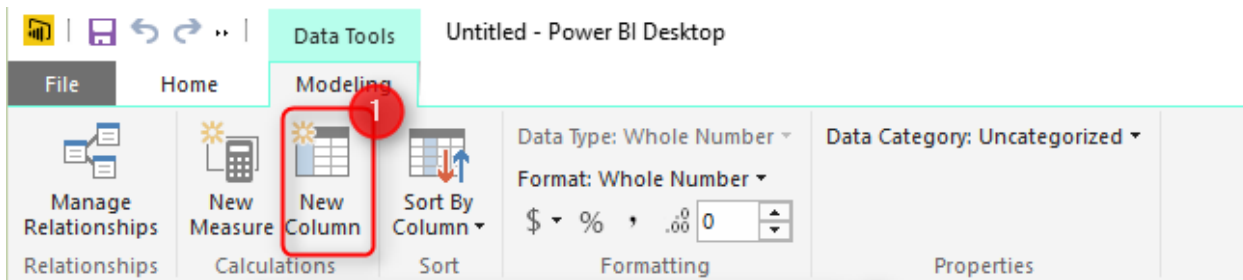
Click on Data Tab to see structure of data in each table. you can see that qty exists in sales table. If you look at titles, you'll see price column there.



stor_id	ord_num	ord_date	qty	payterms	title_id	Order Year
6380	6871	9/14/1994	5	Net 60	BU1032	1994
6380	722a	9/13/1994	3	Net 60	PS2091	1994
7065	A2976	5/24/1993	50	Net 30	PC8888	1993
7066	QA7442.3	9/13/1994	75	ON invoice	PS2091	1994
7067	P2121	9/14/1994	10	Net 60	PS2091	1994
7067	P2121	6/15/1992	40	Net 30	TC3218	1992
7067	P2121	6/15/1992	20	Net 30	TC4203	1992
7067	P2121	6/15/1992	20	Net 30	TC7777	1992
7131	N914008	9/14/1994	20	Net 30	PS2091	1994

Go to sales table (from right pane choose sales), and then click on New Column. In the formula bar type in the formula below

Column = sales[qty]*RELATED(titles[price])



Column = sales[qty]*RELATED(titles[price])

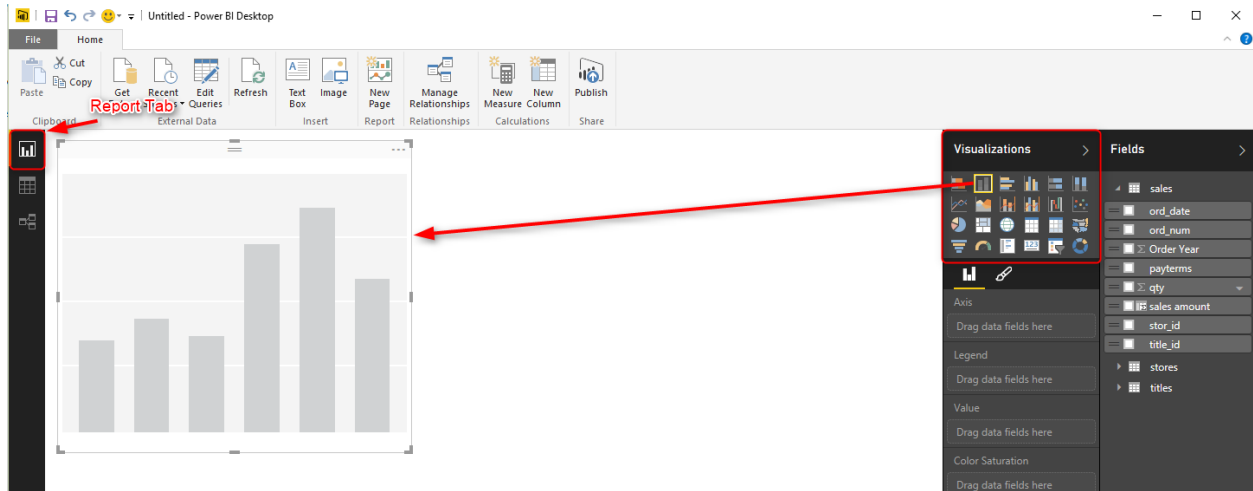
stor_id	ord_num	ord_date	qty	payterms	title_id	Order Year	Column
6380	6871	9/14/1994	5	Net 60	BU1032	1994	
6380	722a	9/13/1994	3	Net 60	PS2091	1994	
7066	A2976	5/24/1993	50	Net 30	PC8888	1993	
7066	QA7442.3	9/13/1994	75	ON invoice	PS2091	1994	
7067	D4482	9/14/1994	10	Net 60	PS2091	1994	
7067	P2121	6/15/1992	40	Net 30	TC3218	1992	
7067	P2121	6/15/1992	20	Net 30	TC4203	1992	
7067	P2121	6/15/1992	20	Net 30	TC7777	1992	
7131	N914008	9/14/1994	20	Net 30	PS2091	1994	
7131	N914014	9/14/1994	25	Net 30	MC3021	1994	
7131	P3087a	5/29/1993	20	Net 60	PS1372	1993	
7131	P3087a	5/29/1993	25	Net 60	PS2106	1993	

then press enter (you'll see that intellisense also helps you to write down the formula). Now you'll see the result in the new column. right click on it, and rename it to sales amount. Congratulations! you've written your first DAX expression here. Related function is a function that brings the relative row to this record based on the relationship of tables. so related(titles[price]) means from table "titles" bring column "price" but only price of that title that is related to this record (based on the relationship of title_id column).

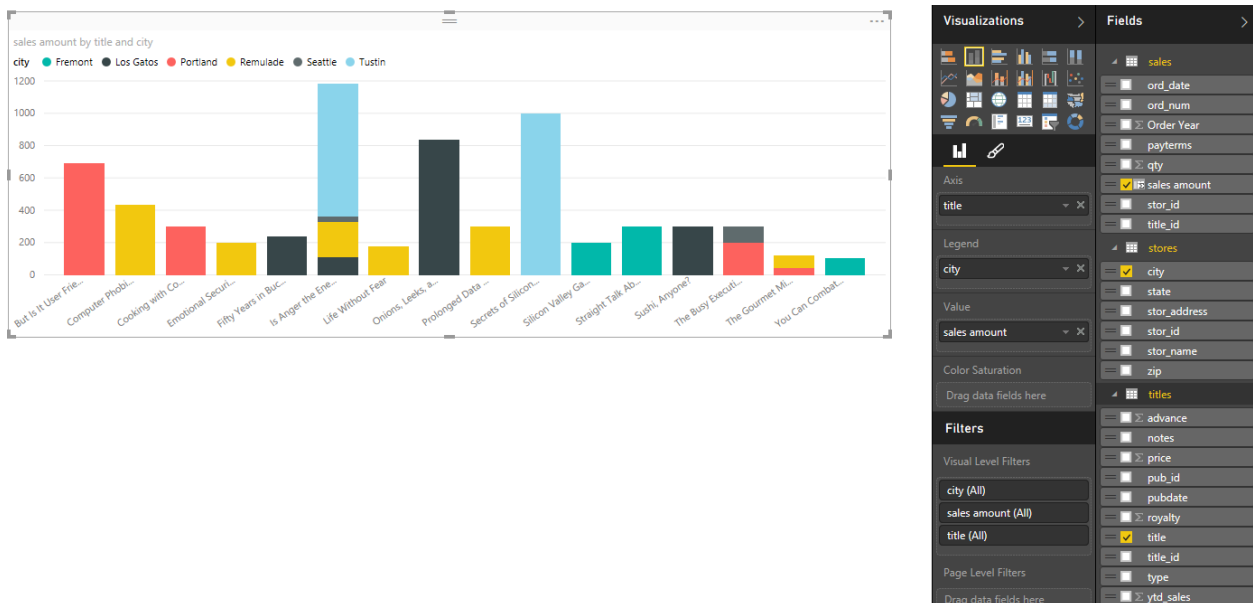
✕	✓	sales amount = sales[qty]*RELATED(titles[price])					
stor_id	ord_num	ord_date	qty	payterms	title_id	Order Year	sales amount
6380	6871	9/14/1994	5	Net 60	BU1032	1994	100
6380	722a	9/13/1994	3	Net 60	PS2091	1994	33
7066	A2976	5/24/1993	50	Net 30	PC8888	1993	1000
7066	QA7442.3	9/13/1994	75	ON invoice	PS2091	1994	821
7067	D4482	9/14/1994	10	Net 60	PS2091	1994	110
7067	P2121	6/15/1992	40	Net 30	TC3218	1992	838
7067	P2121	6/15/1992	20	Net 30	TC4203	1992	239

Visualization

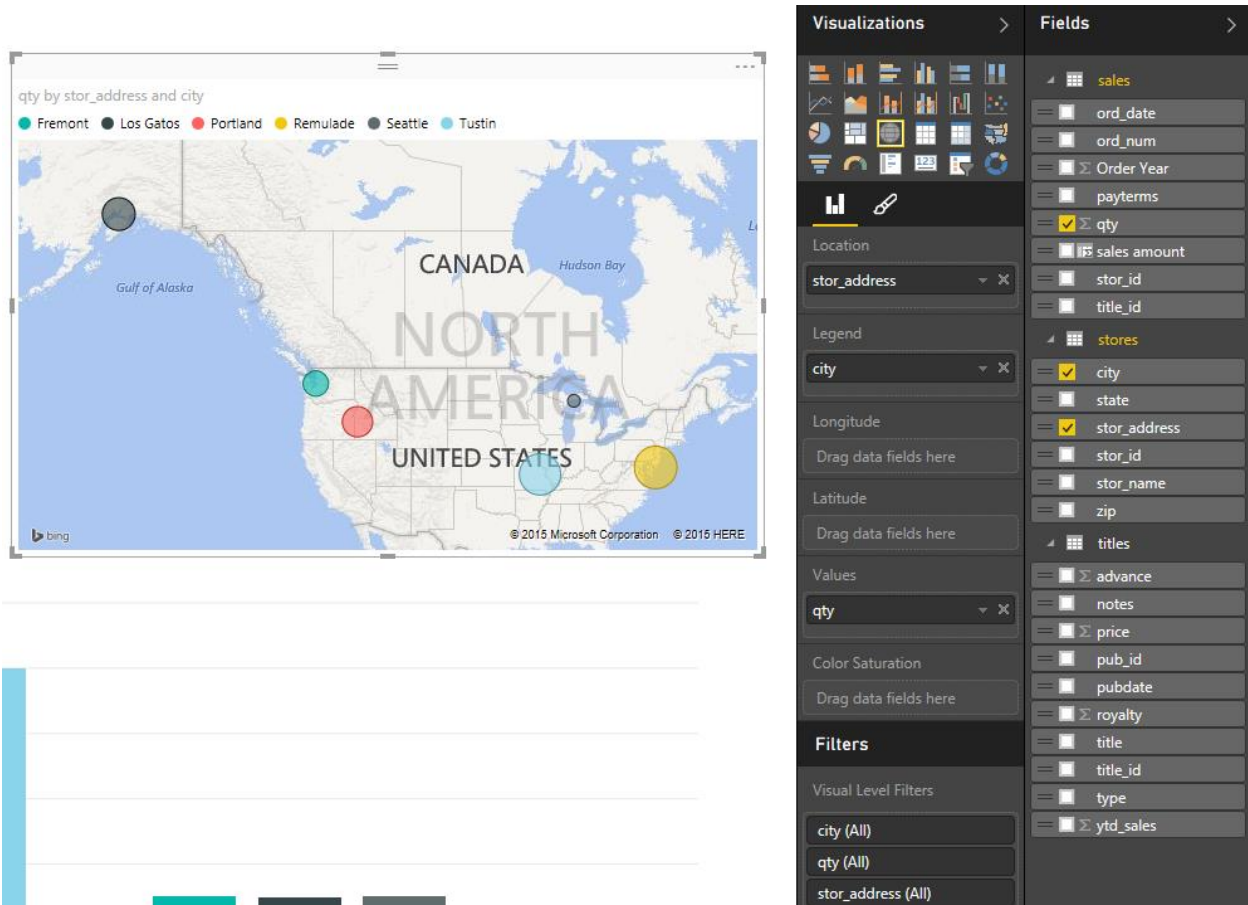
Our model is ready now to be visualized in a report. Go to Report Tab (the default tab in Power BI Desktop). and click on stacked column chart from Visualizations pane. You'll see an empty chart in the report window.



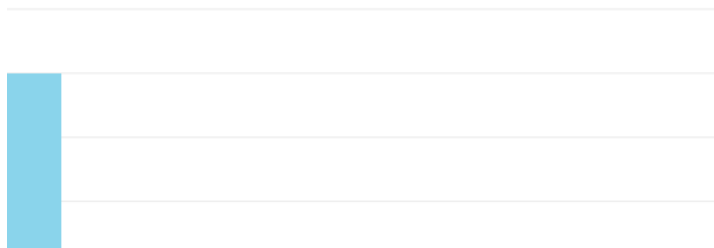
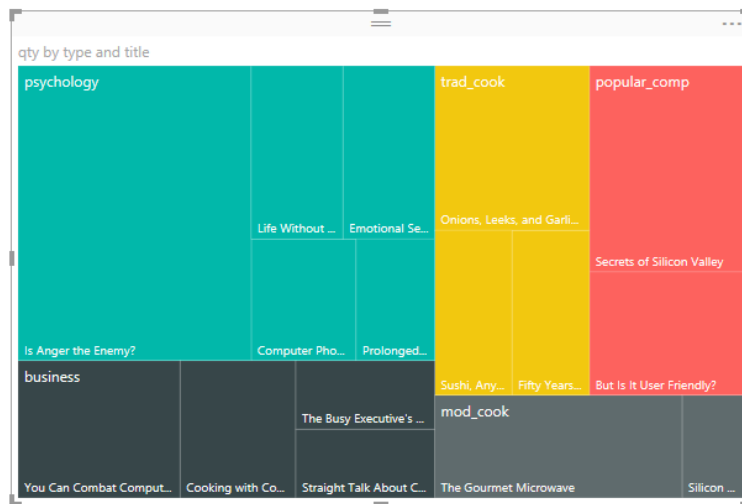
From the fields pane drag and drop sales amount into the Value section of the chart. then drag and drop city (from stores table) into Legend, and then drag and drop title (from titles table) into Axis. Done! you have a clustered column chart already. you can change height and width of the chart simply (I bet you'll figure it out how to).
















Let's create a Map now. Click on an empty section of the report, and then click on Map. set Location as stor_address (from stores table), Values as qty (from sales table), and Legend as city (from stores table). Your map is now ready.



Let's add a treemap to the experience as well. Set the treemap with Group as type (from titles table), Details as title (from titles table), and Values as qty (from sales table).



Visualizations

Group

type

Details

title

Values

qty

Color Saturation

Drag data fields here

Filters

Visual Level Filters

qty (All)

title (All)

type (All)

Page Level Filters

Drag data fields here

Fields

sales

ord_date

ord_num

Order Year

payterms

qty

sales amount

stor_id

title_id

stores

city

state

stor_address

stor_id

stor_name

zip

titles

advance

notes

price

pub_id

pubdate

royalty

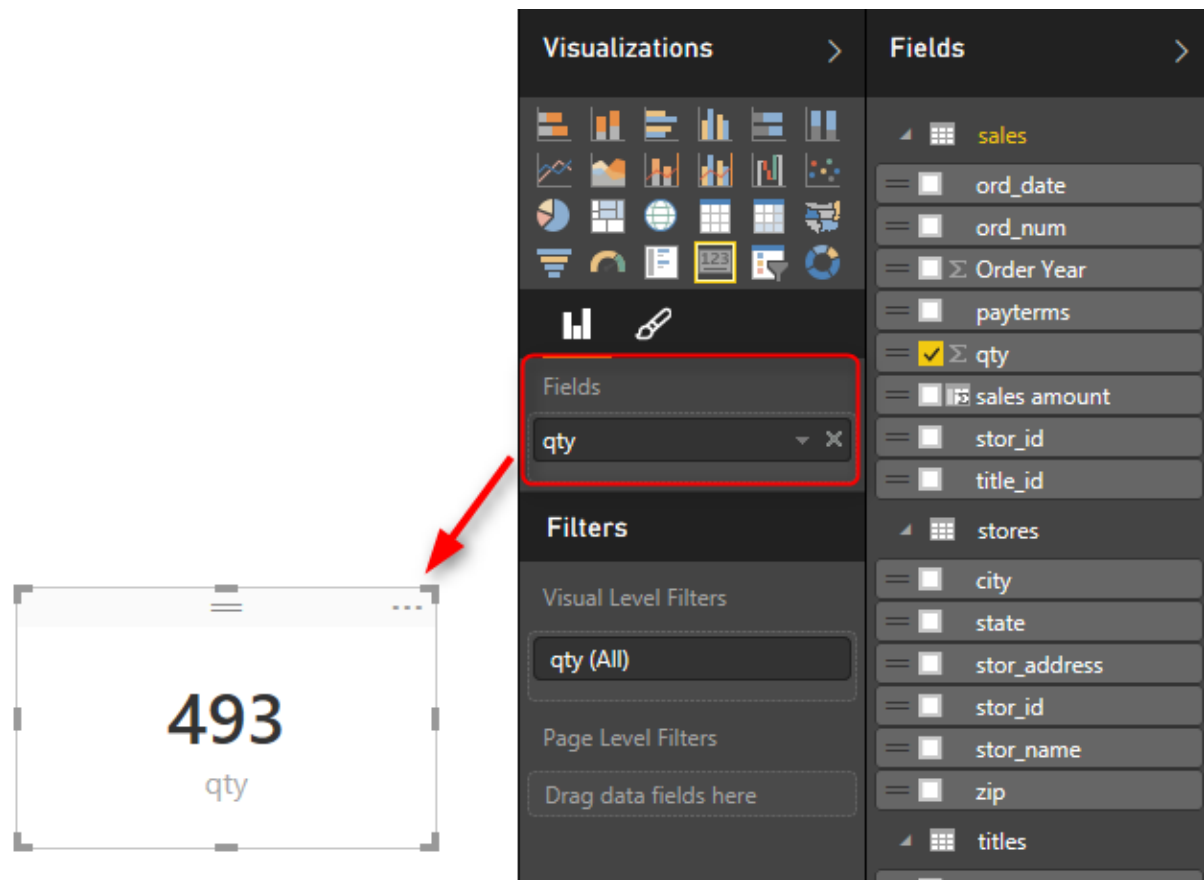
title

title_id

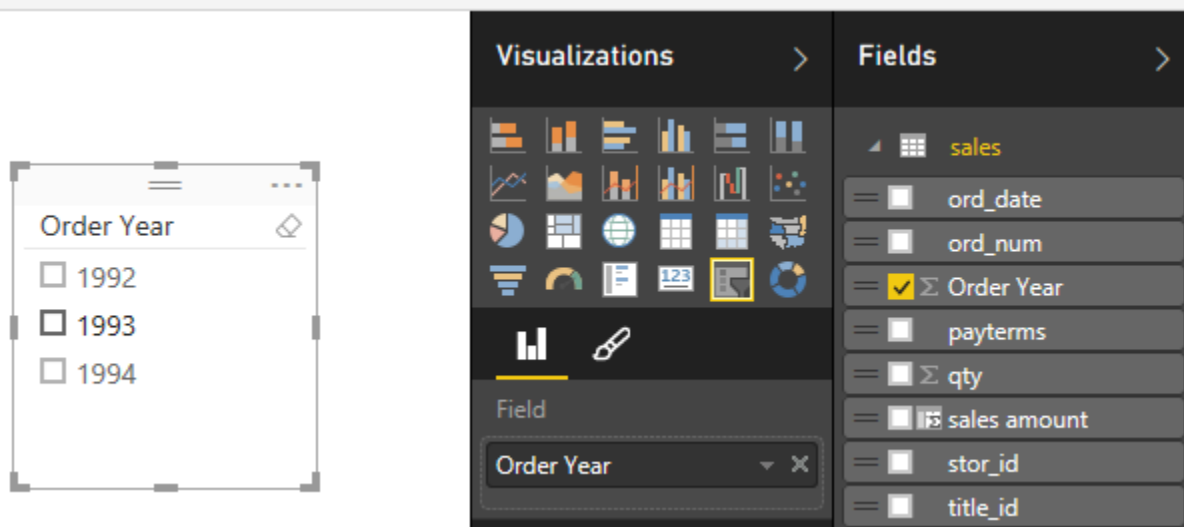
type

ytd_sales

Let's add a total quantity sold as a text box in the report as well. Add a Card from visualization, and choose qty to be displayed there.

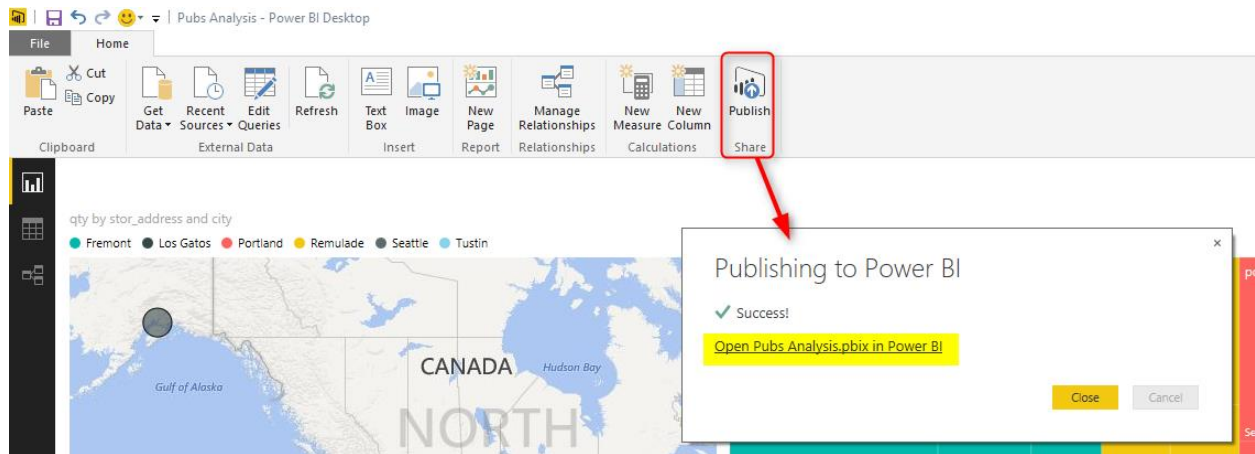


Now add an slicer as the last item for your report. Slicer now can slice and dice the data by Order Year (remember we've added this column in modelling part)

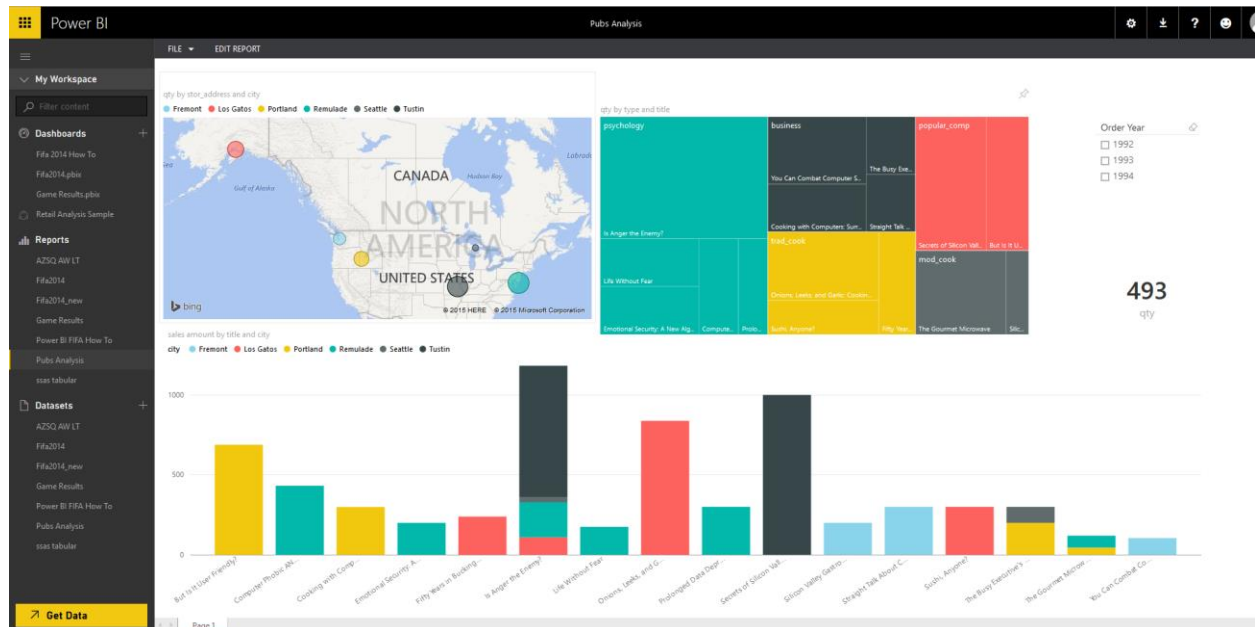


Publish

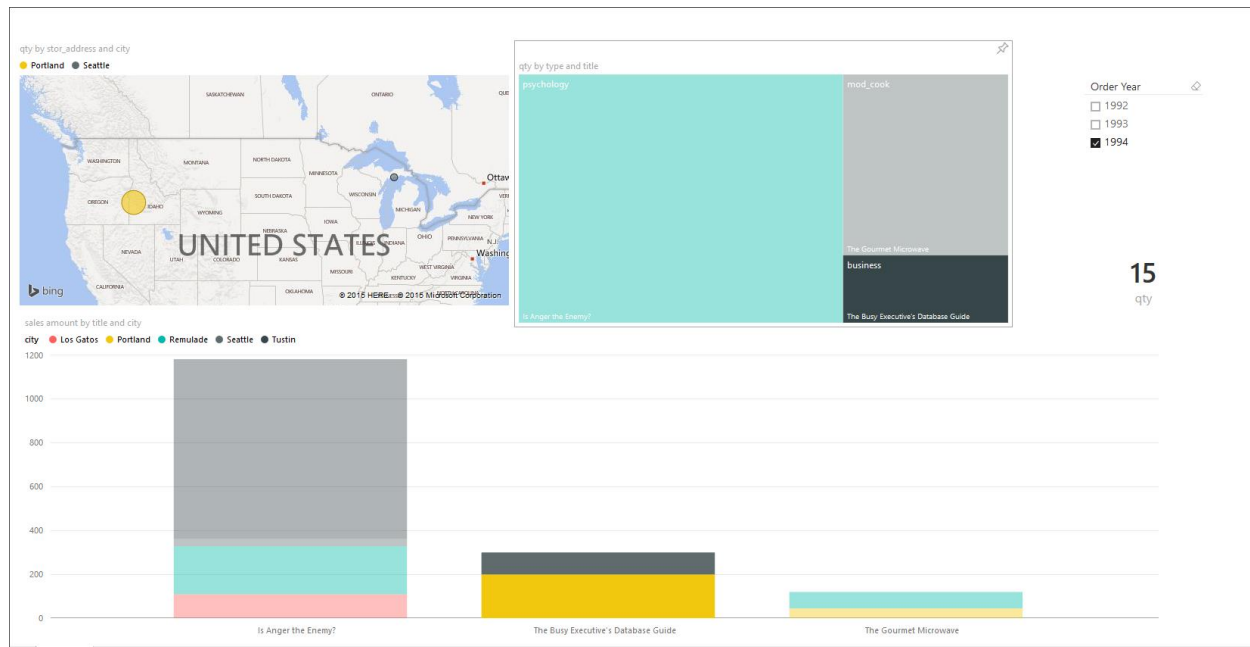
Save the Power BI Desktop file as Pubs Analysis. In the Report tab of Power BI Desktop click on Publish (you'll need to enter your account details here for the first time). You'll get a success message when publish is completed.



Now open the Pubs Analysis in Power BI (just click on the link in publish window), and you'll see your report online in web browser.



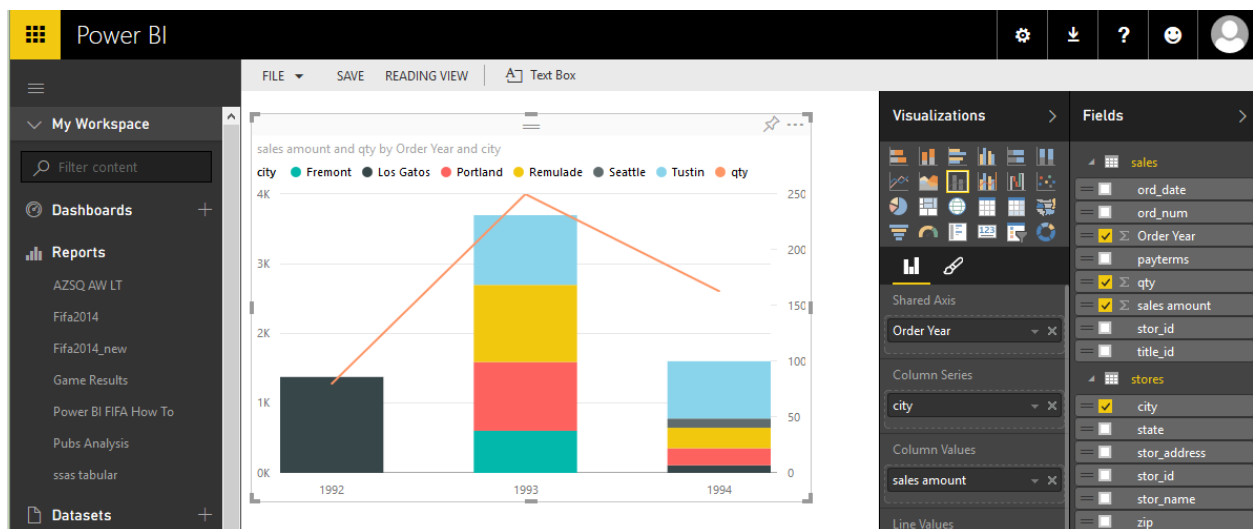
Your report is also fully interactive as well, click on charts and columns and maps to see how highlighting features works interactively.



Well done! you've made your first report with Power BI Desktop, and I know that you'll say it was easy, fun, and interesting. In this section you've learned what is Power BI Desktop. You've connected to a SQL Server as data source and transformed the extracted data. You've loaded data into a model and added a calculated column to it. You've created a report with charts and visualization elements, and finally you published your report in Power BI website. In next section we will go to Power BI website and Apps to see how they works.

Power BI Website: You'll Need Just A Web Browser

Published Date: August 11, 2015



In [previous section](#) (from the [Power BI from Rookie to Rockstar book](#)) you've learned about Power BI Desktop, and you've developed a sample report with that and published it to Power BI website. In this section I'll explain some of Power BI website's functionality. You will understand at the end of this section that Power BI is not just a host for Power BI reports, it gives you option to connect to data sources, and build reports as well, All in web browser! You won't need to install anything to use it. I'll also show you Power BI app for mobile and we'll explore it to see what it gives us in overall.

In this section I will go through below modules with you;

- Create Reports in Power BI Website
- Get Data from Website
- Building Dashboards
- Sharing
- Power Q&A
- Power BI Mobile App

I have to mention it again that this section is still part of Introduction to Power BI chapter of the book, and topics above will be covered lightly. Later in future chapters we will go through all options above deeper in details. Purpose of this section (and the

whole Introduction to Power BI chapter) is to give you an understanding of Power BI tools and services, and prepare you for detailed discussions in future chapters.

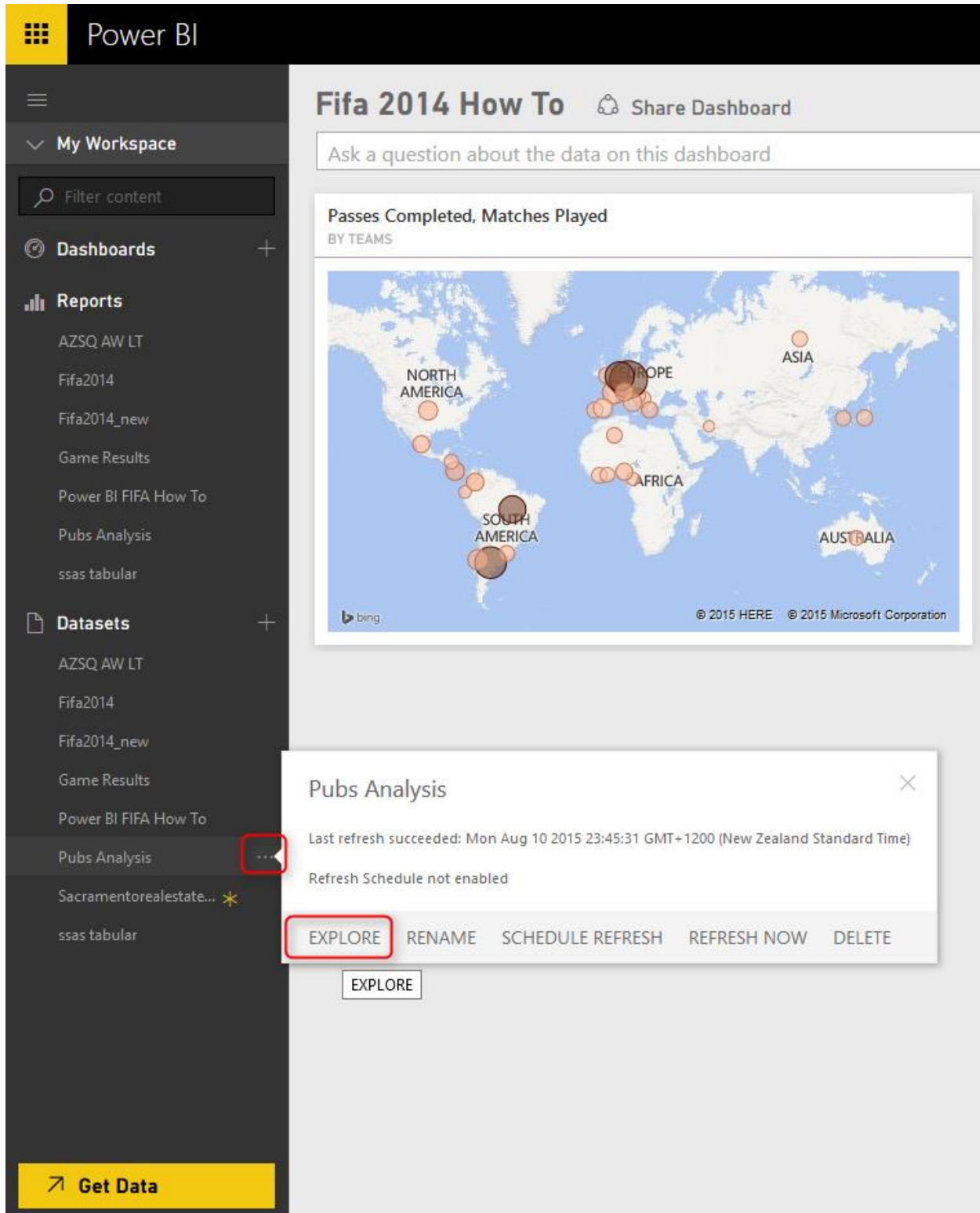
Power BI Website

Power BI Website is a location to host your *.pbix (Power BI) files. You can publish Power BI files directly from the website, or from Power BI Desktop (As you've seen in previous section). However Power BI website is not just for hosting these reports. You can create dashboards from reports in Power BI and share dashboards with others. You can also connect to some data sources directly from the website, and create the report online. Yes, you can do it all with just your web browser. You can also edit an existing report, and re-write it, or save it as a new report. You can schedule data sets to refresh automatically (not all data sources support this feature). As you see Power BI website is not just a hosting location then, it is much more. Sometimes you can do some stuff just with the web browser.

Create Reports in Power BI Website

You can edit reports in the website and rewrite them, or you can save them as a new report. You can also create a new report from an existing or new data source. Let's use the data set that we've built in [previous section](#) and use that as the source for our new report. Follow the instruction below to create a report in the website.

Login to PowerBI.com website with your login. From the left pane menu under Datasets you'll see Pubs Analysis (the data set that we've built from previous example). You can also see Pubs Analysis under reports as well. Pubs Analysis report is the visualization report we've built, and the dataset just contains metadata, and the data. click on ellipsis button near Pubs Analysis data set and choose explore from the popup menu.



Power BI

Fifa 2014 How To [Share Dashboard](#)

Ask a question about the data on this dashboard

Passes Completed, Matches Played
BY TEAMS

Pubs Analysis

Last refresh succeeded: Mon Aug 10 2015 23:45:31 GMT+1200 (New Zealand Standard Time)
Refresh Schedule not enabled

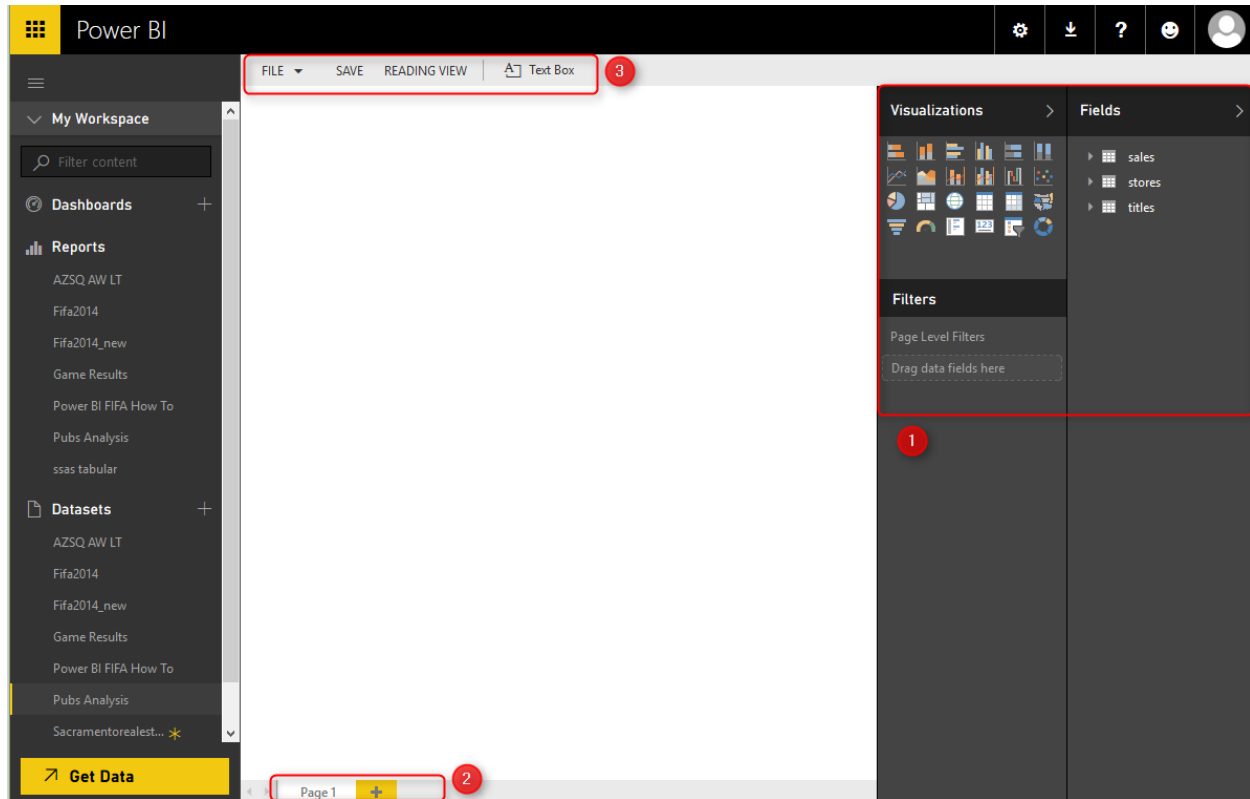
EXPLORE RENAME SCHEDULE REFRESH REFRESH NOW DELETE

EXPLORE

Get Data

When you click explore, a report window will open in the main window. This is similar to Report tab in Power BI Desktop. You'll have the Visualization, Fields, and Filters pane in

the right side (numbered as 1 below), You can add pages (numbered as 2), and you can save the report, or view that, or add a text box to it (numbered as 3). Here is an screenshot of the report editor online

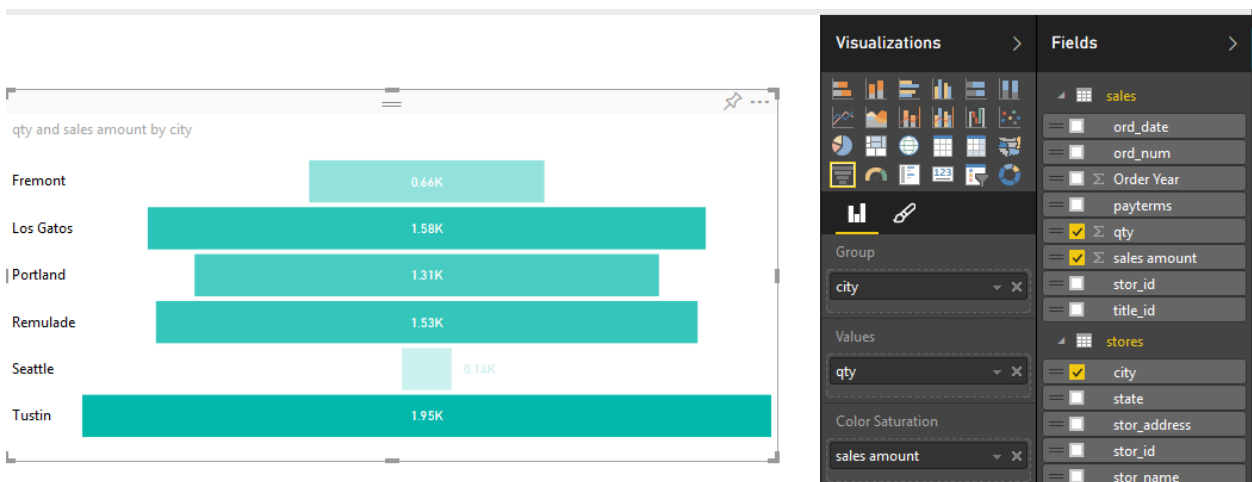


The report editor is not exactly similar to the report tab of Power BI Desktop, there are less functions here (For example you cannot add image here). But most of the functionality is available through the online report editor. and it works perfectly with the web browser. Let's build a simple report;

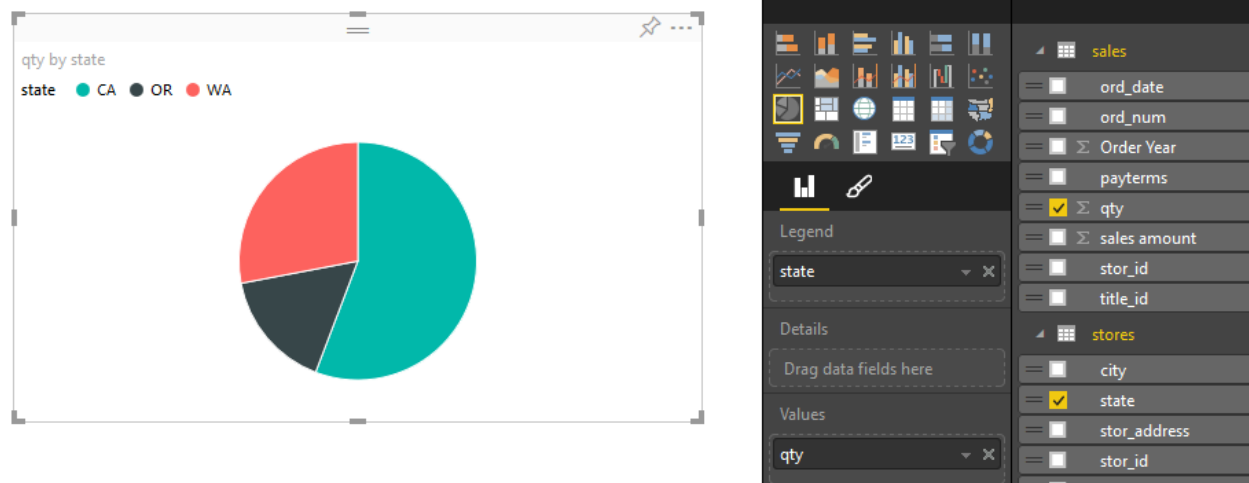
Create a "Line and Stacked Column Chart" from the Visualization pane. Set Shared Axis as Order Year (from sales table), and Column Series as city (from stores table), and Column Values as sales amount (from sales table), and finally Line Values as qty (from sales table). The report illustrated below;



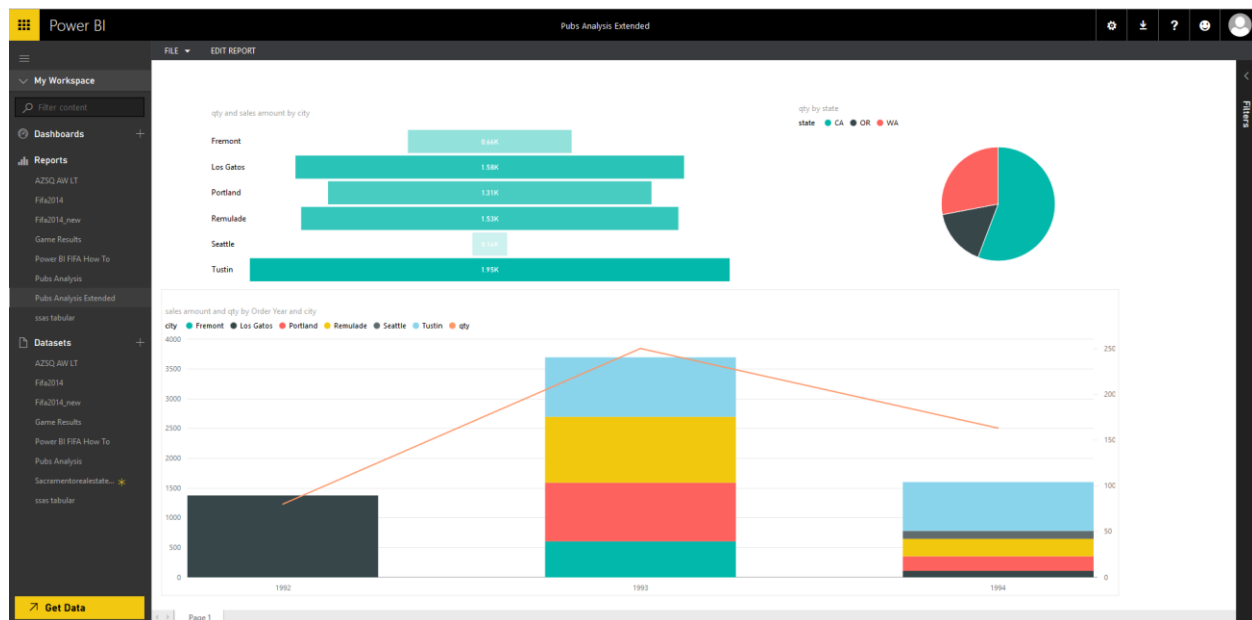
Let's create another visualization in the report. Add a Funnel to the report. Set Group to city, Values to qty. Also set Color Saturation to sales amount. Color saturation will highlights the funnel bars based on sum of sales amount.



For the last chart in this report, let's build a pie chart. I know that pie chart is not good chart for visualization, But I want to use it for 3 distinct categories with really distinguishing values. (Please remember that don't use pie chart with categories are more than 3 and 4, and values are quite close to each other). Add a pie chart to the report. Set it with Legend as state (from stores), and Values as qty. below is the view of pie chart

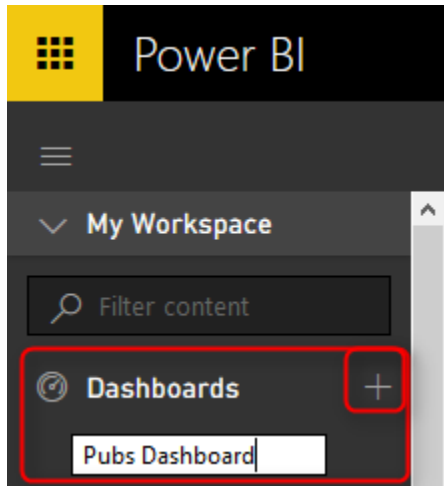


Now let's save the report with name of Pubs Analysis Extended (save option is in top left side of the report editor). You now see the new report under Reports. if you click on it you'll see it in the viewer.



Build Dashboard

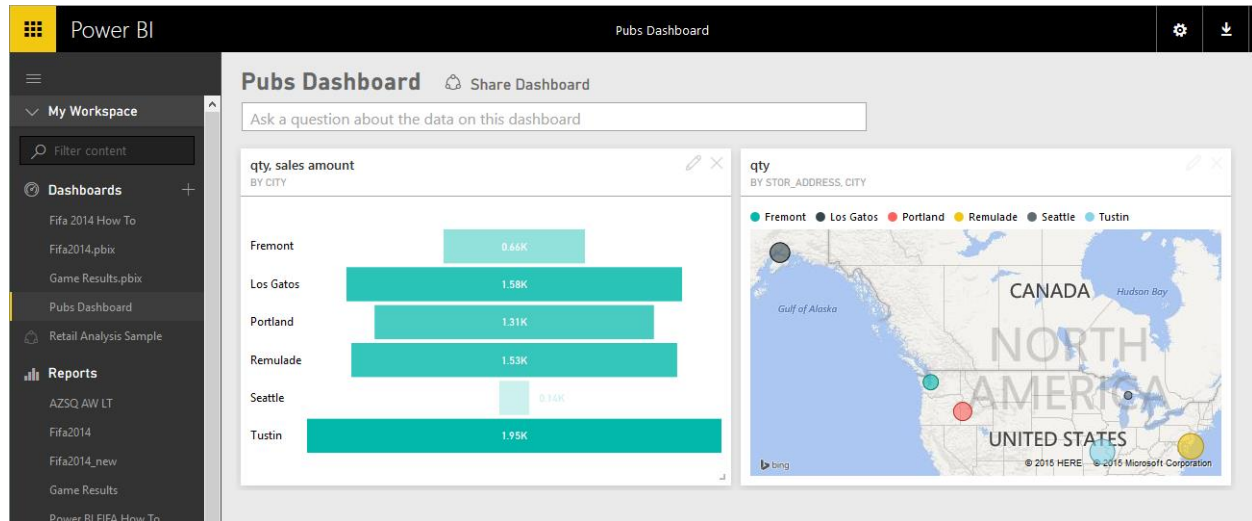
You can build dashboards from your existing reports. dashboards will be primary view point of you or users (from website or mobile app). To create a dashboard, click on Add button close to Dashboards in the left pane, and create a new dashboard with name of Pubs Dashboard.



An empty dashboard will be created. Now go to Pubs Analysis Extended report and click on the right top side of funnel chart to “Pin Visual” as below. You’ll see a success message that chart pinned to dashboard.



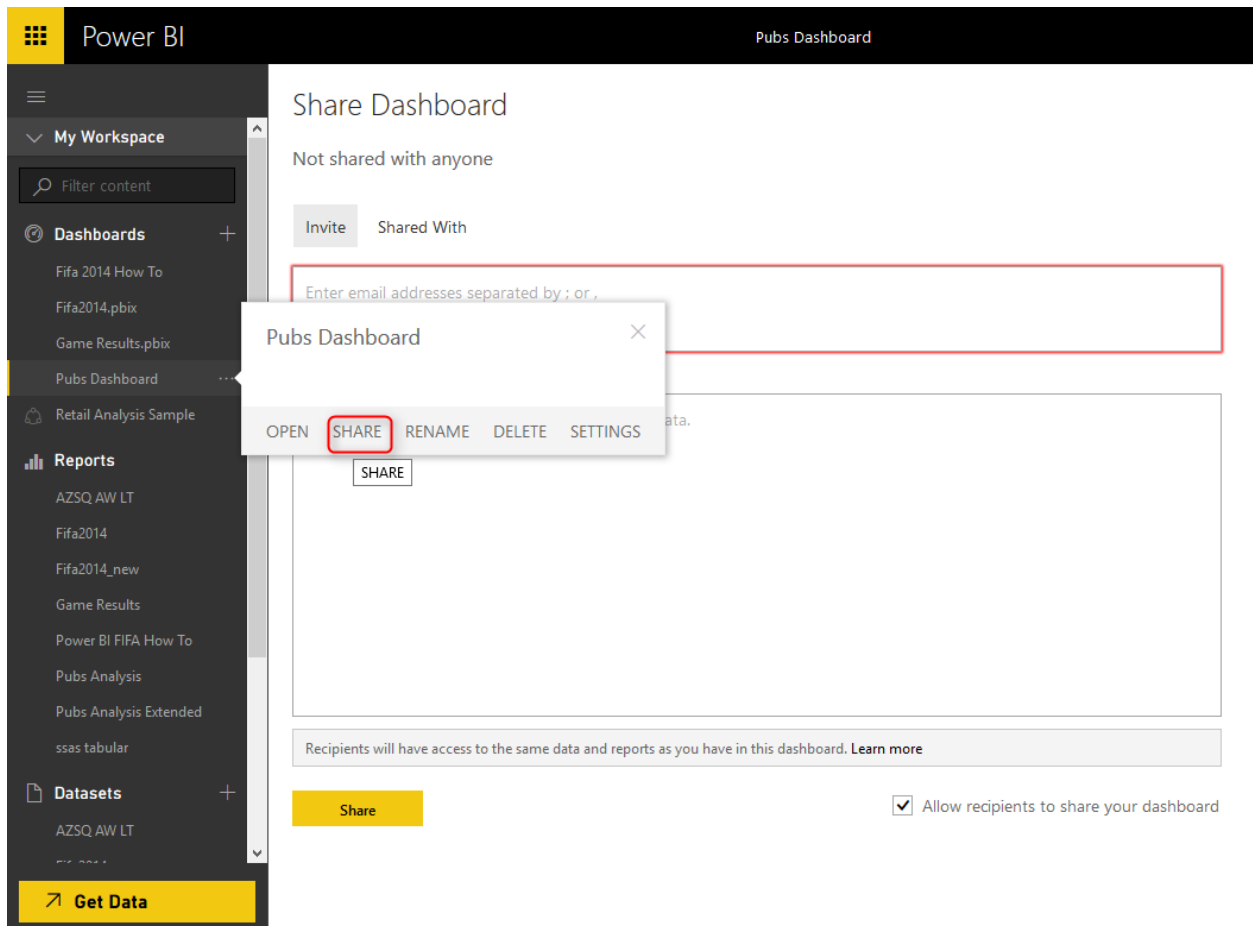
Go to Pubs Analysis report, and pin visual the Map visualization as well. now the Pubs Dashboard looks like below screenshot;



If you click on any item in the dashboard you'll be redirected to the underlying report.

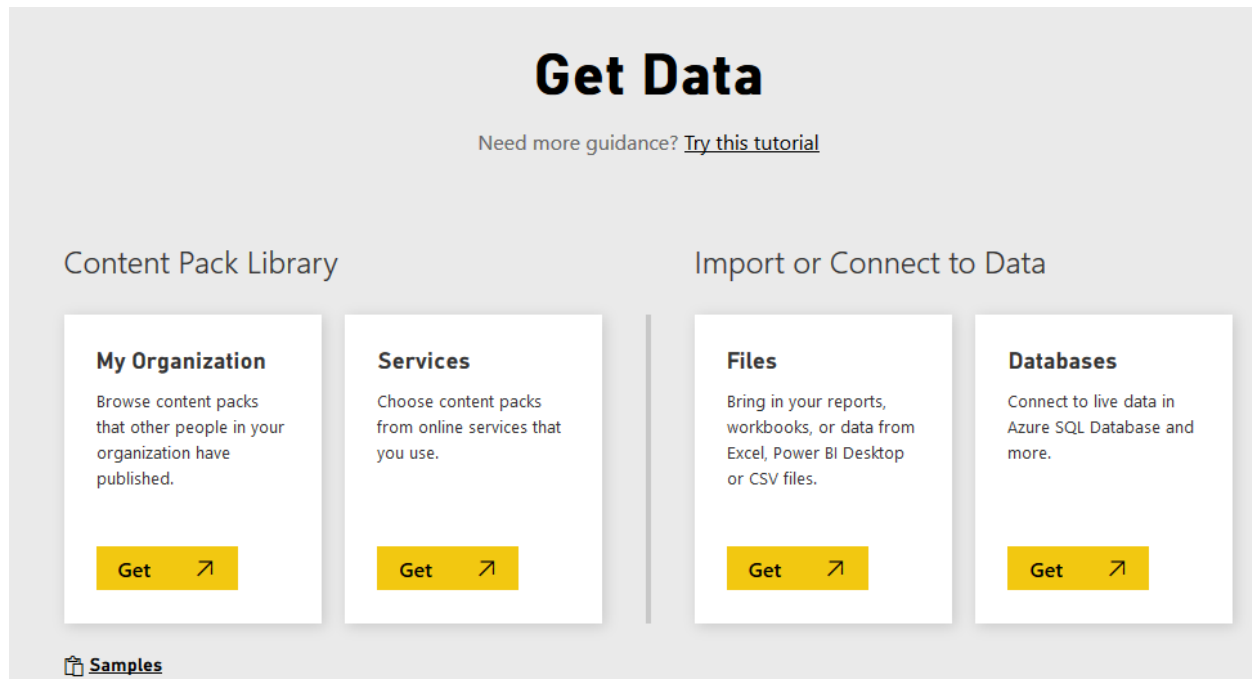
Sharing

Dashboards can be easily shared with other users. just click on ellipsis button of dashboard and choose Share.

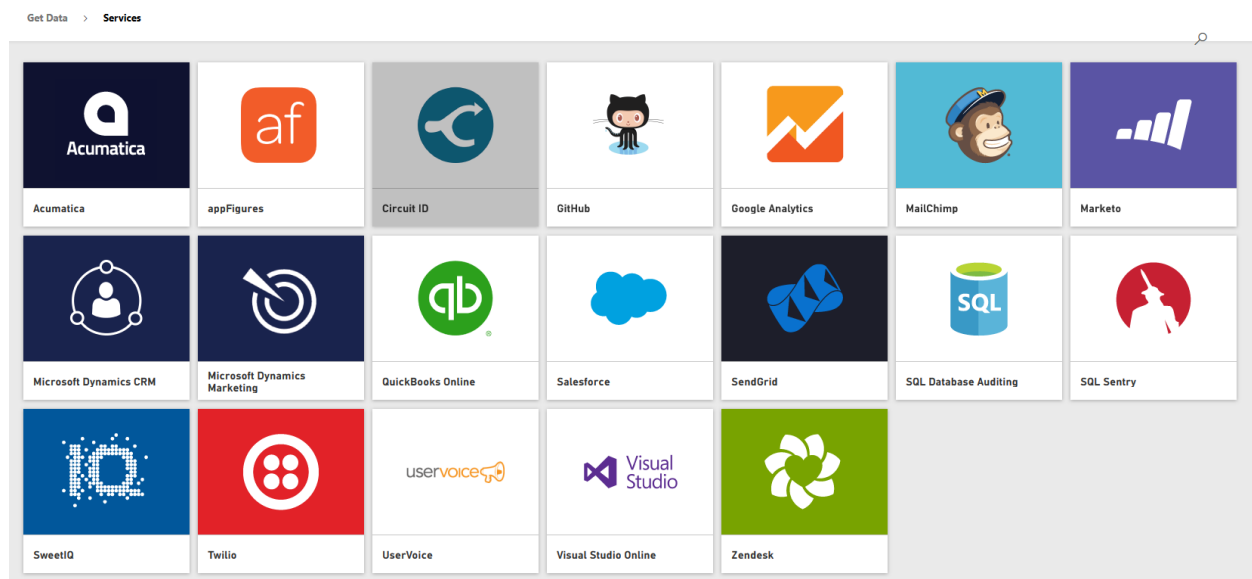


Get Data

You can connect to some data sources from the Power BI website. This option is available through Get Data button.



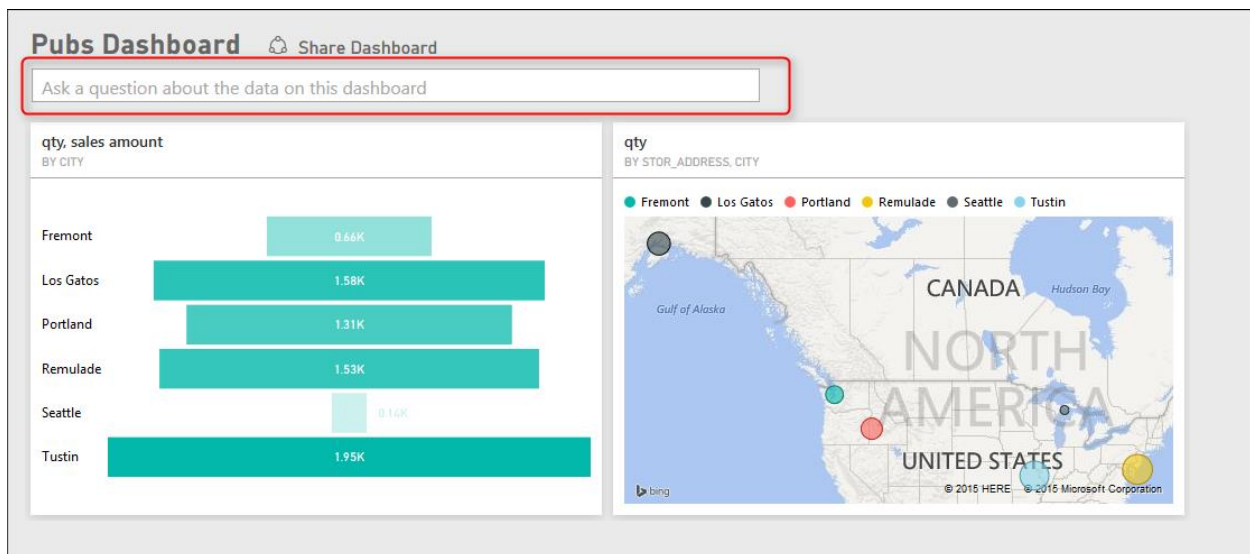
There are some data sources supported in online Get Data. You can get data from services (such as Google Analytics, MailChimp, CRM, Salesforce...), or get data from Files (such as Local File, OneDrive), or get from Databases (such as Azure SQL DB, Azure SQL DW...). Some of these options requires Power BI Pro, but most of them are available through normal Power BI account. screenshot below shows services that supported to get data from (the list updates frequently, and every month new data sources will be added)



Power Q&A

Power Q&A is the question and answering engine of Power BI which works based on natural speaking language. You can ask questions in English language from the Power BI dashboard and get responses! Power Q&A engine will translate your query to machine query and respond that with visualization of the data in the model.

To ask a question simply use the question bar in the dashboard



For example you can ask from Pubs Dashboard: "Show city sales amount", and the response is:

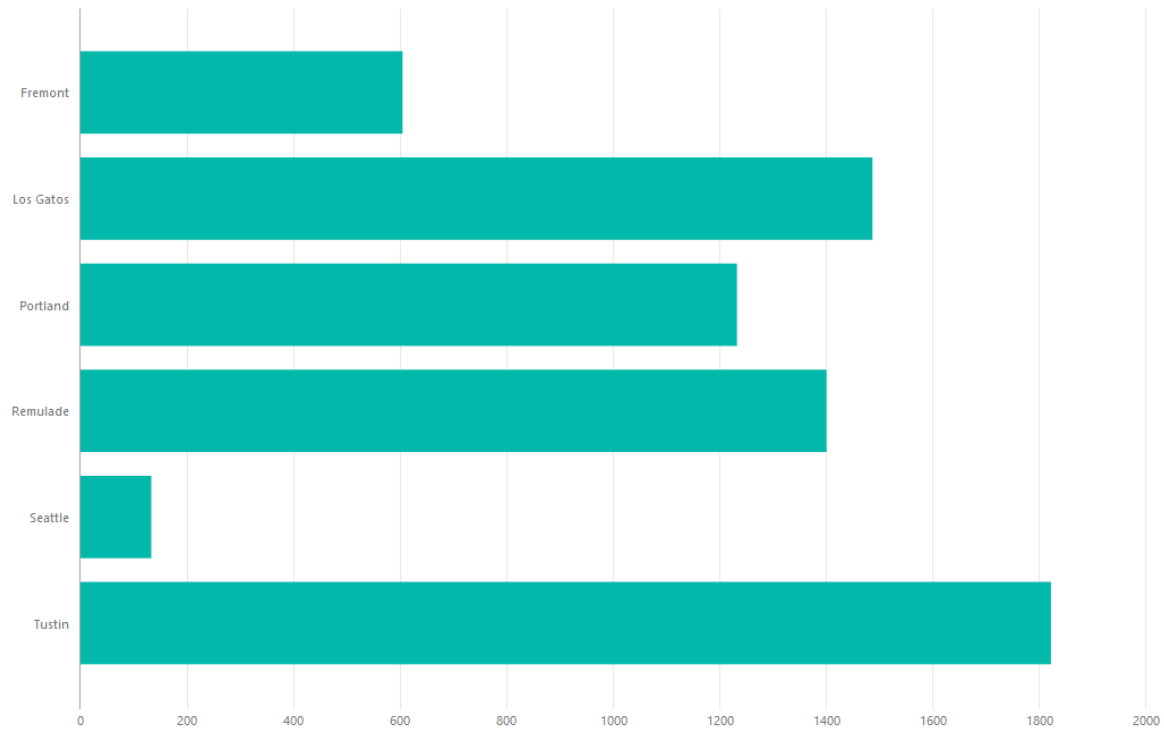
← Pubs Dashboard

show city sales amount



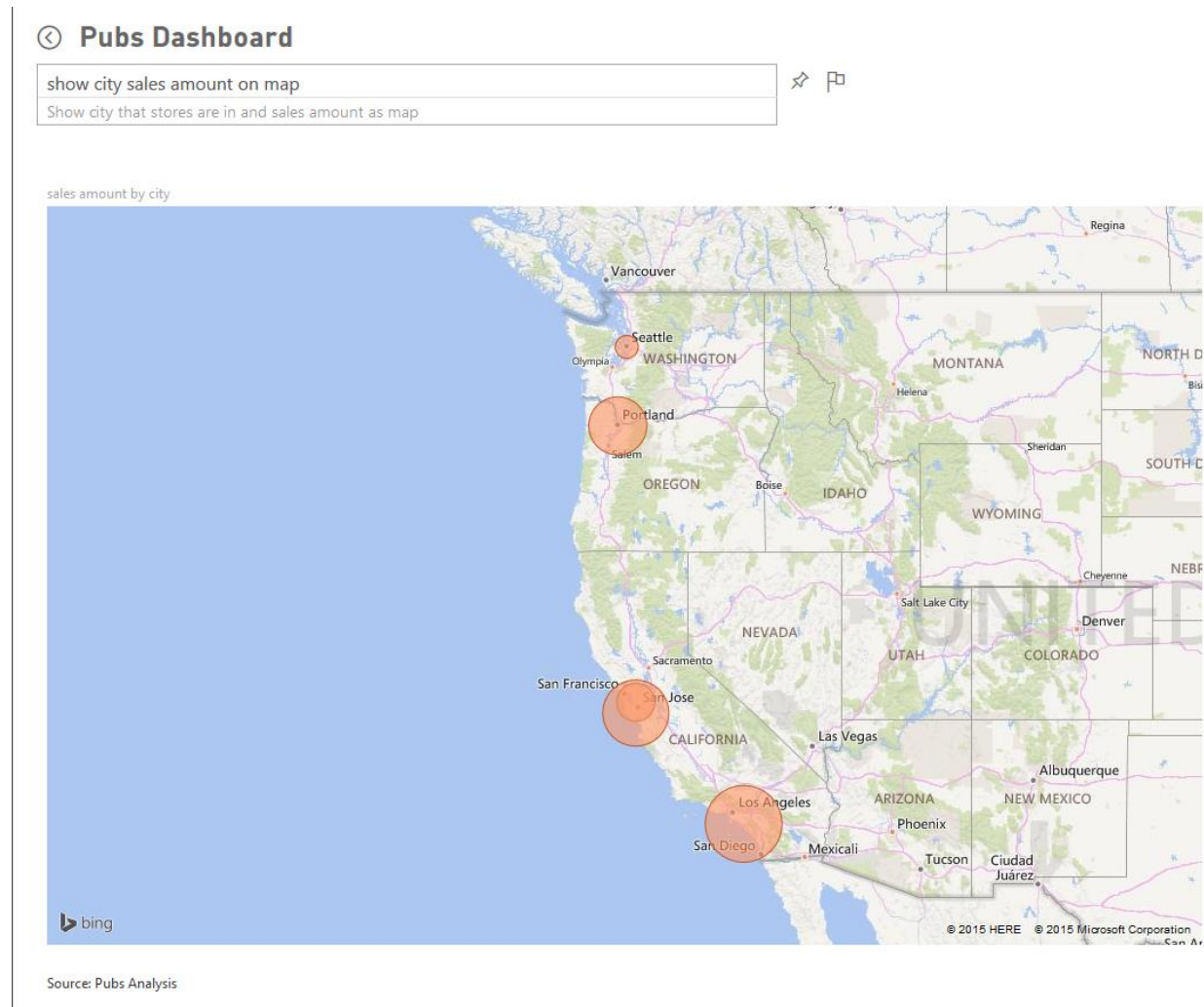
Show city that stores are in and sales amount

sales amount by city



Source: Pubs Analysis

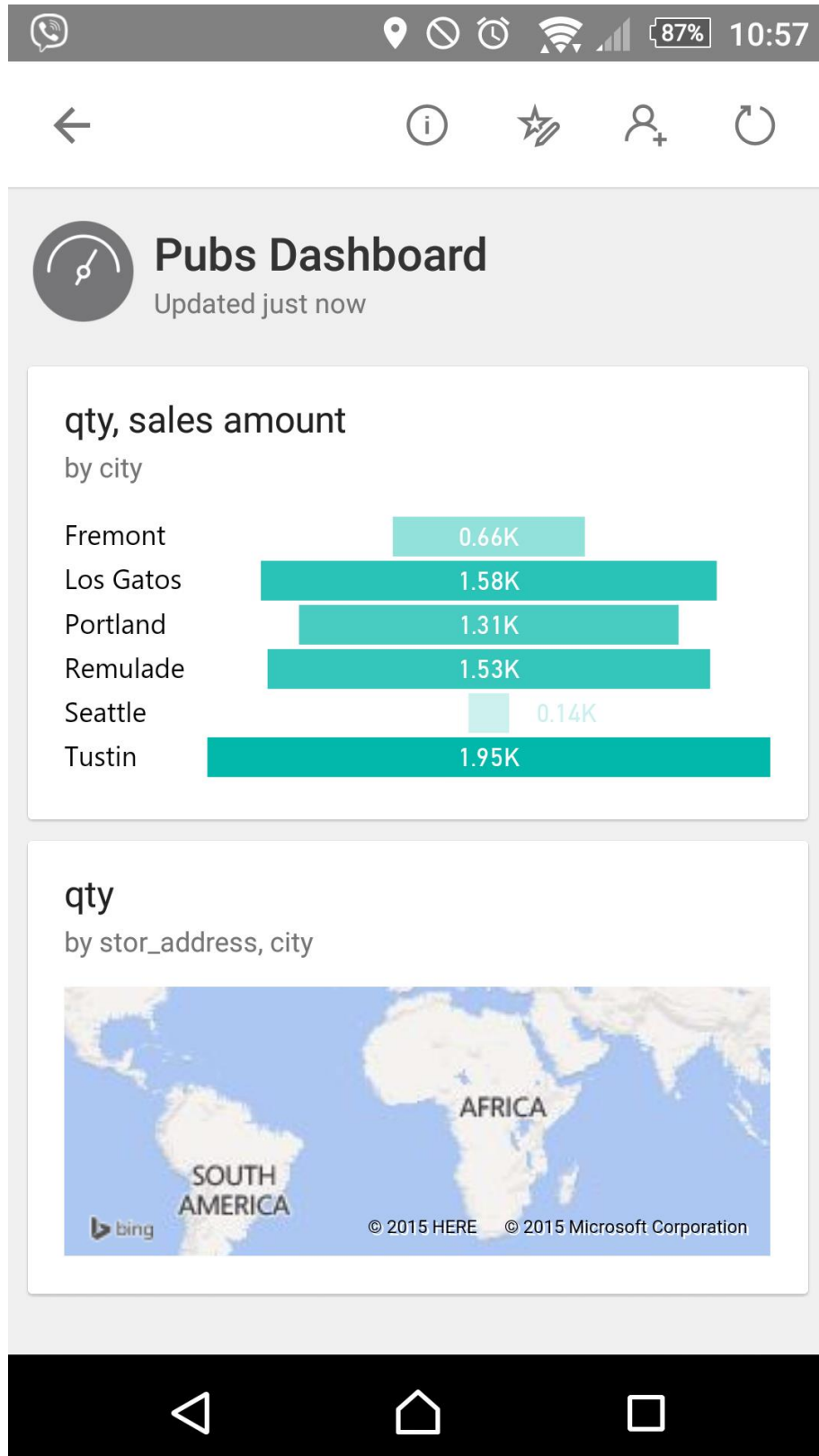
you can even change the visualization to map with this question: "Show city sales amount on map"



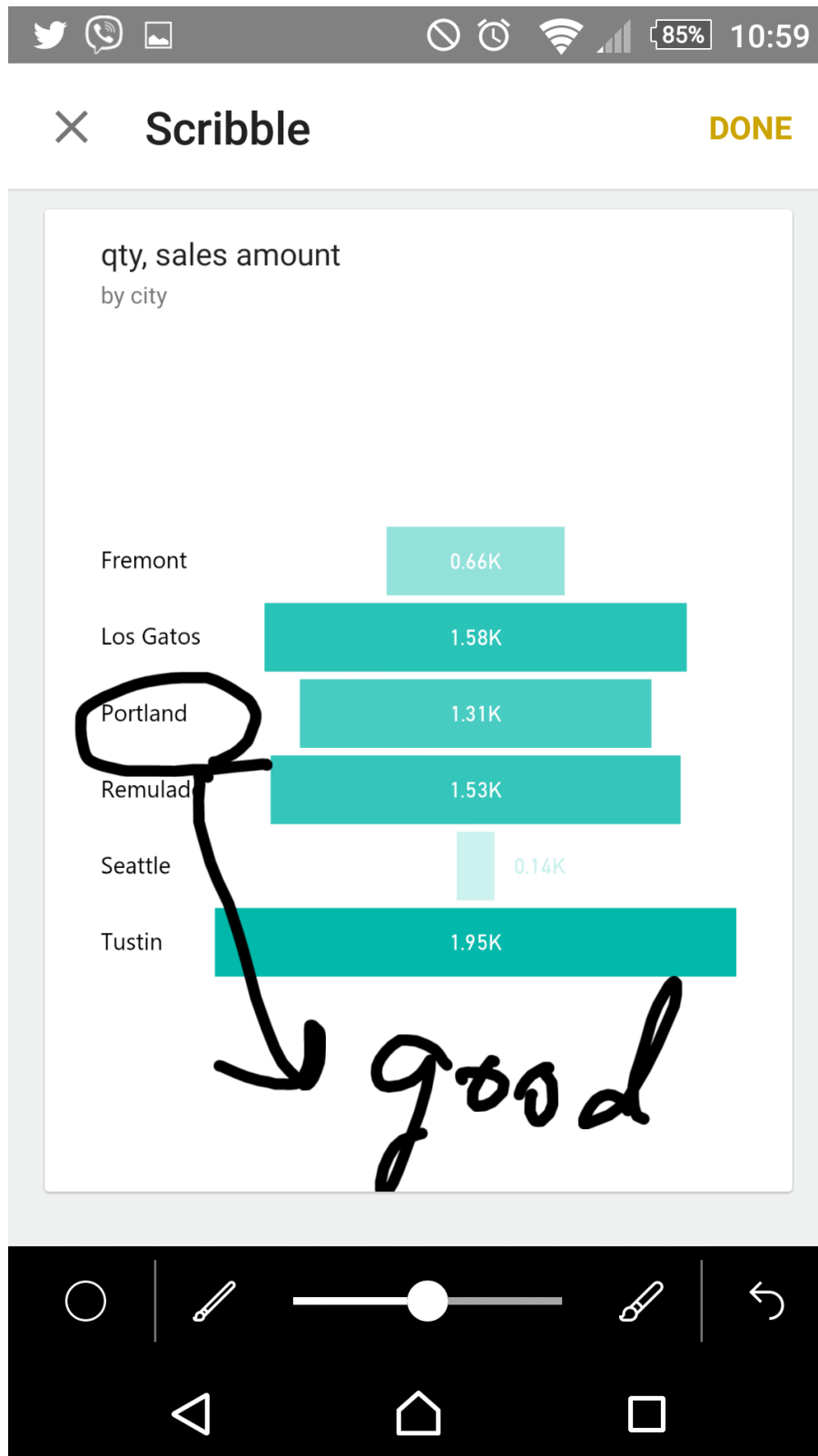
Power Q&A is a very powerful engine. There are some tips and tricks of how to develop your data model so Power Q&A could generate best responses to questions. I'll cover these tips and tricks in future chapters of this book.

Power BI Mobile App

Power BI app is available for Android, Apple, and Windows Phone. simply download it from Goolge Play (Android), or AppStore (Apple), or Windows Store (Windows Phone). After the installation login with your username, and you'll see dashboards there. screenshot below shows the dashboard in Android phone, but other apps are similar to this.



You can highlights some parts of the report and share it as an image



You can even share it from Mobile App



← Share Pubs Dashboard



Enter an email or name to invite

SHARED WITH 0



Reza Rad

Owner



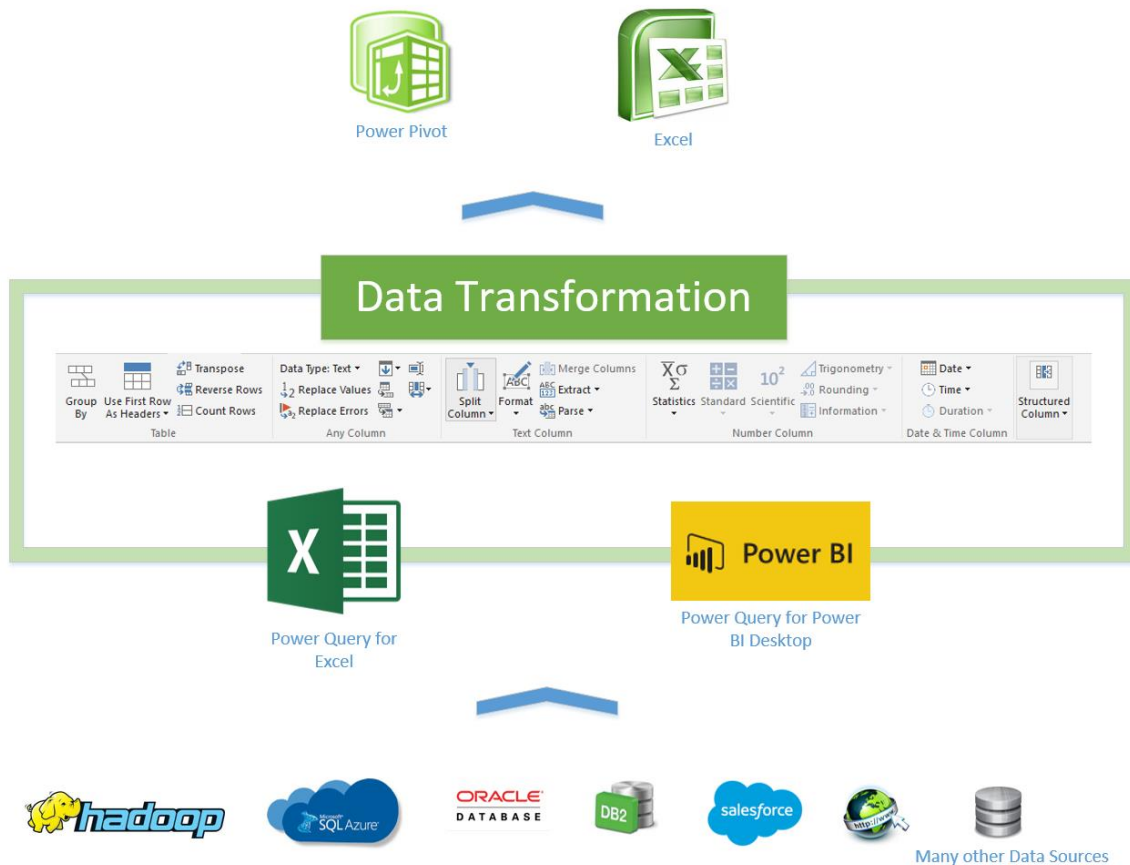
Summary

In summary you've learned about options that are available to you from Power BI website. You've learned that even without Power BI Desktop you can create and edit reports and get data from data sources. You've learned how easy is creating dashboards and sharing it with other users. You've seen some examples of Power Q&A which answers your natural language questions with visualization items and the data in the model. You've also learned that Power BI Mobile Apps interacts nicely with Power BI site and dashboards.

In the next chapter we will go deeper through Get Data Experience of Power BI Desktop and will explore options for different data sources. Data transformations with M formula language also will be discussed in the next chapter.

What Is Power Query? Introduction to Data Mash-up Engine of Power BI

Published Date: August 15, 2015



When you get data in Power BI you actually use Power Query Component. In this chapter you will learn about What Power Query is, and what are different types of sources that Power Query can connect. Power Query has also great list of transformations that can be applied on the data set as well (which will be covered in next chapter), and the Power Query formula language M can be used for complex and powerful data transformation situations (will be covered in a chapter after).

In this section you will read an introduction to Power Query. You will learn;

- What Power Query is?
- What types of works can be done with Power Query?
- What are requirements to run Power Query?

- What are features of Power Query Premium?

What Is Power Query?

Power Query previously named as Data Explorer. Data Explorer has been released as public preview for the [first time at February 2013](#). Data Explorer then renamed to Power Query at July 2013, and from that time it had lots of enhancement on the product. Power Query is on a regular and frequent update plan by Microsoft team and usually you can see monthly updates on this, here is the [latest update notes](#) (released yesterday!) Power Query has been tested a lot during this period and nowadays used in many real world data transformations and BI solutions.

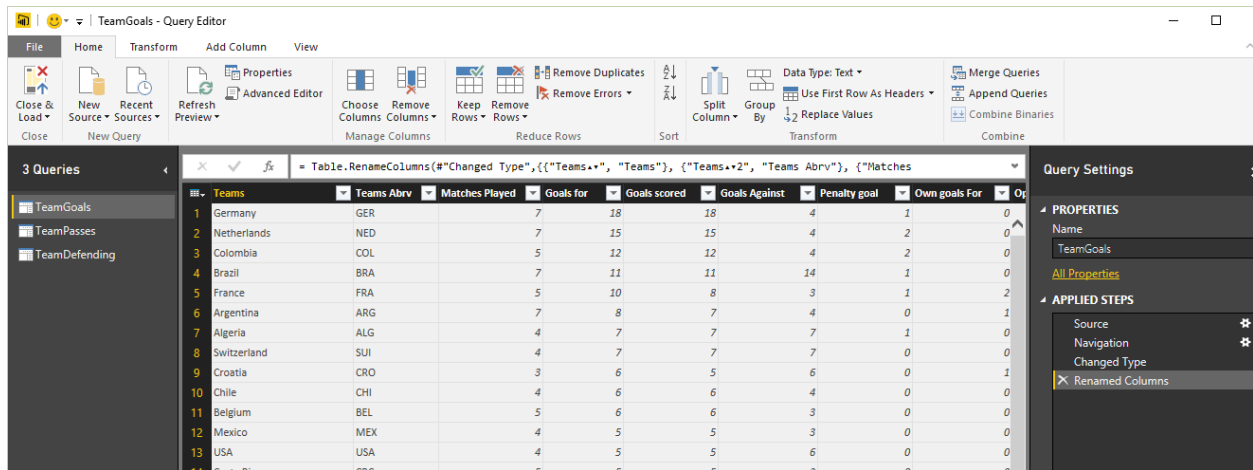
Power Query is a data extraction and transformation engine. The engine comes with a formula language and a graphical tool. The graphical tool has two major setup versions; one embedded in Power BI Desktop tool, and the other one as an Add-In for Excel. The graphical tool has list of transformations that can be applied on a data set, and it also supports different data sources. However the Power Query formula language is much more powerful than the GUI. Actually there are some features in Power Query engine that not yet has been implemented through GUI, but they are available through M (formula language).

Power Query can connects to set of data sources and read data from them. Set of data sources is variable from text files, to web urls, from database systems to some applications. Wide range of data sources is supported. So to respond to one of the very first questions that usually appears when I introduce this product that Can Power Query connect to Oracle? Sure it does! Not only Oracle, but also MySQL, PostgreSQL, DB2, Sybase, and Teradata.

Power Query can apply many transformations to the data set. you can apply simple transformations such as trimming a text value, and applying numeric calculations to complex transformations easily such as pivot and unpivot. Power Query uses a function library for applying transformations, and the function library contains heaps of transformations for every data type such as table, text, record, list, date, number and so on.

Power Query graphical interface is so easy to work with that even business analyst or a power user can work with it, on the other hand Power Query M language is so powerful that can be used for complex real world challenges of data transformations. Power Query can load the result set into an Excel spreadsheet or it can load it into Power Pivot for data modelling. The version of Power Query used in Power BI Desktop load the result set into a Power Pivot model. I will go through details of Power Pivot in future chapters,

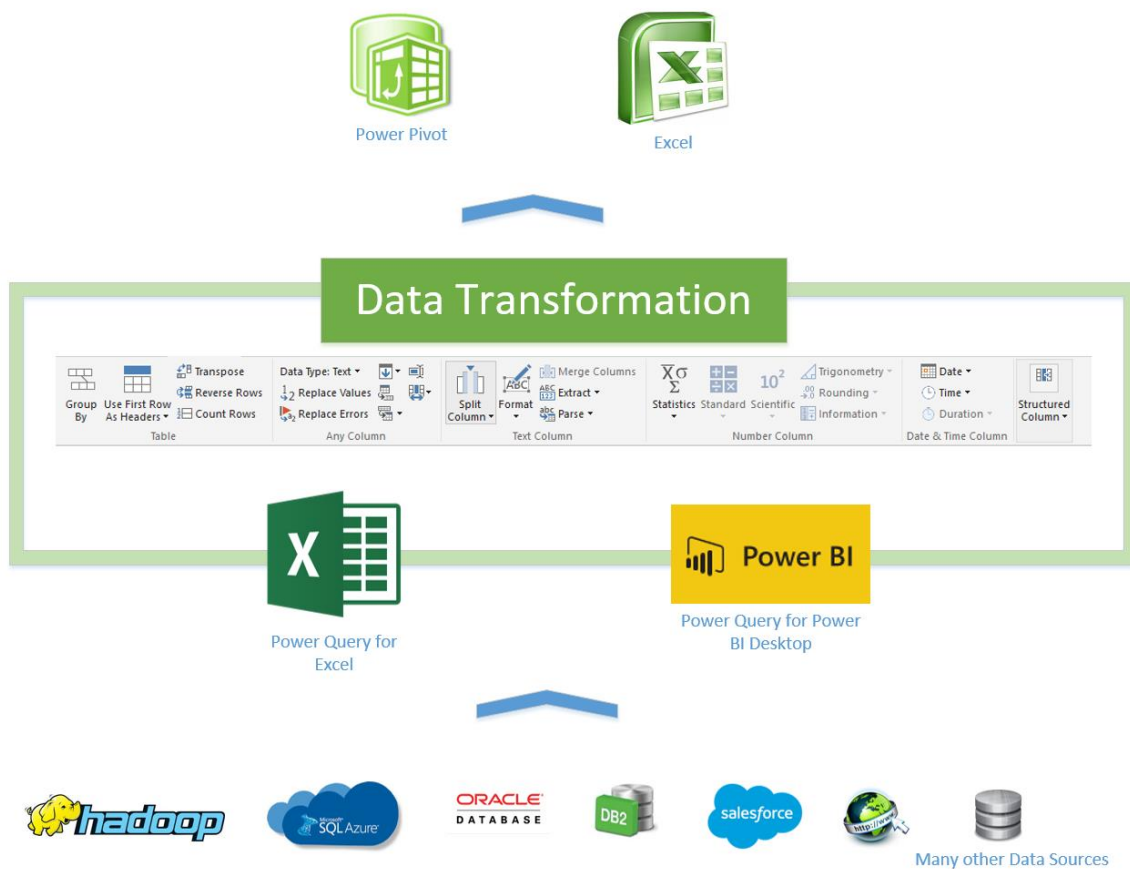
for now it would be enough to know that Power Pivot is In-Memory tabular data model engine. Here is an screenshot or the Query Editor window



The screenshot shows the Power BI Query Editor window with a table named 'Teams'. The table has columns: Teams, Teams Abbrv, Matches Played, Goals for, Goals scored, Goals Against, Penalty goal, and Own goals For. The data is sorted by Teams Abbrv.

	Teams	Teams Abbrv	Matches Played	Goals for	Goals scored	Goals Against	Penalty goal	Own goals For
1	Germany	GER	7	18	18	4	1	0
2	Netherlands	NED	7	15	15	4	2	0
3	Colombia	COL	5	12	12	4	2	0
4	Brazil	BRA	7	11	11	14	1	0
5	France	FRA	5	10	8	3	1	2
6	Argentina	ARG	7	8	7	4	0	1
7	Algeria	ALG	4	7	7	7	1	0
8	Switzerland	SUI	4	7	7	7	0	0
9	Croatia	CRO	3	6	5	6	0	1
10	Chile	CHI	4	6	6	4	0	0
11	Belgium	BEL	5	6	6	3	0	0
12	Mexico	MEX	4	5	5	3	0	0
13	USA	USA	4	5	5	6	0	0
14	Costa Rica	CRI	5	5	5	2	0	0

In below you can see a high level diagram of Power Query conceptually:



How to Use Power Query?

Power Query is available in three different setups:

1. As an Excel Add-In for Excel 2010 and 2013
2. Embedded in Excel 2016
3. Embedded in Power BI Desktop

So if you want to install then you have to install one of the options below:

Excel Add-In for Excel 2010 and 2013:

<https://www.microsoft.com/en-us/download/details.aspx?id=39379>

Please note that the link above might change because Power Query updates frequently and new version will be available almost every month. so you can simply Google it as Power Query Excel add-in.

Excel 2016 download link:

<https://products.office.com/en-us/office-2016-preview>

At the time of writing this blog post Excel 2016 is in preview stage, so the link is likely to change.

Power BI Desktop:

<https://powerbi.microsoft.com/desktop>

What You Can Do With Power Query?

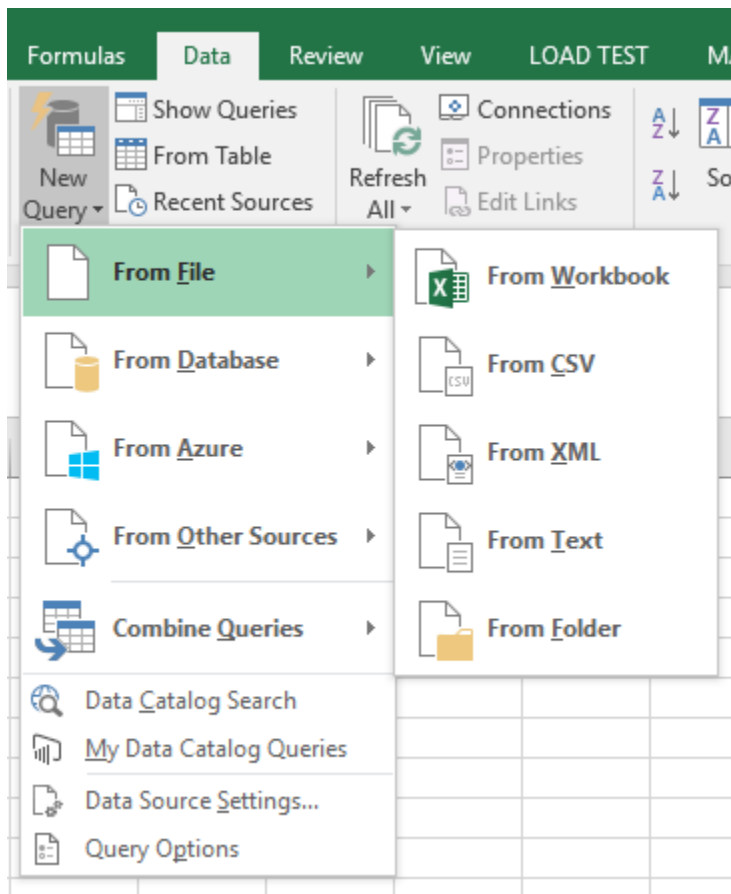
Get Data From Wide Range of Sources

With Power Query you can connect to wide range of data sources. SQL Server or DB2 or Oracle.... All of these database are supported as a source. You can even connect to an Analysis Services instance and fetch data from it. You can connect to file data structures such as text files, XML, CSV, and Excel. You can even read list of files in a folder! You can connect to a range of applications such as Facebook, Salesforce, CRM Online and etc and get data from them. You can get data from Azure services such as Azure SQL

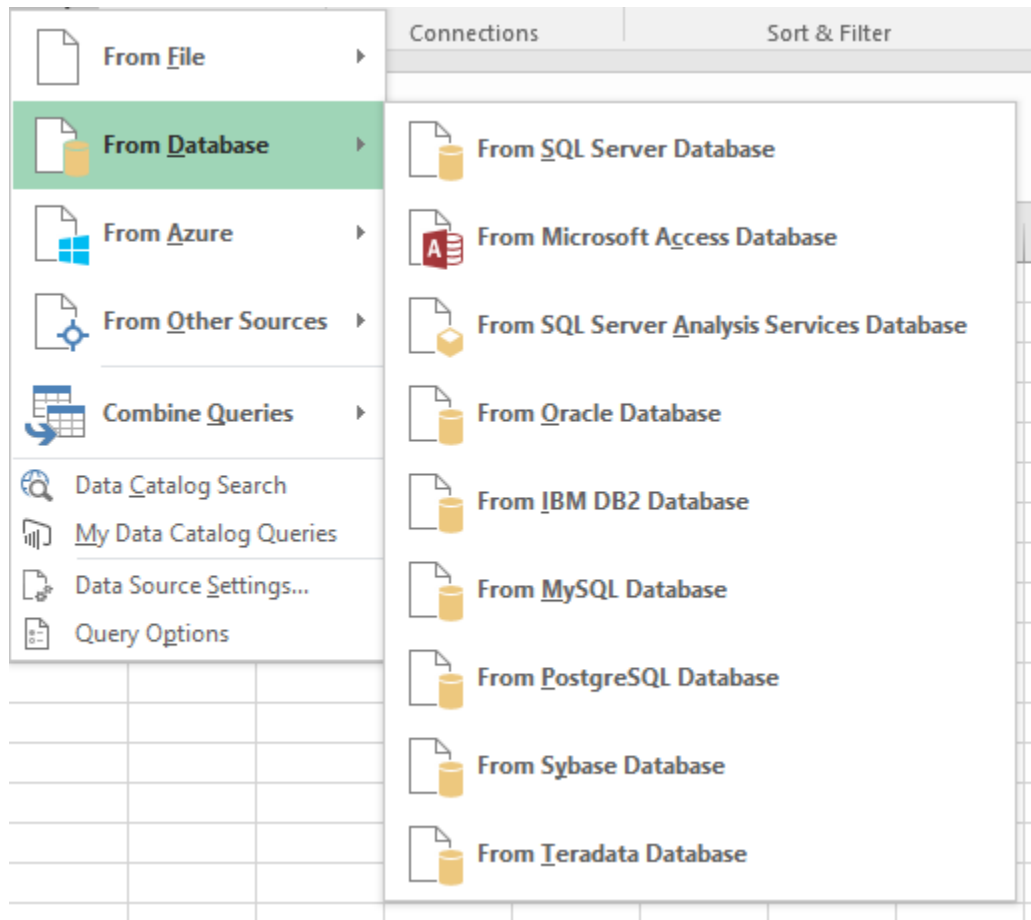
database, Azure HD Insight, Azure Blob storage and etc. There are many data sources supported for Power Query (and obviously for Power BI), also more data sources will be available in every update of Power Query or Power BI.

Here is an example set of data sources supported in Power Query (Excel version):

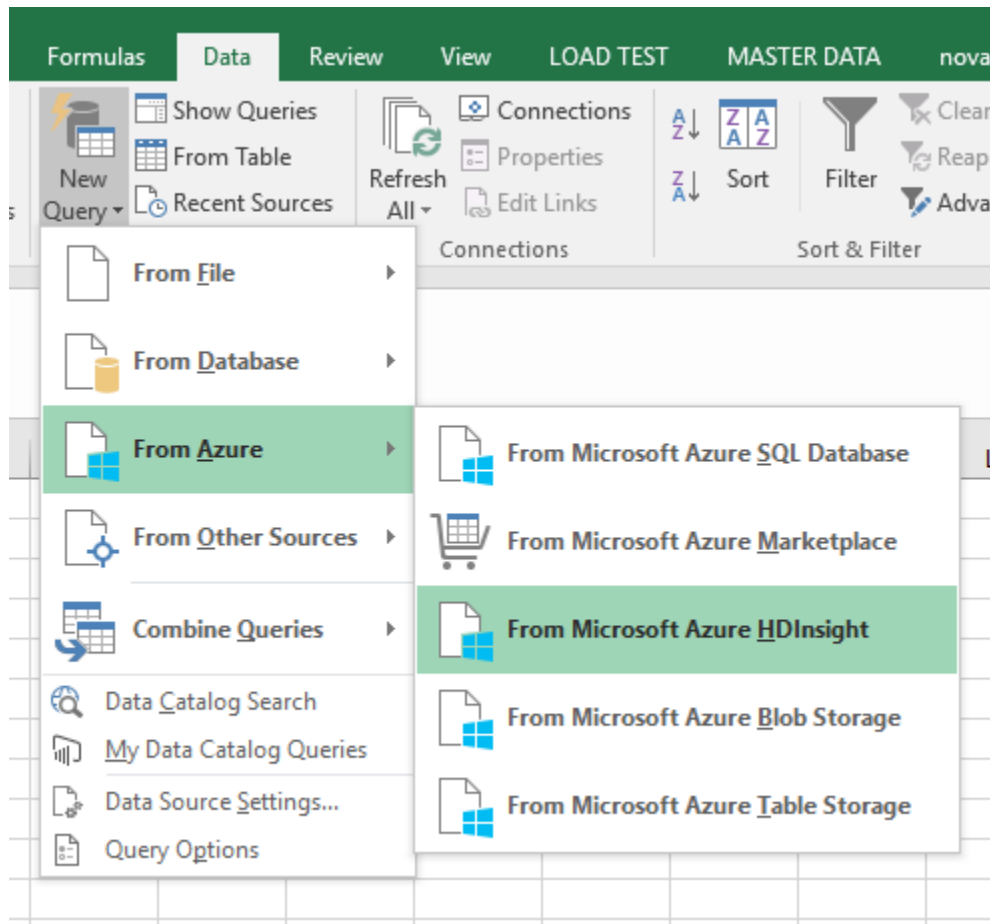
File Data Sources



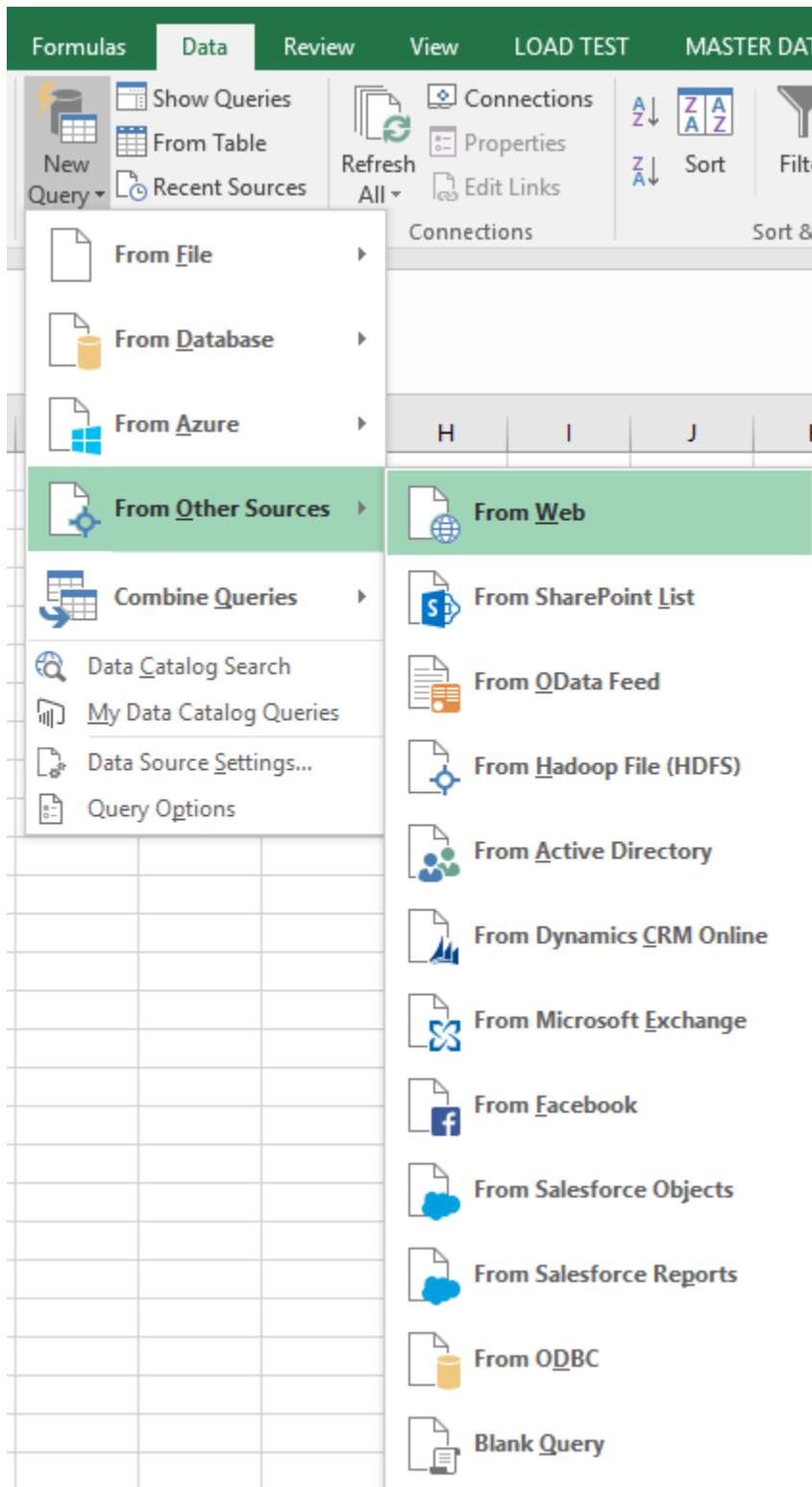
Databases



Azure

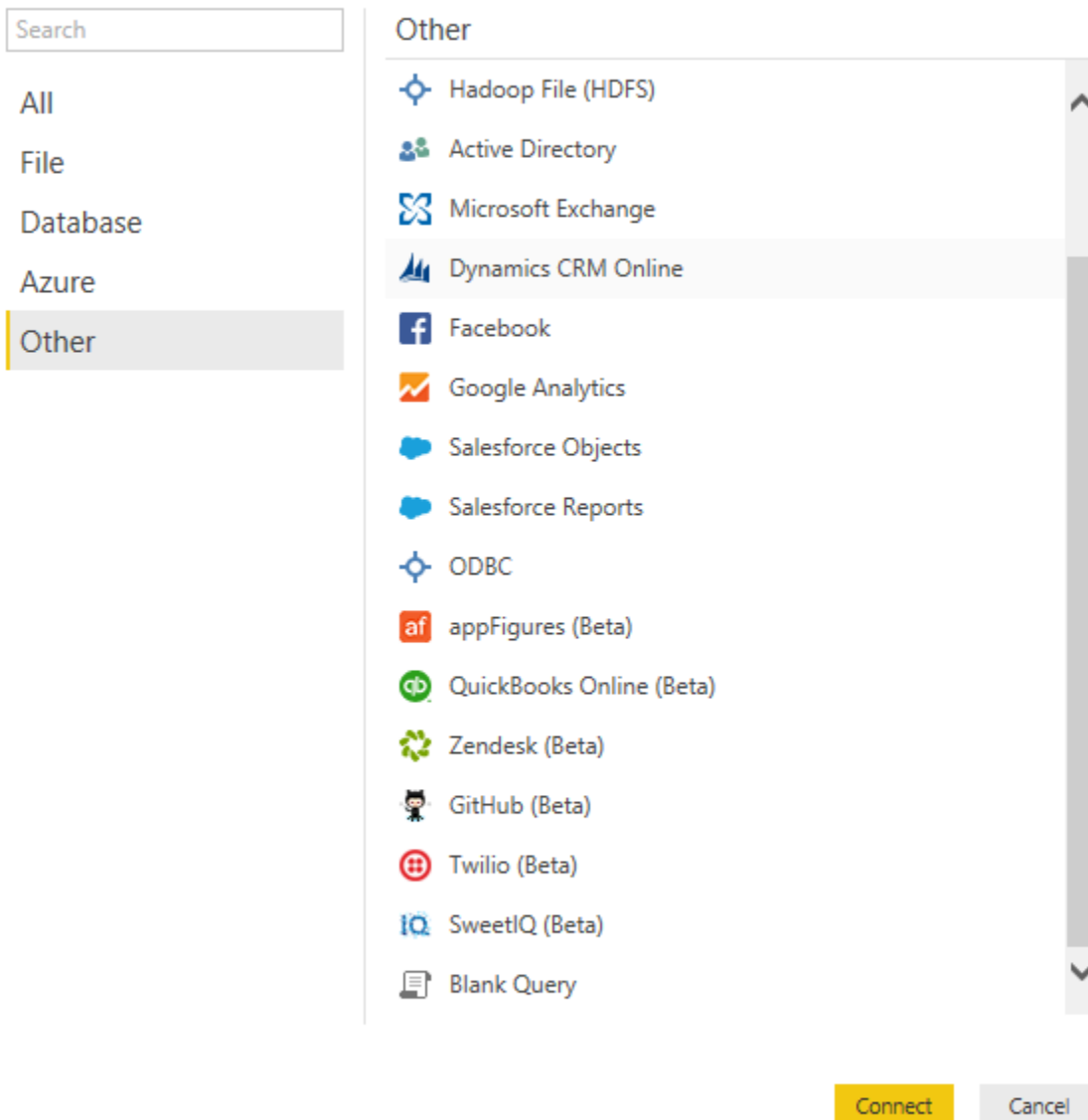


Other Sources



The Power Query version in Power BI Desktop supports some new applications that still is not implemented as Power Query for Excel, you see some of them below:

Get Data



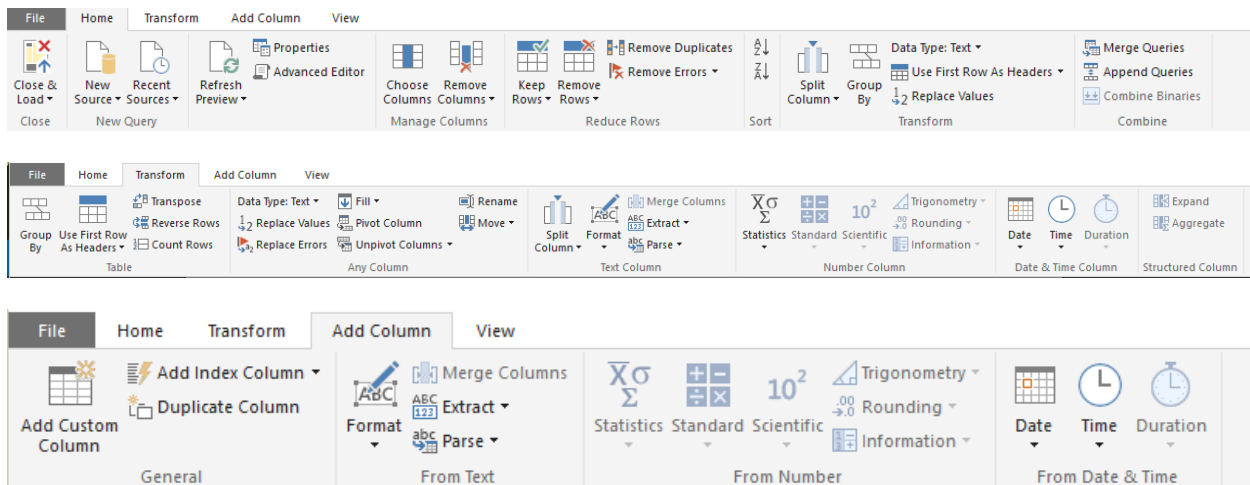
Apply Transformation In a Development Editor

Power Query look at the data values with data types such as : Table, Record, List, DateTime, Text, Number, Boolean and etc. There are many data transformation functions for any of these data types. You can apply Merge (similar to join) or Append (similar to UNION) to two tables. You can apply text functions such as getting part of a string, trimming it or length of string. You can apply mathematical functions. You can apply

datetime functions such as functions for year, Month, day and week. There are two way to apply these transformations;

1. From Query Editor: Graphical User Interface
2. From M query language: scripting language

Query Editor will give you a great experience of most common transformations through very easy user interface. you can apply most of transformation with matter of few clicks. The Query Editor in Power BI Desktop or Power Query Add-In for Excel has many common transformations listed. You can see some of them in below screenshots:

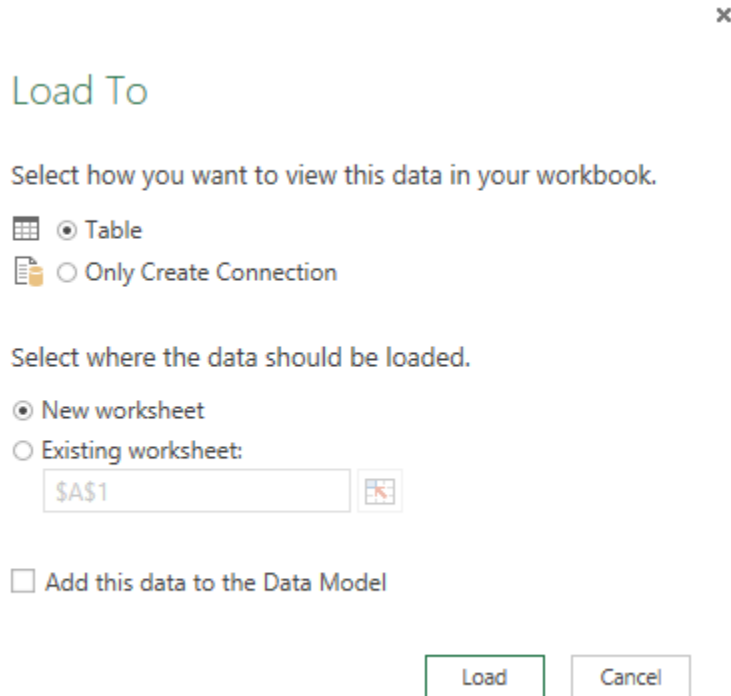


M is the formula language behind the scenes of Power Query. Everything you do in the Query Editor will be translated to an M script. M contains full list of functions that you can use. So the powerful side of Power Query is actually M. I will go deep into details of M in this book, because you would need it for solving complex challenges. M is a functional language and it has a simple structure. screenshot below shows an M Code. The details of information about M scripting will be covered in next sections.

```
let
    Source = Folder.Contents("C:\Users\Reza\Dropbox\Speaking"),
    TypeAdded=Table.AddColumn(Source,"Type",each Value.Is([Content],type table)),
    Folders=Table.SelectRows(TypeAdded, each [Type]=true),
    Sorted=Table.Sort(Folders,{"Date created", Order.Descending})
in
    Sorted
```

Load Data into Destination

You can use Power Pivot as the destination for Power Query to load result set into a data model, or you can use simple Excel spreadsheet for loading data. If you use Power BI Desktop the result set of Power Query automatically will be loaded into a model.



What Are Features of Power Query Premium?

This question might sound weird at first glance, but makes sense when you think about it that all features I mentioned above are available for free! You don't have to pay anything for it. Getting data from different sources, applying all kind of transformations to it, and loading it into a data model is all free. So now the question makes sense; What are features of Power Query Premium?

Using Data Catalog

Data Catalog is a metadata definition service that you can define data sources from your organizational data stores or from public data stores that you trust. You can define descriptors for the data structure so Power Query can search through the Data Catalog and fetch information based on it.

Sharing Queries

You can share your Power Query scripts and queries within your organization

Management using of Shared Queries

You can check the usage of queries that you've shared

As you see in above most of features for Power Query Premium is related to Office 365 usage for sharing or Power BI and Azure for data catalog and structure. Most of the features in Power Query (Essential features I have to say) is available for free!

In summary in this section you've learned about What is Power Query and what are components of it, you've learned features of Power Query, and now you are probably thinking about usage of it in scenarios and challenges that you might have right now! Good start, in next sections I will go through experience of getting data with Power Query and Power BI Desktop.

Get Started with Power Query: Movies Data Mash-Up

Published Date: September 1, 2015



As another section of the [Power BI online book: from Rookie to Rockstar](#), I would like to get start working with Power Query. From my point of view learning through an example is the best way to learn new technology. For this post I have decided to use movie's data to be mashed up. I used this example because movie's data is a fun example at the early sections of the book, you all watch movies, and you will see many familiar titles here. If you want to learn about Power Query or you need a Power Query introduction before this example, read the previous post: [What Is Power Query?](#) [Introduction to Data Mash up Engine of Power BI](#).

You can use either Power Query for Excel or Power Query as part of the Power BI Desktop for running this example. I use two data sets for this example:

1. Worldwide gross sales information of movies

This information are available in <http://www.boxofficemojo.com> website, as below:

Daily Box Office (Sun.) | Weekend Box Office (Aug. 28-30) | #1 Movie: 'Straight Outta Compton' | Showtimes

Box Office Mojo

Search Site

All Time Box Office

Social

Facebook

Twitter

Features

News

Release Sched.

Showtimes

at

Box Office

Daily

Weekend

Weekly

Monthly

Quarterly

Seasonal

Yearly

All Time

Chart Watch

International

Indices

Movies A-Z

Studios

People

WORLDWIDE GROSSES


#1-100 - #101-200 - #201-300 - #301-400 -
#401-500 - #501-600 - #601-615

Pink highlight = official revisions of older movies
Gold highlight = now playing or recent movies

Rank	Title	Studio	Worldwide	Domestic / %	Overseas / %	Year^
1	Avatar	Fox	\$2,788.0	\$760.5 27.3%	\$2,027.5 72.7%	2009^
2	Titanic	Par.	\$2,186.8	\$658.7 30.1%	\$1,528.1 69.9%	1997^
3	Jurassic World	Uni.	\$1,636.7	\$643.1 39.3%	\$993.6 60.7%	2015
4	Marvel's The Avengers	BV	\$1,519.6	\$623.4 41.0%	\$896.2 59.0%	2012
5	Furious 7	Uni.	\$1,511.7	\$351.0 23.2%	\$1,160.7 76.8%	2015
6	Avengers: Age of Ultron	BV	\$1,401.3	\$457.5 32.6%	\$943.8 67.4%	2015
7	Harry Potter and the Deathly Hallows Part 2	WB	\$1,341.5	\$381.0 28.4%	\$960.5 71.6%	2011
8	Frozen	BV	\$1,274.2	\$400.7 31.4%	\$873.5 68.6%	2013
9	Iron Man 3	BV	\$1,215.4	\$409.0 33.7%	\$806.4 66.3%	2013

1. Top 250 movies ranked by people in IMDB website

IMDB is the movie database on the internet that users can rate movies. list of top 250 movies rated by users [listed here](#) in the website as below:




All

















Movies, TV & Showtimes
 Celebs, Events & Photos
 News & Community
 Watchlist

IMDb Charts

Top 250 Movies

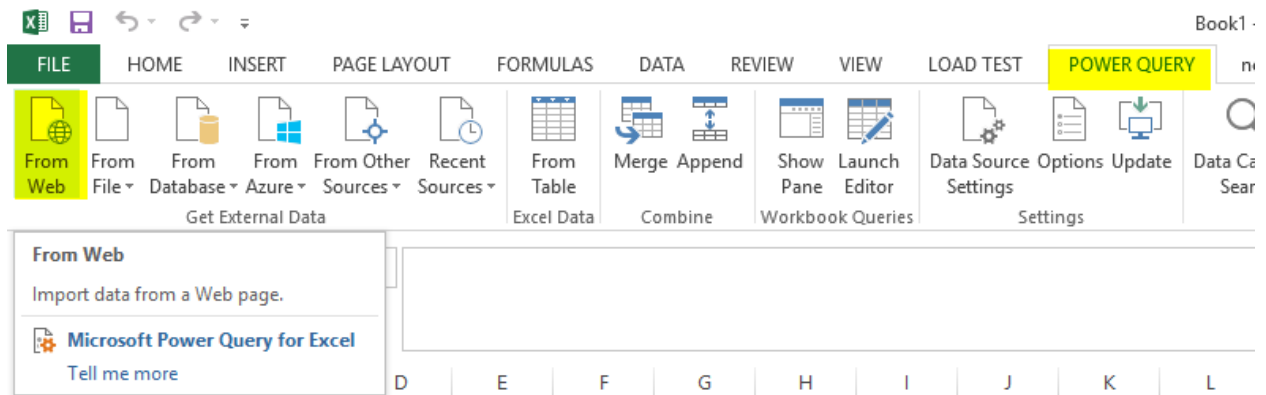
As voted by regular IMDb users

Showing 250 Titles Sort by: Ranking 

Rank & Title	IMDb Rating	Your Rating	
 1. The Shawshank Redemption (1994)	★ 9.2	☆	
 2. The Godfather (1972)	★ 9.2	☆	
 3. The Godfather: Part II (1974)	★ 9.0	☆	
 4. The Dark Knight (2008)	★ 8.9	☆	
 5. Schindler's List (1993)	★ 8.9	☆	
 6. 12 Angry Men (1957)	★ 8.9	☆	
 7. Pulp Fiction (1994)	★ 8.9	☆	
 8. The Good, the Bad and the Ugly (1966)	★ 8.9	☆	

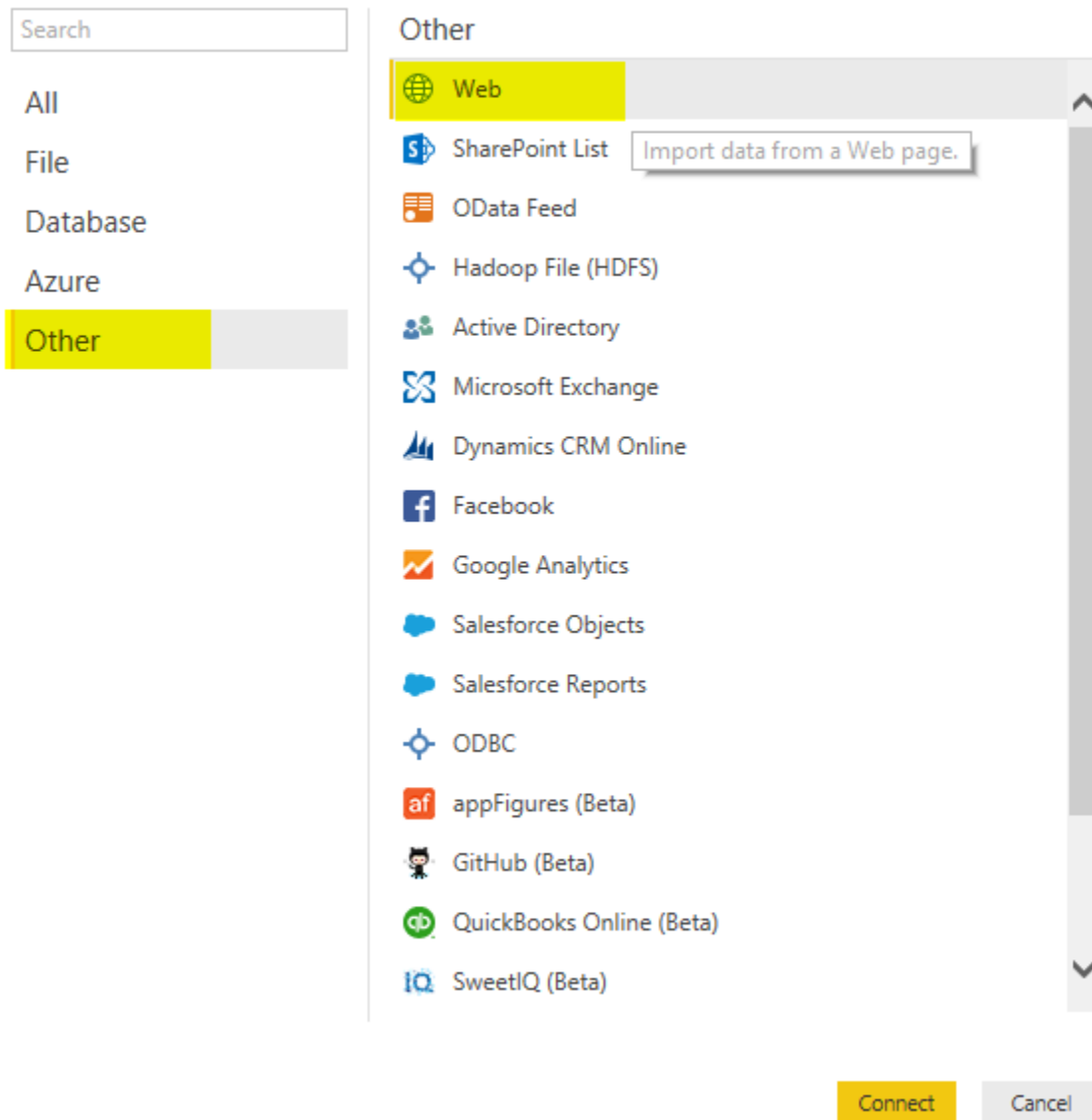
Let's Get Started

Start by getting gross sales data; Open Excel, then Power Query Tab, and then from Web;



Or Open Power BI Desktop and Get Data from Web

Get Data



Then Enter the web page URL for top 100 sold movies all the time from this link:
<http://www.boxofficemojo.com/alltime/world/>

From Web

Enter a Web page URL.

URL

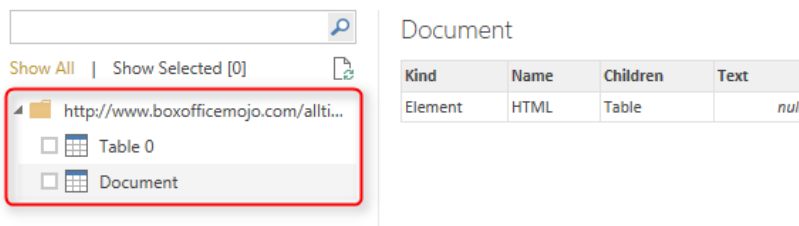
<http://www.boxofficemojo.com/alltime/world/?pagenum=1&p=.htm>

OK

Cancel

Click OK, after a quick processing, you will see a Navigator window. Power Query will check for any tables in the html web page and will come back with a list of tables on the left side under the URL address;

Navigator



Document

Kind	Name	Children	Text
Element	HTML	Table	null

Click on Table 0. You will see a preview of data in the table in main pane. Now tick the checkbox for Table 0 and click on Edit button in Navigator

Navigator

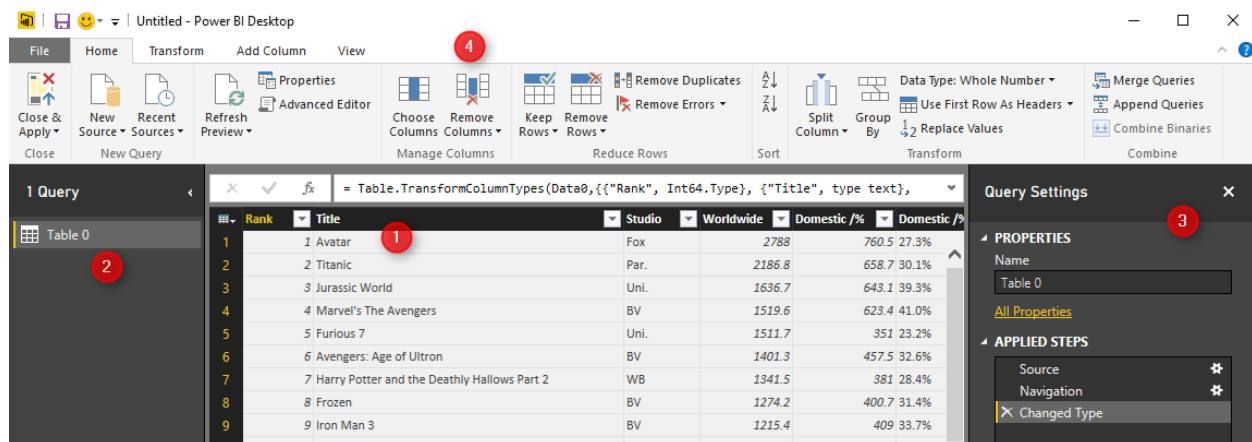
Table 0

Rank	Title	Studio	Worldwide	Domestic
1	Avatar	Fox	2788	
2	Titanic	Par.	2186.8	
3	Jurassic World	Uni.	1636.7	
4	Marvel's The Avengers	BV	1519.6	
5	Furious 7	Uni.	1511.7	
6	Avengers: Age of Ultron	BV	1401.3	
7	Harry Potter and the Deathly Hallows Part 2	WB	1341.5	
8	Frozen	BV	1274.2	
9	Iron Man 3	BV	1215.4	
10	Transformers: Dark of the Moon	P/DW	1123.8	
11	The Lord of the Rings: The Return of the King	NL	1119.9	
12	Skyfall	Sony	1108.6	
13	Transformers: Age of Extinction	Par.	1104	
14	The Dark Knight Rises	WB	1084.9	
15	Pirates of the Caribbean: Dead Man's Chest	BV	1066.2	
16	Toy Story 3	BV	1063.2	
17	Pirates of the Caribbean: On Stranger Tides	BV	1045.7	
18	Jurassic Park	Uni.	1029.2	
19	Star Wars: Episode I - The Phantom Menace	Fox	1027	
20	Alice in Wonderland (2010)	BV	1025.5	
21	The Hobbit: An Unexpected Journey	WB	1021.1	
22	Minions	Uni.	1018.9	
23	The Dark Knight	WB	1004.6	

Load Edit Cancel

Query Editor

After clicking on Edit you will see the Query Editor window opened. This is an editor that you will spend most of your time for data mash-up here.



1 Query

Table 0

Rank	Title	Studio	Worldwide	Domestic /%	Domestic /%
1	Avatar	Fox	2788	760.5	27.3%
2	Titanic	Par.	2186.8	658.7	30.1%
3	Jurassic World	Uni.	1636.7	643.1	39.3%
4	Marvel's The Avengers	BV	1519.6	623.4	41.0%
5	Furious 7	Uni.	1511.7	351	23.2%
6	Avengers: Age of Ultron	BV	1401.3	457.5	32.6%
7	Harry Potter and the Deathly Hallows Part 2	WB	1341.5	381	28.4%
8	Frozen	BV	1274.2	400.7	31.4%
9	Iron Man 3	BV	1215.4	409	33.7%
10	Transformers: Dark of the Moon	P/DW	1123.8	353.4	31.4%

Query Settings

PROPERTIES

Name: Table 0

APPLIED STEPS

Source

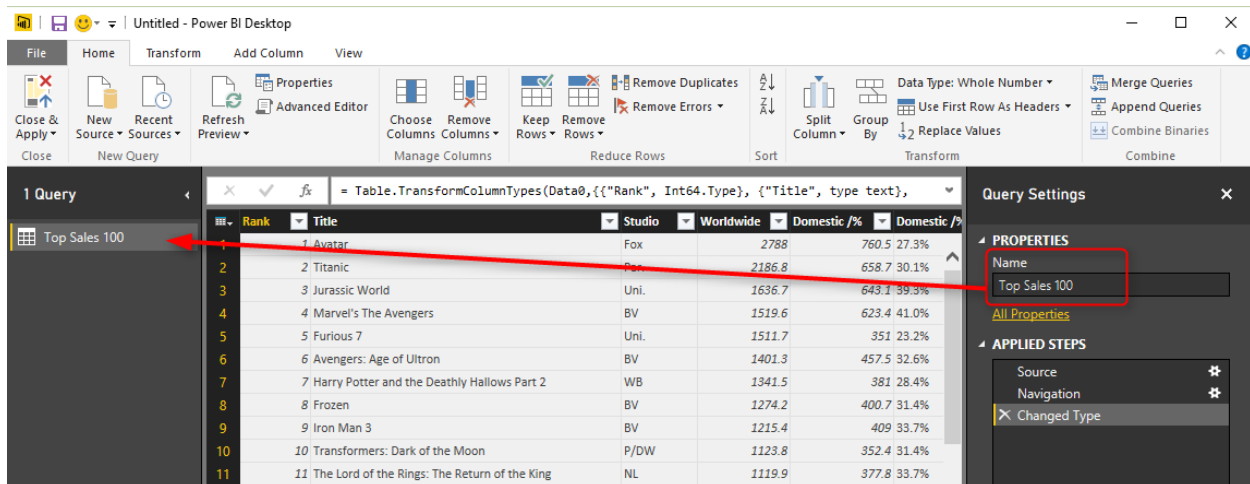
Navigation

Changed Type

Query Editor has four main sections (numbers matched to screenshot above);

1. Main data set pane; This is the central area that the result set will be displayed as preview with limited number of rows
2. List of Queries; Left hand side pane will show list of all queries in this solution or file
3. Query Settings pane; Properties such as Name of the query can be set here. Also list of all applied steps to the current query is visible in this pane.
4. Transformations Menu; Power Query has many transformations options in GUI that are available through the menu in top section

Rename the existing query to Top Sales 100



The screenshot shows the Power BI Query Editor interface. The left pane lists queries, with 'Top Sales 100' selected. The main pane displays a table of movie data. The right pane, 'Query Settings', shows the 'Name' field set to 'Top Sales 100'. A red arrow points from the 'Top Sales 100' query in the left pane to the 'Name' field in the 'Query Settings' pane.

Rank	Title	Studio	Worldwide	Domestic /%	Domestic /%
1	Avatar	Fox	2788	760.5	27.3%
2	Titanic	Par.	2186.8	658.7	30.1%
3	Jurassic World	Uni.	1636.7	643.1	39.3%
4	Marvel's The Avengers	BV	1519.6	623.4	41.0%
5	Furious 7	Uni.	1511.7	351	23.2%
6	Avengers: Age of Ultron	BV	1401.3	457.5	32.6%
7	Harry Potter and the Deathly Hallows Part 2	WB	1341.5	381	28.4%
8	Frozen	BV	1274.2	400.7	31.4%
9	Iron Man 3	BV	1215.4	409	33.7%
10	Transformers: Dark of the Moon	P/DW	1123.8	352.4	31.4%
11	The Lord of the Rings: The Return of the King	NL	1119.9	377.8	33.7%

Our goal in this example is to join the data set of global gross sales with the IMDB user rating, and then analyze to see are best sellers in movie titles among the best rated movies or not? so the more data in gross sales we get would give us better analysis. The above URL only gives us top 100 sold movies. but the option to go to pages for rest of the result set are available;

Box Office Mojo

All Time Box Office

Search Site

Search...

Social

Facebook


Twitter

Features

News

Release Sched.

Showtimes

at 

WORLDWIDE GROSSES

#1-100 - #101-200 - #201-300 - #301-400 -
#401-500 - #501-600 - #601-615

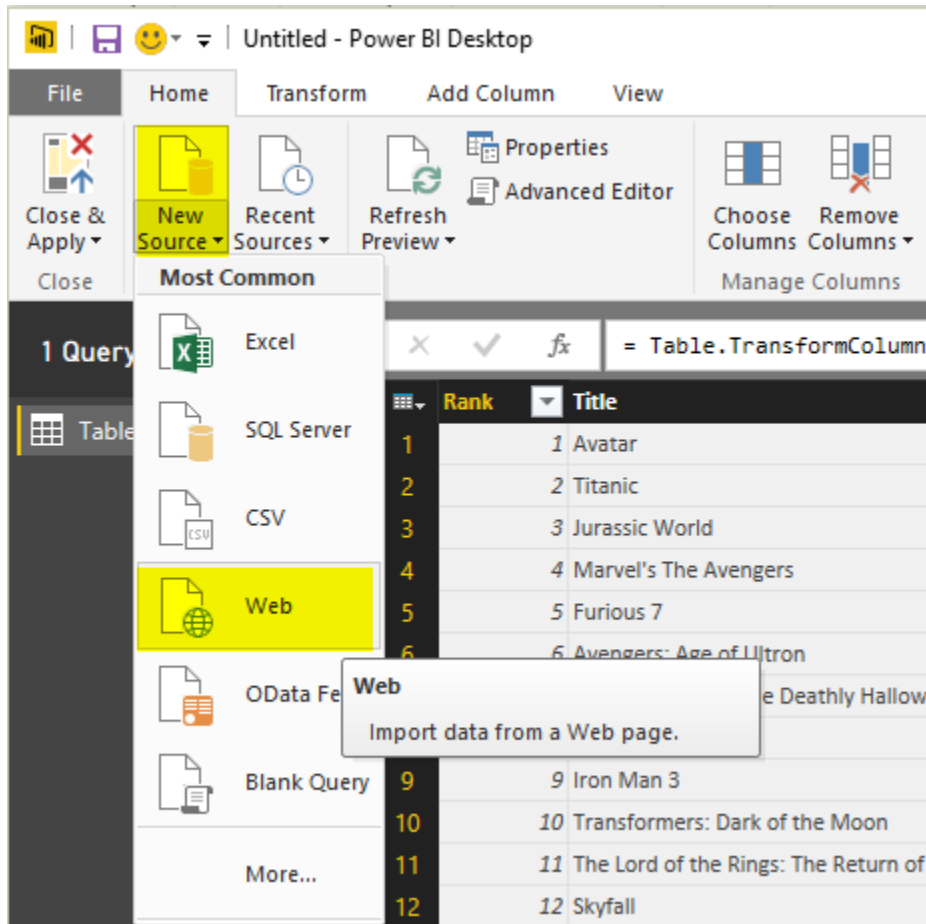
Pink highlight = official revisions of older movies
Gold highlight = now playing or recent movies

Rank	Title	Studio	Worldwide	Domestic / %	Overseas / %	Year^
1	Avatar	Fox	\$2,788.0	\$760.5 27.3%	\$2,027.5 72.7%	2009^

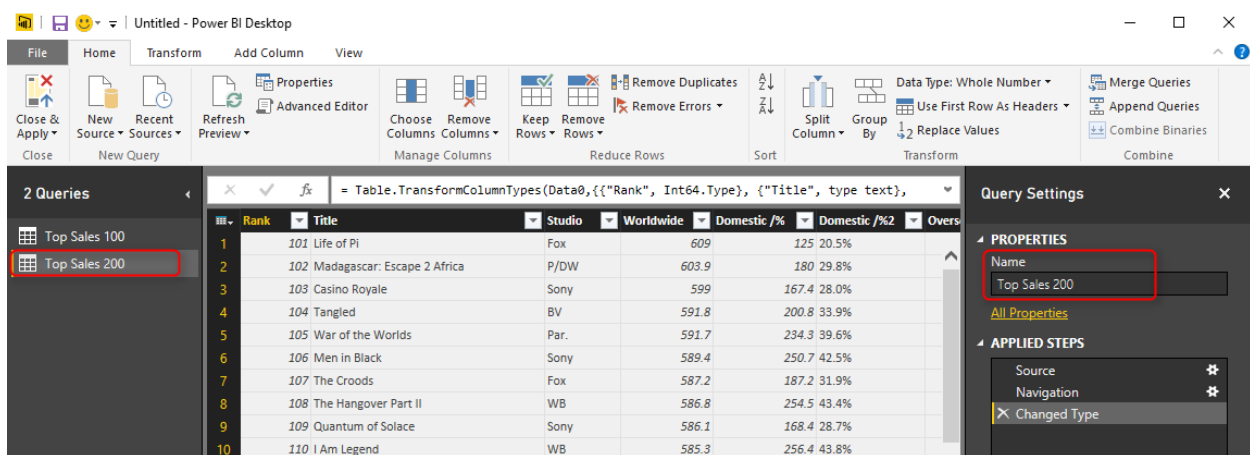
So Let's add list of movies from 101 to 200 in best sellers;

In the existing Query Editor window go to New Source, and then choose From Web.

Enter the URL as : <http://www.boxofficemojo.com/alltime/world/?pagenum=2&p=.htm>



This will lead you to the top second 100 movies sold. Click on Table 0 in navigator window and then OK. in the Query Editor rename this query as Top Sales 200



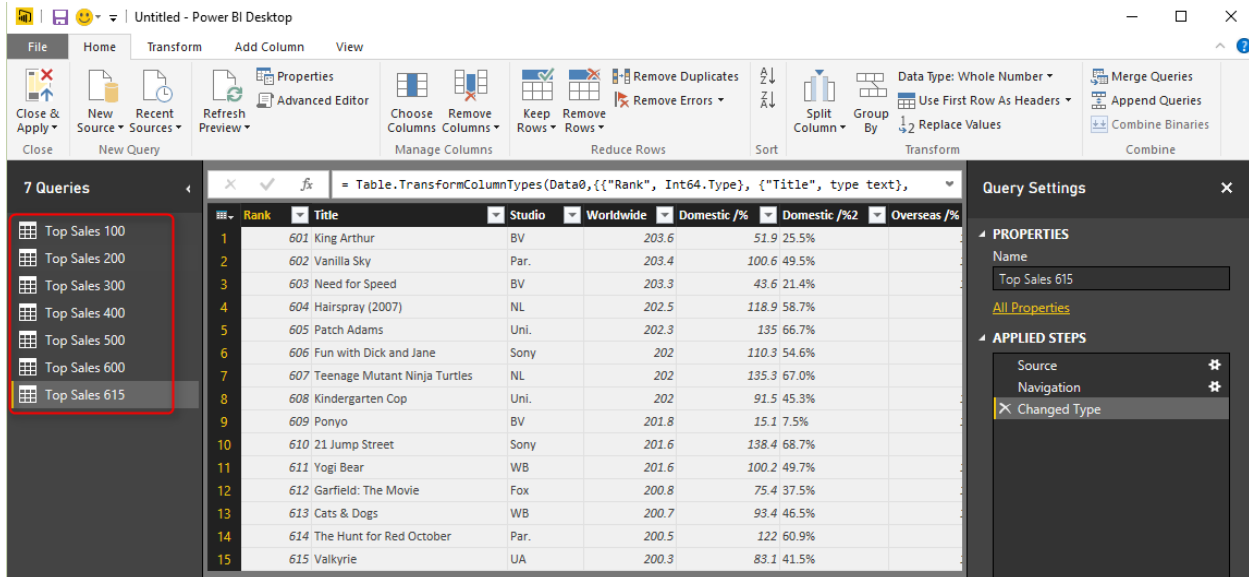
Follow this process for links below;

<http://www.boxofficemojo.com/alltime/world/?pagenum=3&p=.htm>

<http://www.boxofficemojo.com/alltime/world/?pagenum=4&p=.htm>

...

Bring data for all top 615 movies in Power Query



The screenshot shows the Power BI Desktop interface. On the left, a list of 7 queries is displayed, with 'Top Sales 615' selected. The main area shows a table with columns: Rank, Title, Studio, Worldwide, Domestic /%, Domestic /%2, and Overseas /%. The table contains 15 rows of movie data. The right pane shows the 'Query Settings' for 'Top Sales 615', with the 'PROPERTIES' tab active, showing the name 'Top Sales 615' and the 'APPLIED STEPS' list.

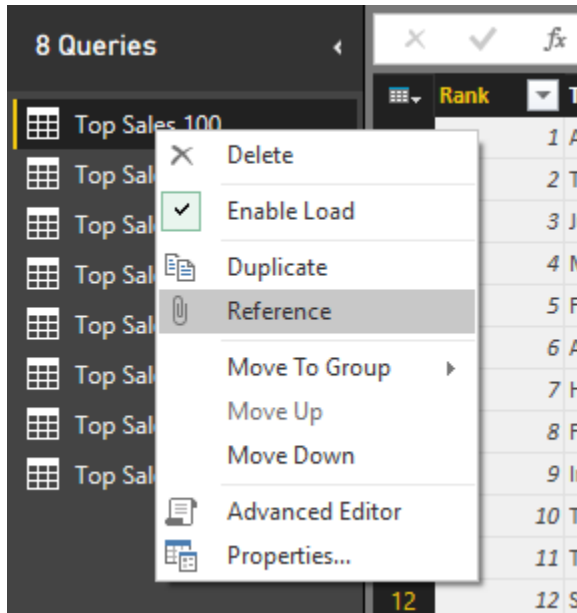
Rank	Title	Studio	Worldwide	Domestic /%	Domestic /%2	Overseas /%
1	601 King Arthur	BV	203.6	51.9	25.5%	
2	602 Vanilla Sky	Par.	203.4	100.6	49.5%	
3	603 Need for Speed	BV	203.3	43.6	21.4%	
4	604 Hairspray (2007)	NL	202.5	118.9	58.7%	
5	605 Patch Adams	Uni.	202.3	135	66.7%	
6	606 Fun with Dick and Jane	Sony	202	110.3	54.6%	
7	607 Teenage Mutant Ninja Turtles	NL	202	135.3	67.0%	
8	608 Kindergarten Cop	Uni.	202	91.5	45.3%	
9	609 Ponyo	BV	201.8	15.1	7.5%	
10	610 21 Jump Street	Sony	201.6	138.4	68.7%	
11	611 Yogi Bear	WB	201.6	100.2	49.7%	
12	612 Garfield: The Movie	Fox	200.8	75.4	37.5%	
13	613 Cats & Dogs	WB	200.7	93.4	46.5%	
14	614 The Hunt for Red October	Par.	200.5	122	60.9%	
15	615 Valkyrie	UA	200.3	83.1	41.5%	

As you see in query editor all of these queries are separate from each other. Let's combine them altogether. in database and SQL world that can be done with UNION. Here in Power Query we can do Append Queries;

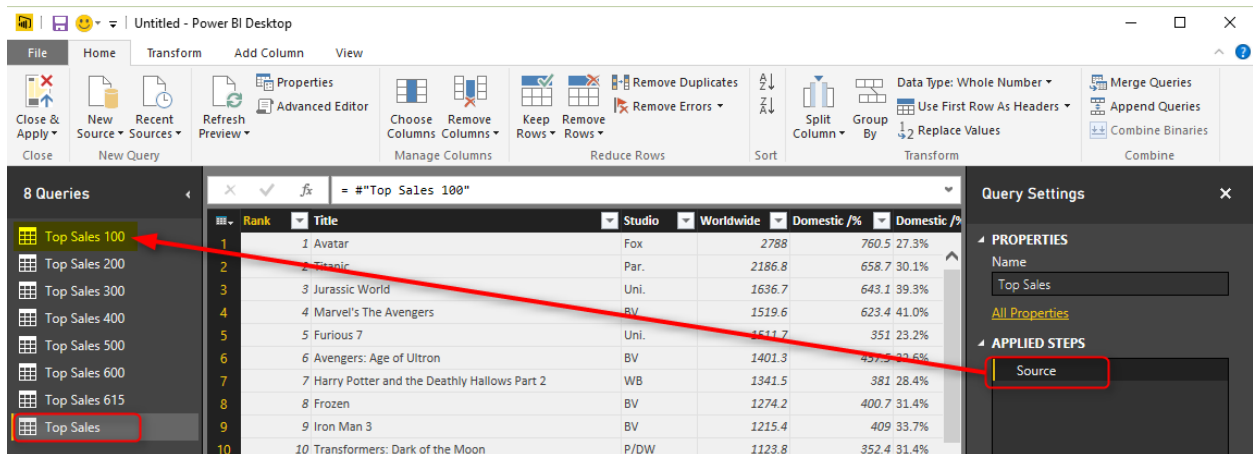
Use a Query as a Reference

First Create a reference from Top Sales 100 (because for this example I want to keep that query as is);

Right click on Top Sales 100, and from the pop up menu choose Reference



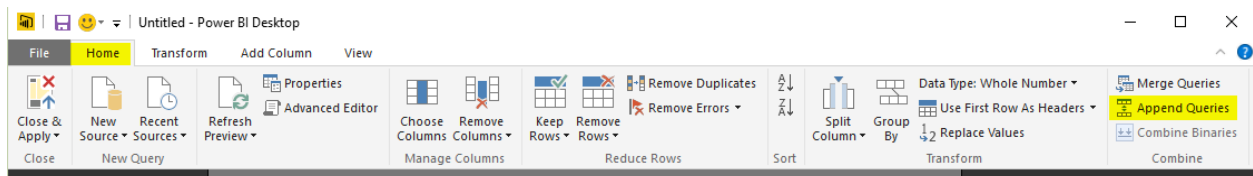
This will create a new query that uses Top Sales 100 as the source (or reference).
Rename this new query to be just "Top Sales"



Append Queries

Now let's combine queries into this new query;

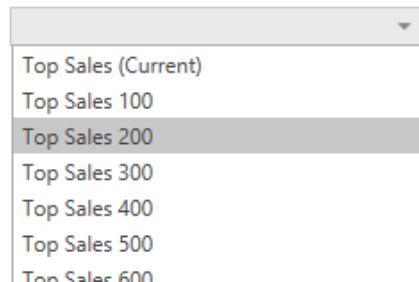
Click on Top Sales and then from the menu (Home) click on Append Queries



For append to work you need two queries; the first query is the query that you are on it (Top Sales), the second query name should be entered in the Append dialog box;

Append

Select the table to append.

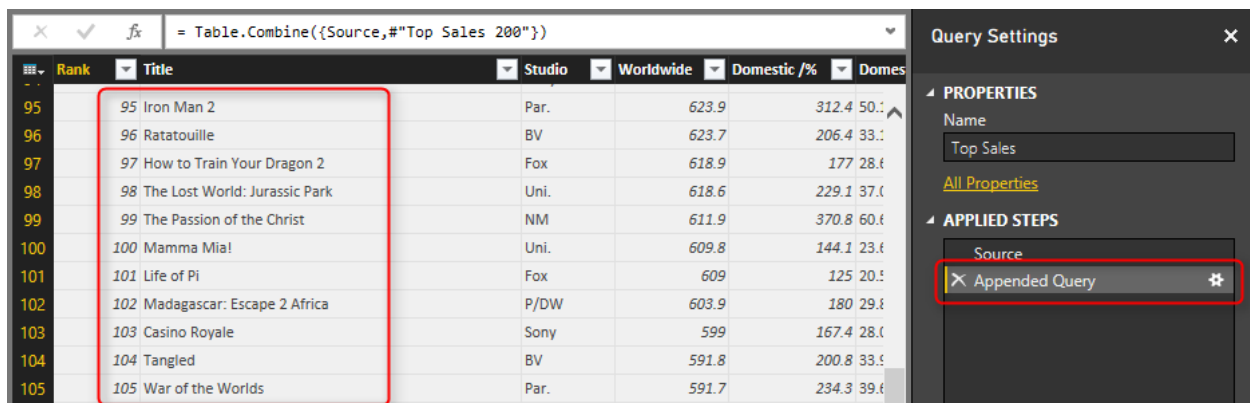


Top Sales (Current)
Top Sales 100
Top Sales 200
Top Sales 300
Top Sales 400
Top Sales 500
Top Sales 600

OK

Cancel

as you see in screenshot above you can choose other queries. For append to work best queries have to be in the same structure (number of columns, order of columns, data type of columns....). Choose Top Sales 200 in this window and click OK. This will create another step in the query setting named Appended Query. and the result set in the main pane (if you scroll down) will show you first top 200 movies sold.



The screenshot shows a table of movie data with columns: Rank, Title, Studio, Worldwide, Domestic /%, and Domes. The data is sorted by Rank, with the top 105 movies displayed. A red box highlights the first 105 rows of the table. The Query Settings pane on the right shows the 'APPLIED STEPS' section with 'Appended Query' selected, indicating that the data from the 'Top Sales 200' query has been appended to the current query.

Continue this process to add all 615 top movies into Top Sales query.

Extract First Characters

After doing this change Let's clean the Year column data; Year column has an special character in some values as below;

Overseas /%2	Year^
7.5 72.7%	2009^
8.1 69.9%	1997^
3.6 60.7%	2015
6.2 59.0%	2012
0.7 76.8%	2015
3.8 67.4%	2015
0.5 71.6%	2011
3.5 68.6%	2013
6.4 66.3%	2013
1.4 68.6%	2011
2.1 66.3%	2003^
4.2 72.5%	2012
8.6 77.8%	2014
6.8 58.7%	2012

Query Settings

PROPERTIES

Name

Top Sales

[All Properties](#)

APPLIED STEPS

Source

Appended Query

Appended Query1

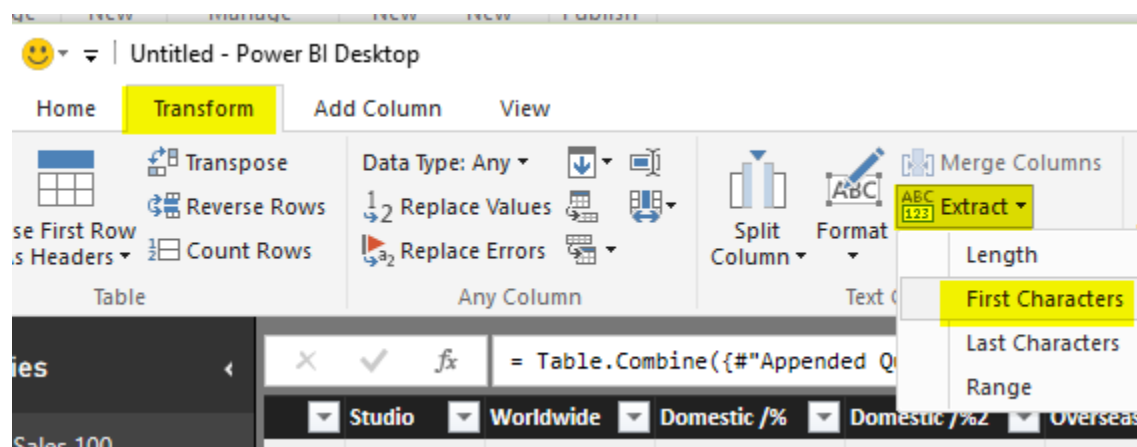
Appended Query2

Appended Query3

Appended Query4

X Appended Query5

Click on Year Column, and then from Transform menu under Text Column click on Extract, and then choose First Characters



Enter 4 in the Extract First Characters dialog box (because year isn't more than 4 characters). then click on OK.

Extract First Characters

Enter how many starting characters to keep.

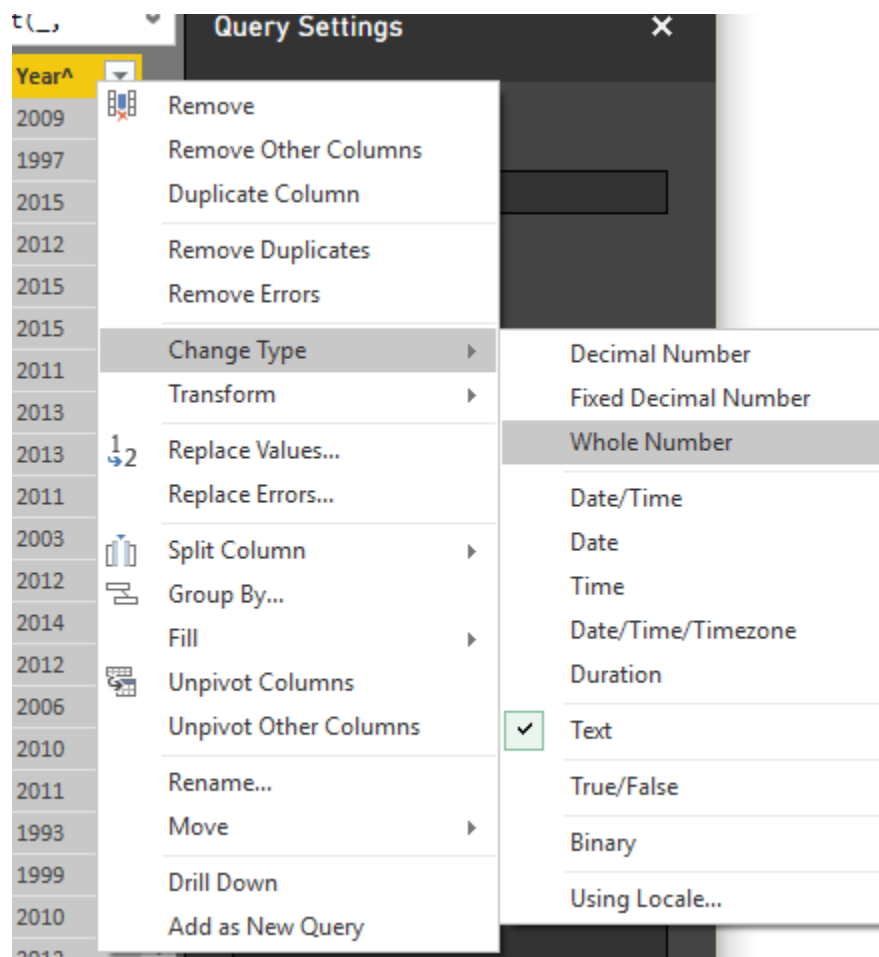
Count

OK

Cancel

You will see that year column is clean now without any extra characters. That was easy data transform, wasn't it? This option in the transformation menu (Extract First Characters) [has been added recently in Power BI Desktop](#).

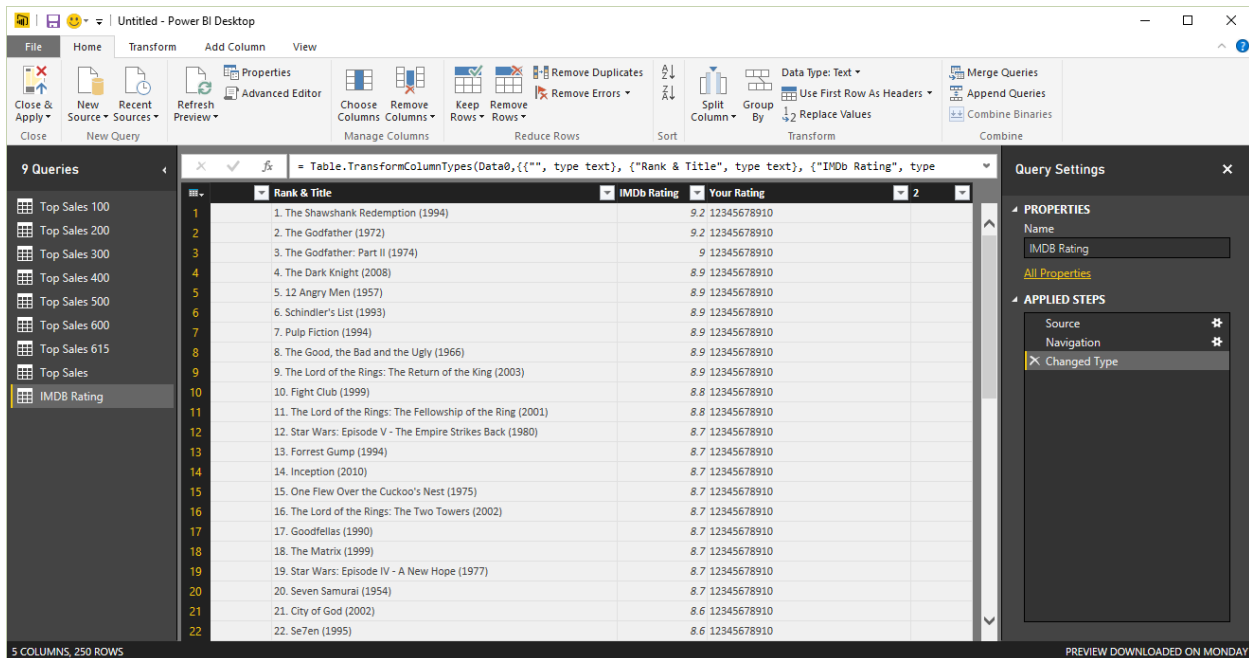
You can even now change data type of this column to whole number. Right click on Year column and then under Change Type choose Whole Number.



Great We've done enough with the first data set. Let's work on the second data set (IMDB user rating);

Go to Home Tab in Query Editor again, and Get data from the New Source and from Web. Enter the URL as: <http://www.imdb.com/chart/top>

In the Navigator window, Table 0 contains the data that we want, so load it with clicking on OK. the data loads into Query Editor as screenshot below illustrates



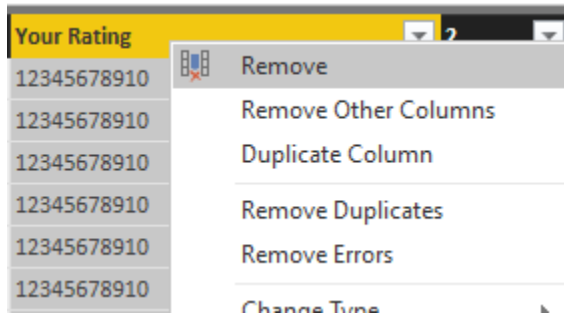
The screenshot shows the Power BI Desktop Query Editor interface. The main window displays a table with the following columns: Rank & Title, IMDB Rating, and Your Rating. The table contains 250 rows of movie data. The left sidebar shows the '9 Queries' list, with 'IMDB Rating' selected. The right sidebar shows the 'Query Settings' pane, which includes the 'PROPERTIES' section (Name: IMDB Rating) and the 'APPLIED STEPS' section (Source, Navigation, Changed Type).

Rank & Title	IMDB Rating	Your Rating
1. The Shawshank Redemption (1994)	9.2	12345678910
2. The Godfather (1972)	9.2	12345678910
3. The Godfather: Part II (1974)	9	12345678910
4. The Dark Knight (2008)	8.9	12345678910
5. 12 Angry Men (1957)	8.9	12345678910
6. Schindler's List (1993)	8.9	12345678910
7. Pulp Fiction (1994)	8.9	12345678910
8. The Good, the Bad and the Ugly (1966)	8.9	12345678910
9. The Lord of the Rings: The Return of the King (2003)	8.9	12345678910
10. Fight Club (1999)	8.8	12345678910
11. The Lord of the Rings: The Fellowship of the Ring (2001)	8.8	12345678910
12. Star Wars: Episode V - The Empire Strikes Back (1980)	8.7	12345678910
13. Forrest Gump (1994)	8.7	12345678910
14. Inception (2010)	8.7	12345678910
15. One Flew Over the Cuckoo's Nest (1975)	8.7	12345678910
16. The Lord of the Rings: The Two Towers (2002)	8.7	12345678910
17. Goodfellas (1990)	8.7	12345678910
18. The Matrix (1999)	8.7	12345678910
19. Star Wars: Episode IV - A New Hope (1977)	8.7	12345678910
20. Seven Samurai (1954)	8.7	12345678910
21. City of God (2002)	8.6	12345678910
22. Se7en (1995)	8.6	12345678910

Rename the query to IMDB Rating.

Remove Columns

You can see that there are three useless columns in the data set; first column, and last two columns. remove these columns simply by clicking on them and then right click and Remove.



Split Column

Now in the result set we have two columns; Rank & Title, and IMDb Rating. Rank & Title is a combined column which contains rank, title, and year of the movie. Let's split these values;

	Rank & Title	IMDb Rating
1	1. The Shawshank Redemption (1994)	9.2
2	2. The Godfather (1972)	9.2
3	3. The Godfather: Part II (1974)	9
4	4. The Dark Knight (2008)	8.9
5	5. 12 Angry Men (1957)	8.9
6	6. Schindler's List (1993)	8.9
7	7. Pulp Fiction (1994)	8.9
8	8. The Good, the Bad and the Ugly (1966)	8.9
9	9. The Lord of the Rings: The Return of the King (2003)	8.9
10	10. Fight Club (1999)	8.8
11	11. The Lord of the Rings: The Fellowship of the Ring (2001)	8.8
12	12. Star Wars: Episode V - The Empire Strikes Back (1980)	8.7
13	13. Forrest Gump (1994)	8.7

Rank is separated by single dot (.). So we can use Split Column transformation to split it easily; Right click on Rank & Title column first. then Choose Split Column, and then By Delimiter

fx = Table.RemoveColumns(#"Changed Type",{ "Your Rating", "2", ""})

	Rank & Title	IMDb Rating
1	1. The Shawshank Redemption (1994)	9.2
2	2. The Godfather (1972)	9.2
3	3. The Godfather: Part II (1974)	9
4	4. The Dark Knight (2008)	8.9
5	5. 12 Angry Men (1957)	8.9
6	6. Schindler's List (1993)	8.9
7	7. Pulp Fiction (1994)	8.9
8	8. The Good, the Bad and the Ugly (1966)	8.9
9	9. The Lord of the Rings: The Return of the King (2003)	8.9
10	10. Fight Club (1999)	8.8
11	11. The Lord of the Rings: The Fellowship of the Ring (2001)	
12	12. Star Wars: Episode V - The Empire Strikes Back (1980)	8.7
13	13. Forrest Gump (1994)	8.7
14	14. Inception (2010)	8.7
15	15. One Flew Over the Cuckoo's Nest (1975)	8.7

In the Split Column by Delimiter dialog box you can choose one of the common delimiters such as comma or color ... or you can use a custom delimiter. Set it to Custom, and enter single dot (.) in the box underneath. You can also specify how the split works. The default option is At each occurrence of the delimiter. This default option might not be best for our case, because sometimes there might be a dot in the movie's title. So select the split method as At the left-most delimiter. This option will scan text from the left, and will stop splitting after finding the first delimiter.

Split Column by Delimiter

Specify the delimiter used to split the text column.

Select or enter delimiter

--Custom--

.

Split

☒ At the left-most delimiter

☐ At the right-most delimiter

☐ At each occurrence of the delimiter

Advanced options

OK

Cancel

after split the result set would look like below;

	Rank & Title.1	Rank & Title.2	IMDb Rating
1	1	The Shawshank Redemption (1994)	9.2
2	2	The Godfather (1972)	9.2
3	3	The Godfather: Part II (1974)	9
4	4	The Dark Knight (2008)	8.9
5	5	12 Angry Men (1957)	8.9
6	6	Schindler's List (1993)	8.9
7	7	Pulp Fiction (1994)	8.9
8	8	The Good, the Bad and the Ugly (1966)	8.9
9	9	The Lord of the Rings: The Return of the King (2003)	8.9
10	10	Fight Club (1999)	8.8

Rename the Rank & Title.1 column to Rank.

Now Let's split title and year. Year value is surrounded between brackets, so we can use same split column method, this time using open bracket as below;

Split Column by Delimiter

Specify the delimiter used to split the text column.

Select or enter delimiter
--Custom--
(
Split
☐ At the left-most delimiter
☒ At the right-most delimiter
☐ At each occurrence of the delimiter

Advanced options

OK

Cancel

The result set looks like below screenshot;

Rank	Rank & Title.2.1	Rank & Title.2.2	IMDb Rating
1	1 The Shawshank Redemption	1994)	9.2
2	2 The Godfather	1972)	9.2
3	3 The Godfather: Part II	1974)	9
4	4 The Dark Knight	2008)	8.9
5	5 12 Angry Men	1957)	8.9
6	6 Schindler's List	1993)	8.9
7	7 Pulp Fiction	1994)	8.9
8	8 The Good, the Bad and the Ugly	1966)	8.9
9	9 The Lord of the Rings: The Return of the King	2003)	8.9
10	10 Fight Club	1999)	8.8
11	11 The Lord of the Rings: The Fellowship of the Ring	2001)	8.8

Replace Values

Rank & Title.2.2 column has the year value with an extra close bracket. Click on this column and then from Transform menu under Any Column click on Replace Values

Transform | Add Column | View

Transpose | Reverse Rows | Count Rows | Data Type: Text | Fill | Replace Values | Replace Errors | Pivot Column | Unpivot Columns | Rename | Move | Split Column | Format | Merge Columns | Extract | Parse | Statistics | Standard S

Replace Values

Replace existing values in the currently selected columns with the specified new value.

Rank	Rank & Title.2.2	IMDb Rating
1	1994)	9.2
2	2 The Godfather 1972)	9.2
3	3 The Godfather: Part II 1974)	9
4	4 The Dark Knight 2008)	8.9
5	5 12 Angry Men 1957)	8.9
6	6 Schindler's List 1993)	8.9
7	7 Pulp Fiction 1994)	8.9
8	8 The Good, the Bad and the Ugly 1966)	8.9
9	9 The Lord of the Rings: The Return of the King 2003)	8.9

Replace close bracket with an empty string as below;

Replace Values

Replace one value with another in the selected columns.

Value To Find

Replace With

☐ Match entire cell contents

OK

Cancel

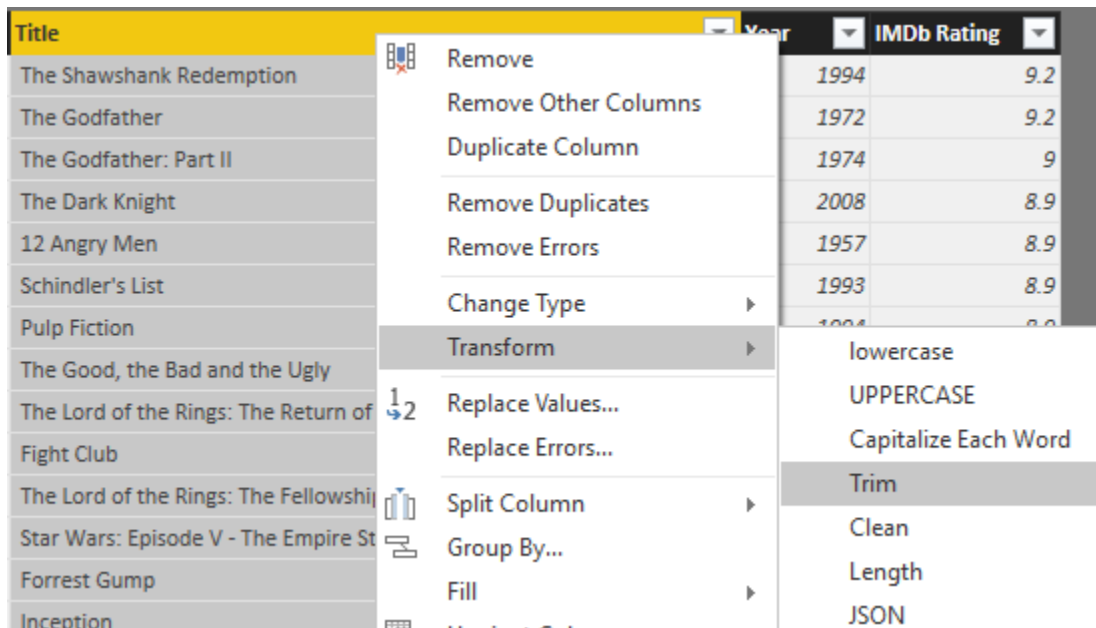
Result set would have the close bracket removed. rename the column to Year, and change it's data type to Whole number (change data type with right click on the column)

Rank	Rank & Title.2.1	Year	IMDb Rating
1	1 The Shawshank Redemption	1994	9.2
2	2 The Godfather	1972	9.2
3	3 The Godfather: Part II	1974	9
4	4 The Dark Knight	2008	8.9
5	5 12 Angry Men	1957	8.9
6	6 Schindler's List	1993	8.9

Trim

Also rename the Rank & Title.2.1 column to Title. Because this column might have extra spaces at the beginning and end of values (as the result of split column steps), let's remove extra spaces;

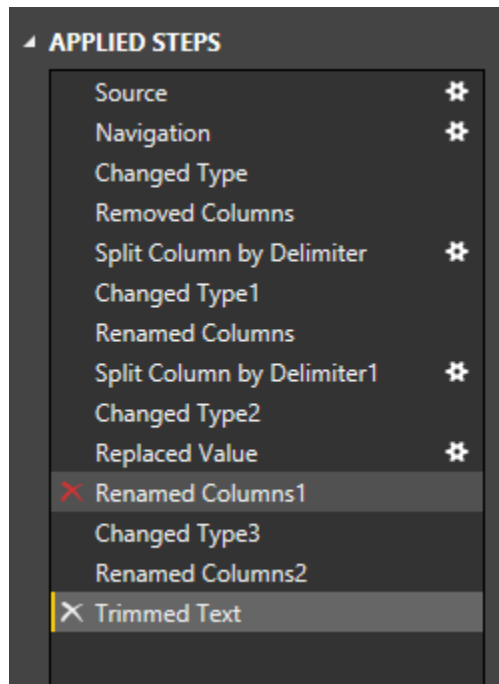
right click on this column and then under transform choose Trim. this will remove all heading and trailing spaces from values in this column.



Awesome, our work with this data set has been finished as well.

Applied Steps

One of the most useful sections of Query Editor window is Applied Steps in the Query Settings Pane. This section of Query Editor window is very useful for debugging and tracking steps and changes. You can see all the steps that you've applied on the current data set in this pane. And this is not all of it! You can click on an step and the main pane will show you the data at that step! such an awesome way of keeping track of steps.

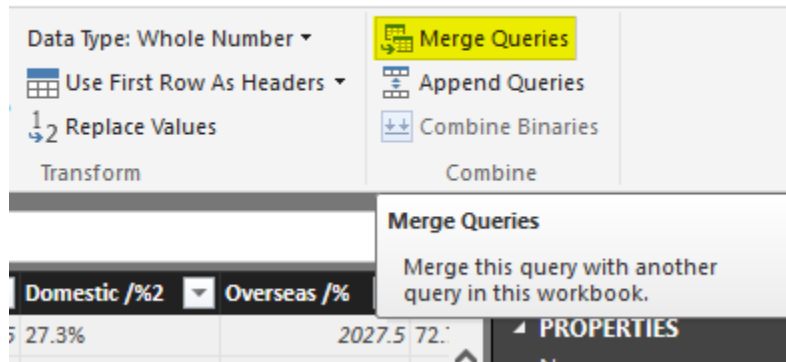


You can even remove an step, or you can change settings of an step with clicking on remove icon (on the left side of step) or setting icon (on the right side of the step, but only for steps that settings is applicable to them).

Final Merge

We've prepared both data sets for a final merge together to see how best selling movies are among top user rated films. So we are one step away from this result. we have to merge these two data sets, or Join them in other word.

Click on Top Sales query and create a reference of it, name the new query as Merge Result. Now click on Merge Result, and then from Home tab, under Combine choose Merge Queries



Merge Queries is equivalent to Join in SQL or database terminology.

Merging queries requires two queries; the first query is the query that you are currently on it (Merge Result), and you can choose the second query in the Merge dialog box. Choose second query as IMDB Rating. Now select joining keys as Title (you can also choose multiple joining columns with pressing ctrl keyboard key). Set also join kind to Left outer join (this will only select all records from the first query with matched rows of that from the second query)

Merge

Select a table and matching columns to create a merged table.


Rank	Title	Studio	Worldwide	Domestic /%	Domestic /%2	Overseas /%	Overseas /%
1	Avatar	Fox	2788	760.5	27.3%	2027.5	72.7%
2	Titanic	Par.	2186.8	658.7	30.1%	1528.1	69.9%
3	Jurassic World	Uni.	1636.7	643.1	39.3%	993.6	60.7%
4	Marvel's The Avengers	BV	1519.6	623.4	41.0%	896.2	59.0%
5	Furious 7	Uni.	1511.7	351	23.2%	1160.7	76.8%

IMDB Rating

Rank	Title	Year	IMDb Rating
1	The Shawshank Redemption	1994	9.2
2	The Godfather	1972	9.2
3	The Godfather: Part II	1974	9
4	The Dark Knight	2008	8.9
5	12 Angry Men	1957	8.9

Join Kind

Inner (only matching rows)

 The selection has matched 58 out of the first 615 rows.

OK

Cancel

Notice in screenshot above that merge dialog mentioned only 58 records out of 615 movies matched! it means only 58 of best seller movies are among top user rated list! such a pity. Screenshot showed only Inner Join result, but you choose Left Outer and then click on OK to look at the data;

Joining experience in Power Query is a bit different from database tables. as a result of join you will get first table with a new column for the new table. This new column holds table values which needs to be expanded. If you click on the column header icon, you can choose which columns of the nested table you want to expand.

fx = Table.NestedJoin(Source,{"Title"},#"IMDb Rating",{"Title"},"NewColumn",JoinKind.Inner)

	Studio	Worldwide	Domestic /%	Domestic /%2	Overseas /%	Overseas /%2	Year^	NewColumn
Return of the King	WB	1004.6	534.9	53.2%				
Fellowship of the Ring	NL	1119.9	377.8	33.7%				
	Par.	677.9	330.3	48.7%				
	WB	1084.9	448.1	41.3%				
	WB	825.5	292.6	35.4%				
Two Towers	BV	1063.2	415	39.0%				
	Uni.	1029.2	402.5	39.1%				
	WB	463.5	171.5	37.0%				
	BV	987.5	422.8	42.8%				
	Par.	675	188	27.9%				
	DW	481.8	216.5	44.9%	265.3	55.1%	1998	Table
	BV	936.7	380.8	40.7%	555.9	59.3%	2003	Table
	Par.	389.9	248.2	63.6%	141.8	36.4%	1981	Table
y	TriS	519.8	204.8	39.4%	315	60.6%	1991	Table

Search Columns to Expand

☒ Expand ☐ Aggregate

☒ (Select All Columns)

☒ Rank

☒ Title

☒ Year

☒ IMDb Rating

☒ Use original column name as prefix

OK Cancel

Let's keep all columns and click OK. You can now see some movies that are among best sellers but not in top 250 user rated list of IMDB, There are movies name ssuch as Iron Man 3, Skyfall, Furious 7 and list goes on. Play with that yourself to see what you explore!

Rank	Title	Studio	Worldwide	NewColumn.Rank	NewColumn.Title	NewColumn.Year	NewColumn.IMDb Rati...
1	23 The Dark Knight	WB	1004.6	4 The Dark Knight		2008	8.9
2	11 The Lord of the Rings: The Return of the King	NL	1119.9	9 The Lord of the Rings: The Return of the King		2003	8.9
3	41 The Lord of the Rings: The Fellowship of the Ring	NL	871.5	11 The Lord of the Rings: The Fellowship of the Ring		2001	8.8
4	80 Forrest Gump	Par.	677.9	13 Forrest Gump		1994	8.7
5	14 The Dark Knight Rises	WB	1084.9	61 The Dark Knight Rises		2012	8.4
6	46 Inception	WB	825.5	14 Inception		2010	8.7
7	16 Toy Story 3	BV	1063.2	78 Toy Story 3		2010	8.3
8	34 The Lord of the Rings: The Two Towers	NL	926	16 The Lord of the Rings: The Two Towers		2002	8.7
9	24 The Lion King	BV	987.5	54 The Lion King		1994	8.4
10	81 Interstellar	Par.	675	29 Interstellar		2014	8.6
11	76 Inside Out	BV	702.5	53 Inside Out		2015	8.4
12	68 Up	BV	731.3	114 Up		2009	8.2
13	82 The Sixth Sense	BV	672.8	159 The Sixth Sense		1999	8.1
14	32 Finding Nemo	BV	936.7	161 Finding Nemo		2003	8.1
15	18 Jurassic Park	Uni.	1029.2	203 Jurassic Park		1993	8
16	56 Guardians of the Galaxy	BV	774.2	210 Guardians of the Galaxy		2014	8
17	87 Pirates of the Caribbean: The Curse of the Black Pearl	BV	654.3	228 Pirates of the Caribbean: The Curse of the Black Pearl		2003	8
18	63 X-Men: Days of Future Past	Fox	748.1	240 X-Men: Days of Future Past		2014	8
19	1 Avatar	Fox	2788			null	null
20	2 Titanic	Par.	2186.8			null	null
21	3 Jurassic World	Uni.	1636.7			null	null
22	4 Marvel's The Avengers	BV	1519.6			null	null
23	5 Furious 7	Uni.	1511.7			null	null
24	6 Avengers: Age of Ultron	BV	1401.3			null	null
25	7 Harry Potter and the Deathly Hallows Part 2	WB	1341.5			null	null
26	8 Frozen	BV	1274.2			null	null
27	9 Iron Man 3	BV	1215.4			null	null
28	10 Transformers: Dark of the Moon	P/DW	1123.8			null	null
29	12 Skyfall	Sony	1108.6			null	null
30	13 Transformers: Age of Extinction	Par.	1104			null	null
31	15 Pirates of the Caribbean: Dead Man's Chest	BV	1066.2			null	null
32	17 Pirates of the Caribbean: On Stranger Tides	BV	1045.7			null	null
33	19 Star Wars: Episode I - The Phantom Menace	Fox	1027			null	null
34	20 Alice in Wonderland (2010)	BV	1025.5			null	null
35	21 The Hobbit: An Unexpected Journey	WB	1021.1			null	null

Summary

In this section you've learned basics of Power Query through an example. You've seen how Power Query can analyze tables in a web page and load it into query editor. You've experienced Query Editor and you've learned how to apply some transformations. You've learned that transformations such as split column, replace values, change data type, and extract part of a text are easy transformations that can be simply done through Power Query editor. In next sections I will explain different types of data sources that Power Query or Power BI can work with through the Get Data Experience. You will see that Power Query and Power BI can get data from text files such as CSV, Text as well as database connections such as MySQL, Oracle and SQL Server, it can also bring data from on-premises data stores as well as cloud Azure based services.

Power BI Get Data from Excel: Everything You Need to Know

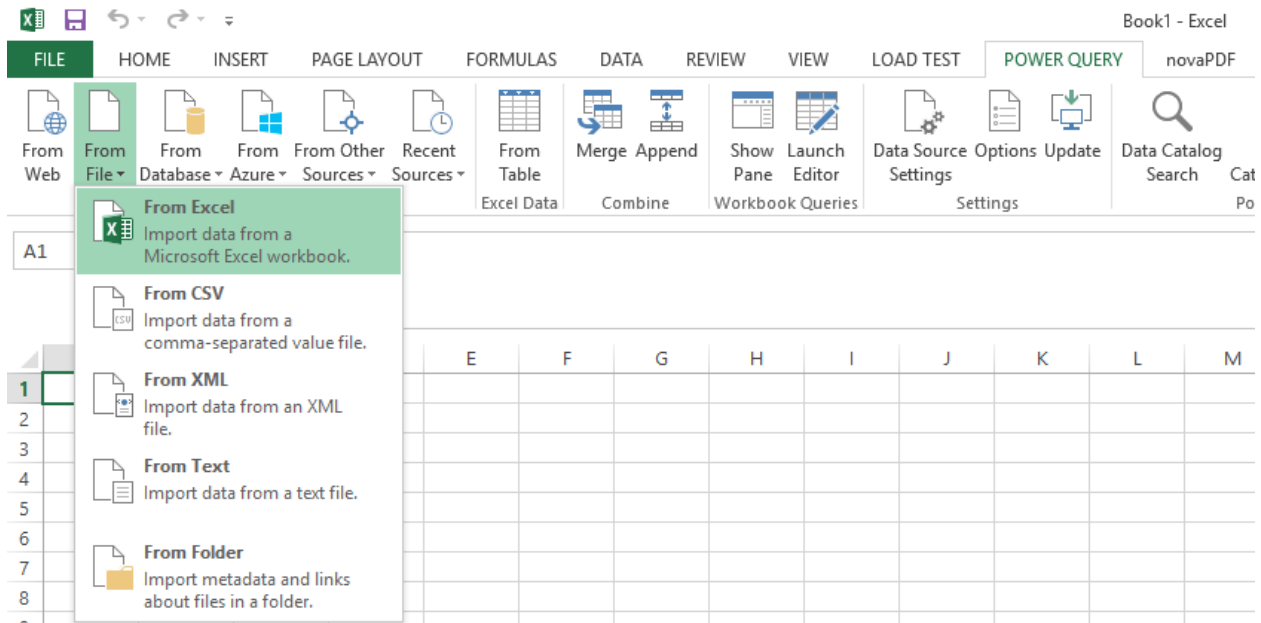
Published Date: September 2, 2015



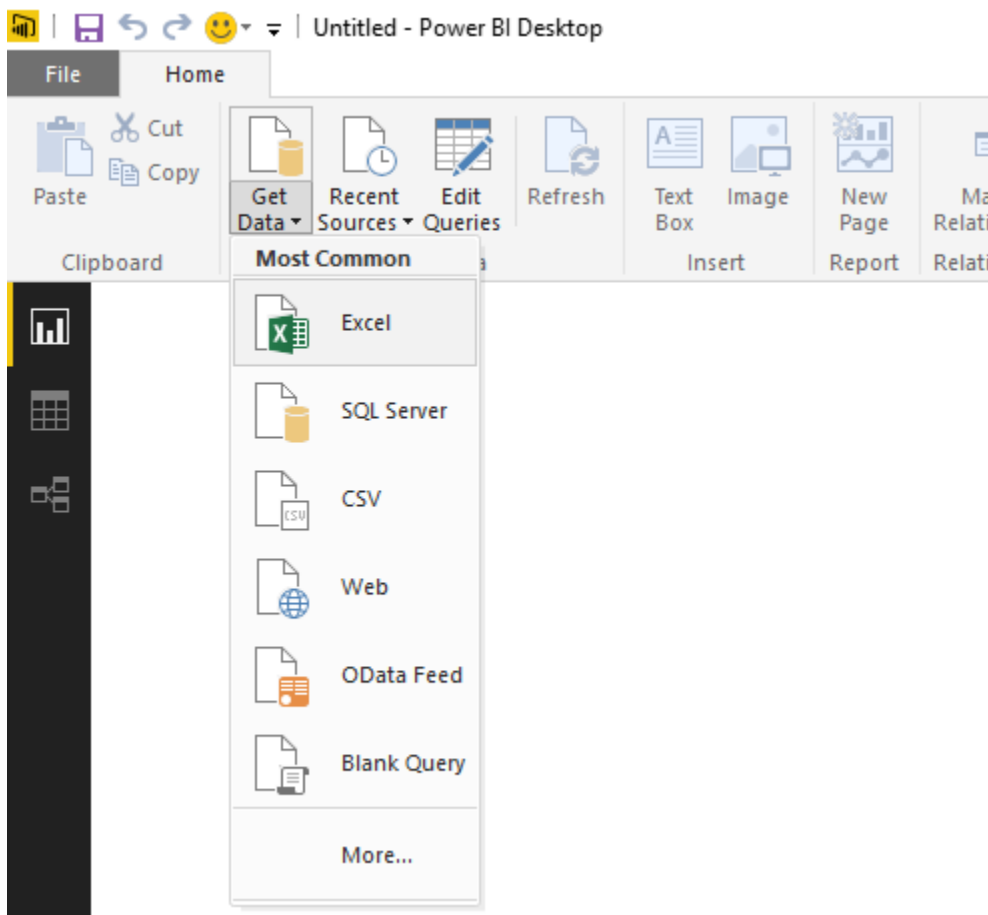
In [Previous section](#) you learned about Power Query through an example of data mash-up of movies. Also prior than that you've learned about [Power BI](#) and its components in [Power BI online book from rookie to rockstar](#). In this section I would like to start exploration of different data sources in Power BI, and I want to start that with an Excel source. Excel source seems to be an easy one, but in the other hand it is one of the most common sources of the data. In this section I want to share some tips about Excel data source and then I show an example of working with Olympic data source in an Excel file.

Excel Data Source

Power Query or Power BI can connect to many data sources, one of the supported data sources in Excel. Power Query for Excel get data from Excel in this way:

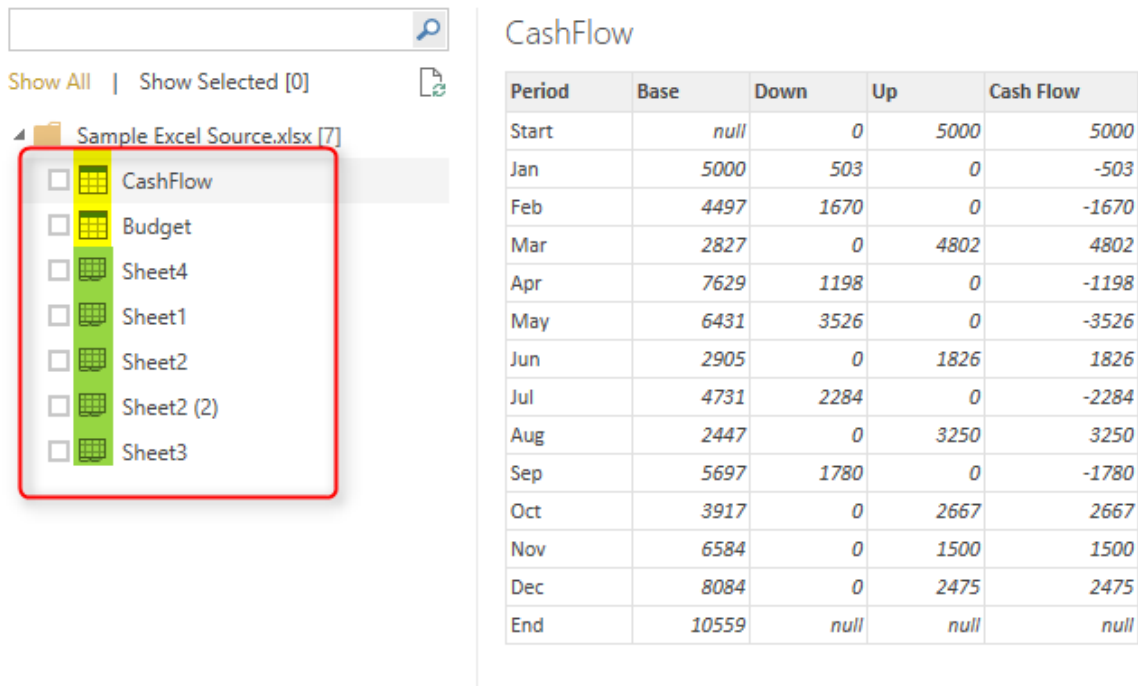


Power BI Desktop connects to Excel through Get Data experience



For getting data from excel you just need to specify path of the file. After specifying the file, Power Query will analyze contents of the file and distinguish all sheets and tables in the file and list them in the Navigator dialog box as preview;

Navigator



CashFlow

Period	Base	Down	Up	Cash Flow
Start	null	0	5000	5000
Jan	5000	503	0	-503
Feb	4497	1670	0	-1670
Mar	2827	0	4802	4802
Apr	7629	1198	0	-1198
May	6431	3526	0	-3526
Jun	2905	0	1826	1826
Jul	4731	2284	0	-2284
Aug	2447	0	3250	3250
Sep	5697	1780	0	-1780
Oct	3917	0	2667	2667
Nov	6584	0	1500	1500
Dec	8084	0	2475	2475
End	10559	null	null	null

As you see in the Navigator dialog box screenshot above, Power Query will distinguish Tables and Sheets and show appropriate icon for any of these types (note to the highlighted section in screenshot above).


Loading Excel Tables into Power Query


If you have data in an Excel Table then it can be easily detected and picked by Power Query as you see in screenshot above, CashFlow is a table in Excel. screenshot below shows the Excel table for CashFlow;

	A	B	C	D	E
1	Period	Base	Down	Up	Cash Flow
2	Start		\$0	\$5,000	\$5,000
3	Jan	\$5,000	\$503	\$0	-\$503
4	Feb	\$4,497	\$1,670	\$0	-\$1,670
5	Mar	\$2,827	\$0	\$4,802	\$4,802
6	Apr	\$7,629	\$1,198	\$0	-\$1,198
7	May	\$6,431	\$3,526	\$0	-\$3,526
8	Jun	\$2,905	\$0	\$1,826	\$1,826
9	Jul	\$4,731	\$2,284	\$0	-\$2,284
10	Aug	\$2,447	\$0	\$3,250	\$3,250
11	Sep	\$5,697	\$1,780	\$0	-\$1,780
12	Oct	\$3,917	\$0	\$2,667	\$2,667
13	Nov	\$6,584	\$0	\$1,500	\$1,500
14	Dec	\$8,084	\$0	\$2,475	\$2,475
15	End	\$10,559			

As you can see in screenshot below Power Query (or Power BI) fetched the table fully

Navigator



Show All | Show Selected [0]


Sample Excel Source.xlsx [7]

- ☒ CashFlow
- ☐ Budget
- ☐ Sheet4
- ☐ Sheet1
- ☐ Sheet2
- ☐ Sheet2 (2)
- ☐ Sheet3

CashFlow

Period	Base	Down	Up	Cash Flow
Start	null	0	5000	5000
Jan	5000	503	0	-503
Feb	4497	1670	0	-1670
Mar	2827	0	4802	4802
Apr	7629	1198	0	-1198
May	6431	3526	0	-3526
Jun	2905	0	1826	1826
Jul	4731	2284	0	-2284
Aug	2447	0	3250	3250
Sep	5697	1780	0	-1780
Oct	3917	0	2667	2667
Nov	6584	0	1500	1500
Dec	8084	0	2475	2475
End	10559	null	null	null

Loading Excel Sheets into Power Query

You can get data directly from Excel sheets as well, No matter if you have tables or not. Power Query will always read the data from Excel sheets from all cells that contains data. if you have even two data sets in one Excel sheet Power Query still read that and load it correctly. Here is an example of Excel sheet with two data sets;

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Period	Base	Down	Up	Cash Flow										
2	Start		0	5000	5000										
3	Jan	5000	503	0	-503										
4	Feb	4497	1670	0	-1670										
5	Mar	2827	0	4802	4802										
6	Apr	7629	1198	0	-1198										
7	May	6431	3526	0	-3526										
8	Jun	2905	0	1826	1826										
9	Jul	4731	2284	0	-2284										
10	Aug	2447	0	3250	3250										
11	Sep	5697	1780	0	-1780										
12	Oct	3917	0	2667	2667										
13	Nov	6584	0	1500	1500										
14	Dec	8084	0	2475	2475										
15	End	10559													
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															

Period	Base	Down	Up	Cash Flow
Start		0	5000	5000
Jan	5000	503	0	-503
Feb	4497	1670	0	-1670
Mar	2827	0	4802	4802
Apr	7629	1198	0	-1198
May	6431	3526	0	-3526
Jun	2905	0	1826	1826
Jul	4731	2284	0	-2284
Aug	2447	0	3250	3250
Sep	5697	1780	0	-1780
Oct	3917	0	2667	2667
Nov	6584	0	1500	1500
Dec	8084	0	2475	2475
End	10559			

When you load this sheet in Power Query only data range of data cells up to the last cell's column and row will be fetched;

Navigator

Show All | Show Selected [0]

- Sample Excel Source.xlsx [7]
 - CashFlow
 - Budget
 - Sheet4
 - Sheet1
 - Sheet2**
 - Sheet2 (2)
 - Sheet3

Sheet2

Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11	Column12	Column13	Column14	Column15
Period	Base	Down	Up	Cash Flow										
Start		null	0	5000	5000	null	null	null	null	null	null	null	null	null
Jan	5000	503	0	-503		null	null	null	null	null	null	null	null	null
Feb	4497	1670	0	-1670		null	null	null	null	null	null	null	null	null
Mar	2827	0	4802	4802		null	null	null	null	null	null	null	null	null
Apr	7629	1198	0	-1198		null	null	null	null	null	null	null	null	null
May	6431	3526	0	-3526		null	null	null	null	null	null	null	null	null
Jun	2905	0	1826	1826		null	null	null	null	null	null	null	null	null
Jul	4731	2284	0	-2284		null	null	null	null	null	null	null	null	null
Aug	2447	0	3250	3250		null	null	null	null	null	null	null	null	null
Sep	5697	1780	0	-1780		null	null	null	null	null	null	null	null	null
Oct	3917	0	2667	2667		null	null	null	null	null	null	null	null	null
Nov	6584	0	1500	1500		null	null	null	null	null	null	null	null	null
Dec	8084	0	2475	2475		null	null	null	null	null	null	null	null	null
End	10559					null	null	null	null	null	null	null	null	null

Load Edit Cancel

What Happens If Excel Contains Formatting?

Any formatting such as color, font, data type of cells would be ignored when it loads into Power Query. The reason is that Power Query is a data mash-up tools not a modelling or visualization tools. You can apply these formatting later in the model (Power Pivot) or in the report (Power View). As an example; if your data values contains decimal points such as 12.94 and in Excel you've formatted cell to have zero decimal points (Excel will show cell value as 13 in this case). Power Query still fetches the original value which is 12.94 .

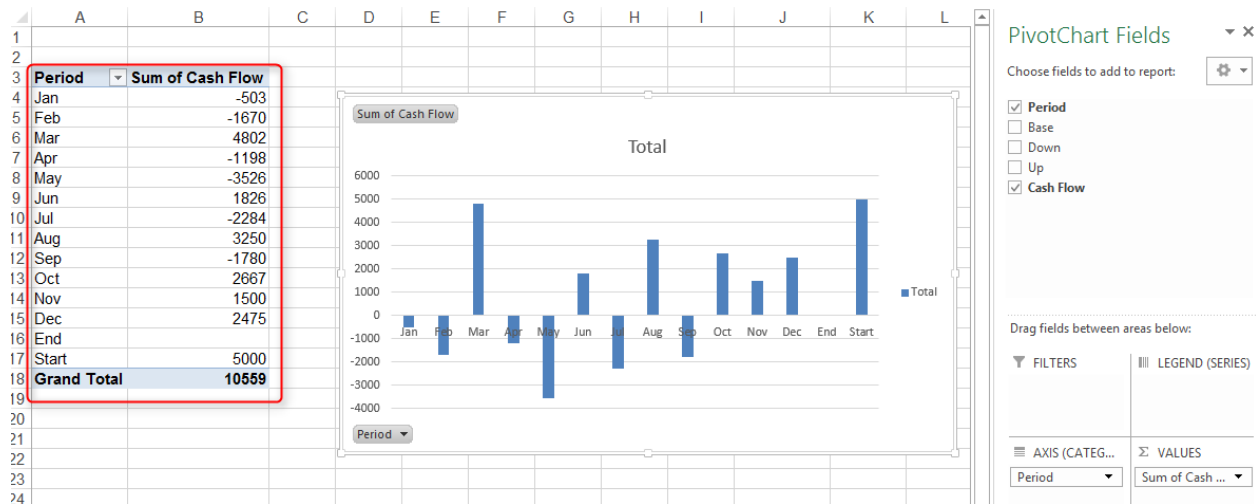
What Happens to Power View Sheets in Excel?

You can have Power View sheets in Excel, and your Power View sheets can contain Data values, such as table. However Power Query won't load anything from a Power View Sheet at this stage. Because usually Power View sheet uses a data source you can use that source directly in Power Query. Power View data source might be sources such as Pivot Table or SSAS Tabular connection that can be both used in Power Query directly.

Pivot Tables and Pivot Charts?

Obviously you cannot fetch data from Pivot Charts! Why? because chart is a visualization element, so same principal that I said for Power View in above paragraph works here, connect to Pivot Chart's source directly. You can get data from PivotTables however.

PivotTables data can be fetched exactly as they shaped in Excel file with same structure of columns and rows. For example, if you have a Pivot Table like below screenshot:



After loading that in Power Query, you will see the Pivot Table exactly the same (without formatting);

Navigator

Show All | Show Selected [0]

Sample Excel Source.xlsx [7]

- ☐ CashFlow
- ☐ Budget
- ☐ Sheet4
- ☐ Sheet1
- ☐ Sheet2
- ☐ Sheet2 (2)
- ☐ Sheet3

Sheet4

Period	Sum of Cash Flow
Jan	-503
Feb	-1670
Mar	4802
Apr	-1198
May	-3526
Jun	1826
Jul	-2284
Aug	3250
Sep	-1780
Oct	2667
Nov	1500
Dec	2475
End	null
Start	5000
Grand Total	10559

What If Your Excel Table Has Merged Cells?

Merged Cells in Excel are commonly used. You can have tables with cells merged vertically or horizontally or even both. Here is an example of a table with merged cells;

Period	Service Center	Budget	Total Budget
1/1	IT	100000	600000
	Finance	500000	
2/1	IT	150000	600000
	Finance	450000	
3/1	IT	170000	570000
	Finance	400000	
4/1	IT	200000	600000
	Finance	400000	

Power Query still reads cells in their original detailed format. It means merged cells won't be merged in Power Query, they will be seen as separate cells.

Navigator

Navigator

Show All | Show Selected [0]

Sample Excel Source.xlsx [6]

- ☐ CashFlow
- ☐ Sheet4
- ☐ Sheet1
- ☐ Sheet2
- ☐ Sheet2 (2)
- ☐ Sheet3

Sheet3

Column1	Column2	Column3	Column4	Column5	Column6	Column7
Period	Service Center	Budget	Total Budget	null	null	
1/1/2015	IT	100000	600000	null	null	
null	Finance	500000	null	null	null	
2/1/2015	IT	150000	600000	null	null	
null	Finance	450000	null	null	null	
3/1/2015	IT	170000	570000	null	null	
null	Finance	400000	null	null	null	
4/1/2015	IT	200000	600000	null	null	
null	Finance	400000	null	null	null	

You can use transformations such as Fill Down to fill null values in the remaining cells of merged cells later on. Also note in above screenshot that green highlighted column is a calculated column. This calculated value will be fetched in Power Query as static values.

Example Excel Data Source: Olympic Games

I've found list of all medalists in Olympic games from the very first game (1896) till 2008 (unfortunately 2012 london games are not included in this data set). This is fortunately a

public list made available by [The Guardian](#), that you can download [here](#). The list is well structured with a main sheet for all medalists as below;

	A	B	C	D	E	F	G	H	I	J
1	List of medalists at the Games of the Olympiad per edition, sport, discipline, gender and event									
2	DISCLAIMER: The IOC Research and Reference Service endeavours to provide you with accurate and up-to-date information. However, it offers no guarantees, express or implied, as to the accuracy or completeness of the information provided.									
3										
4										
5	City	Edition	Sport	Discipline	Athlete	NO	Gender	Event	Event_gender	Medal
6	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
7	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
8	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
9	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
10	Athens	1896	Aquatics	Swimming	CHASAPIS, Spindon	GRE	Men	100m freestyle for sailors	M	Silver
11	Athens	1896	Aquatics	Swimming	CHOROPHAS, Elstathios	GRE	Men	1200m freestyle	M	Bronze
12	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200m freestyle	M	Gold
13	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200m freestyle	M	Silver
14	Athens	1896	Aquatics	Swimming	CHOROPHAS, Elstathios	GRE	Men	400m freestyle	M	Bronze
15	Athens	1896	Aquatics	Swimming	NEUMANN, Paul	AUT	Men	400m freestyle	M	Gold
16	Athens	1896	Aquatics	Swimming	PEPANOS, Antonios	GRE	Men	400m freestyle	M	Silver
17	Athens	1896	Athletics	Athletics	LANE, Francis	USA	Men	100m	M	Bronze
18	Athens	1896	Athletics	Athletics	SZOKOLYI, Alajos	HUN	Men	100m	M	Bronze
19	Athens	1896	Athletics	Athletics	BURKE, Thomas	USA	Men	100m	M	Gold

As you see in above sheet all medalists with their main sport category and discipline and detailed event are available. Name of Athletes and their gender and Medals as well as the Olympic game (year) all is listed. Countries information listed as three character code. This three character codes are available as a reference in another sheet of the file named IOC Country Codes as below;

Country	Int Olympic	ISO code	Country
Afghanistan	AFG	AF	Afghanistan
Albania	ALB	AL	Albania
Algeria	ALG	DZ	Algeria
American Samoa*	ASA	AS	American Samoa*
Andorra	AND	AD	Andorra
Angola	ANG	AO	Angola
Antigua and Barbuda	ANT	AG	Antigua and Barbuda
Argentina	ARG	AR	Argentina
Armenia	ARM	AM	Armenia
Aruba*	ARU	AW	Aruba*
Australia	AUS	AU	Australia
Austria	AUT	AT	Austria
Azerbaijan	AZE	AZ	Azerbaijan
Bahamas	BAH	BS	Bahamas
Bahrain	BRN	BH	Bahrain
Bangladesh	BAN	BD	Bangladesh
Barbados	BAR	BB	Barbados

Data is well structured and loading it into Power BI would be just matter of seconds! Start getting this information by Get Data From Excel and then address the downloaded Excel file. In Navigator dialog box choose All Medalists and IOC Country Codes sheets both to be checked. Also note that All Medalists sheet's data shows in non-merged cell style (as you've learned earlier in this section; merged cells will be un-merged in Power Query). After selecting these two sheets click on Edit.

Navigator

Search:

Show All | Show Selected [2]

- Summer Olympic medallists 1896 to...
- ☒ ALL MEDALISTS
- ☐ BREAKDOWN
- ☐ COUNTRY TOTALS
- ☒ IOC COUNTRY CODES
- ☐ TEAM EVENTS FIXED, ALL YRS T...
- ☐ Sheet8
- ☐ Chart2
- ☐ Chart1
- ☐ IOC COUNTRY CODES 1
- ☐ TOPOLYMPIANS

ALL MEDALISTS

Preview downloaded on Sunday

Column1	Column2	Column3	Column4	Column5
List of medallists at the Games of the Olympiad per edition, sport, disciplir	null	null	null	
null	null	null	null	
DISCLAIMER: The IOC Research and Reference Service endeavours to prov	null	null	null	
null	null	null	null	
City	Edition	Sport	Discipline	Athlete
Athens	1896	Aquatics	Swimming	HAJOS,
Athens	1896	Aquatics	Swimming	HERSCH
Athens	1896	Aquatics	Swimming	DRIVAS
Athens	1896	Aquatics	Swimming	MALOK
Athens	1896	Aquatics	Swimming	CHASAI
Athens	1896	Aquatics	Swimming	CHORO
Athens	1896	Aquatics	Swimming	HAJOS,
Athens	1896	Aquatics	Swimming	ANDRE
Athens	1896	Aquatics	Swimming	CHORO
Athens	1896	Aquatics	Swimming	NEUMA
Athens	1896	Aquatics	Swimming	PEPANI
Athens	1896	Athletics	Athletics	LANE, F
Athens	1896	Athletics	Athletics	SZOKOI
Athens	1896	Athletics	Athletics	BURKE,
Athens	1896	Athletics	Athletics	HOFMA
Athens	1896	Athletics	Athletics	CURTIS
Athens	1896	Athletics	Athletics	GOULD

Load

Edit

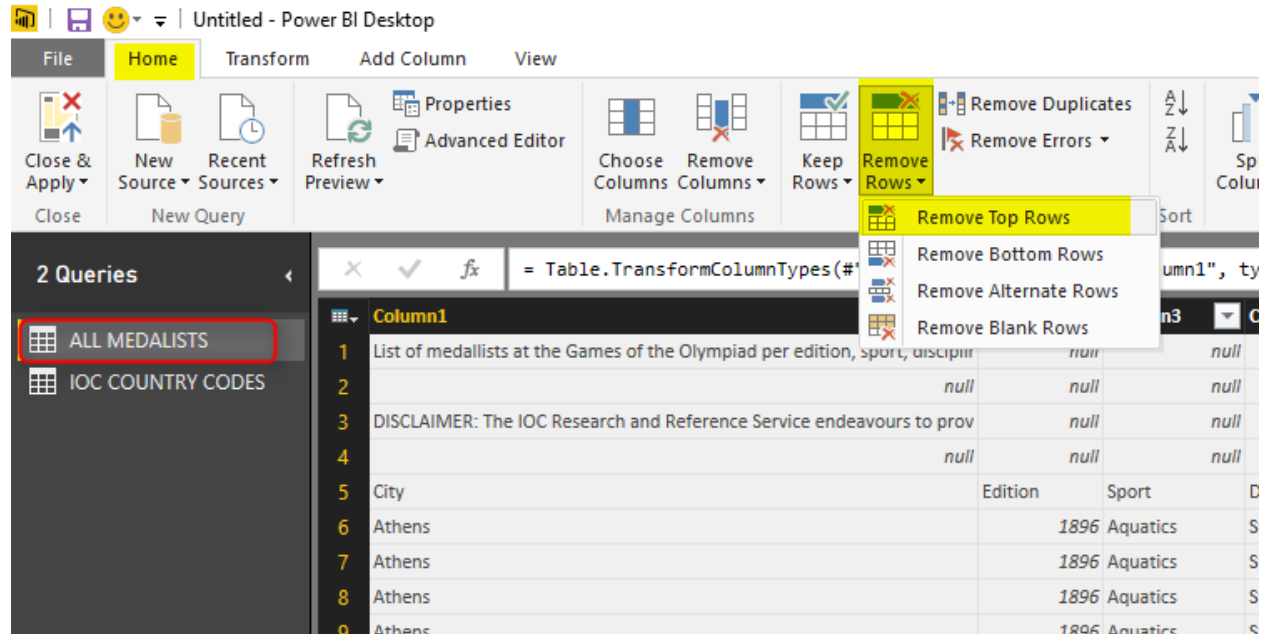
Cancel

There are just few changes that we need to make;

Remove Rows

All Medalists query contains four heading rows which we don't need them (it is just title and disclaimer) so better to remove those;

Go to All Medalists query and then click on Remove Rows in the Home menu. From the Remove Rows popup menu choose Remove Top Rows

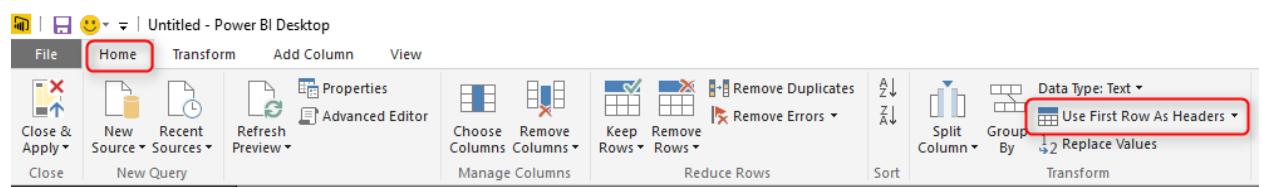


Remove Top 4 Rows in the dialog box by entering 4. and the result will look like below;

Rank	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10
1	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
2	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
3	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
4	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
5	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
6	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver
7	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	1200m freestyle	M	Bronze
8	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200m freestyle	M	Gold
9	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200m freestyle	M	Silver
10	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	400m freestyle	M	Bronze
11	Athens	1896	Aquatics	Swimming	NEUMANN, Paul	AUT	Men	400m freestyle	M	Gold
12	Athens	1896	Aquatics	Swimming	PEPANOS, Antonios	GRE	Men	400m freestyle	M	Silver
13	Athens	1896	Athletics	Athletics	LANE, Francis	USA	Men	100m	M	Bronze
14	Athens	1896	Athletics	Athletics	SZOKOLYI, Alajos	HUN	Men	100m	M	Bronze
15	Athens	1896	Athletics	Athletics	BURKE, Thomas	USA	Men	100m	M	Gold

Use First Row As Headers

Headers of query as you see in screenshot above are Column1, Column2.... Fortunately in the data set the first row contains column headers. We can simply set that to be used for column headers in Power Query. In the Home tab click on Use First Row As Headers.



Data set will look like screenshot below after above change

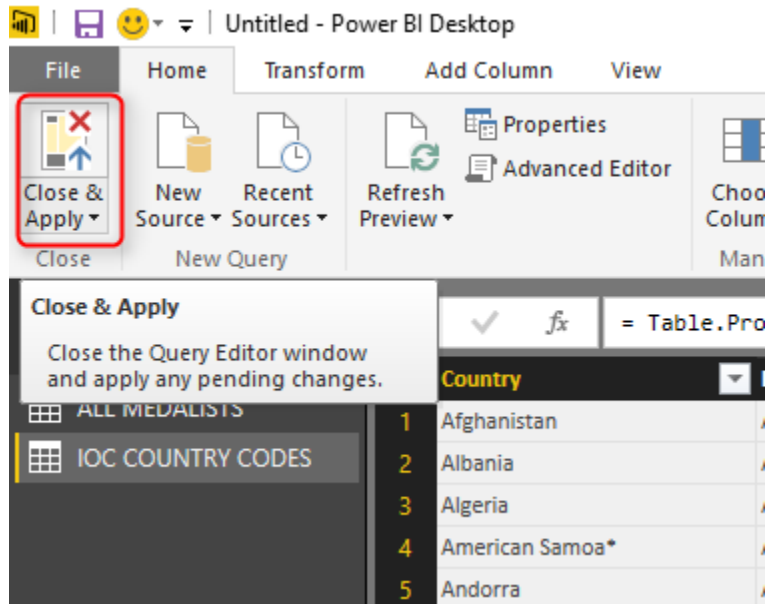
	City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender	Medal
1	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	100m freestyle	M	Gold
2	Athens	1896	Aquatics	Swimming	HERSCHMANN, Otto	AUT	Men	100m freestyle	M	Silver
3	Athens	1896	Aquatics	Swimming	DRIVAS, Dimitrios	GRE	Men	100m freestyle for sailors	M	Bronze
4	Athens	1896	Aquatics	Swimming	MALOKINIS, Ioannis	GRE	Men	100m freestyle for sailors	M	Gold
5	Athens	1896	Aquatics	Swimming	CHASAPIS, Spiridon	GRE	Men	100m freestyle for sailors	M	Silver
6	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	1200m freestyle	M	Bronze
7	Athens	1896	Aquatics	Swimming	HAJOS, Alfred	HUN	Men	1200m freestyle	M	Gold
8	Athens	1896	Aquatics	Swimming	ANDREOU, Joannis	GRE	Men	1200m freestyle	M	Silver
9	Athens	1896	Aquatics	Swimming	CHOROPHAS, Efstathios	GRE	Men	400m freestyle	M	Bronze
10	Athens	1896	Aquatics	Swimming	NEUMANN, Paul	AUT	Men	400m freestyle	M	Gold
11	Athens	1896	Aquatics	Swimming	PEPANOS, Antonios	GRE	Men	400m freestyle	M	Silver
12	Athens	1896	Athletics	Athletics	LANE, Francis	USA	Men	100m	M	Bronze
13	Athens	1896	Athletics	Athletics	SZOKOLYI, Alajos	HUN	Men	100m	M	Bronze
14	Athens	1896	Athletics	Athletics	BURKE, Thomas	USA	Men	100m	M	Gold
15	Athens	1896	Athletics	Athletics	HOFMANN, Fritz	GER	Men	100m	M	Silver
16	Athens	1896	Athletics	Athletics	CURTIS, Thomas	USA	Men	110m hurdles	M	Gold

For IOC Country Codes query just set the first row as headers. No more changes is required

	Country	Int Olympic Committee code	ISO code	Country_1
1	Afghanistan	AFG	AF	Afghanistan
2	Albania	ALB	AL	Albania
3	Algeria	ALG	DZ	Algeria
4	American Samoa*	ASA	AS	American Samoa*
5	Andorra	AND	AD	Andorra
6	Angola	ANG	AO	Angola
7	Antigua and Barbuda	ANT	AG	Antigua and Barbuda
8	Argentina	ARG	AR	Argentina
9	Armenia	ARM	AM	Armenia
10	Aruba*	ARU	AW	Aruba*
11	Australia	AUS	AU	Australia
12	Austria	AUT	AT	Austria
13	Azerbaijan	AZE	AZ	Azerbaijan
14	Bahamas	BAH	BS	Bahamas
15	Bahrain	BRN	BH	Bahrain

Close and Load

After above changes we can now load the result set into Power BI model to build report for it. You can simply click on Close and Apply menu button in Home Tab



Power BI will load result sets into memory for further modeling and reporting.

Apply Query Changes

- ALL MEDALISTS
Waiting for other queries...
- IOC COUNTRY CODES
Evaluating...

Cancel

A bit of Modeling

This section is not about data modeling. However I want you to be able to play with this data set and build some nice reports with it (if you can't wait till modeling and visualization chapters of this book). So Let's create the relationship between All medalists and IOC Country Code. For doing this go to Relationship tab in Power BI Desktop and the click on Manage Relationship. Then create a new relationship as below;

Create Relationship

Select tables and columns that relate to one another.

ALL MEDALISTS
▼

City	Edition	Sport	Discipline	Athlete	NOC	Gender	Event	Event_gender
London	1908	Athletics	Athletics	CARTMELL, John Nathaniel	USA	Men	4x400m relay	M
London	1908	Athletics	Athletics	HAMILTON, William Frank	USA	Men	4x400m relay	M
London	1908	Athletics	Athletics	SHEPPARD, Melvin	USA	Men	4x400m relay	M
London	1908	Athletics	Athletics	TAYLOR, John Baxter	USA	Men	4x400m relay	M
Stockholm	1912	Athletics	Athletics	LINDBERG, Edward F.	USA	Men	4x400m relay	M

<
>

IOC COUNTRY CODES
▼

Country	Int Olympic Committee code	ISO code	Country_1
Afghanistan	AFG	AF	Afghanistan
Albania	ALB	AL	Albania
Algeria	ALG	DZ	Algeria
American Samoa*	ASA	AS	American Samoa*
Andorra	AND	AD	Andorra

⚙ Advanced options

Cardinality

Many to One (*:1)
▼

Cross filter direction

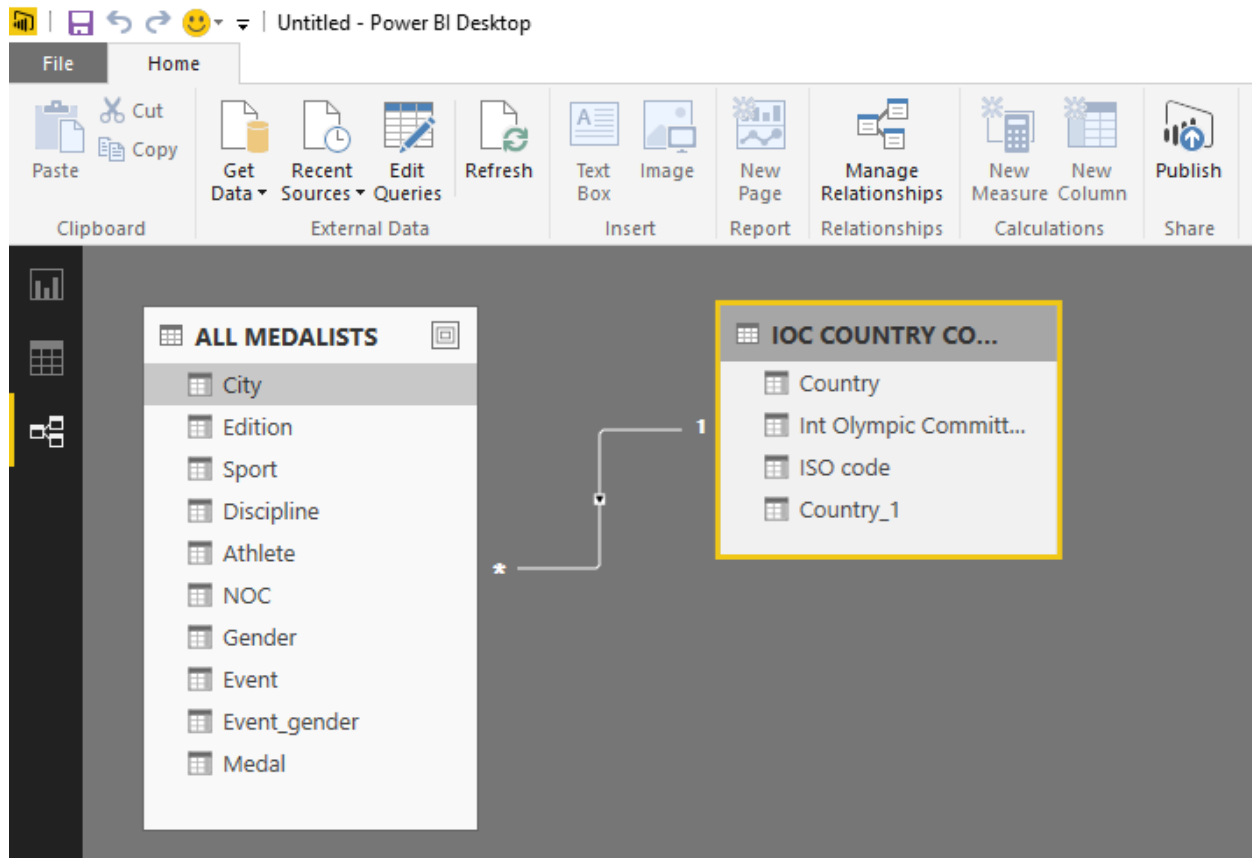
Single
▼

☒ Make this relationship active

OK

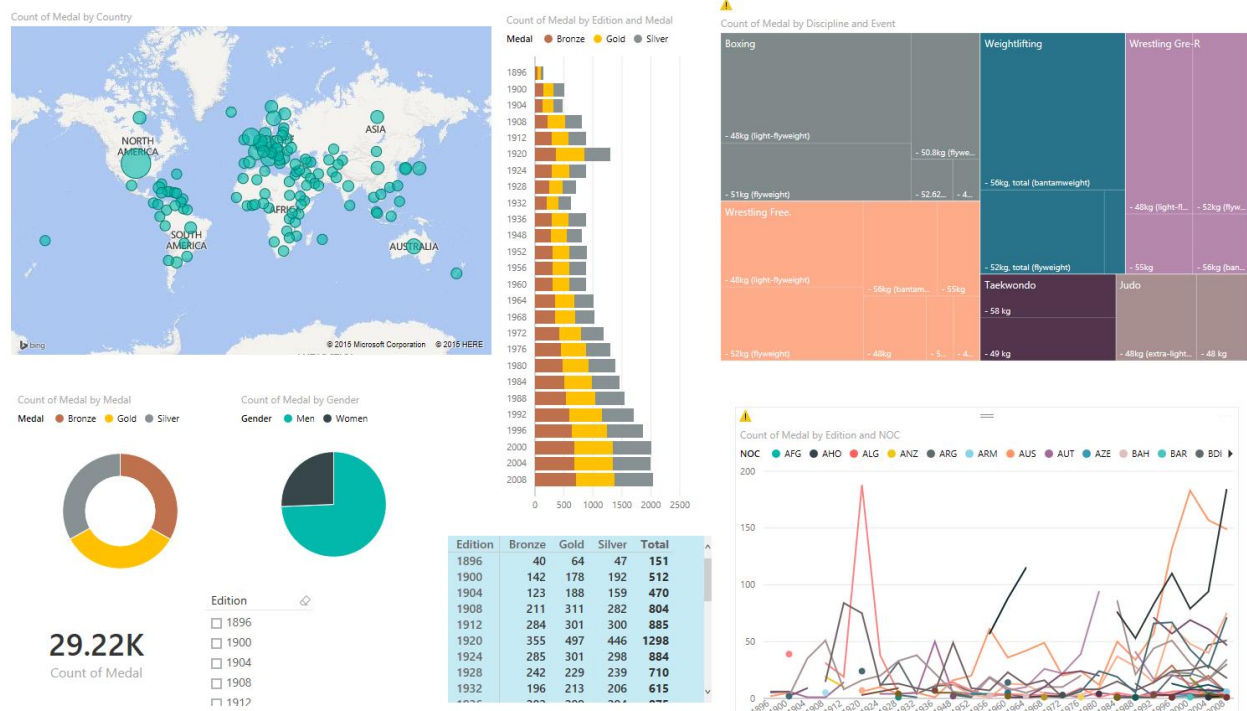
Cancel

After relationship has been created you will see it in the relationship diagram



Play Time!

Now it's your turn to play with this data set and make some nice reports and visualization items. Here is what I've built with this data set:



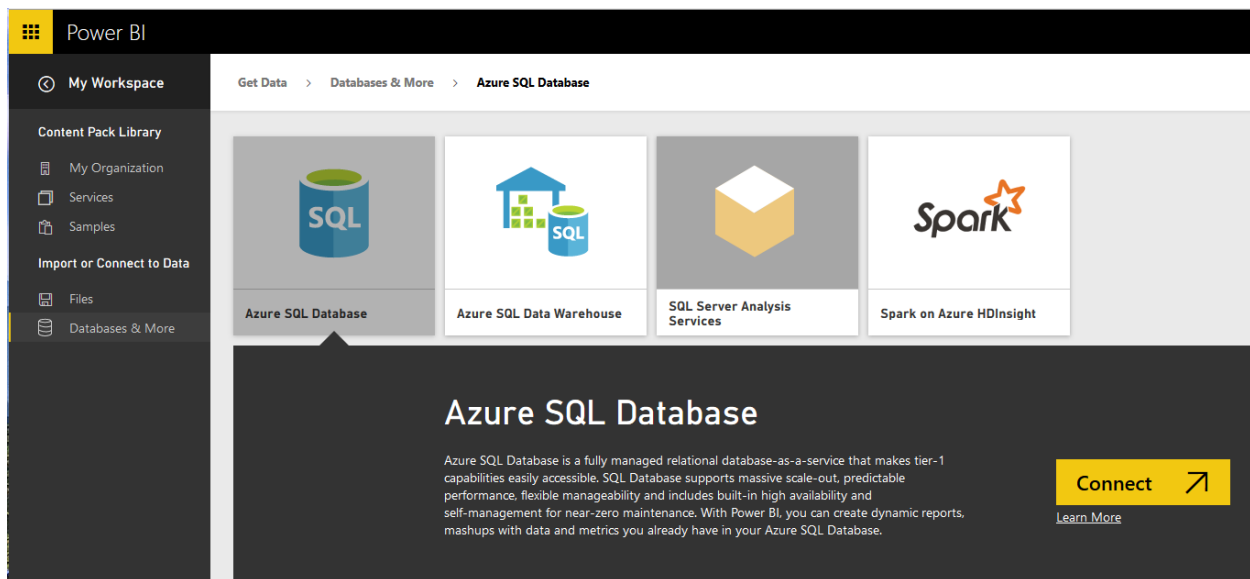
Don't be panic if you can't build report above. I will go through step by step process of building this report later in the Reporting Chapters.

Summary

In this section you've learned some tips of working with Excel Data source from Power Query or Power BI Desktop. You've learned that merged cells would be loaded into Power Query Un-merged. Formatting won't be considered at the time of loading data into Power Query, and Power Query can load data from Pivot Tables, Excel Tables, and Sheets. You've also learned a real world example of fetching Olympic medalists data from Excel file into Power BI Desktop. In next sections I will get you through the journey of Getting data from some other data sources.

Power BI Get Data: From Azure SQL Database

Published Date: September 12, 2015



Power BI and Power Query can connect to files such as Excel, CSV, text files and on-premises databases such as SQL Server, Oracle, MySQL. Power BI can connect to many data sources on cloud such as Azure SQL Database, Azure SQL Data Warehouse and etc. In this part you will learn how to connect from Power BI Desktop to Azure SQL Database. There is also a way of connecting to Azure SQL Database with a direct connection from Power BI website which will be explored in this section as well. You will also learn how you can schedule your report to refresh data loaded from Azure SQL DB. So In general you should expect to learn everything related to Power BI relation to Azure SQL Database in this section.

In this section you will learn;

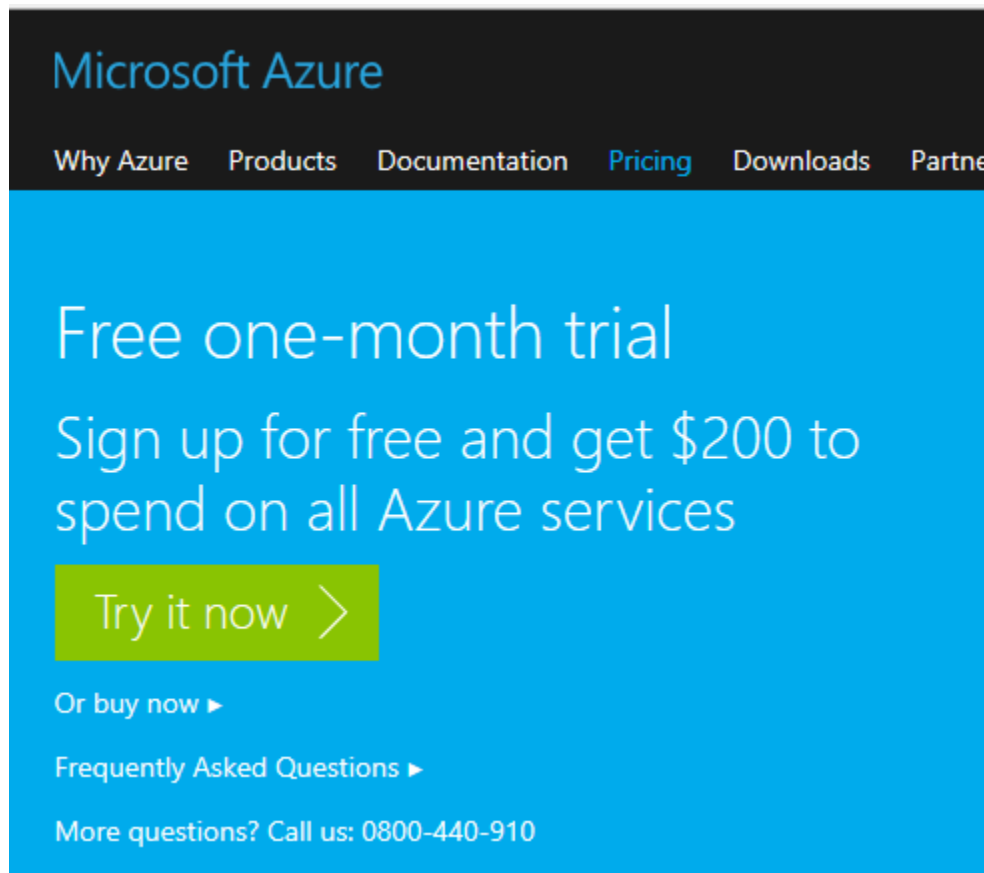
1. How to connect from Power BI Desktop to Azure SQL Database
2. Schedule Power BI for refreshing data from Azure SQL Database
3. Direct Connection to Azure SQL Database from Power BI Website

Preparation

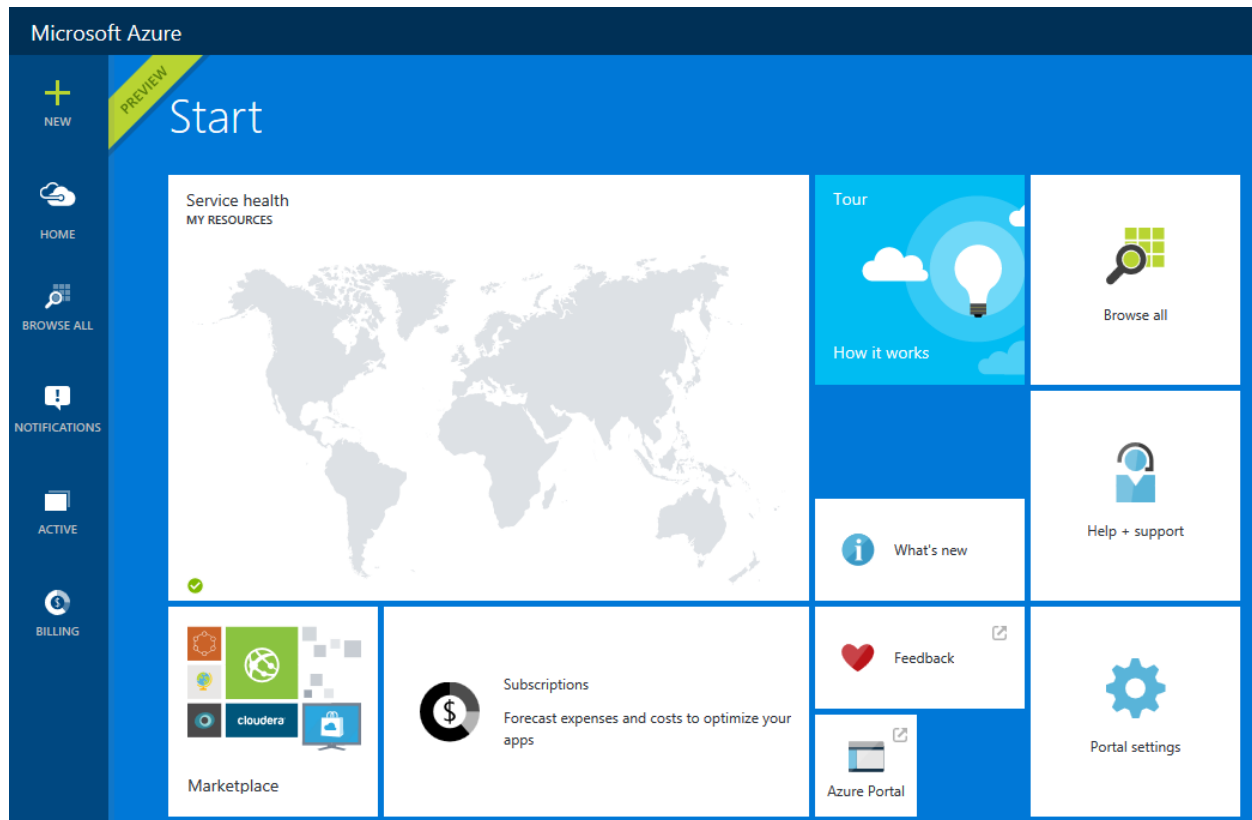
For this section you need an Azure SQL Database. I use AdventureWorksLT example, this is the sample database in Azure SQL Database templates that you can easily install and configure. If you have this database set up on Azure then you can skip this step. For creating an Azure SQL Database for AdventureWorksLT database follow below steps;

Do you need an Azure subscription to run this experiment?

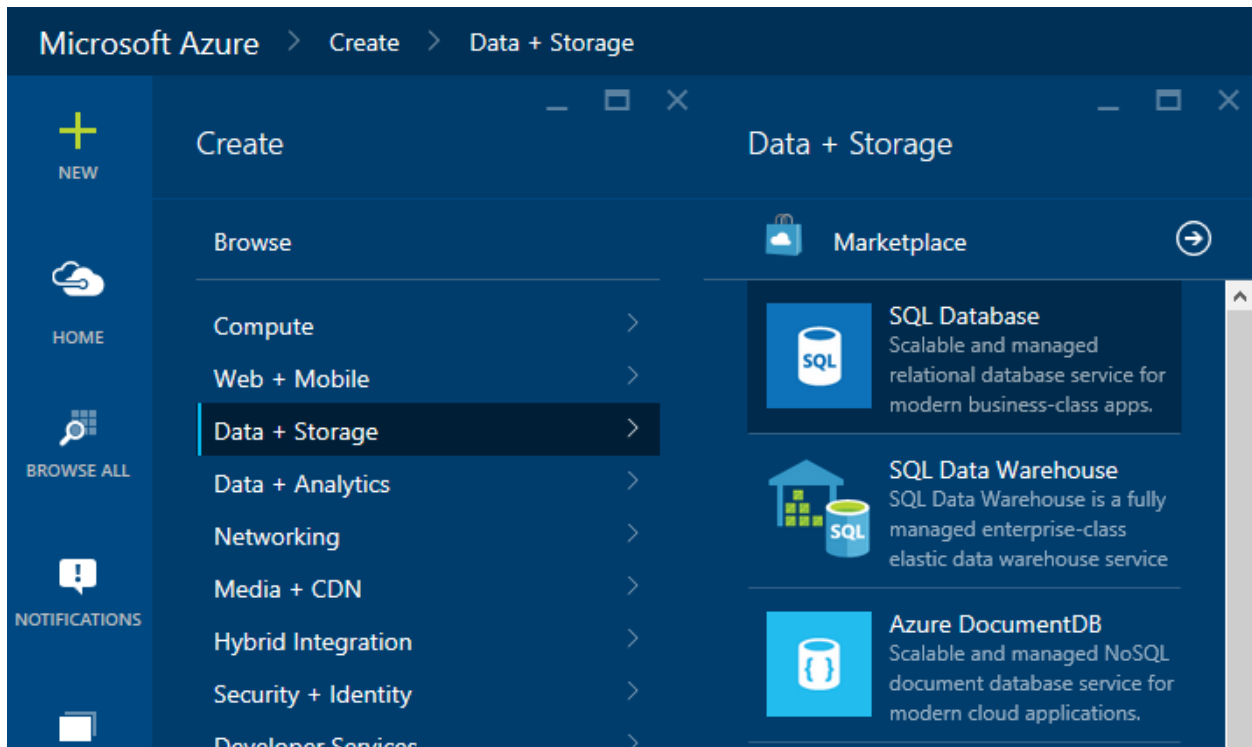
Yes, but don't worry if you don't have it. You can have an Azure subscription free for 25 days with 200\$ free credit for you to use, just use the trial version. You can start trial version by following this URL: <https://azure.microsoft.com/en-us/pricing/free-trial/>



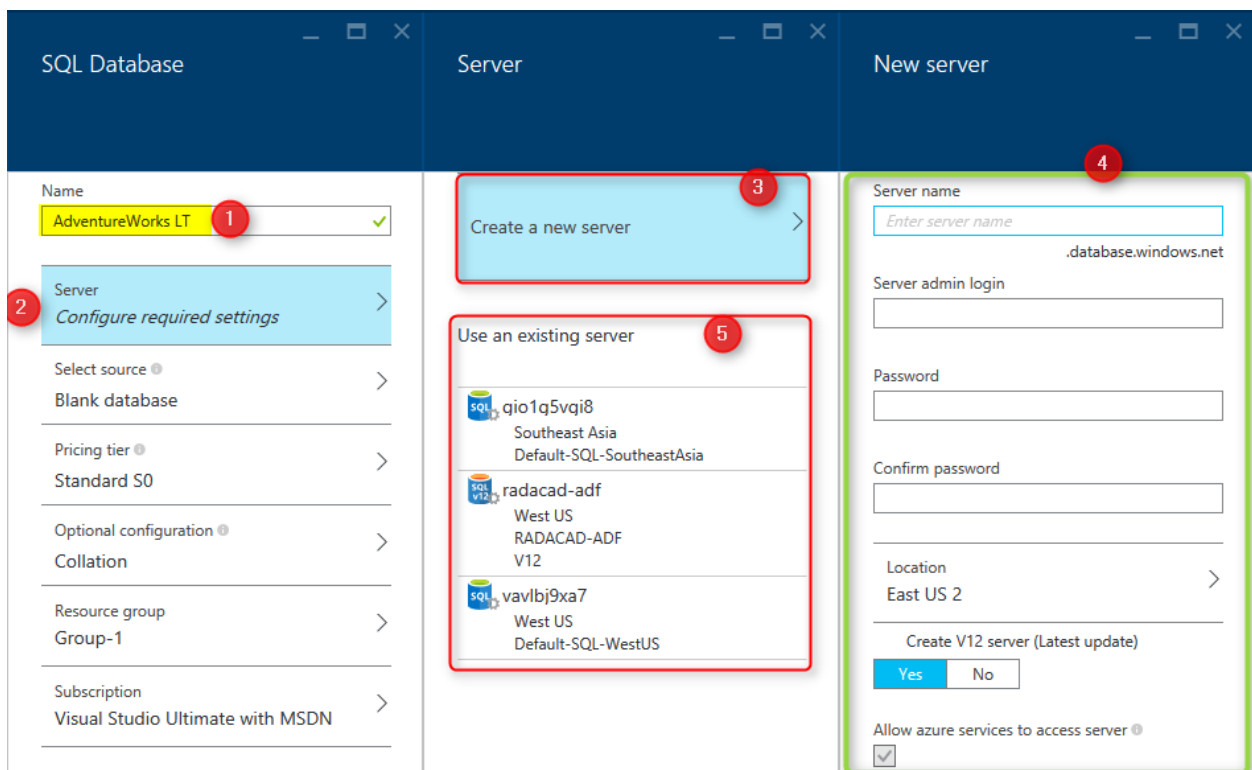
After setting up your Azure Account, go to Azure Portal. I have to mention here that there are two versions of management portal for Azure. the new Azure portal which is tablet friendly, with newer and better look and feel, and the old management portal. screenshots and steps described in this example all has been done in the new Azure Portal. You can go to Azure portal by using this URL: <https://portal.azure.com>



You can manage your Azure services in the management portal by creating new services, editing existing services. Talking about Azure services is out of scope for this example and you need to read books on that topic. However for this example let's smoothly continue steps to create an Azure SQL Database. Click on New on the top left side and then under Data + Storage choose SQL Database

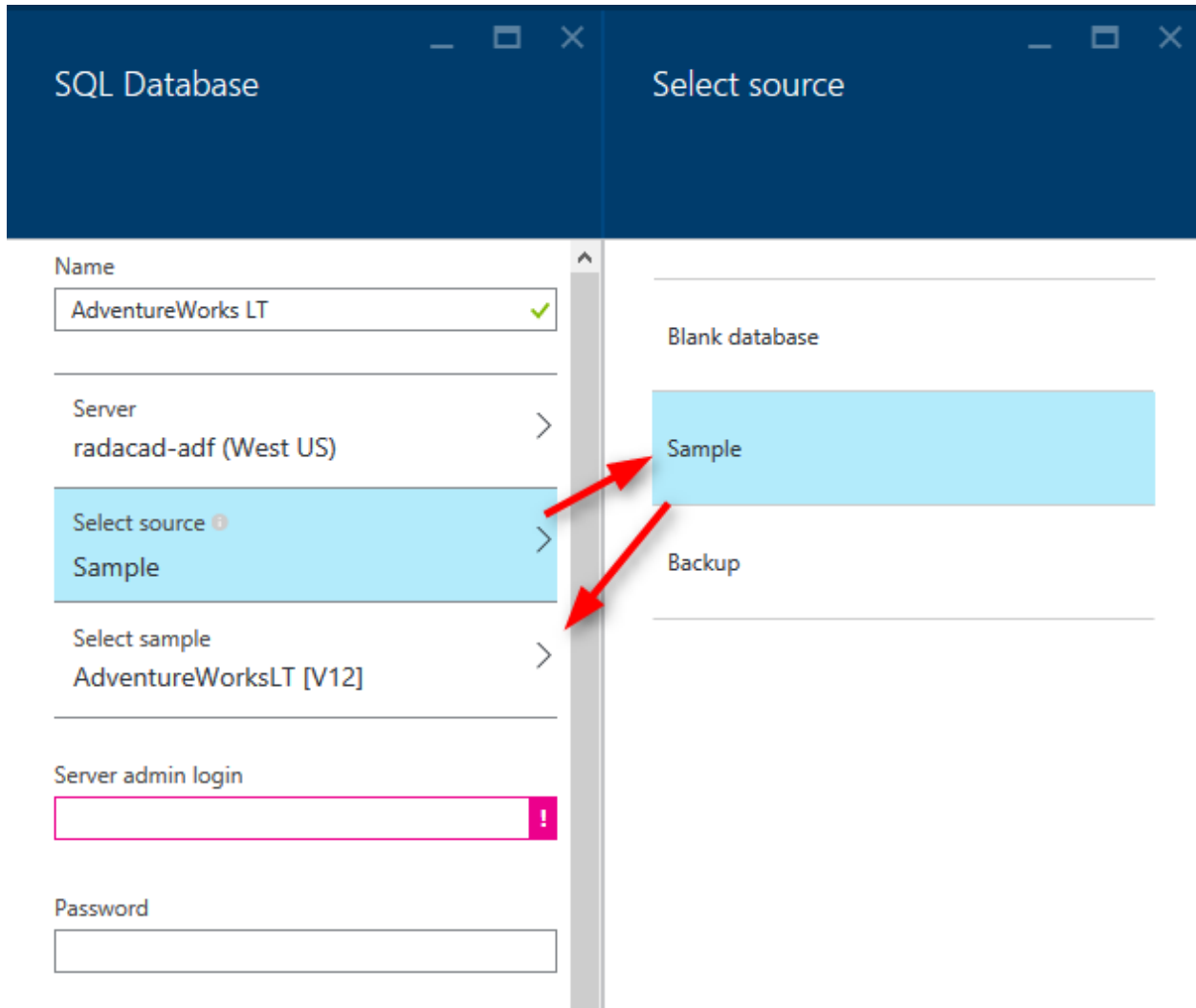


In the SQL Database Create pane, name the database as AdventureWorks LT. You have to choose the server also. Server is like a SQL Server instance that this database will be hosted on that. You can choose from an existing server or you can create a new Server.

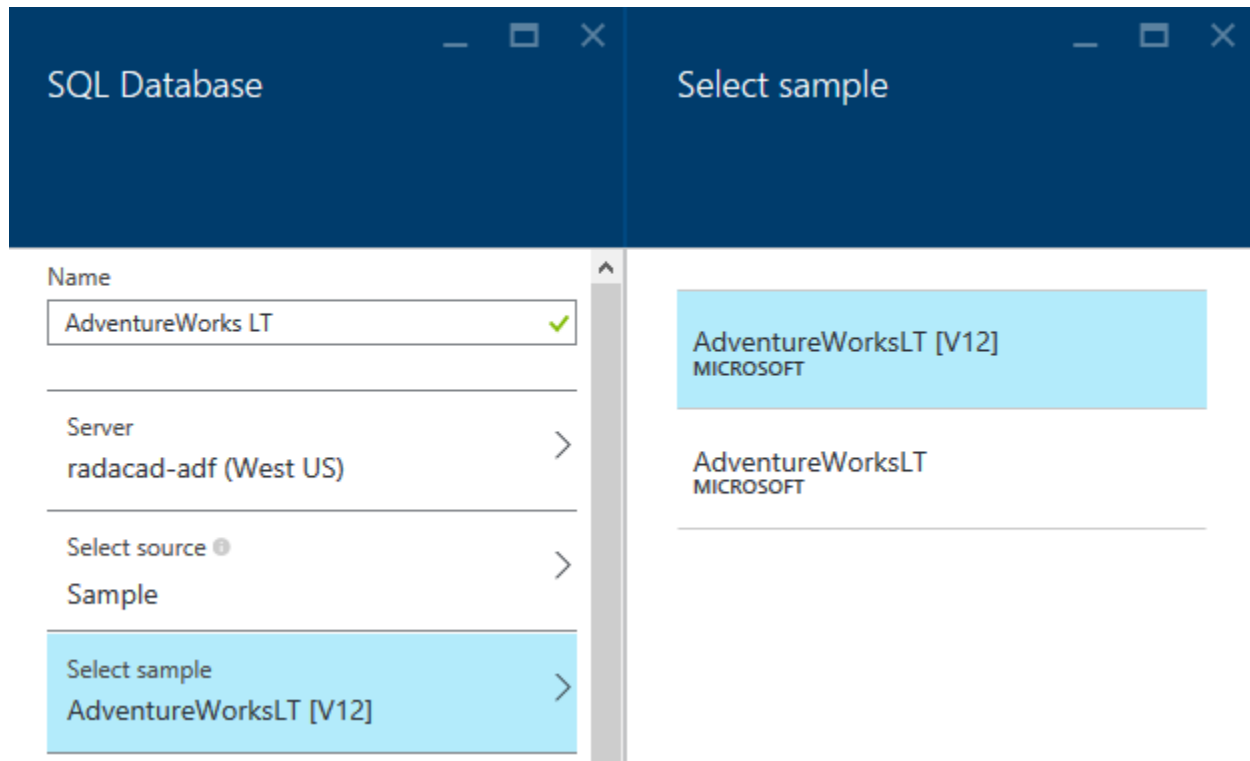


As you can see in screenshot above, after going to Configure required Settings for the Server, you can choose to create new server (numbered 3), which will redirect you to a new pane for setting up the server (numbered 4). or you can choose an existing server (numbered 5).

After setting up the server, you have to select source for the database. for this example choose Sample. after choosing sample you will see the Select Sample option below appears.



Choose the sample as AdventureWorksLT [V12], and then type in the sever admin login and password (you have defined that when you set up the Server)



You have to also choose a pricing tier, and resource group. for pricing tier just use one of the tiers (you can better choose yourself), and then for the resource group you can choose an existing one of create a new. A resource group is a grouping for Azure services, you can have a resource group and add all related azure services under that. for example you can have a resource group for Power BI Online Book and create all examples of this book under that. Please note that the resource group name should not have spaces in the name, but it can have dashes.

SQL Database

Name

AdventureWorks LT ✓

Server

radacad-adf (West US) >

Select source ⓘ

Sample >

Select sample

AdventureWorksLT [V12] >

Server admin login

reza ✓

Password

•••••••• ✓

Pricing tier ⓘ

Standard S0 >

Optional configuration ⓘ

Collation >

Resource group

RADACAD-ADF 🔒


Subscription






Visual Studio Ultimate with MSDN 🔒



☒ Pin to Startboard

Create

After all the configuration click on Create so the SQL DB creates. the tick on check box for pin to Startboard will bring the SQL DB on the first welcome page (start board) of the Azure Portal. It may take a bit time for the database to be created. After completion of creating database process, you will be redirected to database page in azure portal (if you didn't, then click on the AdventureWorks LT database on start board to go to its pane). screenshot below is showing the database created

 **AdventureWorksLT**
SQL Database



Essentials  

Resource group

RADACAD-ADF

Status

Online

Location

West US

Subscription name

Visual Studio Ultimate with MSDN

Subscription ID

71f18500-c4c8-4b70-a37e-97ad201c94ce

Server name

[radacad-adf.database.windows.net](#)

Pricing tier

S0 Standard (10 DTUs)

Geo-replication role

Not configured

Server version

V12

Connection strings

[Show database connection strings](#)

[All settings](#) 

Monitoring

Resource Utilization

[Edit](#)



DTU PERCENTAGE

0 %

Usage

Database Size



0 %

CURRENT

6.94 MB

THRESHOLD

Pricing tier

ADVENTUREWORKSLT

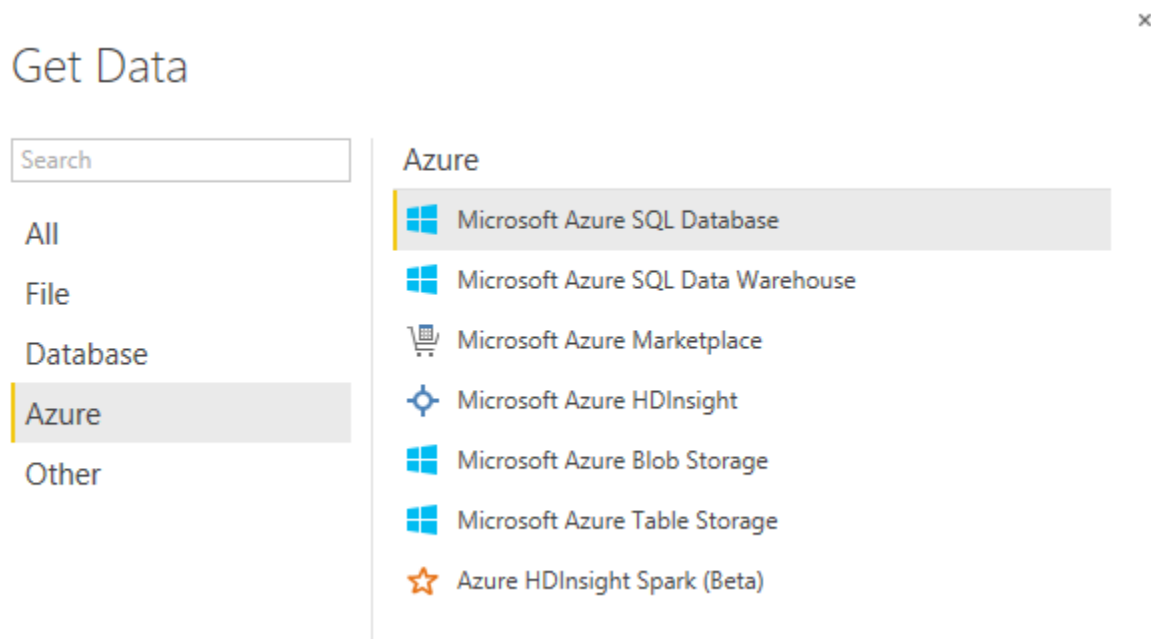


Now you are all set, example database is ready to be used in Power BI.

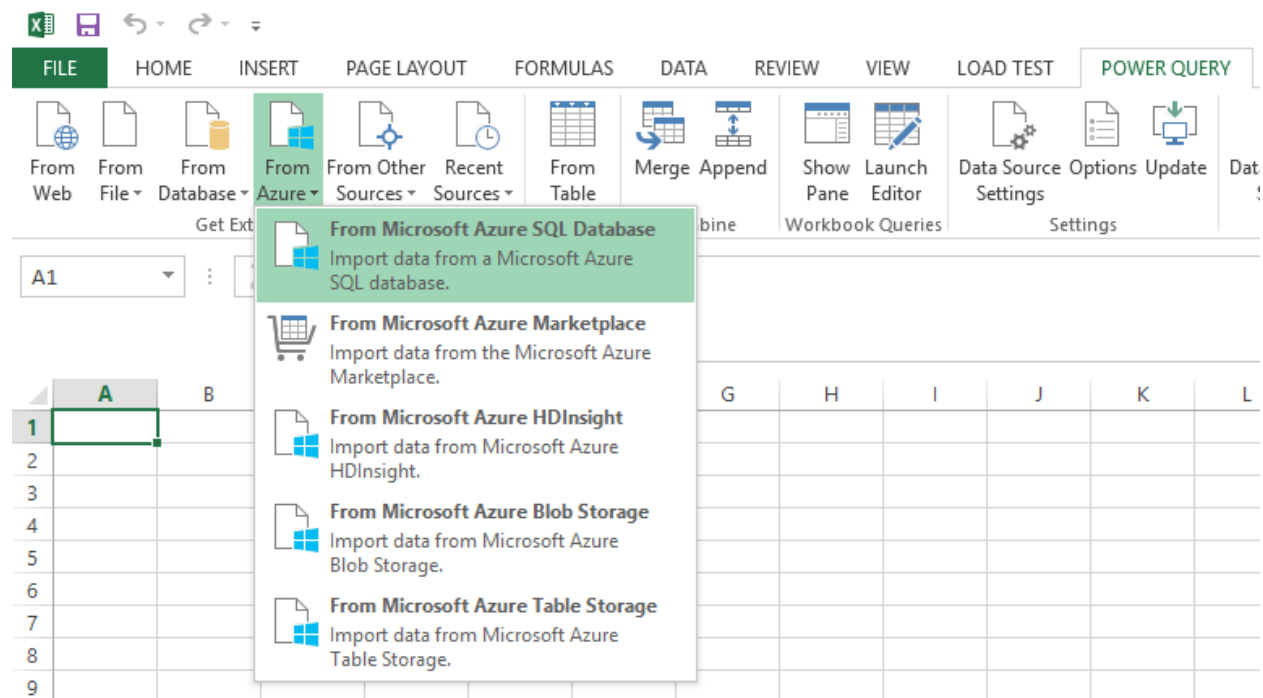
Get Data from Azure SQL Database

You can connect to Azure SQL Database from Power BI Desktop or Power Query for Excel. Both methods work the same. Let's go through the connection from Power BI Desktop. Before starting steps I have to mention that Power BI Desktop connection to Azure SQL Database is an off-line connection. Off-line connection here means the data from Azure SQL Database will be loaded into the Power BI model and then reports will use the data in the model, this disconnected way of connection is what I call off-line. The off-line connection to Azure SQL DB can be scheduled in the Power BI website to be refreshed to populated updated data from the database. In this section we will create the connection from Power BI Desktop to Azure SQL DB, and in the next section following you will learn how to schedule the data refresh.

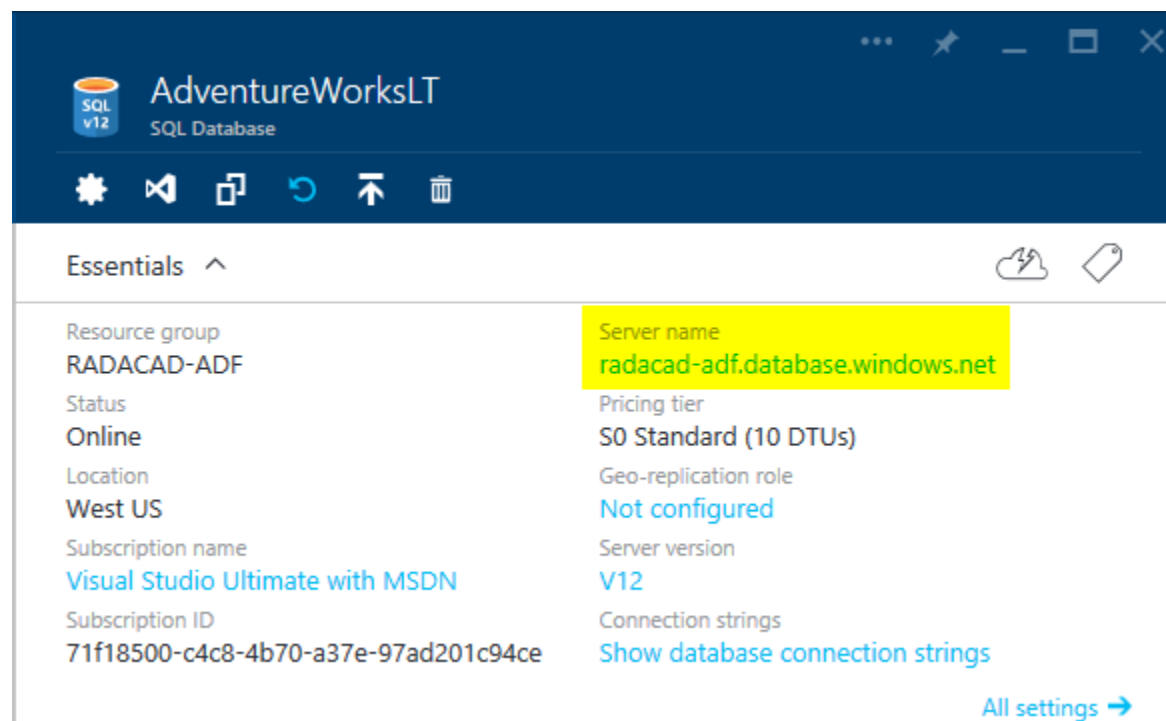
Open the Power BI Desktop and Get Data from Azure SQL Database




In Power Query for Excel you can also follow the path mentioned in screenshot below



You need to enter the server name in SQL Server Database dialog box. Remember that you've set up the server when you created Azure SQL DB. If you don't know what is the server for your database, simply find it through Azure portal under the Azure SQL DB pane;



Type in the server name in SQL Server Database dialog box in Power BI Desktop. and type in the database as AdventureWorks LT. leave the SQL statement as is. then press OK



It is very likely that you get to the window that says Unable to Connect. This window is saying that Power BI Desktop cannot connect to Azure SQL DB and the reason is that the Azure SQL Server didn't allowed your IP address to pass through its firewall. I have to mention that Azure SQL Server by default doesn't allow external IP addresses to connect to it. if you want to connect to any databases on Azure SQL Server you have to allow the IP of that machine to pass through. This is not your internal network IP, this is the IP that your internet connection has. You can find the IP easily. It is mentioned in Unable to Connect error message below!

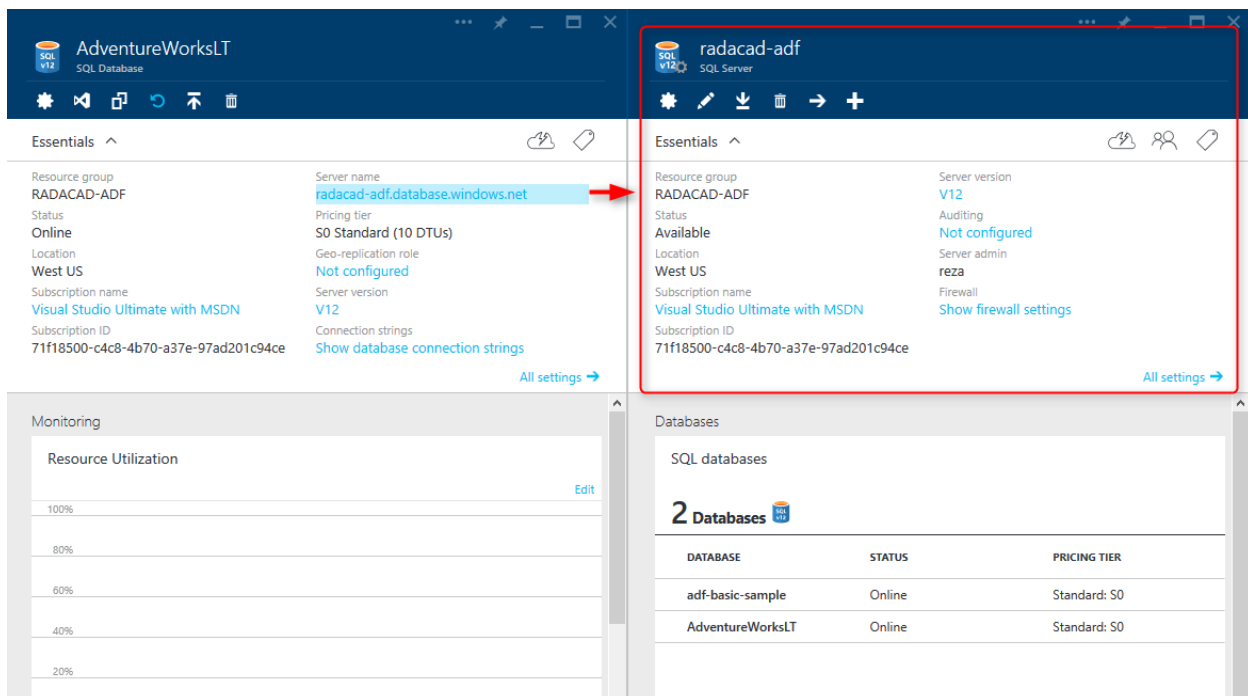
We encountered an error while trying to connect to ".....". Details: Microsoft SQL: Cannot open server'...' requested by the login. Client with IP address'210.246.15.145' is not allowed to access the server. To enable access, use the Windows Azure Management Portal or run sp_set_firewall_rule on the master database to create a firewall rule for this IP address or address range. It may take up to five minutes for this change to take effect."

Unable to Connect

We encountered an error while trying to connect to "radacad-adf.database.windows.net: AdventureWorksLT". Details: "Microsoft SQL: Cannot open server 'radacad-adf' requested by the login. Client with IP address '210.246.15.145' is not allowed to access the server. To enable access, use the Windows Azure Management Portal or run `sp_set_firewall_rule` on the master database to create a firewall rule for this IP address or address range. It may take up to five minutes for this change to take effect."

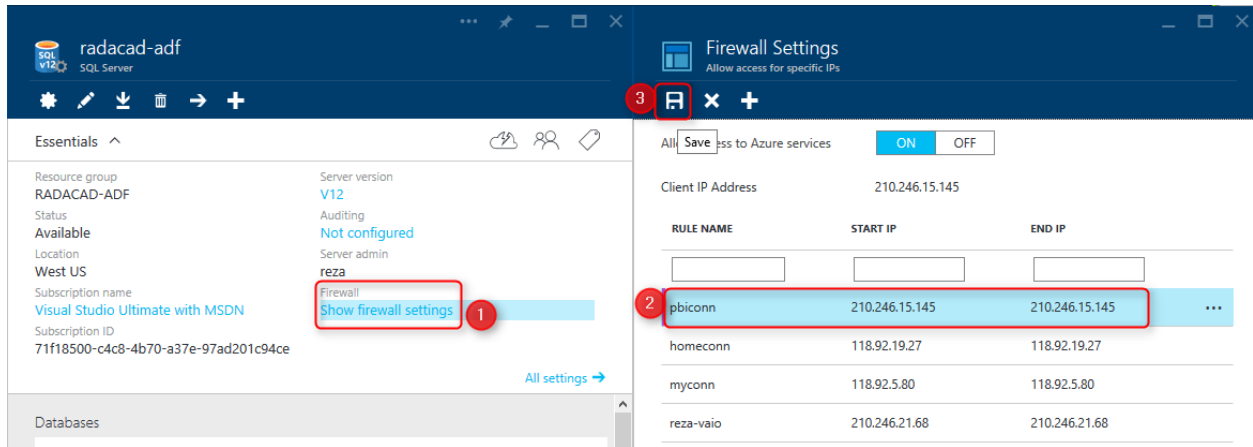
[Retry](#)
[Edit](#)
[Cancel](#)

So Let's add the IP in firewall pass list. Go to Azure Portal again. if you closed it, open it again, go to Browse All, then Choose SQL Databases, then choose AdventureWorksLT database in the list, and then click on the server name of it to open the Azure SQL Server administration pane



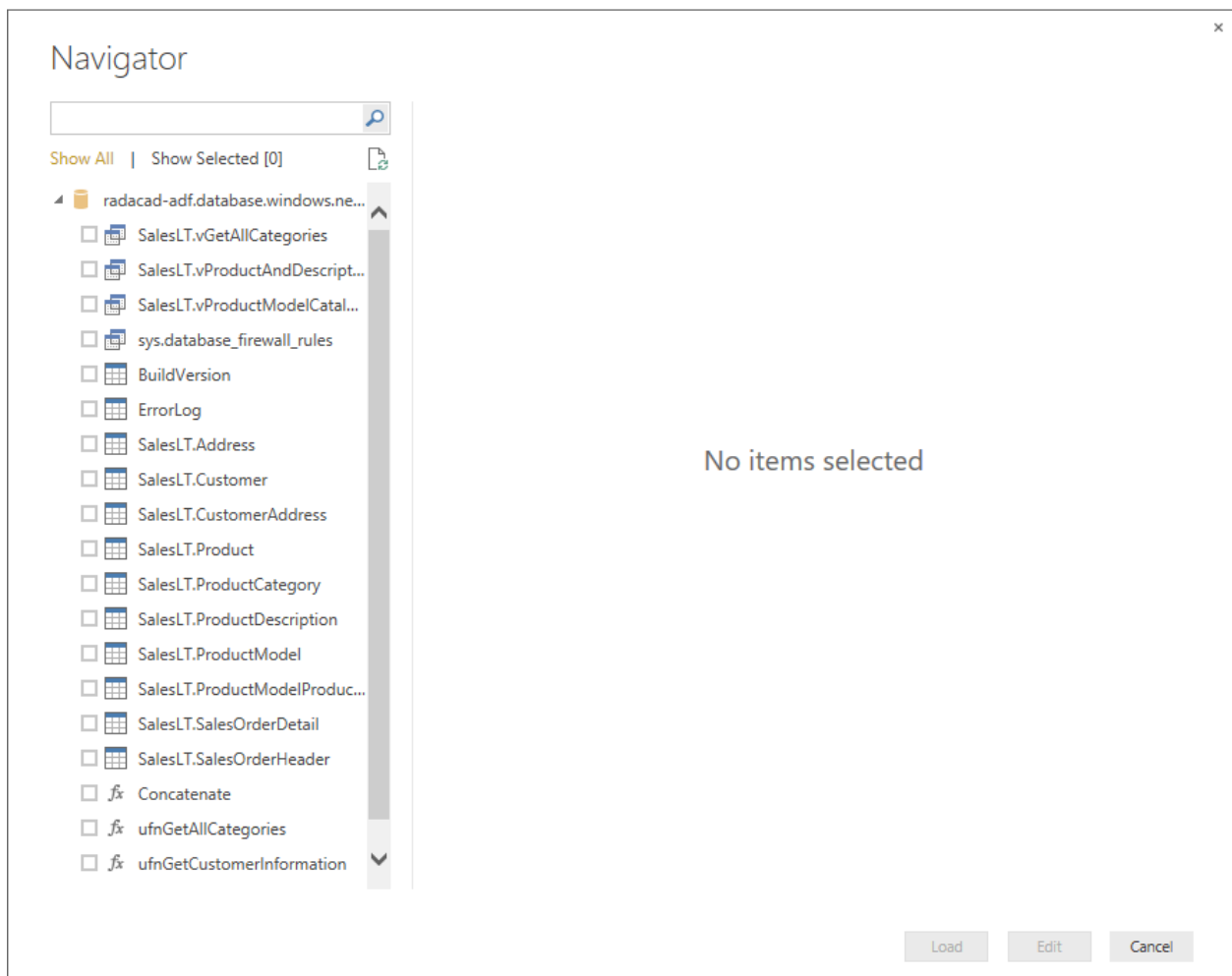
The screenshot displays the Azure SQL Server administration interface for 'radacad-adf'. The 'Essentials' pane on the right provides a summary of the server's configuration, including its resource group, status, location, and server name. A red arrow highlights the server name 'radacad-adf.database.windows.net'. Below this, the 'Monitoring' section features a 'Resource Utilization' chart. The 'Databases' section at the bottom lists the databases associated with the server, showing 'adf-basic-sample' and 'AdventureWorksLT' as online databases with a 'Standard: S0' pricing tier.

Click on Show firewall settings. In the Firewall Settings pane, enter the new IP as a rule, and then save it.



RULE NAME	START IP	END IP
pbiconn	210.246.15.145	210.246.15.145
homeconn	118.92.19.27	118.92.19.27
myconn	118.92.5.80	118.92.5.80
reza-vaio	210.246.21.68	210.246.21.68

Now you can try again to connect from Power BI Desktop (if you get that error again, just wait for few minutes and try again. Sometimes it takes few minutes for changes to take effect). After a successful connection you should be able to see Navigator dialog box with the structure of AdventureWorksLT database



Navigator

Show All | Show Selected [0]

- radacad-adf.database.windows.ne...
 - SalesLT.vGetAllCategories
 - SalesLT.vProductAndDescript...
 - SalesLT.vProductModelCatal...
 - sys.database_firewall_rules
 - BuildVersion
 - ErrorLog
 - SalesLT.Address
 - SalesLT.Customer
 - SalesLT.CustomerAddress
 - SalesLT.Product
 - SalesLT.ProductCategory
 - SalesLT.ProductDescription
 - SalesLT.ProductModel
 - SalesLT.ProductModelProduc...
 - SalesLT.SalesOrderDetail
 - SalesLT.SalesOrderHeader
 - Concatenate
 - ufnGetAllCategories
 - ufnGetCustomerInformation

No items selected

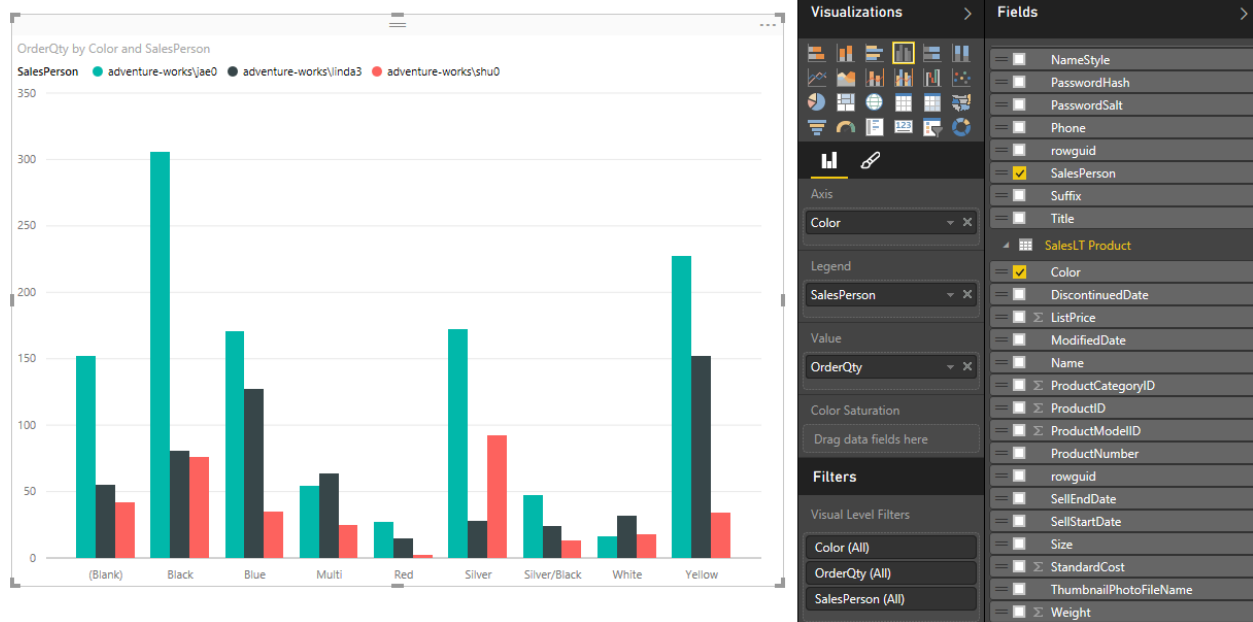
Load Edit Cancel

As you can see in the navigator all views, tables and functions will be listed. You can choose multiple objects and then continue editing them in the Edit Queries or Power Query Editor window. For this example I've chosen these tables: Customer, Product, ProductCategory, ProductModel, SalesOrderDetail, and SalesOrderHeader.

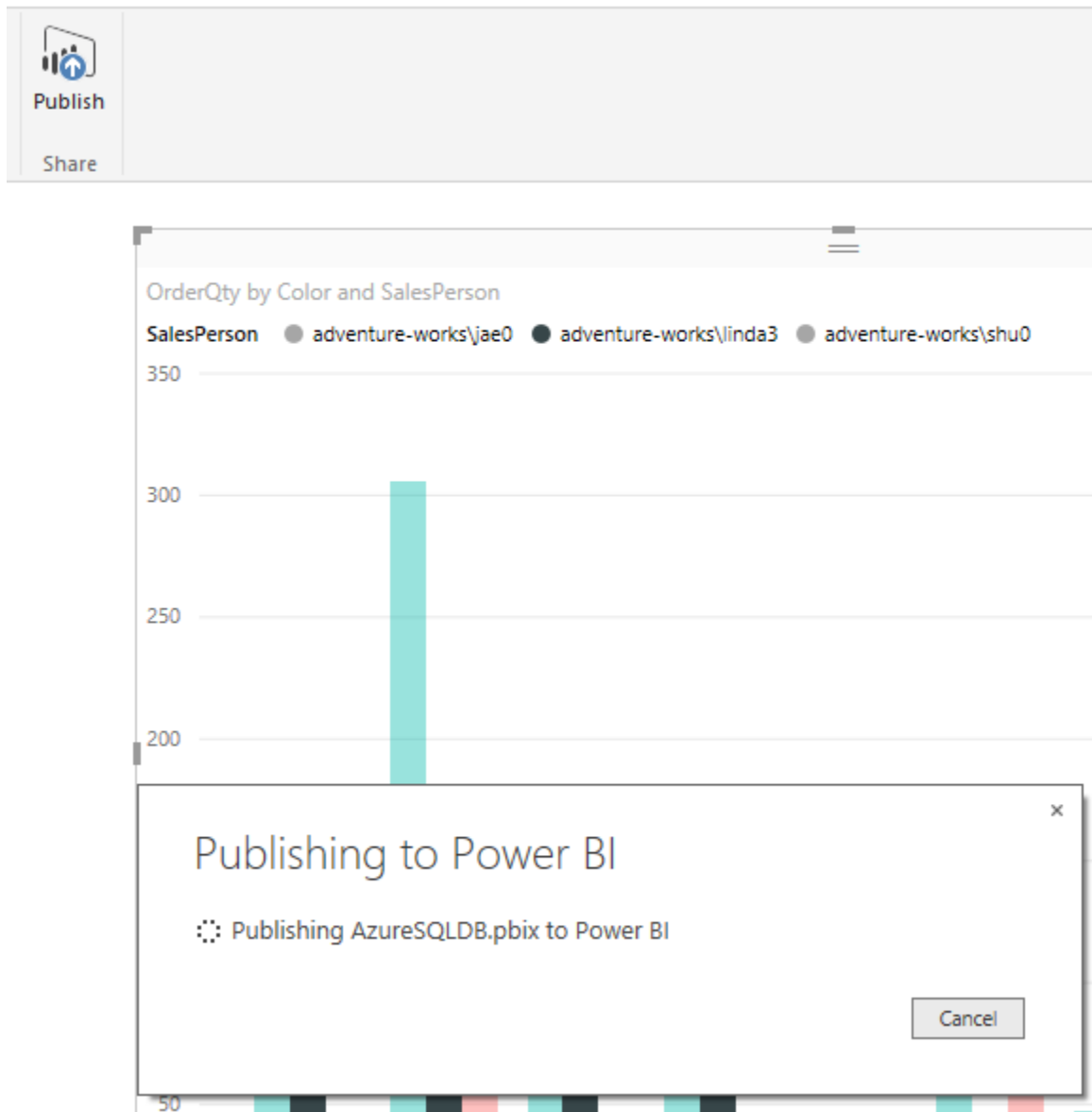
You can then choose these tables with their fields to be used for building a report without any modification in Power Query or Data tab. You can even see that Power BI and Power Query understand the relationship in Azure SQL Database and load the same relationship in the Power BI model.



This section is still one of the early chapters of Power BI online book, and I don't want to discuss visualization and modelling. However for this example I've build a simple chart, the chart is a clustered column chart with Color (from Product table) as Axis, and SalesPerson (from Customer) as Legend, and OrderQty (from SalesOrderDetail) as Value.

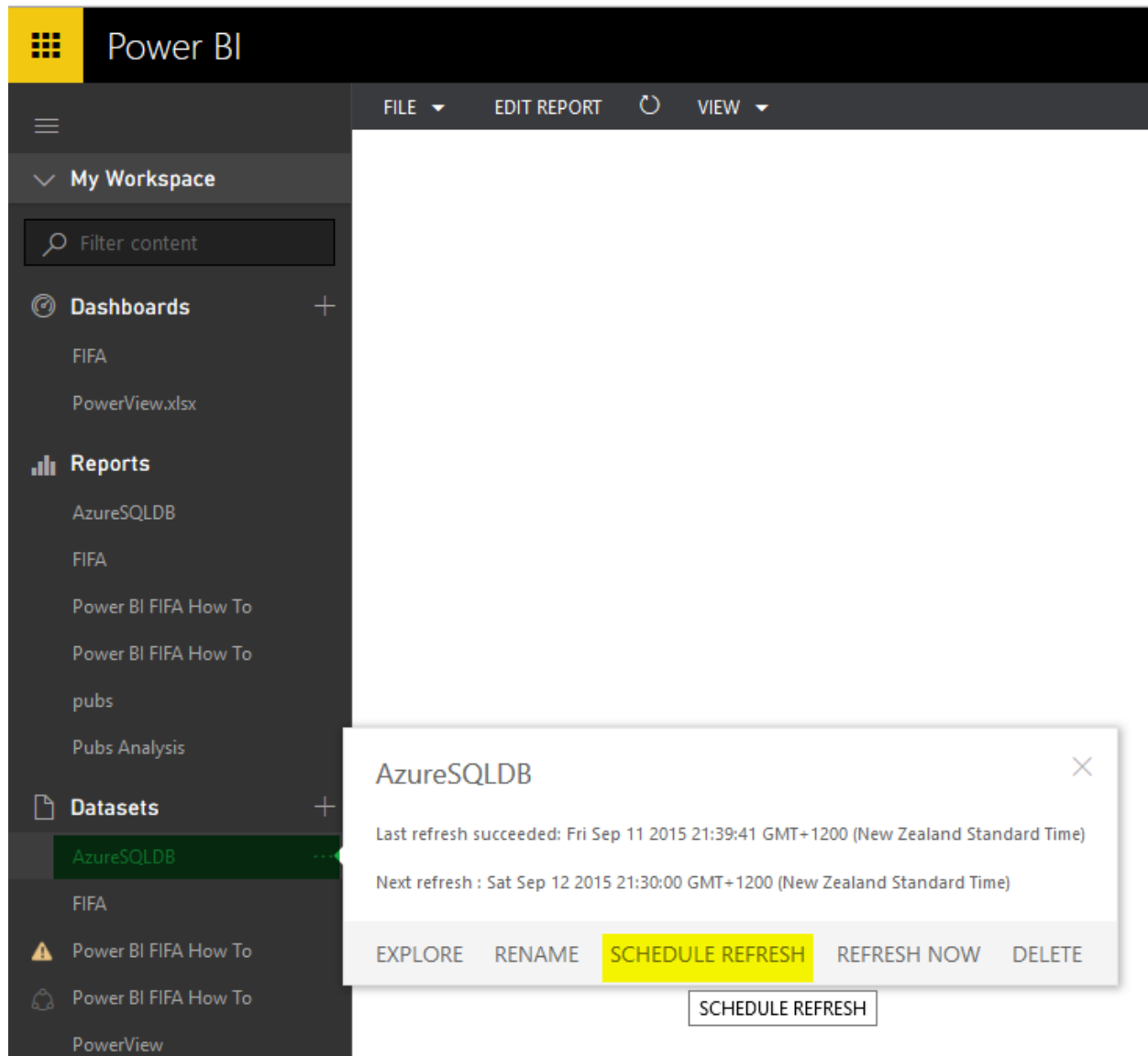


The chart is simple, but still revealing something interesting. Jae0 did the most sales, however color wise each sales person did the best in specific color. shu0 was best at Yellow. linda3 at Silver, and Jae0 at Black. Now you can publish your report into Power BI website with the Publish menu option. after publishing the report you would be able to see that under your Power BI account in the website.



Schedule Data Refresh

In Power BI website you can set up scheduled data refresh for data sets, but not all data sets supports this feature. Fortunately Azure SQL Database supports it. To find out list of all data sets that supports data refresh read [this link](#). To schedule a refresh in Power BI website, under Datasets click on ellipsis besides the data source that you want. and then choose Schedule Refresh.



Set the Data Source Credentials for Azure SQL Database

Settings

General Dashboards **Datasets** Workbooks

AzureSQLDB
FIFA
Power BI FIFA How To
PowerView
pubs
Pubs Analysis

Settings for AzureSQLDB

Last refresh succeeded: Fri Sep 11 2015 21:39:41 GMT+1200 (New Zealand Standard Time)
Next refresh: Sat Sep 12 2015 21:30:00 GMT+1200 (New Zealand Standard Time)

Data Source Credentials

AdventureWorksLT-rac

Schedule Refresh

Configure AzureSQLDB

Server

radacad-adf.database.windows.net

Database

AdventureWorksLT

Authentication Method:

Basic

Username

reza

Password

••••••••

Sign In

Cancel

And then you can schedule refresh. You can choose the frequency to be daily or weekly. and you can add multiple times on the day under that.

Settings for AzureSQLDB

Last refresh succeeded: Fri Sep 11 2015 21:39:41 GMT+1200 (New Zealand Standard Time)

Next refresh: Sat Sep 12 2015 21:30:00 GMT+1200 (New Zealand Standard Time)

▲ Data Source Credentials

AdventureWorksLT-radacad-adf.database.windows.net

[Edit credentials](#)

▲ Schedule Refresh

Keep your data up-to-date

Yes



Refresh frequency

Daily

Time Zone

(UTC+12:00) Auckland, Wellington

Time

9 30 PM X

[Add another time](#)

☒ Send refresh failure notification email to me

[Refresh History](#)

Apply

Discard

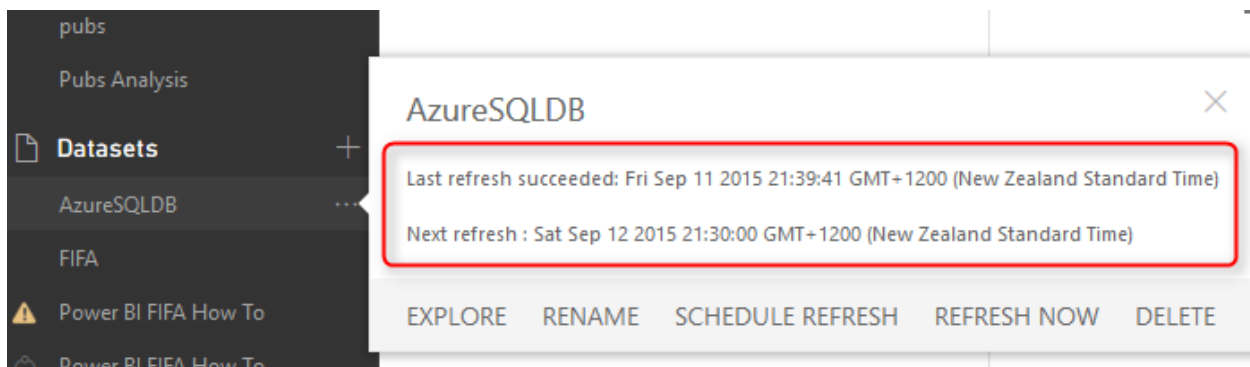
If you have a scheduled refresh set up for a while then you can see the history of refresh as well

Refresh History

Details	Type	Start	End	Status	Fail Message
	Scheduled	9/11/2015, 9:39:01 PM	9/11/2015, 9:39:41 PM	Completed	
	Scheduled	9/10/2015, 9:40:00 PM	9/10/2015, 9:40:32 PM	Completed	
	Scheduled	9/9/2015, 9:42:03 PM	9/9/2015, 9:42:28 PM	Completed	
	Scheduled	9/8/2015, 9:43:01 PM	9/8/2015, 9:43:28 PM	Completed	
	Scheduled	9/7/2015, 9:44:00 PM	9/7/2015, 9:44:29 PM	Completed	
	Scheduled	9/6/2015, 9:35:05 PM	9/6/2015, 9:35:46 PM	Completed	
	Scheduled	9/5/2015, 9:35:02 PM	9/5/2015, 9:35:30 PM	Completed	
	Scheduled	9/4/2015, 9:45:02 PM	9/4/2015, 9:45:33 PM	Completed	
	Scheduled	9/3/2015, 9:45:04 PM	9/3/2015, 9:45:57 PM	Completed	
	Scheduled	9/2/2015, 9:31:00 PM	9/2/2015, 9:31:34 PM	Completed	

Close

After setting up the schedule refresh, you can see the latest refresh and the next refresh information easily with ellipsis button on the data set

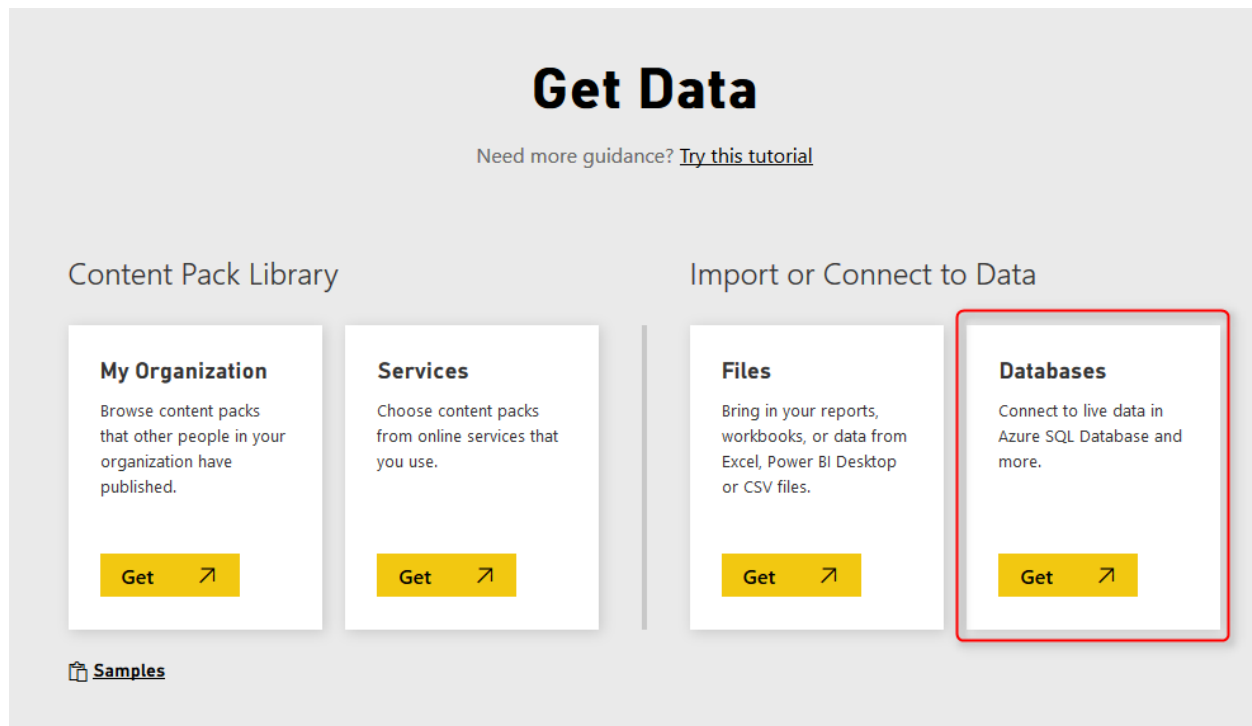


Direct Connection to Azure SQL Database from Power BI Website

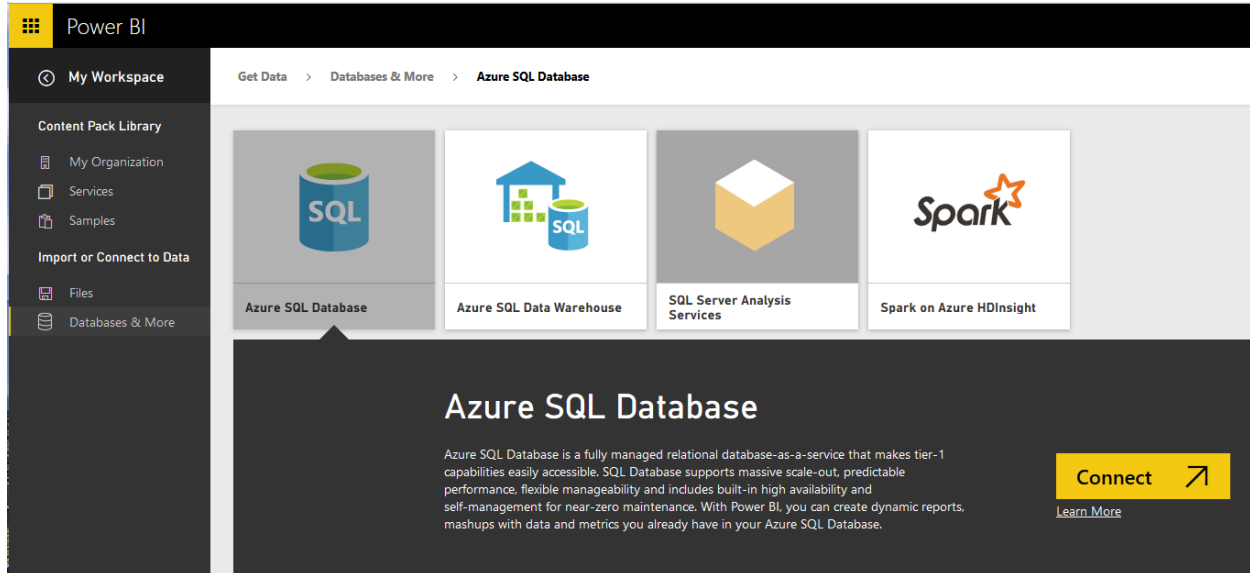
So far you've learned how to connect from Power BI Desktop or Power Query to Azure SQL database. However the connection in that way is off-line and you need to set up a schedule refresh to keep data up-to-date. Fortunately there is another way of connection which is Direct Connection. Direct connection won't load data into Power BI

model, it directly bring data into the report, and you won't need to schedule refresh anymore, because data is always up-to-date. There will be however a lag for loading the report depends on the data and volume required loaded in the report.

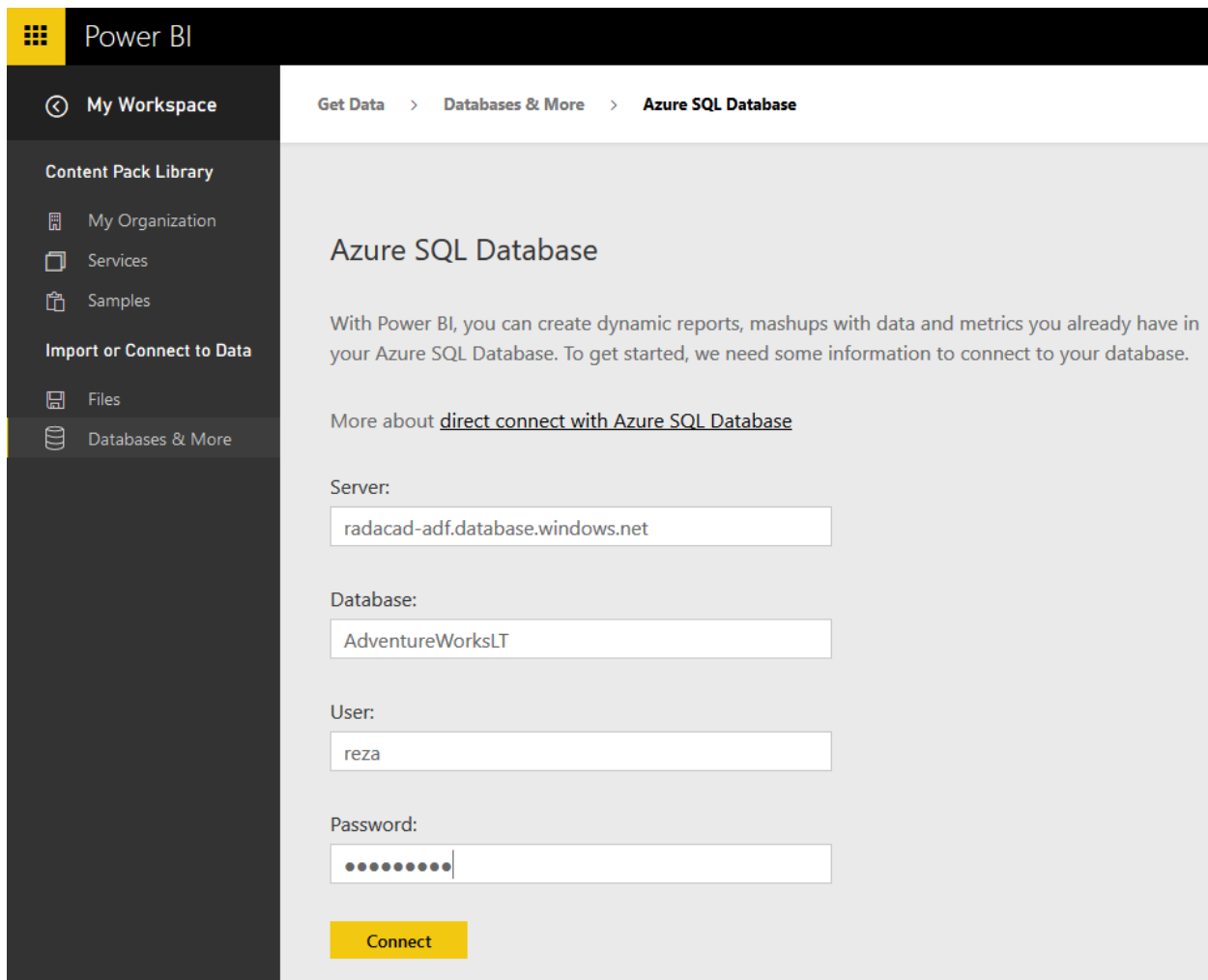
You can set Direct connection only from Power BI website at the moment. And this is one of Power BI Pro features, so you need to buy the pro plan (which costs 9.99\$ per user per month at the moment). To set a direct connection, click on Get Data in Power BI website. you will be redirected to Get Data Page



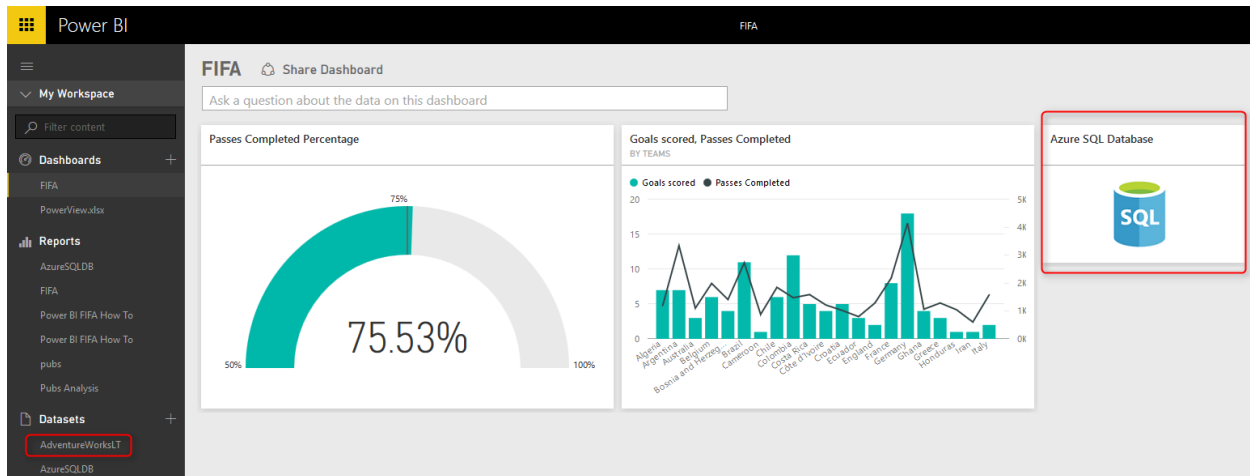
Click on the Get option from Databases. You can see that some databases are supported through this type of connection; Azure SQL Database, Azure SQL Data Warehouse, SQL Server Analysis Services, and Spark on Azure HDInsight. Click on Azure SQL Database and then Connect.



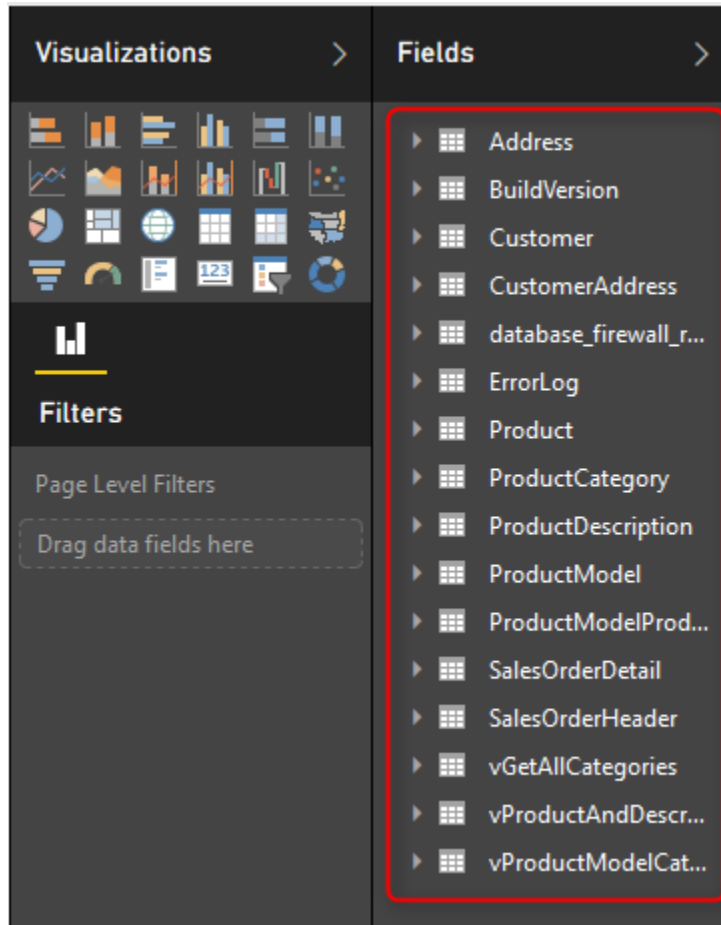
Set the connection information such as server, database name, and user and password



After creating the connection you will see Azure SQL Database in your dashboard and also the dataset in list of datasets.



by click on any of these items you will be redirected to online Power BI report designer. You can see that all tables and views now are listed in the Fields pane, and ready for you to build reports from them.

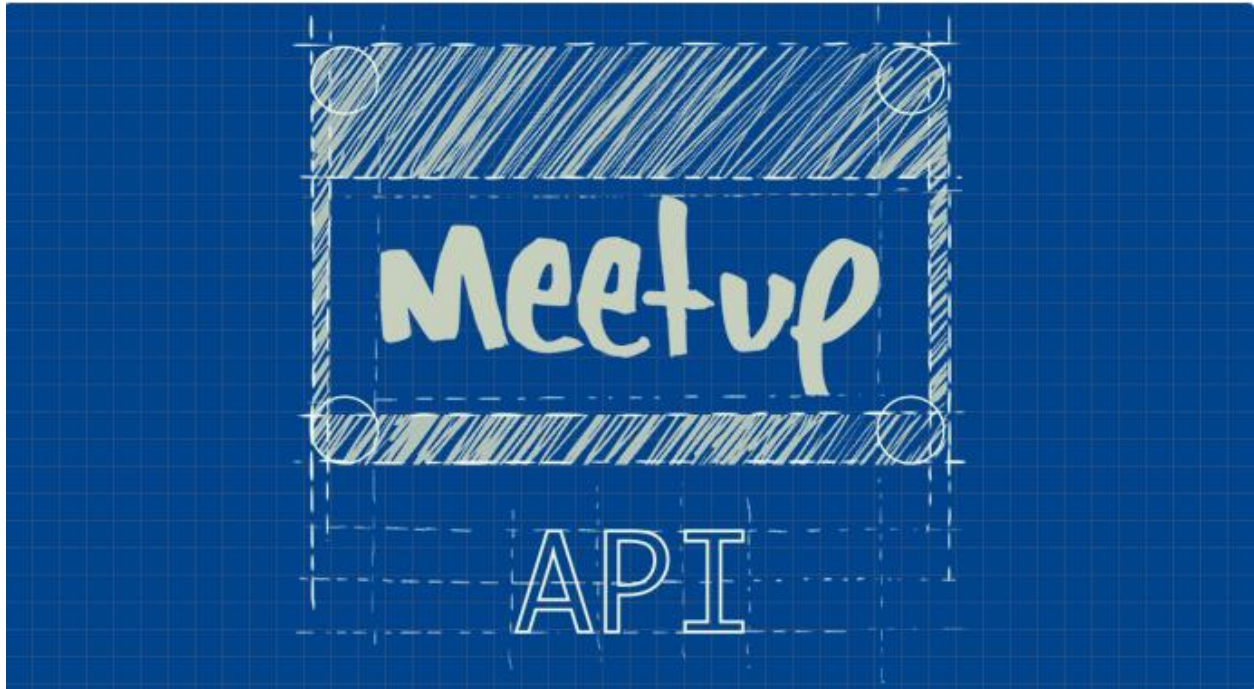


Summary

In this lesson you've learned how to connection from Power BI Desktop or Power Query in Excel to Azure SQL Database. You've also learned some basics about Azure SQL Database settings in Azure Portal. Then you've learned that you need to set firewall in Azure Portal for server to pass your IP for connections. You then learned how you can set scheduled refresh for the data in Power BI website. At the last section you've learned that you can create direct connection from Power BI website to Azure SQL Database that doesn't load data into a model and works online with the data. In next sections of this chapter you will learn about other data sources that you can connect from Power BI.

Meetup Data Source for Power BI

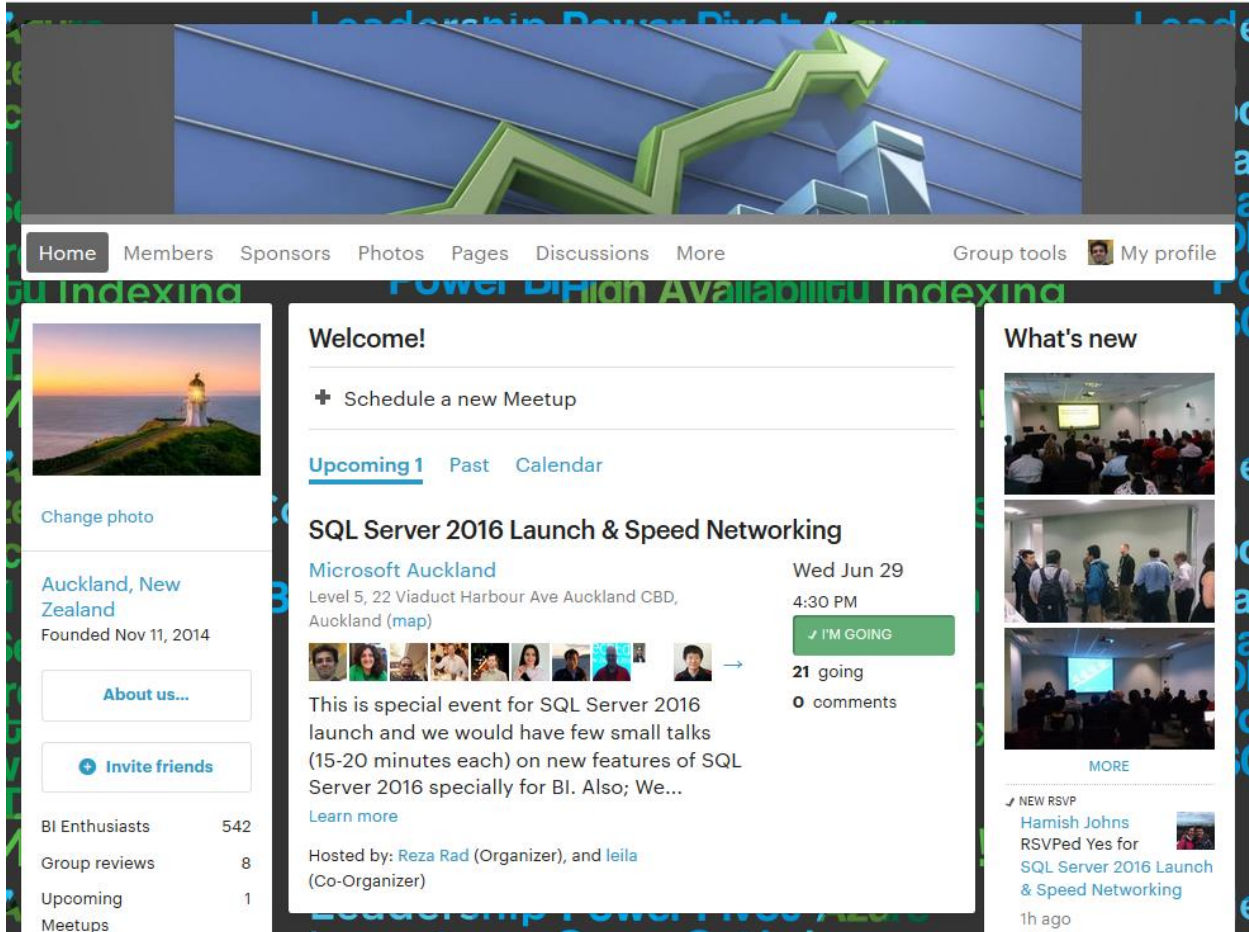
Published Date: June 2, 2016



There are many SaaS (Software as a Service) data sources for Power BI, such as Mailchimp, Salesforce, and etc. However I haven't found anything for Meetup. Meetup on the other hand is a data source that event organizers such as myself use mostly. In this post I will be using Meetup API to connect from Power BI Desktop and read the JSON output of Meetup API to build some nice visualizations in Power BI. You can read more about Power Query in [Power BI online book](#) from Rookie to Rock Star.

About Meetup

If you haven't heard about [Meetup](#) so far; it is a platform for scheduling in-person events. As an organizer I announce meetings for my Meetup group (such as [New Zealand BI user group](#)), and Meetup helps with the registration for the event. As an audience I can join to groups in my area of interest and be informed about their upcoming meetings and RSVP those to attend or not attend. Meetup website itself gives some useful information to organizers, however as a Power BI geek I would love to dig into its data with a better tool.



The screenshot shows a Meetup group page for 'Auckland, New Zealand'. The group was founded on Nov 11, 2014, and has 542 members. The main event featured is 'SQL Server 2016 Launch & Speed Networking' by Microsoft Auckland, scheduled for Wednesday, June 29, at 4:30 PM. The event description mentions small talks on new features of SQL Server 2016. The page also includes a 'What's new' section with recent RSVPs and a 'Welcome!' section with a 'Schedule a new Meetup' button.

Welcome!

+ Schedule a new Meetup

Upcoming 1 Past Calendar

SQL Server 2016 Launch & Speed Networking

Microsoft Auckland

Level 5, 22 Viaduct Harbour Ave Auckland CBD, Auckland (map)

Wed Jun 29
4:30 PM

I'M GOING

21 going
0 comments

This is special event for SQL Server 2016 launch and we would have few small talks (15-20 minutes each) on new features of SQL Server 2016 specially for BI. Also; We...

[Learn more](#)

Hosted by: [Reza Rad](#) (Organizer), and [leila](#) (Co-Organizer)

What's new

NEW RSVP
[Hamish Johns](#)
RSVPed Yes for
SQL Server 2016 Launch
& Speed Networking
1h ago

Auckland, New Zealand
Founded Nov 11, 2014

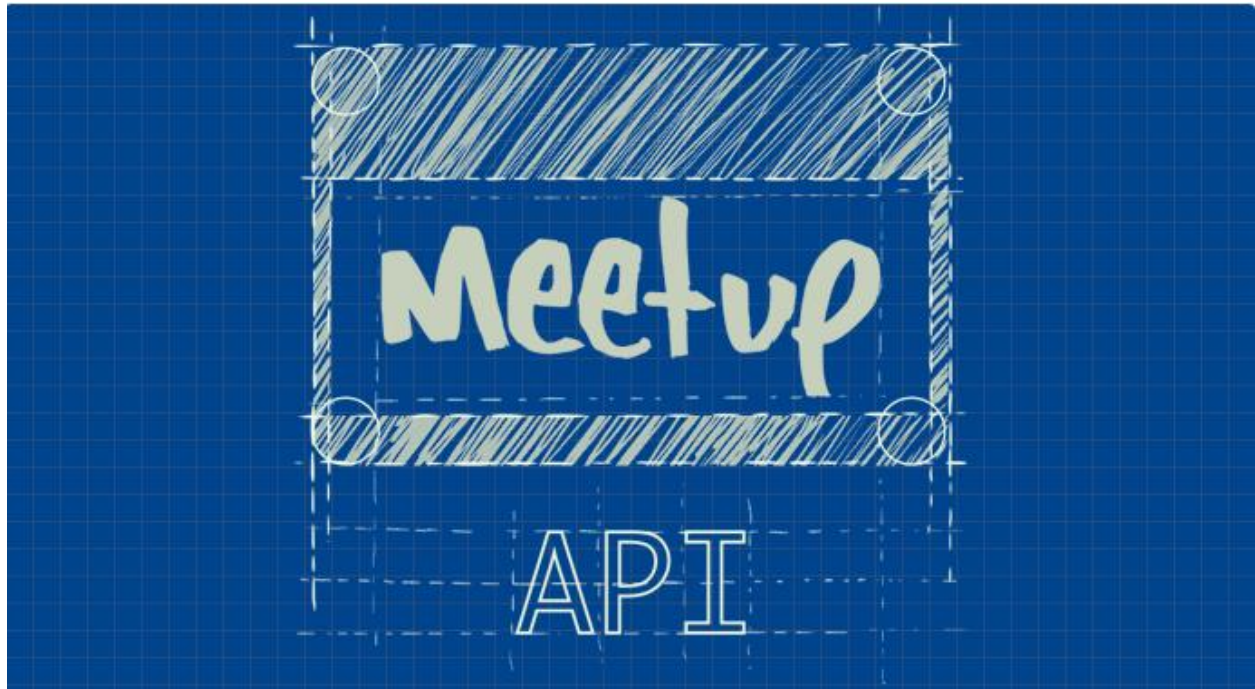
Change photo

About us...


Invite friends

BI Enthusiasts 542
Group reviews 8
Upcoming Meetups 1

Fortunately [Meetup has a set of API](#) restful services which returns output as JSON format.




Restful service means there are some URLs that I can browse and it gives me a response in JSON format. First of all I need to use my API key. API key would be different for anyone, and sorry I cannot share my API key with you, that's why it is blurred in screenshot below. You can access your API key from [here](#).

 **Meetup API** Documentation Console **API Key** OAuth Consum

Getting an API Key

Hello, Reza Rad, click the lock below to reveal your API key

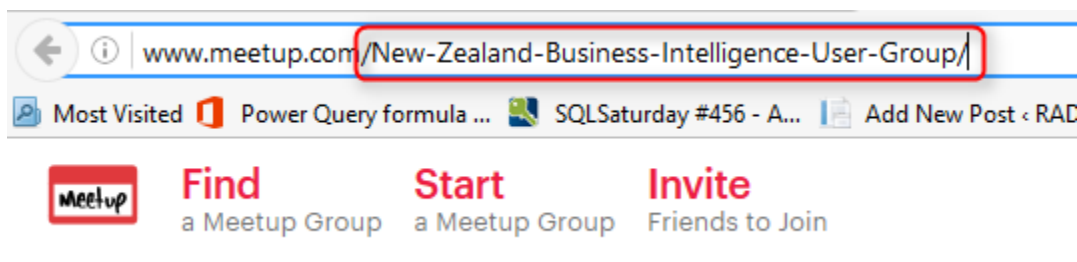
..... 

You'll need to provide this key with every request you make to t
you need to reset your key for any reason, you can do that below
with the old key to fail.

Now I can call each service with this key. Fortunately there is a good documentation on each service, input parameters, and output result set [here](#). As an example if I want to get list of all meetup events in the past for my group, I can use [V2 Events service](#) with input parameters such as URL name of the group, and status to be "past" (means only fetch past meetings);

`https://api.meetup.com/2/events?status=past&group_urlname=New-Zealand-Business-Intelligence-User-Group&key=XYZ`

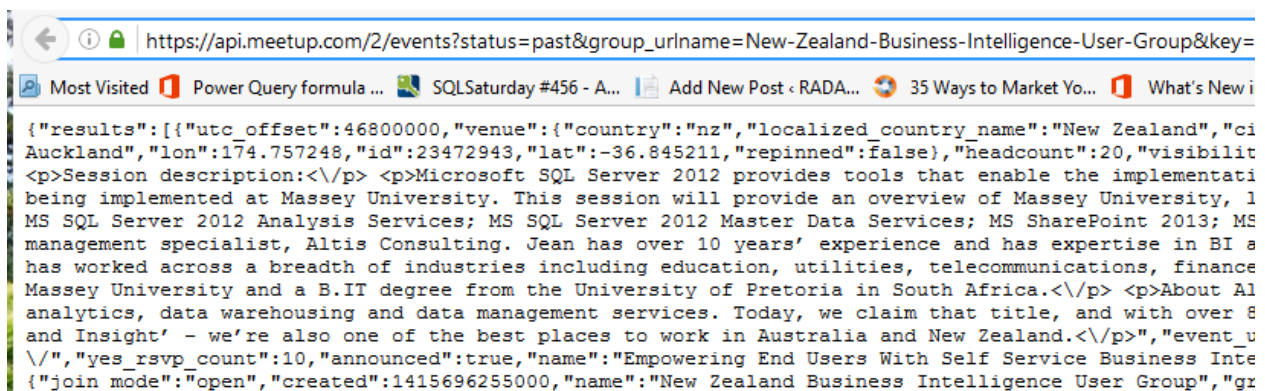
Note that group_urlname of my Meetup group can be easily found on my group web url:



as per screenshot above my group url name is: New-Zealand-Business-Intelligence-User-Group

and you have to use your own API Key instead of XYZ in URL above.

So if I browse URL above it gives me a JSON result. JSON is a format for data which usually doesn't have enter or space between fields, so as a result I would see a big text result like this:

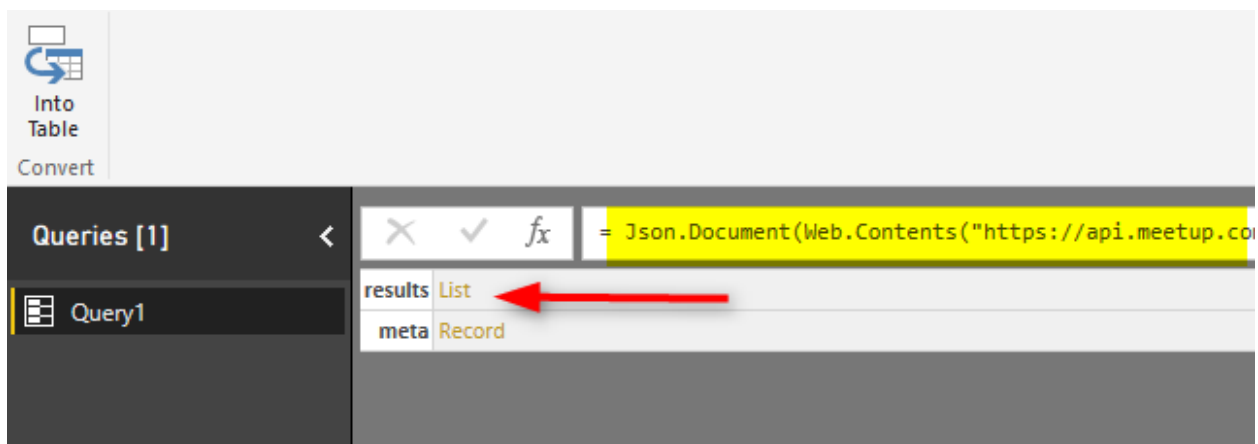


Get Data with Power BI

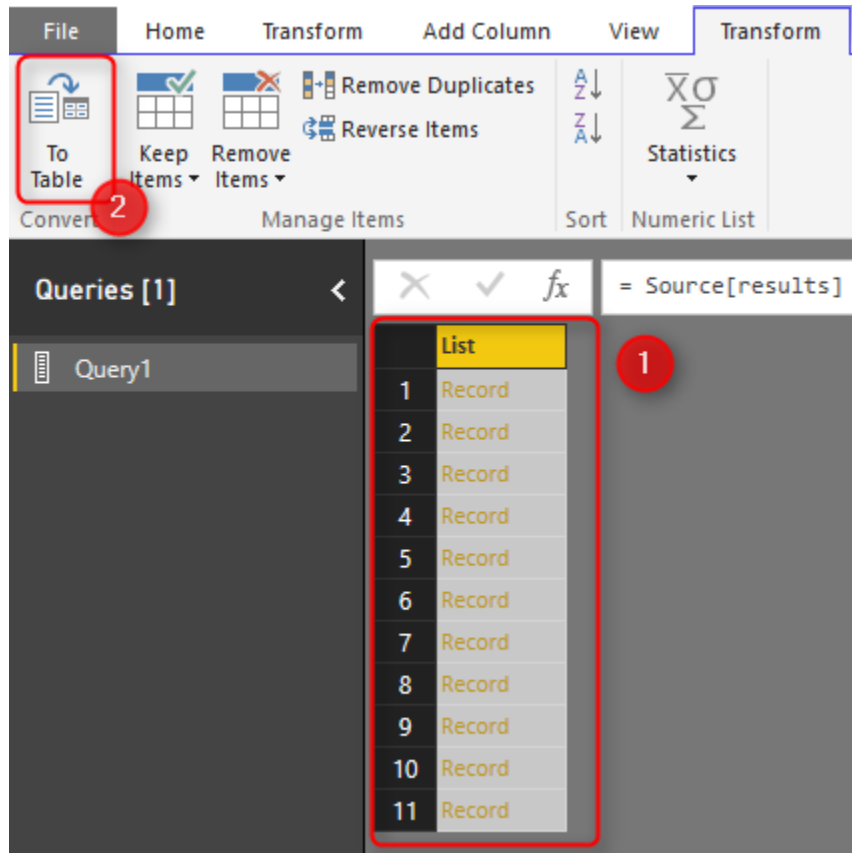
Now let's browse URL above in Power BI and Power Query to see what can I do with it. I open a Power BI Desktop file, and Get Data from Web. and I enter URL above there (with my own API key obviously);




Result of this will show me a record in Power Query editor window with results and meta columns. You can also see in screenshot below that Power Query uses `Json.Document` and `Web.Contents` to get JSON result from a web URL.



Now click on the results section List to expand that. This will drill down into the list and you will see a list with multiple items in the next step. This list is result set in our JSON data. the result set of the URL we run returns list of events. That means this list is list of events, and each row in this list is an event. As you can see I have a record in each row. I can convert the main list to table with first highlighting the whole column, and then click on To Table option from Transform menu in Power Query.



When I click on To Table, I will see To Table dialog box that asks for two configuration options, first one for delimiter, and second one for handling extra columns.



To Table

Create a table from a list of values.

Select or enter delimiter

None

How to handle extra columns

Show as errors

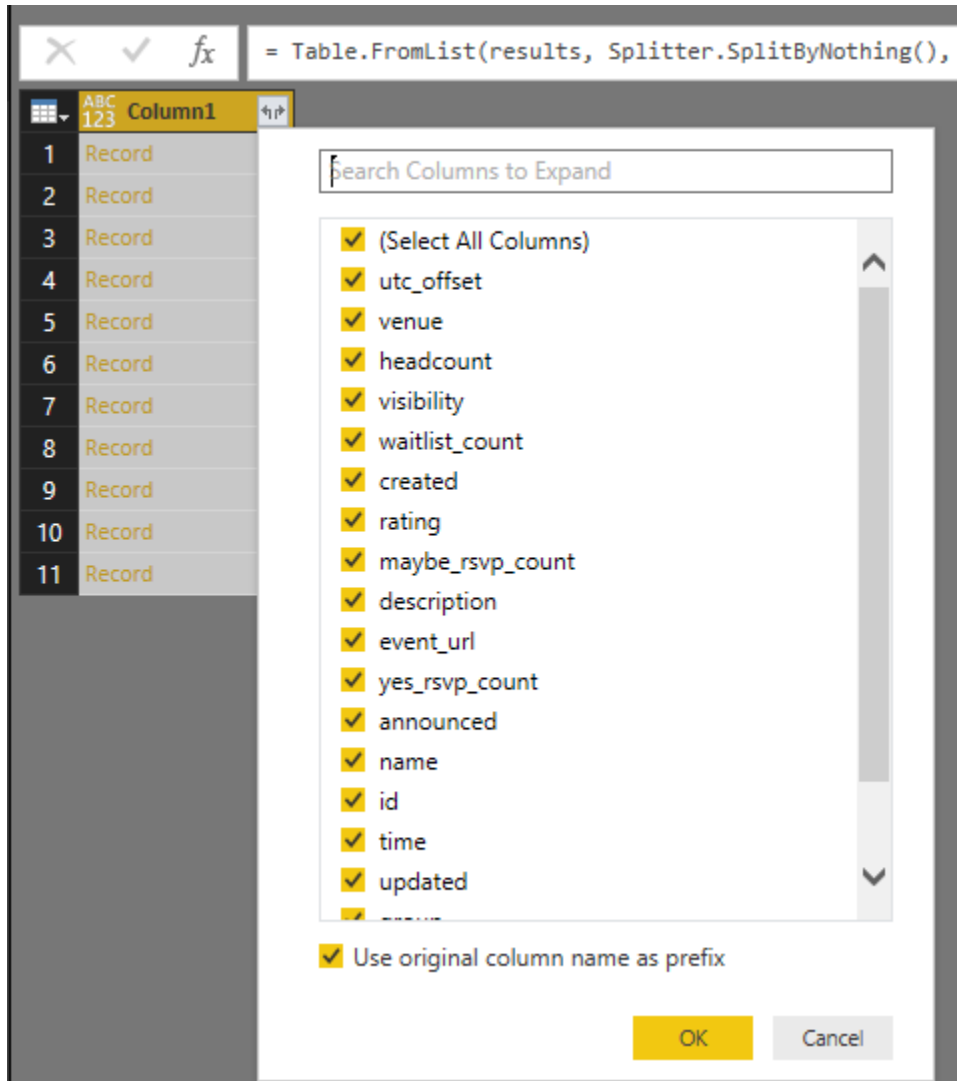
OK

Cancel

I don't do any changes, and click on OK with default values. Result now would be a table with single column, which has a record in each row. I can see the Expand button on my column1 header.



Click on expand button as mentioned in screenshot above, and it will list all columns in records and asks me to choose those I want.



Well let's bring them all to see what we have, so I leave it with select all and click on OK. Here is my result set loaded in Power Query now:

	Column1.utc_offset	Column1.venue	Column1.headcount	Column1.visibility	Column1.waitlist_count	Column1.created	Column1.rating	Column1.maybe_rsvp_count	Column1.description
1	46800000	Record	20	public	0	1.4157E+12	Record	0	<p>Session description</p> <p>Micro
2	43200000	Record	0	public	0	1.42776E+12	Record	0	<p>Session Description</p> <p>This p
3	43200000	Record	0	public	0	1.43121E+12	Record	0	Speaker</p> <p>Darryl Wolfardt</p>
4	43200000	Record	0	public	0	1.43566E+12	Record	0	<p>to>Session Details</p> <p>In
5	43200000	Record	0	public	0	1.43851E+12	Record	0	<p>Session Description</p> <p>Mach

Now I have event information such as Name, Description, venue and other fields. Some values are a record by themselves, such as Venue and Rating. I can expand them if I want to. Let's leave it as is for now. I've loaded my first service result into Power Query. Now one important thing before any other step is to check time value. Because I want to

do date/time analysis for my events, so it is important to see if I have my time value fetched. in screenshot below it shows my Time column as a numeric value.

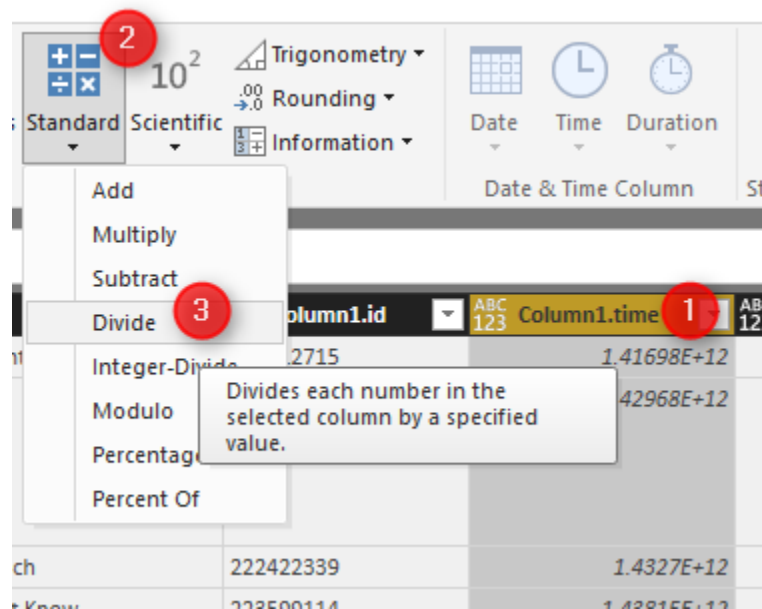
ABC 123 Column1.name	ABC 123 Column1.id	ABC 123 Column1.time	ABC 123 Column1.updated
Empowering End Users With Self Service Business Intelligence	218612715	1.41698E+12	1.46412E+12
BIML is the shizzle!!	221511579	1.42968E+12	1.42969E+12
Show Me Potential Customers: Data Mining Approach	222422339	1.4327E+12	1.43286E+12
Top 5 Functionalities of Power Query that You Don't Know	223599114	1.43815E+12	1.43816E+12
Be Smarter with Azure Machine Learning	224346643	1.44057E+12	1.44221E+12
SQL Saturday Auckland	224997769	1.44382E+12	1.44391E+12

TimeStamp to Date Time

In the documentation of this service in Meetup mentioned that Time column is:

time = UTC start time of the event, in milliseconds since the epoch

That means it is timestamp formatted. Timestamp value is number of seconds from epoch which is 1970-01-01 00:00:00. [I have previously written about how to change timestamp value to date time](#) and it is fairly easy with adding seconds to it. However for this case our value is not seconds, it is milliseconds so I have to first divide it by 1000.



This makes my number smaller now, and I can now feed it into adding a custom column as Event Time as below;

```
#datetime(1970,1,1,0,0,0)+#duration(0,0,0,[Column1.time])
```

Add Custom Column

New column name

Event Time

Custom column formula:

```
=#datetime(1970,1,1,0,0,0)+#duration(0,0,0,[Column1.time])
```

Available columns:

Column1.announced
Column1.name
Column1.id
Column1.time
Column1.updated
Column1.group
Column1.status
Column1.duration

<< Insert

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

OK

Cancel

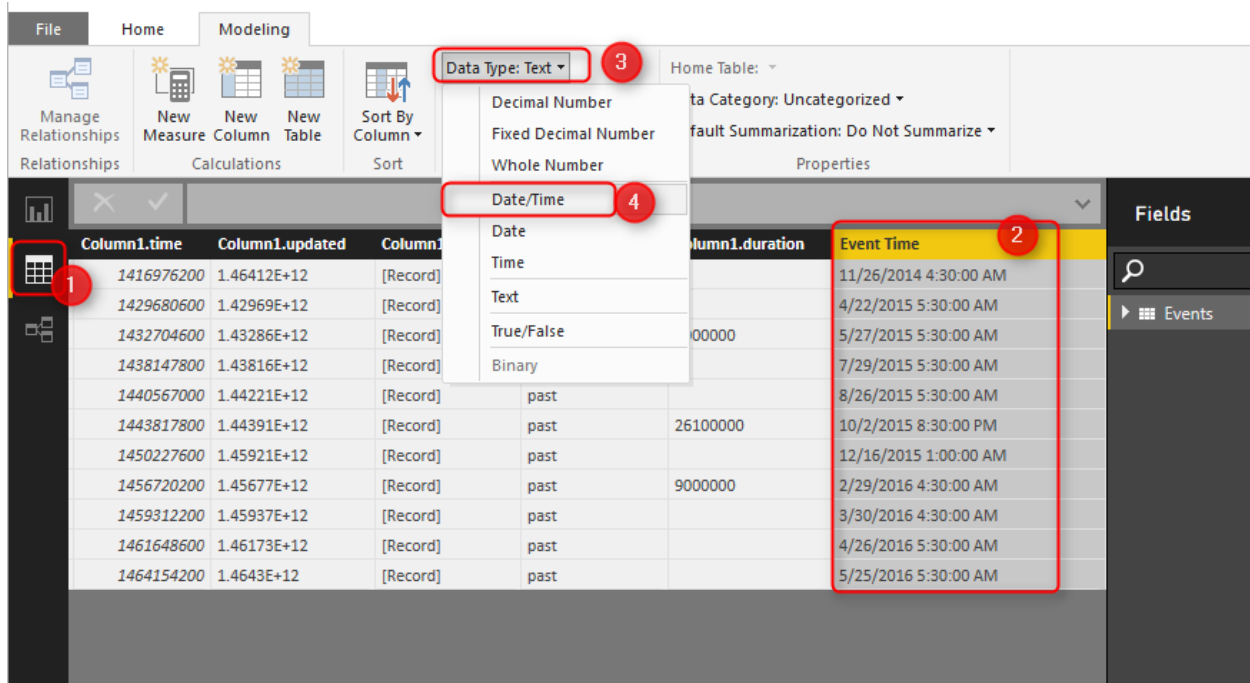
If you want more information about this conversion read my blog post [here](#). Now this will give me a date time value as below:

ABC 123	Event Time
I	11/26/2014 4:30:00 AM
I	4/22/2015 5:30:00 AM
J	5/27/2015 5:30:00 AM
I	7/29/2015 5:30:00 AM
I	8/26/2015 5:30:00 AM
J	10/2/2015 8:30:00 PM

Please note that times are UTC times, We never have meetings 4 AM! I won't be changing UTC time for now, because I want more Month analysis and this is fine for that analysis.

Visualization

Let's visualize what we've got so far. First step I have to make sure the data type of my column in Power BI model loaded as Date Time. I can go to Data Tab, click on my Event Time column and under Modeling tab check the data type, and if it is text or anything else, change it to Date/Time

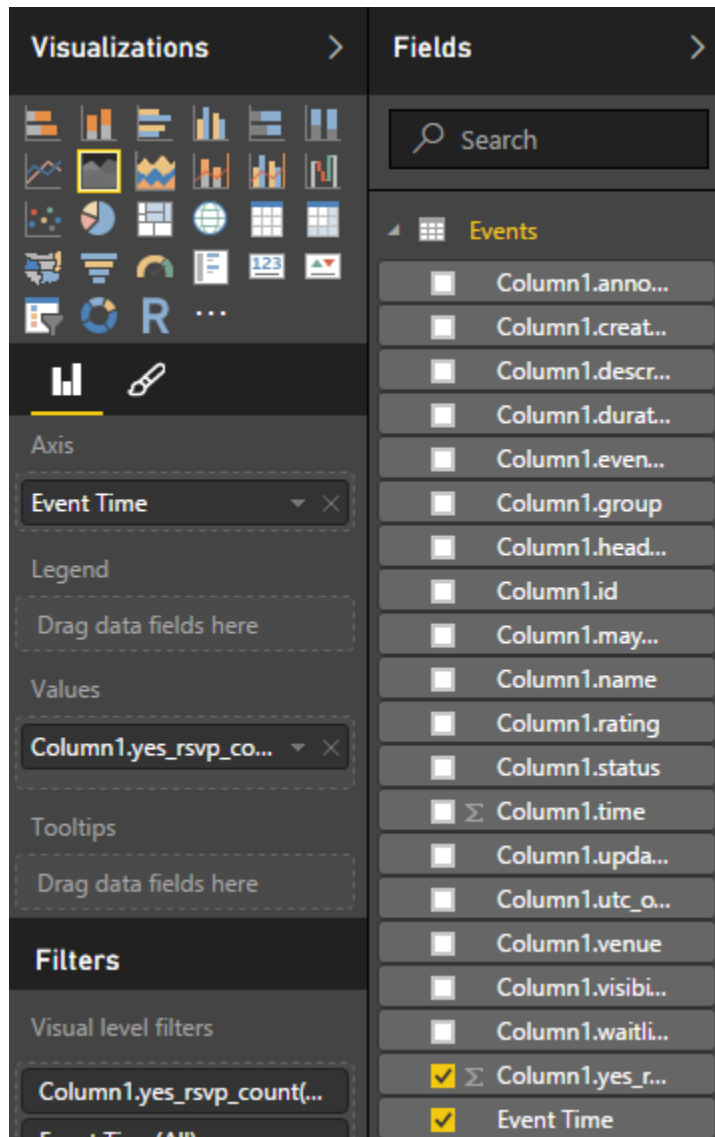


a warning message will be displayed for reports that already using this field, as I don't have any report elements yet, I'll confirm and continue.

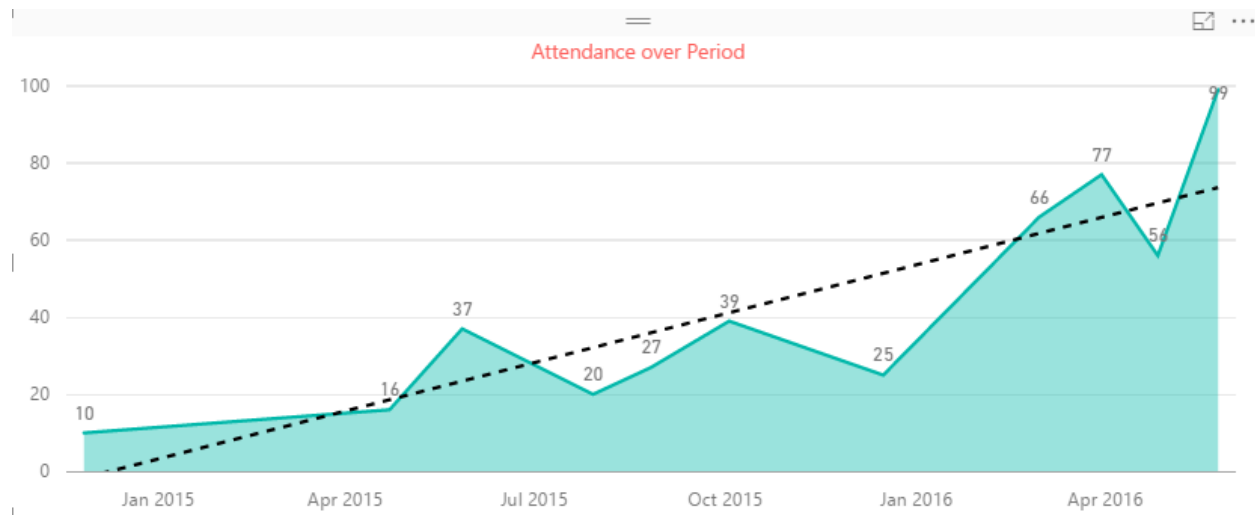
I also have to make sure that yes_rsvp_count column (which is showing number of audience for each session) set as data type whole number. If it is not the case, then I change it to be whole number as below.

Data Type: Whole Number ▾ Format: Whole Number ▾ \$ ▾ % ▾ , ▾ .00 0 ▴ ▾ Formatting		Home Table: ▾ Data Category: Uncategorized ▾ Default Summarization: Sum ▾ Properties
Column1.event_url		Column1.yes_rsvp_count
er 2012 provid	http://www.meetup.com/New-Zealand-Business-Intelligence-User	10
will be a deep c	http://www.meetup.com/New-Zealand-Business-Intelligence-User	16
irketing depart	http://www.meetup.com/New-Zealand-Business-Intelligence-User	37
ic nowadays, P	http://www.meetup.com/New-Zealand-Business-Intelligence-User	20
rovides a facili	http://www.meetup.com/New-Zealand-Business-Intelligence-User	27
rofessionals an	http://www.meetup.com/New-Zealand-Business-Intelligence-User	39
ent for 2015, a	http://www.meetup.com/New-Zealand-Business-Intelligence-User	25
ptional opport	http://www.meetup.com/New-Zealand-Business-Intelligence-User	66
een Microsoft S	http://www.meetup.com/New-Zealand-Business-Intelligence-User	77
ous who would	http://www.meetup.com/New-Zealand-Business-Intelligence-User	56
you can Do wit	http://www.meetup.com/New-Zealand-Business-Intelligence-User	99

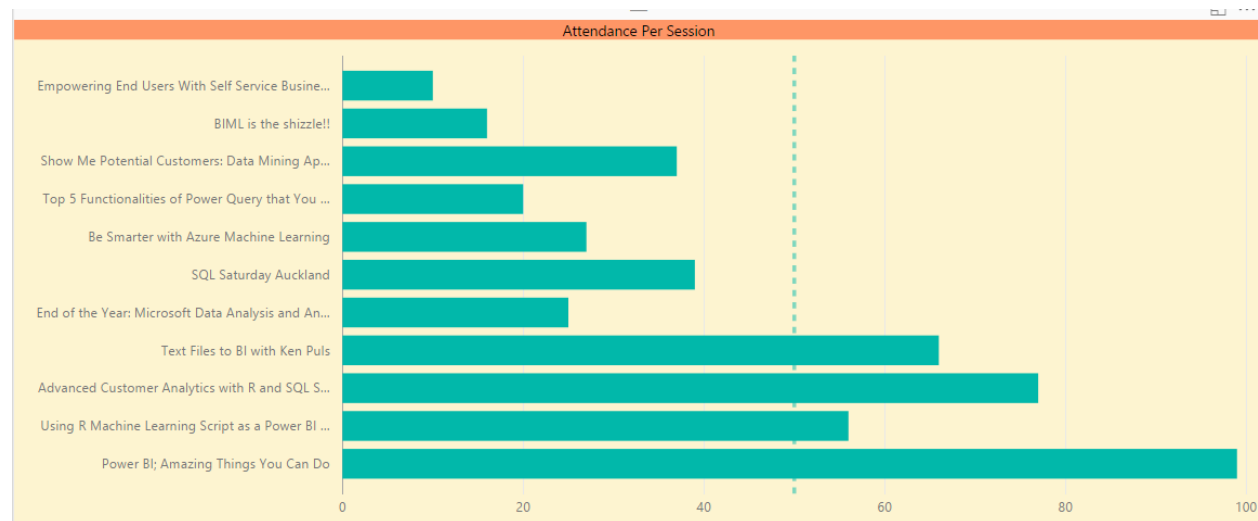
Let's build charts now. I start by an Area Chart showing Event Time as Axis (without hierarchy), and yes_rsvp_count as values



I also apply some formatting and build a chart like this:



Well, it shows our group is performing well, number of audience raised from 10 to 99! and trend line shows that we are progressing better and better. I can build some other visualizations such as bar chart with name of events and number of RSVP yes as below;



I can see the very first event that had more than 50 audience was the event that my friend [Ken Puls Excel MVP](#) from Canada came here to speak in our user group, which was kind of a start into our popular events afterwards.

Adding More Data

I can add more data to build more meaningful insight out of it. For example I can fetch Members information to show their locations (city and country) on the map. For this I use Members API service which I can call as below;

https://api.meetup.com/members?group_urlname=New-Zealand-Business-Intelligence-User-Group&key=XYZ

fx = Table.ExpandRecordColumn("#Converted to Table", "Column1",

	Column1.zip	Column1.country	Column1.city	Column1.joined	Column1.topics
1	meetup1	nz	Auckland	Thu Nov 26 17:29:49 EST 2015	List
2	meetup1	nz	Auckland	Wed Apr 08 06:49:50 EDT 2015	List
3	meetup1	nz	Auckland	Thu Dec 17 02:29:11 EST 2015	List
4	meetup1	nz	Auckland	Sun May 24 23:49:20 EDT 2015	List
5	meetup1	nz	Auckland	Mon May 02 21:29:32 EDT 2016	List
6	meetup1	nz	Auckland	Tue Jan 19 06:24:15 EST 2016	List
7	1150	nz	Auckland	Thu Jan 28 12:06:41 EST 2016	List
8	meetup1	nz	Auckland	Thu Jun 25 20:22:03 EDT 2015	List
9	meetup1	nz	Auckland	Sat Apr 16 18:11:17 EDT 2016	List
10	meetup1	nz	Auckland	Thu Nov 12 04:05:07 EST 2015	List
11	meetup1	nz	Auckland	Wed Sep 09 22:32:28 EDT 2015	List

This time I haven't explained step by step, but process is similar to what we've done for Events, so I skip this part. Now I want to create a custom column concatenated of city and country as below;

Add Custom Column

New column name

City Full

Custom column formula:

= [Column1.city]&" "&[Column1.country]

Available columns:

- Column1.zip
- Column1.country
- Column1.city
- Column1.joined
- Column1.topics
- Column1.link
- Column1.bio

<< Insert

[Learn about Power BI Desktop formulas](#)

✓ No syntax errors have been detected.

OK

Cancel

I then will be using this new City Full column because it gives me more reliable map location than the city itself (because a city with same name might be part of two different countries).

RSVP Data

RSVP data for each event is really useful for me, because I can understand who attend in most of events, and then contact them and thanks them for their commitment to the group. RSVP data can be fetched for each event. So I have to check it for each event ID that what is the RSVP for it. Here is RSVP service URL:

https://api.meetup.com/2/rsvps?event_id=230678189&key=XYZ

Note that you have to use your own event_id for the URL above.

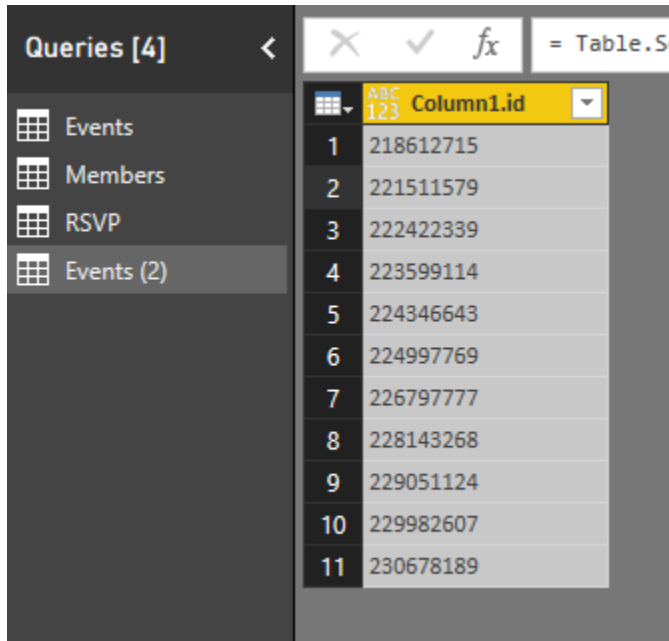
I start by fetching result of above URL into Power Query; and here is what I get after expanding the JSON list and records;

	ABC 123	Column1.mtime	ABC 123	Column1.response	ABC 123	Column1.tallies	ABC 123	Column1.guests	ABC 123	Column1.member.member...	ABC 123	Column1.event
96		1.46372E+12		no		Record		0		5915204		Record
74		1.46175E+12		yes		Record		0		66306872		Record
40		1.46174E+12		yes		Record		0		200870054		Record
21		1.46397E+12		yes		Record		0		195710156		Record
53		1.46288E+12		yes		Record		0		204872467		Record
74		1.4618E+12		yes		Record		0		160955302		Record
80		1.46268E+12		yes		Record		0		204040342		Record
49		1.46407E+12		yes		Record		0		119363682		Record
90		1.46181E+12		yes		Record		0		149123682		Record
94		1.46373E+12		yes		Record		0		155618572		Record
39		1.46414E+12		no		Record		0		160813312		Record
27		1.46173E+12		yes		Record		0		199892152		Record

Note that I've also expanded Members column to get Member ID, this is what I would use later to join to Members query. I also have a column for response which says if this member responded yes to the RSVP or not. I name this query as RSVP.

Add a Parameter and Create Function

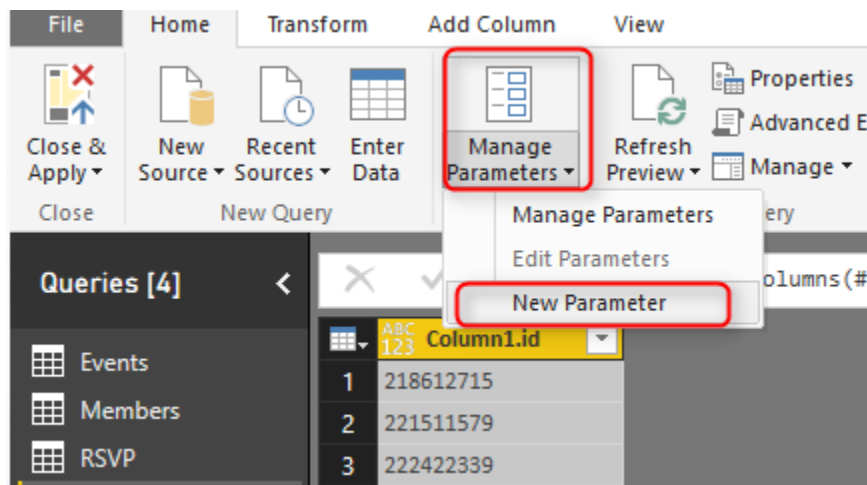
Query above is only for one event, but I am looking for data for every event. So I have to create a function for getting RSVP result and apply that on every row in Events query. I create a duplicate version of Events, and remove everything and keep only column1.id (which is event id).



The screenshot shows the Power BI Desktop interface. On the left, the 'Queries' pane lists 'Events', 'Members', 'RSVP', and 'Events (2)'. The main area displays a table with a single column named 'Column1.id'. The table contains 11 rows of numeric data.

	Column1.id
1	218612715
2	221511579
3	222422339
4	223599114
5	224346643
6	224997769
7	226797777
8	228143268
9	229051124
10	229982607
11	230678189

Now I create a parameter to make my RSVP query parametric. Click on Manage Parameters, and choose New Parameter.



I create a parameter with name EventID and default value as one of my event ids;

Parameters

New
✕

123 EventID
✕

Name

Description

☒ **Required**

Type

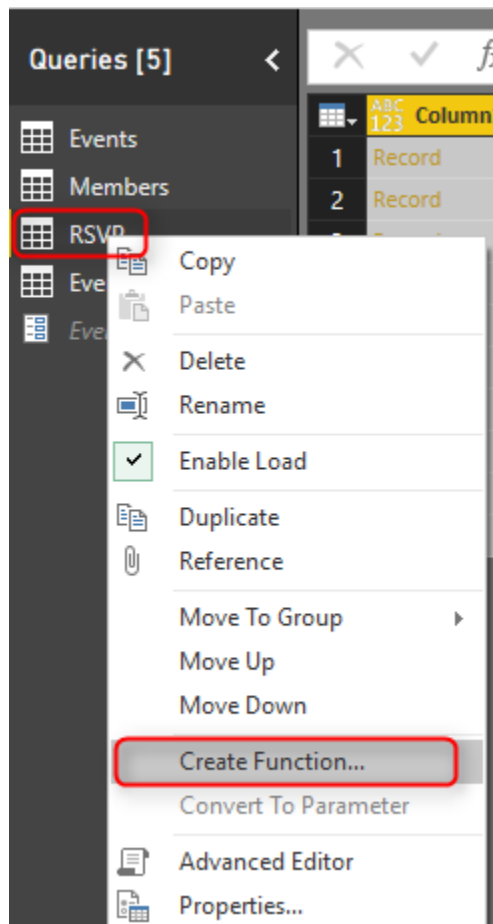
Allowed Values

Current Value


Now I go to advanced editor tab of my RSVP query, and change the script to use EventID variable as below;

```
let
    event=EventID,
    Source =
        Json.Document(Web.Contents("https://api.meetup.com/2/rsvps?event_id=" & Text.From(event) & "&key=295356c2d668094694760525e119")),
    results = Source[results],
    #"Converted to Table" = Table.FromList(results, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    #"Expanded Column1" = Table.ExpandRecordColumn(#"Converted to Table", "Column1", {"venue", "created", "member_photo", "answers", "rsvp_id", "mtime", "response", "tallies", "guests", "member", "event", "group"}, {"Column1.venue", "Column1.created", "Column1.member_photo", "Column1.answers", "Column1.rsvp_id", "Column1.mtime", "Column1.response", "Column1.tallies", "Column1.guests", "Column1.member", "Column1.event", "Column1.group"}),
    #"Expanded Column1.member" = Table.ExpandRecordColumn(#"Expanded Column1", "Column1.member", {"member_id"}, {"Column1.member.member_id"})
in
    #"Expanded Column1.member"
```

Now that my query works with parameter, I can simply right click on it, and Create Function.



I name the function as GetRSVP and it get EventID as input parameter



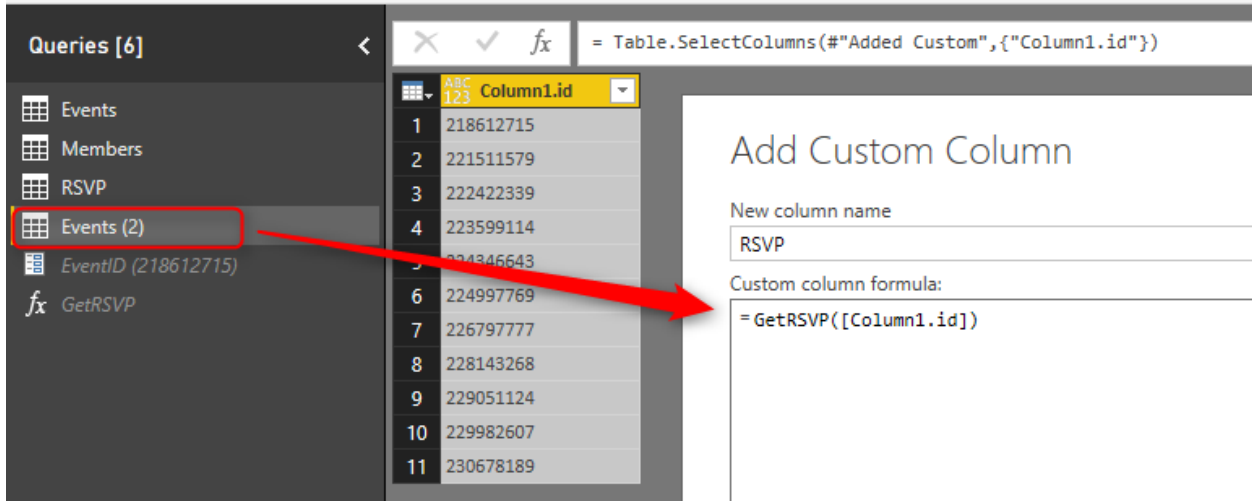
Create Function

Enter names for the new function and its parameters.

Function name

Parameter name (EventID)

Now I can go to my copy of Events query which named Events(2), and add a custom function that calls GetRSVP as below



Queries [6]

- Events
- Members
- RSVP
- Events (2)
- EventID (218612715)
- GetRSVP

Column1.id

1	218612715
2	221511579
3	222422339
4	223599114
5	224346643
6	224997769
7	226797777
8	228143268
9	229051124
10	229982607
11	230678189

Add Custom Column

New column name

RSVP

Custom column formula:

= GetRSVP([Column1.id])

Result set would be a column with tables in each row, which I can expand to get RSVP columns there

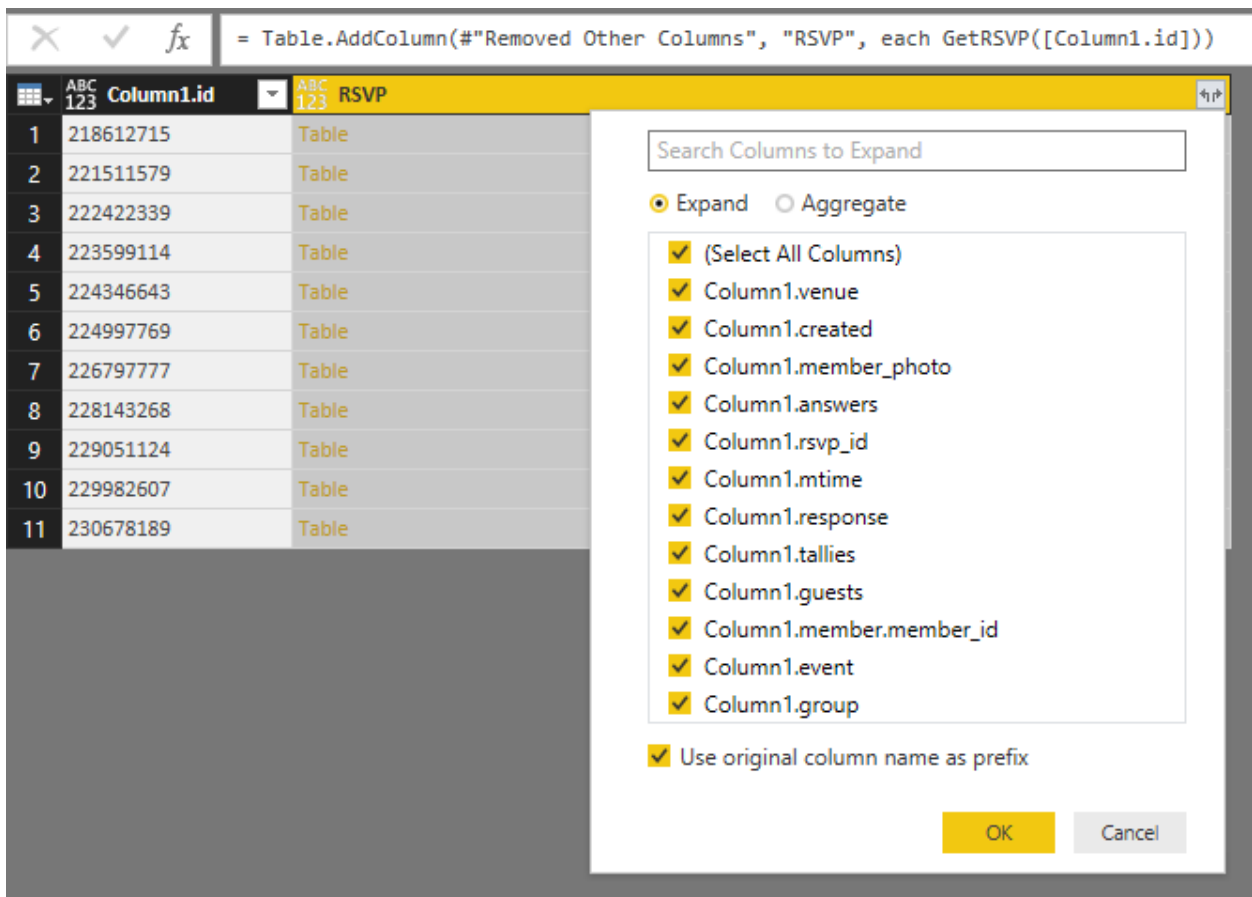


Table.AddColumn("#Removed Other Columns", "RSVP", each GetRSVP([Column1.id]))

Column1.id

1	218612715	Table
2	221511579	Table
3	222422339	Table
4	223599114	Table
5	224346643	Table
6	224997769	Table
7	226797777	Table
8	228143268	Table
9	229051124	Table
10	229982607	Table
11	230678189	Table

Expand Columns

Search Columns to Expand

☒ Expand ☐ Aggregate

- ☒ (Select All Columns)
- ☒ Column1.venue
- ☒ Column1.created
- ☒ Column1.member_photo
- ☒ Column1.answers
- ☒ Column1.rsvp_id
- ☒ Column1.mtime
- ☒ Column1.response
- ☒ Column1.tallies
- ☒ Column1.guests
- ☒ Column1.member.member_id
- ☒ Column1.event
- ☒ Column1.group

☒ Use original column name as prefix

OK Cancel

Note that you may hit the privacy configuration warning and errors at this step; because this is just for test. I went to Files -> Options and Settings -> Option, and change privacy

setting to ignore 😊 This is not best practice obviously, you have to set it up appropriate for production environment, but here is just a demo and test, so let's have fun.

Options

GLOBAL

Data Load
DirectQuery
R Scripting
Security
Privacy
Updates
Usage Data
Diagnostics

Privacy Levels

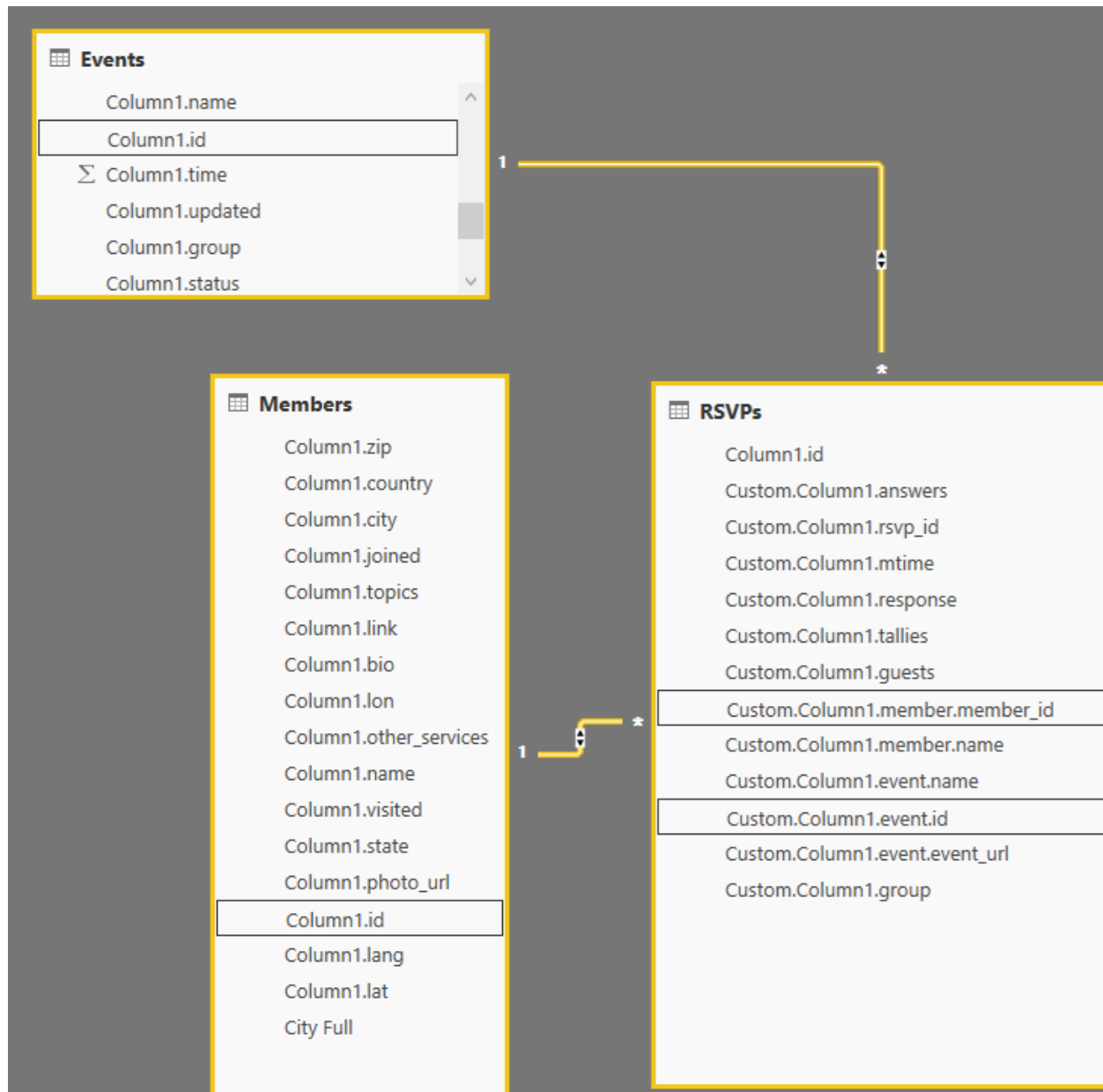
- ☐ Always combine data according to your Privacy Level settings for each source
- ☐ Combine data according to each file's Privacy Level settings
- ☒ Always ignore Privacy Level settings ⓘ

[Learn more about Privacy Levels](#)

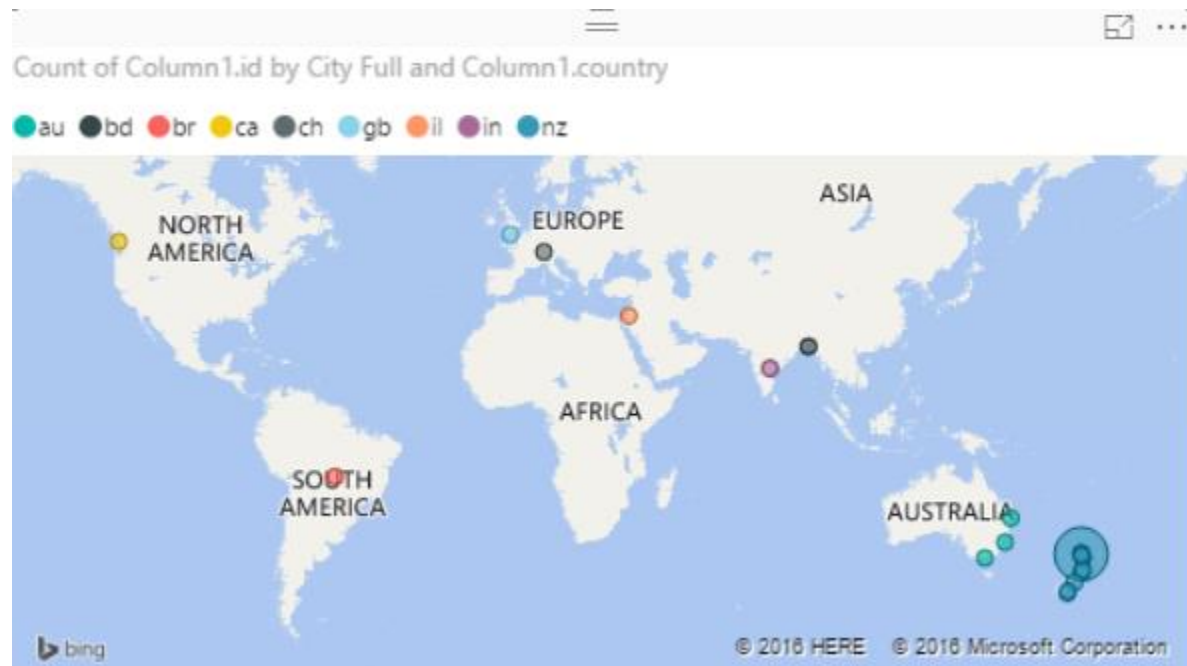
I rename this query as RSVPs, and Close and Load data into Power BI. I'll create relationship as below;

Events and RSVPs= column1.id from Events, Custom.Column1.event.id from RSVPs

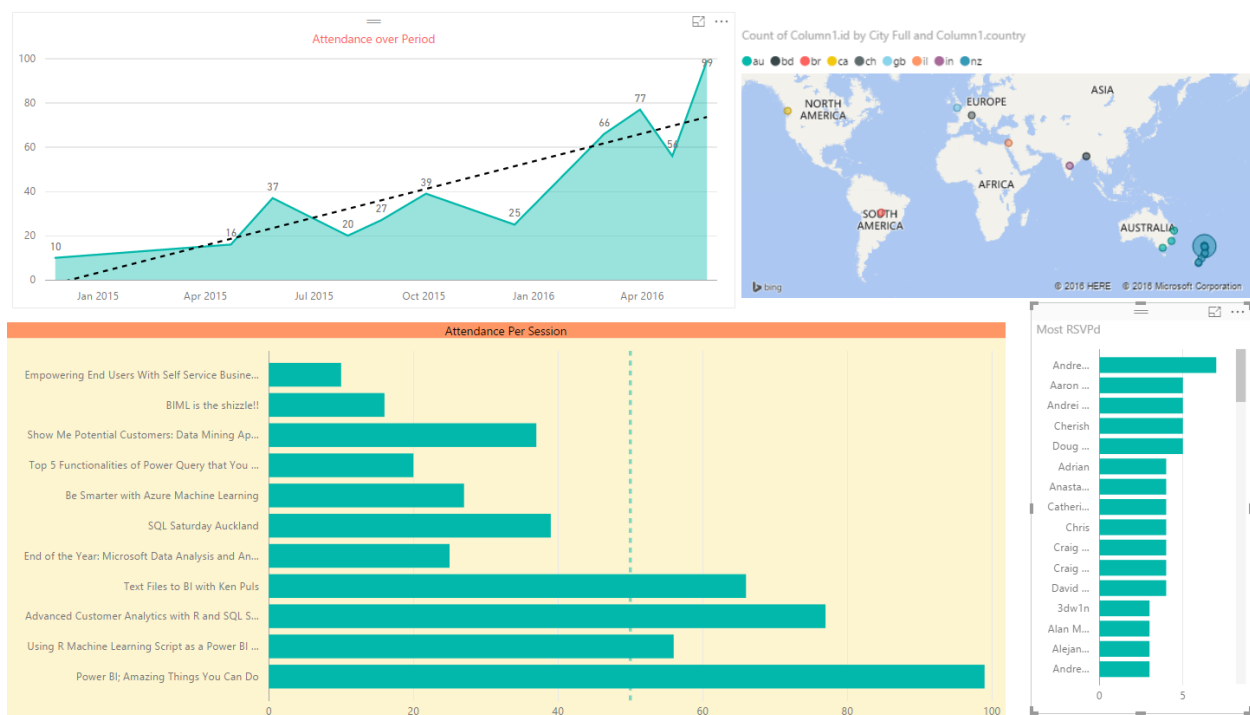
RSVPs and Members= Custom.Column1.member.member_id from RSVPs, Column1.id from Members



I can now visualize members by their geo location city information in a map



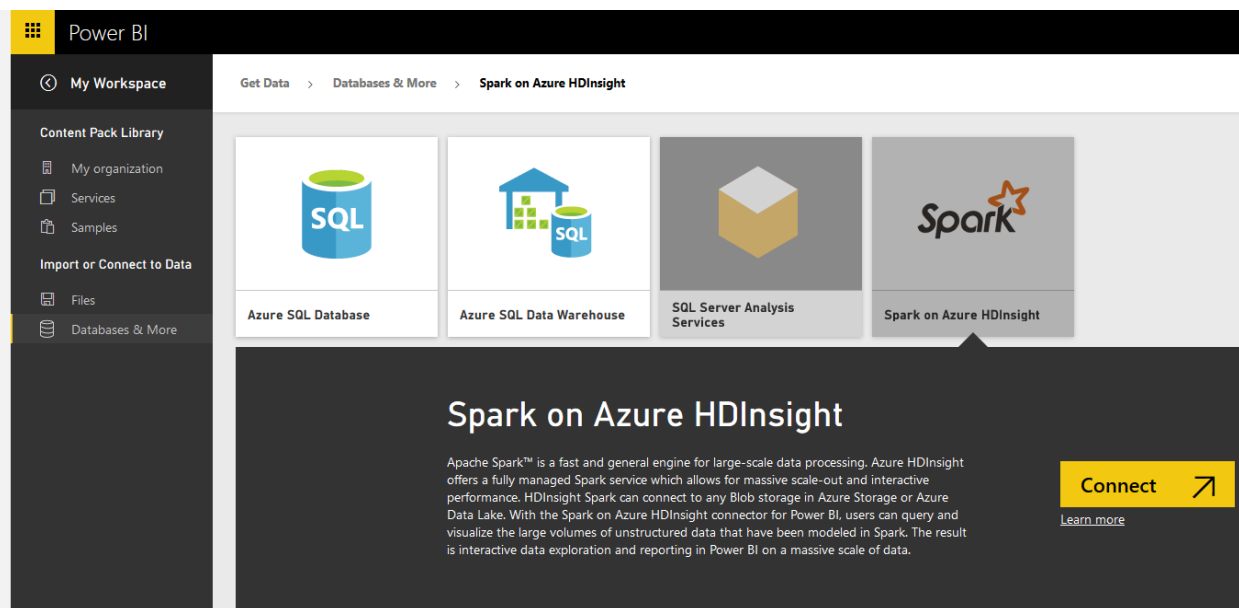
It is interesting that we have some members in our user group in other continents 😊 However the majority are in Auckland, New Zealand as expected. I can also build another bar chart for members who RSVPd yes mostly. Here is my final visualization with some formatting:



I can still dig into more details with adding more data into my Power BI model and visualization, but I leave it to you now from here to see how far you can go with it. Have fun with Power BI and Meetup API.

Power BI and Spark on Azure HDInsight; Step by Step Guide

Published Date: February 1, 2016



Power BI can connect to many data sources as you know, and Spark on Azure HDInsight is one of them. In area of working with Big Data applications you would probably hear names such as Hadoop, HDInsight, Spark, Storm, Data Lake and many other names. Spark and Hadoop are both frameworks to work with big data, they have some differences though. In this post I'll show you how you can use Power BI (either Power BI Desktop or Power BI website) to connect to a sample of Spark that we built on an Azure HDInsight service. by completing this section you will be able to create simple spark on Azure HDInsight, and run few Python scripts from Jupyter on it to load a sample table into Spark, and finally use Power BI to connect to Spark server, load, and visualize the data.

This post is a section of [Power BI from Rookie to Rock Star](#) online book.

Prerequisite

- For running sample mentioned in this section you need to have an [Azure subscription](#). (You can apply for a free trial as well)
- You need to install Microsoft Spark ODBC Driver on your machine (<https://go.microsoft.com/fwlink/?LinkId=616229>). This is for connecting from Power BI Desktop.
- For some features of Spark Connection through Power BI website you'd need Power BI Pro account (You can apply for a trial).

A Bit of Introduction

What is Spark? What is HDInsight?



I've spoken about Power BI well enough in [previous chapters](#). However you might be new to big data names such as Spark and HDInsight, and would like to know what these are? I say it in the most simple method; HDInsight is the framework for running big data, you can have your big data storage as a distributed storage for massive availability. Spark on the other hand is another big data framework that works great for analytics and big data operation, however it doesn't work with distributed files. So the real world big data solution has most of the time Spark on top of an HDFS (Hadoop Distributed File System) or [Azure HDInsight](#) to gain both scalability and powerful analytics. Spark is an Apache application, and often called as [Apache Spark](#). However Spark can be installed on both Windows and Linux servers. In this sample for simplicity we will use Spark on Linux server.

What is Jupyter?

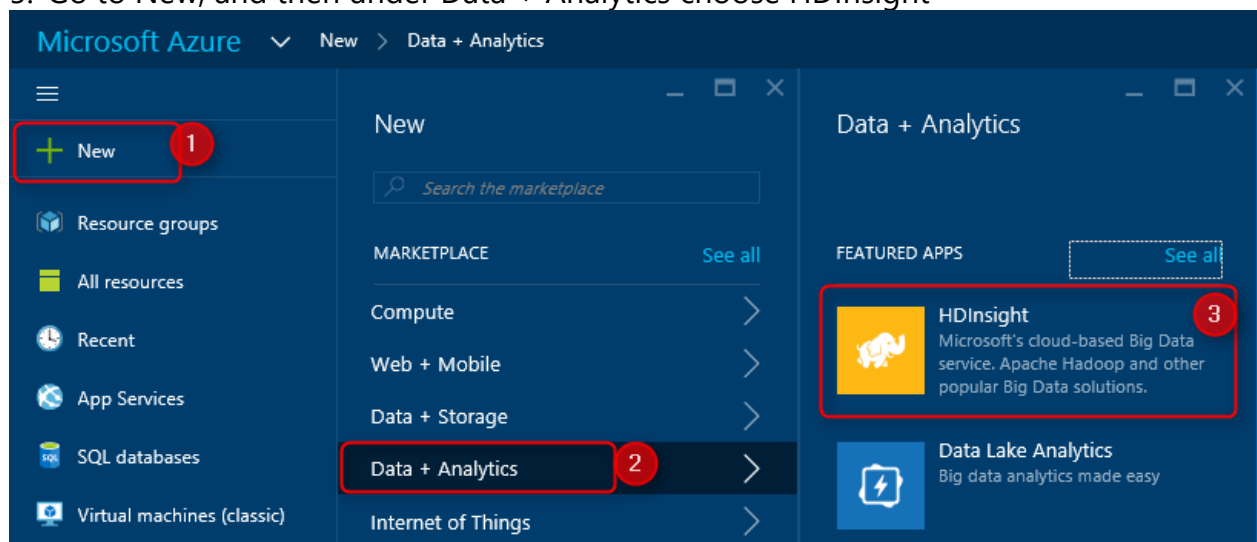


[Jupyter](#) is a live notepad engine for writing and running codes such as Python (it supports more than 40 languages!) on servers (such as Spark server).

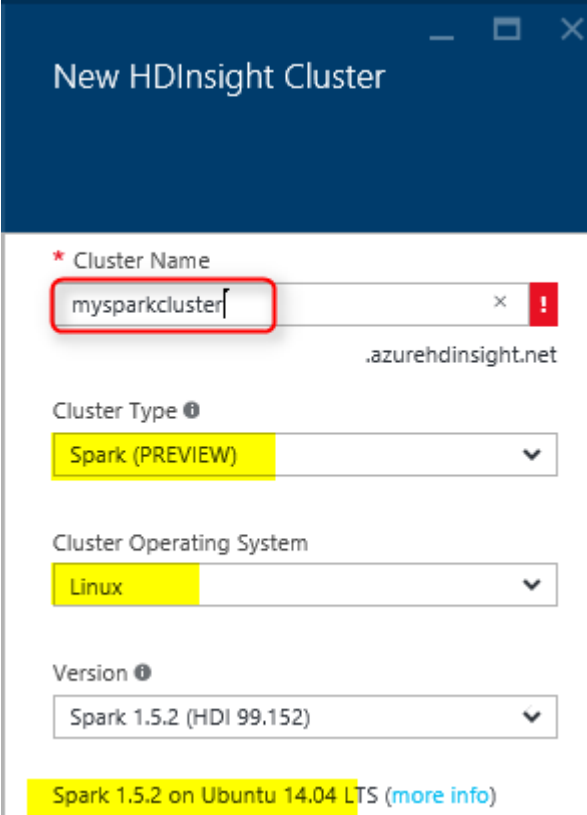
Setup and Configure Spark on Azure HDInsight

Let's start building the first Spark on Azure HDInsight. You can skip this part if you have an Spark on Azure HDInsight already.

4. Login to Azure Portal through <https://portal.azure.com>.
5. Go to New, and then under Data + Analytics choose HDInsight



6. In the New HDInsight Cluster definition section set a name for cluster (my sample is mysparkcluster), then set Cluster Type as Spark (it is still in preview stage), then Cluster Operating System as Linux. When you choose Linux it will install on Ubuntu 14.04 LTS.



New HDInsight Cluster

* Cluster Name
mysparkcluster .azurehdinsight.net

Cluster Type
Spark (PREVIEW)

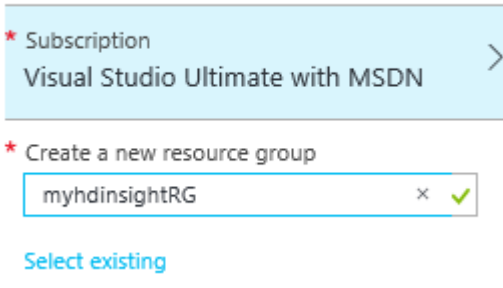
Cluster Operating System
Linux

Version
Spark 1.5.2 (HDI 99.152)

Spark 1.5.2 on Ubuntu 14.04 LTS ([more info](#))

7. Choose one of your subscriptions for Azure (If you have more than one), then assign a resource group. You can choose one of existing resource groups or create a new one.

Spark 1.5.2 on Ubuntu 14.04 LTS ([more info](#))

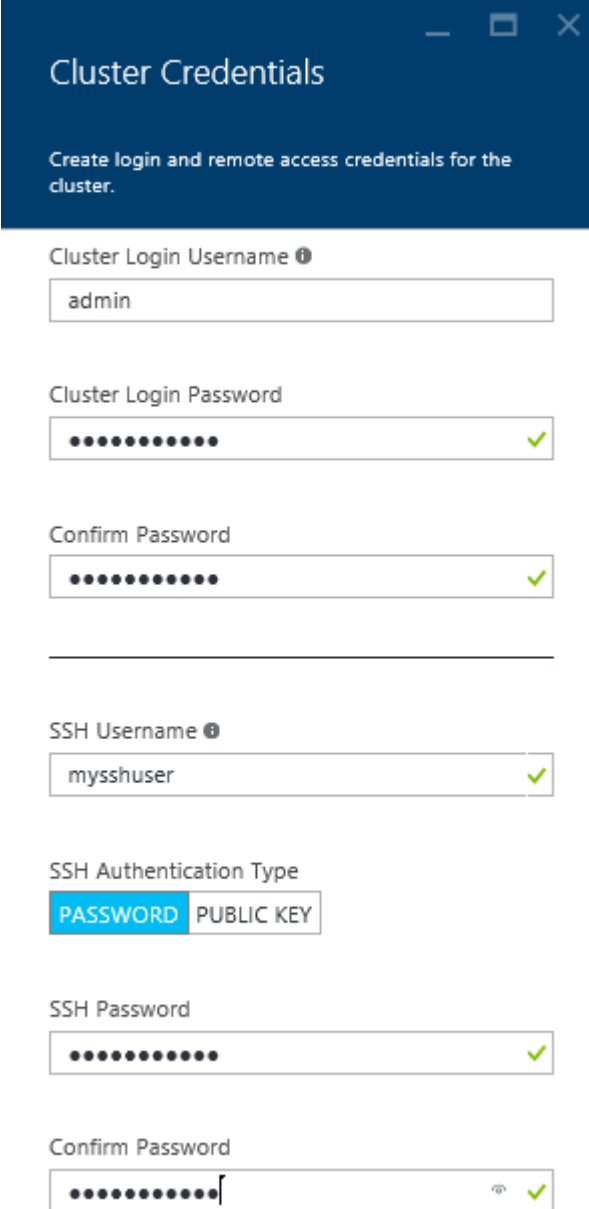


* Subscription
Visual Studio Ultimate with MSDN

* Create a new resource group
myhdinsightRG

[Select existing](#)

8. Set the Credentials for Spark cluster. set a password for admin user. Then write a SSH username and password. SSH username and password used to connect to spark cluster through command line. In this example we won't use command line. There is also a limitation on some reserved SSH user names that you cannot use.



Cluster Credentials

Create login and remote access credentials for the cluster.

Cluster Login Username ⓘ

admin

Cluster Login Password

●●●●●●●● ✓

Confirm Password

●●●●●●●● ✓

SSH Username ⓘ

mysshuser ✓

SSH Authentication Type

PASSWORD PUBLIC KEY

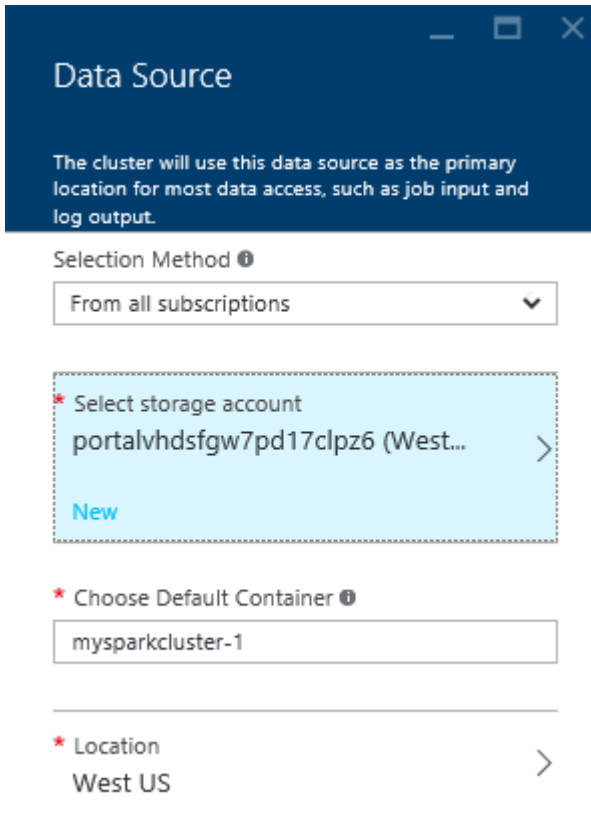
SSH Password

●●●●●●●● ✓

Confirm Password

●●●●●●●● ✓

9. Set Data Source. You can choose an existing storage account, or create a new one. and set the container and location.



Data Source

The cluster will use this data source as the primary location for most data access, such as job input and log output.

Selection Method ⓘ

From all subscriptions ▼

* Select storage account
portalvhdsfgw7pd17clpz6 (West... >

[New](#)

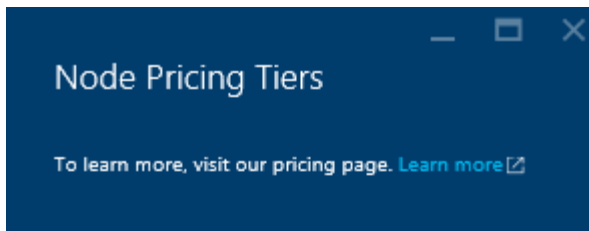
* Choose Default Container ⓘ

mysparkcluster-1

* Location >

West US

10. Choose Node Pricing Tiers. In this example because we are not processing large amount of data I'll do it with couple of nodes. You can see how much this cluster costs you per hour based on your selection of nodes and pricing tier for worker nodes and head node.



Number of Worker nodes ⓘ

2



* Worker Nodes Pricing Tier

D4 (2 nodes, 16 cores)



* Head Node Pricing Tier

D4 (2 nodes, 16 cores)



WORKER NODES 0.84 x 2 = 1.67

HEAD NODES 0.84 x 2 = 1.67

TOTAL COST **3.34**

NZD/HOUR (ESTIMATED)

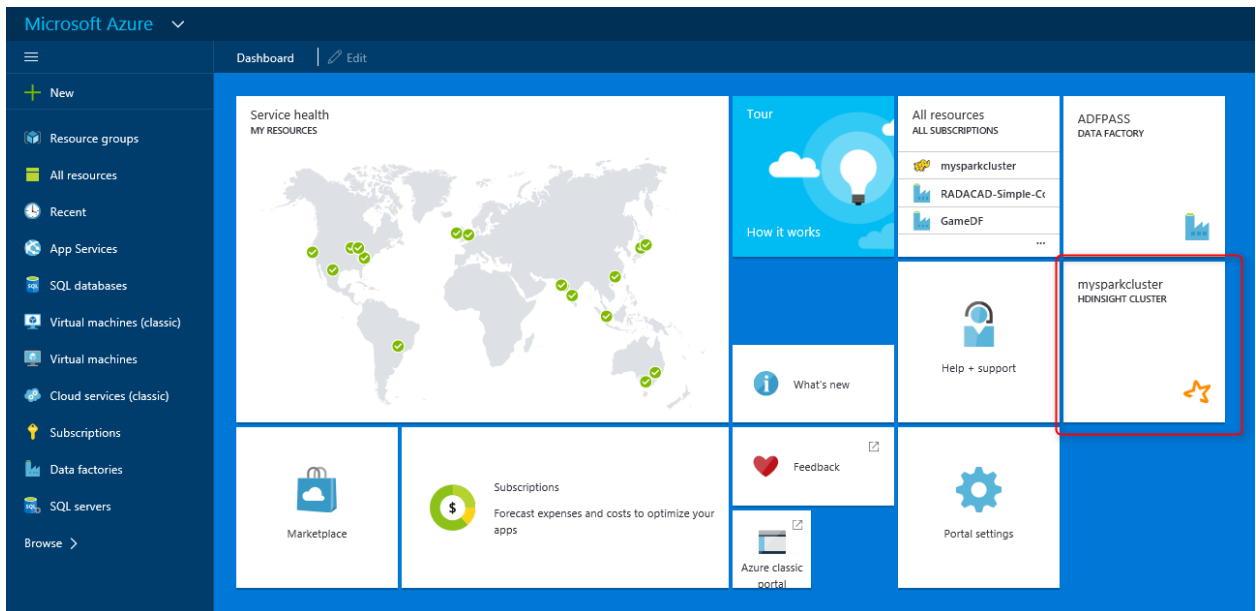
56 of 170 cores would be used in West US.

This price estimate does not include storage costs, network egress costs, or subscription discounts.

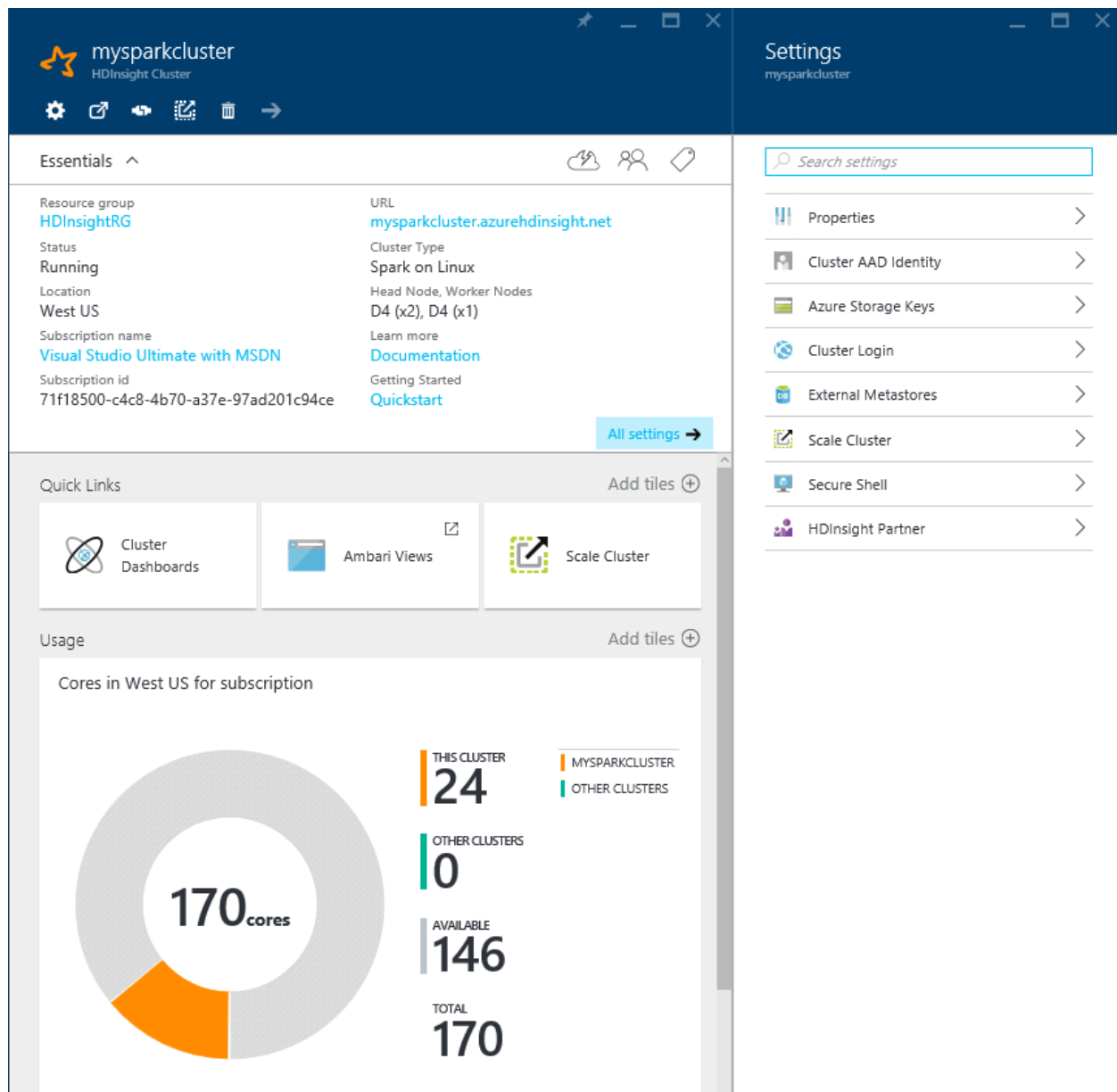
This is a preview feature. I acknowledge that use of this feature is subject to the preview terms in my license agreement (e.g., the Enterprise Agreement, Microsoft Azure Agreement, or Microsoft Online Subscription Agreement), as well as any applicable [Supplemental Terms of Use for Microsoft Azure Previews](#). Microsoft will provide me 30 days' notice via email prior to the expiration of the preview period.

Questions? [Contact billing support](#).

11. Now Create the HDInsight Cluster.
12. It will take about 15 minutes for the Cluster to be created and visible in your Microsoft Azure dashboard.



13. Now you have your Spark on Azure HDInsight ready for the next step.

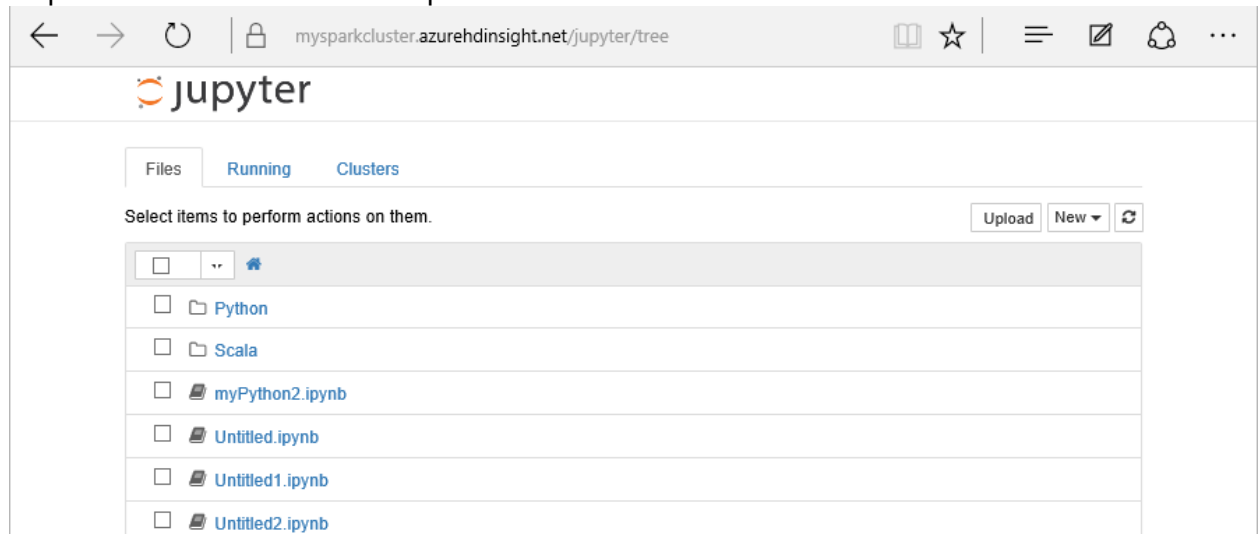


Setup the Sample Table for Spark on Azure HDInsight

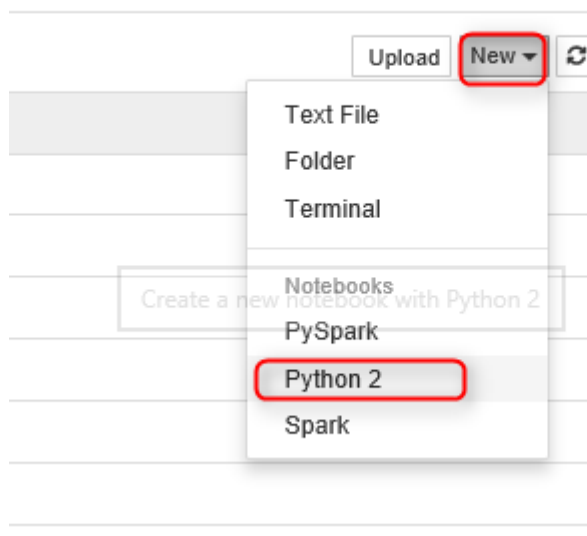
In this section we will load a sample data into a table (inside our Spark on Azure HDInsight cluster) using some codes that we run from Jupyter.

3. Open Jupyter for your Spark on Azure HDInsight with going to <https://<your spark cluster name>.azurehdinsight.net/jupyter>

4. You will be asked for username and password. Enter the admin username and password we've created in previous section.

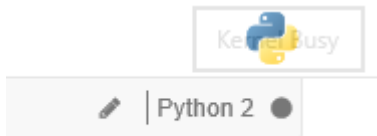


5. Click on New and choose a Python 2 type notebook.



6. You can name the document something meaningful and then copy and paste code below there into a cell
7. from pyspark import SparkContext
 from pyspark.sql import *
 from pyspark.sql import HiveContext
 from pyspark.sql import Row
- # Create Spark and Hive contexts
 sc = SparkContext('yarn-client')
 hiveCtx = HiveContext(sc)

8. Then press Shift+Enter (to run the code and go to the next cell)
9. You will see the Jupyter icon showing that the Kernel is busy after running commands.



Kernel Busy

10. Wait for the command to run, and Kernel back to idle state.



Kernel Idle

11. Now add script below in the second cell;
12. # Create an RDD from sample data
`hvacText = sc.textFile("wasb:///HdiSamples/HdiSamples/SensorSampleData/hvac/HVAC.csv")`

 # Create a schema for our data
`Entry = Row('Date', 'Time', 'TargetTemp', 'ActualTemp', 'BuildingID')`

 # Parse the data and create a schema
`hvacParts = hvacText.map(lambda s: s.split(',')).filter(lambda s: s[0] != 'Date')`
`hvac = hvacParts.map(lambda p: Entry(str(p[0]), str(p[1]), int(p[2]), int(p[3]), int(p[6])))`

 # Infer the schema and create a table
`hvacTable = hiveCtx.createDataFrame(hvac)`
`hvacTable.registerTempTable('hvactemptable')`
`dfw = DataFrameWriter(hvacTable)`
`dfw.saveAsTable('hvac')`

Commands above will populate a hive table called hvac from a CSV sample file located on the server.

13. After pressing Shift+Enter, and waiting for the kernel to go back to idle mode. you can query all hive tables with below command;

14. `hiveCtx.sql("SHOW TABLES").show()`

15. after Shift+Enter you will see the result below;

```
16. +-----+-----+
    |  tableName|isTemporary|
    +-----+-----+
    | hvactemptable|    true|
    |hivesampletable|  false|
    |      hvac|    false|
    +-----+-----+
```

17. You can even query hvac table to see the data;

18. `hiveCtx.sql("SELECT * FROM hvac LIMIT 10").show()`

19. and the result will be;

```
20. +-----+-----+-----+-----+-----+
    | Date|  Time|TargetTemp|ActualTemp|BuildingID|
    +-----+-----+-----+-----+-----+
    |6/10/13| 2:43:51|    68|    63|    3|
    |6/11/13| 3:43:51|    70|    55|   17|
    |6/12/13| 4:43:51|    69|    59|   17|
    |6/13/13| 5:43:51|    69|    68|   19|
    |6/14/13| 6:43:51|    67|    79|    2|
    |6/15/13| 7:43:51|    67|    71|    3|
    |6/16/13| 8:43:51|    69|    63|   16|
    |6/17/13| 9:43:51|    66|    56|    9|
    |6/18/13|10:43:51|    65|    61|   16|
    |6/19/13|11:43:51|    67|    74|    7|
    +-----+-----+-----+-----+-----+
```

Now we have table with the data ready for Power BI to work with, Let's go to the next step.

Jupyter Untitled6 Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Help Not Connected Python 2

Code CellToolbar

```
In [1]: from pyspark import SparkContext
from pyspark.sql import *
from pyspark.sql import HiveContext
from pyspark.sql import Row

# Create Spark and Hive contexts
sc = SparkContext('yarn-client')
hiveCtx = HiveContext(sc)

In [2]: # Create an RDD from sample data
hvacText = sc.textFile("wasb:///HdiSamples/HdiSamples/SensorSampleData/hvac/HVAC.csv")

# Create a schema for our data
Entry = Row('Date', 'Time', 'TargetTemp', 'ActualTemp', 'BuildingID')

# Parse the data and create a schema
hvacParts = hvacText.map(lambda s: s.split(',')).filter(lambda s: s[0] != 'Date')
hvac = hvacParts.map(lambda p: Entry(str(p[0]), str(p[1]), int(p[2]), int(p[3]), int(p[6])))

# Infer the schema and create a table
hvacTable = hiveCtx.createDataFrame(hvac)
hvacTable.registerTempTable('hvactemptable')
dfw = DataFramesWriter(hvacTable)
dfw.saveAsTable('hvac')

In [3]: hiveCtx.sql("SHOW TABLES").show()

+-----+-----+
|   tableName   |isTemporary|
+-----+-----+
| hvactemptable |         true|
| hivesampletable|        false|
|         hvac  |        false|
+-----+-----+

In [4]: hiveCtx.sql("SELECT * FROM hvac LIMIT 10").show()

+-----+-----+-----+-----+-----+
|   Date   |   Time   |TargetTemp|ActualTemp|BuildingID|
+-----+-----+-----+-----+-----+
|6/10/13| 2:43:51|        68|        63|         3|
|6/11/13| 3:43:51|        70|        55|        17|
|6/12/13| 4:43:51|        69|        59|        17|
|6/13/13| 5:43:51|        69|        68|        19|
|6/14/13| 6:43:51|        67|        79|         2|
|6/15/13| 7:43:51|        67|        71|         3|
|6/16/13| 8:43:51|        69|        63|        16|
|6/17/13| 9:43:51|        66|        56|         9|
|6/18/13|10:43:51|        65|        61|        16|
|6/19/13|11:43:51|        67|        74|         7|
+-----+-----+-----+-----+-----+

In [ ]:
```

Connection from Power BI Desktop to Spark on Azure HDInsight

As I said in the very first part this tutorial, you need to install Microsoft Spark ODBC driver to be able to connect from Power BI Desktop to Spark on Azure HDInsight.

- Open Power BI Desktop and go to Get Data Section.
- Choose Azure, and then Azure HDInsight Spark.

Get Data

All


File


Database


Azure


Other


Azure


 Microsoft Azure SQL Database


 Microsoft Azure SQL Data Warehouse


 Microsoft Azure Marketplace


 Microsoft Azure HDInsight

 Microsoft Azure Blob Storage

 Microsoft Azure Table Storage

 Azure HDInsight Spark (Beta)

 Microsoft Azure DocumentDB (Beta)

 Microsoft Azure Data Lake Store (Beta)

Connect

Cancel

- Enter the spark cluster URL as `https://<your spark cluster name>/azurehdinsight.net`

Microsoft Azure HDInsight Spark

Enter the URL of the Azure HDInsight Spark cluster.

URL (e.g. <https://contoso.azurehdinsight.net/>)

<https://mysparkcluster.azurehdinsight.net>

OK

Cancel

- If you don't have Microsoft Spark ODBC Driver installed you will see an error as below;

Unable to Connect

We encountered an error while trying to connect.

Details: "Microsoft Spark ODBC Driver is not installed. To download the driver visit <https://go.microsoft.com/fwlink/?LinkId=616229>. "

Retry

Edit

Cancel

- If you have driver installed you will be asked for the username and password (the admin user that we've created for Spark on Azure HDInsight before)
- After successful connection you will see list of tables. Choose hvac simply and then Load.

Navigator

Search

Show All | Show Selected [1]

https://mysparkcluster.azurehdinsig...

hivesampletable

hvac

hvac

Date	Time	TargetTemp	ActualTemp	BuildingID
6/10/13	2:43:51	68	63	3
6/11/13	3:43:51	70	55	17
6/12/13	4:43:51	69	59	17
6/13/13	5:43:51	69	68	19
6/14/13	6:43:51	67	79	2
6/15/13	7:43:51	67	71	3
6/16/13	8:43:51	69	63	16
6/17/13	9:43:51	66	56	9
6/18/13	10:43:51	65	61	16
6/19/13	11:43:51	67	74	7
6/20/13	12:43:51	65	68	17
6/21/13	13:43:51	67	66	10
6/22/13	14:43:51	67	55	1
6/23/13	15:43:51	70	61	7
6/24/13	16:43:51	65	62	1
6/25/13	17:43:51	67	73	19
6/26/13	18:43:51	66	75	20
6/27/13	19:43:51	70	72	17
6/28/13	20:43:51	66	60	6
6/29/13	21:43:51	67	63	5
6/30/13	22:43:51	65	76	7
6/1/13	23:43:51	69	66	6
6/2/13	0:13:20	68	62	5
6/3/13	1:13:20	69	68	3

Load

Edit

Cancel

- That was really simple from Power BI point of view, wasn't it?!
- Now you can visualize it as you want! for example I've created an Area Chart with Time as axis, and average of ActualTemp and average of TargetTemp as values. I've changed color of data values as below.

Legend On

Data colors

Average of Actua...

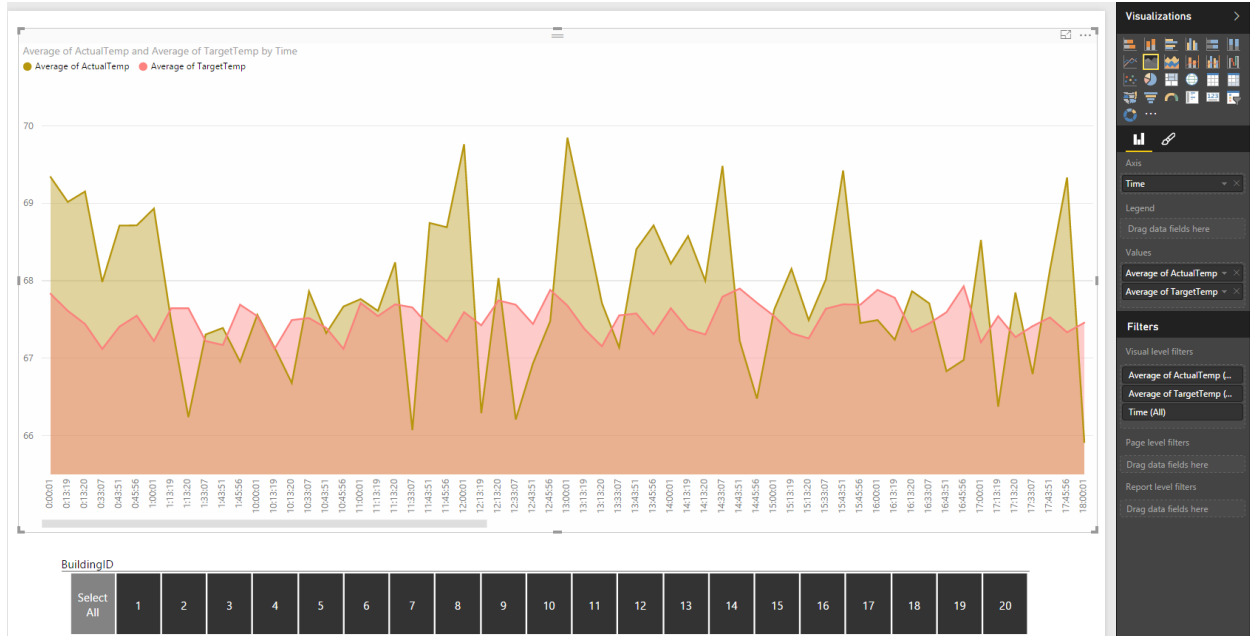
Average of Targe...

Revert to default

- I've also added a slicer (horizontal) for the BuildingID.



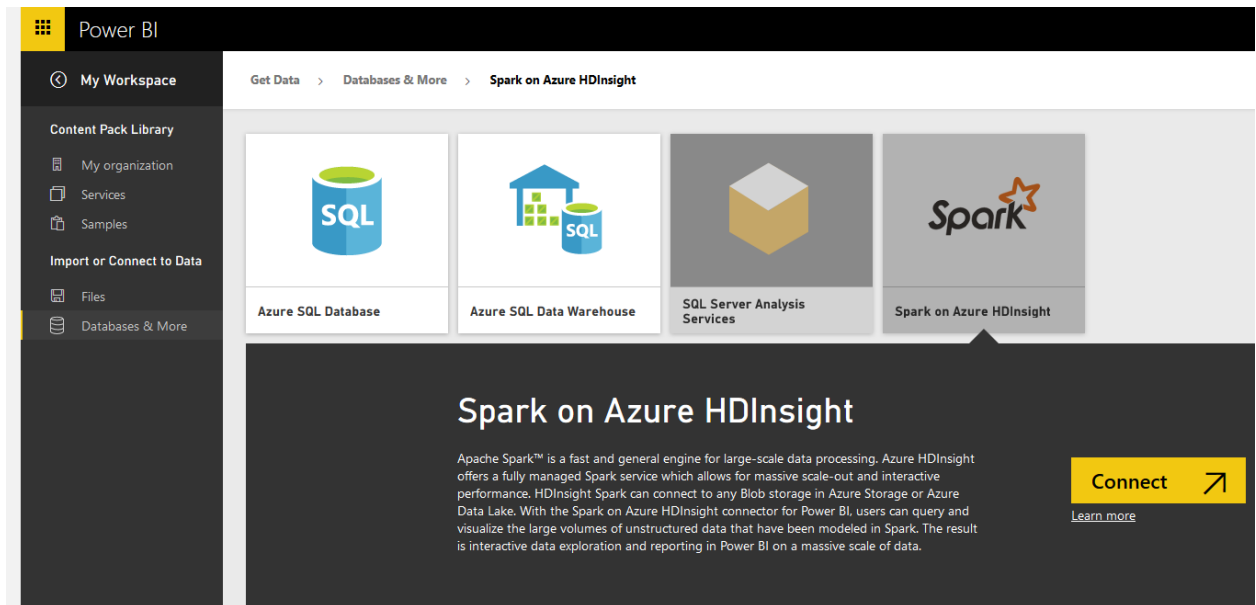
- And here is the result.



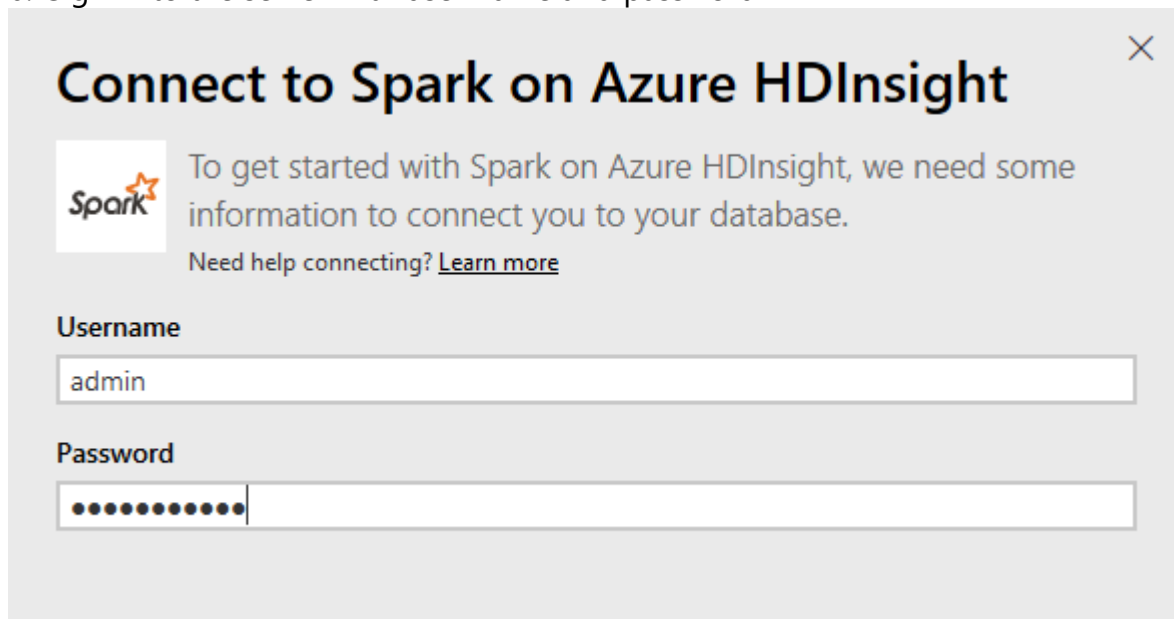
You can simply publish your report to Power BI website. Let's now see how you can connect to Spark from Power BI website itself.

Connection from Power BI website to Spark on Azure HDInsight

4. Login to Power BI website and Get Data from Databases, and Choose Spark on Azure HDInsight as the source.



5. Type in the server name as `https://<your spark cluster name>.azurehdinsight.net` and then connect
6. Sign in to the server with user name and password

The screenshot shows a dialog box titled 'Connect to Spark on Azure HDInsight'. It features the Spark logo and a message: 'To get started with Spark on Azure HDInsight, we need some information to connect you to your database. Need help connecting? [Learn more](#)'. Below the message are two input fields: 'Username' with the text 'admin' and 'Password' with masked characters (dots). A close button (X) is in the top right corner.

7. Now you can build the report you want from the hvac table.

References;

<https://powerbi.microsoft.com/en-us/documentation/powerbi-spark-on-hdinsight-with-direct-connect/>

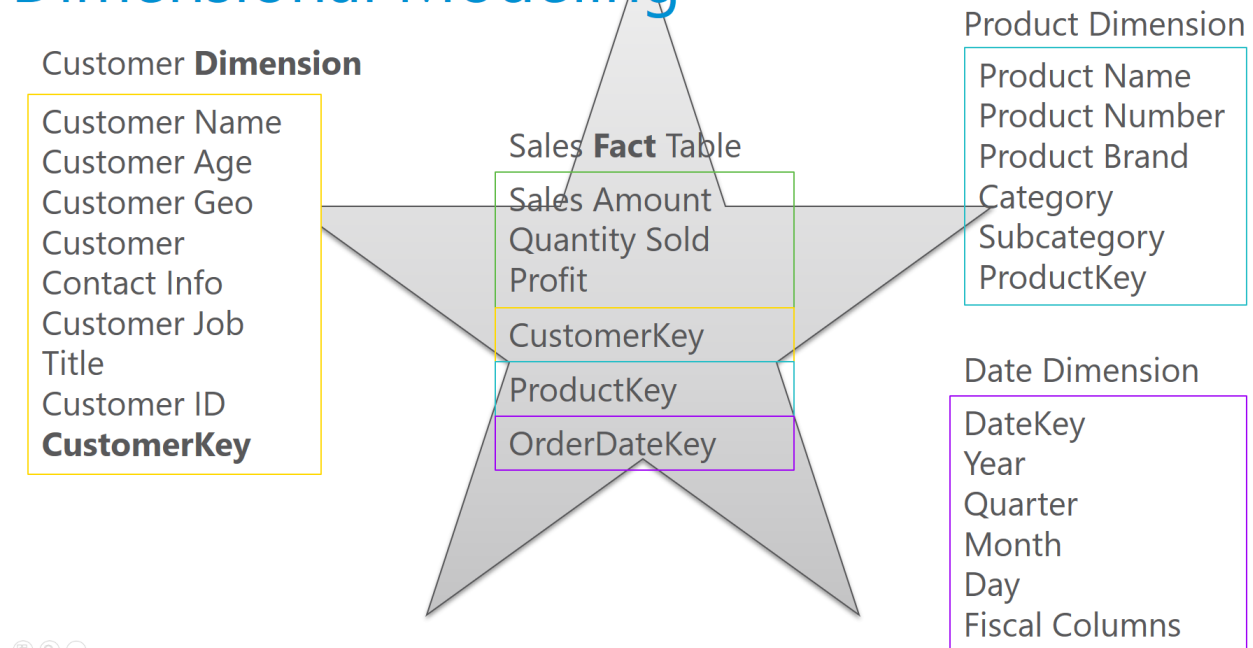
<https://azure.microsoft.com/en-us/documentation/articles/hdinsight-apache-spark-use-bi-tools/>

<https://azure.microsoft.com/en-us/documentation/articles/hdinsight-apache-spark-jupyter-spark-sql/>

Data Preparation; First and Foremost Important Task in Power BI

Published Date: December 21, 2016

Dimensional Modeling

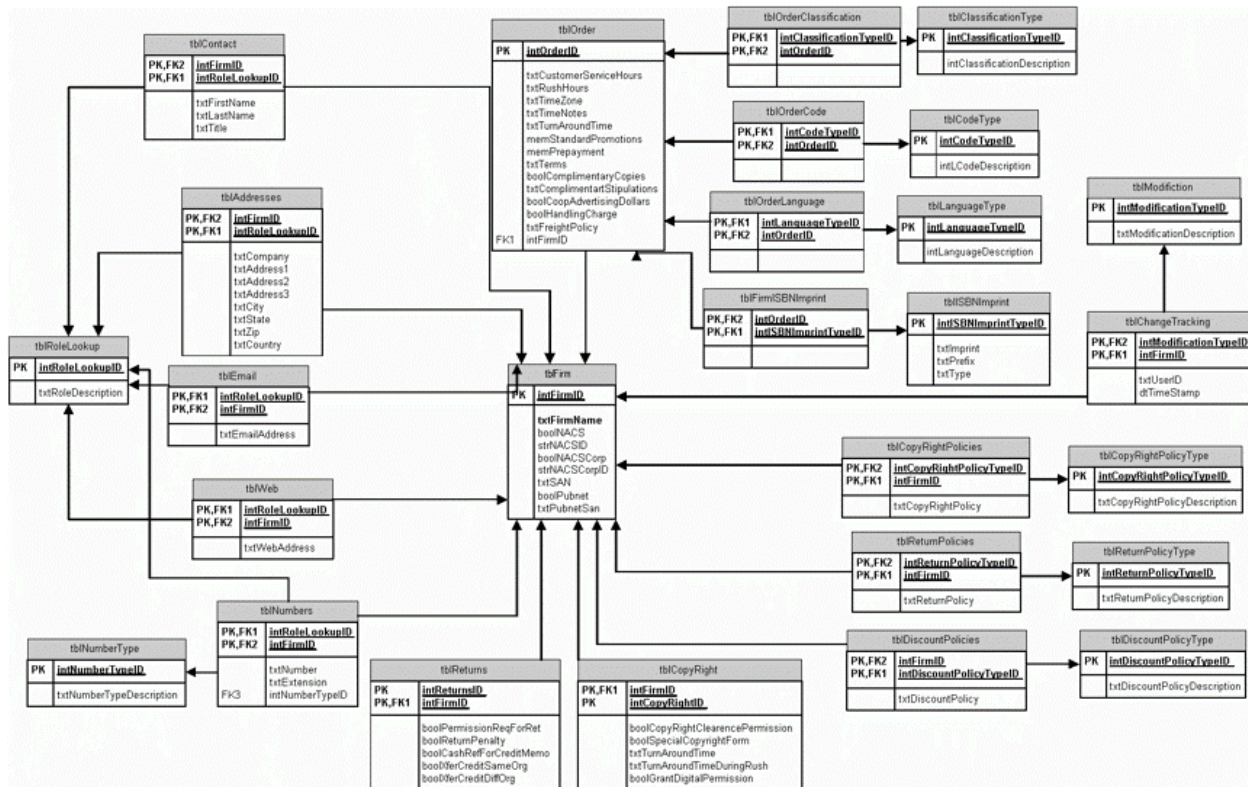


I've talked about Data Preparation many times in conferences such as PASS Summit, BA Conference, and many other conferences. Every time I talk about this I realize how much more I need to explain this. Data Preparation tips are basic, but very important. In my opinion as someone who worked with BI systems more than 15 years, this is the most important task in building in BI system. In this post I'll explain why data preparation is necessary and what are five basic steps you need to be aware of when building a data model with Power BI (or any other BI tools). This post is totally conceptual, and you can apply these rules on any tools.

Why Data Preparation?

All of us have seen many data models like screenshot below. Transactional databases has the nature of many tables and relationship between tables. Transactional databases are build for CRUD operations (Create, Retrieve, Update, Delete rows). Because of this single purpose, transactional databases are build in Normalized way, to reduce

redundancy and increase consistency of the data. For example there should be one table for Product Category with a Key related to a Product Table, because then whenever a product category name changes there is only one record to update and all products related to that will be updated automatically because they are just using the key. There are books to read if you are interested in how database normalization works.

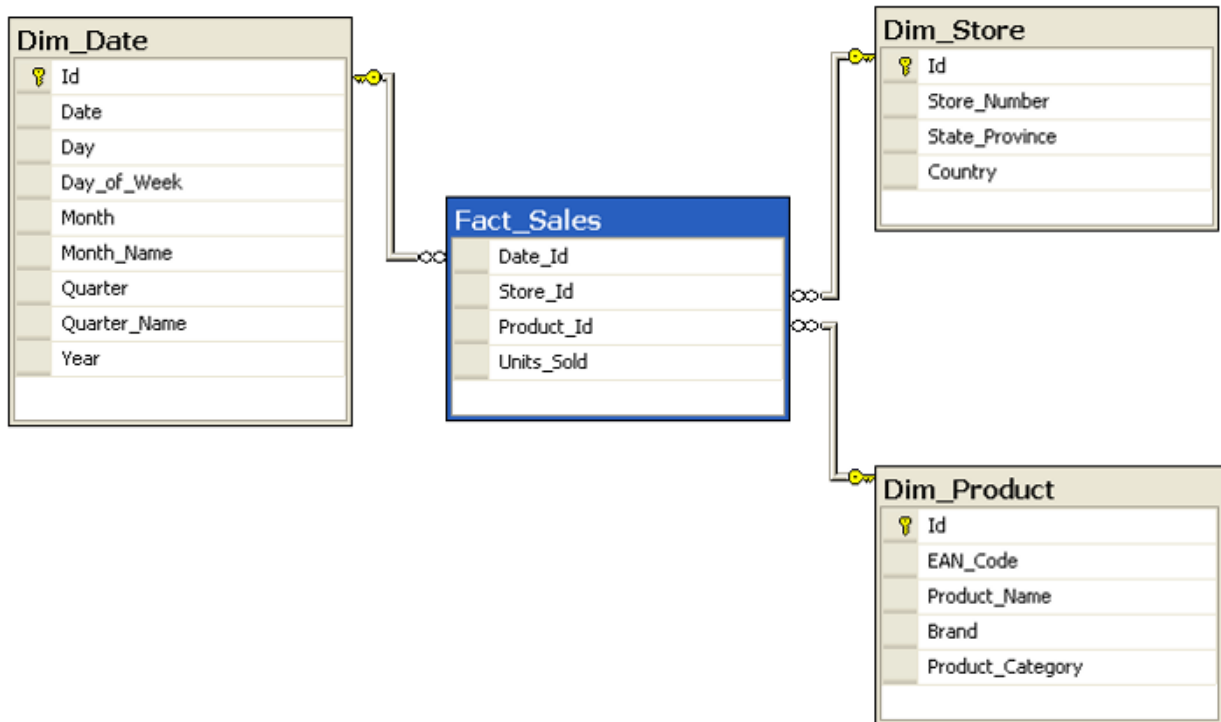


I'm not going to talk about how to build these databases. In fact for building a data model for BI system you need to avoid this type of modeling! This model works perfectly for transactional database (when there are systems and operators do data entry and modifications). However this model is not good for a BI system. There are several reasons for that, here are two most important reasons;

1. The model is hard to understand for a Report User
2. Too many tables and many relationship between tables makes a reporting query (that might use 20 of these tables at once) very slow and not efficient.

You never want to wait for hours for a report to respond. Response time of reports should be fast. You also never wants your report users (of self-service report users) to understand schema above. It is sometimes even hard for a database developer to understand how this works in first few hours! You need to make your model simpler,

with few tables and relationships. Your first and the most important job as a BI developer should be transforming above schema to something like below;



This model is far simpler and faster. There is one table that keeps sales information (Fact_Sales), and few other tables which keep descriptions (such as Product description, name, brand, and category). There is one one relationship that connects the table in the middle (Fact) to tables around (Dimensions). This is the best model to work with in a BI system. This model is called **Star Schema**. Building a star schema or dimensional modeling is your most important task as a BI developer.

How to Design a Star Schema?

To build a star schema for your data model I strongly suggest you to take one step at a time. What I mean by that is choosing one or few use cases and start building the model for that. For example; instead of building a data model for the whole Dynamics AX or CRM which might have more than thousands of tables, just choose Sales side of it, or Purchasing, or GL. After choosing one subject area (for example Sales), then start building the model for it considering what is required for the analysis.

Fact Table

Fact tables are tables that are holding numeric and additive data normally. For example quantity sold, or sales amount, or discount, or cost, or things like that. These values are numeric and can be aggregated. Here is an example of fact table for sales;

Sales **Fact** Table

Sales Amount
Quantity Sold
Profit

Dimension Table

Any descriptive information will be kept in Dimension tables. For example; customer name, customer age, customer geo information, customer contact information, customer job, customer id, and any other customer related information will be kept in a table named Customer Dimension.

Customer **Dimension**

Customer Name
Customer Age
Customer Geo
Customer
Contact Info
Customer Job
Title
Customer ID
CustomerKey

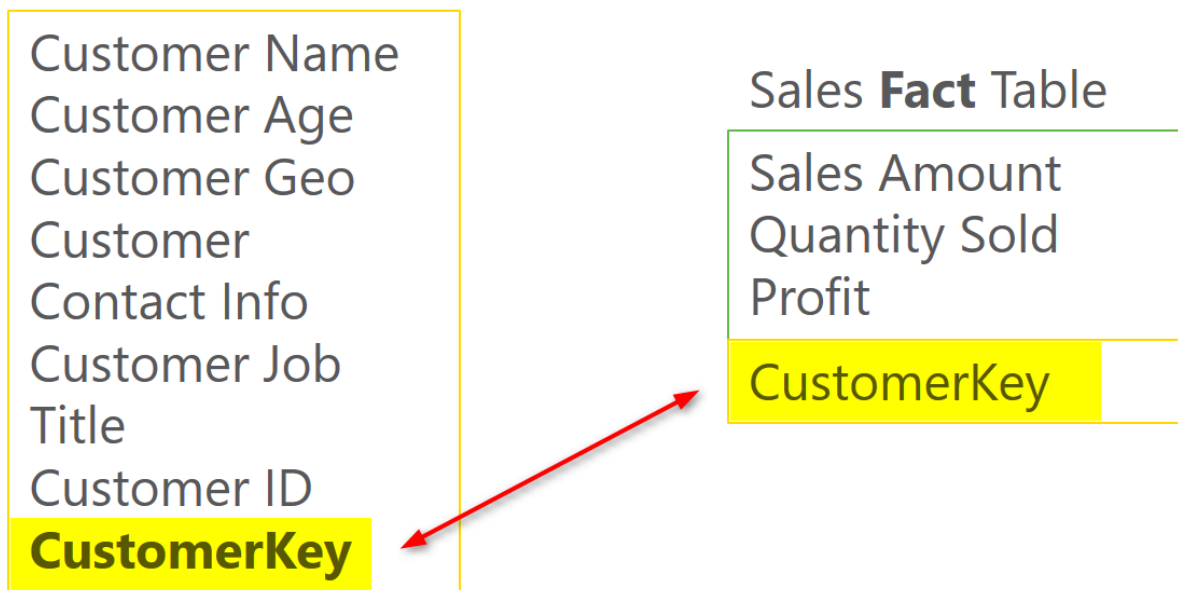
each dimension table should contain a key column. this column should be numeric (integer or big integer depends on the size of dimension) which is auto increment (Identity in SQL Server terminology). This column should be unique per each row in the dimension table. This column will be primary key of this table and will be used in fact table as a relationship. This column SHOULDN'T be ID of the source system. There are several reasons for why. This column is called **Surrogate Key**. Here are few reasons why you need to have surrogate key:

2. Codes (or IDs) in source system might be Text not Integer.

3. Short Int, Int, or Big Int are best data types for surrogate key because these are values which will be used in the fact table. Because fact table is the largest table in the data set, it is important to keep it in the smallest size possible (using Int data types for dimension foreign keys is one of the main ways of doing that).
4. Codes (or IDs) might be recycled.
5. You might want to keep track of changes (Slowly Changing Dimension), so one ID or Code might be used for multiple rows.
- 6....

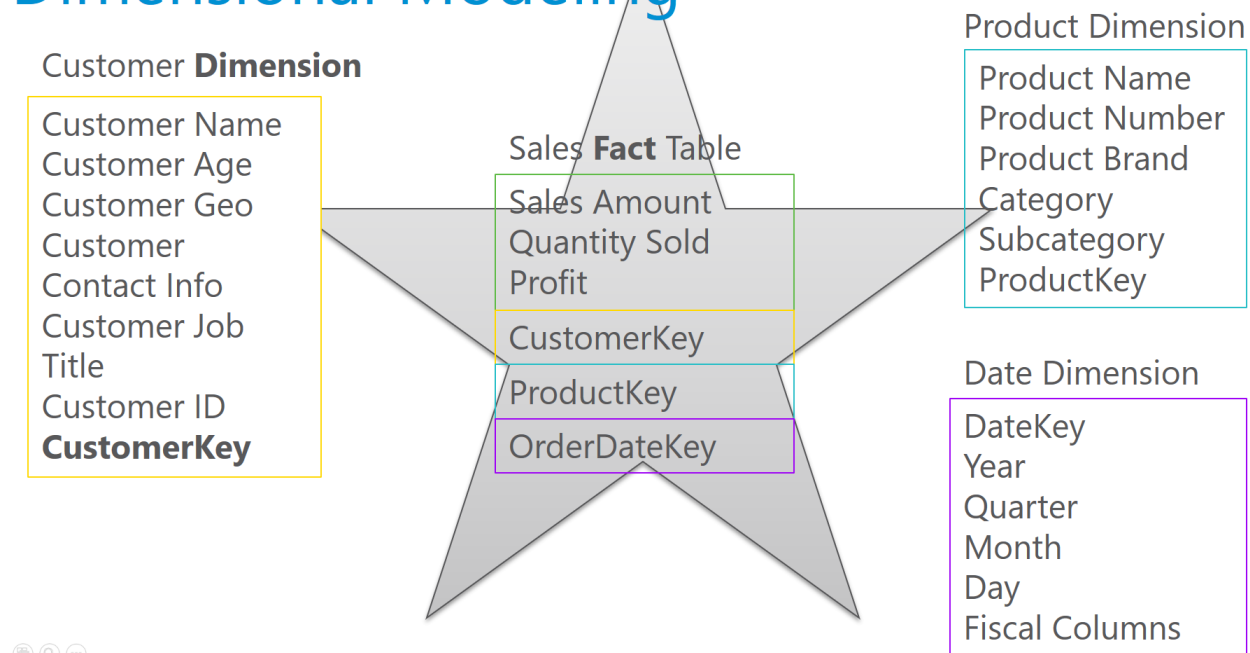
Surrogate key of the dimension should be used in the fact table as foreign key. Here is an example;

Customer **Dimension**



Other dimensions should be also added in the same way. in example below a Date Dimension and Product Dimension is also created. You can easily see in the screenshot below that why it is called star schema; Fact table is in the middle and all other dimensions are around with one relationship from fact table to other dimensions.

Dimensional Modeling



Design Tips

Building star schema or dimensional modeling is something that you have to read books about it to get it right, I will write some more blog posts and explain these principles more in details. However, it would be great to leave some tips here for you to get things started towards better modeling. These tips are simple, but easy to overlook. Number of BI solutions that I have seen suffer from not obeying these rules are countless. These are rules that if you do not follow, you will be soon far away from proper data modeling, and you have to spend ten times more to build your model proper from the beginning. Here are tips:

Tip 1: **DO NOT** add tables/files as is

Why?

Tables can be joined together to create more flattened and simpler structure.

Solution: DO create flattened structure for tables (specially dimensions)

Tip 2: DO NOT flatten your FACT table

Why?

Fact table are the largest entities in your model. Flattening them will make them even larger!

Solution: DO create relationships to dimension tables

Tip 3: DO NOT leave naming as is

Why?

names such as std_id, or dimStudent are confusing for users.

Solution: DO set naming of your tables and columns for the end user

Tip 4: DO NOT leave data types as is

Why?

Some data types are spending memory (and CPU) like decimals. Appropriate data types are also helpful for engines such as Q&A in Power BI which are working based on the data model.

Solution: DO set proper data types based on the data in each field

Tip 5: DO NOT load the whole data if you don't require it

Why?

Filtering part of the data before loading it into memory is cost and performance effective.

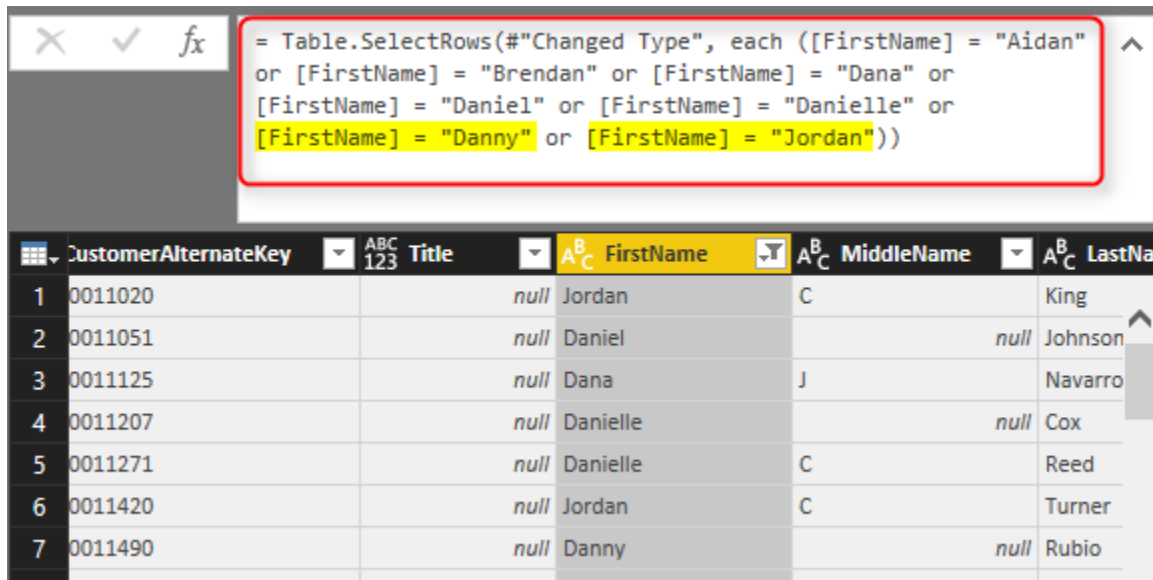
Solution: DO Filter part of the data that is not required.

More to Come

This was just a very quick introduction to data preparation with some tips. This is beginning of blog post series I will write in the future about principles of dimensional modeling and how to use them in Power BI or any other BI tools. You can definitely build a BI solution without using these concepts and principles, but your BI system will slow down and users will suffer from using it after few months, I'll give you my word on it. Stay tuned for future posts on this subject.

Warning! Misleading Power Query Filtering

Published Date: December 15, 2016



The screenshot shows the Power Query Editor. The formula bar contains the following M code, with the last two conditions highlighted in yellow:

```
= Table.SelectRows(#"Changed Type", each ([FirstName] = "Aidan"
or [FirstName] = "Brendan" or [FirstName] = "Dana" or
[FirstName] = "Daniel" or [FirstName] = "Danielle" or
[FirstName] = "Danny" or [FirstName] = "Jordan"))
```

Below the formula bar is a table with the following data:

	CustomerAlternateKey	ABC 123 Title	ABC C FirstName	ABC C MiddleName	ABC C LastName
1	0011020	null	Jordan	C	King
2	0011051	null	Daniel		Johnson
3	0011125	null	Dana	J	Navarro
4	0011207	null	Danielle		Cox
5	0011271	null	Danielle	C	Reed
6	0011420	null	Jordan	C	Turner
7	0011490	null	Danny		Rubio

Filtering in Power Query component of Power BI is easy, however it can be misleading very easily as well. I have seen the usage of this filtering inappropriately in many cases. Many people simply believe in what they see, rather than seeing behind the scenes. Power Query filtering is totally different when you do basic filtering or advanced filtering, and the result of filtering will be different too. In this post I'll show you through a very simple example how misleading it can be and what it the correct way to do filtering in Power Query. This post is a must read for everyone who use filtering in Power Query. If you like to learn more about Power BI, read the [Power BI online book from Rookie to Rock Star](#).

Prerequisite

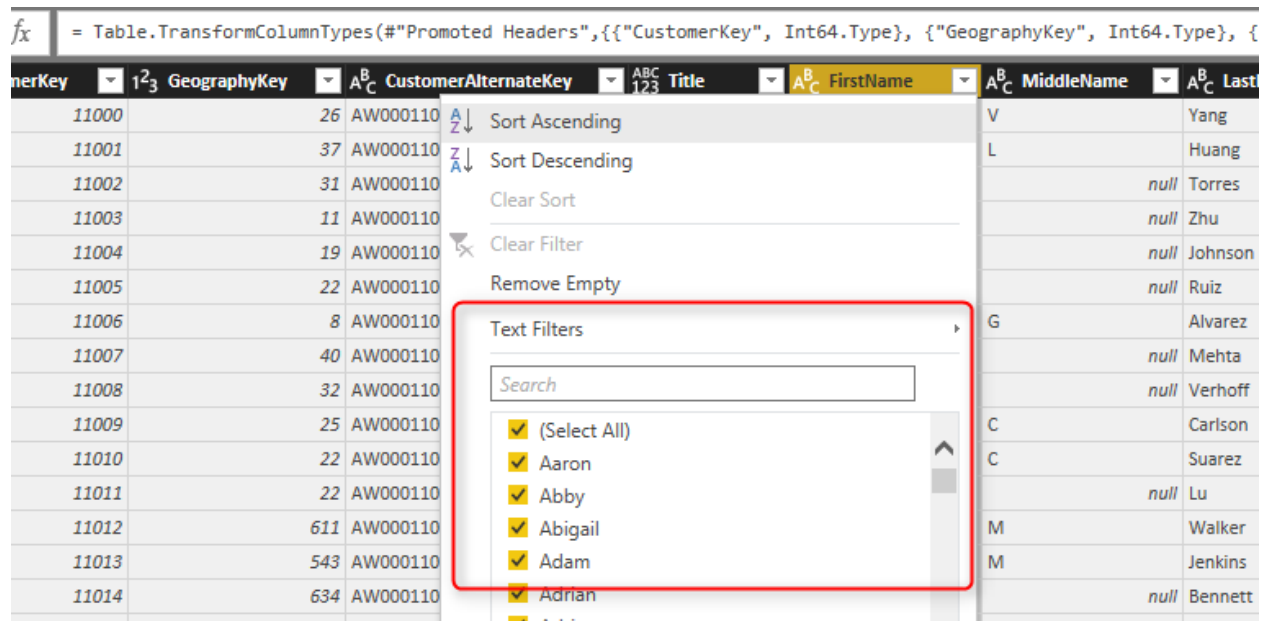
For this example I'll use AdventureWorksDW SQL Server database example. only one table which is DimCustomer.

Filtering in Power Query

Filtering rows is one of the most basic requirements in a data preparation tool such as Power Query, and Power Query do it strongly with an Excel like behavior for filtering. To

see what it looks like open Power BI Desktop (or Excel), and Get Data from the database and table above; DimCustomer, Click on Edit to open Query Editor window.

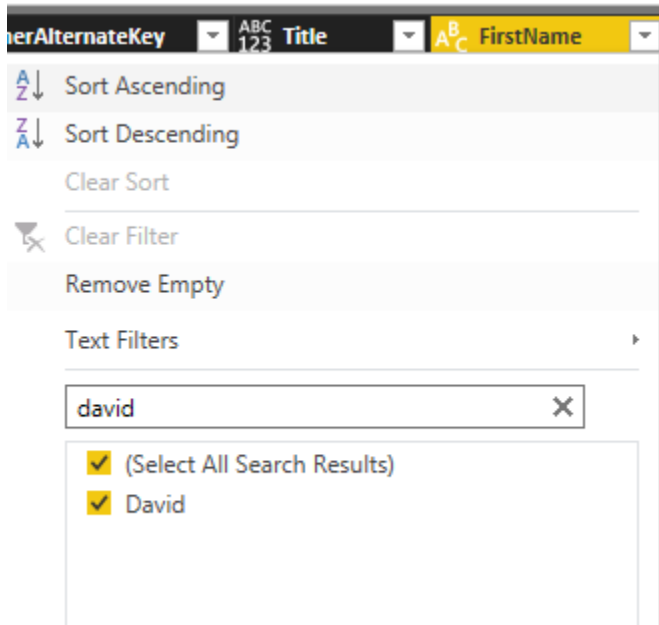
In the Query Editor window there is a button on top right hand side of every column that gives you access to filtering rows



You can simply filter values in the search box provided.

Basic Filtering

For basic filtering all you need to do is typing in the text, number, or date you want in the search box and find values to select from the list. For example if you want to filter all data rows to be only for the FirstName of David, you can simply type it and select and click on OK.



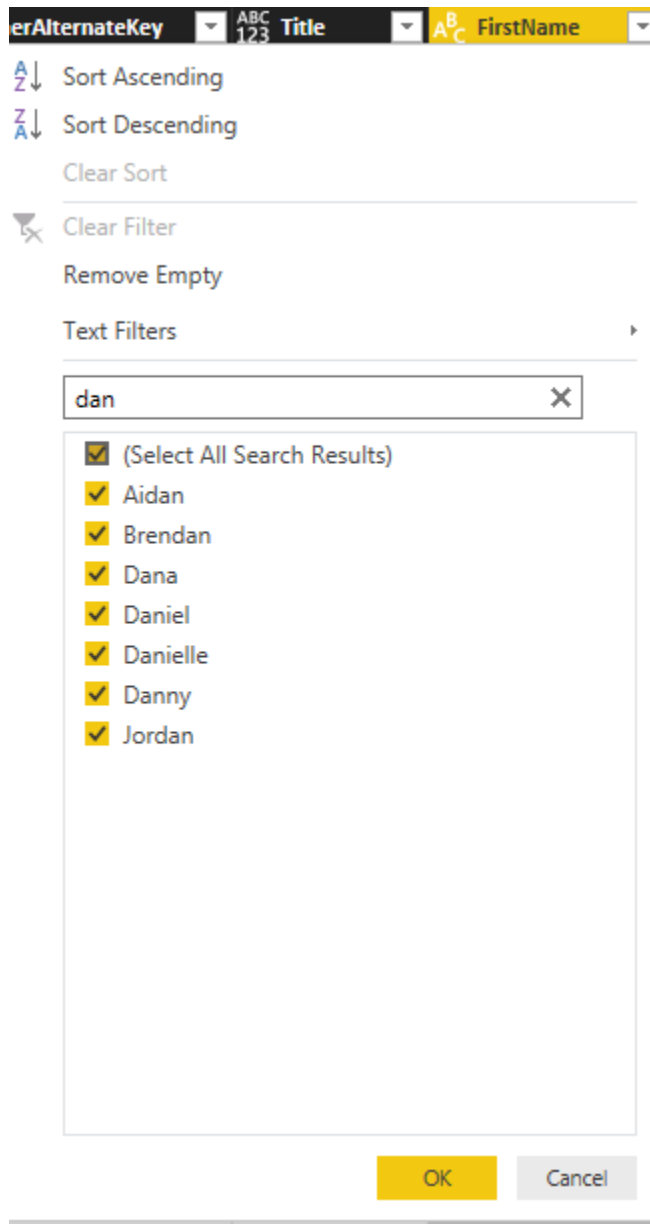
Result then will be only records with David as first name.

	123 CustomerKey	123 GeographyKey	A ^B _C CustomerAlternateKey	ABC 123 Title	A ^B _C FirstName	A ^B _C MiddleName	A ^B _C LastName
1	11168	348	AW00011168	null	David	null	Rodriguez
2	11267	302	AW00011267	null	David	null	Diaz
3	11377	161	AW00011377	Mr.	David	R.	Robinett
4	11701	347	AW00011701	null	David	null	Hayes
5	12017	60	AW00012017	null	David	E	Simmons
6	13396	631	AW00013396	null	David	null	Kumar
7	14374	343	AW00014374	null	David	null	Yang
8	15203	315	AW00015203	null	David	null	Lewis
9	15636	127	AW00015636	null	David	P	White
10	15783	31	AW00015783	null	David	null	Alexander
11	15908	329	AW00015908	null	David	M	Wilson
12	15991	635	AW00015991	null	David	null	Taylor
13	16138	315	AW00016138	null	David	E	Butler

Exactly as you expect. Now let's see when it is misleading.

Misleading Behavior of Basic Filtering

If you want to do equity filtering, basic filtering is great. For example you only want to pick David, and that's what we've done in above example. However what if you want to do similarity filtering? For example, let's say you are interested in FirstNames that has three characters of "Dan" in it. You can do it in the same search box of basic filtering, and this will give you list of all first names that has "Dan" in it.



And then you can click on OK. the result will be showing you all first names with "Dan" characters in it. So everything looks exactly as expected, right?

	123 CustomerKey	123 GeographyKey	A ^B _C CustomerAlternateKey	ABC 123 Title	A ^B _C FirstName	A ^B _C MiddleName	A ^B _C LastName
1	11020		53 AW00011020	null	Jordan	C	King
2	11051		21 AW00011051	null	Daniel		Johnson
3	11125		37 AW00011125	null	Dana	J	Navarro
4	11207		635 AW00011207	null	Danielle		Cox
5	11271		360 AW00011271	null	Danielle	C	Reed
6	11420		189 AW00011420	null	Jordan	C	Turner
7	11490		217 AW00011490	null	Danny		Rubio
8	11504		315 AW00011504	null	Jordan	P	
9	11541		547 AW00011541	null	Aidan	D	
10	11711		54 AW00011711	null	Daniel		
11	11798		611 AW00011798	null	Brendan	Q	
12	11818		312 AW00011818	null	Daniel	C	
13	11858		361 AW00011858	null	Jordan	A	
14	11950		312 AW00011950	null	Jordan		
15	12084		545 AW00012084	null	Aidan	O	
16	12361		22 AW00012361	null	Dana	C	
17	12587		9 AW00012587	null	Danny		
18	12678		15 AW00012678	null	Dana	S	

However as a Power Query geek I always look at the M script of each step to see what is happening behind the scene. For this case, here is the Power Query M script;

✕ ✓ f_x

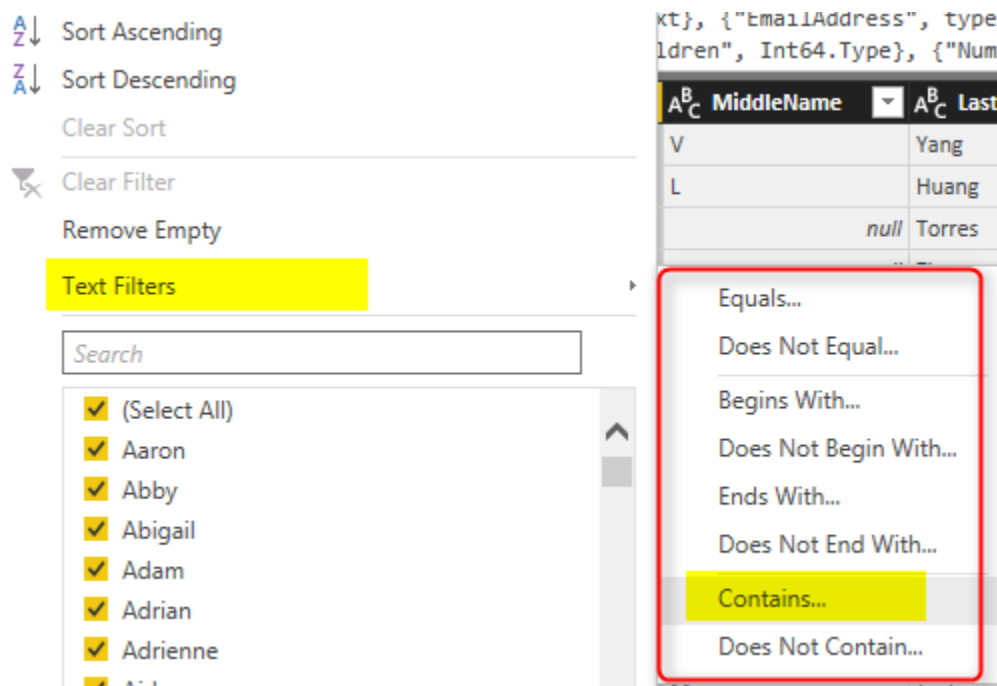
```
= Table.SelectRows("#Changed Type", each ([FirstName] = "Aidan" or [FirstName] = "Brendan" or [FirstName] = "Dana" or [FirstName] = "Daniel" or [FirstName] = "Danielle" or [FirstName] = "Danny" or [FirstName] = "Jordan"))
```

	CustomerAlternateKey	ABC 123 Title	A ^B _C FirstName	A ^B _C MiddleName	A ^B _C LastName
1	0011020		null Jordan	C	King
2	0011051		null Daniel		null Johnson
3	0011125		null Dana	J	Navarro
4	0011207		null Danielle		null Cox
5	0011271		null Danielle	C	Reed
6	0011420		null Jordan	C	Turner
7	0011490		null Danny		null Rubio


The script tells the whole story. Despite the fact that you typed in "Dan" and Power Query showed you all FirstNames that has "Dan" in it. the script still use equity filters for every individual FirstName. For this data set there won't be any issue obviously, because all FirstNames with "Dan" is already selected. However if new data rows coming in to this table in the future, and they have records with FirstNames that are not one of these values, for example Dandy, it won't be picked! As a result the filter won't work exactly as you expect. That's why I say this is misleading.

Advanced Filtering: Correct Filtering

Basic filtering is very handy with that search box and suggestions of items while you typing, but it is misleading easily as you have seen above, and can't give you what you expect in many real-world scenarios. Advanced Filtering is simply what you can see on top of the Search box, which depends on the data type of column can be Text Filters, Number Filters, Date Filters and etc. If we want to do the same "Dan" filter here, I can go to Text Filtering and select Contains.



You can even choose filters such as Begins With or Ends With or many other choices. For this example I'll select Contains, and type in Dan in the text box of Contains.



Filter Rows

☒ Basic ☐ Advanced

Show rows where: FirstName

contains

dan

☒ And ☐ Or

Type or select a value

OK

Cancel

After clicking on OK, you will see same result set (like the time that you did basic filtering with "Dan"), however the M script this time is different;

= Table.SelectRows(#"Changed Type", each Text.Contains([FirstName], "dan"))							
	CustomerAlternateKey	ABC 123 Title	A ^B _C FirstName	A ^B _C MiddleName	A ^B _C LastName		✓ Nam
1	0011020		null Jordan	C	King		
2	0011420		null Jordan	C	Turner		
3	0011504		null Jordan	P	Baker		
4	0011541		null Aidan	D	Ross		
5	0011798		null Brendan	Q	Chande		
6	0011858		null Jordan	A	Coleman		
7	0011950		null Jordan		null Carter		
8	0012084		null Aidan	O	Perry		
9	0012887		null Brendan	L	Pal		
10	0013018		null Brendan		null Xie		
11	0013095		null Jordan		null Griffin		
12	0013103		null Brendan		null Raji		
13	0013250		null Jordan	J	King		

You can see the big different now, This time M script used **Text.Contains** function. This function will pick every new values with "Dan" in the FirstName. There will be no values missing in this way. You will be amazed how many Power Query scenarios are not working correctly because of this very simple reason. There are advanced filters for all types of data types.

Summary

Basic Filtering is good only if you want to do equity filtering for values that exists in the current data set, however it won't work correctly if you want to check ranges, or contains or things that is not an exact equity filter. Advanced Filtering is the correct way of filtering in Power Query, and there are advanced filters for all types of data types; Numbers, Text, Date.... This is very simple fact, but not considering that will bring lots of unwanted behavior and incorrect insight to your Power BI solution.

Flawless Date Conversion in Power Query

Published Date: January 6, 2017

	Date1	Date2	Date3	Date4
1	1/1/2017	1/1/2017	1/1/2017	1/1/2017
2	1/2/2017	1/2/2017	1/2/2017	2/1/2017
3	1/3/2017	1/3/2017	1/3/2017	3/1/2017
4	1/4/2017	1/4/2017	1/4/2017	4/1/2017
5	1/5/2017	1/5/2017	1/5/2017	5/1/2017
6	1/6/2017	1/6/2017	1/6/2017	6/1/2017
7	1/7/2017	1/7/2017	1/7/2017	7/1/2017
8	1/8/2017	1/8/2017	1/8/2017	8/1/2017
9	1/9/2017	1/9/2017	1/9/2017	9/1/2017
10	1/10/2017	1/10/2017	1/10/2017	10/1/2017
11	1/11/2017	1/11/2017	1/11/2017	11/1/2017
12	1/12/2017	1/12/2017	1/12/2017	12/1/2017
13	1/13/2017	1/13/2017	1/13/2017	Error
14	1/14/2017	1/14/2017	1/14/2017	Error
15	1/15/2017	1/15/2017	1/15/2017	Error
16	1/16/2017	1/16/2017	1/16/2017	Error
17	1/17/2017	1/17/2017	1/17/2017	Error
18	1/18/2017	1/18/2017	1/18/2017	Error
19	1/19/2017	1/19/2017	1/19/2017	Error
20	1/20/2017	1/20/2017	1/20/2017	Error
21	1/21/2017	1/21/2017	1/21/2017	Error
22	1/22/2017	1/22/2017	1/22/2017	Error
23	1/23/2017	1/23/2017	1/23/2017	Error
24	1/24/2017	1/24/2017	1/24/2017	Error
25	1/25/2017	1/25/2017	1/25/2017	Error
26	1/26/2017	1/26/2017	1/26/2017	Error
27	1/27/2017	1/27/2017	1/27/2017	Error
28	1/28/2017	1/28/2017	1/28/2017	Error
29	1/29/2017	1/29/2017	1/29/2017	Error
30	1/30/2017	1/30/2017	1/30/2017	Error
31	1/31/2017	1/31/2017	1/31/2017	Error

Date Conversion is one of the simplest conversions in Power Query, however depends on locale on the system that you are working with Date Conversion might return

different result. In this post I'll show you an example of issue with date conversion and how to resolve it with Locale. In this post you'll learn that Power Query date conversion is dependent on the system that this conversion happens on, and can be fixed to a specific format. If you want to learn more about Power BI; Read [Power BI online book; from Rookie to Rock Star](#).

Prerequisite

The sample data set for this post is here: [book1](#)

Different Formats of Date

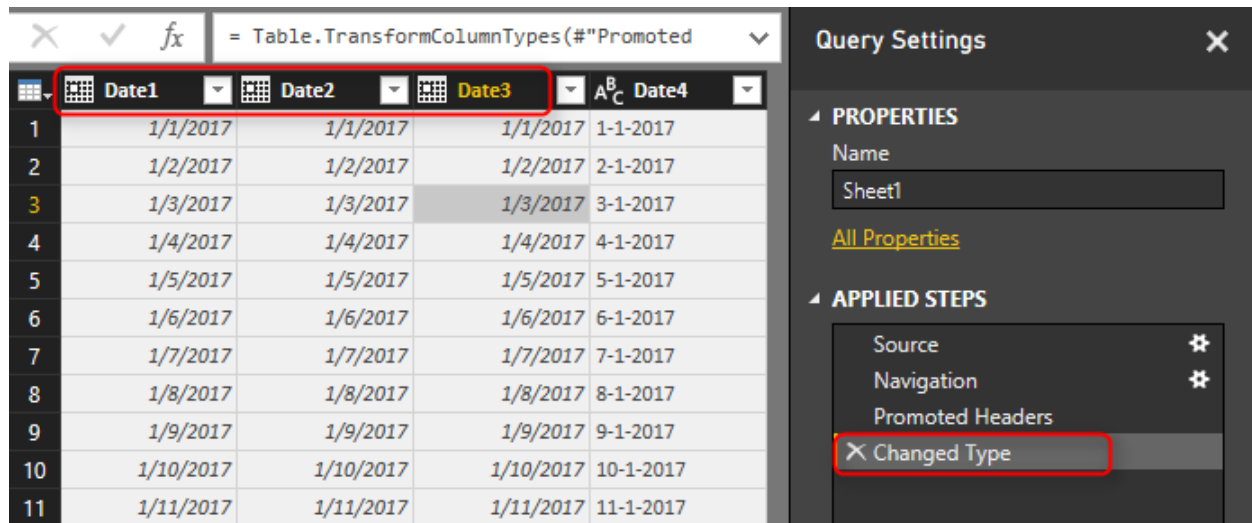
Most of the countries uses YMD format, however some of them use MDY or DMY more frequently. [This wikipedia page](#) explains different formats of date in each country. Data set below have different formats in it;

Date1	Date2	Date3	Date4
2017-1-1	2017/1/1	1-1-2017	1-1-2017
2017-1-2	2017/1/2	1-2-2017	2-1-2017
2017-1-3	2017/1/3	1-3-2017	3-1-2017
2017-1-4	2017/1/4	1-4-2017	4-1-2017
2017-1-5	2017/1/5	1-5-2017	5-1-2017
2017-1-6	2017/1/6	1-6-2017	6-1-2017
2017-1-7	2017/1/7	1-7-2017	7-1-2017
2017-1-8	2017/1/8	1-8-2017	8-1-2017
2017-1-9	2017/1/9	1-9-2017	9-1-2017
2017-1-10	2017/1/10	1-10-2017	10-1-2017
2017-1-11	2017/1/11	1-11-2017	11-1-2017
2017-1-12	2017/1/12	1-12-2017	12-1-2017
2017-1-13	2017/1/13	1-13-2017	13-1-2017
2017-1-14	2017/1/14	1-14-2017	14-1-2017
2017-1-15	2017/1/15	1-15-2017	15-1-2017
2017-1-16	2017/1/16	1-16-2017	16-1-2017
2017-1-17	2017/1/17	1-17-2017	17-1-2017
2017-1-18	2017/1/18	1-18-2017	18-1-2017
2017-1-19	2017/1/19	1-19-2017	19-1-2017
2017-1-20	2017/1/20	1-20-2017	20-1-2017
2017-1-21	2017/1/21	1-21-2017	21-1-2017
2017-1-22	2017/1/22	1-22-2017	22-1-2017
2017-1-23	2017/1/23	1-23-2017	23-1-2017
2017-1-24	2017/1/24	1-24-2017	24-1-2017
2017-1-25	2017/1/25	1-25-2017	25-1-2017
2017-1-26	2017/1/26	1-26-2017	26-1-2017
2017-1-27	2017/1/27	1-27-2017	27-1-2017
2017-1-28	2017/1/28	1-28-2017	28-1-2017
2017-1-29	2017/1/29	1-29-2017	29-1-2017
2017-1-30	2017/1/30	1-30-2017	30-1-2017
2017-1-31	2017/1/31	1-31-2017	31-1-2017

In above table, first two columns (Date1, and Date2) are in YYYY-MM-DD and YYYY/MM/DD format, which is the most common format of date. Date3 is MM-DD-YYYY format (very common in USA and some other countries), and Date4 is DD-MM-YYYY format, which is mostly common in New Zealand, Australia, and some other countries. Now let's see what happens if we load this data into Power BI.

Automatic Type Conversion

Power BI leverages automatic data type conversion. This automatic action sometimes is useful, sometimes not! If you open a Power BI Desktop file and get data from specified data set. in Query Editor you will see the data types converted automatically at some level. Here is the data loaded into Query Editor window with automatic data type conversion;

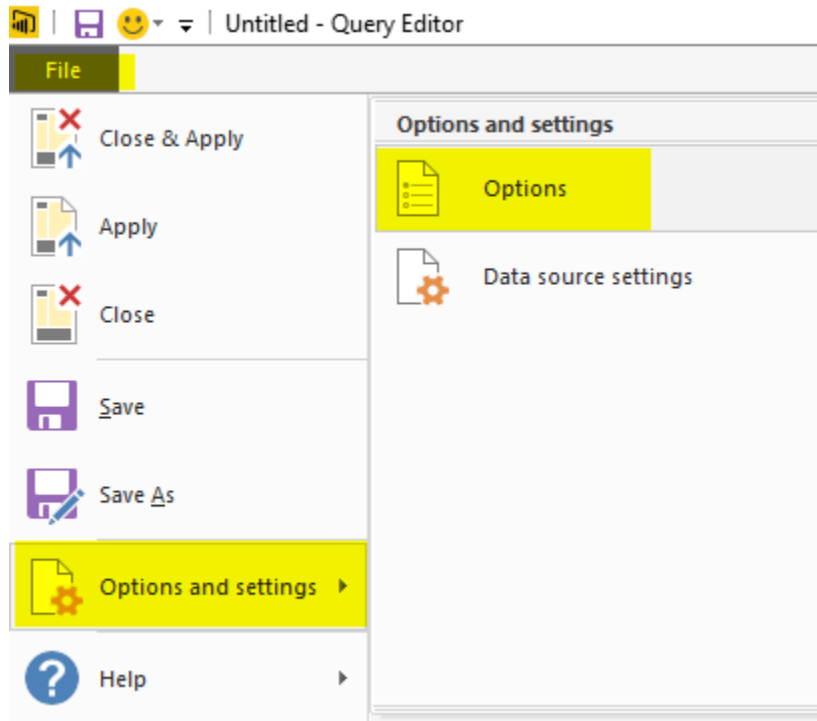


	Date1	Date2	Date3	Date4
1	1/1/2017	1/1/2017	1/1/2017	1-1-2017
2	1/2/2017	1/2/2017	1/2/2017	2-1-2017
3	1/3/2017	1/3/2017	1/3/2017	3-1-2017
4	1/4/2017	1/4/2017	1/4/2017	4-1-2017
5	1/5/2017	1/5/2017	1/5/2017	5-1-2017
6	1/6/2017	1/6/2017	1/6/2017	6-1-2017
7	1/7/2017	1/7/2017	1/7/2017	7-1-2017
8	1/8/2017	1/8/2017	1/8/2017	8-1-2017
9	1/9/2017	1/9/2017	1/9/2017	9-1-2017
10	1/10/2017	1/10/2017	1/10/2017	10-1-2017
11	1/11/2017	1/11/2017	1/11/2017	11-1-2017

You can see the Changed Type step that applied automatically and converted first three columns in the data set to data type of Date, and left the Date4 as data type Text. Automatic Date type conversion understand characters like / or -, and apply conversion correctly in both cases (Date1, and Date2)

* If you are running this example on your machine you might see different result, because Power Query uses the locale of machine to do data type conversion. Locale of my machine is US format, so it understand format of USA and convert it automatically. If you have different locale the conversion might result differently. We will go through that in a second.

If the automatic data type conversion is not something you want, you can remove that step simply. Or if you want to disable the automatic type conversion, Go to File, Options and Settings, then Options.



In the Options window, under Current File, Data Load section, you can enable or disable the automatic data type detection if you wish to. (In this example I'll keep it enabled)

Options

GLOBAL

Data Load
Query Editor
DirectQuery
R Scripting
Security
Privacy
Updates
Usage Data
Diagnostics
Preview Features
Auto Recovery

CURRENT FILE

Data Load
Regional Settings
Privacy
Auto Recovery

Type Detection

☒ Automatically detect column types and headers for unstructured sources

Relationships

☒ Import relationships from data sources ⓘ
☐ Update relationships when refreshing queries ⓘ
☒ Autodetect new relationships after data is loaded ⓘ

Time Intelligence

☒ Auto Date/Time ⓘ

Background Data

☒ Allow data preview to download in the background

Parallel Loading of Tables

☒ Enable parallel loading of tables ⓘ

OK

Cancel

You can also check the Locale of your current machine in the Regional Settings section of Options Window;

Options

GLOBAL

Data Load
Query Editor
DirectQuery
R Scripting
Security
Privacy
Updates
Usage Data
Diagnostics
Preview Features
Auto Recovery

CURRENT FILE

Data Load
Regional Settings
Privacy
Auto Recovery

Locale ⓘ

English (United States) ▼

OK

Cancel

Date Conversion Issue

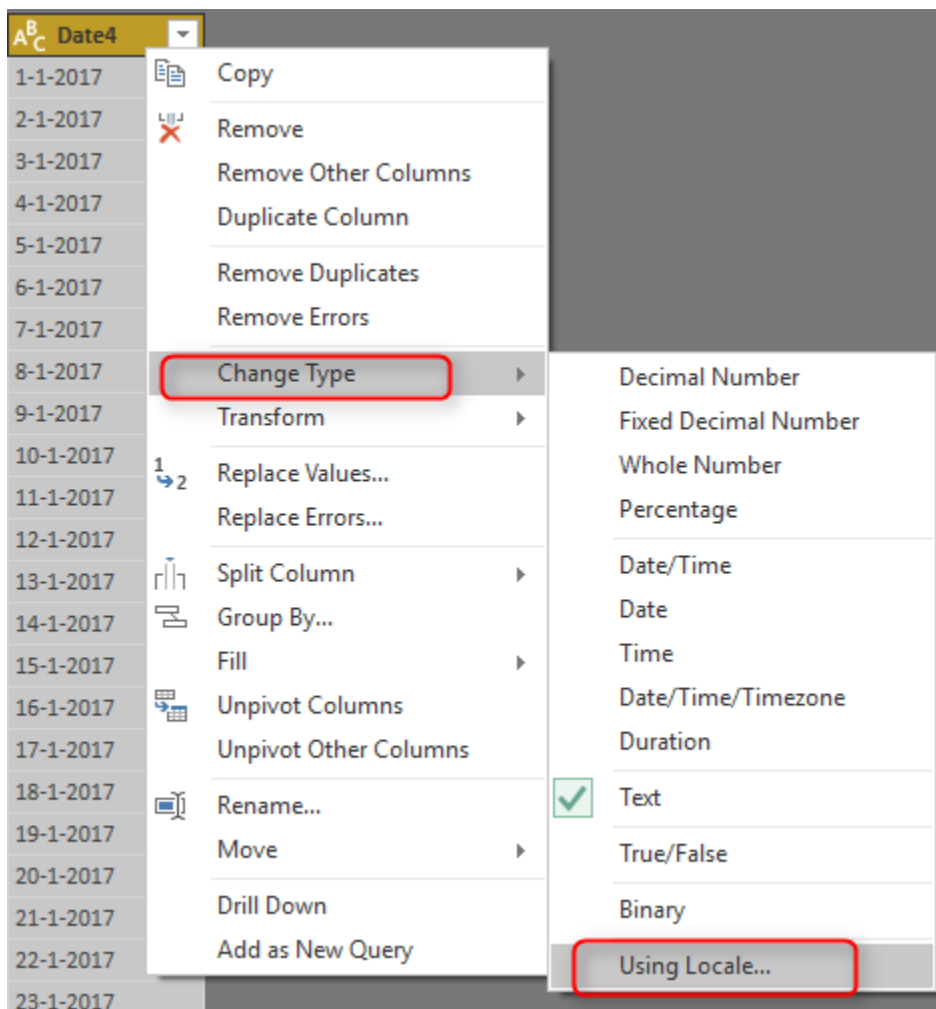
Date4 in the data set isn't converted properly, and that's because the Locale of current system (my machine) is English (United States). I can change the locale to English (New Zealand) in the Options window, and Refresh the data, but this will corrupt the existing Date Conversion of Date3 column. If I try to change the data type of Date4 column myself, the result will not be correct and I'll see some errors;

	Date1	Date2	Date3	Date4
1	1/1/2017	1/1/2017	1/1/2017	1/1/2017
2	1/2/2017	1/2/2017	1/2/2017	2/1/2017
3	1/3/2017	1/3/2017	1/3/2017	3/1/2017
4	1/4/2017	1/4/2017	1/4/2017	4/1/2017
5	1/5/2017	1/5/2017	1/5/2017	5/1/2017
6	1/6/2017	1/6/2017	1/6/2017	6/1/2017
7	1/7/2017	1/7/2017	1/7/2017	7/1/2017
8	1/8/2017	1/8/2017	1/8/2017	8/1/2017
9	1/9/2017	1/9/2017	1/9/2017	9/1/2017
10	1/10/2017	1/10/2017	1/10/2017	10/1/2017
11	1/11/2017	1/11/2017	1/11/2017	11/1/2017
12	1/12/2017	1/12/2017	1/12/2017	12/1/2017
13	1/13/2017	1/13/2017	1/13/2017	Error
14	1/14/2017	1/14/2017	1/14/2017	Error
15	1/15/2017	1/15/2017	1/15/2017	Error
16	1/16/2017	1/16/2017	1/16/2017	Error
17	1/17/2017	1/17/2017	1/17/2017	Error
18	1/18/2017	1/18/2017	1/18/2017	Error
19	1/19/2017	1/19/2017	1/19/2017	Error
20	1/20/2017	1/20/2017	1/20/2017	Error
21	1/21/2017	1/21/2017	1/21/2017	Error
22	1/22/2017	1/22/2017	1/22/2017	Error
23	1/23/2017	1/23/2017	1/23/2017	Error
24	1/24/2017	1/24/2017	1/24/2017	Error
25	1/25/2017	1/25/2017	1/25/2017	Error
26	1/26/2017	1/26/2017	1/26/2017	Error
27	1/27/2017	1/27/2017	1/27/2017	Error
28	1/28/2017	1/28/2017	1/28/2017	Error
29	1/29/2017	1/29/2017	1/29/2017	Error
30	1/30/2017	1/30/2017	1/30/2017	Error
31	1/31/2017	1/31/2017	1/31/2017	Error

You can see that conversion didn't happen correctly, and also it returned Error in some cells, because my machine is expecting MM/DD/YYYY, but the date format in the column is DD/MM/YYYY which is not that format. So it can convert first 12 records, because it places the day as month in the result! and the rest it can't because there is no month 13 or more.

Date Conversion Using Locale

Fortunately you can do date conversion using specific Locale. All you need to do is to go through right click and data type conversion using Locale;



In the Change Type with Locale, choose the Data Type to be Date (Normally Locale is for Date, Time, and Numbers). and then set Locale to be English (New Zealand). You can also see some sample input values for this locale there.



Change Type with Locale


Change the data type and select the locale of origin.

Data Type

Date

Locale

English (New Zealand)

 Sample input values:

29/03/2016

Tuesday, 29 March 2016

29 March

March 2016

OK

Cancel

With this simple change, you can now see the Date4 column converted correctly. You still see that in MM/DD/YYYY format in Query Editor window, and that's because my machine's date format is this. The actual column data type is Date however, which is the correct format to work with.

	Date1	Date2	Date3	Date4
1	1/1/2017	1/1/2017	1/1/2017	1/1/2017
2	1/2/2017	1/2/2017	1/2/2017	1/2/2017
3	1/3/2017	1/3/2017	1/3/2017	1/3/2017
4	1/4/2017	1/4/2017	1/4/2017	1/4/2017
5	1/5/2017	1/5/2017	1/5/2017	1/5/2017
6	1/6/2017	1/6/2017	1/6/2017	1/6/2017
7	1/7/2017	1/7/2017	1/7/2017	1/7/2017
8	1/8/2017	1/8/2017	1/8/2017	1/8/2017
9	1/9/2017	1/9/2017	1/9/2017	1/9/2017
10	1/10/2017	1/10/2017	1/10/2017	1/10/2017
11	1/11/2017	1/11/2017	1/11/2017	1/11/2017
12	1/12/2017	1/12/2017	1/12/2017	1/12/2017
13	1/13/2017	1/13/2017	1/13/2017	1/13/2017
14	1/14/2017	1/14/2017	1/14/2017	1/14/2017
15	1/15/2017	1/15/2017	1/15/2017	1/15/2017
16	1/16/2017	1/16/2017	1/16/2017	1/16/2017
17	1/17/2017	1/17/2017	1/17/2017	1/17/2017
18	1/18/2017	1/18/2017	1/18/2017	1/18/2017
19	1/19/2017	1/19/2017	1/19/2017	1/19/2017
20	1/20/2017	1/20/2017	1/20/2017	1/20/2017
21	1/21/2017	1/21/2017	1/21/2017	1/21/2017
22	1/22/2017	1/22/2017	1/22/2017	1/22/2017
23	1/23/2017	1/23/2017	1/23/2017	1/23/2017
24	1/24/2017	1/24/2017	1/24/2017	1/24/2017
25	1/25/2017	1/25/2017	1/25/2017	1/25/2017
26	1/26/2017	1/26/2017	1/26/2017	1/26/2017
27	1/27/2017	1/27/2017	1/27/2017	1/27/2017
28	1/28/2017	1/28/2017	1/28/2017	1/28/2017
29	1/29/2017	1/29/2017	1/29/2017	1/29/2017
30	1/30/2017	1/30/2017	1/30/2017	1/30/2017
31	1/31/2017	1/31/2017	1/31/2017	1/31/2017

A Look at M Script

A look at formula bar and Power Query script shows that the locale used for this conversion is "en-NZ", while other data type conversion are not using that.

let

```
Source = Excel.Workbook(File.Contents("C:\Users\Reza\SkyDrive\Blog\DateConversion\Book1.xlsx"),
null, true),
```

```

Sheet1_Sheet = Source([Item="Sheet1",Kind="Sheet"])[Data],
#"Promoted Headers" = Table.PromoteHeaders(Sheet1_Sheet),
#"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{
    {"Date1", type date},
    {"Date2", type date},
    {"Date3", type date},
    {"Date4", type text}
}),
#"Changed Type with Locale" = Table.TransformColumnTypes(#"Changed Type", {"Date4", type date}},
"en-NZ")
in
#"Changed Type with Locale"

```

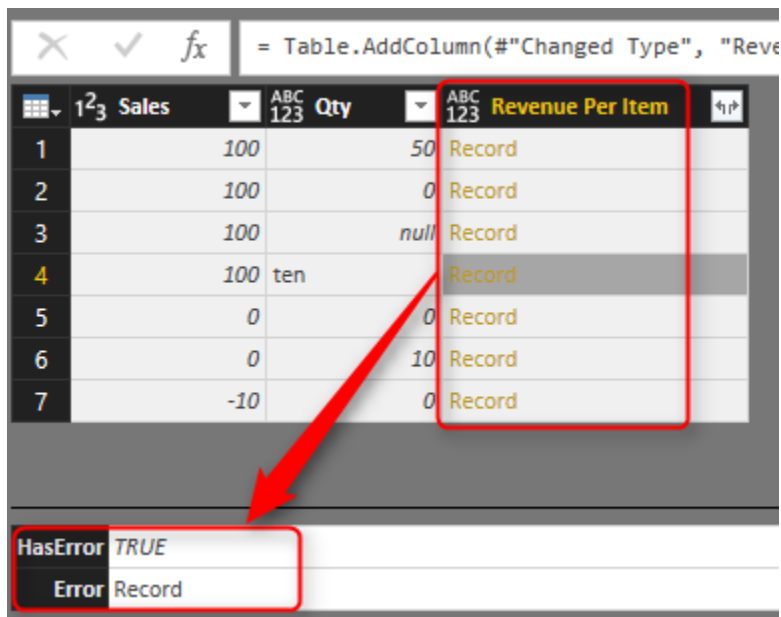
In the script you can see the different of data type change using Locale which uses locale as the last parameter, instead of without locale. This brings a very important topic in mind; correct date type conversion is locale dependent, and you can get it always working if you mention Locale in date type conversion.

Summary

Date Type conversion can be tricky depends on the locale of system you are working on. To get the correct Date Type conversion recommendation is to use Locale for type conversion. some of formats might work even without using Locale. For example I have seen YYYYMMDD is working fine in all locales I worked with so far, but DDMMYYYY or MMDDYYYY might work differently.

Make Your Numeric Division Faultless in Power Query

Published Date: June 20, 2016



The screenshot shows the Power Query Editor interface. The formula bar at the top displays the M code: `= Table.AddColumn(#"Changed Type", "Revenue Per Item", ...)`. Below the formula bar is a table with three columns: Sales, Qty, and Revenue Per Item. The table contains seven rows of data. A red box highlights the Revenue Per Item column, and a red arrow points to the HasError column which shows TRUE for the first record.

	Sales	Qty	Revenue Per Item
1	100	50	Record
2	100	0	Record
3	100	null	Record
4	100	ten	Record
5	0	0	Record
6	0	10	Record
7	-10	0	Record

Below the table, the HasError column is shown with the value TRUE for the first record.

HasError	Error
TRUE	Record

When you work with data it is normal that you apply numeric calculations. Numeric calculations in Power Query depends on the nature of data returns different results. One of the most error prone calculations is division. Power Query behave differently when you divide a number by zero, zero by zero, number by null, and non-numeric values. One of the most frustrating facts is that not all of these calculations ends up to an error. So you can't just remove error rows simply. In this post I'll explain some examples of output for division and a method to find these rows.

Sample Data Set

For this post I'll use a sample excel file which has most of possible combinations that I might face in a division calculation. The table below is some records with Sales Amount and Quantity. and as a simple calculation I want to find out Revenue Per Item which would be result of $[Sales]/[Quantity]$.

	A	B
1	Sales	Qty
2	100	50
3	100	0
4	100	
5	100	ten
6	0	0
7	0	10
8	-10	0

In table above there are nulls, texts, zeros, negative, and positive values. Now let's bring the table into Power Query (Excel or Power BI) and apply the division

Simple Division Calculation

Here is the data set loaded into Power Query. As you can see Quantity column shows the data type as numeric and text.

	123 Sales	ABC 123 Qty
1	100	50
2	100	0
3	100	null
4	100	ten
5	0	0
6	0	10
7	-10	0

Now If I apply a simple division calculation as a new custom column

Add Custom Column

New column name

Revenue Per Item

Custom column formula:

= [Sales]/[Qty]

The result would be as below;

fx = Table.AddColumn(#"Changed Type", "Revenue Per Item", each [Sales]/[Qty])

	Sales	Qty	Revenue Per Item
1	100	50	2
2	100	0	Infinity
3	100	null	null
4	100	ten	Error
5	0	0	NaN
6	0	10	0
7	-10	0	-Infinity

As you see the result set has different outputs depends on the inputs. if number is divided by a zero value result would be positive or negative infinity (depends on the number). if one of the values be null, then result set would be null. if zero divided by zero then result would be NaN! and in case of dividing a number by string or reverse there will be an error raised. Now Let's look at each output separately.

Error Output

Error in sample above happened when one of the values is not number. Fortunately errors can be simply found by TRY keyword. Here is how I change the calculation of Revenue Per Item:

= try [Sales]/[Qty]

Add Custom Column

New column name

Revenue Per Item

Custom column formula:

= try [Sales]/[Qty]

Result set this time would be a Record for each calculation.

fx = Table.AddColumn(#"Changed Type", "Revenue Per Item", ...)

	Sales	Qty	Revenue Per Item
1	100	50	Record
2	100	0	Record
3	100	null	Record
4	100	ten	Record
5	0	0	Record
6	0	10	Record
7	-10	0	Record

HasError	Error
TRUE	Record

The record has two columns: HasError (which says does this record contains error or not), and [Error](#) Record (which would be the error happened in details). So I can add a custom column with a condition on HasError to see if record contains error or not. In expression below if I find an error I will return zero as the result.

= if [Revenue Per Item][HasError] then 0 else [Revenue Per Item][Value]

Add Custom Column

New column name

Revenue Per Item - Error Removed

Custom column formula:

= if [Revenue Per Item][HasError] then 0 else [Revenue Per Item][Value]

Available columns:

- Sales
- Qty
- Revenue Per Item

The result set this time would be:

	1 ² Sales	ABC 123 Qty	ABC 123 Revenue Per Item	ABC 123 Revenue Per Item - Error Removed
1	100	50	Record	2
2	100	0	Record	Infinity
3	100	null	Record	null
4	100	ten	Record	0
5	0	0	Record	NaN
6	0	10	Record	0
7	-10	0	Record	-Infinity

In this example I just returned zero if I find error. but you can return error message if you like with [Revenue Per Item][ErrorMessage]. This method is great error handling method when an error out of blue happens in your data set. I always recommend using TRY method to get rid of errors that might stop the whole solution to work properly.

I have to mention that steps above are separated to show you how the output of try expression looks like. In fact you can combine both steps above in single step with TRY OTHERWISE as below (Thanks to [Maxim Zelensky](#) for pointing this out);

= try [Sales]/[Qty] otherwise 0

Infinity

Error output can be handled with TRY. However Infinity and -Infinity are not errors! These are number values in Power Query, named [Number.PositiveInfinity](#) and [Number.NegativeInfinity](#).

	1 ² Sales	ABC 123 Qty	ABC 123 Revenue Per Item	ABC 123 Revenue Per Item - Error Removed
1	100	50	Record	2
2	100	0	Record	Infinity
3	100	null	Record	null
4	100	ten	Record	0
5	0	0	Record	NaN
6	0	10	Record	0
7	-10	0	Record	-Infinity

- PositiveInfinity happens when a positive number divided by zero
- NegativeInfinity happens when a negative number divided by zero

You can't find these with error handling because as I mentioned earlier these are not error values! You can however check these values to see if a value is `NegativeInfinity` or `PositiveInfinity` with sample code below:

```
Pinfinity=(if x=Number.PositiveInfinity then false else true),
Ninfinity=(if x=Number.NegativeInfinity then false else true)
```

NaN

NaN is another output which happens when zero is divided by zero. [NaN](#) is a number value like positive and negative infinity. So you can't use error handling to spot them out of millions of records.

	1 ² Sales	ABC 123 Qty	ABC 123 Revenue Per Item	ABC 123 Revenue Per Item - Error Removed
1	100	50	Record	2
2	100	0	Record	Infinity
3	100	null	Record	null
4	100	ten	Record	0
5	0	0	Record	NaN
6	0	10	Record	0
7	-10	0	Record	-Infinity

You can find it with [Number.IsNaN](#) function which works as below;

```
Nan=(if Number.IsNaN(x) then false else true)
```

Null Check

Null values always happens in the data, and best practice is always replace them with default values. In numeric calculations if a null value appears in one of the values the result of the calculation will be null.

	1 ² Sales	ABC 123 Qty	ABC 123 Revenue Per Item	ABC 123 Revenue Per Item - Error Removed
1	100	50	Record	2
2	100	0	Record	Infinity
3	100	null	Record	null
4	100	ten	Record	0
5	0	0	Record	NaN
6	0	10	Record	0
7	-10	0	Record	-Infinity


You can simply find nulls with if condition such as below;

Null=(if x=null then false else true)

Function to Check All Anomalies

Anomalies in outputs such as above examples happens in most of the cases, and I found it useful to have a function to check all these options. The function below checks Null, NaN, PositiveInfinity, and NegativeInfinity. It doesn't check errors however. Error handling is best to be applied on the calculation level as we've done earlier in this post. Here is the code for the function:

```
let
    Source = (x as any) =>
        let
            Null=(if x=null then false else true),
            Pinfinity=(if x=Number.PositiveInfinity then false else true),
            Ninfinty=(if x=Number.NegativeInfinity then false else true),
            Nan=(if Number.IsNaN(x) then false else true)
        in
            Null and Pinfinity and Ninfinty and Nan
in
    Source
```

 Advanced Editor

ValidateDivisionResult

```
let
    Source = (x as any) =>
        let
            Null=(if x=null then false else true),
            Pinfinity=(if x=Number.PositiveInfinity then false else true),
            Ninfinty=(if x=Number.NegativeInfinity then false else true),
            Nan=(if Number.IsNaN(x) then false else true)
        in
            Null and Pinfinity and Ninfinty and Nan
in
    Source
```

With function above, now I can add a custom column to my data set, and validate rows;

Add Custom Column

New column name

Validation Result

Custom column formula:

```
=ValidateDivisionResult([#"Revenue Per Item - Error Removed"])
```

Final result set shows me which record is validated and which one is not. Note that the row containing error has been handled previously so it is validating as true here.

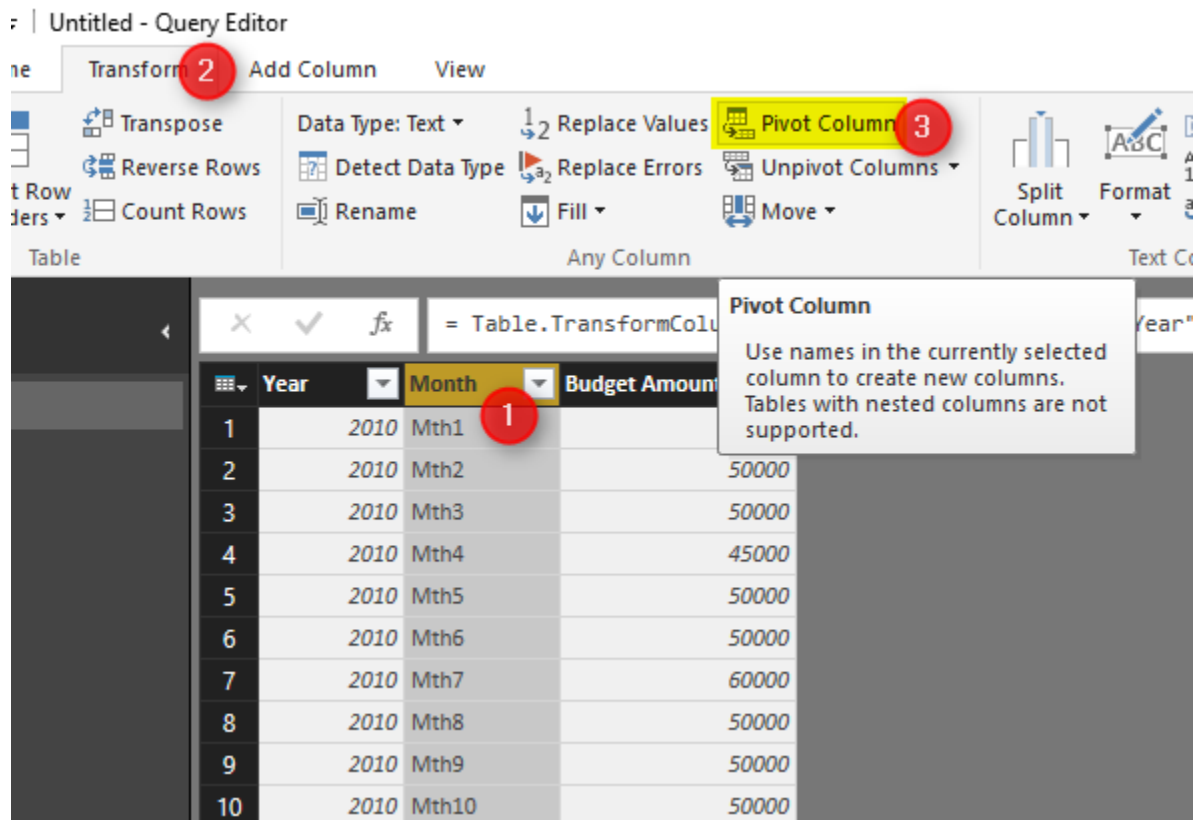
	123 Sales	ABC 123 Qty	ABC 123 Revenue Per Item	ABC 123 Error Removed	ABC 123 Validation Result
1	100	50	Record	2	TRUE
2	100	0	Record	Infinity	FALSE
3	100	null	Record	null	FALSE
4	100	ten	Record	0	TRUE
5	0	0	Record	NaN	FALSE
6	0	10	Record	0	TRUE
7	-10	0	Record	-Infinity	FALSE

Summary

Divide by zero is not the only error that happens in Power Query. Power Query returns different results such as Error, NaN, Positive Infinity, Negative Infinity, and Null. For a proper faultless calculation you have to consider all these exceptions. In this post You've learned how to spot these exceptions easily and make your calculation faultless with small amount of effort. If you like to learn more about Power Query and Power BI read [Power BI online book; from Rookie to Rock Star](#).

Pivot and Unpivot with Power BI

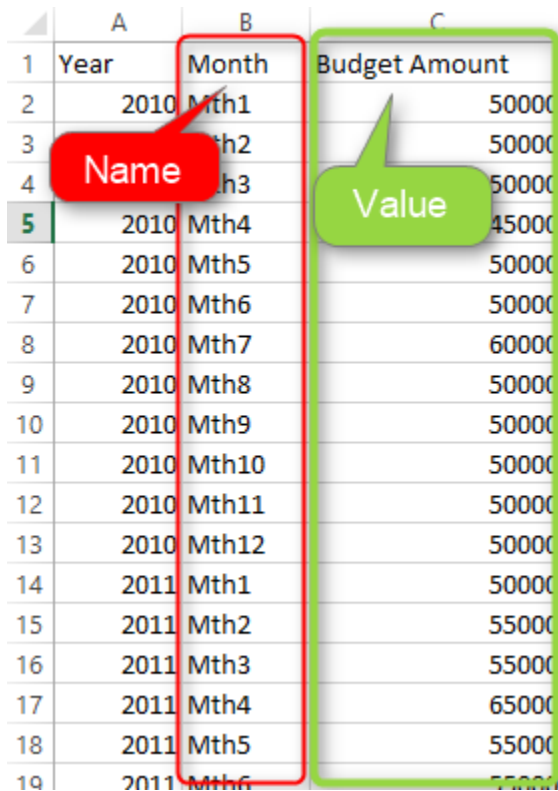
Published Date: April 7, 2016



Turning columns to rows, or rows to columns is easy with Power Query and Power BI. There are many situations that you get a name, value data source, and wants to convert that into columns with values underneath. On the other hand many times you get multiple columns and want to change it to name, value structure with a column for name, another column for value. That's why Pivot and Unpivot is for. In this post I'll get you through basic pivot and unpivot. If you want to read more about Power Query read it from [Power BI online book](#).

Pivot: Turning Name,Value Rows to Columns

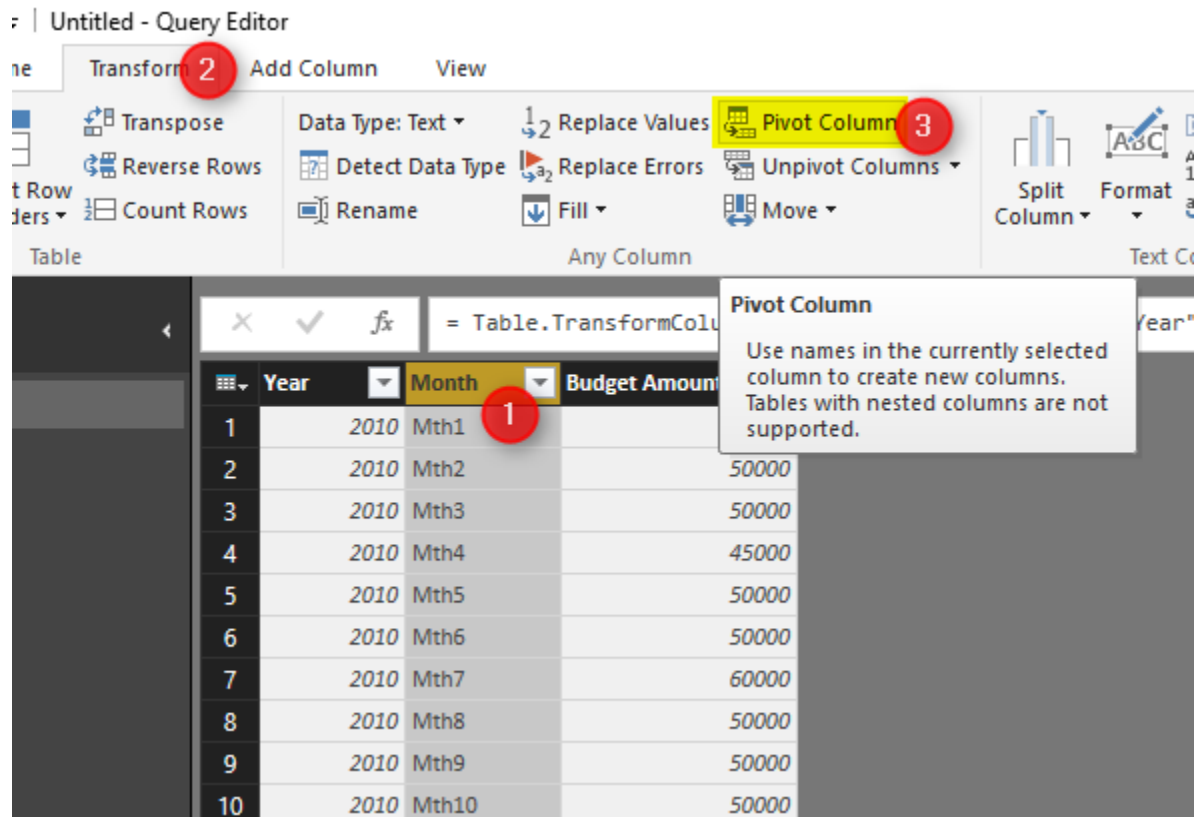
Consider we have a data source like this:



	A	B	C
1	Year	Month	Budget Amount
2	2010	Mth1	50000
3		Mth2	50000
4		Mth3	50000
5	2010	Mth4	45000
6	2010	Mth5	50000
7	2010	Mth6	50000
8	2010	Mth7	60000
9	2010	Mth8	50000
10	2010	Mth9	50000
11	2010	Mth10	50000
12	2010	Mth11	50000
13	2010	Mth12	50000
14	2011	Mth1	50000
15	2011	Mth2	55000
16	2011	Mth3	55000
17	2011	Mth4	65000
18	2011	Mth5	55000
19	2011	Mth6	55000

above data set is budget information. If we want to spread the table with a column for every month, we can simply use Pivot as below:

first click on the column that contains *names*, in this example it would be Month column. Then from Transform menu tab, choose Pivot Column.



The Pivot dialog box asks you to choose the *Value* column, which is Budget Amount in this example

Pivot Column

Use the names in column "Month" to create new columns.

Values Column ⓘ

Budget Amount

▸ Advanced options

[Learn more about Pivot Column](#)

OK

Cancel

and then simply and easily I have the pivoted result set;

Formula Bar: `= Table.Pivot("#Changed Type", List.Distinct("#Changed Type"[Month]), "Month", "Budget Amount", List.Sum)`

	Year	Mth1	Mth2	Mth3	Mth4	Mth5	Mth6	Mth7	Mth8	Mth9	Mth10	Mth11	Mth12
1	2010	50000	50000	50000	45000	50000	50000	60000	50000	50000	50000	50000	50000
2	2011	50000	55000	55000	65000	55000	55000	55000	55000	55000	55000	55000	55000
3	2012	65000	65000	65000	65000	80000	65000	65000	65000	65000	65000	65000	65000
4	2013	70000	70000	70000	70000	65000	70000	70000	70000	80000	70000	70000	70000

You can see that I have a column for each month now. Year column was just passed through. I can have as many as columns I want to pass through. The important factor for Pivot is that there should be a name column, and a value column. You can also see the Table.Pivot script of Power Query generated for this example in above screenshot.

Now let's see what happens if name value is a bit different;

Year	Month	Budget Amount
2010	Mth1	50000
2010	Mth2	50000
2010	Mth3	50000
2010	Mth4	45000
2010	Mth5	50000
2010	Mth6	50000
2010	Mth7	60000
2010	Mth8	50000
2010	Mth9	50000
2010	Mth10	50000
2010	Mth11	50000
2010	Mth11	50000
2010	Mth12	50000
2011	Mth1	50000
2011	Mth2	55000
2011	Mth3	55000
2011	Mth5	55000
2011	Mth6	55000
2011	Mth7	55000
2011	Mth8	55000

In this example I have two records for a single name (Mth 11 2010 is repeated). and for some names I don't have any records at all (Mth 4 2011 is missing)

Pivot dialog has the option to choose the aggregation function, and that is specially for cases that a name appeared more than once in the data set. default aggregation is Sum,

Pivot Column

Use the names in column "Month" to create new columns.

Values Column ⓘ

Budget Amount

Advanced options

Aggregate Value Function ⓘ

Sum

[Learn more about Pivot Column](#)

OK

Cancel

So the default Pivot will result as below:

	Year	Mth1	Mth2	Mth3	Mth4	Mth5	Mth6	Mth7	Mth8	Mth9	Mth10	Mth11	Mth12
1	2010	50000	50000	50000	45000	50000	50000	60000	50000	50000	50000	100000	50000
2	2011	50000	55000	55000	null	55000	55000	55000	55000	55000	55000	55000	55000
3	2012	65000	65000	65000	65000	80000	65000	65000	65000	65000	65000	65000	65000
4	2013	70000	70000	70000	70000	65000	70000	70000	70000	80000	70000	70000	70000

However if the aggregation be set to Do Not Aggregate. then you will get an error when a name is repeated in the data set

Pivot Column

Use the names in column "Month" to create new columns.

Values Column ⓘ

Budget Amount

Advanced options

Aggregate Value Function ⓘ

Don't Aggregate

[Learn more about Pivot Column](#)

OK

Cancel

Here is the error value in the result set;

fx = Table.Pivot(#"Changed Type", List.Distinct(#"Changed Type"[Month]), "Month", "Budget Amount")

	Year	Mth1	Mth2	Mth3	Mth4	Mth5	Mth6	Mth7	Mth8	Mth9	Mth10	Mth11	Mth12
1	2010	50000	50000	50000	45000	50000	50000	60000	50000	50000	50000	Error	50000
2	2011	50000	55000	55000	null	55000	55000	55000	55000	55000	55000	55000	55000
3	2012	65000	65000	65000	65000	80000	65000	65000	65000	65000	65000	65000	65000
4	2013	70000	70000	70000	70000	65000	70000	70000	70000	80000	70000	70000	70000

and the error would be:

Expression.Error: There were too many elements in the enumeration to complete the operation.

Details:

List

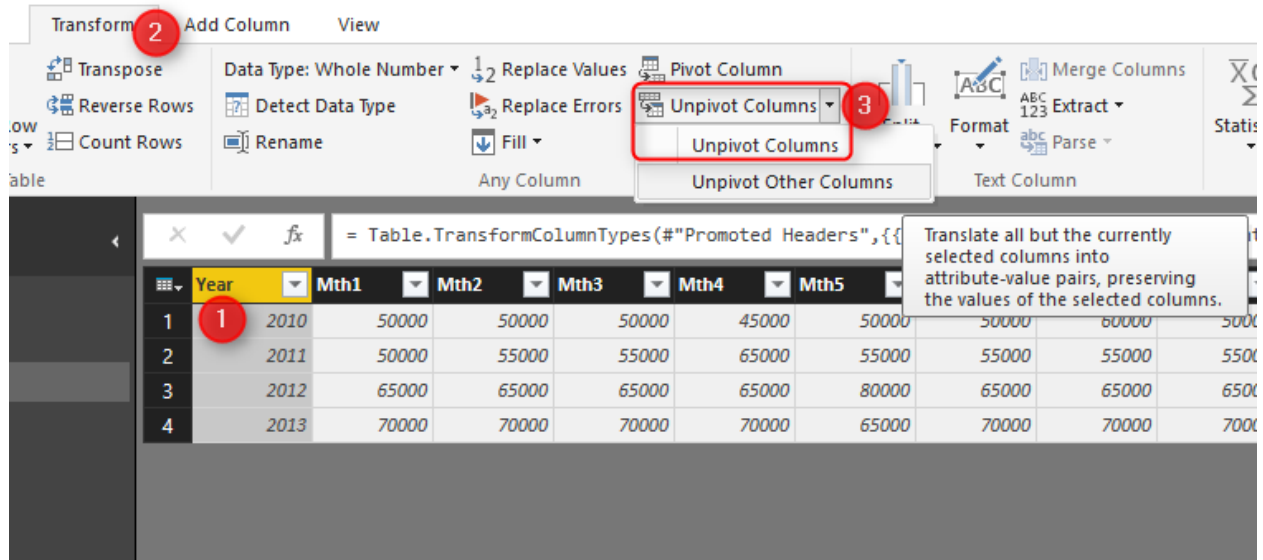
So Pivot is easy and simple to do, but you have to be careful about the nature and quality of source data set. If it is normal to have a name repeated in the source data, then an aggregation needs to be set properly. if you expect each name to appear once, then setting it as Do Not Aggregate works better because you can use error handling mechanism in Power Query to handle error somehow.

Unpivot; Turning Columns to Rows; Name, Values

Unpivot does the reverse. it turns multiple column headers into a single column but in rows. and store their values in another column. Here is an example of Budget data that usually you get from finance department;

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Year	Mth1	Mth2	Mth3	Mth4	Mth5	Mth6	Mth7	Mth8	Mth9	Mth10	Mth11	Mth12
2	2010	50000	50000	50000	45000	50000	50000	60000	50000	50000	50000	50000	50000
3	2011	50000	55000	55000	65000	55000	55000	55000	55000	55000	55000	55000	55000
4	2012	65000	65000	65000	65000	80000	65000	65000	65000	65000	65000	65000	65000
5	2013	70000	70000	70000	70000	65000	70000	70000	70000	80000	70000	70000	70000

You can click on Columns that you want to unpivot, and then select Unpivot columns (or you can do reverse, select pass through columns, and select unpivot other columns);



Transform Add Column View

Transpose Reverse Rows Count Rows

Data Type: Whole Number Replace Values Replace Errors Fill

Pivot Column Unpivot Columns Unpivot Other Columns

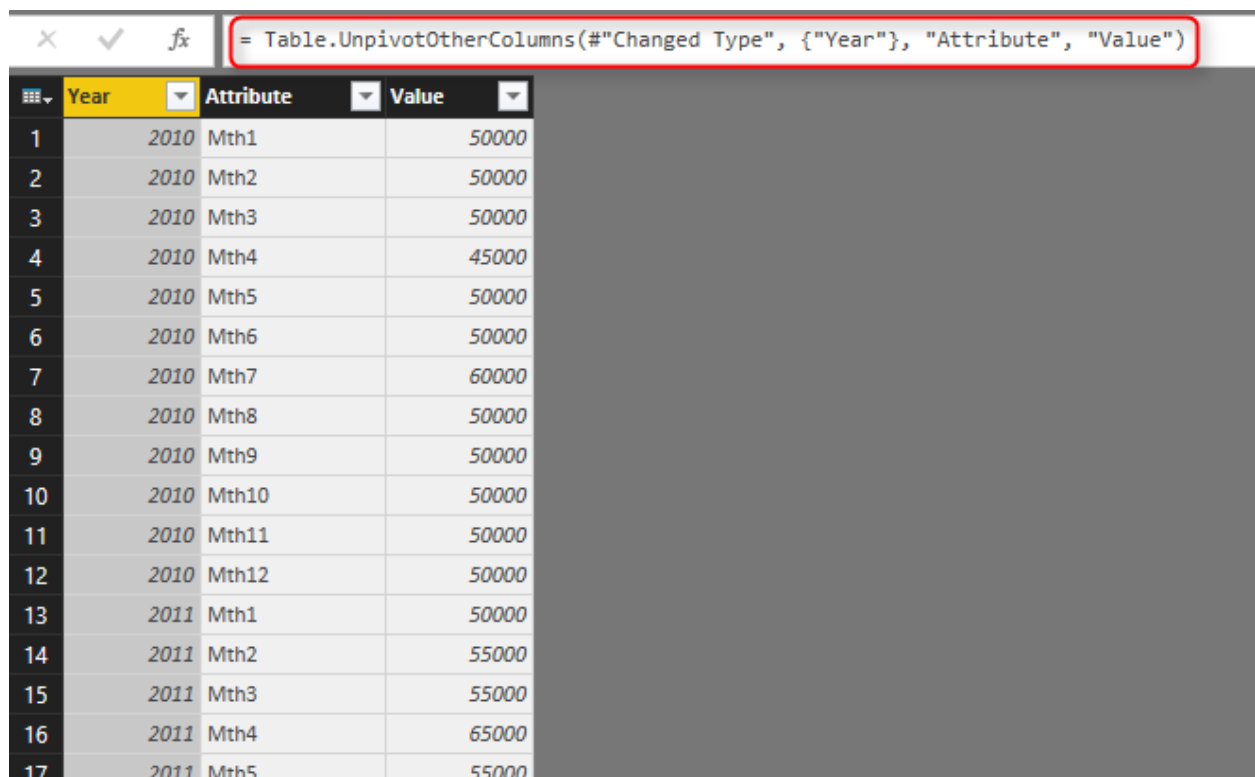
Format Extract Parse

Text Column

Translate all but the currently selected columns into attribute-value pairs, preserving the values of the selected columns.

	Year	Mth1	Mth2	Mth3	Mth4	Mth5
1	2010	50000	50000	50000	45000	50000
2	2011	50000	55000	55000	65000	55000
3	2012	65000	65000	65000	80000	65000
4	2013	70000	70000	70000	70000	65000

and then unpivoted result set would be as below:



= Table.UnpivotOtherColumns("#Changed Type", {"Year"}, "Attribute", "Value")

	Year	Attribute	Value
1	2010	Mth1	50000
2	2010	Mth2	50000
3	2010	Mth3	50000
4	2010	Mth4	45000
5	2010	Mth5	50000
6	2010	Mth6	50000
7	2010	Mth7	60000
8	2010	Mth8	50000
9	2010	Mth9	50000
10	2010	Mth10	50000
11	2010	Mth11	50000
12	2010	Mth12	50000
13	2011	Mth1	50000
14	2011	Mth2	55000
15	2011	Mth3	55000
16	2011	Mth4	65000
17	2011	Mth5	55000

As you see columns and their values are now converted to rows split in only two columns: attribute, and value.

If you get a repetitive column in the source data like below;

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Year	Mth1	Mth2	Mth3	Mth4	Mth5	Mth6	Mth7	Mth8	Mth8	Mth9	Mth10	Mth11	Mth12
2010	50000	50000	50000	45000	50000	50000	60000	50000	50000	50000	50000	50000	50000
2010	50000	50000	50000	55000	50000	50000	60000	50000	50000	50000	50000	50000	50000
2011	50000	55000	55000	65000	55000	55000	55000	55000	55000	55000	55000	55000	55000
2012	65000	65000	65000	65000	80000	65000	65000	65000	65000	65000	65000	65000	65000
2013	70000	70000	70000	70000	65000	70000	70000	70000	70000	80000	70000	70000	70000

Then you would get that repeated in the attribute field after unpivot;

	Year	Attribute	Value
1	2010	Mth1	50000
2	2010	Mth2	50000
3	2010	Mth3	50000
4	2010	Mth4	45000
5	2010	Mth5	50000
6	2010	Mth6	50000
7	2010	Mth7	60000
8	2010	Mth8	50000
9	2010	Mth8_1	50000
10	2010	Mth9	50000
11	2010	Mth10	50000
12	2010	Mth11	50000
13	2010	Mth12	50000

So the best way to handle that is to identify the repetitive column before applying unpivot. You can do these types of checking with Power Query scripts and other functions, If you want to read more about Power Query read it from [Power BI online book](#).

Grouping in Power Query; Getting The Last Item in Each Group

Published Date: August 22, 2016

	CustomerKey	Order Count	Total Revenue	Order Details	First SalesAmount	Last SalesAmount
1	21768	2	4118.26	Table	3578.27	539.99
2	28389	1	3399.99	Table	3399.99	3399.99
3	25863	5	4631.11	Table	3399.99	2.29
4	14501	2	2994.0882	Table	699.0982	2294.99
5	11003	9	8139.29	Table	3399.99	2.29
6	27645	5	6051.31	Table	3578.27	54.99
7	16624	4	5938.25	Table	3578.27	35
8	11005	6	8121.33	Table	3374.99	2384.07
9	11011	4	8133.04	Table	3399.99	53.99
10	27621	3	5971.33	Table	3578.27	8.99
11	27616	5	5998.61	Table	3578.27	2.29

pDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	RevisionNumber	OrderQuantity	UnitPrice	Ex
20050708	25863	1	100		1 SO43699		1	1	3399.99	
20080203	25863	1	100		1 SO62866		1	1	1214.85	
20080203	25863	1	100		1 SO62866		2	1	4.99	
20080203	25863	1	100		1 SO62866		3	1	8.99	
20080203	25863	2	100		1 SO62866		4	1	2.29	

Power BI or Power Query in Excel (or Get Data and Transform as the new name of it) can do many data transformations. One of these transformations is grouping rows by number of fields. If you use the graphical user interface in Power Query for Power BI or Excel you have number of options to get some aggregated results such as count of rows, maximum or minimum, average of each group, or sum... But there are still heaps of operations that you cannot do through GUI, such as getting the last item in each group, or first item. Fortunately with M (Power Query Formula Language) you can apply any of these operations you want. In this post I'll show you how to get benefit of both; start with GUI to write the Group by command for you, and then customize it in M script to achieve what you want. If you like to learn more about Power BI read [Power BI online book; from Rookie to Rock Star](#). If you like to learn more about Power Query, start with [Introduction to Power Query](#).

Learning Objectives for this section

By completing example of this post you will learn;

- How to perform Group By in Power Query/Power BI
- How to leverage pre-defined aggregation functions in Group By GUI window
- How to extend grouping possibilities with changes in Power Query script

Prerequisite

For this example I will be using AdventureWorksDW sample Microsoft SQL Server database. You can download it from [here](#).

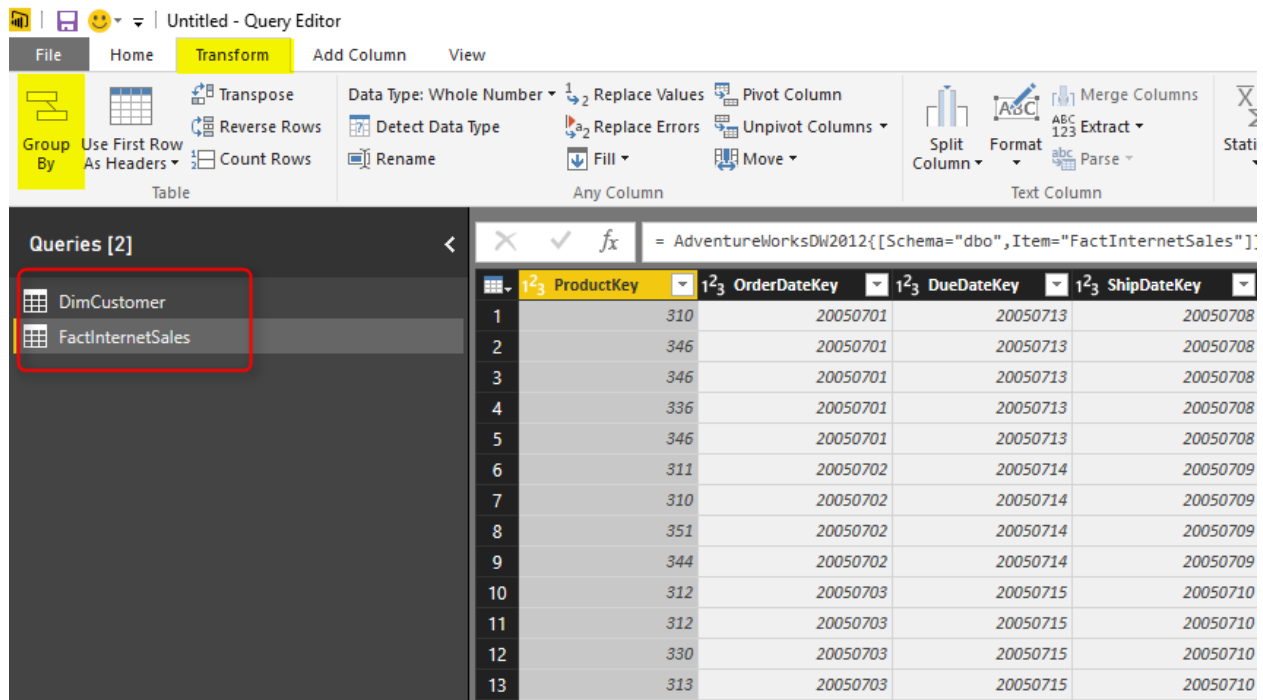
Scenario

Scenario that I want to solve as an example is this:

FactInternetSales has sales transaction information for each customer, by each product, each order date and some other information. We want to have a grouped table by customer, which has the number of sales transaction by each customer, total sales amount for that customer, the first and the last sales amount for that customer. First and last defined by the first and last order date for the transaction.

Get Data

Let's start by getting data from SQL Server, Choose AdventureWorksDW as the source database, and select DimCustomer and FactInternetSales as the only tables for this example. Click on Edit to move into Power Query window.

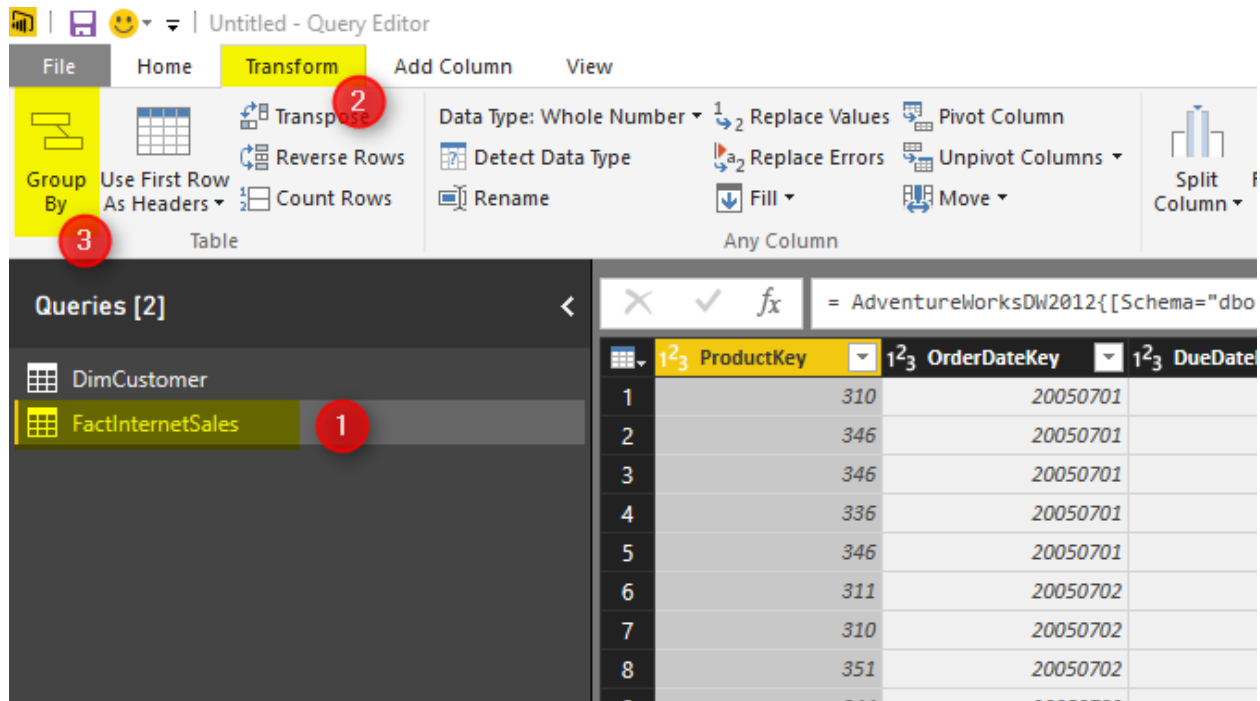


The screenshot shows the Power Query Editor interface. On the left, the 'Queries [2]' pane lists 'DimCustomer' and 'FactInternetSales', both of which are highlighted with a red box. The main area displays a table with the following columns: ProductKey, OrderDateKey, DueDateKey, and ShipDateKey. The table contains 13 rows of data.

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey
1	310	20050701	20050713	20050708
2	346	20050701	20050713	20050708
3	346	20050701	20050713	20050708
4	336	20050701	20050713	20050708
5	346	20050701	20050713	20050708
6	311	20050702	20050714	20050709
7	310	20050702	20050714	20050709
8	351	20050702	20050714	20050709
9	344	20050702	20050714	20050709
10	312	20050703	20050715	20050710
11	312	20050703	20050715	20050710
12	330	20050703	20050715	20050710
13	313	20050703	20050715	20050710

Group By Transformation

FactInternetSales table is the one we want to apply all transformations in. So Click on FactInternetSales first, then from Transform Tab, select Group By option as the first menu option.



This will open the Group By dialog window with some configuration options

Group By

Specify the columns to group by.

Group by

ProductKey

Add grouping

New column name

Count

Operation

Count Rows

Column

Add aggregation

OK

Cancel

By default Group By happens on the selected columns. because usually the first column is the selected column (in our table ProductKey), then the column mentioned under group by section is also ProductKey. You can change this to another column and add or remove columns in this section.

Choose the Group By Field

Based on your final grain of the output table the group by field will be defined. In this example we want the final table to have one record per Customer, so CustomerKey (which is the identifier for each customer) should be our Group By Column.

Group By

Specify the columns to group by.

Group by

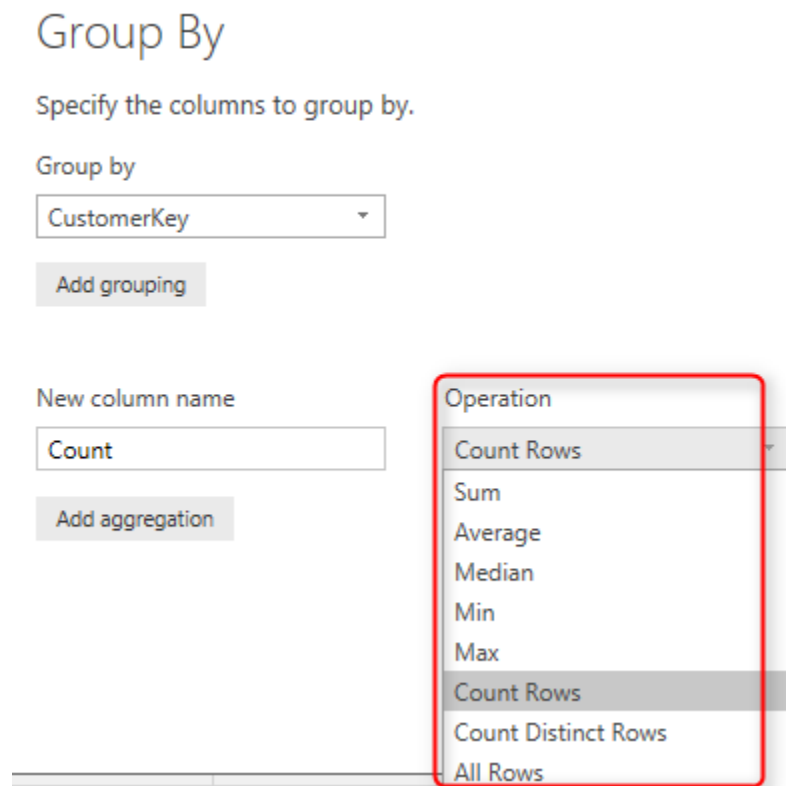
CustomerKey

Add grouping

Note that you can add as many fields as you want in the Group By section. the result would be one record per combination of unique values from all these fields.

Add Aggregation Fields

Group by result would be one record per each unique combination of all fields set in the "group by" section. In addition you can also have some aggregated columns. Here are list of operations you can have by default:



The screenshot shows the 'Group By' configuration in Power BI. It includes a 'Group by' dropdown menu with 'CustomerKey' selected, an 'Add grouping' button, a 'New column name' input field with 'Count', and an 'Add aggregation' button. A red box highlights the 'Operation' dropdown menu, which lists the following aggregation options: Count Rows, Sum, Average, Median, Min, Max, Count Rows (highlighted), Count Distinct Rows, and All Rows.

Most of items above are self explanatory. For example; when you want to count number of sales transaction. you can use Count Rows. If you want total Sales amount for each group you can choose Sum, and then in the Column section choose the column as SalesAmount. All Rows will generate a sub table in each element of the aggregated table that contains all rows in that group.

Columns that I want to create in this section are:

Order Count (Count Rows), Total Revenue (Sum of Sales Amount), Order Details (All Rows)

Group By

Specify the columns to group by.

Group by

CustomerKey ▼

Add grouping

New column name	Operation	Column
Order Count	Count Rows ▼	▼
Total Revenue	Sum ▼	SalesAmount ▼
Order Details	All Rows ▼	▼

Add aggregation

Adding aggregated columns is as easy as that. Now If you click on OK, you will see the result;

	CustomerKey	Order Count	Total Revenue	Order Details
1	21768	2	4118.26	Table
2	28389	1	3399.99	Table
3	25863	5	4631.11	Table
4	14501	2	2994.0882	Table
5	11003	9	8139.29	Table
6	27645	5	6051.31	Table
7	16624	4	5938.25	Table
8	11005	6	8121.33	Table
9	11011	4	8133.04	Table
10	27621	3	5971.33	Table
11	27616	5	5998.61	Table
12	20042	3	3150.3982	Table
13	16351	2	5873.26	Table
14	16517	2	4698.76	Table
15	27606	3	5997.33	Table
16	13513	4	5918.24	Table
17	27601	1	3578.27	Table
18	13591	9	8355.27	Table
19	16483	2	4118.26	Table

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber
346	20050701	20050713	20050708	11003	1	6	9	SO43701
361	20070709	20070721	20070716	11003	1	6	9	SO51315
478	20070709	20070721	20070716	11003	1	6	9	SO51315
477	20070709	20070721	20070716	11003	1	6	9	SO51315
225	20070709	20070721	20070716	11003	1	6	9	SO51315
564	20071111	20071123	20071118	11003	1	6	9	SO57783
541	20071111	20071123	20071118	11003	1	6	9	SO57783
530	20071111	20071123	20071118	11003	1	6	9	SO57783
480	20071111	20071123	20071118	11003	1	6	9	SO57783

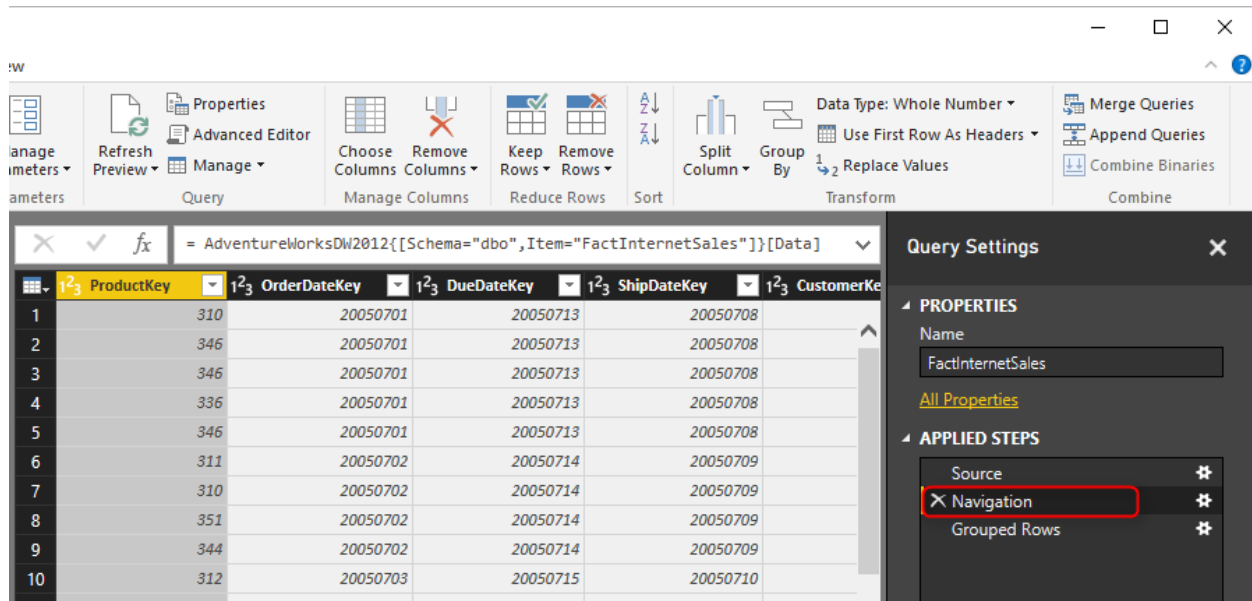
As you can see Order Count and Total Revenue show the aggregated result of each group, and Order Details (if you click not on the “Table” itself, but on a blank area on that cell) contains the actual rows in each group. This detailed view can be used for many other calculations or transformations later on. In many cases you will find the All rows option useful.

First and Last Item in each Group

Getting some default aggregation was as easy as selecting them in Group By window. However not all types of operations are listed there. For example in detailed table above you can see that customer 11003 had 9 sales transaction,, and they happened in different Order dates, getting the first and last order date is easy with Max and Min operations. However getting the sales amount or product key associated with that record, or in the other words getting the first and last item in each group isn’t possible through GUI. Fortunately we can use M (Power Query formula language) to achieve this easily.

Sort By Date

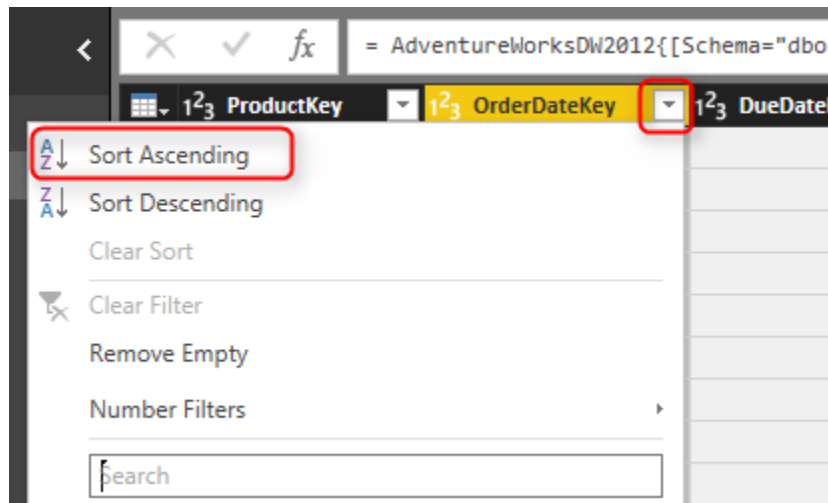
To get the first or last item in each group I have to order the table based on that date column. Sorting is possible simply through GUI. and I have to apply that to the step before group by operation. So from the right hand side applied steps list I'll select Navigation (which is the step before Grouped Rows);



The screenshot shows the Power BI Query Editor interface. The main area displays a table with columns: ProductKey, OrderDateKey, DueDateKey, ShipDateKey, and CustomerKey. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, where 'Navigation' is highlighted with a red box, indicating it is the step before 'Grouped Rows'.

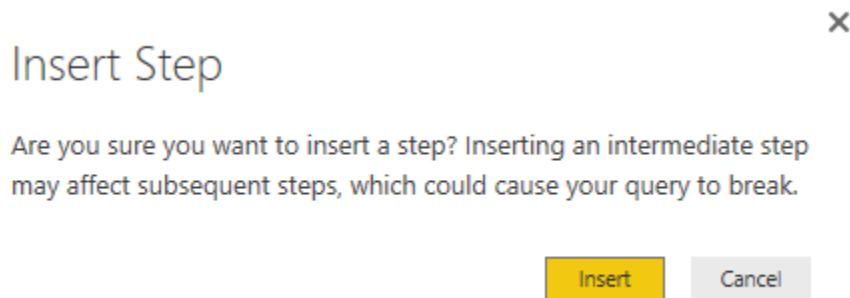
	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey
1	310	20050701	20050713	20050708	
2	346	20050701	20050713	20050708	
3	346	20050701	20050713	20050708	
4	336	20050701	20050713	20050708	
5	346	20050701	20050713	20050708	
6	311	20050702	20050714	20050709	
7	310	20050702	20050714	20050709	
8	351	20050702	20050714	20050709	
9	344	20050702	20050714	20050709	
10	312	20050703	20050715	20050710	

Now in this view you can order simply by clicking on OrderDateKey and Choose Sort Ascending.

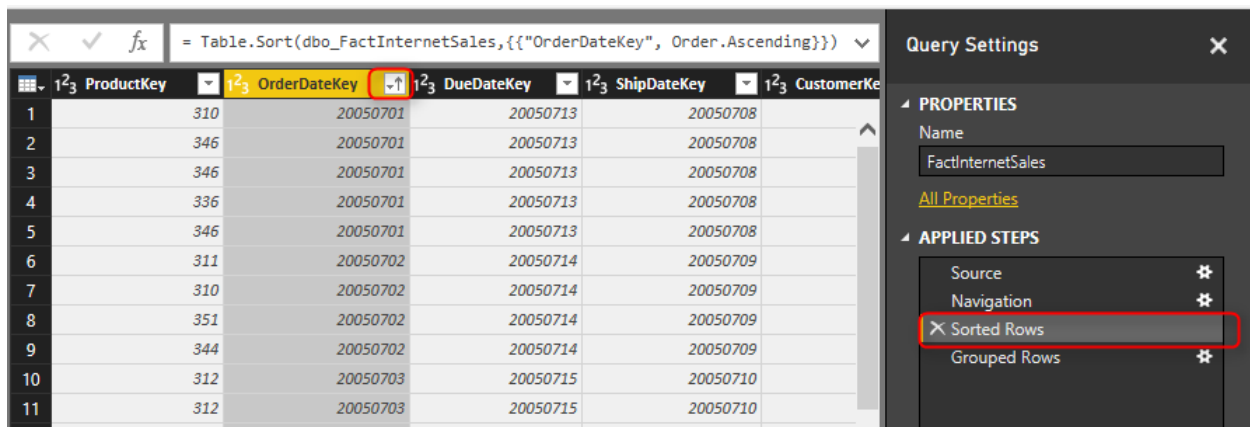


The screenshot shows the 'OrderDateKey' column selected in the table. A dropdown menu is open, showing 'Sort Ascending' as the selected option, highlighted with a red box. Other options include 'Sort Descending', 'Clear Sort', 'Clear Filter', 'Remove Empty', and 'Number Filters'.

This will create another step, and asks you do you want to INSERT this step here?



Click on Insert to confirm you want to insert it here before the Grouped Rows. and then you will see a Sorted Rows step added before Grouped Rows. This means Grouped Rows will use the output of Sorted Rows step as the input for grouping (which is exactly what we want).



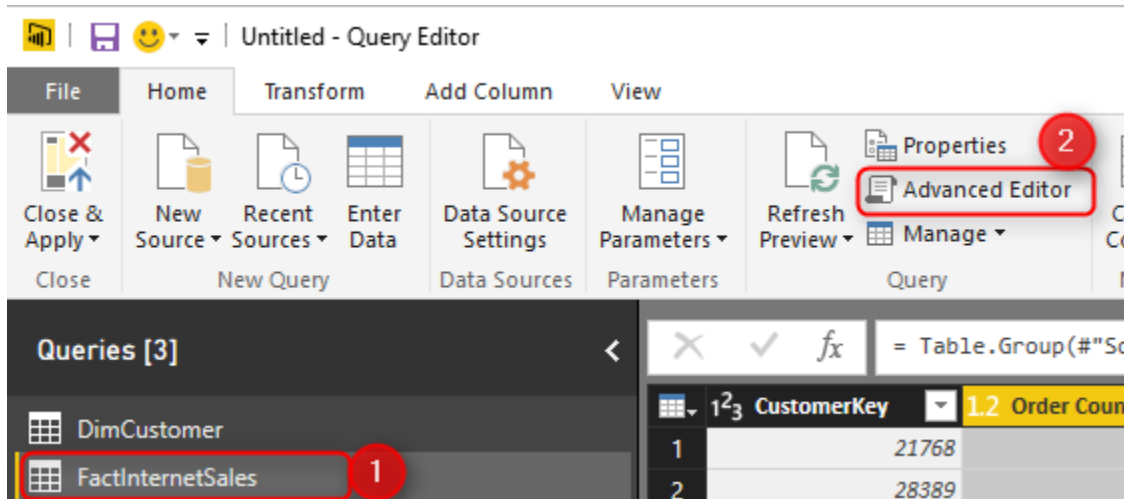
Now you can go to the Grouped Rows step to see the result hasn't changed but the sub tables are sorted now. All we need from here is to get the first and last item in the sub table.

** If you want to sort based on multiple column simply go to the Advanced Editor and add as many as sections you want to Table.Sort input parameters.

List.First and List.Last

Fortunately Power Query has a bunch of operations on List that we can use. List.First will return the first item in the list (based on the order defined in the list), and List.Last will return the last item. So let's use them in the Group By operation to fetch first and last sales amount.

To make changes here you need to go to script editor in Power Query which can be achieved via Advanced Editor option in Home tab. You have to make sure that you are in the FactInternetSales Query first.



Advanced Editor will show you M script that build the output and the group by command as well.

```
let
    Source = Sql.Databases("."),
    AdventureWorksDW2012 = Source{[Name="AdventureWorksDW2012"]}[Data],
    dbo_FactInternetSales = AdventureWorksDW2012{[Schema="dbo",Item="FactInternetSales"]}[Data],
    #"Sorted Rows" = Table.Sort(dbo_FactInternetSales,{"OrderDateKey", Order.Ascending}),
    #"Grouped Rows" = Table.Group(#"Sorted Rows", {"CustomerKey"}, {"Order Count", each
    Table.RowCount(_), type number}, {"Total Revenue", each List.Sum([SalesAmount]), type number}, {"Order
    Details", each _ type table}))
in
    #"Grouped Rows"
```

The script in above code section created automatically when you did transformations through GUI. The line with Table.Group is the line that does all the grouping and aggregation. It is a long line, so let me format it better for easier understanding;

```
let
    Source = Sql.Databases("."),
    AdventureWorksDW2012 = Source{[Name="AdventureWorksDW2012"]}[Data],
    dbo_FactInternetSales = AdventureWorksDW2012{[Schema="dbo",Item="FactInternetSales"]}[Data],
    #"Sorted Rows" = Table.Sort(dbo_FactInternetSales,{"OrderDateKey", Order.Ascending}),
    #"Grouped Rows" = Table.Group(#"Sorted Rows",
        {"CustomerKey"},
        {
            {"Order Count", each Table.RowCount(_), type number},
            {"Total Revenue", each List.Sum([SalesAmount]), type number},
        }
    )
```

```

        {"Order Details", each _ type table}
    }
)
in
#"Grouped Rows"

```

Script below is the same script. I just put some enters and tabs to format it better for reading. The above section shows Table.Group section of the script. As you can see Table.Group gets a table as input, which is the #"Sorted Rows" table from the previous step. The group by field is "CustomerKey". and then a set of aggregated columns one after each other (which is highlighted in code above). Each column has name of the column, type of transformation (or aggregation), and the data type of the column. for example:

```

let
    Source = Sql.Databases("."),
    AdventureWorksDW2012 = Source{[Name="AdventureWorksDW2012"]}[Data],
    dbo_FactInternetSales = AdventureWorksDW2012{[Schema="dbo",Item="FactInternetSales"]}[Data],
    #"Sorted Rows" = Table.Sort(dbo_FactInternetSales,{"OrderDateKey", Order.Ascending}),
    #"Grouped Rows" = Table.Group(#"Sorted Rows",
        {"CustomerKey"},
        {
            {"Order Count", each Table.RowCount(_), type number},
            {"Total Revenue", each List.Sum([SalesAmount]), type number},
            {"Order Details", each _ type table}
        }
    )
in
    #"Grouped Rows"

```

Total Revenue is the name of column. calculation for this column is Sum of [SalesAmount] which is one of the fields in the table, and the output is of type number.

So by now you should thinking of how each is to create a new aggregated column here; by adding similar column in the script. I add the new column after Order Details column, so I need an extra comma (,) after that line, and the new lines would be;

```

let
    Source = Sql.Databases("."),
    AdventureWorksDW2012 = Source{[Name="AdventureWorksDW2012"]}[Data],
    dbo_FactInternetSales = AdventureWorksDW2012{[Schema="dbo",Item="FactInternetSales"]}[Data],
    #"Sorted Rows" = Table.Sort(dbo_FactInternetSales,{"OrderDateKey", Order.Ascending}),
    #"Grouped Rows" = Table.Group(#"Sorted Rows",
        {"CustomerKey"},
        {
            {"Order Count", each Table.RowCount(_), type number},

```

```

        {"Total Revenue", each List.Sum([SalesAmount]), type number},
        {"Order Details", each _, type table},
        {"First SalesAmount", each List.First([SalesAmount]), type number},
        {"Last SalesAmount", each List.Last([SalesAmount]), type number}
    }
)
in
#"Grouped Rows"

```

Marked lines above uses List.First and List.Last on the exact same structure that List.Sum worked. because we have already sorted the table based on OrderDate so the first item would be the first sales transaction, and the last item would be the last.

Here is the output of this change:

	1.2 CustomerKey	1.2 Order Count	1.2 Total Revenue	Order Details	1.2 First SalesAmount	1.2 Last SalesAmount
1	21768	2	4118.26	Table	3578.27	539.99
2	28389	1	3399.99	Table	3399.99	3399.99
3	25863	5	4631.11	Table	3399.99	2.29
4	14501	2	2994.0882	Table	699.0982	2294.99
5	11003	9	8139.29	Table	3399.99	2.29
6	27645	5	6051.31	Table	3578.27	54.99
7	16624	4	5938.25	Table	3578.27	35
8	11005	6	8121.33	Table	3374.99	2384.07
9	11011	4	8133.04	Table	3399.99	53.99
10	27621	3	5971.33	Table	3578.27	8.99
11	27616	5	5998.61	Table	3578.27	2.29

OrderDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	RevisionNumber	OrderQuantity	UnitPrice	Ex
20050708	25863	1	100		1 SO43699		1	1	3399.99	
20080203	25863	1	100		1 SO62866		1	1	1214.85	
20080203	25863	1	100		1 SO62866		2	1	4.99	
20080203	25863	1	100		1 SO62866		3	1	8.99	
20080203	25863	2	100		1 SO62866		4	1	2.29	

You can see that the first and the last SalesAmount picked correctly from each group as two new columns.

** Note that with adding some changes in script editor that are not supported through GUI, you will loose the GUI configuration window section. As a result of the change here you cannot change Grouping configuration in GUI anymore, if you want to change it, you have to go to Advanced Editor for this section. So if you are a GUI fan, better to apply all required configuration first, and then add extra logic in the code.

Append vs. Merge in Power BI and Power Query

Published Date: January 5, 2017

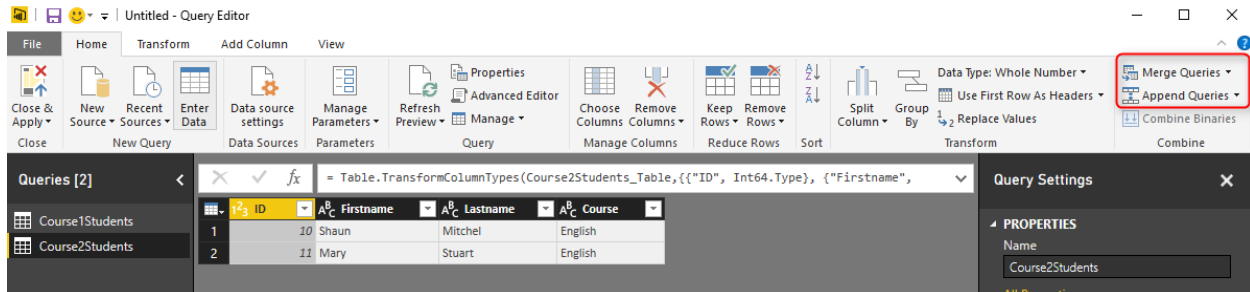


Combining two queries in Power Query or in Power BI is one of the most basic and also essential tasks that you would need to do in most of data preparation scenarios. There are two types of combining queries; Merge, and Append. Database developers easily understand the difference, but majority of Power BI users are not developers. In this post I'll explain the difference between Merge and Append, and situations that you should use each. If you want to learn more about Power BI, read [Power BI online book, from Rookie to Rock Star](#).

Why Combine Queries?

This might be the first question comes into your mind; Why should I combine queries? The answer is that; You can do most of the things you want in a single query, however it will be very complicated with hundreds of steps very quickly. On the other hand your queries might be used in different places. For example one of them might be used as a table in Power BI model, and also playing part of data preparation for another query. Combining queries is a big help in writing better and simpler queries. I'll show you some examples of combining queries.

Result of a combine operation on one or more queries will be only one query. You can find Append or Merge in the Combine Queries section of the Query Editor in Power BI or in Excel.



Append

Append means results of two (or more) queries (which are tables themselves) will be combined into one query in this way:

- Rows will be appended after each other. (for example appending a query with 50 rows with another query with 100 rows, will return a result set of 150 rows)
- Columns will be same number of columns for each query*. (for example col1, col2,..., col10 in first query, after appending with same columns in the second query will result into one query with single set of col1,col2, ..., col10)

There is an exception for number of columns which I'll talk about it later. Let's first look at what Append looks like in action;

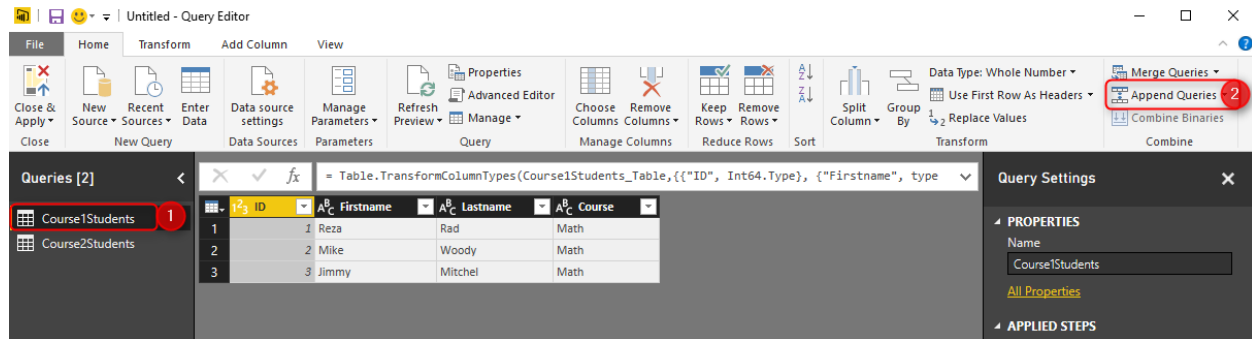
Consider two sample data sets; one for students of each course, Students of course 1:

ID	Firstname	Lastname	Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math

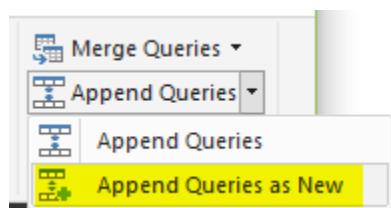
and Students of course 2:

ID	Firstname	Lastname	Course
10	Shaun	Mitchel	English
11	Mary	Stuart	English

To append these queries, Click on one of them and select Append Queries from the Combine section of Home tab in Query Editor



If you want to keep the existing query result as it is, and create a new query with the appended result choose Append Queries as New, otherwise just select Append Queries. In this example I'll do Append Queries as New, because I want to keep existing queries intact.



You can choose what is the primary table (normally this is the query that you have selected before clicking on Append Queries), and the table to append.

Append

☒ Two tables ☐ Three or more tables

Primary table

Course1Students

Table to append to the primary table

Course2Students

You can also choose to append Three or more tables and add tables to the list as you wish. For this example I have only two tables, so I'll continue with above configuration. Append Queries simply append rows after each other, and because column names are exactly similar in both queries, the result set will have same columns.

Queries [3] < ✕ ✓ fx = Table.Combine({Course1Students, Course2Students})

1 ² ₃ ID	A ^B _C Firstname	A ^B _C Lastname	A ^B _C Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math
4	10 Shaun	Mitchel	English
5	11 Mary	Stuart	English

Result of Append as simple as that

1 ² ₃ ID	A ^B _C Firstname	A ^B _C Lastname	A ^B _C Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math

1 ² ₃ ID	A ^B _C Firstname	A ^B _C Lastname	A ^B _C Course
10	Shaun	Mitchel	English
11	Mary	Stuart	English

1 ² ₃ ID	A ^B _C Firstname	A ^B _C Lastname	A ^B _C Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math
10	Shaun	Mitchel	English
11	Mary	Stuart	English

Append is similar to UNION ALL in T-SQL.

What about Duplicates?

Append Queries will NOT remove duplicates. You have to use Group By or Remove Duplicate Rows to get rid of duplicates.

What if columns in source queries are not exactly match?

Append requires columns to be exactly similar to work in best condition. if columns in source queries are different, append still works, but will create one column in the output per each new column, if one of the sources doesn't have that column the cell value of that column for those rows will be null.

Merge

Merge is another type of combining queries which is based on matching rows, rather than columns. The output of Merge will be single query with;

1. There should be joining or matching criteria between two queries. (for example StudentID column of both queries to be matched with each other)
2. Number of rows will be dependent on matching criteria between queries
3. Number of Columns will be dependent of what columns selected in the result set. (Merge will create a structured column as a result).

Understanding how Merge works might look a bit more complicated, but it will be very easy with example, let's have a look at that in action;

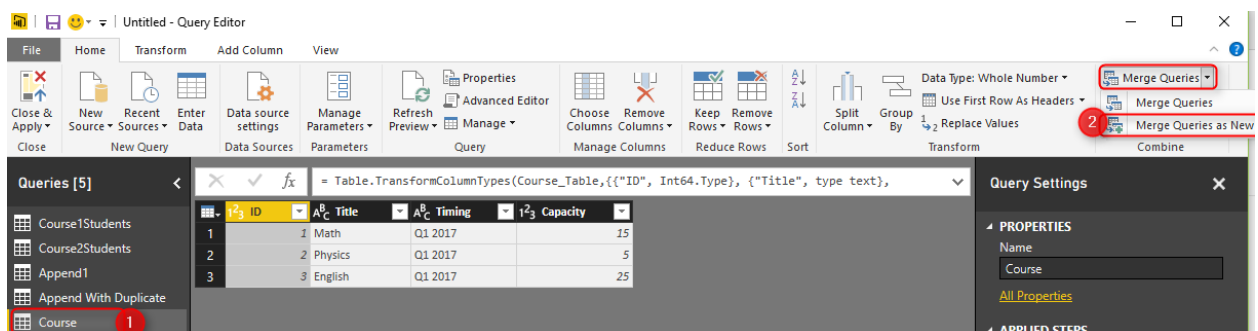
In addition to tables in first example, consider that there is another table for Course's details as below:

1 ² 3 ID	A ^B _C Title	A ^B _C Timing	1 ² 3 Capacity
1	Math	Q1 2017	15
2	Physics	Q1 2017	5
3	English	Q1 2017	25

Now if I want to combine Course query with the Appended result of CourseXStudents to see which students are part of which course with all details in each row, I need to use Merge Queries. Here is the appended result again;

1 ² 3 ID	A ^B _C Firstname	A ^B _C Lastname	A ^B _C Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math
10	Shaun	Mitchel	English
11	Mary	Stuart	English

Select Course Query first, and then Select Merge Queries (as New)



The screenshot shows the Power BI Query Editor interface. The ribbon at the top includes 'File', 'Home', 'Transform', 'Add Column', and 'View'. The 'Merge Queries' button is highlighted in the 'Merge' group of the 'Transform' tab. A red circle and arrow point to the 'Merge Queries as New' option in the dropdown menu. On the left, the 'Queries' pane shows a list of queries: 'Course1Students', 'Course2Students', 'Append1', 'Append With Duplicate', and 'Course'. The 'Course' query is selected, indicated by a red circle and arrow. The main area displays the 'Course' query's data table, which matches the one shown in the previous example.

Merging Queries require joining criteria. Joining criteria is field(s) in each source query that should be matched with each other to build the result query. In this example, I want to Merge Course query with Append1, based on Title of the course.

Merge

Select tables and matching columns to create a merged table.

Course 1

ID	Title 3	Timing	Capacity
1	Math	Q1 2017	15
2	Physics	Q1 2017	5
3	English	Q1 2017	25

Append1 2

ID	Firstname	Lastname	Course 4
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math
10	Shaun	Mitchel	English
11	Mary	Stuart	English

Join Kind

Left Outer (all from first, matching from second)

i The selection has matched 2 out of the first 3 rows.

I'll talk about types of join later. For now continue the selection, and you will see these two queries match with each other based on Course title, result query will be same as the first query (Course in this example), plus one additional column named as NewColumn with a table in each cell. This is a structured column which can be expanded into underlying tables. if you click on an empty area of the cell containing one of these tables, you will see the sub table underneath.

ID	Title	Timing	Capacity	NewColumn
1	Math	Q1 2017	15	Table
2	Physics	Q1 2017	5	Table
3	English	Q1 2017	25	Table

ID	Firstname	Lastname	Course
1	Reza	Rad	Math
2	Mike	Woody	Math
3	Jimmy	Mitchel	Math

Now click on Expand column icon, and expand the New Column to all underneath table structure

ID	Title	Timing	Capacity	NewColumn
1	Math			
2	Physics			
3	English			

Expand

Aggregate

☒ (Select All Columns)
 ☒ ID
 ☒ Firstname
 ☒ Lastname
 ☒ Course

☒ Use original column name as prefix

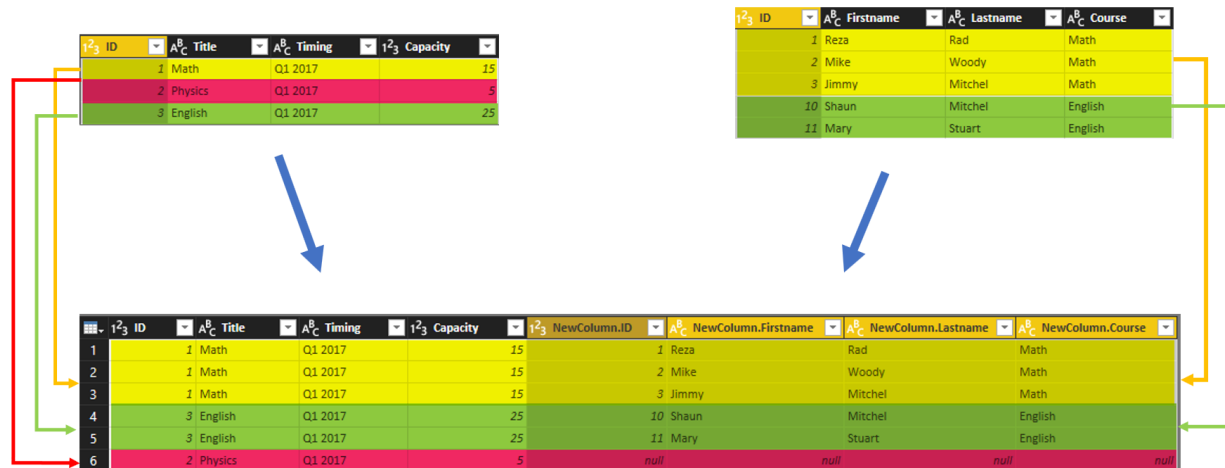
OK

Cancel

Result will be a table including columns from both tables, and rows matching with each other.

ID	Title	Timing	Capacity	NewColumn.ID	NewColumn.Firstname	NewColumn.Lastname	NewColumn.Course
1	Math	Q1 2017	15	1	Reza	Rad	Math
2	Math	Q1 2017	15	2	Mike	Woody	Math
3	Math	Q1 2017	15	3	Jimmy	Mitchel	Math
4	English	Q1 2017	25	10	Shaun	Mitchel	English
5	English	Q1 2017	25	11	Mary	Stuart	English
6	Physics	Q1 2017	5		null	null	null

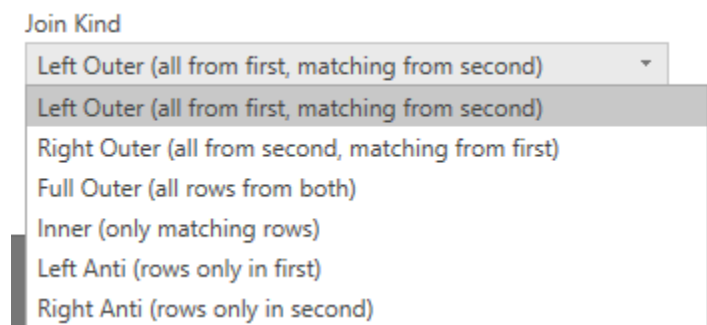
Columns in the left hand side are coming from Course table, columns in the right hand side are coming from Students table. Values in the rows only appears in matching criteria. First three rows are students of Math course, then two students for English course, and because there is no student for Physics course you will see null values for students columns.



Merge is similar to JOIN in T-SQL

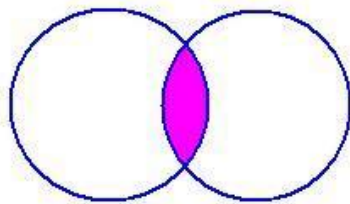
Join Types

There are 6 types of joins supported in Power BI as below, depends on the effect on the result set based on matching rows, each of these types works differently.

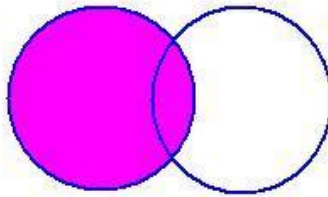


Explaining what each join type will do is totally different post which I will write later. For now this picture explains it very well:

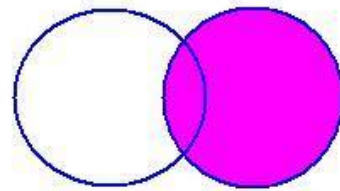
JOINS AND SET OPERATIONS IN RELATIONAL DATABASES



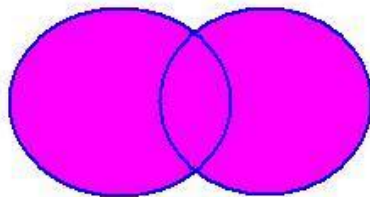
Inner join (result similar to Intersect)



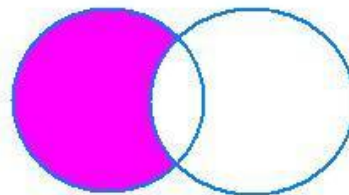
Left outer join



Right outer join



Full outer join



Minus

Picture referenced from: http://www.udel.edu/evelyn/SQL-Class2/SQLclass2_Join.html

Fetch Files and/or Folders with Filtering and Masking: Power Query

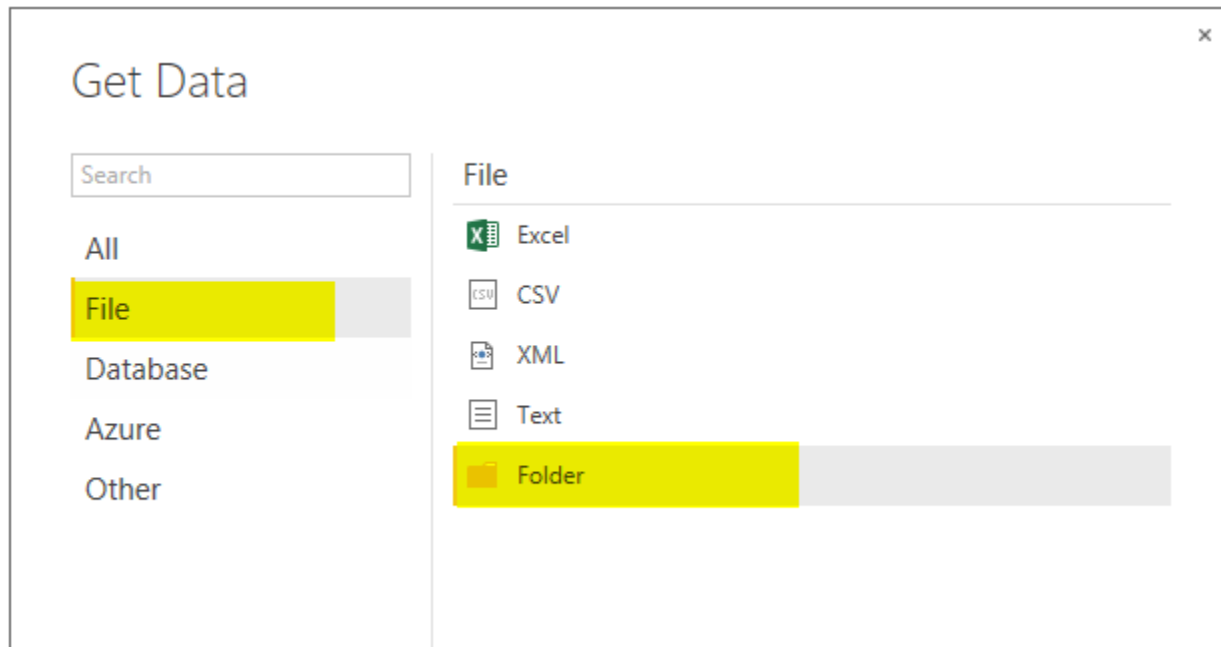
Published Date: August 3, 2015

	Content	Name	Extension	Date accessed	Date modified	Date
1	Table	Auckland CodeCamp 2014		12/13/2014 10:14:54 AM	9/13/2014 2:03:17 PM	9
2	Table	Auckland SQL User Group - 16 May 2012		12/13/2014 10:09:50 AM	1/1/2014 10:09:10 AM	
3	Table	CloudnEnterprise_Symbols_Public_v2.02		2/1/2015 12:43:29 PM	2/1/2015 12:43:29 PM	
4	Binary	CloudnEnterprise_Symbols_Public_v2.02.zip	.zip	2/1/2015 12:43:27 PM	2/1/2015 12:43:10 PM	
5	Binary	DQS.txt	.txt	12/13/2014 10:09:50 AM	11/19/2012 9:23:50 PM	
6	Table	DWBI-VC- Aug 2012		12/13/2014 10:09:51 AM	1/1/2014 9:40:14 AM	
7	Table	Error Handling		12/13/2014 10:04:44 AM	1/1/2014 9:30:34 AM	
8	Table	INF		12/13/2014 10:09:50 AM	1/1/2014 9:37:06 AM	

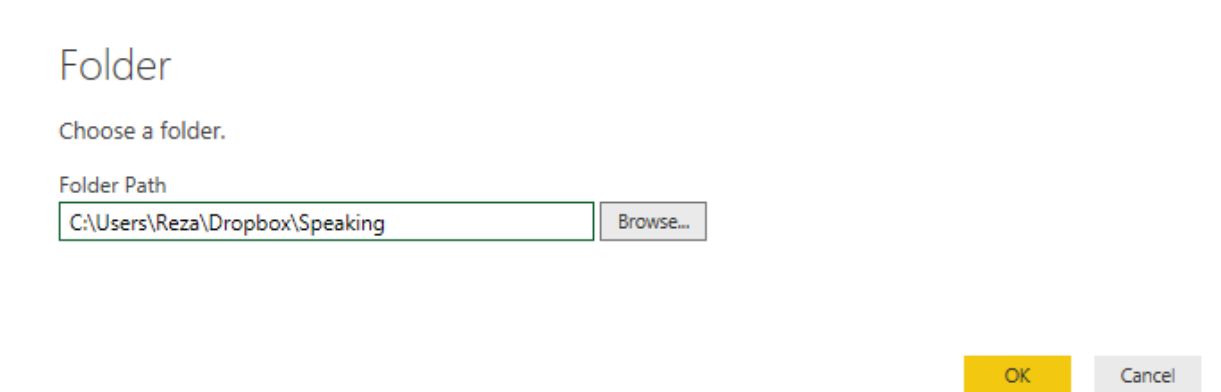
Fetching list of Files in a folder is easy with Power Query, it is one of the built-in source types. However fetching list of folders is not a built-in function. In this post I'll show you how to fetch list of files, and also fetch only list of folders. The method explained in a way that you can customize the code and apply any conditions as you want later on. Conditions such as File or folder name masking to fetch only names that contains special character strings.

Fetch All Files in a Folder

For fetching all files in a folder you can simply use the GUI Get Data, and under File, choose Folder



Browse for Folder



and then simply you will see list of all files.

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
1	Binary	Thumbs.db	7/30/2015 10:32:42 PM	7/30/2015 10:32:42 PM	7/30/2015 10:32:42 PM	Record	C:\Users\Reza\Dropbox\Speaking\CloudEnterprise_Symbols_Public_v2.C
2	Binary	Table Functions.txt	7/27/2015 11:14:38 PM	3/6/2014 10:11:47 AM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
3	Binary	Table Functions Part 1.xlsx	7/27/2015 11:14:38 PM	2/12/2014 1:14:49 AM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
4	Binary	Movies.xlsx	7/27/2015 11:14:38 PM	3/5/2014 9:33:43 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
5	Binary	M.xlsx	7/27/2015 11:14:38 PM	2/10/2014 11:46:30 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
6	Binary	M sample.txt	7/27/2015 11:14:38 PM	2/10/2014 11:46:41 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
7	Binary	Generator.txt	7/27/2015 11:14:38 PM	3/10/2014 2:57:43 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
8	Binary	Error Handling.xlsx	7/27/2015 11:14:38 PM	3/6/2014 1:35:56 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
9	Binary	Error Handling.txt	7/27/2015 11:14:38 PM	3/10/2014 1:38:06 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
10	Binary	Date Dimension.xlsx	7/27/2015 11:14:38 PM	2/15/2014 12:43:03 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
11	Binary	Date Dimension.txt	7/27/2015 11:14:38 PM	3/6/2014 11:05:58 AM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
12	Binary	Date Dimension - With Public Holidays.xlsx	7/27/2015 11:14:38 PM	2/16/2014 2:29:58 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
13	Binary	D2.xlsx	7/27/2015 11:14:38 PM	11/6/2014 3:07:44 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
14	Binary	Custom Function.xlsx	7/27/2015 11:14:38 PM	2/13/2014 4:28:11 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
15	Binary	Custom Function.txt	7/27/2015 11:14:38 PM	3/6/2014 10:42:41 AM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\Demo
16	Binary	Thumbs.db	7/27/2015 11:14:35 PM	5/28/2015 1:29:50 PM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\
17	Binary	Reza Rad_Top 5 Power Query M Functions that You Don't Kn	7/28/2015 12:03:58 AM	7/28/2015 12:03:58 AM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\
18	Binary	Reza Rad_Top 5 Power Query M Functions that You Don't Kn	7/27/2015 11:14:39 PM	11/8/2014 10:33:55 AM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\
19	Binary	PowerQuery Names.xls	7/27/2015 11:14:38 PM	11/8/2014 10:34:27 AM	7/27/2015 11:14:35 PM	Record	C:\Users\Reza\Dropbox\Speaking\NZBIUG July 2015\Power Query\

As you see in the above table extracted, there is a column named Content, which has the content of the file. you can click on that to see content of the file if you want.

What is the M Code Written Behind the Scenes

The code behind the scenes for this transformation used [Folder.Files](#) function

```
let
    Source = Folder.Files("C:\Users\Reza\Dropbox\Speaking")
in
    Source
```

Fetch All Files and Folders

There is another M function to fetch all files and folders listed under a folder, named [Folder.Contents](#). This function returns the Content column with data type of the record and content of it. for Files data type usually is Binary, and for Folders it is table. So it can be easily distinguished and separated. Here is list of all files and folders fetched:

```
let
    Source = Folder.Contents("C:\Users\Reza\Dropbox\Speaking")
in
    Source
```

Result set contains both files and folders

	Content	Name	Extension	Date accessed	Date modified	Date
1	Table	Auckland CodeCamp 2014		12/13/2014 10:14:54 AM	9/13/2014 2:03:17 PM	s
2	Table	Auckland SQL User Group - 16 May 2012		12/13/2014 10:09:50 AM	1/1/2014 10:09:10 AM	
3	Table	CloudnEnterprise_Symbols_Public_v2.02		2/1/2015 12:43:29 PM	2/1/2015 12:43:29 PM	
4	Binary	CloudnEnterprise_Symbols_Public_v2.02.zip	.zip	2/1/2015 12:43:27 PM	2/1/2015 12:43:10 PM	
5	Binary	DQS.txt	.txt	12/13/2014 10:09:50 AM	11/19/2012 9:23:50 PM	
6	Table	DWBI-VC- Aug 2012		12/13/2014 10:09:51 AM	1/1/2014 9:40:14 AM	
7	Table	Error Handling		12/13/2014 10:04:44 AM	1/1/2014 9:30:34 AM	
8	Table	INF		12/13/2014 10:09:50 AM	1/1/2014 9:37:06 AM	

As you see the Content column shows the data type of the value. To check if a record is Folder or not, we have to compare its data type to Table. We can add a column to the table to check data type of the content column in each record. data type can be checked with [Value.Is](#) function.

```
let
    Source = Folder.Contents("C:\Users\Reza\Dropbox\Speaking"),
    TypeAdded=Table.AddColumn(Source,"Type",each Value.Is([Content],type table))
in
    TypeAdded
```

And the result set shows if the record is folder or not

Attributes	Folder Path	Type
6 AM Record	C:\Users\Reza\Dropbox\Speaking\	TRUE
7 AM Record	C:\Users\Reza\Dropbox\Speaking\	TRUE
9 PM Record	C:\Users\Reza\Dropbox\Speaking\	TRUE
7 PM Record	C:\Users\Reza\Dropbox\Speaking\	FALSE
0 AM Record	C:\Users\Reza\Dropbox\Speaking\	FALSE
5 AM Record	C:\Users\Reza\Dropbox\Speaking\	TRUE
5 AM Record	C:\Users\Reza\Dropbox\Speaking\	TRUE
5 AM Record	C:\Users\Reza\Dropbox\Speaking\	TRUE
5 AM Record	C:\Users\Reza\Dropbox\Speaking\	FALSE
6 PM Record	C:\Users\Reza\Dropbox\Speaking\	TRUE

To fetch only folders, we can filter the data set with Table.SelectRows function. I've sorted the result set descending by creating date of the folder. Here is the code:

let

```
Source = Folder.Contents("C:\Users\Reza\Dropbox\Speaking"),
TypeAdded=Table.AddColumn(Source,"Type",each Value.Is([Content],type table)),
Folders=Table.SelectRows(TypeAdded, each [Type]=true),
Sorted=Table.Sort(Folders,{"Date created", Order.Descending})
```

in

Sorted

And the result:

	Content	Name	Extension	Date accessed	Date modified	Date created
1	Table	NZBIUG July 2015		7/27/2015 11:14:35 PM	7/27/2015 11:14:35 PM	7/27/2015 11:14:17 PM
2	Table	SQL Rally Nordic 2015		3/4/2015 8:45:25 PM	3/4/2015 8:45:25 PM	3/3/2015 11:31:56 PM
3	Table	SQL Saturday 374 Vienna		2/28/2015 6:28:23 PM	2/28/2015 6:28:23 PM	2/19/2015 9:03:43 PM
4	Table	CloudnEnterprise_Symbols_Public_v2.02		2/1/2015 12:43:29 PM	2/1/2015 12:43:29 PM	2/1/2015 12:43:29 PM
5	Table	TechDays Hong Kong 2015		2/12/2015 4:14:33 PM	2/12/2015 4:14:33 PM	2/1/2015 12:20:56 PM
6	Table	SQL Saturday 337 Oregon		12/13/2014 10:14:55 AM	11/2/2014 11:15:19 AM	10/31/2014 9:49:55 PM
7	Table	SQL Saturday 352 Sydney		2/7/2015 2:59:03 PM	2/7/2015 2:59:03 PM	10/24/2014 10:59:19 PM
8	Table	SQL Social Brisbane 2014		12/13/2014 10:14:55 AM	9/17/2014 4:28:51 PM	9/17/2014 4:28:26 PM
9	Table	MVP ComCamp 2014		12/13/2014 10:14:53 AM	9/17/2014 4:28:26 PM	9/17/2014 4:28:26 PM
10	Table	PASS Summit 2014		12/13/2014 10:14:52 AM	9/13/2014 6:04:20 PM	9/13/2014 5:44:53 PM
11	Table	Auckland CodeCamp 2014		12/13/2014 10:14:54 AM	9/13/2014 2:03:17 PM	9/13/2014 10:07:16 AM
12	Table	TechEd NZ 2014		1/11/2015 3:45:45 PM	1/11/2015 3:45:45 PM	7/30/2014 5:43:35 PM

Fetch Files and Folders with Masking

Now that you've got through the M code, it would be really easy to apply any masking option to this. Here are couple of examples:

1- Fetch Only Folders Created after Specific Date

let

```
Source = Folder.Contents("C:\Users\Reza\Dropbox\Speaking"),
TypeAdded=Table.AddColumn(Source,"Type",each Value.Is([Content],type table)),
Folders=Table.SelectRows(TypeAdded, each [Type]=true),
Sorted=Table.Sort(Folders,{"Date created", Order.Descending}),
Filtered=Table.SelectRows(Sorted, each [Date created]>DateTime.FromText("2015-1-1"))
```

in

Filtered

Result

	Content	Name	Extension	Date accessed	Date modified	Date created
1	Table	NZBIUG July 2015		7/27/2015 11:14:35 PM	7/27/2015 11:14:35 PM	7/27/2015 11:14:17 PM
2	Table	SQL Rally Nordic 2015		3/4/2015 8:45:25 PM	3/4/2015 8:45:25 PM	3/3/2015 11:31:56 PM
3	Table	SQL Saturday 374 Vienna		2/28/2015 6:28:23 PM	2/28/2015 6:28:23 PM	2/19/2015 9:03:43 PM
4	Table	CloudnEnterprise_Symbols_Public_v2.02		2/1/2015 12:43:29 PM	2/1/2015 12:43:29 PM	2/1/2015 12:43:29 PM
5	Table	TechDays Hong Kong 2015		2/12/2015 4:14:33 PM	2/12/2015 4:14:33 PM	2/1/2015 12:20:56 PM

2- Fetch Only Files with .txt extension and name similar to "amp"

let

```
Source = Folder.Files("C:\Users\Reza\Dropbox\Speaking"),
Sorted=Table.Sort(Source,{"Date created", Order.Descending}),
Filtered=Table.SelectRows(Sorted, each [Extension]=".txt" and Text.Contains([Name],"amp"))
```

in

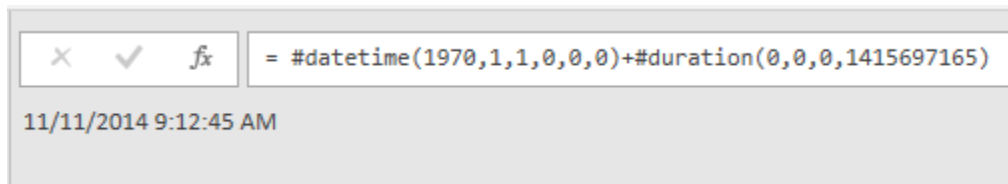
Filtered

Result:

	Content	Name	Extension	Date accessed	Date modified
1	Binary	M sample.txt	.txt	7/27/2015 11:14:38 PM	2/10/2015 10:09:55 AM
2	Binary	M sample.txt	.txt	2/19/2015 9:23:56 PM	2/10/2015 10:09:55 AM
3	Binary	M sample.txt	.txt	12/13/2014 10:15:07 AM	2/10/2015 10:09:55 AM
4	Binary	M sample.txt	.txt	12/13/2014 10:15:04 AM	2/10/2015 10:09:55 AM
5	Binary	M sample.txt	.txt	12/13/2014 10:09:55 AM	2/10/2015 10:09:55 AM

Power Query; Convert Time Stamp to Date Time

Published Date: May 31, 2016



Power Query has number of Date and Time functions that convert values to date and time. However I haven't found a function that converts a timestamp value. Fortunately it is easy to calculate a date time from a timestamp value. In this post I'll explain an easy way of converting timestamp to date time. To Learn more about Power Query read [Power BI online book](#); from Rookie to Rock Star.

What is Timestamp

Timestamp is a whole number value, which is number of seconds from date 1970-01-01 00:00:00. For example; timestamp 100 means 1970-01-01 00:01:40, or timestamp 86400 means 1970-01-02 00:00:00. So the calculation is easy; We have to add timestamp as seconds to the date/time 1970-01-01 00:00:00.

Power Query Convert Timestamp to Date Time

Once we know what is definition of timestamp, and how to calculate date/time from it; easily we can use **#duration(0,0,0,<timestamp value>)** to show duration in seconds, and add it to the **#datetime(1970,1,1,0,0,0)** which is date time 1970-01-01 00:00:00.

So as a result, here is the code to convert timestamp to date time;

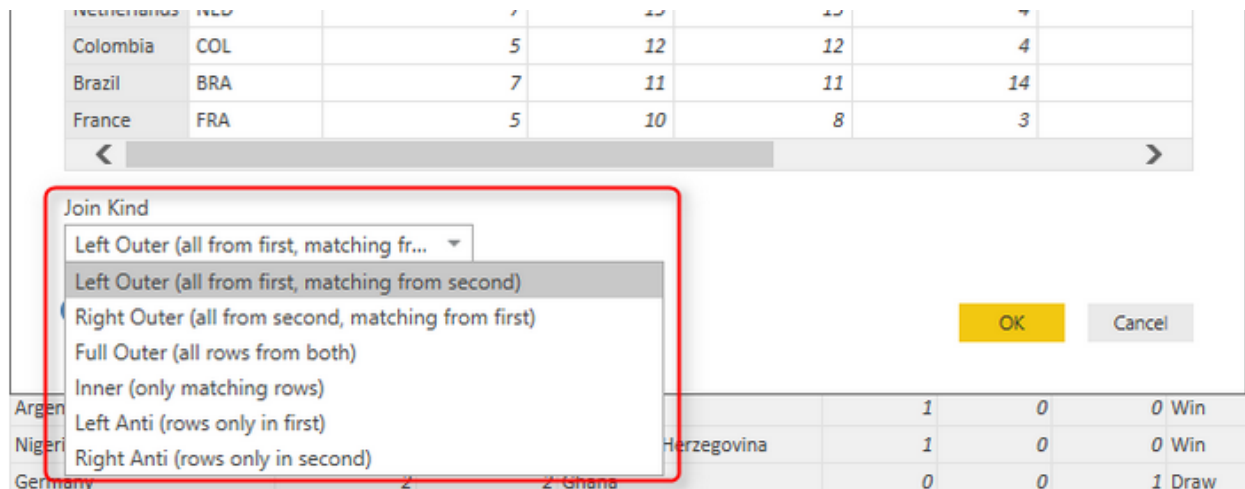
```
DateTimedValue=#datetime(1970,1,1,0,0,0)+#duration(0,0,0,1415697165)
```

above query will respond 11/11/2014 9:12:45 AM.

Please note that you have to replace the 1415697165 number with the field name containing timestamp values, or with your static timestamp value in the query.

How to Change Joining Types in Power BI and Power Query

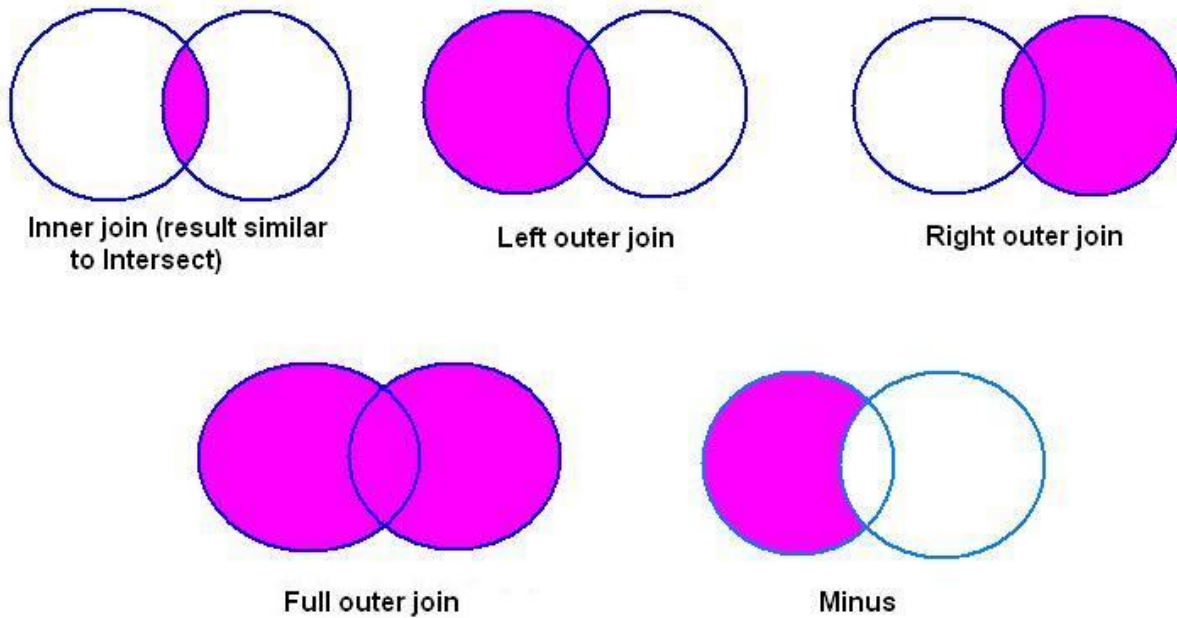
Published Date: July 31, 2015



Joining tables is not a new concept, I bet all readers of my blog at least have a clue about that. However there are different types of joins, and applying these types of Joins are not all possible through Power Query GUI. [Power BI recently took an step and implemented that in the GUI](#), however you might like to know how to apply that in the Power Query. The trick is that M is your friend, You can do whatever you want behind the scenes with M script.

I don't want to go through the details of explaining every join type here. Picture below illustrated it perfectly;

JOINS AND SET OPERATIONS IN RELATIONAL DATABASES

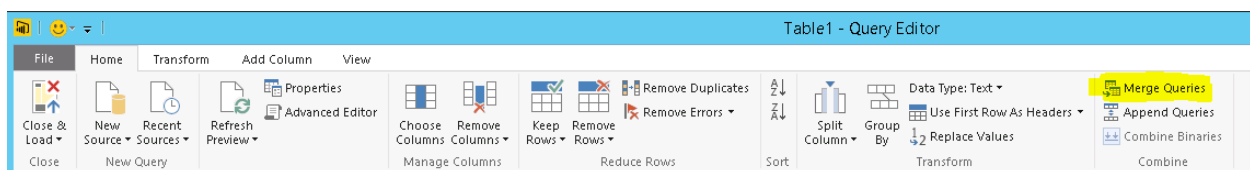


Picture referenced from: http://www.udel.edu/evelyn/SQL-Class2/SQLclass2_Join.html

Now let's see how to use joins through Power BI and Power Query;

Power BI Desktop

In Power BI Desktop you can join two tables with Merge menu item in the Query Editor, in Home tab, Under Combine, Merge Queries.



The Merge Window will appear with ability to select first table (Left part of the join), and the second table (Right part of the join). You can choose columns that you want to participate as joining key within an order (you can choose multiple columns with Ctrl Key). And there is join kind that you can choose.

Merge

Select a table and matching columns to create a merged table.

Venue	Stage	Match	Team 1	Result 1	Result 2	Team 2	Win	Loss	Draw	Resu
Arena Corinthians	Group A	1	Brazil	3	1	Croatia	1	0	0	Win
Arena das Dunas	Group A	2	Mexico	1	0	Cameroon	1	0	0	Win
Arena Fonte Nova	Group B	3	Spain	1	5	Netherlands	0	1	0	Loss
Arena Pantanal	Group B	4	Chile	3	1	Australia	1	0	0	Win
Estádio Mineirão	Group C	5	Colombia	3	0	Greece	1	0	0	Win

TeamGoals

Teams	Teams	Matches Played	Goals for	Goals scored	Goals Against	Penalty goal
Germany	GER	7	18	18	4	
Netherlands	NED	7	15	15	4	
Colombia	COL	5	12	12	4	
Brazil	BRA	7	11	11	14	
France	FRA	5	10	8	3	

Join Kind

- Left Outer (all from first, matching fr...
- Left Outer (all from first, matching from second)
- Right Outer (all from second, matching from first)
- Full Outer (all rows from both)
- Inner (only matching rows)
- Left Anti (rows only in first)
- Right Anti (rows only in second)

OK Cancel

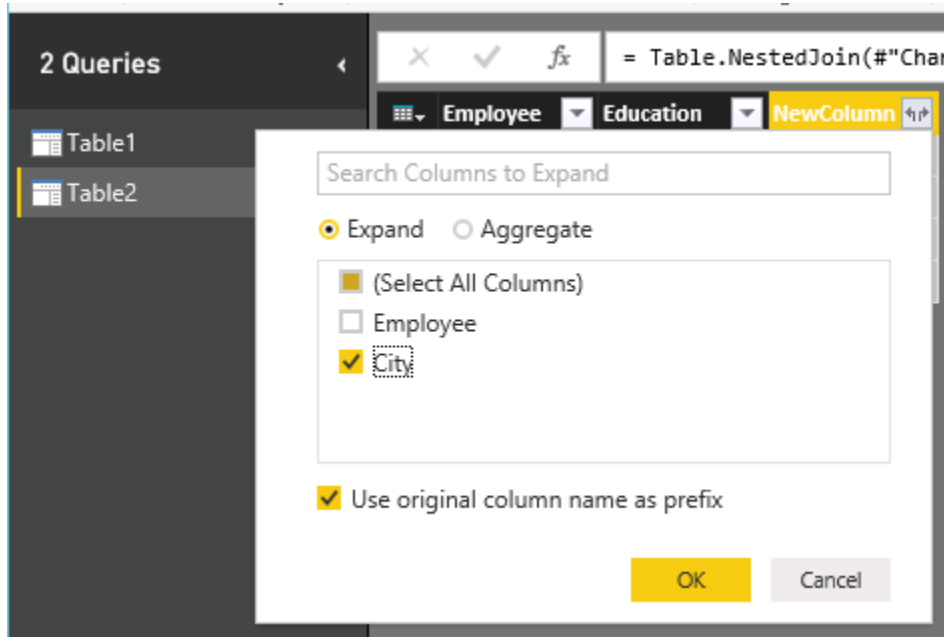
Argentina			1	0	0	Win
Nigeria			1	0	0	Win
Germany	2	2	0	0	1	Draw

The default behavior is left outer join, which means all records from the first table, if there is any record in the first table that matches record(s) in the second table it would be listed as well.

After joining tables, the second table will appear as a field that has table value in it's cells. What you need to do is to select columns that you want to show in the result set.

	Employee	Education	NewColumn
1	Mark	PHD	Table
2	Fernando	MsC	Table
3	Reza	Bachelor	Table
4	Steve	Diploma	Table

Choose columns as below:



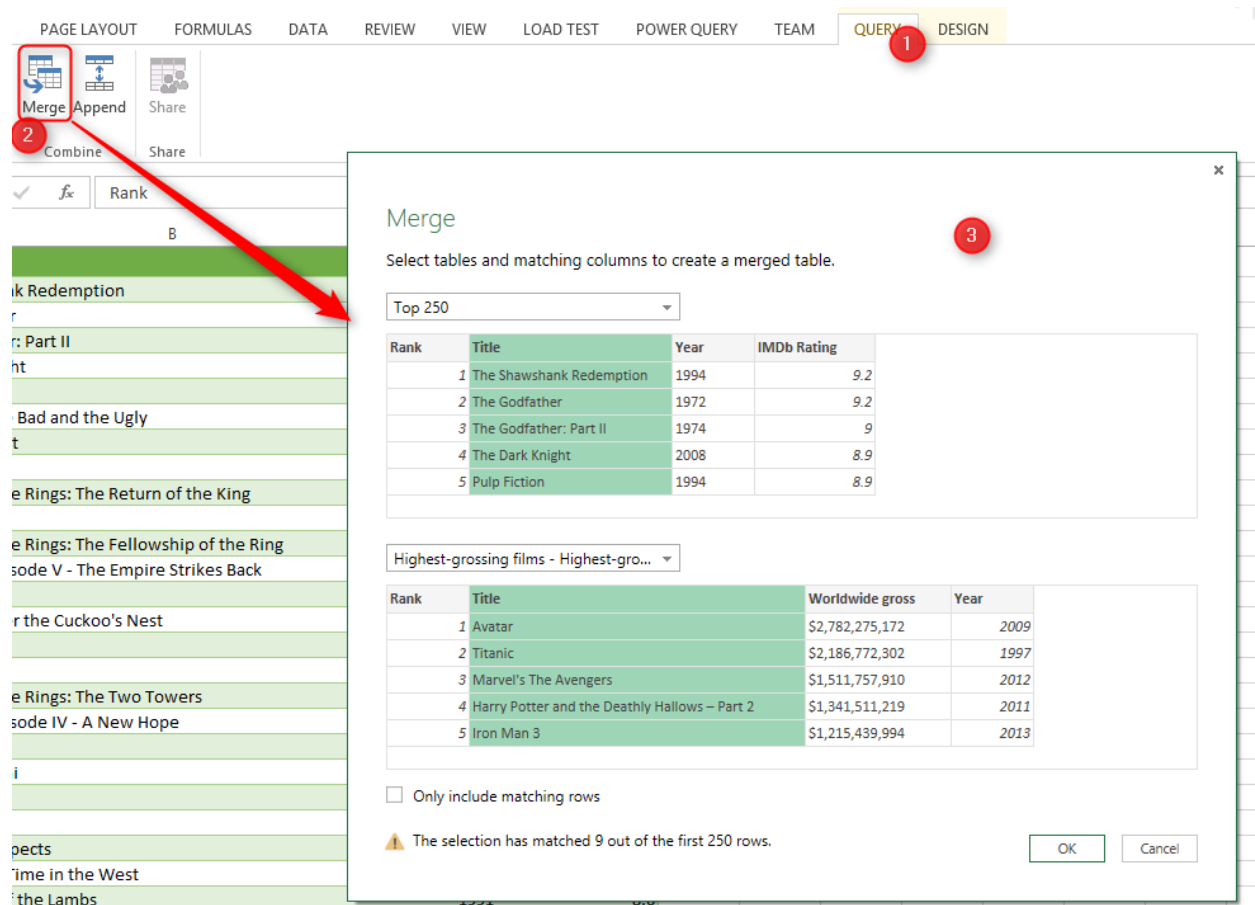
You can also choose from one of other join types as mentioned below:

- Left Outer (all from first, matching from second): this option was the default behavior previously within Merge dialog
- Right Outer (all from second, matching from first)
- Full Outer (all rows from both)
- Inner (only matching rows); this option was available previously through "Choose only matching rows" option in Merge dialog
- Left Anti (rows only in first)
- Right Anti (rows only in second)

Power Query

At the time of writing this blog post Power Query Editor (GUI) only supports two types of joins mentioned above: Left Join, and Inner Join.

You should follow the same path through Merge Queries, and then you will see joining options as below:

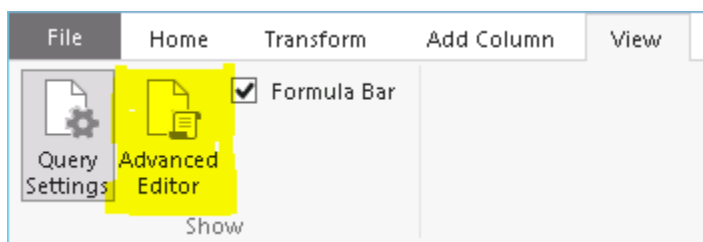


As you see the default behavior is Left join. You can change it to inner join with selecting "Only include matching rows".

Change Through M

You can apply any join type that you want simply by going to advanced editor, and changing the M script as below:

Go to View tab, and click on Advanced Editor:



In the Advanced Editor query window you can see the M script that builds the result set. Find the join function and change the JoinKind.

Table2

```
let
    Source = Excel.Workbook(File.Contents("C:\Users\adm_radr\Documents\join.xlsx"), null, true),
    Table2_Table = Source{[Item="Table2",Kind="Table"]}[Data],
    #"Changed Type" = Table.TransformColumnTypes(Table2_Table,{{"Employee", type text}, {"Education", type text}}),
    #"Merged Queries" = Table.NestedJoin(#"Changed Type",{"Employee"},Table1,{"Employee"},"NewColumn",JoinKind.FullOuter),
    #"Expanded NewColumn" = Table.ExpandTableColumn(#"Merged Queries", "NewColumn", {"City"}, {"NewColumn.City"})
in
    #"Expanded NewColumn"
```

JoinKind is an enumeration type that can have below values:

- JoinKind.Inner=0
- JoinKind.LeftOuter=1
- JoinKind.RightOuter=2
- JoinKind.FullOuter=3
- JoinKind.LeftAnti=4
- JoinKind.RightAnti=5

So you can simply change it as you want.

This feature I reckon soon will be available on Power Query Editor GUI as well, but till that time the above description hopefully help you in any situation that you want to set a join type.

Power Query Library of Functions; Shared Keyword

Published Date: June 20, 2016

Power BI Query Editor interface showing the Record Tools tab and a list of queries.

Record Tools | Untitled - Query Editor

File | **Home** | **Transform** | **Add Column** | **View** | **Convert**

Close & Apply | **New Source** | **Recent Sources** | **Enter Data** | **Manage Parameters** | **Refresh Preview** | **Properties** | **Advanced Editor** | **Manage** | **Choose Columns** | **Remove Columns** | **Manage Columns**

Queries [2]

- Query1
- _Table1

Query1	Record
_Table1	Table
GoogleAnalytics.Accounts	Function
Value.ResourceExpression	Function
Resource.Access	Function
Cube.Parameters	Function
Cube.ApplyParameter	Function
SapHana.Database	Function
SapHanaRangeOperator.Type	Type
SapHanaRangeOperator.GreaterThan	0
SapHanaRangeOperator.LessThan	1
SapHanaRangeOperator.GreaterThanOrEquals	2
SapHanaRangeOperator.LessThanOrEquals	3
SapHanaRangeOperator.Equals	4
SapHanaRangeOperator.NotEquals	5
appFigures.Tables	Function
appFigures.Content	Function
comScore.ReportItems	Function
comScore.NavTable	Function
Github.Contents	Function
Github.PagedTable	Function
Github.Tables	Function
MailChimp.Tables	Function
MailChimp.Contents	Function
MailChimp.Collection	Function
MailChimp.Instance	Function
MailChimp.test	Function
Marketo.Activities	Function
Marketo.Leads	Function

As I mentioned earlier in [Power BI online book](#), Power Query is a functional language. Knowing functions is your best helper when you work with a functional language. Fortunately Power Query both in Excel and Power BI can use shared keyword to reveal a document library of all functions. [I've written about shared keyword](#) almost 2.5 years ago, when it was only an add-in for excel. However I still see people in my webinars who are new with #shared keyword functionality and amazed how helpful this little keyword is. So I decided to explain it with the new Power BI. With the method in this post you can find any function you want easily in Power Query, and you won't need an internet connection to search in functions.

#shared Keyword

Shared is a keyword that loads all functions, and enumerators in the result set. You can simply create a blank query in Power BI (or Excel)

Get Data

All

File











Database

Azure

Online Services

Other

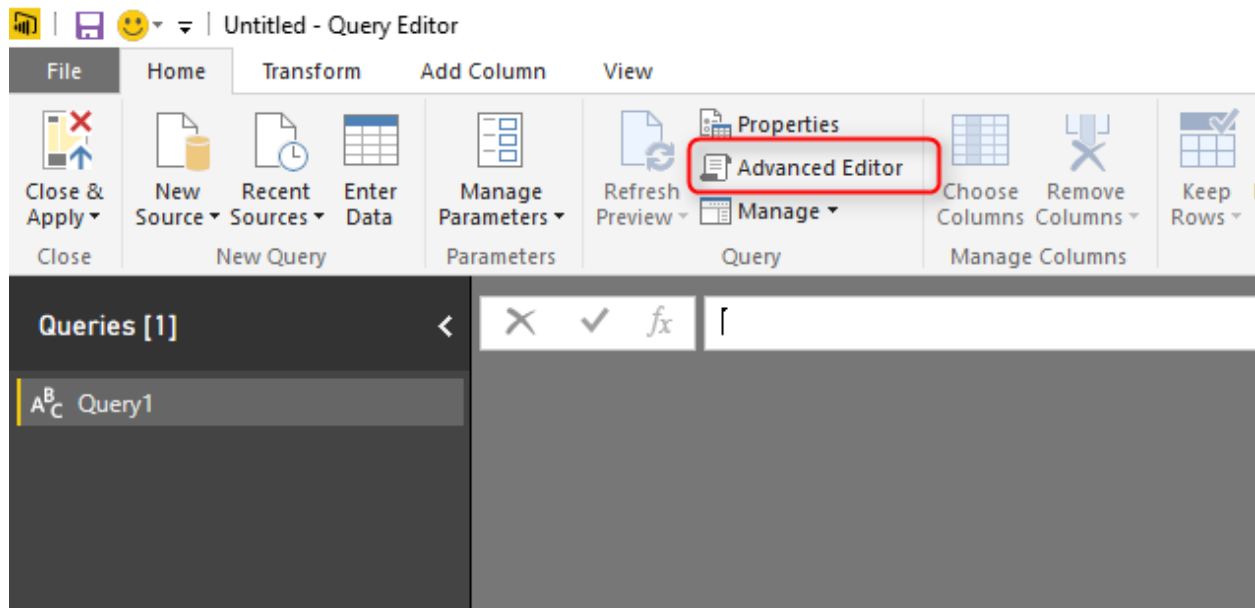
Other

-  Web
-  SharePoint List
-  OData Feed
-  Hadoop File (HDFS)
-  Active Directory
-  Microsoft Exchange
-  ODBC
-  R Script
-  Spark (Beta)
-  **Blank Query**

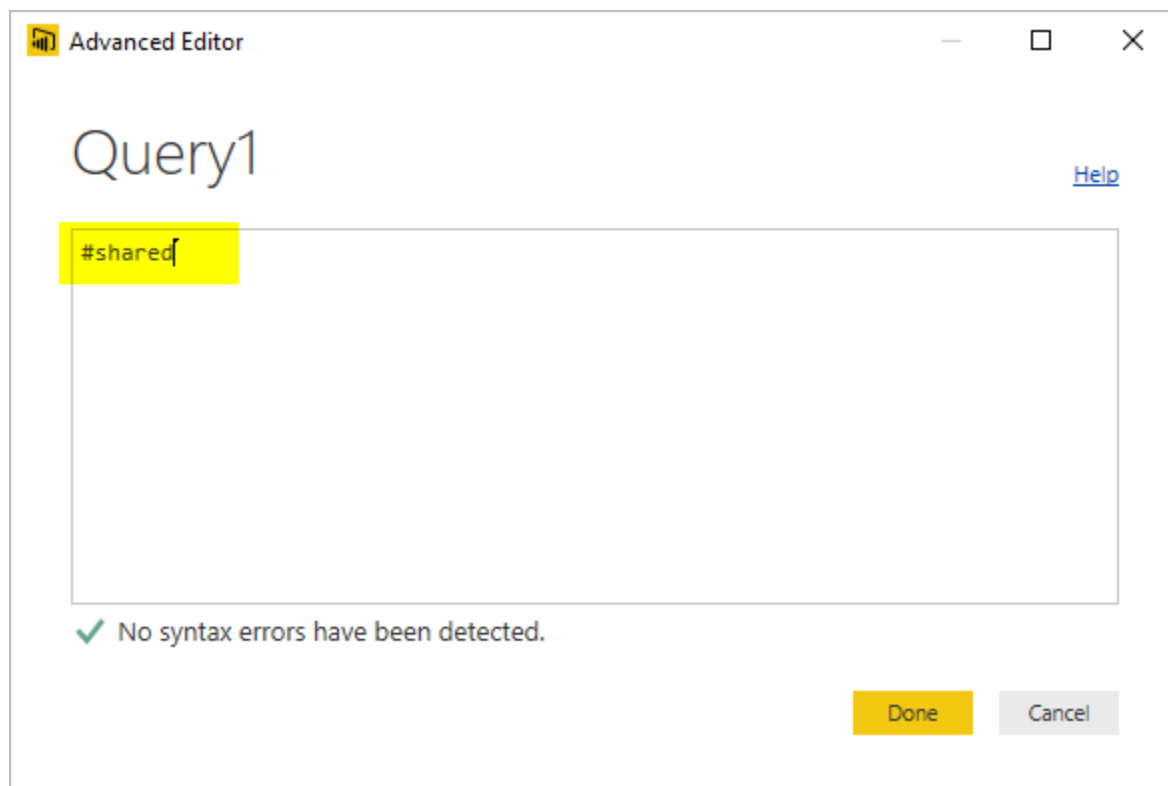
Connect

Cancel

After opening a blank query, go to Advanced Editor



Here in the script editor is where you usually write or modify power query script. Now delete all script here and only type in: #shared



After clicking on Done, you will see list of all functions, and enumerators in power query. You will also see other queries in your Power BI solution or workbook plus other custom functions.

Power BI Query Editor interface showing the Record Tools tab and a list of queries.

Record Tools | Untitled - Query Editor

File | **Home** | **Transform** | **Add Column** | **View** | **Convert**

Close & Apply | **New Source** | **Recent Sources** | **Enter Data** | **Manage Parameters** | **Refresh Preview** | **Properties** | **Advanced Editor** | **Choose Columns** | **Remove Columns** | **Manage Columns**

Queries [2]

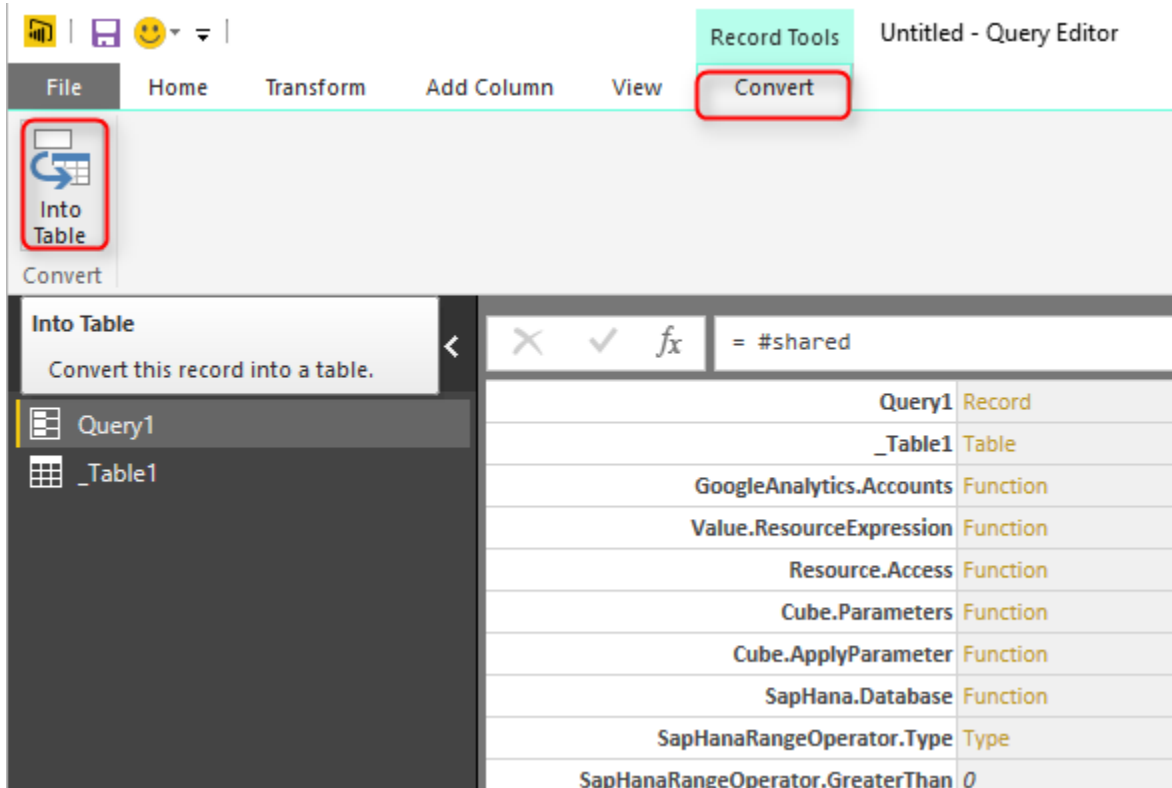
- Query1
- _Table1

Query1	Record
_Table1	Table
GoogleAnalytics.Accounts	Function
Value.ResourceExpression	Function
Resource.Access	Function
Cube.Parameters	Function
Cube.ApplyParameter	Function
SapHana.Database	Function
SapHanaRangeOperator.Type	Type
SapHanaRangeOperator.GreaterThan	0
SapHanaRangeOperator.LessThan	1
SapHanaRangeOperator.GreaterThanOrEquals	2
SapHanaRangeOperator.LessThanOrEquals	3
SapHanaRangeOperator.Equals	4
SapHanaRangeOperator.NotEquals	5
appFigures.Tables	Function
appFigures.Content	Function
comScore.ReportItems	Function
comScore.NavTable	Function
Github.Contents	Function
Github.PagedTable	Function
Github.Tables	Function
MailChimp.Tables	Function
MailChimp.Contents	Function
MailChimp.Collection	Function
MailChimp.Instance	Function
MailChimp.test	Function
Marketo.Activities	Function
Marketo.Leads	Function

Result set here is a Record structure that has functions, enumerators, and queries in each item of the record. Now let's explore the record more in details.

Use the Result Set as a Table

Result above is loaded in Power Query and that is the greatest feature of Power Query itself that can turn this result into a table.



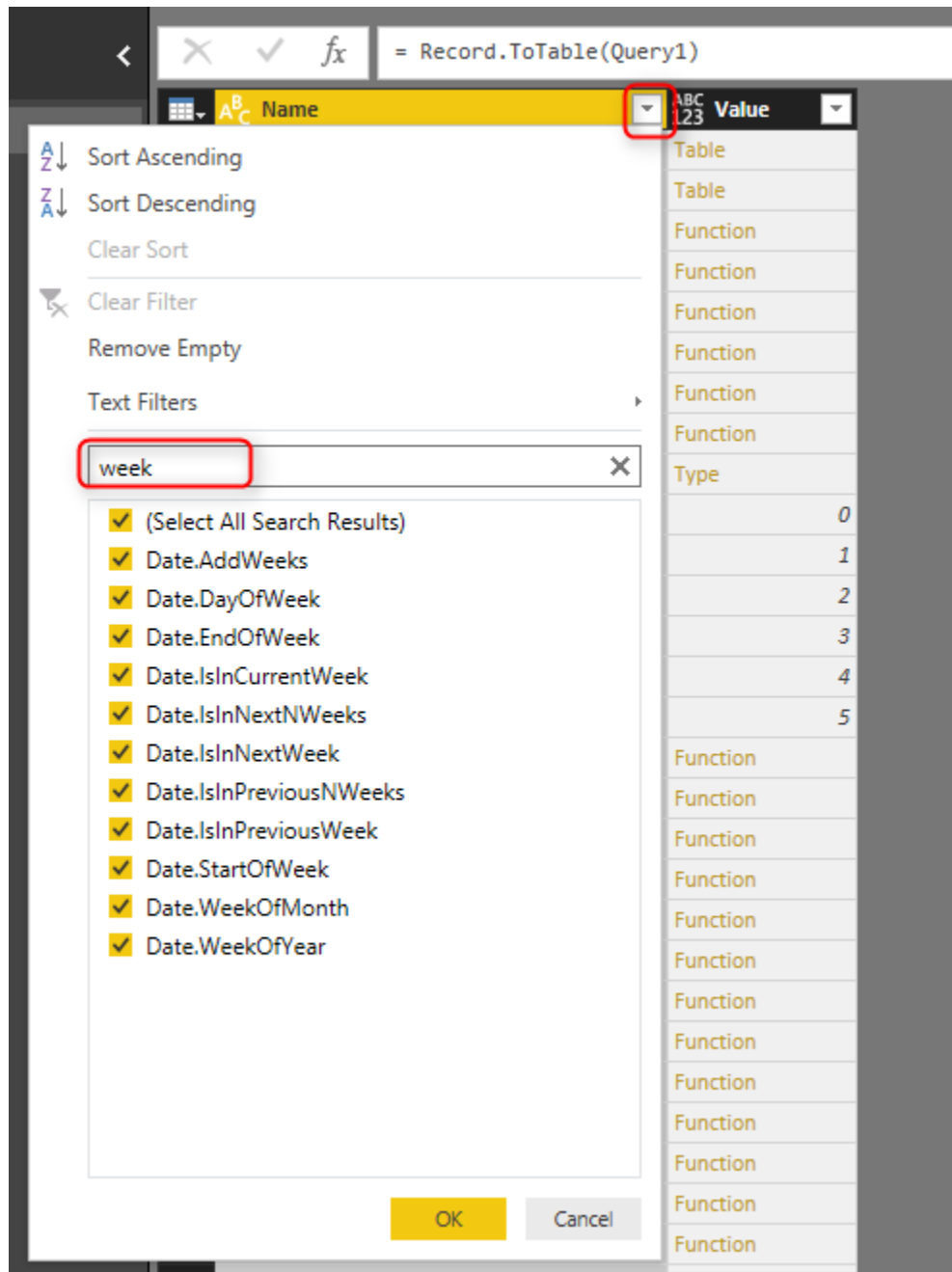
The screenshot shows the Power Query Editor interface. The 'Record Tools' ribbon is active, and the 'Convert' button is highlighted. The 'Into Table' dialog box is open, showing the option to 'Convert this record into a table.' The list of items to be converted includes:

Item	Type
Query1	Record
_Table1	Table
GoogleAnalytics.Accounts	Function
Value.ResourceExpression	Function
Resource.Access	Function
Cube.Parameters	Function
Cube.ApplyParameter	Function
SapHana.Database	Function
SapHanaRangeOperator.Type	Type
SapHanaRangeOperator.GreaterThan	0

This will convert result set record to a table. and table is really easy to search in as you know.

✕ ✓ <i>f_x</i> = Record.ToTable(Query1)		
	ABC 123 Name	ABC 123 Value
1	Query1	Table
2	_Table1	Table
3	GoogleAnalytics.Accounts	Function
4	Value.ResourceExpression	Function
5	Resource.Access	Function
6	Cube.Parameters	Function
7	Cube.ApplyParameter	Function
8	SapHana.Database	Function
9	SapHanaRangeOperator.Type	Type
10	SapHanaRangeOperator.GreaterThan	0
11	SapHanaRangeOperator.LessThan	1
12	SapHanaRangeOperator.GreaterThanOrEquals	2
13	SapHanaRangeOperator.LessThanOrEquals	3
14	SapHanaRangeOperator.Equals	4
15	SapHanaRangeOperator.NotEquals	5
16	appFigures.Tables	Function
17	appFigures.Content	Function
18	comScore.ReportItems	Function
19	comScore.NavTable	Function
20	Github.Contents	Function
21	Github.PagedTable	Function
22	Github.Tables	Function
23	MailChimp.Tables	Function
24	MailChimp.Contents	Function
25	MailChimp.Collection	Function
26	MailChimp.Instance	Function
27	MailChimp.test	Function
28	Marketo.Activities	Function
29	Marketo.Leads	Function
30	Marketo.Tables	Function

I can now simply search in the function list. Let's see for example what function I can find for working with WEEK;



This will filter the table to a sub set that I can see only functions with WEEK in the name of it.

fx = Table.SelectRows(#"Converted to Table", each ([Name] = "Date.AddWeeks" or [Name] = "Date.DayOfWeek" or [Name]

	A ^B C Name	ABC 123 Value
1	Date.IsInPreviousWeek	Function
2	Date.IsInPreviousNWeeks	Function
3	Date.IsInCurrentWeek	Function
4	Date.IsInNextWeek	Function
5	Date.IsInNextNWeeks	Function
6	Date.AddWeeks	Function
7	Date.StartOfWeek	Function
8	Date.EndOfWeek	Function
9	Date.DayOfWeek	Function
10	Date.WeekOfMonth	Function
11	Date.WeekOfYear	Function

function (dateTime as any, numberOfWeeks as number) as any

Returns the date, datetime, or datetimezone result from adding numberOfWeeks weeks to the datetime value dateTime.

dateTime: The date, datetime, or datetimezone value to which weeks are being added.
 numberOfWeeks: The number of weeks to add.

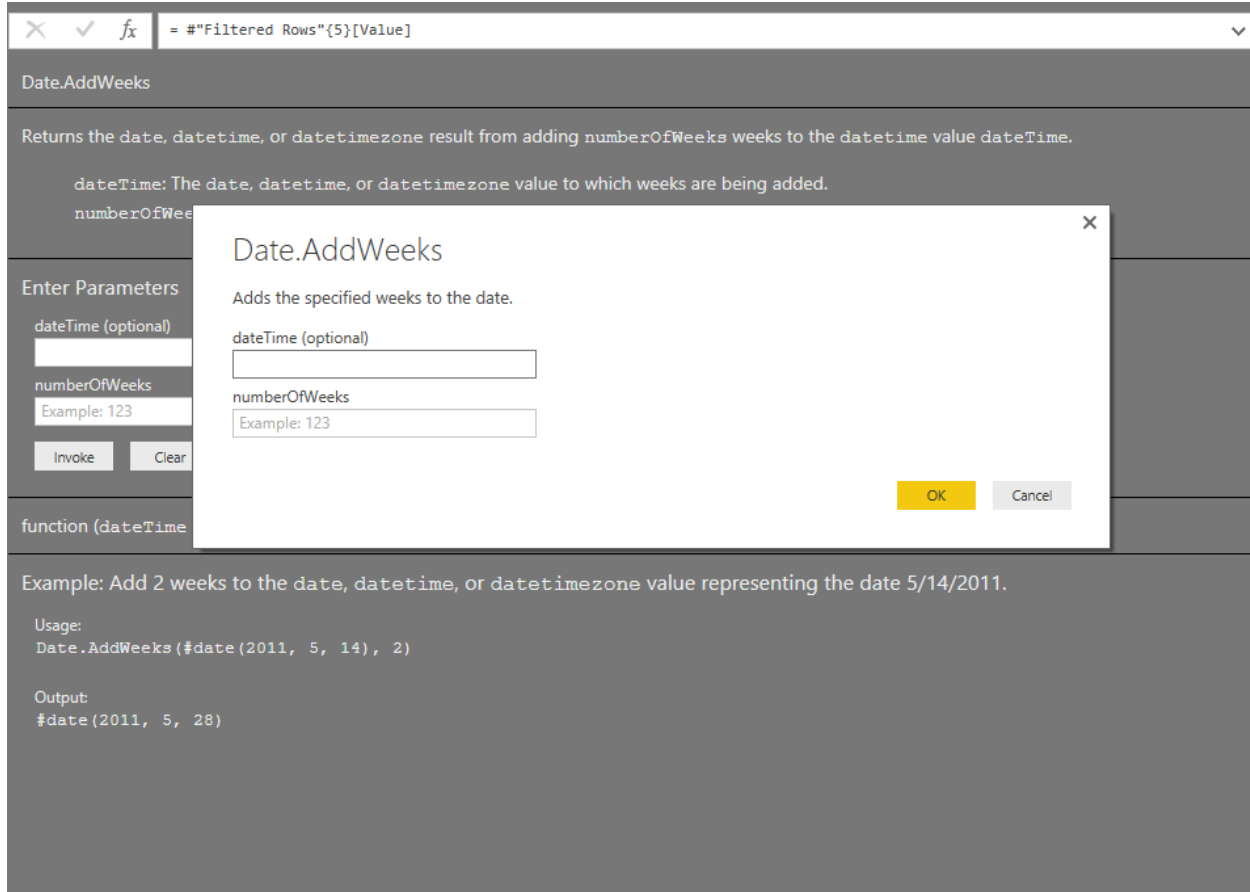
Example: Add 2 weeks to the date, datetime, or datetimezone value representing the date 5/14/2011.

Usage:
 Date.AddWeeks (#date (2011, 5, 14), 2)

Output:
 #date (2011, 5, 28)

Documentation of Function

Now for example if I want to see the Date.AddWeeks function I can click on the "function" link of it in value column and this will redirect me to the documentation of this function, and will bring a dialog box to invoke the function!



`= #\"Filtered Rows\"{5}[Value]`

Date.AddWeeks

Returns the date, datetime, or datetimezone result from adding numberOfWeeks weeks to the datetime value dateTime.

dateTime: The date, datetime, or datetimezone value to which weeks are being added.

numberOfWeeks

Enter Parameters

dateTime (optional)

numberOfWeeks
Example: 123

function (dateTime

Example: Add 2 weeks to the date, datetime, or datetimezone value representing the date 5/14/2011.

Usage:
`Date.AddWeeks(#date(2011, 5, 14), 2)`

Output:
`#date(2011, 5, 28)`

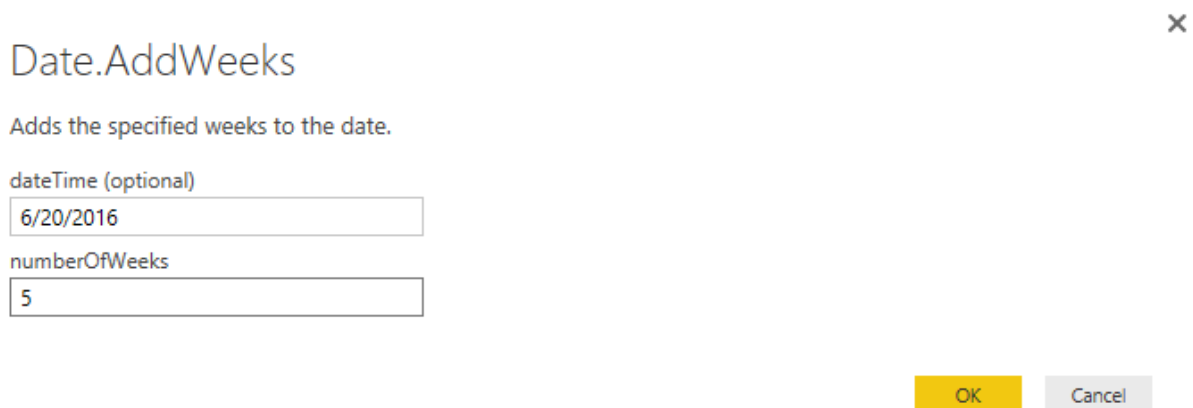
Date.AddWeeks

Adds the specified weeks to the date.

dateTime (optional)

numberOfWeeks
Example: 123

You can see the documentation in grey that also includes examples of how to call this function. For invoking the function I can simply provide parameters, and click OK or Invoke;



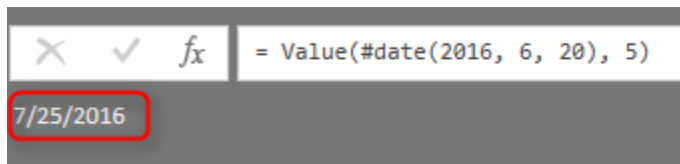
Date.AddWeeks

Adds the specified weeks to the date.

dateTime (optional)

numberOfWeeks

And it calls the function and show me the result as below;



Enumerators

Finding enumerators is also easy with help of #shared keyword. Here I can see enumerators for JoinKind and JoinAlgorithm;

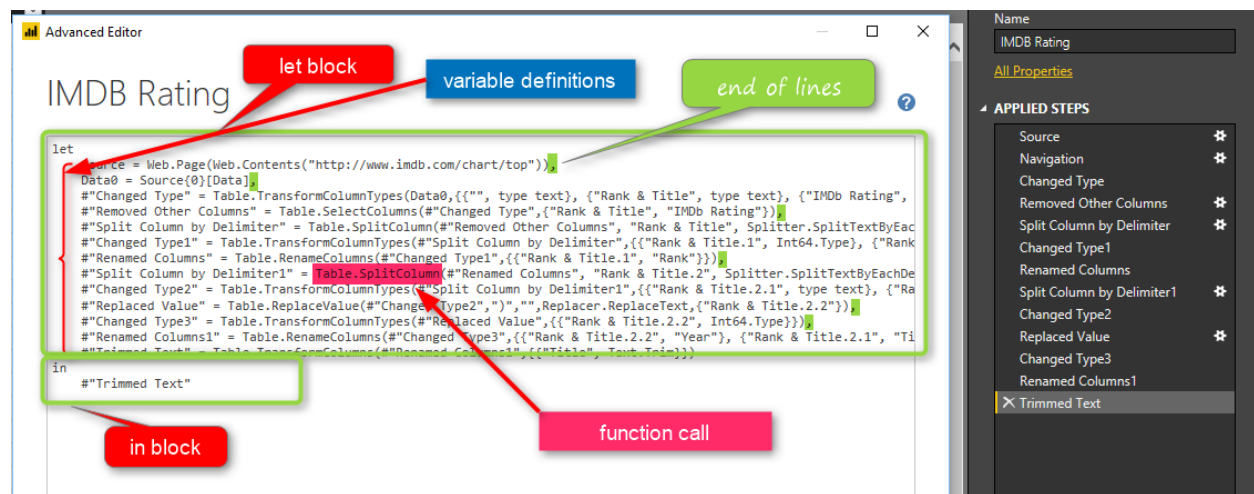
	ABC 123	Name	Value
1		JoinKind.Type	Type
2		JoinKind.Inner	0
3		JoinKind.LeftOuter	1
4		JoinKind.RightOuter	2
5		JoinKind.FullOuter	3
6		JoinKind.LeftAnti	4
7		JoinKind.RightAnti	5
8		Table.Join	Function
9		Table.AddJoinColumn	Function
10		Table.NestedJoin	Function
11		JoinAlgorithm.Type	Type
12		JoinAlgorithm.Dynamic	0
13		JoinAlgorithm.PairwiseHash	1
14		JoinAlgorithm.SortMerge	2
15		JoinAlgorithm.LeftHash	3
16		JoinAlgorithm.RightHash	4
17		JoinAlgorithm.LeftIndex	5
18		JoinAlgorithm.RightIndex	6

Summary

You've learned how #shared keyword can be helpful for getting list of all functions and enumerators in Power Query. You learned you can convert the result set into a table and filter that to find particular function you are looking for. This is superb, especially for people like myself who can't remember things well. You know how I look for functions? this post explained my method! I use #shared keyword to find the function I want and start working on that. for me #shared is the keyword that I use more than any other queries in Power query side of Power BI.

Basics of M: Power Query Formula Language

Published Date: July 10, 2017



M is the powerful language behind the scene of Power Query. Any transformation you apply will be written in M language. For many, M looks like a scary language. In this post I like to explain a bit of basics of M. Not mentioning any functions. Mainly I want to explain to you how the M language syntax is structured. Once you know the syntax, then everything becomes simple. M is a language that you can learn its syntax easily. As a Power Query developer; I highly recommend you to spend time on M, because there are MANY operations that you can with M, but you might not be able to do it simply with the graphical interface. If you would like to learn more about Power BI, read [Power BI book from Rookie to Rock Star](#).

What is M?

M is informal name of Power Query Formula Language. The formal name is so long that no one uses that, everyone call it M! M stands for Data Mashup, some say stands for Data Modeling. M is a functional language, and it is important to know functions of it. However, each language has a structure and syntax which is the beginner level of learning that language. In this post, I will explain the syntax of M. Before learning M, I would like you to read this sentence loud;

M is much more powerful than the graphical interface of Power Query

Yes, you read it correct! The graphical interface of Power Query is changing every month. Every month new functionality comes to this graphical interface. But the fact is all of these functionalities has been in the language for many years! if you knew the language, you could easily use them, instead of waiting for graphical interface option for it. There are heaps of examples for it. one very small example is [here](#): you can extend your Grouping ability in Power Query with a very minor change in M script of it.

Syntax of M

Syntax of this language is simple. it always has two blocks of programming: LET expression block, and IN expression block. Here is the most simple M syntax;

```
let
    x=1
in
    x
```

let and **in** are reserved words. before going even one step further, the first and foremost thing you need to know;

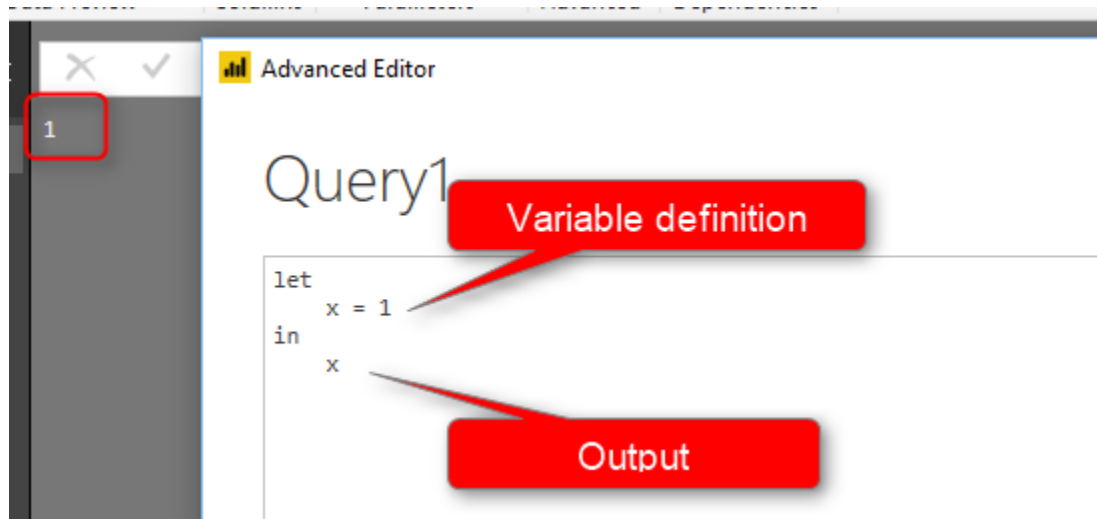
M (Power Query Formula Language) is Case Sensitive. There is a difference between x and X.

what are these two programming blocks:

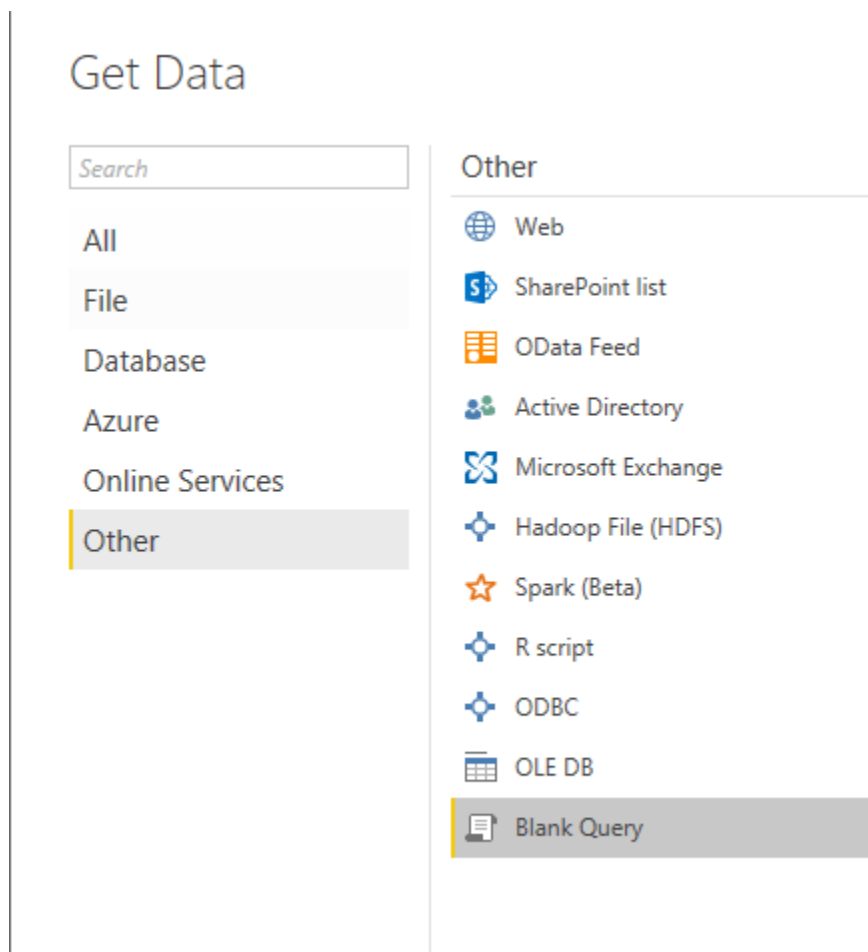
let: definition of all variables

in: output! Yes, in actually means out! just named as in. everything you put in this block will be the output of your query.

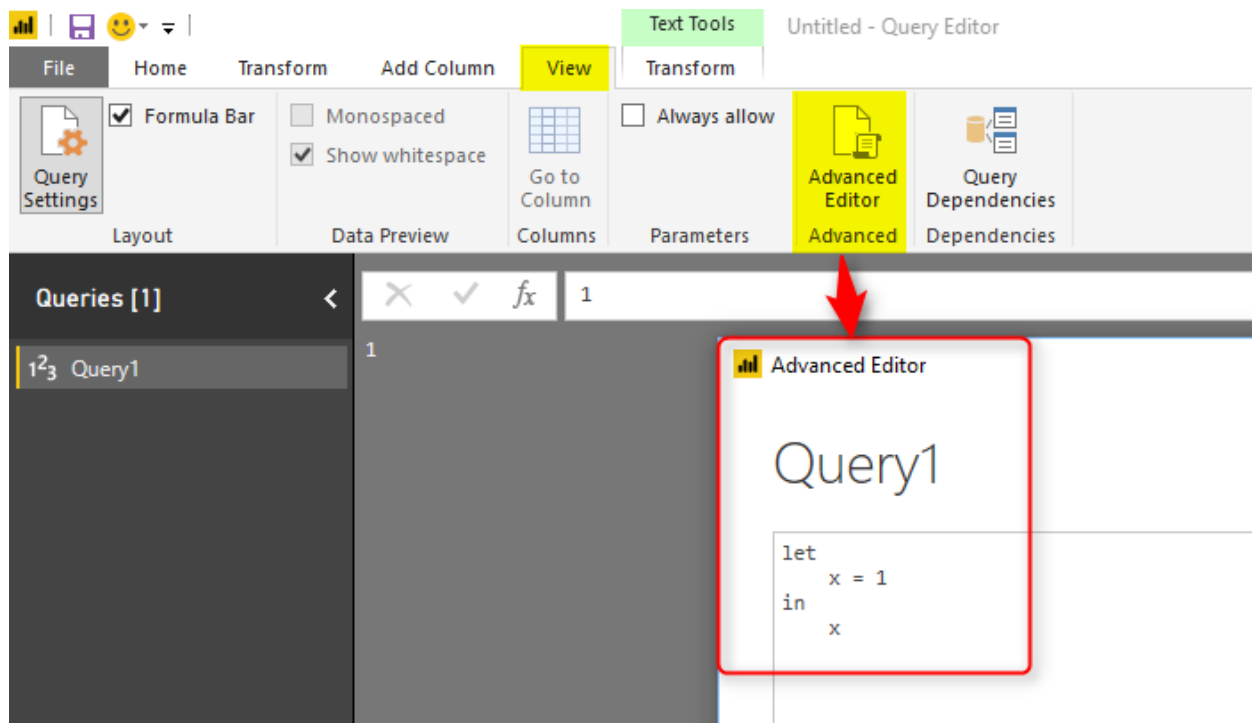
So basically, the query below means defining a variable named as x, assigning the value 1 to it, and showing it as the result set. so the query will return 1.



to run this example, you need to Open Power BI Desktop. Go to Get Data, start with New Blank Query.



then in View tab, select advanced Editor;



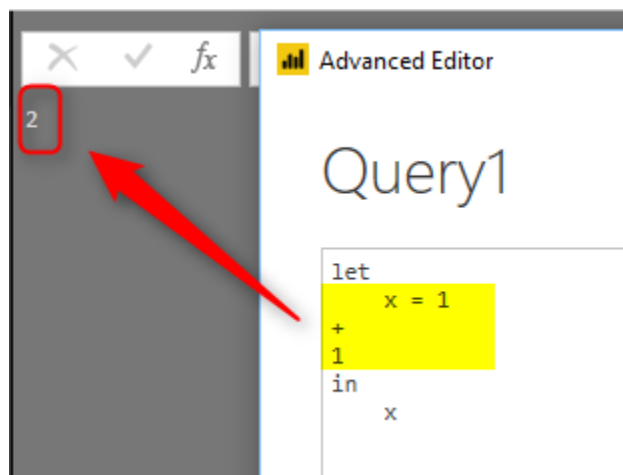
Make sure when you write the script that you put reserved words such as `let` and `in` all lowercase. also your variable name should be the same case in both `let` and `in` section.

As you can see, there is no need to define data types for variable. it will be automatically assigned when the first assignment occurs.

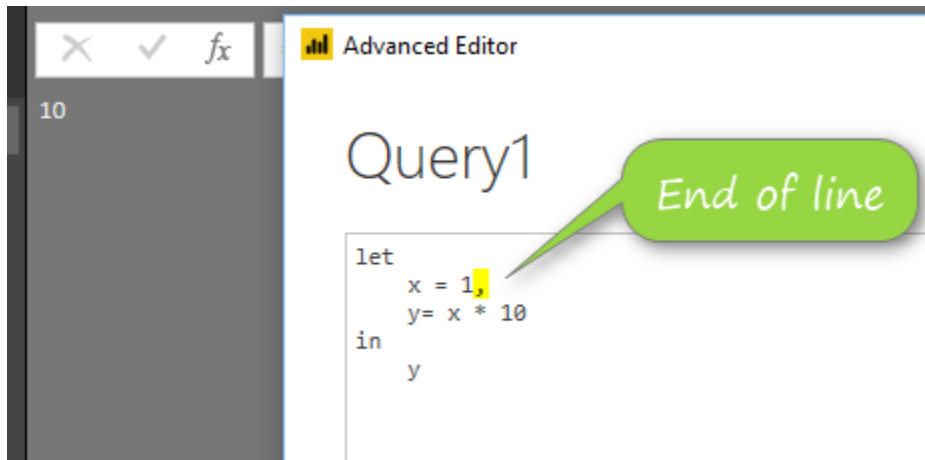
If you specify a text value, then variable would be a text data type automatically.

End of the Line

Lines of codes in M continues if you don't put the end of the line character.



As you can see in above example, the line continues, and x will be equal to $x=1+1$. If you want to put an end for a line use comma(.). example here:

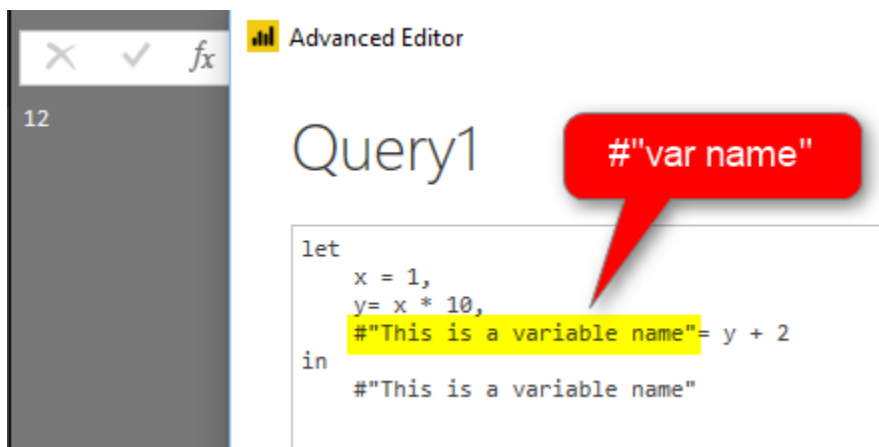


every line needs a comma(,) to finish. except the last line before **in**.

Variable Names

Name of variables can be all one word, like Source. or it can has spaces in it. in case that you have some characters such as space, then you need to put the name inside double quote (") and put a hashtag at the beginning of it(#). something similar to:

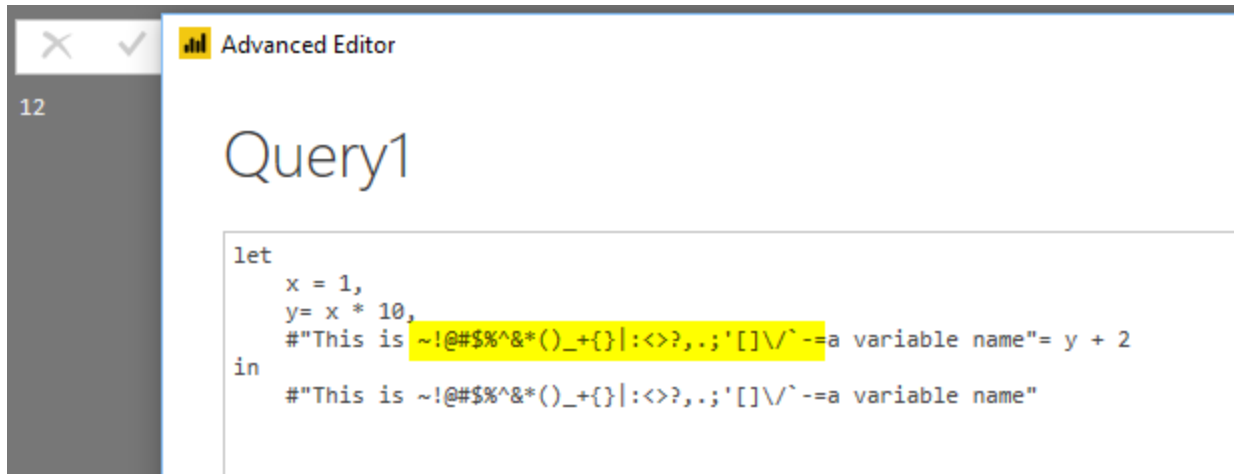
`#"This is a variable name"`



Variable name can contain special characters, here is an example:

Special characters

Variable names can have special character, as you can see below variable has all types of characters in it and still runs good.



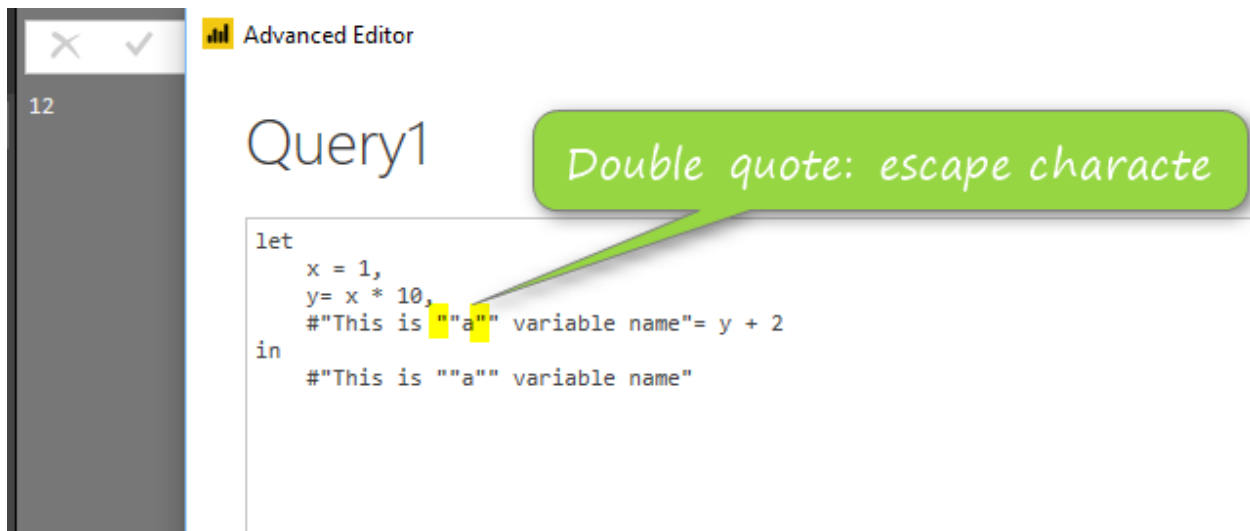
The screenshot shows the Power BI Advanced Editor interface. On the left, a sidebar displays the number '12'. The main area is titled 'Query1' and contains the following M code:

```
let
    x = 1,
    y = x * 10,
    #"This is ~!@#$$%^&*()_+{|:;<>?,.,;'[]\/'-a variable name"= y + 2
in
    #"This is ~!@#$$%^&*()_+{|:;<>?,.,;'[]\/'-a variable name"
```

The variable name in the code is highlighted in yellow.

Escape character

Double quote (") is escape character. you can use it to define variables with names that has another double quote in it. here is an example:



The screenshot shows the Power BI Advanced Editor interface. On the left, a sidebar displays the number '12'. The main area is titled 'Query1' and contains the following M code:

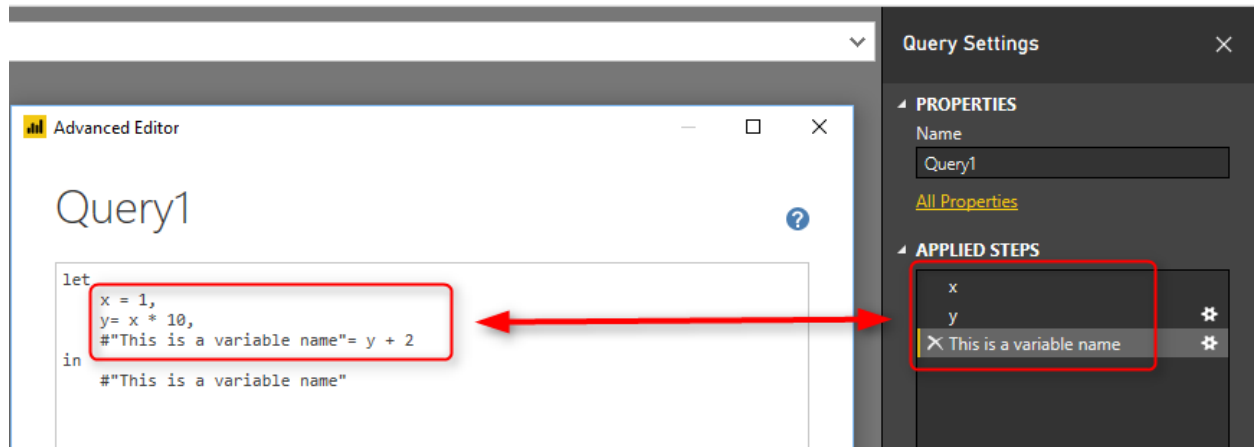
```
let
    x = 1,
    y = x * 10,
    #"This is ""a"" variable name"= y + 2
in
    #"This is ""a"" variable name"
```

The first double quote in the variable name is highlighted in yellow. A green callout bubble points to this quote with the text: "Double quote: escape character".

first double quote (highlighted) above is necessary to be before the second double quote (which is part of the variable name).

Step by Step Coding

Power Query is a step by step transformation. Every transformation usually happens in a step. While you are writing the code, you can also notice that in the right hand side, you will see every variable forms a step.

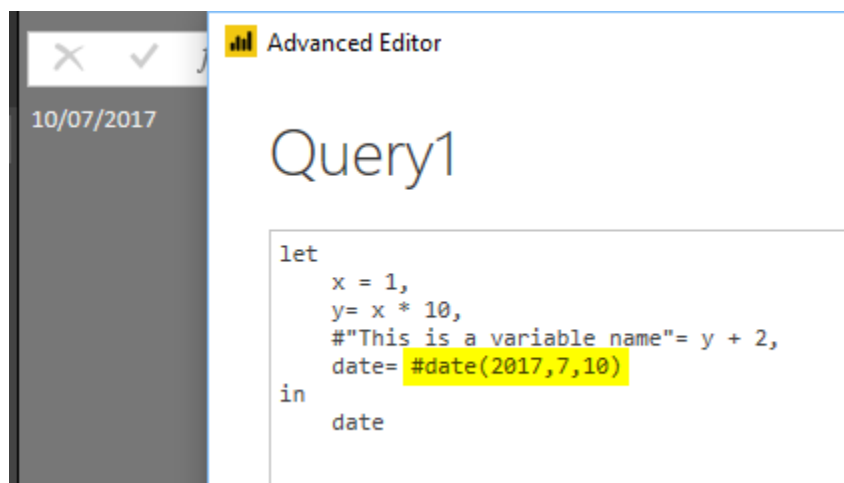


in screenshot above, you can see every variable is determined as a step. and if the variable has space in the name, it will show it with spaces in list of applied steps.

The last variable is always specified in the **in** section.

Literals

There are different ways of defining every literal in Power Query. For example, if you want to define a date variable, here is how to do it;



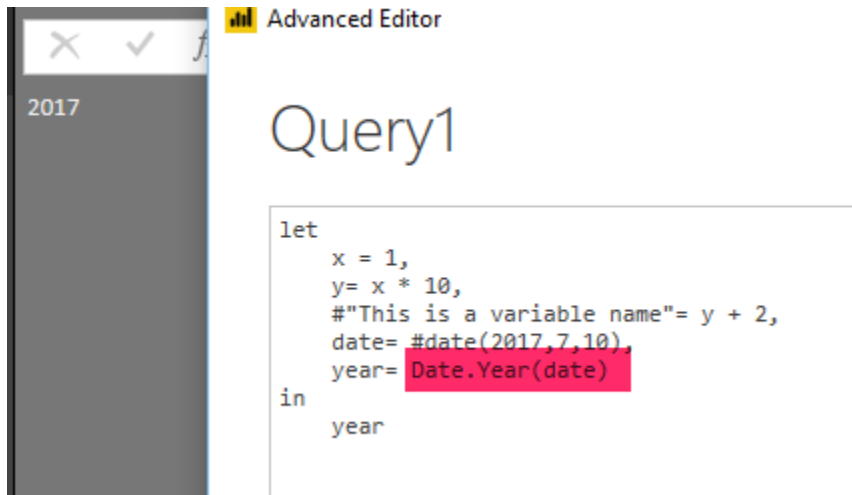
for defining all other types of literals, here is the reference table to use:

Kind	Literal
<i>Null</i>	<code>null</code>
<i>Logical</i>	<code>true false</code>
<i>Number</i>	<code>0 1 -1 1.5 2.3e-5</code>
<i>Time</i>	<code>#time(09,15,00)</code>
<i>Date</i>	<code>#date(2013,02,26)</code>
<i>DateTime</i>	<code>#datetime(2013,02,26, 09,15,00)</code>
<i>DateTimeZone</i>	<code>#datetimezone(2013,02,26, 09,15,00, 09,00)</code>
<i>Duration</i>	<code>#duration(0,1,30,0)</code>
<i>Text</i>	<code>"hello"</code>
<i>Binary</i>	<code>#binary("AQID")</code>
<i>List</i>	<code>{1, 2, 3}</code>
<i>Record</i>	<code>[A = 1, B = 2]</code>
<i>Table</i>	<code>#table({"X","Y"},{{0,1},{1,0}})</code>
<i>Function</i>	<code>(x) => x + 1</code>
<i>Type</i>	<code>type { number } type table [A = any, B = text]</code>

* for function and type; I'll write another post later to explain how these types works.

Function Call

M is a functional language, and for doing almost everything you need to call a function for it. functions can be easily called with name of the function and specifying parameters for it.



screenshot above uses Date.Year function which fetch year part of a date. Functions names starts always with capital letters: **Date.Year()**

Comments

like any programming language, you can put some comments in your code. it can be in two forms;

Single line commentary with double slash (//)

```
let
// this is a comment line and will not be executed
    x = 1,
    y = x * 10,
    #"This is a variable name" = y + 2,
    date = #date(2017, 7, 10),
    year = Date.Year(date)
in
    year
```

Multi line commentary between slash and stars (/* comments */)

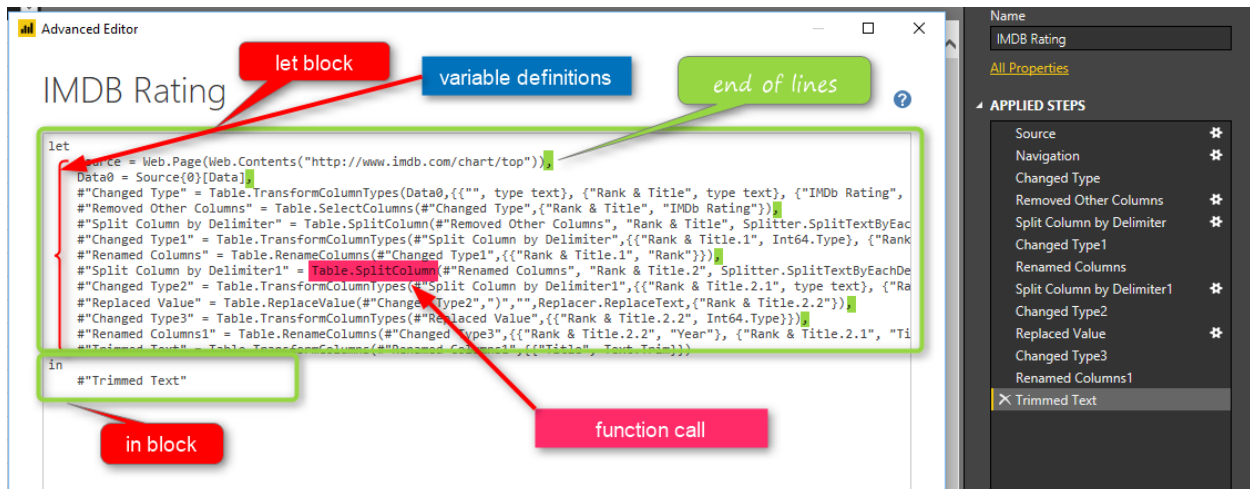
```

let
    /* this is a
    multi
    line
    comment and will not be executed
    */
    x = 1,
    y = x * 10,
    #"This is a variable name" = y + 2,
    date = #date(2017, 7, 10),
    year = Date.Year(date)
in
    year[

```

A real-world example

Now that you know some basics, let's look at an existing query in advanced editor mode and understand it.



The screenshot shows the Advanced Editor window with a query titled "IMDB Rating". The query is written in M language. Annotations with colored boxes and arrows point to specific parts of the query:

- let block**: Points to the "let" keyword at the start of the query.
- variable definitions**: Points to the list of variable assignments within the "let" block.
- end of lines**: Points to the comma character at the end of a line within the "let" block.
- in block**: Points to the "in" keyword at the end of the query.
- function call**: Points to the "Table.TransformColumnTypes" function call within the query.

The right-hand pane shows the "APPLIED STEPS" list, which includes steps like "Source", "Navigation", "Changed Type", "Removed Other Columns", "Split Column by Delimiter", "Changed Type1", "Renamed Columns", "Split Column by Delimiter1", "Changed Type2", "Replaced Value", "Changed Type3", "Renamed Columns1", and "Trimmed Text".

in screenshot above, you can see all basics mentioned so far:

- let and in block
- variable names matching steps applied in the query
- some variable names with hashtag and double quote: #"var name"
- end of the line characters: comma
- calling many functions

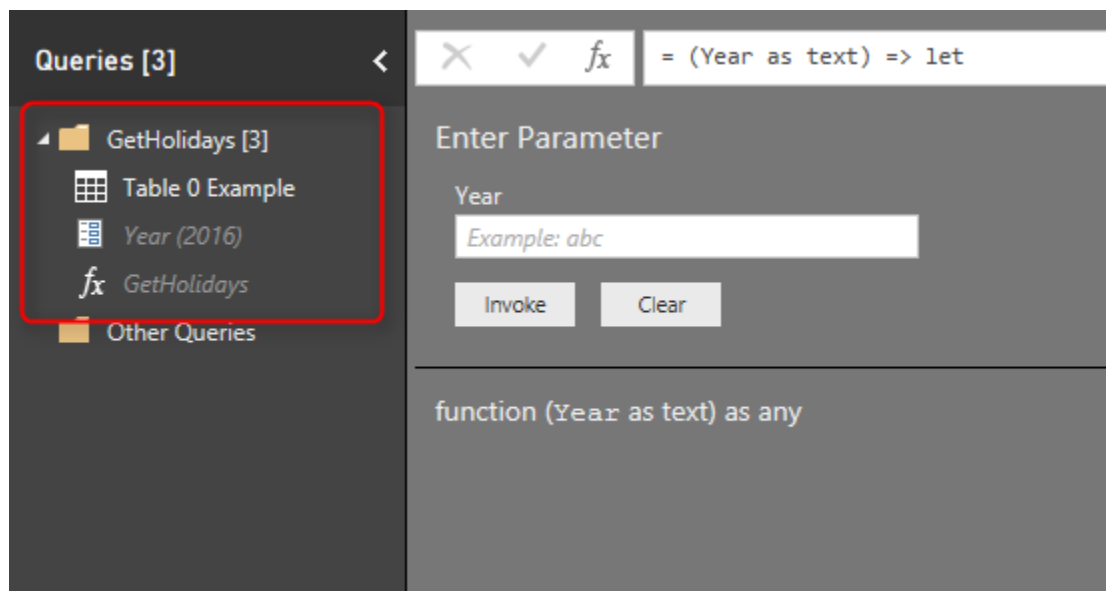
There are still many parts of this query that you might not understand. specially when using functions. you need to learn what functions are doing in order to understand the

code fully. I have written a blog post, that explains how to [use #shared keyword to get documentation of all functions in Power Query](#).

In next posts, I'll explain another levels of structures in M.

Custom Functions Made Easy in Power BI Desktop

Published Date: December 6, 2016



I have written a lot about [Power Query M scripting language](#), and [how to create custom functions](#) with that. With recent updates of Power BI Desktop, creating custom functions made easier and easier every month. This started with bringing Parameters few months ago, and adding source query for the function in [November update of Power BI Desktop](#). In this blog post you will learn how easy is to create a custom function now, what are benefits of doing it in this way, and limitations of it. If you like to learn more about Power BI; read [Power BI online book from Rookie to Rock Star](#).

What is Custom Function?

Custom Function in simple definition is a query that run by other queries. The main benefit of having a query to run by other queries is that you can repeat a number of steps on the same data structure. Let's see that as an example: Website below listed public holidays in New Zealand:

<http://publicholiday.co.nz/>

For every year, there is a page, and pages are similar to each other, each page contains a table of dates and descriptions of holidays. here is for example public holidays of 2016:

<http://publicholiday.co.nz/nz-public-holidays-2016.html>



NZ Public Holidays 2016

Kia ora! We love New Zealand and we love its holidays even more.

2015 2016 2017 2018 2019 School Holidays

	Holiday	Date
✓	New Year's Day	January, Friday 1 st
✓	Day after New Year's Day	January, Saturday 2 nd (observed Monday 4 th)
✓	Wellington Anniversary	January, Monday 25 th
✓	Auckland Anniversary	February, Monday 1 st
✓	Nelson Anniversary	February, Monday 1 st
✓	Waitangi Day	February, Saturday 6 th

Share and get reminded

Follow @PublicHolidayNZ

Tweet

Like 1.8k Share

G+1 136

You can simply use Power BI Get Data from Web menu option of Power Query to get the public holidays of this page. You can also make some changes in the date format to make it proper Date data type. Then you probably want to apply same steps on all other pages for other years (2015, 2017, 2018...), So instead of repeating the process you can reuse an existing query. Here is where Custom Function comes to help.

Benefits of Custom Function

1. Re-Use of Code
2. Increasing Consistency
3. Reducing Redundancy

With a Custom function you are able to re-use a query multiple times. If you want to change part of it, there is only one place to make that change, instead of multiple copies of that. You can call this function from everywhere in your code. and you are reducing redundant steps which normally causes extra maintenance of the code.

How to Create Custom Function?

Well, that's the question I want to answer in this post. Previously (about a year ago), creating custom functions was only possible through M scripting, with lambda expressions, that method still works, but you need to be comfortable with writing M script to use that method (To be honest I am still fan of that method 😊). Recently Power BI Desktop changed a lot, you can now create a function without writing any single line of code.

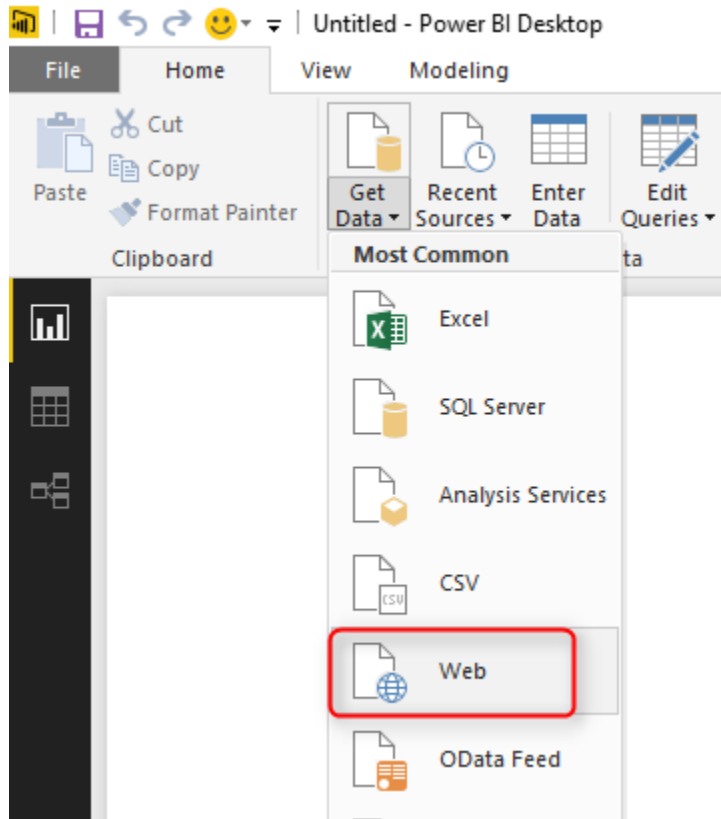
Through an example I'll show you how to create a custom function. This example is fetching all public holidays from the website above, and appending them all in a single table in Power Query. We want the process to be dynamic, so if a new year's data appear in that page, that will be included as well. Let's see how it works.

Building the Main Query

For creating a custom function you always need to build a main query and then convert that to a function. Our main query is a query that will be repeated later on by calling from other queries. In this example our main query is the query that process the holidays table in every page, and return that in proper format of Date data type and Text for description of holiday. We can do that for one of the years (doesn't matter which one). I start with year 2016 which has this url:

<http://publicholiday.co.nz/nz-public-holidays-2016.html>

Open a Power BI Desktop and start by Get Data from Web



Use the 2016's web address in the From Web page;

From Web

☒ Basic ☐ Advanced

Enter a Web page URL.

URL

<http://publicholiday.co.nz/nz-public-holidays-2016.html>

OK

Cancel

In the Navigator you can see that Table 0 is the table containing the data we are after. Select this table and click Edit.

Navigator

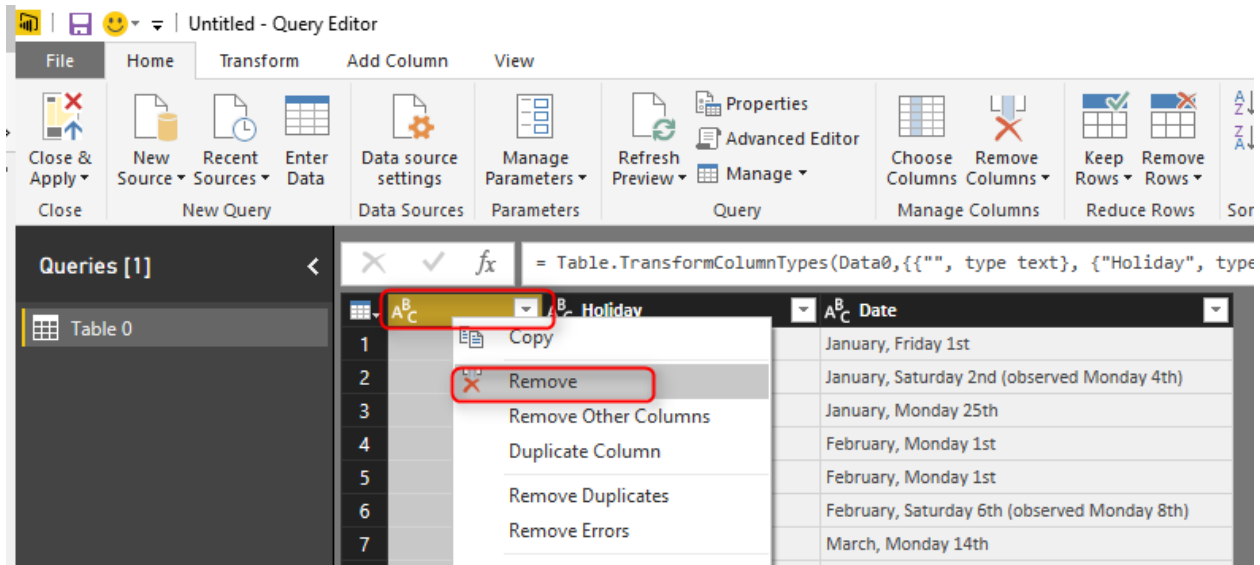
Table View Web View

Table 0

	Holiday	Date
null	New Year's Day	January, Friday 1st
null	Day after New Year's Day	January, Saturday 2nd (observed Monday 4th)
null	Wellington Anniversary	January, Monday 25th
null	Auckland Anniversary	February, Monday 1st
null	Nelson Anniversary	February, Monday 1st
null	Waitangi Day	February, Saturday 6th (observed Monday 8th)
null	Taranaki Anniversary	March, Monday 14th
null	Otago Anniversary	March, Monday 21st
null	Good Friday	March, Friday 25th
null	Easter Monday	March, Monday 28th
null	Easter Tuesday ?	March, Tuesday 29th
null	Southland Anniversary	March, Tuesday 29th
null	Daylight Saving ends	April, Sunday 3rd
null	ANZAC Day	April, Monday 25th
null	Queen's Birthday	June, Monday 6th
null	Daylight Saving starts	September, Sunday 25th
null	South Canterbury Anniversary	September, Monday 26th
null	Hawke's Bay Anniversary	October, Friday 21st
null	Labour Day	October, Monday 24th
null	Marlborough Anniversary	October, Monday 31st
null	Canterbury Anniversary	November, Friday 11th

Load Edit Cancel

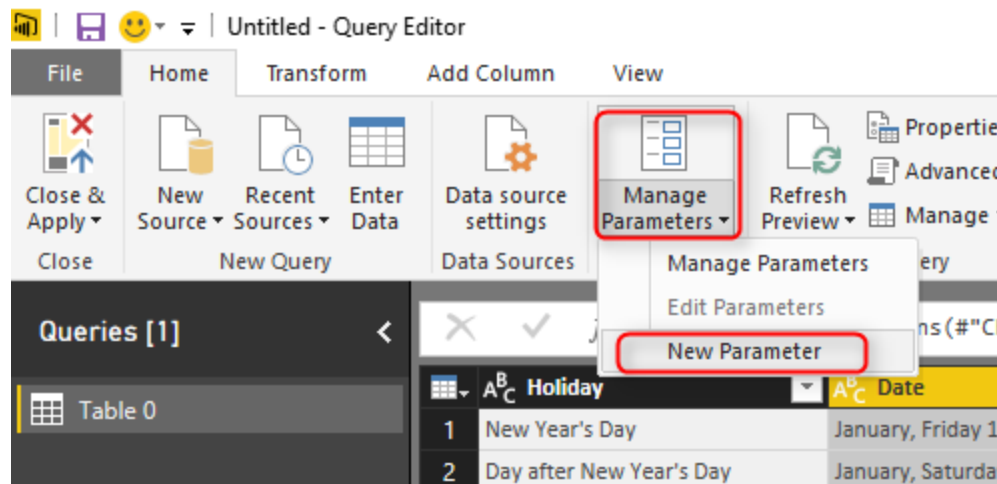
This will open Query Editor Window for you, You can now make some changes in the query, For example remove the first column.



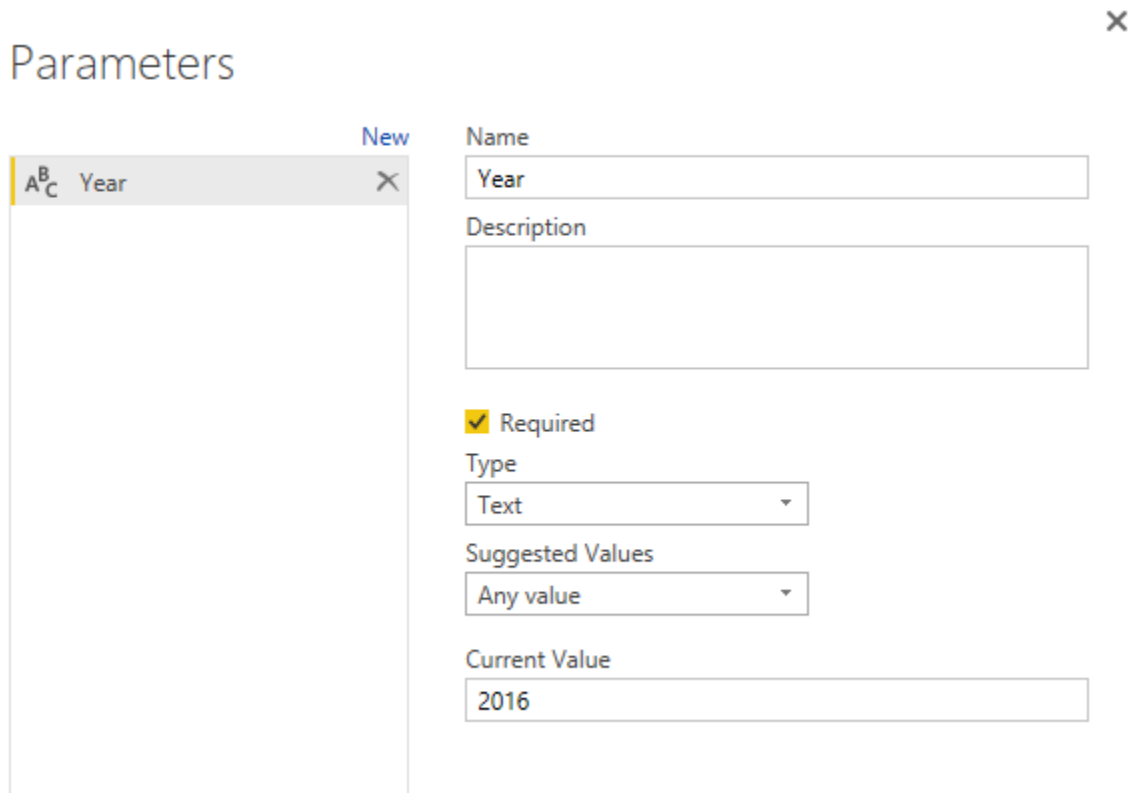
First column was an empty column which we removed. Now you can see two columns; Holiday (which is description of holiday), and Date. Date column is Text data type, and it doesn't have year part in it. If you try to convert it to Date data type you will either get an error in each cell or incorrect date as a result (depends on locale setting of your computer). To convert this text to a date format we need to bring a Year value in the query. The year value for this query can be statistically set to 2016. But because we want to make it dynamic so let's use a Parameter. This Parameter later will be used as input of the query.

Parameter Definition

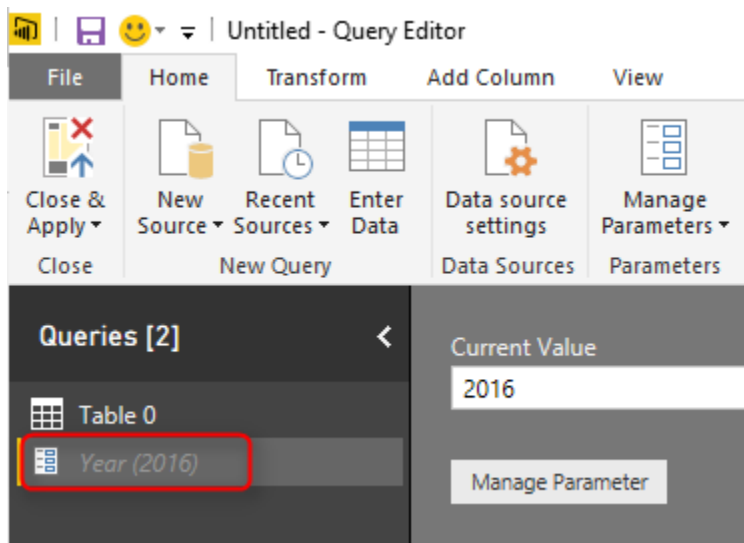
Parameters are ways to pass values to other queries. Normally for custom functions you need to use parameters. Click on Manage Parameters menu option in Query Editor, and select New Parameter.



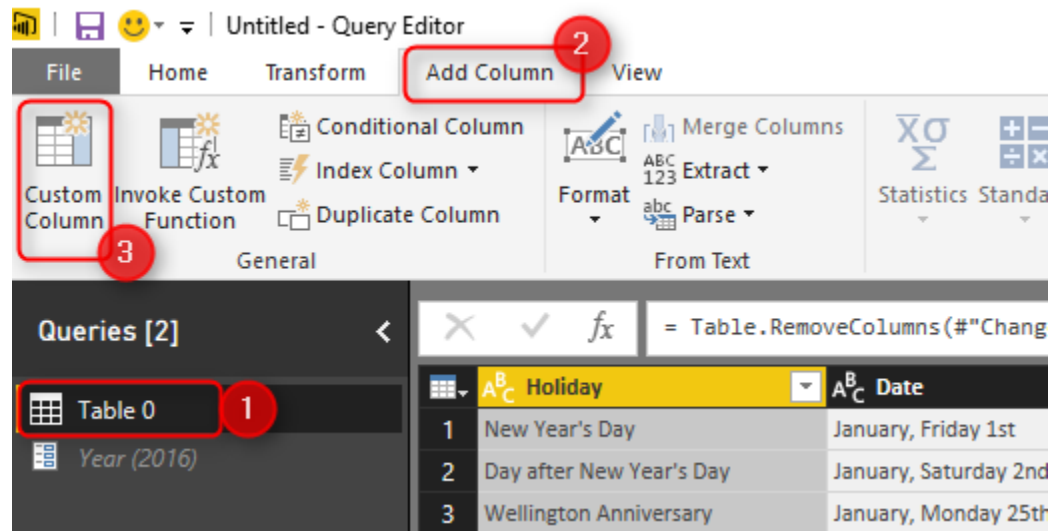
There are different types of parameters you can use, but to keep it simple, create a parameter of type Text, with all default selections. set the Current Value to be 2016. and name it as Year.



After creating the Parameter you can see that in Queries pane with specific icon for parameter.



Now we can add a column in Table 0 with the value from this parameter. Click on Table 0, and from Add Column menu option, click on Add Custom Column.



Name the Custom column as Year, and write the expression to be equal Year (remember names in Power Query are case sensitive)

Add Custom Column

New column name

Year

Custom column formula:

= Year

Now you can see year value added to the table. I have also changed its data type to be Text

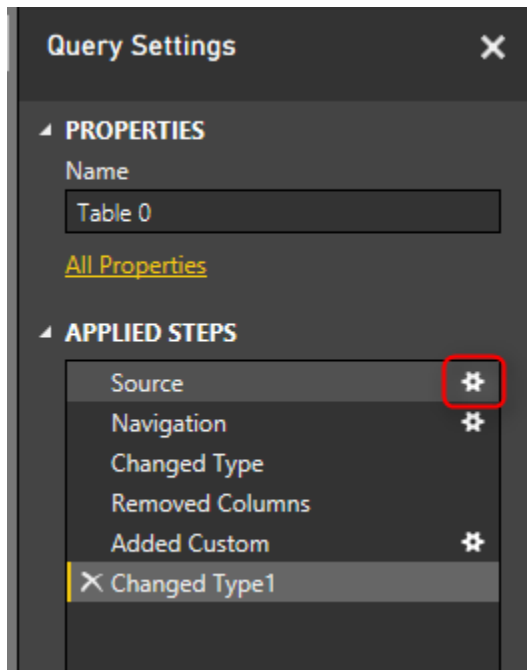
	A ^B _C Holiday	A ^B _C Date	A ^B _C Year
1	New Year's Day	January, Friday 1st	2016
2	Day after New Year's Day	January, Saturday 2nd (observed Monday 4th)	2016
3	Wellington Anniversary	January, Monday 25th	2016
4	Auckland Anniversary	February, Monday 1st	2016
5	Nelson Anniversary	February, Monday 1st	2016
6	Waitangi Day	February, Saturday 6th (observed Monday 8th)	2016
7	Taranaki Anniversary	March, Monday 14th	2016
8	Otago Anniversary	March, Monday 21st	2016
9	Good Friday	March, Friday 25th	2016
10	Easter Monday	March, Monday 28th	2016
11	Easter Tuesday ?	March, Tuesday 29th	2016
12	Southland Anniversary	March, Tuesday 29th	2016
13	Daylight Saving ends	April, Sunday 3rd	2016
14	ANZAC Day	April, Monday 25th	2016
15	Queen's Birthday	June, Monday 6th	2016
16	Daylight Saving starts	September, Sunday 25th	2016
17	South Canterbury Anniversary	September, Monday 26th	2016
18	Hawke's Bay Anniversary	October, Friday 21st	2016
19	Labour Day	October, Monday 24th	2016

Now that we have created Parameter we can use that parameter as an input to the query's source URL as well.

URL Parameterization

One of the main benefits of Parameters is that you can use that in a URL. In our case, the URL which contains 2016 as the year, can be dynamic using this parameter. Also for converting a query to a custom function using parameters is one of the main steps. Let's add Parameter in the source of this query then;

While Table 0 query is selected, in the list of Steps, click on Setting icon for Source step



This will bring the very first step of the query where we get data from web and provided the URL. Not in the top section change the From Web window to Advanced.

From Web

☐ Basic ☒ Advanced

Enter a Web page URL.

URL parts

Add Part

URL preview

<http://publicholiday.co.nz/nz-public-holidays-2016.html>

Open file as

Html Page

Command timeout in minutes (optional)

HTTP Request Header Parameters (optional)

Type or select a value

Add Header

OK

Cancel

Advanced option gives you the ability to split the URL into portions. What we want to do is to put Text portions for beginning and end of string, and make the year part of it dynamic coming from URL. So Add another part, and put setting as below;

From Web

☐ Basic ☒ Advanced

Enter a Web page URL.

URL parts

...

URL preview

`http://publicholiday.co.nz/nz-public-holidays-{Year}.html`

Configuration above means that in the first part of URL we put everything before 2016 which is:

`http://publicholiday.co.nz/nz-public-holidays-`

Second part of URL is coming from parameter, and we use Year parameter for that

third part of URL is the remaining part after the year which is:

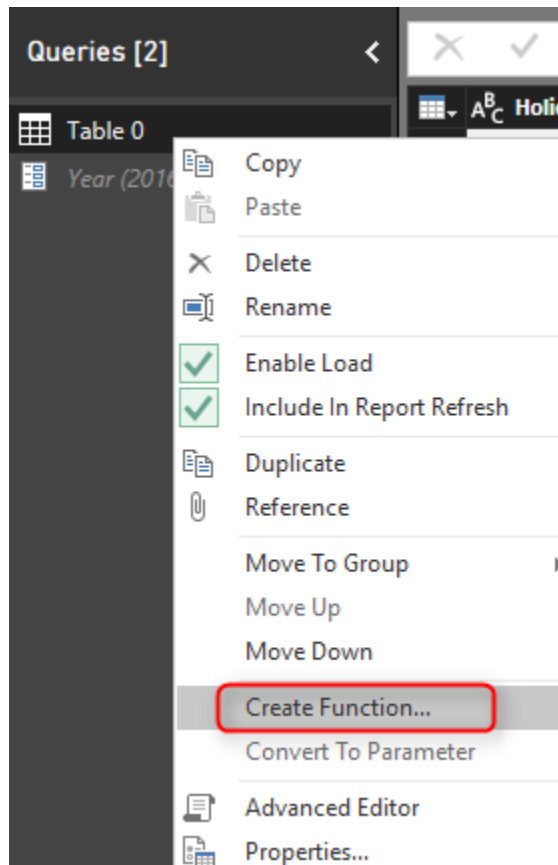
`.html`

altogether these will make the URL highlighted above which instead of {Year} it will have 2016, or 2015 or other values.

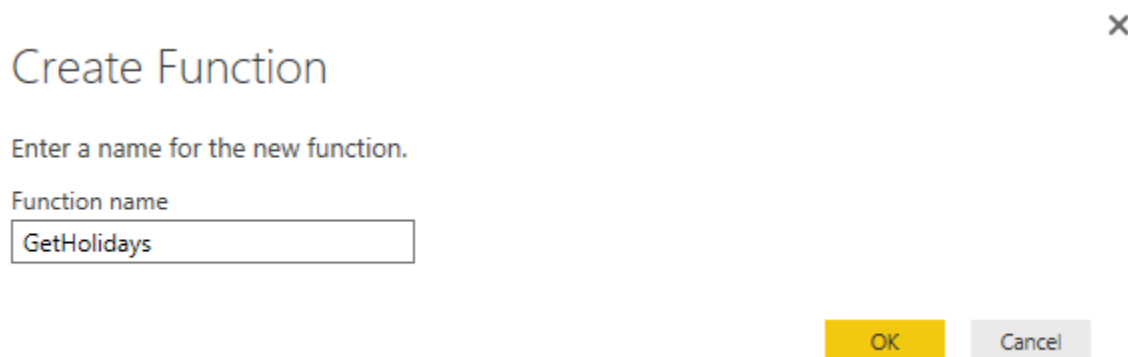
Click on OK. You won't see any changes yet, even if you click on the last step of this query, because we have used same year code as the parameter value. If you change the parameter value and refresh the query you will see changes, but we don't want to do it in this way.

Convert Query to Function

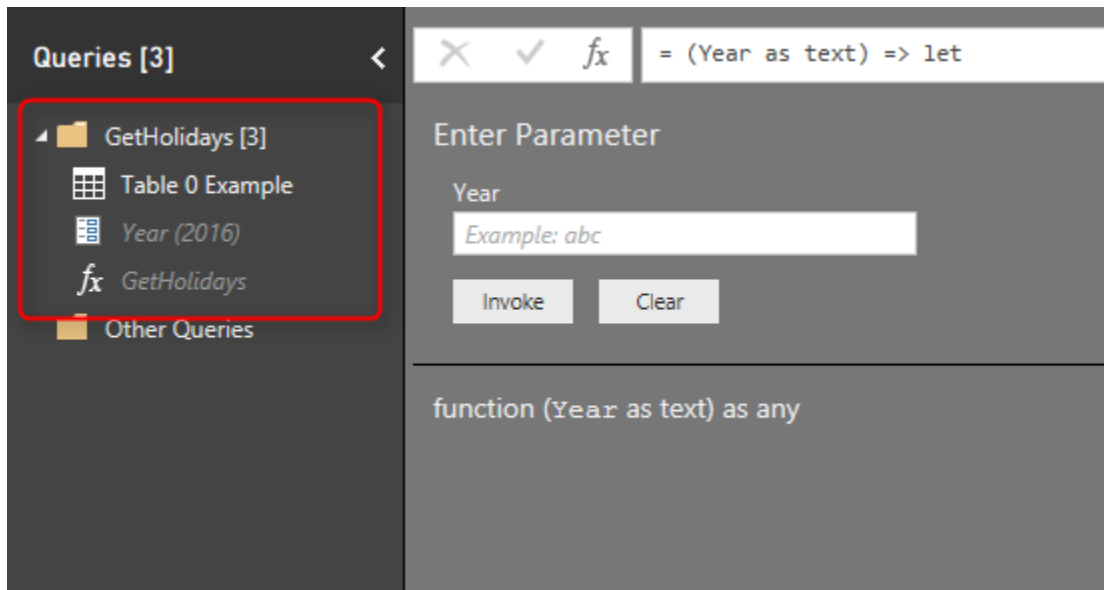
After using parameter in the source of query we can convert it to function. Right click on the Table 0 query and select Create Function.



Name the function as GetHolidays and click on OK.



You will now see a group (folder) created with name of GetHolidays including 3 objects; main query (Table 0), Parameter (year), and function (GetHolidays).



The function itself marked with *fx* icon and that is the function we will use to call from other queries. However the main query and parameter are still necessary for making changes in the function. I will explain this part later.

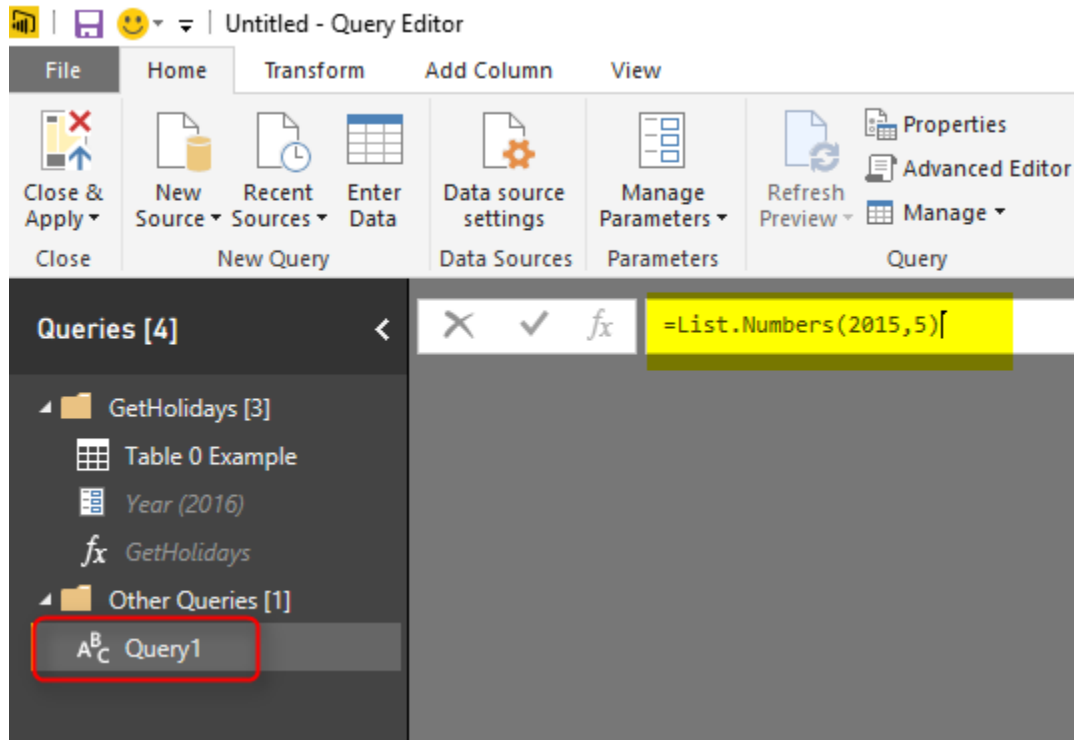
All happened here is that there is a copy of the Table 0 query created as a function. and every time you call this function with an input parameter (which will be year value) this will give you the result (which is public holidays table for that year). Let's now consume this table from another query, but before that let's create a query that includes list of years.

Using Generator

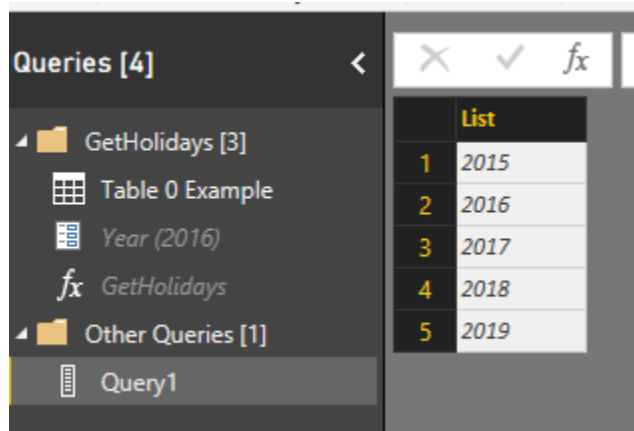
Generators are a topic of its own and can't be discussed in this post. All I can tell you for now is that Generators are functions that generate a list. This can be used for creating loop structure in Power Query. I'll write about that in another post. For this example I want to create a list of numbers from 2015 for 5 years. So I'll use List.Numbers generator function for that. In your Query Editor Window, create a New Source from Home tab, and choose Blank Query.

This will create a Query1 for you. Click on Query1 in Queries pane, and in the Formula bar type in below script:

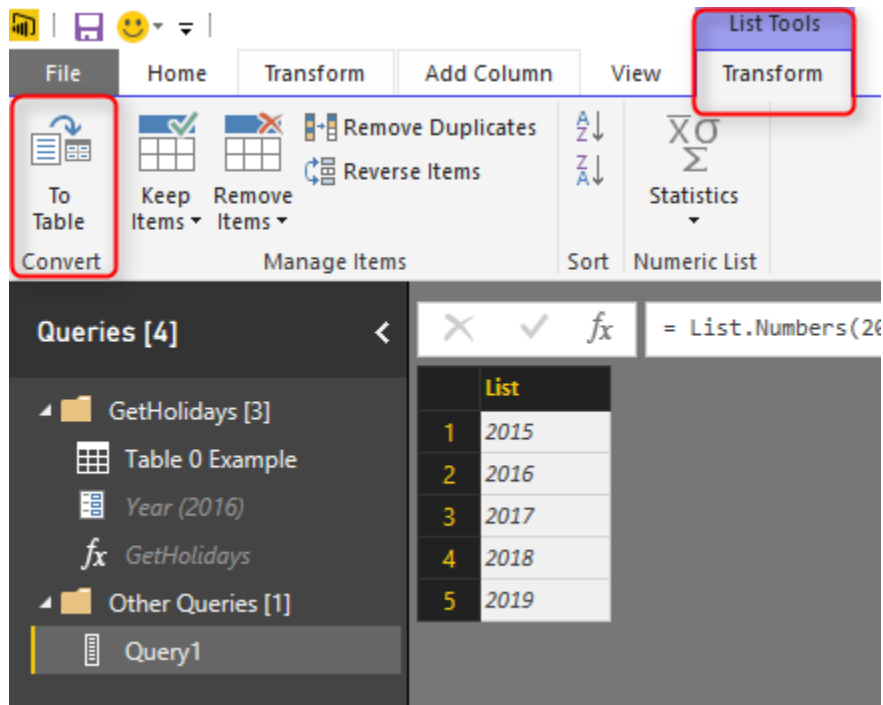
```
= List.Numbers(2015,5)
```



After entering the expression press Enter and you will see a list generated from 2015 for 5 numbers. That's the work done by a generator function.



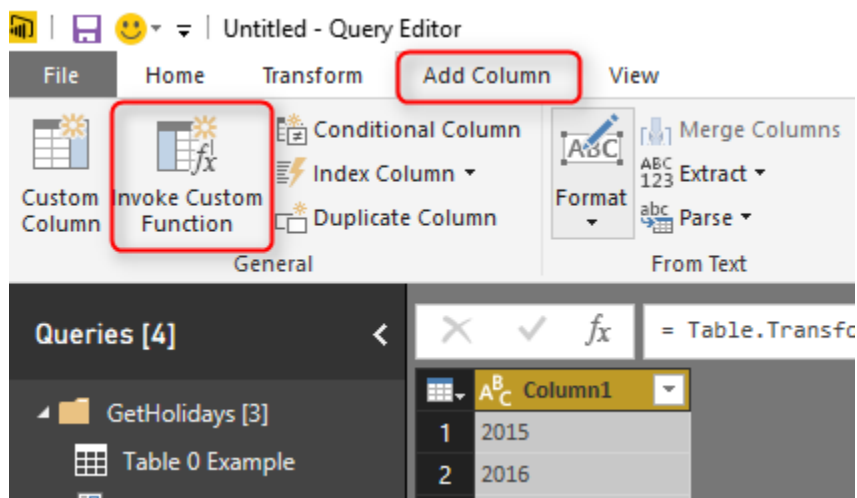
This is a List, and can be converted to Table simple from List Tools menu option.




Convert this to Table with all default settings, and now you can a table with Column1 which is year value. Because the value is whole number I have changed it to Text as well (to match the data type of parameter).

Consuming Function

It is easy to consume a function in Query Editor from a table. Go to Add Columns, and click on Invoke Custom Function option.



In the Invoke Custom Function window, choose the function (named GetHolidays), the input parameter is from the table column name Column1, and name the output column as Holidays.



Invoke Custom Function

Invoke a custom function defined in this file for each row.

New column name

Function query

Year

Now when you click on OK, you will see a new column added with a table in each cell. These tables are results of calling that function with the input parameter which is value of Column1 in each row. If you click on a blank area of a cell with Table hyperlink, you will see the table structure below it.

	Column1	Holidays
1	2015	Table
2	2016	Table
3	2017	Table
4	2018	Table
5	2019	Table

Holiday	Date	Year
New Year's Day	January, Sunday 1st (observed Tuesday 3rd)	2017
Day after New Year's Day	January, Monday 2nd	2017
Wellington Anniversary	January, Monday 23rd	2017
Auckland Anniversary	January, Monday 30th	2017
Nelson Anniversary	January, Monday 30th	2017
Waitangi Day	February, Monday 6th	2017
Taranaki Anniversary	March, Monday 13th	2017
Otago Anniversary	March, Monday 20th	2017
Daylight Saving ends	April, Sunday 2nd	2017
Good Friday	April, Friday 14th	2017
Easter Monday	April, Monday 17th	2017
Easter Tuesday ?	April, Tuesday 18th	2017
Southland Anniversary	April, Tuesday 18th	2017
ANZAC Day	April, Tuesday 25th	2017
Queen's Birthday	June, Monday 5th	2017
Daylight Saving starts	September, Sunday 24th	2017
South Canterbury Anniversary	September, Monday 25th	2017
Hawke's Bay Anniversary	October, Friday 20th	2017
Labour Day	October, Monday 23rd	2017
Marlborough Anniversary	October, Monday 30th	2017

Interesting, isn't it? It was so easy. All done from GUI, not a single line of code to run this function or pass parameters, things made easy with custom functions.

Editing Function

If you want to make modifications in function, you can simply modify the main query (which is Table 0 Example). For example let's create a full date format from that query in this way;

Click on Table 0 Example query and split the Date column with delimiter Comma, you will end up having a column now for Month values and another for Day (Note that I have renamed this columns respectively);

	A ^B _C Holiday	A ^B _C Month	A ^B _C Day	A ^B _C Year
1	New Year's Day	January	Friday 1st	2016
2	Day after New Year's Day	January	Saturday 2nd (observed Monday 4th)	2016
3	Wellington Anniversary	January	Monday 25th	2016
4	Auckland Anniversary	February	Monday 1st	2016
5	Nelson Anniversary	February	Monday 1st	2016
6	Waitangi Day	February	Saturday 6th (observed Monday 8th)	2016
7	Taranaki Anniversary	March	Monday 14th	2016
8	Otago Anniversary	March	Monday 21st	2016
9	Good Friday	March	Friday 25th	2016
10	Easter Monday	March	Monday 28th	2016
11	Easter Tuesday ?	March	Tuesday 29th	2016
12	Southland Anniversary	March	Tuesday 29th	2016
13	Daylight Saving ends	April	Sunday 3rd	2016
14	ANZAC Day	April	Monday 25th	2016
15	Queen's Birthday	June	Monday 6th	2016
16	Daylight Saving starts	September	Sunday 25th	2016
17	South Canterbury Anniversary	September	Monday 26th	2016
18	Hawke's Bay Anniversary	October	Friday 21st	2016
19	Labour Day	October	Monday 24th	2016
20	Marlborough Anniversary	October	Monday 31st	2016
21	Canterbury Anniversary	November	Friday 11th	2016
22	Westland Anniversary	November	Monday 28th	2016
23	Chatham Islands Anniversary	November	Monday 28th	2016
24	Christmas Day	December	Sunday 25th (observed Tuesday 27th)	2016
25	Boxing Day	December	Monday 26th	2016

Now if you go back to Query1, you will see changes in the results table immediately.

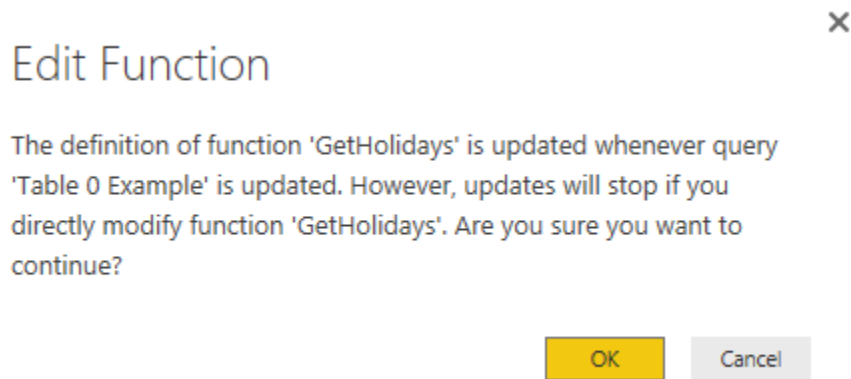
	Column1	Holidays
1	2015	Table
2	2016	Table
3	2017	Table
4	2018	Table
5	2019	Table

Holiday	Month	Day	Year
New Year's Day	January	Sunday 1st (observed Tuesday 3rd)	2017
Day after New Year's Day	January	Monday 2nd	2017
Wellington Anniversary	January	Monday 23rd	2017
Auckland Anniversary	January	Monday 30th	2017
Nelson Anniversary	January	Monday 30th	2017
Waitangi Day	February	Monday 6th	2017
Taranaki Anniversary	March	Monday 13th	2017
Otago Anniversary	March	Monday 20th	2017
Daylight Saving ends	April	Sunday 2nd	2017
Good Friday	April	Friday 14th	2017
Easter Monday	April	Monday 17th	2017
Easter Tuesday ?	April	Tuesday 18th	2017
Southland Anniversary	April	Tuesday 18th	2017
ANZAC Day	April	Tuesday 25th	2017
Queen's Birthday	June	Monday 5th	2017
Daylight Saving starts	September	Sunday 24th	2017
South Canterbury Anniversary	September	Monday 25th	2017
Hawke's Bay Anniversary	October	Friday 20th	2017
Labour Day	October	Monday 23rd	2017
Marlborough Anniversary	October	Monday 30th	2017

Limitations

Edit Script for the Function

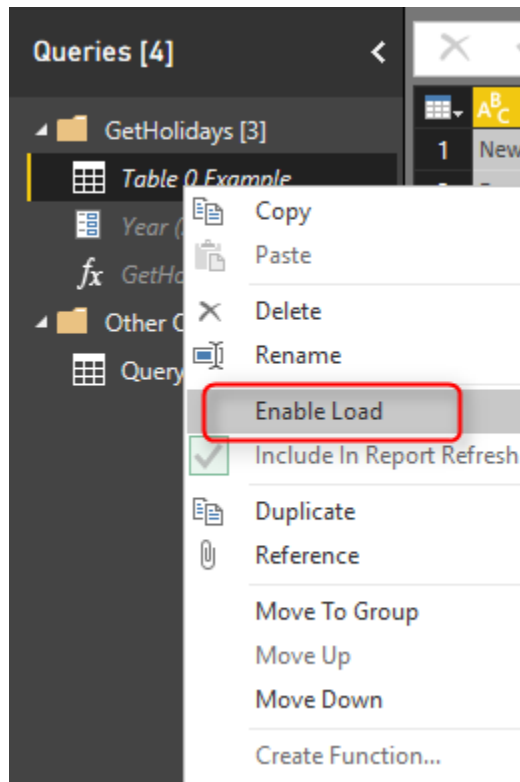
Changes are fine as long as you don't want to edit the M script of the function, if you want to do so, then the function definition and the query definition split apart. and you will get below message:



It is alright to make changes in the advanced Editor of the source query, and then the function will be updated based on that, but if you want to change the function itself, then the query will be separated.

Disable Load of the Source Query

If you have read my [blog post about Enable Load in Power Query](#), you already know that queries that is not used in the model should not be loaded. By Default the source query (in this example named as Table 0 Example) will be loaded into the model. This means one extra table, and consuming more memory. So remember to uncheck the Enable Load for this query;

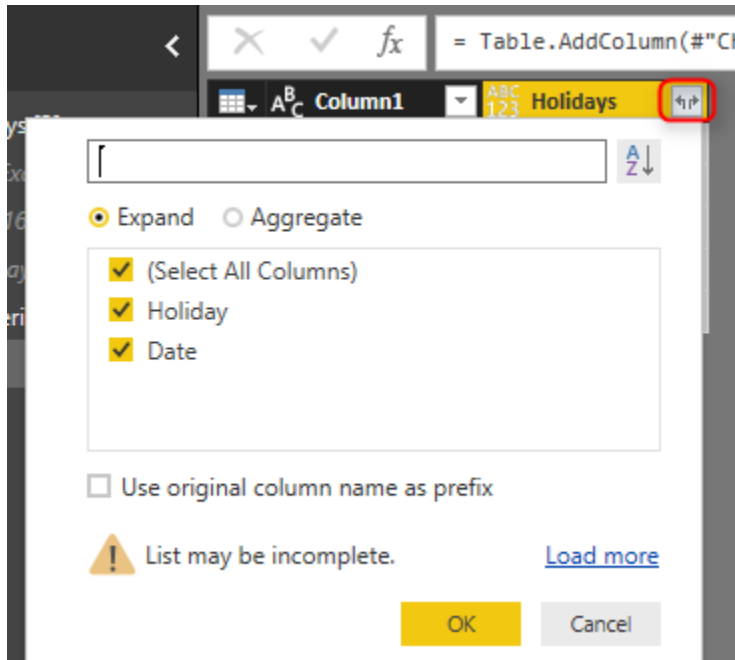


Parameterized URLs

Custom Functions that uses parameterized URLs (like this example) cannot be scheduled to refresh in Power BI. That's a big limitation which I hope be lifted very quickly.

The Example at the End

I have done some other changes, and changed the data type to Date format. Now the final query which is expanded table from all underlying queries include all public holidays. I'll leave that part to you to continue and build rest of the example. For that functionality you need to use Expand;



and here is the final result;

	Holiday	Date
47	Westland Anniversary	11/28/2016
48	Chatham Islands Anniversary	11/28/2016
49	Christmas Day	12/27/2016
50	Boxing Day	12/26/2016
51	New Year's Day	1/3/2017
52	Day after New Year's Day	1/2/2017
53	Wellington Anniversary	1/23/2017
54	Auckland Anniversary	1/30/2017
55	Nelson Anniversary	1/30/2017
56	Waitangi Day	2/6/2017
57	Taranaki Anniversary	3/13/2017
58	Otago Anniversary	3/20/2017
59	Daylight Saving ends	4/2/2017
60	Good Friday	4/14/2017
61	Easter Monday	4/17/2017
62	Easter Tuesday ?	4/18/2017

Summary

In this post you have learned how easy is to create a custom function from a query, all from graphical interface. You have seen that I haven't wrote any single line of code to do it. and all happened through GUI. You have seen that you can change definition of function simply by changing the source query. and you also have seen how easy is to call/consume a function from another query. This method can be used a lot in real world Power Query scenarios.

Day Number of Year, Power Query Custom Function

Published Date: July 28, 2015

```

let
    DayNumberOfYear = (date) =>
        let
            dated = try DateTime.FromText(date),
            month = Date.Month(dated[Value]),
            MonthList = List.Numbers(1, month - 1),
            year = Date.Year(dated[Value]),
            TransformedMonthList = List.Transform(
                MonthList,
                each Text.From(year) & "-" & Text.From(_) & "-1"
            ),
            DateList = List.Transform(
                TransformedMonthList,
                each DateTime.FromText(_)
            ),
            DaysList = List.Transform(
                DateList,
                each Date.DaysInMonth(_)
            )
        in
            if dated[HasError]
            then dated[Error]
            else List.Sum(DaysList) + Date.Day(dated[Value])
in
    DayNumberOfYear("07/28/2015")

```

There are number of Date and DateTime built-in functions in Power Query which are really helpful. There is also a function for DayNumberOfYear. However I've thought it would be a good example to go through writing a function that uses Generators, Each singleton function, and error handling all inside a custom function. Through this post you will also learn;

- how to create Custom Function
- how to use Generators as loop structure
- and how to user Error Handling.

Let's consider this date as today's date : 28th of July of 2015 (this is actually the date of this blog post)

There might be number of methods to calculate the day number of year for this date (which is 209). I just use one of them here. Steps are as below;

- fetch number of days for each month from January of this year (of the date above) to previous month (of the date above).
- Calculate sum of values above will give me the number of days in all month prior than this month.
- add the day number of the date to the calculated sum above.

Some of calculations can be helped through with Power Query Date functions. So let's start;

1 – Create a function in Power Query called DayNumberOfYear as below

If you don't know where to write below code:

- *Open Excel, Go to Power Query Tab, Click On Get Data from Source, Blank Query, In the Query Editor window go to View tab, and click on Advanced Editor.*
- *Open Power BI, click on Edit Queries, In the Query Editor window go to View tab, and click on Advanced Editor.*

```
let
    //start of the code
    DayNumberOfYear = (date) => //Function name, and input parameter
    let
        //start of function body
        dated = DateTime.FromText(date) //date conversion from text
    in
        //start of function output
        dated //function body output
in
    //start of output lines generated
    DayNumberOfYear("07/28/2015") //call function by a value
```

I've put some comments in above script to help you understand each line. In general DayNumberOfYear is name of the function. It accepts an input parameter "date". and convert the parameter from text value to DateTime. the last line of the code calls the function with specific date ("07/28/2015").

**** Note that Date Conversion function is locale dependent. So if the date time of your system is no MM/DD/YYYY then you have to enter date as it formatted in your system (look below the clock on right hand side bottom of your monitor to check the format).**

the result of above script will be:

07/28/2015 12:00:00 a.m.

2 – Fetch Month Number and generate a list of all prior months.

Fetching month number is easily possible with `Date.Month` function. the remaining part is looping through months from January of this year to previous month (of the given date). Unfortunately there is no loop structure in Power Query M language yet, but fortunately we can use Generator functions for that. Generator function is a function that produce/generate a list based on some parameters. For example you can generate a list of dates from a start date, based on given occurrence of a period of time. or you can generate a list of numbers. For this example we want to generate list of numbers, starting from 1 (month January) to the current month number minus 1 (previous month).

Here is the code:

```
let
    DayNumberOfYear= (date) =>
        let
            dated=DateTime.FromText(date),
            month=Date.Month(dated),//month number
            MonthList=List.Numbers(1,month-1) // generate list of months from Jan to previous month
        in
            MonthList
in
    DayNumberOfYear("07/28/2015")
```

The result is a list of month numbers as below:

	List
1	1
2	2
3	3
4	4
5	5
6	6

Generator Function used in above code is `List.Numbers`. this function generate list of numbers starting from a value.

3 – Transform list to full date list

We have to calculate number of days for each month in the function. Number of days in each month can be fetched by `Date.DaysInMonth` function. However this function accept a full `DateTime` data type, and the value that we have in our list members are text. so we have to produce a datetime value from it. for generating a full date we need the year portion as well, we use `Date.Year` function to fetch that.

Here is the code to transform the list:

```
let
    DayNumberOfYear= (date) =>
        let
            dated=DateTime.FromText(date),
            month=Date.Month(dated),
            MonthList=List.Numbers(1,month-1),
            year=Date.Year(dated), // fetch year
            TransformedMonthList=List.Transform // transform list
                (MonthList,
                each Text.From(year)&"-"&Text.From(_)&"-1") // generate full text value
        in
            TransformedMonthList
in
    DayNumberOfYear("07/28/2015")
```

The result is:

	List
1	2015-1-1
2	2015-2-1
3	2015-3-1
4	2015-4-1
5	2015-5-1
6	2015-6-1

As you see the above result is not still of datetime data type, we've only generated full date as text value. Now we can convert values to datetime data type

Here is the code:

```
let
    DayNumberOfYear= (date) =>
        let
            dated=DateTime.FromText(date),
            month=Date.Month(dated),
            MonthList=List.Numbers(1,month-1),
            year=Date.Year(dated),
            TransformedMonthList=List.Transform
                (MonthList,
                each Text.From(year)&"-"&Text.From(_)&"-1"),
            DateList=List.Transform(
                TransformedMonthList,
                each DateTime.FromText(_)//transform to DateTime value
            )
        in
            DateList
```

```

        DateList
in
    DayNumberOfYear("07/28/2015")

```

and the result:

	List
1	1/1/2015 12:00:00 AM
2	2/1/2015 12:00:00 AM
3	3/1/2015 12:00:00 AM
4	4/1/2015 12:00:00 AM
5	5/1/2015 12:00:00 AM
6	6/1/2015 12:00:00 AM

4 – Transform the list to List of DaysNumberOfMonths

We use DaysInMonth function to fetch number of days in each month from the list. here is the code:

```

let
    DayNumberOfYear= (date) =>
        let
            dated=DateTime.FromText(date),
            month=Date.Month(dated),
            MonthList=List.Numbers(1,month-1),
            year=Date.Year(dated),
            TransformedMonthList=List.Transform
                (MonthList,
                 each Text.From(year)&"-"&Text.From(_)&"-1"),
            DateList=List.Transform(
                TransformedMonthList,
                each DateTime.FromText(_)),
            DaysList=List.Transform(
                DateList,
                each Date.DaysInMonth(_))
        in
            DaysList
in
    DayNumberOfYear("07/28/2015")

```

and the result:

	List
1	31
2	28
3	31
4	30
5	31
6	30

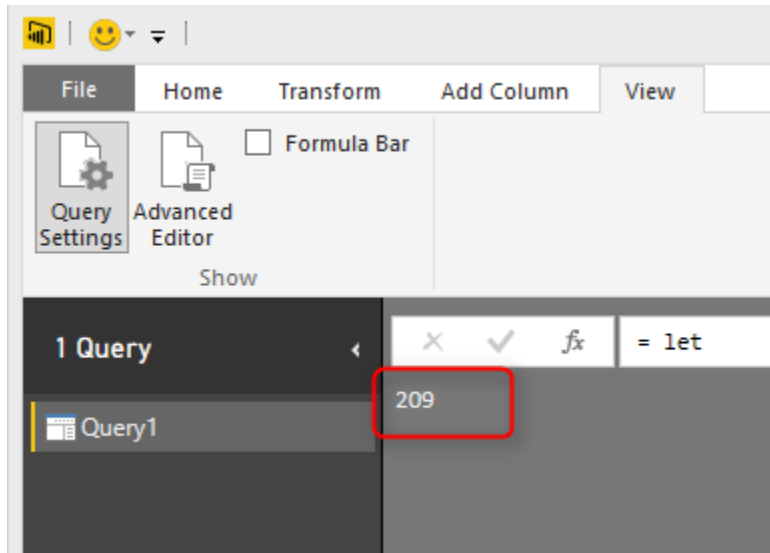
5 – Calculate Sum of Dates and Add day number of this month to it

The list is ready to use, we need to sum it up only. And then add the current day Date.Day from the given date to it.

Here is the code:

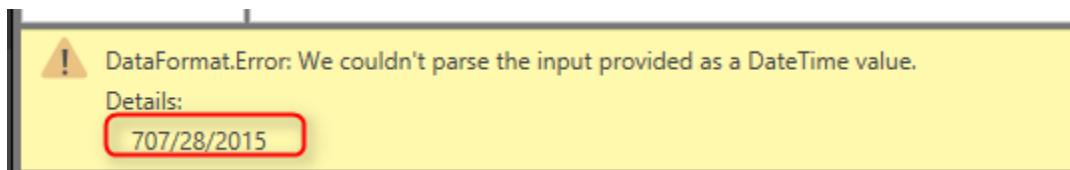
```
let
    DayNumberOfYear= (date) =>
        let
            dated=DateTime.FromText(date),
            month=Date.Month(dated),
            MonthList=List.Numbers(1,month-1),
            year=Date.Year(dated),
            TransformedMonthList=List.Transform
                (MonthList,
                 each Text.From(year)&"-"&Text.From(_)&"-1"),
            DateList=List.Transform(
                TransformedMonthList,
                each DateTime.FromText(_)),
            DaysList=List.Transform(
                DateList,
                each Date.DaysInMonth(_))
        in
            List.Sum(DaysList)//sum of values in the list
            +Date.Day(dated)//current date's day number
in
    DayNumberOfYear("07/28/2015")
```

and the result is:



6- Error Handling

The function is working, but if the given date format be wrong the we will face and error such as below:



So let's add few lines of error handling to the code. We can simply use Try clause to the code as below:

```
let
    DayNumberOfYear= (date) =>
        let
            dated= try DateTime.FromText(date),
            month=Date.Month(dated[Value]),
            MonthList=List.Numbers(1,month-1),
            year=Date.Year(dated[Value]),
            TransformedMonthList=List.Transform
                (MonthList,
                 each Text.From(year)&"-"&Text.From(_)&"-1"),
            DateList=List.Transform(
                TransformedMonthList,
                each DateTime.FromText(_)),
            DaysList=List.Transform(
                DateList,
                each Date.DaysInMonth(_))
        in
```

```

        if dated[HasError]
            then dated[Error]
            else List.Sum(DaysList)+Date.Day(dated[Value])
    in
        DayNumberOfYear("707/28/2015")

```

Result for bad formatted given date is :

fx = let	
Reason	DataFormat.Error
Message	We couldn't parse the input provided as a DateTime value.
Detail	707/28/2015

Here is the full code of the script:

```

let
    DayNumberOfYear= (date) =>
        let
            dated= try DateTime.FromText(date),
            month=Date.Month(dated[Value]),
            MonthList=List.Numbers(1,month-1),
            year=Date.Year(dated[Value]),
            TransformedMonthList=List.Transform
                (MonthList,
                 each Text.From(year)&"-"&Text.From(_)&"-1"),
            DateList=List.Transform(
                TransformedMonthList,
                each DateTime.FromText(_)),
            DaysList=List.Transform(
                DateList,
                each Date.DaysInMonth(_))
        in
            if dated[HasError]
                then dated[Error]
                else List.Sum(DaysList)+Date.Day(dated[Value])
    in
        DayNumberOfYear("707/28/2015")

```

In this post you've learned:

- A Function that calculate Day Number of Year for a given date
- Creating Custom Function
- using Generators as Loop structure

- Error Handling

Power Query Function that Returns Multiple Values

Published Date: July 30, 2015

```

let
    FirstAndLastDayOfTheMonth = (date) =>
        let
            dated=Date.FromText(date) ,
            year=Date.Year(dated) ,
            month=Date.Month(dated) ,
            FirstDateText=Text.From(year)&"-"&Text.From(month)&"-01" ,
            FirstDate=Date.FromText(FirstDateText) ,
            daysInMonth=Date.DaysInMonth(dated) ,
            LastDateText=Text.From(year)&"-"&Text.From(month)&"-"&Text.From(daysInMonth) ,
            LastDate=Date.FromText(LastDateText) ,
            record=Record.AddField([], "First Date of Month", FirstDate) ,
            resultset=Record.AddField(record, "Last Date of Month", LastDate)
        in
            resultset
in
    FirstAndLastDayOfTheMonth("30/07/2015")

```

Yesterday in [NZ BI user group meeting](#), I've been asked that does Power Query custom functions return only one value as the result set? or they can return multiple values. I've answered Yes, and I've explained that through a sentence how to do it with Records, List, or Table. Then I thought this might be a question of many people out there. So I've wrote this blog post to illustrate how to return multiple values from a custom function in Power Query.

If you don't know how to create a custom function, please read my other [blog post with an example of Day Number of Year function for Power Query](#). In this post I'll show you through an example of how to return multiple results from a Power Query function.

As you probably know Power Query function return single value by default, and that is the value result of the operation in the **"in"** clause of the function. Now how to return multiple values? Simply by returning different type of object. The trick is that Power Query custom function can return any single object. and that **object** can be simple structure object such as Date, Text, Number. Or it can be multiple value objects such as Record, List, and Table.

To understand difference between Record, List, and Table:

- Record : Is a single record structure with one or more fields. each field have a field name, and a field value.
- List: Is a single column structure with one or more rows. each row contains a value.
- Table: Multiple rows and columns data structure (as you probably all know it 😊)

Above objects can hold multiple values. So the only thing you need to do is return one of above objects based on your requirement. In example below I've returned a Record as a result set, but you can do it with other two data types.

Return First and Last Dates of Month

As an example I would like to write a function that fetches both first and last date of a month, the input parameter of this function is a date value with text data types, such as "30/07/2015".

**** Note that Date Conversion function is locale dependent. So if the date time of your system is not DD/MM/YYYY then you have to enter date as it formatted in your system (look below the clock on right hand side bottom of your monitor to check the format).**

Let's start by calculation of First Date of the Month

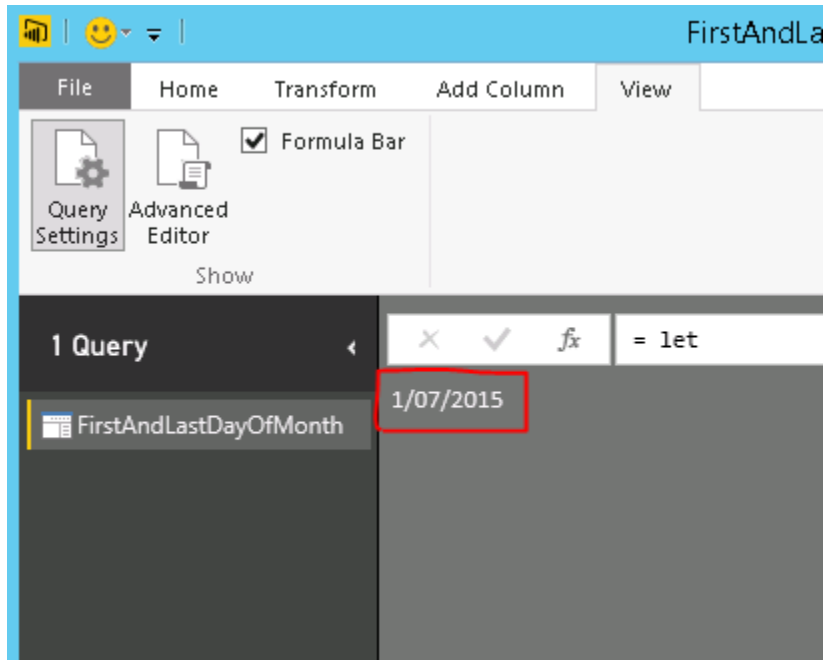
First Date of the Month

We need to fetch the year, and the month, and then build a date string for first day (day 1) of that month and year, and finally convert it to Date datatype.

Here is the script:

```
let
    FirstAndLastDayOfTheMonth = (date) => //function definition
        let
            dated=Date.FromText(date),//convert input text to Date
            year=Date.Year(dated),//fetch year
            month=Date.Month(dated),//fetch month
            FirstDateText=Text.From(year)&"-"&Text.From(month)&"-01",//generate text value of the first date
            FirstDate=Date.FromText(FirstDateText)//convert text value to date
        in
            FirstDate//return result of the function
in
    FirstAndLastDayOfTheMonth("30/07/2015")//function call
```


and the Result:



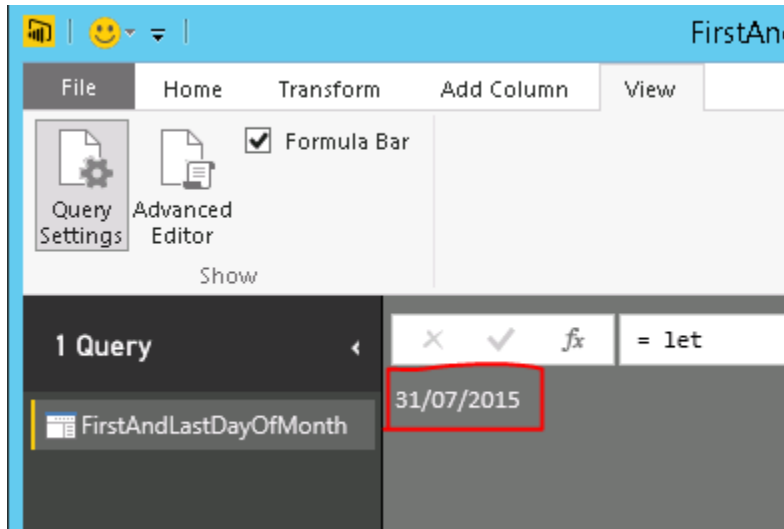
Last Date of the Month

For fetching last date of the month, we use same method of first date, except one change. which is the day part of the calculation should be number of days in the month, which comes from Date.DaysInMonth function.

Here is the script:

```
let
    FirstAndLastDayOfTheMonth = (date) =>
        let
            dated=Date.FromText(date),
            year=Date.Year(dated),
            month=Date.Month(dated),
            FirstDateText=Text.From(year)&"-"&Text.From(month)&"-01",
            FirstDate=Date.FromText(FirstDateText),
            daysInMonth=Date.DaysInMonth(dated),//fetch number of days in month
            LastDateText=Text.From(year)&"-"&Text.From(month)&"-"&Text.From(daysInMonth),
            LastDate=Date.FromText(LastDateText)
        in
            LastDate
    in
        FirstAndLastDayOfTheMonth("30/07/2015")
```

and the result:



Combining both values into a Record and returning Record as a Result

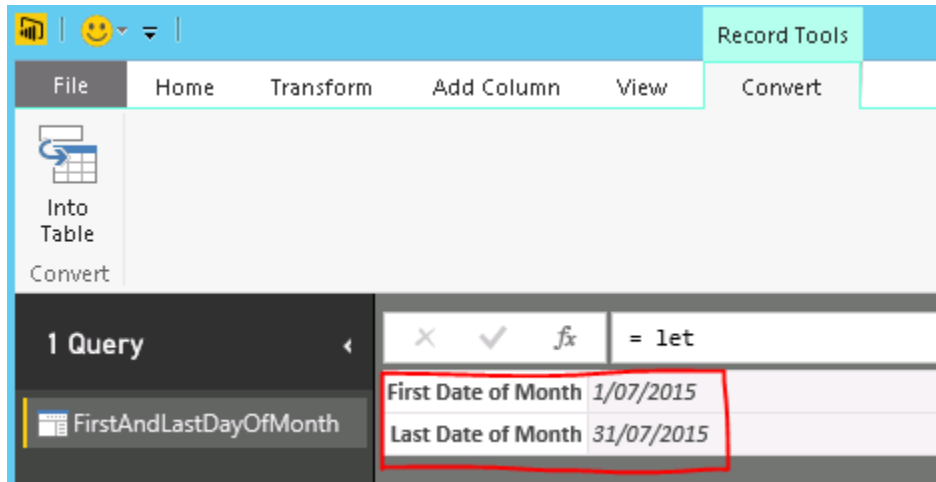
Now we have both values and we want to return them both. I'll create an empty record first. Empty record can be created simply with this : `{}` .

Then I used `Record.AddField` function to add fields one by one. `Record.AddField` gets three parameters: the record that field will be added to it, name of the new field, and value of the new field.

Here is the script:

```
let
    FirstAndLastDayOfTheMonth = (date) =>
        let
            dated=Date.FromText(date),
            year=Date.Year(dated),
            month=Date.Month(dated),
            FirstDateText=Text.From(year)&"-"&Text.From(month)&"-01",
            FirstDate=Date.FromText(FirstDateText),
            daysInMonth=Date.DaysInMonth(dated),
            LastDateText=Text.From(year)&"-"&Text.From(month)&"-"&Text.From(daysInMonth),
            LastDate=Date.FromText(LastDateText),
            record=Record.AddField({}, "First Date of Month", FirstDate),
            resultset=Record.AddField(record, "Last Date of Month", LastDate)
        in
            resultset
in
    FirstAndLastDayOfTheMonth("30/07/2015")
```

and the result:

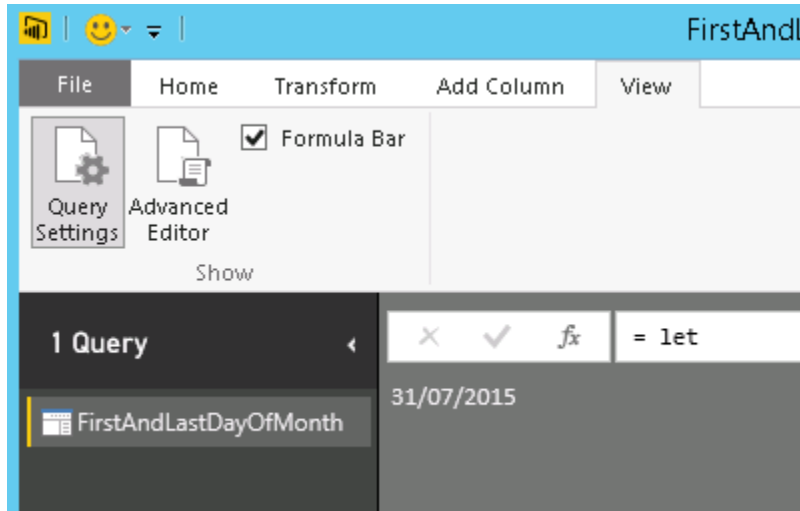


How to access this Record's values

As you see in above the function returns a record with two fields. now we can access fields by name, with script like this:

```
FirstAndLastDayOfTheMonth("30/07/2015")[Last Date of Month]
```

and the result would be a single value



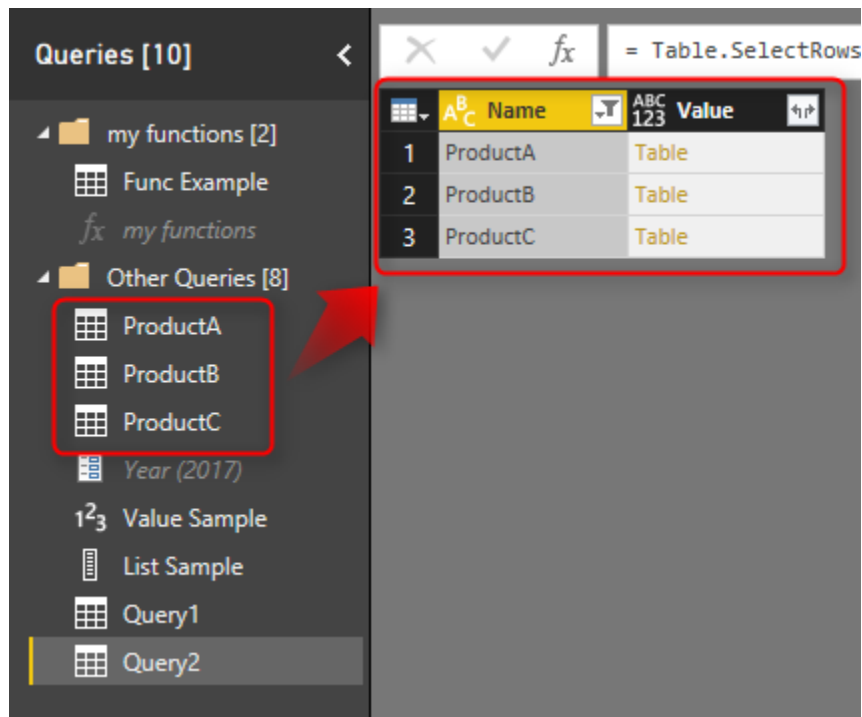
Here is the full code of example if you want to try it yourself.

Don't Limit Yourself

You can have another record in a field's value, you can have a list in a field's value, and you can have a table in a field's value. so you can create any data structure that you want as the result set of your function.

Get List of Queries in Power BI

Published Date: January 16, 2017

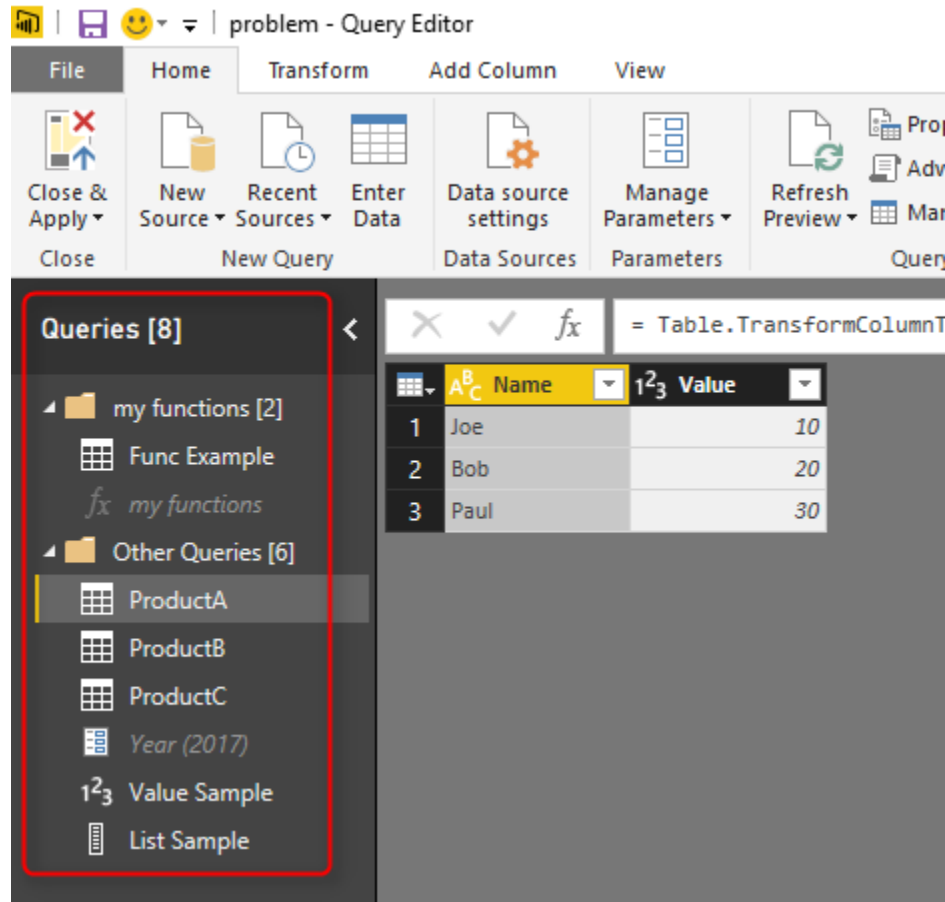


Power Query is the component of getting data in Power BI. But have you used Power Query to get metadata of the Power BI queries itself? In this post I'll show you a quick and simple way of using Power Query to get metadata (name of queries and the data in queries) from all queries in Power BI. I have previously explained how to use #shared keyword to get list of all functions in Power Query, this post shows how to use #shared or #sections to get all queries (and parameters, and functions, and lists...) from Power BI. If you want to learn more about Power BI; read [Power BI online book from Rookie to Rock Star](#).

* Thanks to [Alex Arvidsson](#) who brought this question, and was the cause of writing this blog post.

Question: How to Fetch Name of All Queries into One Query?

Consider below Power BI file, that has functions, parameters, and queries. Queries also returns tables, values, and lists;



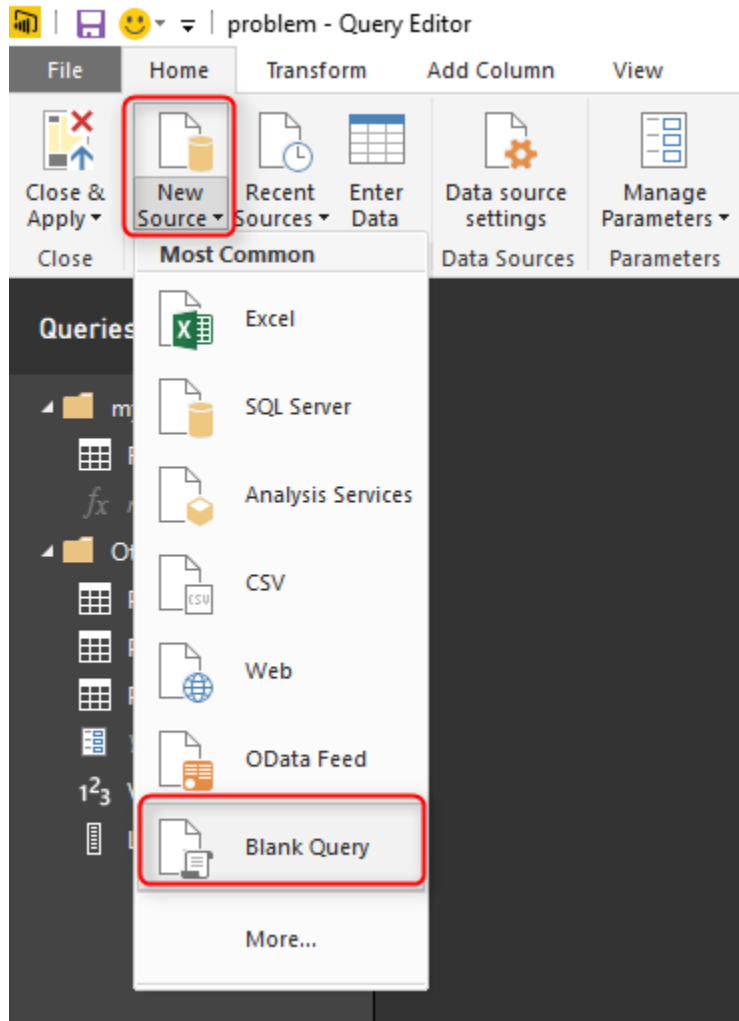
The question is how I can get list of all these queries and their values as a new query?
Let's see the answer

Answer: Using #shared or #sections Keywords

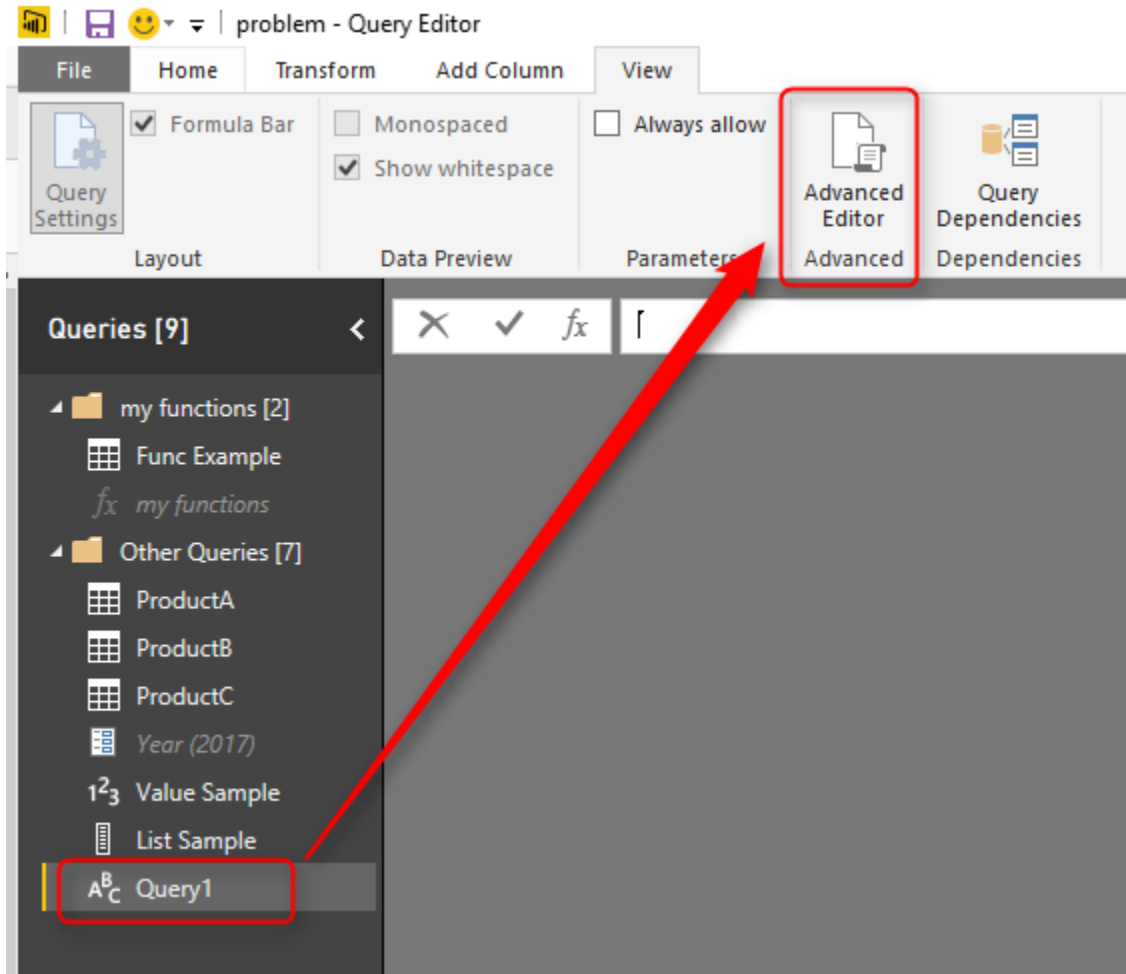
#shared to Get Current File's Queries Plus All Functions

I have previously explained [what #shared keyword does](#); it is a keyword that returns list of all functions, and enumerators in Power Query. It can be used as a document library in the Power Query itself. It will also fetch all queries in the existing Power BI file. Here is how you can use it:

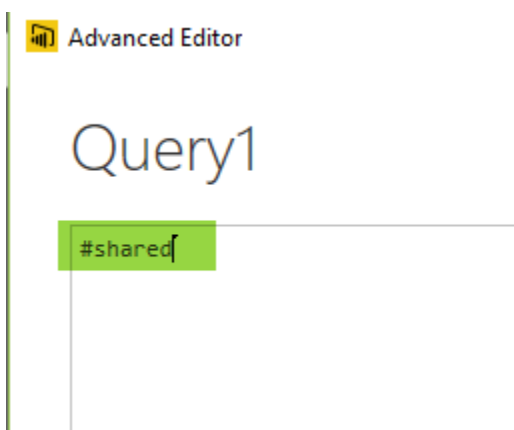
Create a New Source, from Blank Query.



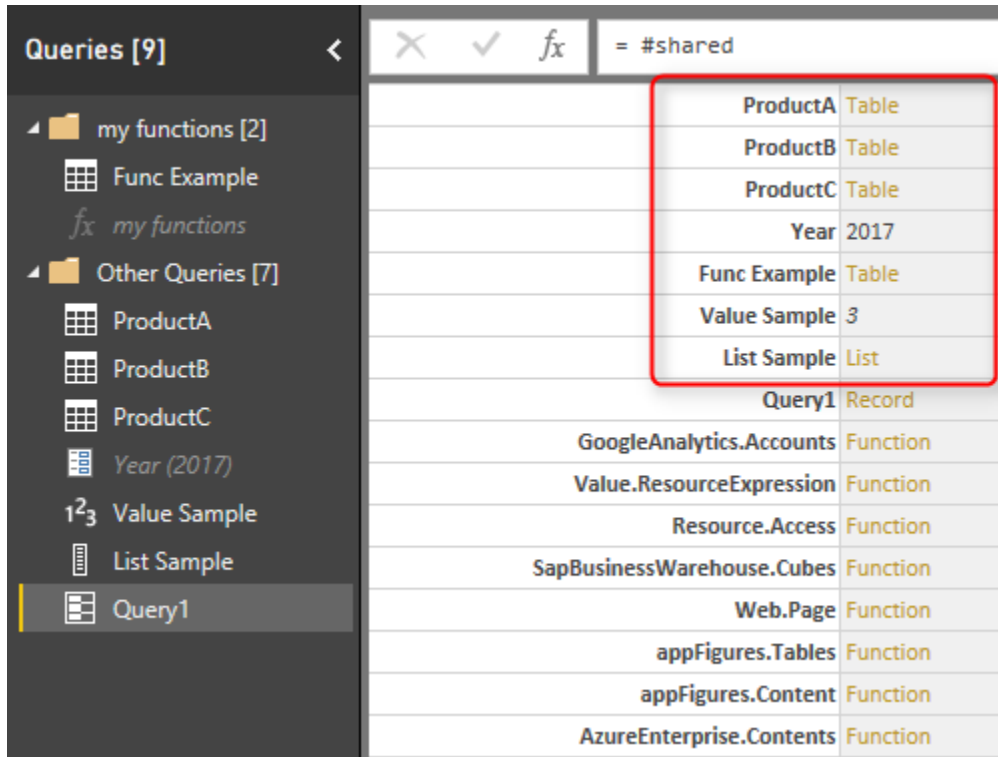
Then go to Advanced Editor of that query (from Home tab or View tab)



write only one keyword in the Advanced Editor: **#shared**



And Click on Done. Result will comes up quickly;

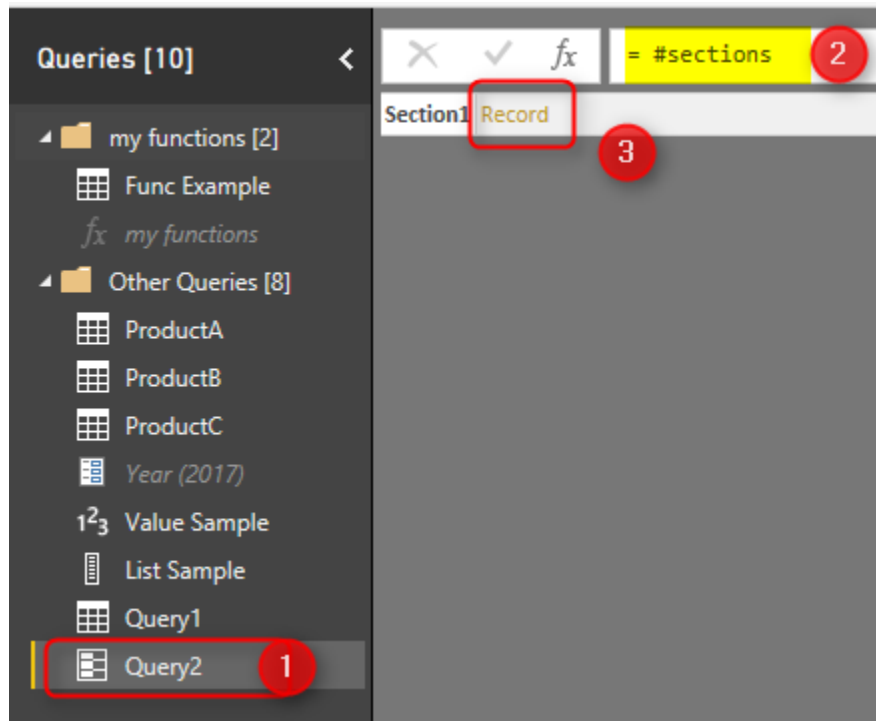


Query Name	Type
ProductA	Table
ProductB	Table
ProductC	Table
Year	2017
Func Example	Table
Value Sample	3
List Sample	List
Query1	Record
GoogleAnalytics.Accounts	Function
Value.ResourceExpression	Function
Resource.Access	Function
SapBusinessWarehouse.Cubes	Function
Web.Page	Function
appFigures.Tables	Function
appFigures.Content	Function
AzureEnterprise.Contents	Function

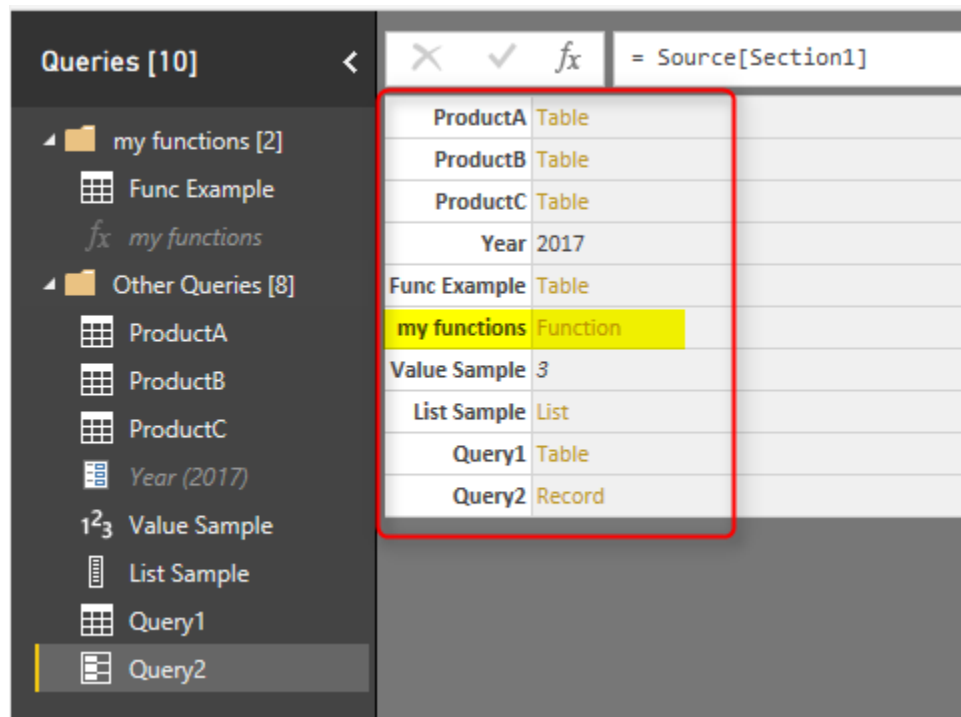
You can see in the result that #shared fetch all existing queries, plus all built-in functions in Power Query. the section which is marked in above result set in where you can find queries from the current file. Note that the Query1 itself (which includes the #shared keyword) is listed there. the limitation of this method is that it won't return your custom function; "my function" in this example. the next method however would pick that as well.

#sections To Get Current File's Queries

The other way of fetching list of queries is using #sections keyword. #sections keyword will give you list of all sections in Power Query (This post isn't right place to explain what sections are, but for now, just consider every query here as a section). so same method this time using #sections will return result below;

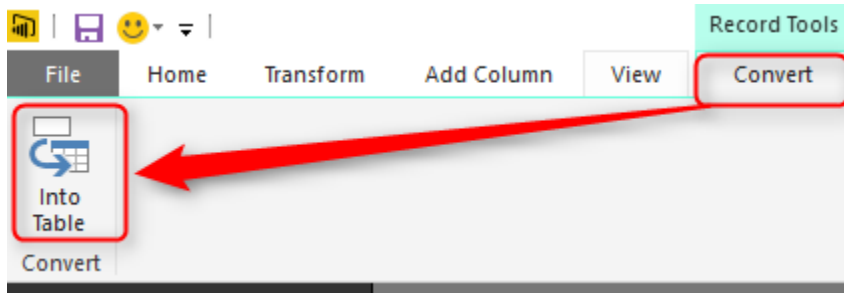


Result is a record, that you can simply click on the Record to see what are columns in there. Columns are queries in the current file:

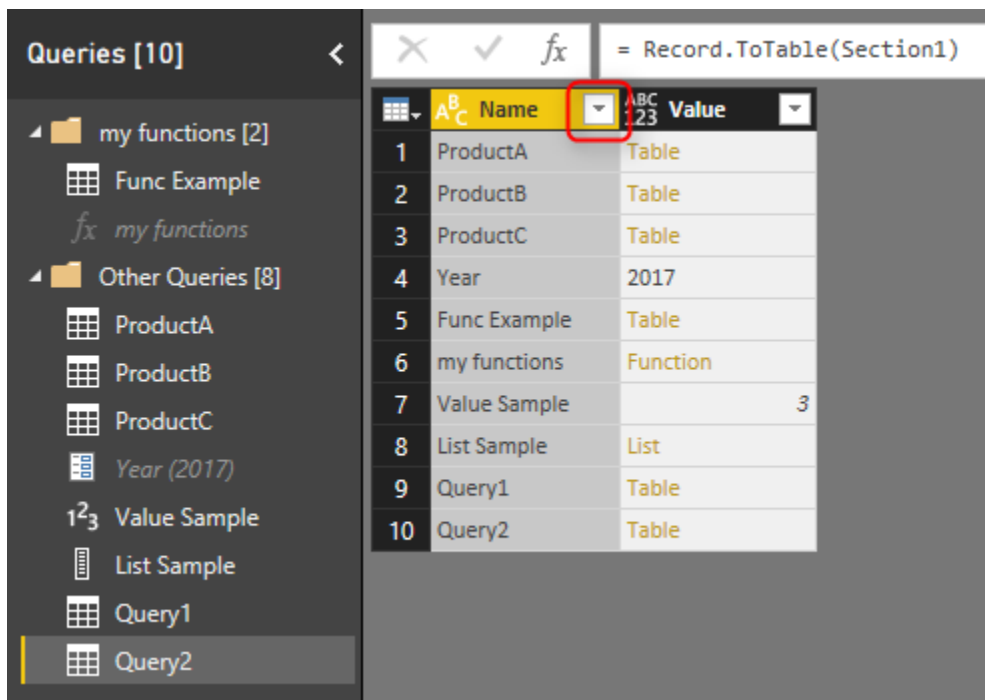


This method also returns functions in the current file, which previous method with #shared didn't. So this is a better method if you are interested to fetch function's names as well.

The result is a record which can be converted to a table (this gives you better filtering options in the GUI). You can find the Convert Into Table under Record Tools Convert Section menu.



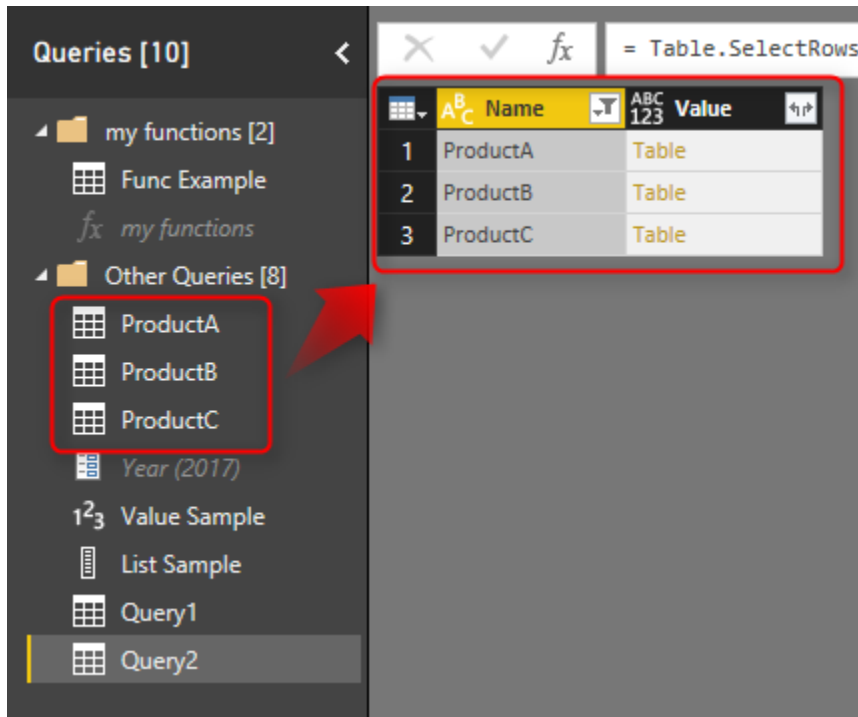
After having this as a table, then you can apply any filters you want;



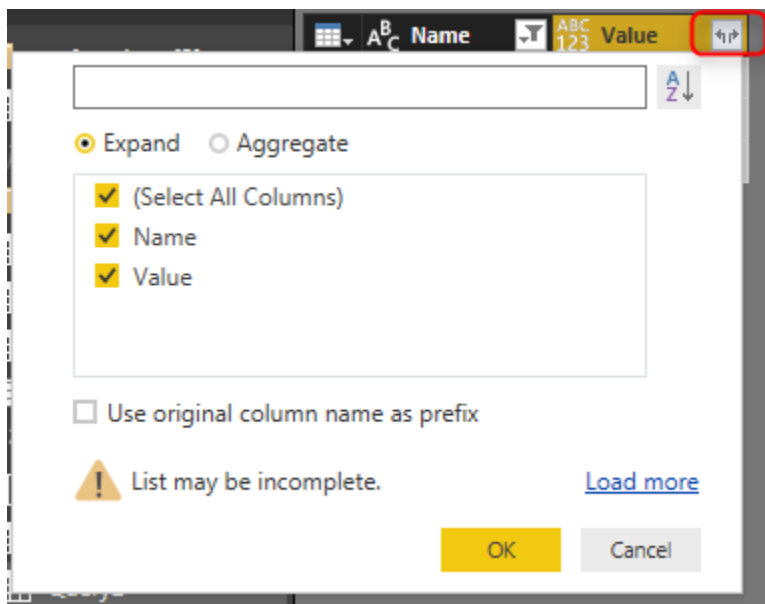
Sample Scenario of Usage

There are many usages of getting name of queries in another query. One sample usage of that can be getting different queries coming from different places, and the only way to identify the source is query name. In this case instead of manually adding a column to

each query and then combining them together, you can use this method to get the query name dynamically. In screenshot below ProductA, ProductB, and ProductC are coming as source queries, and you can simply do a filtering to get them all with their product names.



And you can expand it to tables underneath if you want to (this would work if they all have same data structure);



and as a result you have combined result of all queries, with the query name as another column;

	A ^B C Name	ABC 123 Name.1	ABC 123 Value.1
1	ProductA	Joe	10
2	ProductA	Bob	20
3	ProductA	Paul	30
4	ProductB	Joe	30
5	ProductB	Bob	60
6	ProductB	Paul	90
7	ProductC	Joe	1
8	ProductC	Bob	2
9	ProductC	Paul	3

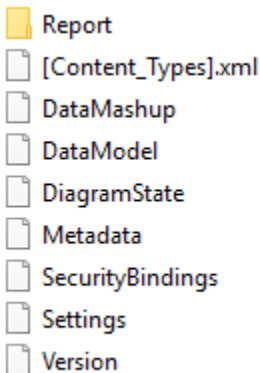
Published Date: June 7, 2017

Power BI Desktop file stores many metadata information in it, one of the most important piece of information stored there is Power Query M scripts. It is always easy to open a PBIX file and realize the queries in Query Editor Window. However, if you have many queries and you want a way to get the script of all of them then this post is for you. I will explain a bit about the structure of PBIX file at the beginning and then I'll explain how you can access the M script from it. If you like to learn more about Power BI and Power Query; read [Power BI book from Rookie to Rock Star](#).

[Gilbert](#) recently written a great blog post about the structure of PBIX file, and I strongly recommend everyone to read it. For this blog post, my focus is on M script section of that. so, I'll go through other parts of the structure briefly. A Power BI Desktop file is a file with *.PBIX extension. This file is a renamed version of a ZIP file. You can simply rename the file to *.zip to see the structure of it. Then you can simply unzip it into a folder.



Files under this folder are;



Here is a very brief explanation of each file or folder;

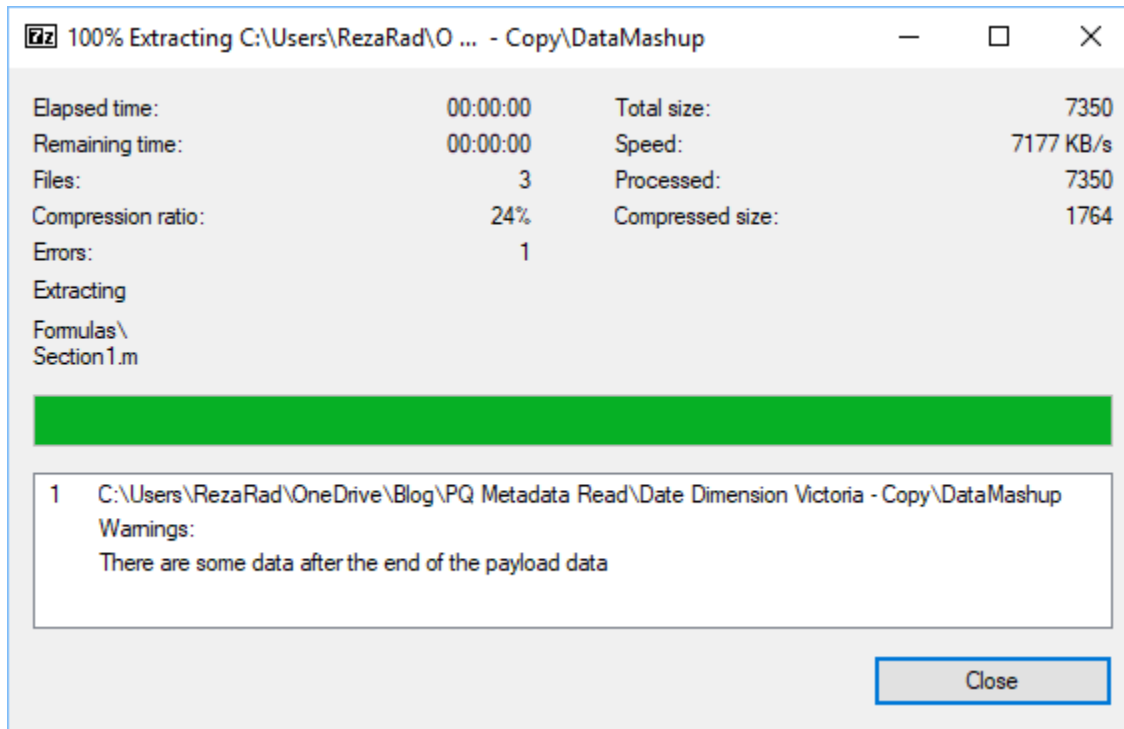
- Report Folder: Layout of the Power BI visual reports (Including two files in the folder)
- [Content_Types].xml: Includes the content structure of this folder in XML format (older PBIX files might have different content structure)
- DataMashup: This is the file which is main topic of this blog post. It includes everything related to Power Query side of the file; M scripts, structure of queries, parameters, and functions.
- DataModel: This actually includes the data in the model (in a compressed format).
- DiagramState: It seems it stores the table and Matrix information.
- Metadata: Contains all labels (Names). In Power BI Desktop, everything has a GUID assigned to it. what we see as a table name, has a GUID assigned to it. This file is mapping of GUIDs and names that we see.
- SecurityBindings: I believe this is Row Level Security configuration in the Power BI stored in binary format.
- Settings: I believe it is some of the settings applied in Power BI settings menu options.
- Version: Version of the file.

in the future I'll write some other posts with detailed explanation of every file and folder. For now, let's focus on DataMashup File.

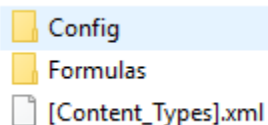
DataMashup File: Everything You Need

Talking about Power Query; DataMashup file is all you need. It includes everything from the structure of queries, tables, parameters, list, to the actual M scripts behind the scene. You can Fetch all of these information from this single file. Let's look at the structure of

this file. If you open this file with a text editor, you will see some binary things first (which are related to the zipped nature of this file), and also some XML information. Yes, this is a zipped file. Let's start with unzipping it into a folder. I've done that with 7-zip application.



In the folder you will have these files:



Here are details of each file;


14. [Content_Types].xml: explains the content of this section, very high level information. it is just saying that M script are somewhere in this folder

```
<?xml version="1.0" encoding="utf-8"?>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
  <Default Extension="xml" ContentType="text/xml" />
  <Default Extension="m" ContentType="application/x-ms-m" />
</Types>
```


21. Config Folder: Includes one file: Package.xml, which is the information about settings of Power Query. For example the culture, versioning, and some other configurations.

```
<?xml version="1.0" encoding="utf-8"?>
<Package xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Version>2.46.4732.721</Version>
  <MinVersion>1.5.3296.0</MinVersion>
  <Culture>en-NZ</Culture>
</Package>
```

- Formulas Folder: Includes M scripts! files with *.m extension.

OneDrive > Blog > PQ Metadata Read > Date Dimension Victoria - Copy > DataMashup~ > Formulas				
Name	Date modified	Type	Size	
 Section1.m	30/05/2017 6:08 PM	M File	7 KB	

You might end up having more than one file if you have more than 1 section, but this is unlikely to happen in usual cases. Usually everything you do in Query Editor goes under single section. If you are a M geek and know how to write multiple sections, then you might end up with having multiple files 😊 I will write another post in the future explaining Sections and their usages. For now, let's look at this *.m file with a text editor:

M Script File

Section1.m File gives you all M script in well formatted text;

```

section Section1;

shared Sheet1 = let
    Source = Excel.Workbook(File.Contents("C:\Users\RezaRad\OneDrive\2-Advanced Power Query_RADACAD\Demo\Book1.xlsx"), null, true),
    Sheet1_Sheet = Source{[Item="Sheet1",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(Sheet1_Sheet, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{("Start Year", Int64.Type), ("Financial Month Start", Int64.Type), ("Years to Generate", Int64.Type)})
in
    #"Changed Type";

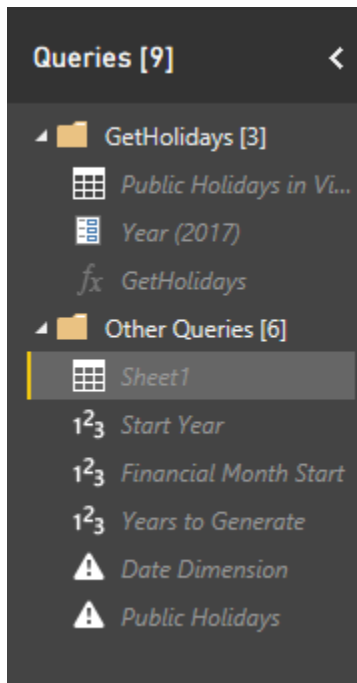
shared #"Start Year" = let
    Source = Excel.Workbook(File.Contents("C:\Users\RezaRad\OneDrive\2-Advanced Power Query_RADACAD\Demo\Book1.xlsx"), null, true),
    Sheet1_Sheet = Source{[Item="Sheet1",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(Sheet1_Sheet, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{("Start Year", Int64.Type), ("Financial Month Start", Int64.Type), ("Years to Generate", Int64.Type)}),
    #"Start Year1" = #"Changed Type"{0}[Start Year]
in
    #"Start Year1";

shared #"Financial Month Start" = let
    Source = Excel.Workbook(File.Contents("C:\Users\RezaRad\OneDrive\2-Advanced Power Query_RADACAD\Demo\Book1.xlsx"), null, true),
    Sheet1_Sheet = Source{[Item="Sheet1",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(Sheet1_Sheet, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{("Start Year", Int64.Type), ("Financial Month Start", Int64.Type), ("Years to Generate", Int64.Type)}),
    #"Financial Month Start1" = #"Changed Type"{0}[Financial Month Start]
in
    #"Financial Month Start1";

shared #"Years to Generate" = let
    Source = Excel.Workbook(File.Contents("C:\Users\RezaRad\OneDrive\2-Advanced Power Query_RADACAD\Demo\Book1.xlsx"), null, true),
    Sheet1_Sheet = Source{[Item="Sheet1",Kind="Sheet"]}[Data],
    #"Promoted Headers" = Table.PromoteHeaders(Sheet1_Sheet, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{("Start Year", Int64.Type), ("Financial Month Start", Int64.Type), ("Years to Generate", Int64.Type)}),
    #"Years to Generate1" = #"Changed Type"{0}[Years to Generate]
in
    #"Years to Generate1";

```

If you need to send your M scripts to someone this is the place to copy them from. This includes every single M script all in one place (as long as they are all under one section). Here is the actual Power Query's Query Editor list:



all of these queries, parameters, and functions are in the section1.m file. You can easily understand if they are query, parameter or function. Look at screenshot below;

```

shared #Public Holidays in Victoria in 2017" = let
    Source = Web.Page(Web.Contents("http://www.officeholidays.com/countries/australia/victoria/" & Year & ".php")),
    Data0 = Source[0][Data],
    #Added Custom" = Table.AddColumn(Data0, "Year", each Year),
    #Merged Columns" = Table.CombineColumns(#Added Custom,{"Year", "Date"},Combiner.CombineTextByDelimiter(" ", QuoteStyle.None),"Full Date"),
    #Changed Type" = Table.TransformColumnTypes(#Merged Columns,{{"Full Date", type date}}),
    #Removed Columns" = Table.RemoveColumns(#Changed Type,{"Day"})
in
    #Removed Columns";

[ Description = "Input Year column for Public Holidays" ]
shared Year = "2017" meta [IsParameterQuery=true, Type="Text", IsParameterQueryRequired=true];

[ FunctionQueryBinding = "{"exemplarFormulaName":"","Public Holidays in Victoria in 2017"}" ]
GetHolidays = let
    Source = (Year as text) => let
        Source = Web.Page(Web.Contents("http://www.officeholidays.com/countries/australia/victoria/" & Year & ".php")),
        Data0 = Source[0][Data],
        #Added Custom" = Table.AddColumn(Data0, "Year", each Year),
        #Merged Columns" = Table.CombineColumns(#Added Custom,{"Year", "Date"},Combiner.CombineTextByDelimiter(" ", QuoteStyle.None),"Full Date"),
        #Changed Type" = Table.TransformColumnTypes(#Merged Columns,{{"Full Date", type date}}),
        #Removed Columns" = Table.RemoveColumns(#Changed Type,{"Day"})
    in
        #Removed Columns"
in
    Source;

```

Queries are just starting with let expressions. However, for parameters there is a meta information:

meta [IsParameterQuery=true, Type="Text", IsParameterQueryRequired=true]

for functions; there is a FunctionQueryBinding section which explain where the function is sourced from;

[FunctionQueryBinding = "{"exemplarFormulaName":"","Public Holidays in Victoria in 2017"}"]

Green section above is a simple query

Red section is a parameter

and Yellow section is a function

So, very easily you can access all queries pane from this area. However, this doesn't tell you the folder structure, or some other query editor settings. For example, many of these queries are not loading in Power BI and disabled for load. This *.m file doesn't tell you that. That information, however, exists somewhere else. let's look at that.

Metadata Information in XML Format

If you open the DataMashup file in a text editor (like Notepad++) you will see another story. content starts with some binary characters, but then some XML codes appears.

Starting binary data are related to the zipped nature of this file (which we've already went through it in previous part of this post). After binary characters, some XML code starts, which includes some schema definition. You will easily find all the metadata information between these tags;

all useful information are under LocalPackageMetadataFile tags. you can simply remove everything else. After removing everything else and keeping only the content between these tags, then you will have XML structure that includes all metadata information about Power Query's Query Editor;

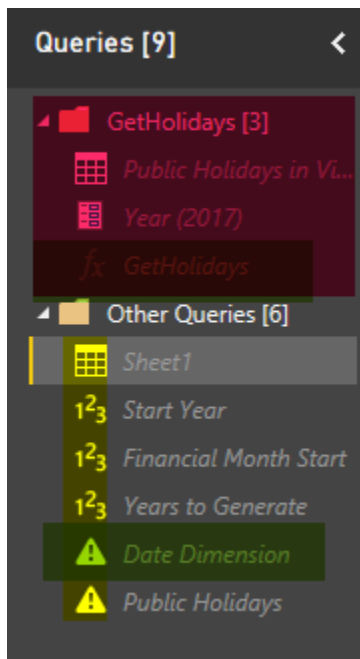
358 | Page

As you can see in above screenshot, there are XML tags for all attributes; name of every step and query (highlighted yellow), and their properties (highlighted green) such as LoadToReportDisabled, ResultType and Value.

If you be a Power Query geek like myself, you would dig into this structure with Power Query Itself! I have opened another Power BI Desktop file, and got data from this file; DataMashup File. I have done some steps, such as removing everything before and after main xml tags, then drilled into different sections of it, and I've got some results such as this:

	ItemLocation.ItemPath	IsPriv...	NameUpdatedAfterFit...	IsDirectQuery	ResultType	LoadToReportDisabled	QueryGroupID	FillToDataModelEnabled	FillEnabled	FillObjectType
1	Section1/Date Dimension	10	11	10	sTable					
2	Section1/Financial Month Start	10	11	10	sNumber	11				
3	Section1/GetHolidays				sFunction		420ca76-9e57-4271-98cd-f2fc3d8			
4	Section1/Public Holidays	10	11	10	sTable	11				
5	Section1/Public Holidays in Victoria in 2017	10	11	10	sTable		420ca76-9e57-4271-98cd-f2fc3d8			
6	Section1/Sheet1	10	11	10	sTable	11				
7	Section1/Start Year	10	11	10	sNumber	11				
8	Section1/Year	10	11		sText	11	420ca76-9e57-4271-98cd-f2fc3d8	10	10	sConnectionOnly
9	Section1/Years to Generate	10	11	10	sNumber	11				

As you can see it is fairly obvious that is the query Parameter, Function or a normal Query(highlighted yellow). Is the query inside a group or not (highlighted red), or is it enabled load in the report (highlighted green), and many other information.



Here you go. You have all you need in terms of understanding the metadata of tables, queries, parameters, and functions.

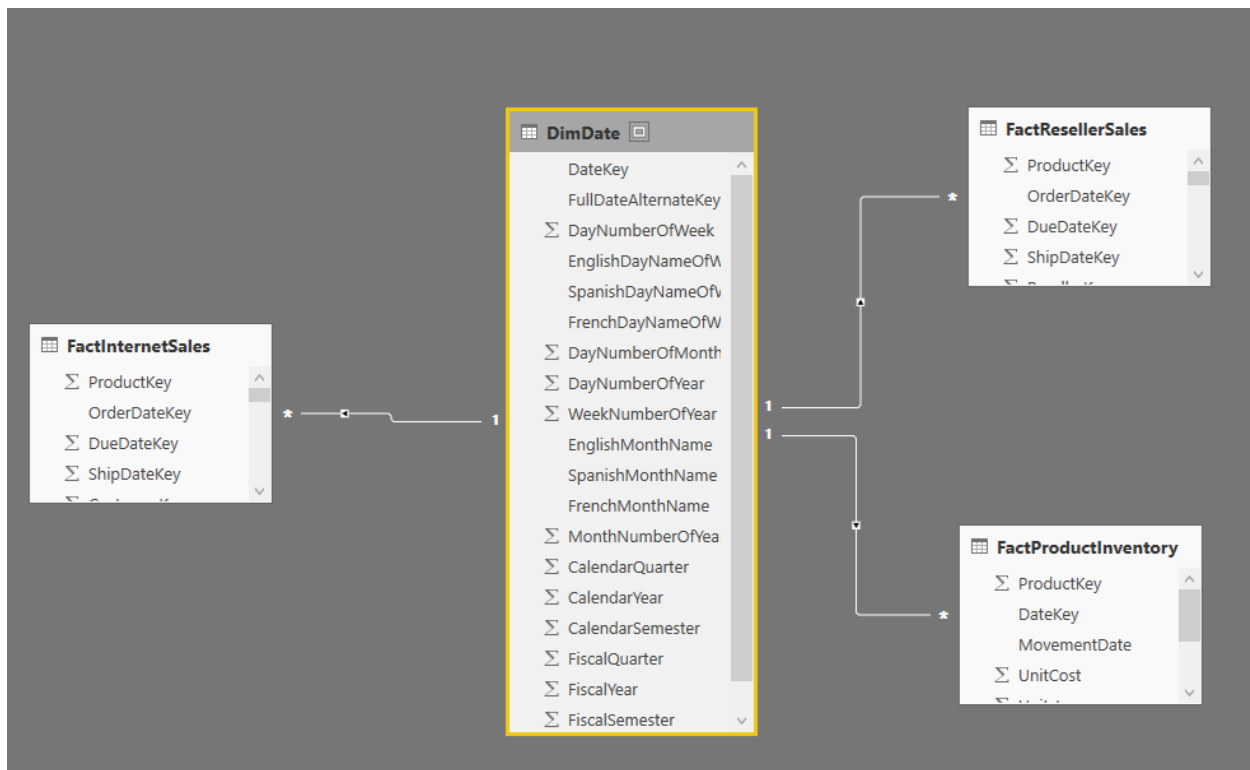
Summary

DataMashup file in *.pbix file included all you need to access to Query Editor's metadata information, and also the M scripts. You can simply use this file to access to the whole code behind of Power Query component of your Power BI solution. In future posts I'll go through steps of exploring data from the XML content in DataMashup file and also explain other files under *.pbix file. There are still a lot of information in XML content of DataMashup file, I encourage you all to explore it with Power Query and open your eyes to the world of metadata of Power Query 😊

Tell me what part of this DataMashup file and metadata information you like most in the comments.

Do You Need a Date Dimension?

Published Date: January 13, 2017



I've hear this question many times; "Do you think we need a date dimension?". some years ago, date dimension has been used more, but nowadays less! The main reason is that there are some tools that are using their own built-in date hierarchy. As an example Power BI has a built-in date hierarchy that is enabled by default on top of all date fields which gives you a simple hierarchy of year, quarter, month, and day. Having this hierarchy brings the question most of the time that do I need to have a date dimension still? of I can simply use this date hierarchy? In this post I'll answer this question and explain it with reasons. Majority of this post is conceptual and can be used across all BI tools, however part of it is Power BI focused. If you like to learn more about Power BI, read [Power BI online book from Rookie to Rock Star](#).

What is Date Dimension?

Date Dimension is a table that has one record per each day, no more, no less! Depends on the period used in the business you can define start and end of the date dimension. For example your date dimension can start from 1st of Jan 1980 to 31st of December of

2030. For every year normally you will have 365 records (one record per year), except leap years with 366 records. here is an example screenshot of a date dimension records;

	Year	Month	Day	FullDateAL...	DateKey	DateFullName	Fiscal Year	Fiscal Quarter	Calendar Quar...	IsWeekDay	DayOfWeek	Month Name	Day of Week Name
1	2013	6	15	6/15/2013	20130615	15 June 2013	2013	4	2	0	6	June	Saturday
2	2013	6	16	6/16/2013	20130616	16 June 2013	2013	4	2	0	0	June	Sunday
3	2013	6	17	6/17/2013	20130617	17 June 2013	2013	4	2	1	1	June	Monday
4	2013	6	18	6/18/2013	20130618	18 June 2013	2013	4	2	1	2	June	Tuesday
5	2013	6	19	6/19/2013	20130619	19 June 2013	2013	4	2	1	3	June	Wednesday
6	2013	6	20	6/20/2013	20130620	20 June 2013	2013	4	2	1	4	June	Thursday
7	2013	6	21	6/21/2013	20130621	21 June 2013	2013	4	2	1	5	June	Friday
8	2013	6	22	6/22/2013	20130622	22 June 2013	2013	4	2	0	6	June	Saturday
9	2013	6	23	6/23/2013	20130623	23 June 2013	2013	4	2	0	0	June	Sunday
10	2013	6	24	6/24/2013	20130624	24 June 2013	2013	4	2	1	1	June	Monday
11	2013	6	25	6/25/2013	20130625	25 June 2013	2013	4	2	1	2	June	Tuesday
12	2013	6	26	6/26/2013	20130626	26 June 2013	2013	4	2	1	3	June	Wednesday
13	2013	6	27	6/27/2013	20130627	27 June 2013	2013	4	2	1	4	June	Thursday
14	2013	6	28	6/28/2013	20130628	28 June 2013	2013	4	2	1	5	June	Friday
15	2013	6	29	6/29/2013	20130629	29 June 2013	2013	4	2	0	6	June	Saturday
16	2013	6	30	6/30/2013	20130630	30 June 2013	2013	4	2	0	0	June	Sunday
17	2013	7	1	7/1/2013	20130701	01 July 2013	2014	1	3	1	1	July	Monday
18	2013	7	2	7/2/2013	20130702	02 July 2013	2014	1	3	1	2	July	Tuesday
19	2013	7	3	7/3/2013	20130703	03 July 2013	2014	1	3	1	3	July	Wednesday
20	2013	7	4	7/4/2013	20130704	04 July 2013	2014	1	3	1	4	July	Thursday
21	2013	7	5	7/5/2013	20130705	05 July 2013	2014	1	3	1	5	July	Friday
22	2013	7	6	7/6/2013	20130706	06 July 2013	2014	1	3	0	6	July	Saturday

Date Dimension is not a big dimension as it would be only ~3650 records for 10 years, or even ~36500 rows for 100 years (and you might never want to go beyond that). Columns will be normally all descriptive information about date, such as Date itself, year, month, quarter, half year, day of month, day of year....

Date Dimension will be normally loaded once, and used many times after it. So it shouldn't be part of your every night ETL or data load process.

Why Date Dimension?

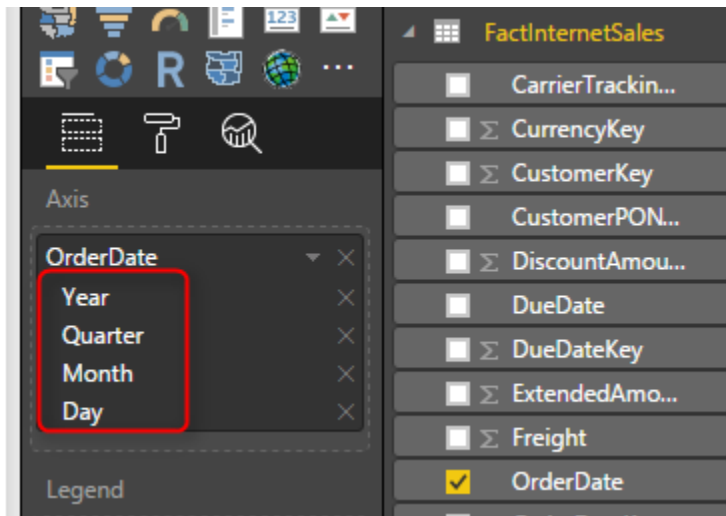
Date Dimension is useful for scenarios mentioned below;

- Ability to slice and dice by many date attributes such as week number, half year, day of year, and etc.
- Consistency in Analysis
- Ability to do analysis based on public holidays (Easter Monday, Good Friday, and etc)
- Some BI tools extended functions need to work with a Date Dimension

Let's look at reasons one by one;

Powerful Slice and Dice

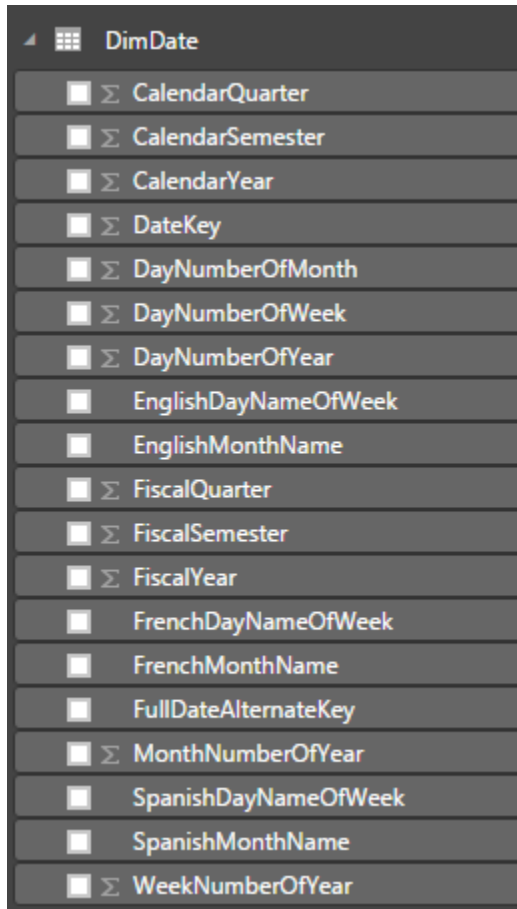
If you have worked with Power BI you know that there is a date hierarchy built-in which gives you fields such as Year, Quarter, Month, and Day. This hierarchy normally adds automatically to date fields. here is an example of how it looks like;



This built-in hierarchy is very good help, but what about times that you want to slice and dice date fields based on something other than these 4 fields? for example questions that you cannot answer with this hierarchy are;

15. Weekday analysis of revenue (by Monday, Tuesday, ..., Sunday)
16. Which week in the year generates least revenue or most revenue?
17. What about Fiscal? What is the revenue for Fiscal year?
18. and many other questions.

The fact is; when you use the simple hierarchy of Year, Quarter, Month, and Day, you are limited to only slice and dice by these fields. If you want to step beyond that, then you have to create additional fields to do it. That's when a date dimension with all of these fields is handy. Here are some fields that you can slice and dice based on that in Date Dimension (and this is just a very short list, a real date dimension might have twice, three times, or 4 times more fields than this! Yes more fields means more power in slicing and dicing)



DimDate
CalendarQuarter
CalendarSemester
CalendarYear
DateKey
DayNumberOfMonth
DayNumberOfWeek
DayNumberOfYear
EnglishDayNameOfWeek
EnglishMonthName
FiscalQuarter
FiscalSemester
FiscalYear
FrenchDayNameOfWeek
FrenchMonthName
FullDateAlternateKey
MonthNumberOfYear
SpanishDayNameOfWeek
SpanishMonthName
WeekNumberOfYear

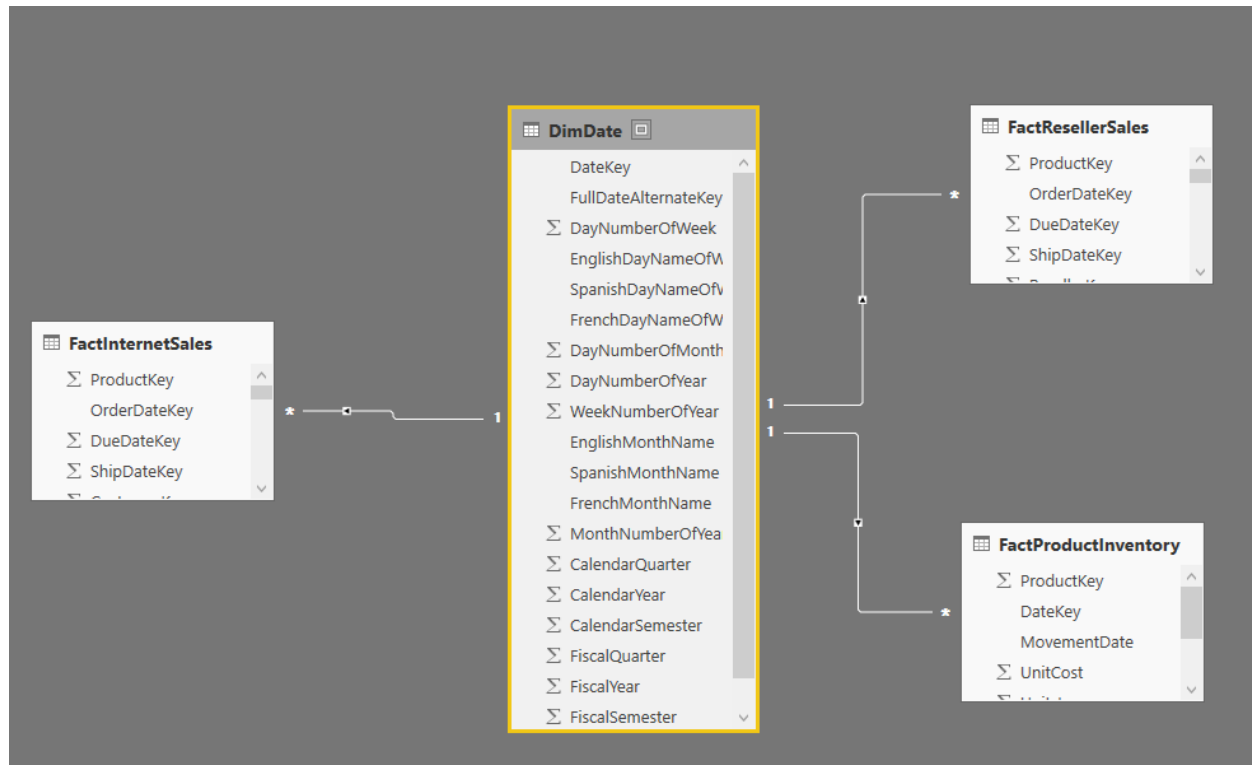
You can answer questions above without a date dimension of course, for example; to answer the first question above, you will need to add Weekday Name, because you would need to sort it based on Weekday number, so you have to bring that as well. To answer second question you need to bring Week number of year. For third question you need to bring fiscal columns, and after a while you will have heaps of date related columns in your fact table!

And worst part is that this is only this fact table (let's say Sales fact table). Next month you'll bring Inventory fact table, and the story begins with date fields in Inventory fact table. This simply leads us to the second reason for date dimension; Consistency.

Consistency in Analysis

You don't want to be able to slice and dice your Sales data by week number of year, but not having this feature in Inventory data! If you repeat these columns in all fact tables you will have big fact tables which are not consistent though, if you add a new column somewhere you have to add it in all other tables. What about a change in calculation? Believe me you never want to do it.

Date Dimension on the other hand is a dimension that is shared between all fact tables. You add all fields and calculations here, and fact tables are only related to this. This is a consistent approach. Here is a simple view of a data warehouse schema with a date dimension shared between two fact tables.



Public Holidays Insight

In most of businesses nowadays public holidays or special event's insight is an important aspect of reporting and analysis. some of analysis are like;

22. Was the sales in Easter this year better than last year? (you don't want to calculate Easter in Power BI of course)
23. How is the revenue is public holidays compared with weekends?
24. How was the sales for store in opening day? (special event)
25. and many other questions....

One of the best use cases of date dimension is public holidays and special events. I have shown previously in another post [how to fetch public holidays live in Power BI](#). Calculating some of public holidays are time consuming and not easy. Easter for example changes every year. You can also have some fields in date dimension that mention what the special day is (opening, closing, massive sale....). Your date dimension

can be enhanced a lot with insight from these fields. Here is an example of public holidays information in a date dimension;

	A ^B _C Holiday	Date
47	Westland Anniversary	11/28/2016
48	Chatham Islands Anniversary	11/28/2016
49	Christmas Day	12/27/2016
50	Boxing Day	12/26/2016
51	New Year's Day	1/3/2017
52	Day after New Year's Day	1/2/2017
53	Wellington Anniversary	1/23/2017
54	Auckland Anniversary	1/30/2017
55	Nelson Anniversary	1/30/2017
56	Waitangi Day	2/6/2017
57	Taranaki Anniversary	3/13/2017
58	Otago Anniversary	3/20/2017
59	Daylight Saving ends	4/2/2017
60	Good Friday	4/14/2017
61	Easter Monday	4/17/2017
62	Easter Tuesday ?	4/18/2017

Extended Functions for Time Intelligence

BI Tools in the market has some extended functions that gives you insight related to date, named as time intelligence functions. For examples, talking about Power BI, you can leverage heaps of DAX functions, such as TotalYTD to calculate year to date, ParallelPeriod to find the period parallel to the current period in previous year or quarter, and many other functions. These functions sometimes need to work with a date dimension, otherwise they won't return correct result!

Yes, you've read it correctly, DAX time intelligence functions won't work properly if you don't have a proper date dimension. proper date dimension is a table that has a full date column in one of the fields, and has **no gaps** in dates. This means that if you are using Sales transaction table which has an OrderDate field as an input to DAX time intelligence functions, and if there is no sales transaction for 1st of January, then the result of time intelligence functions won't be correct! It won't give you an error, but it won't show you the correct result as well, this is the worst type of error might happen. So having a Date Dimension can be helpful in this kind of scenarios as well.

CalendarYear	EnglishMonthName ▲	SalesAmount	Last Year Sales
2005	July	\$473,388.16	
	August	\$506,191.69	
	September	\$473,943.03	
	October	\$513,329.47	
	November	\$543,993.41	
	December	\$755,527.89	
	Total	\$3,266,373.66	
2006	January	\$596,746.56	
	February	\$550,816.69	
	March	\$644,135.20	
	April	\$663,692.29	
	May	\$673,556.20	
	June	\$676,763.65	
	July	\$500,365.16	\$473,388.16
	August	\$546,001.47	\$506,191.69
	September	\$350,466.99	\$473,943.03
	October	\$415,390.23	\$513,329.47
	November	\$335,095.09	\$543,993.41
	December	\$577,314.00	\$755,527.89
	Total	\$6,530,343.53	\$3,266,373.66
2007	January	\$438,865.17	\$596,746.56
	February	\$489,090.34	\$550,816.69
	March	\$485,574.79	\$644,135.20
	April	\$506,399.27	\$663,692.29
	May	\$562,772.56	\$673,556.20
	June	\$554,799.23	\$676,763.65
	July	\$886,668.84	\$500,365.16
	August	\$847,413.51	\$546,001.47
	September	\$1,010,258.13	\$350,466.99
	October	\$1,080,449.58	\$415,390.23
	November	\$1,196,981.11	\$335,095.09
	December	\$1,731,787.77	\$577,314.00
	Total	\$9,791,060.30	\$6,530,343.53
2008	January	\$1,340,244.95	\$438,865.17
	February	\$1,462,479.83	\$489,090.34
	March	\$1,480,905.18	\$485,574.79
	April	\$1,608,750.53	\$506,399.27
	May	\$1,878,317.51	\$562,772.56
	June	\$1,949,361.11	\$554,799.23
	July	\$50,840.63	\$886,668.84
	August		\$847,413.51
	September		\$1,010,258.13
	October		\$1,080,449.58
	November		\$1,196,981.11

In this post I've explained [how to use some of DAX time intelligence functions in Power BI](#).

Do I Need a Date Dimension?

Now is the time to answer the question; "Do I need a date dimension"? the answer is: Yes, IMHO. Let me answer that in this way: Obviously I can do a BI solution without date dimension, but would you build a table without hammer?! Would you build a BI solution that has some date columns in each fact table, and they are not consistent with each other? would you overlook having special event or public holidays dates insight in your solution? would you only stick to year, quarter, month, and date slicing and dicing? would you like to write all your time intelligence functions yourself? Obviously you don't. Everyone has the same answer when I explain it all.

I have been working with countless of BI solutions in my work experience, and there is always requirement for a Date Dimension. Sooner or later you get to a point that you need to add it. In my opinion add it at the very beginning so you don't get into the hassle of some rework.

How to Build a Date Dimension?

Now that you want to build a date dimension, there are many ways to build it. Date Dimension is a dimension table that you normally load once and use it always. because 1st of Jan 2017 is always 1st of Jan 2017! data won't change, except bringing new holiday or special event dates. I mention some of ways to build a date dimension as below (there are many more ways to do it if you do a bit of search in Google);

Date Dimension T-SQL Script

You can use a written T-SQL Script and load date dimension into a relational database such as SQL Server, Oracle, MySQL, and etc. Here is a [T-SQL example](#) I have written few years ago.

Power Query Script

I have written a [blog post](#) explaining how you can do the whole date dimension in Power Query. the blog post is a bit old, I will write a new one in near future and will show an easier way of doing this in Power BI.

Fetch Public Holidays

I have written another [blog post](#) that mentioned how to fetch public holidays from internet and attach it to a date dimension in Power BI.

DAX Calendar Functions

You can use [Calendar\(\)](#) function in DAX to generate a date dimension very quickly as well.

Using Date Dimension Multiple Times?

Using a dimension multiple times in a data warehouse called Role Playing dimension. I have explained how to do role playing dimension for DimDate in [this blog post](#).

Script to Generate Date Dimension with Power Query M - With Financial Columns

Published Date: February 14, 2014

There are many date dimensions scripts on the internet, and even there are some scripts for the Power Query (Thanks to [Chris Webb for his very early version of the date dimension](#), and also to [Matt Masson for his version of Date Dimension](#)). I've built a date dimension from scratch (not a copy or extended version), because one of my reasons to build this dimension was Practice! So I did that to practice Power Query more in action. There are some other reasons that why I built this dimension.

Why I built the Date dimension;

- To use it: Date Dimension is one of the most common dimensions in Data Model. It is too rare to you see a data model without date dimension.
- I wanted to build a Power Query copy of my date dimension ([previously I've done a date dimension with T-SQL](#))
- I used this as a practice to learn Power Query more in action.
- I need a demo for my upcoming speaking and presentations on Power Query.
- I like to share something useful in community.

The Date dimension that I've built contains:

- Financial calendar (configured in the config table)
- Weekend and Weekdays

*Note: This version of date dimension doesn't contains public holidays, I'll work on the newer version with public holidays and write another blog post about it.

Here is the script to build the Date Dimension:

```
let
// Read Config Table
ConfigTable=Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
```



```

YearsToAppend=Table.First(ConfigTable)[YearsToGenerate],
FinancialYearStartingMonth=Table.First(ConfigTable)[FinancialYearStartingMonth],
// Generate base table
Source =
List.Dates(Date.From(Table.First(ConfigTable)[StartDate]),YearsToAppend*365,#duration(1,
0, 0, 0)),
Transformed=List.Transform(Source, each Date.ToRecord(_),
Tabled=Table.FromList(Transformed,Record.FieldValues,{"Year","Month","Day"}),
//Add Full Date Column
FullDateAddedTable=Table.AddColumn(Tabled,"FullDateAlternateKey",each
Date.FromText(Text.From([Year])&"-"&Text.From([Month])&"-"&Text.From([Day]))),
DateKeyAdded=Table.AddColumn(FullDateAddedTable,"DateKey",each
([Year]*10000)+([Month]*100)+[Day]),
FullDateNameAdded=Table.AddColumn(DateKeyAdded,"DateFullName",each
DateTime.ToText(DateTime.From([FullDateAlternateKey]),"dd MMMM yyyy")),
// Fiscal Year
FiscalYearAdded=Table.AddColumn(FullDateNameAdded,
"Fiscal Year",
each
if Date.Month([FullDateAlternateKey])>=FinancialYearStartingMonth then
Date.Year([FullDateAlternateKey])+1
else
Date.Year([FullDateAlternateKey])
),
// Fiscal Month
FiscalQuarterAdded=Table.AddColumn(FiscalYearAdded,
"Fiscal Quarter",
each
if Date.Month([FullDateAlternateKey])>=FinancialYearStartingMonth then
Number.IntegerDivide((Date.Month([FullDateAlternateKey])-
FinancialYearStartingMonth),3)+1
else
Number.IntegerDivide((12+Date.Month([FullDateAlternateKey])-
FinancialYearStartingMonth),3)+1
),
// Calendar Quarter
CalendarQuarterAdded=Table.AddColumn(FiscalQuarterAdded, "Calendar Quarter",
each Number.IntegerDivide(Date.Month([FullDateAlternateKey])-1,3)+1
),
// Is Week Day

```

```

WeekDayAdded=Table.AddColumn(CalendarQuarterAdded, "IsWeekDay",
each
if
Date.DayOfWeek(DateTime.From([FullDateAlternateKey]))=Day.Sunday
or
Date.DayOfWeek(DateTime.From([FullDateAlternateKey]))=Day.Saturday
then 0 else 1 ),
// Day Of Week
DayOfWeek=Table.AddColumn(WeekDayAdded,"DayOfWeek",each
Date.DayOfWeek(DateTime.From([FullDateAlternateKey]))),
// Month Name
MonthName=Table.AddColumn(DayOfWeek,"Month Name",each
DateTime.ToText(DateTime.From([FullDateAlternateKey]),"MMMM")),
// Day of Week Name
DayOfWeekName=Table.AddColumn(MonthName,"Day of Week Name",each
DateTime.ToText(DateTime.From([FullDateAlternateKey]),"dddd"))

in
DayOfWeekName

```

You can download the Sample workbook with config table and the script from here:

[date-dimension-with-public-holidays-start](#)

What I've learned through this example:

Loop Structure

As you probably noticed, Power Query (up to current version) doesn't contain a loop structure.

This is a weakness for this version, but there are methods to do what you want with other structures. List and Table are structures that contains item and records.

Fortunately List can be generated with some functions that called as **GENERATORS**. If you notice to the line of text with List.Dates, you would see that this line of code generates a list of dates from the start date with the duration specified up to specific number.

This line of code :

```

Source =
List.Dates(Date.From(Table.First(ConfigTable)[StartDate]),YearsToAppend*365,#duration(1,
0, 0, 0)),

```

Above line generates the list of dates, and below lines transform that date to a record containing day, month, and year, and finally it would be converted to a table with Table.FromList function:

```
Transformed=List.Transform(Source, each Date.ToRecord(_)),
Tabled=Table.FromList(Transformed,Record.FieldValues,{"Year","Month","Day"}),
```

So as you seen above we generated a list from scratch, and then converted that to a table. you can apply expressions to a new column of the table with EACH single parameter function, and that means you can simulate loop like structure with List/Table structures. I would dedicate a blog post about this topic later with more illustration and detailed samples.

Date Functions

There are bunch of date functions used in this sample, such as :

Date.Month : returns the month of specific date

Date.DayOfWeek: returns enumeration values of the weekday, zero would be Sunday, and 6 would be Saturday

DateTime.ToText: generates a text FORMATTED from the date time with specified format.

There are many date functions which I will write about them in another blog post in future.

Reading information from Excel Workbook

In this script I used a config table, that table sits in an excel workbook, and I used lines below to read that table from the Excel workbook:

```
// Read Config Table
ConfigTable=Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
YearsToAppend=Table.First(ConfigTable)[YearsToGenerate],
FinancialYearStartingMonth=Table.First(ConfigTable)[FinancialYearStartingMonth],
```

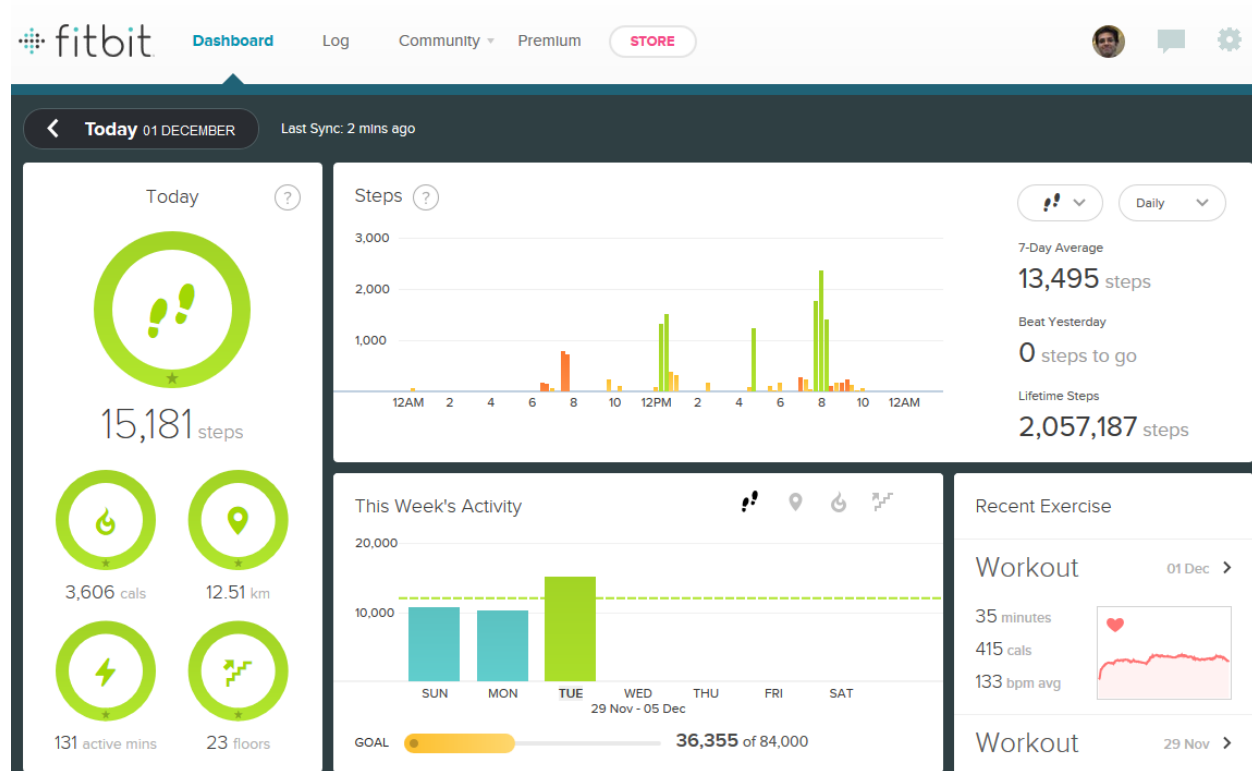
Having a config table such as the one above, helps to re-generate data with just a REFRESH button instead of changing hard-coded values in the script.

Be Fitbit BI Developer in Few Steps: First Step Get Data from CSV

Published Date: December 1, 2015



Power BI made things really easy, it isn't hard to build dashboards and data analysis on almost any types of data sources. Previously in other sections of the [online book; Power BI from Rookie to Rockstar](#) I've mentioned few data sources and ways of connecting to them. In this post I like to show you how Power BI easily works with CSV or Text files. CSV or Text files can be in different formats, but for this example I through using Fitbit exported CSV data sounds to be great demo. The reason is that you probably have seen your Fitbit dashboard in the website as below:



So for this post we are going to build that dashboard (not all of that obviously, because we don't have the data required for all of that), but most part of it with Power BI. You will see how easy and powerful is Power BI in this kind of scenarios, and you will see how you can be the BI Developer of Fitbit in a few steps of building this demo.

Before start I have to apologize in advance to Fitbit developer team, and I have to say that I don't want to say they are not doing anything. Obviously they are doing great job, and the most part of this great job is hidden from our view, they are mashing up the data to build it in the way that can be visualized best. So I admire them because of their efforts. In this post I just want to show how easy is to build dashboards with Power BI when the data is available in CSV or Text format.

So Let's get started. Fitbit devices are usually smart bands that measure your steps, number of floors, calories spent, heart rate and some other information. You can always download an extracted version of your activities as CSV file from your account page in Fitbit website. This always gives you a month period of data. So if you want the data to be exported for two different month, you will end up with two CSV files.

Settings
a.raad.g@gmail.com

View profile >

Personal Info

Notifications

Devices

Silent Alarms

Privacy

Data Export

Applications

Sharing

Export your data

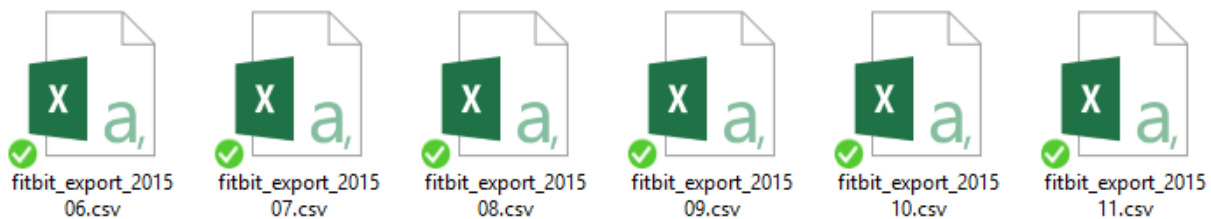
Your data belongs to you!
Download your Fitbit data as an **XLS** file (Microsoft Excel) or **CSV**, which can be opened by many paid, free or shareware applications.

Time period
☐ This week
☐ This month
☐ Last week
☒ Last month
☐ Custom:
from
to
You can export up to 30 days of data.

Data
☒ Body
☐ Foods
☐ Activities
☐ Sleep
File format

Download

I have exported my Fitbit activities history from the time that I got it as a gift (Great gift I have to say 😊) till now, which is 5 months.



Here is sample data set in files:

```

1 Activities
2 Date,Calories Burned,Steps,Distance,Floors,Minutes Sedentary,Minutes Lightly Active,Minu
3 "01-07-2015","3,953","13,361","10.64","16","1,058","196","71","87","2,393"
4 "02-07-2015","3,432","10,818","8.76","10","726","224","19","45","1,803"
5 "03-07-2015","4,102","15,537","12.38","16","701","237","55","88","2,550"
6 "04-07-2015","3,959","14,490","11.54","6","685","255","46","69","2,443"
7 "05-07-2015","4,028","14,652","11.68","3","1,003","283","93","61","2,598"
8 "06-07-2015","3,538","12,161","9.84","16","1,144","168","76","52","1,968"
9 "07-07-2015","3,550","11,103","8.99","17","785","250","24","41","1,950"
10 "08-07-2015","3,238","10,029","8.22","10","1,204","152","36","48","1,587"
11 "09-07-2015","3,994","13,951","11.29","10","733","183","85","96","2,489"
12 "10-07-2015","3,998","12,516","9.5","9","650","194","92","92","2,508"
13 "11-07-2015","2,802","6,750","5.18","14","594","164","14","29","1,115"
14 "12-07-2015","4,156","19,471","15.39","21","472","272","92","96","2,781"
15 "13-07-2015","3,523","14,730","11.73","17","654","208","39","73","1,959"
16 "14-07-2015","3,039","9,543","7.24","16","712","195","10","38","1,409"
17 "15-07-2015","2,750","6,663","5.06","15","1,222","206","6","6","1,085"
18 "16-07-2015","3,373","14,603","11.46","12","745","198","22","69","1,813"
19 "17-07-2015","3,948","15,787","11.98","29","649","252","23","115","2,471"
20 "18-07-2015","3,917","16,652","13.35","19","543","332","52","55","2,512"
21 "19-07-2015","3,447","11,983","9.84","3","561","255","48","37","1,945"
22 "20-07-2015","3,430","12,816","10.35","15","709","174","37","74","1,834"
23 "21-07-2015","3,693","16,979","13.62","39","744","162","64","100","2,167"
24 "22-07-2015","3,461","13,088","10.52","15","717","211","29","59","1,863"
25 "23-07-2015","3,400","15,143","12.36","14","696","185","31","69","1,827"
26 "24-07-2015","4,092","20,077","15.82","24","690","172","70","127","2,602"
27 "25-07-2015","3,736","16,374","13.54","14","576","248","30","68","2,208"
28 "26-07-2015","3,453","13,676","11.18","4","576","232","45","55","1,933"
29 "27-07-2015","3,298","12,026","9.77","14","775","173","44","52","1,684"
30 "28-07-2015","3,654","17,401","14.14","17","1,140","163","37","100","2,115"
31 "29-07-2015","3,588","14,711","11.82","15","681","241","40","58","2,062"
32 "30-07-2015","3,059","8,671","6.58","30","790","231","7","20","1,417"
33 "31-07-2015","3,562","14,522","11.05","14","673","187","58","80","2,023"

```

I start reading data from CSV in Power BI Desktop with Get Data from CSV. But this is the result I get:

fitbit_export_201507.csv ×

Column1
Activities
Date
01-07-2015
02-07-2015
03-07-2015
04-07-2015
05-07-2015
06-07-2015
07-07-2015
08-07-2015
09-07-2015

Load Edit Cancel

Well, this is happening because of the very first record in CSV file which is the header row, as you see below;

```

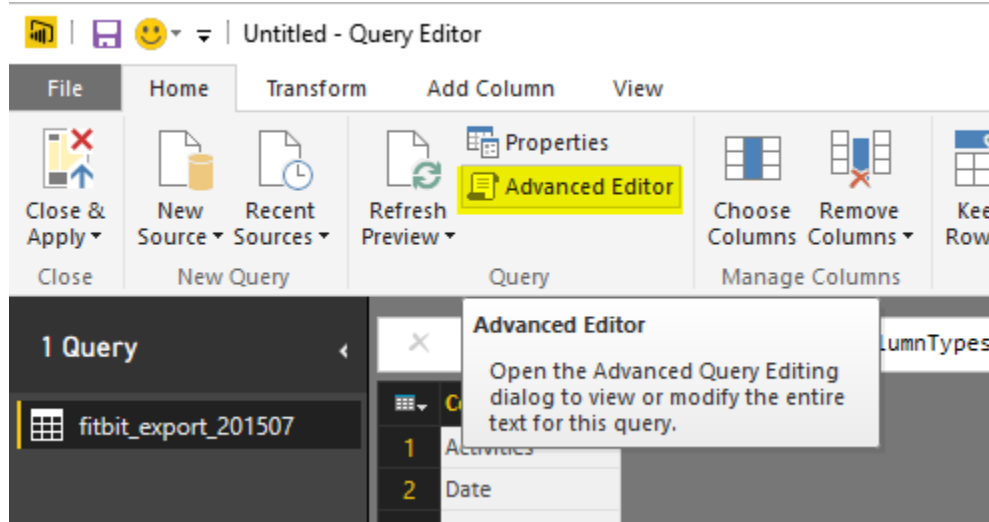
1 Activities
2 Date,Calories Burned,Steps,Distance,Floors,Minutes Sedentary,Minutes Lightly Active,Minu
3 "01-07-2015","3,953","13,361","10.64","16","1,058","196","71","87","2,393"
4 "02-07-2015","3,432","10,818","8.76","10","726","224","19","45","1,803"
5 "03-07-2015","4,102","15,537","12.38","16","701","237","55","88","2,550"
6 "04-07-2015","3,959","14,490","11.54","6","685","255","46","69","2,443"
7 "05-07-2015","4,028","14,652","11.68","3","1,003","283","93","61","2,598"
8 "06-07-2015","3,538","12,161","9.84","16","1,144","168","76","52","1,968"
9 "07-07-2015","3,550","11,103","8.99","17","785","250","24","41","1,950"
10 "08-07-2015","3,238","10,029","8.22","10","1,204","152","36","48","1,587"
11 "09-07-2015","3,994","13,951","11.29","10","733","183","85","96","2,489"
12 "10-07-2015","3,998","12,516","9.5","9","650","194","92","92","2,508"
13 "11-07-2015","2,802","6,750","5.18","14","594","164","14","29","1,115"
14 "12-07-2015","4,156","19,471","15.39","21","472","272","92","96","2,781"
15 "13-07-2015","3,523","14,730","11.73","17","654","208","39","73","1,959"
16 "14-07-2015","3,039","9,543","7.24","16","712","195","10","38","1,409"
17 "15-07-2015","2,750","6,663","5.06","15","1,222","206","6","6","1,085"
18 "16-07-2015","3,373","14,603","11.46","12","745","198","22","69","1,813"
19 "17-07-2015","3,948","15,787","11.98","29","649","252","23","115","2,471"
20 "18-07-2015","3,917","16,652","13.35","19","543","332","52","55","2,512"
21 "19-07-2015","3,447","11,983","9.84","3","561","255","48","37","1,945"
22 "20-07-2015","3,430","12,816","10.35","15","709","174","37","74","1,834"
23 "21-07-2015","3,693","16,979","13.62","39","744","162","64","100","2,167"
24 "22-07-2015","3,461","13,088","10.52","15","717","211","29","59","1,863"
25 "23-07-2015","3,400","15,143","12.36","14","696","185","31","69","1,827"
26 "24-07-2015","4,092","20,077","15.82","24","690","172","70","127","2,602"
27 "25-07-2015","3,736","16,374","13.54","14","576","248","30","68","2,208"
28 "26-07-2015","3,453","13,676","11.18","4","576","232","45","55","1,933"
29 "27-07-2015","3,298","12,026","9.77","14","775","173","44","52","1,684"
30 "28-07-2015","3,654","17,401","14.14","17","1,140","163","37","100","2,115"
31 "29-07-2015","3,588","14,711","11.82","15","681","241","40","58","2,062"
32 "30-07-2015","3,059","8,671","6.58","30","790","231","7","20","1,417"
33 "31-07-2015","3,562","14,522","11.05","14","673","187","58","80","2,023"

```

The first row doesn't have any delimiter, and that is why Power BI Desktop THINKS that this is a one column CSV file. To fix this issue you can use two approaches:

- Manually change each CSV file and remove the header row, and save it back.
- Write few lines in Power Query scripts and get rid of the first line through script automatically.

The first approach isn't my favorite as it is manual, and won't be so much of use in real world cases when you have 1000s of files. so let's go with the second. Click on Edit and go to Query Editor window. There you can click on Advanced Editor to Change the code.



In the Advanced Editor window you see the M script that generates the output.



Change the Code to below;

```
let
    Source =
        Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles\fitbit_export_201507.csv"),null,null,1252)})
in
    Source
```

Now you can see the result as full data in one column rather than only first column with date values.

= Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PO1	
Column1	
1	Activities
2	Date,Calories Burned,Steps,Distance,Floors,Minutes Sedentary,Minutes Lightly Active,Minutes Fairly Active,Minutes Very Active,Activity Calories
3	"01-07-2015","3,953","13,361","10.64","16","1,058","196","71","87","2,393"
4	"02-07-2015","3,432","10,818","8.76","10","726","224","19","45","1,803"
5	"03-07-2015","4,102","15,537","12.38","16","701","237","55","88","2,550"
6	"04-07-2015","3,959","14,490","11.54","6","685","255","46","69","2,443"
7	"05-07-2015","4,028","14,652","11.68","3","1,003","283","93","61","2,598"
8	"06-07-2015","3,538","12,161","9.84","16","1,144","168","76","52","1,968"
9	"07-07-2015","3,550","11,103","8.99","17","785","250","24","41","1,950"
10	"08-07-2015","3,238","10,029","8.22","10","1,204","152","36","48","1,587"
11	"09-07-2015","3,994","13,951","11.29","10","733","183","85","96","2,489"
12	"10-07-2015","3,998","12,516","9.5","9","650","194","92","92","2,508"
13	"11-07-2015","2,802","6,750","5.18","14","594","164","14","29","1,115"

Follow steps below to shape the data for using the best in report;

Get rid of first record with Remove Rows, and then choosing 1 top row to remove;

y Editor

Add Column View

Properties Advanced Editor

Query

Manage Columns

Choose Columns Remove Columns Keep Rows Remove Rows

Remove Rows

Remove Top Rows

Remove Bottom Rows

Remove Alternate Rows

Remove Blank Rows

Remove Duplicates

Remove Errors

Sort

Split Column

Table

= Table.FromColumns({Lines.From

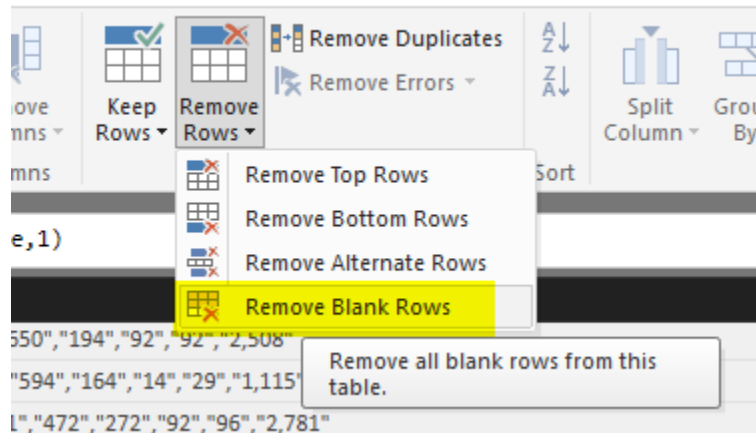
Column1

1 Activities

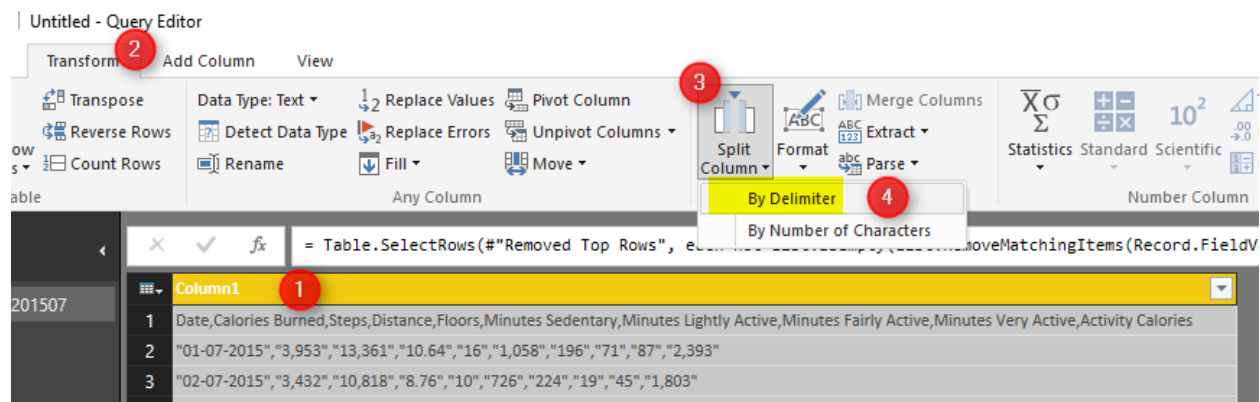
2 Date,Calories Burned,Steps,Distance,Floors,Minutes Sedentary,Minutes Lightly Active,Minutes Fairly Active,Mi

3 "01-07-2015","3,953","13,361","10.64","16","1,058","196","71","87","2,393"

You can also remove any blank rows at the end of report



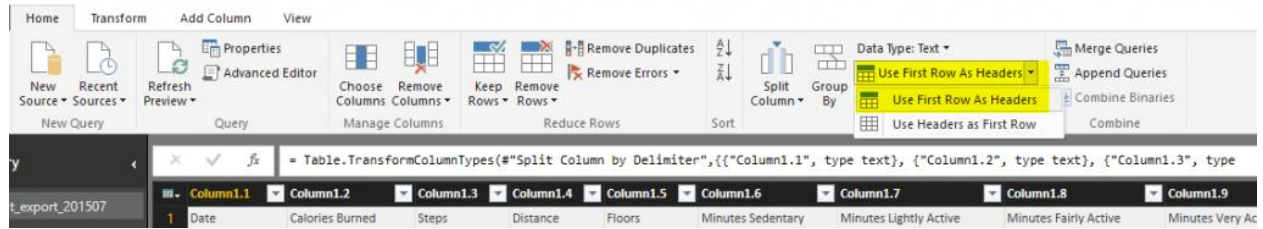
Then select Column1 and Got to Transform Tab, Choose Split Column, then By Delimiter



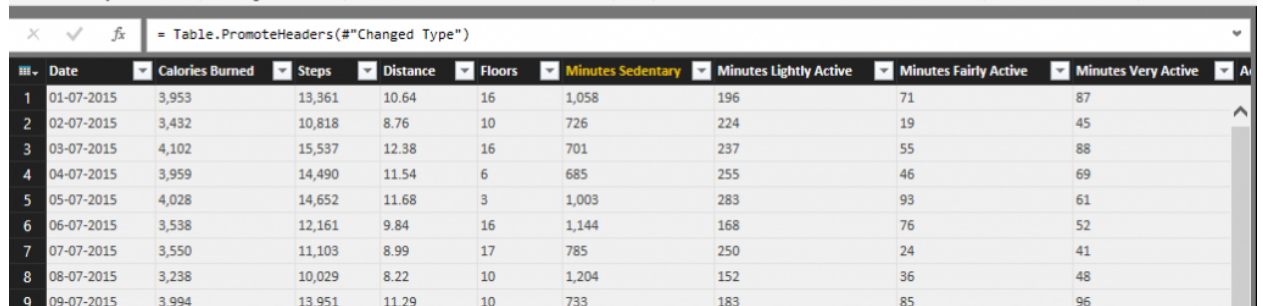
Leave the options as default (which is delimiter Comma, and Split at each occurrence of the delimiter) in Split Column by Delimiter dialog box and continue. You will see how columns all split as below;

	Column1.1	Column1.2	Column1.3	Column1.4	Column1.5	Column1.6	Column1.7	Column1.8	Column1.9
1	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active
2	01-07-2015	3,953	13,361	10.64	16	1,058	196	71	87
3	02-07-2015	3,432	10,818	8.76	10	726	224	19	45
4	03-07-2015	4,102	15,537	12.38	16	701	237	55	88
5	04-07-2015	3,959	14,490	11.54	6	685	255	46	69
6	05-07-2015	4,028	14,652	11.68	3	1,003	283	93	61
7	06-07-2015	3,538	12,161	9.84	16	1,144	168	76	52
8	07-07-2015	3,550	11,103	8.99	17	785	250	24	41
9	08-07-2015	3,238	10,029	8.22	10	1,204	152	36	48

Set the first row as header, which going to Home Tab in Query Editor window and click on Use First Row As Headers option.



This option will rename all column headers to the values from the first row;



The screenshot shows the Power BI Transform ribbon with the 'Promote Headers' option selected in the 'Data Type: Text' dropdown menu. The ribbon also shows other options like 'Merge Queries', 'Append Queries', 'Combine Binaries', and 'Combine'.

	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active
1	01-07-2015	3,953	13,361	10.64	16	1,058	196	71	87
2	02-07-2015	3,432	10,818	8.76	10	726	224	19	45
3	03-07-2015	4,102	15,537	12.38	16	701	237	55	88
4	04-07-2015	3,959	14,490	11.54	6	685	255	46	69
5	05-07-2015	4,028	14,652	11.68	3	1,003	283	93	61
6	06-07-2015	3,538	12,161	9.84	16	1,144	168	76	52
7	07-07-2015	3,550	11,103	8.99	17	785	250	24	41
8	08-07-2015	3,238	10,029	8.22	10	1,204	152	36	48
9	09-07-2015	3,994	13,951	11.29	10	733	183	85	96

Now change data type of Distance column to Decimal, and all other columns (except Date) to Whole Number

	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly	
1	01-07-2015	3,953			10.64	16	1,058	196
2	02-07-2015	3,432			8.76	10	726	224
3	03-07-2015	4,102			12.38	16	701	237
4	04-07-2015	3,959			11.54	6	685	255
5	05-07-2015	4,028			11.68	3	1,003	283
6	06-07-2015	3,538			9.84	16	1,144	168
7	07-07-2015	3,550			8.99	17	785	250
8	08-07-2015	3,238			8.22	10	1,204	152
9	09-07-2015	3,994						183
10	10-07-2015	3,998						194
11	11-07-2015	2,802						164
12	12-07-2015	4,156						272
13	13-07-2015	3,523						208
14	14-07-2015	3,039						195
15	15-07-2015	2,750						206
16	16-07-2015	3,373						198
17	17-07-2015	3,948		15,787	12.36	14	696	185
18	18-07-2015	3,917		16,652				
19	19-07-2015	3,447		11,983				
20	20-07-2015	3,430		12,816				
21	21-07-2015	3,693		16,979				
22	22-07-2015	3,461		13,088				
23	23-07-2015	3,400		15,143				

Copy

Remove Columns

Remove Other Columns

Remove Duplicates

Remove Errors

12

Replace Values...

Fill

Change Type

Transform

Merge Columns

Group By...

Unpivot Columns

Unpivot Other Columns

Move

Decimal Number

Fixed Decimal Number

Whole Number

Date/Time

Date

Time

Date/Time/Timezone

Duration

Text

True/False

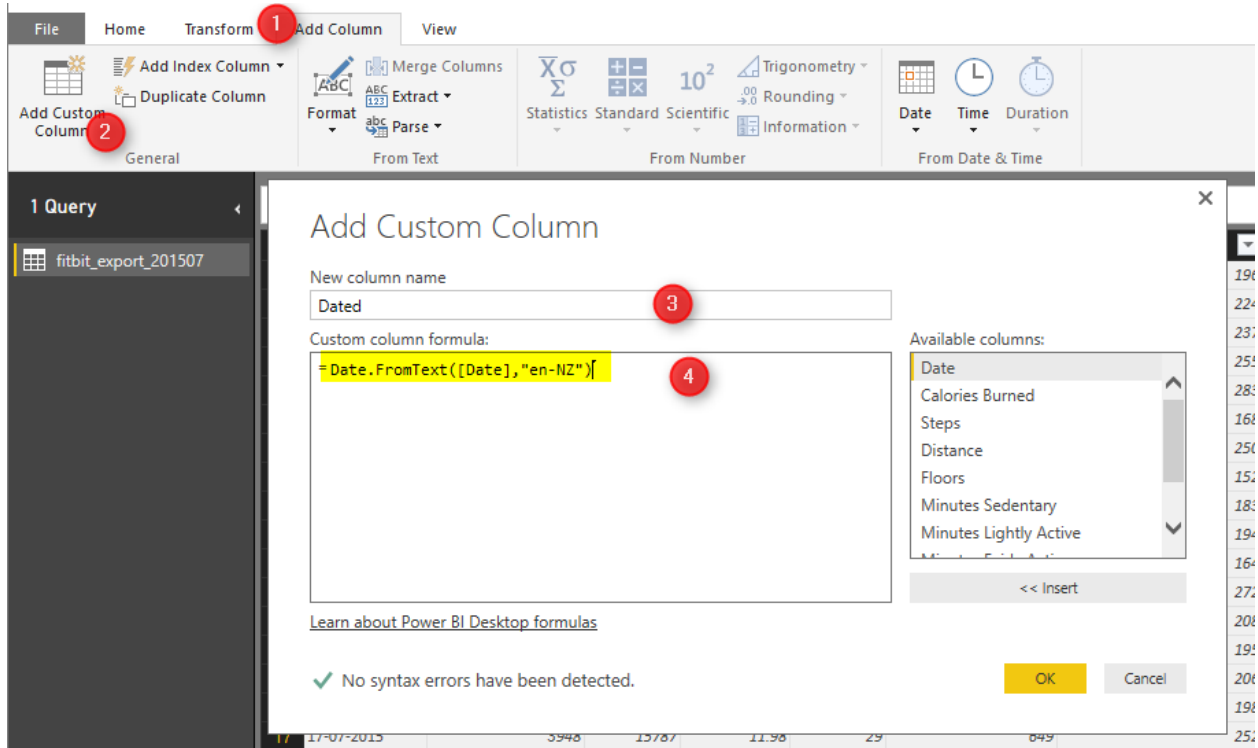
Binary

Using Locale...

For the Date Column itself you can't always change the type that easily, because the Date conversion works with locale of your system. As you see in screenshot below the Date format is DD-MM-YYYY which not always equal to locale of all systems.

	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active
1	01-07-2015	3953	13361	10.64	16	1058	196	71	81
2	02-07-2015	3432	10818	8.76	10	726	224	19	41
3	03-07-2015	4102	15537	12.38	16	701	237	55	81
4	04-07-2015	3959	14490	11.54	6	685	255	46	61
5	05-07-2015	4028	14652	11.68	3	1003	283	93	61
6	06-07-2015	3538	12161	9.84	16	1144	168	76	51
7	07-07-2015	3550	11103	8.99	17	785	250	24	41
8	08-07-2015	3238	10029	8.22	10	1204	152	36	41
9	09-07-2015	3994	13951	11.29	10	733	183	85	91
10	10-07-2015	3998	12516	9.5	9	650	194	92	91
11	11-07-2015	2802	6750	5.18	14	594	164	14	21
12	12-07-2015	4156	19471	15.39	21	472	272	92	91
13	13-07-2015	3523	14730	11.73	17	654	208	39	71
14	14-07-2015	3039	9543	7.24	16	712	195	10	31
15	15-07-2015	2750	6663	5.06	15	1222	206	6	1

If you try to convert it you might get an error. So because of that I convert it through M script. Click on Add Column, and then choose Custom Column. Here is where you can write M script to generate new column. the script is:



`Date.FromText([Date], "en-NZ")`

The locale that you might be using might be different. [Here](#) are list of culture and locale information. After doing that you will see Dated Column added to the end of table with Date data type. Now you can remove the original Date column and move the new Dated Column to the beginning of the table as below;

Dated	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Acti
7/1/2015	3953	13361	10.64	16	1058	196	71	87	
7/2/2015	3432	10818	8.76	10	726	224	19	45	
7/3/2015	4102	15537	12.38	16	701	237	55	88	
7/4/2015	3959	14490	11.54	6	685	255	46	69	
7/5/2015	4028	14652	11.68	3	1003	283	93	61	
7/6/2015	3538	12161	9.84	16	1144	168	76	52	
7/7/2015	3550	11103	8.99	17	785	250	24	41	
7/8/2015	3238	10029	8.22	10	1204	152	36	48	
7/9/2015	3994	13951	11.29	10	733	183	85	96	

So our data is ready now to work with. And this is the final M script of our work;

let


```

Source =
Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV
\SourceFiles\fitbit_export_201507.csv"),null,null,1252)}),
    #"Removed Top Rows" = Table.Skip(Source,1),
    #"Split Column by Delimiter" = Table.SplitColumn(#"Removed Top
Rows","Column1",Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv),{"Column1.1", "Column1.2",
"Column1.3", "Column1.4", "Column1.5", "Column1.6", "Column1.7", "Column1.8", "Column1.9",
"Column1.10"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type
text}, {"Column1.2", type text}, {"Column1.3", type text}, {"Column1.4", type text}, {"Column1.5", type text},
{"Column1.6", type text}, {"Column1.7", type text}, {"Column1.8", type text}, {"Column1.9", type text},
{"Column1.10", type text}}),
    #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
    #"Removed Blank Rows" = Table.SelectRows(#"Promoted Headers", each not
List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null}))),
    #"Changed Type1" = Table.TransformColumnTypes(#"Removed Blank Rows",{{"Steps", Int64.Type},
{"Calories Burned", Int64.Type}, {"Floors", Int64.Type}, {"Minutes Sedentary", Int64.Type}, {"Minutes Lightly
Active", Int64.Type}, {"Minutes Fairly Active", Int64.Type}, {"Minutes Very Active", Int64.Type}, {"Activity
Calories", Int64.Type}, {"Distance", type number}}),
    #"Added Custom" = Table.AddColumn(#"Changed Type1", "Dated", each Date.FromText([Date],"en-
NZ")),
    #"Changed Type2" = Table.TransformColumnTypes(#"Added Custom",{{"Dated", type date}}),
    #"Removed Columns" = Table.RemoveColumns(#"Changed Type2",{"Date"}),
    #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns",{"Dated", "Calories Burned",
"Steps", "Distance", "Floors", "Minutes Sedentary", "Minutes Lightly Active", "Minutes Fairly Active",
"Minutes Very Active", "Activity Calories"})
in
    #"Reordered Columns"

```

However we have some other files to work with as well, and they all have same structure. We can use M Script to loop through all files in the source directory and load them one by one automatically, but that would make this post a very long post, and requires also a bit of scripting which you might not like. I will explain that part in the next section. For now just for you to stay interested here is a glance of a sample report at the end of this effort (we will build that together through next steps);



Be Fitbit BI Developer in Few Steps: Step 2 Loop Through All CSV Files

Published Date: December 2, 2015

= Folder.Files("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles")									
	Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path	
1	Binary	fitbit_export_201506.csv	.csv	12/1/2015 3:29:56 PM	9/24/2015 12:29:44 AM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
2	Binary	fitbit_export_201507.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:49:38 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
3	Binary	fitbit_export_201508.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:48:52 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
4	Binary	fitbit_export_201509.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:48:11 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
5	Binary	fitbit_export_201510.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:47:09 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
6	Binary	fitbit_export_201511.csv	.csv	12/1/2015 3:30:38 PM	12/1/2015 3:07:38 PM	12/1/2015 3:07:36 PM	Record	C:\Users\Reza\SkyDi	

In the [first step](#) you've learned how to manipulate and mash up the data with few changes in Power Query M script. In spite of Power BI Desktop not being able to recognize your CSV file correctly because of the header row, you managed to write script and fetch data as you required. However this process needs to be repeated for the second file, and then the third file, and god knows what happens if you have 1000s of files. Real case scenarios contains many files, even millions. So a manual process won't be the answer. You have to create your data transformation process in a way that works automatically not matter how many files are there.

Warning: This post contains scripting!

In this post we will write M Power Query script to loop through all CSV files in the source folder and process them one by one with the fetch table script that we produced in previous step. So be prepared to write some code, and don't scare from it 😊 It is almost impossible to create a BI solution for real world problem solving that doesn't include scripting (even a bit), So you have to be familiar with that. Yes, Power BI is not just a nice drag and drop dashboard building tool, and has many powerful features. You have to spend time to learn it. So here you go, let's write some codes here.

Looping Through CSV Files

Unfortunately Power Query or let's say Power BI doesn't have a loop structure, and that is because of the functional structure of this language. However there are data structures such as Table and List that can be easily used with **each** singleton function to work exactly as a loop structure does. Here in this post I will get you through the process of looping into files in a directory and processing them all, and finally

combining them into a large big table. You will also learn some Power Query M functions through this process.

Where We've been in Previous Step

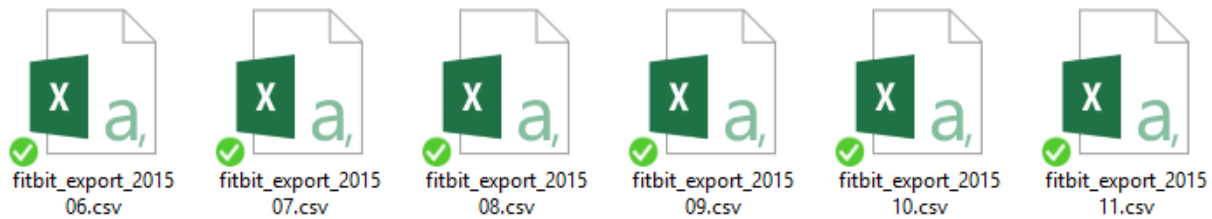
In previous step we ended up with below script

```
let
    Source =
    Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles\fitbit_export_201507.csv"),null,null,1252)}),
    #"Removed Top Rows" = Table.Skip(Source,1),
    #"Split Column by Delimiter" = Table.SplitColumn(#"Removed Top Rows", "Column1", Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv), {"Column1.1", "Column1.2", "Column1.3", "Column1.4", "Column1.5", "Column1.6", "Column1.7", "Column1.8", "Column1.9", "Column1.10"}),
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type text}, {"Column1.3", type text}, {"Column1.4", type text}, {"Column1.5", type text}, {"Column1.6", type text}, {"Column1.7", type text}, {"Column1.8", type text}, {"Column1.9", type text}, {"Column1.10", type text}}),
    #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
    #"Removed Blank Rows" = Table.SelectRows(#"Promoted Headers", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null}))),
    #"Changed Type1" = Table.TransformColumnTypes(#"Removed Blank Rows",{{"Steps", Int64.Type}, {"Calories Burned", Int64.Type}, {"Floors", Int64.Type}, {"Minutes Sedentary", Int64.Type}, {"Minutes Lightly Active", Int64.Type}, {"Minutes Fairly Active", Int64.Type}, {"Minutes Very Active", Int64.Type}, {"Activity Calories", Int64.Type}, {"Distance", type number}}),
    #"Added Custom" = Table.AddColumn(#"Changed Type1", "Dated", each Date.FromText([Date], "en-NZ")),
    #"Changed Type2" = Table.TransformColumnTypes(#"Added Custom",{{"Dated", type date}}),
    #"Removed Columns" = Table.RemoveColumns(#"Changed Type2", {"Date"}),
    #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns", {"Dated", "Calories Burned", "Steps", "Distance", "Floors", "Minutes Sedentary", "Minutes Lightly Active", "Minutes Fairly Active", "Minutes Very Active", "Activity Calories"}),
in
    #"Reordered Columns"
```

That generates table below from an exported CSV file of Fitbit activities

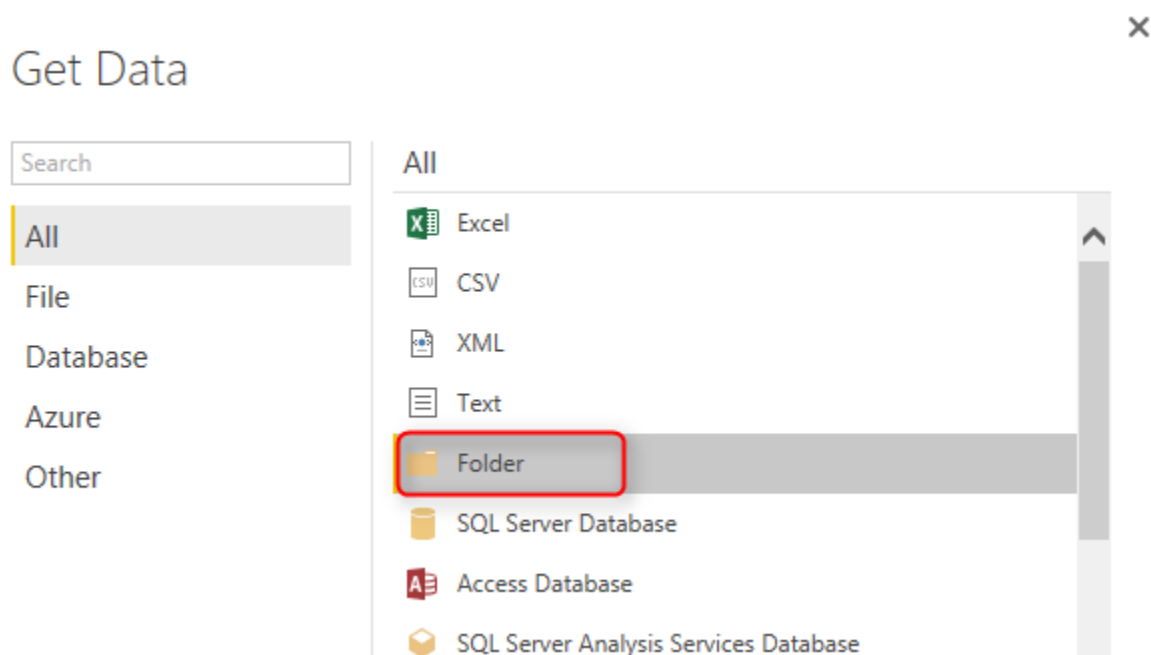
Dated	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity
7/1/2015	3953	13361	10.64	16	1058	196	71	87	
7/2/2015	3432	10818	8.76	10	726	224	19	45	
7/3/2015	4102	15537	12.38	16	701	237	55	88	
7/4/2015	3959	14490	11.54	6	685	255	46	69	
7/5/2015	4028	14652	11.68	3	1003	283	93	61	
7/6/2015	3538	12161	9.84	16	1144	168	76	52	
7/7/2015	3550	11103	8.99	17	785	250	24	41	
7/8/2015	3238	10029	8.22	10	1204	152	36	48	
7/9/2015	3994	13951	11.29	10	733	183	85	96	

Now There are multiple files in the source folder that I want to repeat the process for all of them;



Get Data From Folder

Let's start the process with reading all files in a folder. Power BI Desktop can do that easily. Through Get Data options choose from Folder and the select the folder containing source files as above.



This will load all files and their general information into a single table. There are columns for name of file, folder path, date created, modified, and etc. There is also a column named Content. This is an important column as it contains the whole content of each file in a Binary format.

= Folder.Files("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles")									
#	Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path	
1	Binary	fitbit_export_201506.csv	.csv	12/1/2015 3:29:56 PM	9/24/2015 12:29:44 AM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
2	Binary	fitbit_export_201507.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:49:38 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
3	Binary	fitbit_export_201508.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:48:52 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
4	Binary	fitbit_export_201509.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:48:11 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
5	Binary	fitbit_export_201510.csv	.csv	12/1/2015 3:29:56 PM	11/11/2015 10:47:09 PM	12/1/2015 3:29:56 PM	Record	C:\Users\Reza\SkyDi	
6	Binary	fitbit_export_201511.csv	.csv	12/1/2015 3:30:38 PM	12/1/2015 3:07:38 PM	12/1/2015 3:07:36 PM	Record	C:\Users\Reza\SkyDi	

Custom Function

So far we have list of all files and the script to convert their content into a desired table structure. We can create a custom function with that script and re-use it for each file's content. The process of creating custom functions are easy. I'm not going to explain every parts of creating custom function because that's a whole separate topic. If you are interested to read more about [Creating Custom Functions in Power Query read this post](#).

Go into Advanced Editor, and create the function as below highlighted section;

```
let
    FetchTable = (table) =>
        let
            Source = Table.FromColumns({Lines.FromBinary(table,null,null,1252)}),
            #"Removed Top Rows" = Table.Skip(Source,1),
            #"Split Column by Delimiter" = Table.SplitColumn(#"Removed Top Rows", "Column1", Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv), {"Column1.1", "Column1.2", "Column1.3", "Column1.4", "Column1.5", "Column1.6", "Column1.7", "Column1.8", "Column1.9", "Column1.10"}),
            #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type text}, {"Column1.3", type text}, {"Column1.4", type text}, {"Column1.5", type text}, {"Column1.6", type text}, {"Column1.7", type text}, {"Column1.8", type text}, {"Column1.9", type text}, {"Column1.10", type text}}),
            #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
            #"Removed Blank Rows" = Table.SelectRows(#"Promoted Headers", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null}))),
            #"Changed Type1" = Table.TransformColumnTypes(#"Removed Blank Rows",{{"Steps", Int64.Type}, {"Calories Burned", Int64.Type}, {"Floors", Int64.Type}, {"Minutes Sedentary", Int64.Type}, {"Minutes Lightly Active", Int64.Type}, {"Minutes Fairly Active", Int64.Type}, {"Minutes Very Active", Int64.Type}, {"Activity Calories", Int64.Type}, {"Distance", type number}}),
            #"Added Custom" = Table.AddColumn(#"Changed Type1", "Dated", each Date.FromText([Date], "en-NZ")),
            #"Changed Type2" = Table.TransformColumnTypes(#"Added Custom",{{"Dated", type date}}),
            #"Removed Columns" = Table.RemoveColumns(#"Changed Type2",{"Date"}),
```

```

        #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns",{ "Dated",
"Calories Burned", "Steps", "Distance", "Floors", "Minutes Sedentary", "Minutes Lightly Active", "Minutes
Fairly Active", "Minutes Very Active", "Activity Calories"})
        in
        #"Reordered Columns",

Files = Folder.Files("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles")
in
Files

```

Note that all the highlighted part from line 2 to line 16 in code above is the code for the custom function. The first line of function (line 2 in above script) contains name of the function; **FetchTable**, and the input parameter; **table**. From line 3 to line 16 is all the script we've copied from [first step](#) of this exercise.

To Make sure that the function works properly you can call it as below;

```

FetchTable(Files[{"Folder
Path"="C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles\",Name="fitbit_export_201509.cs
v"}][Content])

```

Add above single line of code in the IN section of script instead of Files. and you will see the result table as below:

	Dated	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories
1	9/1/2015	3683	15620	12.81	19	852	210	25	88	2138
2	9/2/2015	3239	9065	6.88	24	975	191	24	50	1572
3	9/3/2015	3443	13903	11.62	21	1151	207	25	57	1868
4	9/4/2015	4045	16437	12.54	37	661	220	83	105	2599
5	9/5/2015	4091	17423	14.06	20	1048	237	66	89	2674
6	9/6/2015	3819	13478	10.82	10	596	245	69	67	2356
7	9/7/2015	3579	11851	9.55	16	1121	240	29	50	2019
8	9/8/2015	3218	9437	7.16	18	759	235	18	41	1613
9	9/9/2015	3614	13441	11.26	41	771	178	25	85	2007
10	9/10/2015	3504	12736	10.34	19	847	197	29	70	1911

So this shows the function definition and function call work perfectly. Now let's loop through files.

EACH Singleton Function

EACH is a singleton function, which means it has one input parameter and one output. It can be applied on EACH record in a table or EACH item in a list. with using that you can apply calculations, data transformations, adding custom columns and many other operations. For our case I want to use EACH to apply FetchTable function on every Content value of the Files table. Each is easy to use as below;

```
Files = Folder.Files("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles"),
```

```
TableAdded= Table.AddColumn(Files,"Data",each FetchTable([Content])),
DataColumnSelected=Table.SelectColumns(TableAdded,{"Data"})
in
DataColumnSelected
```

As you see the single line above and specially the section ***each FetchTable([Content])*** will apply FetchTable function for every Content value in each record of the table. and as a result we will have a new column called Data that includes tables. line 3 of script above just select Data column (because we don't need other columns). Here is the result;



	Data
1	Table
2	Table
3	Table
4	Table
5	Table
6	Table

As you see in the result set above we have one table with one column (named Data), and it includes a Table in every item. This is the table that we want to use, so we have to combine all of these tables into one table. Fortunately there is a function for it in Power Query called Table.Combine .

Combining, Appending, or UNION of Tables

[Table.Combine](#) does the UNION of multiple tables into one table. This function get a LIST of tables (they should be all with the same structure), and returns a single table which is appended version (or UNION version) of all the table. Here is the definition of this function;

```
Table.Combine(tables as list) as table
```

As you mentioned probably this function needs a LIST as an input parameter. And so far we have built a TABLE (single column table) which needs to be converted to list. There are different methods of converting a table into a list. I'll show you one method here. I first fetch only a single column list from this table with [Table.ToColumns](#) function, and then I get the only first value of the list with [List.First](#) function. Finally I use the Table.Combine with the list parameter, here's the code:

```
Files = Folder.Files("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles"),
TableAdded= Table.AddColumn(Files,"Data",each FetchTable([Content])),
DataColumnSelected=Table.SelectColumns(TableAdded,{"Data"}),
```



```

TablesColumn=Table.ToColumns(DataColumnSelected),
TablesList=List.First(TablesColumn),
AllTables=Table.Combine(TablesList)
in
AllTables

```

Line 4 converts the table to a list, but a list that has another list within. Because a table might have more than one column.

Line 5 gets the first item of the list, which would be the first column of the table. For our example as we have only a single column then only that column would be fetched.

Line 6 finally combines all the tables in the list into one large table.

And the result:

	Dated	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories
55	7/25/2015	3736	16374	13.54	14	576	248	30	68	2208
56	7/26/2015	3453	13676	11.18	4	576	232	45	55	1933
57	7/27/2015	3298	12026	9.77	14	775	173	44	52	1684
58	7/28/2015	3654	17401	14.14	17	1140	163	37	100	2115
59	7/29/2015	3588	14711	11.82	15	681	241	40	58	2062
60	7/30/2015	3059	8671	6.58	30	790	231	7	20	1417
61	7/31/2015	3562	14522	11.05	14	673	187	58	80	2023
62	8/1/2015	4090	18259	14.81	5	996	280	65	99	2693
63	8/2/2015	3412	10894	8.82	10	662	193	56	52	1841
64	8/3/2015	3311	13734	11	19	724	170	24	72	1693
65	8/4/2015	3017	7562	5.74	14	817	176	29	37	1376
66	8/5/2015	3278	11924	10.07	9	809	181	14	48	1635
67	8/6/2015	3142	11182	9.04	18	783	159	52	43	1519
68	8/7/2015	3638	12549	9.61	14	776	212	55	83	2104
69	8/8/2015	3980	16805	13.82	2	700	241	65	83	2493

You can see that I have now all tables combined as I have all months together in this final table.

Here is the full script of this demo so far:

```

let
    FetchTable = (table) =>
        let
            Source = Table.FromColumns({Lines.FromBinary(table,null,null,1252)}),
            #"Removed Top Rows" = Table.Skip(Source,1),
            #"Split Column by Delimiter" = Table.SplitColumn(#"Removed Top Rows", "Column1", Splitter.SplitTextByDelimiter(",", QuoteStyle.Csv), {"Column1.1", "Column1.2", "Column1.3", "Column1.4", "Column1.5", "Column1.6", "Column1.7", "Column1.8", "Column1.9", "Column1.10"}),
            #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type text}, {"Column1.3", type text}, {"Column1.4", type text}, {"Column1.5", type text}, {"Column1.6", type text}, {"Column1.7", type text}, {"Column1.8", type text}, {"Column1.9", type text}, {"Column1.10", type text}}),
            #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),

```

```

        #"Removed Blank Rows" = Table.SelectRows(#"Promoted Headers", each not
List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null}))),
        #"Changed Type1" = Table.TransformColumnTypes(#"Removed Blank Rows",{{"Steps",
Int64.Type}, {"Calories Burned", Int64.Type}, {"Floors", Int64.Type}, {"Minutes Sedentary", Int64.Type},
{"Minutes Lightly Active", Int64.Type}, {"Minutes Fairly Active", Int64.Type}, {"Minutes Very Active",
Int64.Type}, {"Activity Calories", Int64.Type}, {"Distance", type number}}),
        #"Added Custom" = Table.AddColumn(#"Changed Type1", "Dated", each
Date.FromText([Date],"en-NZ")),
        #"Changed Type2" = Table.TransformColumnTypes(#"Added Custom",{{"Dated", type
date}}),
        #"Removed Columns" = Table.RemoveColumns(#"Changed Type2",{"Date"}),
        #"Reordered Columns" = Table.ReorderColumns(#"Removed Columns",{{"Dated",
"Calories Burned", "Steps", "Distance", "Floors", "Minutes Sedentary", "Minutes Lightly Active", "Minutes
Fairly Active", "Minutes Very Active", "Activity Calories"}}
in
        #"Reordered Columns",

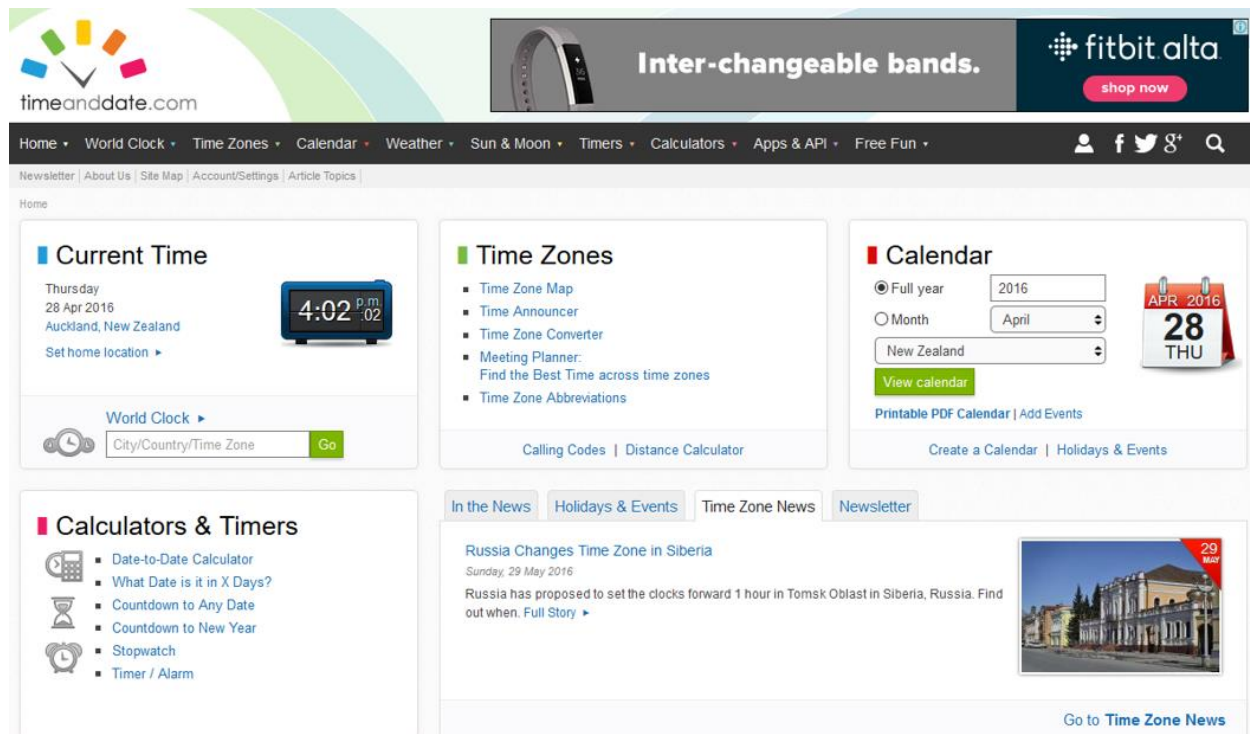
Files = Folder.Files("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles"),
TableAdded= Table.AddColumn(Files,"Data",each FetchTable([Content])),
DataColumnSelected=Table.SelectColumns(TableAdded,{"Data"}),
TablesColumn=Table.ToColumns(DataColumnSelected),
TablesList=List.First(TablesColumn),
AllTables=Table.Combine(TablesList)
in
    AllTables

```

That ends this step. You've learned some scripting skills here today, and I hope you've enjoyed it. With few lines of scripting like this you can add massive power to your queries, you can automate things and get rid of manual extra work. Power Query is one of the most power full components of Power BI that gives you ability to apply many transformations and data mash up. The next step will be about Visualization of the data we've prepared so far, so stay tuned!

Power Query Not for BI: Event Date and Time Scheduler - Part 1

Published Date: May 13, 2016



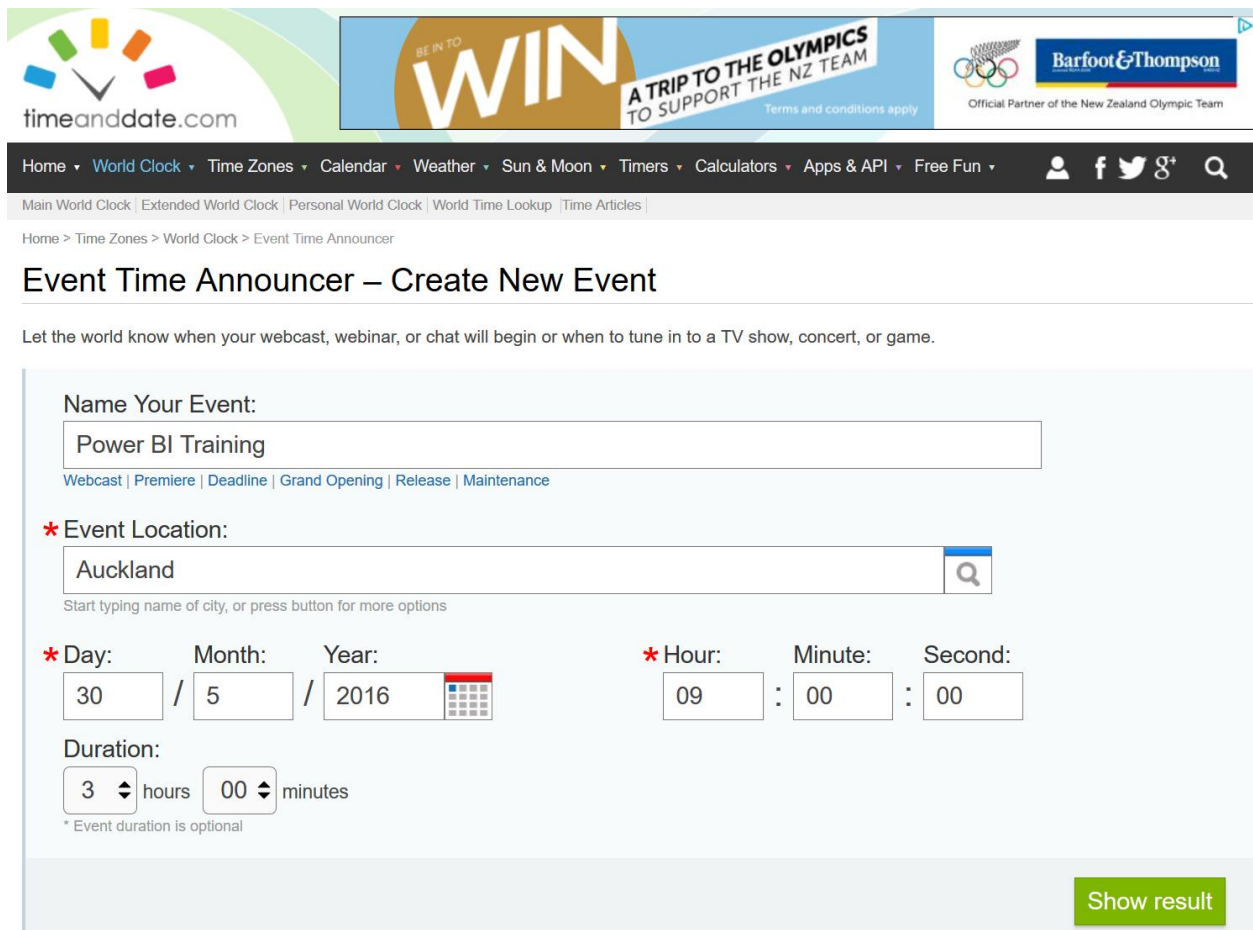
[Previously](#) you have read how to use Power Query as the data transformation component of Power BI and you [learned how to use it in BI applications](#). You can read more about Power Query usage in BI in [Power BI online book from Rookie to Rock Star](#). In this series I will explain how to use Power Query for non-BI usage. This data transformation tool can be used anywhere to use input parameters, fetch the data from web or any other data sources, mash up the data to form the required shape, and automate the process. In this series I will show you a real use case of using Power Query for non-BI application.

Why?

As an introduction to this series, I want to take you to the path that leads me to use Power Query here. You might be aware that I am teaching Power BI courses, and most

of my courses are online and Live. This means that courses are not recorded videos, it is me on the other side of the line with full interactive audio and video experience with students with Go2Meeting application. Students connecting to me from other places in the world. So I do need an event date/time scheduler that I can announce date and time of the event in different time zones.

Fortunately there is a very good [website](#) that helps to find a date/time in different time zones. In this website I can set my input parameters as the date/time of my event locally (in my city), and name of the event, and duration.



The screenshot shows the 'Event Time Announcer – Create New Event' page on timeanddate.com. The form includes the following fields and options:

- Name Your Event:** A text input field containing 'Power BI Training'. Below it are links for 'Webcast', 'Premiere', 'Deadline', 'Grand Opening', 'Release', and 'Maintenance'.
- * Event Location:** A text input field containing 'Auckland' with a search button. Below it is the instruction: 'Start typing name of city, or press button for more options'.
- * Day:** A dropdown menu showing '30'.
- Month:** A dropdown menu showing '5'.
- Year:** A text input field showing '2016' with a calendar icon.
- * Hour:** A dropdown menu showing '09'.
- Minute:** A dropdown menu showing '00'.
- Second:** A dropdown menu showing '00'.
- Duration:** A section with '3' hours and '00' minutes, each in a dropdown menu.
- Below the duration is the text: '* Event duration is optional'.
- A green 'Show result' button is located at the bottom right of the form.

Then the website produces a result set of important cities in different time zones with the date and time of the event in their time zones. That's brilliant, isn't it?

[Africa](#) | [North America](#) | [South America](#) | [Asia](#) | [Australia/Pacific](#) | [Europe](#)

The [Event Time Announcer](#) converts the time of your planned event to local times all over the world for easy sharing and linking.

Event Times around the world					
Sort by:	City			Cities shown:	Most Popular (144)
Location	Start time	End time	Location	Start time	End time
Accra	Sun 9:00 p.m.	Sun 12:00 Midn	Kiritimati	Mon 11:00 a.m.	Mon 2:00 p.m.
Addis Ababa	Sun 12:00 Midn	Mon 3:00 a.m.	Kolkata	Mon 2:30 a.m.	Mon 5:30 a.m.
Adelaide	Mon 6:30 a.m.	Mon 9:30 a.m.	Kuala Lumpur	Mon 5:00 a.m.	Mon 8:00 a.m.
Algiers	Sun 10:00 p.m.	Mon 1:00 a.m.	Kuwait City	Sun 12:00 Midn	Mon 3:00 a.m.
Almaty	Mon 3:00 a.m.	Mon 6:00 a.m.	Kyiv *	Sun 12:00 Midn	Mon 3:00 a.m.
Amman *	Sun 12:00 Midn	Mon 3:00 a.m.	La Paz	Sun 5:00 p.m.	Sun 8:00 p.m.
Amsterdam *	Sun 11:00 p.m.	Mon 2:00 a.m.	Lagos	Sun 10:00 p.m.	Mon 1:00 a.m.
Anadyr	Mon 9:00 a.m.	Mon 12:00 Noon	Lahore	Mon 2:00 a.m.	Mon 5:00 a.m.
Anchorage *	Sun 1:00 p.m.	Sun 4:00 p.m.	Las Vegas *	Sun 2:00 p.m.	Sun 5:00 p.m.
Ankara *	Sun 12:00 Midn	Mon 3:00 a.m.	Lima	Sun 4:00 p.m.	Sun 7:00 p.m.
Antananarivo	Sun 12:00 Midn	Mon 3:00 a.m.	Lisbon *	Sun 10:00 p.m.	Mon 1:00 a.m.
Asuncion	Sun 5:00 p.m.	Sun 8:00 p.m.	London *	Sun 10:00 p.m.	Mon 1:00 a.m.
Athens *	Sun 12:00 Midn	Mon 3:00 a.m.	Los Angeles *	Sun 2:00 p.m.	Sun 5:00 p.m.
Atlanta *	Sun 5:00 p.m.	Sun 8:00 p.m.	Madrid *	Sun 11:00 p.m.	Mon 2:00 a.m.
Auckland	Mon 9:00 a.m.	Mon 12:00 Noon	Managua	Sun 3:00 p.m.	Sun 6:00 p.m.
Baghdad	Sun 12:00 Midn	Mon 3:00 a.m.	Manila	Mon 5:00 a.m.	Mon 8:00 a.m.
Bangkok	Mon 4:00 a.m.	Mon 7:00 a.m.	Melbourne	Mon 7:00 a.m.	Mon 10:00 a.m.
Barcelona *	Sun 11:00 p.m.	Mon 2:00 a.m.	Mexico City *	Sun 4:00 p.m.	Sun 7:00 p.m.

The main issue here is that; I want to use this list in my course announcement web page, and list above is overwhelming! If I copy and paste list above, it will make my page so long. Audience will lose interest to read such a long page to find their city and local time zone. What I need is this:

- List of all above cities with the event date time on their time zone, **categorized by time**. That means; Los Angeles, San Francisco, Las Vegas, Seattle, and Portland, should be all listed in one row of my result set table.

Start time	End time	Cities
Mon 1:00 a.m.	Mon 4:00 a.m.	Accra, Reykjavik
Mon 4:00 a.m.	Mon 7:00 a.m.	Addis Ababa, Amman, Ankara, Antananarivo, Athens, Baghdad, Beirut
Mon 10:30 a.m.	Mon 1:30 p.m.	Adelaide, Darwin
Mon 2:00 a.m.	Mon 5:00 a.m.	Algiers, Casablanca, Dublin, Kinshasa, Lagos, Lisbon, London
Mon 7:00 a.m.	Mon 10:00 a.m.	Almaty, Dhaka
Mon 3:00 a.m.	Mon 6:00 a.m.	Amsterdam, Barcelona, Belgrade, Berlin, Brussels, Budapest, Cairo, Ca
Mon 1:00 p.m.	Mon 4:00 p.m.	Anadyr, Auckland, Suva
Sun 5:00 p.m.	Sun 8:00 p.m.	Anchorage
Sun 9:00 p.m.	Sun 12:00 Midn	Asuncion, Atlanta, Boston, Caracas, Columbus, Detroit, Havana, Indian
Mon 8:00 a.m.	Mon 11:00 a.m.	Bangkok, Hanoi, Jakarta
Mon 9:00 a.m.	Mon 12:00 Noon	Beijing, Hong Kong, Kuala Lumpur, Manila, Perth, Shanghai, Singapore,
Mon 6:30 a.m.	Mon 9:30 a.m.	Bengaluru, Kolkata, Mumbai, New Delhi
Sun 8:00 p.m.	Sun 11:00 p.m.	Bogota, Chicago, Dallas, Houston, Kingston, Lima, Mexico City, Minnea
Sun 10:00 p.m.	Mon 1:00 a.m.	Brasilia, Buenos Aires, Halifax, Montevideo, Rio de Janeiro, Santiago, S
Mon 11:00 a.m.	Mon 2:00 p.m.	Brisbane, Canberra, Melbourne, Sydney
Sun 7:00 p.m.	Sun 10:00 p.m.	Calgary, Denver, Edmonton, Guatemala, Managua, Salt Lake City, San S
Mon 5:00 a.m.	Mon 8:00 a.m.	Dubai
Sun 3:00 p.m.	Sun 6:00 p.m.	Honolulu
Mon 6:00 a.m.	Mon 9:00 a.m.	Islamabad, Karachi, Lahore, Tashkent
Mon 5:30 a.m.	Mon 8:30 a.m.	Kabul, Tehran
Mon 6:45 a.m.	Mon 9:45 a.m.	Kathmandu
Mon 3:00 p.m.	Mon 6:00 p.m.	Kiritimati
Sun 6:00 p.m.	Sun 9:00 p.m.	Las Vegas, Los Angeles, Phoenix, San Francisco, Seattle, Vancouver
Mon 10:00 a.m.	Mon 1:00 p.m.	Seoul, Tokyo
Sun 10:30 p.m.	Mon 1:30 a.m.	St. John's
Mon 7:30 a.m.	Mon 10:30 a.m.	Yangon

26. I want this list to be in Excel format, so I can send it to my mailing list or attach it as an excel table
27. I don't want to produce the data from website. I would like to change parameters in my Excel file, and it automatically fetches data from this website, and load result set back to my Excel spreadsheet.

So as a result I decided to use Power Query, because with Power Query I can;

- Have input parameters in an Excel sheet for local date/time, and duration of the event.
- Browse web URL and fetch result set.
- Mash up the output data to produce my desired data set.
- Load result set into Excel.

So Let's now start building this solution step by step.

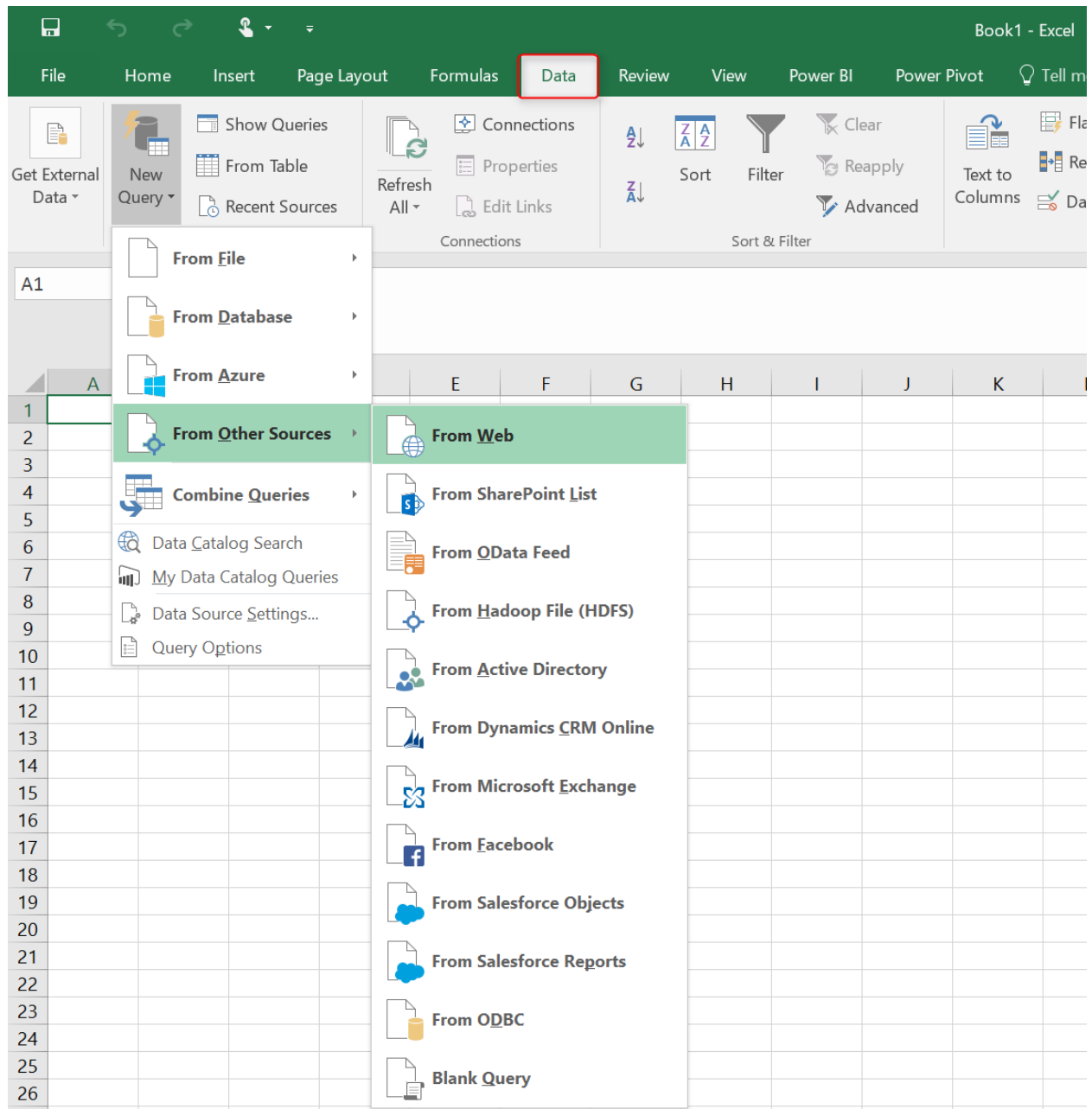
What You Will Learn?

By completing this scenario, you will learn a real-world use case of Power Query for non-BI solution. You will learn using Power Query for;

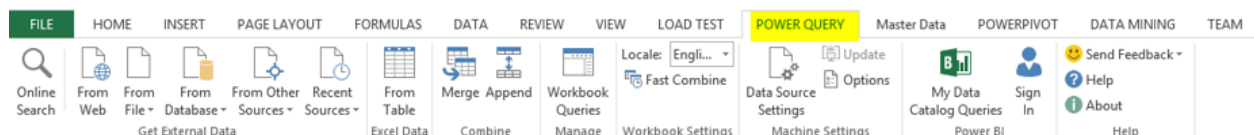
8. Defining a table for input parameters in Excel sheet
9. Using Power Query to read values from that table as input parameter and build web URL for timeanddate.com web query.
10. Querying web URL and Fetching the data
11. Data Manipulation with Power Query such as; Removing Columns, Replacing Values, Appending Tables, Grouping Rows, Adding Custom Columns, Changing Tables to Column, Columns to Lists, Lists to Tables, and Concatenating Table Columns with Comma.
12. Automating the Power Query solution to take the parameters and produce result set when you hit Refresh button.

Prerequisite

This example has been done with Excel 2016 which has Power Query embedded. So if you have Excel 2016 you don't need to install anything. If you are using Excel 2013 or 2010 you need to download and install Power Query for Excel Add-in. Power Query editor experience in all of these versions of Excel are the same. The only difference is that Power Query menu options in Excel 2016 is located under Data Tab;



And in Excel 2013 and 2010 are located under Power Query Tab;

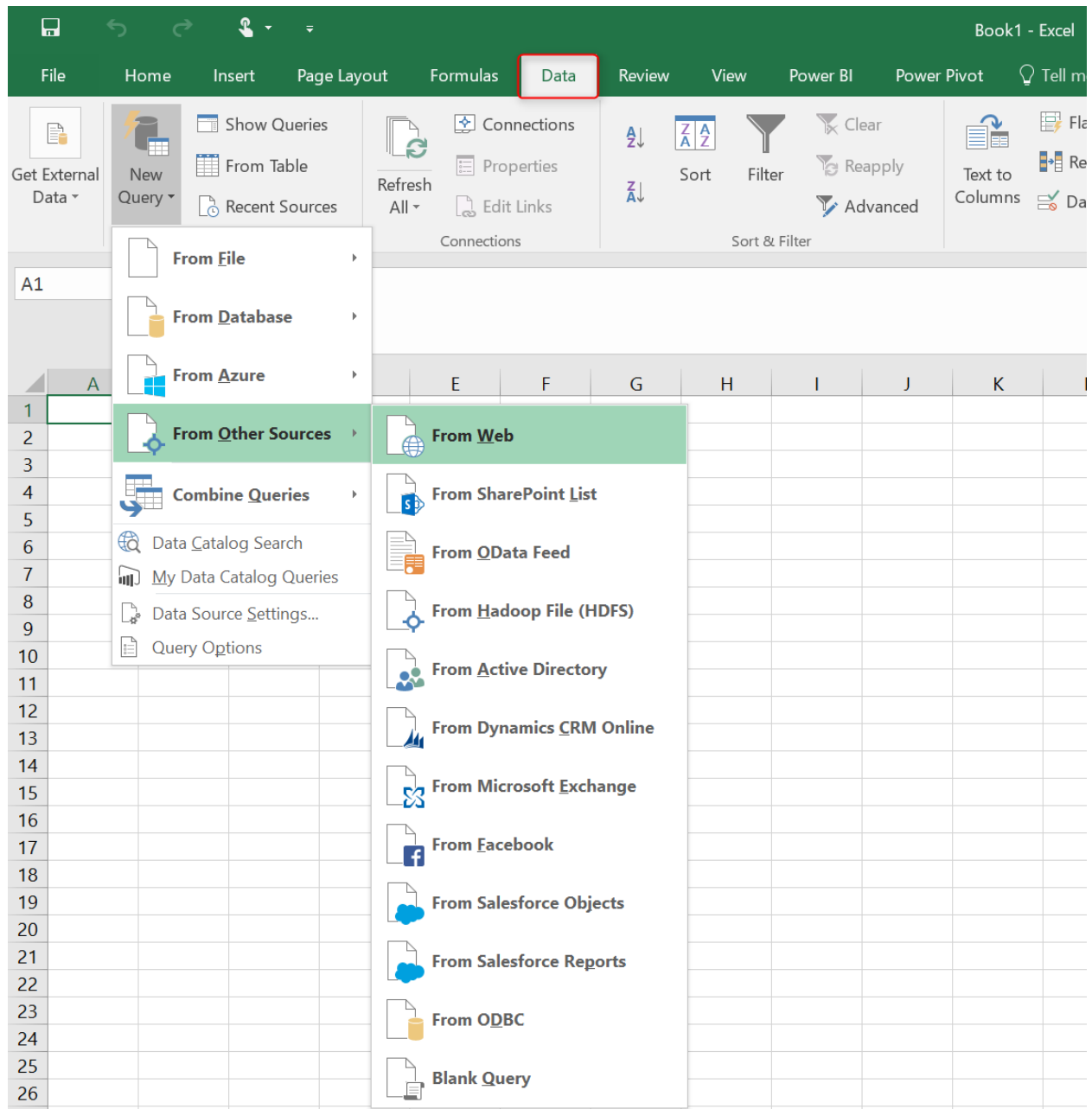


Get Data from Web

I'll start by the easy part of the work, which is using a static web URL like this:

<http://www.timeanddate.com/worldclock/fixedtime.html?msg=Power+BI+Training&iso=20160530T09&p1=22&ah=3>

In Power Query I can fetch tables produced in this web page. Follow the menu as: Go to Data Tab (In excel 2013/2010 Power Query Tab), New Query, From Other Data Sources, and then From Web



Enter the URL mentioned above in the dialog box

From Web

Enter a Web page URL.

URL

ixedtime.html?msg=Power+BI+Training&iso=20160530T09&p1=22&ah=3

OK

Cancel

Power Query is smart enough to separate each table, and It shows you a preview of each table. Choose the table that has values with cities and date/times.

Navigators

☐ Select multiple items

Show All | Show Selected [1]

http://www.timeanddate.com/worldclock/fixed...

Document

Table 0

Table 0

Header	Location	Start time	End time
Sort by: Event Times around the world Cities shown:	Accra	Sun 9:00 p.m.	Sun 12:00 Midn
Sort by: Event Times around the world Cities shown:	Addis Ababa	Sun 12:00 Midn	Mon 3:00 a.m.
Sort by: Event Times around the world Cities shown:	Adelaide	Mon 6:30 a.m.	Mon 9:30 a.m.
Sort by: Event Times around the world Cities shown:	Algiers	Sun 10:00 p.m.	Mon 1:00 a.m.
Sort by: Event Times around the world Cities shown:	Almaty	Mon 3:00 a.m.	Mon 6:00 a.m.
Sort by: Event Times around the world Cities shown:	Amman*	Sun 12:00 Midn	Mon 3:00 a.m.
Sort by: Event Times around the world Cities shown:	Amsterdam*	Sun 11:00 p.m.	Mon 2:00 a.m.
Sort by: Event Times around the world Cities shown:	Anadyr	Mon 9:00 a.m.	Mon 12:00 Noc
Sort by: Event Times around the world Cities shown:	Anchorage*	Sun 1:00 p.m.	Sun 4:00 p.m.
Sort by: Event Times around the world Cities shown:	Ankara*	Sun 12:00 Midn	Mon 3:00 a.m.
Sort by: Event Times around the world Cities shown:	Antananarivo	Sun 12:00 Midn	Mon 3:00 a.m.

Load

Edit

Cancel

I can see in preview above that there are some extra columns, and also I need to apply some transformations. So I Click on Edit to lunch query editor for further data manipulation.

Table 0 - Query Editor

File Home Transform Add Column View

Close & Load Refresh Preview Advanced Editor Choose Columns Remove Columns Keep Rows Remove Rows Remove Duplicates Remove Errors Split Column Group By Data Type: Text Use First Row As Headers Replace Values Merge Queries Append Queries Combine Binaries New Source Recent Sources

1 query Table 0

Table.TransformColumnTypes(Data0,{{"Header", type text}, {"Location", type text}, {"Start time", type text}, {"End time", type text}, {"Location2", type text}, {"Start time2", type text}, {"End time2", type text}}

	Header	Location	Start time	End time	Location2	Start time2	End time2
1	Sort by: Event Times around the world Cities shown:	Accra	Sun 9:00 p.m.	Sun 12:00 Midn	Kiritimati	Mon 11:00 a.m.	Mon 2:00 p.m.
2	Sort by: Event Times around the world Cities shown:	Addis Ababa	Sun 12:00 Midn	Mon 3:00 a.m.	Kolkata	Mon 2:30 a.m.	Mon 5:30 a.m.
3	Sort by: Event Times around the world Cities shown:	Adelaide	Mon 6:30 a.m.	Mon 9:30 a.m.	Kuala Lumpur	Mon 5:00 a.m.	Mon 8:00 a.m.
4	Sort by: Event Times around the world Cities shown:	Algiers	Sun 10:00 p.m.	Mon 1:00 a.m.	Kuwait City	Sun 12:00 Midn	Mon 3:00 a.m.
5	Sort by: Event Times around the world Cities shown:	Almaty	Mon 3:00 a.m.	Mon 6:00 a.m.	Kyiv*	Sun 12:00 Midn	Mon 3:00 a.m.
6	Sort by: Event Times around the world Cities shown:	Amman*	Sun 12:00 Midn	Mon 3:00 a.m.	La Paz	Sun 5:00 p.m.	Sun 8:00 p.m.
7	Sort by: Event Times around the world Cities shown:	Amsterdam*	Sun 11:00 p.m.	Mon 2:00 a.m.	Lagos	Sun 10:00 p.m.	Mon 1:00 a.m.
8	Sort by: Event Times around the world Cities shown:	Anadyr	Mon 9:00 a.m.	Mon 12:00 Noon	Lahore	Mon 2:00 a.m.	Mon 5:00 a.m.
9	Sort by: Event Times around the world Cities shown:	Anchorage*	Sun 1:00 p.m.	Sun 4:00 p.m.	Las Vegas*	Sun 2:00 p.m.	Sun 5:00 p.m.
10	Sort by: Event Times around the world Cities shown:	Ankara*	Sun 12:00 Midn	Mon 3:00 a.m.	Lima	Sun 4:00 p.m.	Sun 7:00 p.m.
11	Sort by: Event Times around the world Cities shown:	Antananarivo	Sun 12:00 Midn	Mon 3:00 a.m.	Lisbon*	Sun 10:00 p.m.	Mon 1:00 a.m.
12	Sort by: Event Times around the world Cities shown:	Asuncion	Sun 5:00 p.m.	Sun 8:00 p.m.	London*	Sun 10:00 p.m.	Mon 1:00 a.m.
13	Sort by: Event Times around the world Cities shown:	Athens*	Sun 12:00 Midn	Mon 3:00 a.m.	Los Angeles*	Sun 2:00 p.m.	Sun 5:00 p.m.

Query Settings

PROPERTIES

Name

Table 0

All Properties

APPLIED STEPS

Source

Navigation

Changed Type

Apply Basic Transformations

When query editor lunches and Power Query loads the preview of table, it also automatically applies some data type changes as the very first step. Here is the data set when it loads in query editor for the first time;

Table.TransformColumnTypes(Data01,{{"Header", type text}, {"Location", type text}, {"Start time", type text}, {"End time", type text}, {"Location2", type text}, {"Start time2", type text}, {"End time2", type text}}

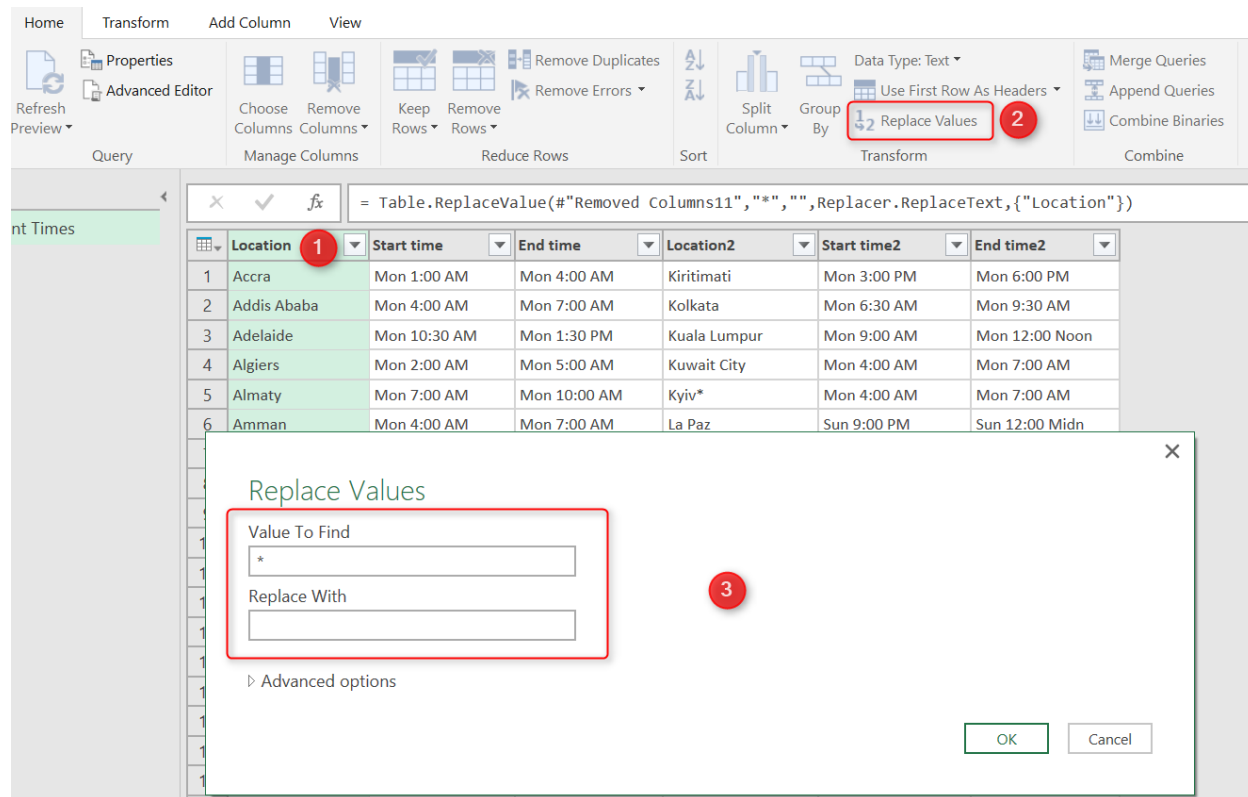
	Header	Location	Start time	End time	Location2	Start time2	End time2
1	Sort by: Event Times around the world Cities shown:	Accra	Mon 1:00 AM	Mon 4:00 AM	Kiritimati	Mon 3:00 PM	Mon 6:00 PM
2	Sort by: Event Times around the world Cities shown:	Addis Ababa	Mon 4:00 AM	Mon 7:00 AM	Kolkata	Mon 6:30 AM	Mon 9:30 AM
3	Sort by: Event Times around the world Cities shown:	Adelaide	Mon 10:30 AM	Mon 1:30 PM	Kuala Lumpur	Mon 9:00 AM	Mon 12:00 Noon
4	Sort by: Event Times around the world Cities shown:	Algiers	Mon 2:00 AM	Mon 5:00 AM	Kuwait City	Mon 4:00 AM	Mon 7:00 AM
5	Sort by: Event Times around the world Cities shown:	Almaty	Mon 7:00 AM	Mon 10:00 AM	Kyiv*	Mon 4:00 AM	Mon 7:00 AM
6	Sort by: Event Times around the world Cities shown:	Amman*	Mon 4:00 AM	Mon 7:00 AM	La Paz	Sun 9:00 PM	Sun 12:00 Midn
7	Sort by: Event Times around the world Cities shown:	Amsterdam*	Mon 3:00 AM	Mon 6:00 AM	Lagos	Mon 2:00 AM	Mon 5:00 AM
8	Sort by: Event Times around the world Cities shown:	Anadyr	Mon 1:00 PM	Mon 4:00 PM	Lahore	Mon 6:00 AM	Mon 9:00 AM
9	Sort by: Event Times around the world Cities shown:	Anchorage*	Sun 5:00 PM	Sun 8:00 PM	Las Vegas*	Sun 6:00 PM	Sun 9:00 PM

There is a Header column which is not giving me any useful information, So I'll remove it by right click on the column and Remove column

Table.TransformColumnTypes(Data01,{{"Header", type text}, {"Location", type text}, {"Start time", type text}, {"End time", type text}, {"Location2", type text}, {"Start time2", type text}, {"End time2", type text}}

	Header	Location	Start time	End time	Location2	Start time2	End time2
1	Sort by: Event Times around the world Cities shown:	Accra	Mon 1:00 AM	Mon 4:00 AM	Kiritimati	Mon 3:00 PM	Mon 6:00 PM
2	Sort by: Event Times around the world Cities shown:	Addis Ababa	Mon 4:00 AM	Mon 7:00 AM	Kolkata	Mon 6:30 AM	Mon 9:30 AM
3	Sort by: Event Times around the world Cities shown:	Adelaide	Mon 10:30 AM	Mon 1:30 PM	Kuala Lumpur	Mon 9:00 AM	Mon 12:00 Noon
4	Sort by: Event Times around the world Cities shown:	Algiers	Mon 2:00 AM	Mon 5:00 AM	Kuwait City	Mon 4:00 AM	Mon 7:00 AM
5	Sort by: Event Times around the world Cities shown:	Almaty	Mon 7:00 AM	Mon 10:00 AM	Kyiv*	Mon 4:00 AM	Mon 7:00 AM
6	Sort by: Event Times around the world Cities shown:	Amman*	Mon 4:00 AM	Mon 7:00 AM	La Paz	Sun 9:00 PM	Sun 12:00 Midn
7	Sort by: Event Times around the world Cities shown:	Amsterdam*	Mon 3:00 AM	Mon 6:00 AM	Lagos	Mon 2:00 AM	Mon 5:00 AM

There are also some * characters in cities names (at the end of some cities, such as Barcelona*). So I remove them by clicking on the Location Column, then choosing Replace Values from icons under Home menu, and then in replace values dialog box I replace * with blank.



Home Transform Add Column View

Refresh Preview Advanced Editor

Choose Columns Remove Columns Manage Columns

Keep Rows Remove Rows Reduce Rows

Remove Duplicates Remove Errors

Sort Split Column Group By

Data Type: Text Use First Row As Headers

Replace Values

Merge Queries Append Queries Combine Binaries

Query

nt Times

Location Start time End time Location2 Start time2 End time2

1 Accra Mon 1:00 AM Mon 4:00 AM Kiritimati Mon 3:00 PM Mon 6:00 PM

2 Addis Ababa Mon 4:00 AM Mon 7:00 AM Kolkata Mon 6:30 AM Mon 9:30 AM

3 Adelaide Mon 10:30 AM Mon 1:30 PM Kuala Lumpur Mon 9:00 AM Mon 12:00 Noon

4 Algiers Mon 2:00 AM Mon 5:00 AM Kuwait City Mon 4:00 AM Mon 7:00 AM

5 Almaty Mon 7:00 AM Mon 10:00 AM Kyiv* Mon 4:00 AM Mon 7:00 AM

6 Amman Mon 4:00 AM Mon 7:00 AM La Paz Sun 9:00 PM Sun 12:00 Midn

Replace Values

Value To Find

*

Replace With

Advanced options

OK Cancel

I'll do the same step again Location2 Column this time. Here is what my result set looks like so far;



Location Start time End time Location2 Start time2 End time2

1 Accra Sun 9:00 p.m. Sun 12:00 Midn Kiritimati Mon 11:00 a.m. Mon 2:00 p.m.

2 Addis Ababa Sun 12:00 Midn Mon 3:00 a.m. Kolkata Mon 2:30 a.m. Mon 5:30 a.m.

3 Adelaide Mon 6:30 a.m. Mon 9:30 a.m. Kuala Lumpur Mon 5:00 a.m. Mon 8:00 a.m.

4 Algiers Sun 10:00 p.m. Mon 1:00 a.m. Kuwait City Sun 12:00 Midn Mon 3:00 a.m.

5 Almaty Mon 3:00 a.m. Mon 6:00 a.m. Kyiv Sun 12:00 Midn Mon 3:00 a.m.

6 Amman Sun 12:00 Midn Mon 3:00 a.m. La Paz Sun 5:00 p.m. Sun 8:00 p.m.

7 Amsterdam Sun 11:00 p.m. Mon 2:00 a.m. Lagos Sun 10:00 p.m. Mon 1:00 a.m.

8 Anadyr Mon 9:00 a.m. Mon 12:00 Noon Lahore Mon 2:00 a.m. Mon 5:00 a.m.

9 Anchorage Sun 1:00 p.m. Sun 4:00 p.m. Las Vegas Sun 2:00 p.m. Sun 5:00 p.m.

10 Ankara Sun 12:00 Midn Mon 3:00 a.m. Lima Sun 4:00 p.m. Sun 7:00 p.m.

11 Antananarivo Sun 12:00 Midn Mon 3:00 a.m. Lisbon Sun 10:00 p.m. Mon 1:00 a.m.

12 Asuncion Sun 5:00 p.m. Sun 8:00 p.m. London Sun 10:00 p.m. Mon 1:00 a.m.

13 Athens Sun 12:00 Midn Mon 3:00 a.m. Los Angeles Sun 2:00 p.m. Sun 5:00 p.m.

14 Atlanta Sun 5:00 p.m. Sun 8:00 p.m. Madrid Sun 11:00 p.m. Mon 2:00 a.m.

Query Settings

PROPERTIES

Name

Event Times

All Properties

APPLIED STEPS

Source

Navigation

Changed Type

Removed Columns

Replaced Value

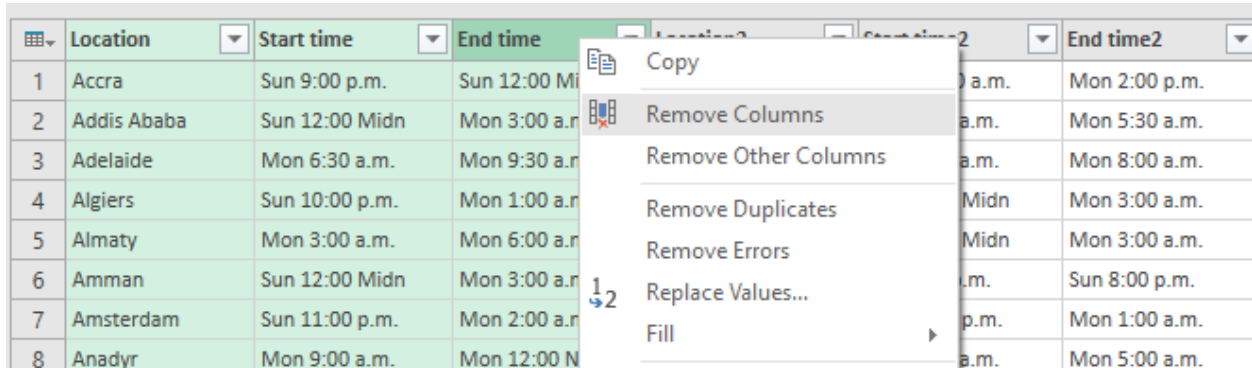
Replaced Value1

As you can see in screenshot above all applied steps so far are listed in the right pane (numbered 3 in screenshot above). This is one of the great features of Power Query that allows you rollback changes anytime you want and check preview of query for specific step.

In screenshot above I have to set of columns; first one numbered 1, and the second one numbered 2. These are not repetitive values. You can check city names in columns

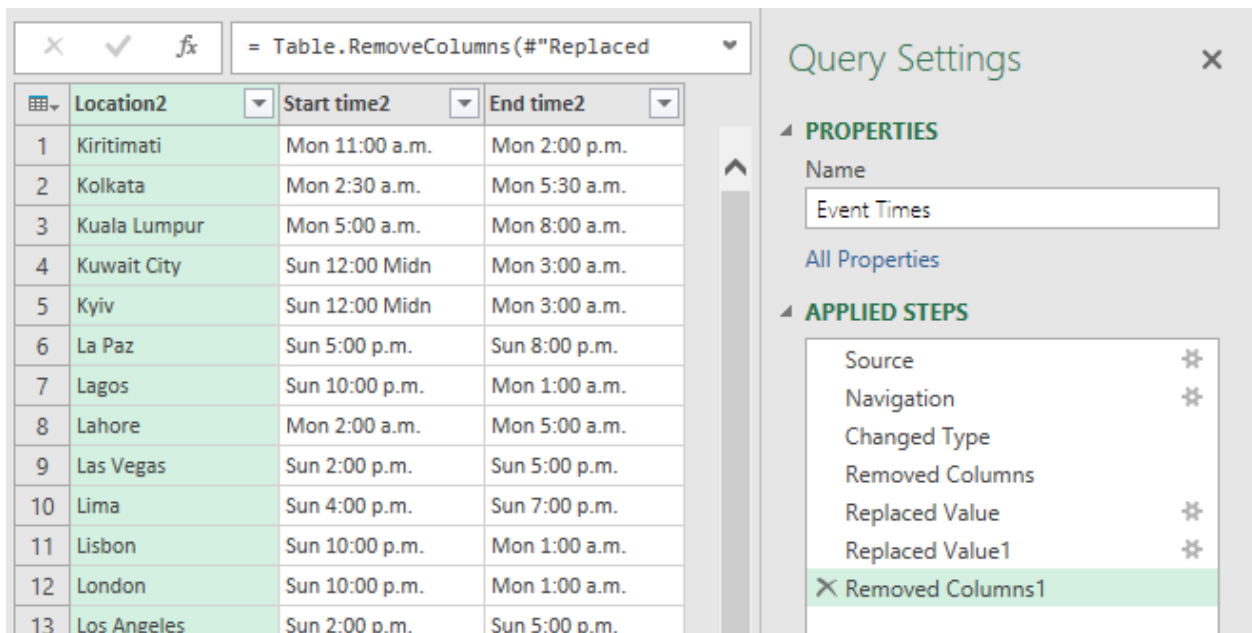
Location and Location2. The fact is that timeanddate.com website split the result set into two set of columns and produced HTML table like that. So for reading the full data set, I have to separate these two set of columns and then combine their values into a single set of columns.

So I'll produce a result set for the second set of columns with removing the first set;



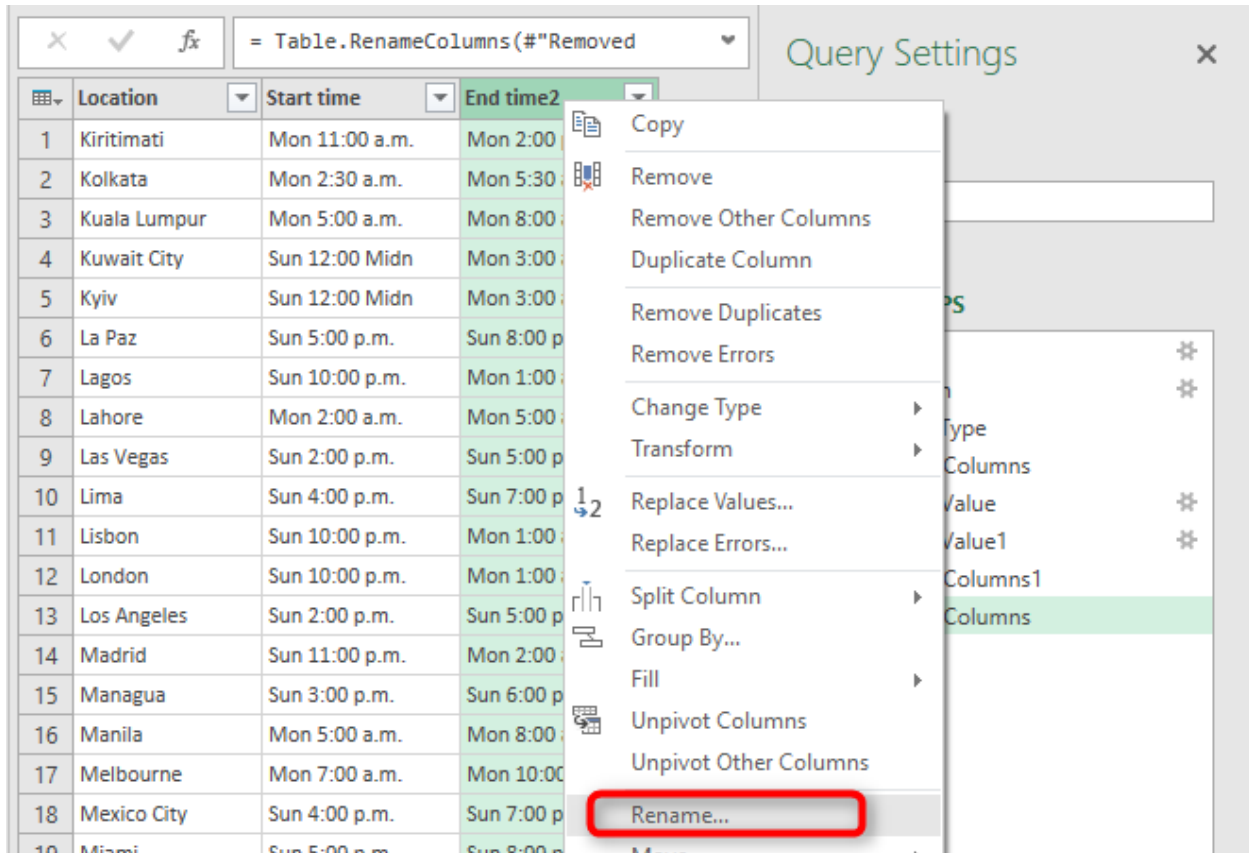
	Location	Start time	End time	Location2	Start time2	End time2
1	Accra	Sun 9:00 p.m.	Sun 12:00 M			
2	Addis Ababa	Sun 12:00 Midn	Mon 3:00 a.m.			
3	Adelaide	Mon 6:30 a.m.	Mon 9:30 a.m.			
4	Algiers	Sun 10:00 p.m.	Mon 1:00 a.m.			
5	Almaty	Mon 3:00 a.m.	Mon 6:00 a.m.			
6	Amman	Sun 12:00 Midn	Mon 3:00 a.m.			
7	Amsterdam	Sun 11:00 p.m.	Mon 2:00 a.m.			
8	Anadyr	Mon 9:00 a.m.	Mon 12:00 N			

This will give me the second set of columns as below;



	Location2	Start time2	End time2
1	Kiritimati	Mon 11:00 a.m.	Mon 2:00 p.m.
2	Kolkata	Mon 2:30 a.m.	Mon 5:30 a.m.
3	Kuala Lumpur	Mon 5:00 a.m.	Mon 8:00 a.m.
4	Kuwait City	Sun 12:00 Midn	Mon 3:00 a.m.
5	Kyiv	Sun 12:00 Midn	Mon 3:00 a.m.
6	La Paz	Sun 5:00 p.m.	Sun 8:00 p.m.
7	Lagos	Sun 10:00 p.m.	Mon 1:00 a.m.
8	Lahore	Mon 2:00 a.m.	Mon 5:00 a.m.
9	Las Vegas	Sun 2:00 p.m.	Sun 5:00 p.m.
10	Lima	Sun 4:00 p.m.	Sun 7:00 p.m.
11	Lisbon	Sun 10:00 p.m.	Mon 1:00 a.m.
12	London	Sun 10:00 p.m.	Mon 1:00 a.m.
13	Los Angeles	Sun 2:00 p.m.	Sun 5:00 p.m.

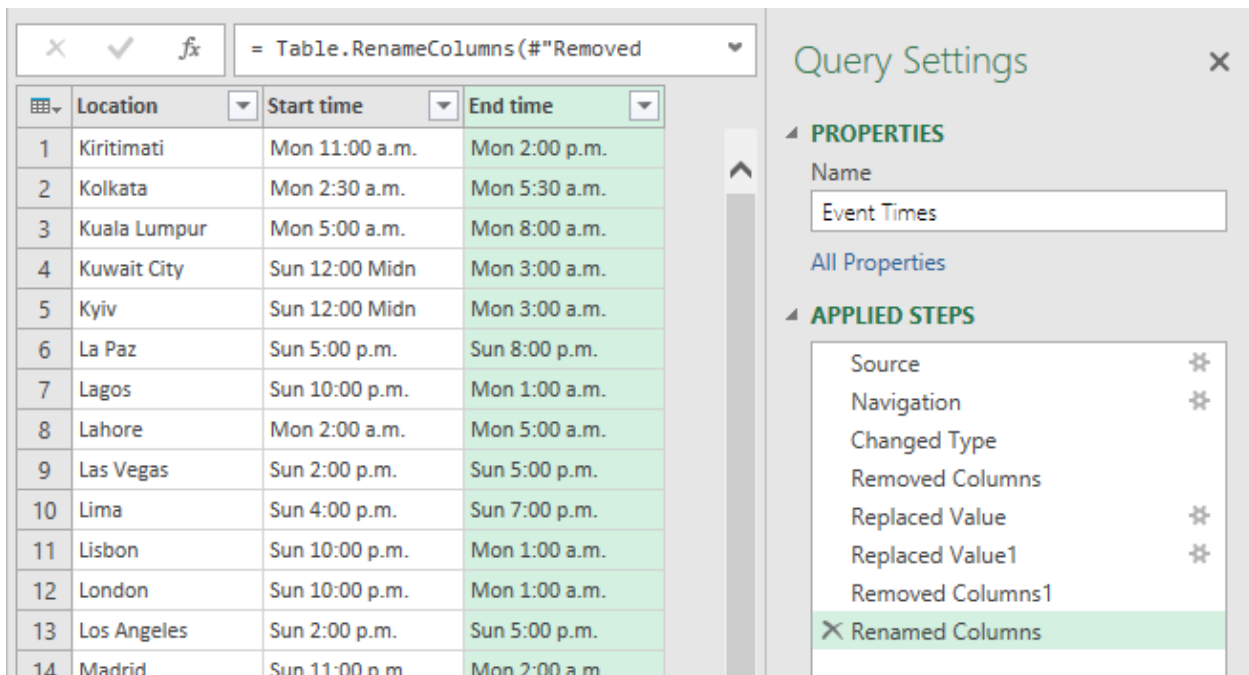
Then I rename the column names for this data set. The reason for this rename is that later on when I want to combine two queries together column names should be the same. I'll rename Columns to be Location, Start Time, End Time. for rename I just right click on each column and Rename



The screenshot shows a Power BI data table with columns: Location, Start time, and End time2. A context menu is open over the 'End time2' column, listing various transformation options. The 'Rename...' option is highlighted with a red rectangle.

	Location	Start time	End time2
1	Kiritimati	Mon 11:00 a.m.	Mon 2:00
2	Kolkata	Mon 2:30 a.m.	Mon 5:30
3	Kuala Lumpur	Mon 5:00 a.m.	Mon 8:00
4	Kuwait City	Sun 12:00 Midn	Mon 3:00
5	Kyiv	Sun 12:00 Midn	Mon 3:00
6	La Paz	Sun 5:00 p.m.	Sun 8:00 p
7	Lagos	Sun 10:00 p.m.	Mon 1:00
8	Lahore	Mon 2:00 a.m.	Mon 5:00
9	Las Vegas	Sun 2:00 p.m.	Sun 5:00 p
10	Lima	Sun 4:00 p.m.	Sun 7:00 p
11	Lisbon	Sun 10:00 p.m.	Mon 1:00
12	London	Sun 10:00 p.m.	Mon 1:00
13	Los Angeles	Sun 2:00 p.m.	Sun 5:00 p
14	Madrid	Sun 11:00 p.m.	Mon 2:00
15	Managua	Sun 3:00 p.m.	Sun 6:00 p
16	Manila	Mon 5:00 a.m.	Mon 8:00
17	Melbourne	Mon 7:00 a.m.	Mon 10:00
18	Mexico City	Sun 4:00 p.m.	Sun 7:00 p
19	Miami	Sun 5:00 p.m.	Sun 8:00 p

And here is my result set for the second data set:



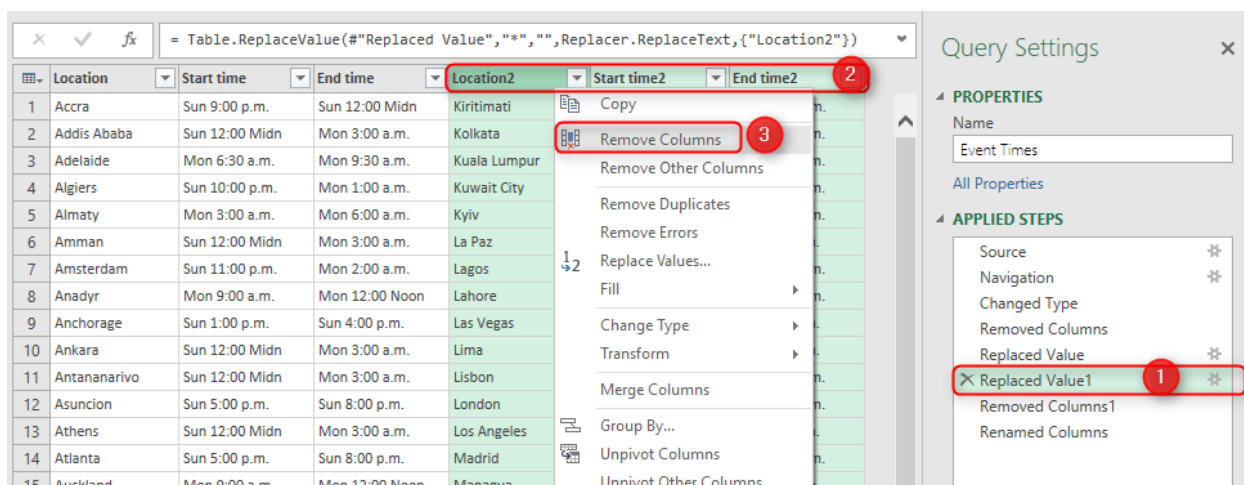
The screenshot shows the final result set after renaming the column. The 'End time' column is now correctly labeled. The 'Query Settings' pane on the right shows the 'Renamed Columns' step in the 'APPLIED STEPS' list.

	Location	Start time	End time
1	Kiritimati	Mon 11:00 a.m.	Mon 2:00 p.m.
2	Kolkata	Mon 2:30 a.m.	Mon 5:30 a.m.
3	Kuala Lumpur	Mon 5:00 a.m.	Mon 8:00 a.m.
4	Kuwait City	Sun 12:00 Midn	Mon 3:00 a.m.
5	Kyiv	Sun 12:00 Midn	Mon 3:00 a.m.
6	La Paz	Sun 5:00 p.m.	Sun 8:00 p.m.
7	Lagos	Sun 10:00 p.m.	Mon 1:00 a.m.
8	Lahore	Mon 2:00 a.m.	Mon 5:00 a.m.
9	Las Vegas	Sun 2:00 p.m.	Sun 5:00 p.m.
10	Lima	Sun 4:00 p.m.	Sun 7:00 p.m.
11	Lisbon	Sun 10:00 p.m.	Mon 1:00 a.m.
12	London	Sun 10:00 p.m.	Mon 1:00 a.m.
13	Los Angeles	Sun 2:00 p.m.	Sun 5:00 p.m.
14	Madrid	Sun 11:00 p.m.	Mon 2:00 a.m.

Insert a Step

Now I want to produce another variable for the first set of columns. Instead of creating the whole process again from Get data from web, I start from one step ago which was before removing columns. As I've mentioned earlier I can grab that status easily from the Applied Steps section in Query Settings pane. There is a step named Replaced Value1. This step belongs to where I replaced * characters with blank for Location2 columns.

I Click on Replaced Value1 step and then choose second set of columns to remove.



When I click on Remove columns as above, I see a message says; Are you sure you want to insert a step? I get this message because this is not the latest step in my query editor. As long as I know what I am doing adding an extra step is fine.



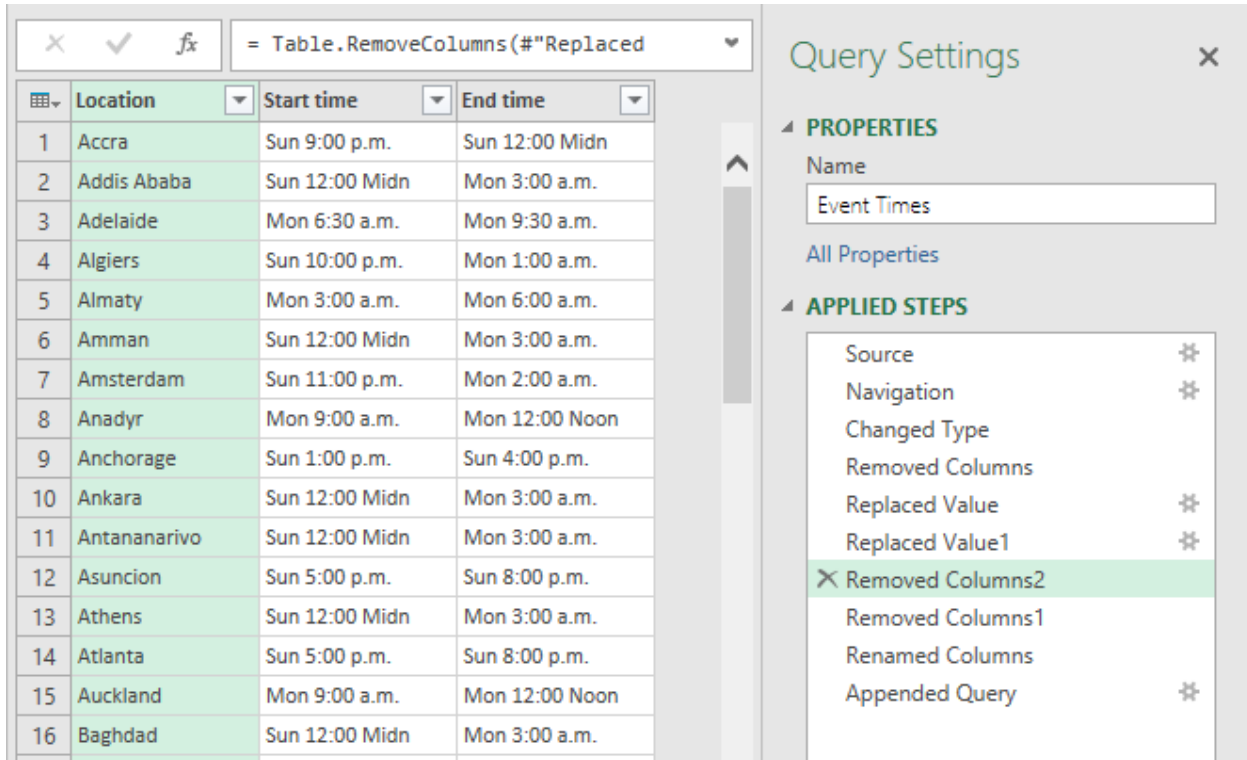
Insert Step

Are you sure you want to insert a step? Inserting an intermediate step may affect subsequent steps, which could cause your query to break.

Insert

Cancel

I hit Insert for the message above and a new step will be created called Removed Columns2. With the result set as below;



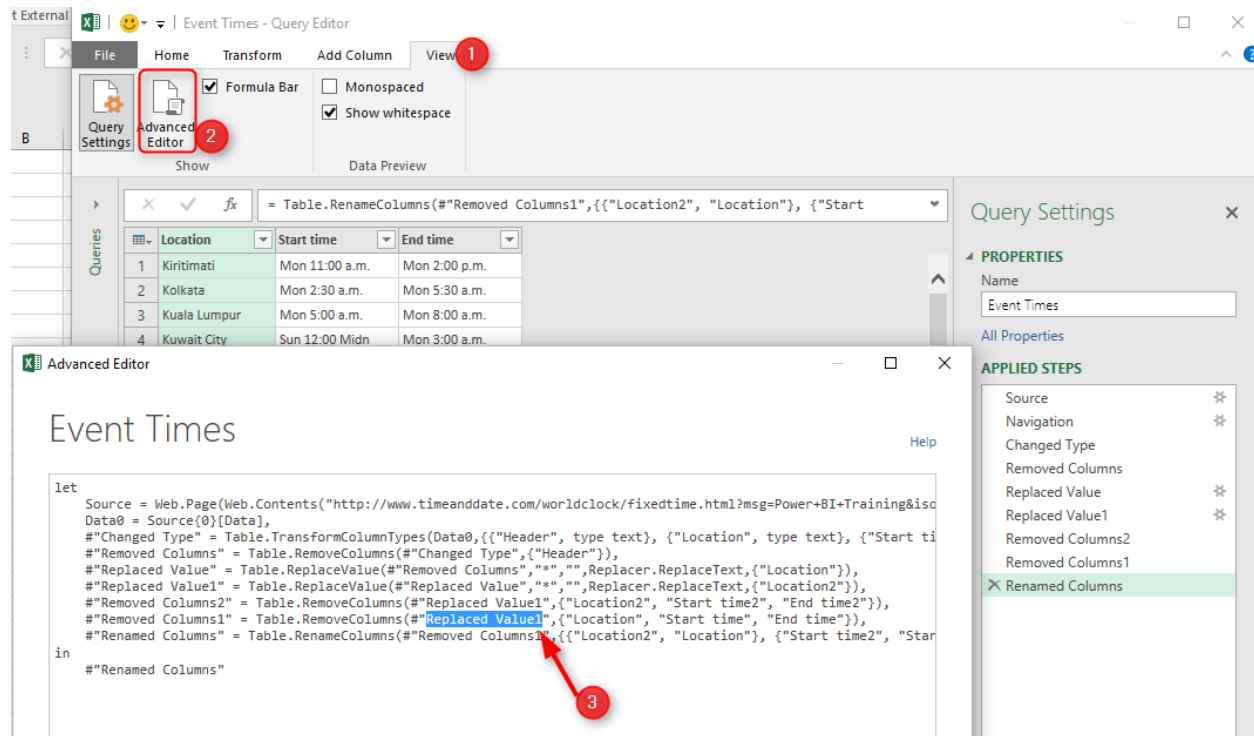
The screenshot shows the Power Query Editor interface. On the left, a table with 4 columns: Location, Start time, and End time. The table contains 16 rows of data. On the right, the 'Query Settings' pane is open, showing the 'APPLIED STEPS' list. The steps are: Source, Navigation, Changed Type, Removed Columns, Replaced Value, Replaced Value1, **Removed Columns2** (highlighted), Removed Columns1, Renamed Columns, and Appended Query.

	Location	Start time	End time
1	Accra	Sun 9:00 p.m.	Sun 12:00 Midn
2	Addis Ababa	Sun 12:00 Midn	Mon 3:00 a.m.
3	Adelaide	Mon 6:30 a.m.	Mon 9:30 a.m.
4	Algiers	Sun 10:00 p.m.	Mon 1:00 a.m.
5	Almaty	Mon 3:00 a.m.	Mon 6:00 a.m.
6	Amman	Sun 12:00 Midn	Mon 3:00 a.m.
7	Amsterdam	Sun 11:00 p.m.	Mon 2:00 a.m.
8	Anadyr	Mon 9:00 a.m.	Mon 12:00 Noon
9	Anchorage	Sun 1:00 p.m.	Sun 4:00 p.m.
10	Ankara	Sun 12:00 Midn	Mon 3:00 a.m.
11	Antananarivo	Sun 12:00 Midn	Mon 3:00 a.m.
12	Asuncion	Sun 5:00 p.m.	Sun 8:00 p.m.
13	Athens	Sun 12:00 Midn	Mon 3:00 a.m.
14	Atlanta	Sun 5:00 p.m.	Sun 8:00 p.m.
15	Auckland	Mon 9:00 a.m.	Mon 12:00 Noon
16	Baghdad	Sun 12:00 Midn	Mon 3:00 a.m.

Fixing Insert Step reference issue

When I insert a step in Power Query, the step after that will source from inserted step. This will cause some issues in my scenario. I inserted an step and removed Location2, Start time2, and End time2. and then in Removed Columns1 I want to remove other columns and in Renamed Columns rename columns that does not exists anymore. So I have to fix the reference issue happened by inserting the step. Here is how I fix it:

I go to View tab, Advanced Editor (or I can find Advanced Editor button in Home tab). This will open Advanced Editor with M code for me. Then I replace the highlighted part to be referenced from Replaced Value1.



Now my data sets exists as below:

Removed Columns2: the data set for first set of columns

Renamed Columns: the data set for second set of columns

Append Queries

Now It's the time to combine these two data sets. There is a menu option that combines two data sets. However, it only appends two QUERIES. As I have everything here in one query (but different steps or let's say variables), so I can't use that option. Fortunately, I can do that with a single line of M scripting.

I click on Advanced Editor to see the M code generated behind the scene. Then I Insert code line below at the end of LET clause, after adding a single comma (which is line separator in Power Query) in the line before it.

```
#"Appended Query" = Table.Combine({#"Removed Columns2", #"Renamed Columns" })
```

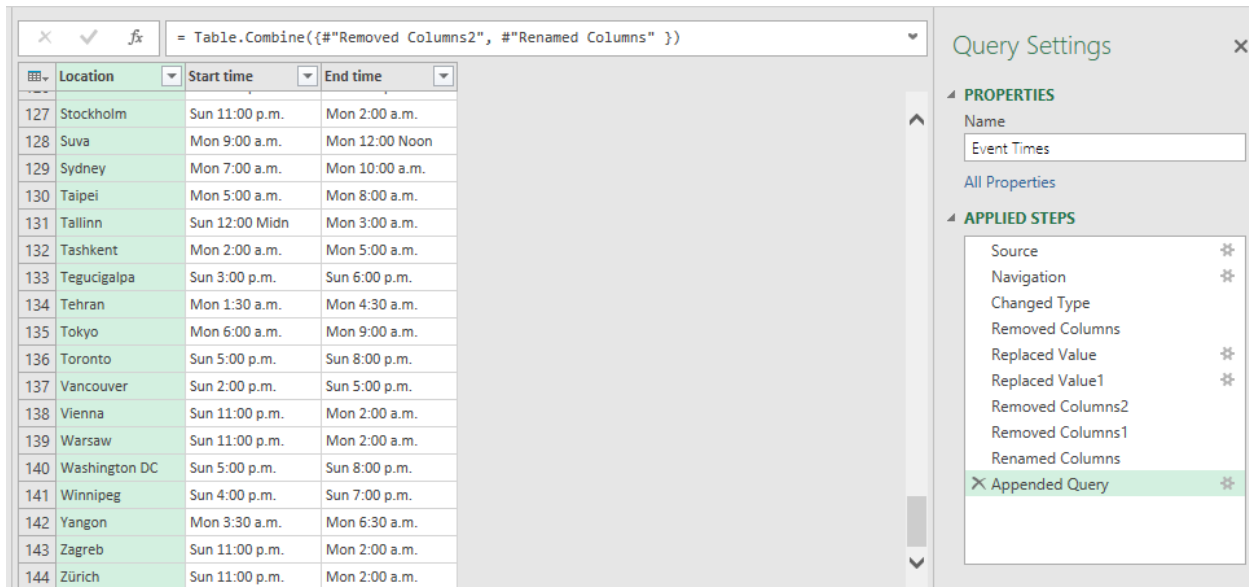
Statement above creates a new data set as a variable called Appended Query. I've used Table.Combine to append queries (or UNION them in SQL and Database terminologies). This function is a simple function and gets data sets to be combined which are Removed Columns1 and Renamed Columns1.

Then I show this step in the result set with writing that in IN section:

in

#"Appended Query"

Now my result set have both data sets combined into one. (I can check this includes 144 columns for 144 cities; combined list of 72 cities in each of previous data sets).



The screenshot shows the Power BI Query Editor interface. The main area displays a table with 4 columns: Location, Start time, and End time. The table contains 14 rows of data, numbered 127 to 144. The formula bar at the top shows the query: `= Table.Combine({#"Removed Columns2", #"Renamed Columns" })`. On the right, the 'Query Settings' pane is open, showing the 'APPLIED STEPS' list. The steps include: Source, Navigation, Changed Type, Removed Columns, Replaced Value, Replaced Value1, Removed Columns2, Removed Columns1, Renamed Columns, and 'Appended Query' (which is highlighted in green).

	Location	Start time	End time
127	Stockholm	Sun 11:00 p.m.	Mon 2:00 a.m.
128	Suva	Mon 9:00 a.m.	Mon 12:00 Noon
129	Sydney	Mon 7:00 a.m.	Mon 10:00 a.m.
130	Taipei	Mon 5:00 a.m.	Mon 8:00 a.m.
131	Tallinn	Sun 12:00 Midn	Mon 3:00 a.m.
132	Tashkent	Mon 2:00 a.m.	Mon 5:00 a.m.
133	Tegucigalpa	Sun 3:00 p.m.	Sun 6:00 p.m.
134	Tehran	Mon 1:30 a.m.	Mon 4:30 a.m.
135	Tokyo	Mon 6:00 a.m.	Mon 9:00 a.m.
136	Toronto	Sun 5:00 p.m.	Sun 8:00 p.m.
137	Vancouver	Sun 2:00 p.m.	Sun 5:00 p.m.
138	Vienna	Sun 11:00 p.m.	Mon 2:00 a.m.
139	Warsaw	Sun 11:00 p.m.	Mon 2:00 a.m.
140	Washington DC	Sun 5:00 p.m.	Sun 8:00 p.m.
141	Winnipeg	Sun 4:00 p.m.	Sun 7:00 p.m.
142	Yangon	Mon 3:30 a.m.	Mon 6:30 a.m.
143	Zagreb	Sun 11:00 p.m.	Mon 2:00 a.m.
144	Zürich	Sun 11:00 p.m.	Mon 2:00 a.m.

Next Steps

It will be really long if I mention all steps in a single post, So stay tuned. Next step would be Grouping rows and applying some transformations to get the desired output. Finally I will show you how to use parameters in the query and automate the process.

Power Query Not for BI: Event Date and Time Scheduler - Part 2

Published Date: May 14, 2016

	Start time	End time	Cities
1	Mon 1:00 AM	Mon 4:00 AM	Accra, Reykjavik
2	Mon 4:00 AM	Mon 7:00 AM	Addis Ababa, Amman, Ankara, Antananarivo, Athens, Baghdad, Beirut, Buchare...
3	Mon 10:30 AM	Mon 1:30 PM	Adelaide, Darwin
4	Mon 2:00 AM	Mon 5:00 AM	Algiers, Casablanca, Dublin, Kinshasa, Lagos, Lisbon, London
5	Mon 7:00 AM	Mon 10:00 AM	Almaty, Dhaka
6	Mon 3:00 AM	Mon 6:00 AM	Amsterdam, Barcelona, Belgrade, Berlin, Brussels, Budapest, Cairo, Cape Town,...
7	Mon 1:00 PM	Mon 4:00 PM	Anadyr, Auckland, Suva
8	Sun 5:00 PM	Sun 8:00 PM	Anchorage
9	Sun 9:00 PM	Sun 12:00 Midn	Asuncion, Atlanta, Boston, Caracas, Columbus, Detroit, Havana, Indianapolis, La...

In [first part of this series](#) I've explained that Power Query can be used for non-BI scenarios, and here is a real-world use case example; Creating an event date and time scheduler which consider multiple time zones. In [previous post](#) you've learned how to use Power Query to do some data mash up to build a unified list of cities in different time zones and start and end time of event in each city. In this post we will continue steps to build the final result set. If you want to learn more about Power Query read my Power BI online book; [Power BI from Rookie to Rock Star](#).

Group Rows

After creating the full list, I want to group rows by their start and end time. I want to create groups of cities where start time of the event is the same (and definitely end time would be the same).

I start by clicking on Start time and End time columns and then I choose Group By from the Home menu under Transform. In Group By window I can see that Start time and End time already selected as Group By columns (I can add or remove from this window by clicking on – or + buttons if I want to).

I create two new columns for group by result set, one as a count of all rows in each group (this is just out of curiosity to see how many cities I have in each start/end time

group), and the other one as all rows for each group (this will generate a table for each group, containing all cities of that group; all cities with same start/end time);

×

Group By

Specify the columns to group by.

Group by

+

Start time

-

End time

-

Group By Columns

New Output Columns

New column name	Operation	Column	+
Count	Count Rows		-
Cities	All Rows		-

OK Cancel

Output of this step will be a table with unique start/end times plus two new columns; count of all rows, and a table embedded column for list of all cities.

	Start time	End time	Count	Cities
1	Mon 1:00 AM	Mon 4:00 AM	2	Table
2	Mon 4:00 AM	Mon 7:00 AM	22	Table
3	Mon 10:30 AM	Mon 1:30 PM	2	Table
4	Mon 2:00 AM	Mon 5:00 AM	7	Table
5	Mon 7:00 AM	Mon 10:00 AM	2	Table

I can click on a cell in the Cities column to see contents of it (Do not click on the Table word, click on an empty space in that cell. If you click on the Table word a new subsequent step will be generated to fetch that specific table).

Query Settings

PROPERTIES

Name

Event Times

All Properties

APPLIED STEPS

changedtype

date

time

duration

Source1

Data0

Changed Type

Removed Columns

Replaced Value

Replaced Value1

Removed Columns1

Removed Columns2

Renamed Columns1

Appended Query

Grouped Rows

Query Formula: = Table.Group(#"Appended Query", {"Start

	Start time	End time	Count	Cities
1	Mon 1:00 AM	Mon 4:00 AM	2	Table
2	Mon 4:00 AM	Mon 7:00 AM	22	Table
3	Mon 10:30 AM	Mon 1:30 PM	2	Table
4	Mon 2:00 AM	Mon 5:00 AM	7	Table
5	Mon 7:00 AM	Mon 10:00 AM	2	Table
6	Mon 3:00 AM	Mon 6:00 AM	22	Table
7	Mon 1:00 PM	Mon 4:00 PM	3	Table
8	Sun 5:00 PM	Sun 8:00 PM	1	Table
9	Sun 9:00 PM	Sun 12:00 Midn	19	Table
10	Mon 8:00 AM	Mon 11:00 AM	5	Table
11	Mon 9:00 AM	Mon 12:00 Noon	8	Table

Location	Start time	End time
Addis Ababa	Mon 4:00 AM	Mon 7:00 AM
Amman	Mon 4:00 AM	Mon 7:00 AM
Ankara	Mon 4:00 AM	Mon 7:00 AM
Antananarivo	Mon 4:00 AM	Mon 7:00 AM
Athens	Mon 4:00 AM	Mon 7:00 AM
Baghdad	Mon 4:00 AM	Mon 7:00 AM
Beirut	Mon 4:00 AM	Mon 7:00 AM
Bucharest	Mon 4:00 AM	Mon 7:00 AM

Concatenate Rows

Content of each cell in above screenshot is a table containing all cities in that group. For example, for the selected cell above I have a table of 22 cities.

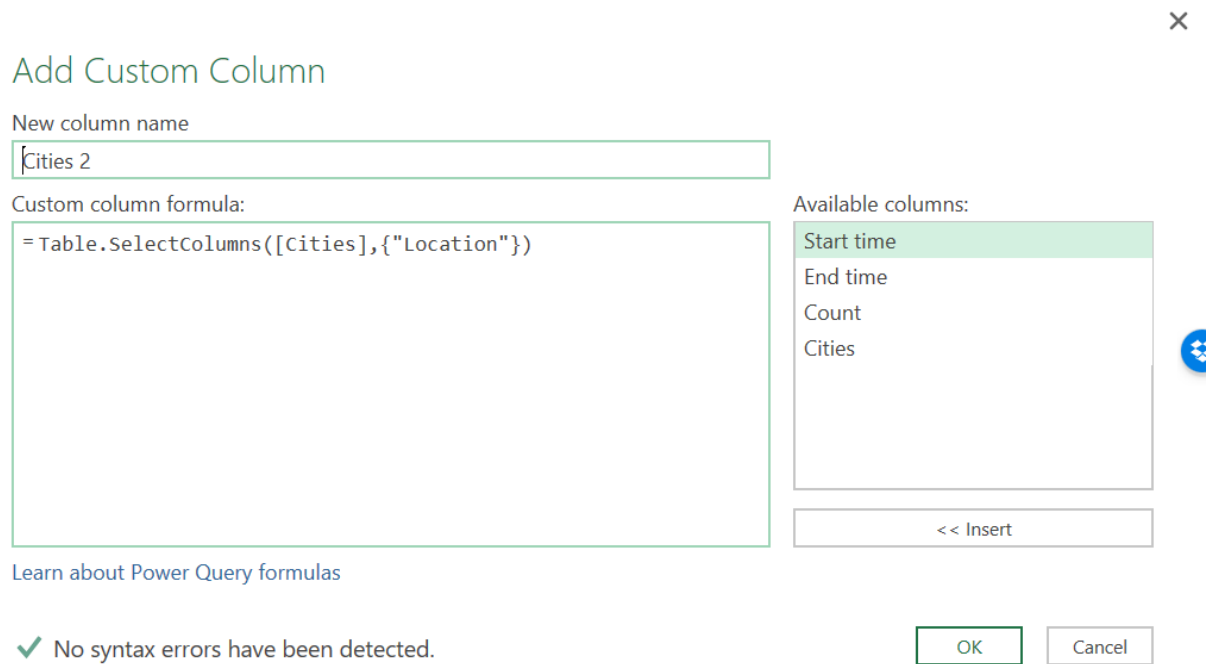
So now my next step is to create a string concatenated list of all these city names, like: Addis Ababa, Amman, Ankara, Antananarivo, Athens...

I can transform this column to be concatenated list all in one, but for simplicity and ease of understanding I'll do that step by step by adding a custom column in each step;

Step 1 to Concatenate: Select Single Column

As the first step to concatenate, I'll just fetch single column from the sub-table which is Location .

I create a new custom column by click on the Add Column menu, and then Add Custom Column icon. This will open Add Custom Column window for me where I can define a new column with the name and the expression to calculate the new column.



×

Add Custom Column

New column name
Cities 2

Custom column formula:
= Table.SelectColumns([Cities],{"Location"})

Available columns:
Start time
End time
Count
Cities

<< Insert

Learn about Power Query formulas

✓ No syntax errors have been detected.

OK Cancel

I name this as Cities 2, and the expression as

```
= Table.SelectColumns([Cities],{"Location"})
```

Table.SelectColumns only fetches specific columns of the table that we mention in the {}. Here I only fetched Location column, so it will be a single column table as a result;

fx = Table.AddColumn(#"Grouped Rows", "Cities 2", each Table

	Start time	End time	Count	Cities	Cities 2
1	Mon 1:00 AM	Mon 4:00 AM	2	Table	Table
2	Mon 4:00 AM	Mon 7:00 AM	22	Table	Table
3	Mon 10:30 AM	Mon 1:30 PM	2	Table	Table
4	Mon 2:00 AM	Mon 5:00 AM	7	Table	Table
5	Mon 7:00 AM	Mon 10:00 AM	2	Table	Table
6	Mon 3:00 AM	Mon 6:00 AM	22	Table	Table
7	Mon 1:00 PM	Mon 4:00 PM	3	Table	Table
8	Sun 5:00 PM	Sun 8:00 PM	1	Table	Table
9	Sun 9:00 PM	Sun 12:00 Midn	19	Table	Table
10	Mon 8:00 AM	Mon 11:00 AM	3	Table	Table
11	Mon 9:00 AM	Mon 12:00 Noon	8	Table	Table

Location
Addis Ababa
Amman
Ankara
Antananarivo
Athens
Baghdad
Beirut
Bucharest

Step 2 to Concatenate: Rows to Columns

For concatenation I want to use a method that needs values to be on COLUMNS not on rows. So I have to transform the inner table to show values in Columns rather than rows. Fortunately it is easy to do with Table.Transpose. I add a new custom column with this expression:

```
= Table.Transpose([Cities 2])
```

This simple expression will turn my single-column row values into multiple columns in a single row. For using Transpose All you need is table name. Table should be single

columned. And then it transforms it to rows. Column headers would be auto named as Column1, Column2,.... So this would be the result;

fx = Table.AddColumn("#Added Custom", "Cities 4", each Table.Transpose([Cities

	Start time	End time	Count	Cities	Cities 2	Cities 4
1	Mon 1:00 AM	Mon 4:00 AM	2	Table	Table	Table
2	Mon 4:00 AM	Mon 7:00 AM	22	Table	Table	Table
3	Mon 10:30 AM	Mon 1:30 PM	2	Table	Table	Table
4	Mon 2:00 AM	Mon 5:00 AM	7	Table	Table	Table
5	Mon 7:00 AM	Mon 10:00 AM	2	Table	Table	Table
6	Mon 3:00 AM	Mon 6:00 AM	22	Table	Table	Table
7	Mon 1:00 PM	Mon 4:00 PM	3	Table	Table	Table
8	Sun 5:00 PM	Sun 8:00 PM	1	Table	Table	Table
9	Sun 9:00 PM	Sun 12:00 Midn	19	Table	Table	Table
10	Mon 8:00 AM	Mon 11:00 AM	3	Table	Table	Table
11	Mon 9:00 AM	Mon 12:00 Noon	8	Table	Table	Table

Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8
Addis Ababa	Amman	Ankara	Antananarivo	Athens	Baghdad	Beirut	Bucharest

A bit of code behind

You might wonder what is happening when you add a custom column which is based on an expression which uses an existing column. With a glance at Advanced Editor you can see this is the script line generated for adding this custom column;

```
#"Added Custom2" = Table.AddColumn("#Added Custom", "Cities 4", each Table.Transpose([Cities 2]))
```

Table.AddColumn adds a new column with name and expression. This function gets three parameters;

- Name of the table to add column to it: #"Added Custom"
- Name of the new column to generate: Cities 4
- Expression to populate the new column: each Table.Transpose([Cities 2])

EACH is a singleton function that applies on every row of the main table, and produces the result of expression for that row into the new column.

Step 3 to Concatenate: Concatenate to List

Now that I have values in multiple columns I can concatenate them all into one string with Table.ToList function which converts a table to List. This function can concatenate all columns of table into one column (because List is a single columned data structure).

The actual concatenation happens by Combiner function;

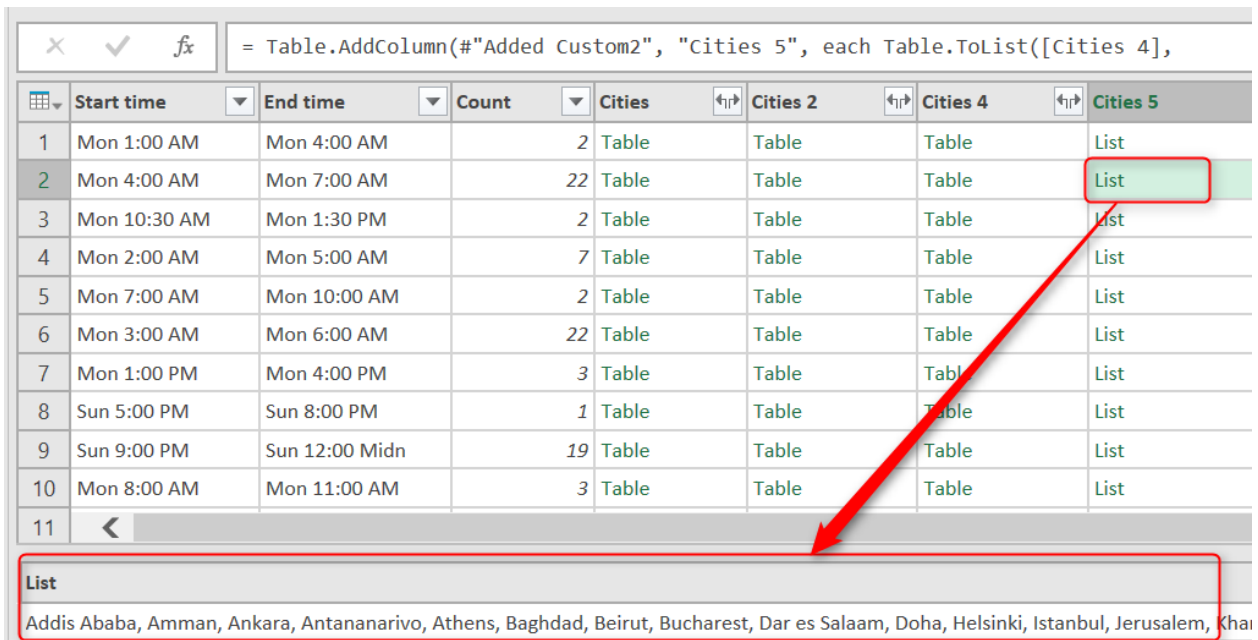
Combiner.CombineTextByDelimiter(",", ") which concatenate values with a delimiter which I set to be comma. So here is the expression for my new custom column:

```
=Table.ToList([Cities 4],
```

```
Combiner.CombineTextByDelimiter(",", ")
```

```
)
```

And the result would be a list which contains concatenated string value;



The screenshot shows a Power BI Desktop interface with a table. The table has columns: Start time, End time, Count, Cities, Cities 2, Cities 4, and Cities 5. The Cities 5 column contains the value 'List'. A red box highlights the 'List' value in the Cities 5 column, and a red arrow points from it to a detailed view of the list below the table.

	Start time	End time	Count	Cities	Cities 2	Cities 4	Cities 5
1	Mon 1:00 AM	Mon 4:00 AM	2	Table	Table	Table	List
2	Mon 4:00 AM	Mon 7:00 AM	22	Table	Table	Table	List
3	Mon 10:30 AM	Mon 1:30 PM	2	Table	Table	Table	List
4	Mon 2:00 AM	Mon 5:00 AM	7	Table	Table	Table	List
5	Mon 7:00 AM	Mon 10:00 AM	2	Table	Table	Table	List
6	Mon 3:00 AM	Mon 6:00 AM	22	Table	Table	Table	List
7	Mon 1:00 PM	Mon 4:00 PM	3	Table	Table	Table	List
8	Sun 5:00 PM	Sun 8:00 PM	1	Table	Table	Table	List
9	Sun 9:00 PM	Sun 12:00 Midn	19	Table	Table	Table	List
10	Mon 8:00 AM	Mon 11:00 AM	3	Table	Table	Table	List
11							

Below the table, a red box highlights the 'List' value, and a red arrow points from it to a detailed view of the list below the table.

List

Addis Ababa, Amman, Ankara, Antananarivo, Athens, Baghdad, Beirut, Bucharest, Dar es Salaam, Doha, Helsinki, Istanbul, Jerusalem, Khai

Step 4 to Concatenate: Expand

All I have to now is to expand the list to its values. Expanding a column means expanding its underneath structure. Expand column appears when you have a multi value structure in each cell. Multi-value structures in Power Query can be; Table, List, and Record. You can find Expand Column Icon simply besides the column heading and click on it.

fx = Table.AddColumn("#Added Custom2", "Cities 5", each Table.ToList([Cities 4],

	nd time	Count	Cities	Cities 2	Cities 4	Cities 5
1	Mon 4:00 AM	2	Table	Table	Table	List
2	Mon 7:00 AM	22	Table	Table	Table	List
3	Mon 1:30 PM	2	Table	Table	Table	List
4	Mon 5:00 AM	7	Table	Table	Table	List
5	Mon 10:00 AM	2	Table	Table	Table	List
6	Mon 6:00 AM	22	Table	Table	Table	List

After expanding I can see concatenated text values in my main table as below;

fx = Table.ExpandListColumn("#Added Custom3", "Cities 5")

	Start time	End time	Count	Cities	Cities 2	Cities 4	Cities 5
1	Mon 1:00 AM	Mon 4:00 AM	2	Table	Table	Table	Accra, Reykjavik
2	Mon 4:00 AM	Mon 7:00 AM	22	Table	Table	Table	Addis Ababa, Amman, Ankara, Antananarivo, Athens, Baghdad, Beirut,...
3	Mon 10:30 AM	Mon 1:30 PM	2	Table	Table	Table	Adelaide, Darwin
4	Mon 2:00 AM	Mon 5:00 AM	7	Table	Table	Table	Algiers, Casablanca, Dublin, Kinshasa, Lagos, Lisbon, London

Final Polish

Now I have Start/End time for each time zone, and concatenated list of all cities in that time zone, so I just do a bit of polishing with removing extra columns. Only keeping Start Time, End Time, and Cities 5. And renaming Cities 5 to Cities.

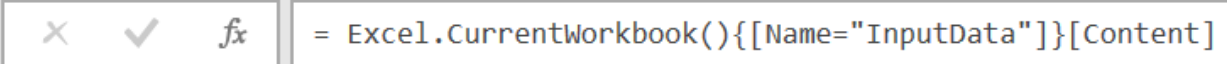

	Start time	End time	Cities
1	Mon 1:00 AM	Mon 4:00 AM	Accra, Reykjavik
2	Mon 4:00 AM	Mon 7:00 AM	Addis Ababa, Amman, Ankara, Antananarivo, Athens, Baghdad, Beirut, Buchare...
3	Mon 10:30 AM	Mon 1:30 PM	Adelaide, Darwin
4	Mon 2:00 AM	Mon 5:00 AM	Algiers, Casablanca, Dublin, Kinshasa, Lagos, Lisbon, London
5	Mon 7:00 AM	Mon 10:00 AM	Almaty, Dhaka
6	Mon 3:00 AM	Mon 6:00 AM	Amsterdam, Barcelona, Belgrade, Berlin, Brussels, Budapest, Cairo, Cape Town,...
7	Mon 1:00 PM	Mon 4:00 PM	Anadyr, Auckland, Suva
8	Sun 5:00 PM	Sun 8:00 PM	Anchorage
9	Sun 9:00 PM	Sun 12:00 Midn	Asuncion, Atlanta, Boston, Caracas, Columbus, Detroit, Havana, Indianapolis, La...

Next Steps

I've made my result set as I want with Power Query so far, but still one step left, which is using parameters and Automation; I'll be using an Excel table as input parameters to set my local event date and time, and then Power Query will use that input parameters to refresh the whole result set. Stay tuned for the next post to see how it can be done.

Power Query Not for BI: Event Date and Time Scheduler - Part 3

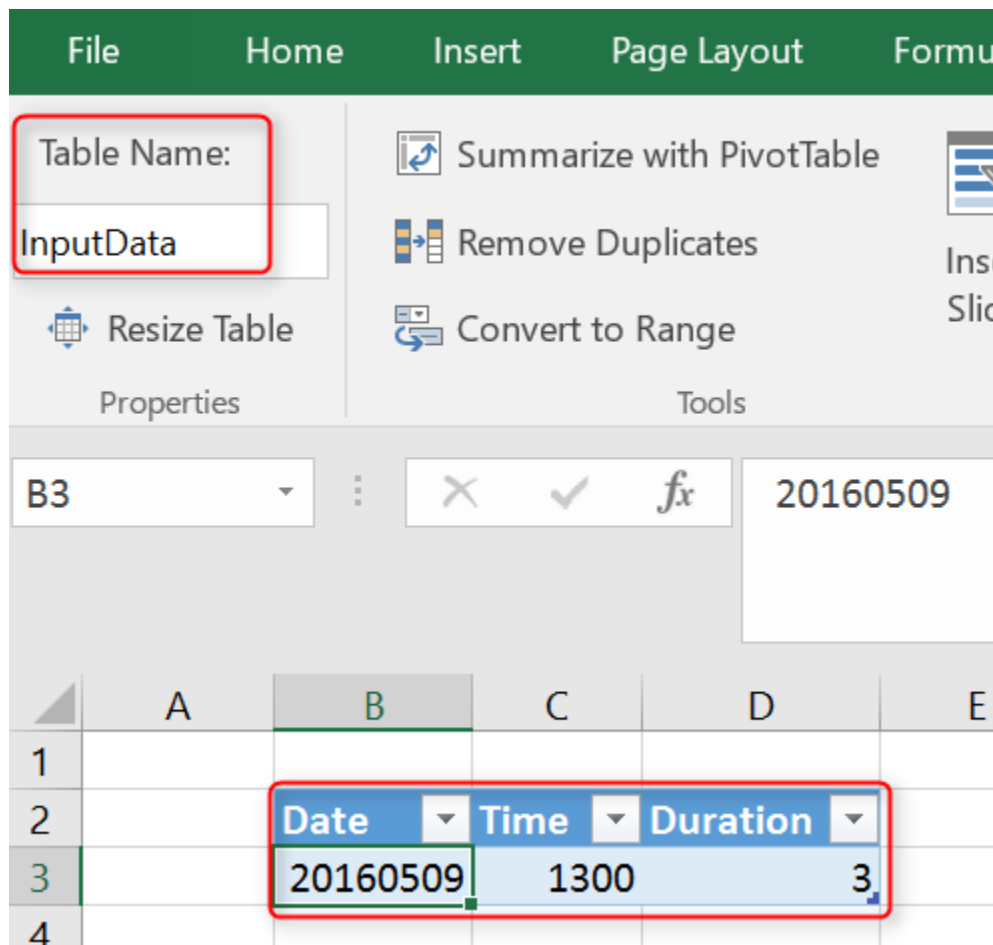
Published Date: May 16, 2016

			
	Date	Time	Duration
1	20160509	1300	3

Previously You have seen in [Part 1](#) and [Part 2](#) of this series, that how you can use Power Query to do data transformation and turn the result of event date/time scheduler of timeanddate.com website to a better format. We've used number of basic transformations, and some functions such as Table.Transpose to fetch desired output. This part is the last step which we will use parameters to make our query dynamic. At the end of this part you will learn how to automate your Power Query solution, with a change in parameters in an Excel table, and hit refresh button a new result set with desired format will be created. To learn more about Power Query, read my Power BI online book; [Power BI from Rookie to Rock Star](#).

Using Parameters

We've made all the data manipulation and mash up, and the output result set is exactly as desired. However, one step is left: Using Parameters and making this query dynamic or automate the process. For parameters I use an Excel table as my configuration table. So I create a table as below;



This table has three columns: Date, Time, and Duration. I separated the date and time for simplicity of this example. Date to be formatted as YYYYMMDD, and Time as HHMM, and duration as an integer value illustrating hours.

Configuration above means the event starts at 9th of May 2016, at 1:00 pm New Zealand time (this is what my local time is), with duration of 3 hours. I named this table as InputData.

I can fetch this table in my Query Editor with a simple M script line as below;

```
input = Excel.CurrentWorkbook(){[Name="InputData"]}[Content]
```

Excel.CurrentWorkbook reads a table from the current excel workbook. Name of the table passed as input parameter. And [Content] will load content field of the table (which is a table by itself of all columns and rows of the Excel table) into the variable named input.

<div> <div>×</div> <div>✓</div> <div>fx</div> </div> <div>= Excel.CurrentWorkbook(){[Name="InputData"]}[Content]</div>			
	Date	Time	Duration
1	20160509	1300	3

I can then read first record of this table into a variable called inputrecord. And use it for fetching each parameter later on. Here is my inputrecord

<div> <div>×</div> <div>✓</div> <div>fx</div> </div> <div>= Table.First(input)</div>	
Date	20160509
Time	1300
Duration	3

```
inputrecord=Table.First(input),
```

Now I can read Date value easily by expression below:

```
date=Text.From(inputrecord[Date]),
```

For Time I do also use a PadStart with 0 to generate values like 0400 when time is 4AM. The reason is that TimeandDate.com asks for a full HHMM string with leading 0 to work correctly. Duration calculation is simple as well.

```
time=Text.PadStart(Text.From(inputrecord[Time]),4,"0"),
```

```
duration=Text.From(inputrecord[Duration]),
```

Now that I have all three parameters in variables named; date, time, and duration. I can simply use them in generating web url to pass to timeanddate.com website as below;

```
WebURL="http://www.timeanddate.com/worldclock/fixedtime.html?msg=Power+BI+Training&iso="&date&"T"&time&"&p1=22&ah="&duration,
```

String concatenation character in Power Query is: &. and expression above generates the web url for the date/time of event with local time of Auckland, New Zealand with duration specified. All I have to do now is to fetch data from this URL with the WebURL variable. So I'll change line below;

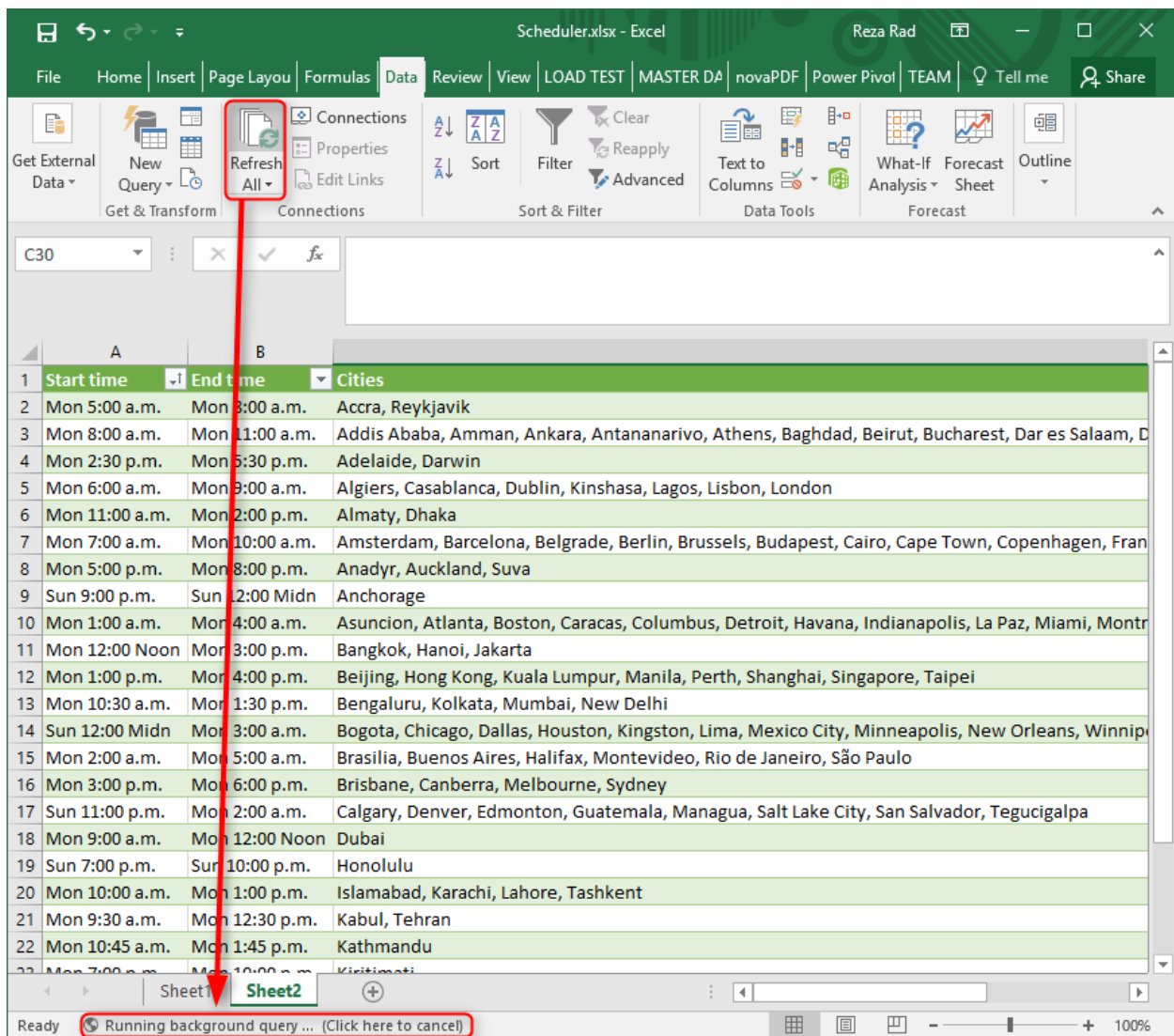
Source1 =

```
Web.Page(Web.Contents("http://www.timeanddate.com/worldclock/fixedtime.html?msg=Power+BI+Training&iso=20160509T1300&p1=22&ah=3"))
```

To this:

```
Source1 = Web.Page(Web.Contents(WebURL)),
```

Now I have a dynamic query based on parameters. Let's change the parameters in Excel and Refresh the data.



When I refresh the data I can see in the status bar that Power Query is refreshing the whole data set based on input parameters (illustrated in screenshot above).

And here is my final result:

	A	B	
1	Start time	End time	Cities
2	Mon 3:30 a.m.	Mon 6:30 a.m.	Accra, Reykjavik
3	Mon 6:30 a.m.	Mon 9:30 a.m.	Addis Ababa, Amman, Ankara, Antananarivo
4	Mon 1:00 p.m.	Mon 4:00 p.m.	Adelaide, Darwin
5	Mon 4:30 a.m.	Mon 7:30 a.m.	Algiers, Casablanca, Dublin, Kinshasa, Lagos
6	Mon 9:30 a.m.	Mon 12:30 p.m.	Almaty, Dhaka
7	Mon 5:30 a.m.	Mon 8:30 a.m.	Amsterdam, Barcelona, Belgrade, Berlin, Br
8	Mon 3:30 p.m.	Mon 6:30 p.m.	Anadyr, Auckland, Suva
9	Sun 7:30 p.m.	Sun 10:30 p.m.	Anchorage
10	Sun 11:30 p.m.	Mon 2:30 a.m.	Asuncion, Atlanta, Boston, Caracas, Columb
11	Mon 10:30 a.m.	Mon 1:30 p.m.	Bangkok, Hanoi, Jakarta
12	Mon 11:30 a.m.	Mon 2:30 p.m.	Beijing, Hong Kong, Kuala Lumpur, Manila, F
13	Mon 9:00 a.m.	Mon 12:00 Noon	Bengaluru, Kolkata, Mumbai, New Delhi
14	Sun 10:30 p.m.	Mon 1:30 a.m.	Bogota, Chicago, Dallas, Houston, Kingston,
15	Mon 12:30 a.m.	Mon 3:30 a.m.	Brasilia, Buenos Aires, Halifax, Montevideo,
16	Mon 1:30 p.m.	Mon 4:30 p.m.	Brisbane, Canberra, Melbourne, Sydney
17	Sun 9:30 p.m.	Mon 12:30 a.m.	Calgary, Denver, Edmonton, Guatemala, Ma
18	Mon 7:30 a.m.	Mon 10:30 a.m.	Dubai
19	Sun 5:30 p.m.	Sun 8:30 p.m.	Honolulu
20	Mon 8:30 a.m.	Mon 11:30 a.m.	Islamabad, Karachi, Lahore, Tashkent
21	Mon 8:00 a.m.	Mon 11:00 a.m.	Kabul, Tehran
22	Mon 9:15 a.m.	Mon 12:15 p.m.	Kathmandu

Here is the full script of example above:

```
let
```

```
// read Input Data
```

```
input = Excel.CurrentWorkbook()[[Name="InputData"]][Content],
```

```
inputrecord=Table.First(input),
```

```
date=Text.From(inputrecord[Date]),
```

```
time=Text.PadStart(Text.From(inputrecord[Time]),4,"0"),
```

```
duration=Text.From(inputrecord[Duration]),
```

```
WebURL="http://www.timeanddate.com/worldclock/fixedtime.html?msg=Power+BI+Training&iso="&date&"T"&time&"&p1=22&ah="&duration,
```

```
Source = Web.Page(Web.Contents(WebURL)),
```

```
Data0 = Source{0}[Data],
#"Changed Type" = Table.TransformColumnTypes(Data0,{{"Header", type text}, {"Location", type text}, {"Start time", type text}, {"End time", type text}, {"Location2", type text}, {"Start time2", type text}, {"End time2", type text}}),
#"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"Header"}),
#"Replaced Value" = Table.ReplaceValue(#"Removed Columns", "*", "", Replacer.ReplaceText,{"Location"}),
#"Replaced Value1" = Table.ReplaceValue(#"Replaced Value", "*", "", Replacer.ReplaceText,{"Location2"}),
#"Removed Columns2" = Table.RemoveColumns(#"Replaced Value1",{"Location2", "Start time2", "End time2"}),
#"Removed Columns1" = Table.RemoveColumns(#"Replaced Value1",{"Location", "Start time", "End time"}),
#"Renamed Columns" = Table.RenameColumns(#"Removed Columns1",{{"Location2", "Location"}, {"Start time2", "Start time"}, {"End time2", "End time"}}),
```

```
#"Appended Query" = Table.Combine({#"Removed Columns2", #"Renamed Columns" }),
```

```
#"Grouped Rows" = Table.Group(#"Appended Query", {"Start time", "End time"}, {{"Count", each Table.RowCount(_), type number}, {"Cities", each _ , type table}}),
```

```
#"Added Custom" = Table.AddColumn(#"Grouped Rows", "Cities 2", each Table.SelectColumns([Cities],{"Location"})),
```

```
#"Added Custom2" = Table.AddColumn(#"Added Custom", "Cities 4", each Table.Transpose([Cities 2])),
```

```
#"Added Custom3" = Table.AddColumn(#"Added Custom2", "Cities 5", each Table.ToList([Cities 4],
```

```
Combiner.CombineTextByDelimiter(", ")
```

```
)),
```

```
#"Expanded Cities 5" = Table.ExpandListColumn(#"Added Custom3", "Cities 5"),
```

```
#"Removed Other Columns" = Table.SelectColumns(#"Expanded Cities 5",{"Start time", "End time", "Cities 5"}),
```

```
Final = Table.RenameColumns(#"Removed Other Columns",{{"Cities 5", "Cities"}})
```

```
in
```

```
Final
```

Summary

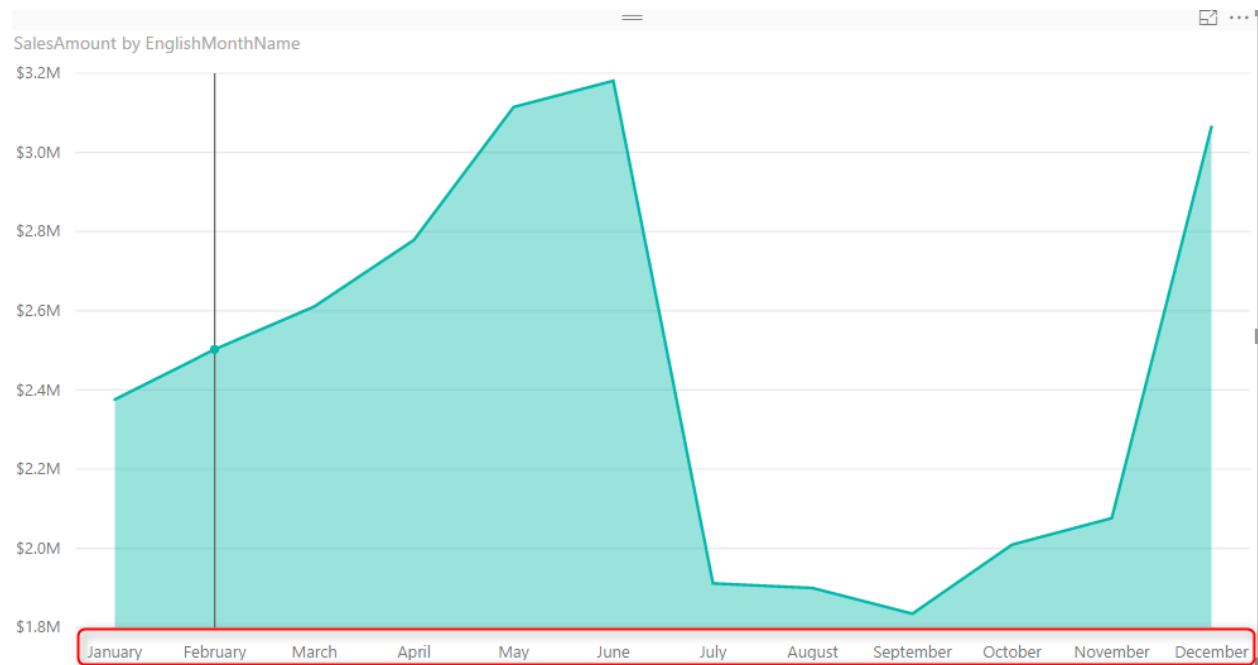
You have seen how Power Query can be useful for even none BI use case scenarios. You've learned how to use Power Query to get input parameters dynamically from Excel spreadsheet and call a web URL with that, fetches the data set, and do data transformations. This was a real world use case that showed you some strengths of Power Query. No matter if you are a BI developer or not, you will find use cases of Power Query in many situations.

Call to Action

Tell me real use case scenarios that you have used Power Query for, how did you find Power Query in solving that challenge?

Sort by Column in Power BI

Published Date: January 10, 2017



Power BI users Sorting in most of the visualizations, you can choose to sort ascending or descending based on specified data fields. However the field itself can be sorted based on another column. This feature called as Sort By Column in Power BI. This is not a new or advanced feature in Power BI. This is very basic feature that enhance your visualization significantly. Let's look at this simple feature with an example. If you like to learn more about Power BI; read [Power BI online book from Rookie to Rock Star](#).

Prerequisite

For running example of this post you will need AdventureWorksDW sample database, or you can download **Excel version** of it from here:

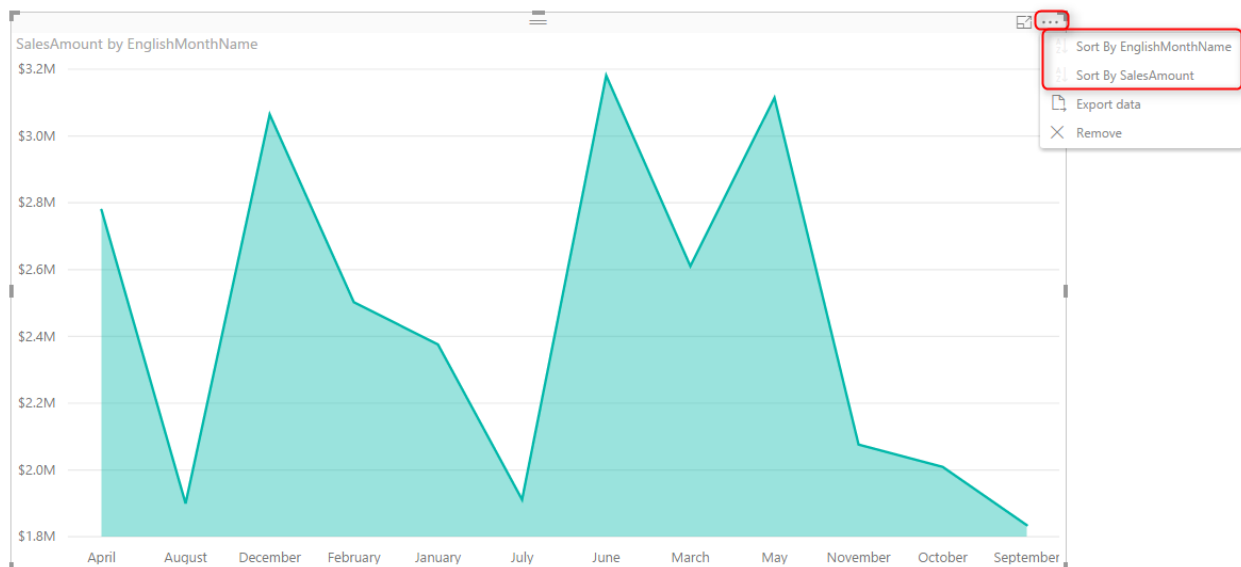
Enter Your Email to download the file (required)

[Download]

Sorting in Power BI

By Default a field's data is sorted by that field it self. It means if the field is numeric it will be ordered based on the number, if it is text it will be ordered alphabetically. Let's look at it through an example;

Create a New Power BI Desktop file, and Get Data from AdventureWorksDW, from DimDate, and FactInternetSales Tables. Check the relationship between these tables to be only based on an active relationship between OrderDateKey in the FactInternetSales and DateKey in the DimDate. remove any extra relationships. Create a visualization (Area Chart for example) with SalesAmount as Values, and EnglishMonthName as Axis.



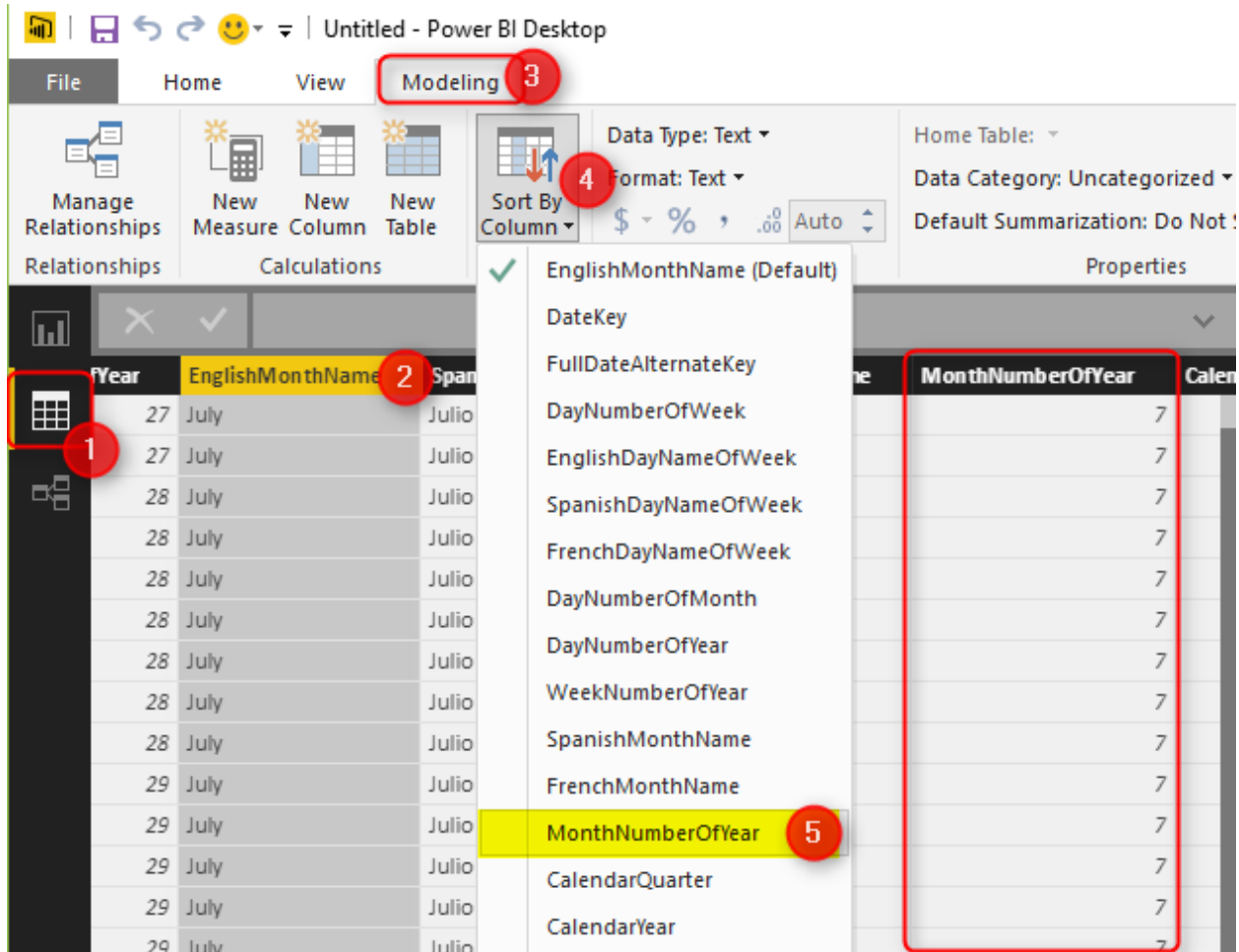
If you click on the three dot button on top right hand side of the chart you can see the sorting option for this visual. You can choose to sort based on fields that used in this visual, which can be either SalesAmount, or EnglishMonthName. If you sort it based on Sales Amount it is obviously showing you from biggest value to the lowest or reverse. This is the normal behaviour, a field will be ordered by values in the field itself.

Problem in Sorting

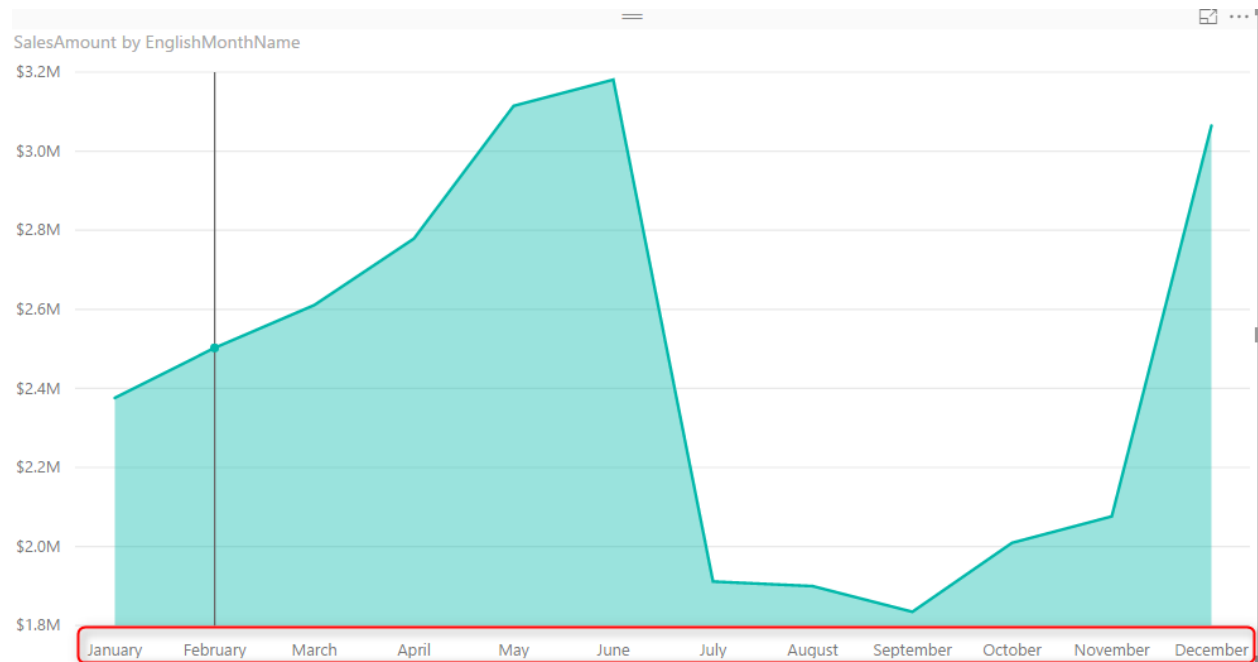
Problem happens when you want a Text field to be ordered based on something different than the value of the field. For example if you look at above chart you can see that months ordered from April to September. This is not order of months, this is alphabetical order. If you change the sorting of visual, it will only change it from A to Z, or Z to A. To make it in the order of month numbers you have to do it differently.

Sort By Column

If you wish a column to be sorted by values of another column you can simply do that in the Data tab of Power BI Desktop. First go to Data Tab, Select the field that you want to be sorted (EnglishMonthName in this example), and then from the menu option under Modeling choose Sort Column By, and select the field that contains numeric values of months (1, 2, 3, ...12). This field in our example is MonthNumberOfYear.



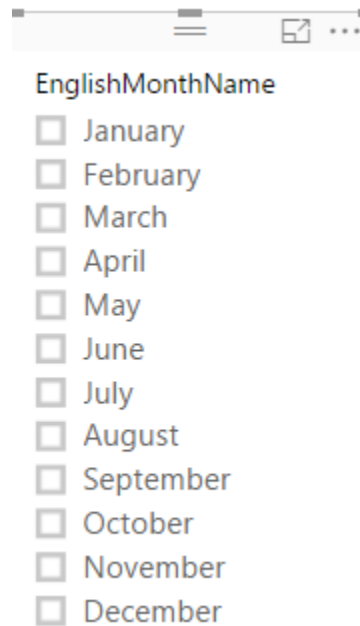
You can see in the screenshot above that MonthNumberOfYear is showing the numeric value of each month. After applying this change, simply go back to the report, and you will see the correct ordering of EnglishMonthName now.



More Cases to use Sort By Column

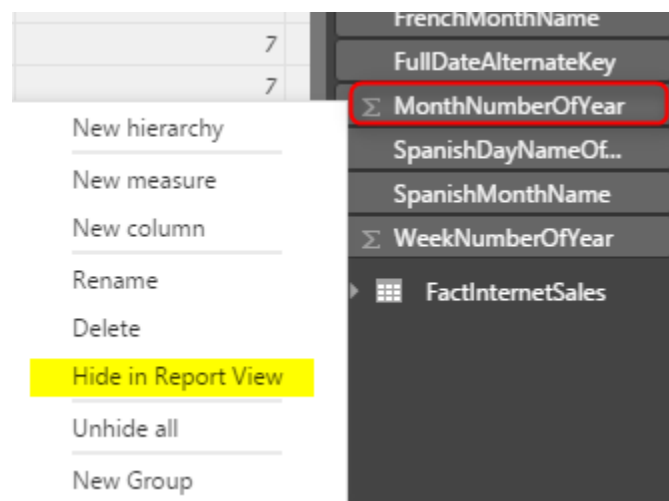
When you use this type of sorting, all visuals will be working based on this ordering. If you have a slicer, items in the slicer will be sorted and ordered based on this. This would be the way of filtering for that column from now onward. You can also think about many other possibilities and usages of this feature, here are few;

- Ordering Month Names based on Financial Period order. In this case you will need a FiscalMonthNumberOfYear column.
- Ordering items in slicer based on something different than the text. For example Priority or categories or anything like that.



Hide the Column Used for Sorting

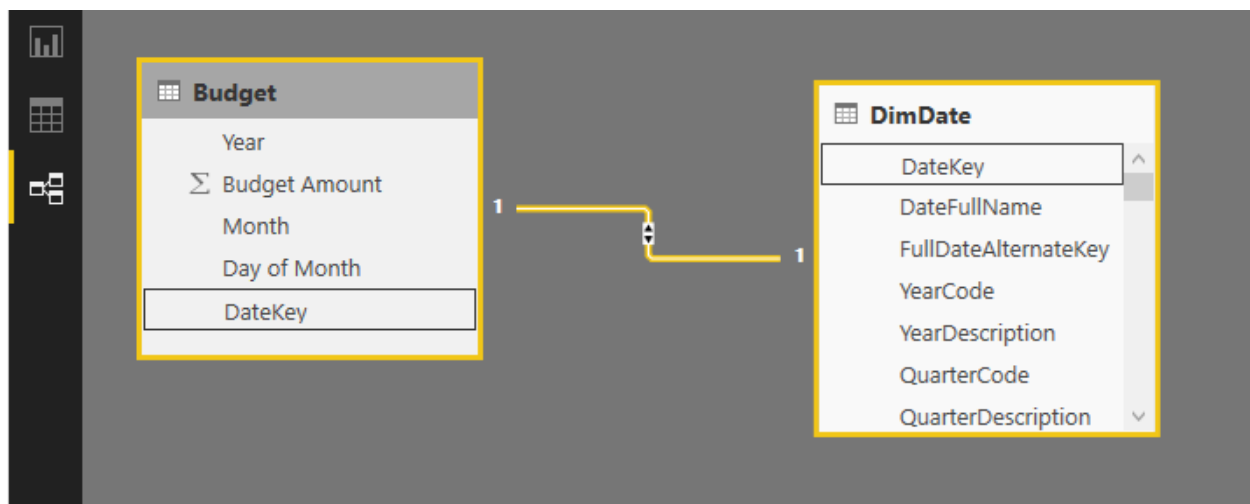
The column used for sorting (in this example MonthNumberOfYear) is not normally used for visualization itself. So you can simply Hide it from Report View.



This is a recommended approach because having too many fields in the report view is confusing for end users.

Relationship in Power BI with Multiple Columns

Published Date: April 6, 2016



You can create relationships in Power BI between tables. Relationships are useful for some functions to work across multiple tables and produce the result. Relationship between tables also makes visualization and report elements more efficient, because result of selection in one chart can affect another chart from different table. However there is a limitation in Power BI relationship that you can't create relationship based on more than one column. In other words if you want to create relationship (or join two tables) with two or more columns, you cannot! Fortunately there is a work around that I'll explain in this post. For this post you need to be familiar with Power BI and Power Query, if you are not, read them through [Power BI online book](#).

Defining the Problem

Assume that we have a budget table with fiscal year, fiscal period, and budget amount. Here is an screenshot of this table:

Year	Month	Budget Amount
2010	Mth1	50000
2010	Mth2	50000
2010	Mth3	50000
2010	Mth4	45000
2010	Mth5	50000
2010	Mth6	50000
2010	Mth7	60000
2010	Mth8	50000
2010	Mth9	50000
2010	Mth10	50000
2010	Mth11	50000
2010	Mth12	50000
2011	Mth1	50000
2011	Mth2	55000
2011	Mth3	55000
2011	Mth4	65000
2011	Mth5	55000

If I want to do date analysis and date based calculations it is best to create a relationship between the budget table and a date dimension. Here is a view of my date dimension:
(Here is an example of creating [date dimension with Power Query](#), and the script for creating [date dimension in SQL Server](#))

erOffYear	ISOWeekYearCode	WeekDay	WeekDayName	NZFiscalYearCode	NZFiscalYearDescription	NZFiscalQuarterCode	NZFiscalQuarterYearCode	NZFiscalQuarterDescription	NZFiscalMonthCode	NZFiscalMonthYearCode	NZFiscalMonthL
27	199027	3	Tuesday	1991	FY 1991	2	19912	QTR 2	4	199104	Month 4
28	199028	3	Tuesday	1991	FY 1991	2	19912	QTR 2	4	199104	Month 4
29	199029	3	Tuesday	1991	FY 1991	2	19912	QTR 2	4	199104	Month 4
30	199030	3	Tuesday	1991	FY 1991	2	19912	QTR 2	4	199104	Month 4
31	199031	3	Tuesday	1991	FY 1991	2	19912	QTR 2	4	199104	Month 4
32	199032	3	Tuesday	1991	FY 1991	2	19912	QTR 2	5	199105	Month 5
33	199033	3	Tuesday	1991	FY 1991	2	19912	QTR 2	5	199105	Month 5
34	199034	3	Tuesday	1991	FY 1991	2	19912	QTR 2	5	199105	Month 5
35	199035	3	Tuesday	1991	FY 1991	2	19912	QTR 2	5	199105	Month 5
27	199127	3	Tuesday	1992	FY 1992	2	19922	QTR 2	4	199204	Month 4
28	199128	3	Tuesday	1992	FY 1992	2	19922	QTR 2	4	199204	Month 4
29	199129	3	Tuesday	1992	FY 1992	2	19922	QTR 2	4	199204	Month 4
30	199130	3	Tuesday	1992	FY 1992	2	19922	QTR 2	4	199204	Month 4
31	199131	3	Tuesday	1992	FY 1992	2	19922	QTR 2	4	199204	Month 4
32	199132	3	Tuesday	1992	FY 1992	2	19922	QTR 2	5	199205	Month 5
33	199133	3	Tuesday	1992	FY 1992	2	19922	QTR 2	5	199205	Month 5
34	199134	3	Tuesday	1992	FY 1992	2	19922	QTR 2	5	199205	Month 5
35	199135	3	Tuesday	1992	FY 1992	2	19922	QTR 2	5	199205	Month 5
28	199228	3	Tuesday	1993	FY 1993	2	19932	QTR 2	4	199304	Month 4

To join these two tables I can add a day column to the budget table, and then join them based on three columns: fiscal year, fiscal period, and day (day of month). So here is the budget table with the new day column added and month value a bit polished to remove "Mth" from the month label;

	Year	Month	Budget Amount	Day of Month
1	2010	1	50000	1
2	2010	2	50000	1
3	2010	3	50000	1
4	2010	4	45000	1
5	2010	5	50000	1
6	2010	6	50000	1
7	2010	7	60000	1
8	2010	8	50000	1
9	2010	9	50000	1
10	2010	10	50000	1
11	2010	11	50000	1
12	2010	12	50000	1
13	2011	1	50000	1
14	2011	2	55000	1
15	2011	3	55000	1

Now if I want to create relationship between date dimension and budget table based on these three columns I cannot! The create relationship dialog doesn't allow me to select multiple columns, and because with a single column a Key field won't be identified so the relationship can't be created.



Create Relationship

Select tables and columns that relate to one another.

DimDate
▼

Code	NZFiscalQuarterDescription	NZFiscalMonthCode	NZFiscalMonthYearCode	NZFiscalMonthDescription	Flag
19912	QTR 2	4	199104	Month 4	
19912	QTR 2	4	199104	Month 4	
19912	QTR 2	4	199104	Month 4	


Budget
▼

Year	Budget Amount	Month	Day of Month
2010	50000	1	1
2010	50000	2	1
2010	50000	3	1

Cardinality
▼
Cross filter direction
▼

☐ Make this relationship active

☐ Assume Referential Integrity.

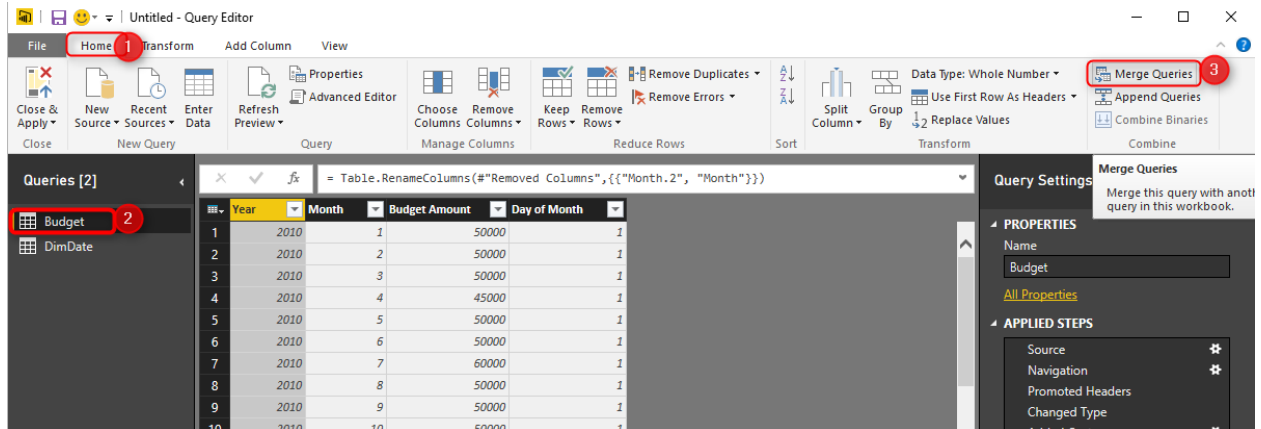
 You can't create a relationship between these two columns because one of the columns must have unique values.

OK
Cancel

Workaround

The workaround for this problem is easy. Power BI doesn't allow relationship in model based on multiple columns, but Power Query can join tables with as many as columns we want. So what I can do as a workaround is to join budget table to date dimension in Power Query and fetch the date key. then I'll use the date key as a single field relationship in Power BI modelling section.

First I open Merge Queries from the Combine section of Home tab;



Here is how I join two tables based on multiple columns: I can hold CTRL key and select columns one by one (in the right order of joining)

Merge

Select a table and matching columns to create a merged table.

Budget

Year	1	Month	2	Budget Amount	Day of Month	3
2010		1		50000		1
2010		2		50000		1
2010		3		50000		1
2010		4		45000		1
2010		5		50000		1

DimDate

Name	NZFiscalYearCode	1	NZFiscalMonthCode	2	DayNumberOfMonth	3	NZFiscalMonthYearCode	NZFiscalYear
	1990		10		1		199010	FY 1990
	1990		10		2		199010	FY 1990
ay	1990		10		3		199010	FY 1990
	1990		10		4		199010	FY 1990
	1990		10		5		199010	FY 1990

Join Kind

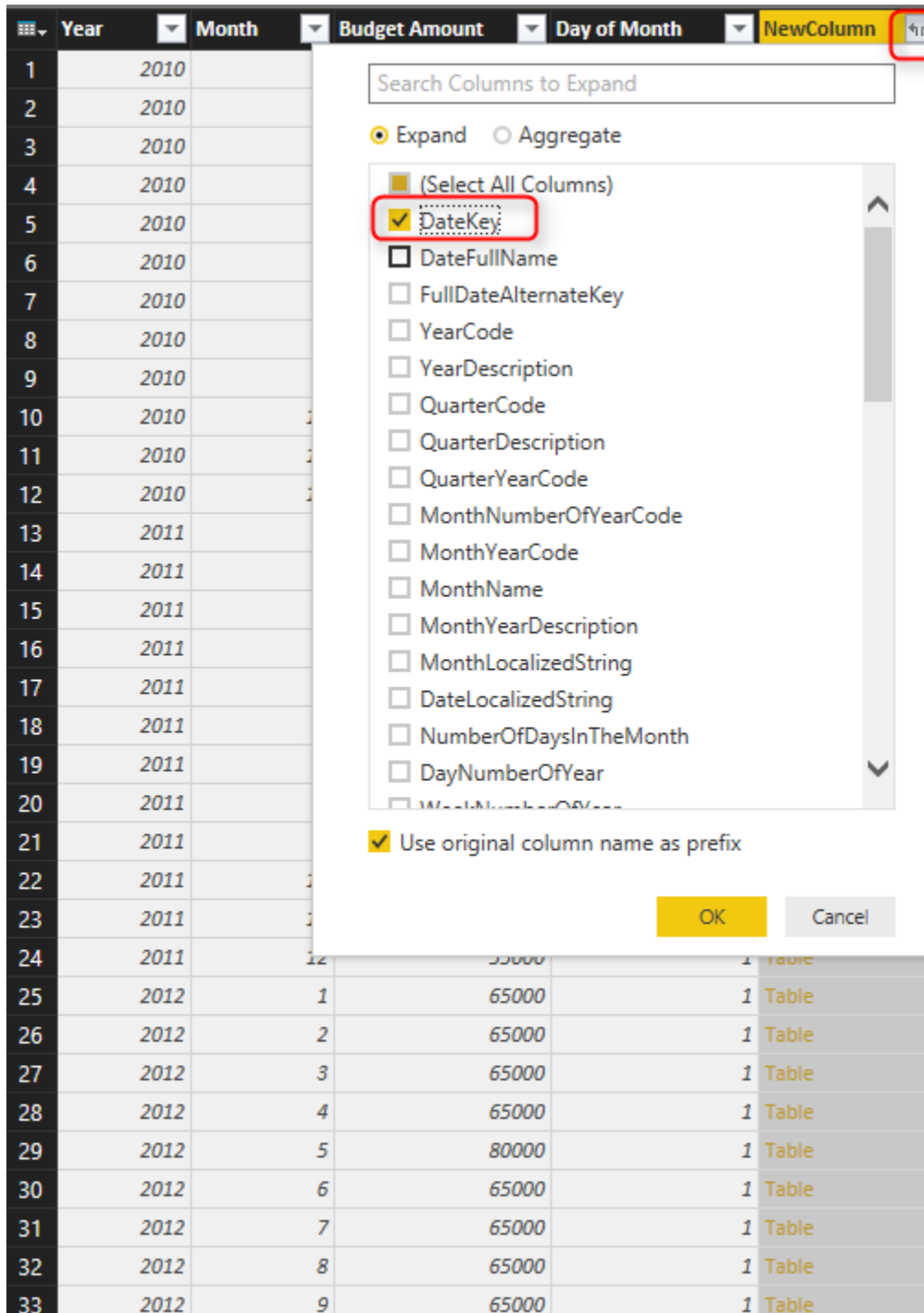
Left Outer (all from first, matching from second)

The selection has matched 48 out of the first 48 rows.

OK

Cancel

Then I'll have the new table embedded as a result of the join;



Search Columns to Expand

☒ Expand ☐ Aggregate

☒ (Select All Columns)

☒ DateKey

☐ DateFullName

☐ FullDateAlternateKey

☐ YearCode

☐ YearDescription

☐ QuarterCode

☐ QuarterDescription

☐ QuarterYearCode

☐ MonthNumberOfYearCode

☐ MonthYearCode

☐ MonthName

☐ MonthYearDescription

☐ MonthLocalizedString

☐ DateLocalizedString

☐ NumberOfDaysInTheMonth

☐ DayNumberOfYear

☐ WeekNumberOfYear

☒ Use original column name as prefix

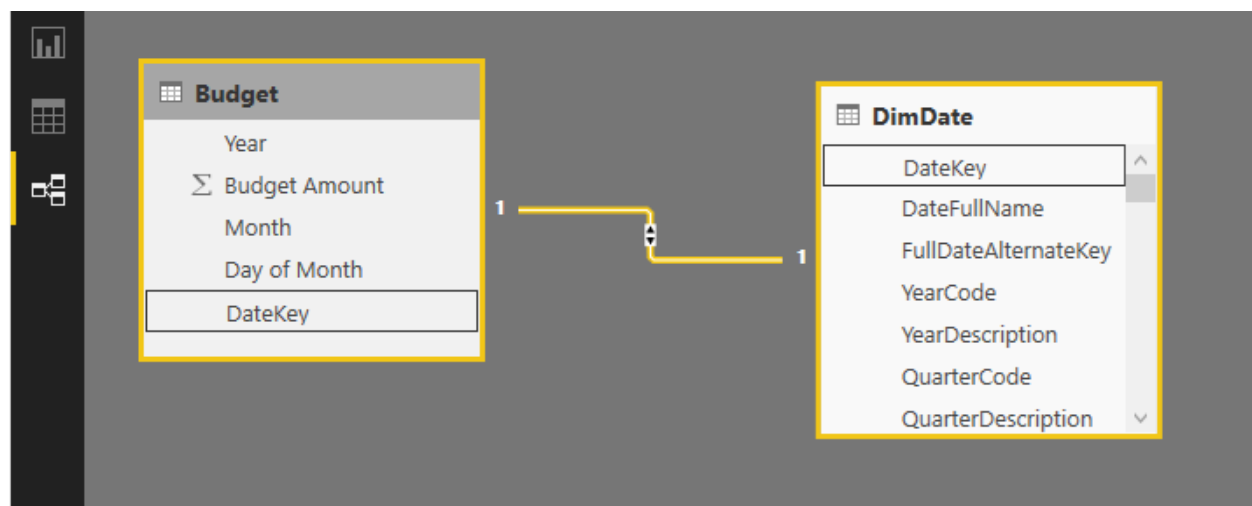
OK Cancel

	Year	Month	Budget Amount	Day of Month	NewColumn
1	2010				
2	2010				
3	2010				
4	2010				
5	2010				
6	2010				
7	2010				
8	2010				
9	2010				
10	2010				
11	2010				
12	2010				
13	2011				
14	2011				
15	2011				
16	2011				
17	2011				
18	2011				
19	2011				
20	2011				
21	2011				
22	2011				
23	2011				
24	2011	12	55000	1	Table
25	2012	1	65000	1	Table
26	2012	2	65000	1	Table
27	2012	3	65000	1	Table
28	2012	4	65000	1	Table
29	2012	5	80000	1	Table
30	2012	6	65000	1	Table
31	2012	7	65000	1	Table
32	2012	8	65000	1	Table
33	2012	9	65000	1	Table

So I'll just pick the key field from embedded table;

	Year	Month	Budget Amount	Day of Month	DateKey
1	2010	1	50000	1	20090401
2	2010	2	50000	1	20090501
3	2010	3	50000	1	20090601
4	2010	4	45000	1	20090701
5	2010	5	50000	1	20090801
6	2010	6	50000	1	20090901
7	2010	7	60000	1	20091001
8	2010	8	50000	1	20091101
9	2010	9	50000	1	20091201
10	2010	10	50000	1	20100101
11	2010	11	50000	1	20100201
12	2010	12	50000	1	20100301
13	2011	1	50000	1	20100401

And after save and closing query editor window I can create relationship in Power BI model based on a single column;



M or DAX? That is the Question!

Published Date: March 3, 2017



What is the main difference between M and DAX? Why we can do a calculated column in two different places? What are pros and cons of each? Which one should I use for creating a profit column? Why I cannot do all of it in only one; DAX or M! Why two different languages?! Why structure of these two are so different? ... If any of these are your questions, then you need to read this post. In this post I'll go through differences of these two languages, and explain why, when, where of it. Normally I don't get this question asked from students of my Power BI course, because I elaborate the difference in details. However, if you have this question, this is a post for you. If you would like to learn more about Power BI; read [Power BI book; from Rookie to Rook Star](#).

What is M?

M is the scripting language behind the scene for Power Query. M is the informal name of this language. the formal name is: Power Query Formula Language! Which is really long, and even Microsoft refer it to M. M stands for many things, but one of the most common words of it is Mashup. Which means this language is capable of data mashup, and transformation. M is a functional language. and structure of M script can be similar to this:


```

let
    FirstAndLastDayOfTheMonth = (date) =>
        let
            dated=Date.FromText(date) ,
            year=Date.Year(dated) ,
            month=Date.Month(dated) ,
            FirstDateText=Text.From(year)&"-"&Text.From(month)&"-01" ,
            FirstDate=Date.FromText(FirstDateText) ,
            daysInMonth=Date.DaysInMonth(dated) ,
            LastDateText=Text.From(year)&"-"&Text.From(month)&"-"&Text.From(daysInMonth) ,
            LastDate=Date.FromText(LastDateText) ,
            record=Record.AddField([], "First Date of Month",FirstDate) ,
            resultset=Record.AddField(record,"Last Date of Month",LastDate)
        in
            resultset
in
    FirstAndLastDayOfTheMonth("30/07/2015")

```

[Source: Day Number of Year Function in Power Query](#)

M is a step by step language structure. Usually (Not always), every line in M script is a data transformation step. and the step after that will use the result of previous step. It is usually easy to follow the structure of M language for a programmer. because it is understandable with programming blocks of Let and In, and some other programming language features alike.

What is DAX?

DAX is Data Analysis eXpression Language. This is the common language between SQL Server Analysis Services Tabular, Power BI, and Power Pivot in Excel. DAX is an expression language, and unlike M, it is very similar to Excel functions. In fact, DAX has many common functions with Excel. However DAX is much more powerful than Excel formula in many ways. Here is an example DAX expression:

```

Sales Rolling 12 Months =
CALCULATE(
    SUM(FactInternetSales[SalesAmount]),
    DATESBETWEEN
        (DimDate[FullDateAlternateKey],
        NEXTDAY(SAMEPERIODLASTYEAR(LASTDATE(DimDate[FullDateAlternateKey]))),
        LASTDATE(DimDate[FullDateAlternateKey])
    ),
    ALL(DimDate)
)

```

[Source: Secret of Time Intelligence Functions in Power BI](#)

DAX calculations are built in a way that makes sense mostly for Excel users. Normally Excel users are very comfortable with this language. Everything goes through functions. DAX doesn't have programming blocks in it, and is combination of function uses, filters, and expressions.

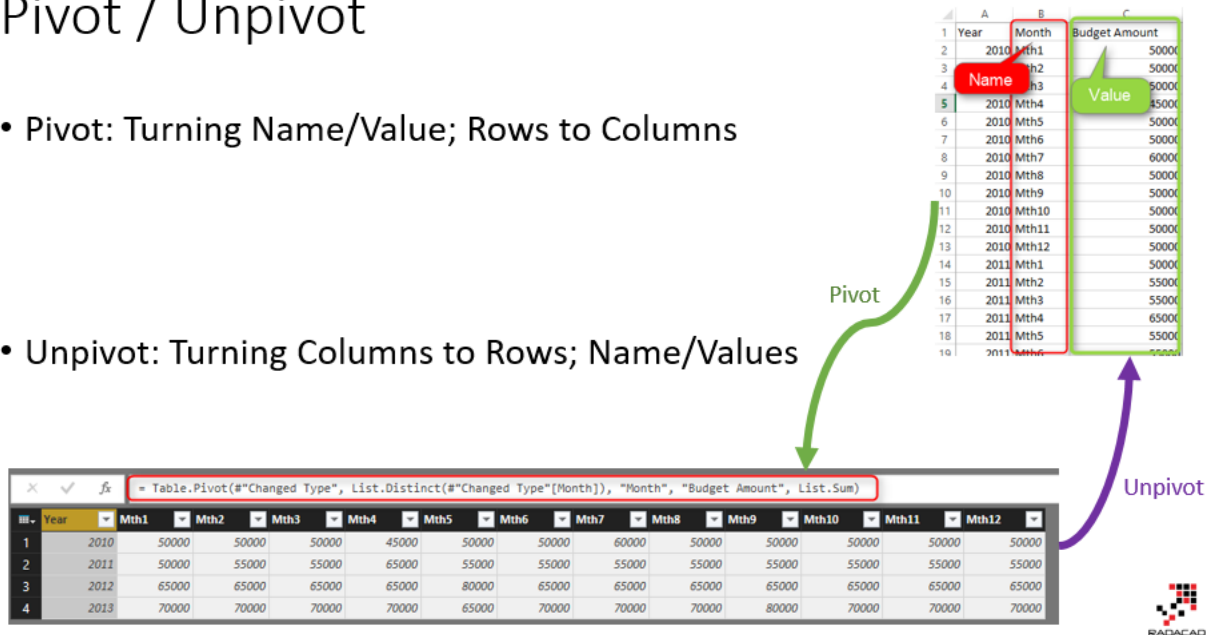
Example Usage of M

M can be used in many data transformation scenarios. As an example, it can be used to Pivot or Unpivot Data, To [Group](#) it based on a number of columns. Here is how a [Pivot/Unpivot](#) can work in Power Query;

Pivot / Unpivot

- Pivot: Turning Name/Value; Rows to Columns

- Unpivot: Turning Columns to Rows; Name/Values



Example Usage of DAX?

DAX can be used for many calculation for analyzing data. For example, calculating Year To Date, Calculating [Rolling 12 Months Average](#), or anything like that. Here is an example which based on a selection criteria in the report and few simple DAX expressions we can do a [customer retention](#) case with DAX;

FullName	Total Revenue	Last Period Revenue	Lost Customers	New Customers
Aaron Adams	\$118	\$118	0	1
Aaron Alexander	\$70	\$70	0	1
Aaron Allen	\$3,400		1	0
Aaron Baker	\$1,751	\$1,751	0	1
Aaron Bryant	\$134	\$134	0	1
Aaron Butler	\$15	\$15	0	1
Aaron Campbell	\$1,155	\$1,155	0	1
Aaron Carter	\$40	\$40	0	1
Aaron Chen	\$40	\$40	0	1
Aaron Coleman	\$62	\$62	0	1
Aaron Collins	\$6,047	\$2,469	0	0
Aaron Diaz	\$6,030	\$2,451	0	0
Aaron Edwards	\$94	\$94	0	1
Aaron Evans	\$2,433	\$2,433	0	1

Period

☐ 30

☐ 60

☐ 120

☐ 180

☐ 365

☒ 1461

☐ 2922

1461
Selected Period

EnglishProductName

☐ Adjustable Race

☐ All-Purpose Bike Stand

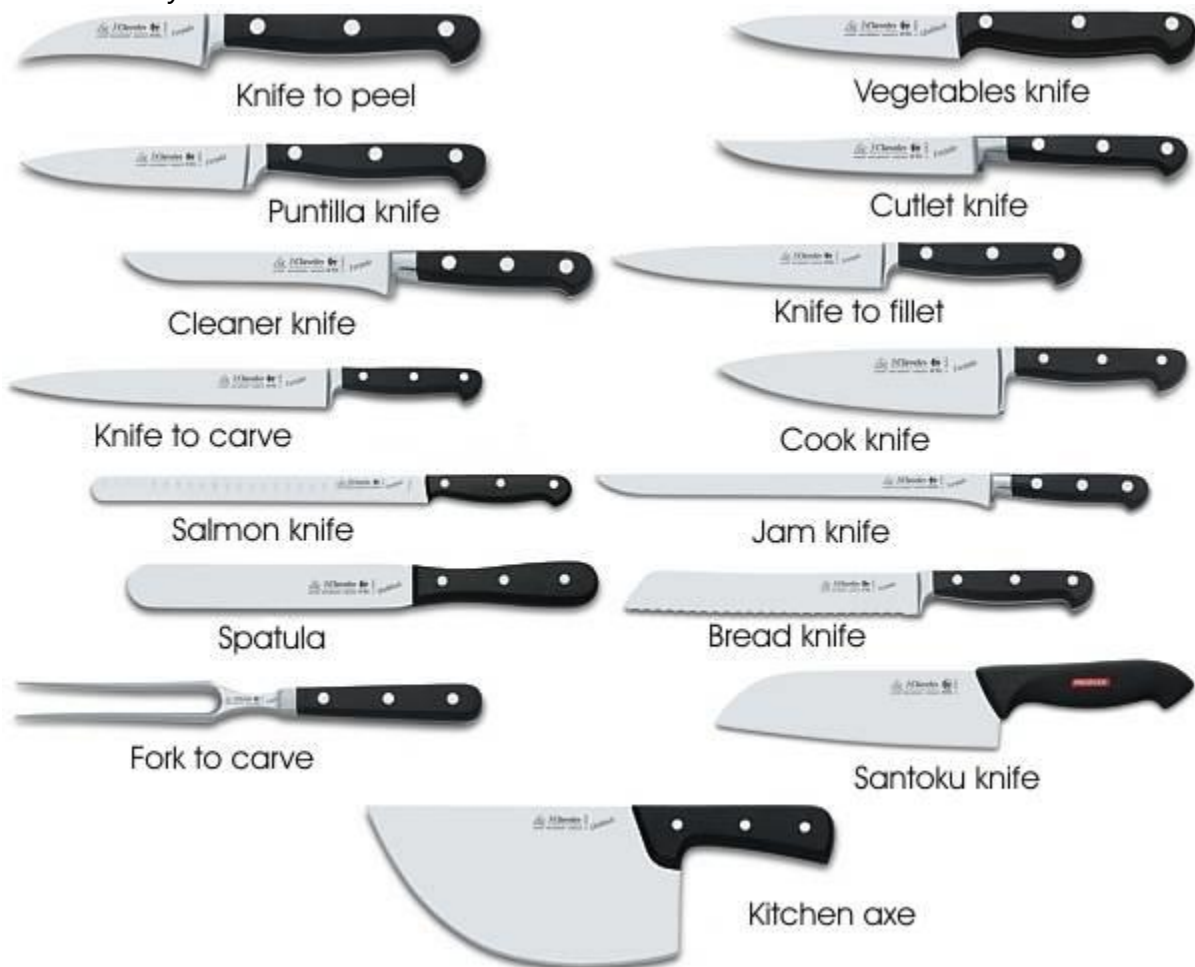
☐ AWC Logo Cap

☐ BB Ball Bearing

Calculated Column Dilemma

The main question of choosing between DAX and M comes from calculated column dilemma in my opinion. You can create many calculated columns in both M or DAX, and it is confusing where is the best place to do it, or why there are two different places to do it?! As an example; you can create a full name which is concatenated of FirstName and LastName column. You can do that in M, and also in DAX. So this question comes up that: Why two different places? Which one is best to use? can we always use one language?

To answer this question, I would like to use another example; There are many types of knives, and you can use almost all of them to cut cheese!



reference: <http://forkitchen.blogspot.co.nz/2008/10/what-are-different-types-of-kitchen.html>

Almost every knife in above picture can be used for cutting cheese except one of them! So why there are so many knives for cutting cheese?! The answer is that; these are not knives for cutting cheese! each knife is good for doing one special case. for cutting

bread, bread knife gives you the best result. for cutting a fillet you normally need another type of knife. but as you agree, for some cases (such as cutting cheese!) you can use many of these knives. Let's know go back to the original question;

Why I can create same calculated column in DAX or M?

These two languages built independently. They built in a way that they can handle most of business related solutions. So as a result there are some use cases that both languages are capable of doing it. As an example both of these languages can easily be used to create a concatenated column of two other columns.

Which one is best?

Quick answer is Depends! Depends on type of usage. If you want to create a concatenated column; Power Query (M) is better option in my view, because that is normally like the ETL part of your BI solution, you can simply build your model and data sets in a way you like it to be. But if you want to create something like Year To Date; Obviously you can do that in Power Query or M, but it will be lots of code, and you have to consider many combinations of possibilities to create a correct result, while in DAX you can simply create that with usage of TotalYTD function. So the answer is; there is no best language between these two. The type of usage identifies which one is best. Normally any changes to prepare the data for the model is best to be done in M, and any analysis calculation on top of the model is best to be done in DAX.

Two Languages for Two Different Purposes

There are many programming languages in the world, each language has its own pros and cons. JavaScript is a language of web scripting, which is very different from ASP.NET or PHP. Same thing happens here. When M born, it meant to be a language for data transformation, and it is still that language. DAX was created to answer business analysis questions.

What Questions DAX Can Answer?

DAX is the analytical engine in Power BI. It is best language to answer analytical questions which their responses will be different based on the selection criteria in the report. For example; You might want to calculate Rolling 12 Months Average of Sales. It is really hard if you want to calculated that in M, because you have to consider all different types of possibilities; Rolling 12 months for each product, for every customer, for every combinations and etc. However if you use a DAX calculation for it, the analytical engine of DAX take care of all different combinations selected through Filter Context in the report.

What Questions M Can Answer?

M is Data Transformation engine in Power BI. You can use M for doing any data preparation and data transformation before loading that into your model. Instead of bringing three tables of DimProduct, DimProductSubcategory, and DimProductCategory, you can merge them all together in Power Query, and create a single DimProduct including all columns from these tables, and load that into the model. Loading all of these into the model and using DAX to relate these with each other means consuming extra memory for something that is not required to be in the model. M can simply combine those three tables with each other and based on "Step Based" operational structure of M, they can be simply used to create a final data set.

As a Power BI Developer Which Language Is Important to Learn?

Both! With no hesitation! M is your ETL language, and DAX is the analytical language. You cannot live with only one. If you want to be an expert in Power BI, you should be an expert in both of these languages. There are some cases, that one of the languages will be used more than the other one. However, you will need very good understanding of both languages to understand which one is best for which purpose, and easily can use it in real-world scenarios.

Secret of Time Intelligence Functions in Power BI

Published Date: April 5, 2016

CalendarYear	EnglishMonthName ▲	SalesAmount	Last Year Sales
2005	July	\$473,388.16	
	August	\$506,191.69	
	September	\$473,943.03	
	October	\$513,329.47	
	November	\$543,993.41	
	December	\$755,527.89	
	Total	\$3,266,373.66	
2006	January	\$596,746.56	
	February	\$550,816.69	
	March	\$644,135.20	
	April	\$663,692.29	
	May	\$673,556.20	
	June	\$676,763.65	
	July	\$500,365.16	\$473,388.16
	August	\$546,001.47	\$506,191.69
	September	\$350,466.99	\$473,943.03
	October	\$415,390.23	\$513,329.47
	November	\$335,095.09	\$543,993.41
	December	\$577,314.00	\$755,527.89
	Total	\$6,530,343.53	\$3,266,373.66
2007	January	\$438,865.17	\$596,746.56
	February	\$489,090.34	\$550,816.69
	March	\$485,574.79	\$644,135.20
	April	\$506,399.27	\$663,692.29
	May	\$562,772.56	\$673,556.20
	June	\$554,799.23	\$676,763.65
	July	\$886,668.84	\$500,365.16
	August	\$847,413.51	\$546,001.47
	September	\$1,010,258.13	\$350,466.99
	October	\$1,080,449.58	\$415,390.23
	November	\$1,196,981.11	\$335,095.09
	December	\$1,731,787.77	\$577,314.00
	Total	\$9,791,060.30	\$6,530,343.53
2008	January	\$1,340,244.95	\$438,865.17
	February	\$1,462,479.83	\$489,090.34
	March	\$1,480,905.18	\$485,574.79
	April	\$1,608,750.53	\$506,399.27
	May	\$1,878,317.51	\$562,772.56
	June	\$1,949,361.11	\$554,799.23
	July	\$50,840.63	\$886,668.84
	August		\$847,413.51
	September		\$1,010,258.13
	October		\$1,080,449.58
	November		\$1,196,981.11

If you be familiar with Power Pivot you know that there is a functionality in Power Pivot for setting a table as a Date Table. This enables time intelligence functions such as year to date to work properly over your data model. However this option is not available in Power BI Desktop. There is no way that you mark a table as a date table. So time intelligence functions might not work properly in Power BI. In this post I'll show you some methods that you can get time intelligence functions working.

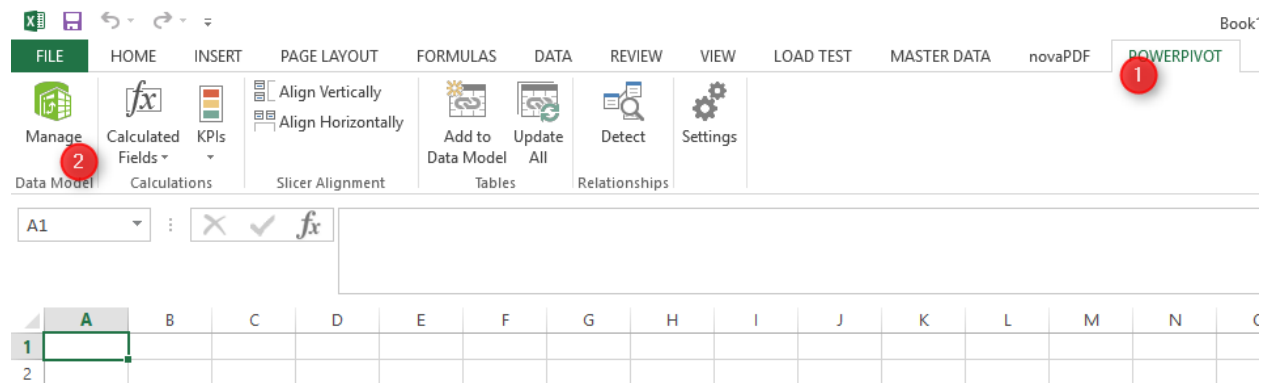
Prerequisite

For running samples of this post you need to have AdventureWorksDW database installed on your SQL Server. If you don't have it download it from [here](#) and restore it on your machine.

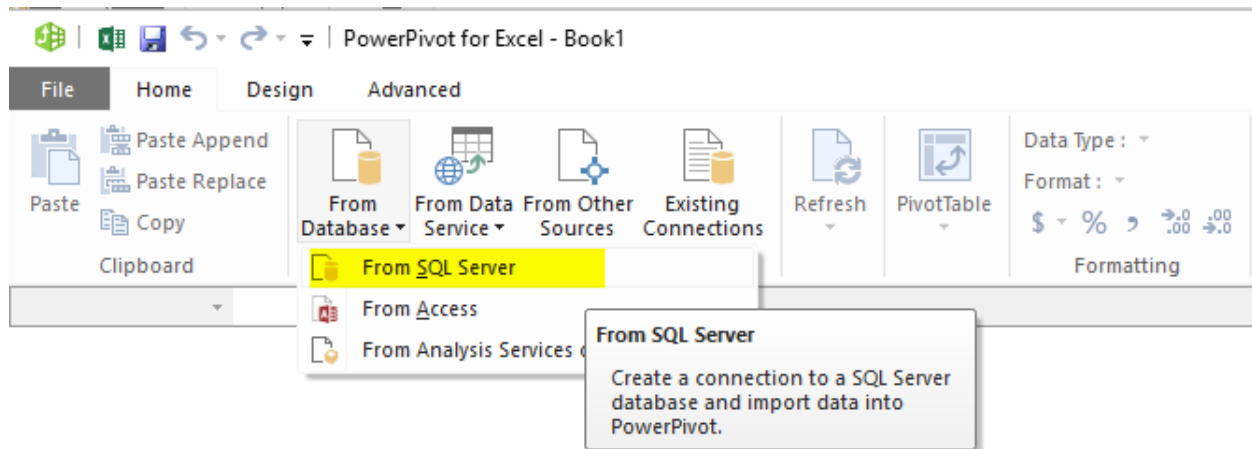
If you are not Familiar with Time Intelligence functions, or DAX, or even Power BI; read [Power BI online book](#)'s other chapters.

Time Intelligence in Power Pivot

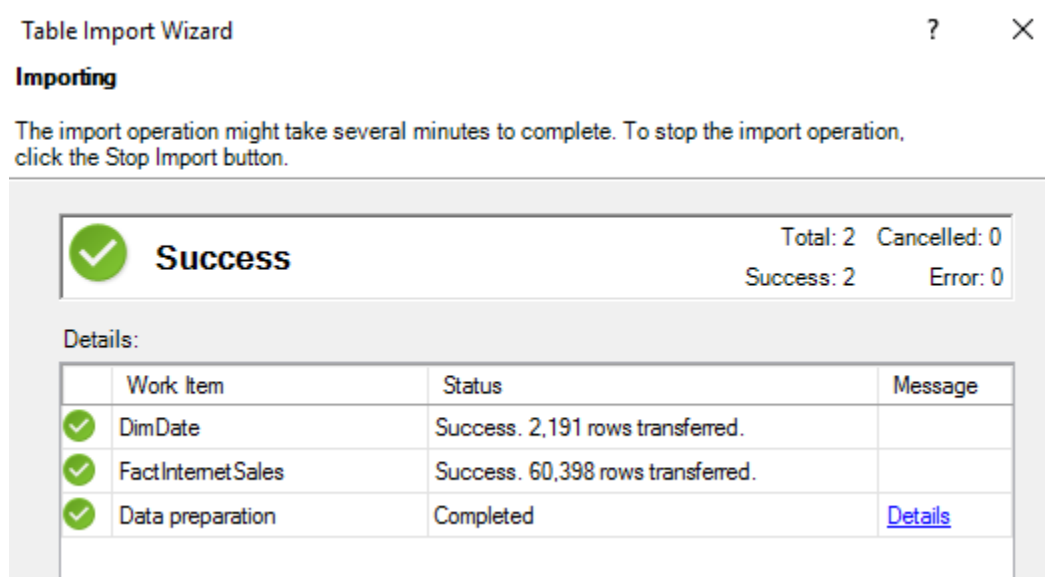
Let's start with calculating year to date in Power Pivot. So I'll start by creating an Excel file and going to Power Pivot tab of it, then Manage



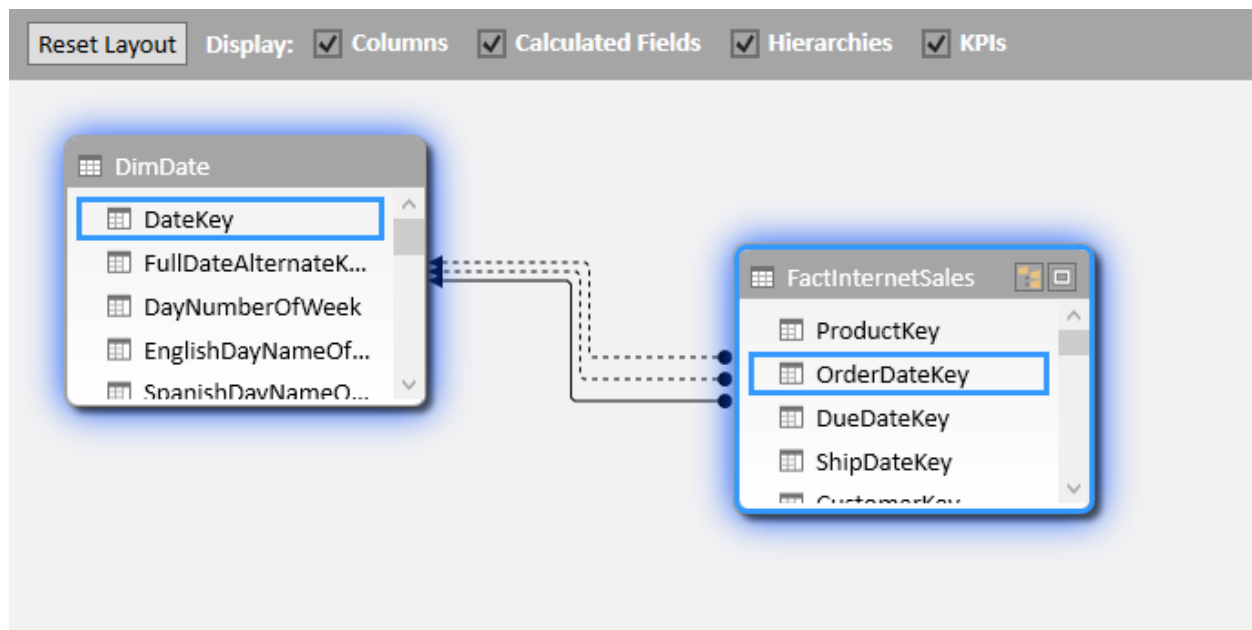
In Power Pivot editor I choose to get data from external data sources, and from SQL Server.



And I choose two tables from AdventureWorksDW database; DimDate, and FactInternetSales.



I then make sure that the connection between tables is based on OrderDateKey (active relationship);



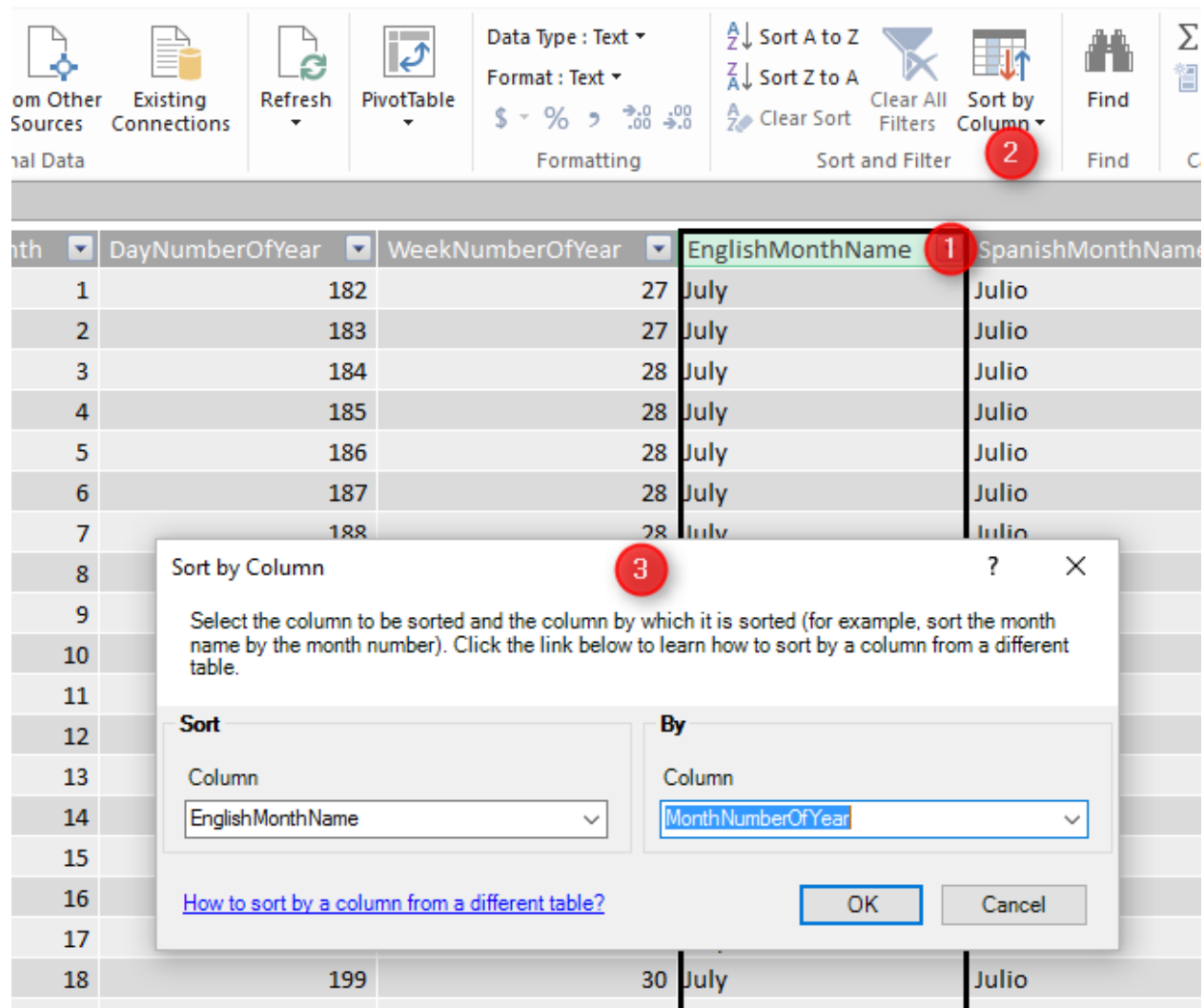
Now let's create year to date measure in FactInternetSales with simple TotalYTD Dax function as below:

Sales YTD:=TOTALYTD(SUM(FactInternetSales[SalesAmount]),DimDate[FullDateAlternateKey])

PowerPivot for Excel - Book1

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey
528	20070801	20070813	20070808	14870	1	10
528	20070802	20070814	20070809	15319	1	10

I also order months with the correct order of their number as below



om Other Sources Existing Connections Refresh PivotTable

Data Type : Text Format : Text

Sort A to Z Sort Z to A Clear Sort Clear All Filters Sort by Column

Find Find

Month	DayNumberOfYear	WeekNumberOfYear	EnglishMonthName	SpanishMonthName
1	182	27	July	Julio
2	183	27	July	Julio
3	184	28	July	Julio
4	185	28	July	Julio
5	186	28	July	Julio
6	187	28	July	Julio
7	188	28	July	Julio
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18	199	30	July	Julio
...

Sort by Column

Select the column to be sorted and the column by which it is sorted (for example, sort the month name by the month number). Click the link below to learn how to sort by a column from a different table.

Sort

Column

EnglishMonthName

By

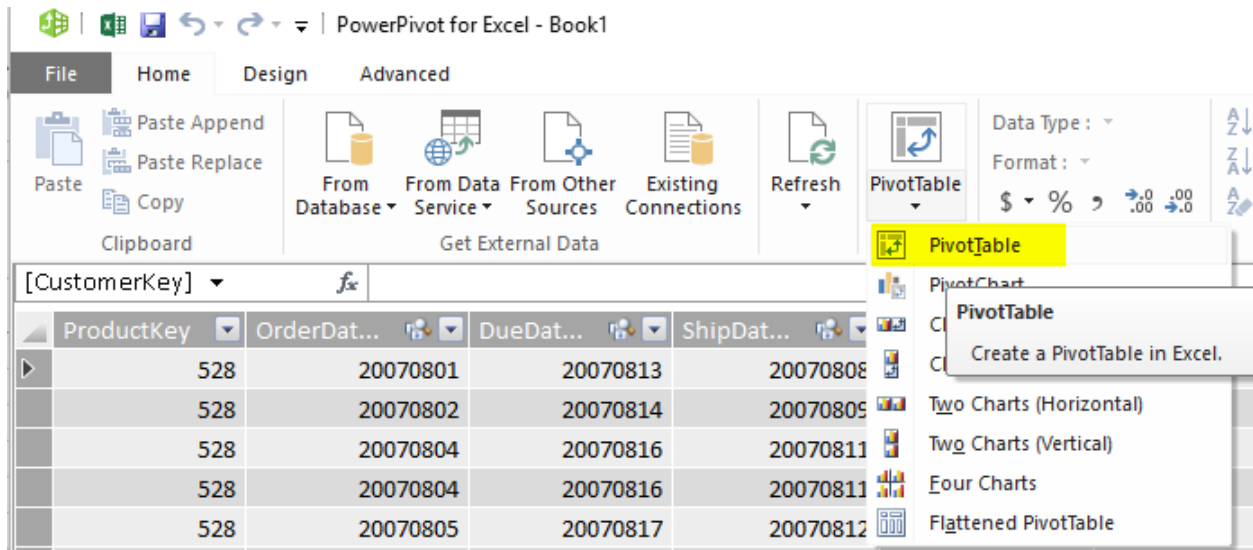
Column

MonthNumberOfYear

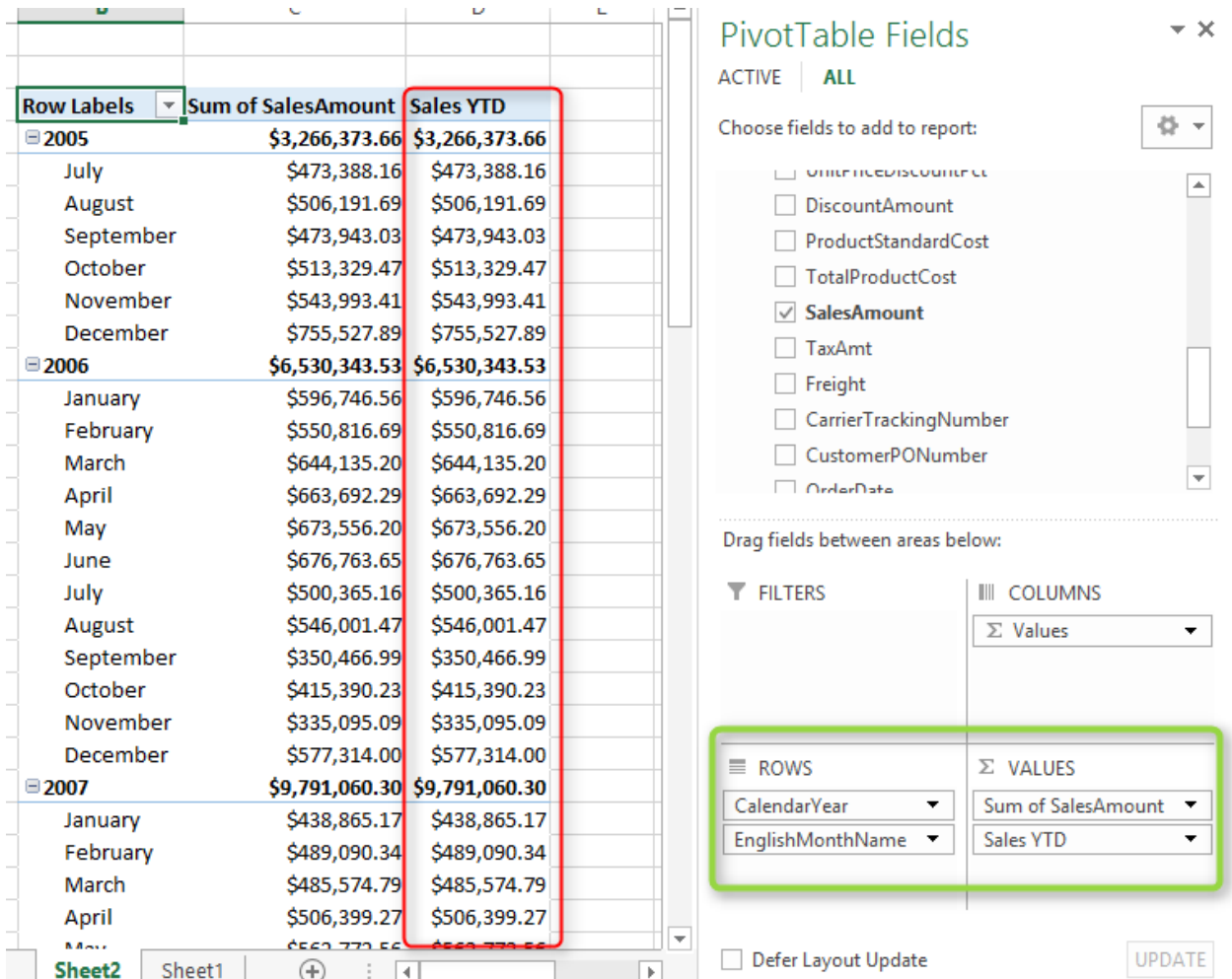
[How to sort by a column from a different table?](#)

OK Cancel

Now I can analyze this model in Excel PivotTable;



When I look at Sales year to date value I see that it is not working properly, and it only shows same Sales Amount as YTD value. which is not correct obviously



The screenshot shows the PivotTable Fields task pane on the right and the PivotTable data on the left. The task pane shows the following fields:

- ACTIVE: ALL
- Choose fields to add to report:
 - ☐ UnitPriceDiscount
 - ☐ DiscountAmount
 - ☐ ProductStandardCost
 - ☐ TotalProductCost
 - ☒ SalesAmount
 - ☐ TaxAmt
 - ☐ Freight
 - ☐ CarrierTrackingNumber
 - ☐ CustomerPONumber
 - ☐ OrderDate
- Drag fields between areas below:
 - FILTERS:**
 - COLUMNS:** Σ Values
 - ROWS:** CalendarYear, EnglishMonthName
 - VALUES:** Σ Values, Sum of SalesAmount, Sales YTD
- ☐ Defer Layout Update
- UPDATE

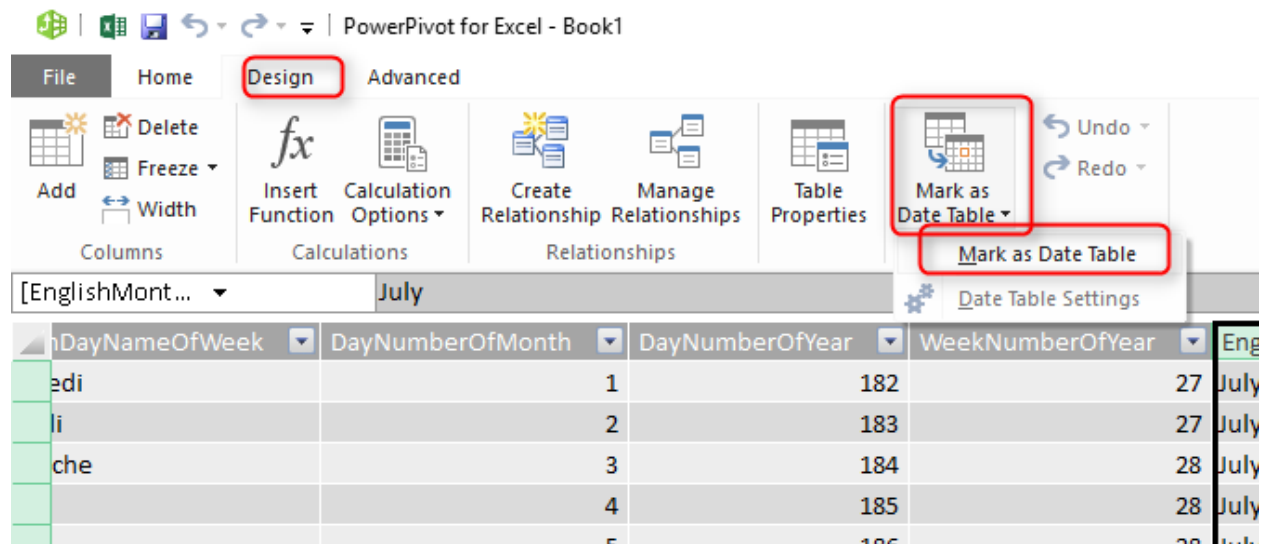
The PivotTable data is as follows:

Row Labels	Sum of SalesAmount	Sales YTD
2005	\$3,266,373.66	\$3,266,373.66
July	\$473,388.16	\$473,388.16
August	\$506,191.69	\$506,191.69
September	\$473,943.03	\$473,943.03
October	\$513,329.47	\$513,329.47
November	\$543,993.41	\$543,993.41
December	\$755,527.89	\$755,527.89
2006	\$6,530,343.53	\$6,530,343.53
January	\$596,746.56	\$596,746.56
February	\$550,816.69	\$550,816.69
March	\$644,135.20	\$644,135.20
April	\$663,692.29	\$663,692.29
May	\$673,556.20	\$673,556.20
June	\$676,763.65	\$676,763.65
July	\$500,365.16	\$500,365.16
August	\$546,001.47	\$546,001.47
September	\$350,466.99	\$350,466.99
October	\$415,390.23	\$415,390.23
November	\$335,095.09	\$335,095.09
December	\$577,314.00	\$577,314.00
2007	\$9,791,060.30	\$9,791,060.30
January	\$438,865.17	\$438,865.17
February	\$489,090.34	\$489,090.34
March	\$485,574.79	\$485,574.79
April	\$506,399.27	\$506,399.27

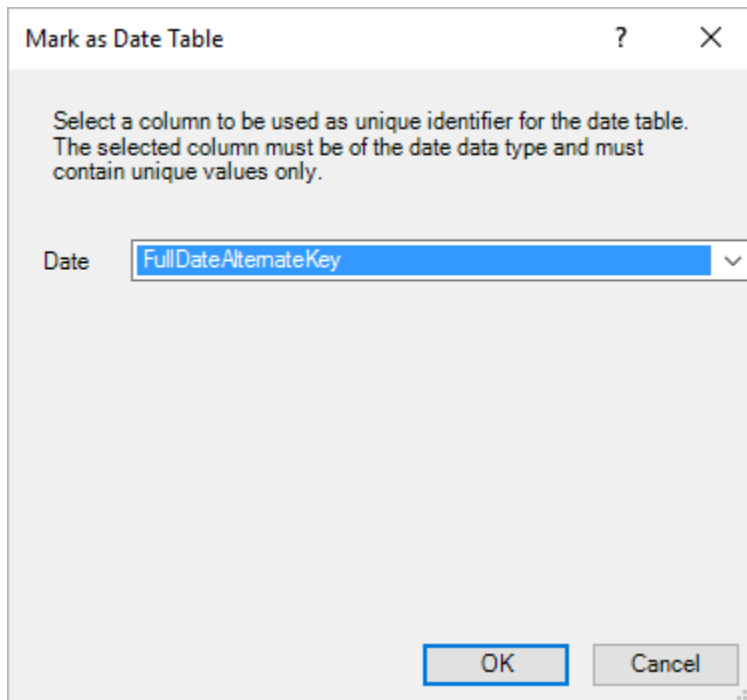
Reason for this behavior is that TotalYTD is a DAX time intelligence function and for that to work correctly I have to mark a table as a Date Table in my model. There are some requirements for the table to be marked as date table;

7. it should have one record per each individual date
8. there should be no gap in dates for the range of dates in your data set
9. there should be a full date column (with date data type) in the date table.

fortunately DimDate in our example meets all criteria above, so I'll go to model and under DimDate, under the Design tab, I choose Mark as Date Table;



There will be a dialog box asking for the full date column in your table, which in our case is FullDateAlternateKey (the column with date data type)



After this change now I can see that Sales YTD shows me the correct value (sales amounts adds up for each year):

Row Labels	Sum of SalesAmount	Sales YTD
2005	\$3,266,373.66	\$3,266,373.66
July	\$473,388.16	\$473,388.16
August	\$506,191.69	\$979,579.85
September	\$473,943.03	\$1,453,522.89
October	\$513,329.47	\$1,966,852.36
November	\$543,993.41	\$2,510,845.77
December	\$755,527.89	\$3,266,373.66
2006	\$6,530,343.53	\$6,530,343.53
January	\$596,746.56	\$596,746.56
February	\$550,816.69	\$1,147,563.25
March	\$644,135.20	\$1,791,698.45
April	\$663,692.29	\$2,455,390.74
May	\$673,556.20	\$3,128,946.94
June	\$676,763.65	\$3,805,710.59
July	\$500,365.16	\$4,306,075.74
August	\$546,001.47	\$4,852,077.21
September	\$350,466.99	\$5,202,544.20
October	\$415,390.23	\$5,617,934.44
November	\$335,095.09	\$5,953,029.53
December	\$577,314.00	\$6,530,343.53
2007	\$9,791,060.30	\$9,791,060.30
January	\$438,865.17	\$438,865.17
February	\$489,090.34	\$927,955.51
March	\$485,574.79	\$1,413,530.30
April	\$506,399.27	\$1,919,929.57
May	\$562,772.56	\$2,482,702.13

So as you see it is easy to get time intelligence functions in Power Pivot working by only marking a table as a date table. Things aren't that easy in Power BI unfortunately.

Time Intelligence Functions Issue in Power BI

There is not an option for marking a table as date table in Power BI (at least at the time of writing this post). So if I create same model in Power BI with same relationship, and same measure I'll see it won't work correctly

CalendarYear	EnglishMonthName ▲	SalesAmount	Sales YTD Not Correct
2005	July	\$473,388.16	\$473,388.16
	August	\$506,191.69	\$506,191.69
	September	\$473,943.03	\$473,943.03
	October	\$513,329.47	\$513,329.47
	November	\$543,993.41	\$543,993.41
	December	\$755,527.89	\$755,527.89
	Total	\$3,266,373.66	\$3,266,373.66
2006	January	\$596,746.56	\$596,746.56
	February	\$550,816.69	\$550,816.69
	March	\$644,135.20	\$644,135.20
	April	\$663,692.29	\$663,692.29
	May	\$673,556.20	\$673,556.20
	June	\$676,763.65	\$676,763.65
	July	\$500,365.16	\$500,365.16
	August	\$546,001.47	\$546,001.47
	September	\$350,466.99	\$350,466.99
	October	\$415,390.23	\$415,390.23
	November	\$335,095.09	\$335,095.09
	December	\$577,314.00	\$577,314.00
	Total	\$6,530,343.53	\$6,530,343.53
2007	January	\$438,865.17	\$438,865.17
	February	\$489,090.34	\$489,090.34
	March	\$485,574.79	\$485,574.79
	April	\$506,399.27	\$506,399.27

Everything for building above model is same as what I've explained in Power Pivot. the only difference is that I've used Matrix from Power BI visualization to show report above.

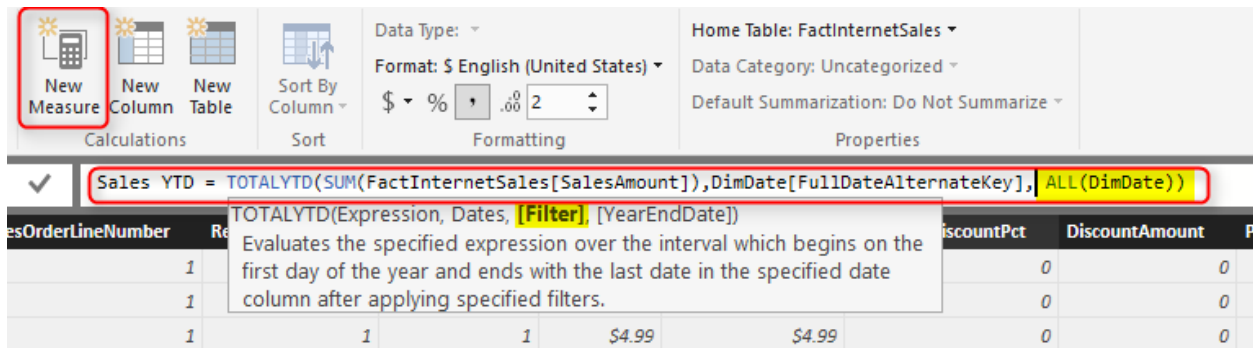
I've found two ways to fix the issue so far;

4. adding ALL(<date table>)
5. Importing date table from Power Pivot

Adding ALL(<Date Table>)

I can all an ALL(<date table name>) filter to the time intelligence function to get it working right. Here is the correct version of year to date calculation in Power BI;

Sales YTD = TOTALYTD(SUM(FactInternetSales[SalesAmount]),DimDate[FullDateAlternateKey],
ALL(DimDate))



New Measure | **Calculations** | **Sort** | **Formatting** | **Properties**

Data Type: **\$ English (United States)** | Home Table: **FactInternetSales** | Data Category: **Uncategorized** | Default Summarization: **Do Not Summarize**

Sales YTD = TOTALYTD(SUM(FactInternetSales[SalesAmount]),DimDate[FullDateAlternateKey], ALL(DimDate))

TOTALYTD(Expression, Dates, [Filter], [YearEndDate])
Evaluates the specified expression over the interval which begins on the first day of the year and ends with the last date in the specified date column after applying specified filters.

esOrderLineNumber	Re	DiscountPct	DiscountAmount	P
1		0	0	
1		0	0	
1		0	0	

Two important points here:

5. You should create year to date as a New Measure not column.
6. use ALL(DimDate) in the Filter section of the function.

So now as a result I have correct year to date:

CalendarYear	EnglishMonthName ▲	SalesAmount	Sales YTD
2005	July	\$473,388.16	\$473,388.16
	August	\$506,191.69	\$979,579.85
	September	\$473,943.03	\$1,453,522.89
	October	\$513,329.47	\$1,966,852.36
	November	\$543,993.41	\$2,510,845.77
	December	\$755,527.89	\$3,266,373.66
	Total	\$3,266,373.66	\$3,266,373.66
2006	January	\$596,746.56	\$596,746.56
	February	\$550,816.69	\$1,147,563.25
	March	\$644,135.20	\$1,791,698.45
	April	\$663,692.29	\$2,455,390.74
	May	\$673,556.20	\$3,128,946.94
	June	\$676,763.65	\$3,805,710.59
	July	\$500,365.16	\$4,306,075.74
	August	\$546,001.47	\$4,852,077.21
	September	\$350,466.99	\$5,202,544.20
	October	\$415,390.23	\$5,617,934.44
	November	\$335,095.09	\$5,953,029.53
	December	\$577,314.00	\$6,530,343.53
	Total	\$6,530,343.53	\$6,530,343.53
2007	January	\$438,865.17	\$438,865.17
	February	\$489,090.34	\$927,955.51
	March	\$485,574.79	\$1,413,530.30
	April	\$506,399.27	\$1,919,929.57
	May	\$562,772.56	\$2,482,702.13
	June	\$554,799.23	\$3,037,501.36
	July	\$886,668.84	\$3,924,170.20
	August	\$847,413.51	\$4,771,583.71

Let's test it with another function:

SamePeriodLastYear is a DAX function that returns the same period (date, month, quarter) from the last year. I want to use it to calculate last year's sale. If I do it without ALL I see this

Last Year Sales Incorrect =
CALCULATE(SUM(FactInternetSales[SalesAmount]),SAMEPERIODLASTYEAR(DimDate[FullDateAlternateKey])
)

CalendarYear	EnglishMonthName ▲	SalesAmount	Last Year Sales Incorrect
2005	July	\$473,388.16	
	August	\$506,191.69	
	September	\$473,943.03	
	October	\$513,329.47	
	November	\$543,993.41	
	December	\$755,527.89	
	Total	\$3,266,373.66	
2006	January	\$596,746.56	
	February	\$550,816.69	
	March	\$644,135.20	
	April	\$663,692.29	
	May	\$673,556.20	
	June	\$676,763.65	
	July	\$500,365.16	
	August	\$546,001.47	
	September	\$350,466.99	
	October	\$415,390.23	
	November	\$335,095.09	
	December	\$577,314.00	
	Total	\$6,530,343.53	
2007	January	\$438,865.17	
	February	\$489,090.34	
	March	\$485,574.79	
	April	\$506,399.27	
	May	\$562,772.56	
	June	\$554,799.23	
	July	\$886,668.84	
	August	\$847,413.51	
	September	\$1,010,258.13	
	October	\$1,080,449.58	
	November	\$1,196,981.11	
	December	\$1,731,787.77	
	Total	\$9,791,060.30	
2008	January	\$1,340,244.95	
	February	\$1,462,479.83	
	March	\$1,480,905.18	
	April	\$1,608,750.53	
	May	\$1,878,317.51	
	June	\$1,949,361.11	
	July	\$50,840.63	
	Total	\$9,770,899.74	
Total		\$29,358,677.22	\$29,358,677.220699999

This shows me nothing!

Now if I add ALL(DimDate) in the filter area of SamePeriodLastYear as below:

Last Year Sales =

CALCULATE(SUM(FactInternetSales[SalesAmount]),SAMEPERIODLASTYEAR(DimDate[FullDateAlternateKey]),ALL(DimDate))

Last Year Sales = CALCULATE(SUM(FactInternetSales[SalesAmount]),SAMEPERIODLASTYEAR(DimDate[FullDateAlternateKey]),ALL(DimDate))									
CALCULATE(Expression, [Filter1], [Filter2], ...)									
Evaluates an expression in a context modified by filters.									
LineNumber	RevisionNumber	ProductID	ProductDescription	UnitPrice	DiscountPct	DiscountAmount	ProductStandardCost	TotalDue	TotalDue
1	1	1	1	\$4.99	0	0	\$1.8663	\$1.8663	\$1.8663
1	1	1	1	\$4.99	0	0	\$1.8663	\$1.8663	\$1.8663
1	1	1	1	\$4.99	0	0	\$1.8663	\$1.8663	\$1.8663

And the result would be correct:

CalendarYear	EnglishMonthName ▲	SalesAmount	Last Year Sales
2005	July	\$473,388.16	
	August	\$506,191.69	
	September	\$473,943.03	
	October	\$513,329.47	
	November	\$543,993.41	
	December	\$755,527.89	
	Total	\$3,266,373.66	
2006	January	\$596,746.56	
	February	\$550,816.69	
	March	\$644,135.20	
	April	\$663,692.29	
	May	\$673,556.20	
	June	\$676,763.65	
	July	\$500,365.16	\$473,388.16
	August	\$546,001.47	\$506,191.69
	September	\$350,466.99	\$473,943.03
	October	\$415,390.23	\$513,329.47
	November	\$335,095.09	\$543,993.41
	December	\$577,314.00	\$755,527.89
	Total	\$6,530,343.53	\$3,266,373.66
2007	January	\$438,865.17	\$596,746.56
	February	\$489,090.34	\$550,816.69
	March	\$485,574.79	\$644,135.20
	April	\$506,399.27	\$663,692.29
	May	\$562,772.56	\$673,556.20
	June	\$554,799.23	\$676,763.65
	July	\$886,668.84	\$500,365.16
	August	\$847,413.51	\$546,001.47
	September	\$1,010,258.13	\$350,466.99
	October	\$1,080,449.58	\$415,390.23
	November	\$1,196,981.11	\$335,095.09
	December	\$1,731,787.77	\$577,314.00
	Total	\$9,791,060.30	\$6,530,343.53
2008	January	\$1,340,244.95	\$438,865.17
	February	\$1,462,479.83	\$489,090.34
	March	\$1,480,905.18	\$485,574.79
	April	\$1,608,750.53	\$506,399.27
	May	\$1,878,317.51	\$562,772.56
	June	\$1,949,361.11	\$554,799.23
	July	\$50,840.63	\$886,668.84
	August		\$847,413.51
	September		\$1,010,258.13
	October		\$1,080,449.58
	November		\$1,196,981.11

Even If I do more complex time intelligence functions such as Rolling 12 month ([more information on Alberto's post here](#)), I can add ALL(DimDate) to get it working correctly:

Sales Rolling 12 Months =

```
CALCULATE(
    SUM(FactInternetSales[SalesAmount]),
    DATESBETWEEN
        (DimDate[FullDateAlternateKey],
            NEXTDAY(SAMEPERIODLASTYEAR(LASTDATE(DimDate[FullDateAlternateKey]))),
            LASTDATE(DimDate[FullDateAlternateKey])
        ),
    ALL(DimDate)
)
```

CalendarYear	EnglishMonthName ▲	SalesAmount	Sales Rolling 12 Months
2005	July	\$473,388.16	\$473,388.16
	August	\$506,191.69	\$979,579.85
	September	\$473,943.03	\$1,453,522.89
	October	\$513,329.47	\$1,966,852.36
	November	\$543,993.41	\$2,510,845.77
	December	\$755,527.89	\$3,266,373.66
	Total	\$3,266,373.66	\$3,266,373.66
2006	January	\$596,746.56	\$3,863,120.21
	February	\$550,816.69	\$4,413,936.91
	March	\$644,135.20	\$5,058,072.11
	April	\$663,692.29	\$5,721,764.40
	May	\$673,556.20	\$6,395,320.59
	June	\$676,763.65	\$7,072,084.24
	July	\$500,365.16	\$7,099,061.24
	August	\$546,001.47	\$7,138,871.02
	September	\$350,466.99	\$7,015,394.98
	October	\$415,390.23	\$6,917,455.73
	November	\$335,095.09	\$6,708,557.42
	December	\$577,314.00	\$6,530,343.53
	Total	\$6,530,343.53	\$6,530,343.53
2007	January	\$438,865.17	\$6,372,462.14
	February	\$489,090.34	\$6,310,735.78
	March	\$485,574.79	\$6,152,175.37
	April	\$506,399.27	\$5,994,882.35
	May	\$562,772.56	\$5,884,098.72
	June	\$554,799.23	\$5,762,134.30
	July	\$886,668.84	\$6,148,437.98
	August	\$847,413.51	\$6,449,850.02
	September	\$1,010,258.13	\$7,109,641.16
	October	\$1,080,449.58	\$7,774,700.51
	November	\$1,196,981.11	\$8,636,586.53
	December	\$1,731,787.77	\$9,791,060.30
	Total	\$9,791,060.30	\$9,791,060.30
2008	January	\$1,340,244.95	\$10,692,440.08
	February	\$1,462,479.83	\$11,665,829.57
	March	\$1,480,905.18	\$12,661,159.96
	April	\$1,608,750.53	\$13,763,511.22

I haven't checked ALL(<date table>) with all time intelligence functions in Power BI to see if it works for all of them or not, but I believe it would work for most as most of them accept filter. However if you found somewhere that it doesn't work, try the second method as below;

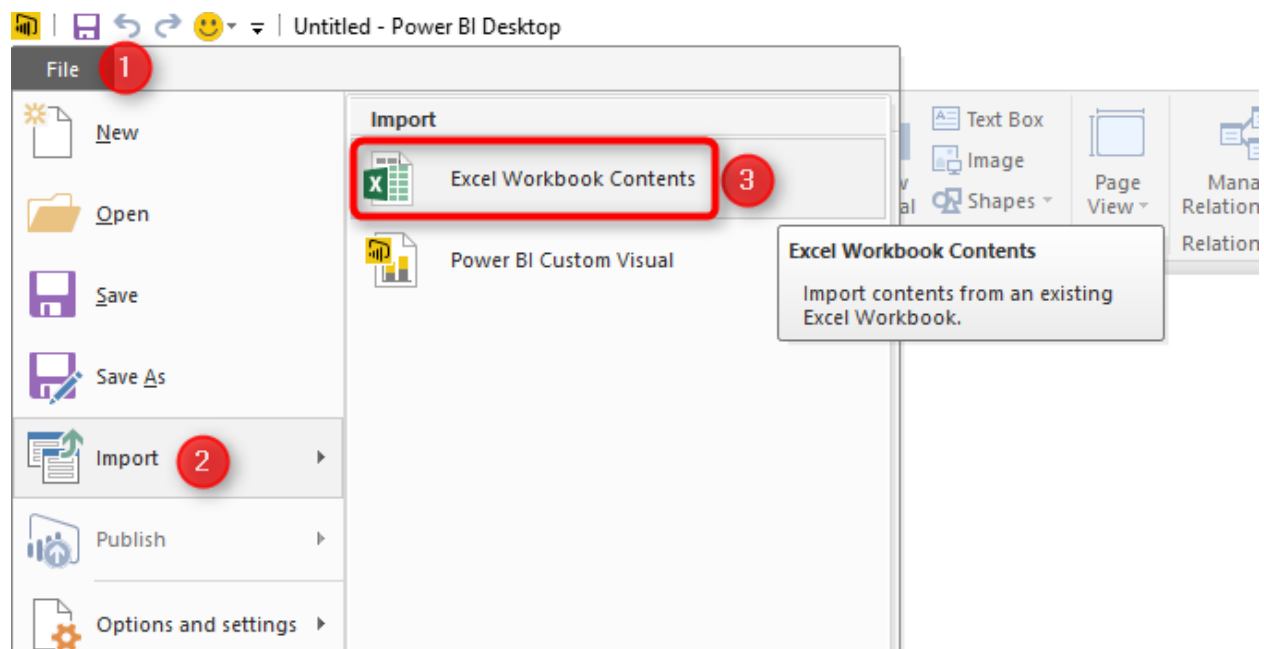
Import Date Table from Power Pivot

I saw this first on [Ginger Grant's blog post](#), so all credit goes to her. this method is simple, but works perfectly! The big warning for using this method is that if you want to apply this on an existing model you might need to re-create your model, because Power BI after the Import will re-write the solution. So let's see how it works;

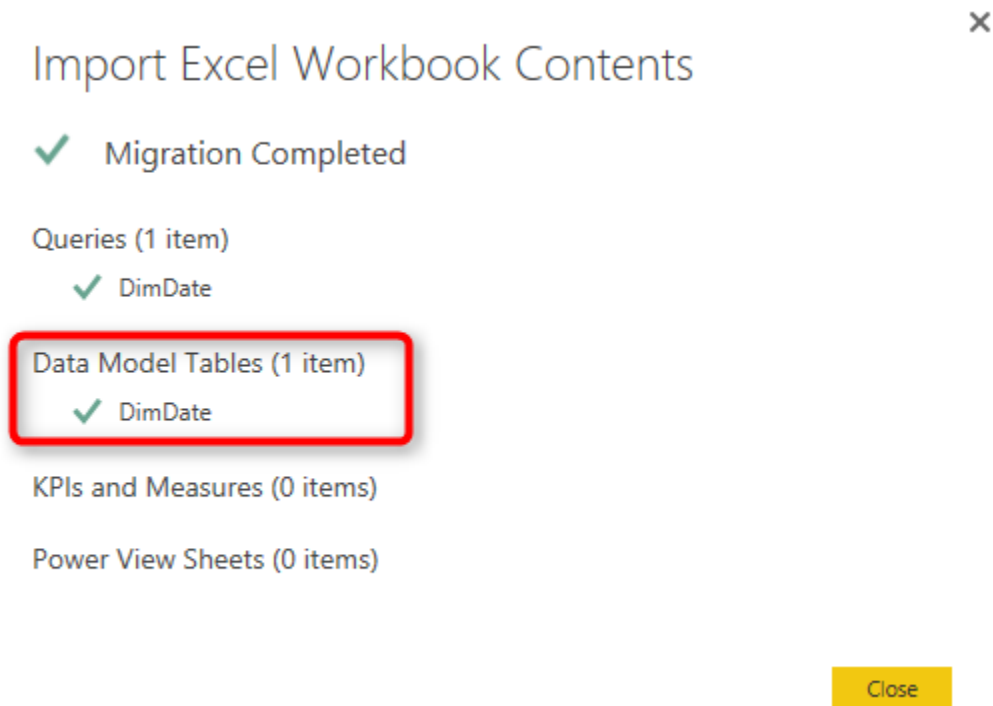
You've seen how I've created an Excel file with Power Pivot and the DimDate marked as date table earlier in this post. What you need to do is to do same but only for your date dimension;

Create an empty Excel file with a Power Pivot model with only single table: your date dimension. Mark this table as Date Table in Power Pivot, and then save the Excel file somewhere.

In Power BI **import** the excel file: Note that **Import** is different from Get Data. To Import; go to File menu, then Import, and then choose Excel Workbook Contents.



Address the Excel file here, and the DimDate table from Power Pivot will be loaded into the model;



Now you can add all other tables you want and write time intelligence functions normally. Don't worry about anything because you have the date table configured.

For my example I add FactInternetSales, and create the relationship between that table and DimDate, and write YTD calculation as a measure with this script:

```
Sales YTD = TOTALYTD(SUM(FactInternetSales[SalesAmount]),DimDate[FullDateAlternateKey])
```

And here is the result correctly as expected:

CalendarYear	EnglishMonthName	SalesAmount	Sales YTD
2005	July	\$473,388.16	\$473,388.16
	August	\$506,191.69	\$979,579.85
	September	\$473,943.03	\$1,453,522.89
	October	\$513,329.47	\$1,966,852.36
	November	\$543,993.41	\$2,510,845.77
	December	\$755,527.89	\$3,266,373.66
	Total	\$3,266,373.66	\$3,266,373.66
2006	January	\$596,746.56	\$596,746.56
	February	\$550,816.69	\$1,147,563.25
	March	\$644,135.20	\$1,791,698.45
	April	\$663,692.29	\$2,455,390.74
	May	\$673,556.20	\$3,128,946.94
	June	\$676,763.65	\$3,805,710.59
	July	\$500,365.16	\$4,306,075.74
	August	\$546,001.47	\$4,852,077.21
	September	\$350,466.99	\$5,202,544.20
	October	\$415,390.23	\$5,617,934.44
	November	\$335,095.09	\$5,953,029.53
	December	\$577,314.00	\$6,530,343.53
	Total	\$6,530,343.53	\$6,530,343.53
2007	January	\$438,865.17	\$438,865.17
	February	\$489,090.34	\$927,955.51
	March	\$485,574.79	\$1,413,530.30
	April	\$506,399.27	\$1,919,929.57
	May	\$562,772.56	\$2,482,702.13
	June	\$554,799.23	\$3,037,501.36
	July	\$886,668.84	\$3,924,170.20
	August	\$847,413.51	\$4,771,583.71
	September	\$1,010,258.13	\$5,781,841.84
	October	\$1,080,449.58	\$6,862,291.42
	November	\$1,196,981.11	\$8,059,272.53
	December	\$1,731,787.77	\$9,791,060.30
	Total	\$9,791,060.30	\$9,791,060.30
2008	January	\$1,340,244.95	\$1,340,244.95
	February	\$1,462,479.83	\$2,802,724.78

As I've mentioned earlier the limitation (or difficulty) with this method is that you have to do your date table first and then rest of the model afterwards. If you have an existing model in Power BI then with Import your model have to be re-created which is a pain in most of the cases.

I'll update this post again if I found other methods.

Lost Customers DAX Calculation for Power BI

Published Date: November 22, 2016

FullName	Total Revenue	Last Period Revenue	Lost Customers	New Customers
Aaron Adams	\$118	\$118	0	1
Aaron Alexander	\$70	\$70	0	1
Aaron Allen	\$3,400		1	0
Aaron Baker	\$1,751	\$1,751	0	1
Aaron Bryant	\$134	\$134	0	1
Aaron Butler	\$15	\$15	0	1
Aaron Campbell	\$1,155	\$1,155	0	1
Aaron Carter	\$40	\$40	0	1
Aaron Chen	\$40	\$40	0	1
Aaron Coleman	\$62	\$62	0	1
Aaron Collins	\$6,047	\$2,469	0	0
Aaron Diaz	\$6,030	\$2,451	0	0
Aaron Edwards	\$94	\$94	0	1
Aaron Evans	\$2,433	\$2,433	0	1

Period

☐ 30

☐ 60

☐ 120

☐ 180

☐ 365

☒ 1461

☐ 2922

1461

Selected Period

EnglishProductName

☐ Adjustable Race

☐ All-Purpose Bike Stand

☐ AWC Logo Cap

☐ BB Ball Bearing

One of the sample scenarios that DAX can be used as a calculation engine is customer retention. In the area of customer retention businesses might be interested to see who there lost customers or new customers are in the specific period. Obviously this can be calculated in the Power Query or in the data source level, however if we want to keep the period dynamic then DAX is the great asset to do it. In this post I'll explain how you can write a simple DAX calculation to find new customers and lost customers over a dynamic period of time in Power BI. If you are interested to learn more about Power BI read the [Power BI online book; from Rookie to Rock Star](#).

Defining Terms

Before start let's define terms that we use here. There might be different definitions for Lost or New Customers, the definition that I use is the very simple definition that I have seen some business use to easily measure their customer retention.

New Customer(s): Any customer who purchased any product in the last specified period of time considering the fact that the customer didn't purchased anything from this business before this period.

Lost Customer(s): Any customer who haven't purchased any product in the last specified period of time considering the fact that he/she previously purchased from this business.

The method explained below through an example.

Prerequisite

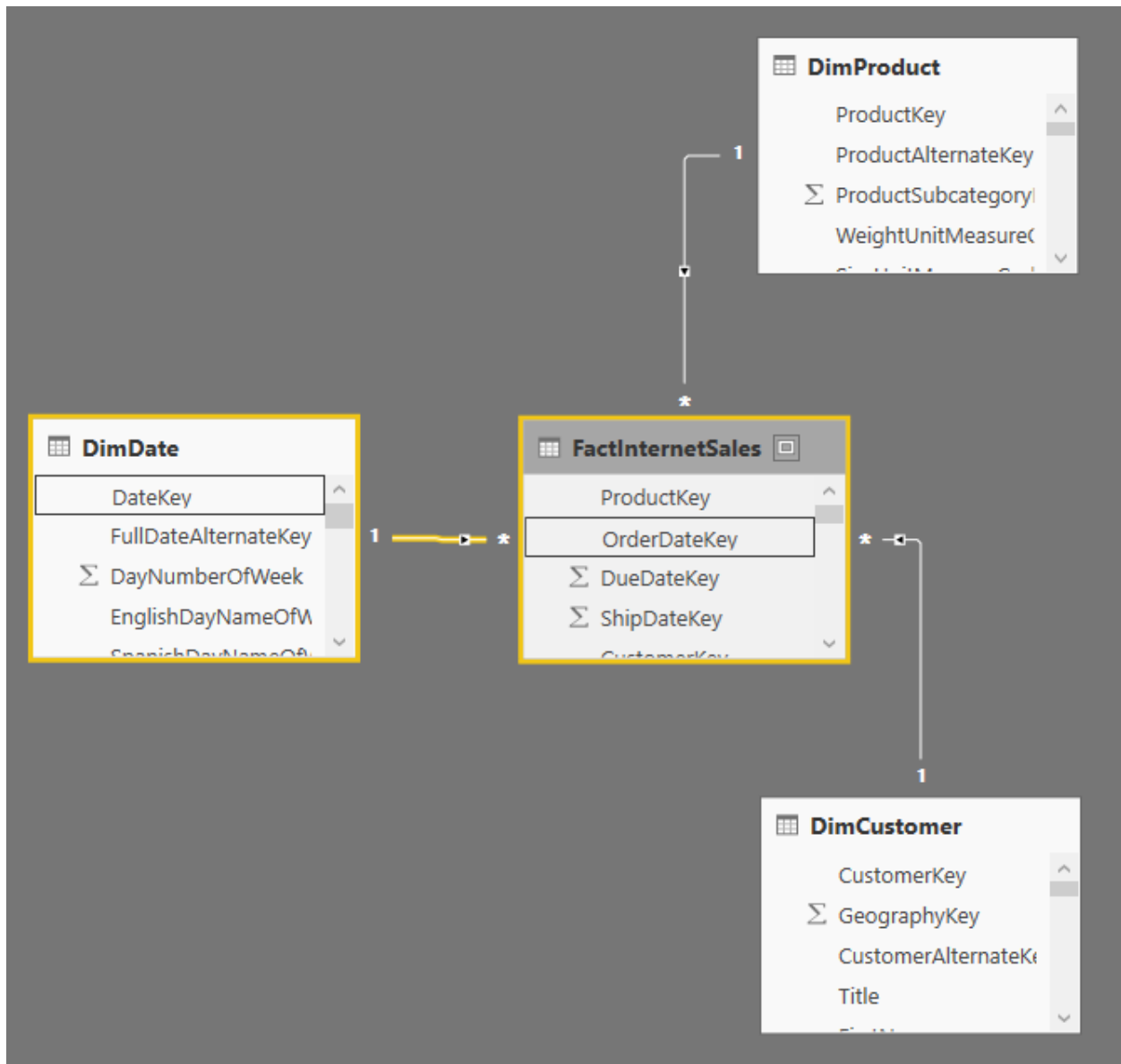
For running this example you need to have AdventureWorksDW database or the Excel version of it.

Scenario

In the AdventureWorksDW, we want to find out who our new customers and also lost customers are. We want to have the flexibility to select the period (that determines if customer is lost or new) in the slicer. We need these tables in AdventureWorksDW; FactInternetSales, DimDate, DimCustomer, and DimProduct.

The Model

Let's start by getting data from AdventureWorksDW SQL Server database. Tables that we need for this model are: FactInternetSales, DimDate, DimCustomer, and DimProduct. You don't need to apply any Power Query transformation. Just load the tables in the Power BI. and check the relationship to be as below. Note that relationship between DimDate and FactInternetSales is only one active relationship based on OrderDateKey (remove other relationships between these two tables).



Period Slicer

Also we need to have a Period Slicer. This is for being able to dynamically select the period (for example last 30 days, or last 365 days). You can create this table any where, in SQL Server db. in Power Query, in Excel.... I have created that in my model itself through Enter Data. Here is how the table looks like;

1 ² 3	Period
1	30
2	60
3	120
4	180
5	365
6	1461
7	2922

Table Name is Period, and the column name is also Period.

Method

What I do for the calculation is to calculate the total revenue from each customer first. This is the total revenue regardless of period. Means sum of SalesAmount for everything that this customer bought for all times (I call this Total Revenue). Then I create another measure to calculate sum of SalesAmount only for the selected period (in Days), Let's call this Last Period Revenue. Once we have these calculations finding new or lost costumers are easy as below;

Lost Customer(s): Customers that their Total Revenue is greater than zero, but their Last Period Revenue is not.

New Customer(s): Customers that their Total Revenue is equal to their Last Period Revenue, and greater than zero.

The reason for greater than zero condition is that in the database we might have some customer records that haven't purchased anything at all (seems odd, but we have to consider it, as there might be some exceptions in some business). I do all of steps one by one in a separate measure, this is just to give you an understanding of the process. Obviously these all can be done in one measure.

To start the very first thing is to identify the selected value in the slicer through measure.

Measure for the Selected Value in the Slicer: Selected Period

First things first is to identify what item is selected in the Period slicer. DAX doesn't understand the SELECTED item, however it understand the CONTEXT in which the DAX expression evaluates. This means; if I select one or more values in the slicer, DAX expression will execute on top of that selection. This selection is the filter context of DAX expression. That's simple. Now for this particular slicer we want user to always select one

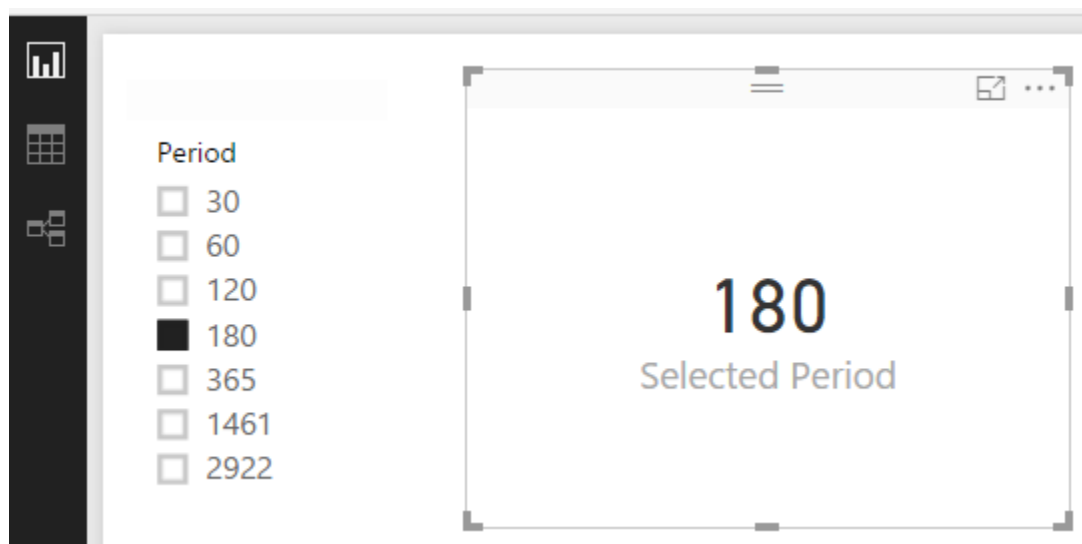
item, so it is also good to check if multiple items is selected or not. So let's create the **measure** (You can create a measure in the Data Tab, under Modeling, New Measure), named Selected Period under Period table with calculation below (Note that this should be a measure, not calculated column);

```
Selected Period = IF(COUNTROWS(Period)=1,
                    MAX(Period[Period]),
                    BLANK()
                    )
```

Expression above simply just check if number of records under the selected context (which will be selected slicer items) is one, then it will fetch the MAX of period. Why maximum when it is only one row? because DAX always works with a set of rows. even though we know it will be one row after passing this condition, DAX still works with a set of rows. the set of rows might have one or more rows. when it has only one row, we can use MIN or MAX or things like that to fetch it as a single value from the set of rows. And finally when multiple items are selected it will return blank.

Let's see how the measure works in action;

Create a sample report with going to Report tab, and create a slicer with Period. Also add a Card visualization with Selected Period measure (created above) as the value. Now if you select items in the slicer you will see what item is selected in the card visualization.



This Selected Period measure in the calculation of Sales Amount for selected period. Which we will do it in next steps.

Total Revenue

There are multiple ways of calculating the total revenue for each customer. I use a method that use a new Measure. create a new measure in Data tab, under DimCustomer and name it Total Revenue. write DAX expression below for it;

```
Total Revenue = SUMX(RELATEDTABLE(FactInternetSales),FactInternetSales[SalesAmount])
```

This DAX expression uses SumX function which will calculate sum of an expression (FactInternetSales[SalesAmount]) on a filtered table. The filtered table in this example is RelatedTable(FactInternetSales). RelatedTable will go through each Customer record in DimCustomer, and considering the existing relationship between DimCustomer and FactInternetSales in the model (As showed in the diagram earlier in this post), it will generate a sub set of FactInternetSales for each customer, which is only the subset that has this customerKey in it. So the result of sum of SalesAmount for this subset will be total revenue for each customer.

Let's see how this works in action;

Go back to the Report page, and create a Table with Full Name from DimCustomer, and also the new measure Total Revenue

FullName	Total Revenue
Aaron Adams	\$118
Aaron Alexander	\$70
Aaron Allen	\$3,400
Aaron Baker	\$1,751
Aaron Bryant	\$134
Aaron Butler	\$15
Aaron Campbell	\$1,155
Aaron Carter	\$40
Aaron Chen	\$40
Aaron Coleman	\$62
Aaron Collins	\$6,047
Aaron Diaz	\$6,030
Aaron Edwards	\$94
Aaron Evans	\$2,433
Aaron Flores	\$1,539
Aaron Foster	\$4,912
Aaron Gonzales	\$1,810
Aaron Gonzalez	\$133
Aaron Green	\$27
Aaron Griffin	\$72
Aaron Hall	\$29
Aaron Hayes	\$3,113
Aaron Henderson	\$27
Aaron Hernandez	\$94
Aaron Hill	\$36
Aaron Hughes	\$4,456
Aaron Jai	\$575
Aaron Jenkins	\$120
Aaron King	\$4,758
Aaron Kumar	\$2,049
Aaron Lal	\$2,310
Aaron Li	\$2,170
Total	\$29,358,677

In front of each customer you can see the total revenue for that customer. Now let's calculate selected period's revenue.

Total Revenue for the Selected Period in the Slicer

Let's call this Last Period Revenue. In this last period revenue all we want to calculate is the total sales for each customer in the period of last X days, where X is coming from

the selection of period slicer. We can use few functions to help along the way. Here is the measure with full DAX expression. I'll explain details after;

```
Last Period Revenue = CALCULATE(
    SUM(FactInternetSales[SalesAmount]),
    DATESBETWEEN(
        DimDate[FullDateAlternateKey],
        DATEADD(LASTDATE(DimDate[FullDateAlternateKey]),-1*[Selected Period],DAY),
        LASTDATE(DimDate[FullDateAlternateKey])
    )
)
```

Let's Start from Calculate; Calculation function is a simple function in terms of the basics of how it works, and it is also the motherhood of all functions in DAX, there are heaps of calculations you can do with this function. How this works? Calculate simply calculates an expression (first parameter) on a set of filters (second parameter, third parameter....). In the example above the expression is Sum(FactInternetSales[SalesAmount]), and the filter is all the parameters after that. In above expression you can see that we have only one filter which is DatesBetween. So calculate simply resolve sum of SalesAmount for the subset of data that comes out of DatesBetween function. Let's see what DatesBetween Does;

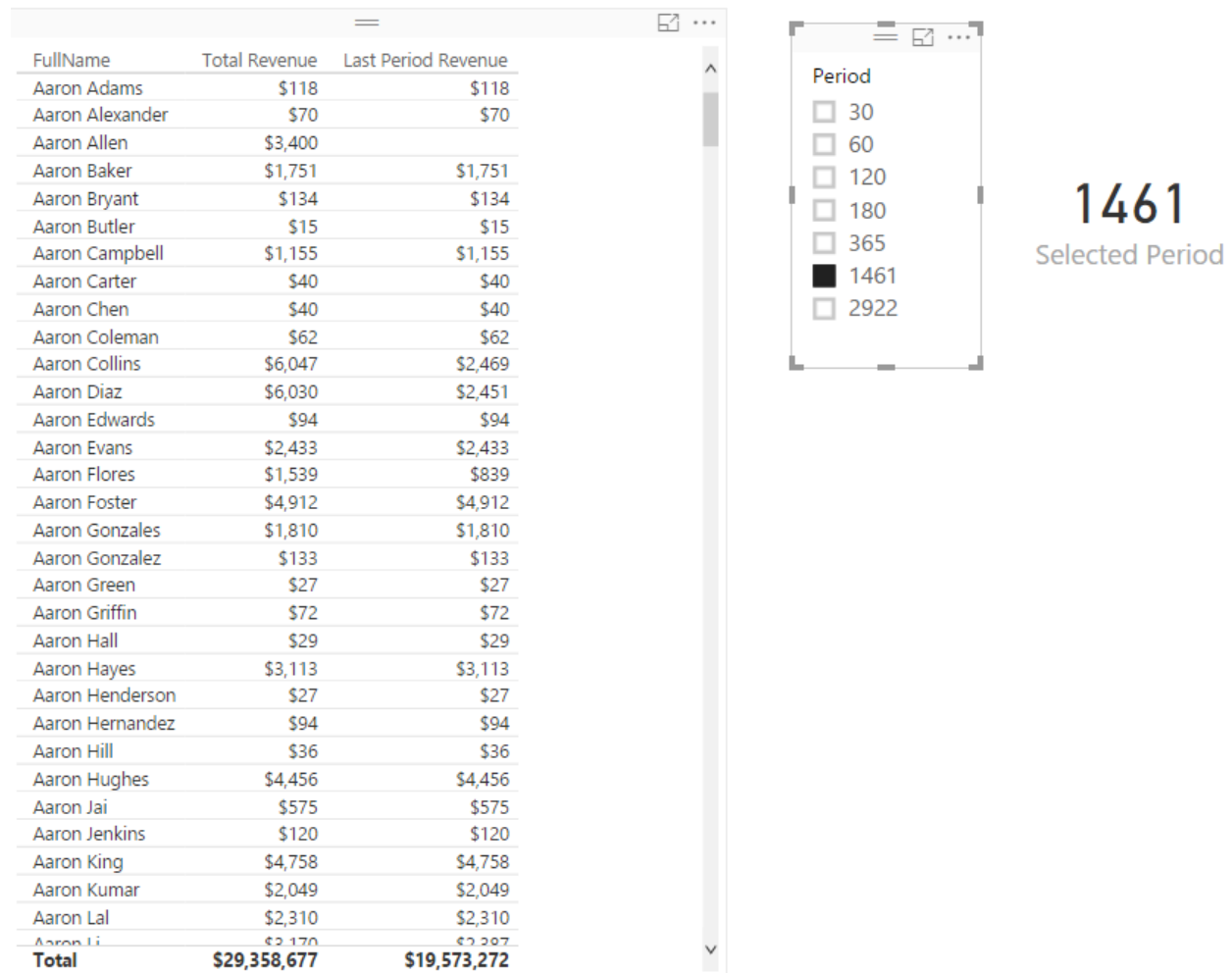
DatesBetween is self explanatory function. It will return a subset of data for the dates between start date and end date! It has three parameters; date field, start date, and end date;

```
DATESBETWEEN(
    DimDate[FullDateAlternateKey],
    DATEADD(LASTDATE(DimDate[FullDateAlternateKey]),-1*[Selected Period],DAY),
    LASTDATE(DimDate[FullDateAlternateKey])
)
```

In example above end date is LASTDATE(DimDate[FullDateAlternateKey]). This means the maximum date that we have considering what filter context we are in. The main important thing is calculating the start date. Start date is the last date minus selected period. For doing this calculation I used DATEADD function which reduce number of days with (-1*[Selected Period]) in the highlighted line above in the expression.

As a result this whole DAX expression returns total sales amount only for the selected period. let's see how it works in the report;

Go back to the report and add the new measure to table.



Last Period Revenue by default when no period is selected in the slicer will be blank. When you select values, this will bring total revenue for that number of days from the last date in DimDate. In my sample data set I can only start something if I select last 1461 days which is last 4 years. In different version of AdventureWorksDW you might get different results because their data is different. Now let's find what customers are lost.

Lost Customer(s) for the Selected Period

What table above says is that for a selected period, some customers have not purchased anything. For example Aaron Allen has the total revenue of \$3,400 but in the last 1461 days he didn't purchased anything. So what this says is that this customer is lost in the selected period! So Lost Customer(s) calculation is as simple as this:

Lost Customers = IF([Total Revenue]>0 && [Last Period Revenue]<=0
,1
,0)

This is a simple IF expression which says if customer has a Total Revenue greater than zero, and did not purchased anything in the last period (Last Period Revenue is less than or equal to zero), then this customer is lost.

New Customer(s) for the Selected Period

New customers on the other hand are customers who only purchased in the last period. This means their Total Revenue is equal to Last Period Revenue, and greater than zero. Here is the calculation;

New Customers = IF([Total Revenue]>0 && [Last Period Revenue]=[Total Revenue]
,1
,0)

Let's see the final result in the report; Add Lost Customers and New Customers to the table and you will see the result;

FullName	Total Revenue	Last Period Revenue	Lost Customers	New Customers
Aaron Adams	\$118	\$118	0	1
Aaron Alexander	\$70	\$70	0	1
Aaron Allen	\$3,400		1	0
Aaron Baker	\$1,751	\$1,751	0	1
Aaron Bryant	\$134	\$134	0	1
Aaron Butler	\$15	\$15	0	1
Aaron Campbell	\$1,155	\$1,155	0	1
Aaron Carter	\$40	\$40	0	1
Aaron Chen	\$40	\$40	0	1
Aaron Coleman	\$62	\$62	0	1
Aaron Collins	\$6,047	\$2,469	0	0
Aaron Díaz	\$6,030	\$2,451	0	0
Aaron Edwards	\$94	\$94	0	1
Aaron Evans	\$2,433	\$2,433	0	1
Aaron Flores	\$1,539	\$839	0	0
Aaron Foster	\$4,912	\$4,912	0	1
Aaron Gonzales	\$1,810	\$1,810	0	1
Aaron Gonzalez	\$133	\$133	0	1
Aaron Green	\$27	\$27	0	1
Aaron Griffin	\$72	\$72	0	1
Aaron Hall	\$29	\$29	0	1
Aaron Hayes	\$3,113	\$3,113	0	1
Aaron Henderson	\$27	\$27	0	1
Total	\$29,358,677	\$19,573,272	0	0

As you can see our formula successfully determined that Aaron Allen is lost in the last 1461 days period (if you increase or decrease the period you will see different result). Also those customers which their total revenue is equal to last period revenue are considered as New Customers (that's why you see so many new customers in above table). Note that a customer might not fall in any of these categories. for example;

FullName ▼	Total Revenue	Last Period Revenue	Lost Customers	New Customers
Aaron Kumar	\$2,049	\$2,049	0	1
Aaron Lal	\$2,310	\$2,310	0	1
Aaron Li	\$3,170	\$2,387	0	0
Aaron McDonald	\$120	\$120	0	1
Aaron Mitchell	\$49	\$49	0	1
Aaron Nelson	\$2,882	\$810	0	0
Aaron Patterson	\$1,184	\$1,184	0	1

Aaron Li and Aaron Nelson are neither lost or new customers. They are customers who purchased before this period, and still purchased in the last period as well.

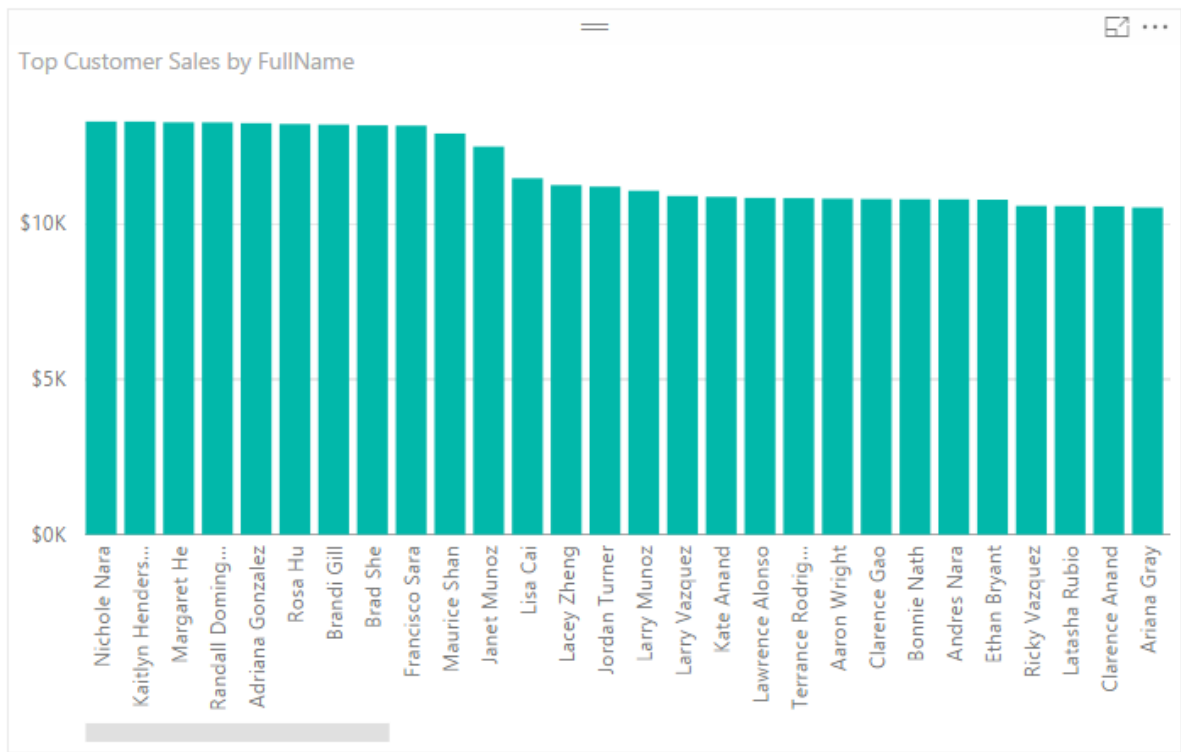
Bringing Other Dimensions

These measures also works if you bring another table which is related to other tables in the model as a new dimension. For example you can bring Product as another slicer and then you will see product by product which customers are lost and which are new customers. Hope this calculations helps you to expand customer retention in your Power BI Model.



Scenarios of Using Calculated Tables in Power BI

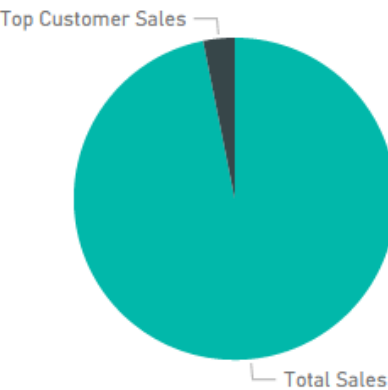
Published Date: April 1, 2016



Total Sales and Top Customer Sales

\$955.44K
Top Customer Sales

\$29.36M
Total Sales



Calculated tables first introduced in [September 2015 update of Power BI Desktop](#). The name speak for itself; these are tables created by calculation. As these are in-memory tables their calculation is based on DAX (Data Analysis eXpression language). There are many benefits of using Calculated tables, such as using them for role playing dimensions (for example having more than one date dimension in a model), There are some DAX functions and expressions that returns a table as a result, and using them as table in your model sometimes is really helpful. for example, you might want to create a table for top 10 customers and then use that as the main source table in many reports. In this post I'll explain to you some use cases of calculated tables.

Prerequisite

For running examples of this post, you need to install [Adventure Works DW](#) database on SQL Server, or you can download **Excel version** of it from here:

Enter Your Email to download the file (required)

[Download]

Role Playing Dimension

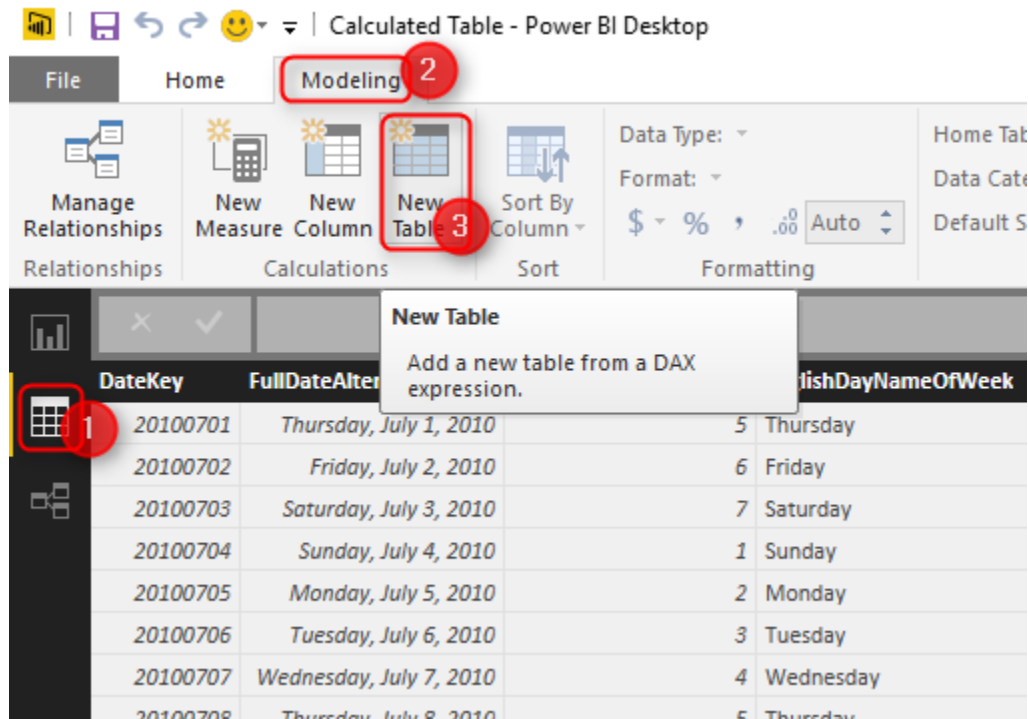
The very first functionality that appears in mind when we talk about Calculated Tables is ability to create role play dimensions. Role Play dimensions are dimensions with the same structure and data rows that plays different roles in our data model. For example Date Dimension is a generic dimension. However in a sales transaction table you might have more than one date columns to relate with the date dimension. in example below we have three date fields in FactInternetSales table: Order Date, Ship Date, and Due Date.

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey
528	20130128	20130209	20130204	14870	1	100	4
528	20130129	20130210	20130205	15319	1	100	4
528	20130131	20130212	20130207	16384	1	100	4
528	20130131	20130212	20130207	15476	1	100	4
528	20130201	20130213	20130208	15861	1	100	4
528	20130203	20130215	20130210	26017	1	100	4
528	20130203	20130215	20130210	14761	1	100	4
528	20130204	20130216	20130211	22038	1	100	4
528	20130204	20130216	20130211	22163	1	100	4
528	20130204	20130216	20130211	16018	1	100	4
528	20130205	20130217	20130212	25839	1	100	4

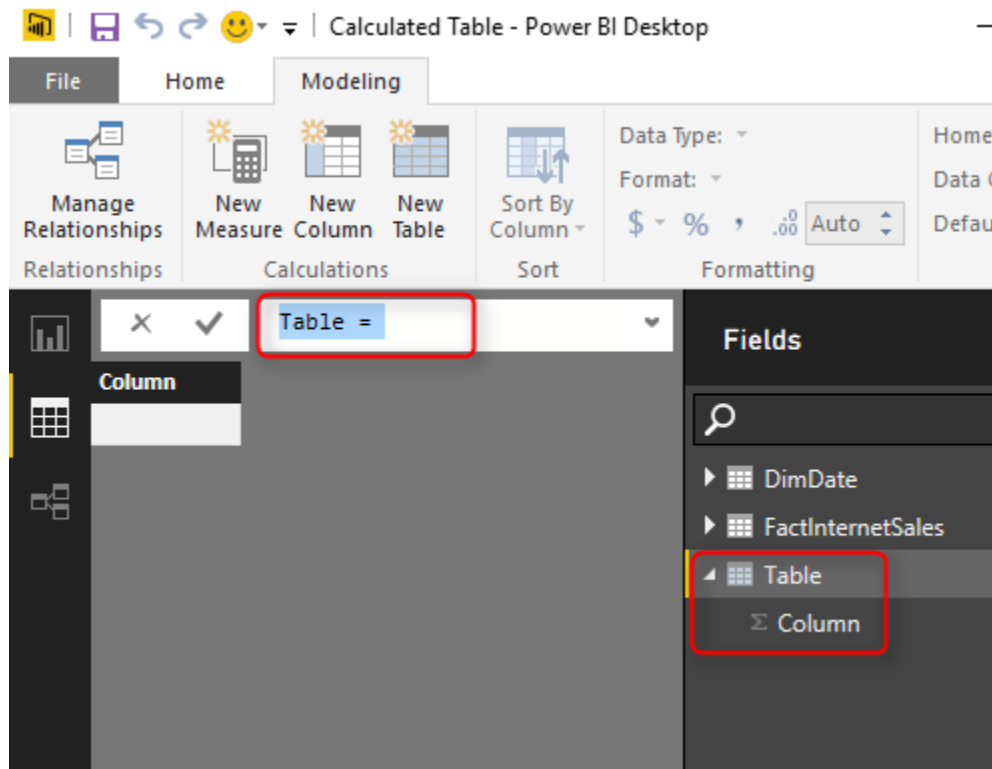
There three fields should be related to three different date dimensions (because tabular model which Power BI is based on that, doesn't support role playing dimensions built-in). So what you can do is to just load the date dimension once in Get Data section from the SQL Server ([Here is the T-SQL script for creating date dimension](#)), or by a blank query from Power Query script ([Here is the Power Query script for creating the date dimension](#)). Here is the example date dimension loaded in Power BI Desktop (through Get Data):

DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek
20100701	Thursday, July 1, 2010	5	Thursday	Jueves	Jeudi
20100702	Friday, July 2, 2010	6	Friday	Viernes	Vendredi
20100703	Saturday, July 3, 2010	7	Saturday	Sábado	Samedi
20100704	Sunday, July 4, 2010	1	Sunday	Domingo	Dimanche
20100705	Monday, July 5, 2010	2	Monday	Lunes	Lundi
20100706	Tuesday, July 6, 2010	3	Tuesday	Martes	Mardi
20100707	Wednesday, July 7, 2010	4	Wednesday	Miércoles	Mercredi
20100708	Thursday, July 8, 2010	5	Thursday	Jueves	Jeudi
20100709	Friday, July 9, 2010	6	Friday	Viernes	Vendredi
20100710	Saturday, July 10, 2010	7	Saturday	Sábado	Samedi
20100711	Sunday, July 11, 2010	1	Sunday	Domingo	Dimanche
20100712	Monday, July 12, 2010	2	Monday	Lunes	Lundi
20100713	Tuesday, July 13, 2010	3	Tuesday	Martes	Mardi
20100714	Wednesday, July 14, 2010	4	Wednesday	Miércoles	Mercredi
20100715	Thursday, July 15, 2010	5	Thursday	Jueves	Jeudi
20100716	Friday, July 16, 2010	6	Friday	Viernes	Vendredi
20100717	Saturday, July 17, 2010	7	Saturday	Sábado	Samedi
20100718	Sunday, July 18, 2010	1	Sunday	Domingo	Dimanche
20100719	Monday, July 19, 2010	2	Monday	Lunes	Lundi
20100720	Tuesday, July 20, 2010	3	Tuesday	Martes	Mardi
20100721	Wednesday, July 21, 2010	4	Wednesday	Miércoles	Mercredi

Now I can create role playing dimensions with creating a Calculated table:

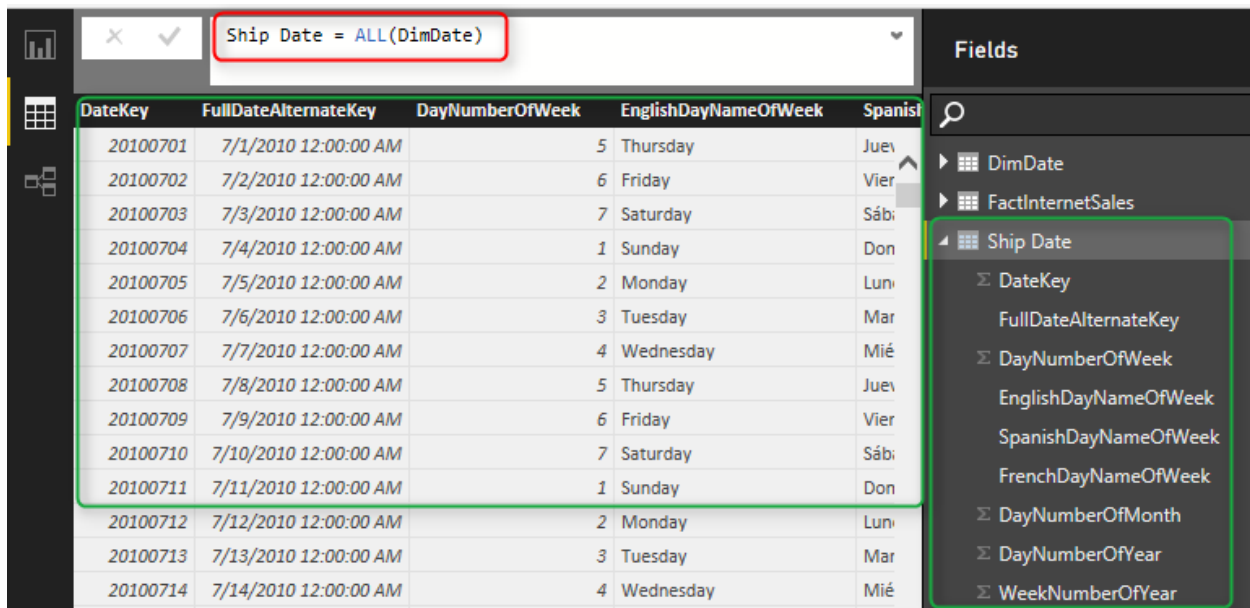


and this will create a table in memory for me and allows me to write definition of the table



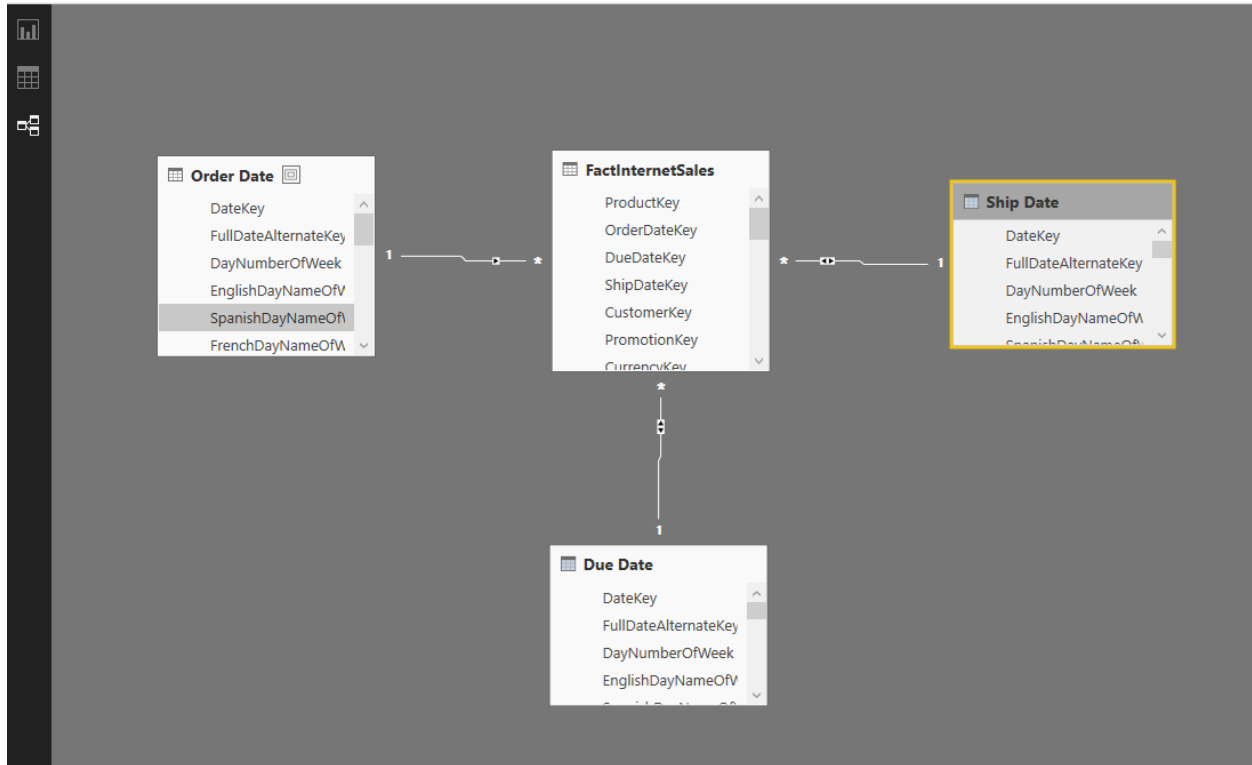
The language for table definition is DAX (Data Analysis eXpression). If you want to know more about DAX you have to read DAX section of the [Power BI book](#). For now let's keep it simple to see how it works in action. We want an exact copy of the DimDate table here, so I can simply use [ALL function in DAX](#) as below:

Ship Date = ALL(DimDate)



DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek
20100701	7/1/2010 12:00:00 AM	5	Thursday	Juev
20100702	7/2/2010 12:00:00 AM	6	Friday	Vier
20100703	7/3/2010 12:00:00 AM	7	Saturday	Sáb
20100704	7/4/2010 12:00:00 AM	1	Sunday	Don
20100705	7/5/2010 12:00:00 AM	2	Monday	Lun
20100706	7/6/2010 12:00:00 AM	3	Tuesday	Mar
20100707	7/7/2010 12:00:00 AM	4	Wednesday	Mié
20100708	7/8/2010 12:00:00 AM	5	Thursday	Juev
20100709	7/9/2010 12:00:00 AM	6	Friday	Vier
20100710	7/10/2010 12:00:00 AM	7	Saturday	Sáb
20100711	7/11/2010 12:00:00 AM	1	Sunday	Don
20100712	7/12/2010 12:00:00 AM	2	Monday	Lun
20100713	7/13/2010 12:00:00 AM	3	Tuesday	Mar
20100714	7/14/2010 12:00:00 AM	4	Wednesday	Mié

As soon as I type the expression above and press Enter I'll see the result underneath it as data rows, and also list of columns in the Fields pane. I've created my role dimension as simple as that. Now I can set up the relationship;

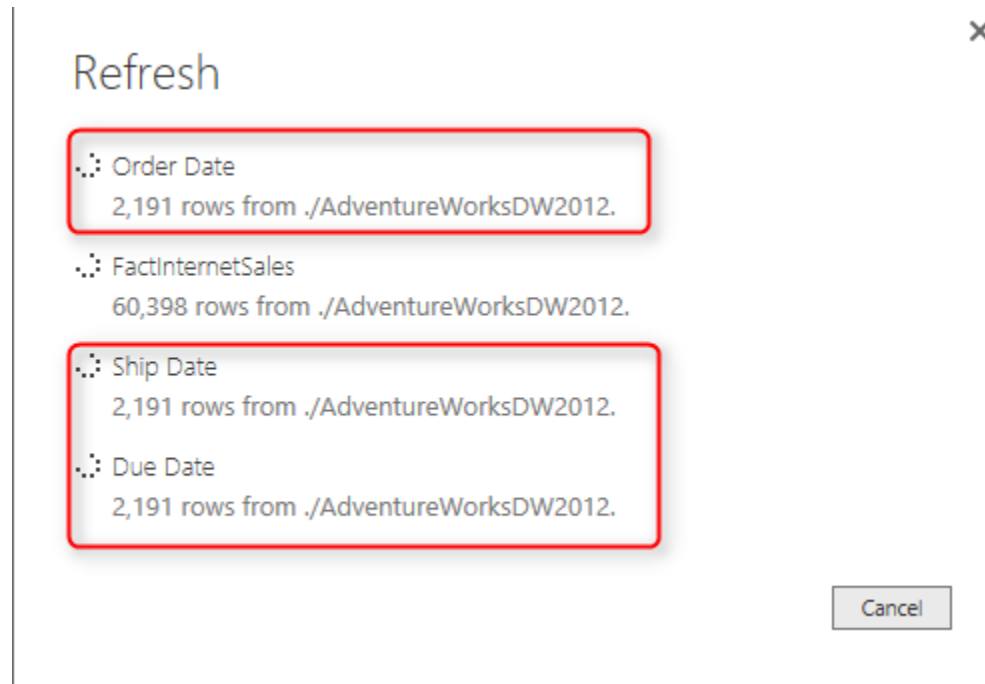


For the relationship above I've created also a Due Date dimension, and renamed the original DimDate to Order Date.

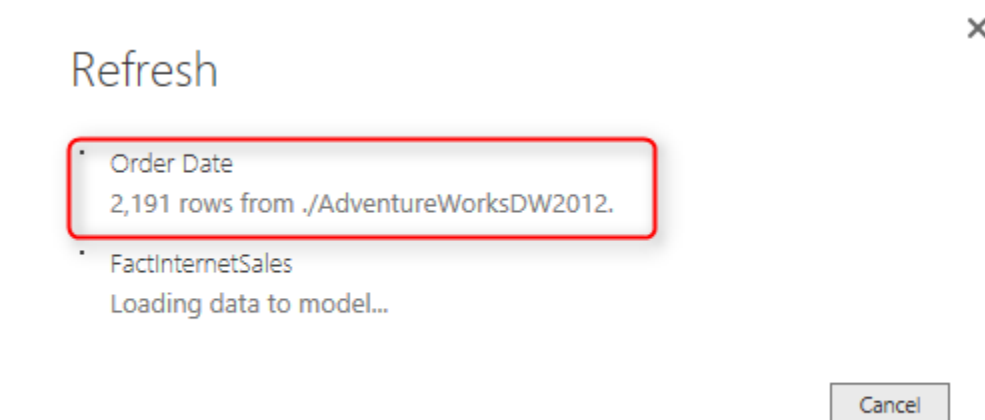
In-Memory Structure, Less Refresh Time

Calculated table loads into memory, so your Power BI file size will increase. However you don't need to read them again from the external data source. Yes, you can create multiple views in the source database and connect to them through Get Data section with Power Query, however their data need to populate from the source database every time a refresh happens (either scheduled or manual).

WITHOUT Calculated Tables: Here is example of three date tables loaded from the external data source:



WITH Calculated Tables: and here is only one date dimension loaded (for the role playing dimension example above):



As you can see this is much more efficient in terms of reducing the refresh time. However the memory consumption would be the same in both methods.

Date dimension was a narrow data set example. You might need role playing for big data tables, so Calculated tables will save you a lot of time in refreshing data in such case.

DAX Table Functions

There are some DAX functions that return a table. For example ALL function which I used in the role playing sample above. ALL was a simple example of a DAX function that returns the whole copy of the source table. Let's have a look at some other examples and see how it works in other scenarios.

Top 100 Customers as a Calculated Table

There are many examples that a business considers top 10 or top 20 customers and filter down whole dashboard and set of reports only for them. Usually the main reason is that top 10, 20 customers will bring majority of revenue to the business. Fortunately there is a [TOPN function in DAX](#) which helps us to build such calculation. TOPN function returns a table. With TOPN we can choose how many rows we want in the result set, and the grouping function to be applied (if there is any) and the aggregation (if there is any).

In this example I want to show you how to use calculated table to generate a list of top 100 customers. As a business requirement I want to visualize the total revenue from top 100 customers and compare it with total revenue of the whole business. There might be different ways to calculate and visualize it, but I want to do it with calculated table as a sample scenario.

Summarize

[Summarize](#) is a DAX function that generates a grouped by list from a table. Summarize works similar to Group By in T-SQL. So if I want to create a table with CustomerKeys and their total sales amount, I can write this expression:

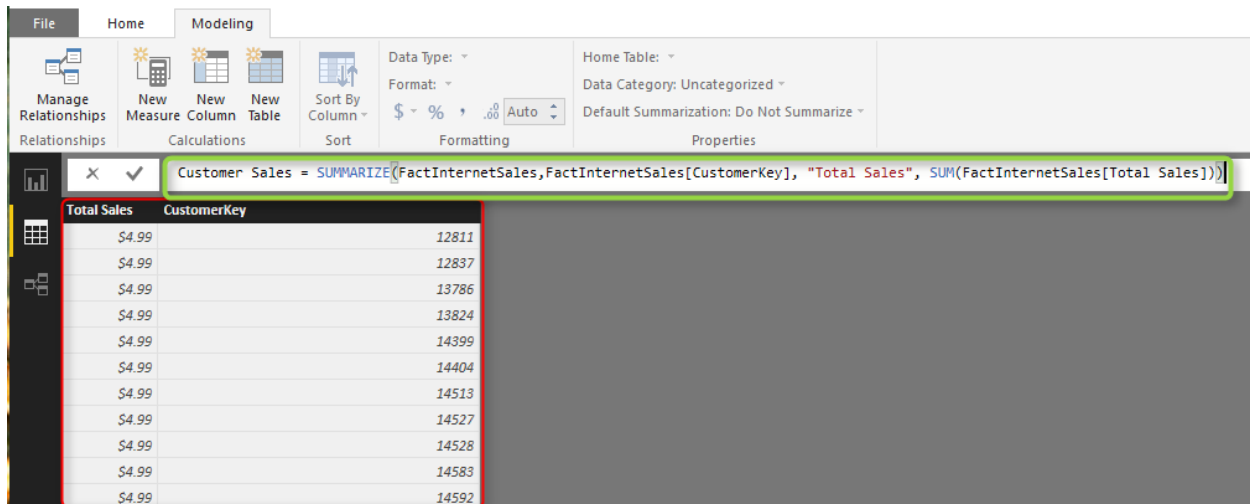
```
Customer Sales = SUMMARIZE(FactInternetSales,FactInternetSales[CustomerKey], "Total Sales",  
SUM(FactInternetSales[Total Sales]))
```

Here are details about parameters I passed in the expression above to Summarize function:

1. First parameter: Source Table. FactInternetSales is the source table that I want the group by (summarize) operation to be applied on it.
2. Second Parameter: Group by Column. CustomerKey in the FactInternetSales table is the column that I want to use as the key for grouping.
3. Third parameter: Output Column Name. I named the output calculated column name as Total Sales.

4. Forth parameter: Output Column Calculation. Here I write the calculation for the output column, which is simply sum of Total Sales Column.

So as a result I will have a table with CustomerKey and Total Sales.



TOPN

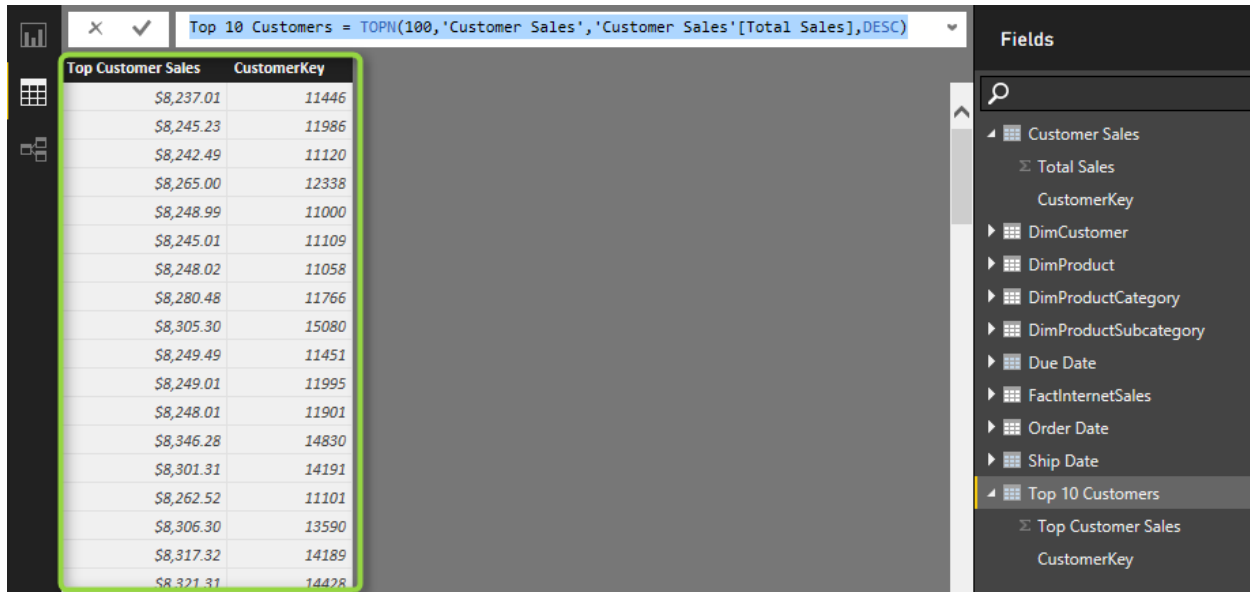
Now that we have list of customers with their total sales, it is easy to get top 100 customers. I can simply use TOPN function like this to create another calculated table (I could do this example with only one calculated table instead of two, but I only did it with two table to help you understand the logic better);

Top 10 Customers = TOPN(100,'Customer Sales','Customer Sales'[Total Sales],DESC)

And expression above means:

1. First parameter: Number of rows to return. 100 for top 100 customers.
2. Second parameter: Source Table. Customer Sales is the source of this operation which we want to fetch top 100 customers from it.
3. Third parameter: Order By Column. I want to order the table based on Total Sales of each Customer.
4. Forth parameter: Order By expression (ASC, or DESC). To get top 100 customers I have to order it by Total Sales DESC.

and here is the result:

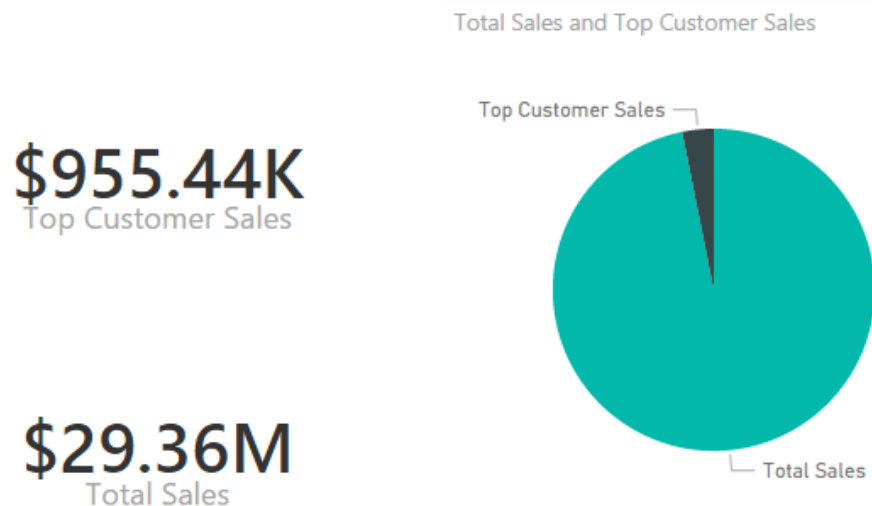
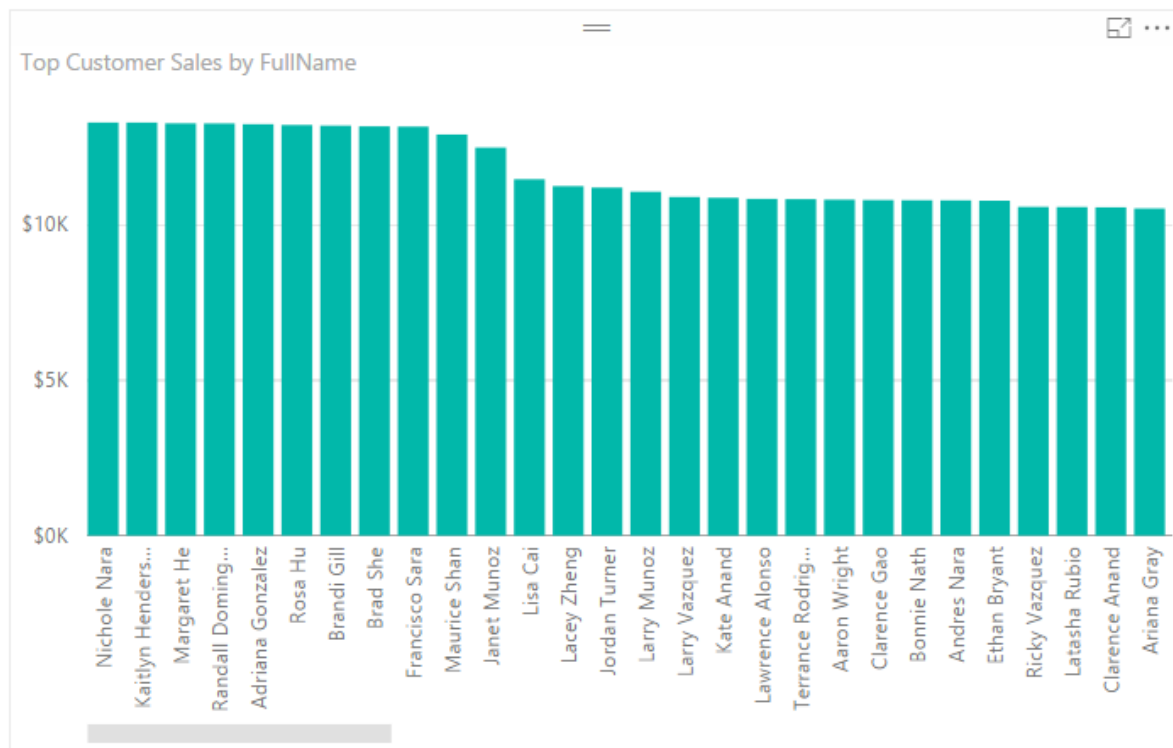


The screenshot shows the Power BI interface. At the top, a formula bar displays the DAX formula: `Top 10 Customers = TOPN(100, 'Customer Sales', 'Customer Sales'[Total Sales], DESC)`. Below the formula bar, a table titled 'Top Customer Sales' is displayed with two columns: 'Top Customer Sales' and 'CustomerKey'. The table contains 15 rows of data. To the right of the table, the 'Fields' pane is visible, showing a hierarchy of tables and columns. The 'Top 10 Customers' table is highlighted in the Fields pane.

Top Customer Sales	CustomerKey
\$8,237.01	11446
\$8,245.23	11986
\$8,242.49	11120
\$8,265.00	12338
\$8,248.99	11000
\$8,245.01	11109
\$8,248.02	11058
\$8,280.48	11766
\$8,305.30	15080
\$8,249.49	11451
\$8,249.01	11995
\$8,248.01	11901
\$8,346.28	14830
\$8,301.31	14191
\$8,262.52	11101
\$8,306.30	13590
\$8,317.32	14189
\$8,321.31	14428

I also renamed the Total Sales column to Top Customer Sales (as you see in screenshot above).

Now I can simply build a report in Power BI to show the difference between Total Sales and Top Customer Sales (If you don't know how to create visualization below follow the visualization chapters of [Power BI book](#)):



Great, We've used calculated tables for getting some insight out of top 100 customers and compared it with the total. There are many other cases that you can use Calculated Table for. [Chris Webb mentioned using Calendar function in his blog post](#) as a calculated table to start building a date dimension.

Limitations

As any other DAX related limitations, calculated tables very first limitation is memory. You need to have enough memory to use these tables. This limitation is also an advantage on the other hand, because in-memory structure makes these calculations really fast.

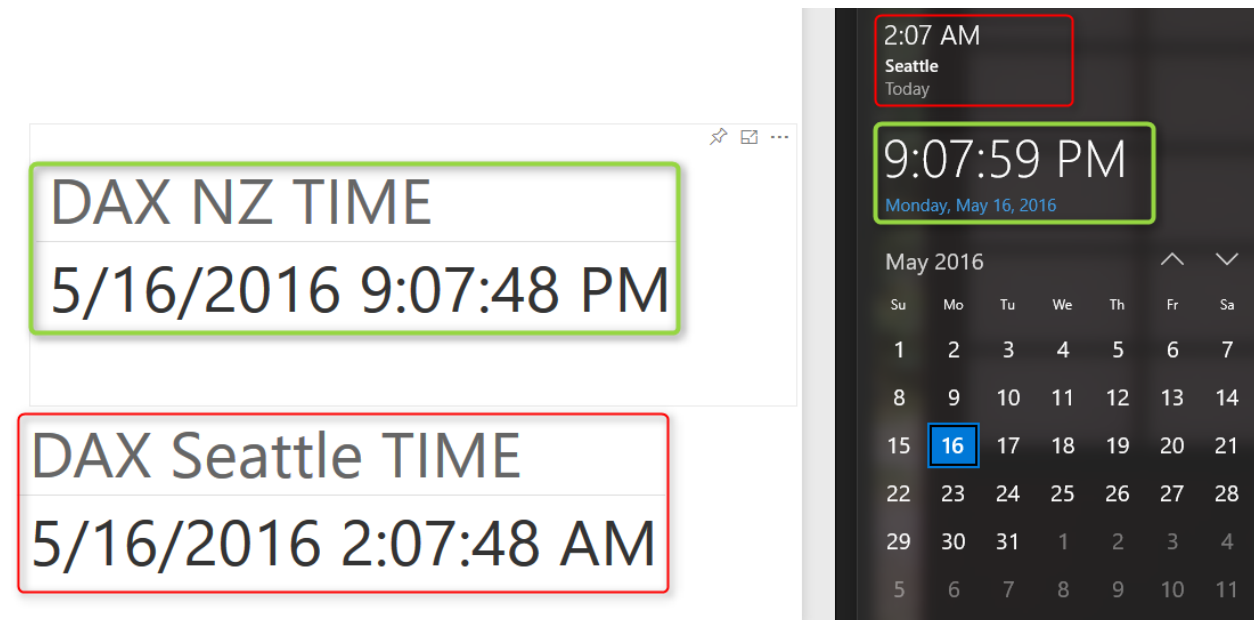
The other limitation which I like to mention at this stage is: Not Inheriting Formatting.

By not inheriting formatting I mean the calculated table doesn't inherit format from the source table. In some complex scenarios where the calculation comes from many tables that might not be necessary. but for our role playing simple example above; If my original date dimension has some formatting configuration. such as setting DateKey to a "Do Not Summarize", or some other configuration, then I would like to see the same in the calculated table fetched out of this.

The formatting applied on the calculated table columns also will be overwritten after each change in the DAX expression.

Solving DAX Time Zone Issue in Power BI

Published Date: May 17, 2016

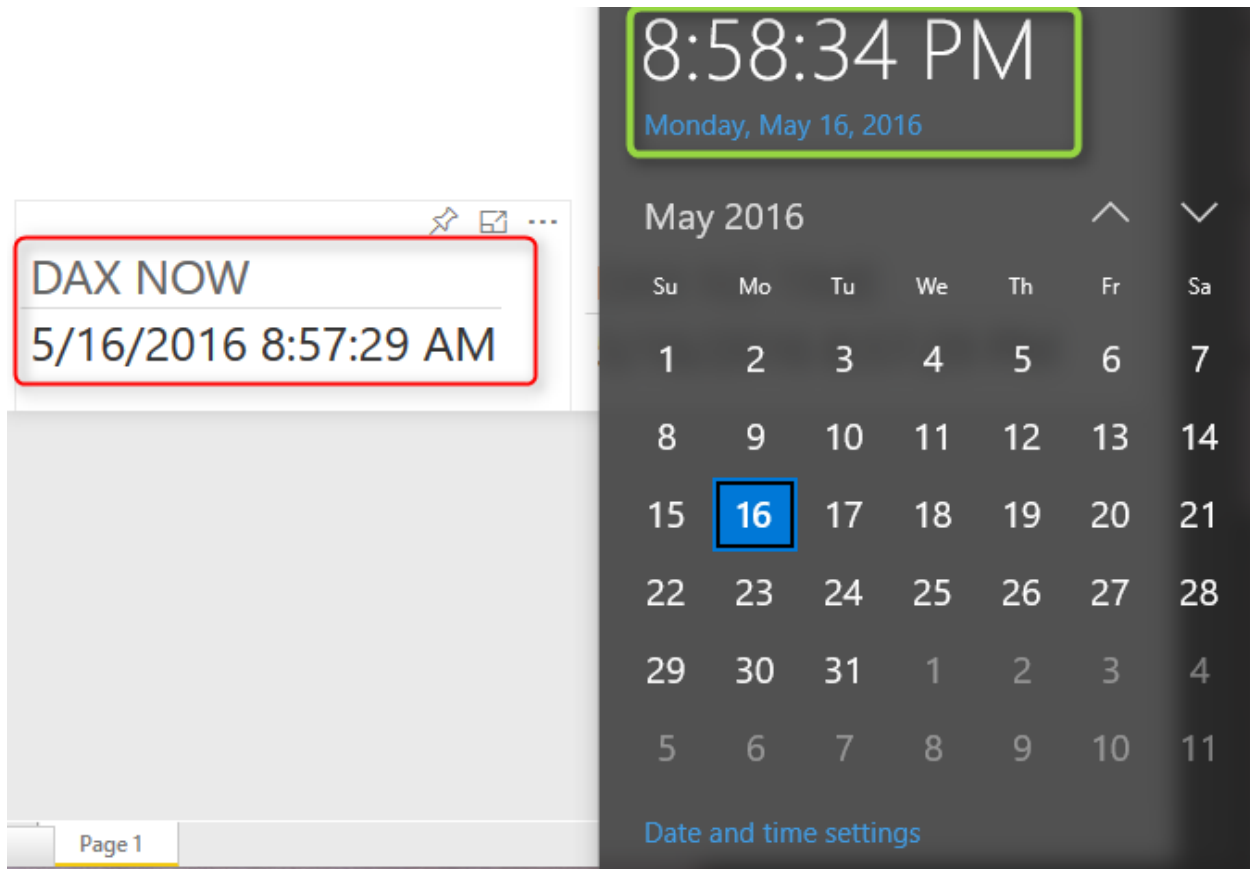


Power BI is a cloud service, and that means Power BI files are hosted somewhere. Some DAX functions such as Date/Time functions work on system date/time on the server their file is hosted on. So If you use DAX functions such as TODAY() or NOW() you will not get your local date/time, You will fetch server's date/time. In this blog post I'll explain methods of solving this issue, so you could use Power BI to resolve your specific time zone's date and time. If you want to learn more about Power BI read Power BI online book; [Power BI from Rookie to Rock Star](#).

Defining the Problem

Using DAX functions on your local Power BI file is totally different from what you will see in Power BI website especially when date and time functions has been used. The reason is that DAX works with the date and time of the system that hosted the Power BI file. Power BI is a could based service, and that means Power BI files will be hosted on a server somewhere in the world, that might not be on the same time zone as your city is. So as a result when you used functions that works with the current date and time; such as TODAY() or NOW() in DAX you will not get your local current date and time. At this stage there is time zone feature in DAX functions to help resolving this, so I suggest few options to resolve it as below.

Screenshot below shows a Power BI report published on Power BI website, and the result of DAX NOW() function there compared with the local date/time on the client system. Please note that you won't see this anomaly in Power BI Desktop, because in that case file is running on your local system, and the result would be your local date/time, you will only face this issue when you publish solution to Power BI website.



Method 1 – DAX Formula Manipulation

One easy way of solving this is to add time offset to the date/time function in DAX. Power BI date/time seems to be GMT. So if I want to show my local time in Auckland, I have to add 12 hours to it. Or for Seattle I have to reduce 7 hours from it.

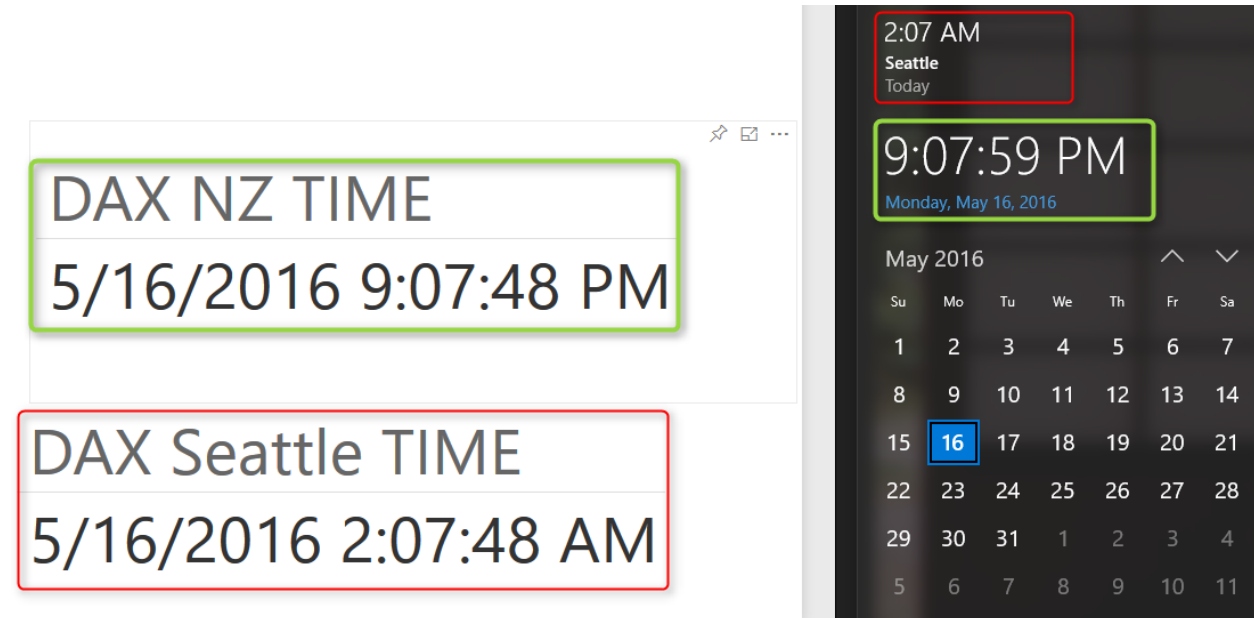
So I create a new calculation as DAX NZ TIME with this code:

DAX NZ TIME = NOW() **+(12/24)**

and another for DAX Seattle Time with this code:

DAX Seattle TIME = NOW() **-(7/24)**

Here is corrected result as below;



The screenshot displays two DAX queries and their results on the left, and a Windows desktop clock and calendar on the right. The DAX results are as follows:

DAX Query	Result
DAX NZ TIME	5/16/2016 9:07:48 PM
DAX Seattle TIME	5/16/2016 2:07:48 AM

The Windows clock on the right shows the time as 2:07 AM in Seattle on Monday, May 16, 2016. The calendar below the clock shows the month of May 2016, with the 16th highlighted.

This method works but has an issue which I deal with it later on.

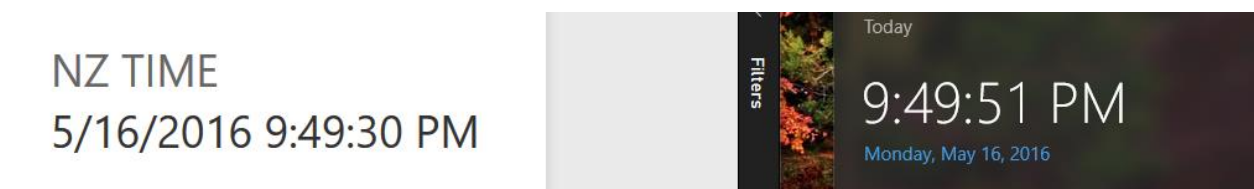
Method 2 – Power Query DateTimeZone Functions

Thanks to my friend [Ken Puls](#) who mentioned this method to me in PASS BA conference, I come with this second option. Fortunately in Power Query there is set of functions for [DateTimeZone](#). Ken already has a [blog post](#) about time zones with Power Query which is a good read and recommended. DateTimeZone functions has options such as fetching local time or switching time zones. For this purpose I can use [DateTimeZone.SwitchZone](#) function to switch server's local time to my time zone's date and time.

```
= DateTimeZone.SwitchZone(DateTimeZone.LocalNow(),12,0)
```

12 is hours, and 0 is minutes for the new time zone. script above will turn the local time zone to NZ time. for turning that into Seattle time I have to set parameters to -7, and 0.

And here is the result set:



The screenshot displays the DAX results on the left and a Windows desktop clock on the right. The DAX result is as follows:

DAX Query	Result
NZ TIME	5/16/2016 9:49:30 PM

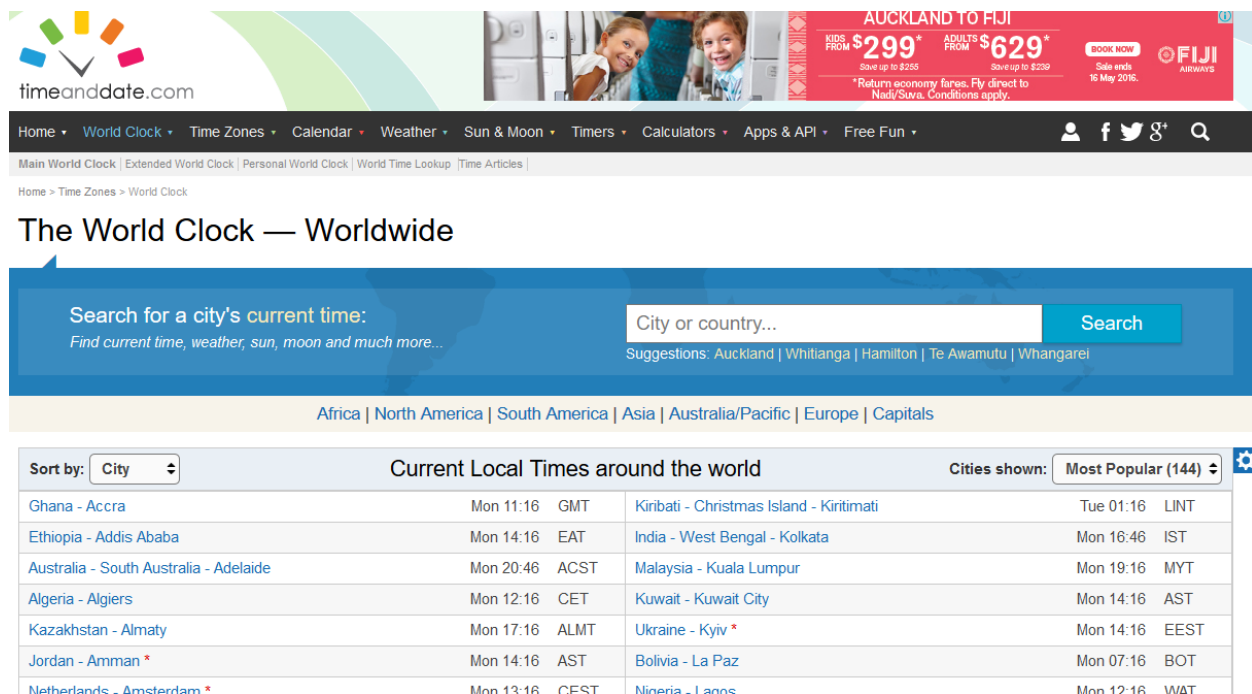
The Windows clock on the right shows the time as 9:49:51 PM on Monday, May 16, 2016.

You can also use other functions such as [DateTime.AddZone](#) in Power Query to turn the local date time to specific time zone.

Well above solution works like DAX method, but both suffer from similar issue; Day Light Saving Time. This is the reason that if you try code above in summer or winter you might get different result!

Method 3 – Web Query with Power Query

Day Light Saving is a big challenge, because each time zone, city, or country might have different day light saving time. even same city might have different dates for [DST](#) (Daylight Saving Time) for different years! Power Query is intelligence enough to help with Time Zone issue, but doesn't have a directory of all DST times for all time zones. Fortunately Power Query can query web URL. And there are some websites that give you the current date and time for your specific city, country, or time zone. And those websites usually consider DST correctly. One of these websites is [TimeAndDate.com](#). As you see in screenshot below this website gives me the current date and time for most of cities around the world;



timeanddate.com

Home • World Clock • Time Zones • Calendar • Weather • Sun & Moon • Timers • Calculators • Apps & API • Free Fun •

Main World Clock | Extended World Clock | Personal World Clock | World Time Lookup | Time Articles |

Home > Time Zones > World Clock

The World Clock — Worldwide

Search for a city's current time:
Find current time, weather, sun, moon and much more...

City or country...

Suggestions: Auckland | Whitianga | Hamilton | Te Awamutu | Whangarei

Africa | North America | South America | Asia | Australia/Pacific | Europe | Capitals

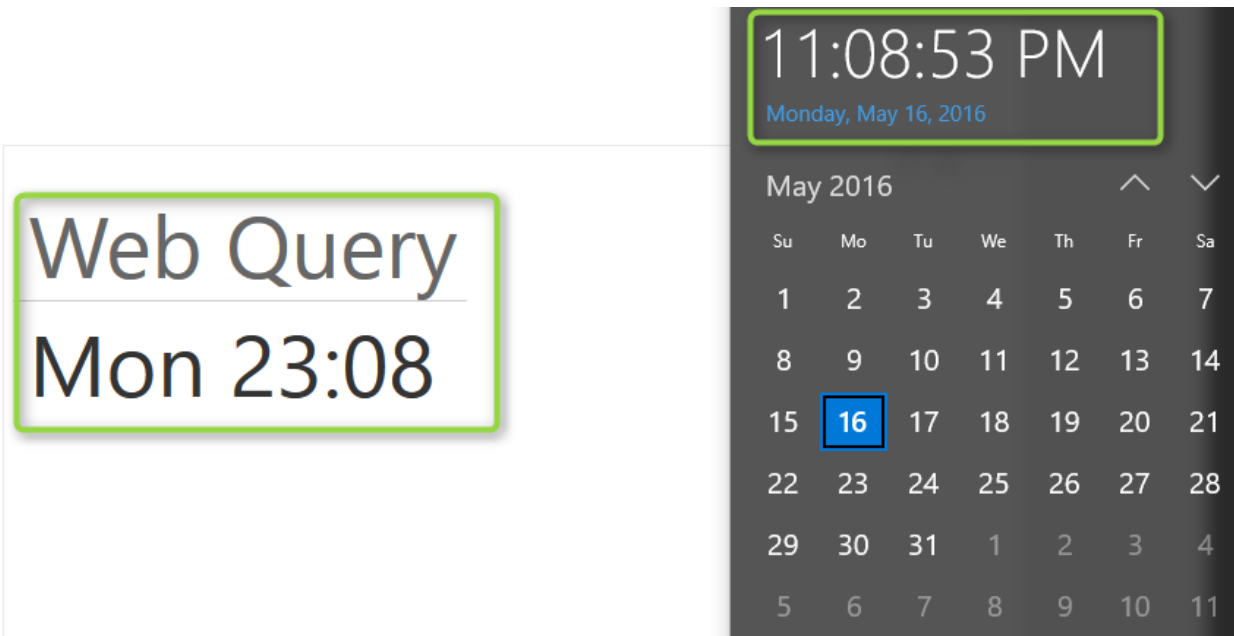
Sort by: <input type="text" value="City"/>	Current Local Times around the world		Cities shown: <input type="text" value="Most Popular (144)"/>
Ghana - Accra	Mon 11:16 GMT	Kiribati - Christmas Island - Kiritimati	Tue 01:16 LINT
Ethiopia - Addis Ababa	Mon 14:16 EAT	India - West Bengal - Kolkata	Mon 16:46 IST
Australia - South Australia - Adelaide	Mon 20:46 ACST	Malaysia - Kuala Lumpur	Mon 19:16 MYT
Algeria - Algiers	Mon 12:16 CET	Kuwait - Kuwait City	Mon 14:16 AST
Kazakhstan - Almaty	Mon 17:16 ALMT	Ukraine - Kyiv *	Mon 14:16 EEST
Jordan - Amman *	Mon 14:16 AST	Bolivia - La Paz	Mon 07:16 BOT
Netherlands - Amsterdam *	Mon 13:16 CEST	Nigeria - Lagos	Mon 12:16 WAT

In Power Query we can use functions such as [Web.Page\(\)](#) and [Web.Contents\(\)](#) to read tables in a web page, and then fetch part of it that we want with some other transformations. I won't be explaining details of using timeanddate.com URL to fetch the

local city here because it would make this post very long. I just refer you to my other post about reading some date/time information for different time zones which is similar to method I've used here. If you want to understand how code below works read the post [here](#). For this part I will be using [another website](#) which gives me current date and time in Auckland, and here is Power Query code:

```
let
    Source = Web.Page(Web.Contents("http://localtimes.info/Oceania/New_Zealand/Auckland/")),
    Data1 = Source{1}[Data],
    #"Changed Type" = Table.TransformColumnTypes(Data1,{{"Column1", type text}, {"Column2", type text}}),
    date = #"Changed Type"{1}[Column2],
    time = #"Changed Type"{0}[Column2],
    datetime = DateTime.FromText(date & " " & time)
in
    datetime
```

And here is the result with the correct DST and time zone;

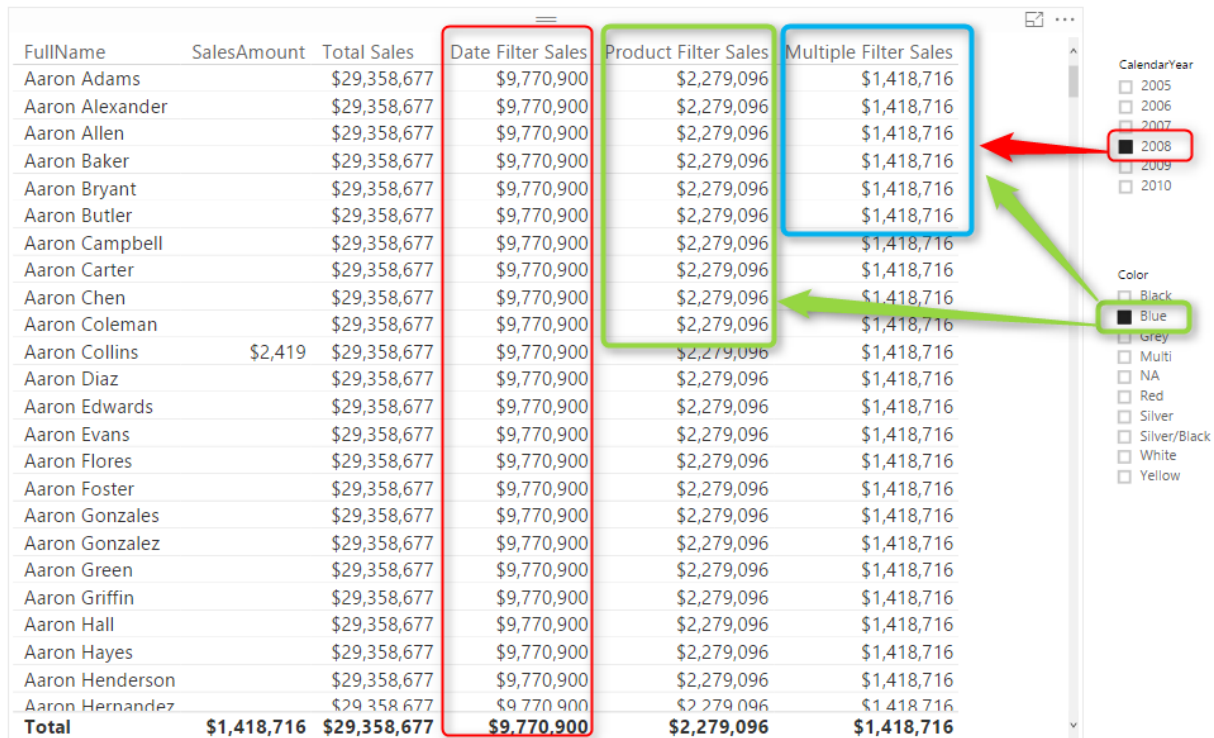


Other Methods

At the time of writing this post, I've only thought about these three methods. You might have an idea about another method. In that case, don't hesitate to share it here.

Overwrite Interaction of Power BI with DAX

Published Date: January 24, 2017



FullName	SalesAmount	Total Sales	Date Filter Sales	Product Filter Sales	Multiple Filter Sales
Aaron Adams		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Alexander		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Allen		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Baker		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Bryant		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Butler		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Campbell		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Carter		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Chen		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Coleman		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Collins	\$2,419	\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Diaz		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Edwards		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Evans		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Flores		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Foster		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Gonzales		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Gonzalez		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Green		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Griffin		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Hall		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Hayes		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Henderson		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Hernandez		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Total	\$1,418,716	\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716

I have written a blog post while ago about using Edit Interaction to [control the interaction of visuals](#). There are however sometimes that you want to have some of filters to be applied on (some measures) of a single visual, but others not. In this post I'll explain one method of writing DAX expressions in a way that overwrite the way that Power BI visuals interact. You will learn how to write a DAX expression that some filters affect on that, some not. Let's see how the method works. If you want to learn more about Power BI; read [Power BI online book from Rookie to Rock Star](#).

Prerequisite

For running example of this post you will need AdventureWorksDW sample database, or you can download **Excel version** of it from here:

Enter Your Email to download the file (required)

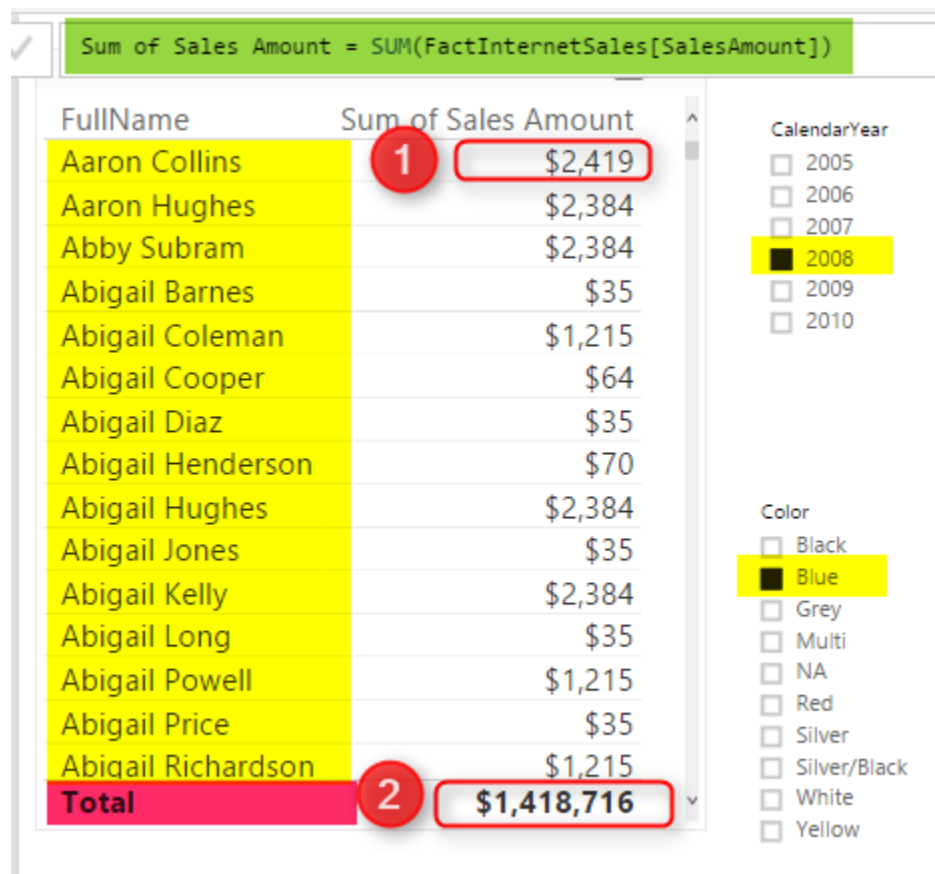
[Download]

Filter Context in DAX

It is impossible to start explaining this method without talking about filter context. Filter context is everything you used for filtering and slicing and dicing in the report. As an example; create a Power BI model based on DimDate, DimCustomer, DimProduct, and FactInternetSales. Make sure that you have only one active relationship between FactInternetSales (OrderDateKey) and DimDate (DateKey). Create a Measure with DAX code below:

Sum of Sales Amount = SUM(FactInternetSales[SalesAmount])

Now Create a report with a Table of Full Name (from DimCustomer), Sum of Sales Amount. also create two slicers; one for Calendar Year, and another for Product Color.



In above screenshot you can see that result of Sum of Sales Amount is not always same value. IT DEPENDS! Depends on what filter you have selected, or what values you have sliced and diced based on. For example Highlight numbered 1, shows sum of Sales Amount for product Color Blue, Calendar Year 2008, and Customer Full Name "Aaron Collins". While the highlight numbered 2, shows sum of Sales Amount for Year 2008, and color Blue, but for all Customers. What you see here is Filter Context.

Filter Context is all filters, slicers, highlight, and slicing and dicing applied to a report or visual. Filter Context for number 1 in above image is: product Color Blue, Calendar Year 2008, and Customer Full Name "Aaron Collins".

Everything in DAX resolves based on Filter Context and Row Context. However there are some ways to control the context. Controlling the context means controlling interaction of visuals. In above example, with any change in the slicer, filter context changes, and result of Sum(SalesAmount) also changes. However if we write a DAX expression that doesn't change with selecting a slicer, that means we have controlled the context. Let's look at some examples.

Total Sales Regardless of Filters

As an example; you can create a measure that returns total sales amount regardless of what is selected in slicers or filters, regardless of what the filter context is. For this you can use either Iterators (SumX, MinX, MaxX,...) or Calculate function. Create measure below;

```
Total Sales = SUMX(ALL(FactInternetSales),FactInternetSales[SalesAmount])
```

In above DAX expression **ALL** function will act regardless of filter context. No matter what Filter context is ALL will return everything, and as a result **SUMX** will calculate sum of SalesAmount for all rows. Here is a screenshot of report;

FullName	Sum of Sales Amount	Total Sales
Aaron Adams		\$29,358,677
Aaron Alexander	\$70	\$29,358,677
Aaron Allen		\$29,358,677
Aaron Baker	\$1,751	\$29,358,677
Aaron Bryant	\$75	\$29,358,677
Aaron Butler	\$15	\$29,358,677
Aaron Campbell	\$1,155	\$29,358,677
Aaron Carter	\$40	\$29,358,677
Aaron Chen	\$40	\$29,358,677
Aaron Coleman	\$62	\$29,358,677
Aaron Collins	\$2,469	\$29,358,677
Aaron Diaz		\$29,358,677
Aaron Edwards		\$29,358,677
Aaron Evans		\$29,358,677
Aaron Flores	\$839	\$29,358,677
Total	\$9,770,900	\$29,358,677

CalendarYear

- ☐ 2005
☐ 2006
☐ 2007
☒ 2008
☐ 2009
☐ 2010

Color

- ☐ Black
☐ Blue
☐ Grey
☐ Multi
☐ NA
☐ Red
☐ Silver
☐ Silver/Black
☐ White
☐ Yellow

Doesn't matter what filter you select, or what slicer you click, result for measure is always total value. Now let's control the context a bit different.

Total Sales Filterable Only by Date Selection

Let's take one step forward with bringing one selection criteria in the measure. For this measure we want to create a Total Sales that can be only changed when a date selection happens (Year in our example), but nothing else.

Because we need multiple filters now, I'll do it this time with CALCULATE function where I can specify multiple filters. Here is the code:

```

Date Filter Sales = CALCULATE( SUM(FactInternetSales[SalesAmount]),
    DATESBETWEEN(DimDate[FullDateAlternateKey],
        FIRSTDATE(DimDate[FullDateAlternateKey]),
        LASTDATE(DimDate[FullDateAlternateKey])
    ),
    ALL(FactInternetSales)
)
  
```

In measure above we have two filters; ALL(FactInternetSales), and DatesBetween(). DatesBetween brings everything from the FirstDate to the LastDate. FirstDate and

LastDate will be depends on the date selection in slicer. as a result DatesBetween will return the filter context of date selection, however everything else will be ignored by ALL(FactInternetSales). Result will be a filter which is junction of these two filter. Here is the result;

FullName	Sum of Sales Amount	Total Sales	Date Filter Sales
Aaron Adams		\$29,358,677	\$9,770,900
Aaron Alexander	\$70	\$29,358,677	\$9,770,900
Aaron Allen		\$29,358,677	\$9,770,900
Aaron Baker	\$1,751	\$29,358,677	\$9,770,900
Aaron Bryant	\$75	\$29,358,677	\$9,770,900
Aaron Butler	\$15	\$29,358,677	\$9,770,900
Aaron Campbell	\$1,155	\$29,358,677	\$9,770,900
Aaron Carter	\$40	\$29,358,677	\$9,770,900
Aaron Chen	\$40	\$29,358,677	\$9,770,900
Aaron Coleman	\$62	\$29,358,677	\$9,770,900
Aaron Collins	\$2,469	\$29,358,677	\$9,770,900
Aaron Diaz		\$29,358,677	\$9,770,900
Aaron Edwards		\$29,358,677	\$9,770,900
Aaron Evans		\$29,358,677	\$9,770,900
Aaron Flores	\$839	\$29,358,677	\$9,770,900
Total	\$9,770,900	\$29,358,677	\$9,770,900

CalendarYear

☐ 2005

☐ 2006

☐ 2007

☒ 2008

☐ 2009

☐ 2010

Color

☐ Black

☐ Blue

☐ Grey

☐ Multi

☐ NA

☐ Red

☐ Silver

☐ Silver/Black

☐ White

☐ Yellow

You can see that the value in this new measure (Date Filter Sales) changes by any selection in Calendar Year slicer, but nothing else. Result of this measure will be always sum of sales amount for all transactions in the selected year. If nothing is selected in Year slicer, then this column's value will be similar to Total Sales Measure.

How if we want to enable multiple filters then? let's look at another measure.

Measure Filterable with Multiple Selections

Let's go even one step further and add a measure that can be only affected by selecting values in date and product slicer, but nothing else. You know the answer already I believe. You just need to add one more filter to the list of filters. I'll do it this time with a RelatedTable function, but you can do it with other methods as well. Here is the new measure;

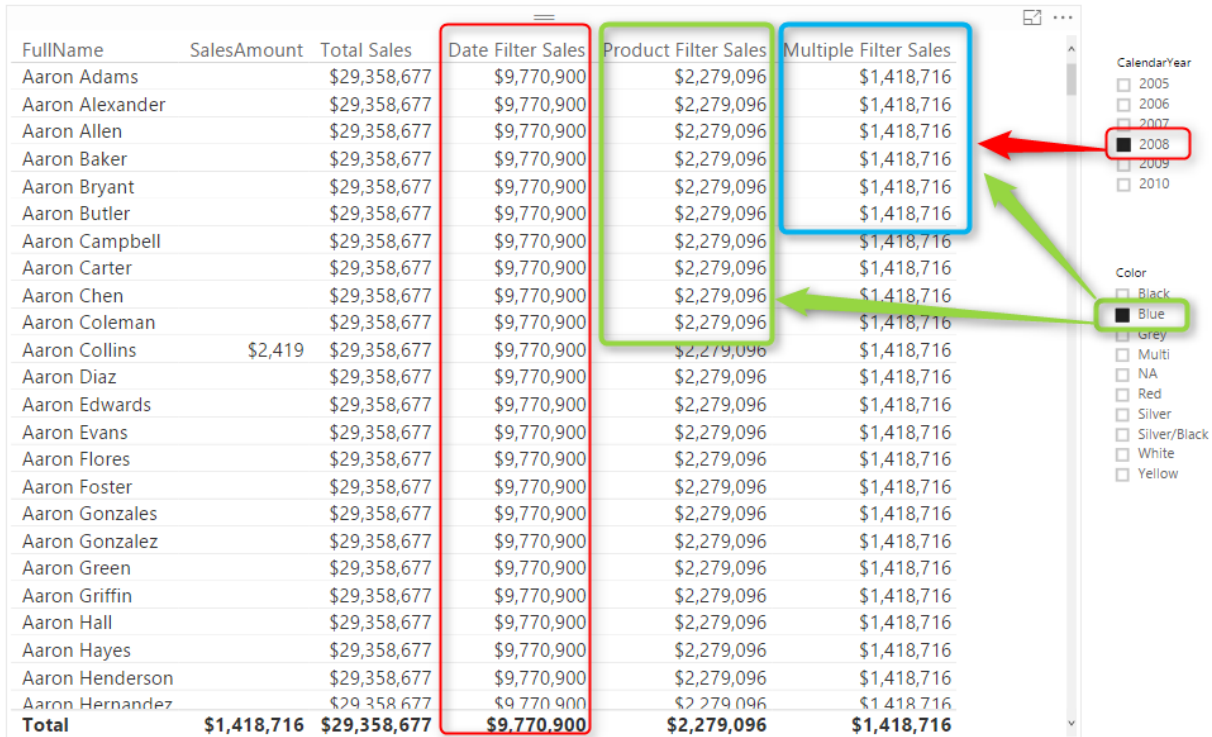
```
Multiple Filter Sales = CALCULATE( SUM(FactInternetSales[SalesAmount]),
    ALL(FactInternetSales),
    RELATEDTABLE(DimProduct),
```

```

DATESBETWEEN(DimDate[FullDateAlternateKey],FIRSTDATE(DimDate[FullDateAlternateKey]),LAST
DATE(DimDate[FullDateAlternateKey]))
)

```

Above measure is similar to the previous measure with only one more filter: RelatedTable(DimProduct). This filter will return only sub set of select products. As a result for this measure Product and Date selection will be effective;



FullName	SalesAmount	Total Sales	Date Filter Sales	Product Filter Sales	Multiple Filter Sales
Aaron Adams		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Alexander		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Allen		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Baker		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Bryant		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Butler		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Campbell		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Carter		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Chen		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Coleman		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Collins	\$2,419	\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Diaz		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Edwards		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Evans		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Flores		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Foster		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Gonzales		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Gonzalez		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Green		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Griffin		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Hall		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Hayes		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Henderson		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Aaron Hernandez		\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716
Total	\$1,418,716	\$29,358,677	\$9,770,900	\$2,279,096	\$1,418,716

Summary

As you can see simply with DAX expressions you can control the filter context, or in other words you can control the interaction in Power BI. Note that you can write DAX expressions in many different ways, the expression above are not the only way of controlling filter context. Iterators and Calculate function can be very helpful in changing this interaction.

Previous Dynamic Period DAX Calculation

Published Date: January 12, 2017



Parallel Period is a function that help you fetching previous period of a Month, Quarter, or Year. However if you have a dynamic range of date, and you want to find the previous period of that dynamic selection, then Parallel Period can't give you the answer. As an example; if user selected a date range from 1st of May 2008 to 25th of November 2008, the previous period should be calculated based on number of days between these two dates which is 208 days, and based on that previous period will be from 5th of October 2007 to 30th of April 2008. The ability to do such calculation is useful for reports that user want to compare the value of current period with whatever period it was before this. In this post I'll show you an easy method for doing this calculation, I will be using one measure for each step to help you understand the process easier. If you like to learn more about DAX and Power BI, read [Power BI online book from Rookie to Rock Star](#).

Prerequisite

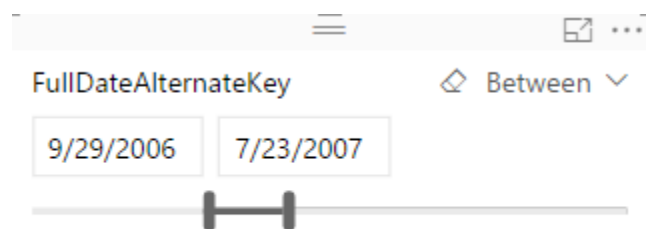
For running example of this post you will need AdventureWorksDW sample database, or you can download **Excel version** of it from here:

Enter Your Email to download the file (required)

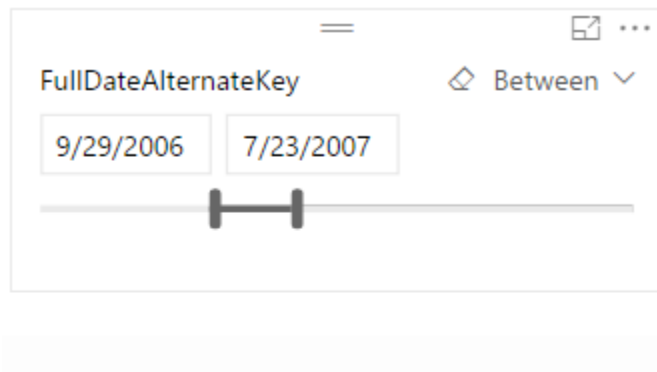
[Download]

Current Period

I will go through this with an example; Create a new Power BI Desktop file and choose DimDate, and FactInternetSales from AdventureWorksDW. Make sure that there is only one **Active** relationship between these two tables based on OrderDateKey in the FactInternetSales table and DateKey in the DimDate table. Now add a slicer for FullDateAlternateKey in the page



Also add a Card visual which shows SalesAmount from FactInternetSales table.



\$21.71M
This Period Sales

I normally prefer to create an explicit measure for this type of calculations, that's why I have create a measure named "This Period Sales" with DAX code below; (the measure for This Period Sales is not necessary, because Power BI does the same calculation automatically for you)

This Period Sales = SUM(FactResellerSales[SalesAmount])

Start of Current Period

To understand the current period, an easy way can be calculating start, end of period and number of days between these two. Start of Period is simple. I just create a measure under DimDate, as below:

Start of This Period = FIRSTDATE(DimDate[FullDateAlternateKey])

FirstDate() DAX function returns the first available date in the current evaluation context, which will be whatever filtered in the date range.

End of Current Period

Same as start of period, for end of period I will use a simple calculation, but this time with *LastDate()* to find the latest date in the current selection.

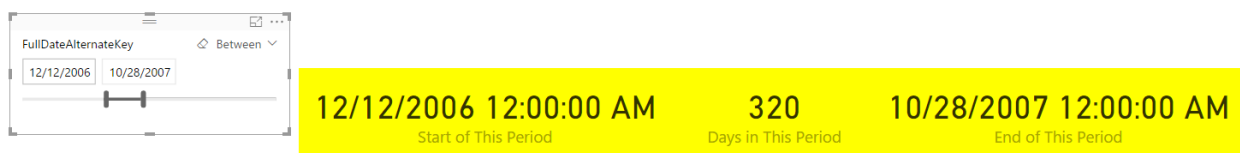
End of This Period = LASTDATE(DimDate[FullDateAlternateKey])

Days in This Period

Next easy step is understanding number of days between start and end of period, which is simply by using *DateDiff()* DAX function as below;

Days in This Period = DATEDIFF([Start of This Period],[End of This Period],DAY)

I add them all in the report as Card Visuals (one for each measure), and here is the result so far;



Previous Period

After finding number of days in this period, start, and end of current period, it is a simple calculation to find the previous period. Previous period calculation should be number of days in this period minus start of current period. to exclude the start of period to calculate twice, I'll move one more day back. Here is the calculation step by step, I'll start with Start of Previous Period;

Start of Previous Period

Using DateAdd to reduce number of days from DimDate

DATEADD(DimDate[FullDateAlternateKey],-1*[Days in This Period],DAY)

DateAdd() DAX function adds a number of intervals to a **date set**. In this example interval is DAY, and date set is all dates in DimDate[FullDateAlternateKey] field (because DateAdd doesn't work with single date), and the number of intervals is Days in This Period multiplied by -1 (to move dates backwards rather than forward).

Fetch the First Date of the result

FIRSTDATE(DATEADD(DimDate[FullDateAlternateKey],-1*[Days in This Period],DAY))

FirstDate() used here to fetch first value only.

Move one day back

```
PREVIOUSDAY(FIRSTDATE(DATEADD(DimDate[FullDateAlternateKey],-1*[Days in This Period],DAY)))
```

To exclude current date from the selection we always move one day back, that's what *PreviousDay()* DAX function does. it always returns a day before the input date.

These are not three separate DAX expressions or measure, this is only one measure which I explained step by step. here is the full expression:

```
Start of Previous Period = PREVIOUSDAY(FIRSTDATE(DATEADD(DimDate[FullDateAlternateKey],-1*[Days in This Period],DAY)))
```

End of Previous Period

Similar to the Start of Previous Period calculation, this calculation is exactly the same the only difference is using *LastDate()*;

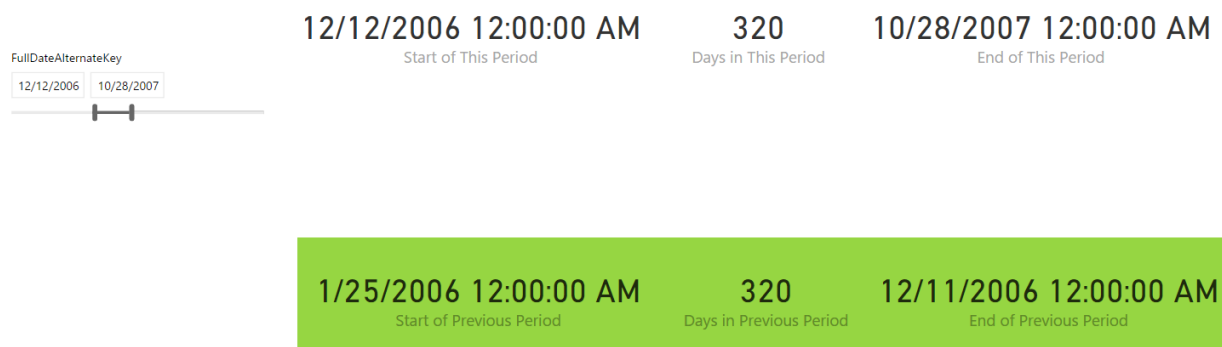
```
End of Previous Period = PREVIOUSDAY(LASTDATE(DATEADD(DimDate[FullDateAlternateKey],-1*[Days in This Period],DAY)))
```

Days in Previous Period

You don't need to create this measure, I have only created this to do a sanity check to see do I have same number of days in this period compared with previous period or not;

```
Days in Previous Period = DATEDIFF([Start of Previous Period],[End of Previous Period],DAY)
```

Now if I add all of these measure to the report with card visuals again I can see previous period calculation works correctly;



With every change you apply in date range slicer you can see the previous period calculates the range again, it will be always same number of days as the current period,

but same number of days BEFORE. in the screenshot above you can see that start of previous period is 321 days before start of this period (1 more days because the end of previous period is not exactly start of this period, it is one day before. we don't want to duplicate values of date in current and previous calculations).

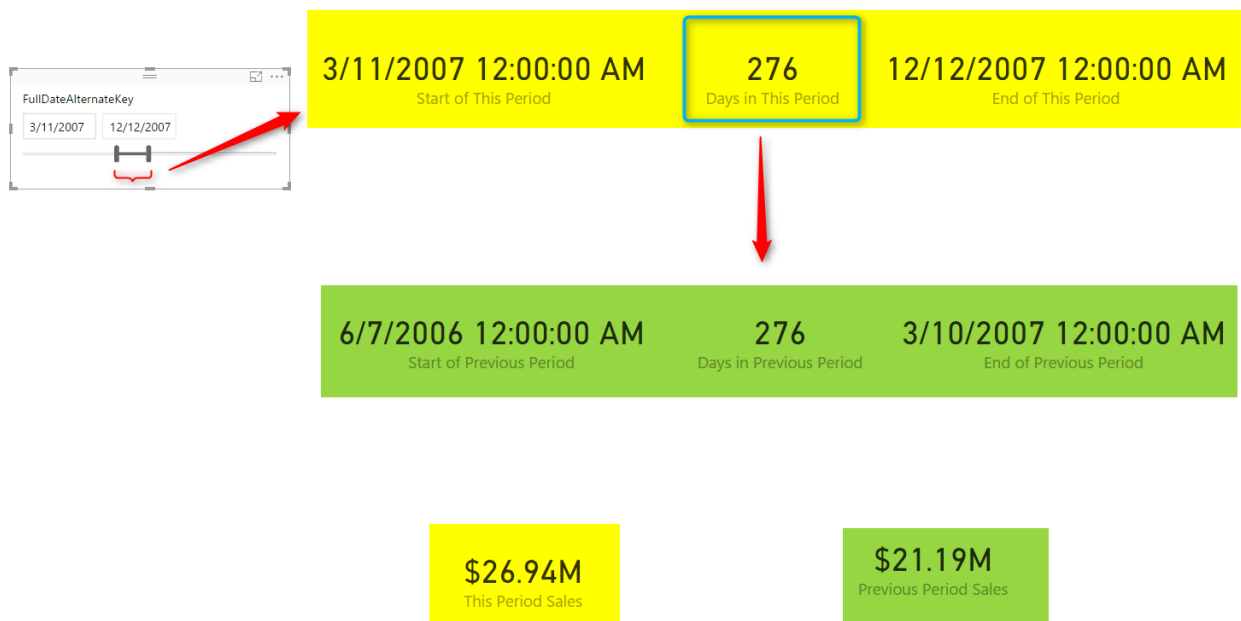
Previous Period Sales

Now as an example I have created another measure to show you the sum of SalesAmount for the previous period. the calculation here uses DatesBetween() DAX function to fetch all the dates between start of previous period and end of previous period;

Previous Period Sales = CALCULATE(SUM(FactResellerSales[SalesAmount])
,DATESBETWEEN(

DimDate[FullDateAlternateKey],
[Start of Previous Period],
[End of Previous Period]),
ALL(DimDate))

Showing all of these in a page now;



Summary

This was a very quick and simple post to show you a useful DAX calculation to find Dynamic Previous Period based on the selection of date range in Power BI report page. I have used number of DAX functions such as `FirstDate()`, `LastDate()`, `DateAdd()`, `DateDiff()`, and `PreviousDate()` to do calculations. Calculation logic is just counting number of days in the current period and reducing it from the start and end of the current period to find previous period.

IF and Filter are Different! Be Careful (DAX)

Published Date: March 20, 2017

Color	Sum of Red Products - With Filter ▼
Red	\$7,724,330.52
Total	\$7,724,330.52

Color	Sum of Red Products - With IF
Black	\$0.00
Blue	\$0.00
Multi	\$0.00
NA	\$0.00
Red	\$7,724,330.52
Silver	\$0.00
White	\$0.00
Yell...	\$0.00
Total	\$7,724,330.52

DAX has many functions to write conditional expressions. For example you might want to calculate sum of sales amount for all "Red" products. you can achieve it by using SUMX or Calculate, and functions such as IF or Filter to write a conditional expression for product color to be equal to "Red". At the first you might think these functions will be have same in the result set, but there is a difference that should not be overlooked. In this post I'll explain what type of problem might happen if you don't use these functions wisely. If you want to learn more about Power BI; read [Power BI book from Rookie to Rock Star](#).

Prerequisite

For running this example you would need to have a copy of AdventureWorksDW database in SQL Server, or you can download Excel version of it from here:

Enter Your Email to download the file (required)

[Download]

Brief of Functions

IF

"IF" is a conditional filtering expression function for DAX. You can simply write a conditional expression including Then and Else part of it. It simply works with this syntax;

IF(<conditional expression>, <what happens if true>, <what happens if false>)

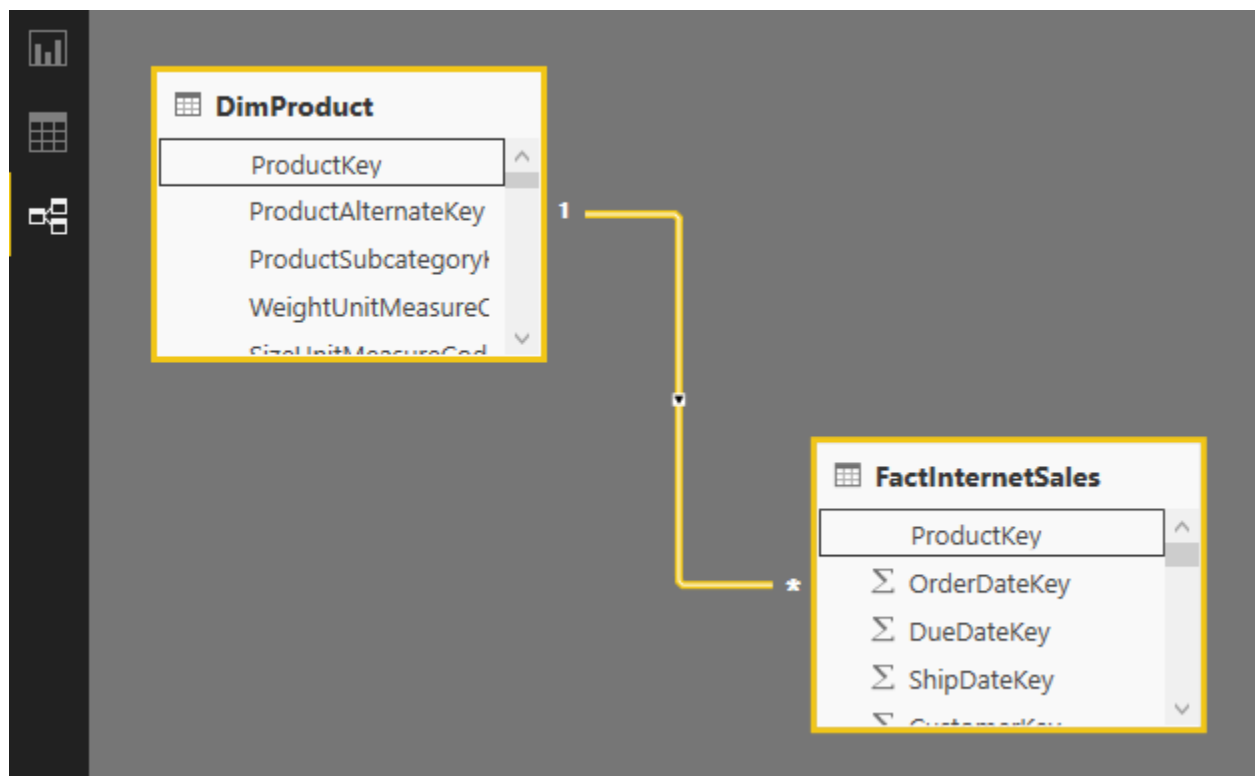
Filter

"Filter" is a function that filters data set based on a custom filter. For example you can filter only products with "Red" color. Here is an example Filter expression;

FILTER(<table>, <filter condition>)

Sample Data Set

For this example, you need to bring FactInternetSales, and DimProduct into your Power BI Model. Relationship between these tables automatically should be detected by Power BI. It should be based on ProductKey between two tables. Here is how the relationship looks like;



Conditional SUM

There are multiple ways of calculating conditional sum in DAX. you can use SUMX or CALCULATE. both of these functions calculate an expression (In this case it would be sum of sales amount from FactInternetSales), based on a filter (which would be our conditional expression to find "Red" products). I will use SUMX in this example, but same concept applies on Calculate function as well. Here is how you can use SUMX for calculating sum of "Red" products;

Method 1 – SumX with FILTER

In the first method, I can use SUMX expression and filter the data set to be only "Red" products. Create a new Measure in FactInternetSales with this expression;

```
Sum of Red Products - With Filter = SUMX(  
    FILTER(FactInternetSales,  
  
    RELATED(DimProduct[Color])="Red")  
  
    ,FactInternetSales[SalesAmount]  
  
    )
```

As you can see in the above expression, I have used a simple FILTER function to filter everything in FactInternetSales when Color or product is "Red". I have used RELATED function because Color is a column in DimProduct and Related Function goes through the relationship from Many (FactInternetSales) to One (DimProduct) and allow us to do the filtering based on a column in a related table.

Method 2 – SumX with IF

We can achieve same result with SUMX and IF condition together. In this case the condition comes as IF statement in the expression part of SUMX. Here is the new measure's code;

```
Sum of Red Products - With IF = SUMX(  
  
    FactInternetSales,
```



```

IF(RELATED(DimProduct[Color])="Red",

FactInternetSales[SalesAmount],

0)

)

```

In this expression, instead of filtering data with FILTER function, I have used a conditional expression to identify if the color of product is "Red" or not, if it is "Red", then I use SalesAmount for sum calculation, otherwise I use zero (means don't summarize for other product colors).

Method 3 – Calculate with Simple Conditional Expression

There are many other methods of calculating the conditional sum, but just adding this one because it looks different; If I use Calculate Function with simple expression for checking the color of product as a new measure;

```

Sum of Red Products - Calculate Simple Expression = CALCULATE (

SUM(FactInternetSales[SalesAmount]),

DimProduct[Color]="Red"

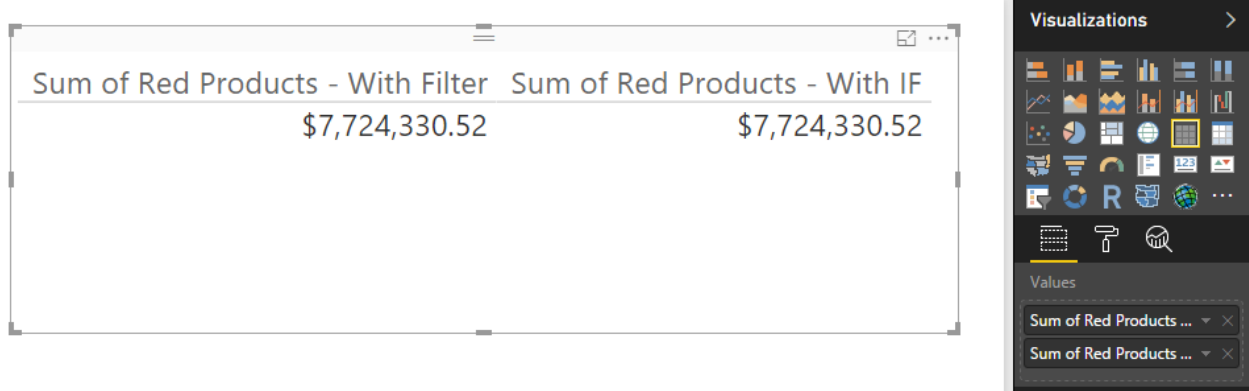
)

```

Writing DimProduct[Color]="Red" in this way is similar to writing a condition for every result set. The final result will be sum of Red Products.

Testing Results – Similar

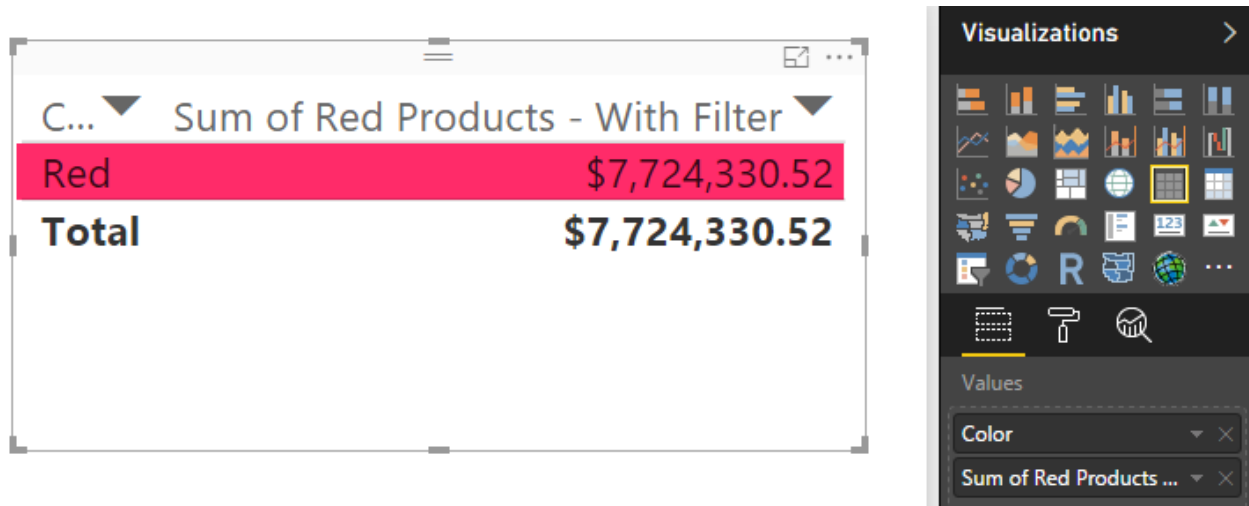
If you bring both measures in Power BI as a Table Visual you will see the result of both are identical, and it will show you total sales amount for products with "Red" Color correctly;



Sum of Red Products - With Filter	Sum of Red Products - With IF
\$7,724,330.52	\$7,724,330.52

Different Results

The result for measure is perfectly similar, however if you use one of these measures for a data set you will see the result of data set is different, and it changes the result significantly. For example, if you use "Sum of Red Products – With Filter" only in a table with "Color" from DimProduct, here is what you will see:



C...	Sum of Red Products - With Filter
Red	\$7,724,330.52
Total	\$7,724,330.52

If you use "Sum of Red Products – With IF" only in a table with "Color" from DimProduct, you will see different result;

Color	Sum of Red Products - With IF
Black	\$0.00
Blue	\$0.00
Multi	\$0.00
NA	\$0.00
Red	\$7,724,330.52
Silver	\$0.00
White	\$0.00
Yell...	\$0.00
Total	\$7,724,330.52

In both cases, the grand total is similar. However, the table with a FILTER measure will automatically filter the data set, and only shows result set for RED products, where the second table with IF measure, will show all products with zero in front of all colors, except Red. These two are VERY different in user point of view, while the final total value is similar. The reason is that IF just apply a conditional expression on the result set, where FILTER works differently and filters the data set to the custom filter expression. Notice that we don't have any Visual, Report, or Page Level filter applied in this example. Filtering happened automatically because of FILTER function.

If you bring the last method's result into a table (Sum of Red Products – Calculate Simple Expression), you will see the calculation happens on every row in the result set. It won't filter the data set, but the filter applies on calculating the final result for every row.

Color	Sum of Red Products - Calculate Simple Expression
Black	\$7,724,330.52
Blue	\$7,724,330.52
Grey	\$7,724,330.52
Multi	\$7,724,330.52
NA	\$7,724,330.52
Red	\$7,724,330.52
Silver	\$7,724,330.52
Silver/Black	\$7,724,330.52
White	\$7,724,330.52
Yellow	\$7,724,330.52
Total	\$7,724,330.52

Which One to Use?

Like many other situations; It Depends! Are you only interested in final result (sum of sales amount for "Red" products in this example)? if yes, then FILTER gives you faster result, because it first filters the data set, and then apply SUM on it. However, if you want to show a detailed view of records, then FILTER function will also filter the data set, which might not be something you want, in those cases IF would give you the correct response. If you want to calculate percentage for each row, the Calculate method might generate more reliable result. Notice that there is nothing wrong about these functions, they are working exactly as they should. However, Not knowing their actual behavior might cause some confusion for you and your users. So use them wisely and don't overlook their differences.

Control the Interaction in Power BI Report

Published Date: November 29, 2016



Power BI reports are highly interactive, If you select a column in a column chart other charts will be highlighted. Selecting a slicer value will filter all other visuals in the report. This interactivity can be controlled easily. Despite the fact that this feature has been released in early phases of Power BI, there are still many clients who don't know how to control the interaction of visuals in Power BI report. In this post you will learn how easy and useful is controlling the interaction between Power BI visual elements. If you want to learn more about Power BI; read [Power BI online book from Rookie to Rock Star](#).

What is Interactivity of Power BI Visuals?

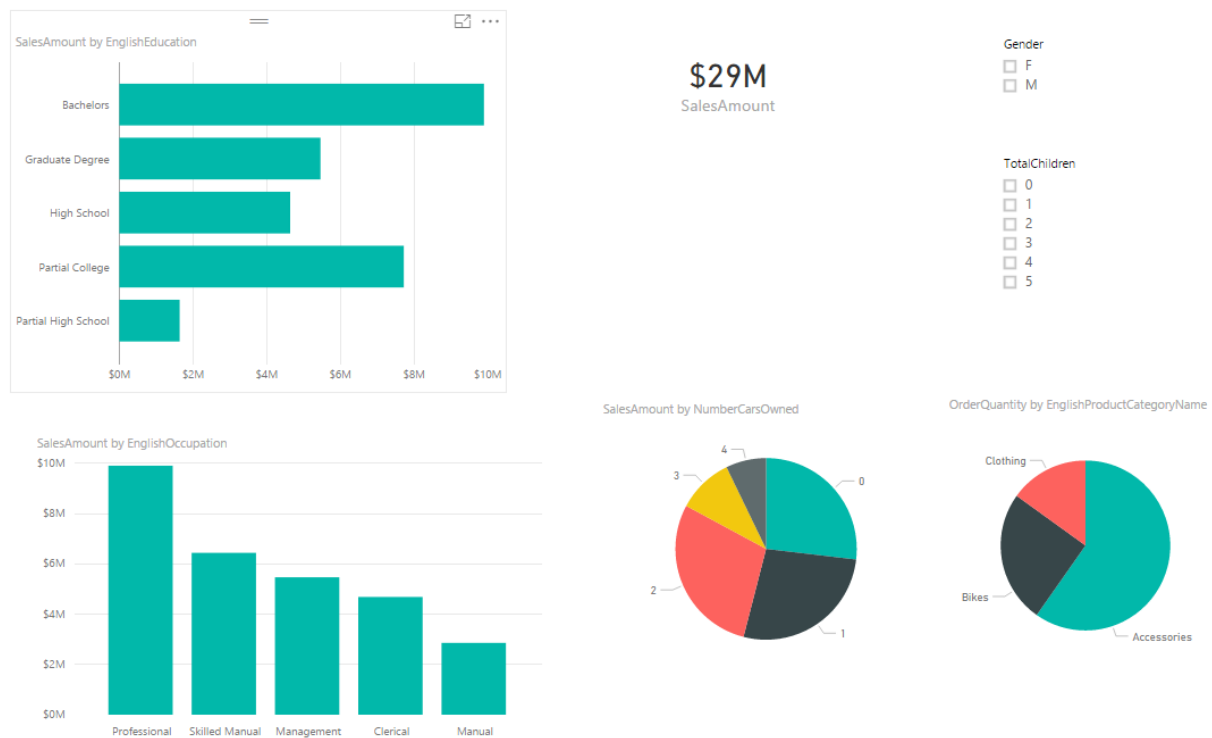
Power BI visuals are interacting with each other. Selecting an item in a visual will effect on the display of another chart. sometimes this effect is highlighting items in another chart, and sometimes filtering values in the other visual. By default all visuals in a report page are interactive with each other, however this interactivity can be controlled and modified. This functionality is very easy to change, but because there are still many people not aware of it, I ended up writing this post about it.

Prerequisite

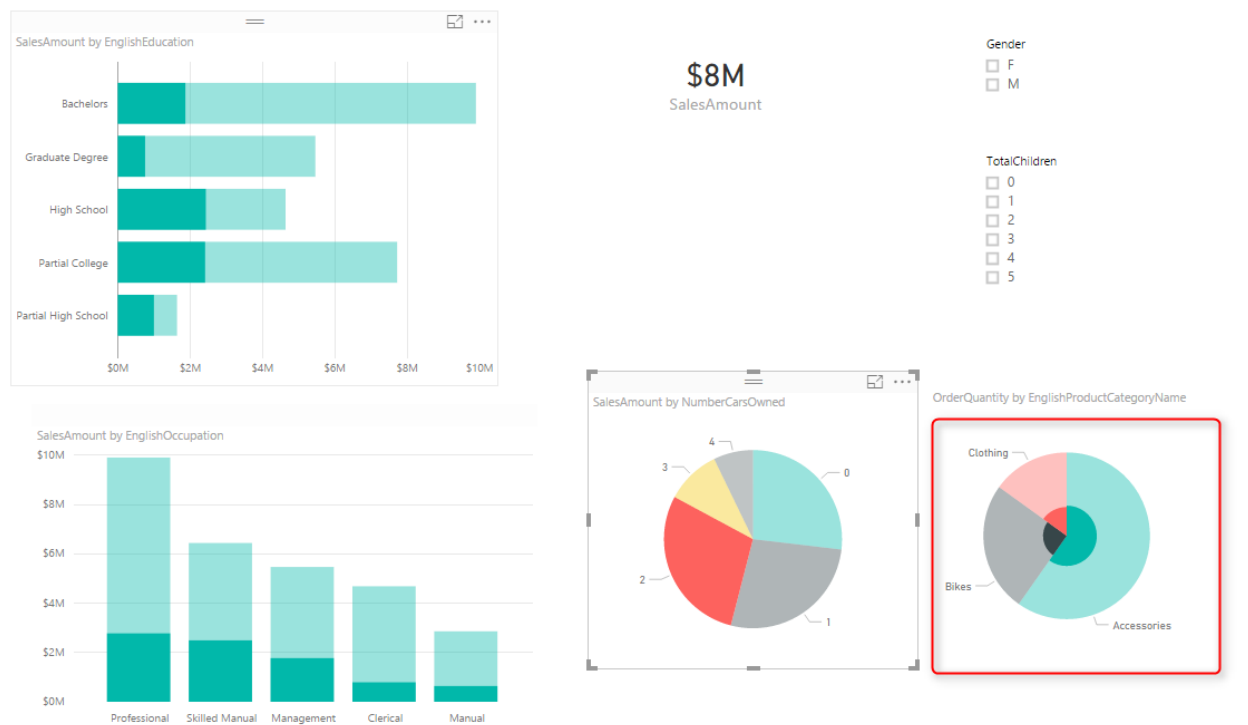
Let's start by building a very simple report from AdventureWorksDW database with few visuals. For this example bring these tables into your model: FactInternetSales, DimCustomer, DimProductCategory, DimProductSubcategory, DimProduct.

Default Behavior

Build a sample report with a Bar Chart on EnglishEducation (from DimCustomer) as Axis, and SalesAmount (from FactInternetSales) as Values. Build a Column Chart with EnglishOccupation (from DimCustomer) as Axis, and SalesAmount (from FactInternetSales) as Values. Build a Card visual with total of SalesAmount. Build a Pie Chart with NumberCarsOwned (from DimCustomer) as Legend and SalesAmount as the value. Another Pie Chart with EnglishProductCategoryName (from DimProductCategory) as Legend, and OrderQuantity (from FactInternetSales) as Value. Also create two filters; one with Gender (from DimCustomer), and another with TotalChildren (from DimCustomer). here is a view of the report;



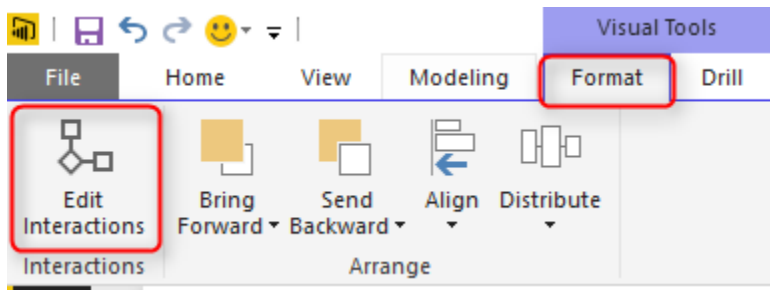
As I mentioned earlier, All visuals in Power BI report page are interactive with each other. If you click on a chart other charts will be highlighted:



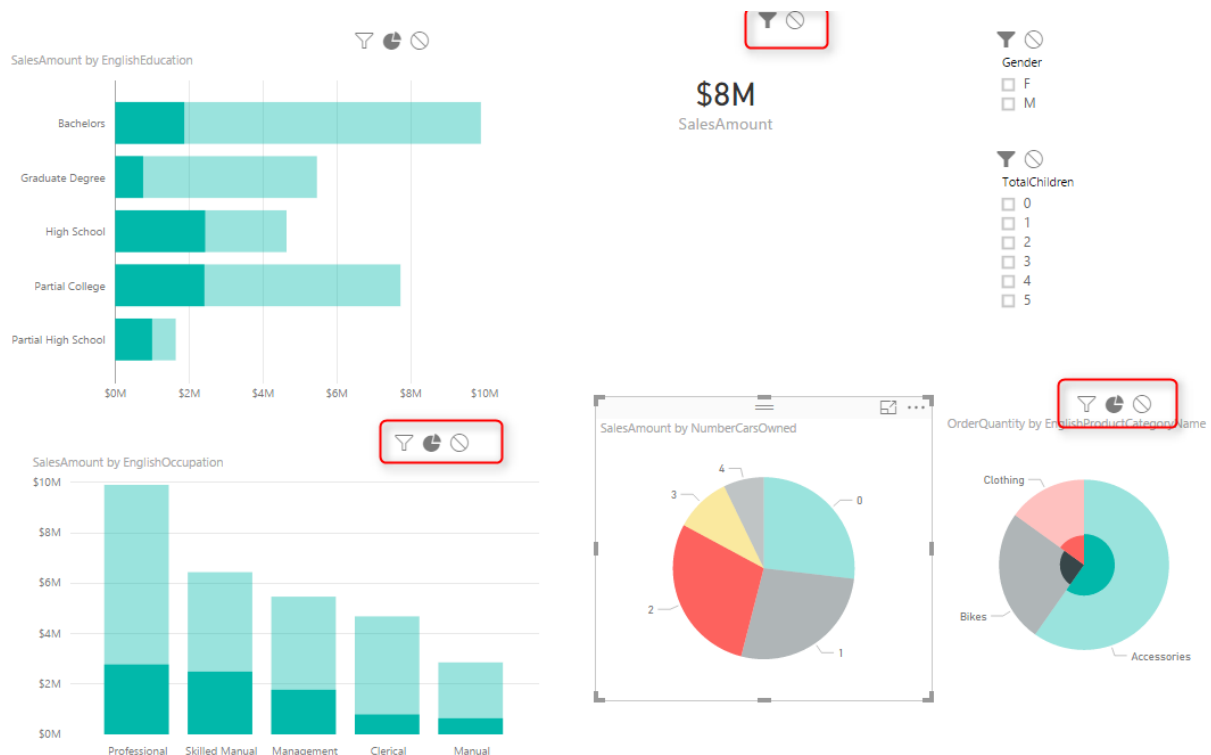
In the sample report above NumberCarsOwned 2 is selected in the Pie Chart, and related values to it highlighted in column chart and bar chart respectively. Also the total Sales Amount in the card visual changed from \$29 million to \$8 million. However if you look at the other Pie Chart (mentioned with red area in screenshot above) the Pie chart for product category and quantities are highlighted, but it is not easy to find out exactly which one is bigger (is Clothing bigger or Bikes?). Despite the nature of Pie Chart which makes things hard to visualize. There are still more things we can do here to make it more readable. If instead of highlighting this pie chart it was filtered (like the card visual) we would have better insight out of this visual.

Changing the Interaction

To change the interaction of a chart with other charts, simply select the main chart (the chart that you want to control effect of that on other charts), and then from Visual Tools menu, under Format, click on Edit Interactions.



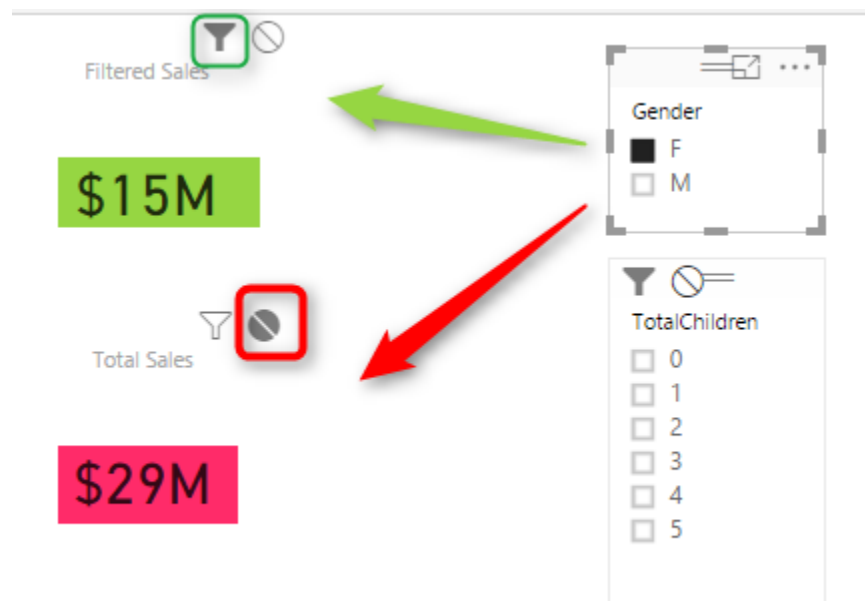
You will now see all other visuals in the page with two or three buttons on the top right hand side corner of each. These are controls of interaction.



The darker color in this icon set shows which interaction will apply. For example for the interaction from NumberCarsOwned Pie Chart and the SalesAmount Card Visual the selection is Filter. but for the ProductCategory Pie Chart is highlighting (the middle option). Now change this option to be Filter. and you will see immediately that second pie chart (product category) shows a filtered view of the selected item in NumberCarsOwned pie chart. Actually the first pie chart is acting as a filter for the second pie chart.



This behavior also can be set to None. For example let's say you want to have a total of sales amount regardless of gender selection, and then a total of sales amount for the selected gender in slicer. To do this copy the SalesAmount Card Visual, and then click on Gender Slicer. click on Edit Interaction, and set one of the card visuals to None, the other one as default with Filter.



As you can see you can set interaction between each two individual elements in a page. This is extremely helpful when you are expecting selecting an item to have different

impact on different visualization. Hope this simple tip helps to make your visualization better with Power BI.

Grouping and Binning; Step Towards Better Data Visualization

Published Date: November 3, 2016

YearlyIncome (bins)	YearlyIncome	SalesAmount
\$0	\$10,000	\$1,396,359
\$0	\$20,000	\$2,010,650
\$30,000	\$30,000	\$2,922,275
\$30,000	\$40,000	\$4,226,593
\$30,000	\$50,000	\$805,523
\$60,000	\$60,000	\$4,230,549
\$60,000	\$70,000	\$4,484,239
\$60,000	\$80,000	\$2,391,472
\$90,000	\$90,000	\$2,067,048
\$90,000	\$100,000	\$1,073,018
\$90,000	\$110,000	\$1,019,528
\$120,000	\$120,000	\$760,616
\$120,000	\$130,000	\$1,192,928
\$150,000	\$150,000	\$241,471
\$150,000	\$160,000	\$211,123
\$150,000	\$170,000	\$325,286
Total		\$29,358,677

The latest update of Power BI Desktop (October 2016) has [many features](#). Two of these features are grouping and binning. These features used to create groups of items and visualize them better in the report. Previously you could do that by Power Query or DAX calculated columns, now it is all possible simply through the graphical user interface of report editor. In this post I'll show you how to create banding (binning) and grouping simply with these features. If you like to learn more about Power BI; read the [Power BI online book from Rookie to Rock Star](#).

Prerequisite

For running examples of this post you need to have AdventureWorksDW SQL Server database sample installed. Or you can get the excel version of it.

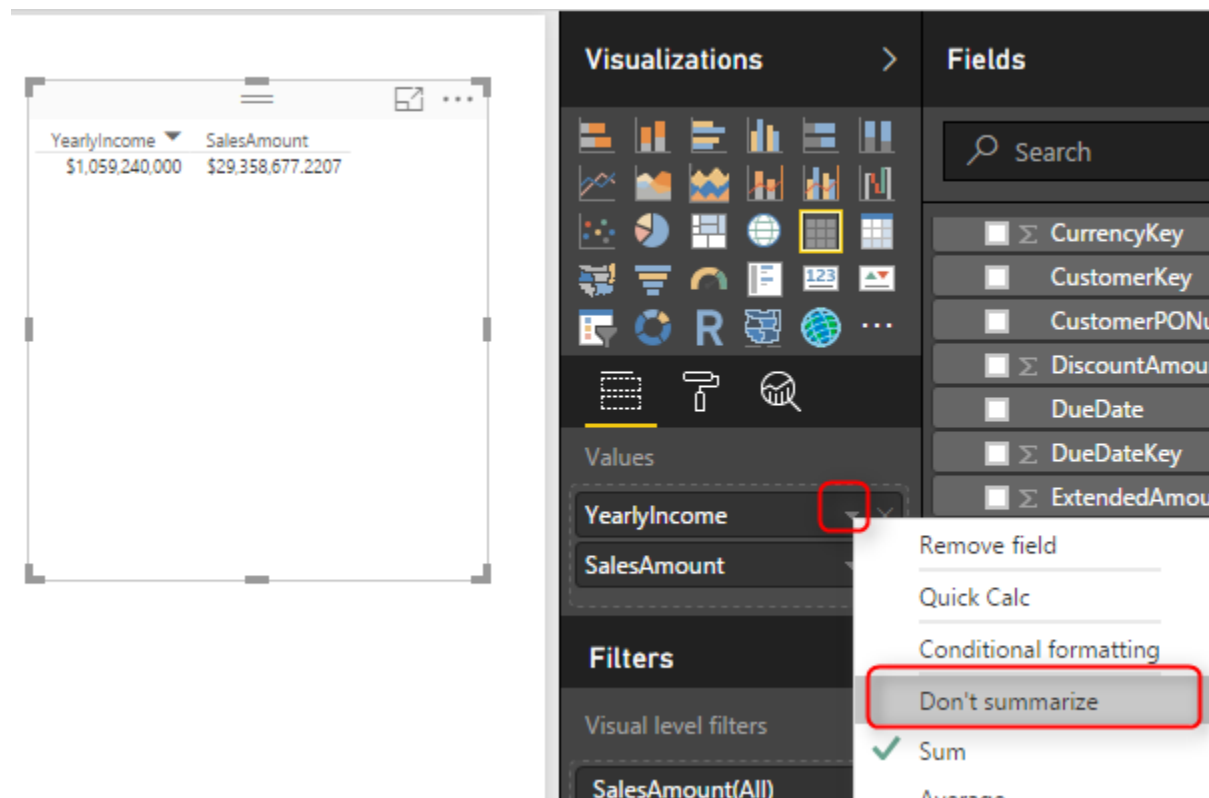
Get Data; Start with Get Data from AdventureWorksDW, and choose tables: FactInternetSales, DimDate, and DimCustomer.

Set the relationship between DimDate and FactInternetSales to be only based on OrderDateKey as Active relationship, remove inactive relationships between these two tables.

Banding or Binning

Binning is grouping a numeric field based on a division. This type of grouping is called Banding as well. For example you might have customers with different yearlyIncome range from \$10,000 to \$100,000 and you want to create a banding by \$25,000. This will generate 4 groups of yearly income for you. This is exactly what Binning in Power BI does. Let's look at the example.

Create a Table in Power BI Report and visualize YearlyIncome (from DimCustomer), and SalesAmount (from FactInternetSales) in it. Change the aggregation of YearlyIncome from Sum to Do Not Summarize as below

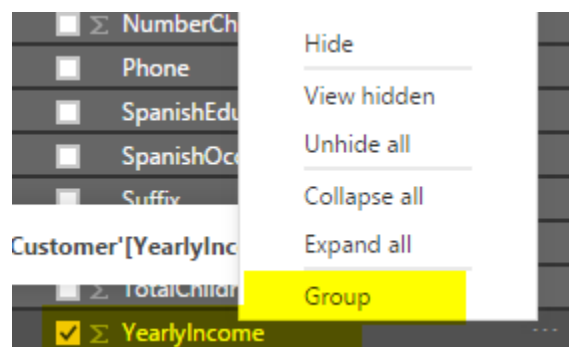


The screenshot shows the Power BI interface. On the left, a table is displayed with two columns: YearlyIncome and SalesAmount. The values are \$1,059,240,000 and \$29,358,677.2207 respectively. On the right, the Fields pane is visible. The 'Visualizations' pane shows a table icon selected. The 'Fields' pane lists various fields, including CurrencyKey, CustomerKey, CustomerPONu, DiscountAmount, DueDate, DueDateKey, and ExtendedAmount. The 'Values' section shows YearlyIncome and SalesAmount. A context menu is open for YearlyIncome, with options: Remove field, Quick Calc, Conditional formatting, Don't summarize (highlighted with a red box), Sum (checked), and Average.

This will visualize each yearlyIncome value with the sales amount associated to it.

YearlyIncome	SalesAmount
\$10,000	\$1,396,359
\$20,000	\$2,010,650
\$30,000	\$2,922,275
\$40,000	\$4,226,593
\$50,000	\$805,523
\$60,000	\$4,230,549
\$70,000	\$4,484,239
\$80,000	\$2,391,472
\$90,000	\$2,067,048
\$100,000	\$1,073,018
\$110,000	\$1,019,528
\$120,000	\$760,616
\$130,000	\$1,192,928
\$150,000	\$241,471
\$160,000	\$211,123
\$170,000	\$325,286
Total	\$29,358,677

Now let's create groups of people based on their yearly income categorized in buckets of \$30,000. Right click on the YearlyIncome column in the fields section and select Group.



In the Groups window make sure Bin is selected as the group type (by default for numeric or date/time columns this will be the default). and set the size to 30000. This is the size that values of YearlyIncome will be split based on it.

✕

Groups

Name

Field

Group type

Bin ▼

Binning splits numeric or date time data into equally sized groups. Enter bin size.

Bin size

30000

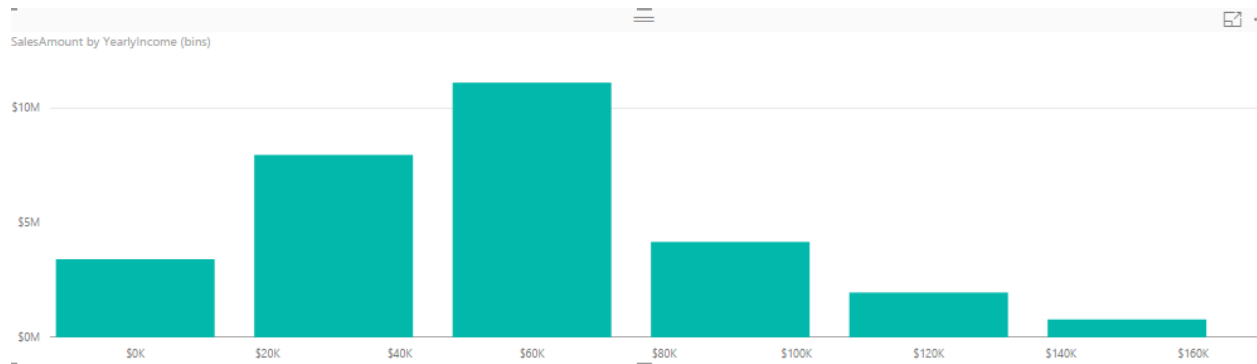
⬆ ⬇ ⬆

Press OK in the Group Window, and bring this new generated field (named YearlyIncome (bins)) to the table.

YearlyIncome (bins)	YearlyIncome	SalesAmount
\$0	\$10,000	\$1,396,359
\$0	\$20,000	\$2,010,650
\$30,000	\$30,000	\$2,922,275
\$30,000	\$40,000	\$4,226,593
\$30,000	\$50,000	\$805,523
\$60,000	\$60,000	\$4,230,549
\$60,000	\$70,000	\$4,484,239
\$60,000	\$80,000	\$2,391,472
\$90,000	\$90,000	\$2,067,048
\$90,000	\$100,000	\$1,073,018
\$90,000	\$110,000	\$1,019,528
\$120,000	\$120,000	\$760,616
\$120,000	\$130,000	\$1,192,928
\$150,000	\$150,000	\$241,471
\$150,000	\$160,000	\$211,123
\$150,000	\$170,000	\$325,286
Total		\$29,358,677

Color coding above shows that YearlyIncome values grouped by the division result of the value of yearly income by group size which is \$30,000. If you want less groups you can increase the group size, or if you want more detailed groups you can decrease the group size.

Now you can remove the YearlyIncome from table view, change the table to be Column Chart to see which group generates the most revenue.



This binning can be applied to datetime values as well. For example if you want to see revenue based on half year's banding, you can simply create a group with Bin size of 6 by Month. Let's see the example; Right click on FullDateAlternateKey column (from DimDate), and create the group;

Groups

Name: Field:

Group type:

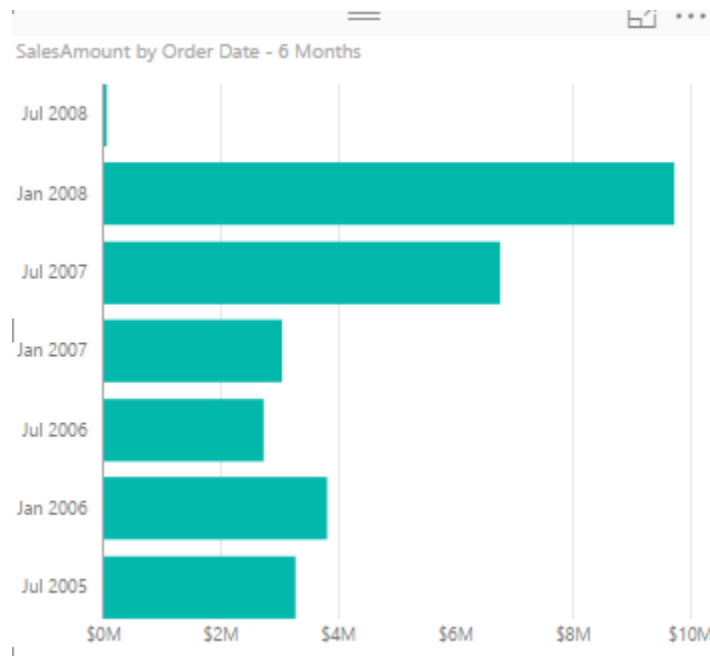
Binning splits numeric or date time data into equally sized groups. Enter bin size.

Bin size:

Bin type:

- Month
- Year
- Month
- Day
- Hour
- Minute
- Second

Now create a Bar chart with this new column (named Oder Date – 6 Months), and Sales Amount.



You can simply create other banding or binning for Years, Months, Days, Hours, and Minutes as well.

Grouping

You Can Create a group of multiple items in a visualization. This group can be simply created by Ctrl+Click and selecting items, then grouping them. As an example; create a Column Chart with English Education (from DimCustomer), and SalesAmount (from FactInternetSales).



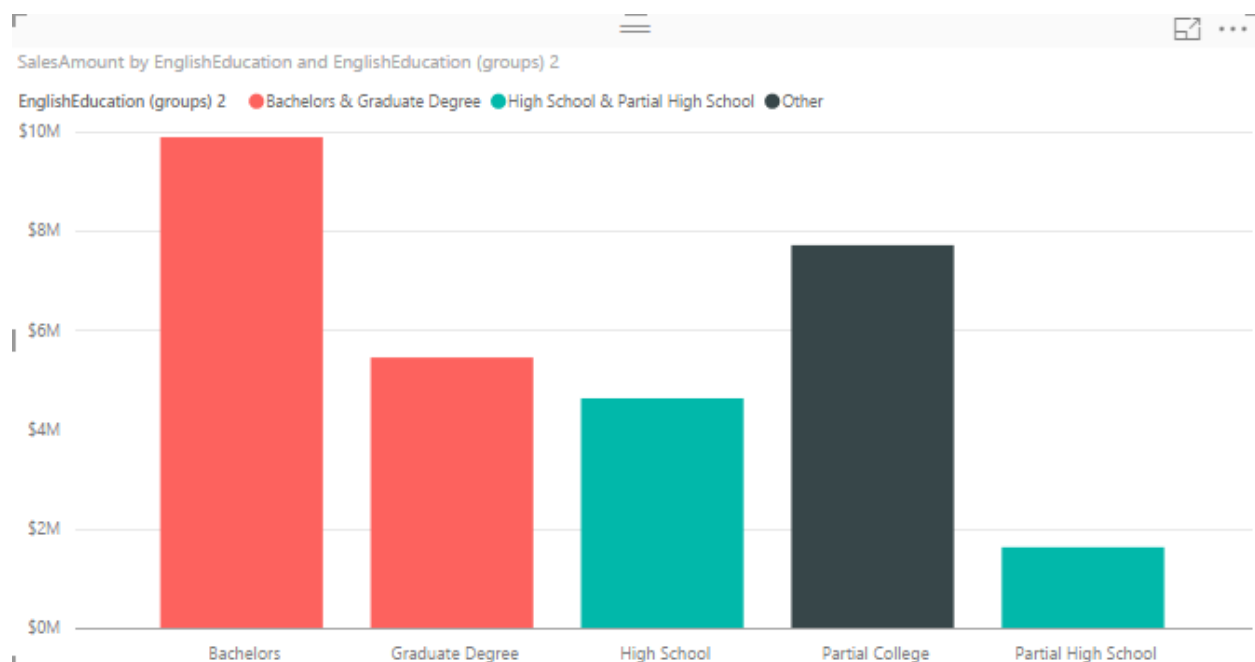
Now Ctrl Click and select High School and Partial High School, then right click and select Group.



You will see now them grouped together with a legend on the top of chart showing this group, and an "other" group.



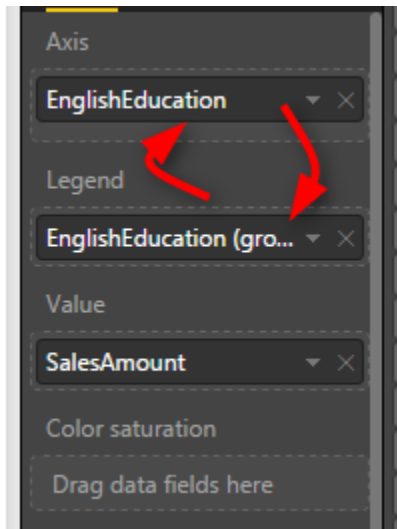
Now create another group for Bachelors and Graduate Degree in the same way.



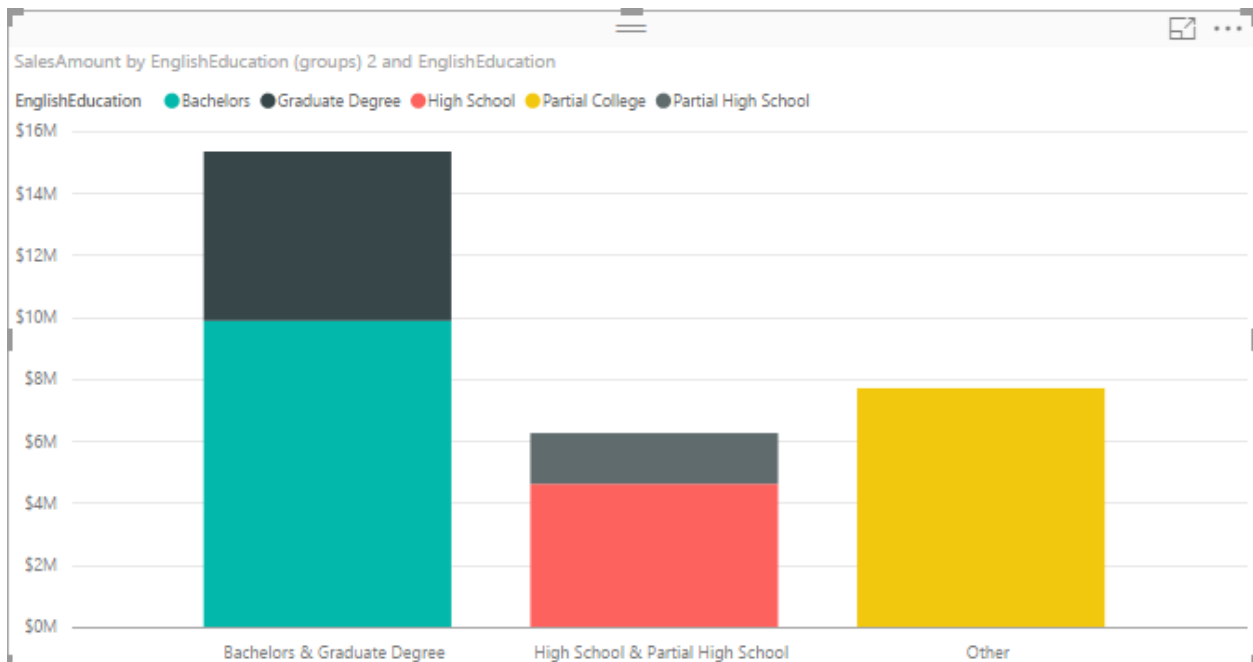
You have now three groups; higher education (Bachelors and Graduate Degree), College (Others), and High School (High School and Partial High School).

Once you have the grouping you can do many nice visualizations with that, an example can be creating a Stacked Column Chart with These Groups as the Axis, and the English

Education as the Legend. To do so; Change these two in the chart and change it to be Stacked Column Chart.



Now the chart will look like this;



This shows simply how much revenue each category generates in total, and also inside the category which sub category (English education) generates the most or least revenue.

Behind the Scene

What is happening behind the scene is that there is a **DAX calculated column** (calculation is not visible in Power BI) created for each binning. This was exactly the method of implementing such thing previously. There is also another method which is creating the banding through Power Query.

Grouping and Binning is part of your model. That means if you create another report or chart you can use the Group there. Even if you create calculated tables or any DAX calculations you can use these new Group columns there.

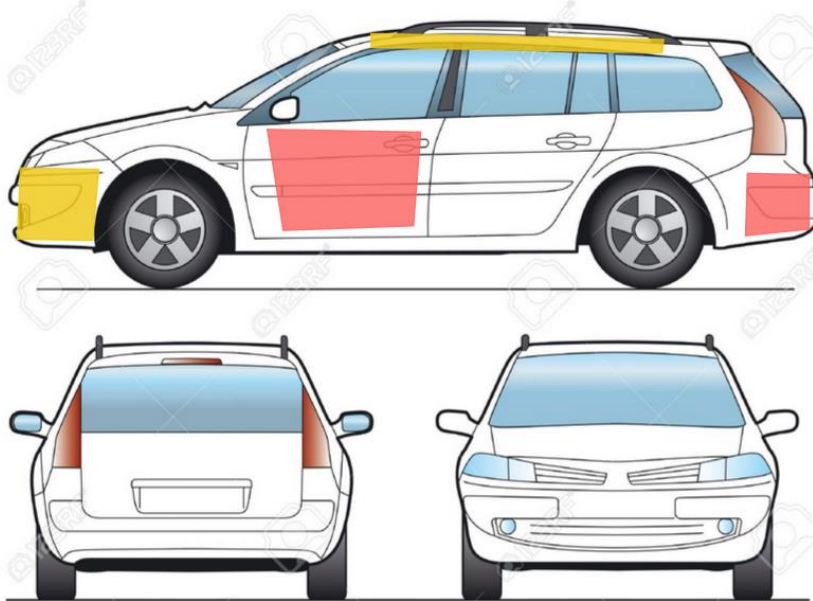
Summary

There are heaps of possibilities that these new grouping features bring to your visualization. Having these grouping as part of you model also is great to have unique experience in your reporting solution. However There are always complex grouping conditions that still need to write a bit of DAX code or Power Query M script.

Custom Visuals in Power BI; Build Whatever You Want

Published Date: March 9, 2016

Count of Rego and Count of Rego by Damage



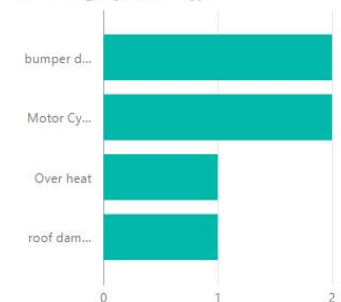
Accident Type



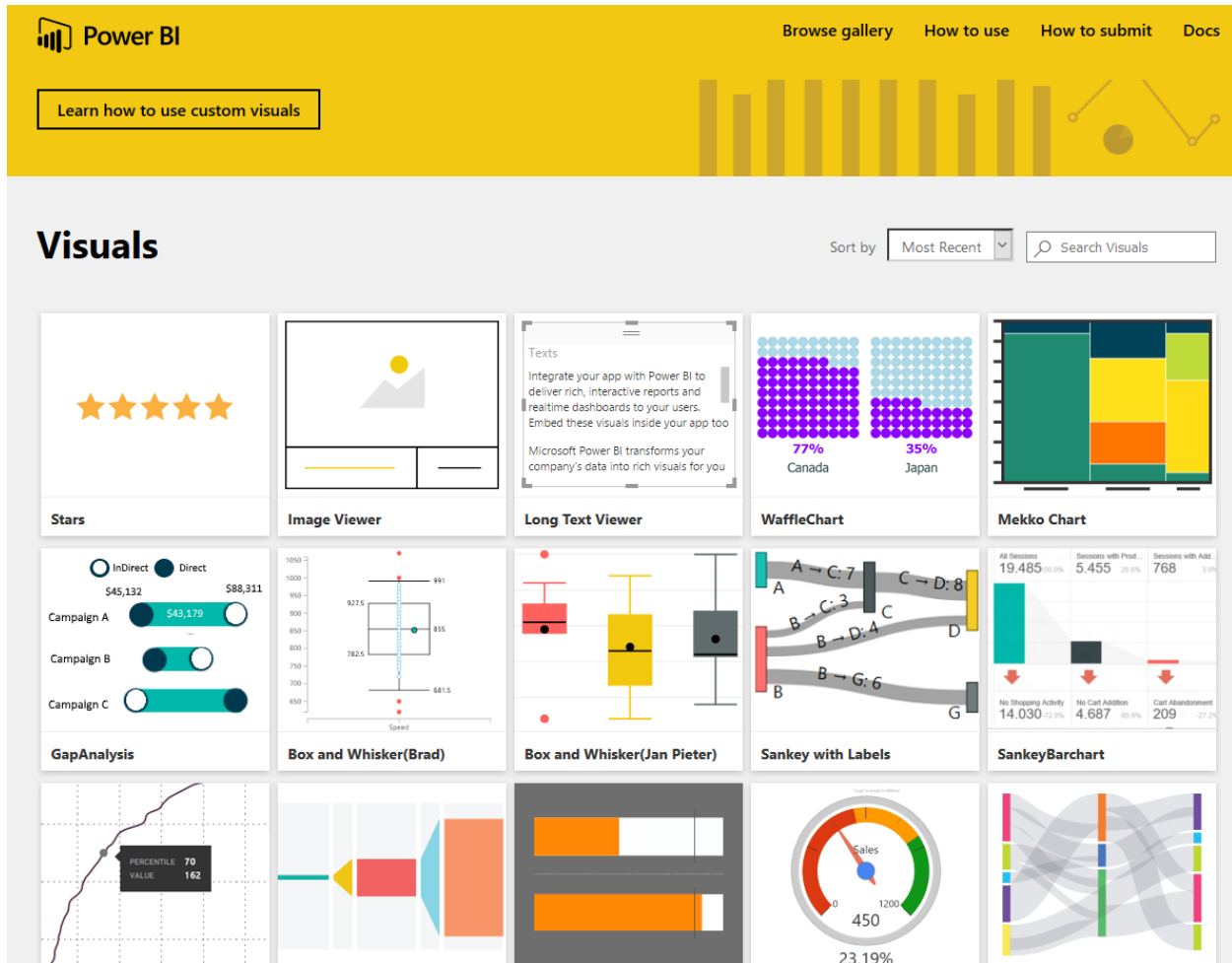
Severity



Count of Rego by accident Type




In previous posts of [Power BI from Rookie to Rock Star](#) you've learned that Power BI has a bunch of useful built-in visualizations such as bar chart, column chart, [waterfall chart](#), tree map, and many other visualizations. However there is always a requirement for new types of visualizations. You might want to visualize parts of the data in a custom image, or any other visualization that is not built-in. Fortunately Power BI gives you the ability to create your own custom visualizations. You can then share your visualization with others in your organization or even with the world through public [custom visuals website](#). In this post I'll show you some of the great examples of custom visuals in Power BI and how to use them. In future posts we will go through the scenario of creating one.



Synoptic Panel by SQL BI

This is one of the coolest custom visuals I've seen so far. The reason is that this custom visual has a customization in it! with this visualization you can define regions in any picture or images, and map data points to the image in your Power BI report. The image can be everything; human body, airplane seat layout, shop floor layout or football field. You choose the image you want, then you define regions. Let's have a closer look at this visual.

You can download the latest version of Synoptic Panel from [Power BI visuals gallery](#) or from the [Synoptic Design website](#). Please note that custom visuals might be developed by third parties not Microsoft, so you are using them at your own risk (I mean in terms of support if an issue happens). Synoptic Panel is a custom visual built by [SQLBI](#) company. After downloading this visual you will have a *.pbviz file.



Stock room

Bluetooth He... MP4&MP3

Recordin... Camcorders Camera... Digital Came...

Water Heaters Microwaves

Movie DVD Refrigerators

Washers & Dr...

Coffee Mach...

Fans

Lamps

Desktops

Laptops

Home & Office

Smart phones...

Touch Screen...

Computers Ac...

Cell phones Accessories

VCD & DVD

Televisions

Registers

Entry

Cell phones Computers Games and ... Home Appl... Music, Mov... TV and Video

SynopticPanel by SQLBI

Published by [SQLBI](#)

The Synoptic Panel connects areas in a picture with attributes in the data model, coloring each area with a state (red/yellow/green) or with a saturation of a color related to the value of a measure. Starting from any image, you draw custom areas using <https://synoptic.design>, which generates a SVG file you import in the Synoptic Panel. You can visualize data over a map, a planimetry, a diagram, a flow chart.

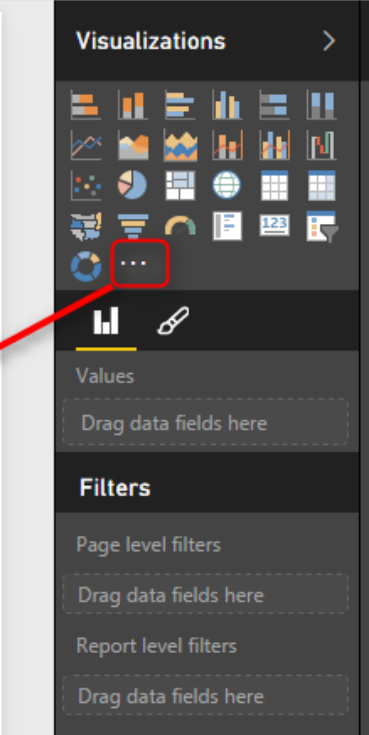
[License](#) [Privacy Statement](#)

[Download Visual](#)

Version 1.2.6 — Date added 12/8/2015

[Contact Author](#) [Support](#) [GitHub](#)

You can import that file into Power BI desktop with clicking on ellipsis button in visualization pane and choosing the *.pbiviz file.



Visualizations

Caution: Import Custom Visual

Custom visuals are not provided by Microsoft and could contain security or privacy risks. Only import a custom visual if you trust its author and source.

[Learn More](#)

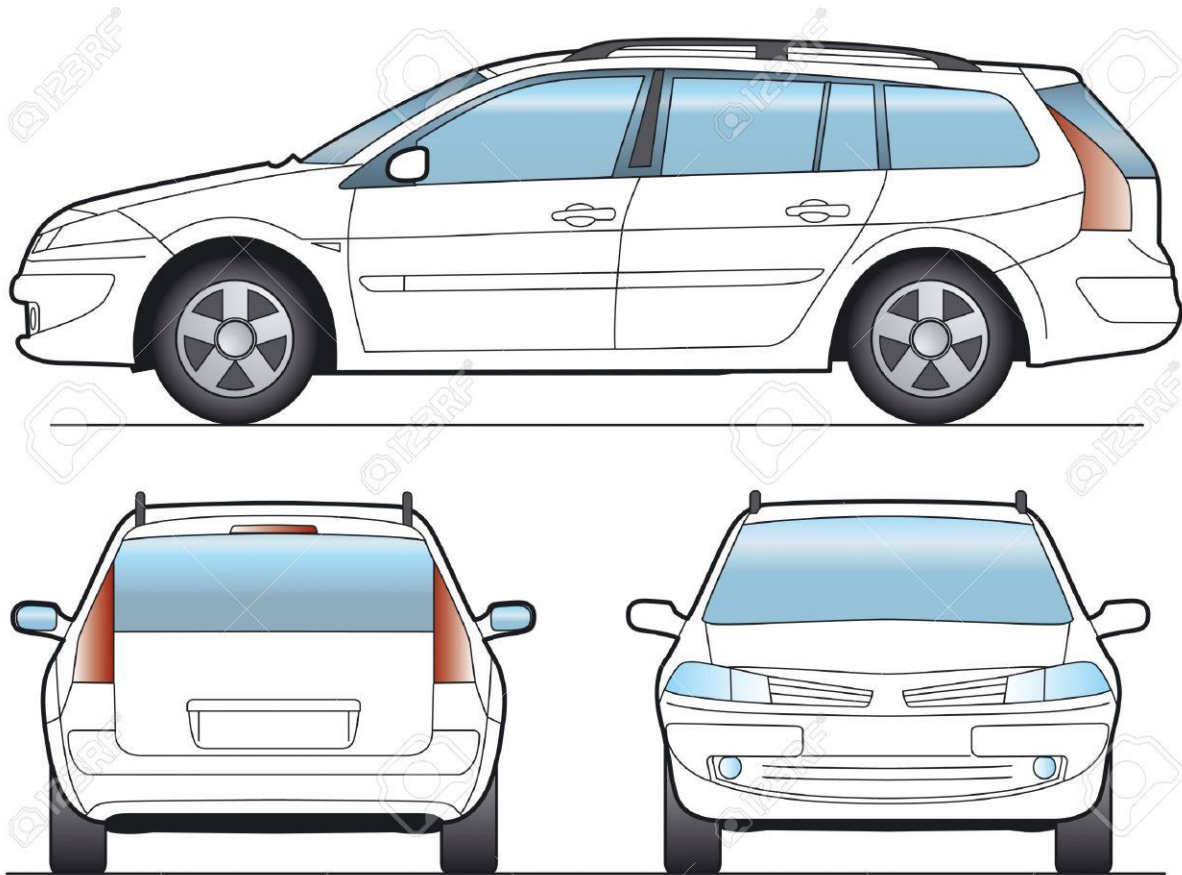
[Import](#) [Cancel](#)

After importing you will see the visual listed in visualization pane.

Now let's do an example with that. Consider we have a dataset of car accidents which has date of accident, registration number of the car, car part which broke in the accident, accident type and severity of the accident. Here is the example data set:

Date	Rego	Damage	Severity	accident Type
1/15/2015	wht345	roof	High	roof damage
6/25/2015	dsg562	bumper	Low	bumper damage
12/10/2015	lkn282	door	Low	Motor Cycle accident
10/23/2015	HGB345	front bumper	Medium	Over heat
12/12/2015	lkn265	door	Low	Motor Cycle accident
3/25/2015	dsg545	bumper	Low	bumper damage
10/23/2015	HGB345	front bumper	Medium	Over heat
12/12/2015	lkn265	bumper	Low	Motor Cycle accident
3/25/2015	dsg545	bumper	Low	bumper damage

As you see in the dataset above the Damage column shows which part of car the damage has been occurred. So a very useful visualization is having a layout of car exterior design (considering that damage only happened on the exterior for simplicity). With a simple search in google I've found image below:



Source of the image: http://www.123rf.com/photo_1648382_sw-car-layout-for-presentation-vector.html

What I need to do is to define regions on this image. Synoptic Panel also has a designer tool that allows me to draw regions! so let's do that. Go to [Synoptic Designer](https://synopticdesigner.com/) website and upload your image there.



SYNOPTIC DESIGNER FOR POWER BI

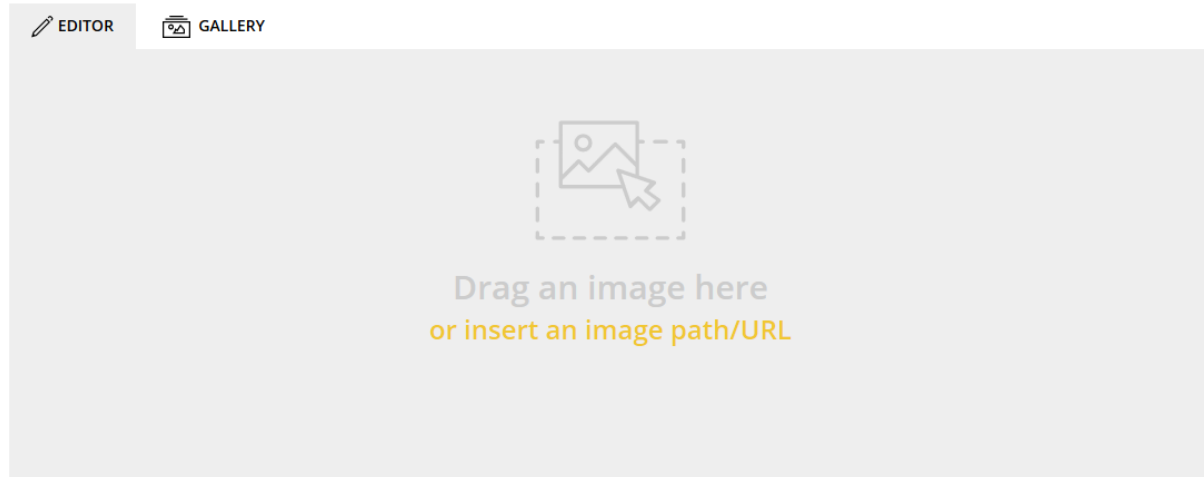
Synoptic Designer is the companion tool of **Synoptic Panel by SQLBI**, grand prize winner of the Power BI Best Visual contest. It allows you to draw custom areas over any map image and export it as SVG file to Power BI. Then, through our Synoptic Panel, you will be able to color the different areas based on your data.

Read [this post](#) on Microsoft blog or [watch a step-by-step video](#) to get more information.

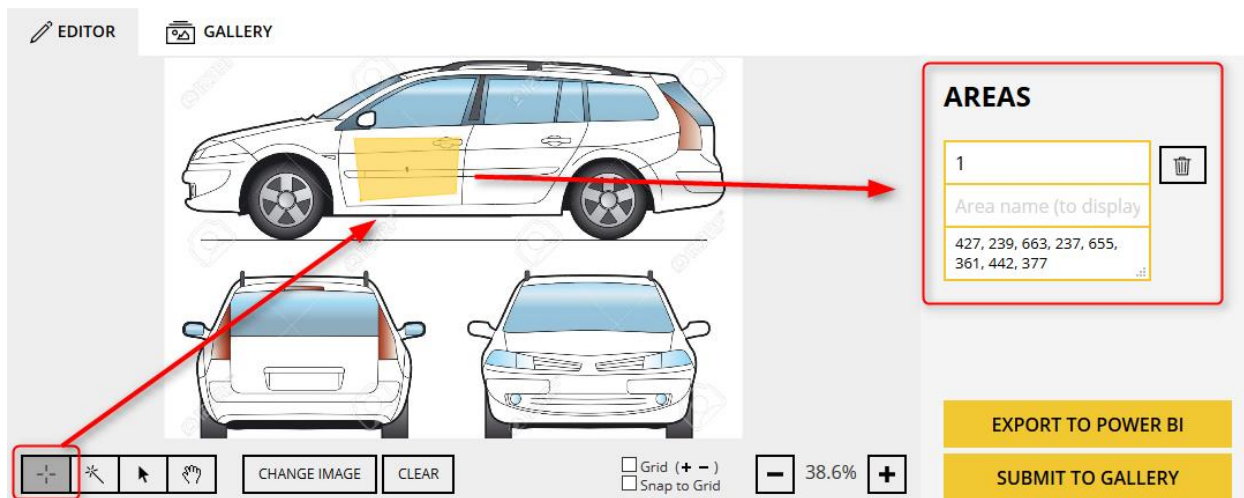
SYNOPTIC PANEL BY SQLBI



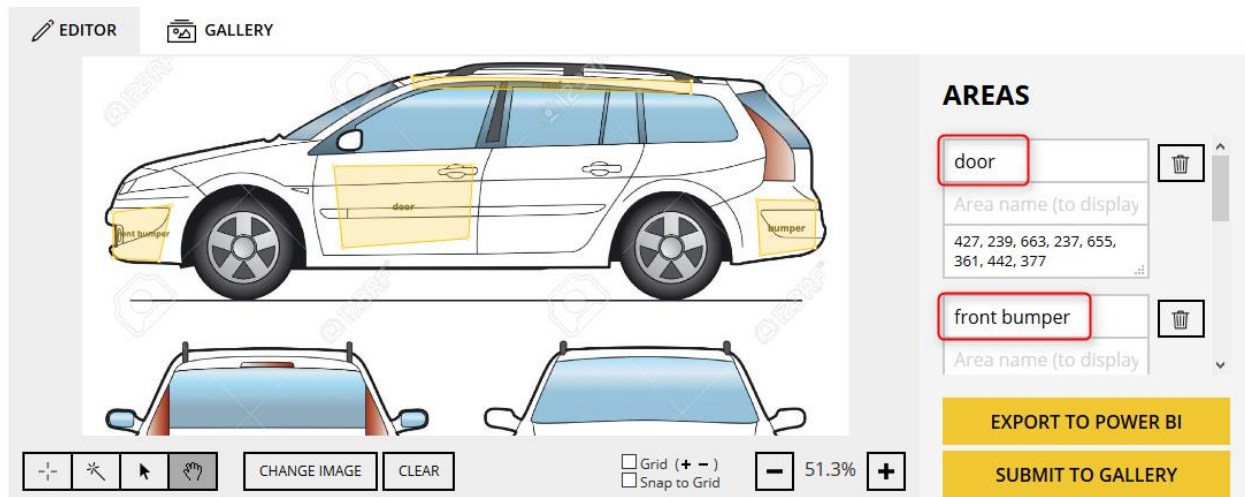
Get the latest version
v1.2.6 - released on Jan 30, 2016
([Github repo](#) | [Power BI gallery](#))

[RECEIVE UPDATES](#)


add the image there, and then use the designer tool for drawing an area. after drawing the area you will see area highlighted and listed on the right hand side.



Define other areas (for bumper, roof...). and name each area as the values you have in your dataset's Damage column.



Now you can export this design into your Power BI with clicking on Export To Power BI. this will creates a *.svg file for you to use in the Synoptic Panel visual.

EXPORT TO POWER BI

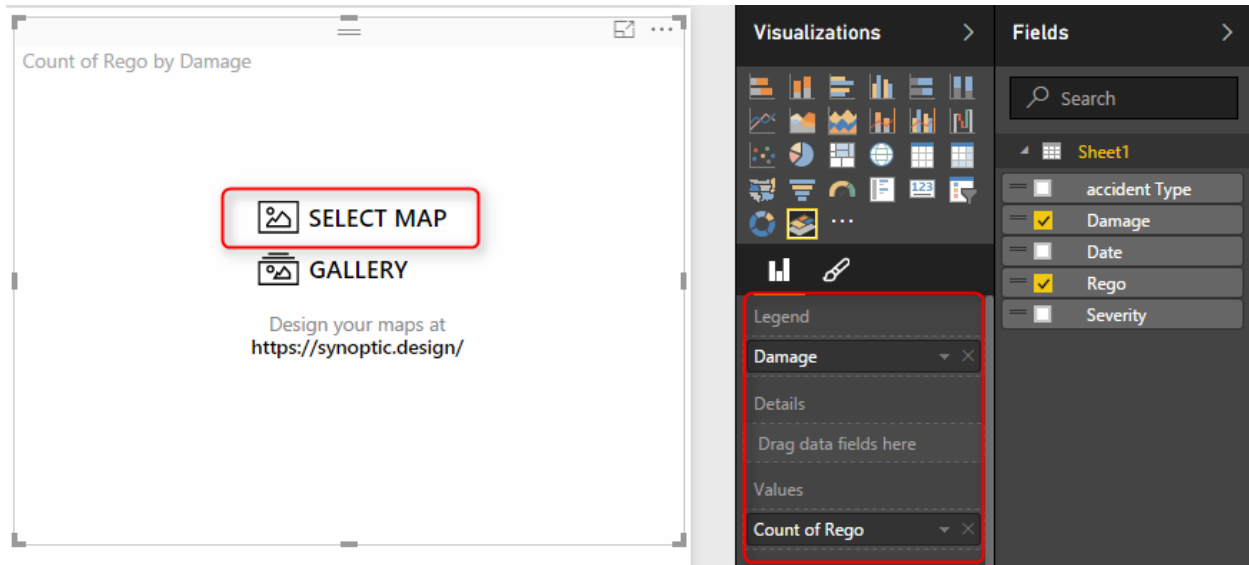
Fantastic work! Now you can import the following file in Synoptic Panel by SQLBI in Power BI and start highlighting your data:



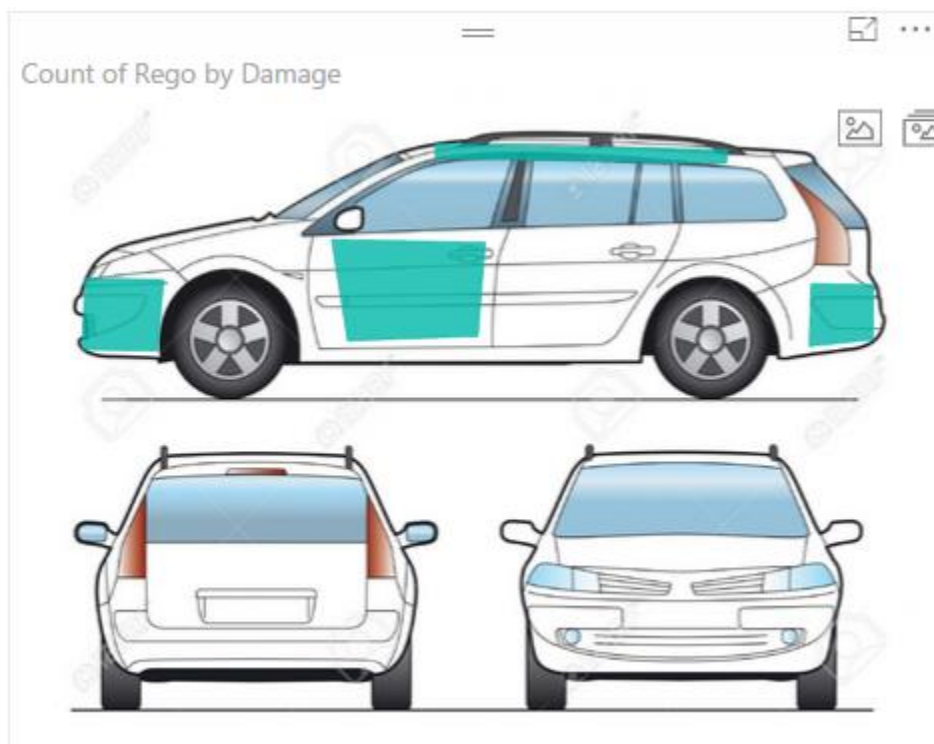
DOWNLOAD SVG

Note that it is not the original image you uploaded, but a generated SVG file that includes definition of areas.

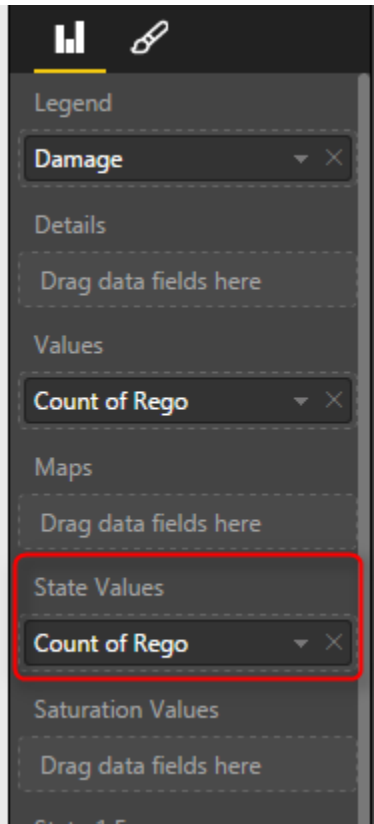
Import the *.svg file into Power BI desktop's Synoptic Panel visual. As you can see I've mapped Damage field to Legend of the custom visual and count of Regos to values field.



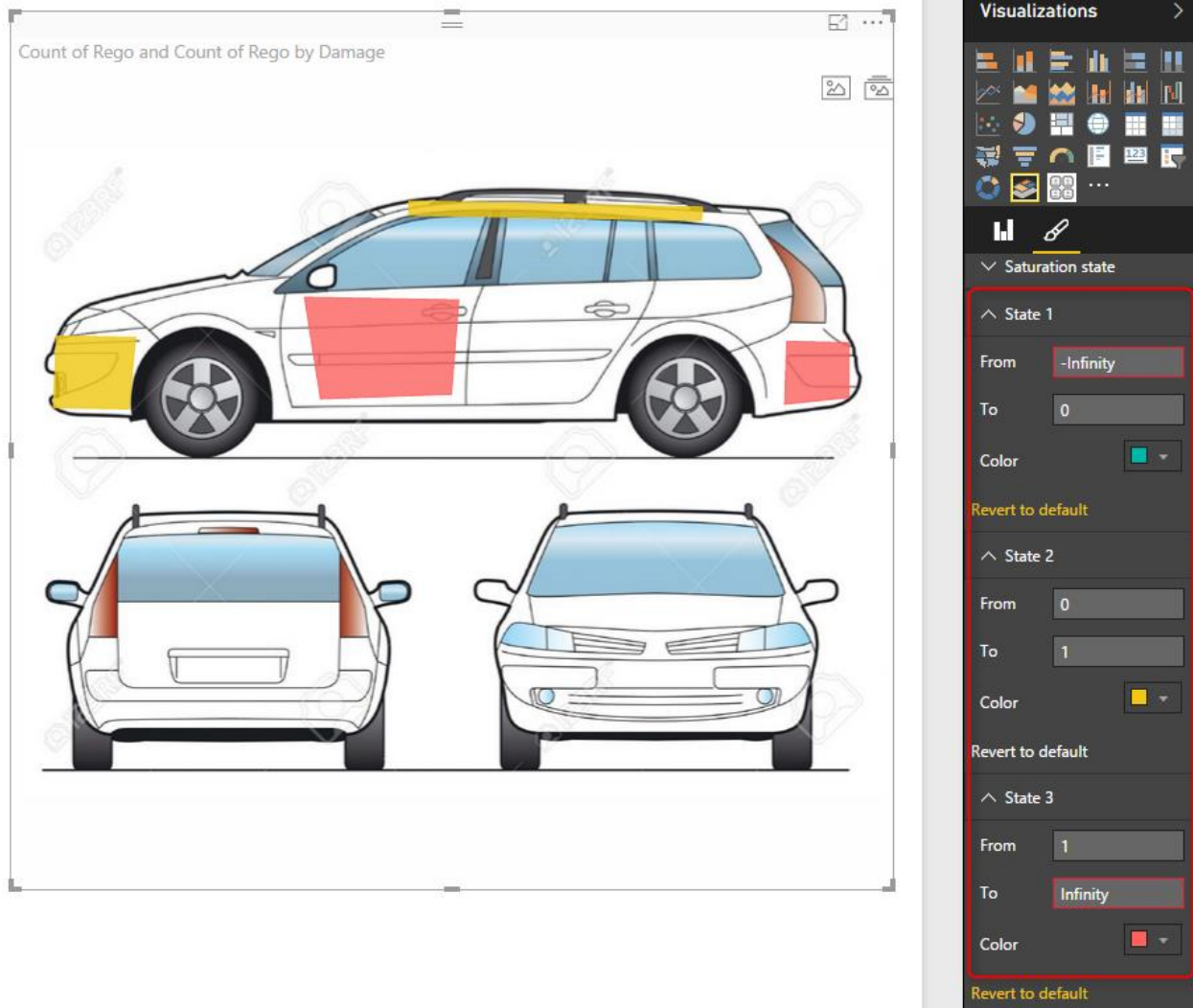
After adding the *.svg file as the map you will see regions highlighted. This shows that there are some values for those regions.



You can set a color state for this visual. Before doing this you should set a field as the state field for this visual;



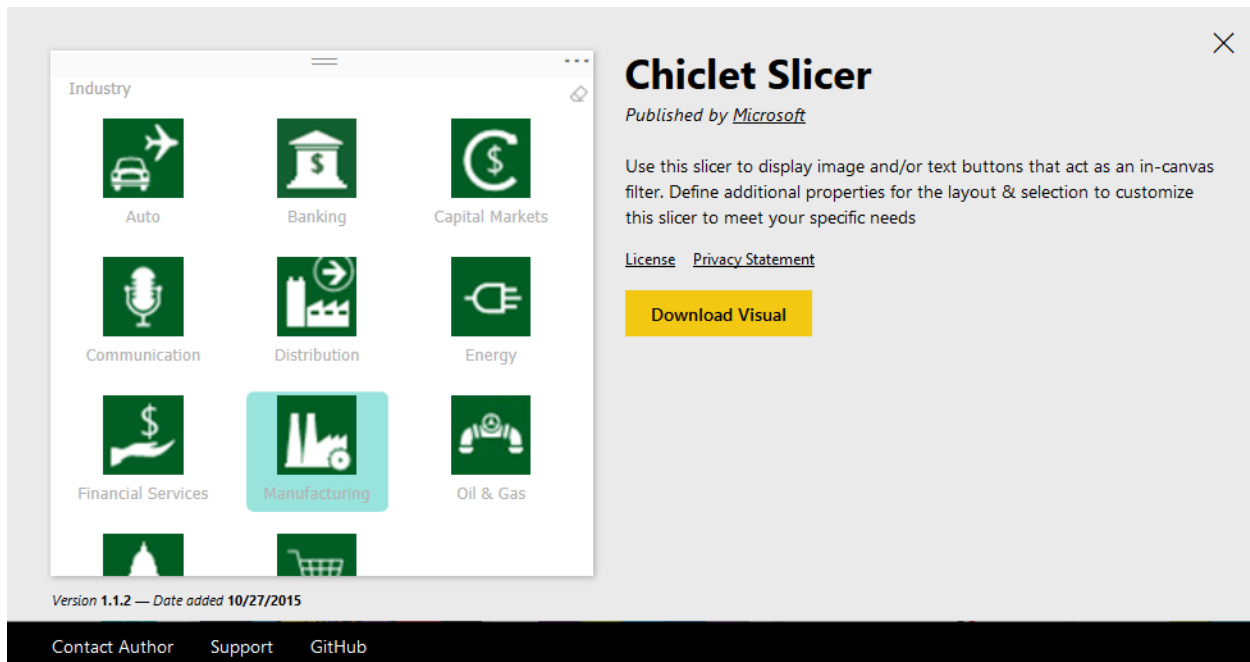
Then you define rules for coloring: For example of value is between 0 to 5 then state color one with Green color. if it is between 6 to 10 then state color 2.... Here is what I defined:



Nice visualization, isn't it? There are some other customization that you can apply with this visual. I'll leave that part to you for further studying if you are interested. [Watch SQLBI YouTube video](#) which illustrates some of this visual's features. There are also some per-built maps for this visual that you can use if you want.

Chiclet Slicer

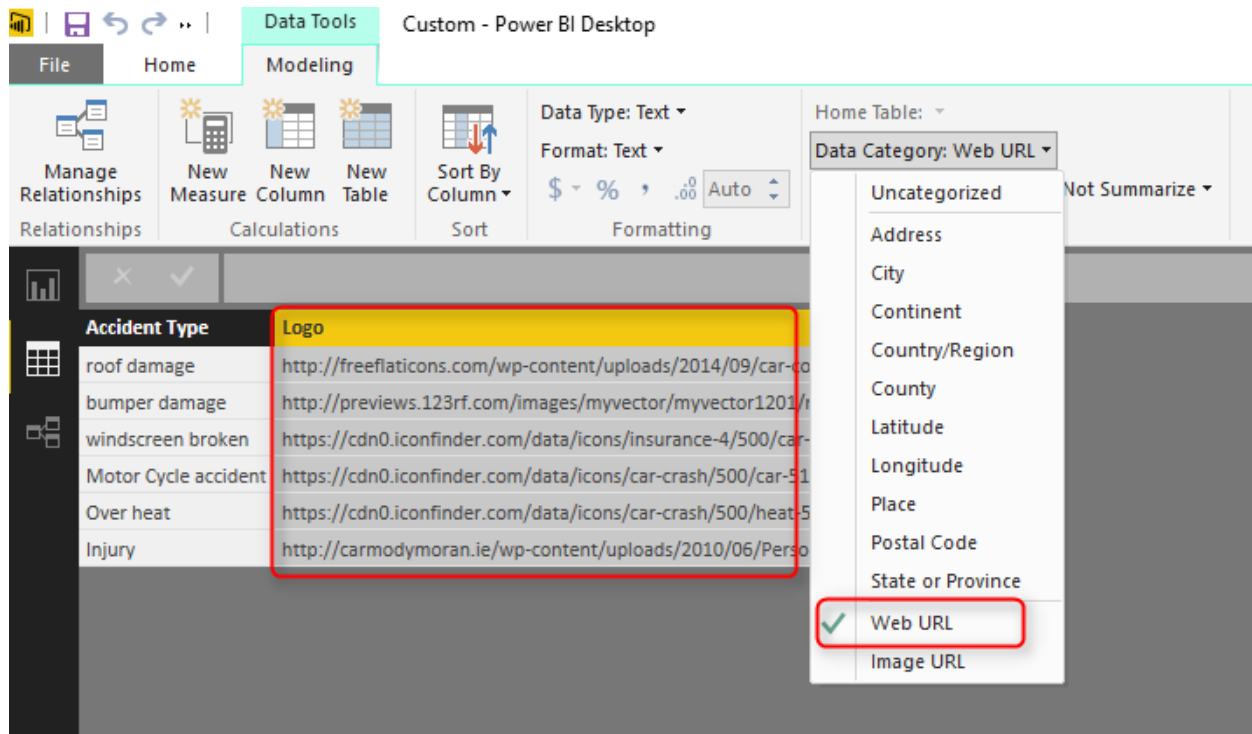
The other custom visual I want to talk about in this post is Chiclet Slicer. This is a custom visual built by Microsoft itself. The default slicer in Power BI is very basic, you can't do much formatting on it. Chiclet slicer allows you to do text formatting, box formatting, and also adding images to your slicer items.



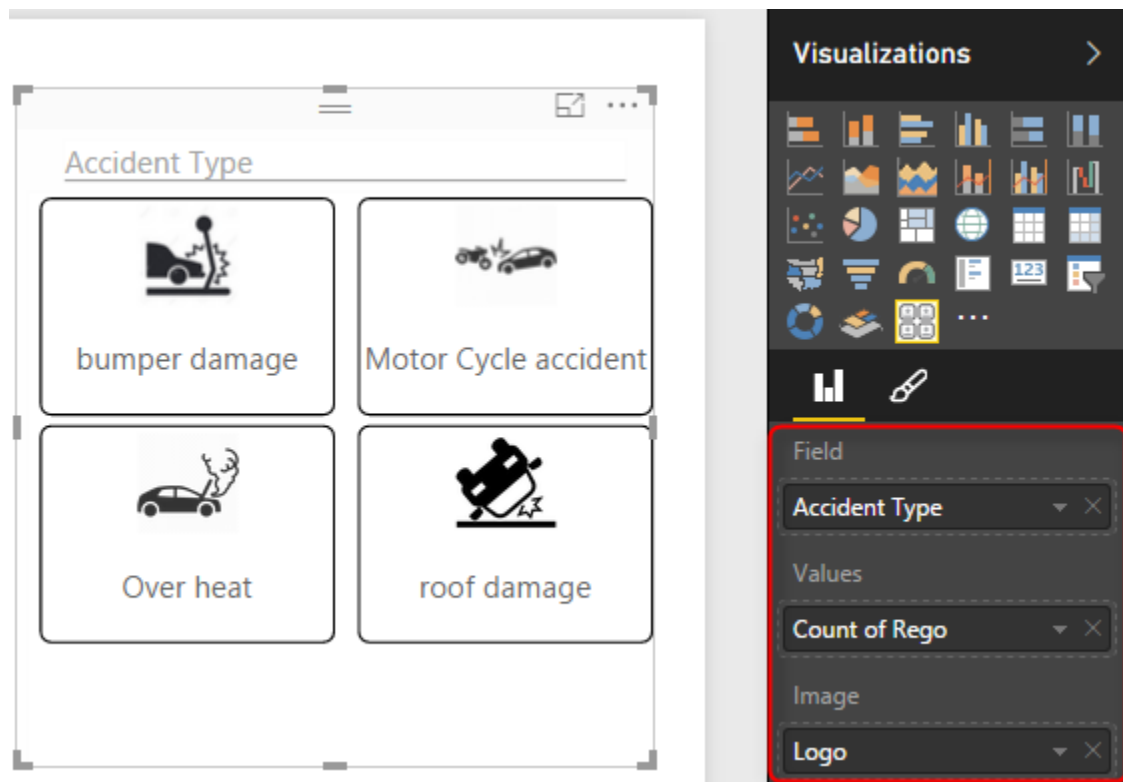
Let's add this slicer to our existing car accident example. As you've seen in dataset above we have accident type field which contains values such as roof damage, motor Cycle accident, over heat and etc. It is good to have a slicer that shows these accident types as images. So what I've done was creating another spreadsheet with accident types and a field called logo which is a URL for an image related to each item.

Accident Type	Logo
roof damage	http://freeflaticons.com/wp-content/uploads/2014/09/car-copy-141057655684kng.png
bumper damage	http://previews.123rf.com/images/myvector/myvector1201/myvector120101576/12126552-Vector-illustration
windscreen broken	https://cdn0.iconfinder.com/data/icons/insurance-4/500/car-512.png
Motor Cycle accident	https://cdn0.iconfinder.com/data/icons/car-crash/500/car-512.png
Over heat	https://cdn0.iconfinder.com/data/icons/car-crash/500/heat-512.png
Injury	http://carmodymoran.ie/wp-content/uploads/2010/06/Personal-Injury-icon-e1333481462230-210x300.png

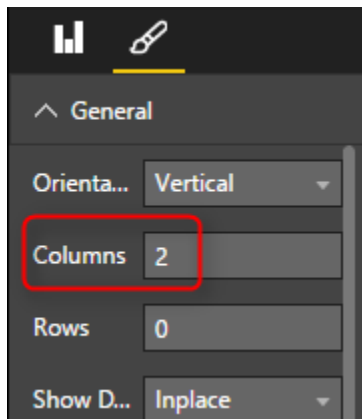
I've then loaded this dataset into my Power BI model and created relationship based on AccidentType. Then I've changed the Data Category of Logo field in my Data tab to Web URL.



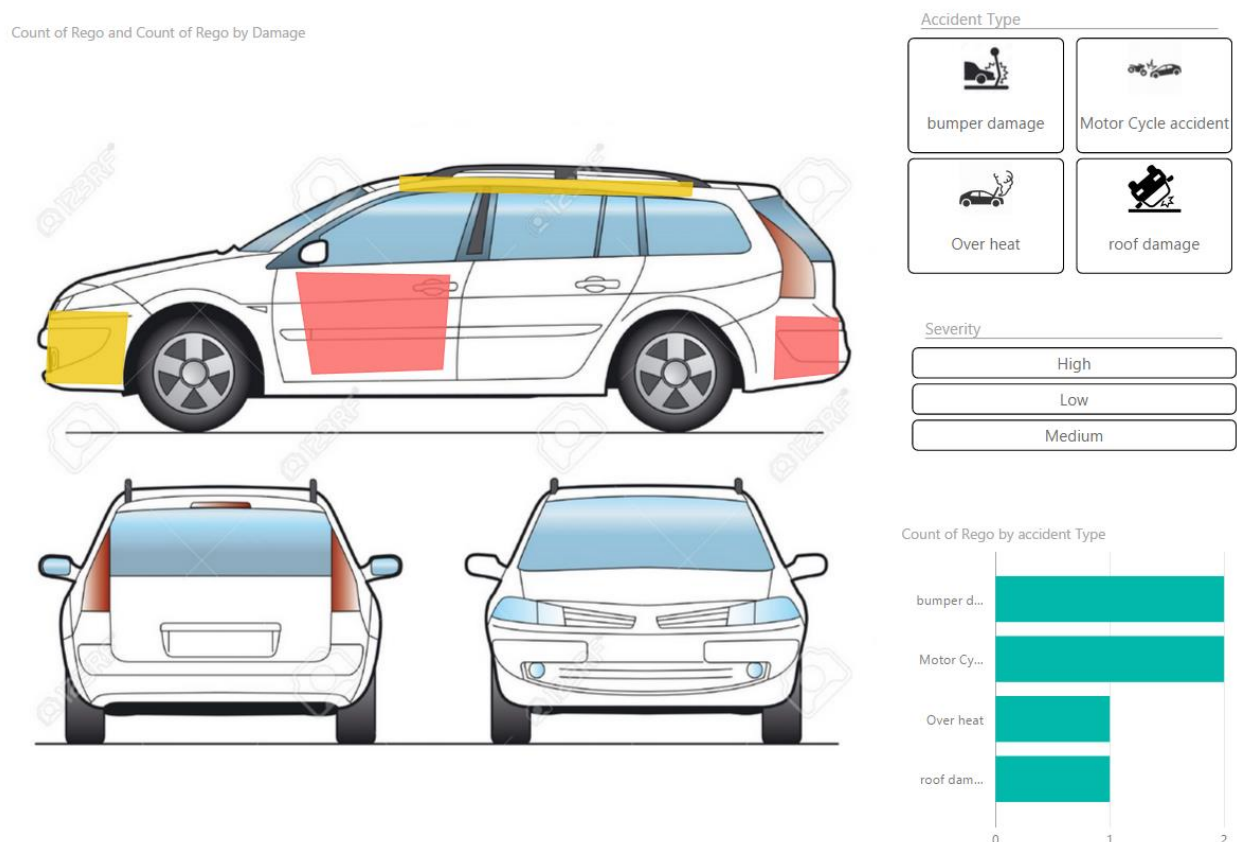
Now in the Chiclet Slicer custom visual I set fields as accident type, and Image as Logo, and count of rego as Values. Here is the result.



For the result above I've also configured number of columns in my chiclet slicer to be 2.



Even if I don't have an image for chiclet slicer still I can use it as it shows text field in a way much better. I've used it for Severity field as well. and the final result is here:

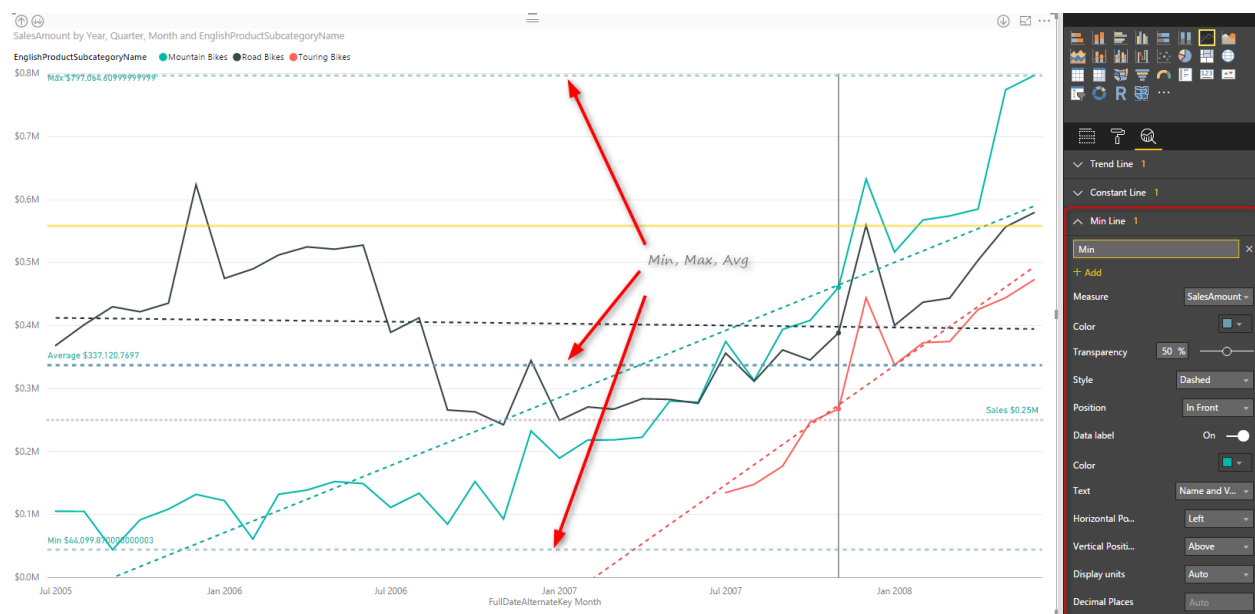


You can see that adding two custom visuals made my report much better in terms of both look and feel and also understanding the insight better. In future posts of [Power BI](#)

[from Rookie to Rock Star book](#) I'll explain some other useful custom visuals and the method of developing your own custom visual.

Power Behind the Line Chart in Power BI; Analytics

Published Date: August 28, 2016



Line chart is one of the most basic charts that you can see in every visualization tool. Power BI like other visualization tools has the line chart visualization which is good for showing one or two measure's value across an axis which can be a category, time period and etc. [Recently](#) Power BI line chart added the feature to create dynamic reference lines and the Analytics pane which brings more power into this useful chart. In this post I'll show you an example of using these features in Power BI line chart. If you are new to Power BI or you want to learn more make sure to read [Power BI; from Rookie to Rock Star](#).

Prerequisite

For running the sample in this post you will need to [download](#) and install AdventureWorksDW sample Microsoft SQL Server database.

Introduction to line types in the line chart

Value Line

Line chart can show one or more measures as measures, such as Sales Amount, Total Costs, Quantity of the goods sold and etc.

Trend Line

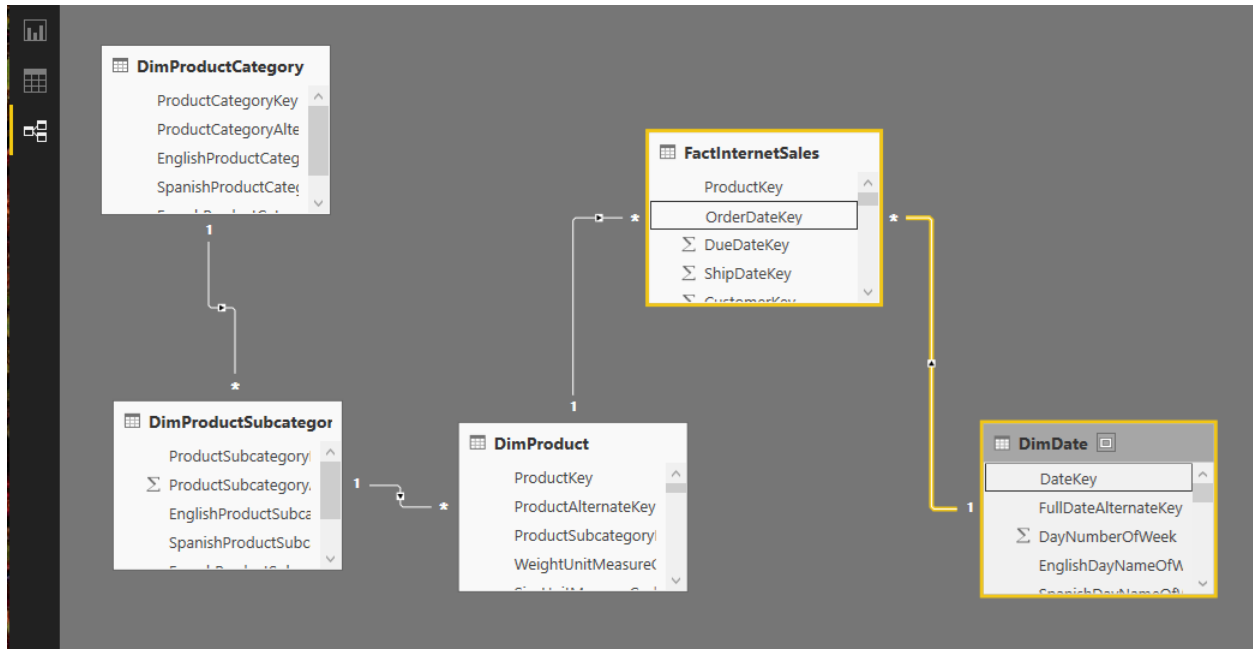
Depends on the variety of values across a time period, Line chart can illustrate a straight line as a trend. This trend is good to understand how in overall products are selling, is revenue going up or down for example.

Reference Line

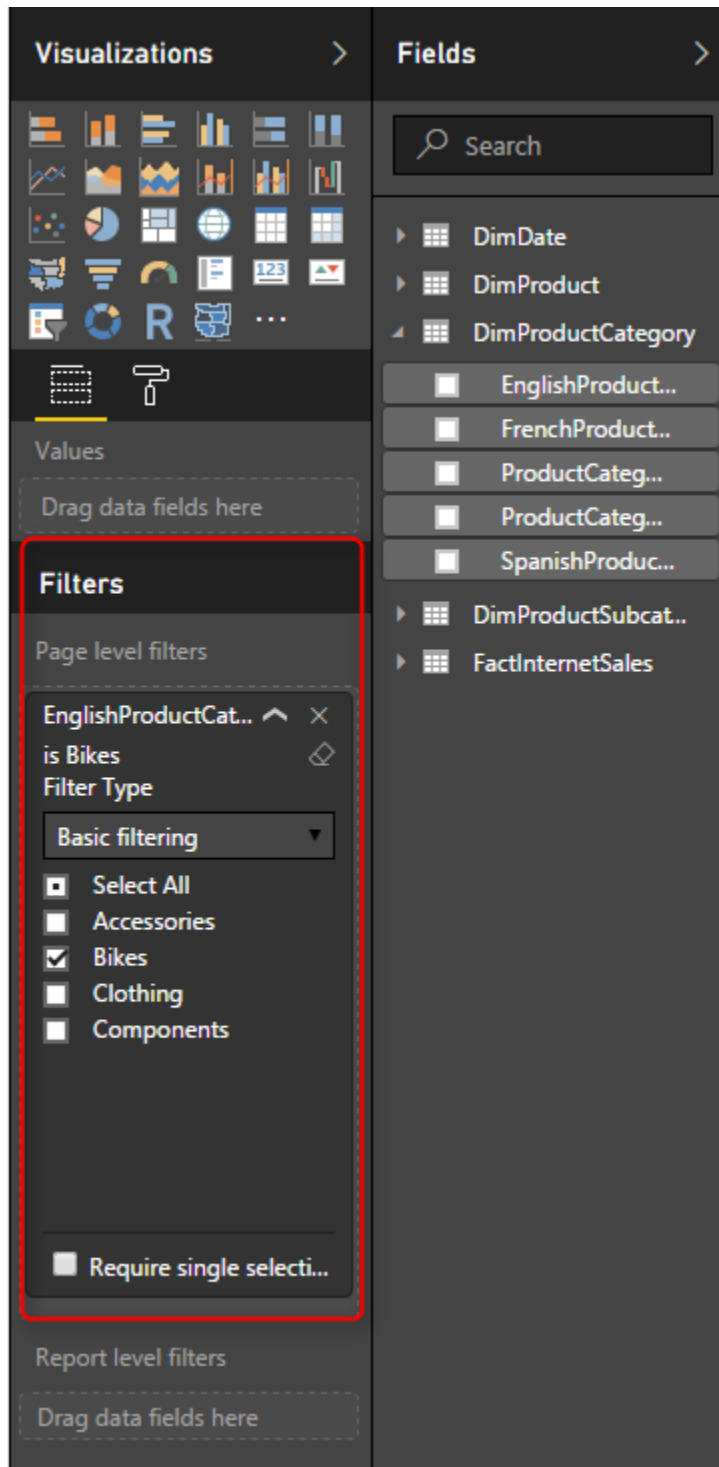
You might want to define minimum, maximum, average, or median values for your line chart as separate lines, and compare values lines with these lines. These are reference lines which can be dynamically created based on value of measures in the chart.

Building a Line Chart

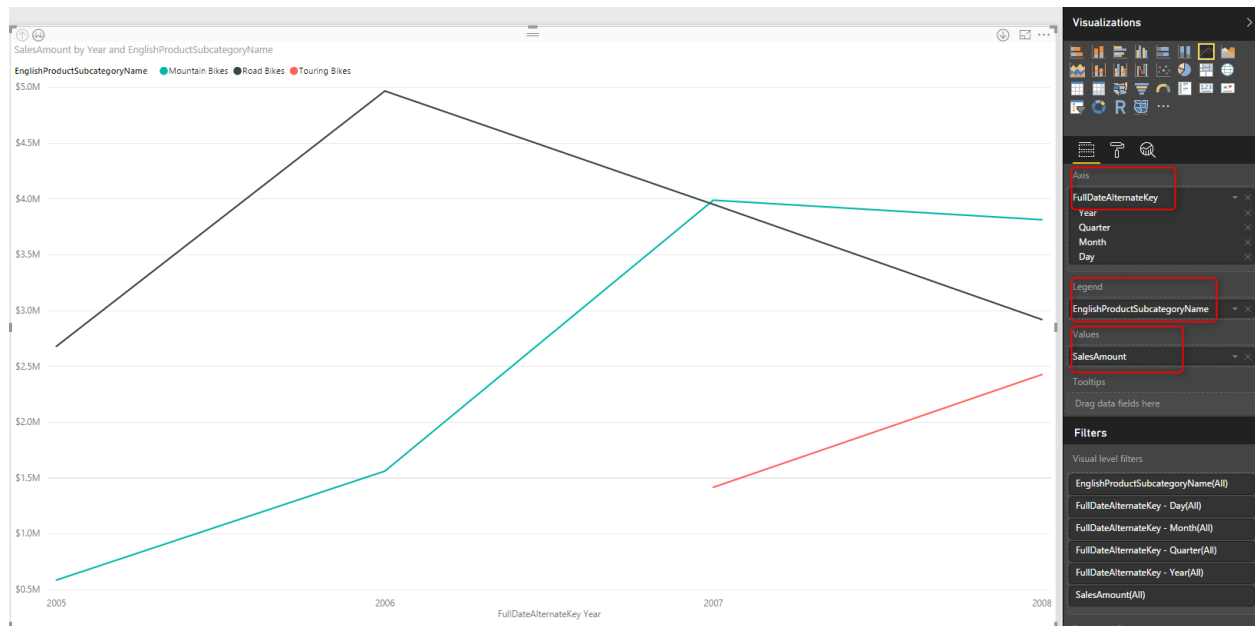
Create a new Power BI solution with Get Data from AdventureWorksDW SQL Server database, and choose tables as DimDate, DimProductCategory, DimProductSubCategory, DimProduct, and FactInternetSales. Click on Load, and in the relationship tab. remove extra relationships between FactInternetSales and DimDate to be only one active relationship based on OrderDateKey.



Go to Report tab, and add EnglishProductCategory to the Page Level Filters. Filter data to be only Bikes.

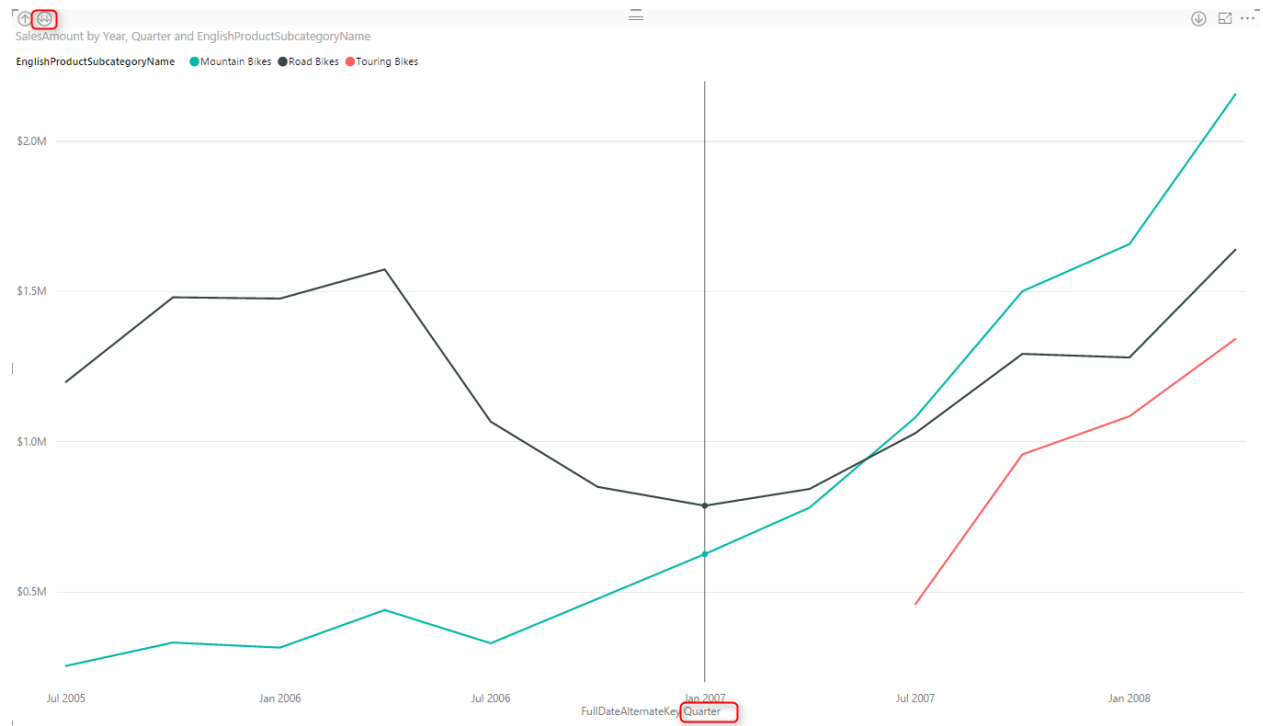


Page level filter will filter all visualizations in this page to only show Bikes information. Now add a Line Chart from visualization pane. Set the value to be SalesAmount (from FactInternetSales), Axis to be FullDateAlternateKey (from DimDate), and EnglishProductSubCategory (from DimProductSubCategory).

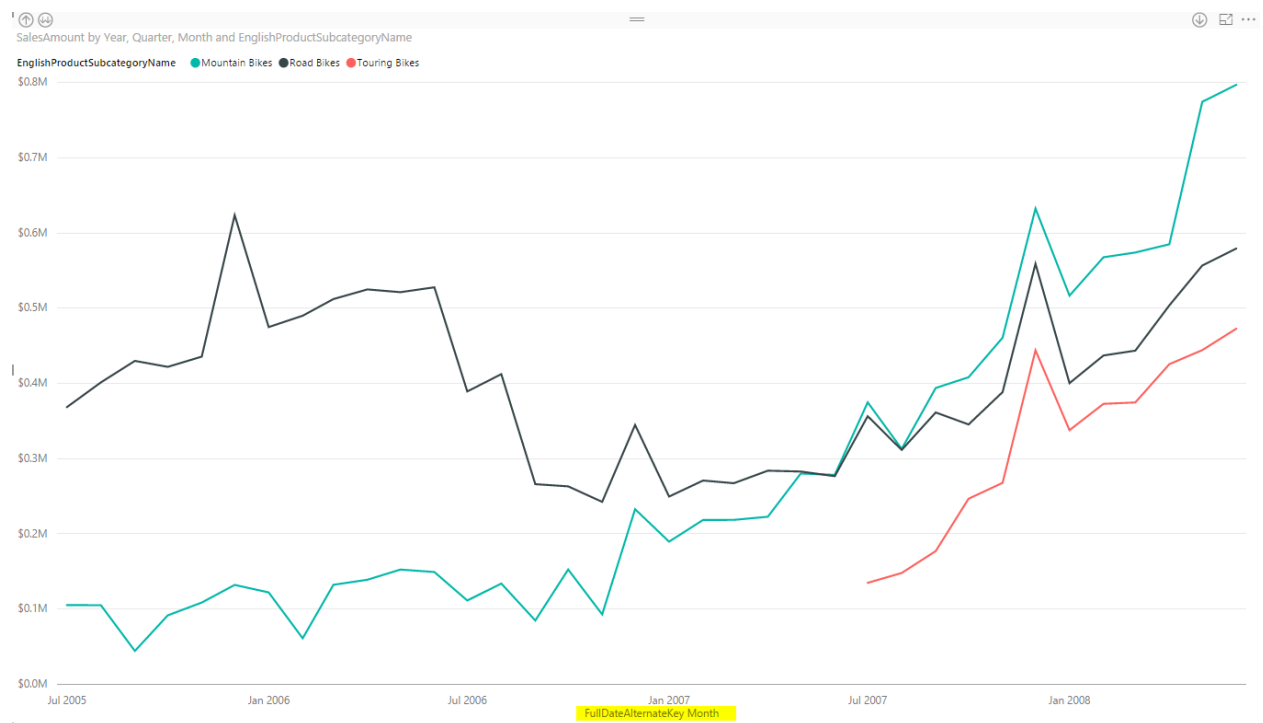


This line chart now simply shows three product sub categories of Bike; Mountain Bikes, Touring Bikes, and Road Bikes. FullDateAlternateKey is a full date column which in this view showed as a Year level. This behavior is because Power BI automatically add Date Hierarchy to date fields. If you want to change this behavior you can go to options and do so.

Let's now drill down into Quarter by clicking on the Drill Down button on the top left hand side corner of the line chart. You can see after drill down in the button center of the chart it shows the level of drill down is on Quarter now.



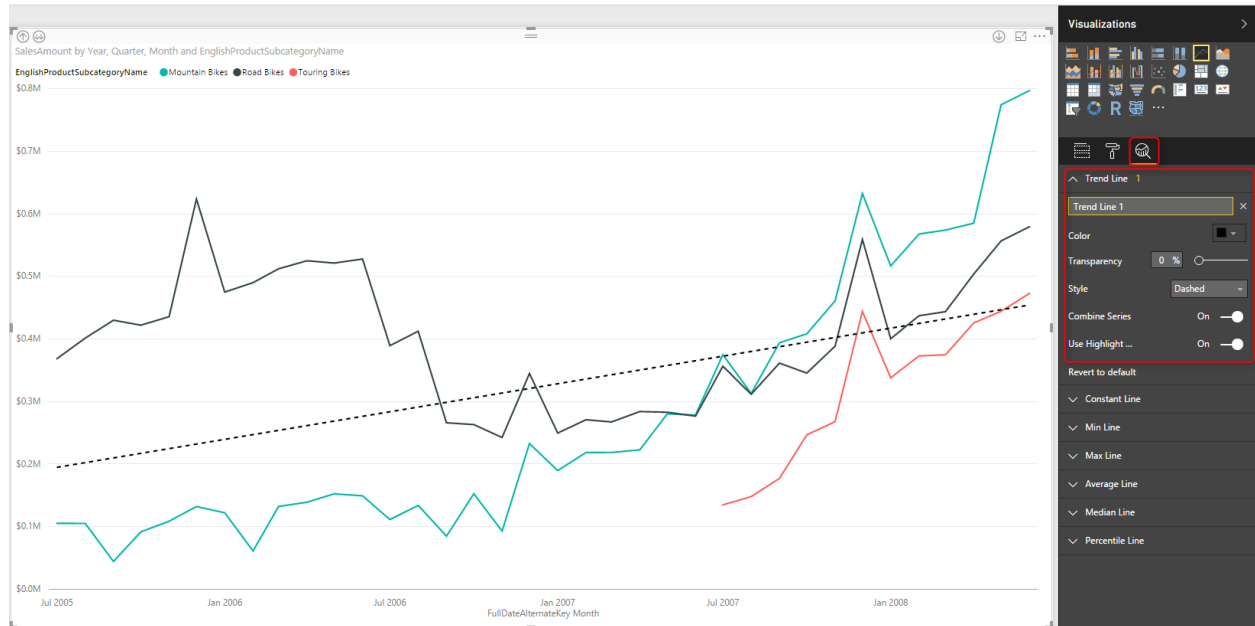
Drill down into Month Level afterwards.



Month level view shows better visualization as a detailed view, and not as messy as Day view (you can drill down to day to see what I mean by messy).

Trend Line

As I explained earlier, Line chart can simply show a trend line to investigate overall behavior of the value over the period of time. For adding this trend line, simply go to Analytics Tab (when you have line chart selected), and under Trend Lines click on Add.



As you see in the screenshot above, this trend line simply show the trend of Total Sales Amount over the period, which is increasing. You can have this trend as total as is, or have a separate trend line per each product subcategory.

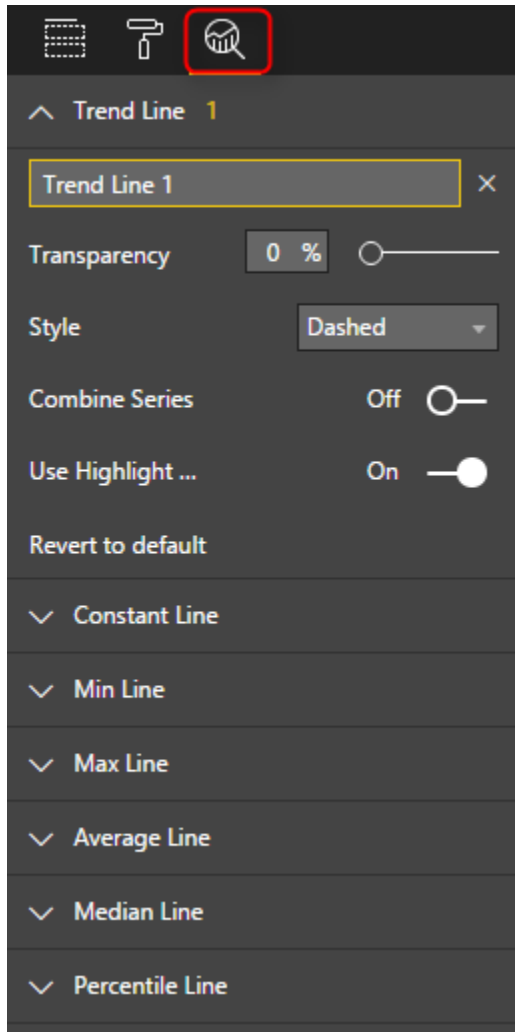
Trend Line by Category

Change the Combine Series option in the Trend Line setting to Off. You will see now a trend line per each subcategory. This is very useful visualization to see how each subcategory works. For example you can now simply see the sales amount for black solid line (road bikes) increased in last few months of the data, however it is still trending down (black dashed line). and the trend for other two subcategories is going upwards.



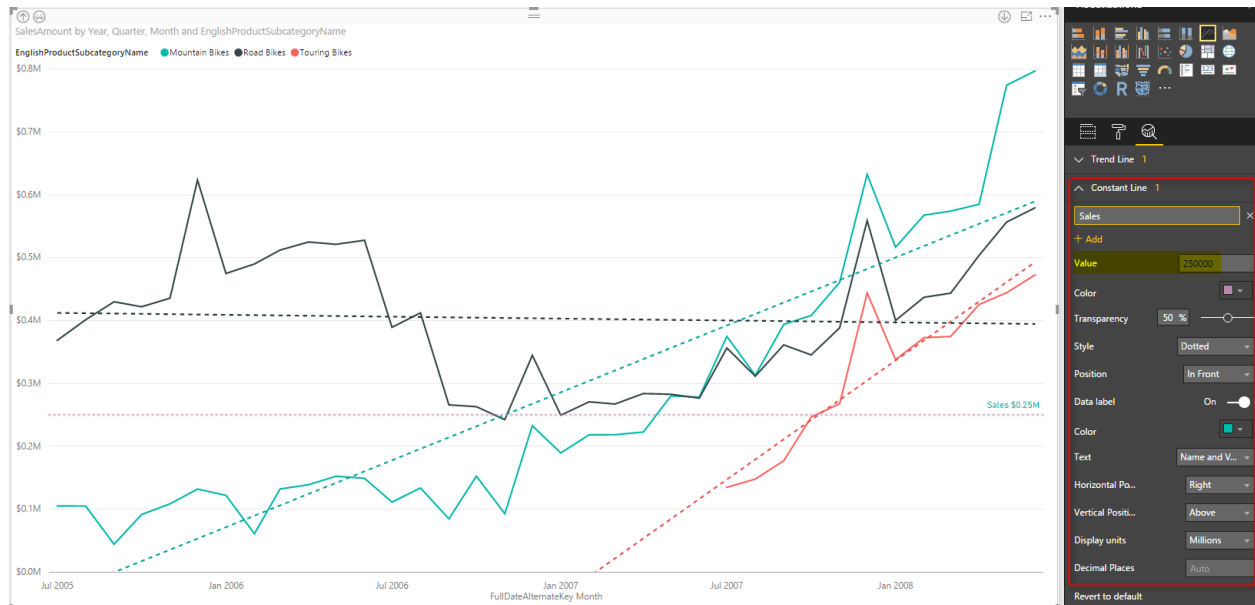
Analytics Tab

In Line chart (and few other charts) there is an Analytics tab that brings the power of adding extra insight, named Analytics Tab. In line chart we can have the extra insight as Trend Line, Constant Line(s), and Dynamic Reference Line(s). You have seen already how helpful a trend line is. next other lines are reference lines which can be useful to measure values in line chart and compare them to a reference value. Let's dig more into it now.



Constant Line

A constant line is a constant value showed as a line. It is good for comparing the value line to a constant line. Here I have created a constant line for value \$250000. and did some formatting for that and made it dotted.



This is good to compare values with especially when we DO NOT want to drill down or up. The reason for that is that if I drill up one level, then because I am in quarter level most of values goes above this constant value. So normally constant line used when you have the main value as the Average, rather than sum or count.

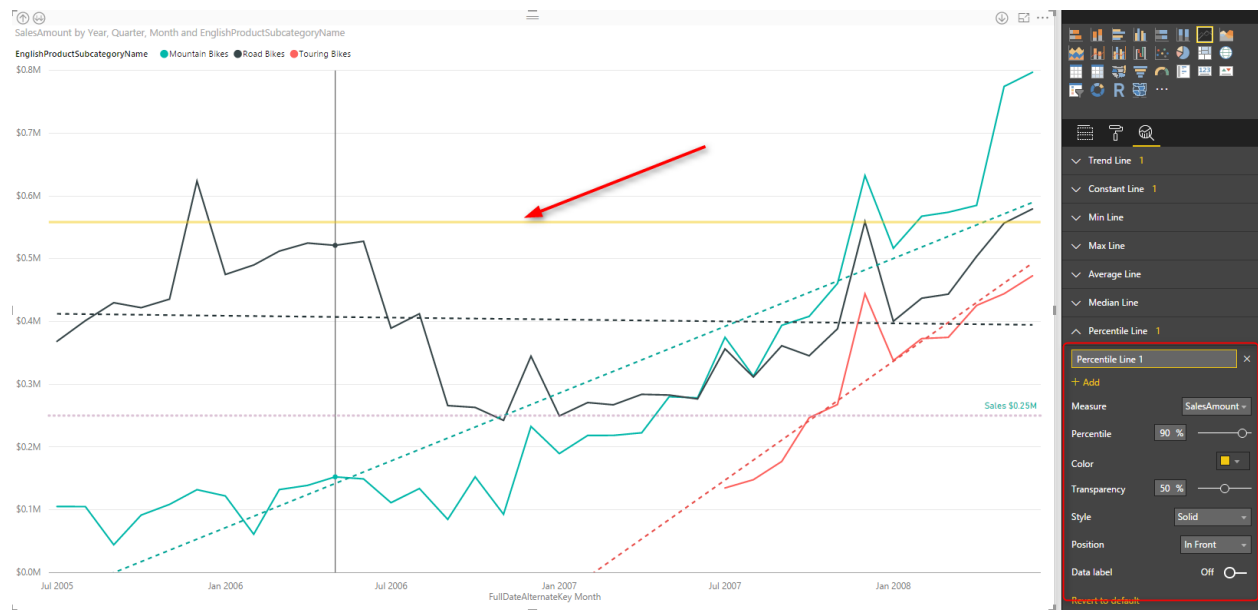


Dynamic Reference Lines

Constant line as you see above are not good when we drill down or up, because their value might not be close to compare with the value lines in that level. So dynamic reference lines are the substitute for such case. For example you can create an Average reference line, minimum, or maximum which works based on the level you are in, or filter applied, or what other slicers selected. This would be based on what data is showed in the value line. There are number of dynamic reference lines that you can create; Min, Max, Average, Median, and Percentile.

Percentile Line

You can simply add a percentage line, based on the percentage number you want. This can be percentage of one of measures viewed in the line chart.



Min, Max, Average, Median Lines

For adding any of these lines, just click add, set the formatting, name the line as something and then choose to show data labels with name and values or without it.



Do Not Overuse

Line Chart is a simple chart and there is a reason for this simplicity, to figure out values, trends, and analyze it with a quick glance. Adding more lines to it will add more insight obviously, but overusing it with adding all possible reference lines, trend lines, and multiple constant lines will make it a very busy line chart which then doesn't convey the message as it should. Consider using this visualization in the best way, get benefit of reference and trend lines, but only when it adds value.

Stacked Chart or Clustered? Which One is the Best?

Published Date: January 11, 2017



Column chart and Bar chart are two of the most basic charts used in every report and dashboard. There are normally two types of these charts: Stacked, and Clustered. At the first glance they seem to do the same action; showing values by categories and sub-categories. However, they are different. Your storytelling of data would be totally different when you use one of these charts. The difference is simple, but knowing it and considering it in your visualization will take your visualization simply one level up. In this post I'll show you the difference and when to use which. I will use Power BI for this example, but this is a conceptual post and works with any other BI tool. If you like to learn more about Power BI; read [Power BI online book, from Rookie to Rock Star](#).

Prerequisite

For running example of this post you will need AdventureWorksDW sample database, or you can download **Excel version** of it from here:

Enter Your Email to download the file (required)

[Download]

Simple Column Chart

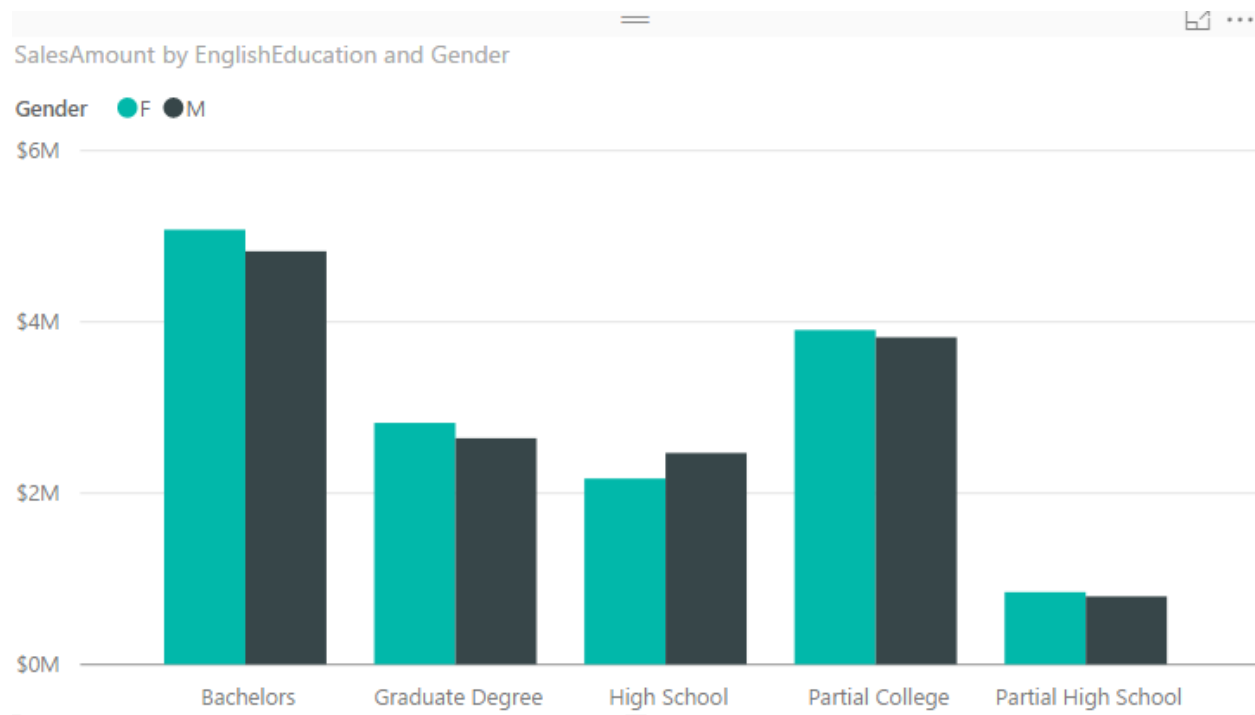
One of the best options to visualize difference between quantitative values across multiple items is column chart or bar chart. I have created a Power BI Desktop file connected to AdventureWorksDW, and loaded DimCustomer and FactInternetSales into the model. The chart below is a Column chart with EnglishEducation from DimCustomer as Axis, and SalesAmount from FactInternetSales as Value;



You can see with a single glance that Bachelors are producing the most revenue in this data set, and then Partial College after that. You can also simply understand Partial High School type of customers are lowest revenue generating customers. Column Chart and Bar chart are best for comparing a quantitative value (SalesAmount) based on categories/items (EnglishEducation). Now let's bring another dimension here.

Clustered Chart

Clustered charts are best for comparing all categories and their sub categories as part of a whole. If you add Gender from DimCustomer as Legend, and choose the Clustered Column Chart type for your visual, this is what you will see;



Similar to normal column chart you can easily figure out which sub category producing the most revenue: Bachelors Female. and Partial High School Male is the least revenue generation sub category. You can also very easily realize that in each category which sub category has higher value than the other one;



You can simply say that Males are producing less revenue than females in the Partial College category. Even though in Partial College values of Male and Female sub categories are very close to each other. These are questions that you can answer with this chart easily:

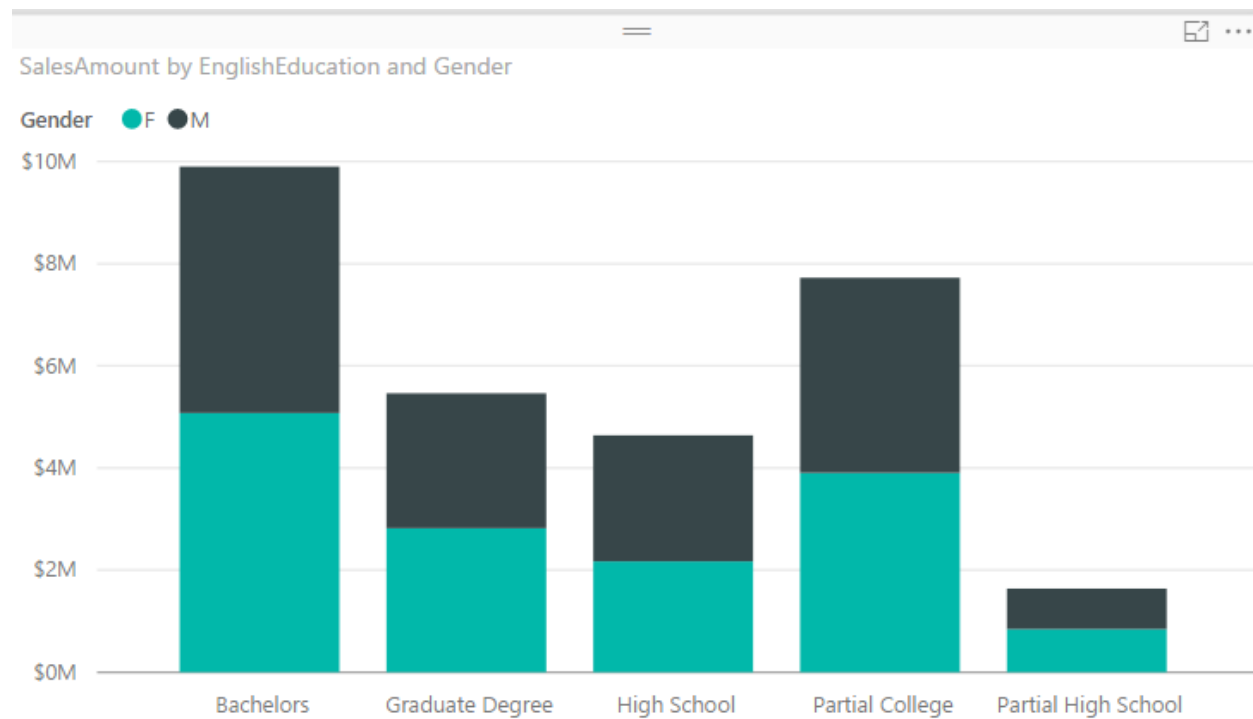
- Q: Which sub category generating most revenue?
- A: Bachelors Female.
- Q: Which sub category in Partial College generating least revenue?
- A: Male.

How about this question: Which category produces most total revenue? You would say Bachelors probably fairly quickly because both Female and Males has highest values of females and males across all categories. However if I ask you between High School and Graduate Degree, which one has the most total revenue based on only Clustered Chart above, that might be challenging to answer. The point is:

Clustered Chart are not good for comparing totals of categories

Stacked Chart

Point above leads us to Stacked Chart which is good for comparing totals and also values of sub categories. Change type of visual to Stacked Column Chart;

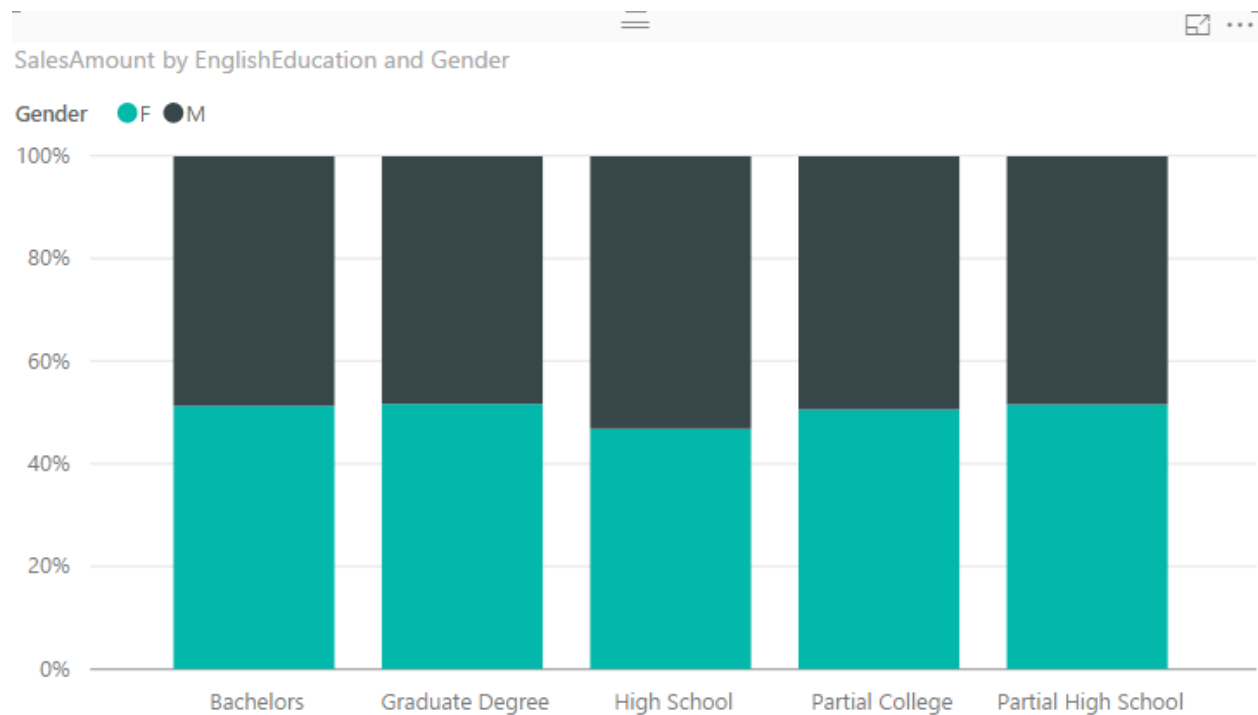


With this chart you can simply say which Education category provide most revenue, because the total is showed. between Graduate Degree and High School, it is obviously Graduate Degree. in Stacked chart however isn't easy to compare values of a category. For example finding out is Male or Female producing more revenue in High School category is almost impossible through this. Stacked Chart is good when value difference is high between sub categories in each category.

Stacked Chart are not good for comparing values in each category

100% Stacked Chart

There is another type of stacked chart, named as 100% Stacked Chart. This chart is good for comparing percentages normally. To find out how sub categories of each category are doing compared with other sub categories. If you change the type of chart to 100% Stacked Column Chart, here is what you will see:

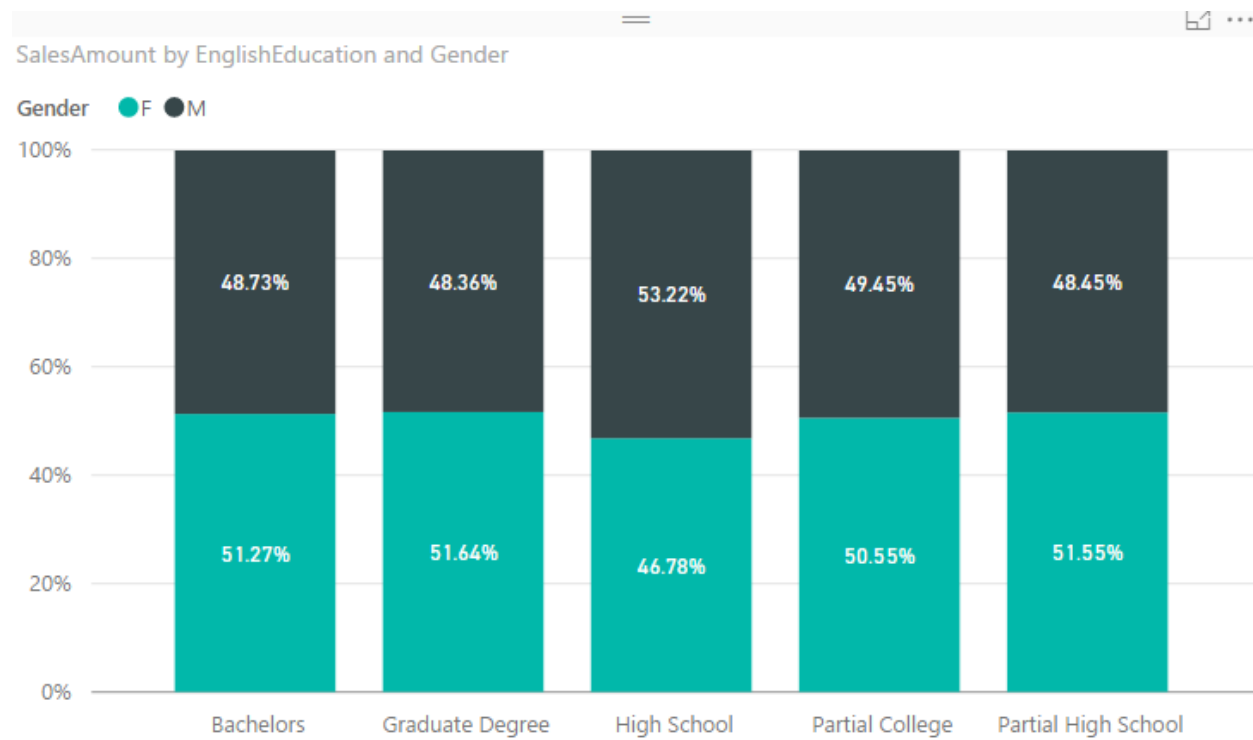


You can realize very quickly that portion of Males in all Education categories are less than High School's Males. However You can never use 100% stacked chart for comparing actual values. This chart is only good for percentages. When you use this chart in Power BI, it will automatically uses percentages calculation for it.

100% Stacked Charts are not good for comparing actual values.

Data Label is Your Asset

Regardless of what type of chart (Stacked, 100% Stacked, or Clustered) you are using, Data Labels are always helpful. Here is how charts can be powered with labels;



Your Story of Data

Deciding which chart is the best is all depends on what you want to tell as the story of the data;

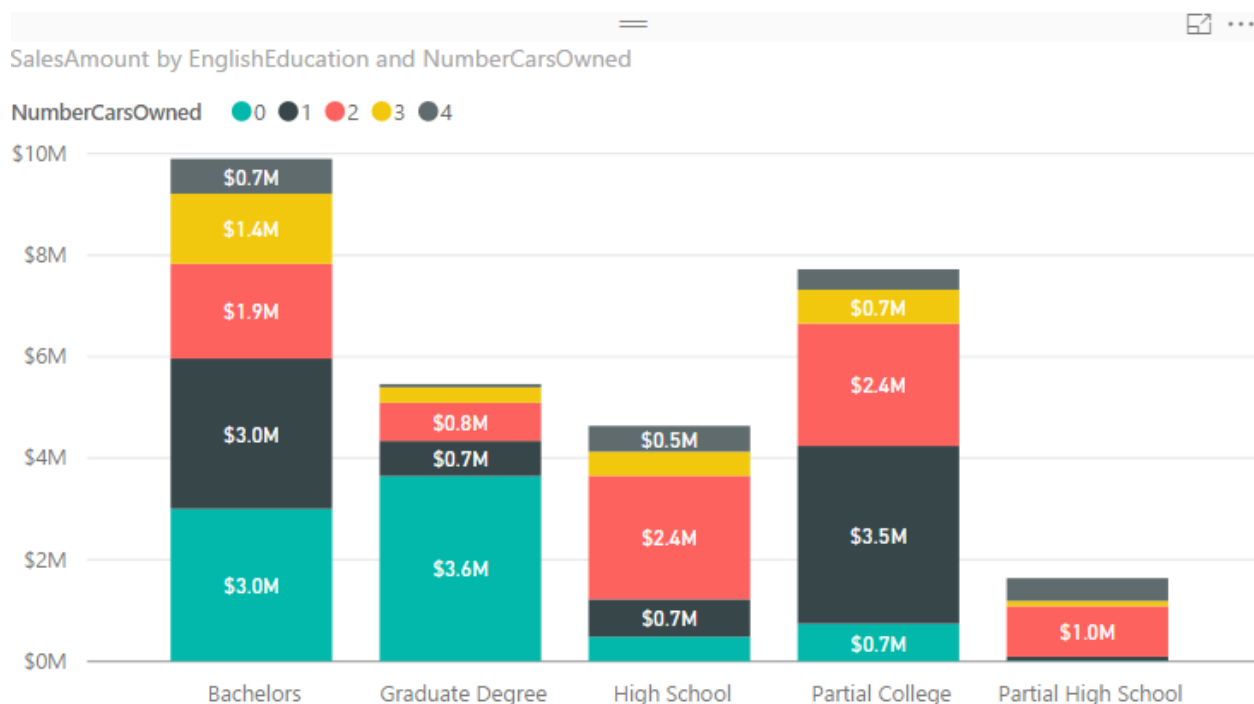
13. If you want to visualize totals of each category with separation of sub categories; Stacked Chart is your option.
14. If you want to visualize differences of percentage of sub categories value with other sub categories in other categories; 100% Stacked is the best.
15. If you want to compare values for all sub categories, and understand which one is the best or worst, inside a category or overall; Clustered Chart is your asset.

There is no one single chart telling the whole story. You need to use combination of these, but use them wisely. For example don't use Stacked Chart if you want to understand which sub category of a single category performs better or worst, or don't use Clustered chart if you want to compare totals of categories together.



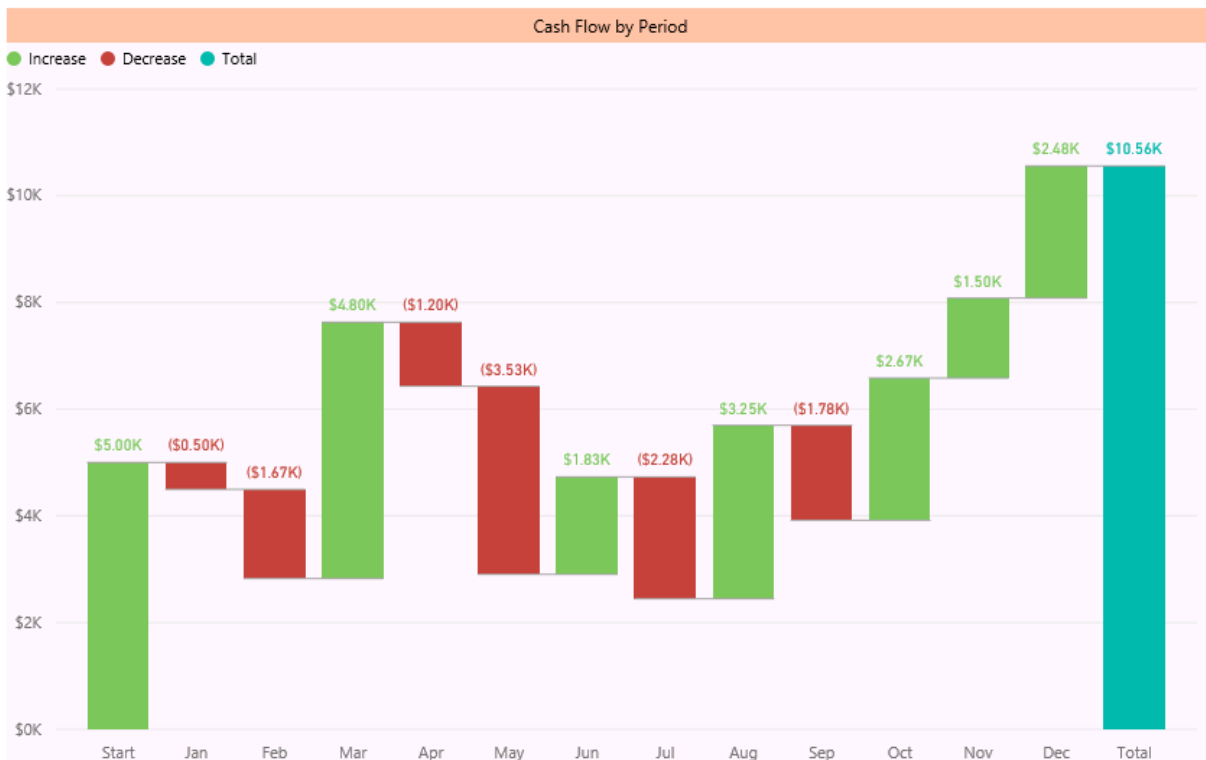
Be Careful of Too Many Items

All of these charts works perfectly when you have few number of items. if you want to spread these across many sub category these charts would be hard to understand. Best would be splitting that variety in multiple charts.



Power BI Waterfall Chart: What's That All About?

Published Date: July 29, 2015



You've heard the news about Power BI Desktop release with bunch of new features. If you don't, read [blog post here to understand new features of Power BI](#). Waterfall chart is one of the new visualization elements in this product. For finance people this chart makes sense, but not everyone knows the usage of that. In this post I want to explain what is Waterfall chart, when to use it? and how to use it? So you would learn what's Waterfall chart all about?

There are many charts that are self-explanatory, such as Bar chart, line chart.... Some charts might need a little bit working with to understand what is the best situation to use them, such as scatter chart (especially with play dimension). Some charts looks easy but you might not get their reason to exists, and the type of problem that they solve. From feedback that I've heard I think waterfall chart is in the latest category. But don't be afraid, it is an easy chart. you only need to know when and how to use it.

What is Waterfall Chart?

Waterfall Chart is a type of chart that usually used for

“understanding the cumulative effect of sequentially introduced positive or negative values”.

Don't blame me! I've get that definition from [Wikipedia](#) 😊 My simple definition however is: Waterfall chart is for analysis of up and downs for an additive measure. as an example you can analysis cashflow based on up and downs of it through all months of a financial year.

When to use Waterfall Chart?

You can also use waterfall chart for analysis of inventory, credit, performance, and everything else that falling or raising of its value matters. sequential of attribute that describe the value is also another helpful dimension of the waterfall chart. For example you want to know up and downs of performance through the last year with the sequence of quarters that you've done performance reviews.

So Don't limit yourself, this chart visualize and tell many stories, just think big and find situations that you can use this chart to respond to the existing requirement.

How to use this Chart?

Easy! When you know the usage of chart, and you have the data set to work with, then just use Waterfall chart in Power BI Desktop. I'll show it through an example to you.

Sample Data

I used the data set provided by [Debra Dalglish](#) in [this post](#). The data set is a cashflow data set as below:

	A	B	C	D	E
1	Period ▾	Base ▾	Down ▾	Up ▾	Cash Flow ▾
2	Start		\$0	\$5,000	\$5,000
3	Jan	\$5,000	\$503	\$0	-\$503
4	Feb	\$4,497	\$1,670	\$0	-\$1,670
5	Mar	\$2,827	\$0	\$4,802	\$4,802
6	Apr	\$7,629	\$1,198	\$0	-\$1,198
7	May	\$6,431	\$3,526	\$0	-\$3,526
8	Jun	\$2,905	\$0	\$1,826	\$1,826
9	Jul	\$4,731	\$2,284	\$0	-\$2,284
10	Aug	\$2,447	\$0	\$3,250	\$3,250
11	Sep	\$5,697	\$1,780	\$0	-\$1,780
12	Oct	\$3,917	\$0	\$2,667	\$2,667
13	Nov	\$6,584	\$0	\$1,500	\$1,500
14	Dec	\$8,084	\$0	\$2,475	\$2,475
15	End	\$10,559			

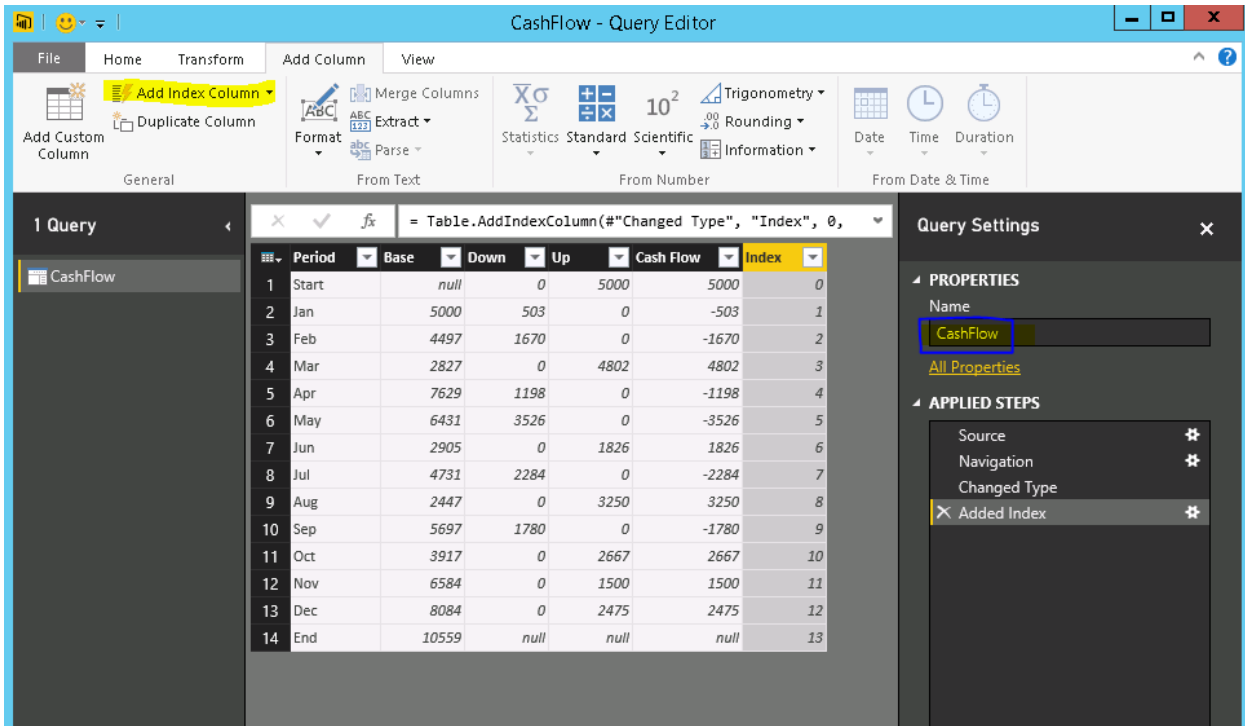
As you see in above data set we have periods (months), base, up, down, and the cash flow. If you want to understand how the calculation of cashflow works. it is as below:

- Base of the start period is coming from previous period (year) balance.
- Base of each period is the Base from previous period + Up of previous period – Down of the previous period
- CashFlow is equal to Up minus Down
- value of End of Period calculated in the same way of base for each period
- As an example $B3 = B2 + D2 - C2$ (this formula continues to down with numbers changed based on rows)

So Let's use this data set in the Power BI.

Extract the Data

I just fetch the data from the excel file as is. with no changes. the only change I make is to add an Index Column to it (this index column will be used for sorting later on);



CashFlow - Query Editor

File Home Transform Add Column View

General From Text From Number From Date & Time

1 Query

CashFlow

Formula Bar: `= Table.AddIndexColumn(#"Changed Type", "Index", 0,`

	Period	Base	Down	Up	Cash Flow	Index
1	Start	null	0	5000	5000	0
2	Jan	5000	503	0	-503	1
3	Feb	4497	1670	0	-1670	2
4	Mar	2827	0	4802	4802	3
5	Apr	7629	1198	0	-1198	4
6	May	6431	3526	0	-3526	5
7	Jun	2905	0	1826	1826	6
8	Jul	4731	2284	0	-2284	7
9	Aug	2447	0	3250	3250	8
10	Sep	5697	1780	0	-1780	9
11	Oct	3917	0	2667	2667	10
12	Nov	6584	0	1500	1500	11
13	Dec	8084	0	2475	2475	12
14	End	10559	null	null	null	13

Query Settings

PROPERTIES

Name: **CashFlow**

APPLIED STEPS

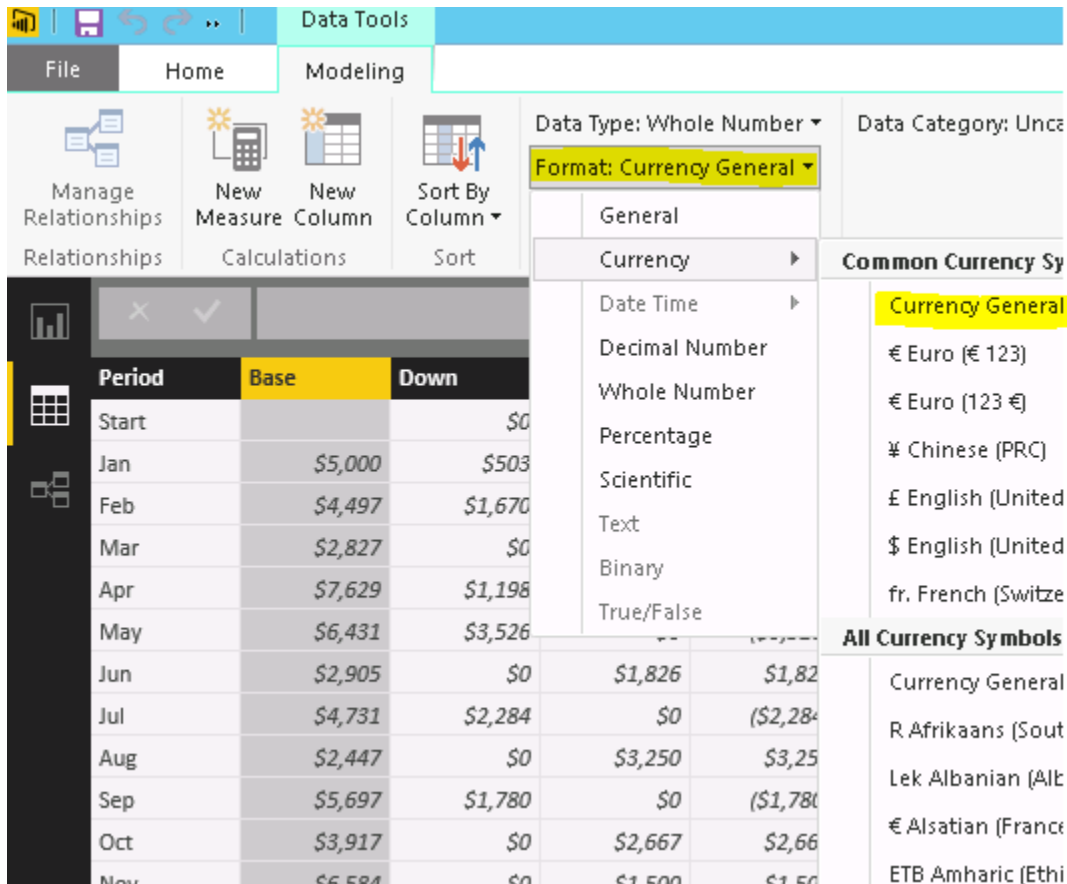
- Source
- Navigation
- Changed Type
- Added Index**

After adding the index column and changing name of the query to CashFlow. I'll start preparing the model.

Modelling

There is only one data table in our model, so modelling shouldn't be a problem. However there are some changes that I like to make as below:

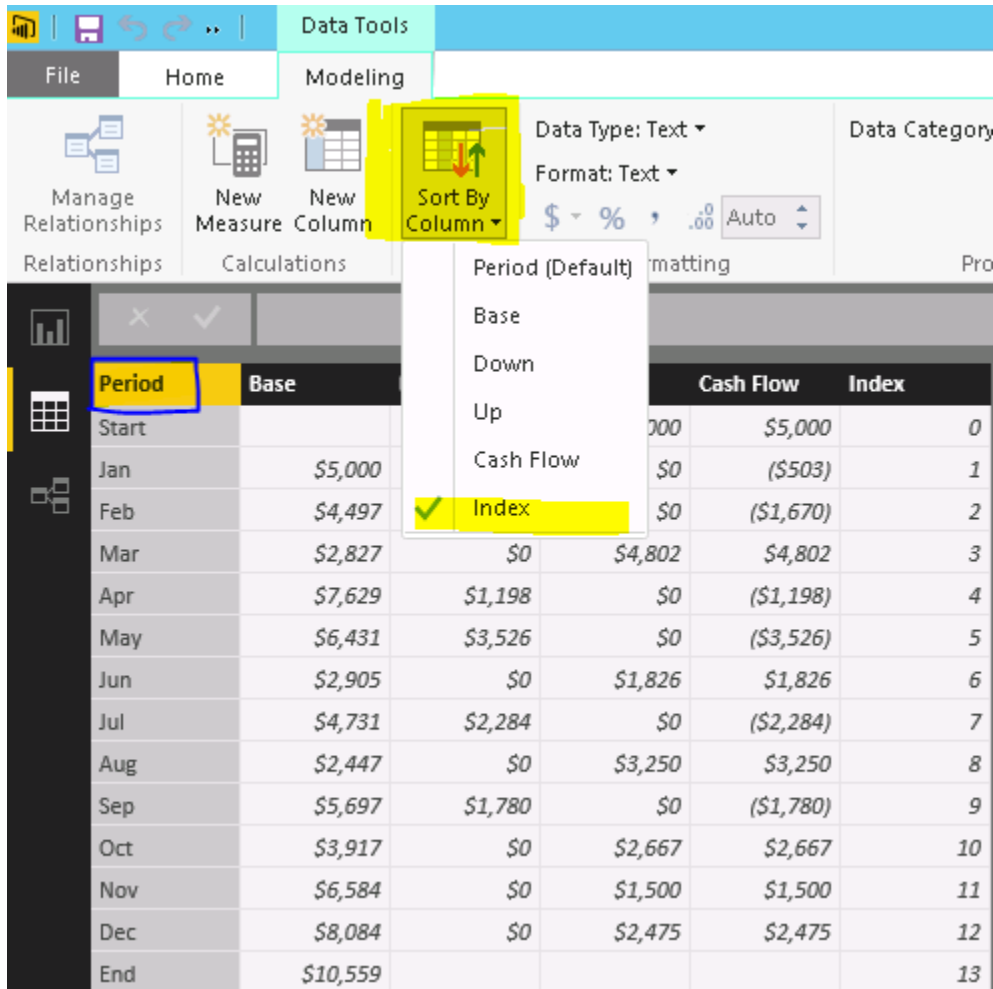
- Changing Format of columns (Base, Down, Up, and Cash Flow) as Currency



The screenshot shows the Power BI Desktop interface. The 'Data Tools' ribbon is active, and the 'Sort By Column' dropdown menu is open. The menu lists various data types, and 'Currency General' is selected. The background shows a table with columns 'Period', 'Base', and 'Down'.

Period	Base	Down
Start		\$0
Jan	\$5,000	\$503
Feb	\$4,497	\$1,670
Mar	\$2,827	\$0
Apr	\$7,629	\$1,198
May	\$6,431	\$3,526
Jun	\$2,905	\$0
Jul	\$4,731	\$2,284
Aug	\$2,447	\$0
Sep	\$5,697	\$1,780
Oct	\$3,917	\$0
Nov	\$6,584	\$0

- Change Sorting of Period Column to be based on Index column



Period	Base	Cash Flow	Index
Start			
Jan	\$5,000		
Feb	\$4,497		
Mar	\$2,827	\$0	\$4,802
Apr	\$7,629	\$1,198	\$0
May	\$6,431	\$3,526	\$0
Jun	\$2,905	\$0	\$1,826
Jul	\$4,731	\$2,284	\$0
Aug	\$2,447	\$0	\$3,250
Sep	\$5,697	\$1,780	\$0
Oct	\$3,917	\$0	\$2,667
Nov	\$6,584	\$0	\$1,500
Dec	\$8,084	\$0	\$2,475
End	\$10,559		

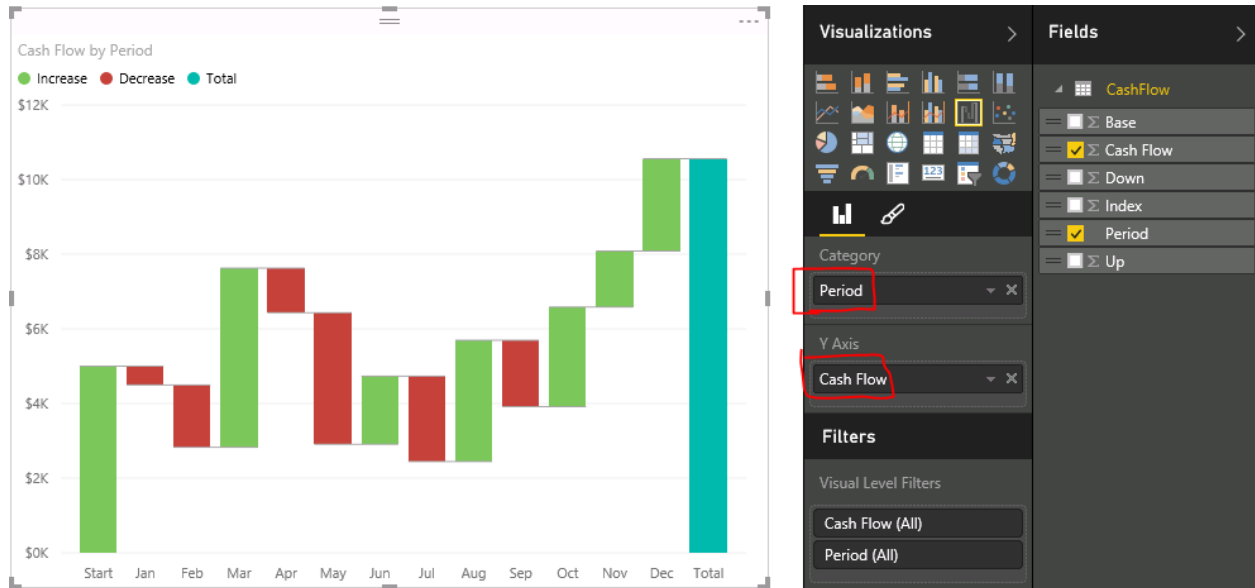
For doing above change; select Period column first, then from Modeling tab, click on Sort By Column, and then choose Index. If you don't apply this change waterfall chart will sort periods based on their alphabetical order.

Visualization

Go to Report tab. double click on Waterfall chart from Visualizations pane in the right hand side.

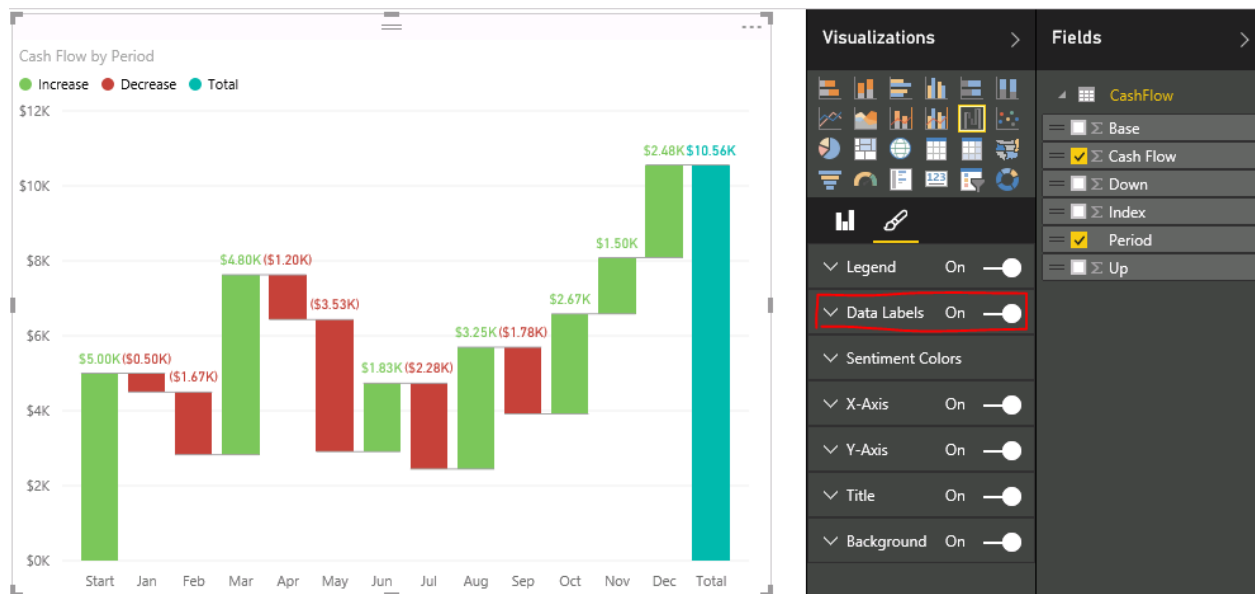
You have to set only two parameters:

- Category: Set it to Period
- Y Axis: Set it to Cash Flow



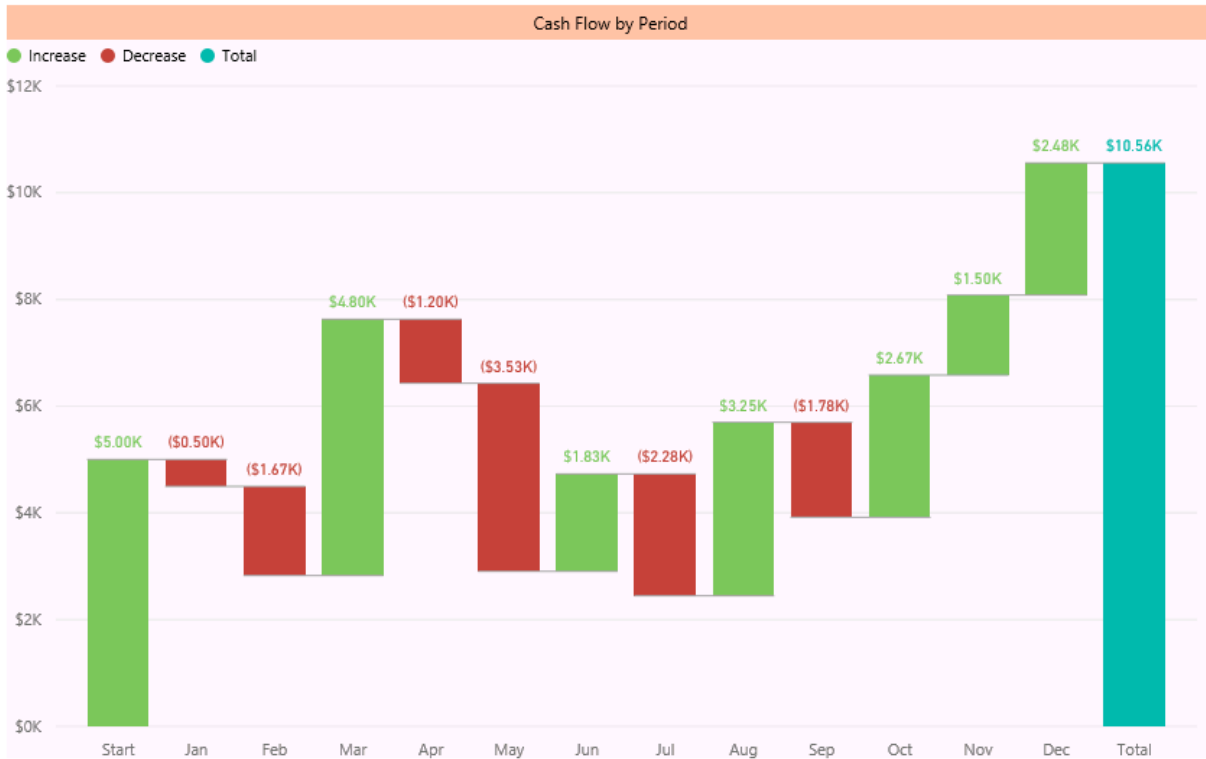
As you see in above chart, the cashflow and its up and down showed beautifully in the chart. Order of period category is also right, and that is because of the Sort By Column setting we've done in previous section.

You can apply formatting as you want to this chart, like changing colors, backgrounds, title and so on. I just like to set data labels for now.



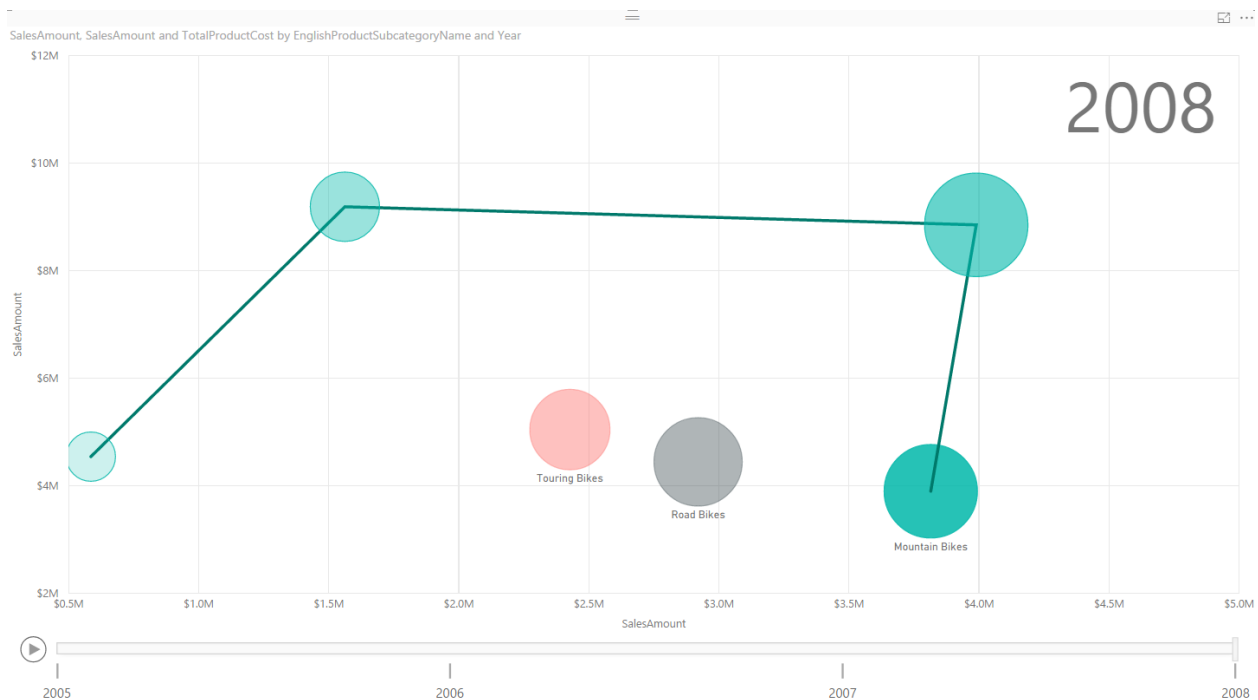
Result

And this is the final chart (with few formatting):



Storytelling with Power BI Scatter Chart

Published Date: July 30, 2016



Column or Bar chart can be easily used for showing a single measure's insight across a category. Mixed charts such as Line and Column chart can be used for showing two measure and comparing their values across a set of categories. However there are some charts that can be used to show values of three measures, such as Scatter Chart. Scatter chart not only shows values of three measure across different categories, it also has a special Play axis that helps you to tell the story behind the data. In this post you'll learn how easy is to visualize something with Scatter chart and tell a story with that. If you like to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

Prerequisite

For this example like many other examples in this [Power BI book](#), I'll be using AdventureWorksDW SQL Server database. If you don't have this installed, you can easily download it from [here](#) and install it.

Scatter Chart

Scatter chart is a built-in chart in Power BI that you can show up to three measure with a categorization in it. Three measures can be visualized in position of X axis, Y axis, and size of bubbles for scatter chart. You can also set up a date field in play axis, and then scatter chart will animate how measure values are compared to each other in each point of a time. Let's start building something simple with this chart and see how it is working in action. At the end of example you will see a summary chart as below;

Visualization Item	Scatter Chart	Comments
Measure 1	X Axis	
Measure 2	Y Axis	
Measure 3	Size of bubbles	
Category	Yes	As Legend
Detailed Category	Yes	As Details
Play Axis	Yes	
Color Formatting	Needs Improvement	Data Colors can't be set
Text Formatting	Needs Improvement	Title for X/Y axis can't be set
General Formatting	Needs Improvement	No Formatting option for Play Axis
Custom/Builtin	Builtin Chart	

Building Sample Report

Open Power BI Desktop, and start by Get Data from SQL Server AdventureWorksDW database. For this example you need to import data for 6 tables; FactInternetSales, FactResellerSales, DimDate, DimProduct, DimProductCategory, and DimProductSubCategory.

Navigator

Display Options ▾

- ☐ DimCustomer
- ☒ DimDate
- ☐ DimDepartmentGroup
- ☐ DimEmployee
- ☐ DimGeography
- ☐ DimOrganization
- ☒ DimProduct
- ☒ DimProductCategory
- ☒ DimProductSubcategory
- ☐ DimPromotion
- ☐ DimReseller
- ☐ DimSalesReason
- ☐ DimSalesTerritory
- ☐ DimScenario
- ☐ FactAdditionalInternationalP...
- ☐ FactCallCenter
- ☐ FactCurrencyRate
- ☐ FactFinance
- ☒ FactInternetSales
- ☐ FactInternetSalesReason
- ☐ FactProductInventory
- ☒ FactResellerSales
- ☐ FactSalesQuota
- ☐ FactSurveyResponse
- ☐ ProspectiveBikeBuyersProba...

DimDate

Preview downloaded on Wednesday, July

DateKey	FullDateAlternateKey
20050101	1/1/2005
20050102	1/2/2005
20050103	1/3/2005
20050104	1/4/2005
20050105	1/5/2005
20050106	1/6/2005
20050107	1/7/2005
20050108	1/8/2005
20050109	1/9/2005
20050110	1/10/2005
20050111	1/11/2005
20050112	1/12/2005
20050113	1/13/2005
20050114	1/14/2005
20050115	1/15/2005
20050116	1/16/2005

The data in the preview has been tr

Just Load the data, don't edit. Then go to Relationship tab to verify relationship. remove inactive relationships between DimDate and FactInternetSales/FactResellerSales. Only keep one relationship active based on ORDER DATE. to change a relationship to active

double click on the relationship line and change it to active. You can only have one active relationship between two tables.

Edit Relationship

Select tables and columns that relate to one another.

FactResellerSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	ResellerKey	EmployeeKey	PromotionKey	CustomerKey
317	20050801	20050813	20050808	403	282	1	
322	20050801	20050813	20050808	403	282	1	
316	20050801	20050813	20050808	403	282	1	

DimDate

DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek
20050701	Friday, July 1, 2005	6	Friday	Viernes	Vendredi
20050702	Saturday, July 2, 2005	7	Saturday	Sábado	Samedi
20050703	Sunday, July 3, 2005	1	Sunday	Domingo	Dimanche

Cardinality

Many to One (*:1)

Cross filter direction

Single

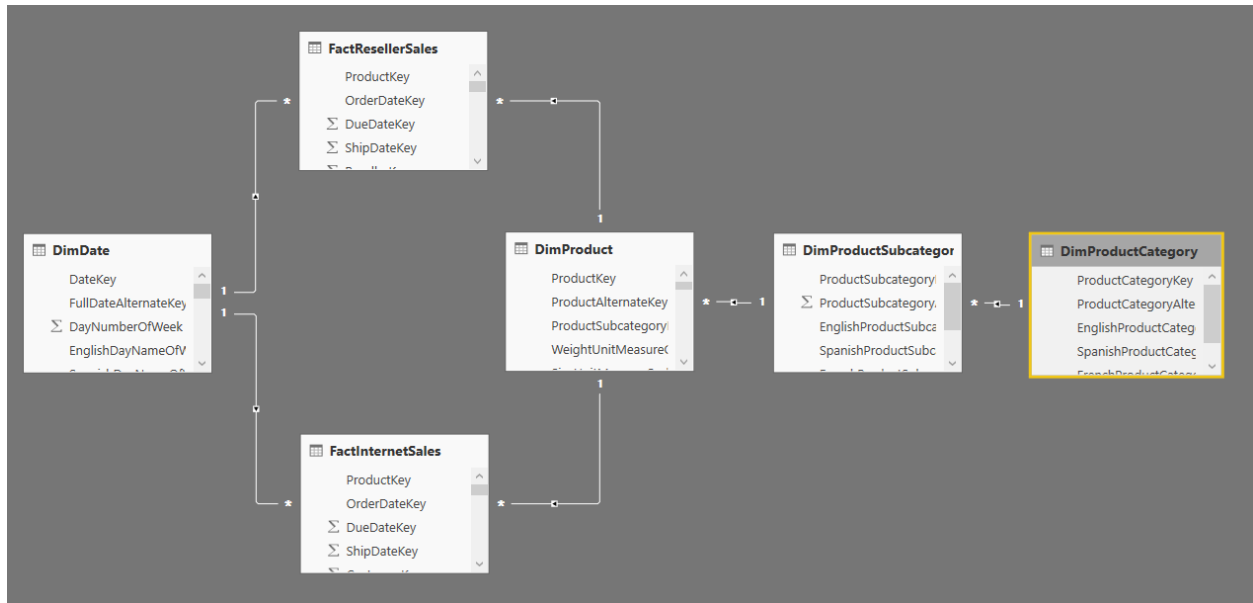
☒ Make this relationship active

☐ Assume Referential Integrity

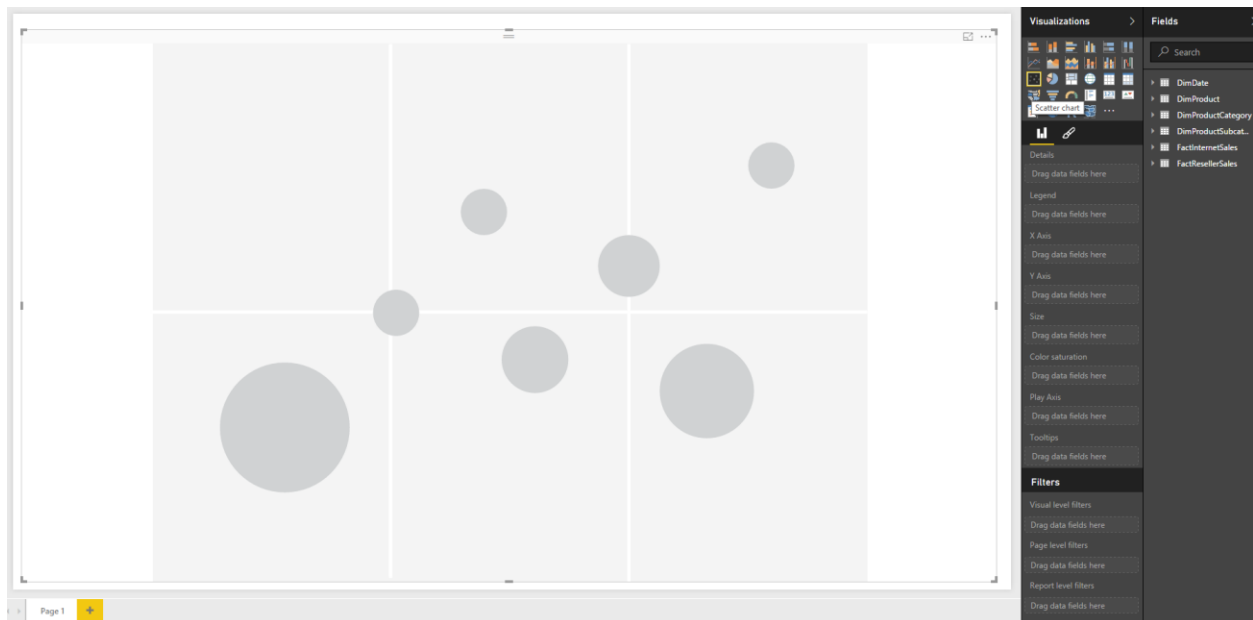
OK

Cancel

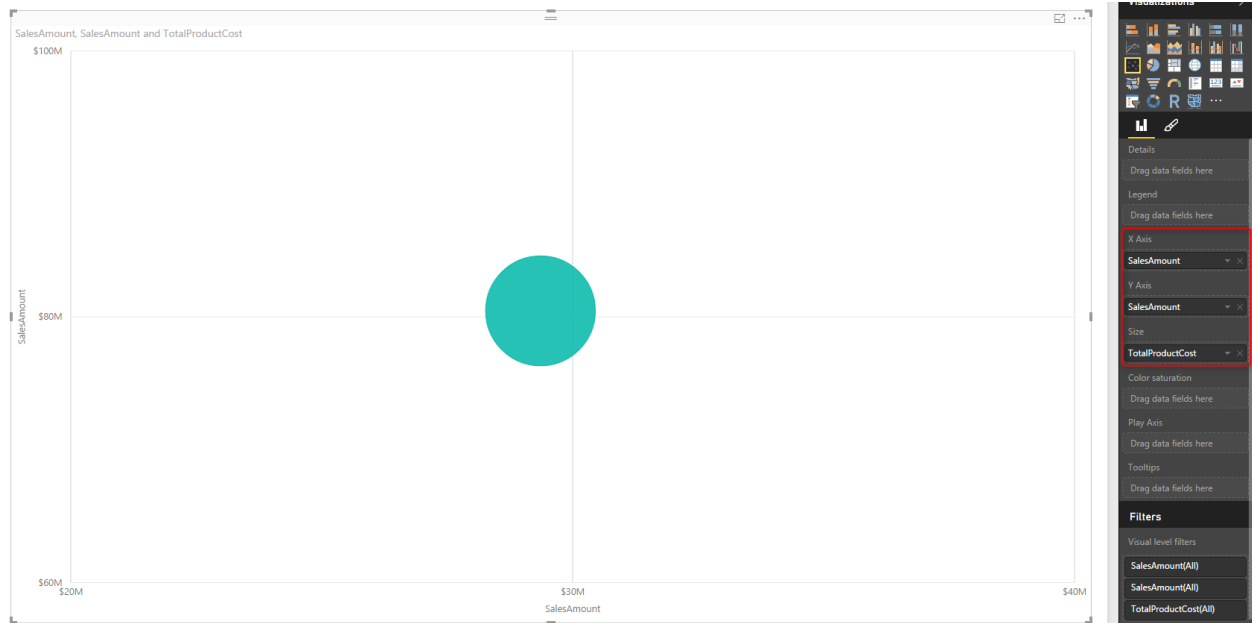
Your final relationship diagram should look like this;



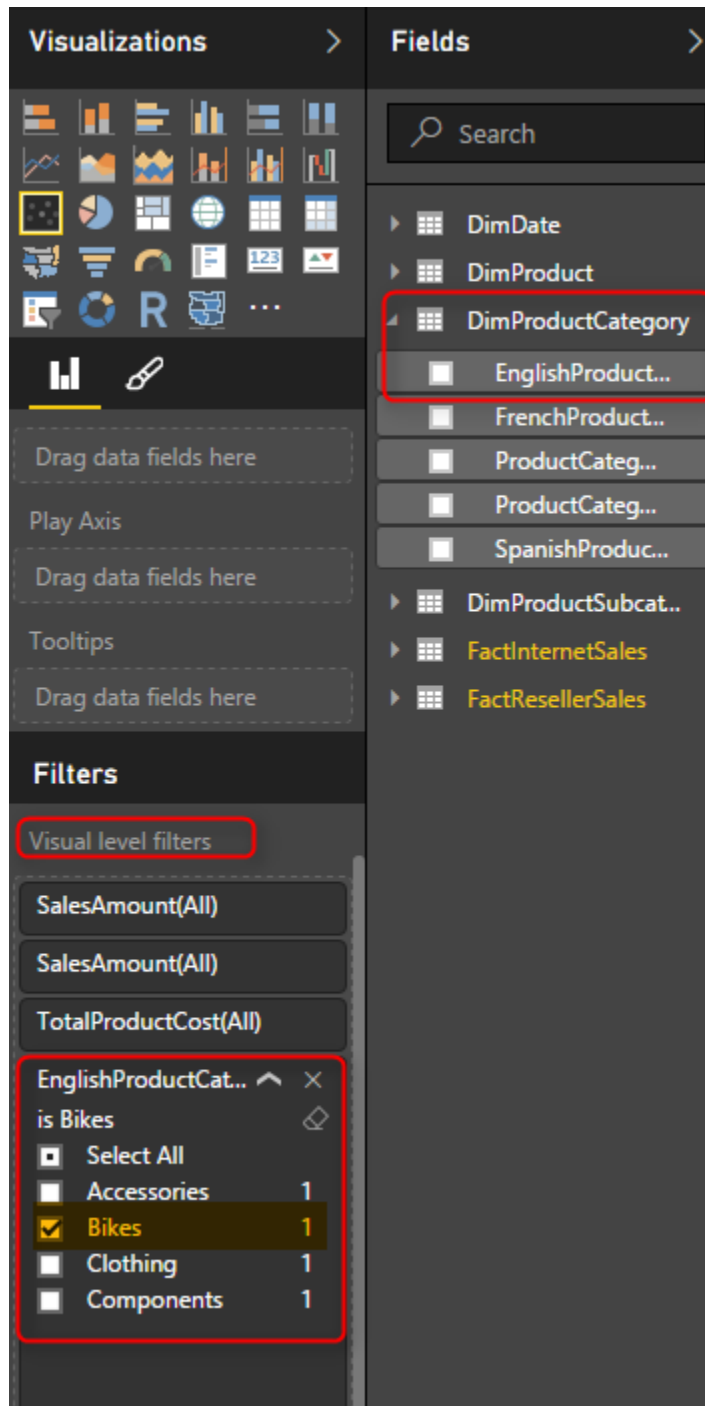
You might notice that this is not a best practice star schema model to work with, but let's make this example simple for now, and focus on the visualization side. Go to the Report, and create a Scatter Chart. expand it to cover the whole report page.



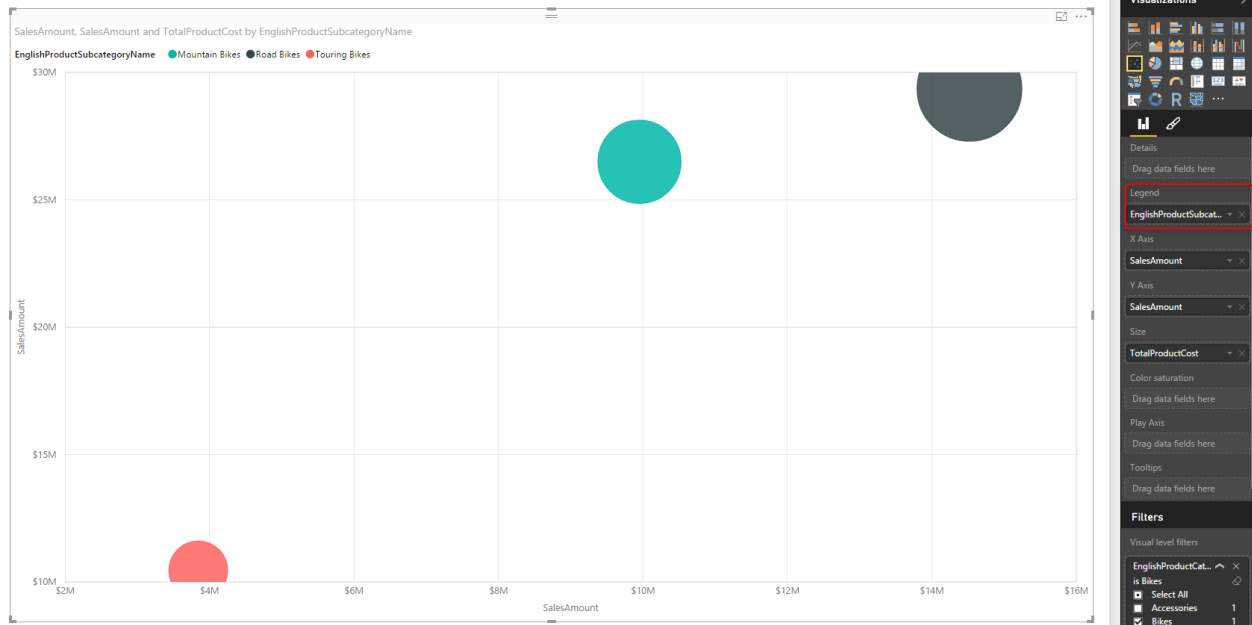
Add Sales Amount from FactInternetSales to X Axis, and then Sales Amount from FactResellerSales to Y Axis. Also bring TotalProductCost from FactInternetSales into Size. These are three measures that we want to visualize here;



Now Let's see how each product sub category of Bike is for these measure. For that add a Visual Level Filter for EnglishProductCategoryName from DimProductCategory, and set it to be Bike.



Now bring the EnglishProductSubCategoryName from DimProductSubCategory into the Legend.

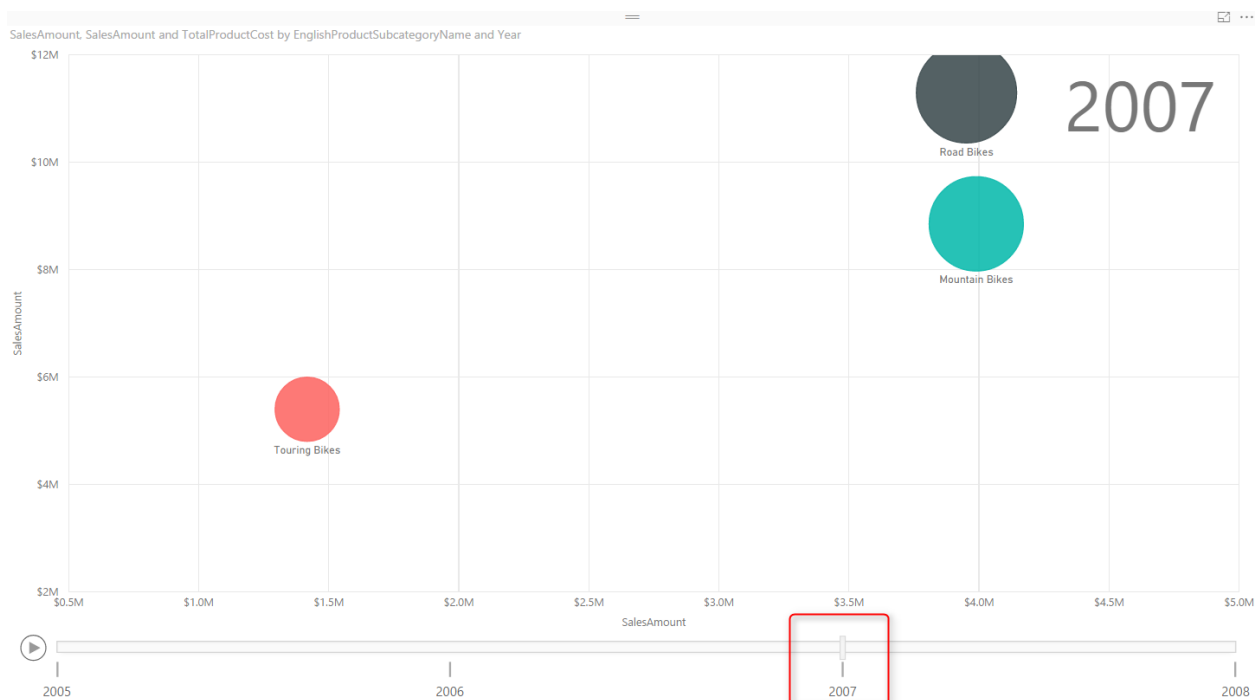


This simply shows the difference of categories for each measure. For example you can see that the total internet sales and reseller sales of road bikes (black) is greater than other two; Mountain bikes, and Touring bikes. Let's see was this the case all the time? or just the total is different? Fortunately with this chart we can do that with Play Axis.

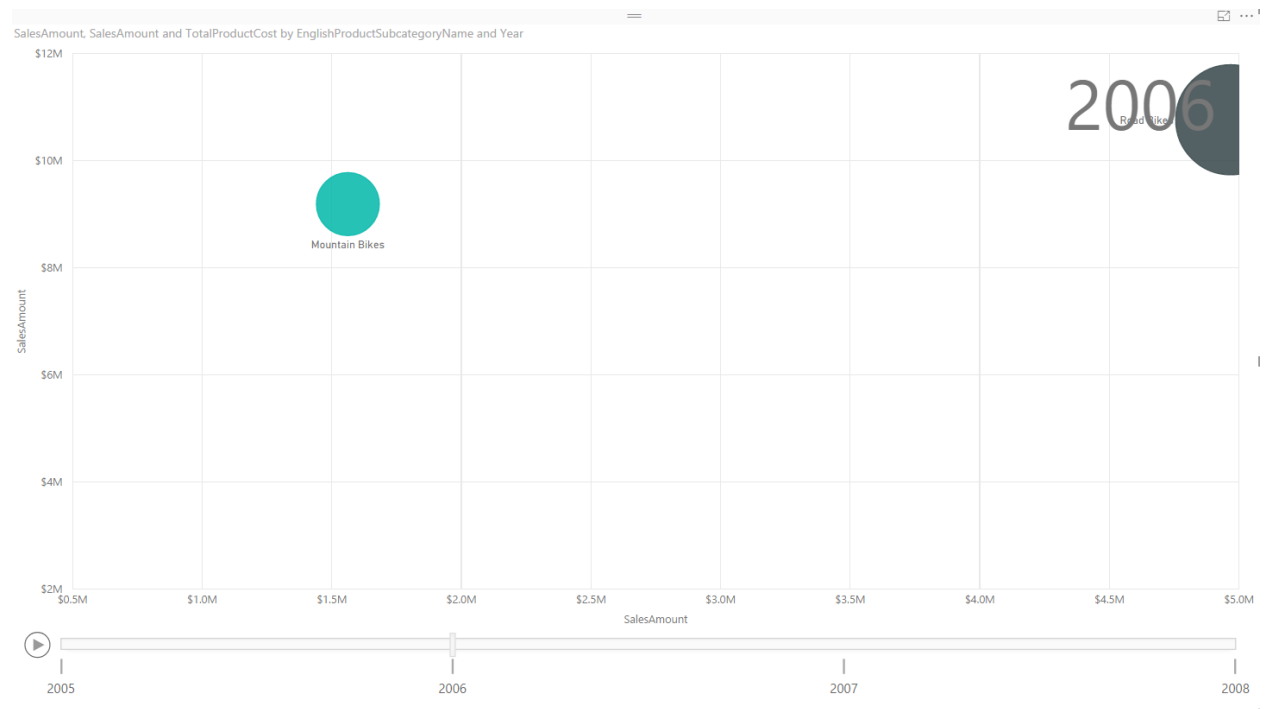
Drag and drop FullDateAlternateKey from DimDate to Play Axis. and you will see values in both axis change immediately. the reason is that now this chart will show value of one year at a time. and you can see the year title in the top right hand corner of scatter chart.



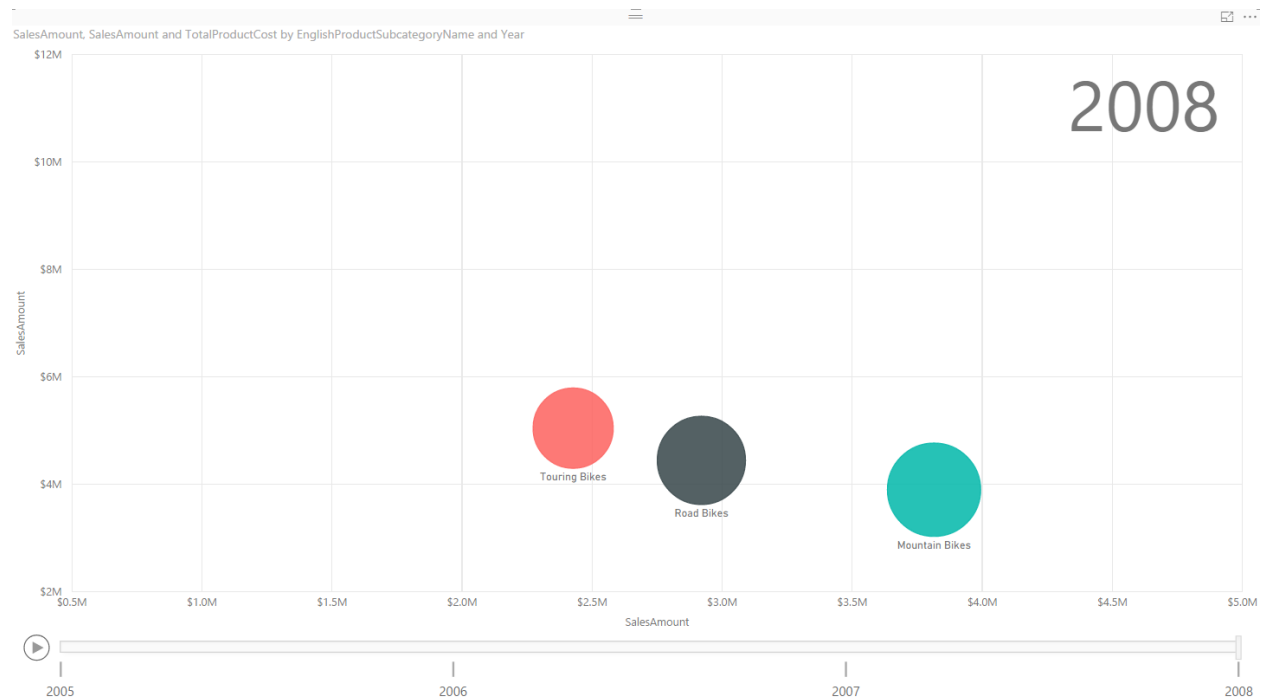
If you click on Play axis you can see that this the sub categories are moving around because their sales for each year is different. and easily you can see how their sales was during the period of these years. You can even move the player's bar navigator to a year like 2007 and see values at that point of the time. Screenshot below for year 2007 simply shows that mountain bikes and road bikes at that year had almost same internet sales amount, however the reseller sales of road bikes was more.



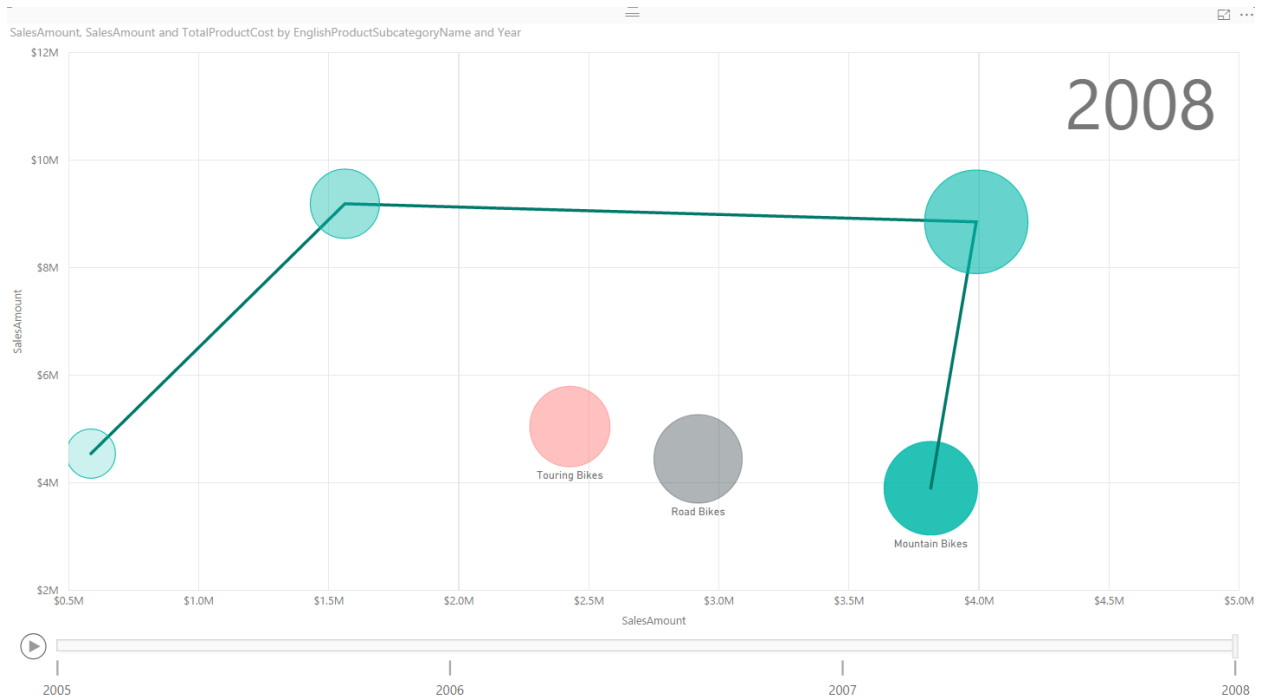
And if you go a year before you can see that there was no Touring Bikes even exists at that time, and sales of Road bikes was much more than mountain bikes in both internet sales and reseller sales.



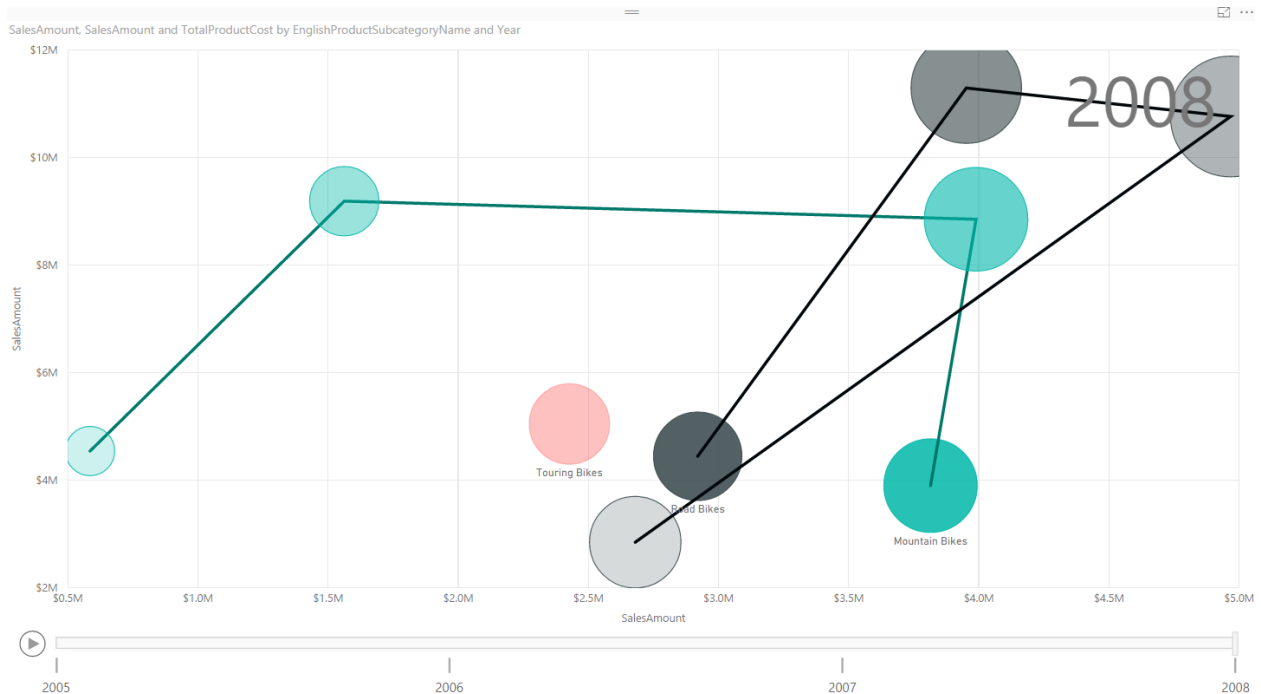
Also you will see that at the year 2008 the internet sales of road bikes is less that two other types of bike.



It is easy to compare year to year changes in this way. You can even click on each individual category and see how it worked through the time. Scatter chart shows you a line of all states for that category.



Even you can take one step further and select multiple items and simply compare them through the years. Click on Ctrl and choose Road and Mountain Bikes.



Now you can easily see that road bikes had better sales in years 2006 and 2007, but in 2008 it sold even less than mountain bikes for internet sales. The reseller sales of road bikes still is higher in year 2008. It is easy to compare all three measures through the time and based on different categories with Scatter chart. Scatter chart helps you to tell story behind the data easily.

You can also add more details to it. As an example let's add Class from DimProduct to Details of this chart. Now you will see more bubbles, for each sub category we have bubbles with same color, but different bubbles based on each product class.



Scatter chart also has a field for Color saturation and Tooltips which I haven't demonstrated here, feel free to play with it and enhance your visualization. Scatter chart is a powerful visualization that helps you to tell a story about a data of multiple measure and categories with their values through changes of time.

Formatting

Unfortunately will all power in this visualization, there isn't enough formatting options for it at this time. You can apply general formatting like a title for the whole chart, but you cannot change the title for X or Y Axis, which seems necessary. In this example the field name we've used was Sales Amount identical name but from two different tables. It is a basic need that I can change the title of axis, but I can't. The only work around for this is to change name of column in data tap of Power BI, or in Power Query.

Data colors can't be set as well. This option exists in many other charts such as column/bar chart and I believe it will be available soon here for this chart.

There is no control on the speed of Play axis or any formatting on this axis at this stage.

Summary

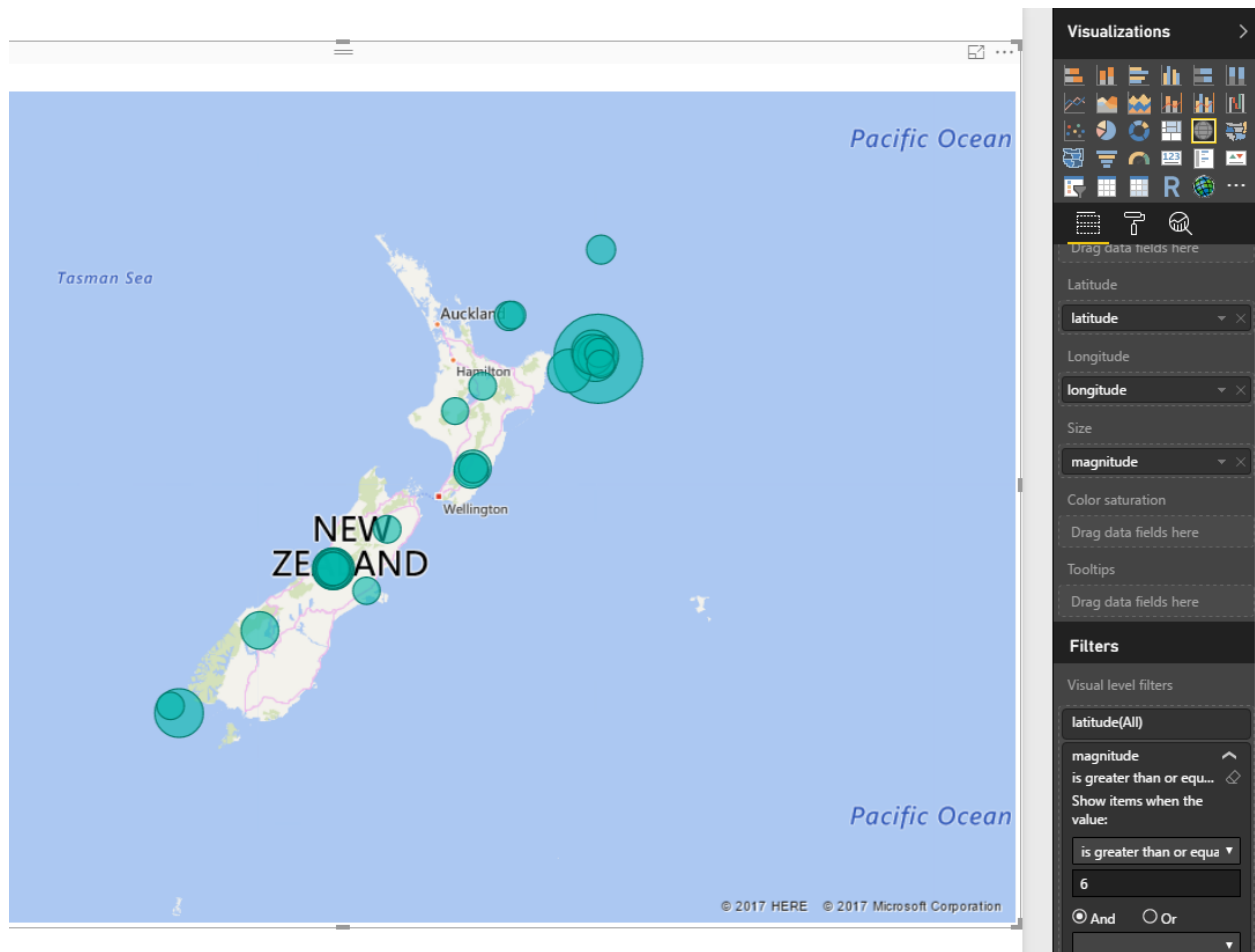
In Summary Scatter chart is one of the most useful charts in Power BI which helps for storytelling with multiple measures, categories, and changes of values through the time. However this visualization item, still needs some improvements for formatting, which I

believe with great efforts of Power BI team will be applied in the very near future. Here is a summary of features for this visualization item;

Visualization Item	Scatter Chart	Comments
Measure 1	X Axis	
Measure 2	Y Axis	
Measure 3	Size of bubbles	
Category	Yes	As Legend
Detailed Category	Yes	As Details
Play Axis	Yes	
Color Formatting	Needs Improvement Data Colors can't be set	
Text Formatting	Needs Improvement Title for X/Y axis can't be set	
General Formatting	Needs Improvement No Formatting option for Play Axis	
Custom/Builtin	Builtin Chart	

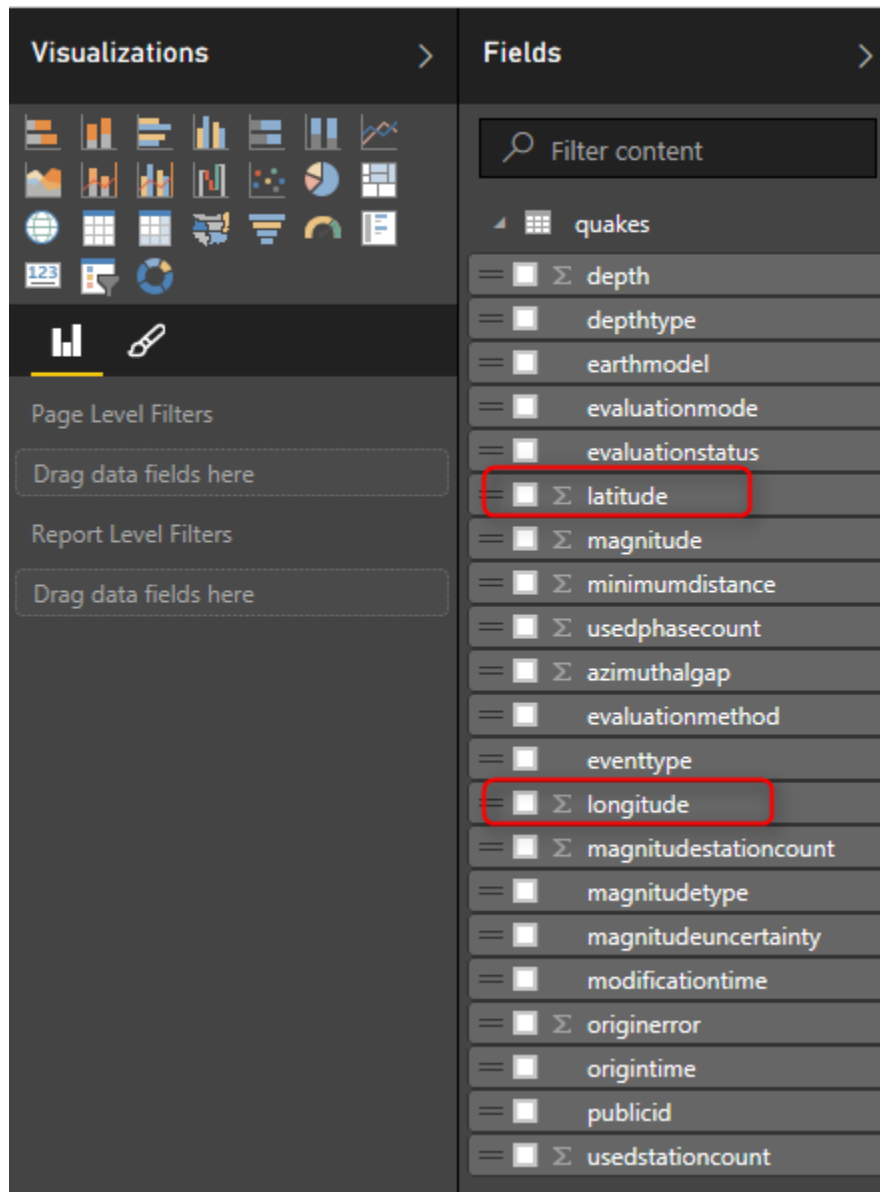
How to Do Power BI Mapping With Latitude and Longitude Only

Published Date: September 24, 2015

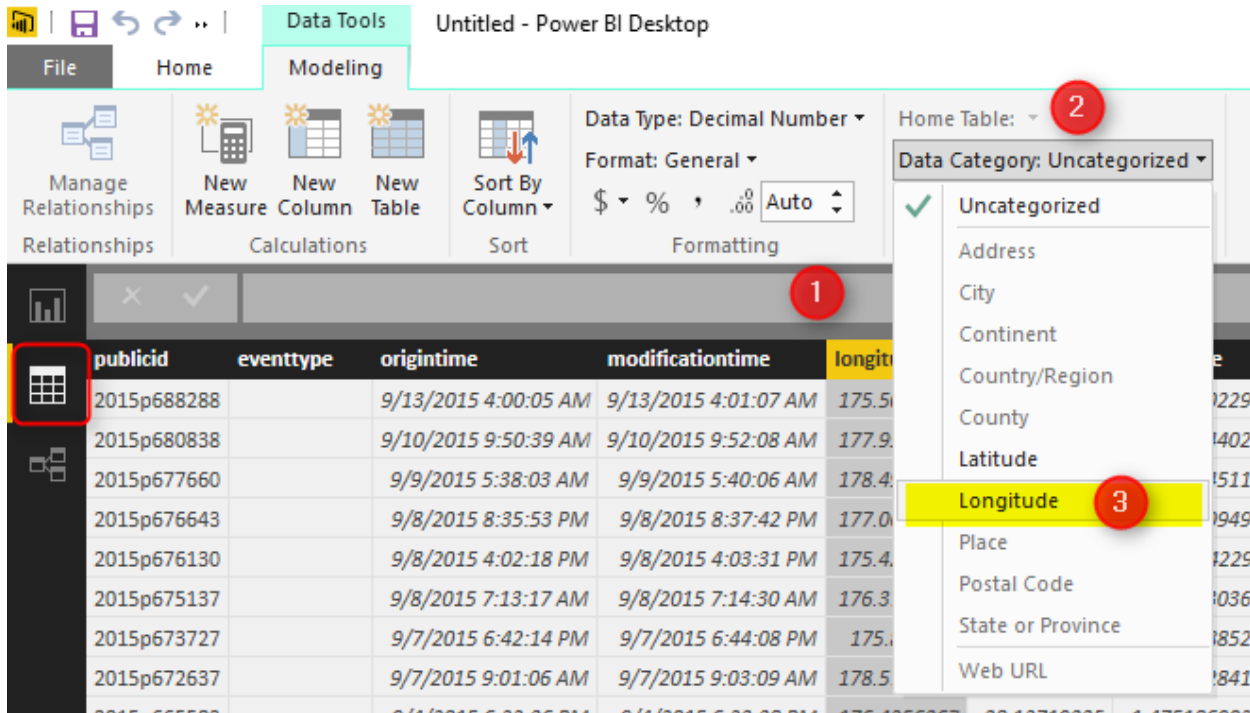


You may have seen many videos or blog posts so far that Power BI Desktop showed the data on the map visualization based on address, suburb, city, state, and country. Fortunately Bing Map helps a lot to search the point on the map based on address fields. However sometimes you don't have address fields, actually in some types of the data there is no address field. As an example an earth quake most of the time happens somewhere deep in the ocean where there is no street address! All you have is latitude and longitude as the Geo graphical data. However latitude and longitude is precise enough for any GPS device to point out the exact location. Unfortunately to your surprise, at this point of the time Power BI Desktop map doesn't support visualization

only based on latitude and longitude! So this post is about how to visualize it with Power BI Desktop map easily.



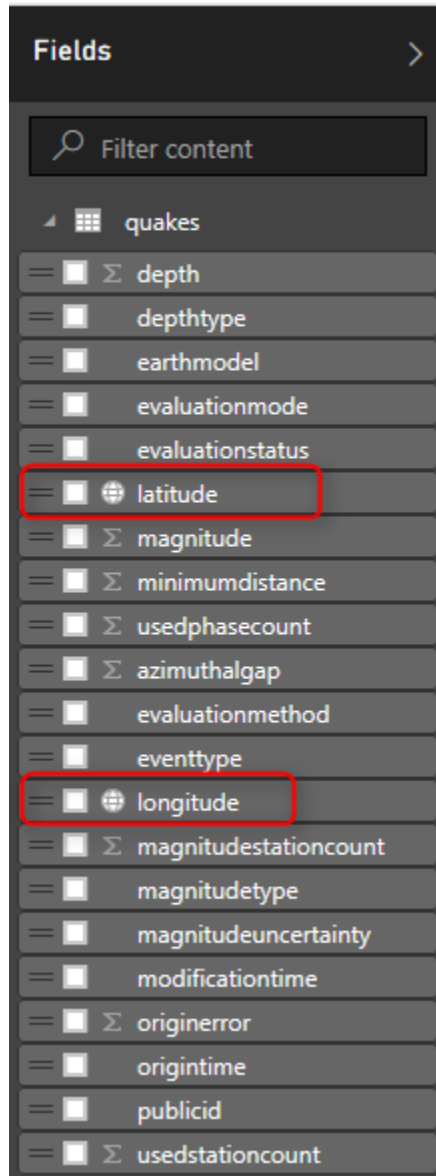
As you see Power BI considered these fields as numeric fields and summarized them in the view (you can see sigma icon besides them in screenshot above). This can be easily changed in the data tab. Select the Column and then change the Data Category under the Modeling menu to respective field (latitude or longitude).



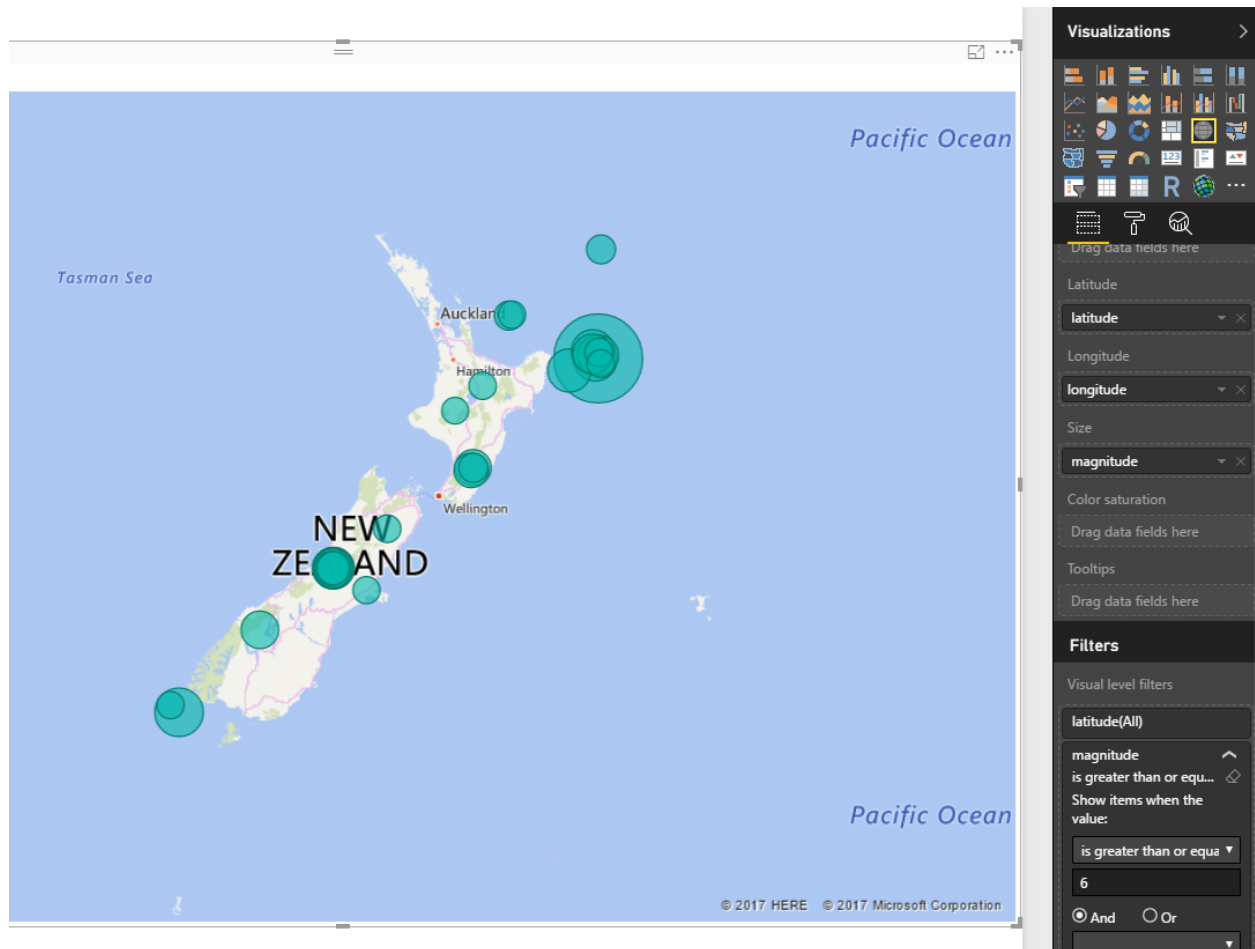
The screenshot shows the Power BI Desktop interface with the Data Tools ribbon selected. The ribbon includes sections for Relationships, Calculations, Sort, and Formatting. A data table is displayed with columns: publicid, eventtype, origintime, modificationtime, and longitude. A red box labeled '1' highlights the table icon in the left sidebar. A red circle labeled '2' highlights the 'Home Table' dropdown menu, which is open, showing a list of data categories. A red circle labeled '3' highlights the 'Longitude' category in the list.

publicid	eventtype	origintime	modificationtime	longitude
2015p688288		9/13/2015 4:00:05 AM	9/13/2015 4:01:07 AM	175.5
2015p680838		9/10/2015 9:50:39 AM	9/10/2015 9:52:08 AM	177.9
2015p677660		9/9/2015 5:38:03 AM	9/9/2015 5:40:06 AM	178.4
2015p676643		9/8/2015 8:35:53 PM	9/8/2015 8:37:42 PM	177.0
2015p676130		9/8/2015 4:02:18 PM	9/8/2015 4:03:31 PM	175.4
2015p675137		9/8/2015 7:13:17 AM	9/8/2015 7:14:30 AM	176.3
2015p673727		9/7/2015 6:42:14 PM	9/7/2015 6:44:08 PM	175.1
2015p672637		9/7/2015 9:01:06 AM	9/7/2015 9:03:09 AM	178.5

Now go back to Report tab. This time you will see these two fields with map icons besides them.



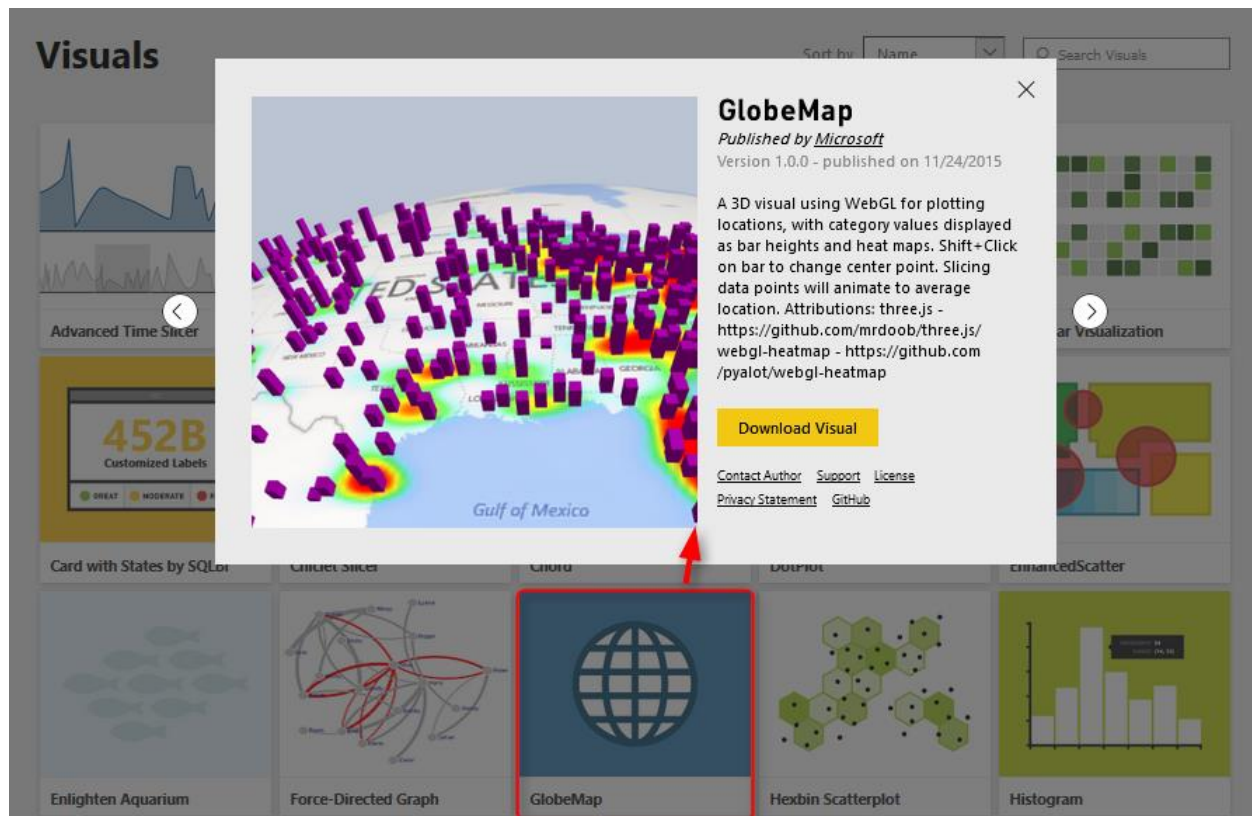
Now let's create a Map. Map works with Latitude and Longitude (which we have fortunately), I also add magnitude as values. For chart below I've also showed magnitude in values, and color saturated it based on depth of the earth quake. as the data rows were too many (19K quakes), I've filtered to show only those that has magnitude greater than 6. Here is the result;



Map still might seems scary for you to see that many earth quakes with more than 6 magnitude in only one year, But if you wanna live in a magnificent country you should take some risk 😊

Power BI Says Hi to 3D Maps

Published Date: November 25, 2015



If you've read my previous articles about Power BI, you know that Power BI has five main components: Power Query, Power Pivot, Power View, Power Q&A, and Power Map. So far all of these components were now supported in PowerBI.com website as part of your Power BI solution except Power Map! Power Map was the only component that you could use in Excel for Office 365 subscription, and it allowed you great 3D Geo-spatial visualization with ability to tell stories with videos based on animated tours. The great news today is that Power BI Desktop now supports 3D map visualization, but is that visualization same as what Power Map exactly does? Let's find out.

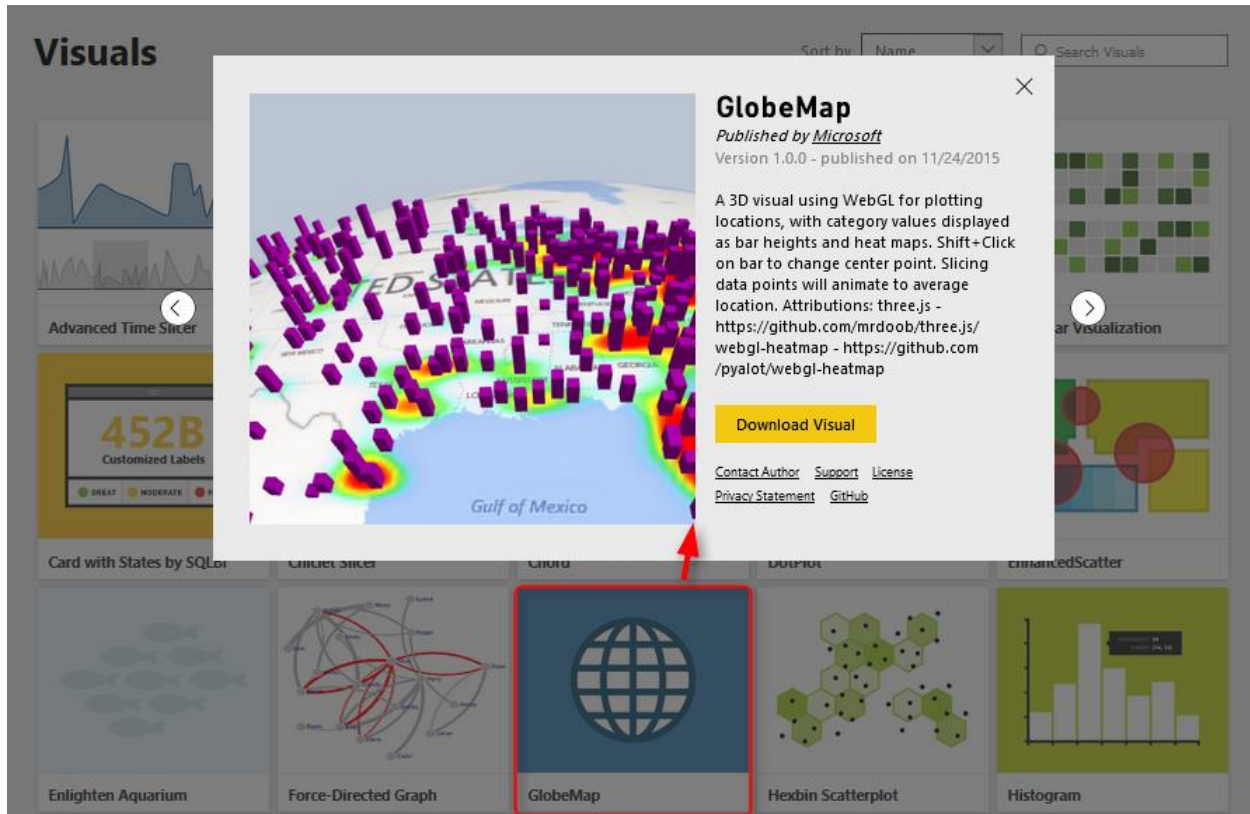
3D Visualization in Power BI Desktop

I've heard of this features first in a session at PASS Summit 2015, and I was looking forward to see the feature be available to run some tests on it. My Particular interest was

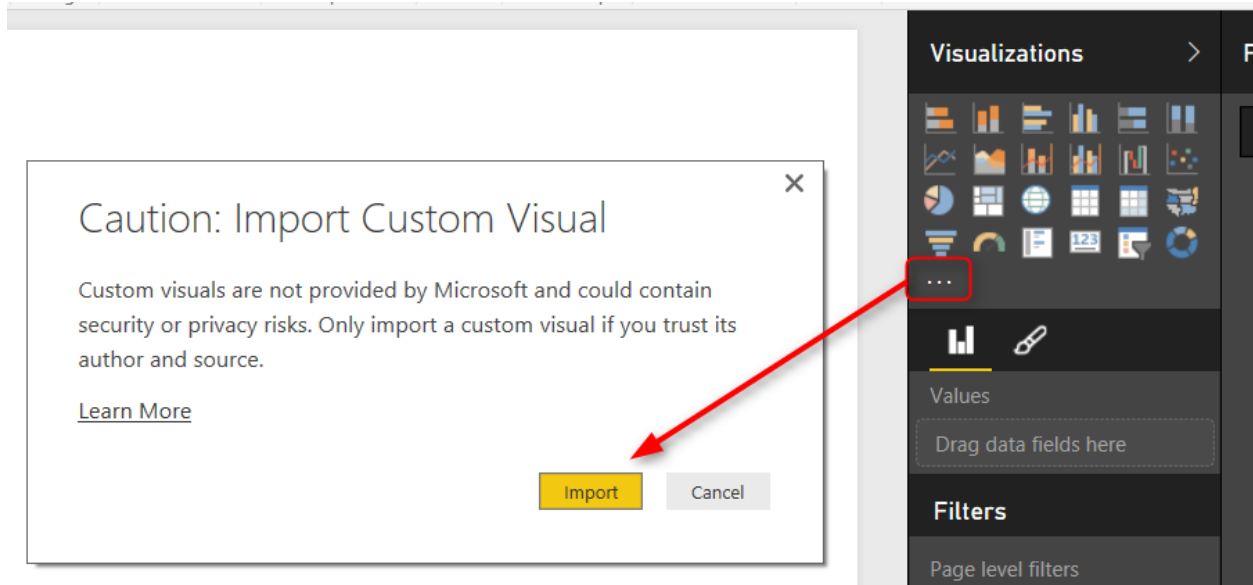
because I've done some interesting projects on Geo-spatial data visualization with Power Map for different clients, and I know what are their requirements in that area.

Fortunately today [Microsoft Power BI team announced](#) that 3D map feature added to Power BI through Custom Visuals. So let's see how it looks like. For using this visual you have to download it first from Power BI Visuals Gallery:

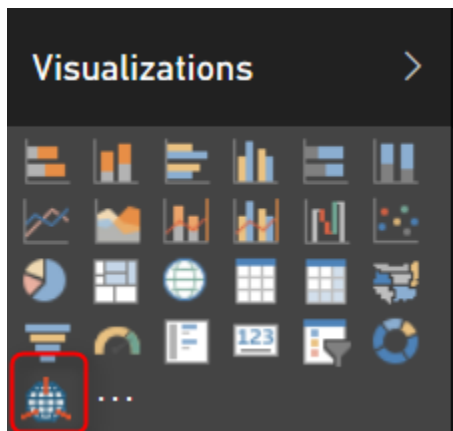
<https://app.powerbi.com/visuals>



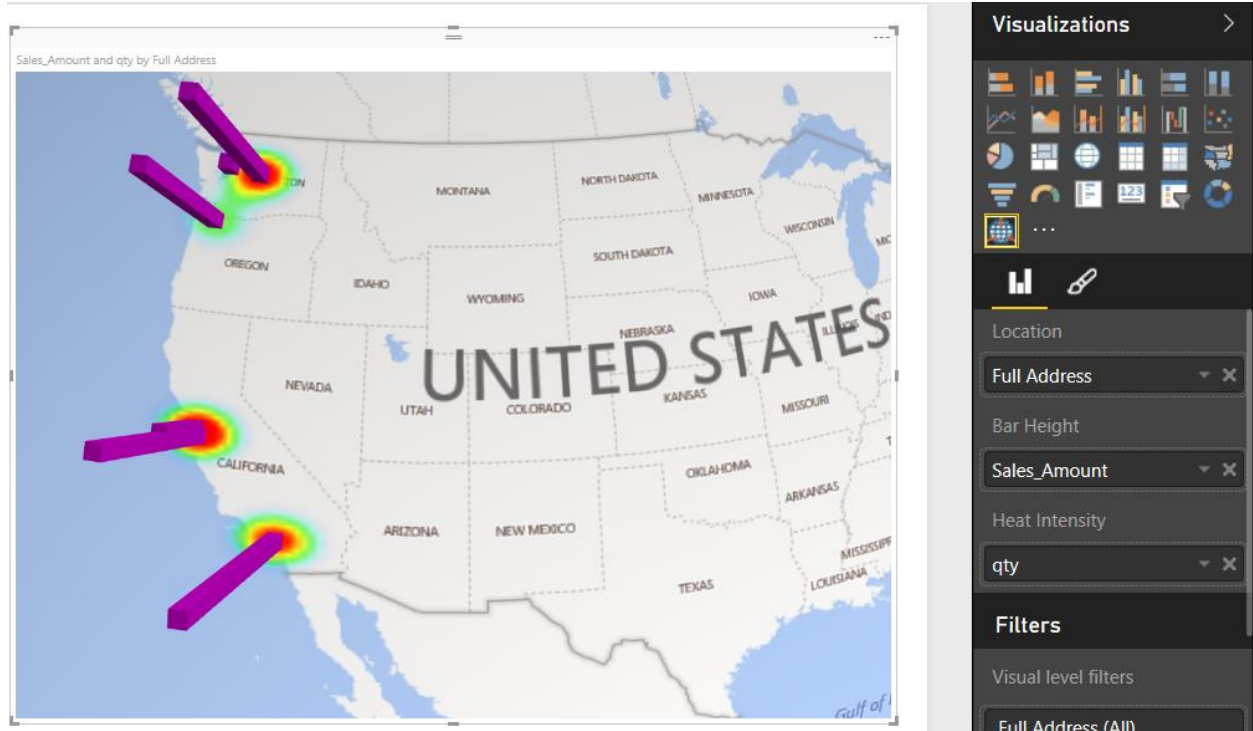
After downloading this visual you can import it into Power BI desktop



Now you see Globe Map visual added to the visualization pane



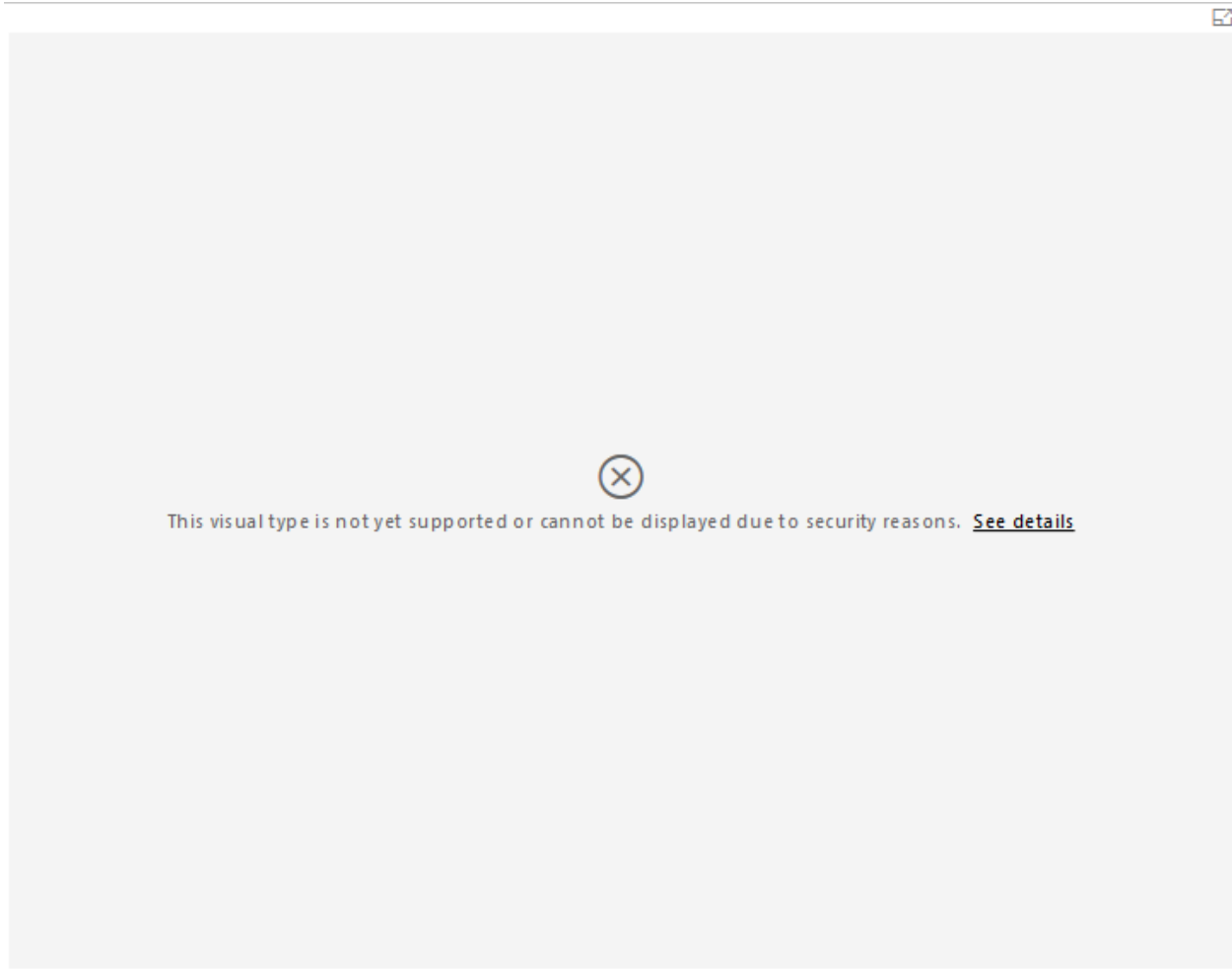
I show my sample [Pubs map demo of Power BI](#) here with Globe and here it is;



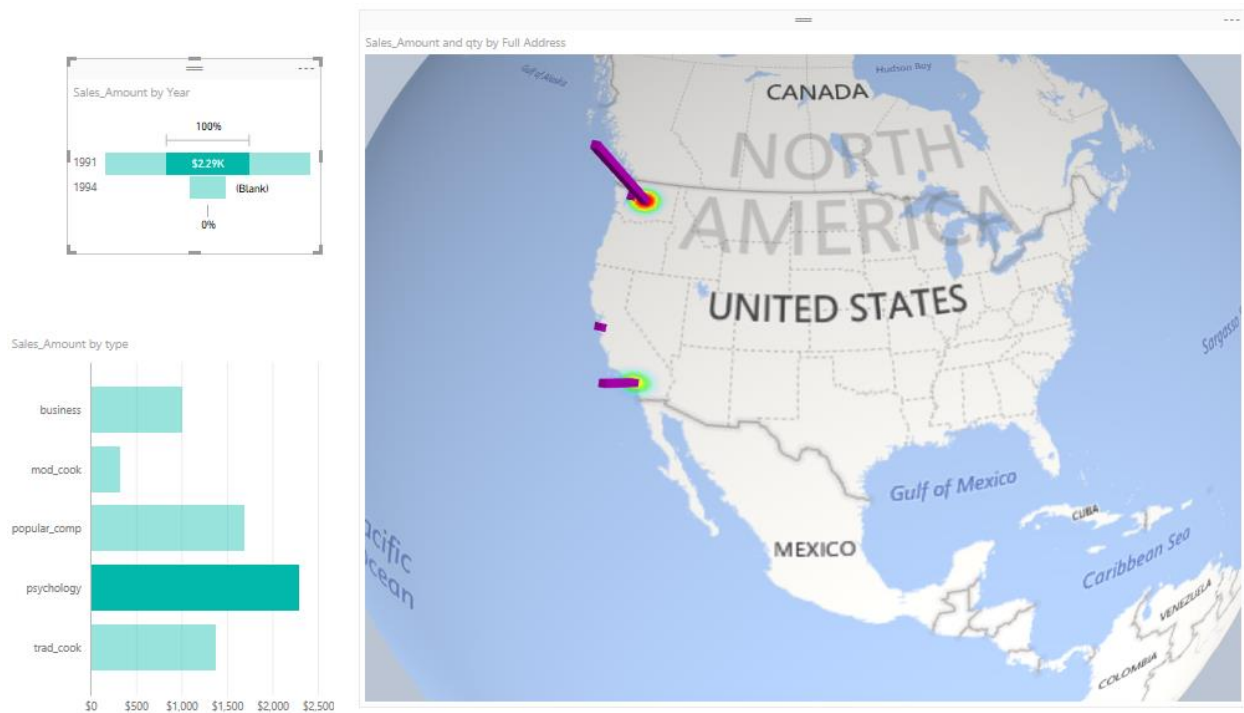
As you see a nice 3D map showed here easily. However there is no particular options for it. it is just a 3D map! If I wanted to zoom into part of the map on a particular angle I cannot do that.

The map however is able to show two measures; one as the height of bar chart, and another one as heat map. you can see that above map shows sales amount as height, and quantity of sales as heat map.

You can not yet see Globe visual in PowerBI website, I reckon that would come very soon. but for now if you publish your report you will see this;

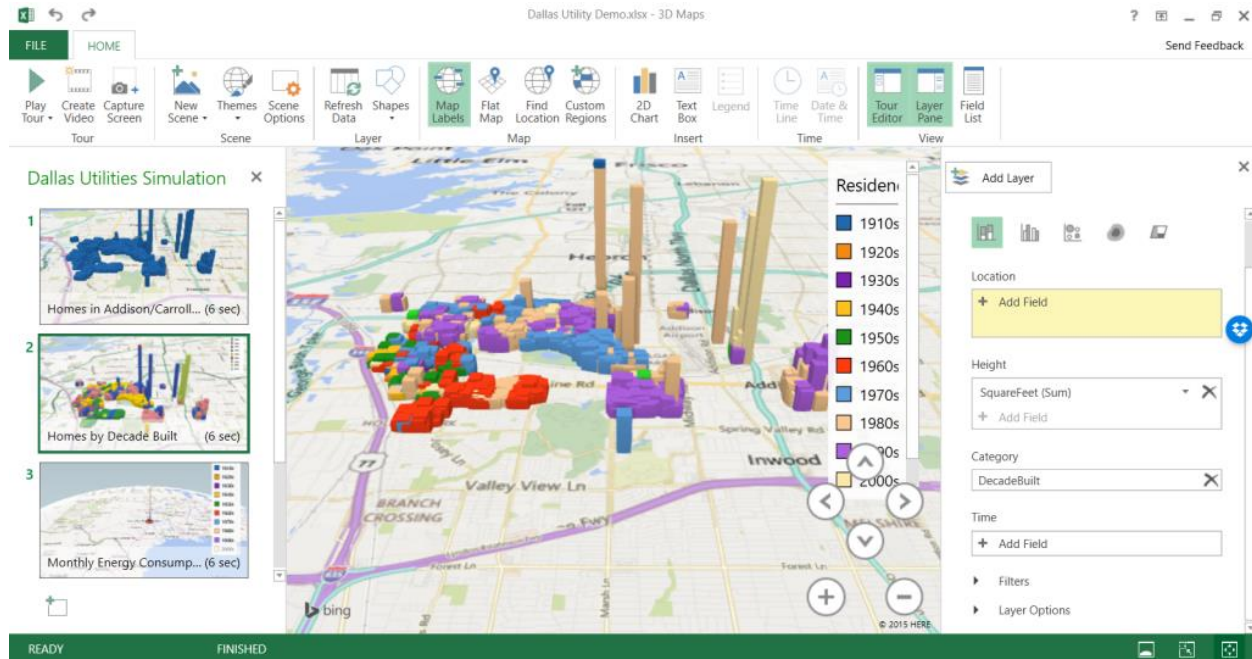


One specific good feature of Power BI Globe visual is that it sync simply with other visualization items in the report (it is not isolated component as Power Map) , so if you click on column chart's specific column you will see only related part of the data for that visualized in map.



Power Map

Well Power Map is much more mature in 3D Geo-Spatial visualization, you can have different layers of visualization (such as column chart and heat map, and region visualization). You can zoom into map on particular angle if you want to. You can have a play axis which is very important for story telling.



Here are some features on Power Map, and the information about either they are or not yet supported in Power BI;

Feature	Power Map	Power BI 3D Globe Visual
3D Geo-Spatial Map	Yes	Yes
Bing Map	Yes	Yes
Column Chart	Yes	Yes
Heat Map	Yes	Yes
Zoom in/out	Yes	Yes
Stacked Column	Yes	No
Bubble Chart	Yes	No
Region Visualization	Yes	No
Synchronization With Other Charts in Report	No	Yes
Customization on size and opacity of columns	Yes	No
Map Labels option	Yes	No
Custom Regions and Custom Map	Yes	No
Scene Creation	Yes	No
Video creation	Yes	No
Camera Movement	Yes	No
Time/Play Axis	Yes	No
Host and Show 3D map on Web	No	No

Conclusion

Power BI team did a great step forward with adding 3D map visual in Power BI Desktop. Thank you Microsoft Power BI team because of that! It is really useful for some scenarios that users need to see visualization on 3D map. However this feature is far behind Power Map features for story telling, creating tours, play axis, and many other features. I

believe that soon many of these features will be added into Power BI. So I say Power BI said hi to 3D maps, but please be quick on that Microsoft team because Power Map raised expectations of our clients to a very high level! They will be looking for same features (at least) in Power BI desktop.

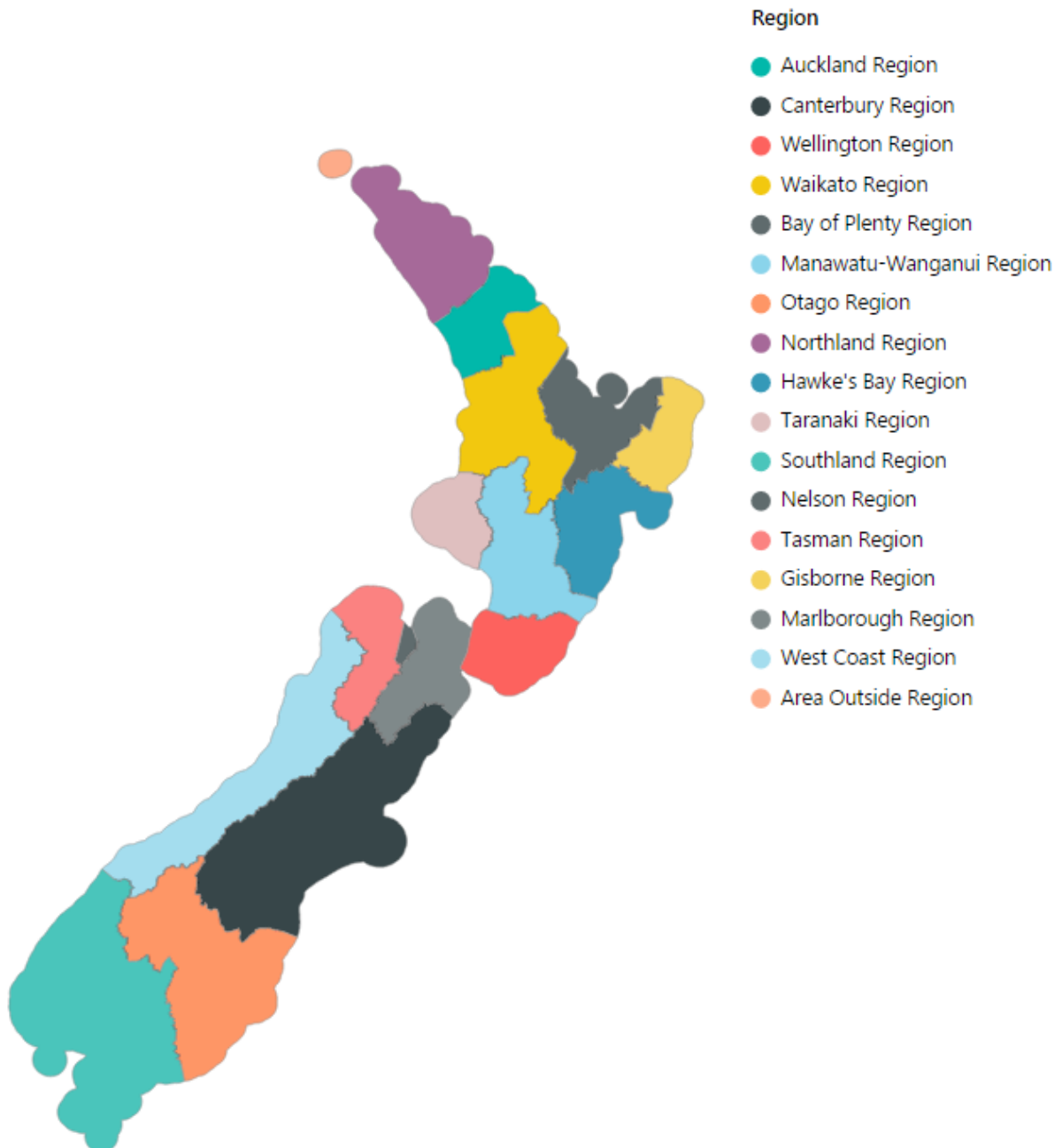
A Question for You

Have you ever used Power Map or any other 3D Geo-Spatial data visualization tools? what are features of that (those) tools that you liked most? what are features that you didn't used? What about Power Map? what are great features of this product from your (or your customers) point of view? and what needs to be added?

Shape Map Better than the Filled Map

Published Date: March 7, 2017

Population by Region and Region



I have previously written some blog posts about [Map visuals](#) in Power BI. One of them was specifically about [Filled Map](#), titled as Filled Map; the Good, the Bad, the Ugly! Why? you need to read that post to find the reason. In this post I want to explain the power of

Shape Map which is one of the visuals Power BI team published recently. This visual is still at preview mode at the time of writing this post. This visual is much more powerful than what it looks. The actual power behind it is that you can have your own map added to it. Let's take a closer look at this visual with an example. If you want to learn more about Power BI; read [Power BI from Rookie to Rock Star](#).

Maps in Power BI

There are many ways to visualize something on Map in Power BI, some of it are as below;

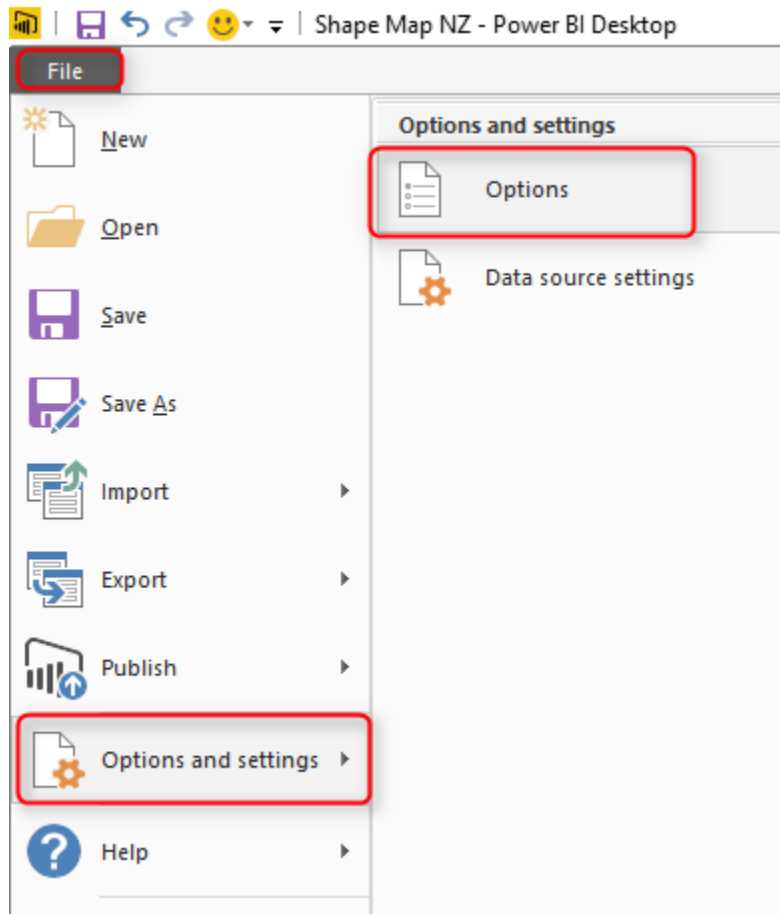
- [Map Visual](#)
- [Filled Map Visual](#)
- [Custom Visuals – Synoptic Panel by OKviz](#)
- [Custom Visuals – GlobeMap by Microsoft](#)
- Shape Map Visual – Preview
- ArcGIS Map Visual – Preview

I have written about them all except last two. This post focus is on Shape Map, and in another post I'll explain ArcGIS visual. Hopefully at the end I will write a post about comparison.

Shape Map

Shape Map is a visual first introduced few months ago with the ability to highlight regions (similar to Filled Map), and the power of adding your own map file to it. This feature makes it different from Filled Map, because the normal filled map visualize anything on a Bing map layer, and if for some reason it can't find that geo location information result won't be great with that. If you want to see an example of that, read this post; [Filled Map, the good, the bad, the ugly!](#) Don't get me wrong; Filled Map is good at some stage, but there are better ways some other times.

Because Shape Map is still in preview mode (at least at the time of writing this post), you need to enable it in Options section of Power BI Desktop; From File Menu -> Options and Settings -> Options.



You need to select Shape Map in the list of Preview Features tab;

Options

GLOBAL

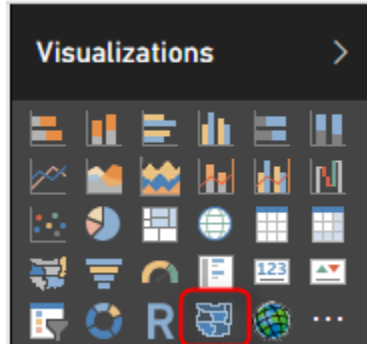
Data Load
Query Editor
DirectQuery
R Scripting
Security
Privacy
Updates
Usage Data
Diagnostics
Preview Features
Auto Recovery

Preview Features

The following features are available for you to try in this release. Preview features might change or be removed in future releases.

- ☐ Amazon Redshift [Learn more](#)
- ☐ Impala [Learn more](#)
- ☐ Snowflake [Learn more](#)
- ☒ Shape Map Visual [Learn more](#)
- ☒ Enable cross filtering in both directions for DirectQuery [Learn more](#)
- ☒ Clustering [Learn more](#)
- ☒ ArcGIS Maps for Power BI [Learn more](#)

After clicking OK, you need to close your Power BI Desktop and re-open it again. You should be able to see Shape Map visual in list of your visuals now.



Sample Data Set and Demo Files

For this example I will be using an Excel file including all New Zealand regions and population of each region. Why New Zealand? Regions of New Zealand can't fit great into filled map (I've explained it [here](#)), so it can be a very good example for Shape Map then. You can download all files for this demo [here](#);

Enter Your Email to download the file (required)

[Download]

How Shape Map Works?

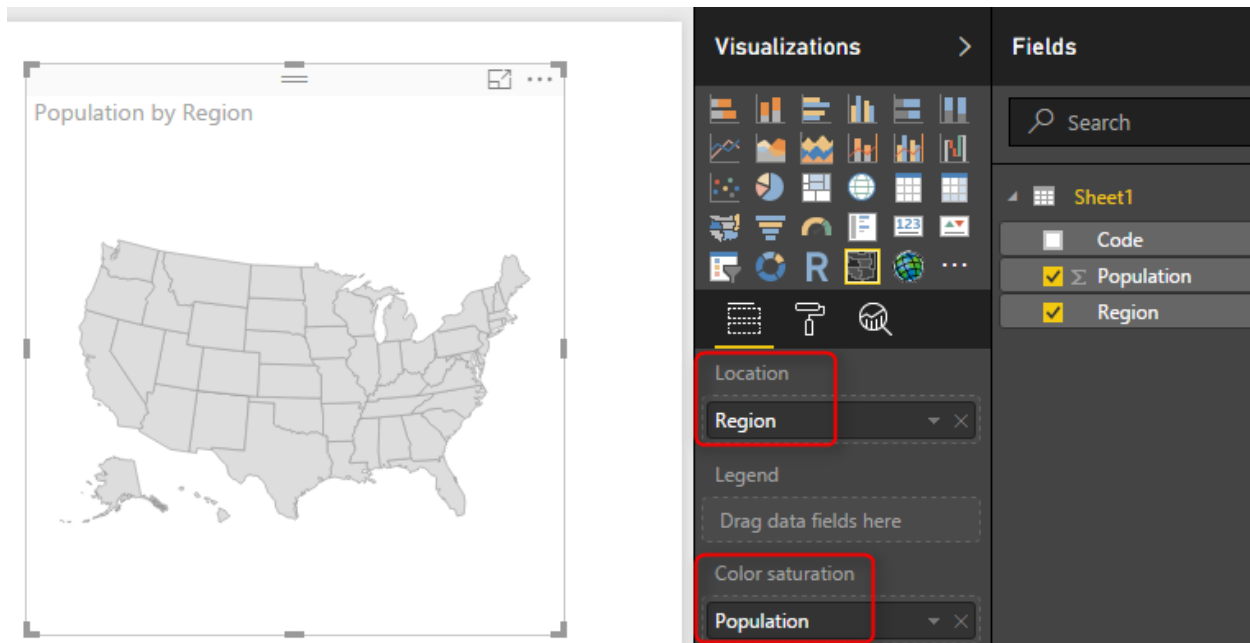
After enabling the preview feature of Shape Map, get the data from Regions Excel file, which includes data set as below;

Region	Code	Population
Nelson Region	17	50600
Marlborough Region	18	45500
Northland Region	01	171400
Auckland Region	02	1614300
Waikato Region	03	449200
Bay of Plenty Region	04	293500
Gisborne Region	05	47900
Hawke's Bay Region	06	161500
Taranaki Region	07	116600
Manawatu-Wanganui Region	08	234500
Wellington Region	09	504900
West Coast Region	12	32600
Canterbury Region	13	600100
Otago Region	14	219200
Southland Region	15	98000
Tasman Region	16	50300
Area Outside Region	99	0

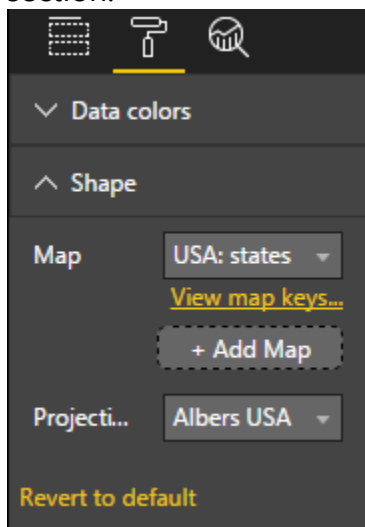
When you get data from Power BI Desktop from above data set, Power BI automatically change the data types to preferred data types. Make sure that you get correct data types; Region and Code should be Text, and Population should be Whole number;

	^A _C Region	^A _C Code	¹ ₂ ₃ Population
1	Nelson Region	17	50600
2	Marlborough Region	18	45500
3	Northland Region	01	171400
4	Auckland Region	02	1614300

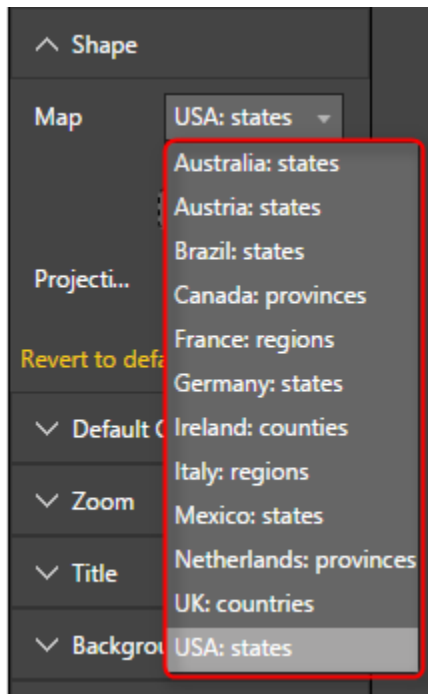
Close and Apply query editor, and drag and drop a Shape Map visual in the report page. To get the first view of this visual, you need to drag and drop a field into it. Start with using Region as the Location, and Use Population as Color Saturation.



As you can see in above image, the first view would be States in USA, which is not exactly what we are looking for. The reason is that Shape Map works based on a predefined map. To select to map, go to Format of this visual, and expand Shape section.



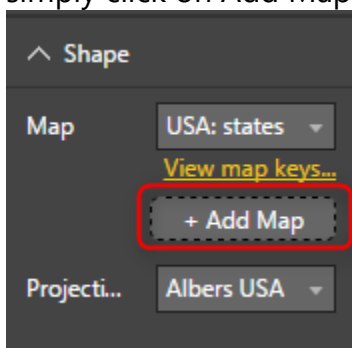
Now you can see that the default Map view is USA: States. You can simply change it to other maps in the list;



There are few maps/shapes in this list, but obviously there are many options missing. For example; New Zealand regional map is missing. Even if you want to use USA Cities Map, or Cities of a specific state in USA you can't find it here. Don't turn around with seeing that, Shape Map allows you to add your own Map. If your data set is dealing with any of existing maps, then you don't need to add a custom map for it. Just use the related map, and you are good to go. If your map is not there, here is the guide to add it;

Add a Custom Map

Shape Map allows you add your own custom map! For using this option you need to simply click on Add Map in Shape section of Format of this visual.



The map should be a TOPO JSON file. TOPO JSON is a standard for defining geo location information. In this type of file, information such as boundaries of each region and name/id can be stored. Here is an example TOPO JSON content;


```
{ "objects": { "05720e46d7d482ca89887573389a5d41": { "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
  "type": "GeometryCollection", "geometries": [ { "type": "Polygon", "properties": { "REGC12": "17", "NAME": "Nelson Region" }, "arcs": [ [ 0, 1, 2, 3, 4, 5, 6 ] ] }, { "type": "Polygon", "properties": { "REGC12": "18", "NAME": "Marlborough Region" }, "arcs": [ [ 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63 ] ] }, { "type": "Polygon",
  "properties": { "REGC12": "01", "NAME": "Northland Region" }, "arcs": [ [ 64, 65 ] ] }, { "type": "Polygon", "properties": { "REGC12":
  "02", "NAME": "Auckland Region" }, "arcs": [ [ 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
  84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95 ] ] }, { "type": "Polygon", "properties": { "REGC12": "03", "NAME": "Waikato Region"
  }, "arcs": [ [ 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
  121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999 ] ] ] }
```

As you can see in the above screenshot, information such as boundaries of each region plus region information listed in the file with JSON format. Don't panic! You don't need to write JSON to create this file, there are much easier ways;

Where to Find Map?

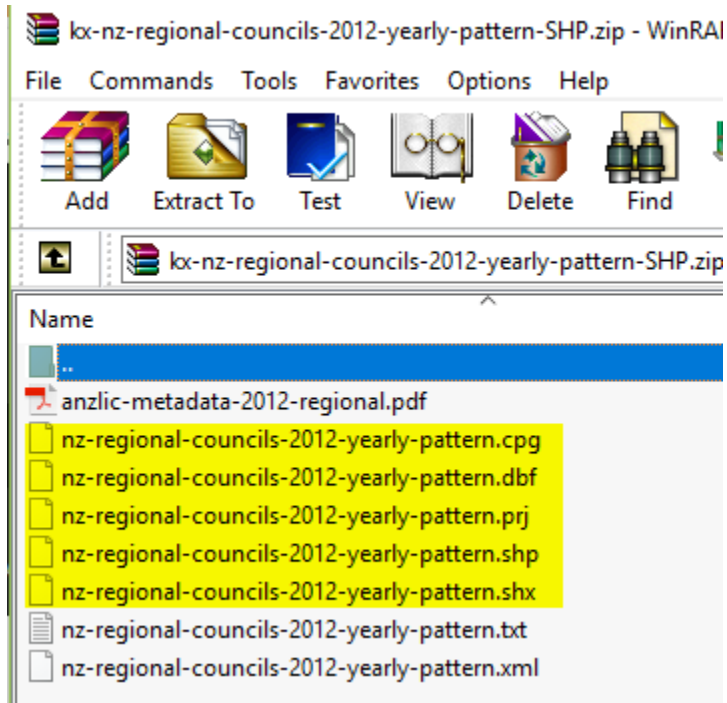
Normally you can easily find Shape files of every map over internet. For example, here is where I have found Shape files for New Zealand Region map;

<https://koordinates.com/layer/4240-nz-regional-councils-2012-yearly-pattern/download/>



The screenshot shows the Koordinates.com website interface. On the left, a sidebar displays the dataset title "NZ Regional Councils (2012 Yearly Pattern)" and its source "Statistics New Zealand". It also shows the date added (18 May 2012) and usage statistics (3537 views, 805 downloads). Below this are buttons for "EXPORT LAYER" and "ADD TO COLLECTION". The main area features a map of New Zealand with regional council boundaries highlighted in blue. Various cities are labeled on the map, including Auckland, Hamilton, Tauranga, Rotorua, Palmerston North, Wellington, Nelson, Christchurch, Dunedin, Invercargill, and Queenstown. The Tasman Sea is labeled to the west of the North Island. The top of the page has a search bar and navigation tabs for "MAP" and "CROP".

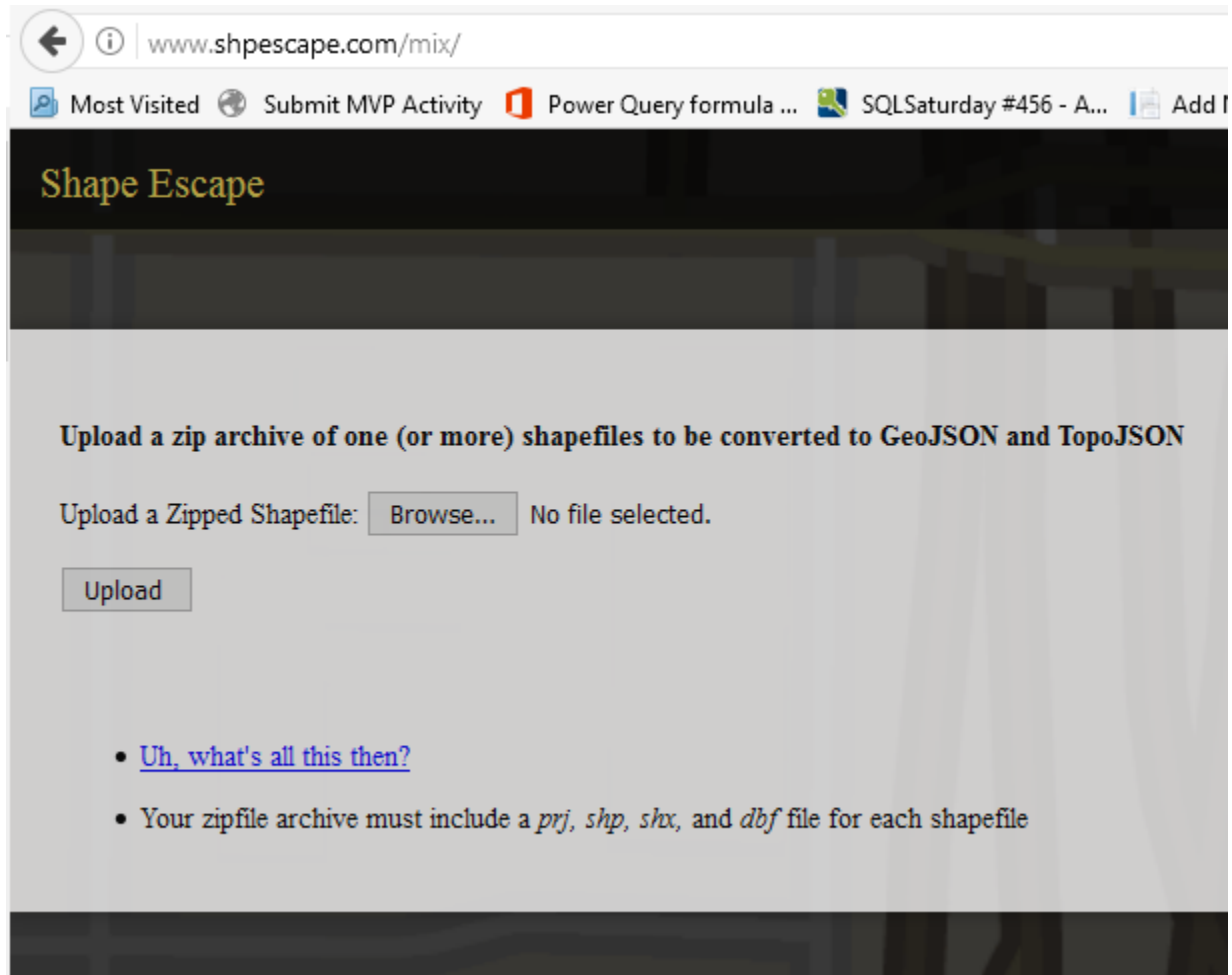
In the Koordinates.com website, you can find many other maps, and there are also heaps of other websites that give you Shape file for the map. Shape files cannot be used directly with Shape Map. Shape files are normally a zip file including a bunch of files;



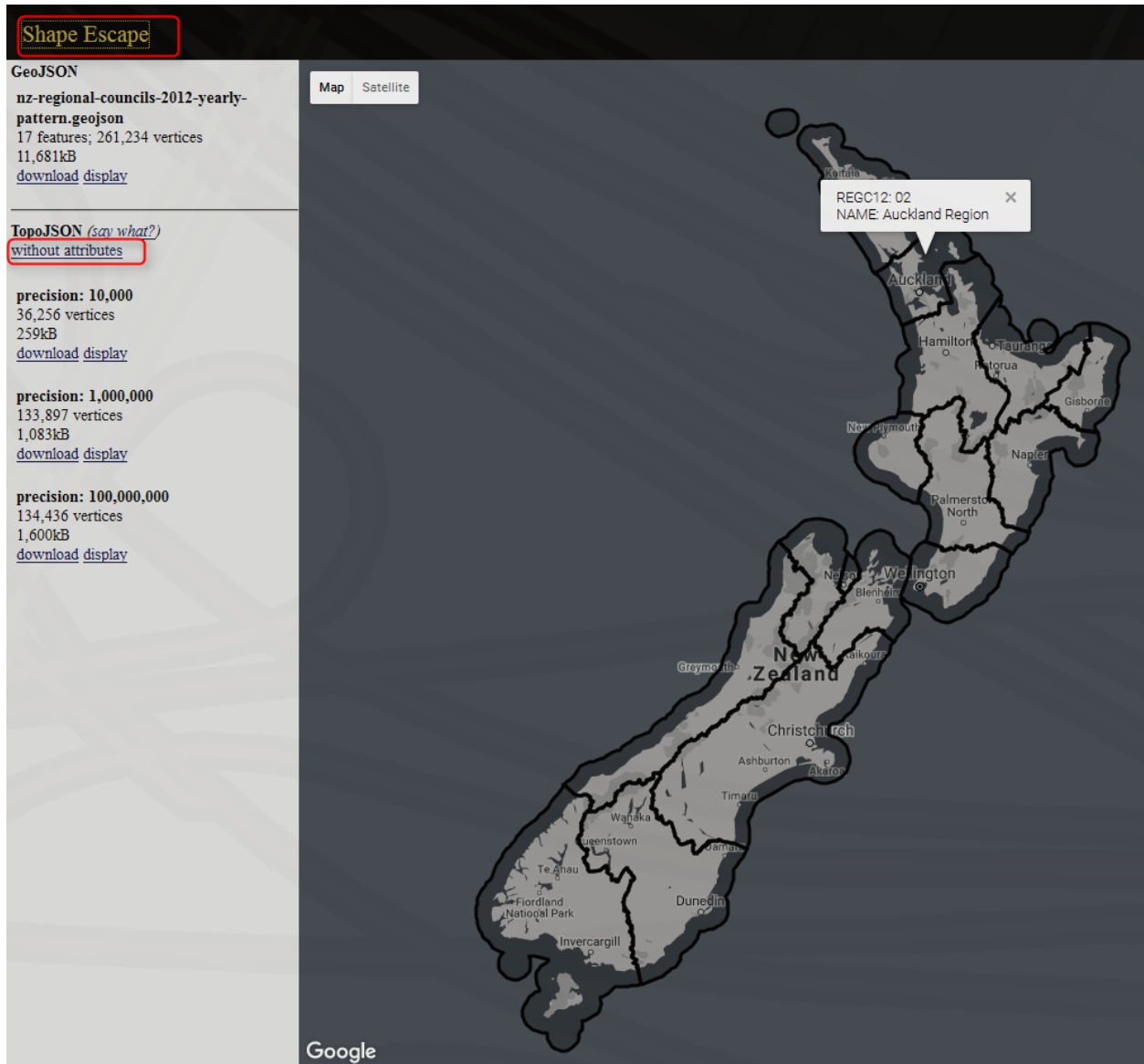
Not all files above are normally part of Shape file zipped directory, but normally .shp, .shx, and .dbf are. Once you have the shape file, you can easily convert it to TOPO JSON. there are heaps of converters for it. Here is one of them which one of students in my courses found it and works perfectly;

Convert Shape File to JSON

There are many websites that does the conversion of Shape File to TOPO JSON. as an example [ShapeEscape](#) does that perfectly;

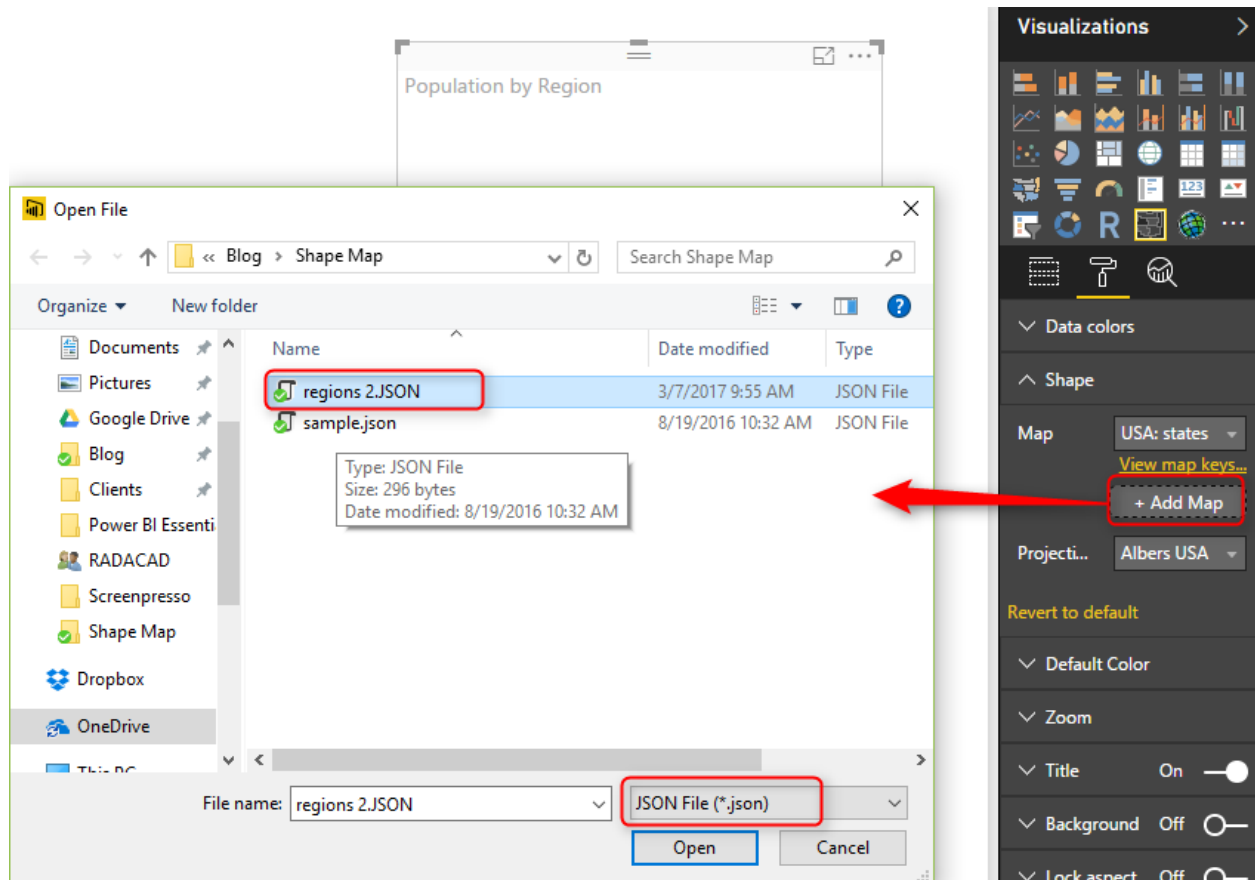


All you need to do is to upload shape file (the zip file including everything) into the website, and it will simply show you the map. make sure that you have selected TopoJSON map WITH attributes. To verify it; when you click on a region you should see a pop up text section showing name and id of that region (like screenshot below).

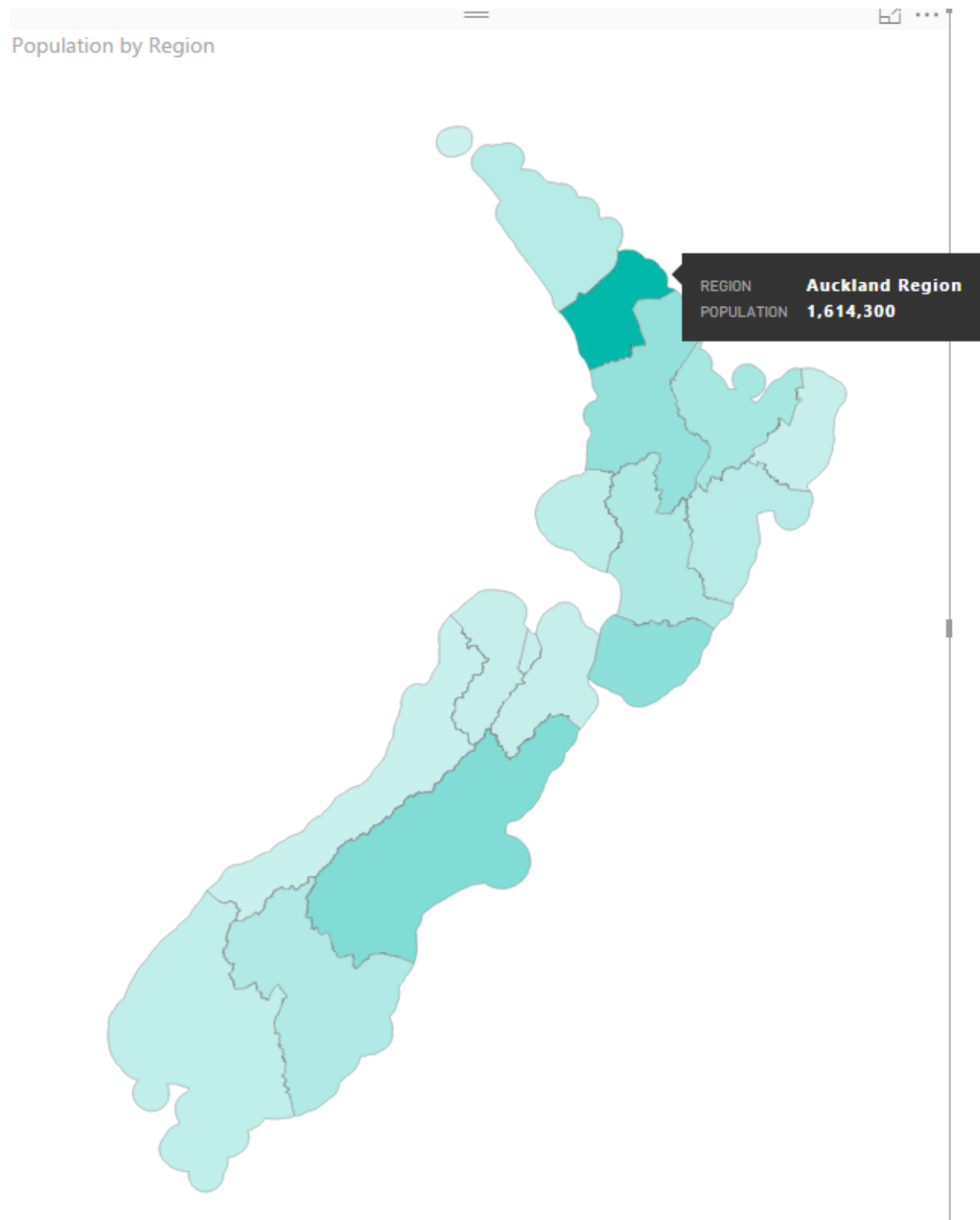


Maps can have variation of details, and depends on that the file size changes, you can select whatever details level you want. for this example, I took the most basic one with precision of 10,000. When you click on download, you can copy the entire JSON code into a text file, and put a .JSON extension for it.

Now you can simply add the custom map to Shape Map visual;

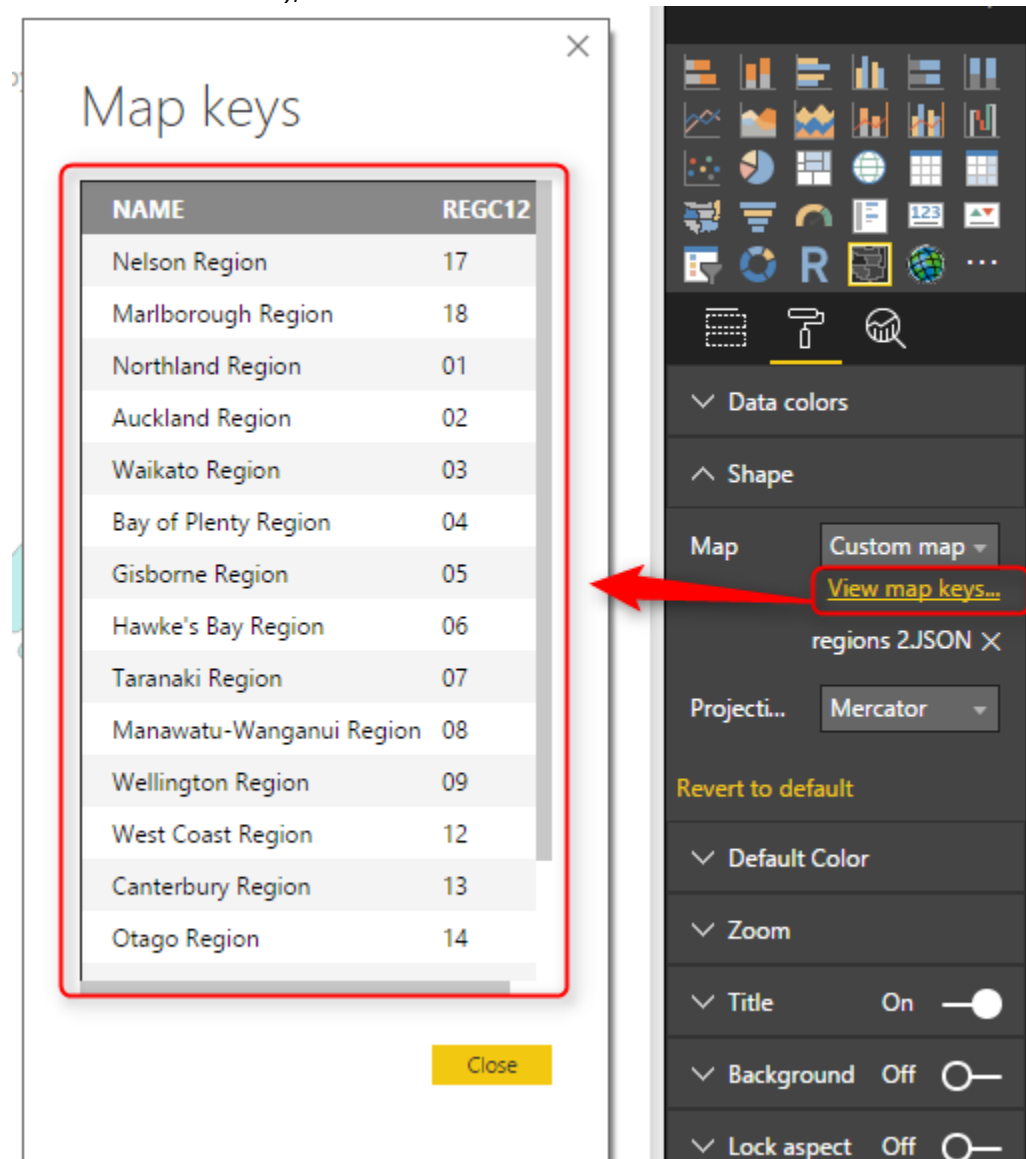


It will take few seconds for map to load, and then you will see your regional map visualized:



If you cannot see regions color coded, probably the attributes part of map file is missing. Verify that in Format -> Shape section with clicking on "View Map Keys". You should see region names and their keys. If you see nothing, or names are not correctly

populated, then try to convert shape file again (make sure that you get the converted file WITH attributes);



Pros and Cons of Shape Map

Shape Map despite its simplicity, is very powerful. Think about ability to add your own map, and that means no limit! All you need is just some tools to build your map, and convert it to Topo JSON format. There are many free or paid tools for it. There are however some limitations for this visual, which I hope get fixed before the general availability 😊 Here are pros and cons in nutshell;

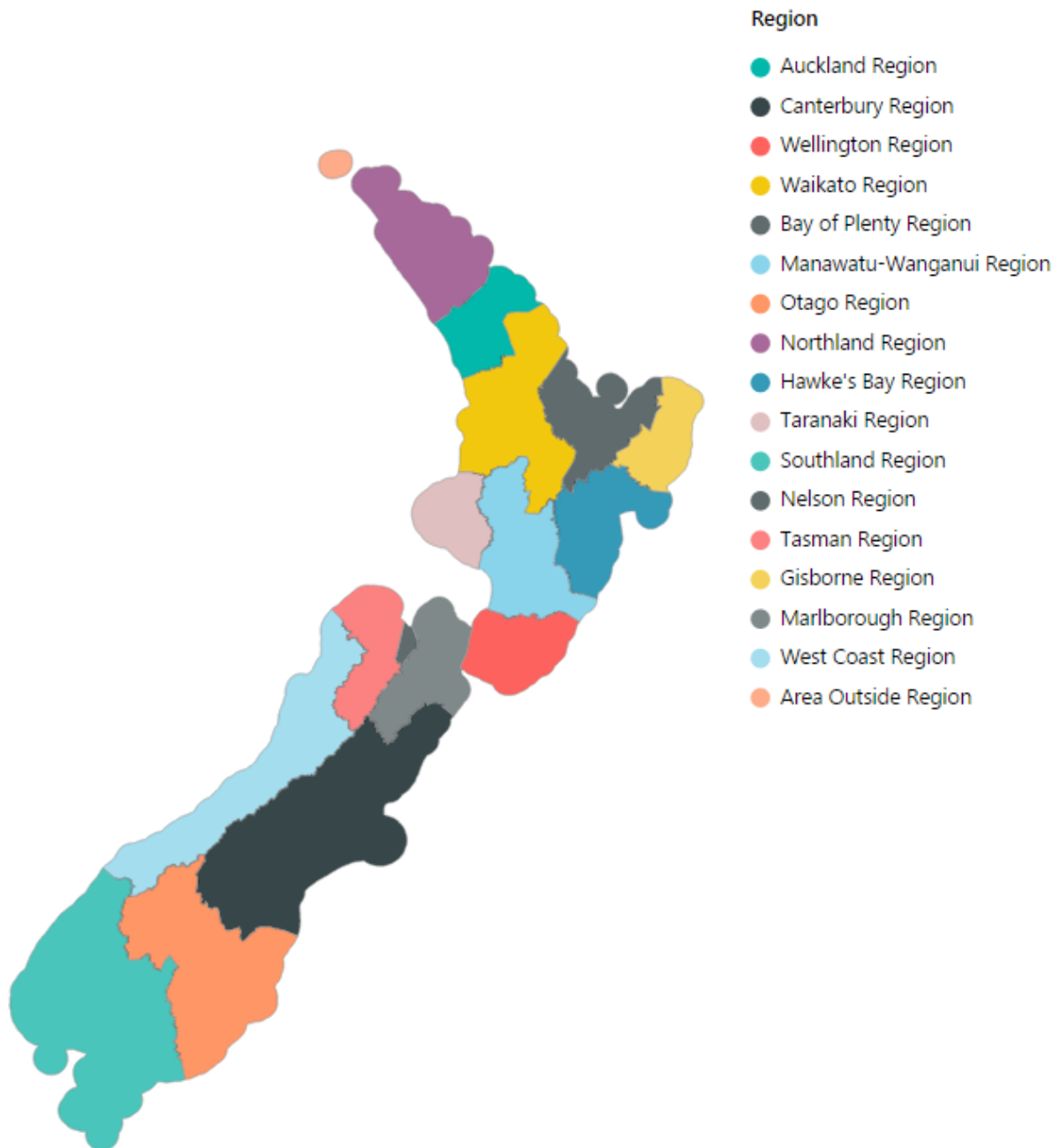
Advantage: Add any Map!

It is not common feature in Power BI map visuals, that you can add your own map. I have previously explained about [SynopticPanel visual by OKviz](#) which allows you to do that. However that is a custom visual, and this one is a built-in visual. and this one works with the Topo JSON standard of geo location.

Limitation: General Map Features

This visual is still in preview mode, and it lacks of many features that can be available for map visuals. for example, you can specify different colors for every region through Legend. However drill down/up is not possible. You cannot add other layers of map to it (for example if you want to add heatmap, or bubbles to it).

Population by Region and Region



Summary

In summary Shape map in my point of view is a powerful map visual for Power BI especially because of the feature to add your own map. Finding a map isn't easy with help of Google and websites that does the conversion (such as ShapeEscape). This visual

still has some limitations due to preview mode of it, which I believe will be sorted in next few months.

Power BI Filled Map; the Good, the Bad, and the Ugly

Published Date: March 23, 2016



Power BI utilizes two built-in map charts; Map and Filled Map. Filled Map is a map that shows data points as Geo-spatial areas rather than points on map. Areas can be continent, country, region, state, city, or county. Working with Filled Map however isn't as easy and convenient as the map chart is. You need to set the data category of fields in the data model appropriately to get the Filled Map working correctly. There are also

some limitations in filled map with countries that address hierarchy is different from State, City, County. In this post I'll get you through some tips of using filled map in your Power BI solution. I also show you some existing issues in the current version of filled map and things that you need to be aware of when you work with this visualization element.

The Good: How to Use Filled Map

As I mentioned earlier working with filled map requires a bit more steps than the regular map. Let's go through that with a real example; [This Wikipedia page](#) includes list of all regions in New Zealand with their population. So let's create a filled map for this data to show areas with their population.

✦	Region	✦	Regional council	✦	Chair	✦	Council seat	✦	Island	✦	Area (km ²) ^[19]	✦	Population ^[20]	✦	ISO 3166-2 Code	✦
1	Northland		Northland Regional Council		Bill Shepherd ^[21]		Whangarei		North		12,498		168,300		NZ-NTL	
2	Auckland ⁽¹⁾		Auckland Council		Len Brown		Auckland		North		4,940		1,570,500		NZ-AUK	
3	Waikato		Waikato Regional Council		Paula Southgate		Hamilton		North		23,900		439,200		NZ-WKO	
4	Bay of Plenty		Bay of Plenty Regional Council		John Cronin		Whakatāne		North		12,071		287,100		NZ-BOP	
5	Gisborne ⁽¹⁾⁽²⁾		Gisborne District Council		Meng Foon		Gisborne		North		8,386		47,400		NZ-GIS	
6	Hawke's Bay		Hawke's Bay Regional Council		Fenton Wilson		Napier		North		14,137		160,100		NZ-HKB	
7	Taranaki		Taranaki Regional Council		David MacLeod		Stratford		North		7,254		115,800		NZ-TKI	

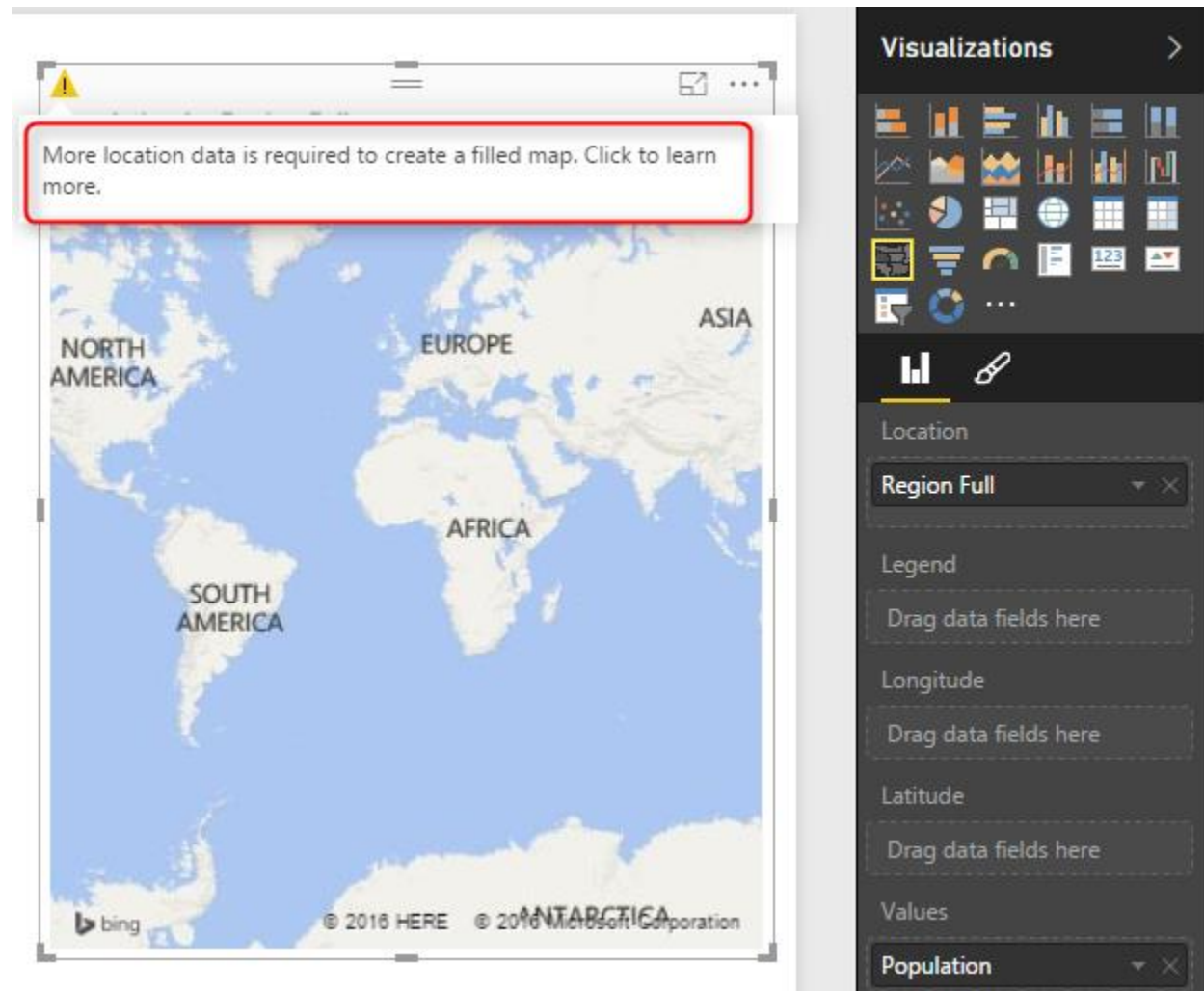
I can use Get Data -> From Web and use the URL of the Wikipedia page to explore the data set. I can remove some extra columns and rows in Query Editor window easily (If you don't know how to work with Query Editor read the [Power Query introduction](#) part of this Power BI series). I also add a custom column called Full Region with the country name attached to the end of region name (because there might be another Auckland in another country which is not what I want to show in my map). So here is what I have at my query at the end:

`= Table.AddColumn("#Changed Type1", "Region Full", each [Region]&"", New Zealand)`

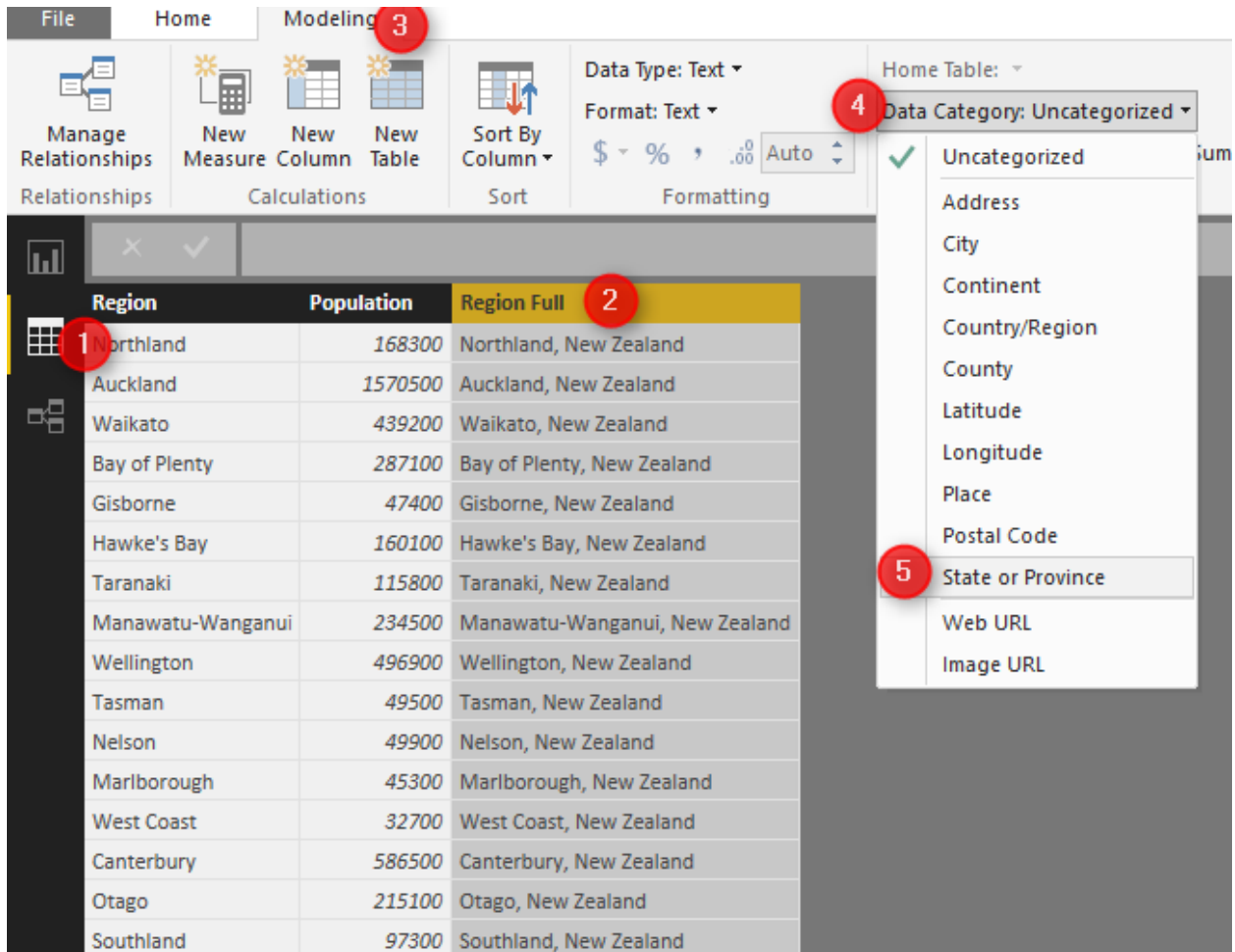
	Region	Population	Region Full
1	Northland	168300	Northland, New Zealand
2	Auckland	1570500	Auckland, New Zealand
3	Waikato	439200	Waikato, New Zealand
4	Bay of Plenty	287100	Bay of Plenty, New Zealand
5	Gisborne	47400	Gisborne, New Zealand
6	Hawke's Bay	160100	Hawke's Bay, New Zealand
7	Taranaki	115800	Taranaki, New Zealand
8	Manawatu-Wanganui	234500	Manawatu-Wanganui, New Zealand
9	Wellington	496900	Wellington, New Zealand
10	Tasman	49500	Tasman, New Zealand
11	Nelson	49900	Nelson, New Zealand
12	Marlborough	45300	Marlborough, New Zealand
13	West Coast	32700	West Coast, New Zealand
14	Canterbury	586500	Canterbury, New Zealand
15	Otago	215100	Otago, New Zealand
16	Southland	97300	Southland, New Zealand

Now if I directly use Region Full in a Filled Map, I'll see an error message that says; More location data is required to create a filled map. Click to learn more.

Details: To create a filled map, the location data should include Bing Map-supported geographic data, such as country/region, state/province, or postal code.



Configuring a data field as a geographic field is easy; All I need to do is to go to Data Tab, click on my query, and then on my data field. Then from the Modeling tab under the Properties pane set Data Category to State or Province.



The screenshot shows the Power BI Desktop Modeling ribbon. The 'Data Category' dropdown is open, showing a list of categories. The 'State or Province' option is highlighted. The table below shows the data being modeled.

Region	Population	Region Full
Northland	168300	Northland, New Zealand
Auckland	1570500	Auckland, New Zealand
Waikato	439200	Waikato, New Zealand
Bay of Plenty	287100	Bay of Plenty, New Zealand
Gisborne	47400	Gisborne, New Zealand
Hawke's Bay	160100	Hawke's Bay, New Zealand
Taranaki	115800	Taranaki, New Zealand
Manawatu-Wanganui	234500	Manawatu-Wanganui, New Zealand
Wellington	496900	Wellington, New Zealand
Tasman	49500	Tasman, New Zealand
Nelson	49900	Nelson, New Zealand
Marlborough	45300	Marlborough, New Zealand
West Coast	32700	West Coast, New Zealand
Canterbury	586500	Canterbury, New Zealand
Otago	215100	Otago, New Zealand
Southland	97300	Southland, New Zealand

After the change when I get back to the Filled map I can see that it illustrates regions on the map nicely.

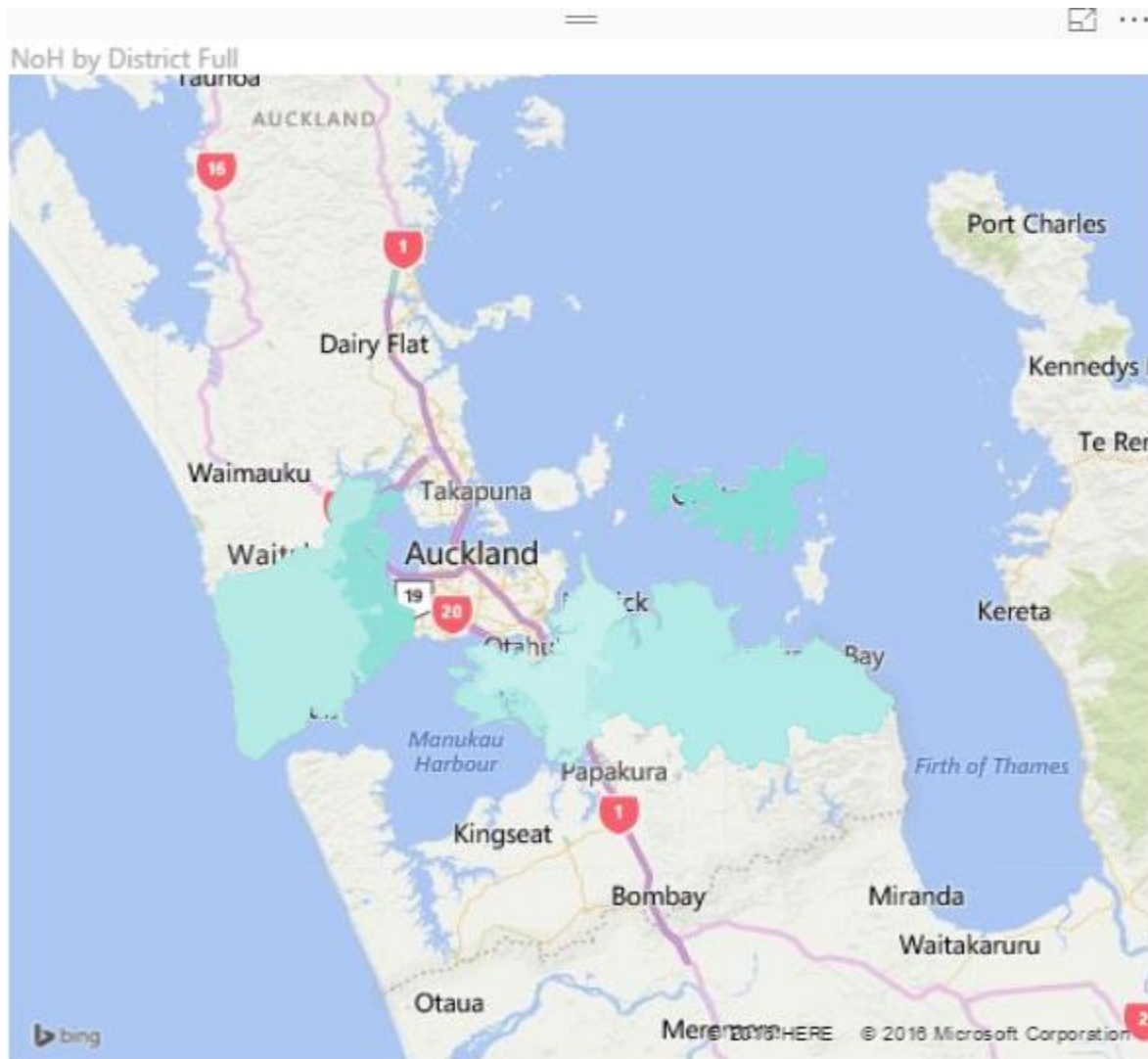


As you see in the Data Category options you can set the field to be Continent, Country/Region, State or Province, City, County, Post Code.... If you have the geographic values in your data set then you can easily map them to these data categories and filled map will draw them nicely for you.

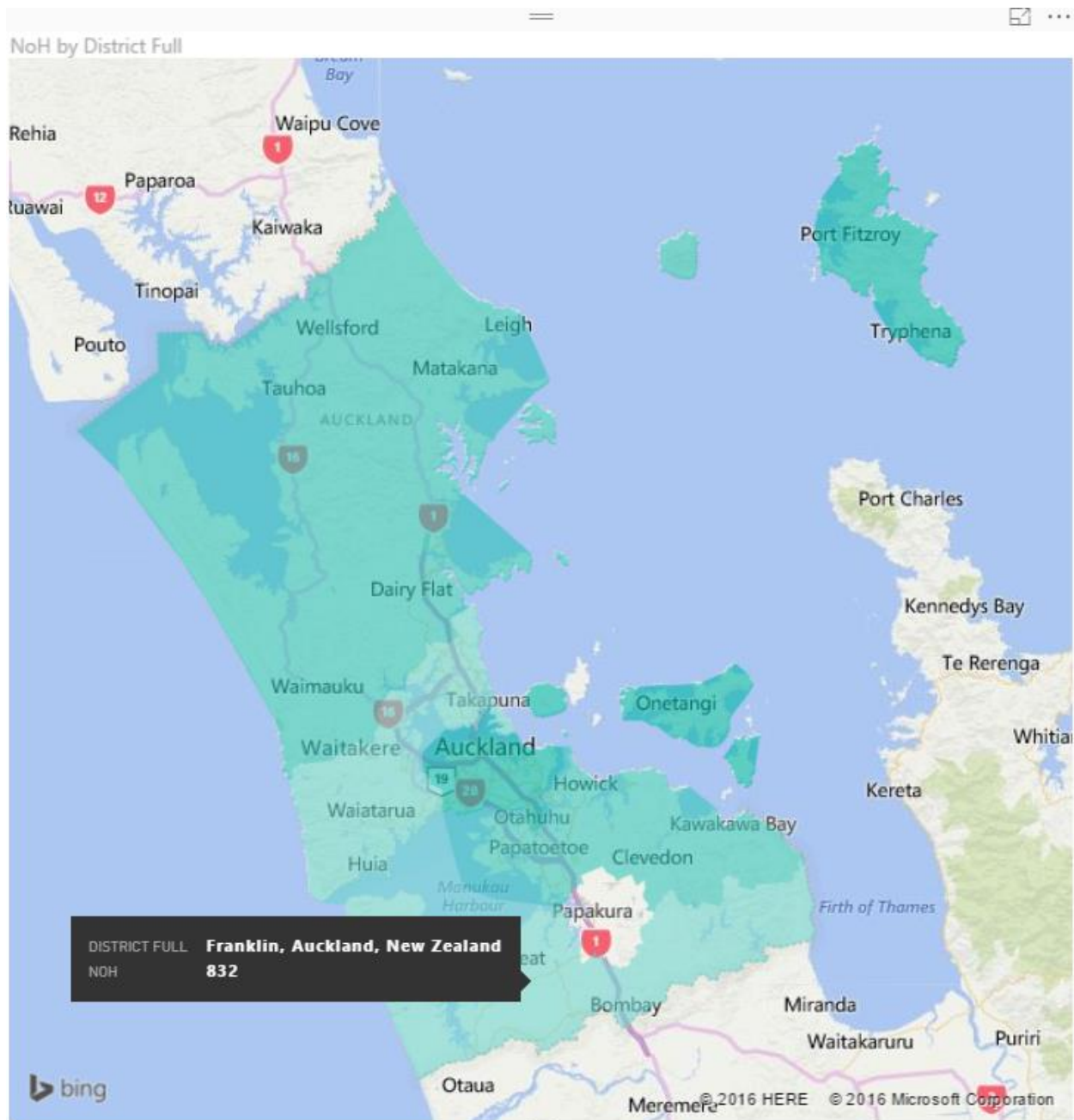
The Ugly: When Geographic Structure is Different

I'm living in New Zealand (if you don't know know it already ;)) and the geographic structure here is different from State, City, County. What we have is more like Country, Region, District, Suburb. for example: Country= New Zealand, Region= Auckland, District =Franklin, Suburb= Pukekohe. You've seen in example above that I used State or Province data category for Region and it worked fine. So I thought I can map District to

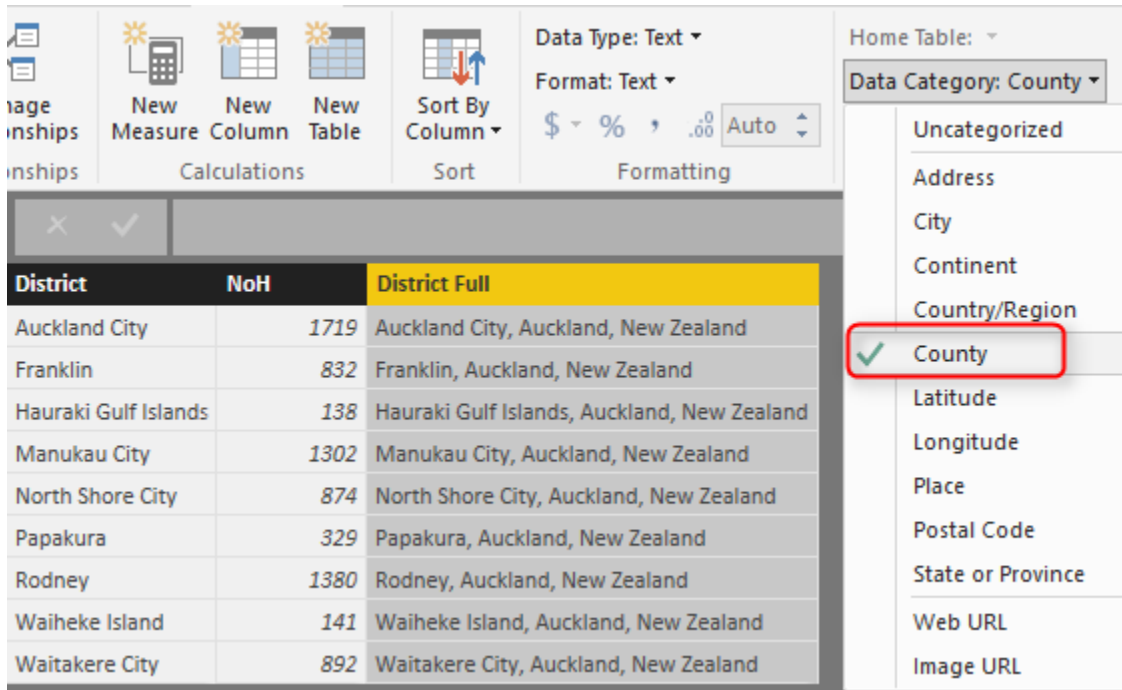
the City data category as well, but Here is what I get when I map cities in Auckland to a City data category:



However when I change data category to County I get it working (not perfect though! there are some white spots in the map!);

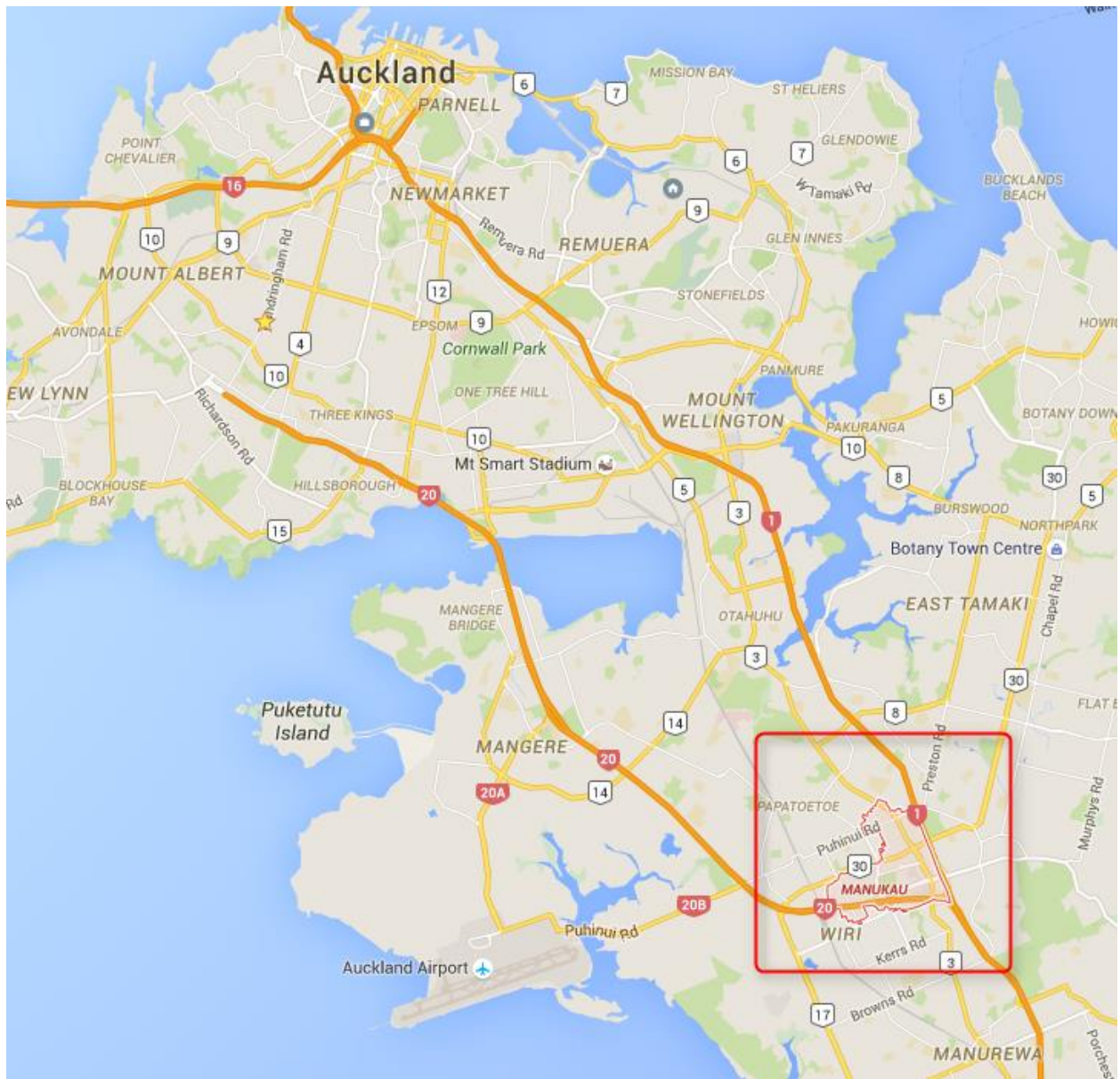


and Here is the category I choose for it:

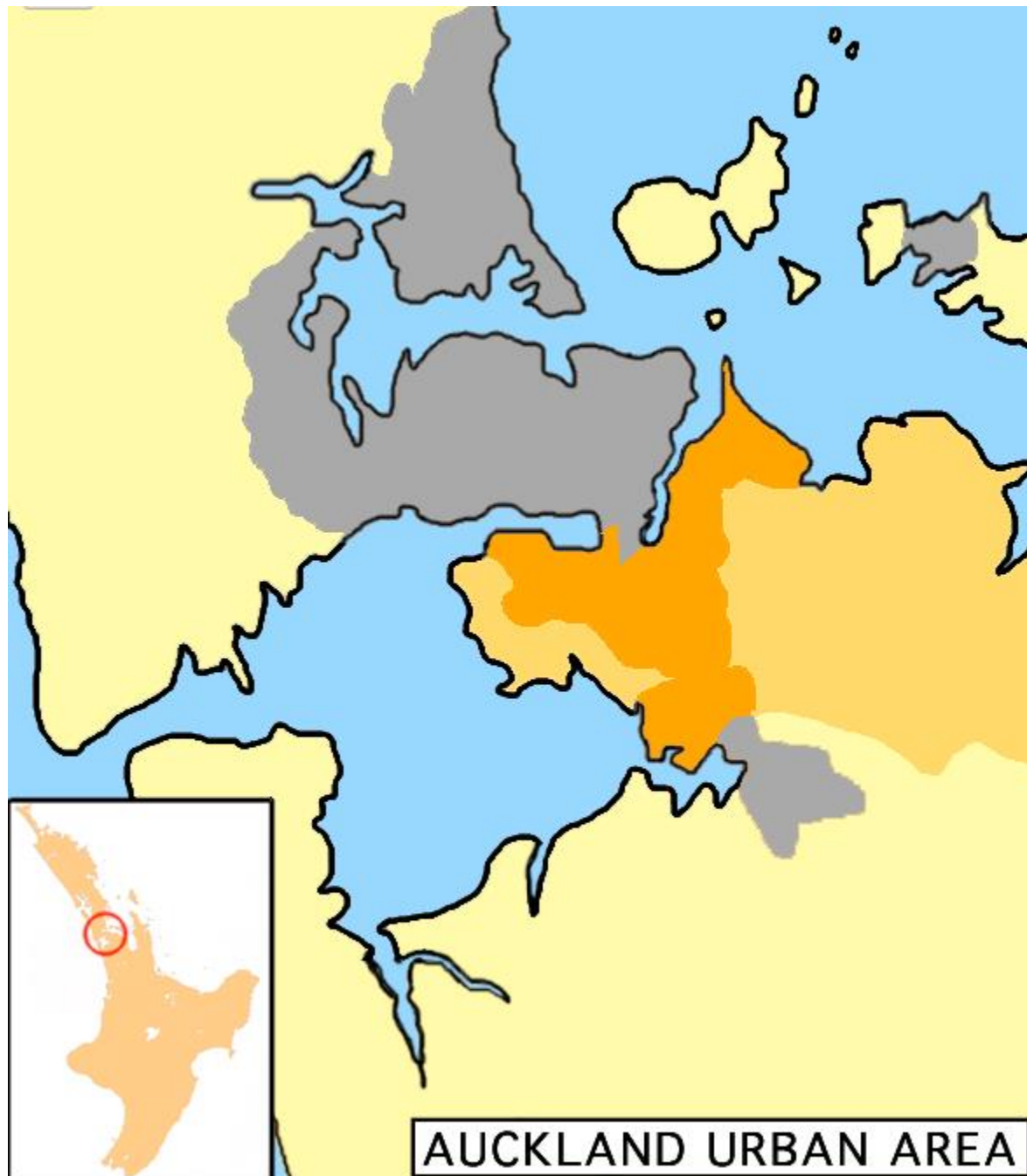


District	NoH	District Full
Auckland City	1719	Auckland City, Auckland, New Zealand
Franklin	832	Franklin, Auckland, New Zealand
Hauraki Gulf Islands	138	Hauraki Gulf Islands, Auckland, New Zealand
Manukau City	1302	Manukau City, Auckland, New Zealand
North Shore City	874	North Shore City, Auckland, New Zealand
Papakura	329	Papakura, Auckland, New Zealand
Rodney	1380	Rodney, Auckland, New Zealand
Waiheke Island	141	Waiheke Island, Auckland, New Zealand
Waitakere City	892	Waitakere City, Auckland, New Zealand

So as you see I have to play with to map my data fields to a data category which generates a result close to what it is. And as you see I don't get a clean result. I see an area (Papakura) in the map above which is white (no data points) which should be drawn as part of one of Districts (Manukau City). I did a search on Google Map and Bing Map to see if they can show me boundaries of districts, and found out that they can't. Here is what [Google Map](#) shows me:



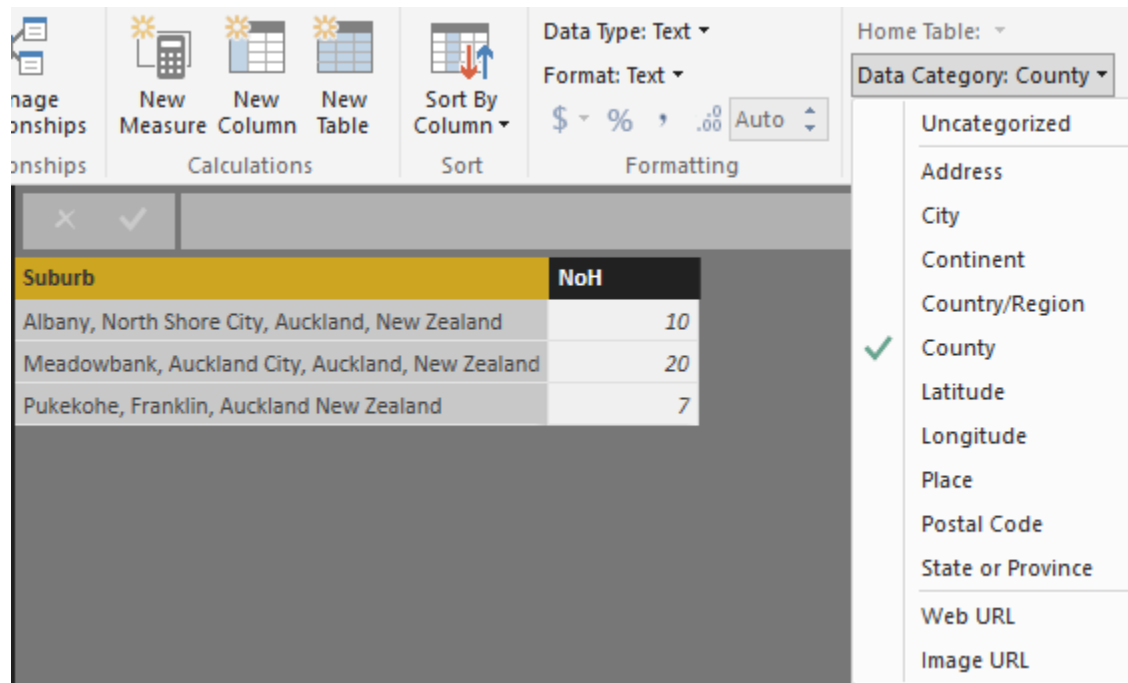
And here is what the actual boundaries should be (from [Wikipedia](https://en.wikipedia.org/wiki/Manukau))



The orange area in map above is much bigger than the area mentioned in Google Map. So no wonder why Power BI shows some areas blank. So I can't blame Power BI, because it seems to be related to the boundaries definition in Bing maps which is not up-to-date. However at the end of the day this won't give me the result I'm looking for.

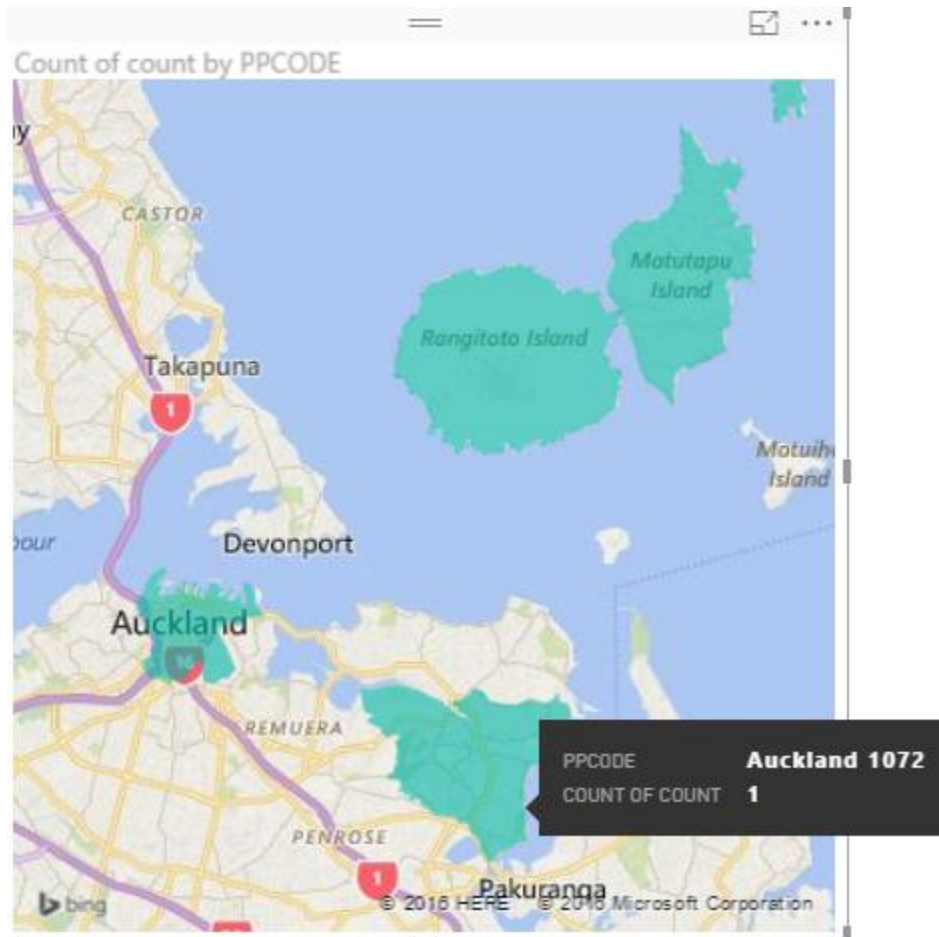
The Bad: When Data Category is Useless

Believe it or not, there are some situations that you can't find any appropriate data categories to map to your geographic data field. This might not happen in USA geographic information, but I found it when I tried to go one level down to the Suburb level. The sad fact is that I couldn't find a data category that maps to suburbs in some areas in New Zealand! Here is what I've done;

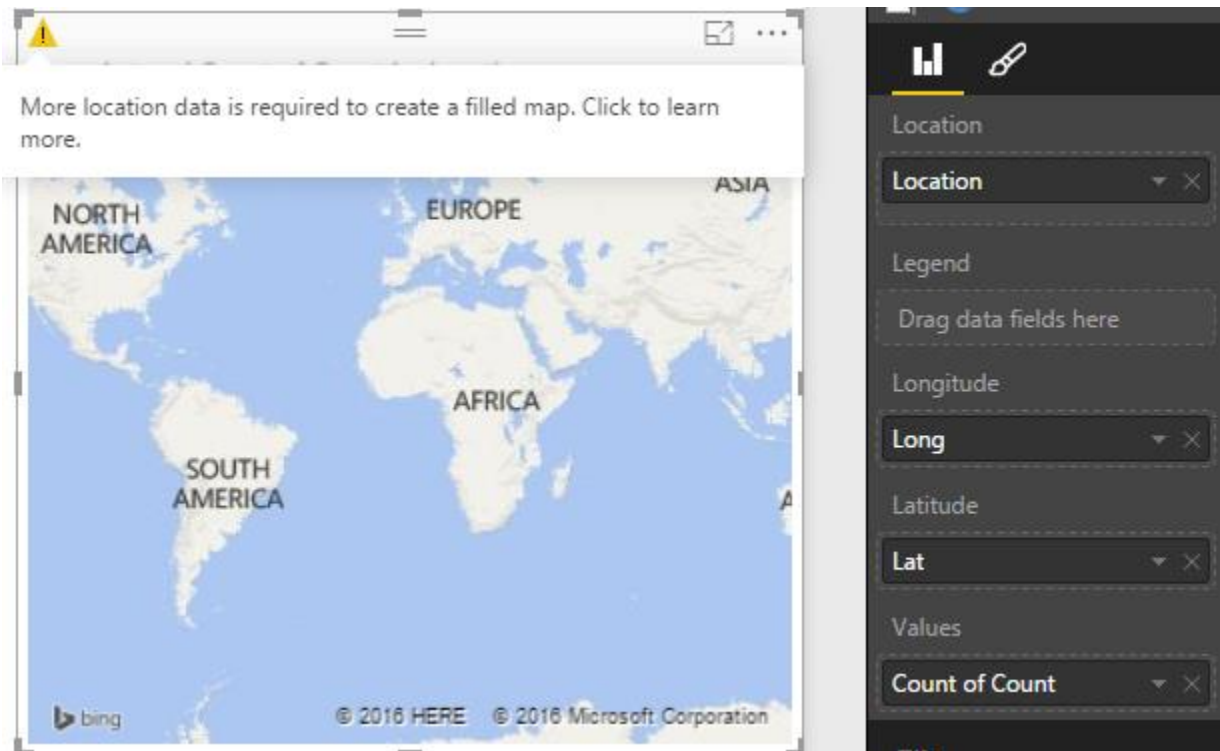


I've mapped suburbs to County because that was the lowest level I've found in data category for geographic information. (Place and Address cannot be used for Filled Map at the time or writing this post). and I got Nothing! Not even a small area on the map. I've tried then removing the district and putting suburb, region, country format with County as the data category which didn't help again.

I've found that I can map some locations based on Postal Code as you see below. However not Postal Code is not always good distinguishing field for a region, as multiple regions might have a postal code shared.



Then I tried to find Latitude and Longitude of each suburb in Bing Maps and use that for Filled Map, but I've found that Lat and Long fields in Filled Map isn't working in the way they supposed to work!



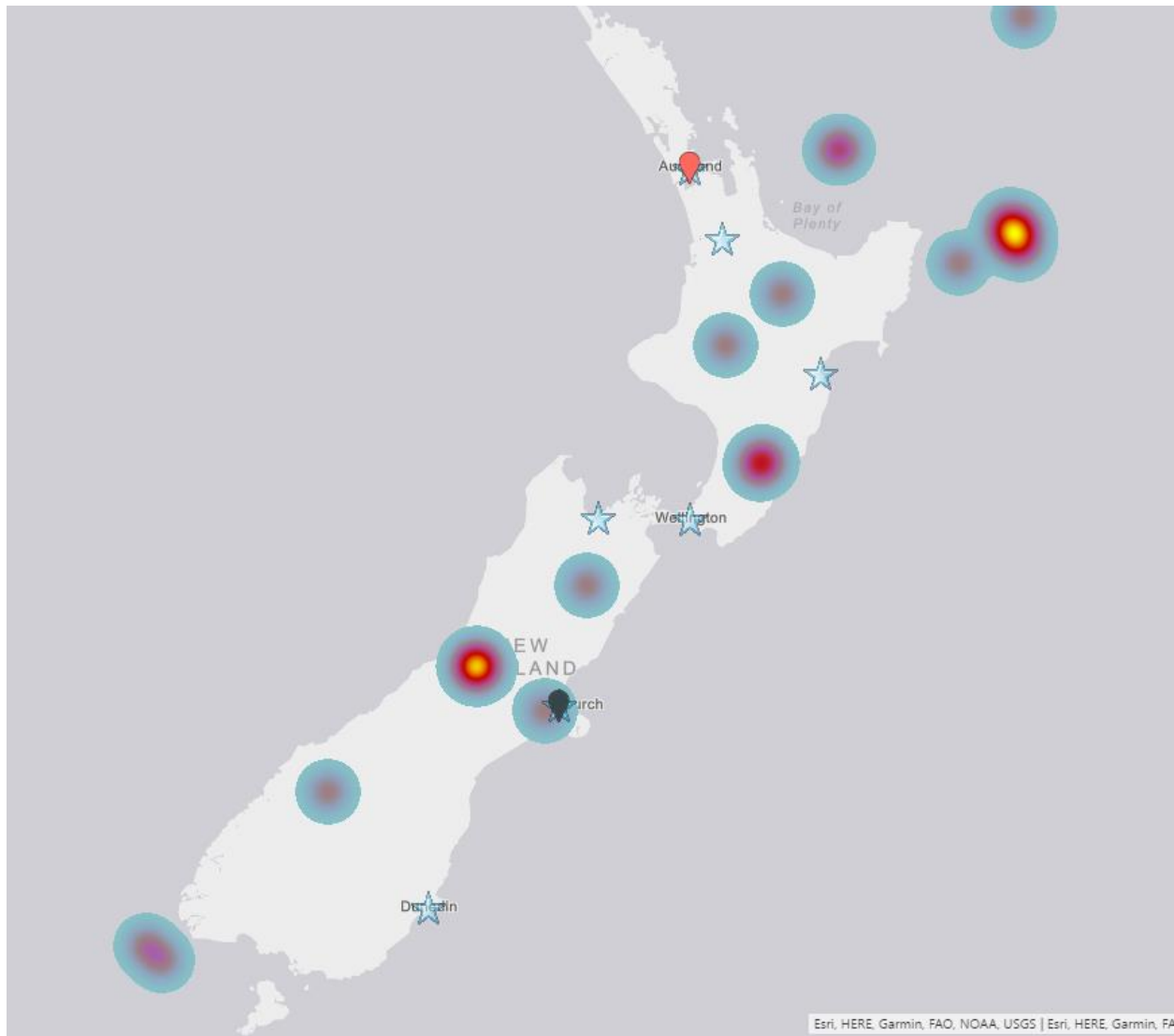
There was no other method I could try to map suburbs with Filled Map unfortunately. So the sad end of this experiment is that with the current version of Filled Map I cannot show some suburbs with Filled Map.

Summary

Filled Map is good visualization element but it has some limitations. You can use it for areas which their boundaries are clear, such as Country, Region, States. However when geographic structure is different from built-in structure in Data Category types, then you might face some difficulties and won't be able to generate the visualization you want. I hope Microsoft BI team fix issues related to this graph soon (And I believe they will, because they did a really good job in Power BI extensive improvement through past months). If you want to learn more about Power BI feel free to read rest of [Power BI online book from Rookie to Rock Star](#).

ArcGIS Map in Power BI; Simple, But Insightful

Published Date: April 13, 2017



I have written about 4 other types of map visual with Power BI before. In this post I want to explain how ArcGIS visual work as a map visual. There are some specific features in this visual which is simple to use, but will give you a lot of insight. With this type of map visual, you can add multiple layers of data, and use even some of existing geo location information from ArcGIS map directory. Let's go through an example and see what are

features with this visual. To learn more about Power BI, read [Power BI book from Rookie to Rock Star](#).

Maps in Power BI

There are many ways to visualize something on Map in Power BI, some of it are as below;

- [Map Visual](#)
- [Filled Map Visual](#)
- [Custom Visuals – Synoptic Panel by OKviz](#)
- [Custom Visuals – GlobeMap by Microsoft](#)
- [Shape Map Visual – Preview](#)
- ArcGIS Map Visual – Preview

I have written about them all except last one. This post focus is on ArcGIS Map visual. Hopefully at the end I will write a post about comparison.

Sample Data Set

For this example I will be using a sample data set which includes earthquakes in New Zealand in 2015! Not all of them, but part of earthquakes. Guess how many earthquakes listed in this file? 20,000! But don't worry, it has listing of everything even earthquakes with magnitude of less than 1. To download the CSV data set file, use link below;

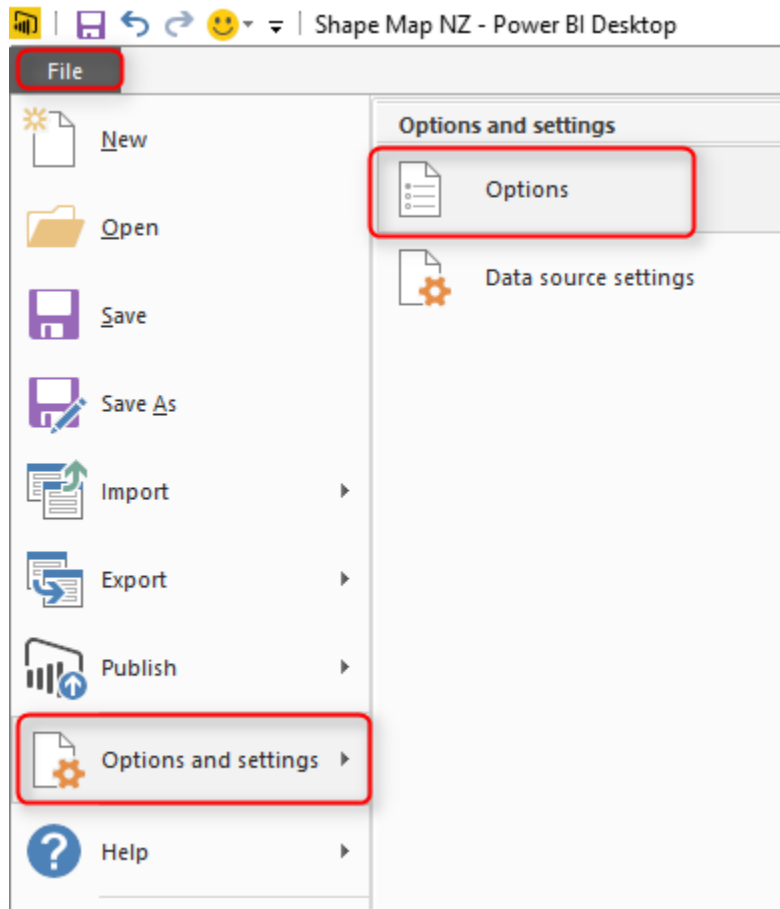
Enter Your Email to download the file (required)

[Download]

ArcGIS Map Visual

This visual is still preview at the time of writing this post, and for using it, you need to enable this preview feature. To enable it go through:

File Menu -> Options and Settings -> Options.



In preview features tab, select ArcGIS Maps for Power BI.

Options

GLOBAL

Data Load
Query Editor
DirectQuery
R scripting
Security
Privacy
Updates
Usage Data
Diagnostics
Preview features
Auto recovery

CURRENT FILE

Data Load
Regional Settings
Privacy
Auto recovery

Preview features

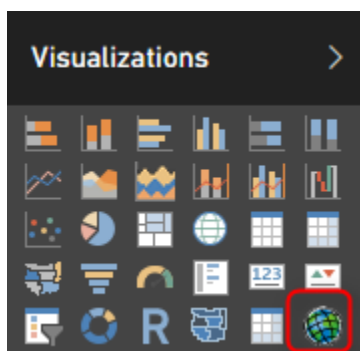
The following features are available for you to try in this release. Preview features might change or be removed in future releases.

- ☐ Impala [Learn more](#)
- ☐ Snowflake [Learn more](#)
- ☒ Shape map visual [Learn more](#)
- ☒ Custom report themes [Learn more](#)
- ☒ Enable cross filtering in both directions for DirectQuery [Learn more](#)
- ☒ **ArcGIS Maps for Power BI** [Learn more](#)
- ☒ New matrix visual [Learn more](#)
- ☒ Numeric range slicer [Learn more](#)
- ☐ Spanish language support for Power BI Q&A [Learn more](#)
- ☒ Power BI Service Live Connection [Learn more](#)
- ☒ Quick measures [Learn more](#)

OK

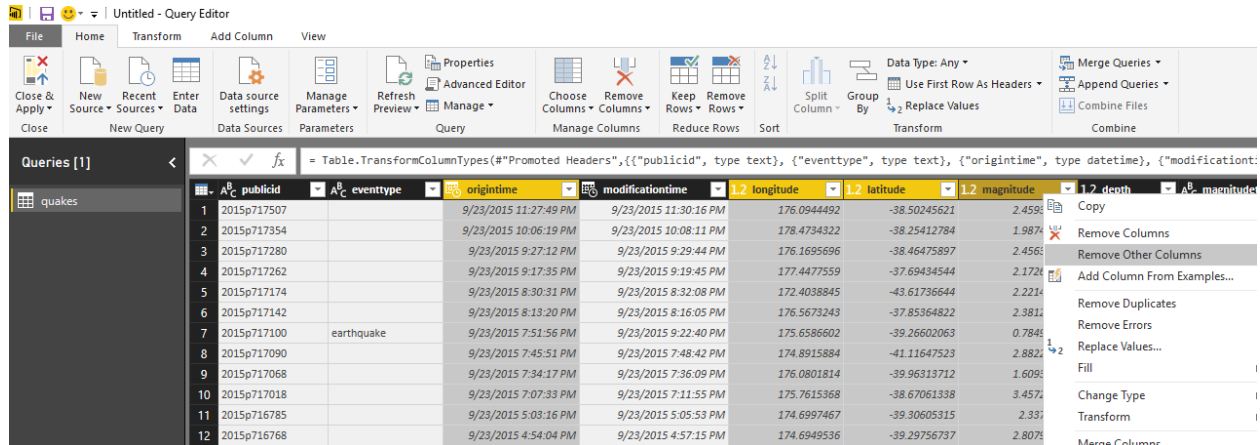
Cancel

After clicking OK, you need to close your Power BI Desktop and re-open it again. You should be able to see ArcGIS Map visual in list of your visuals now.



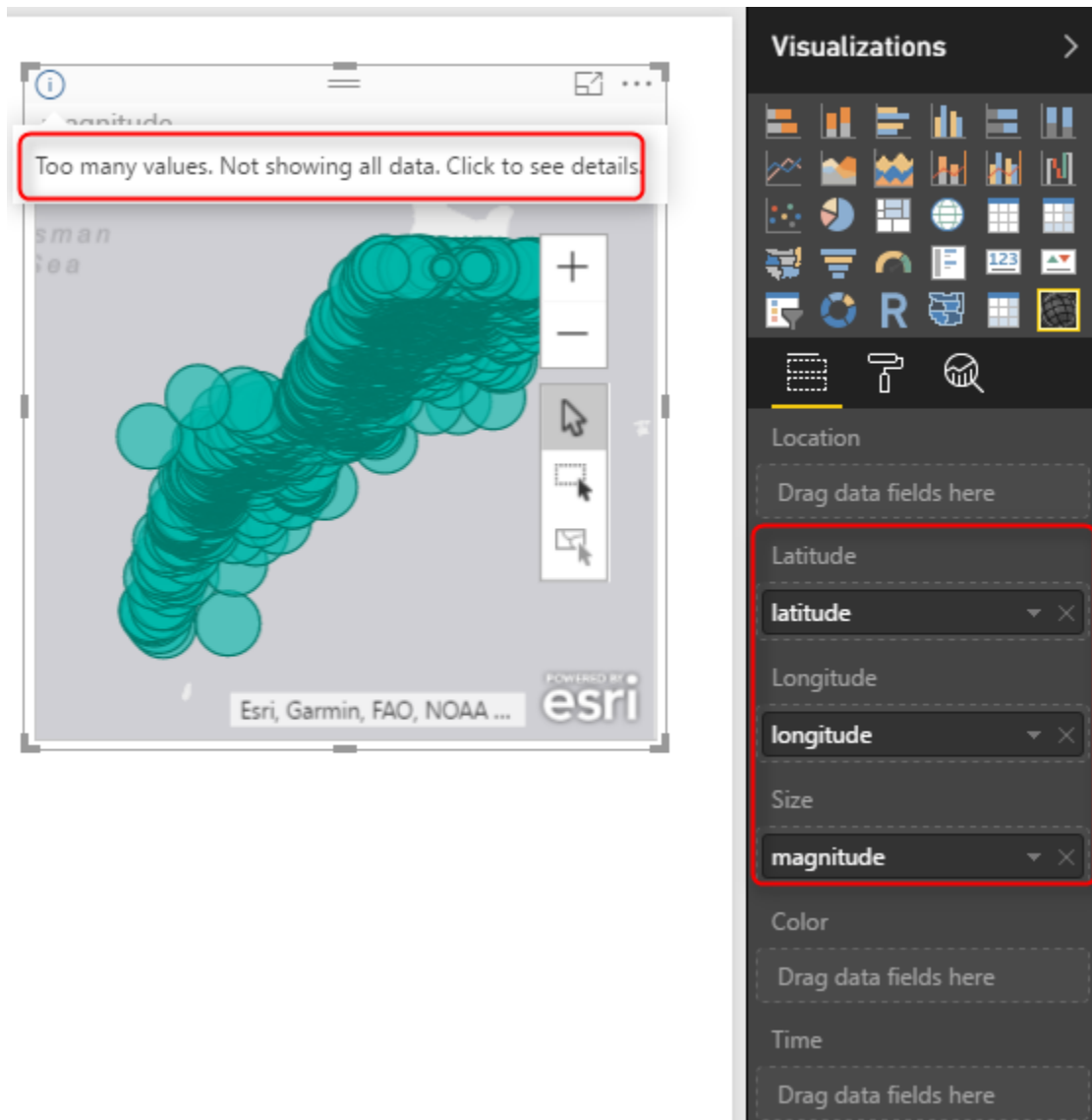
Get Data

To start with the sample data set, Get Data from Text/CSV, and select the quakes.csv file (link above in this post), Click on Edit in Navigator window, and you will see the whole data set. Only keep these columns and remove the rest: Longitude, Latitude, Magnitude, and Origintime. That means; just select these columns, an from right click menu click on Remove Other Columns.

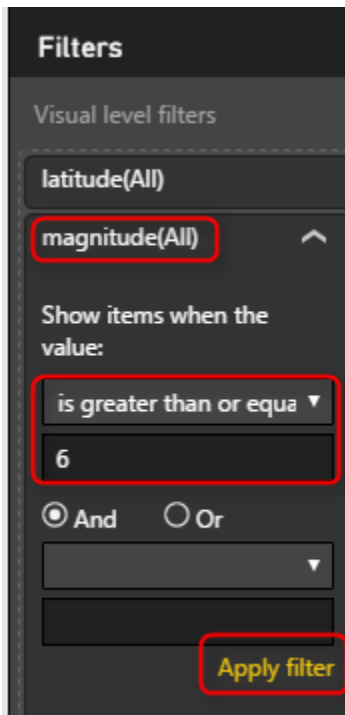


The screenshot shows the Power BI Query Editor interface. The top ribbon includes tabs for File, Home, Transform, Add Column, and View. The 'Transform' tab is active, showing various data transformation options. The main area displays a table of earthquake data with columns: publicid, eventtype, origintime, modificationtime, longitude, latitude, magnitude, depth, and magnitude. A right-click context menu is open over the 'magnitude' column, showing options like 'Remove Columns', 'Remove Other Columns', 'Add Column From Examples...', 'Remove Duplicates', 'Remove Errors', 'Replace Values...', 'Fill', 'Change Type', 'Transform', and 'Merge Columns'.

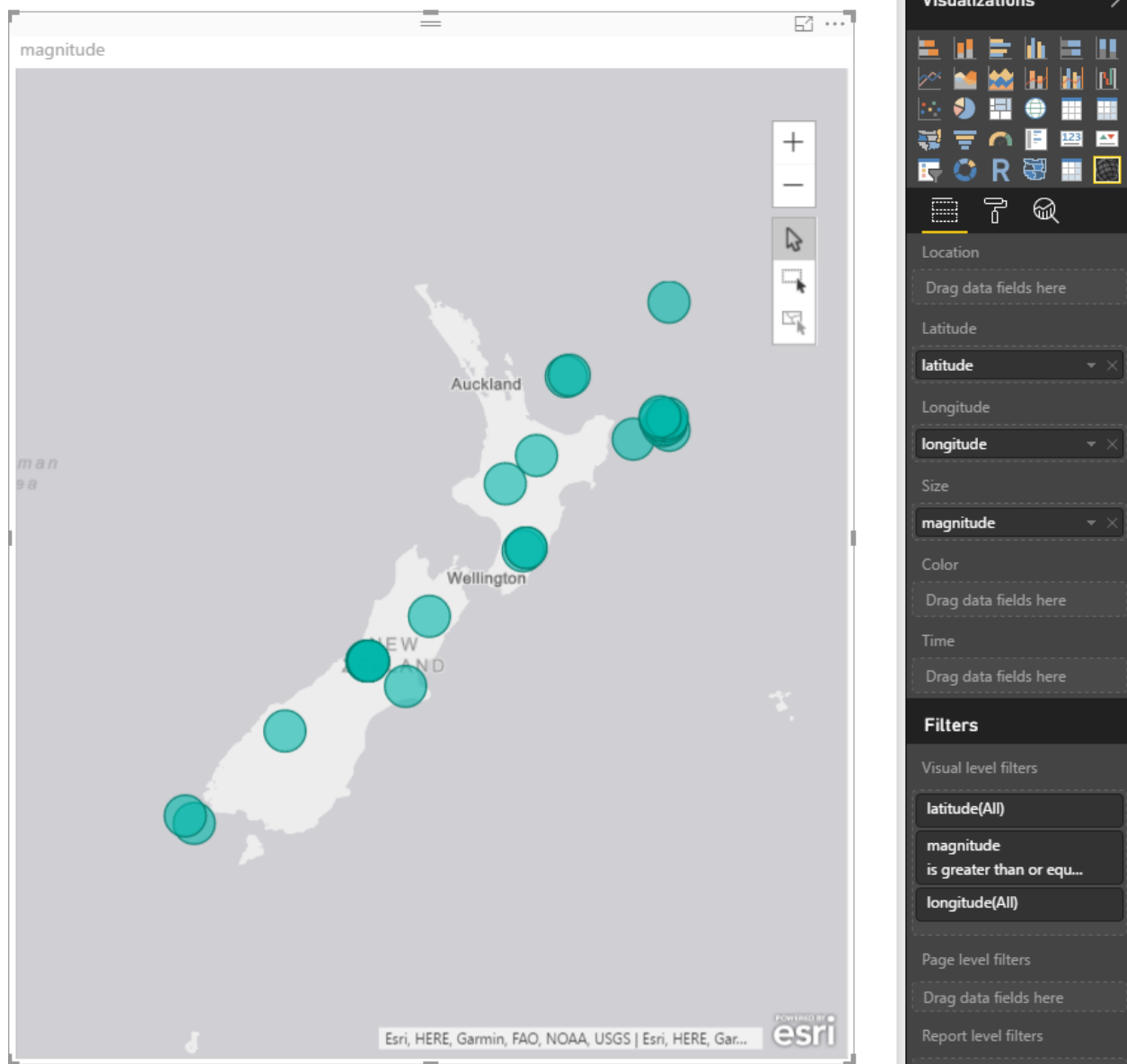
Close and Apply the Query Editor window. Drag and Drop a new ArcGIS map visual in the report, and assign values from latitude to latitude, longitude to longitude, and magnitude from fields list to the Size section of the visual.



This visual, like the basic map visual in Power BI, has a limitation on the number of data points to show. because we have 20K data points it cannot visualize them all. Let's filter the data set, and only focus on everything with magnitude greater or equal to 6. For doing this in visual level filter expand the Magnitude filter and apply filtering. Remember to click on Apply Filter, otherwise nothing happens.



After applying filter, you would simply see only few numbers.

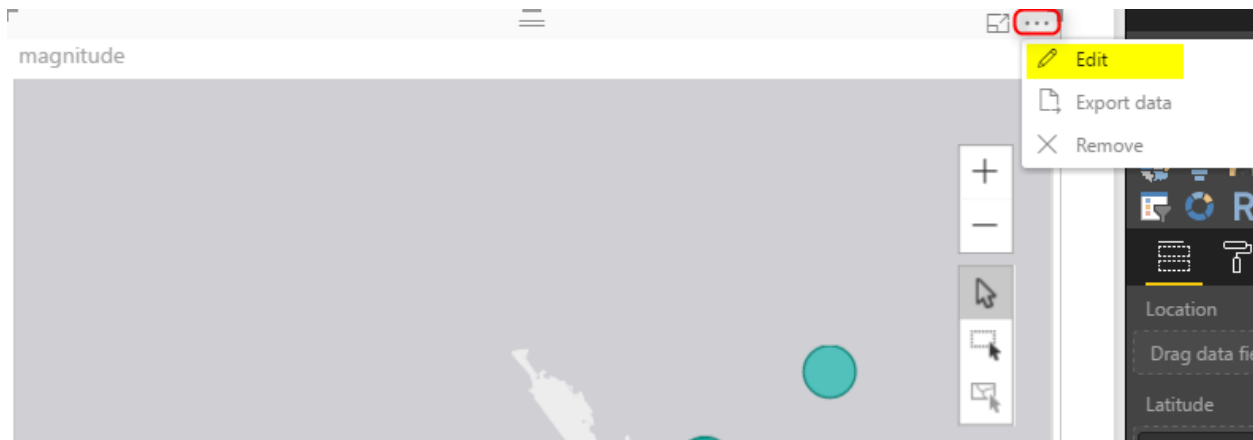


All of these earthquakes with magnitude more than 6 happened in 2015! can you imagine?!!!! anyway, let's go back to ArcGIS map;

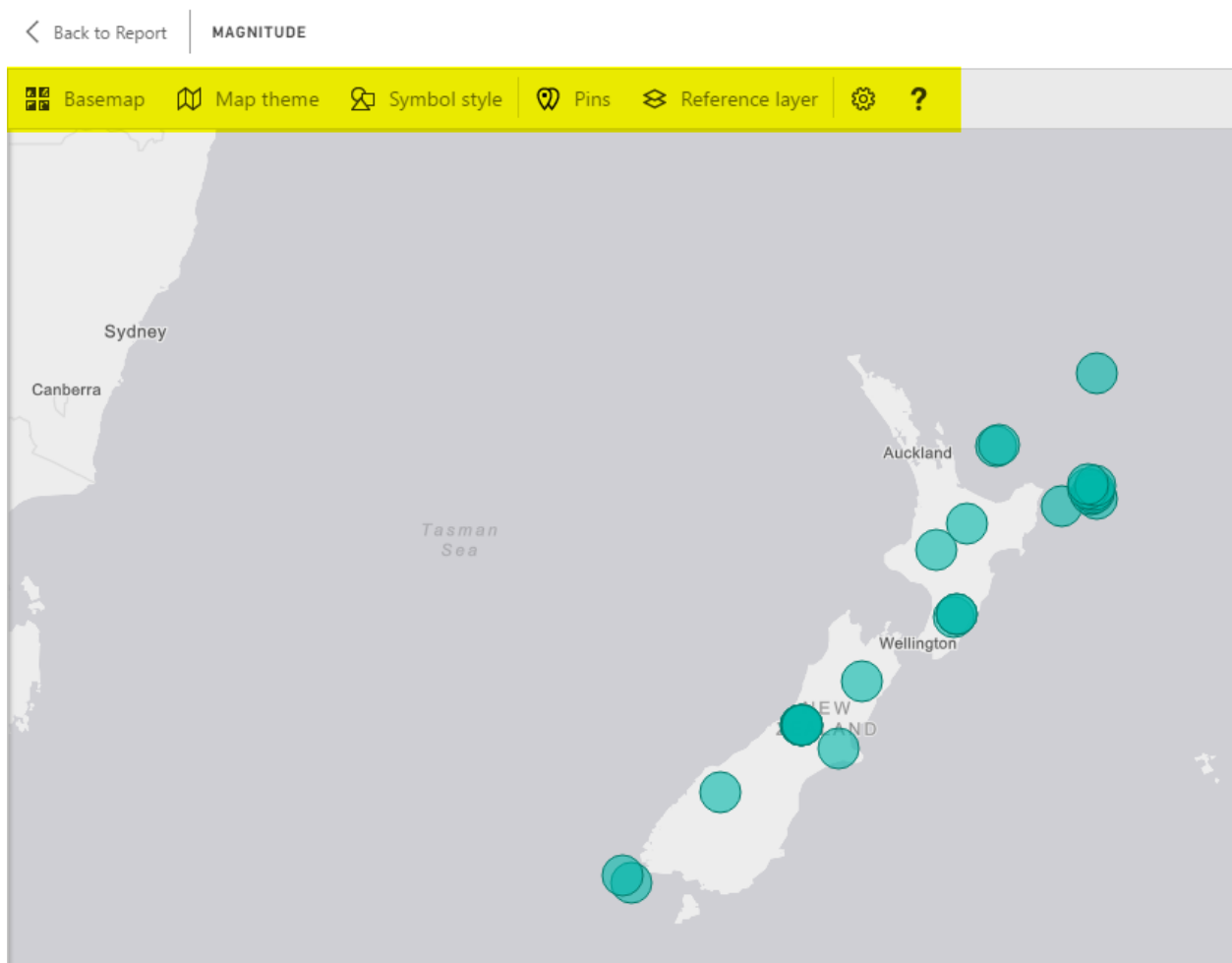
So far, the map is very similar to normal Power BI basic map visual with bubbles. Now let's see what other options we have with ArcGIS visual;

ArcGIS Features

To go through features of this visual, you need to click on three dots icon on the top right corner of visual and click on Edit.

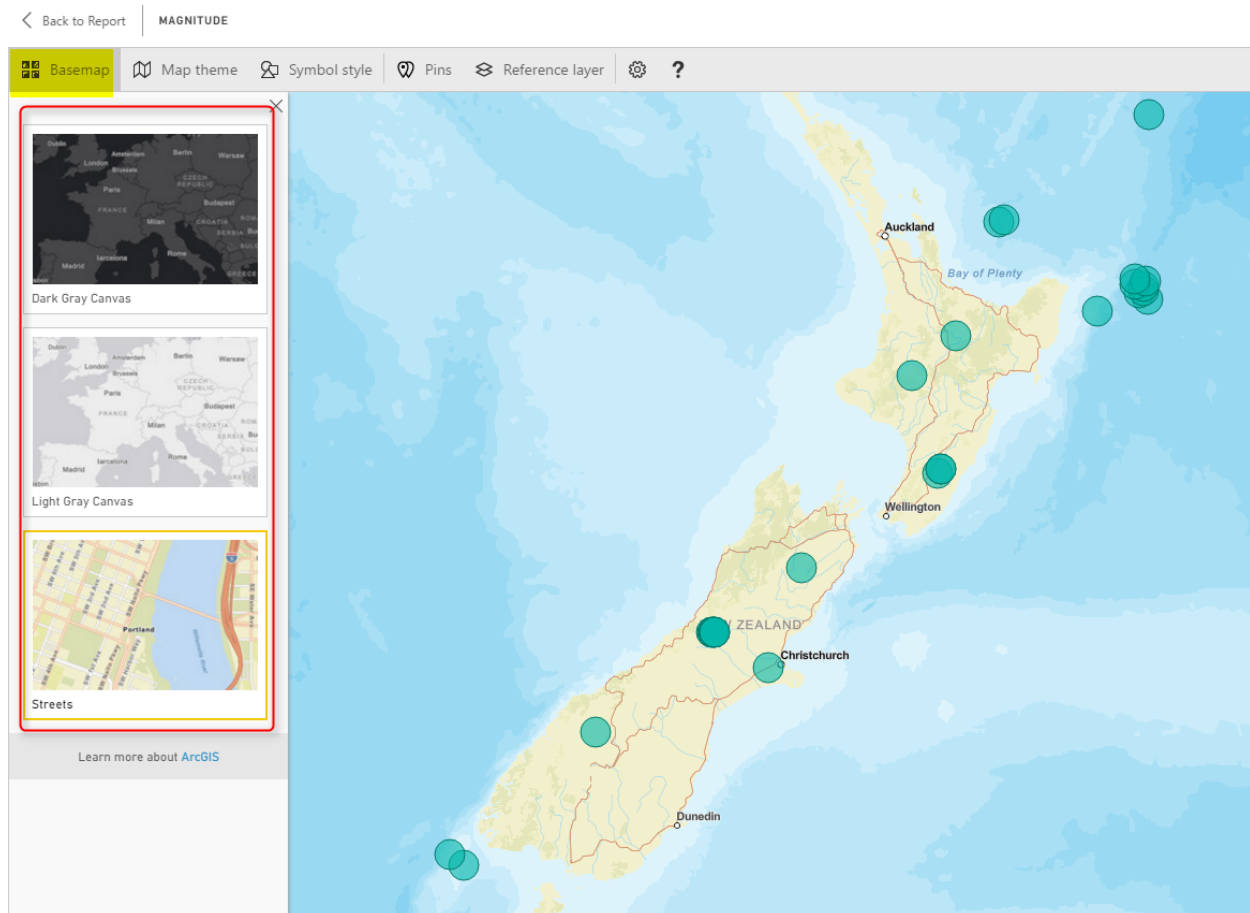


You will be redirected to an Edit mode, where you will see the map full screen, and number of menu options at the top;



Change Base Map

The first change you can make, it so change the base map, to be something different. You can choose from Dark Gray Canvas, Light Gray Canvas (Default), or Streets.



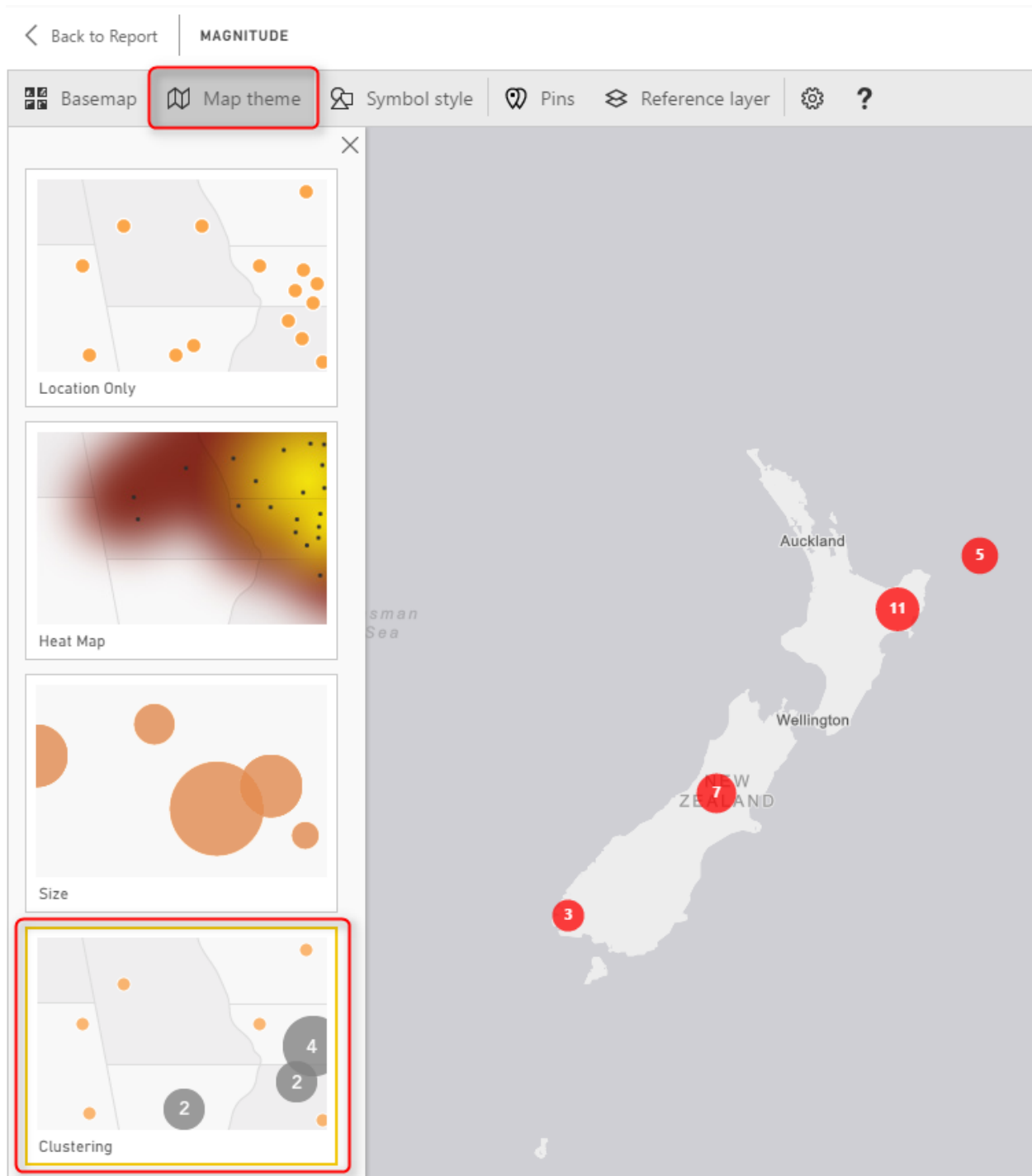
This feature is good to match the map with theme of your report, or even get some street information.

Map Theme

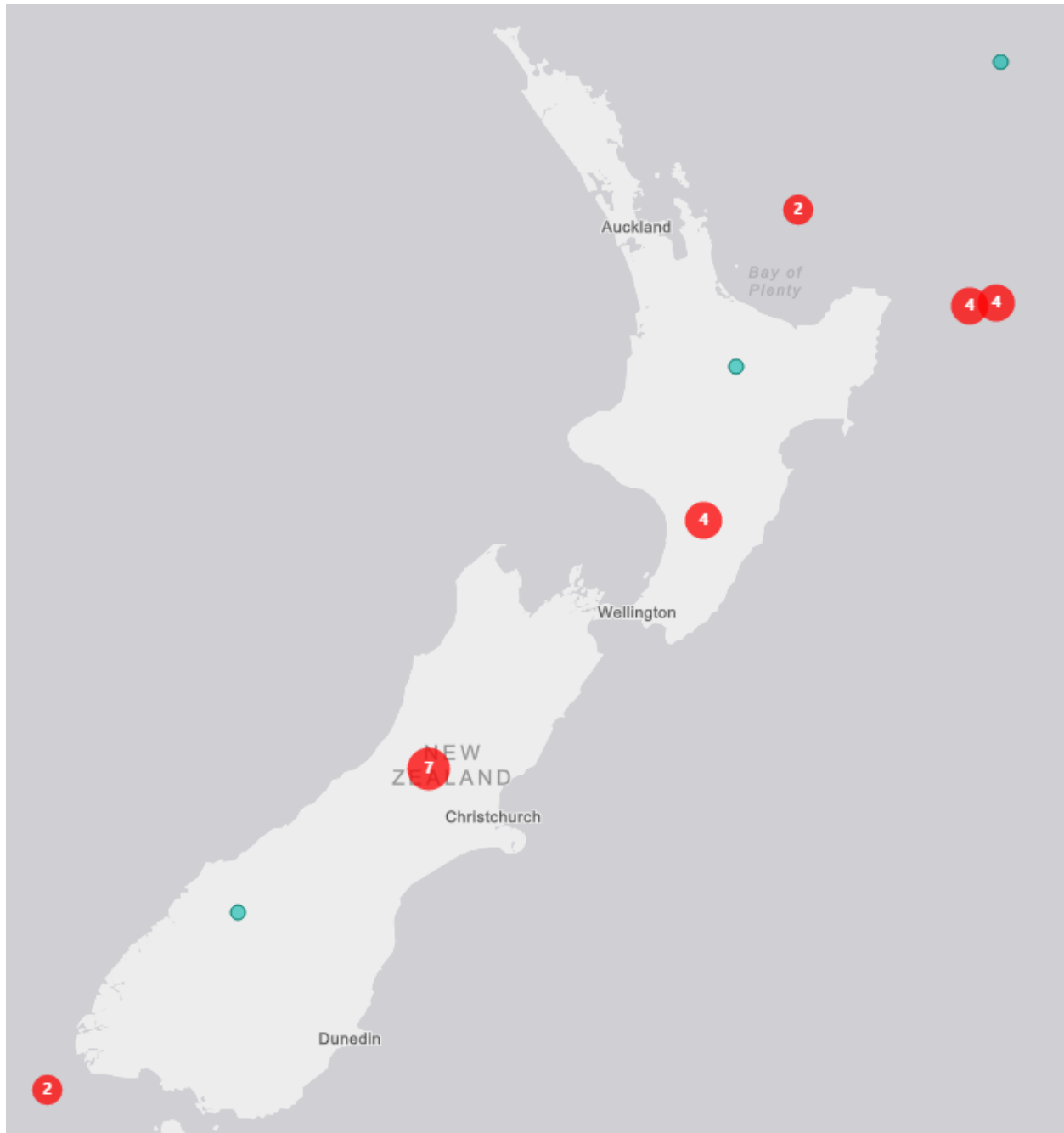
One of the interesting features of this visual, is to change theme of Map. This type of map can give you four themes;

16. Location Only – no size difference between bubbles
17. Heat Map
18. Size – bubbles with different sizes based on measure's value
19. Clustering – Good option for drill down

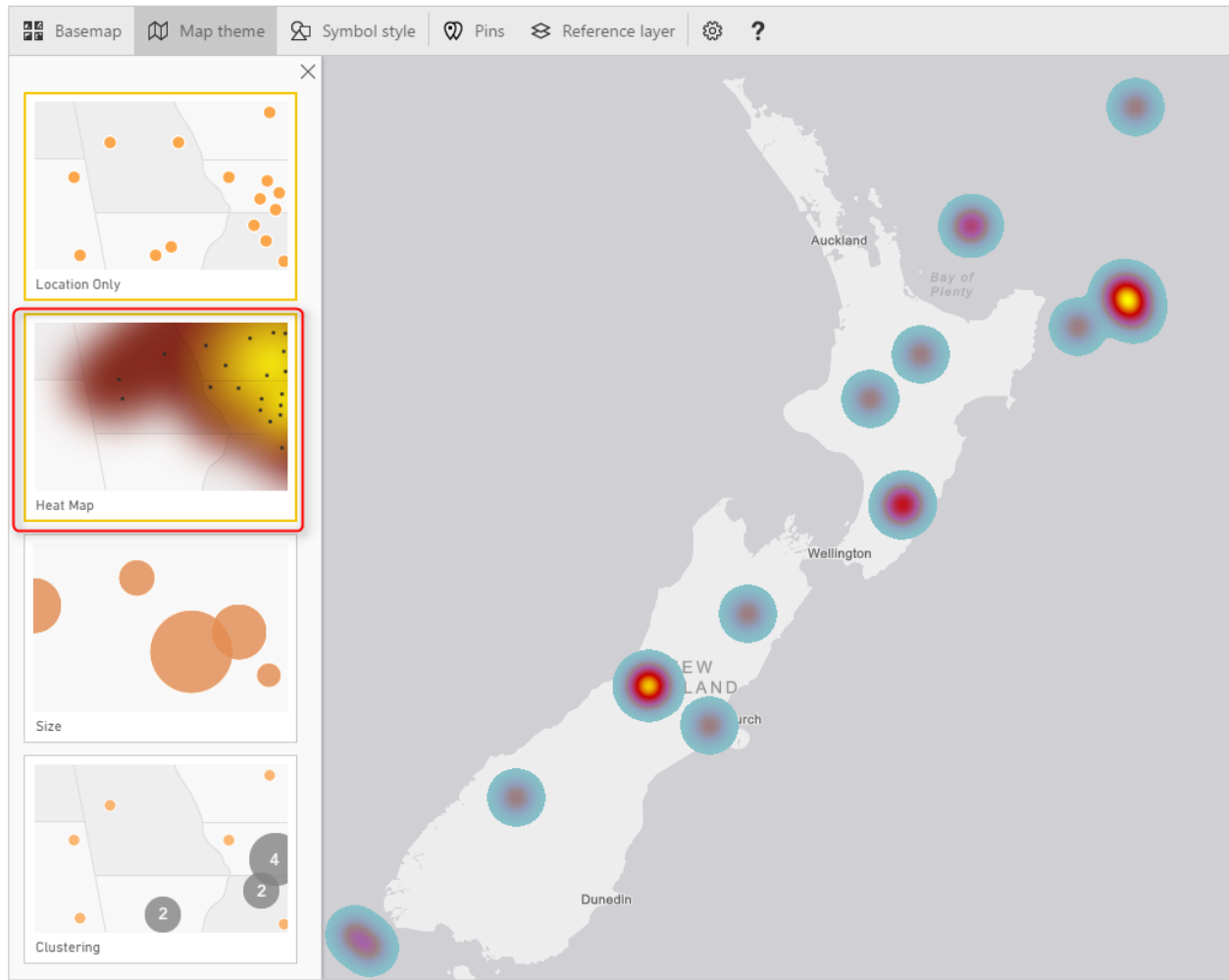
Clustering is good if you want to have high level view, and then ability to drill down into each region. Here is a view of clustering;



Each bubble shows you the total count of data point in that region (more you zoom out, bigger your region will be). and if you double click on a bubble you will drill down into that region;



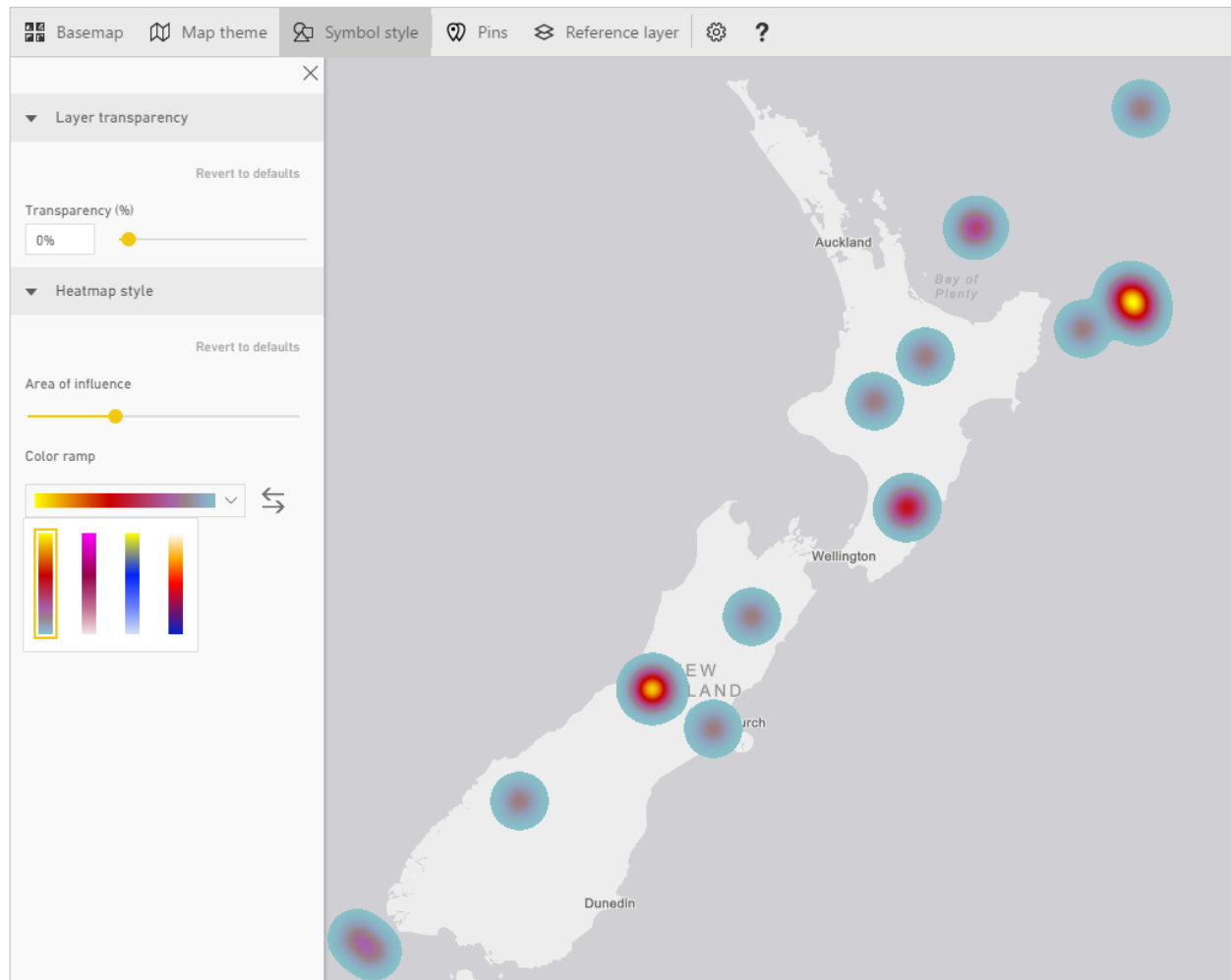
Heat map is a good visualization for showing the severity of earthquake in each area more graphically;



You can also use Location only or Size, which are simple to use.

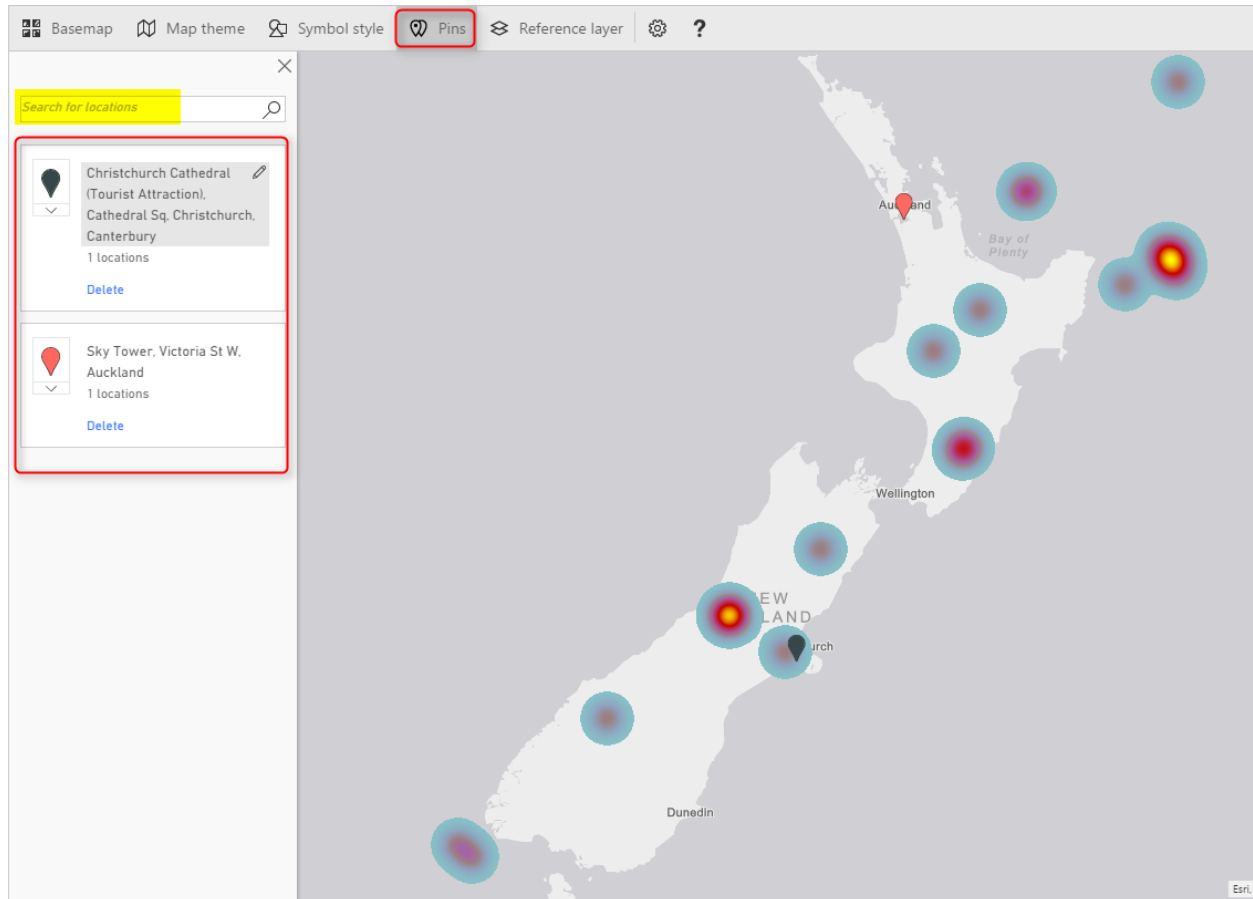
Symbol Style

Depends on what Map Theme, you've selected, you can do the details configuration of that in Symbol Style section. For example, if you have selected heat map, you can change the color. If you have selected size, you can change the bubble sizes and etc.



Pins

You can add your own pins. These pins can be locations of branches, hospitals, schools, or whatever else you want to use as a pin on the map.




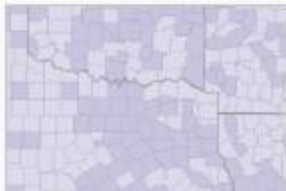

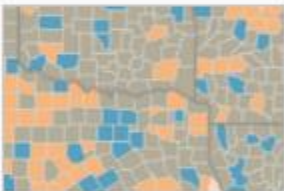

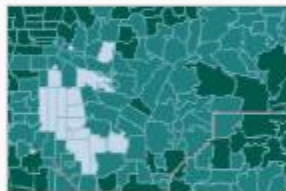


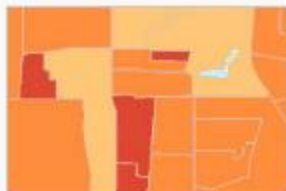
Reference Layer

The last feature, but not least is adding reference layer. In my opinion this is one of the best features of this visual. You can add another layer of visualization based on a map in directory of ArcGIS maps. [ArcGIS](#) has many maps that you can use, almost map of everything! if you click on Reference Layer menu option, you can see some of existing maps for USA.

[< Back to Report](#) | **MAGNITUDE**

Basemap Map theme Symbol style Pins **Reference layer**

Demographics ArcGIS Remove layer

 2016 USA Average Household Income <input type="checkbox"/> Add	 2016 USA Average Household Size <input type="checkbox"/> Add	 2016 USA Diversity Index <input type="checkbox"/> Add
 2016 USA Median Age <input type="checkbox"/> Add	 2016 USA Median Disposable Income <input type="checkbox"/> Add	 2016 USA Median Home Value <input type="checkbox"/> Add
 2016 USA Median Household Income <input type="checkbox"/> Add	 2016 USA Per Capita Income <input type="checkbox"/> Add	 2016 USA Population Density <input type="checkbox"/> Add

To access more maps, click on ArcGIS at the top, and there you can search for maps. For example below are some maps in New Zealand area;


Back to Report
MAGNITUDE

Basemap
Map theme
Symbol style
Pins
Reference layer


Demographics

ArcGIS


Remove layer




New Zealand Demographics for Emergency Management
☐ Add




New_Zealand_Predator_Control_NWR_May2014
☐ Add




New Zealand Schools
☐ Add




TrafficCameraLocations
☐ Add




New Zealand Authoritative Boundaries
☐ Add




New_Zealand_50k
☐ Add



New Zealand Roads
☐ Add



New Zealand Cellphone Towers
☐ Add

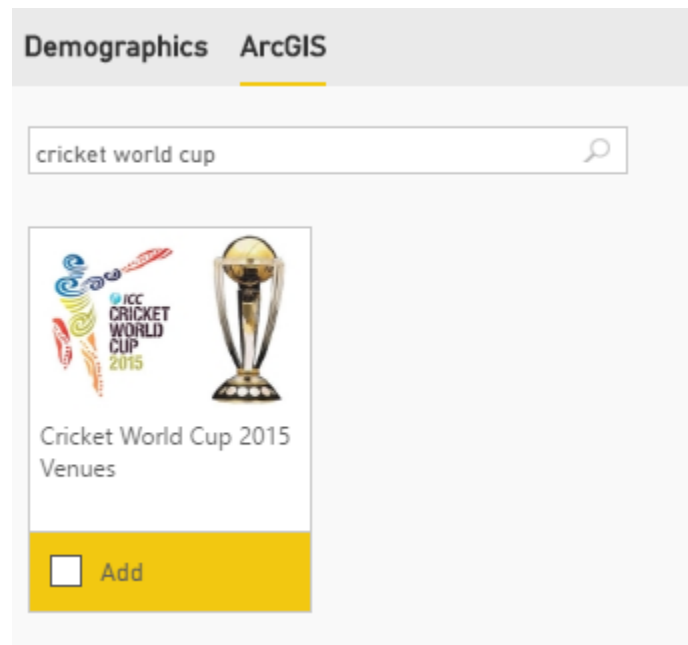


Pest Free Islands and Sanctuaries
☐ Add

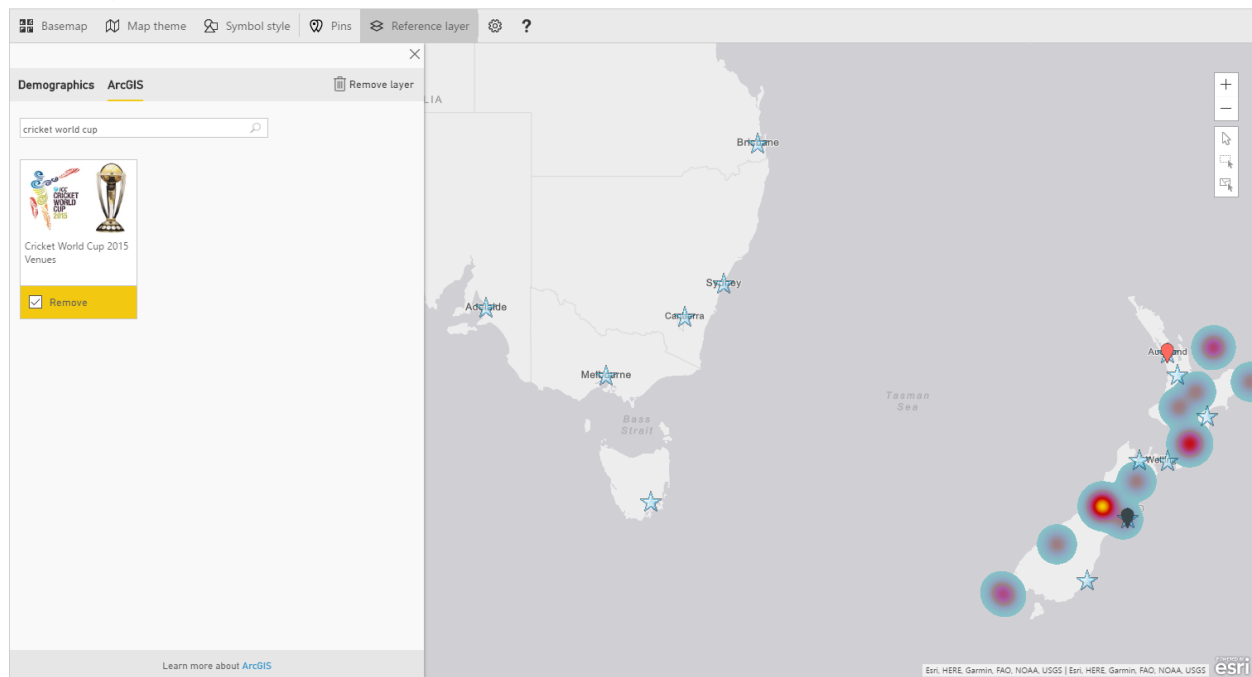
[Learn more about ArcGIS](#)

There are maps for schools, pipelines, gas stations, almost everything. Even there is a map for **Traffic Camera Locations**, Very useful map!

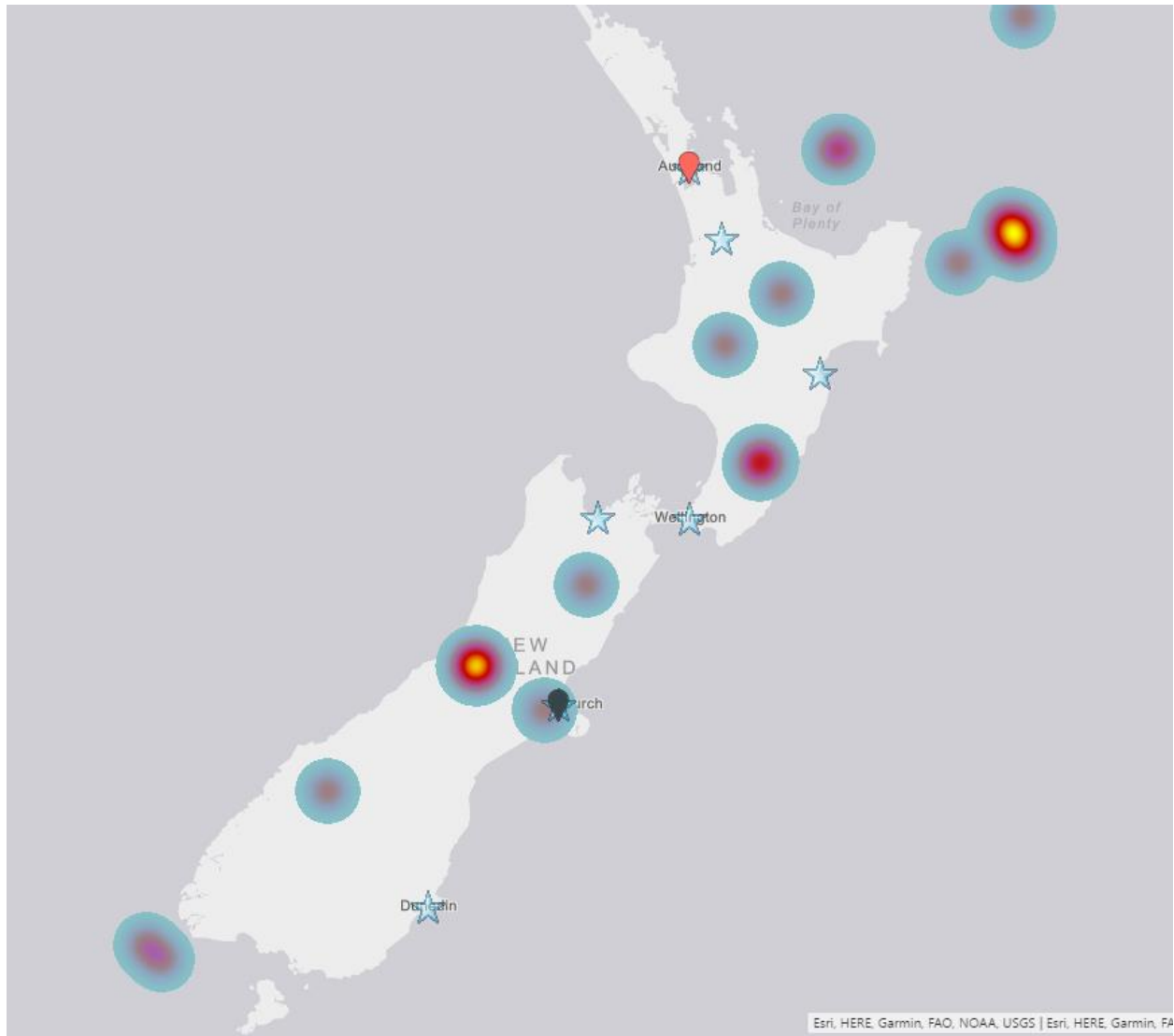
Because the current data set is for New Zealand 2015, and there was a cricket world cup in New Zealand and Australia at that time, so it might be good to bring a map of venues for that and realize are those cricket world cup venues close to the earthquake centers or not. Simply you can search for cricket world cup, and you will get the map very easily. Click on Add to add the layer to map;



It will take few seconds for the layer to load, and now you will see locations with star mark on it. Wonderful, isn't it?

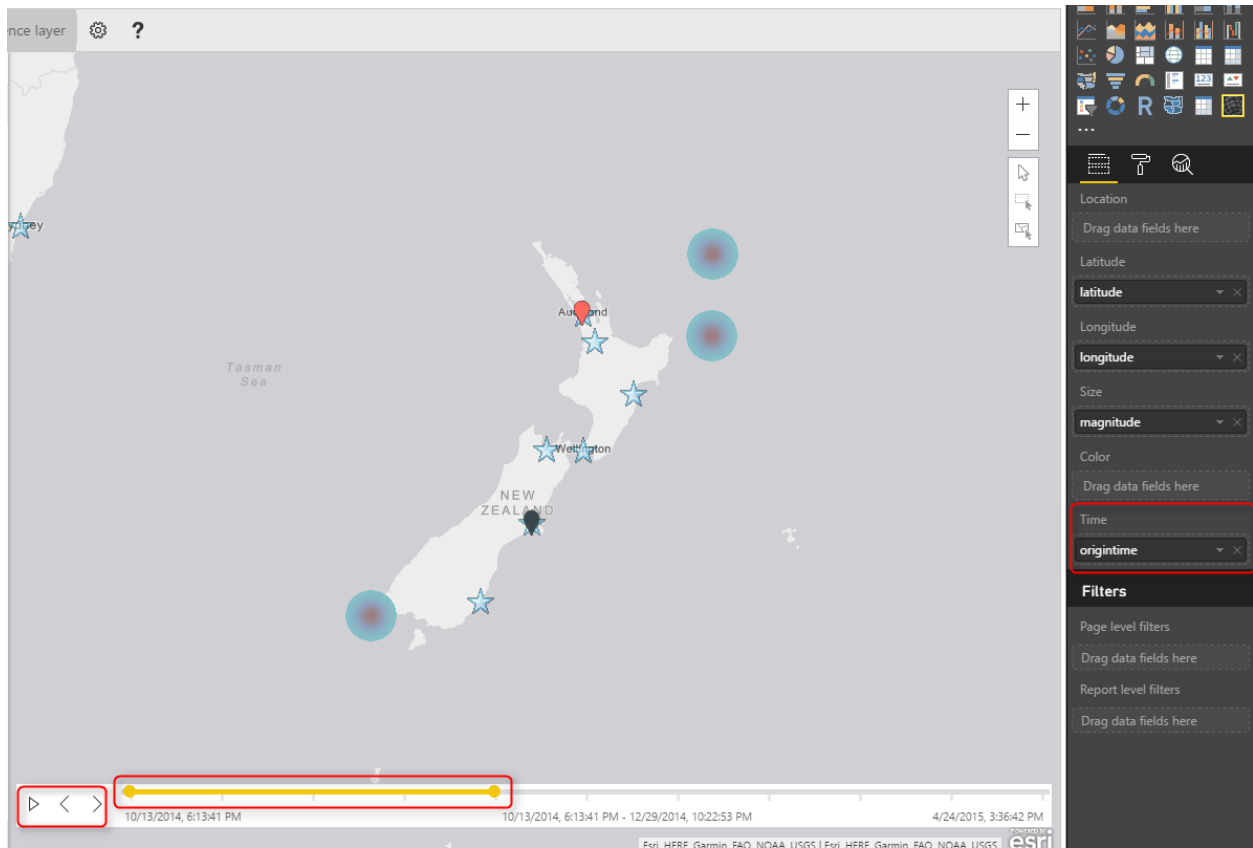


Now if you zoom in more to the New Zealand part of the map, you will see how close are venues. The venue in Christchurch was pretty close to one of earthquakes actually.



Adding Time Dimension

All of above features can be much more insightful with adding a time dimension, and understanding how measure values (in our case earthquakes) happened through the time. If you bring Origintime to the Time section of this visual you will see a new time bar under the map, which you can select the time range for it, and click on play to see how earthquakes happened through the time. Similar to the [Play Axis feature in Scatter Chart](#), Very insightful dimension to add.



Summary

I hope demos of this post has been motivate enough to show you the power of this visual. ArcGIS map visual is not like other map visuals, it is the best of all I would say. There are obviously pros and cons (which I will cover in another post later). However this visual gives you many things; using different themes, adding reference layers from ArcGIS map directory, and adding time dimension. These features are not easy features to use with other types of maps. There is just one part which is not 100% clear, and that is the licensing part of this visual. This visual at the moment is preview and free. However, I'm not sure how long it will be available for free. Because this visual is still in preview mode, expect to see some errors and bugs in it. sometimes you get an error. and you need to close Power BI Desktop and open it, which will be fixed I believe in the general availability version.

Download Demo Files

Enter Your Email to download the file (required)

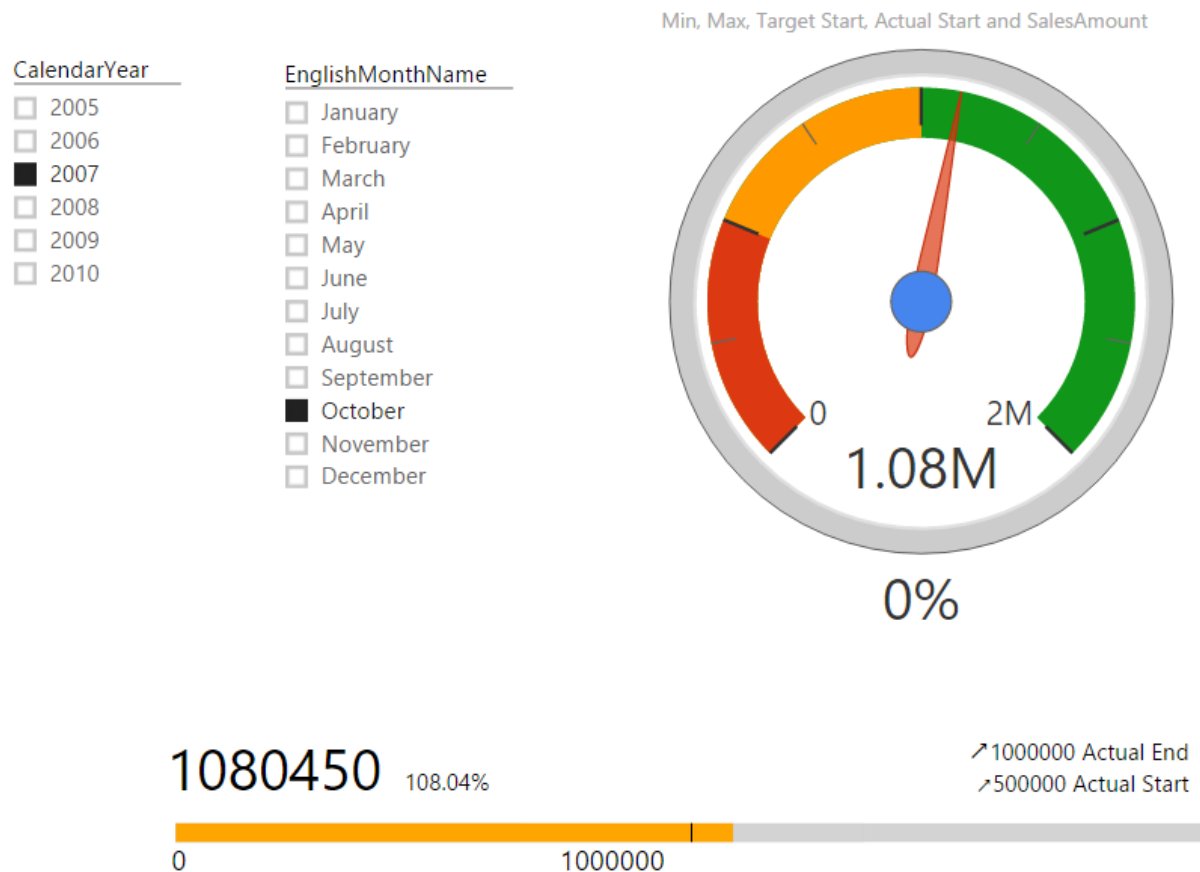
Power BI from Rookie to Rock Star



[Download]

KPIs and Power BI; Visualization Aspect

Published Date: April 12, 2016



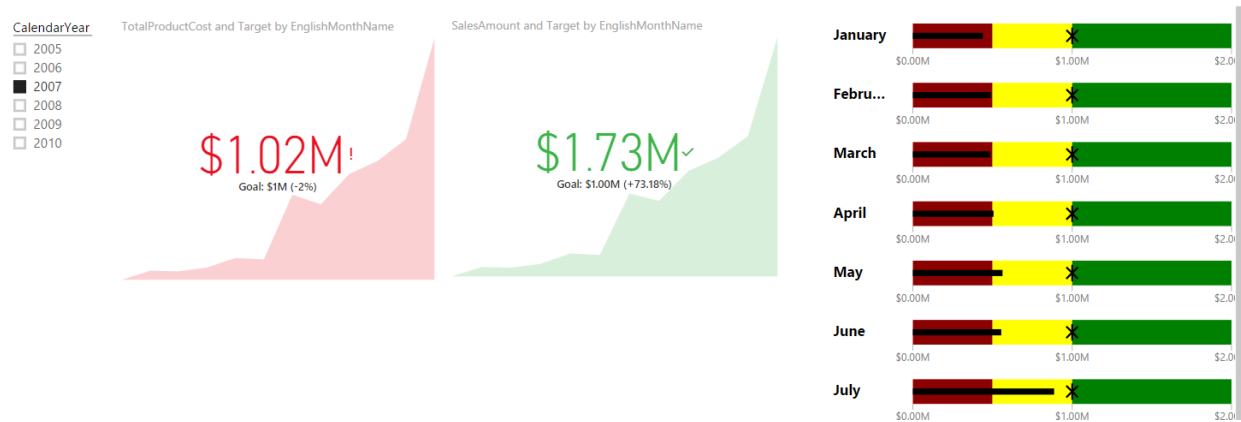
In every reporting solution you might feel KPI as a requirement. KPI or Key Performance Indicator is a measure for business to understand how they are acting in specific area that is very important for their business. Power BI supports built-in visualization for KPI, and there are also some good custom visuals that can be used for this purpose. In this post you will learn how to create KPI visualizations with Power BI.

What is KPI?

Key Performance Indicator is a measure for business to understand how they perform in specific area that is important and many times critical for their business. For example they would like to see how the current year to date sales is going against what they estimated? Is it higher or lower? What is the trend? Is it going upward or downward? There are some visualization options to show KPIs, and from them the most common

are; Gauges, Charts with arrows that shows upward or downward, and color code of red, amber, or green (visualizing the status).

Fortunately Power BI has a built-in KPI visualization ([released 29th of Feb 2016](#)) which can be used for this purpose. There are also some great custom visualizations that I found really useful and like to share them with you here.



KPI Elements

KPIs measuring something and show trend and status. So they have components to work with as below:

- Value; The main measure which we want to evaluate
- Target; What we want to compare the Value with
- Trend; How Value performs in a time period, is it going upward, downward...?
- Status; What is the current status of Value against Target?

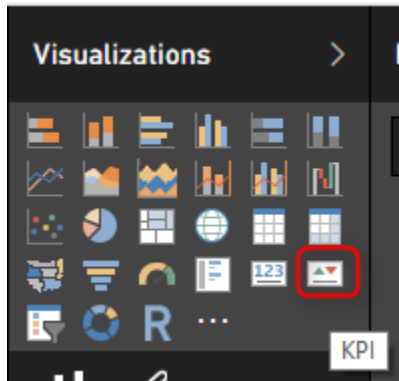
There might be other components as well, but let's keep it simple and work on these elements. In below example you'll see how we apply attributes and configurations for each visualization element to visualize KPI there.

Prerequisite

For running examples of this post you need to install AdventureWorksDW like other examples of [Power BI online book](#).

Built-in KPI

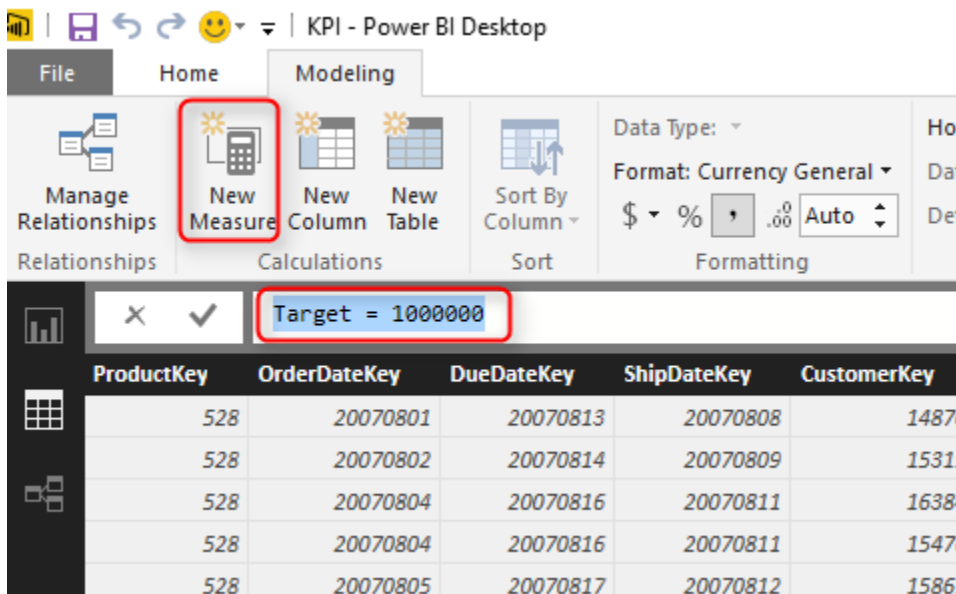
As I mentioned earlier KPI recently added to Power BI and you can find it in built-in visualization list in visualization pane.



Using this item is easy as it only has three data fields to configure:

- Indicator: This would be your KPI Value, or the measure to evaluate.
- Trend Axis; dimension attribute to calculate trend on it. for example Months of year to check the trend over a period of year.
- Target Goals; Your target value. something that KPI value will be compared with.

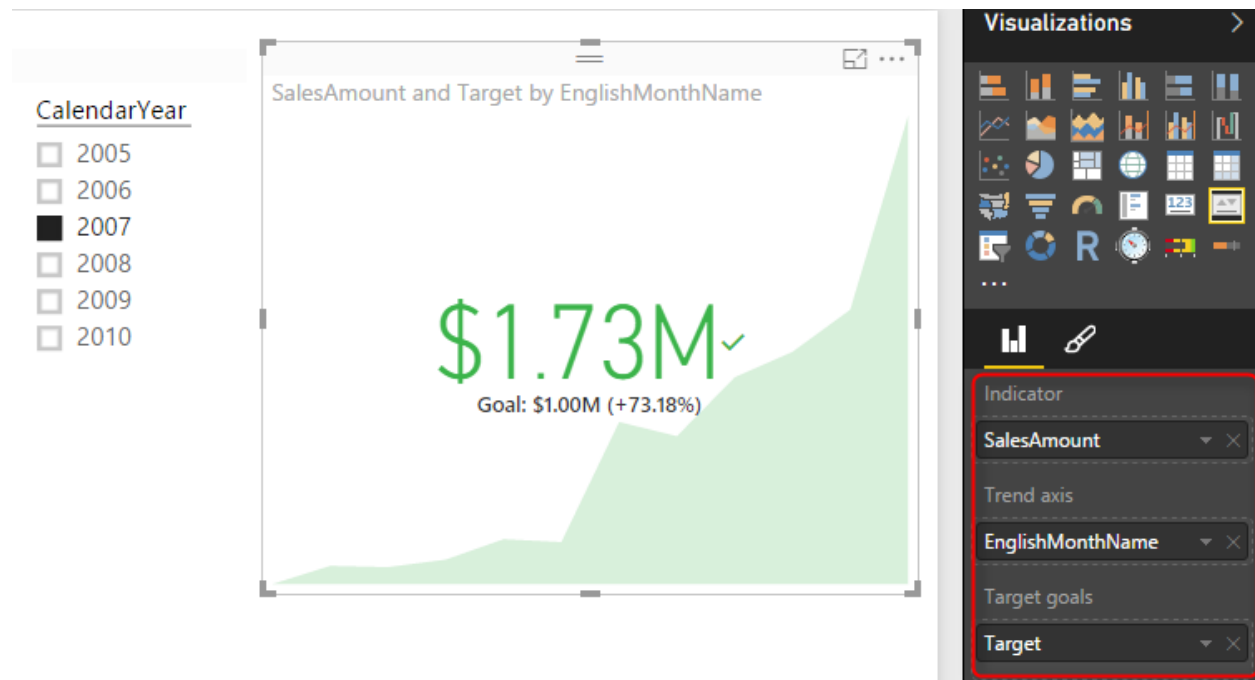
I've only used FactInternetSales and DimDate from AdventureWorksDW in this demo. For simplicity I got the target value as static value. I've created a measure in FactInternetSales called Target, and assigned the static value 1,000,000\$.



As I said this is just for simplicity. Your target value in real world scenarios might come from a forecast table, or a budget table or something like that.

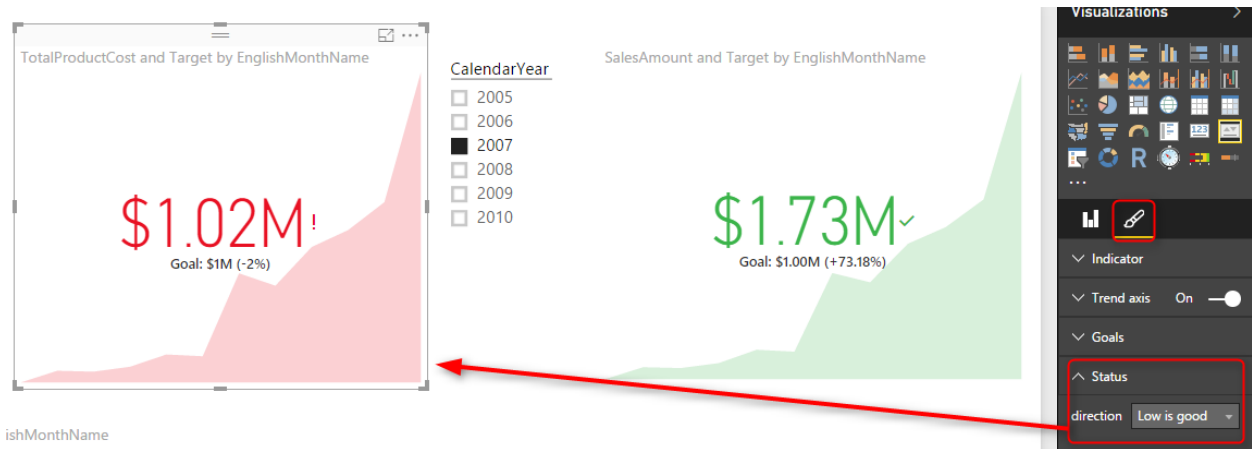
In the report area, create a slicer on Calendar Year. and then drag and drop a KPI element into the report. set Indicator as SalesAmount, Target goals to be the Target

measure, and Trend Axis to be EnglishMonthName (Before doing that you have to Sort EnglishMonthName column by MonthNumberOfYear in the Data Tab). Then this is what you get:



As you see the KPI now shows me the value or indicator (which is Sales Amount) in the middle with large font (\$1.73M), and the Goal or target under that with smaller font (\$1M), and the percentage of the value compared with goal besides it (+73.18%). You also see this colored green which means status is good or in the other words value is higher than expected. in the back ground you also see an area chart of value over period of months (Green area chart shows value over months). You can change the calendar year in slicer and see how the KPI change.

Now lets see another KPI and how to change Status on that. For the other KPI I want to see TotalProductCost as Indicator, and everything else exactly the same. For Costs usually the lower is better which is different from previous KPI. In the first KPI the higher value means the better result, the higher sales is desired. But for Costs I expect decrease. So I can simply change the Status in the Formatting area of KPI to have the direction as Low is good. and here would be the result;



Now you see that despite the fact that TotalProductCost is greater than the goal still it shows it as red, because low is good, and high is bad. The default behavior for Status is that High is good.

Bullet Chart

Bullet chart is a custom visualization published by Microsoft. You need to download it from [Visuals Gallery](#) and then import it into Power BI. This chart has many fields to configure, you can configure them all or part of them. Required fields are Value, Target Value. You need to also set Minimum and Maximum to appropriate values. There are also areas with red, amber, and green colors that can be assigned with Needs Improvement, Satisfactory, Good, and Very Good fields. For this chart I've used a sample table with single line as below:

Min	Max	Target Start	Target End	Actual Start	Actual End	Percentage
0	2000000	1000000	2000000	500000	1000000	10

And I configured Bullet chart with fields from this table combined with FactInternetSales, and DimDate as below;

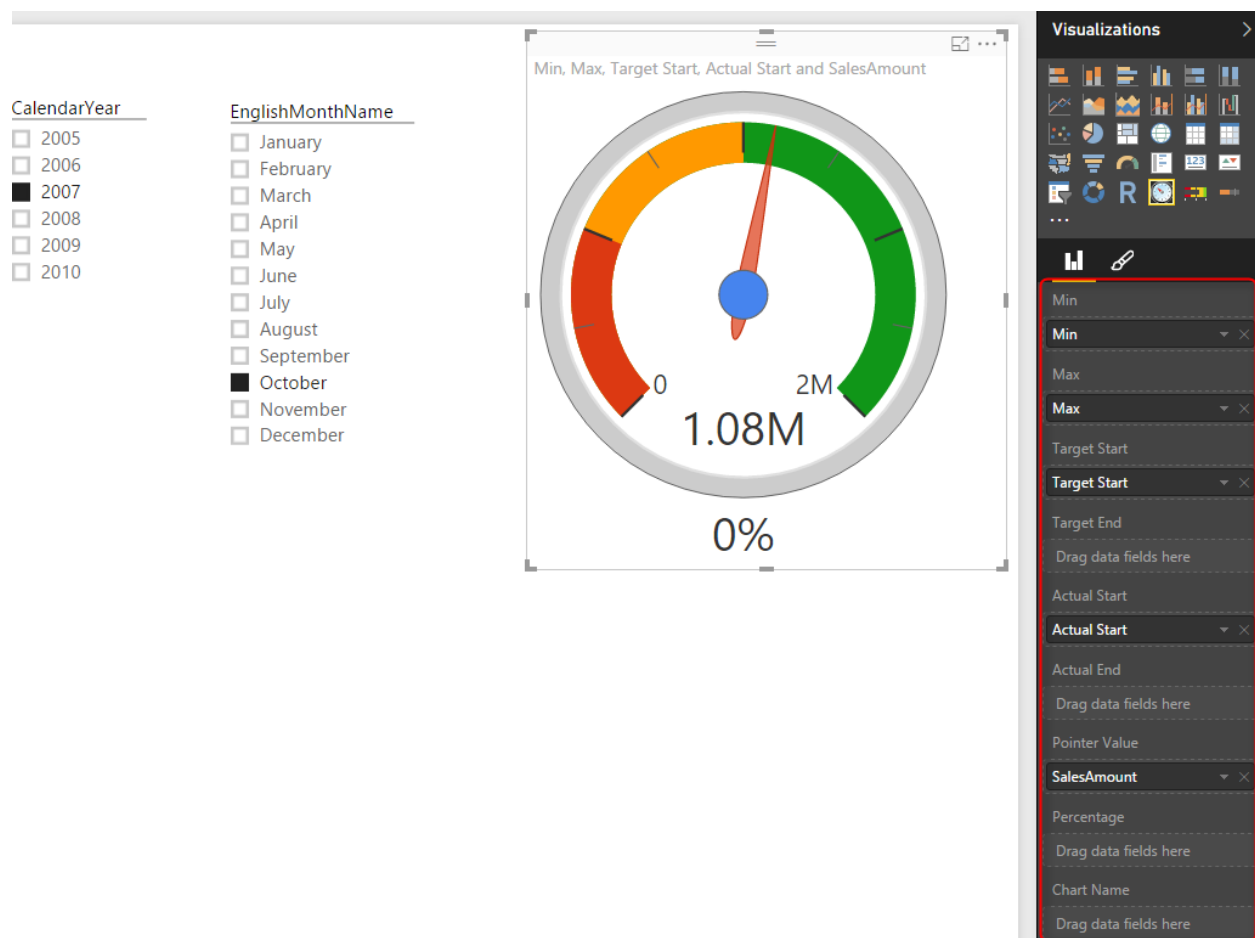


As you can see in the above chart which is for calendar year 2007; that for some months like December the sales amount is in good range (from \$1M to \$2M). for some months it is in amber range like August and June (between \$500K and \$1M), and for some it is below \$500K and are in red range such as January.

As you see in above configuration I haven't used Satisfactory and Good range. Have a play with this chart and configure it to see what you get out of it.

Dial Gauge

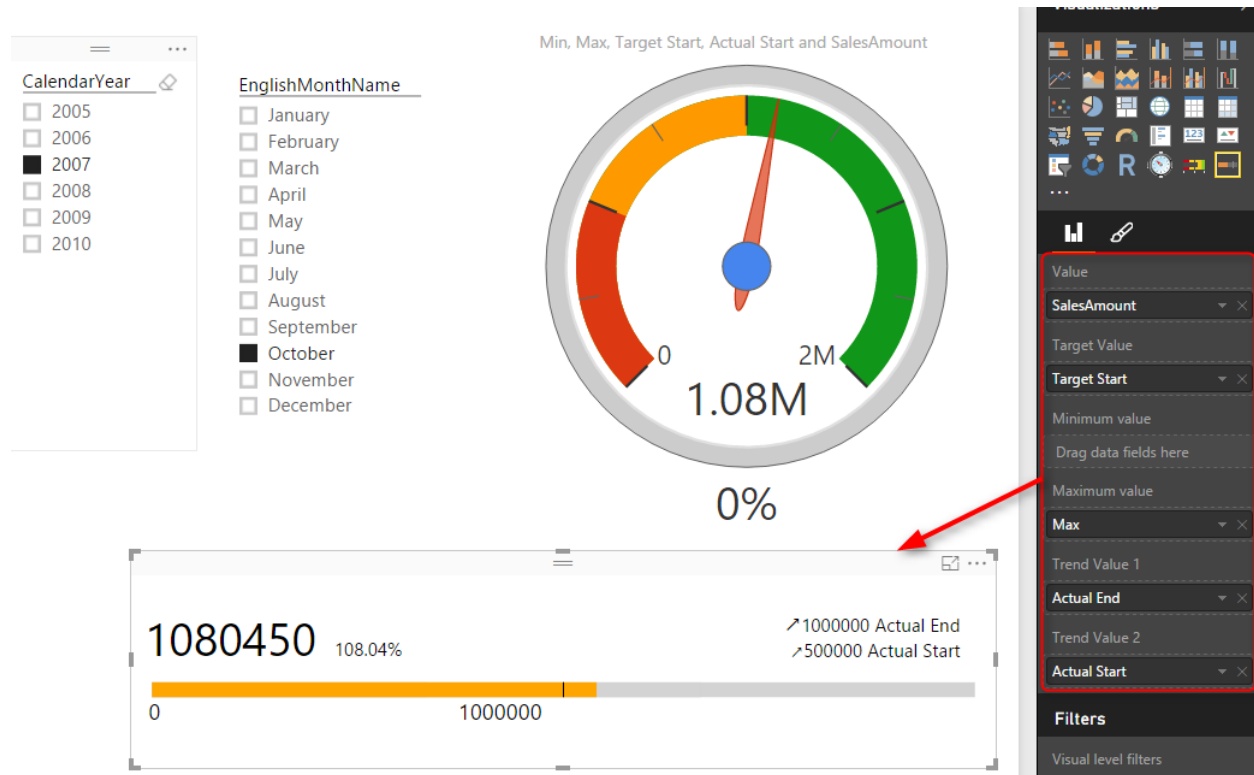
Another custom visualization which gives us the gauge visualization is DialGauge. This visual has been published by CloudFonts Technologies LLP. like the normal gauge for KPI this component gives you flexibility to define ranges for green, amber, and red. The Pointer Value is the value we want to evaluate. You can also set a percentage for this gauge. I've used this gauge with the sample table I've mentioned earlier in this post. and SalesAmount as the pointer value. For this chart I've used another slicer for the month in addition to the slicer for the year.



As you see the pointer value stays in green area because sales amount of October 2007 is \$1.08M which is between \$1M and \$2M. You can check what you see if you change the month to August (expect pointer to fall into amber area).

Linear Gauge

LinearGauge is the custom visual published by MAQ Software. You can set minimum and maximum values, target value, and two trend values. As an example in below chart I've used SalesAmount as the value, and other information from the sample table above.



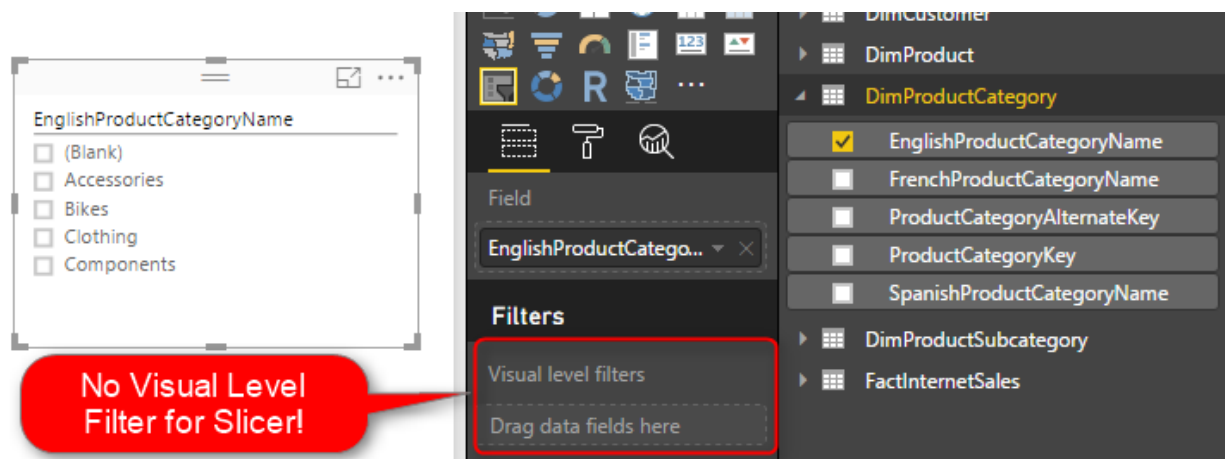
As you see the target value showed with a vertical line in \$1M point. the orange filled section shows the value which is sales amount \$1.08M for month October 2007. Percentage against target is 108.04% which is +8.04%. Trend values shows that sales amount is going upwards compared to each trend value.

Summary

In this post you've seen some visualization items that are useful to show KPIs. There will be definitely more charts and custom visuals in upcoming months. However this post gave you a better feeling about how to use them, or what items to look for and configure when you deal with KPIs. If you are interested to learn more about Power BI, please continue reading [Power BI online book](#).

Filtering Slicer Resolved in Power BI

Published Date: September 26, 2016



You can filter all visuals in Power BI except Slicer! In fact slicer itself is a type of filter, however there are some times that number of items in slicer is too many. So this is normal that we want to filter the items in the slicer itself. Unfortunately current version of Power BI Desktop doesn't support visual level filter on slicer, and I'm sure this feature will come soon. However in the meantime here is a work around for filtering slicers using calculated tables. If you want to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

Defining the Problem

The idea of this blog post came from a question that one of students in my [Power BI course](#) asked to me, and I've found this as a high demand in internet as well. So I've decided to write about it.

You might have too many items to show in a slicer. a slicer for customer name when you have 10,000 customers isn't meaningful! You might be only interested in top 20 customers. Or you might want to pick few items to show in the slicer. With all other visual types (Such as Bar chart, Column chart, line chart....) you can simply define a visual level filter on the chart itself. Unfortunately this feature isn't supported at the time of writing this post for Slicers. However the demand for this feature is already high! you can see the [idea published here](#) in Power BI user voice, so feel free to vote for such feature 😊

The work around in this post might get dated very soon because I expect Power BI team to fix this soon, however in the meantime this workaround will help some people I believe.

There are number of workarounds for this challenge;

1. One can be defining the filter in the page level or report level. but then other elements in the report will be filtered based on predefined items. This might not be what you want.
2. Using some specific visuals as slicer can be the other option. such as stacked column chart or bar chart. However this won't be exactly same feature as slicer. because with clicking on another visual the selection on slicer remains but not the selection of an item in another visual.
3. Using custom visuals. As always custom visuals can be helpful. However some of you might prefer built-in visuals, because some of custom visuals are not designed for mobile friendly reports.
4. Using Calculated tables; Calculated tables can be used to generate filtered views of other tables in the model. Hence this can be used as a workaround. In this blog post I'll explain to you how to leverage calculated table as a filter for Slicer.

Calculated Tables

I have written a [blog post about what Calculated tables are](#) and how to use them for features such as role playing dimensions in Power BI, and fetching filtered views such as top 20 customers. I strongly encourage you to read that post to learn about usages of calculated tables. In summary calculated tables are DAX generated tables, these tables will be part of the model, and can have relationship with other tables in the model.

Calculated Tables to Filter Data

For filtering data in a slicer I thought using calculated tables for such filtering. I know it won't be that easy filtering. and I would need to write a bit of DAX instead of picking items from a list, but at least I have full DAX features to use and create whatever filter I want. Once we create the filtered view as a calculated table, then we can create relationship between that table and other tables, and use it in the slicer. Let's go through that with an example.

Prerequisite

For running this example you need to download and install AdventureWorksDW database from [here](#).

Sample Scenario

Start with Get Data from AdventureWorksDW database, and choose these tables to import: FactInternetSales, DimProduct, DimProductSubCategory, DimProductCategory, and DimCustomer.

Navigator

Display Options ▾

- ☒ DimCustomer
- ☐ DimDate
- ☐ DimDepartmentGroup
- ☐ DimEmployee
- ☐ DimGeography
- ☐ DimOrganization
- ☒ DimProduct
- ☒ DimProductCategory
- ☒ DimProductSubcategory
- ☐ DimPromotion
- ☐ DimReseller
- ☐ DimSalesReason
- ☐ DimSalesTerritory
- ☐ DimScenario
- ☐ FactAdditionalInternationalP...
- ☐ FactCallCenter
- ☐ FactCurrencyRate
- ☐ FactFinance
- ☒ FactInternetSales
- ☐ FactInternetSalesReason

FactInternetSales

Preview downloaded on Wednesday, July 13, 2016

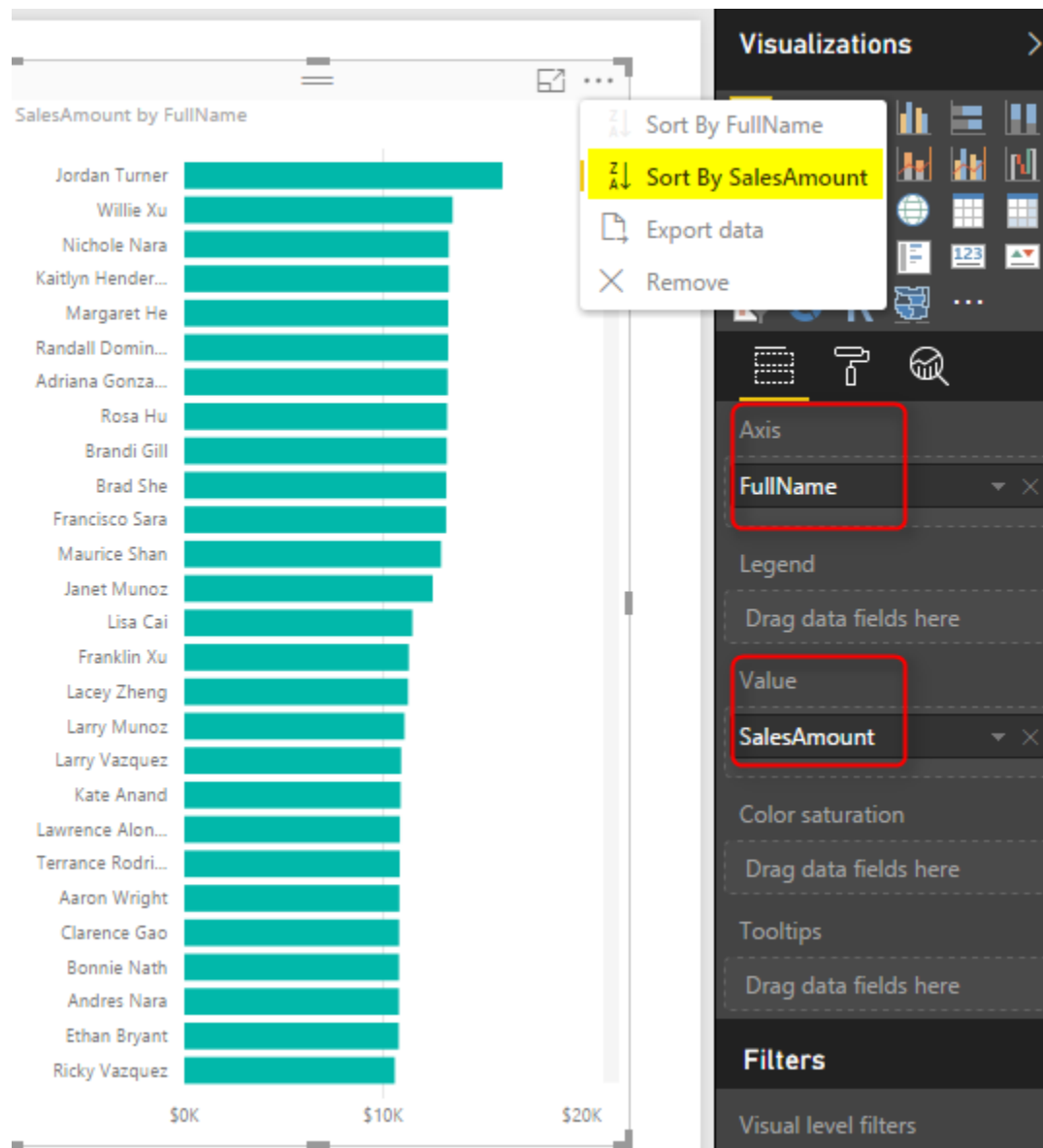
ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey
310	20050701	20050713	20050708	21768	
346	20050701	20050713	20050708	28389	
346	20050701	20050713	20050708	25863	
336	20050701	20050713	20050708	14501	
346	20050701	20050713	20050708	11003	
311	20050702	20050714	20050709	27645	
310	20050702	20050714	20050709	16624	
351	20050702	20050714	20050709	11005	
344	20050702	20050714	20050709	11011	
312	20050703	20050715	20050710	27621	
312	20050703	20050715	20050710	27616	
330	20050703	20050715	20050710	20042	
313	20050703	20050715	20050710	16351	
314	20050703	20050715	20050710	16517	

The data in the preview has been truncated due to size limits.

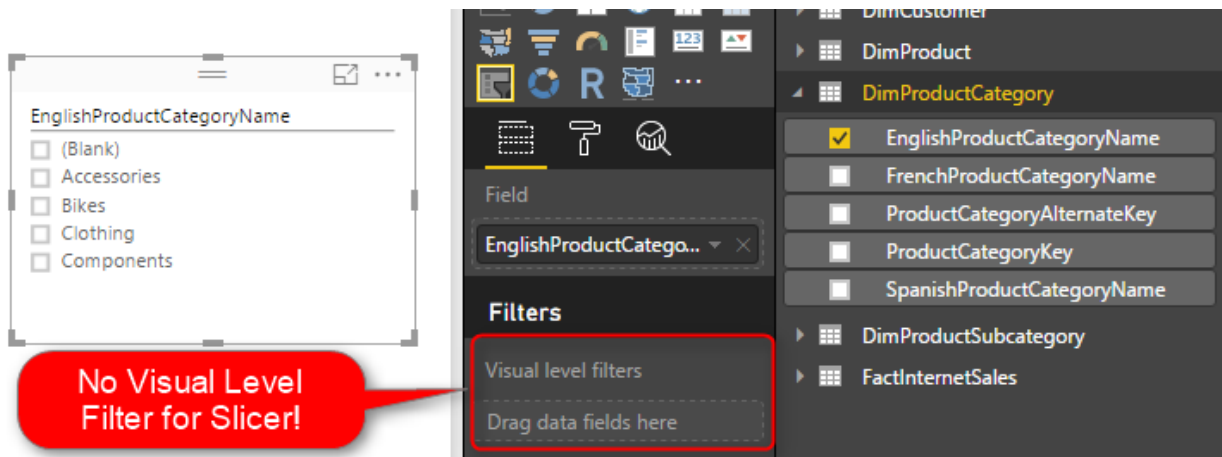
Select Related Tables

Load Edit Cancel

Click on Load, and then build a bar chart with Full Name from DimCustomer as Axis, and SalesAmount from FactInternetSales as Value. Also sort the chart by SalesAmount in descending order.

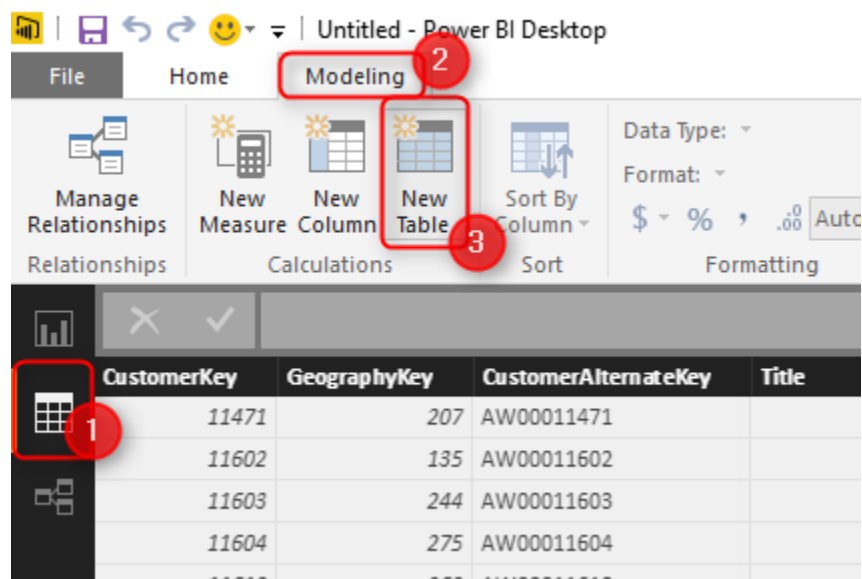


Then Create a Slicer with EnglishProductCategoryName from DimProductCategory. Note that if you drag and drop something into visual level filter it won't be placed there! in this version of Power BI Desktop Visual Level Filters are not supported for slicer! But don't worry we will solve it with calculated tables.

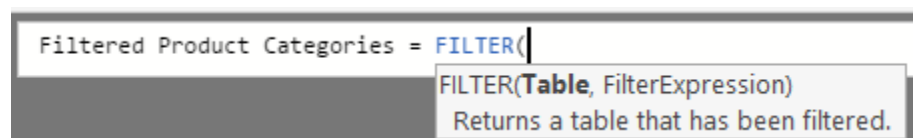


Create Calculated Table for Filtered View

Go to Data Tab, and from Modeling tab, choose Create Calculated Table



Let's create a table for filtered list of product categories. As a simple example, let's show everything except Bikes. I use FILTER function of DAX which gets a table to filter as the first parameter, and the expression as the second parameter.



Now set DimProductCategory as the table, and everything except Bikes as the expression as below;

Filtered Product Categories = `FILTER(DimProductCategory,DimProductCategory[EnglishProductCategoryName]<>"Bikes")`

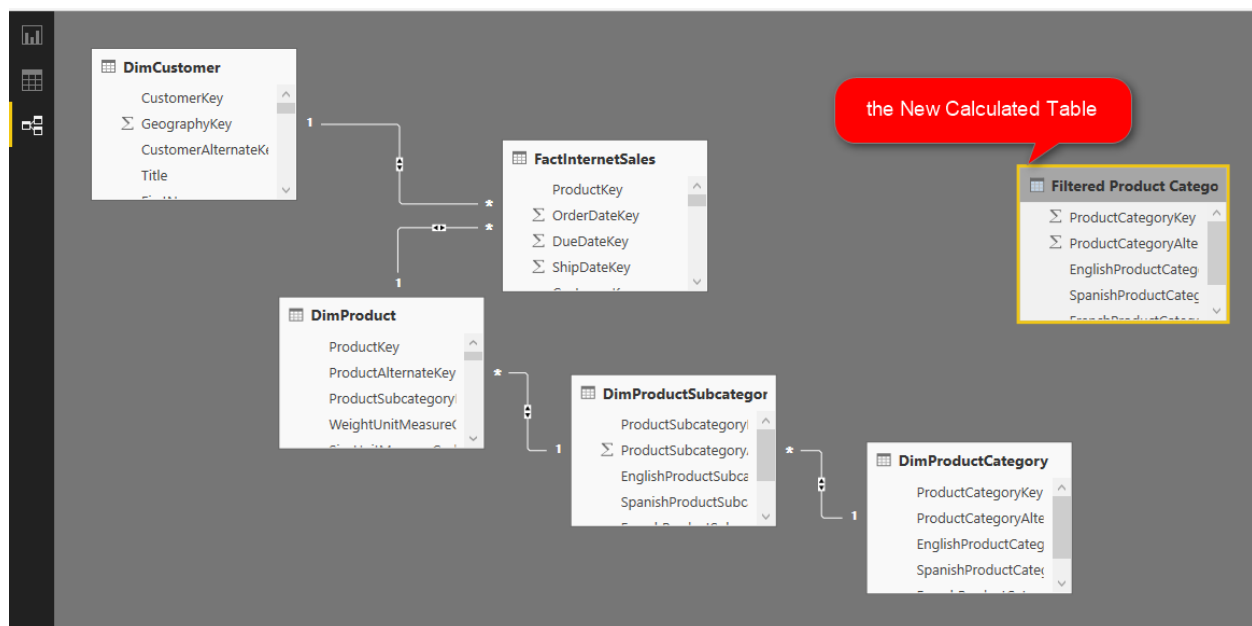
ProductCategoryKey	ProductCategoryAlternateKey	EnglishProductCategoryName	SpanishProductCategoryName	FrenchProductCategoryName
2	2	Components	Componente	Composant
3	3	Clothing	Prenda	Vêtements
4	4	Accessories	Accesorio	Accessoire

Here is the DAX code;

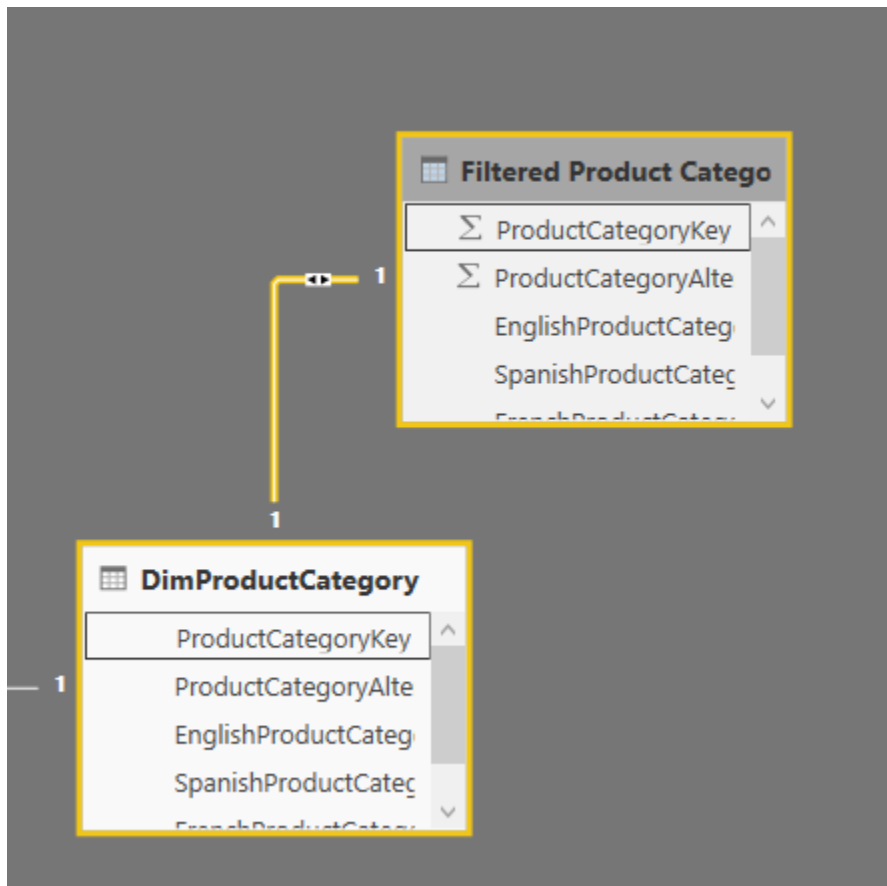
Filtered Product Categories =

`FILTER(DimProductCategory,DimProductCategory[EnglishProductCategoryName]<>"Bikes")`

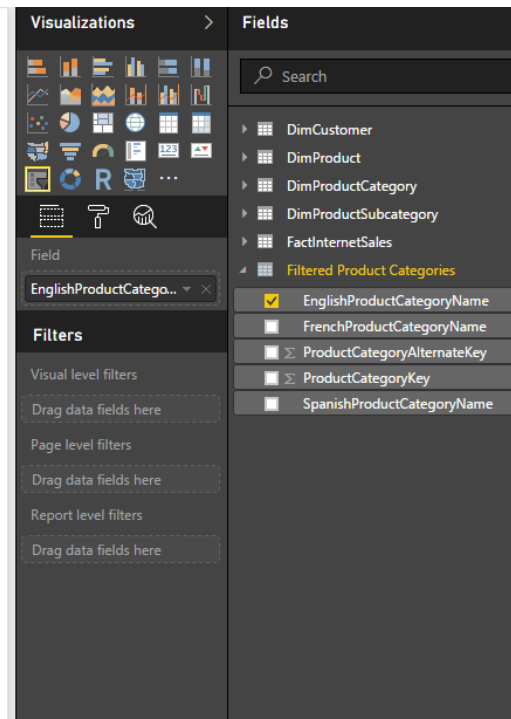
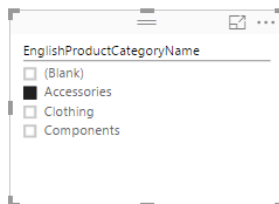
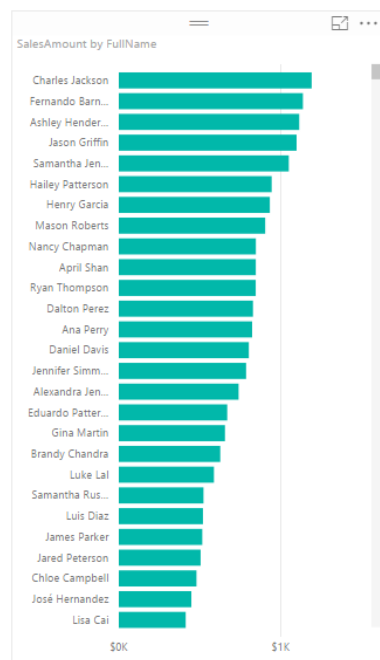
As you can see the filter simply selects everything but Bikes. After creating the table, go to Relationship tab



Create a relationship between DimProductCategory, and Filtered Product Category based on ProductCategoryKey.



Now go back to the Report tab, and in the slicer use EnglishProductCategoryName from Filtered Product Category.



You can see that the slicer is now showing everything but Bikes. Other elements in the report will show everything if slicer hasn't filtered them yet. If slicer filter their data then they will show only filtered data.

Expand the Possibilities

The example I've showed you here was a very basic example of filter. DAX is fully featured language. you can define a very complex filter as well. Think about all possible filters you can create with calculated tables. things like [top 10 customers](#), or any other filters that you cannot simply achieve with visual level filters are possible here.

Limitations

Calculated tables can be created only if you use Import Data into Power BI. these are generated based on DAX functions, and are not available as DirectQuery.

Calculated tables consumes memory, but you can reduce it with selecting minimum number of columns in calculated tables, and some DAX performance tuning considerations.

Summary

In summary I explained to you how to create a visual level filter on a slicer using calculated tables. This method might not be as easy as picking some items from a list in a visual level filter section of the report, but it is very powerful because it uses DAX filtering for generating calculated tables. creating filters such as top 10 customers are possible here with DAX expressions. There are some limitation though, DAX is only supported if you import data into Power BI. the DirectQuery don't support calculated tables.

Be Fitbit BI Developer in Few Steps: Step 3 Visualization

Published Date: December 3, 2015



In previous steps you've learned [how to extract and mash up the data from CSV](#) and [automate the process of looping through all CSV files in the source folder](#). In this step we are going to build visualizations based on the data set we've built so far. In this post you will see how easy is to build charts in Power BI desktop and how customize-able they are. You will also learn about using few DAX calculations to produce new measures that helps building visualization elements.

Adding Date Dimension

Before going to visualization part let's add a date dimension to our queries. I have previously explained an example of [date dimension with Power Query](#), so All you need to do is to use that as the source. The Date dimension query produces table below as result:

Year	Month	Day	FullDateAlt	DateKey	DateFullName	Fiscal Year	Fiscal Qu.	Calendar Q	IsWeek	DayOf	Month Nan	Day of Week Na	HolidayDescription	IsPubl
2015	1	1	42005	20150101	01 January 2015	2015	3	1	1	1	4 January	Thursday	New Year's Day	
2015	1	2	42006	20150102	02 January 2015	2015	3	1	1	1	5 January	Friday	Day after New Year's Day	
2015	1	3	42007	20150103	03 January 2015	2015	3	1	0	0	6 January	Saturday		
2015	1	4	42008	20150104	04 January 2015	2015	3	1	0	0	0 January	Sunday		
2015	1	5	42009	20150105	05 January 2015	2015	3	1	1	1	1 January	Monday		
2015	1	6	42010	20150106	06 January 2015	2015	3	1	1	1	2 January	Tuesday		
2015	1	7	42011	20150107	07 January 2015	2015	3	1	1	1	3 January	Wednesday		
2015	1	8	42012	20150108	08 January 2015	2015	3	1	1	1	4 January	Thursday		
2015	1	9	42013	20150109	09 January 2015	2015	3	1	1	1	5 January	Friday		
2015	1	10	42014	20150110	10 January 2015	2015	3	1	0	0	6 January	Saturday		
2015	1	11	42015	20150111	11 January 2015	2015	3	1	0	0	0 January	Sunday		
2015	1	12	42016	20150112	12 January 2015	2015	3	1	1	1	1 January	Monday		
2015	1	13	42017	20150113	13 January 2015	2015	3	1	1	1	2 January	Tuesday		
2015	1	14	42018	20150114	14 January 2015	2015	3	1	1	1	3 January	Wednesday		
2015	1	15	42019	20150115	15 January 2015	2015	3	1	1	1	4 January	Thursday		
2015	1	16	42020	20150116	16 January 2015	2015	3	1	1	1	5 January	Friday		
2015	1	17	42021	20150117	17 January 2015	2015	3	1	0	0	6 January	Saturday		

It is also configurable as it mentioned in the [blog post](#), and it includes public holidays (New Zealand holidays actually, you have to change it a bit to cover your country holidays which is easy).

I have above query in an Excel file and I use that as a source in Power BI Desktop. So here is what I have in my Query Editor; A query for Date (which comes from date dimension query example explained above), and the Tracking Data query (which we have built it in previous step of this exercise);

2 Queries

Tracking Data

Date

×

✓

fx

= Table.TransformColumnTypes(Merge1_Table,{{"Year", Int64.Type}, {"Month", Int64.Type}, {"Day", Int64.Type}, {"FullDateAlternateKey", Date.Type}, {"DateKey", Int64.Type}, {"DateFullName", Text.Type}, {"Fiscal Year", Int64.Type}, {"Fiscal Quarter", Int64.Type}, {"Calendar Quarter", Int64.Type}})

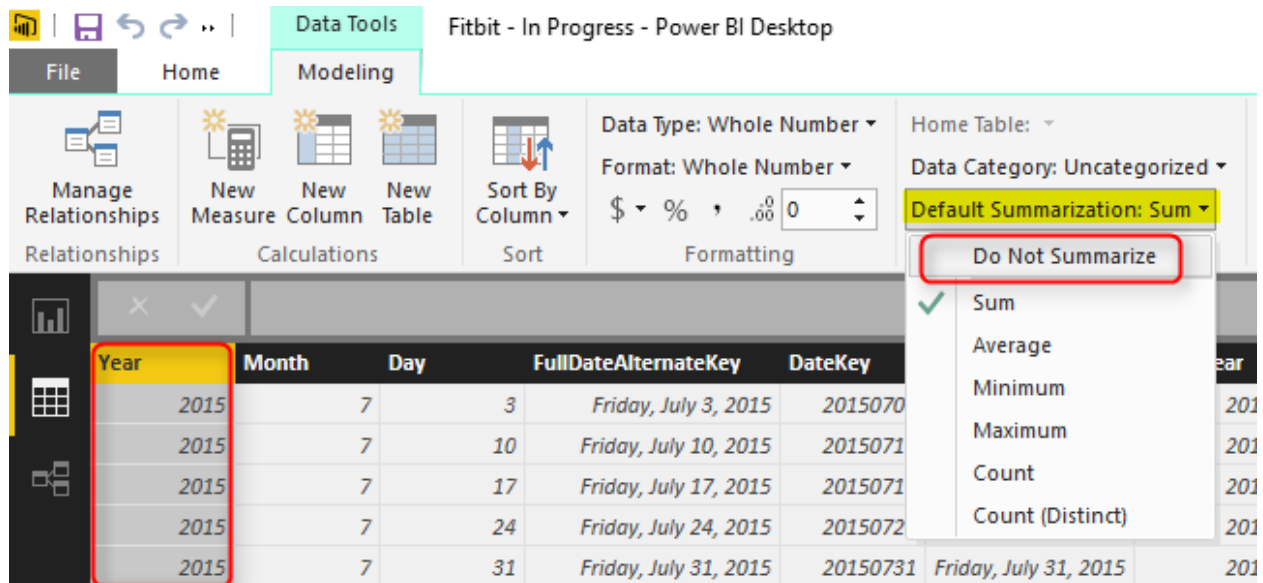
	Year	Month	Day	FullDateAlternateKey	DateKey	DateFullName	Fiscal Year	Fiscal Quarter	Calendar Quarter
1	2015	1	1	42005	20150101	1/1/2015	2015	3	
2	2015	1	2	42006	20150102	1/2/2015	2015	3	
3	2015	1	3	42007	20150103	1/3/2015	2015	3	
4	2015	1	4	42008	20150104	1/4/2015	2015	3	
5	2015	1	5	42009	20150105	1/5/2015	2015	3	
6	2015	1	6	42010	20150106	1/6/2015	2015	3	
7	2015	1	7	42011	20150107	1/7/2015	2015	3	
8	2015	1	8	42012	20150108	1/8/2015	2015	3	
9	2015	1	9	42013	20150109	1/9/2015	2015	3	
10	2015	1	10	42014	20150110	1/10/2015	2015	3	
11	2015	1	11	42015	20150111	1/11/2015	2015	3	
12	2015	1	12	42016	20150112	1/12/2015	2015	3	
13	2015	1	13	42017	20150113	1/13/2015	2015	3	
14	2015	1	14	42018	20150114	1/14/2015	2015	3	

As you see in the above table the FullDateAlternateKey is not showing the right date format. I just right click on it and change it to Date format so I have proper date field there;

	Year	Month	Day	FullDateAlternateKey	DateKey	DateFullName	Fiscal Year	Fiscal Quarter
1	2015	1	1	1/1/2015	20150101	1/1/2015	2015	
2	2015	1	2	1/2/2015	20150102	1/2/2015	2015	
3	2015	1	3	1/3/2015	20150103	1/3/2015	2015	
4	2015	1	4	1/4/2015	20150104	1/4/2015	2015	
5	2015	1	5	1/5/2015	20150105	1/5/2015	2015	
6	2015	1	6	1/6/2015	20150106	1/6/2015	2015	
7	2015	1	7	1/7/2015	20150107	1/7/2015	2015	
8	2015	1	8	1/8/2015	20150108	1/8/2015	2015	

A Touch to Modelling

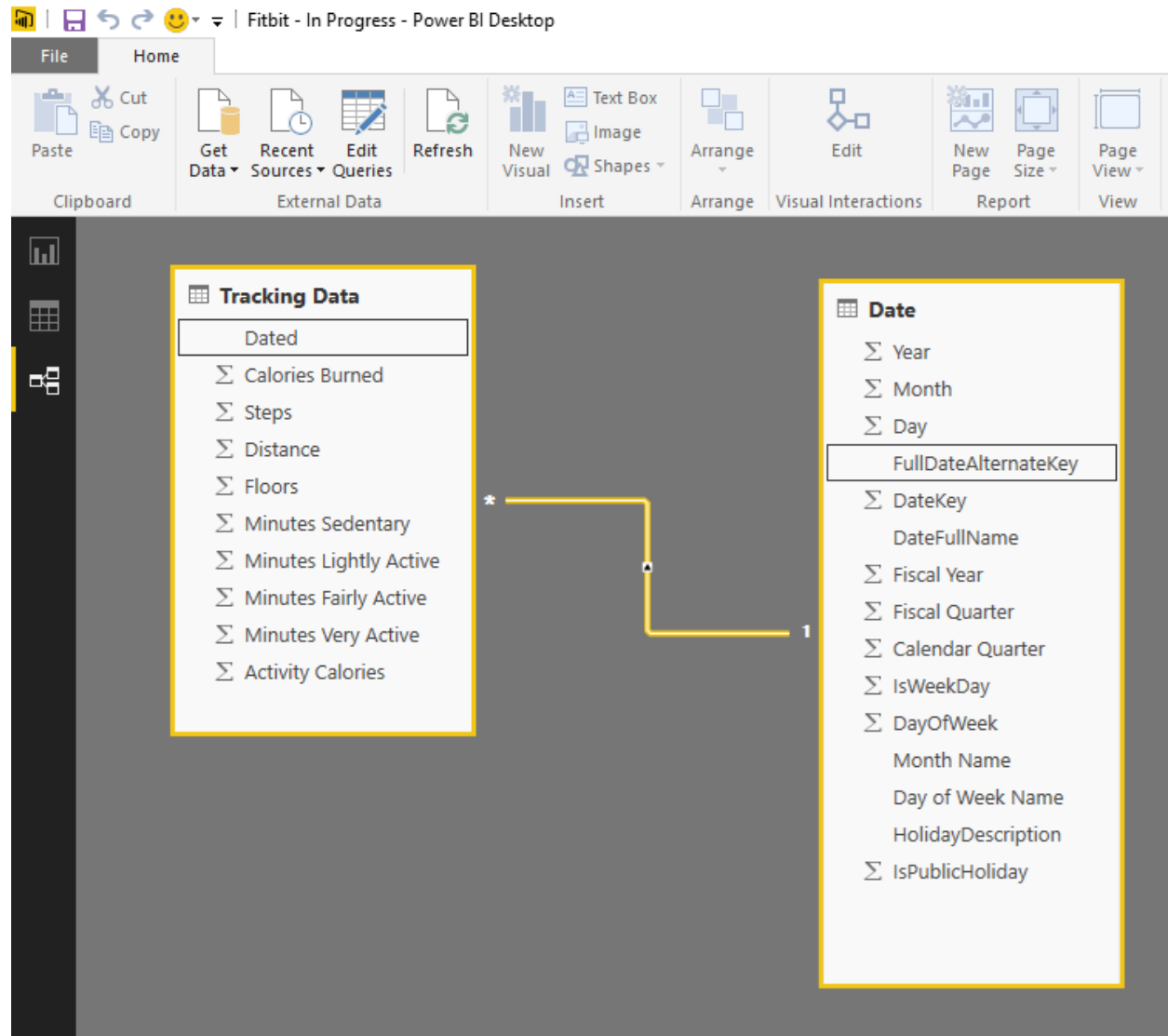
Now Close the query editor and apply. This will load result of both queries into the memory and model for Power BI Power Pivot component. Go to Data tab, and expand the Date under fields section. You will see that many columns have a sigma icon beside them, the reason is that Power BI desktop based on their data type (whole number or decimal) decided that these are values that can be summarized. You can change this behavior by choosing any of those columns and change default stigmatization to Do Not Summarize.



The screenshot shows the Power BI Desktop interface with the Modeling tab selected. The ribbon includes options like Manage Relationships, New Measure, New Column, New Table, Sort By Column, and Formatting. The data table below has columns: Year, Month, Day, FullDateAlternateKey, and DateKey. The 'Year' column is highlighted with a red box. A context menu is open over the 'Year' column, showing the 'Default Summarization' dropdown set to 'Sum'. The 'Do Not Summarize' option is highlighted with a red box.

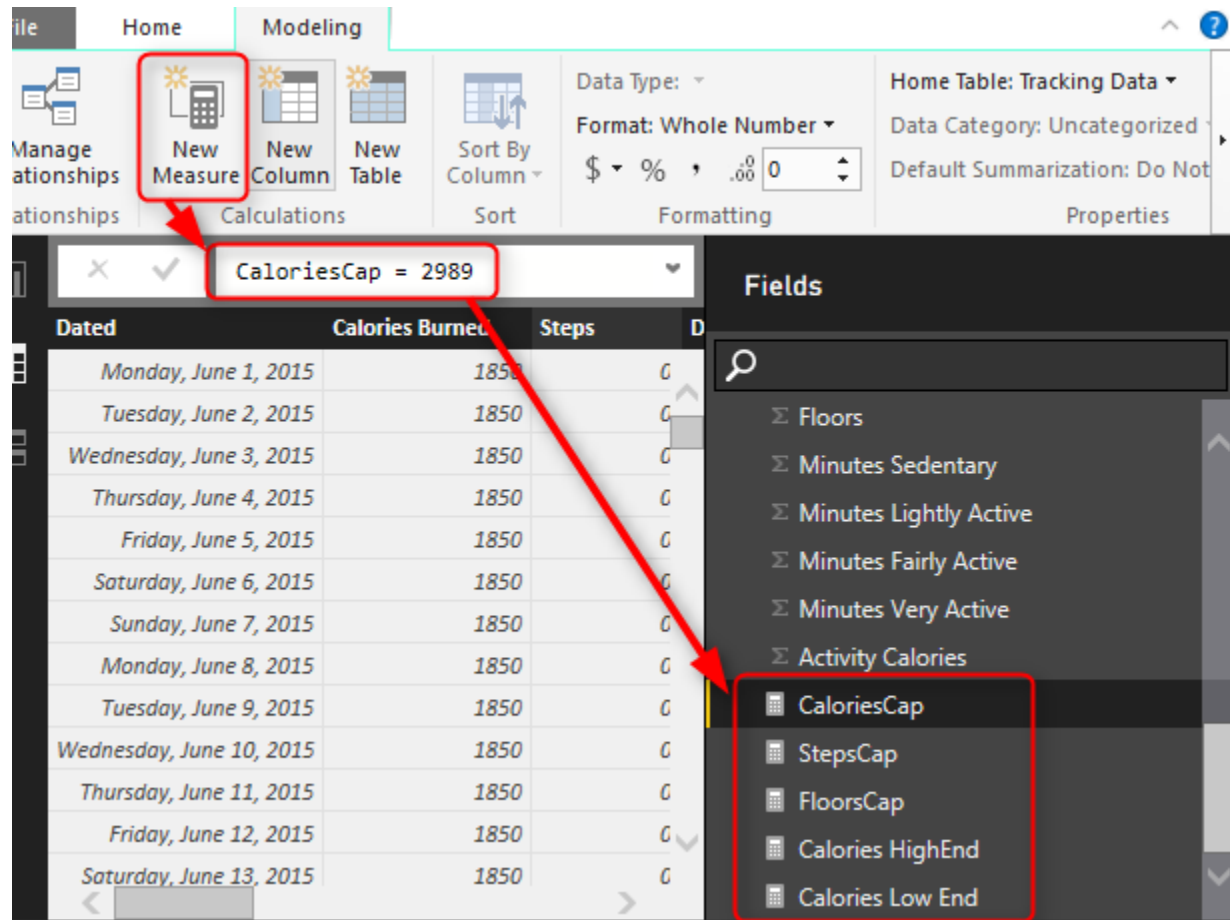
Year	Month	Day	FullDateAlternateKey	DateKey
2015	7	3	Friday, July 3, 2015	20150703
2015	7	10	Friday, July 10, 2015	20150710
2015	7	17	Friday, July 17, 2015	20150717
2015	7	24	Friday, July 24, 2015	20150724
2015	7	31	Friday, July 31, 2015	20150731

Now go to Relationship tab and create relationship between two tables based on their full date columns which are; FullDateAlternateKey Column from Date, and Dated Column in Tracking Data table.



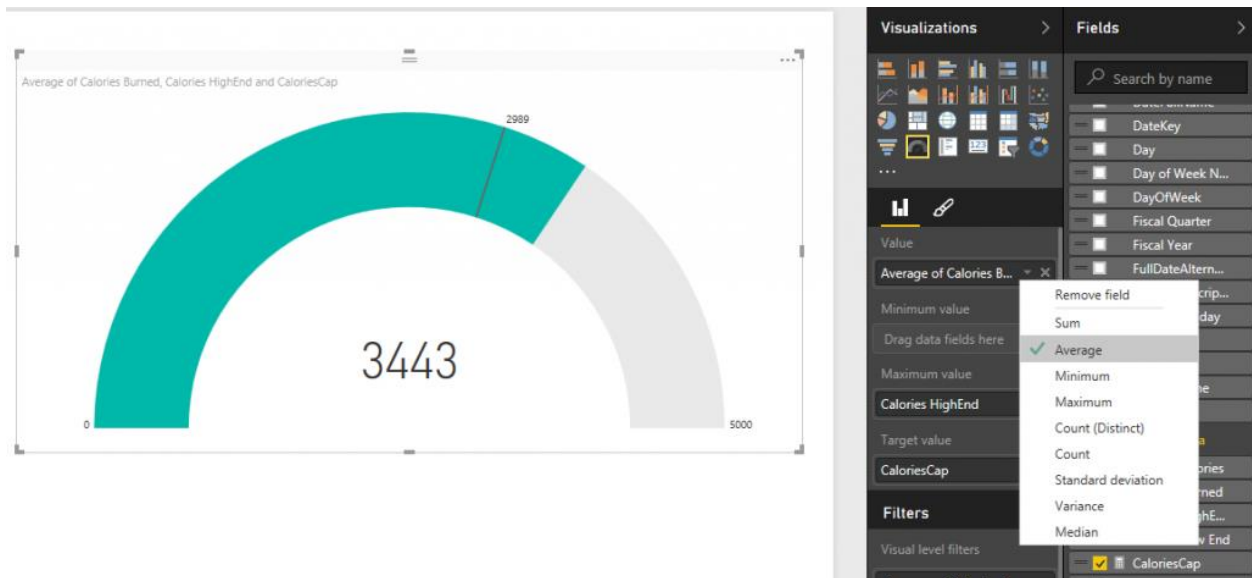
Adding Few Static Measures

Fitbit calculates based on my current weight and age (I assume) how much calories I have to spend each day. I don't know that calculation, So I create a static measure with the value of 2989 for the amount of calories I have to spend each day. I also create StepsCap measure with 12000 value showing that I have to walk 12000 steps a day, and another one for FloorCap with the value of 10. I created a Calories HighEnd measure with 5000 calories as value (I will die if I burn more than that!). You can create all these measures easily in Data tab.

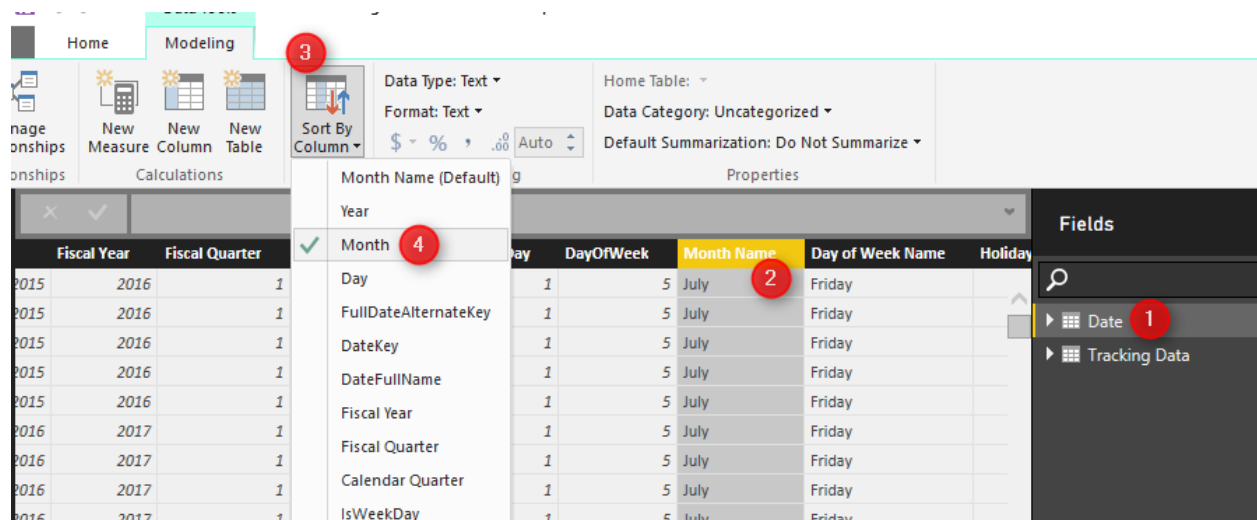


Let's Visualize

It's time to visualize what we have so far in the data set. Go to Report tab, and start with building a Gauge. Then drag and drop CaloriesCap as Target Value. You can also put Calories HighEnd measure as Maximum. and then drag and drop Calories measure into the value. change the aggregation from sum to Average as it showed in below image.



Easy! and lucky me who spent (in average) more than the calories I have to spend each day. Do the same for Floor with FloorCap and Floors measure. Now Add Three Slicers for; Year, Month Name, and Day. When you create slicer for Month Name you will see month names are ordered by their alphabetic order of names which is not right. Go to data tab, and click on Month Name Column of Date table. Then from the menu options of Modeling choose Sort By Column and then choose Month (which is the number of month in year);



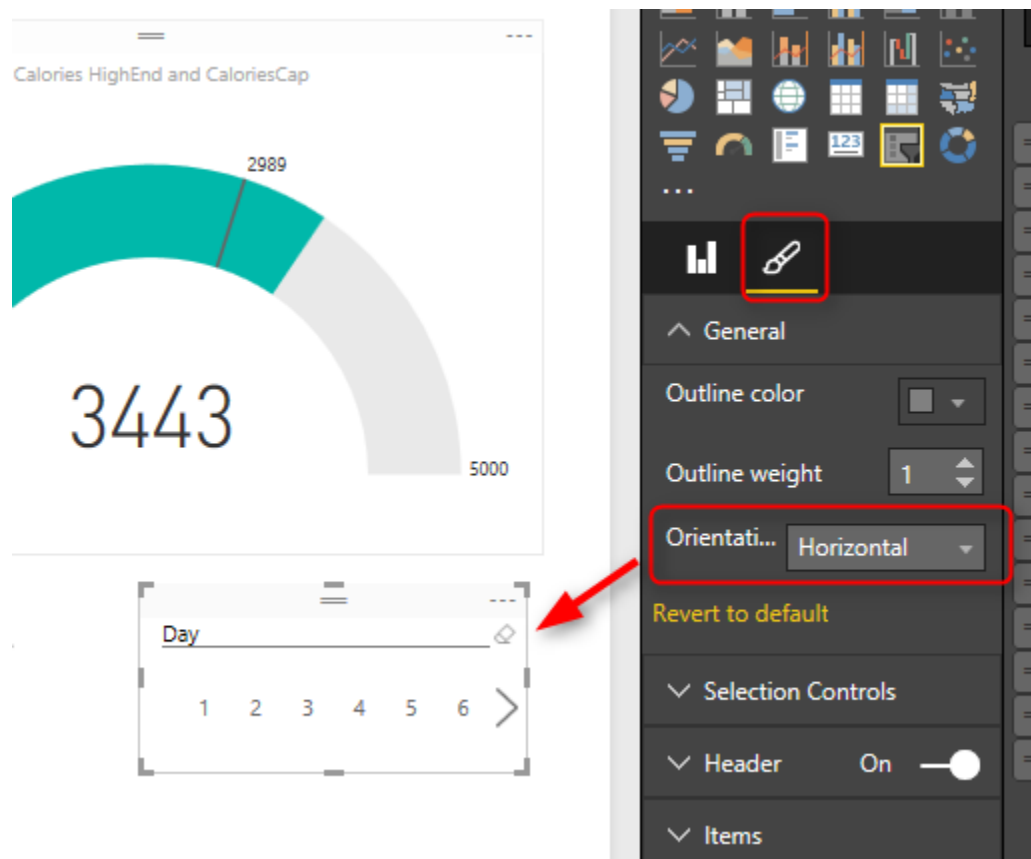
The screenshot shows the Power BI Data View with the Date table. The columns are: Day, DayOfWeek, Month Name, Day of Week Name, and Holiday. The 'Month Name' column is highlighted in yellow. The 'Sort By Column' menu is open, showing the 'Month' column selected. The 'Month' column is highlighted in the table. The 'Month Name' column is highlighted in the table. The 'Month' column is highlighted in the table.

Fiscal Year	Fiscal Quarter	Day	DayOfWeek	Month Name	Day of Week Name	Holiday
2015	2016	1	5	July	Friday	
2015	2016	1	5	July	Friday	
2015	2016	1	5	July	Friday	
2015	2016	1	5	July	Friday	
2015	2016	1	5	July	Friday	
2015	2016	1	5	July	Friday	
2016	2017	1	5	July	Friday	
2016	2017	1	5	July	Friday	
2016	2017	1	5	July	Friday	
2016	2017	1	5	July	Friday	

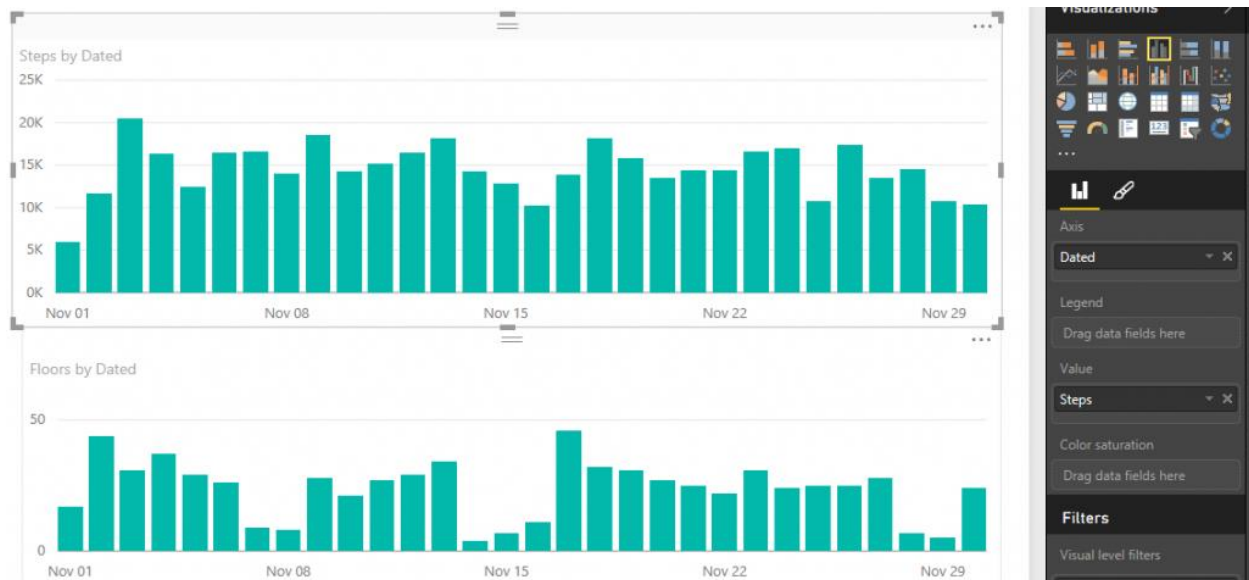
Now that you've set ordering you will be able to see Month names in the correct order in the slicer as below;

Year	Month Name	Day
<input type="checkbox"/> 2015	<input type="checkbox"/> January	<input type="checkbox"/> 1
<input type="checkbox"/> 2016	<input type="checkbox"/> February	<input type="checkbox"/> 2
<input type="checkbox"/> 2017	<input type="checkbox"/> March	<input type="checkbox"/> 3
<input type="checkbox"/> 2018	<input type="checkbox"/> April	<input type="checkbox"/> 4
<input type="checkbox"/> 2019	<input type="checkbox"/> May	<input type="checkbox"/> 5
<input type="checkbox"/> 2020	<input type="checkbox"/> June	<input type="checkbox"/> 6
<input type="checkbox"/> 2021	<input type="checkbox"/> July	<input type="checkbox"/> 7
<input type="checkbox"/> 2022	<input type="checkbox"/> August	<input type="checkbox"/> 8
	<input type="checkbox"/> September	<input type="checkbox"/> 9
	<input type="checkbox"/> October	<input type="checkbox"/> 10
	<input type="checkbox"/> November	<input type="checkbox"/> 11
	<input type="checkbox"/> December	<input type="checkbox"/> 12
		<input type="checkbox"/> 13
		<input type="checkbox"/> 14

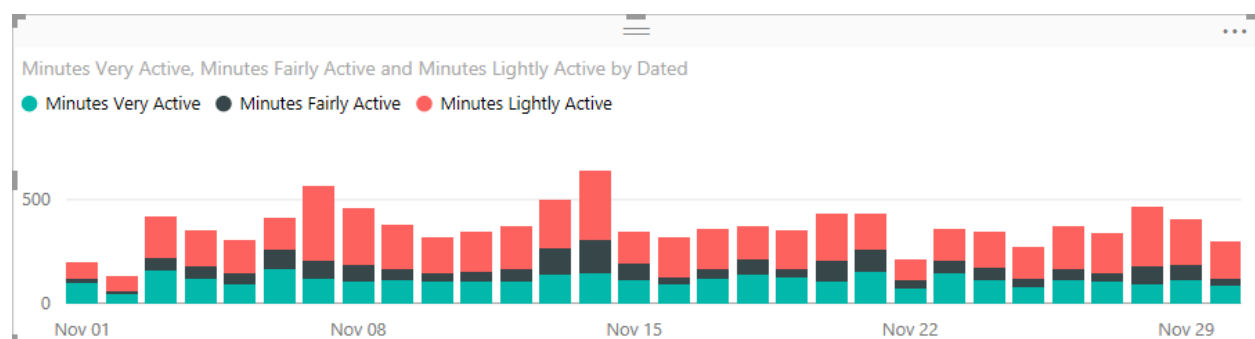
Change the orientation of Day slicer to be horizontal rather than vertical in the formatting option of the slicer



Now let's add two column charts; one for Steps as Value, and Dated column (from Tracking Data table) as axis. The other one with same axis, and Floors as value



Add a stacked column chart with Dated as axis, and three measures as value: Minutes Very Active, Minutes Fairly Active, and Minutes Lightly Active.



Add Also an average of all of three measure above in a pie chart to show the total average. and Add two Card visualization one for Average of steps, and another for average of distance.

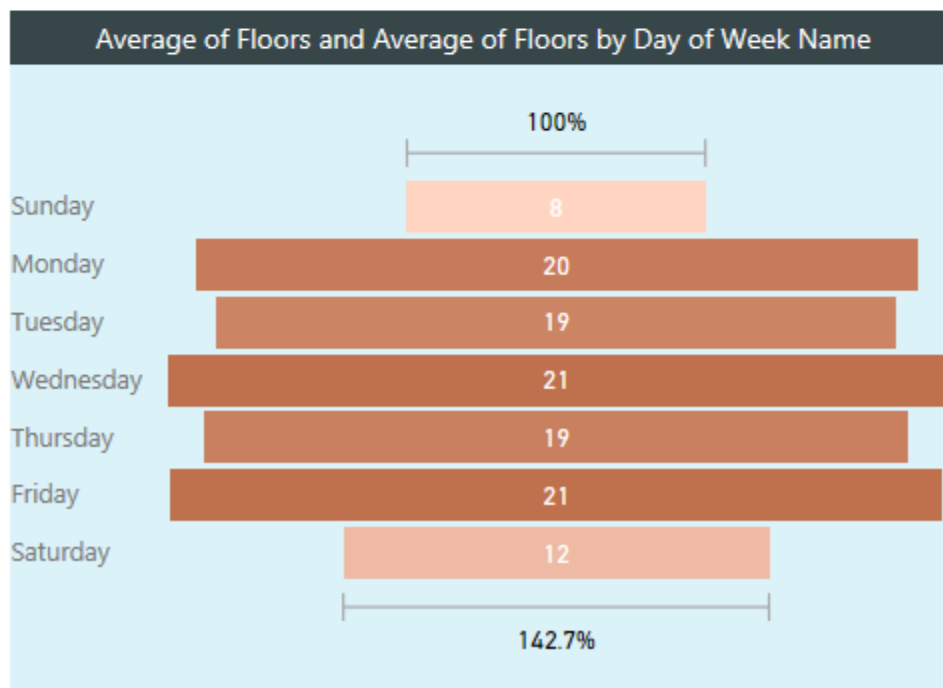
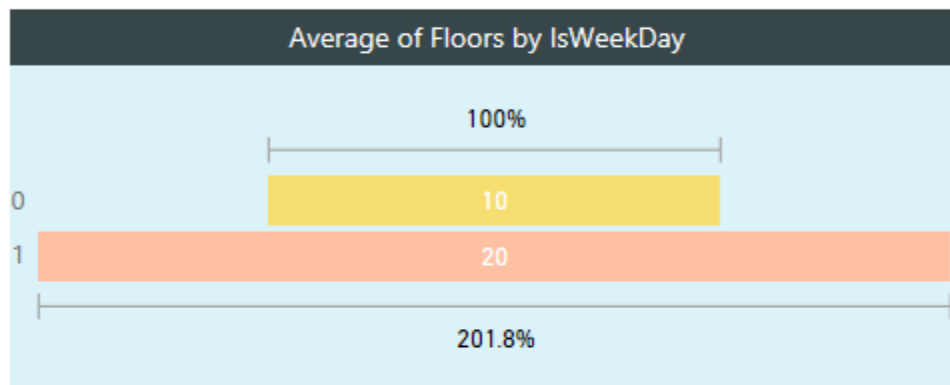
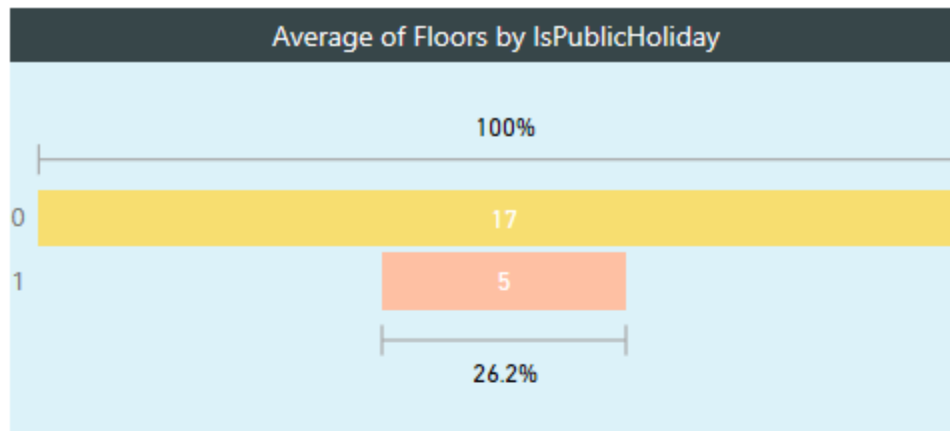
Well, you can customize the visualization as you want in the formatting option of each chart or element. Have a play with it and create something remarkable. I'm sure you design it better than me, here is what I built in few minutes;



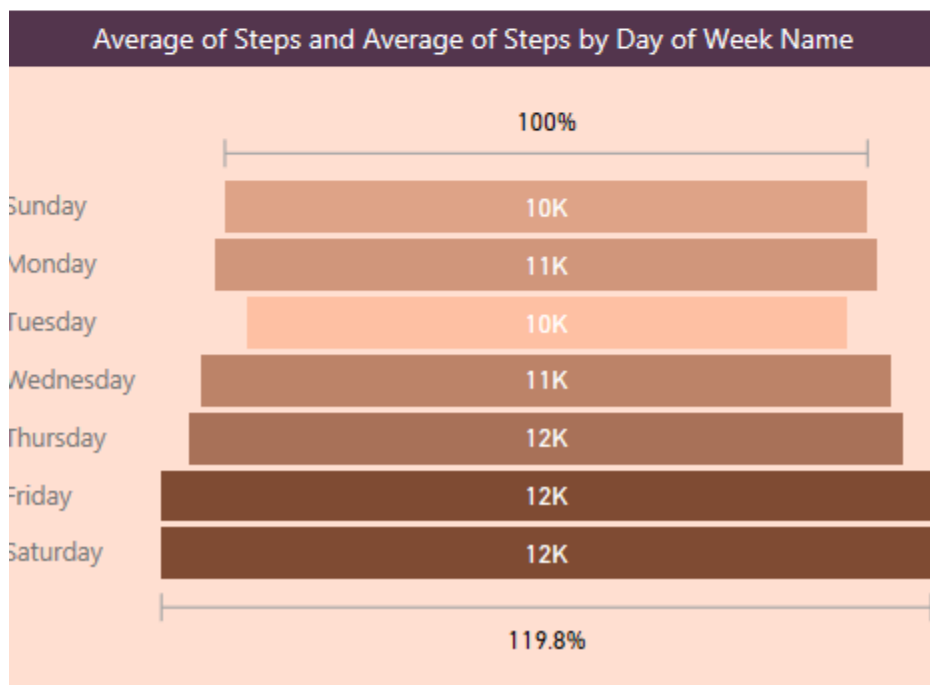
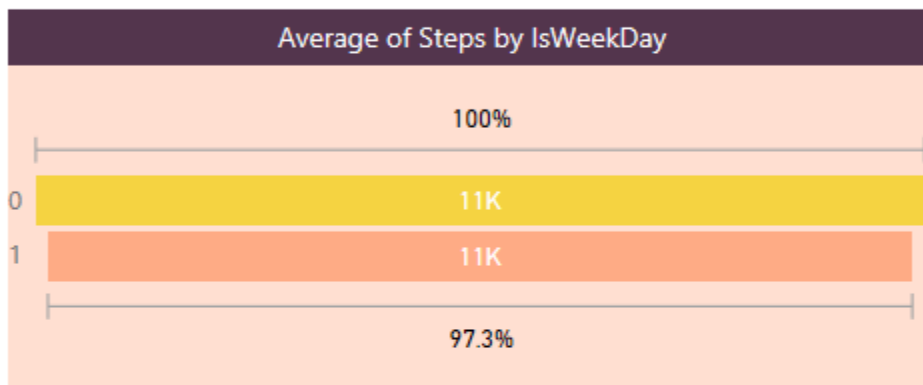
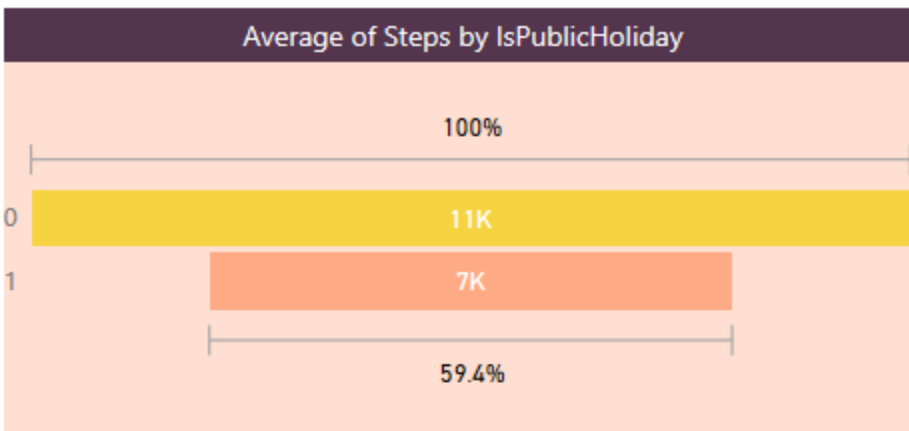
We are using the date dimension here, and one of the benefits of having date dimension is that I can analyze the data based on weekdays, and public holidays. and see which day of week usually I perform best in walking! Here we go;



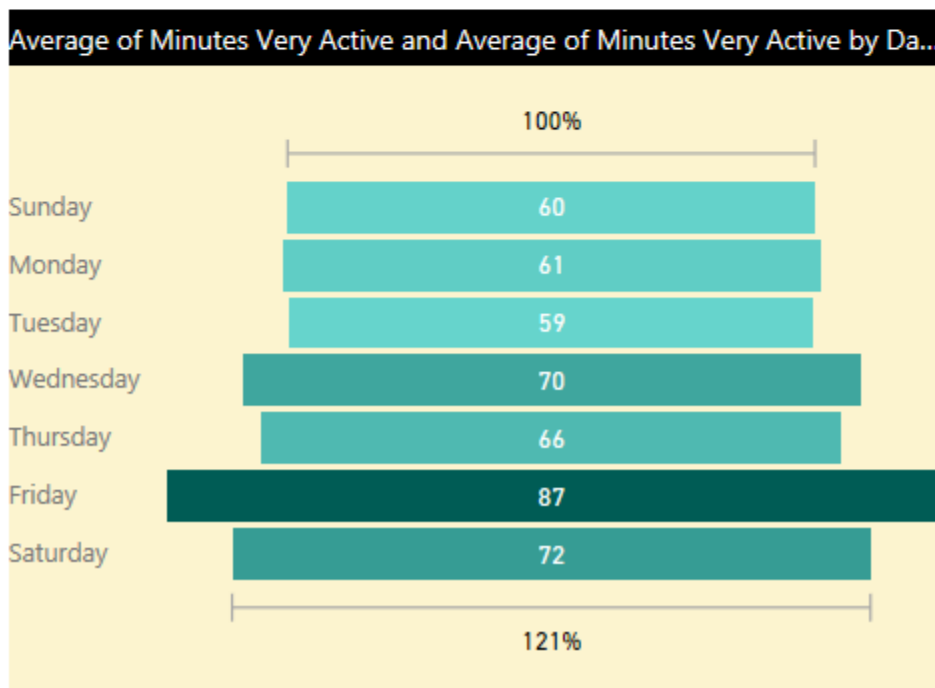
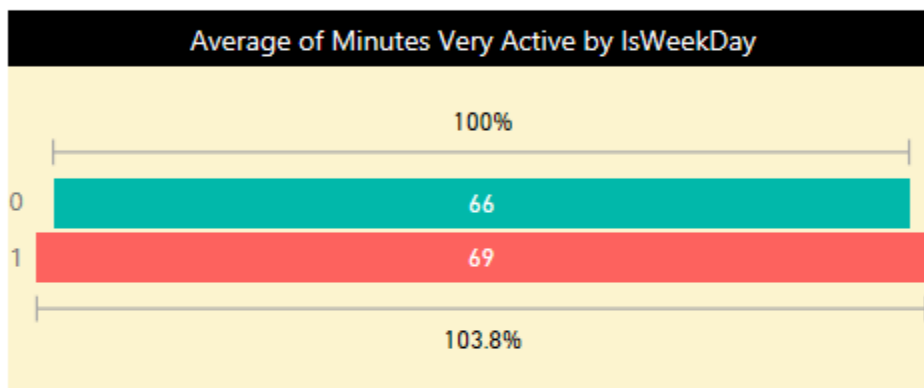
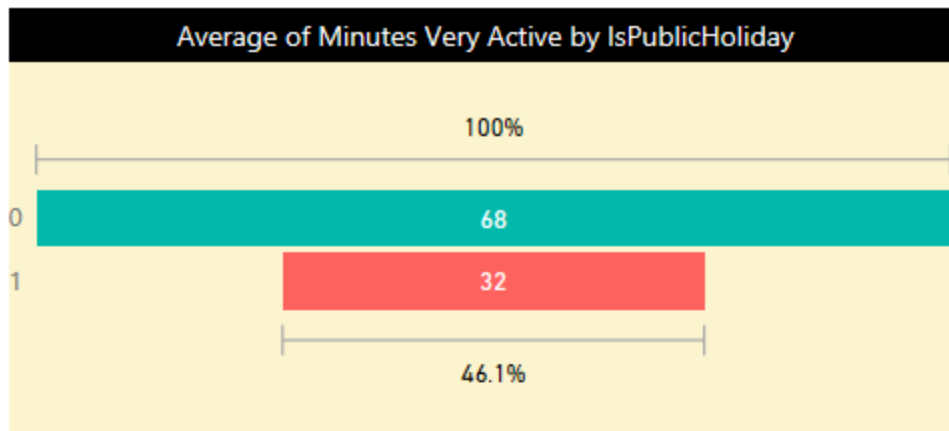
Interesting! Isn't it? I can get more insights out of this visualization than what I see in Fitbit dashboard. I can see that I'm terribly performing bad in taking floors in public holidays! Well our home is flat single floor. However I'm doing a little bit better in weekends. My best days for Floors is always weekdays and there isn't really so much difference between them.



Now if I check my steps; I'm not doing that bad in public holidays, 7K in public holidays in comparison with 11K is other days is fine for me 😊 And I'm doing even very close in weekends and weekdays. If I look at each day's average I can see that I'm a bit tired in Sundays and prefer to rest more, however Saturdays are good days for me! so my overall weekend average goes up because of that. and for some unknown reasons Tuesdays I walk less! That's something I have to consider really why it happens.



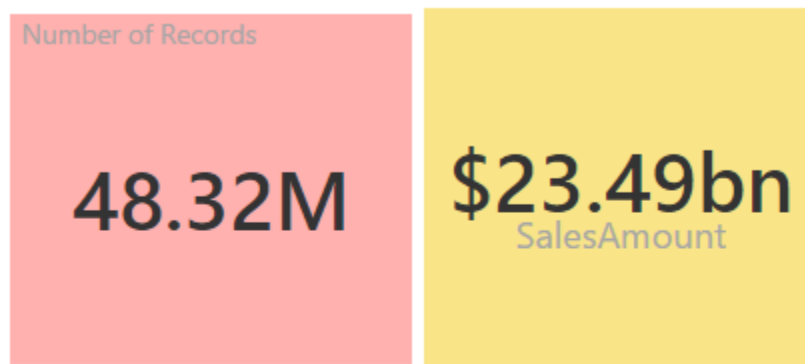
In the spit of what I see in Floors and Steps, I'm still doing a bit closer to my steps results in my very active minutes average. and my best active minutes comes Fridays. There is an obvious reason for it. Fridays I play indoor soccer with some friends, and I really feel my heart is coming out of my chest at some minutes in the game! Here is the evidence; I got most active minutes in Fridays!



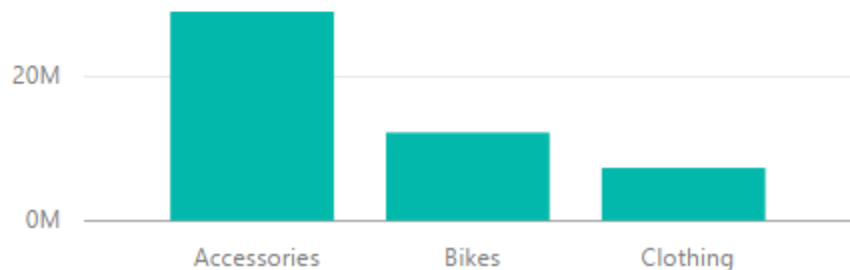
We can go on and on with building more visualizations on this data set. There is no limitation on the amount of insight you can get from it, so I leave it to you to use your own creativity and build dashboards and charts that gives you more insights. You've seen so far how easy is to be a BI developer if you use Power BI 😊

Step Beyond the 10GB Limitation of Power BI

Published Date: April 10, 2016



OrderQuantity by Product Category



Power BI free allows you to have 1GB data per user, and pro 10GB. However in many organizations you have much bigger size of data than 10GB, sometimes you deal with Tera bytes, even Peta bytes of data. Is Power BI limiting you for the visualization? Short answer; No! In this post I'll explain you options that you have when you deal with large data sets. As any other solutions there are pros and cons of working with large data sets with Power BI, Let's explore them together.

Compressed Structure of Power BI

First of all I have to mention that Power BI compress each data set effectively before loading into memory, this is one of the big advantages of xVelocity In-memory engine that Power BI, Power Pivot, and SSAS Tabular built on top of that. Compression happens automatically, that means you don't need to set a configuration, or allow Power BI to do

something. Any data set will be compressed in a reasonable level when you import the data into Power BI.

As an example I've imported a CSV file with ~800 MB data size. When it loaded into Power BI I had only 8MB power BI file to work with. That's an impressive compression.

However compression rate isn't always like that, it depends on type of data set, data type of columns and some other facts. The bottom line in this section is that, if you have a data set that is more than 10GB and won't grow significantly, then consider loading that into Power BI to check how compressed it would be, maybe you get it under 10GB limitation and then you are good to go 😊 If that is not the case then continue reading.

Live Connection

With Power BI we can connect live to some data sources; On premises or in cloud. Well, here's the trick; if your data set is large, then use Live connection! simple but useful trick. You can have a SQL Server, Oracle or whatever data source you want with any size of data you need, and create a live connection to it with Power BI. Live connection won't import data into the model in Power BI. Live connection brings the metadata and data structure into Power BI, and then you can visualize data based on that. with every visualization a query will be sent to the data source and brings the response.

Live Connection to SQL Server On Premises

Let's look at an example of such Live connection to SQL Server on premises database. I have a database with a large table contains 48 million records of data. the table itself is ~800 MB

Table Properties - FactInternetSales

Select a page

- General
- Permissions
- Change Tracking
- Storage
- Extended Properties

Script Help

Compression

Compression type	ColumnStore
------------------	-------------

Filegroups

Filegroup	PRIMARY
Text filegroup	
Table is partitioned	False
FILESTREAM filegroup	

General

Vardecimal storage format is enabled	False
Index space	0.000 MB
Row count	48318400
Data space	837.195 MB

Connection

Server:

.

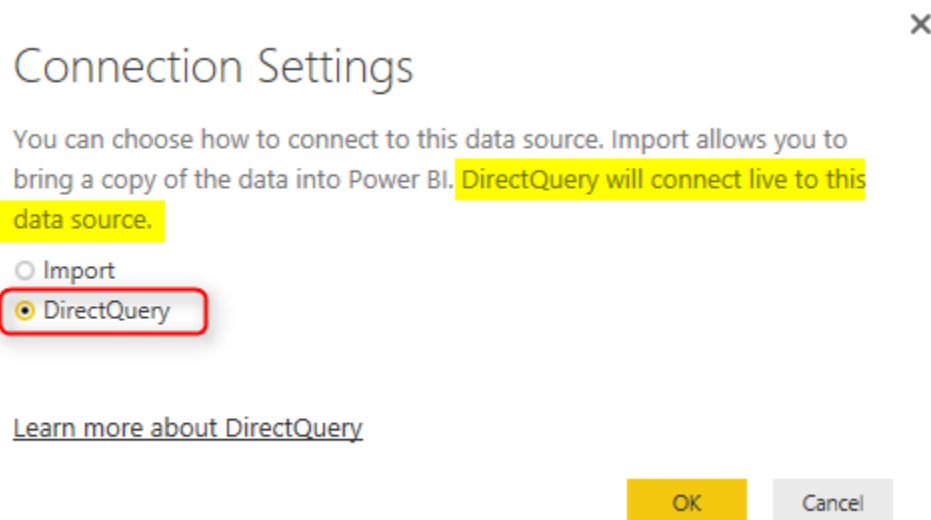
Connection:

REZA-VAIO\Reza

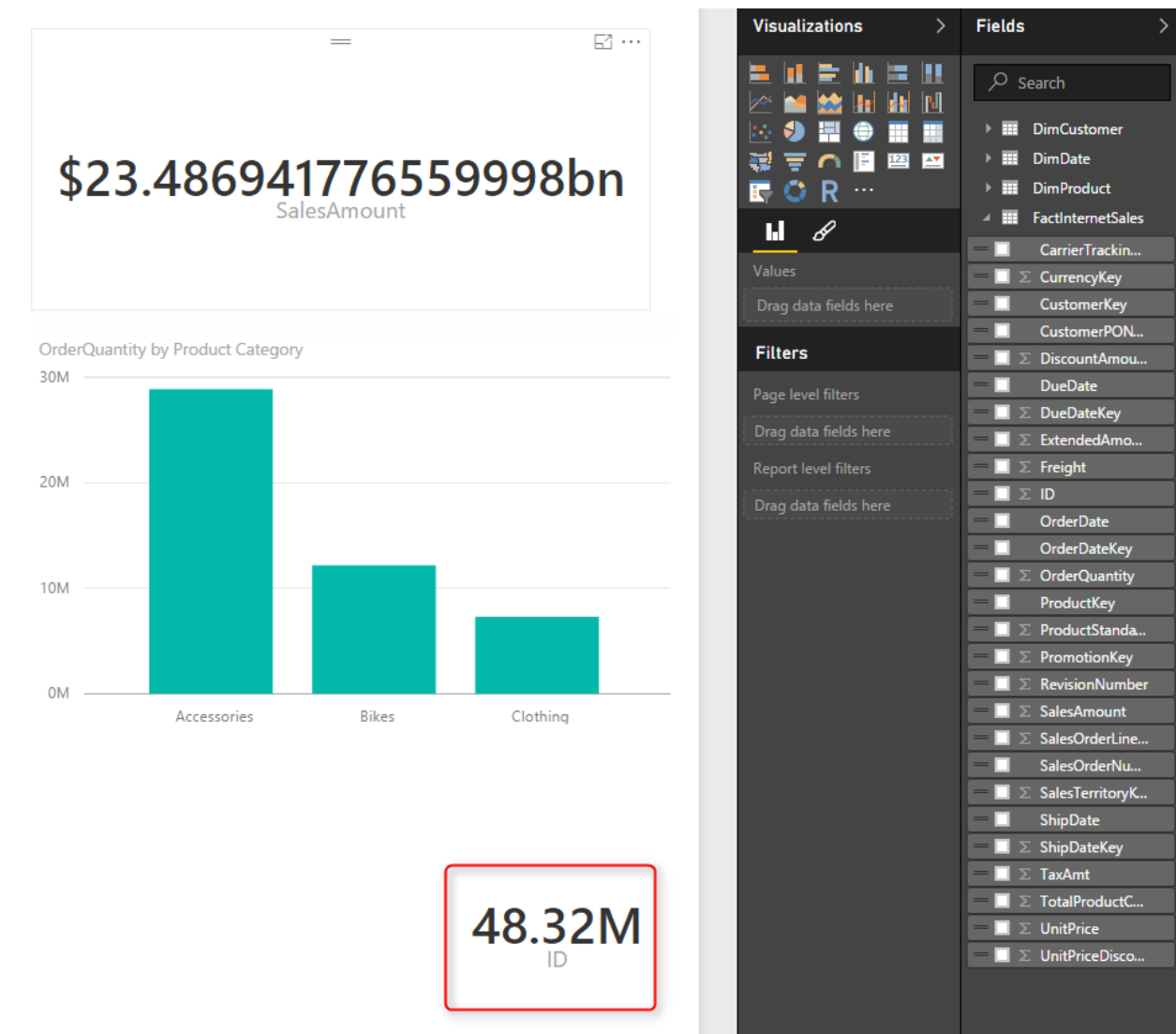
[View connection properties](#)

I know it is not more than 10GB, but the method would be the same even if I have 10TB sized data table. this table is big enough to show how live connection works for this example.

When you go through Get Data and select this table (or any other tables in the data set) and click Load, you'll see the option to choose between Live connection or Import. DirectQuery means Live Connection.



After creating the live Connection you can create relationship in the model, or build visualization for that.



There are however some limitations with Live connection which I'll explore later in this post.

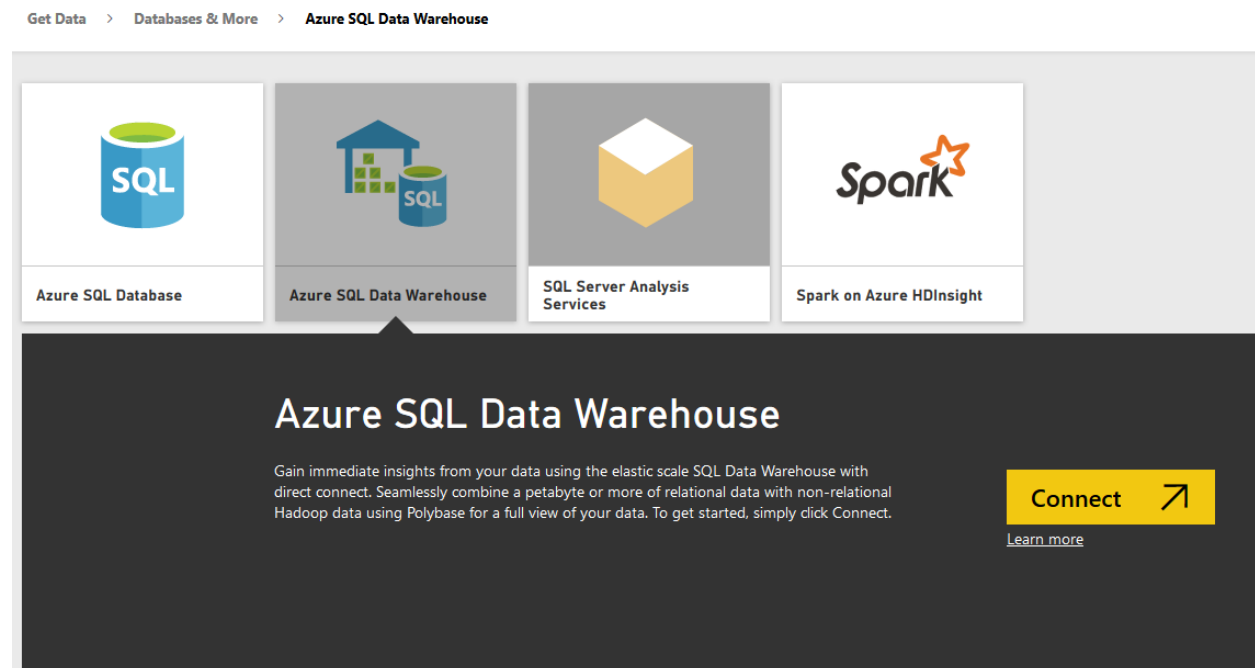
** For the Live connection to on-premises data sources you need to have Power BI Enterprise or [Personal Gateway](#) installed and configured.

You can connect to wide range of on-premises data sources, such as other databases, and also SSAS. as an example you can do:

4. [Live Connection to SSAS Tabular](#)
5. [Live Connection to SSAS Multi-Dimensional](#)

Live Connection to Azure

There are also many Azure data sources that you can use for the live connection Azure SQL Database which is the similar database engine of SQL Server on-premises (with some differences of course) but on cloud. Azure SQL Data Warehouse is the cloud database structure which supports both unstructured and structured data. Azure SQL DW is capable of expanding the compute engine of the database regardless of the storage engine of that. And Sparks on Azure HDInsight is the other azure data source that can be used for live connection to big data structure on azure.



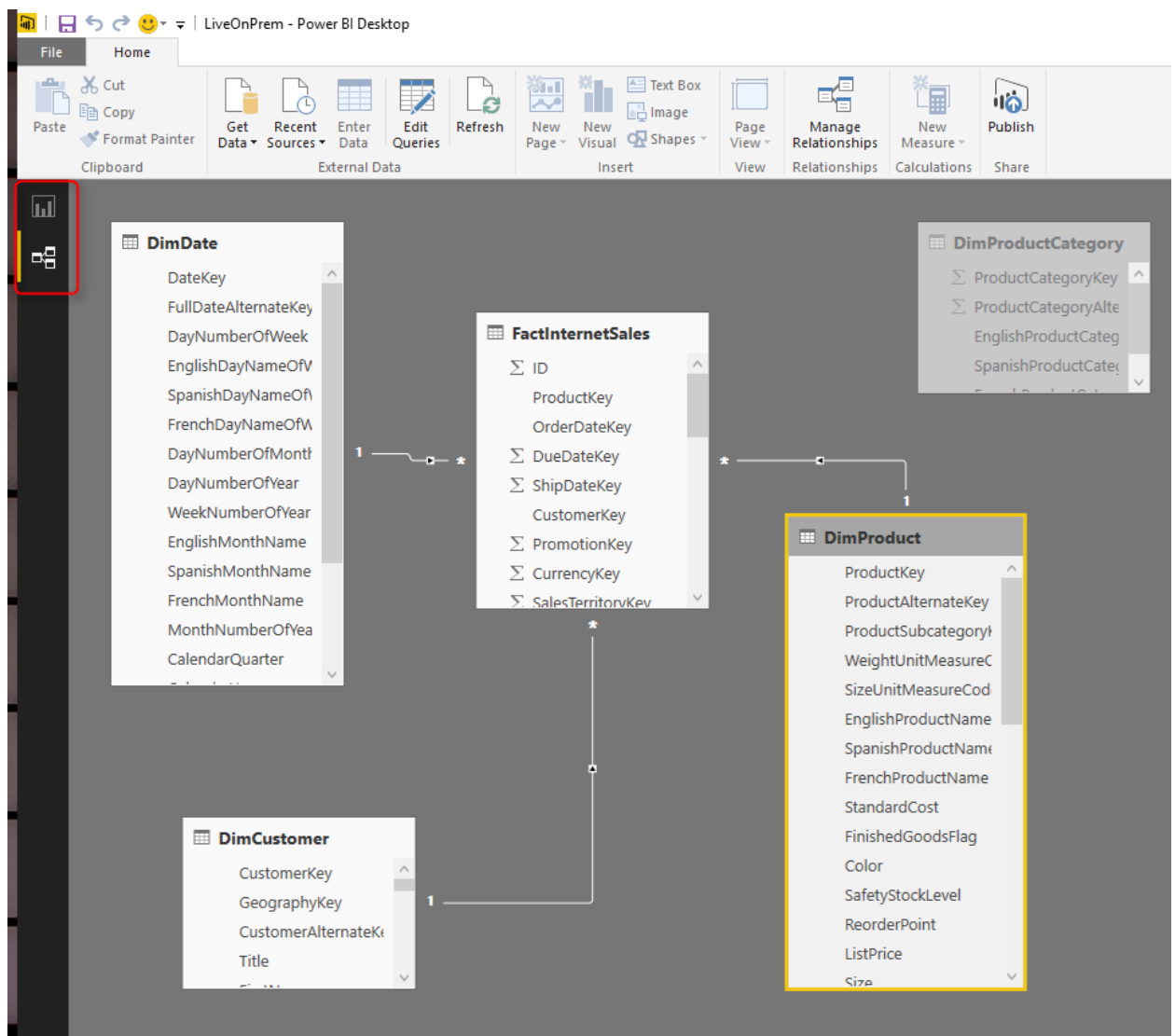
3. [Live Connection to Azure SQL Database](#)
4. Live Connection to Azure SQL Data Warehouse
5. [Live Connection to Spark on Azure HDInsight](#)

Limitations of Live Connection

Live connection is good to connect with large data sets, and won't load the data into the model, so Power BI solution would be very small. However there are some limitations for this type of connection, such as; No DAX support, no data model formatting in the Data tab of Power BI, No multiple data sources support, and No Power Q&A support. Let's go through them one by one;

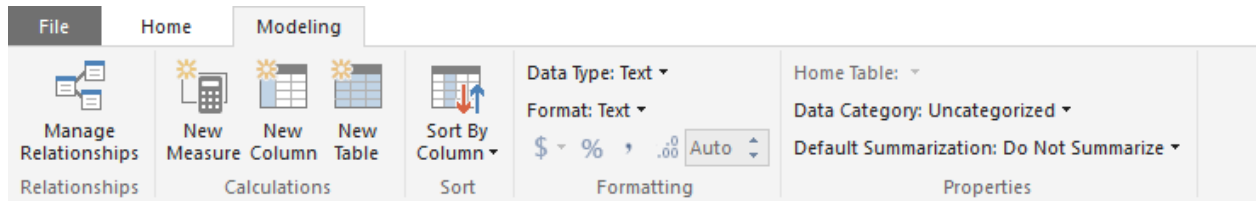
No DAX Support

With Live connection there won't be any Data tab in Power BI to create calculated measures, columns or tables. You have to create all calculations in the data source level. It means that if you have SQL Server as the source, create all calculated columns or measures in SQL Server source tables. Your modelling should be done through the data source. In Power BI desktop you would be able to only set up relationship and that's all in terms of modelling in Power BI.



No Data Tab, No DAX in Live Connection from Power BI

Formatting is not also available through the Modeling tab of Power BI, So if you want to set decimal points, or set a data type of column it is not possible through Power BI. You need to handle all of these through the data source (or Power Query, read further down).



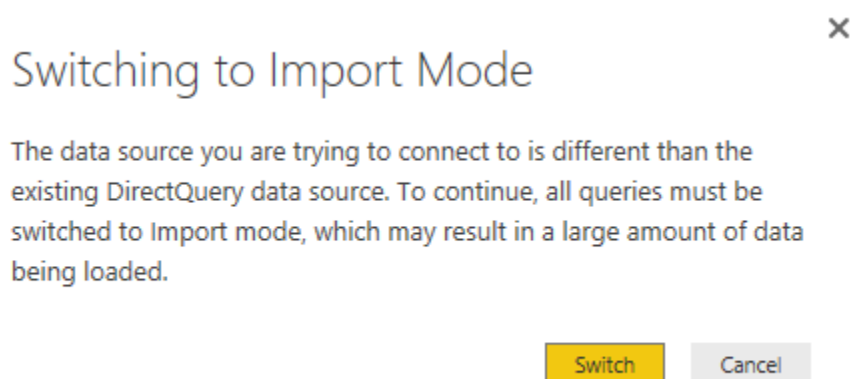
No Data Modeling tab with Live Connection in Power BI

Full Visualization Support

Fortunately visualization part of Power BI is fully supported in Live connection mode. The underlying reason is that visualization is a separate engine in Power BI, and this is one of awesome reasons that a product built on top of separate components usually works better than a product all in one with no underlying component. You can build any visualization and you won't be limited at all.

Multiple Data Sources is Not Supported

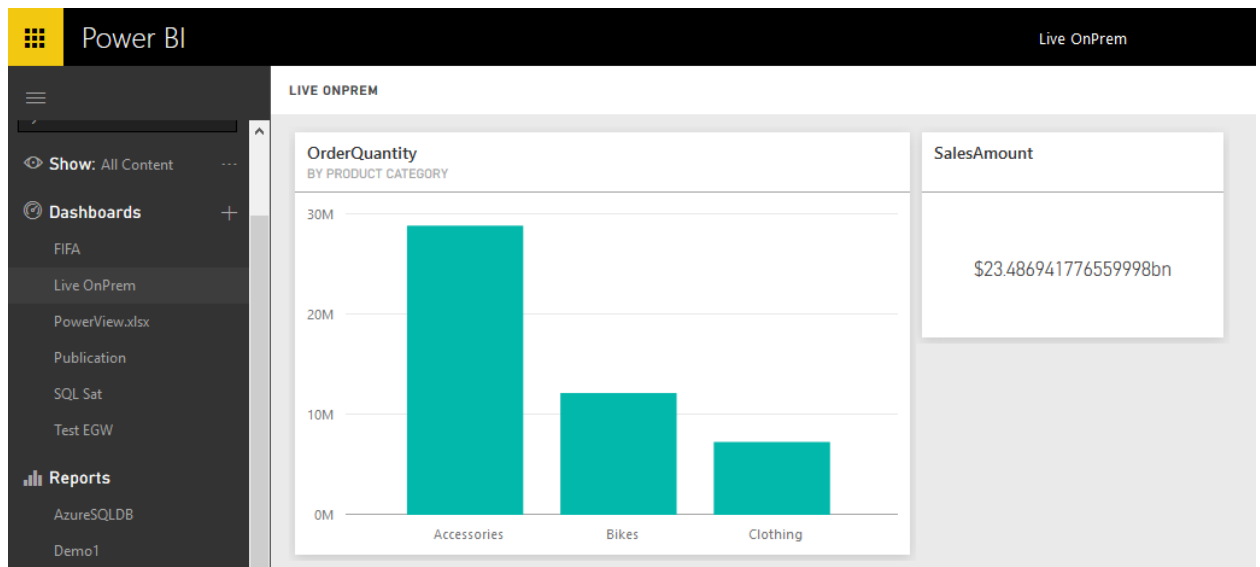
With Power BI one of the great advantages is that you can combine data sets from multiple sources, multiple data bases or files can be participated in building a model. Unfortunately when you work with Live connection you won't be able to have data from more than one data source. Even if you want to get data from another database on the same server you will face messages like below;



So adding another data source requires data to be imported, which is not useful for the scenario of working with large data sets.

No Power Q&A

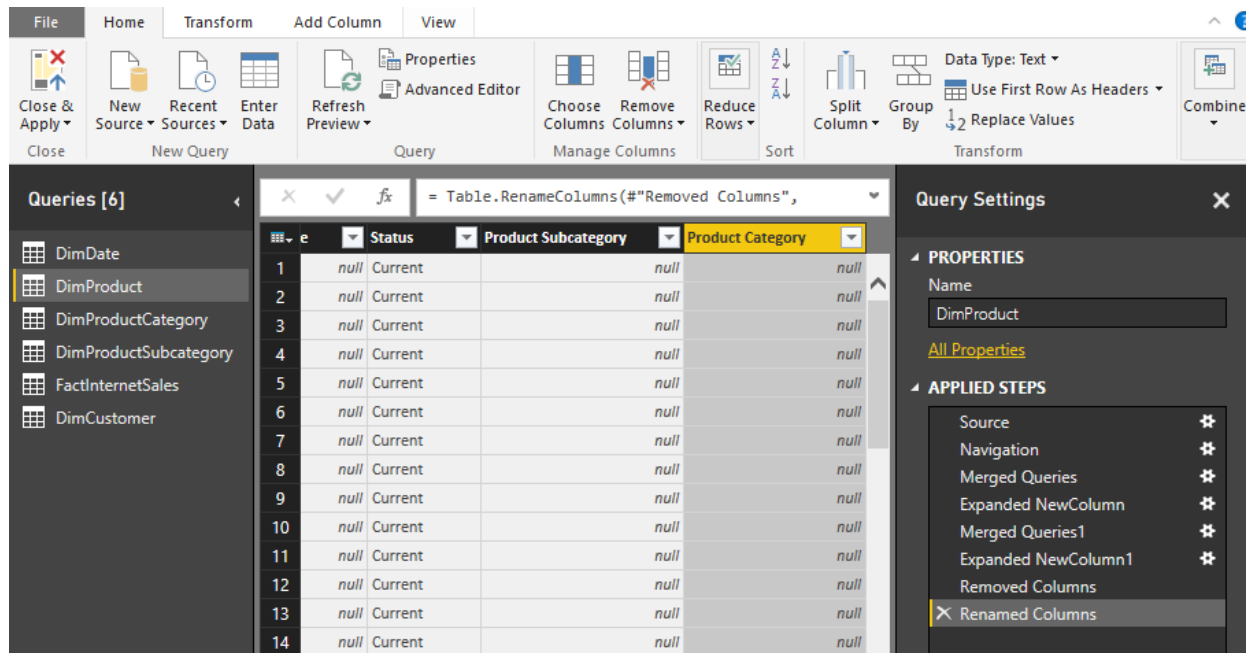
As you know one of the features of Power BI website is natural language questions and answering engine called [Power Q&A](#). This feature is not available (at the time of writing this post) in Power BI Live Connection. when you have a live connection your dashboard won't have Power Q&A question box on the top.



No Power Q&A with Power BI Live Connection

Don't Forget the Power Query

Fortunately Power Query still is available with Live Connection. This gives you ability to join tables, flatten them if you require, apply data transformation and prepare the data as you want. Power Query can also set the data types in a way that be more familiar for the Power BI model to understand. If you want to learn more about Power Query read [Power Query](#) sections of [Power BI online book](#).



The screenshot displays the Power BI Desktop interface. The top ribbon includes tabs for File, Home, Transform, Add Column, and View. The Transform tab is active, showing various data transformation tools. The main area shows a table with the following columns: Status, Product Subcategory, and Product Category. The table contains 14 rows of data, all with 'null' values. The 'Query Settings' pane on the right shows the 'Renamed Columns' step applied.

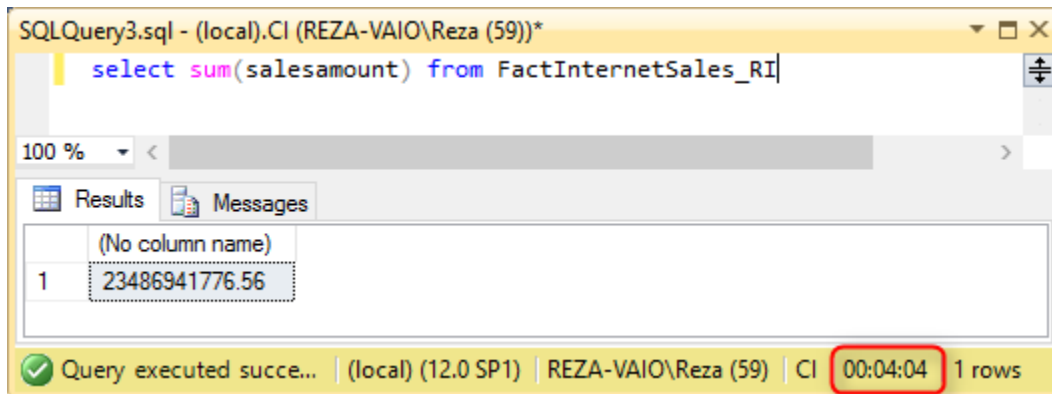
	Status	Product Subcategory	Product Category
1	null	Current	null
2	null	Current	null
3	null	Current	null
4	null	Current	null
5	null	Current	null
6	null	Current	null
7	null	Current	null
8	null	Current	null
9	null	Current	null
10	null	Current	null
11	null	Current	null
12	null	Current	null
13	null	Current	null
14	null	Current	null

In screenshot above you can see that I've joined DimProduct, DimProductSubCategory, and DimProductCategory to bring all fields in single table; DimProduct. And this all happened with Live Connection.

Optimization at Data Source Level

Live connection to data source means report will send queries to data source. Data Sources are different in terms of response time. SSAS tabular might produce faster result, and normal SQL Server database slower. Don't forget that all performance and indexing tips should be considered carefully when you work with Live connection. If you are working for example with SQL Server consider proper indexing, column-store indexes and many other optimization and performance tuning tips.

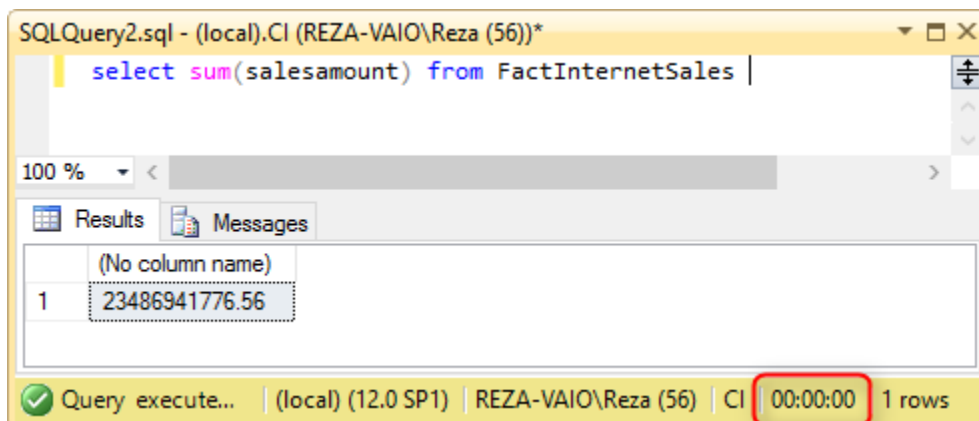
Just for a very small example of performance tuning; here is what performance I get when I have normal index on my table with 48 Million records;



Regular Index

A regular select sum from my table with 48 million records takes 4 minutes and 4 seconds to run, and the same query responds in less than a second when I have a clustered column store index;

and significantly improved performance when I have a Clustered Column Store index on the same table with same amount of data rows;

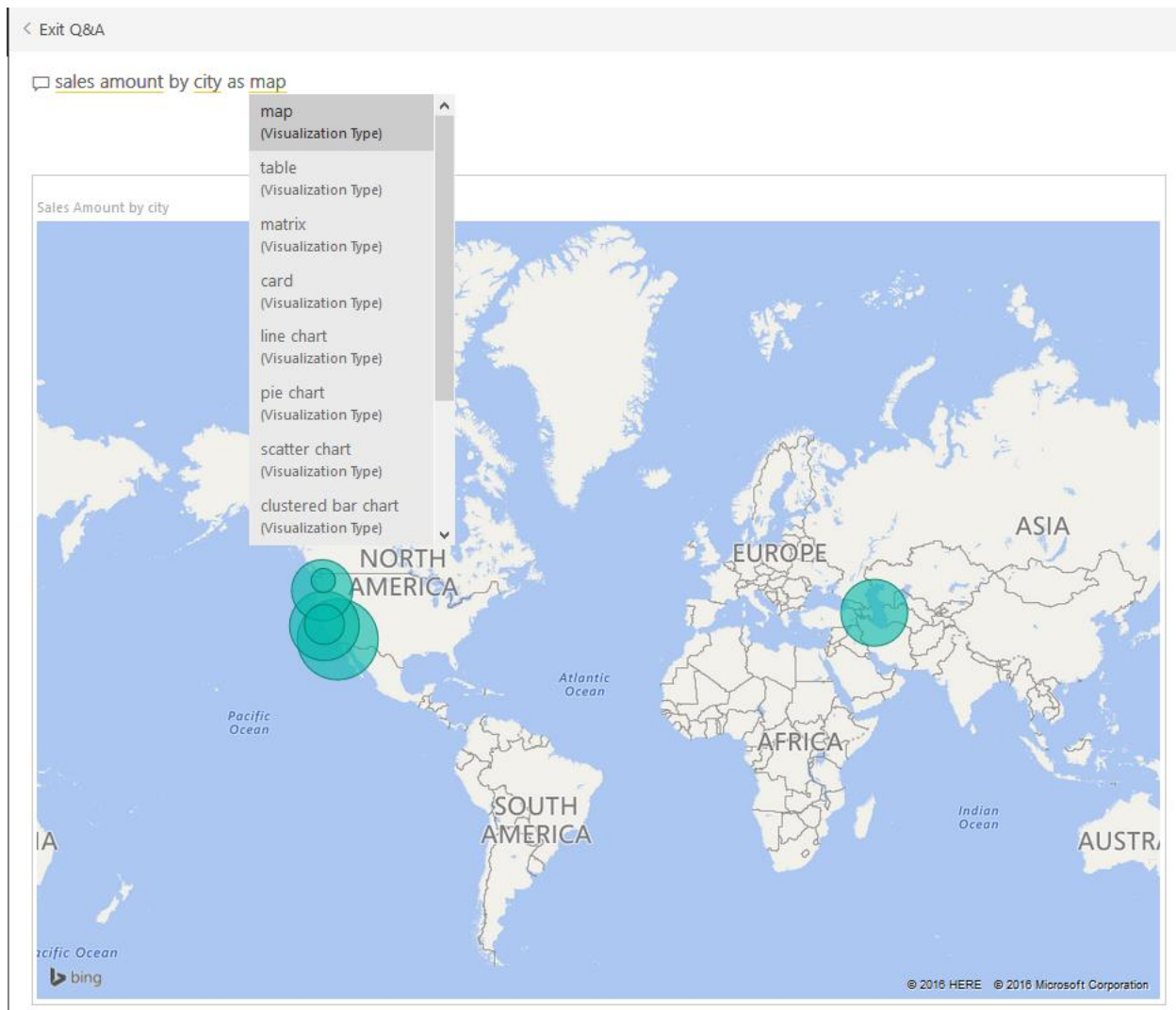


Performance Boost with Clustered Column Store Index

I'm not going to teach you all performance tuning in this post, and I can't do it, because you have to read books, blog posts, watch videos to learn that. That is a whole different topic on its own, and the most important thing is that; Performance tuning is different for each data source. Performance tuning for Oracle, SQL Server, and SSAS are totally different. Your friend for this part would be Google, and vast amount of free content available on the internet for you to study.

Dashboard vs Report; Differences At a Glance - Power BI

Published Date: October 10, 2016



This question is one of the main questions that people ask when they are at the preliminary stages of building a Power BI solutions. There is a misconception that Report is a detailed tabular report, and dashboard is interactive visualizations with chart. However this definition isn't what Power BI reports and dashboards stands for. There are differences in these two main components of a Power BI solution, Understanding differences will help you to leverage their power in a best way. In this post I'll explain what are differences between these two and Where, When, Why, Which is best to use? If

you are interested to learn more about Power BI; read [Power BI online book from Rookie to Rock Star](#).

Definition

Dashboard: General

[Stephen Few](#)'s definition of Dashboard: *A dashboard is a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance.*

Report: General

A Report on the other hand is any informational work. This information can be at any format. Table, Chart, text, number or anything else.

Power BI Report

Power BI Report is combination of multiple visual elements (charts, texts, values...) on a page that can be inter-related with each other. Data visualized in the report can be sliced and diced with slicers. Power BI report is fully interactive from user. and It can be filtered based on some criteria.

Power BI Dashboard

Power BI Dashboard is a high level view of some of key KPIs of one or more reports. Dashboard is a day-to-day view of KPIs, and provide the navigation point to the detailed reports. Power BI Dashboard isn't built for slicing and dicing.

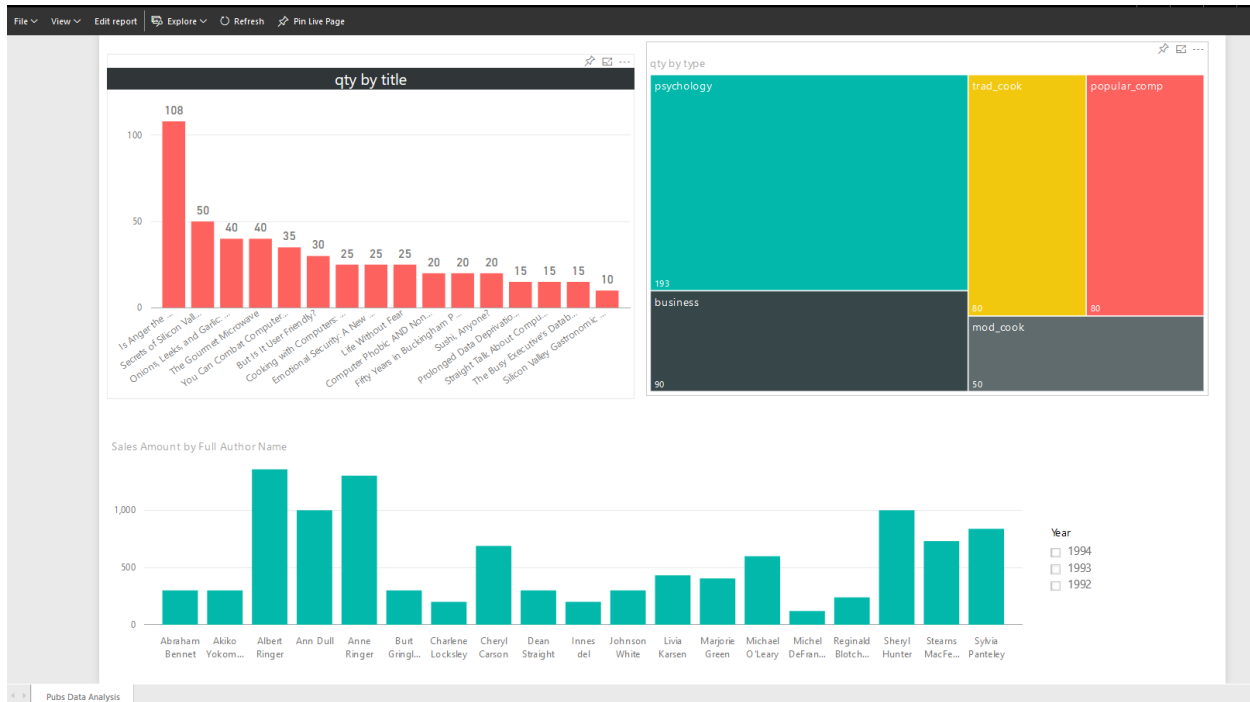
You can see that definition of Power BI Dashboard and Report fits into the general definition of these two component we've had earlier. Now let's look closer at these two.

Beyond the Definition

Power BI Report

Reports in Power BI can have multiple pages. In each page there might be multiple visualization elements. Slicing and dicing, hovering and highlighting is possible in the Report. We can drill down through a hierarchical data structure, or select a particular column in a column chart and see the related data to it in other visualization elements. All of these means Report in Power BI is for slicing and dicing. Report built for end users to play with different handles (slicers or visuals or filters), and achieve what they want.

For example they might want to understand why Sales Amount in month August is lower than other month? or Which product is selling best, and how it is distributed through branches. Here is an example of Power BI report;



If you like to know how to create this report, please read [this](#) post.

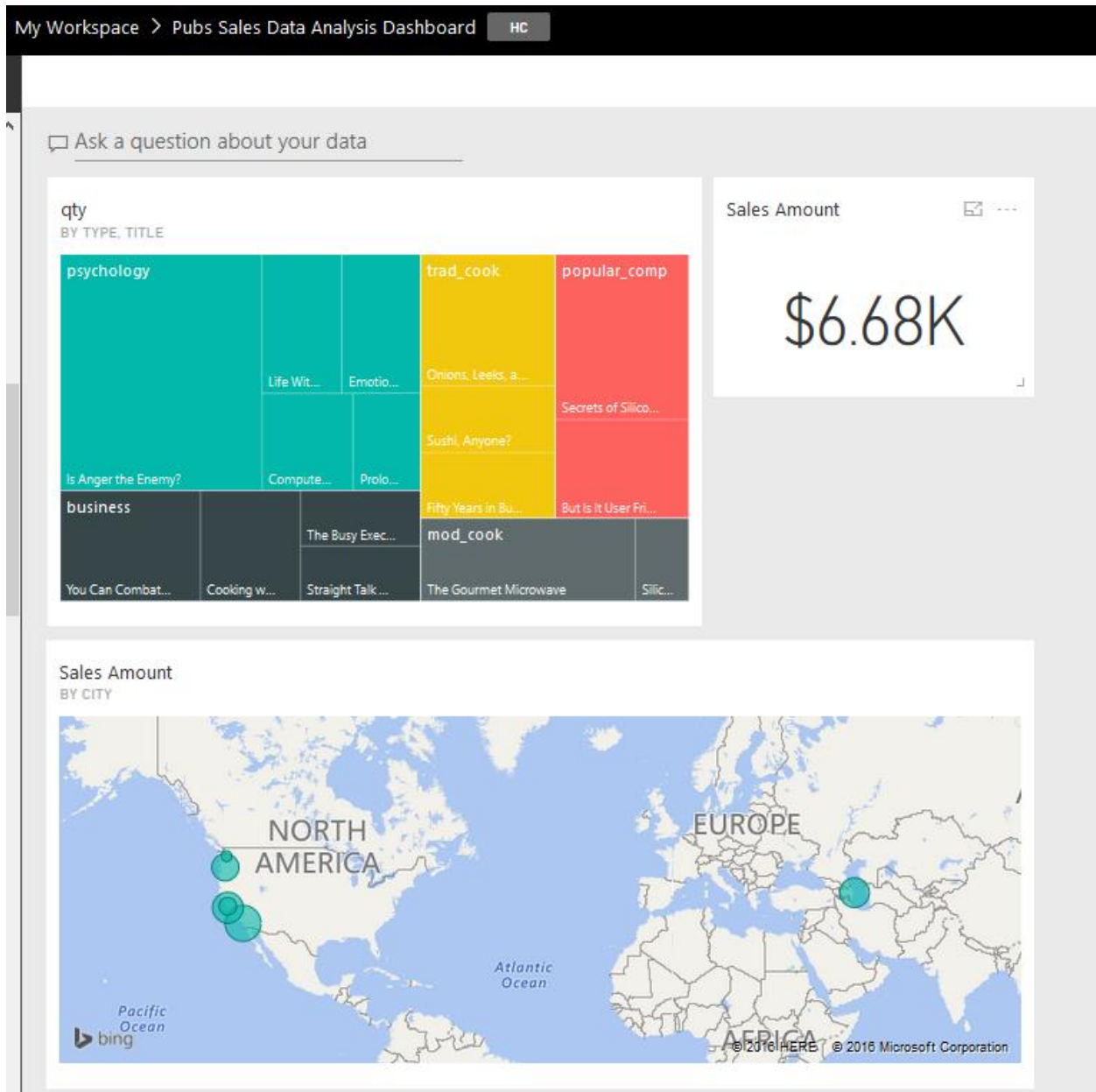
Unique Features of a Power BI Report

28. Slicers
29. Multi-Pages
30. Interactivity
31. Drill Down/Up
32. Publish to Web
33. Explore Data

Power BI Dashboard

A Dashboard in Power BI is a navigation point to reports, and a very high level day-to-day view of main KPIs of business. For example for a particular business need some KPIs might be required from multiple reports. For example year to date revenue from sales report, stock in hand from inventory report, and something from production report. With a Dashboard visualization elements from multiple reports and pages can be *pinned* to one main place. This place then will work as a navigation point. with clicking on each

of these visuals user will be redirected to the report and page that has this element.
Here is an example of a dashboard;



Unique Features of a Power BI Dashboard

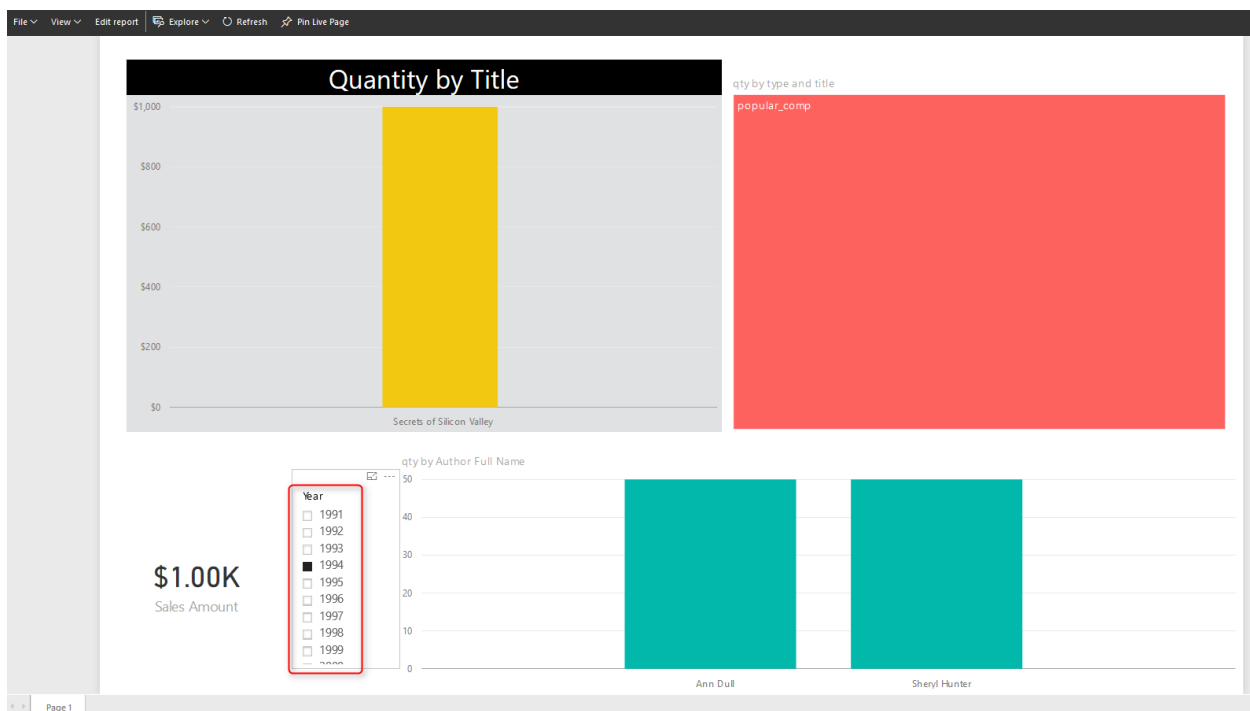
- Automatic Refresh
- Sharing
- Customization by User is Easy
- Featured Dashboard
- Power Q&A

- Real-time Monitoring
- Alerts
- Related Insights

Unique Features of a Power BI Report

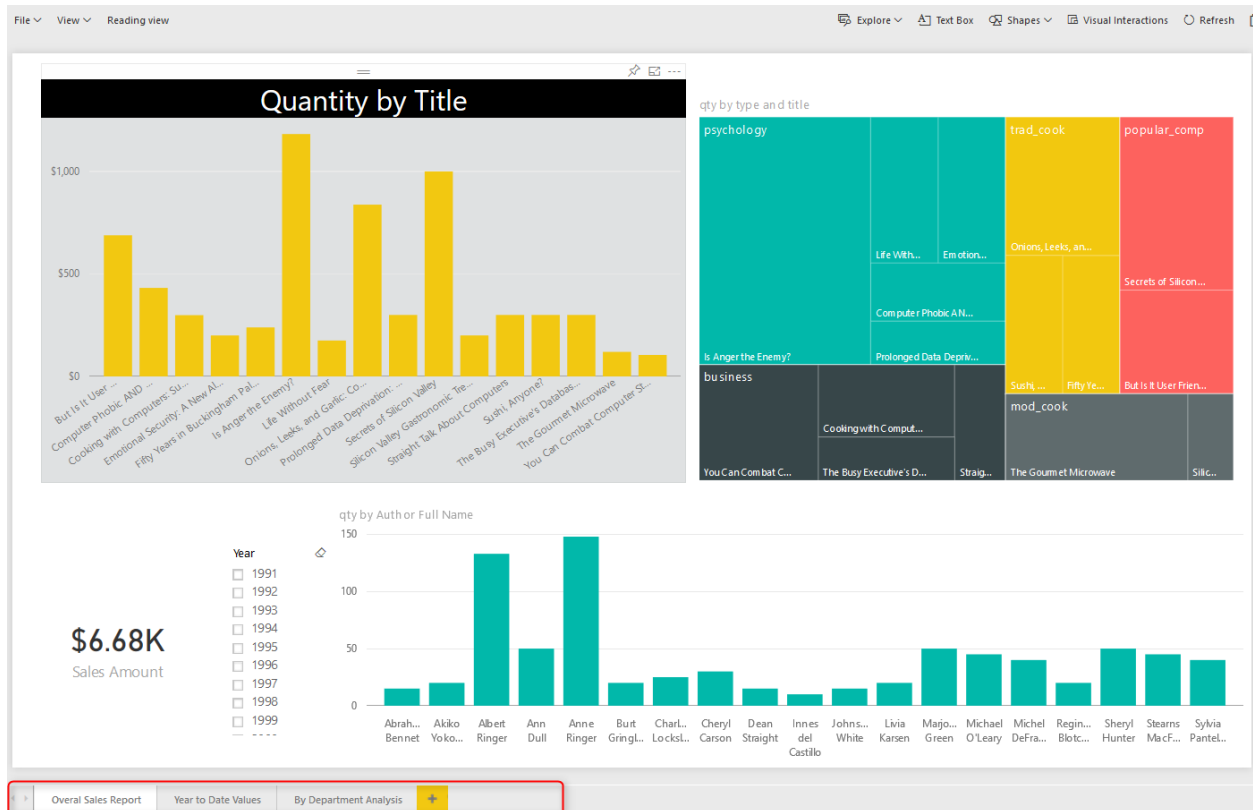
Slicers

You can simply slice and dice the data in the report with slicers. This is unique to reports, In dashboard there is no way of using slicers (Except using "Pin Live Page" option which will pin the whole page in the dashboard)



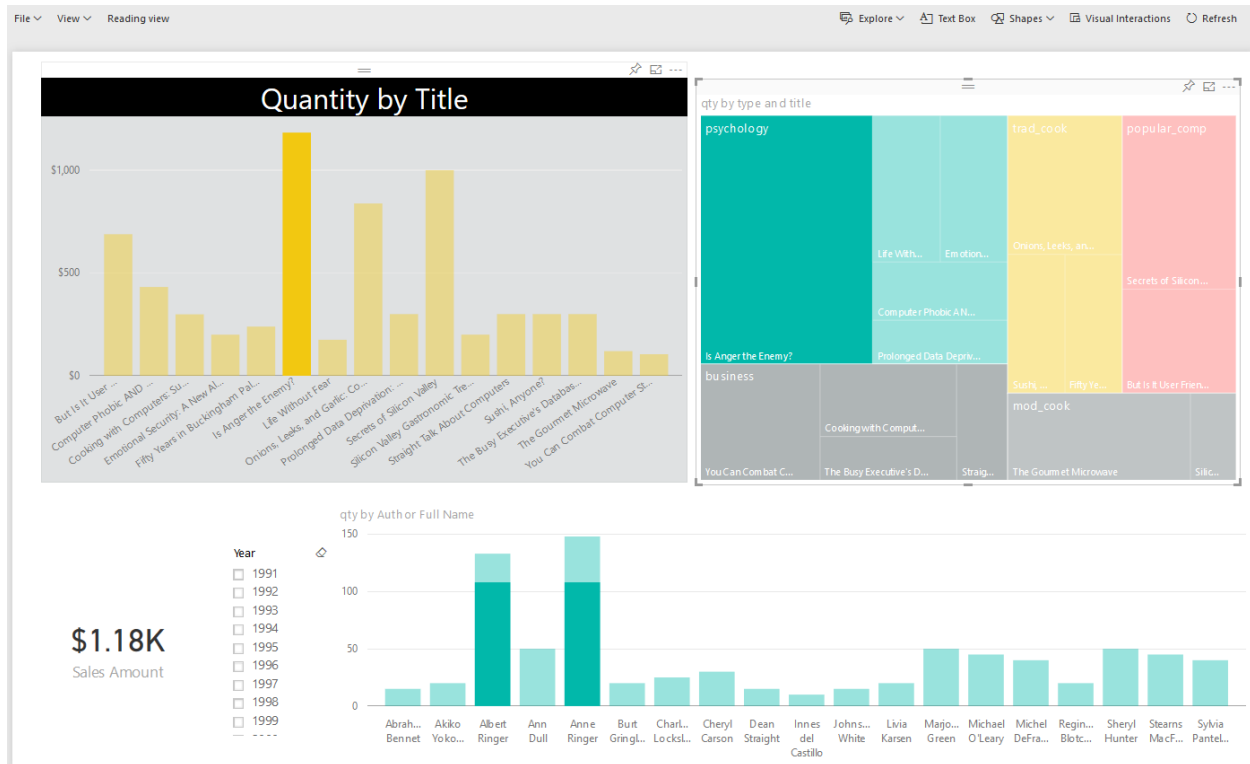
Multi-Pages

You can have multiple pages in a report. Navigation between pages is simply possible through the navigation pane at the bottom of report. In dashboard you can have as many as tiles you want. But there is no concept of pages. everything is in one page, If content doesn't fit into one page scroll bars will appear (Which is not recommended)



Interactivity

In addition to slicers, you can select a particular element in a chart and it will highlight other elements. User can simply interact with report elements and get more insight from it. In dashboard with clicking on a tile you will be redirected to the report/page that built the element.



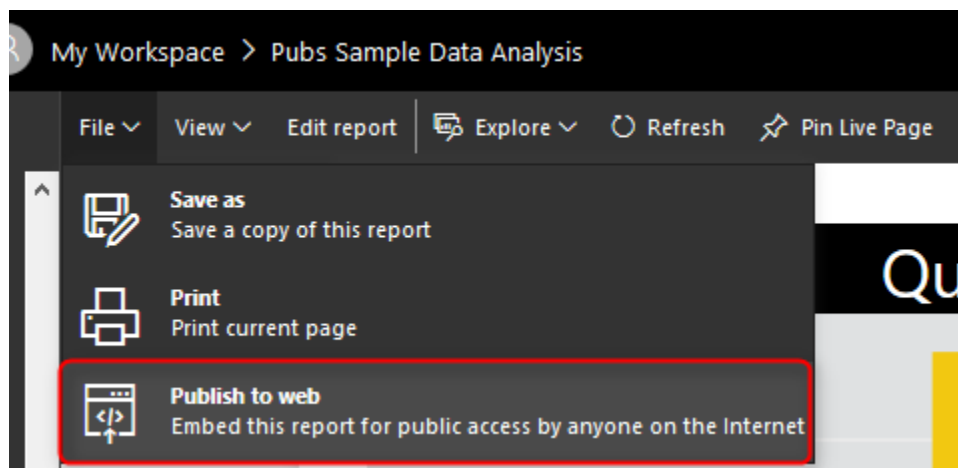
Drill Down/Up

In Power BI you can have hierarchies; such as Product Group, or Calendar. And you can drill down or up in different levels of hierarchy through some of visualization elements. However in Dashboard drill down/up is not possible, and with a click on the visual you will be redirected to the report/page that built the element.



Publish to Web

Publish to Web is a specific feature for a report. With Publish to web report can be published as a public web page, or can be embedded in a publicly available web page. Dashboards at the time of writing this post, cannot be published publicly on the web. To read more about Publish to Web, read [this blog post](#).



Explore Data

Users can explore the data of a particular visual element in the report with options in Explore Data or See Records. These options will help users to drill even down to the record level and see roots of a particular value in a chart. In dashboard only the data of dashboard (not data records) can be exported.

< Back to Report QUANTITY BY TITLE

title	Sales Amount ▼	stor_id	ord_num	payterms	title_id
Is Anger the Enemy?	\$821.25	7066	QA744...	ON invoice	PS20...
Is Anger the Enemy?	\$219.00	7131	N914008	Net 30	PS20...
Is Anger the Enemy?	\$109.50	7067	D4482	Net 60	PS20...
Is Anger the Enemy?	\$32.85	6380	722a	Net 60	PS20...

Unique Features of a Power BI Dashboard

Automatic Refresh

Automatic refresh is one of the main benefits of dashboards vs reports. A dashboard can be designed in this way that it be open for many hours and it will refresh automatically (depends on elements explained later). On the other hand if you open a report, and if data set of that report get refreshed, you need to manually refresh your report otherwise your report won't be refreshed automatically.

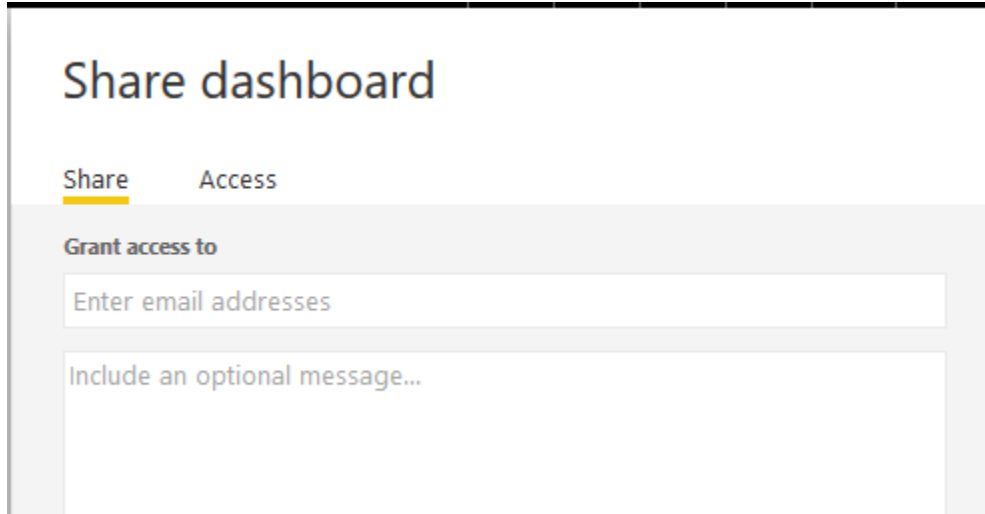
Well when I get to this point, many students in my courses asks why this behaves like that?! Why it is not getting refreshed automatically like the dashboard. I believe the main reason for this is scenarios like this: Consider that an analyst has opened an inventory report, and is checking number of that with a static report that he/she has in Excel or even on paper. His/Her excel or paper report is static, and he/she is doing a sanity check to see if numbers match or not. if the report be dynamic then nothing can be checked. There might be other reasons as well, but this is I believe the main reason for it.

Power BI Dashboards refreshes whenever the data set refreshes. This is for when we import the data into Power BI. If the data set is scheduled to refresh, or if we refresh that manually, the dashboard will be refreshed automatically.

Power BI Dashboards for DirectQuery data sets refreshes every 15 minutes.

Sharing

Dashboards, Reports, and Data Sets in Power BI can be shared through different methods; such as Content Packs, and [Power BI Work Groups](#). However Dashboards can be shared through a basic sharing method as well. This is an addition to the other methods of sharing. With this method person/people that this dashboard will be shared with them will receive an email with the dashboard link.

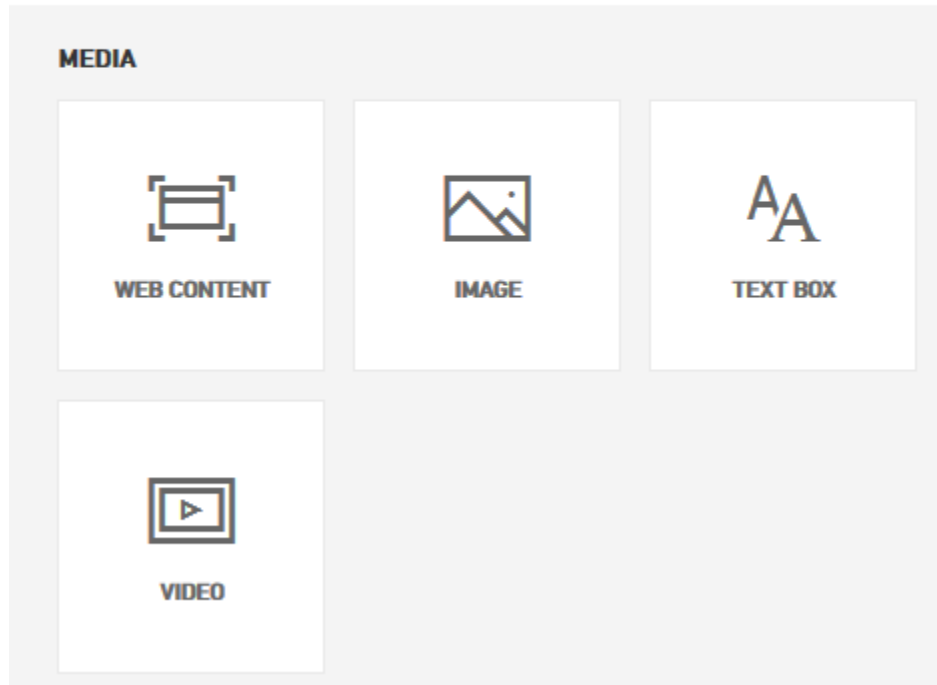


Customization by User is Easy

It is easy to change size of dashboard tiles, and change order of them, also to add new tiles (Image, Text, Web, and Video) in dashboard. That said it is also possible to make changes to a report as well. But that required clicking on Edit report, going to edit mode, applying changes, and saving. Dashboard is like user's view of the world, so simply can be adjusted to what he/she wants to see.

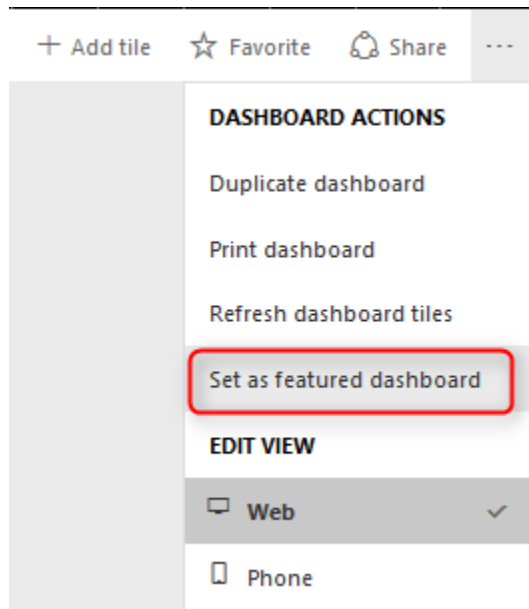
Add tile

Select source



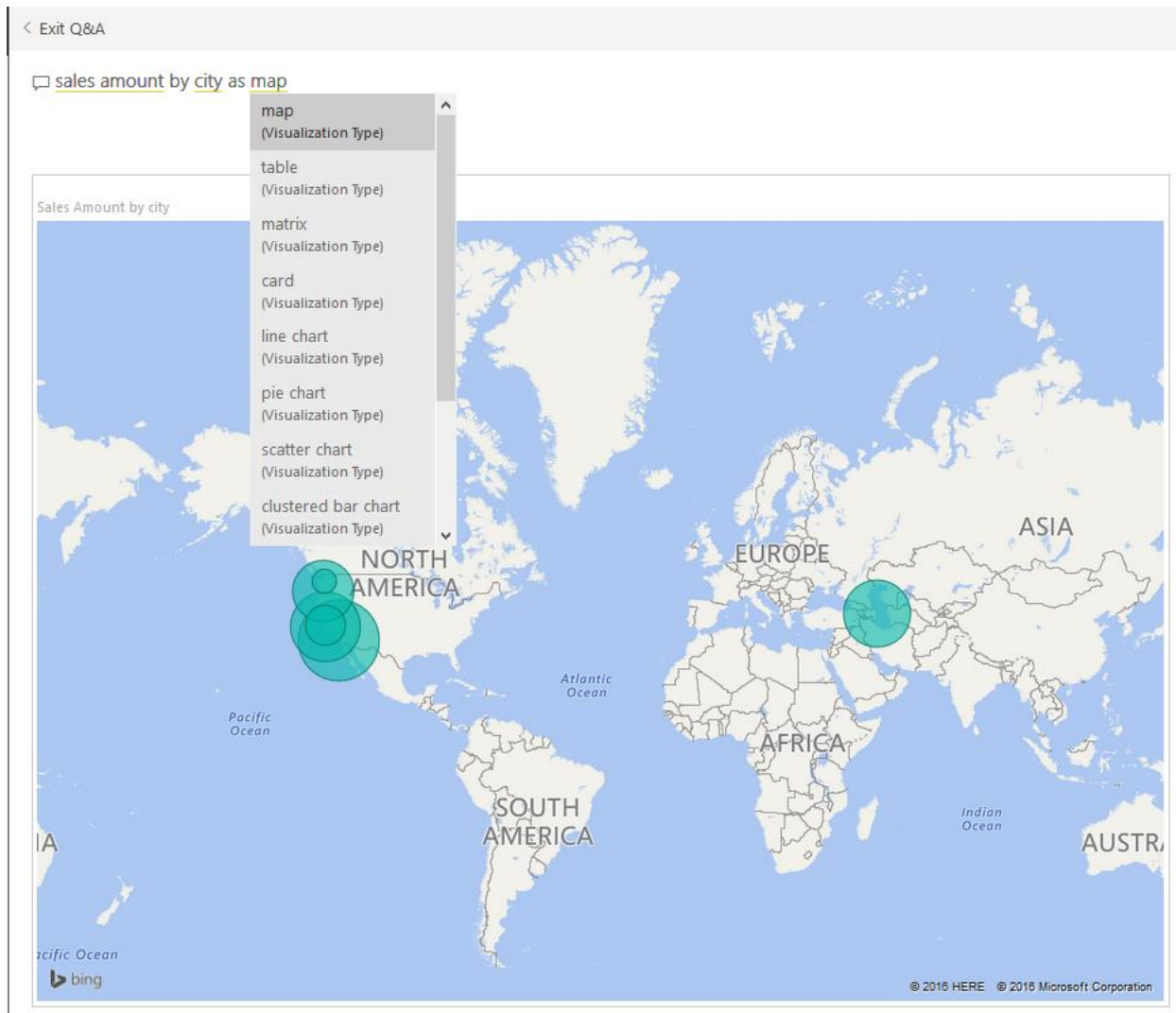
Featured Dashboard

A dashboard can be set as a landing page with setting that as Featured Dashboard. There can be only one featured Dashboard. If you set another dashboard to be featured, the previous featured dashboard will be replaced with this new one.



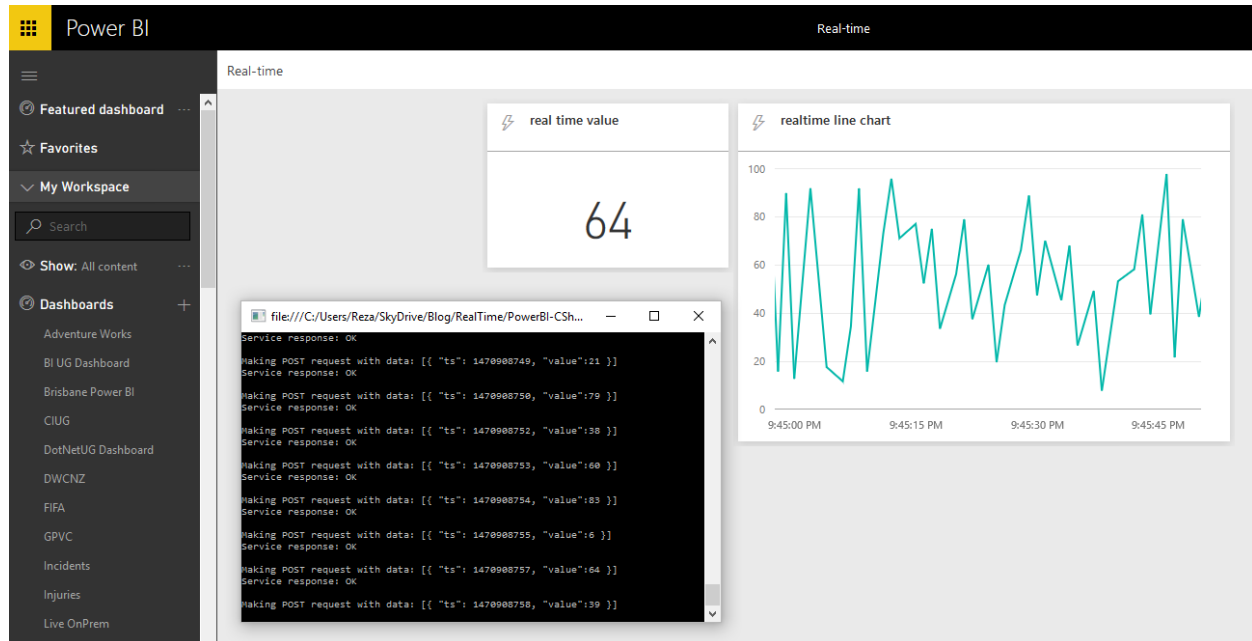
Power Q&A

If you haven't heard about Q&A I encourage you to read [this blog post](#). Q&A is an engine of top of Power BI model which will respond to your natural English language questions. Q&A box is only available on top of dashboards (Not all types of dashboards, at the time of writing this post, Q&A on dashboards that has Row Level Security enabled, or data sets used DirectQuery is not possible).



Real-time Monitoring

I've mentioned that dashboards refreshes automatically. However this doesn't mean real-time monitoring. Real-time monitoring means using PUSH approach. That means anytime a new data in the source is available it should be monitored. It is possible to have real-time monitoring of events with Power BI Dashboards. The data set at the moment can be sourced from PubNub, Azure Stream Analytics, or REST API. To learn more about real-time monitoring with REST API [read this post](#), and Azure stream Analytics [read this post](#).




Alerts



You can define alerts for each data driven tile in the report. Alerts can be as simple as if number goes above or below something send an email to me.

SALES AMOUNT

...

Manage alerts

 Add alert rule

 Alert for Sales Amount 

Active

☒ On


Alert title

Alert for Sales Amount

Set alerts rule for

Sales Amount

Condition

Above 

Threshold

6677

Maximum notification frequency

☒ At most every 24 hours

☐ At most once an hour

Alerts are only sent if your data changes.

By default, you'll receive notifications on the service in the notification center.

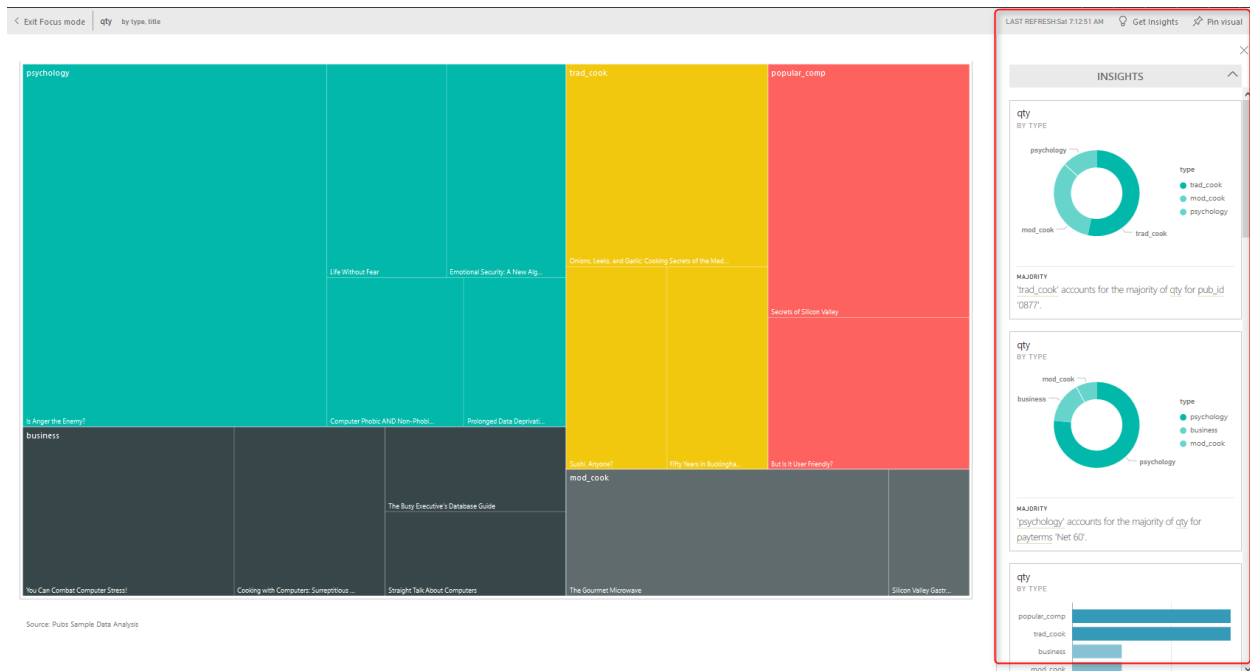
☒ Send me email, too

Save and close

Cancel

Related Insights

In dashboards you can get some more insight with selecting the Related Insights feature of the tile. This option will search through patterns in the data set and visualize them automatically.

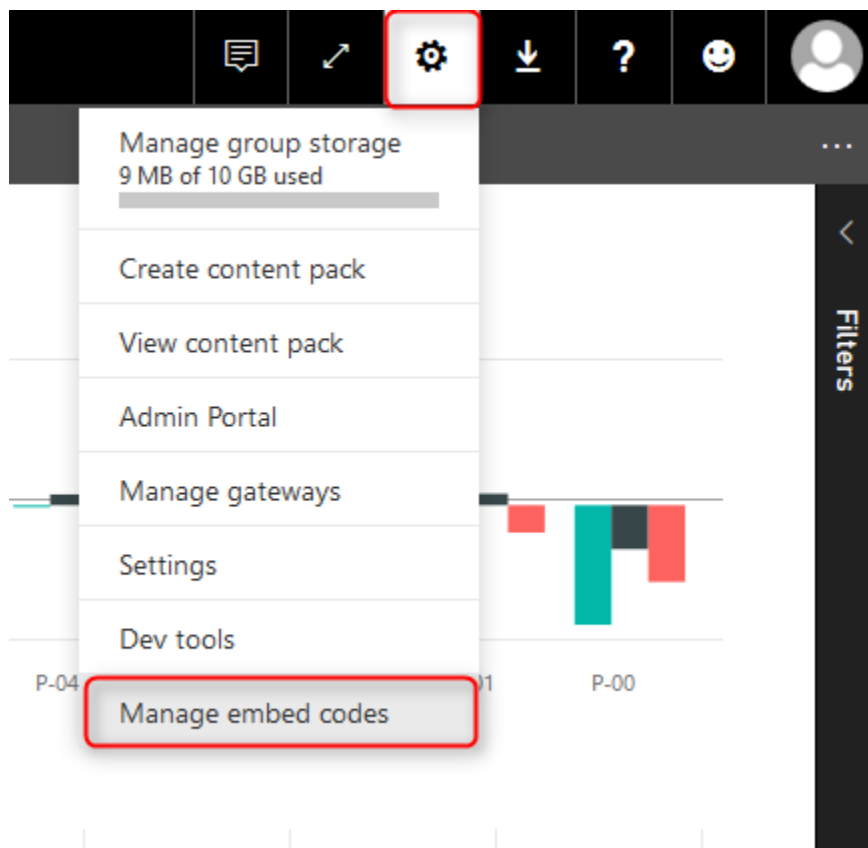


Differences At a Glance

As you see dashboards and reports are not exactly the same. there are many major differences between these two. majority of the difference is that; Dashboard is day-to-day single page view of main KPIs, which can be refreshed automatically and can visualize real-time events. Dashboard can be used to navigate to reports. Reports are interactive data visualization elements that can be used by users to slice and dice, highlight and interact with to investigate numbers and insight more in details. For a great visualization solution with Power BI you would need both, these are complements of each other, not replacements.

Power BI Publish to Web - Questions Answered

Published Date: July 22, 2016



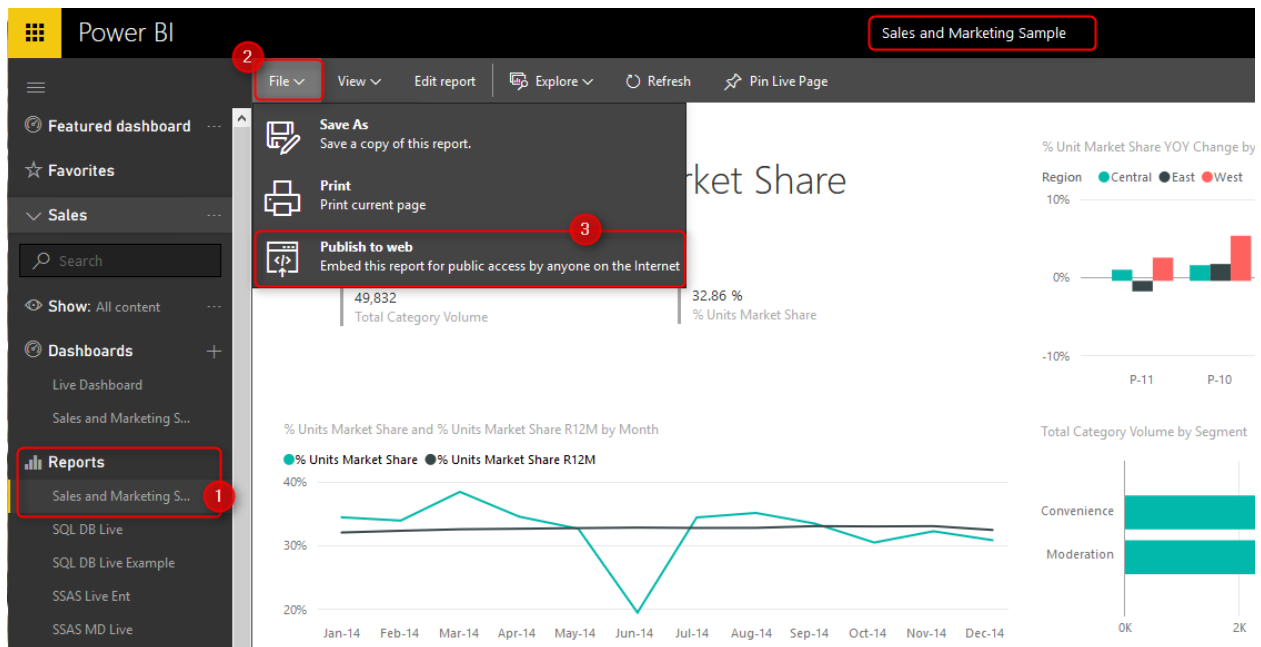
Publish to Web feature of Power BI is not a very new or hot topic, However I still get questions about what this feature is, and how it is difference with Power BI Embedded. It would be a very long discussion if I want to both explain these two features, and compare them here. So in this post I'll explain Publish to Web, and in next posts I'll explain about Power BI embedded, and their differences. In this post you will learn how easy is to share your report with public through a web page which can be your blog post, and HTML page, or any other web pages. Some questions also about this feature answered through the content of this post. If you like to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

What is Publish to Web?

Once you published your Power BI report into Power BI Service (or website), then you can share it with others through creating a dashboard, or [groups in Power BI](#). What if you want to share it with public through the web? Let's say you want everyone to see the report and play with it (with all interactive features that you have in Power BI). The answer is; use Publish to Web. Publish to web allows you to create an embedded code for Power BI report, and use that code in a web page. This simple feature will enable everyone to access the report. They won't be able to edit the report, but they will see the report and the report will be fully interactive for them so they can highlight items, select slicers, and drill down.

How to Do It?

Using this feature is very simple. All you need to do is first deploy or publish the report into Power BI website or service. and there click on the report (not dashboard). Once you opened the report click on File menu option and choose Publish to Web.



Then you will be informed about this feature in a message box that mentions this step will create a link and embed code for you to share with the world through a website or even email. Click on Create embed code below.

Embed in a public website



Get a link or embed code that you can include on a website, or send in email.

Publish a live version that will remain synchronized with the source report in Power BI. Any changes you make to the report will immediately be reflected in this version.

Create embed code

Close

Again because this is all about sharing a report and obviously the data in the report, you will be informed again to check confidentiality of the data and report, and make sure you are sharing the content that is not harmful for an organization or someone when it is view-able for public.

Embed in a public website



You are about to create an embed code for this report. Once published, anyone on the Internet will be able to access the report and the data it contains.

Before publishing this report, ensure you have the right to share the data and visualizations publicly. Do not publish confidential or proprietary information. If in doubt, check your organization's policies before publishing.

Publish

Close

After clicking on Publish above you will see the embed code plus a link to share through email if you want. You can also choose the size of screen for embed code.

Success!



Link you can send in email

<https://app.powerbi.com/view?r=eyJrIjojMWUyNGRjZDAtYTYSOS00YjI4LV>

Html you can paste into your blog or website

```
<iframe width="800" height="600" src="https://app.powerbi.com/view?r=
```

Size

800 x 600 px	▼
680 x 510 px	
800 x 600 px	
933 x 700 px	

Close

app.powerbi.com/view?r=eyJrIjoibWUyNGRjZDA4YTYS00YjI4LWZDctMmU1ZGZkNGFjZmZmlwidCl6ImNjODBmOGQ2LTIiLCJ0Ij06IiwiaWUiOiJ1IiwiaW91Ij06IiwiaW92Ij06IiwiaW93Ij06IiwiaW94Ij06IiwiaW95Ij06IiwiaW96Ij06IiwiaW97Ij06IiwiaW98Ij06IiwiaW99Ij06IiwiaW10Ij06IiwiaW11Ij06IiwiaW12Ij06IiwiaW13Ij06IiwiaW14Ij06IiwiaW15Ij06IiwiaW16Ij06IiwiaW17Ij06IiwiaW18Ij06IiwiaW19Ij06IiwiaW20Ij06IiwiaW21Ij06IiwiaW22Ij06IiwiaW23Ij06IiwiaW24Ij06IiwiaW25Ij06IiwiaW26Ij06IiwiaW27Ij06IiwiaW28Ij06IiwiaW29Ij06IiwiaW30Ij06IiwiaW31Ij06IiwiaW32Ij06IiwiaW33Ij06IiwiaW34Ij06IiwiaW35Ij06IiwiaW36Ij06IiwiaW37Ij06IiwiaW38Ij06IiwiaW39Ij06IiwiaW40Ij06IiwiaW41Ij06IiwiaW42Ij06IiwiaW43Ij06IiwiaW44Ij06IiwiaW45Ij06IiwiaW46Ij06IiwiaW47Ij06IiwiaW48Ij06IiwiaW49Ij06IiwiaW50Ij06IiwiaW51Ij06IiwiaW52Ij06IiwiaW53Ij06IiwiaW54Ij06IiwiaW55Ij06IiwiaW56Ij06IiwiaW57Ij06IiwiaW58Ij06IiwiaW59Ij06IiwiaW60Ij06IiwiaW61Ij06IiwiaW62Ij06IiwiaW63Ij06IiwiaW64Ij06IiwiaW65Ij06IiwiaW66Ij06IiwiaW67Ij06IiwiaW68Ij06IiwiaW69Ij06IiwiaW70Ij06IiwiaW71Ij06IiwiaW72Ij06IiwiaW73Ij06IiwiaW74Ij06IiwiaW75Ij06IiwiaW76Ij06IiwiaW77Ij06IiwiaW78Ij06IiwiaW79Ij06IiwiaW80Ij06IiwiaW81Ij06IiwiaW82Ij06IiwiaW83Ij06IiwiaW84Ij06IiwiaW85Ij06IiwiaW86Ij06IiwiaW87Ij06IiwiaW88Ij06IiwiaW89Ij06IiwiaW90Ij06IiwiaW91Ij06IiwiaW92Ij06IiwiaW93Ij06IiwiaW94Ij06IiwiaW95Ij06IiwiaW96Ij06IiwiaW97Ij06IiwiaW98Ij06IiwiaW99Ij06IiwiaW100Ij06IiwiaW101Ij06IiwiaW102Ij06IiwiaW103Ij06IiwiaW104Ij06IiwiaW105Ij06IiwiaW106Ij06IiwiaW107Ij06IiwiaW108Ij06IiwiaW109Ij06IiwiaW110Ij06IiwiaW111Ij06IiwiaW112Ij06IiwiaW113Ij06IiwiaW114Ij06IiwiaW115Ij06IiwiaW116Ij06IiwiaW117Ij06IiwiaW118Ij06IiwiaW119Ij06IiwiaW120Ij06IiwiaW121Ij06IiwiaW122Ij06IiwiaW123Ij06IiwiaW124Ij06IiwiaW125Ij06IiwiaW126Ij06IiwiaW127Ij06IiwiaW128Ij06IiwiaW129Ij06IiwiaW130Ij06IiwiaW131Ij06IiwiaW132Ij06IiwiaW133Ij06IiwiaW134Ij06IiwiaW135Ij06IiwiaW136Ij06IiwiaW137Ij06IiwiaW138Ij06IiwiaW139Ij06IiwiaW140Ij06IiwiaW141Ij06IiwiaW142Ij06IiwiaW143Ij06IiwiaW144Ij06IiwiaW145Ij06IiwiaW146Ij06IiwiaW147Ij06IiwiaW148Ij06IiwiaW149Ij06IiwiaW150Ij06IiwiaW151Ij06IiwiaW152Ij06IiwiaW153Ij06IiwiaW154Ij06IiwiaW155Ij06IiwiaW156Ij06IiwiaW157Ij06IiwiaW158Ij06IiwiaW159Ij06IiwiaW160Ij06IiwiaW161Ij06IiwiaW162Ij06IiwiaW163Ij06IiwiaW164Ij06IiwiaW165Ij06IiwiaW166Ij06IiwiaW167Ij06IiwiaW168Ij06IiwiaW169Ij06IiwiaW170Ij06IiwiaW171Ij06IiwiaW172Ij06IiwiaW173Ij06IiwiaW174Ij06IiwiaW175Ij06IiwiaW176Ij06IiwiaW177Ij06IiwiaW178Ij06IiwiaW179Ij06IiwiaW180Ij06IiwiaW181Ij06IiwiaW182Ij06IiwiaW183Ij06IiwiaW184Ij06IiwiaW185Ij06IiwiaW186Ij06IiwiaW187Ij06IiwiaW188Ij06IiwiaW189Ij06IiwiaW190Ij06IiwiaW191Ij06IiwiaW192Ij06IiwiaW193Ij06IiwiaW194Ij06IiwiaW195Ij06IiwiaW196Ij06IiwiaW197Ij06IiwiaW198Ij06IiwiaW199Ij06IiwiaW200Ij06IiwiaW201Ij06IiwiaW202Ij06IiwiaW203Ij06IiwiaW204Ij06IiwiaW205Ij06IiwiaW206Ij06IiwiaW207Ij06IiwiaW208Ij06IiwiaW209Ij06IiwiaW210Ij06IiwiaW211Ij06IiwiaW212Ij06IiwiaW213Ij06IiwiaW214Ij06IiwiaW215Ij06IiwiaW216Ij06IiwiaW217Ij06IiwiaW218Ij06IiwiaW219Ij06IiwiaW220Ij06IiwiaW221Ij06IiwiaW222Ij06IiwiaW223Ij06IiwiaW224Ij06IiwiaW225Ij06IiwiaW226Ij06IiwiaW227Ij06IiwiaW228Ij06IiwiaW229Ij06IiwiaW230Ij06IiwiaW231Ij06IiwiaW232Ij06IiwiaW233Ij06IiwiaW234Ij06IiwiaW235Ij06IiwiaW236Ij06IiwiaW237Ij06IiwiaW238Ij06IiwiaW239Ij06IiwiaW240Ij06IiwiaW241Ij06IiwiaW242Ij06IiwiaW243Ij06IiwiaW244Ij06IiwiaW245Ij06IiwiaW246Ij06IiwiaW247Ij06IiwiaW248Ij06IiwiaW249Ij06IiwiaW250Ij06IiwiaW251Ij06IiwiaW252Ij06IiwiaW253Ij06IiwiaW254Ij06IiwiaW255Ij06IiwiaW256Ij06IiwiaW257Ij06IiwiaW258Ij06IiwiaW259Ij06IiwiaW260Ij06IiwiaW261Ij06IiwiaW262Ij06IiwiaW263Ij06IiwiaW264Ij06IiwiaW265Ij06IiwiaW266Ij06IiwiaW267Ij06IiwiaW268Ij06IiwiaW269Ij06IiwiaW270Ij06IiwiaW271Ij06IiwiaW272Ij06IiwiaW273Ij06IiwiaW274Ij06IiwiaW275Ij06IiwiaW276Ij06IiwiaW277Ij06IiwiaW278Ij06IiwiaW279Ij06IiwiaW280Ij06IiwiaW281Ij06IiwiaW282Ij06IiwiaW283Ij06IiwiaW284Ij06IiwiaW285Ij06IiwiaW286Ij06IiwiaW287Ij06IiwiaW288Ij06IiwiaW289Ij06IiwiaW290Ij06IiwiaW291Ij06IiwiaW292Ij06IiwiaW293Ij06IiwiaW294Ij06IiwiaW295Ij06IiwiaW296Ij06IiwiaW297Ij06IiwiaW298Ij06IiwiaW299Ij06IiwiaW300Ij06IiwiaW301Ij06IiwiaW302Ij06IiwiaW303Ij06IiwiaW304Ij06IiwiaW305Ij06IiwiaW306Ij06IiwiaW307Ij06IiwiaW308Ij06IiwiaW309Ij06IiwiaW310Ij06IiwiaW311Ij06IiwiaW312Ij06IiwiaW313Ij06IiwiaW314Ij06IiwiaW315Ij06IiwiaW316Ij06IiwiaW317Ij06IiwiaW318Ij06IiwiaW319Ij06IiwiaW320Ij06IiwiaW321Ij06IiwiaW322Ij06IiwiaW323Ij06IiwiaW324Ij06IiwiaW325Ij06IiwiaW326Ij06IiwiaW327Ij06IiwiaW328Ij06IiwiaW329Ij06IiwiaW330Ij06IiwiaW331Ij06IiwiaW332Ij06IiwiaW333Ij06IiwiaW334Ij06IiwiaW335Ij06IiwiaW336Ij06IiwiaW337Ij06IiwiaW338Ij06IiwiaW339Ij06IiwiaW340Ij06IiwiaW341Ij06IiwiaW342Ij06IiwiaW343Ij06IiwiaW344Ij06IiwiaW345Ij06IiwiaW346Ij06IiwiaW347Ij06IiwiaW348Ij06IiwiaW349Ij06IiwiaW350Ij06IiwiaW351Ij06IiwiaW352Ij06IiwiaW353Ij06IiwiaW354Ij06IiwiaW355Ij06IiwiaW356Ij06IiwiaW357Ij06IiwiaW358Ij06IiwiaW359Ij06IiwiaW360Ij06IiwiaW361Ij06IiwiaW362Ij06IiwiaW363Ij06IiwiaW364Ij06IiwiaW365Ij06IiwiaW366Ij06IiwiaW367Ij06IiwiaW368Ij06IiwiaW369Ij06IiwiaW370Ij06IiwiaW371Ij06IiwiaW3

You can also use the embed code and add it to your HTML page, or blog post or wherever you want them to see the report. Here is I embedded my code;



722 | Page

Security Thoughts?

What You Share is for Everyone!

The first thing you might think is usually security. How you can manage security on this? The short answer is there is no security here. The report is shared through the web, or email, with EVERYONE. so everyone who has the link or embed code can access the report. They cannot edit it. but they can view it with no restriction.

All Report Pages Are Visible

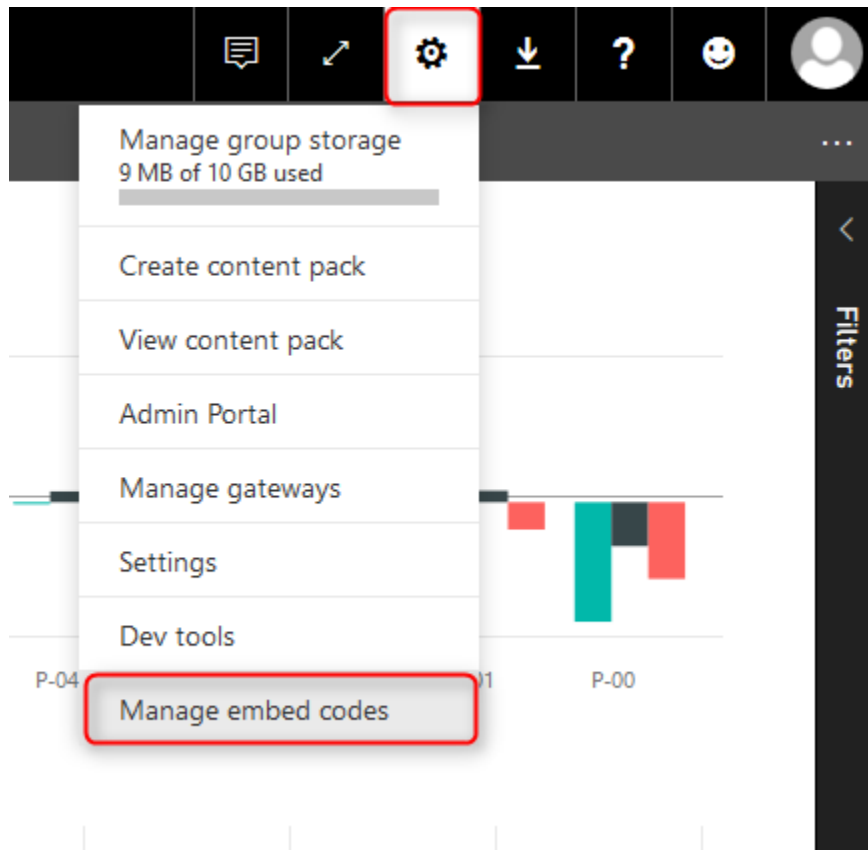
If you have a report with 10+ pages, all of them would be visible to browsers. You can not limit which pages you want to show and which you don't. as an example the report you see above has more than one page, and you can view all of them. I recommend creating different reports if you want to restrict some pages, and share them separately.

Link or Embed code is Synchronized with the Report

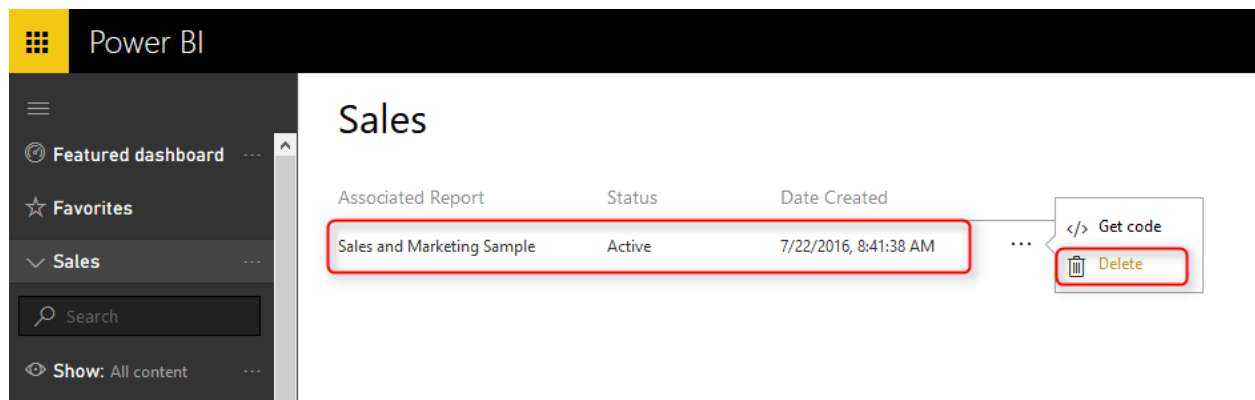
If you make any changes to the report, all changes will be synchronized, because link or embed code are just references to this report. So People will see always the latest version of your report. If you also want to keep the report up-to-date you can schedule it for data refresh.

Removing the Access is Easy!

If for some reasons, you want to revoke the access for everyone to the report you can do it easily. Just go to power bi website or service, and under Setting, click on Manage Embed Codes



Here you will see the embed code created, and you can delete it.



Please note that once you delete the embed code, no-one would be able to access this report from public web. You will see a notification message about it.

Delete publish to web code



You are about to delete a published to web code for this report. The report will no longer be available in the blogs or websites you put it in. Are you sure you want to delete this publish to web code?

[Delete](#)[Close](#)

If you go ahead, and delete the embed code, all links and embed codes will show a message to public web users that this content is not available.

This content is not available.

[Learn more about Power BI.](#)

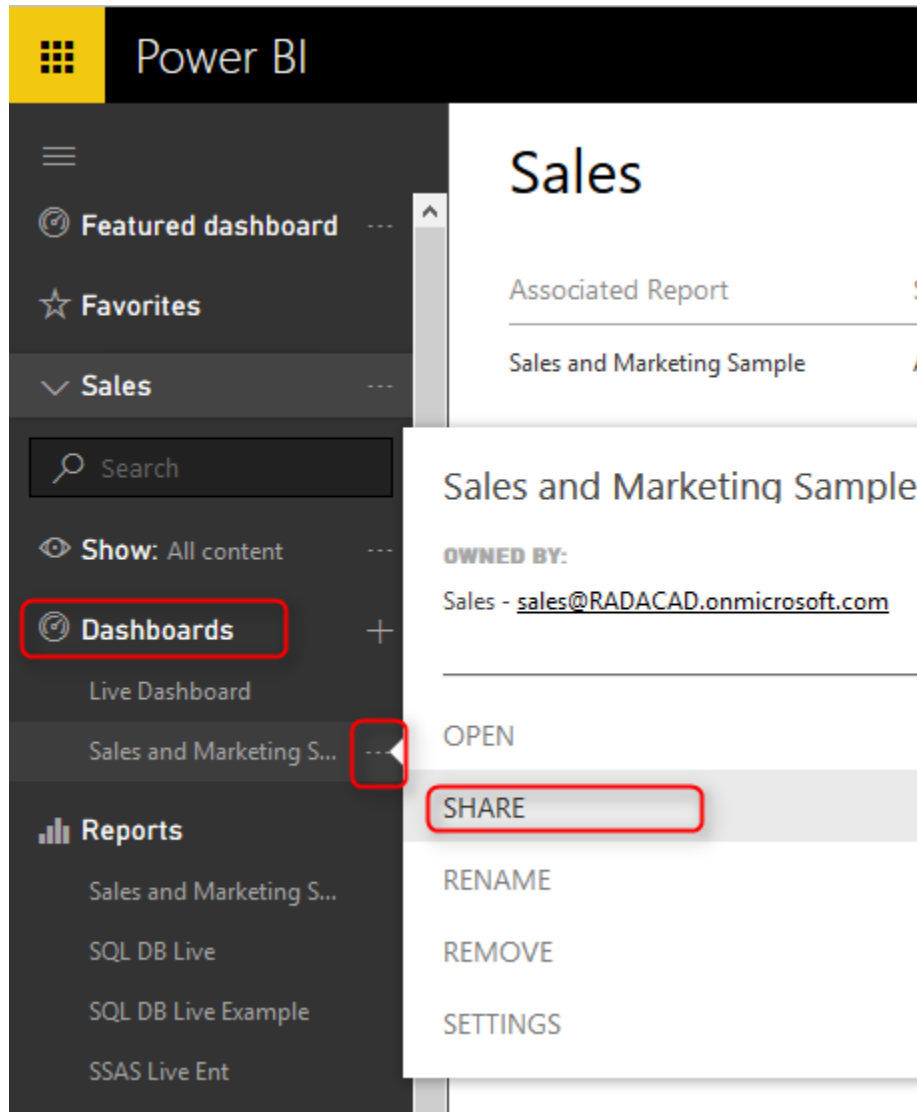
Difference with Sharing?

If you played with Power BI before, you know that you can share your dashboards with people in your organization. This feature is totally different from embed code. Here are some differences between sharing dashboard and embed code;

1. You Share a Dashboard not a Report
2. Only those who has access to the dashboard will see the content
3. You can also give EDIT access to those who you want
4. You can Share dashboard, report, and data set with a group in Power BI
5. Dashboard link which is shared to public won't show anything if they are not authorized to see it.

You Share a Dashboard not a Report

If you click on the ellipsis button beside a dashboard you can share it with others in your organizations with their Power BI accounts.



Only those who has access to the dashboard will see the content

Once you share a report, you can choose who has access to see it based on their Power BI accounts. You can give them Edit access if you want as well.



Share dashboard


Share

Access

Grant access to

Leila Etaati X Enter email addresses

Include an optional message...

 Recipients will have access to the same data, reports, and workbooks as you have in this dashboard, unless their access is restricted by row-level security defined for the dataset. [Learn more](#)

☒ Allow recipients to share your dashboard

☒ Send email notification to recipients

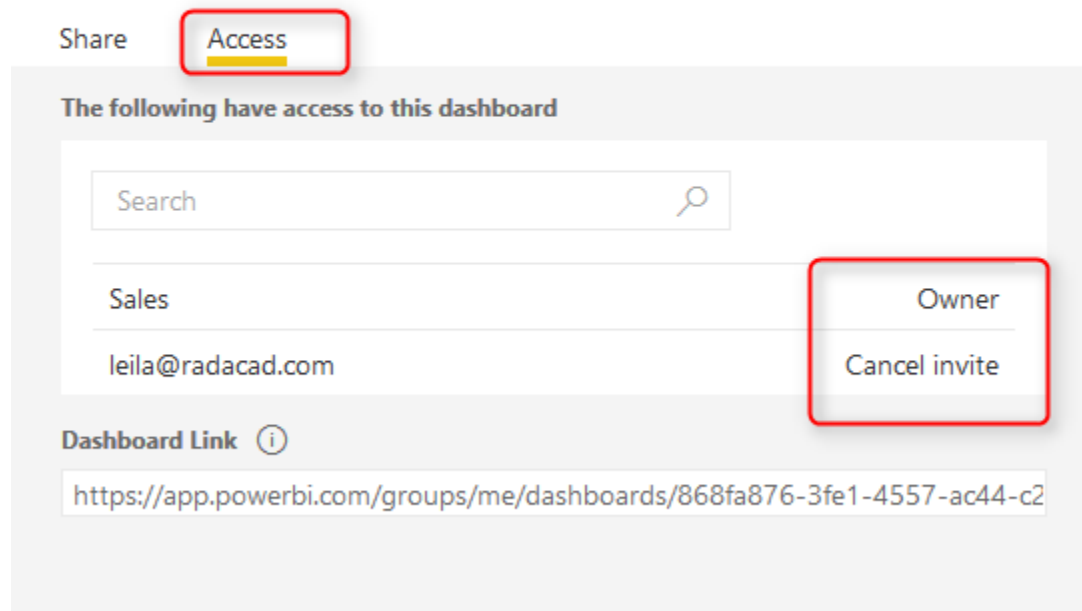
Share

Cancel

You can also give EDIT access to those who you want

You can set access to Owner, Edit, or View for each user or group. and you can revoke this access anytime you want from the Access tab in sharing pane.

Share dashboard



Share **Access**

The following have access to this dashboard

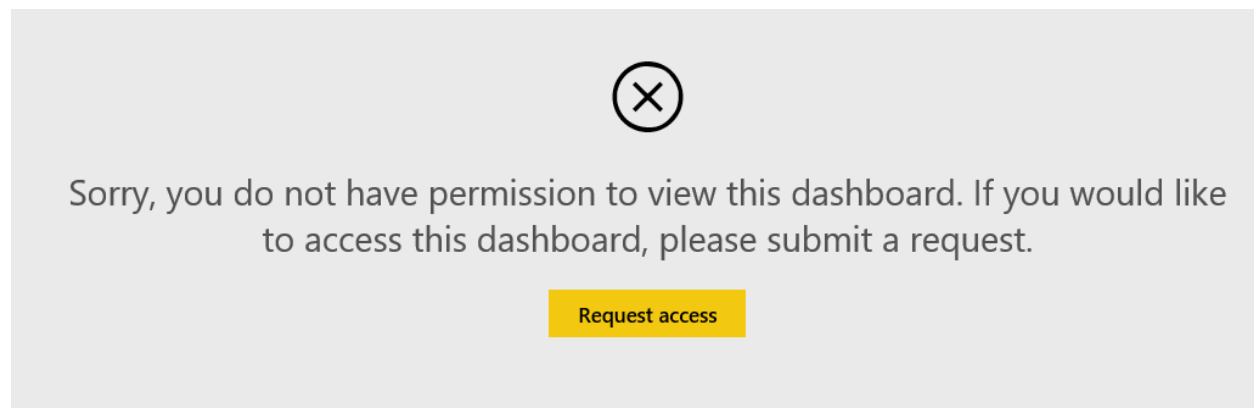
Search
Sales
leila@radacad.com
Owner
Cancel invite

Dashboard Link ⓘ

<https://app.powerbi.com/groups/me/dashboards/868fa876-3fe1-4557-ac44-c2>

Dashboard link Works Only for Authorized Users

In Spite of having the dashboard link available in the Access tab of sharing pane; Only authorized users can see the content when browse the link. Otherwise they will see a message that says they don't have permission to view this dashboard.



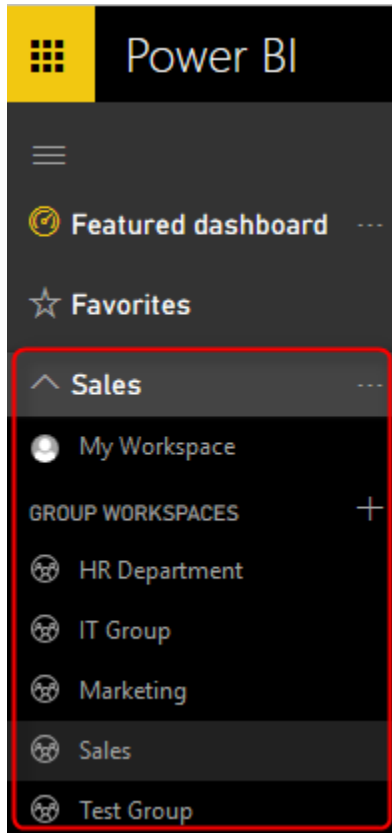
⊗

Sorry, you do not have permission to view this dashboard. If you would like to access this dashboard, please submit a request.

Request access

Power BI Groups is For Authorized Group of Users in Your Organization

You can share dashboards, reports, and data sets with a group which is an Office 365 group in your organization. And users of this group will have access to all content shared under the group. Anyone outside of the group won't be able to see anything. To learn more about [Groups in Power BI read this post](#).



Public Access or Organizational Sharing

Last but not least; the difference between Publish to web, and Sharing is all about difference between giving public access or sharing a content in the organization.

With Sharing dashboards or using Power BI groups you can share content with other users. These users SHOULD BE Power BI users, they cannot access content anonymously. They need to LOGIN to Power BI service to access the content.

With Publish to Web EVERYONE access the report even if they don't have Power BI account. They don't need to login. they can browse the page that contains Power BI embed code with no restriction on viewing the report.

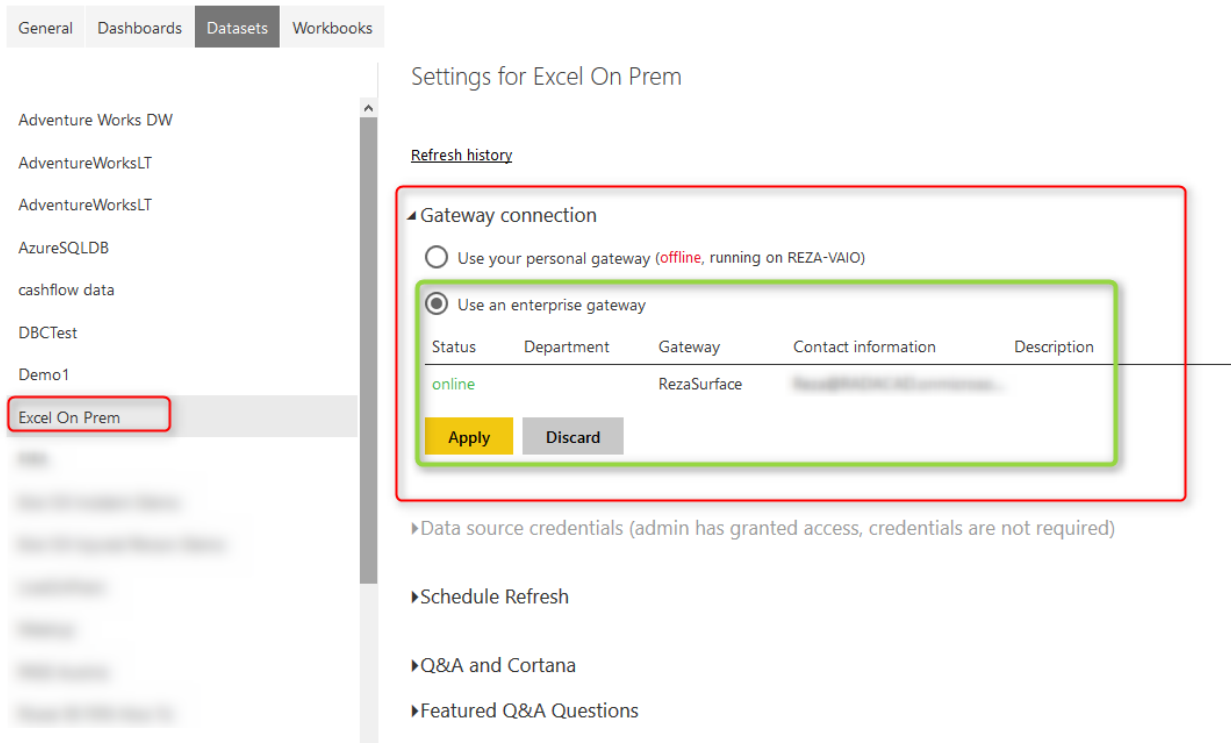
Power BI Embedded

Now that you know about Publish to web, I can explain how Power BI embedded works in a nutshell. Power BI Embedded brings Power BI into applications. Yes you can share your Power BI report through an application with API Keys. And you share the report with application users, even if they don't have Power BI accounts. However you are much more flexible here. You can choose which reports you want to share with which users in the application. Power BI Embedded is the way to bring Power BI experience into an application with security configuration enabled for users. I will explain in a separate post how to use Power BI Embedded. Stay tuned.

Schedule Refresh Local Files on Power BI Web Site

Published Date: June 8, 2016

Settings



The screenshot shows the 'Settings' page in the Power BI web interface. The left sidebar has tabs for 'General', 'Dashboards', 'Datasets', and 'Workbooks'. Under 'Datasets', several data sources are listed, with 'Excel On Prem' highlighted. The main content area is titled 'Settings for Excel On Prem'. It includes a 'Refresh history' link and a 'Gateway connection' section. In the 'Gateway connection' section, two radio buttons are present: 'Use your personal gateway (offline, running on REZA-VAIO)' and 'Use an enterprise gateway'. The 'Use an enterprise gateway' option is selected. Below this, a table displays gateway details:

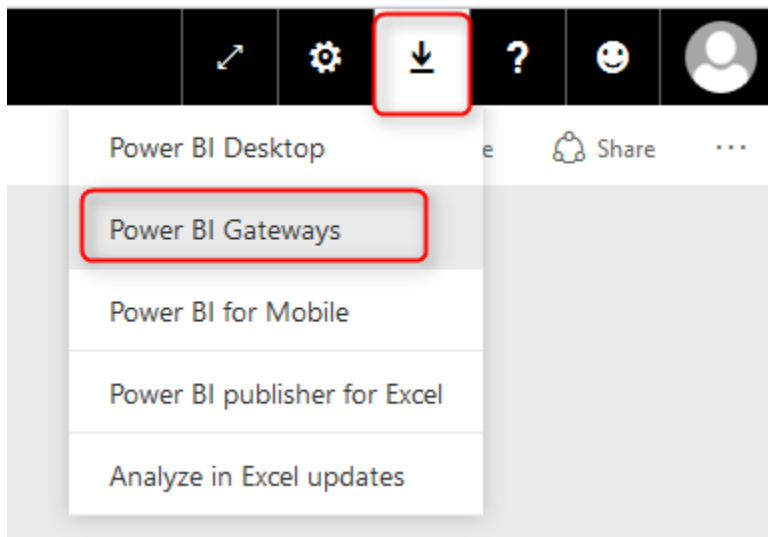
Status	Department	Gateway	Contact information	Description
online		RezaSurface		

Below the table are 'Apply' and 'Discard' buttons. Further down, there are expandable sections for 'Data source credentials (admin has granted access, credentials are not required)', 'Schedule Refresh', 'Q&A and Cortana', and 'Featured Q&A Questions'.

I get this question asked in webinars, speakings, and comments on my blog posts that Is it possible to use a local file as a source in Power BI and schedule that to be refreshed? The answer is Yes, of Course. You can host your files locally or on a server on premises, and then use gateways to create the connection from Power BI website to the local file. And it will be able to schedule automatically. In this post I will show a full walk through of using an Excel file on my local machine as a source, and scheduling that to be refreshed from Power BI service (or website). I won't go into details of what is gateway, or how to build a sample report in Power BI, if you are interested to learn more on those subjects read posts on [Power BI online book; From Rookie to Rock Star](#).

Install and Configure Gateway

There are two gateways that allows us to connect from Power BI service or web site (which is a cloud service) to on-premises data sources; [Personal](#) and Enterprise gateway. There are some differences in these two gateways which I will explain in separate post. In this example I will be using Enterprise Gateway to connect to on-premises Excel file. You can download the gateway from the Power BI website after you logged in, through options mentioned in screenshot below;



Then choose Enterprise Gateway for this example

Choose the gateway that best fits your needs

For personal use

Refresh your on-premises data quickly without waiting for an IT Admin. Designed for use with personal data sets, there is no central monitoring capabilities for this gateway.

Power BI Gateway - Personal

[Download](#)[Learn more](#)

For enterprise deployments

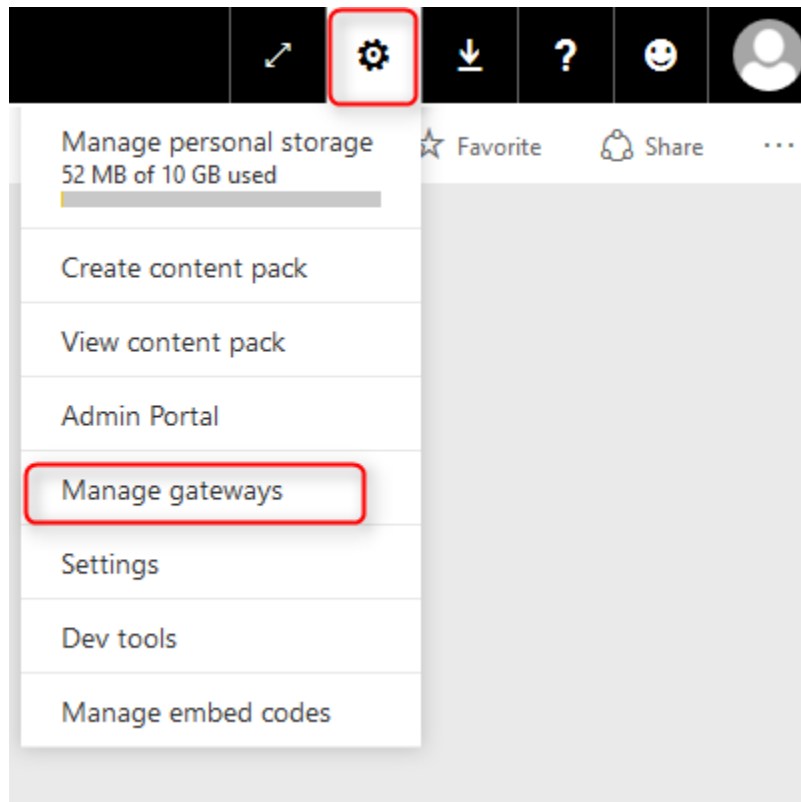
This gateway is used by organizations to serve a large number of users. It also enables administrators to set up access control for individual data sources and monitor usage.

Power BI Gateway - Enterprise

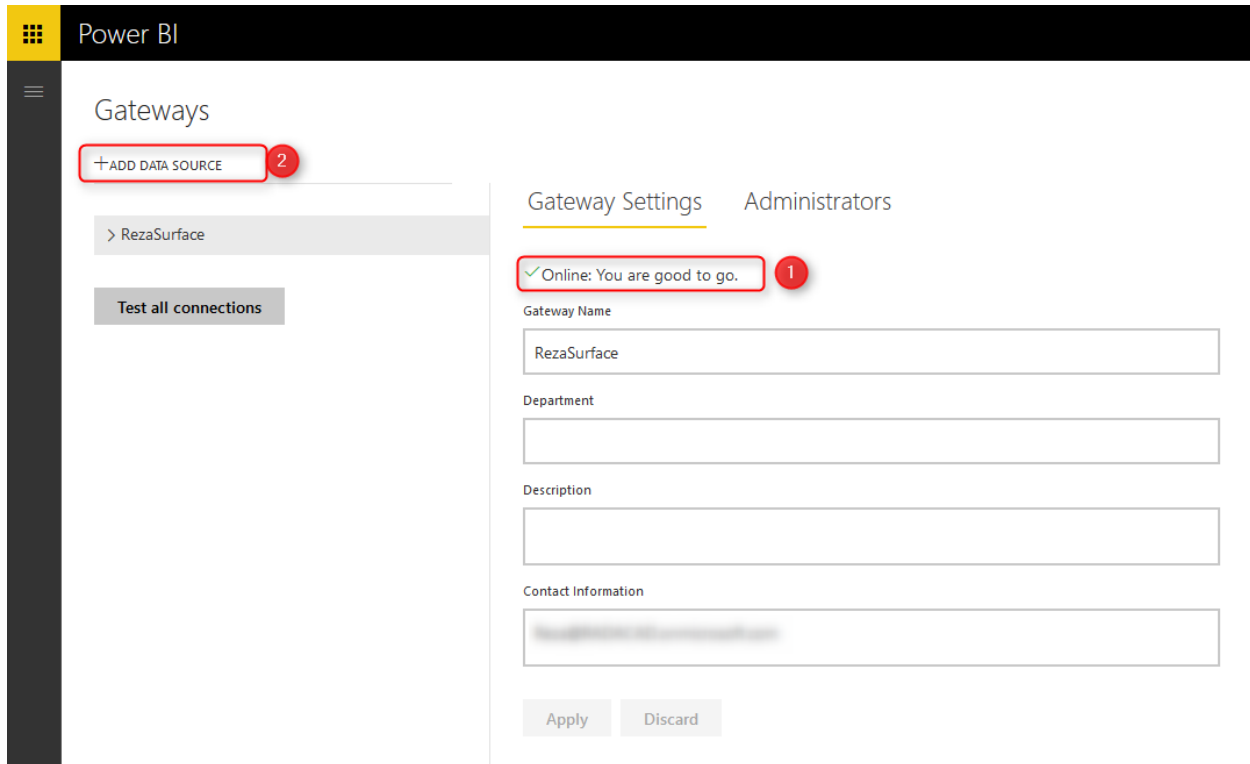
[Download](#)[Learn more](#)

After downloading follow the setup wizard with few steps and you will have Enterprise gateway installed with few clicks. There are some limitations though; You cannot have Enterprise and Personal Gateway on the same machine. I won't talk about their differences, that would be a whole separate post which I will write later. In this post we only go through an example of using this gateway.

After installation, you can configure your gateway. All you have to do is to sign in to your Power BI account from gateway, set a name for your gateway and a code or recovery key. then you will be good to go. I've already setup my gateway and named it as RezaSurface. Now I can manage my gateway through Power BI website, through Setting icon, and then Manage Gateways.



In the Manage Gateways page I can see the gateway I've configured named RezaSurface. And I can also see that it is online with a green checkbox close to it. which means everything works correctly.



Now I can create a Data Source for the file that I want to use as a source. Click on Add Data Source (numbered 2 in screenshot above). as you can see in screenshot below I name the new data source as Excel On Prem, and choose data source type as File, and set full path of file. This should be the local path of file in the computer that has gateway installed on it. Also I enter Windows username and password for the computer that has enterprise gateway installed on it. This username and password will be used to access the local file.

Gateways

+ ADD DATA SOURCE

v RezaSurface

New data source

New data source 1

Test all connections

Data Source Settings

Data Source Name

Excel On Prem

Data Source Type

File

Full path

C:\GetDataExcel\cashflow data.xlsx

The credentials are encrypted using the key stored on-premises on the gateway server. [Learn more](#)

Windows username

Windows password

••••••••••

>Advanced settings

Add

Discard

Note that for this example I will be using sample [Cashflow Data.xlsx](#) file which I previously built a Waterfall chart with it. If you want to learn more about building a waterfall Power BI chart, read [this post](#). After adding data source above I can see the connection is successful to my local excel file.

Gateways

+ ADD DATA SOURCE

▼ RezaSurface

New data source

Excel On Prem

Test all connections

Data Source Settings

Users

✓ Connection Successful

ⓘ Next Step: Go to the [Users tab](#) above and add users to this Data Source

Data Source Name

Excel On Prem

Data Source Type

File

Full path

C:\GetDataExcel\cashflow data.xlsx

The credentials are encrypted using the key stored on-premises on the gateway server. [Learn more](#)

Windows username

●●●●●●●●

Windows password

●●●●●●●●

>Advanced settings

Apply

Discard

Deploy Power BI Solution

Power BI solution that I have is a simple waterfall chart on a cash flow data table in an Excel source. Here is my sample excel source table:

	A	B	C	D	E
1	Period ▼	Base ▼	Down ▼	Up ▼	Cash Flow ▼
2	Start		\$0	\$5,000	\$5,000
3	Jan	\$5,000	\$503	\$0	-\$503
4	Feb	\$4,497	\$1,670	\$0	-\$1,670
5	Mar	\$2,827	\$0	\$4,802	\$4,802
6	Apr	\$7,629	\$1,198	\$0	-\$1,198
7	May	\$6,431	\$3,526	\$0	-\$3,526
8	Jun	\$2,905	\$0	\$1,826	\$1,826
9	Jul	\$4,731	\$2,284	\$0	-\$2,284
10	Aug	\$2,447	\$0	\$3,250	\$3,250
11	Sep	\$5,697	\$1,780	\$0	-\$1,780
12	Oct	\$3,917	\$0	\$2,667	\$2,667
13	Nov	\$6,584	\$0	\$1,500	\$1,500
14	Dec	\$8,084	\$0	\$2,475	\$2,475
15	End	\$10,559			

and here is the waterfall chart I've built for the data set above in Power BI Desktop



If you need more details information about how to build report above read [Waterfall Chart blog post here](#).

I named this Power BI file as Excel On Prem, and publish it to my workspace in Power BI



Publish to Power BI

Select a destination

My Workspace

HR Department

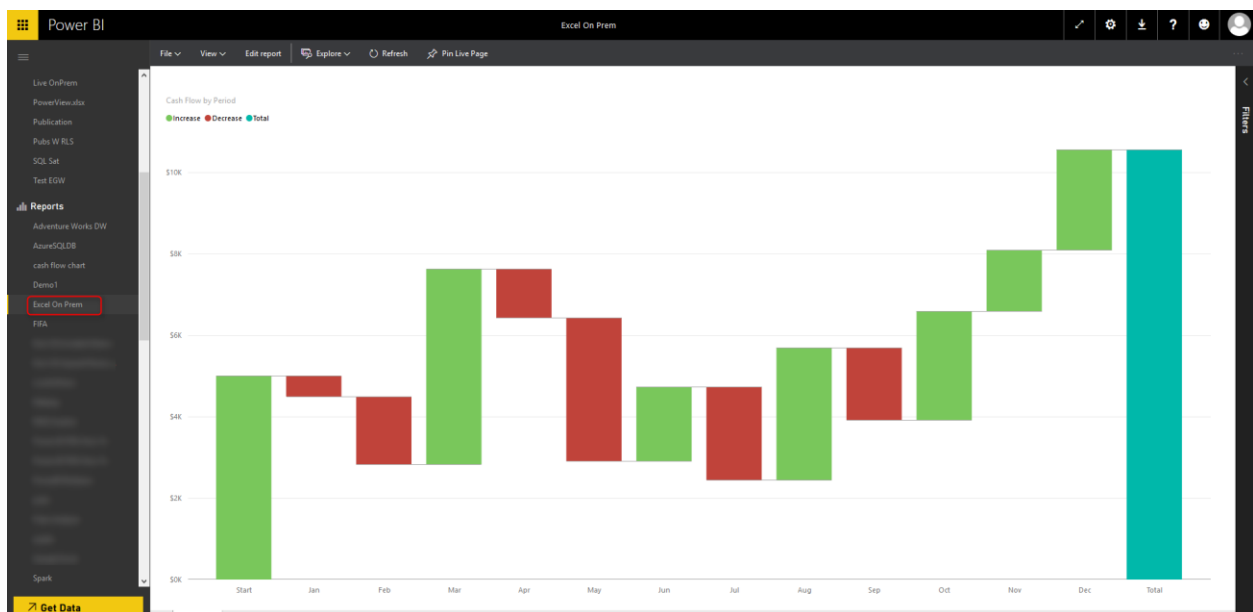
Sales

Test Group

Publish

Cancel

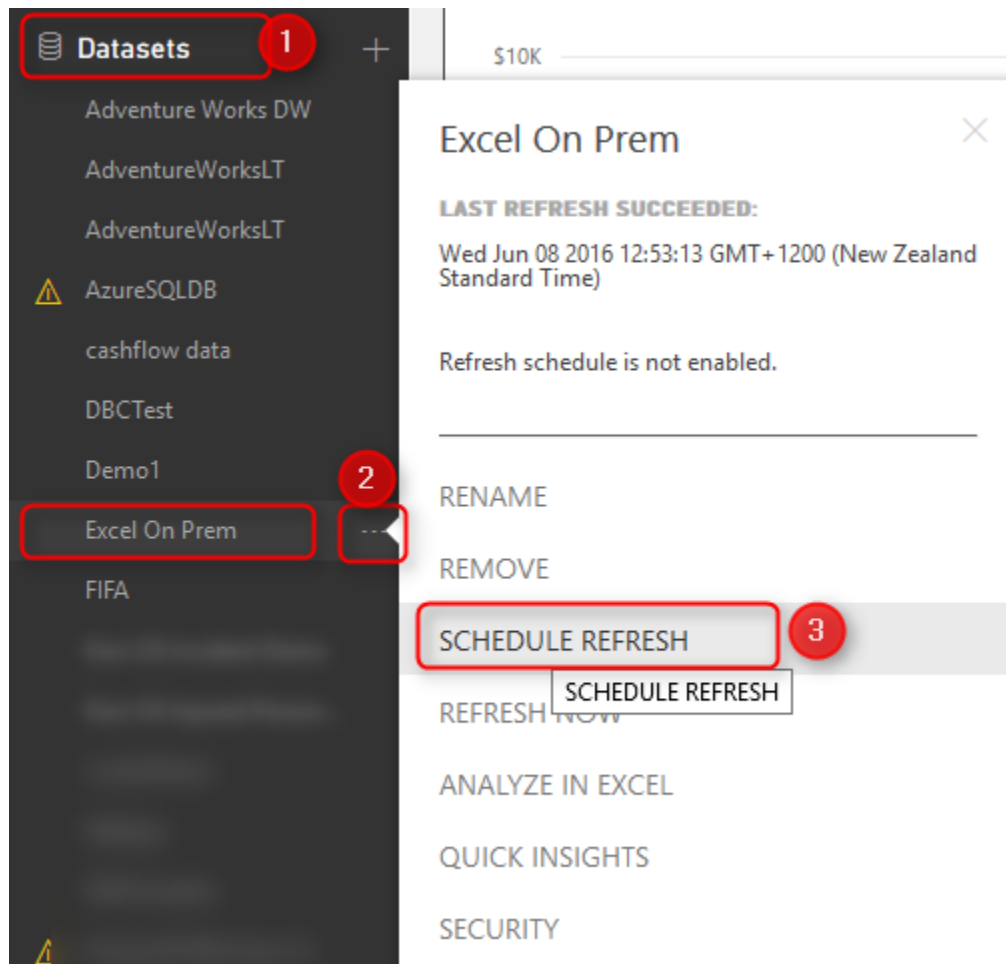
Now I can see my report in Power BI website;



Schedule Refresh

Now that I have my report and data set published in Power BI service, and gateway configured with the data source, I can connect these two together and schedule refresh.

Click on ellipsis button besides the Excel On Prem data set and choose Schedule Refresh.



Then I will be redirected to Settings page where I can configure settings for this data set. As you see in screenshot below, I expand Gateway connection, and I choose using an enterprise gateway which I've already created a data source for this file there before. and then I apply changes.

Settings

General Dashboards **Datasets** Workbooks

Adventure Works DW
AdventureWorksLT
AdventureWorksLT
AzureSQLDB
cashflow data
DBCTest
Demo1
Excel On Prem

Settings for Excel On Prem

[Refresh history](#)

Gateway connection

☐ Use your personal gateway (offline, running on REZA-VAIO)

☒ Use an enterprise gateway

Status	Department	Gateway	Contact information	Description
online		RezaSurface		

Apply Discard

►Data source credentials (admin has granted access, credentials are not required)

►Schedule Refresh

►Q&A and Cortana

►Featured Q&A Questions

The connection between the data set in my Power BI report and the data source in gateway is now created. and I can Schedule Refresh based on the schedule I would like daily or weekly...

Settings

- General
- Dashboards
- Datasets**
- Workbooks

Adventure Works DW

AdventureWorksLT

AdventureWorksLT

AzureSQLDB

cashflow data

DBCTest

Demo1

Excel On Prem

FIFA

New 100 random items

New 100 random items

Load/Refresh

Refresh

Refresh

Refresh

Refresh

Refresh

Refresh

Settings for Excel On Prem

Next refresh: Thu Jun 09 2016 09:00:01 GMT+1200 (New Zealand)

[Refresh history](#)

► Gateway connection

► Data source credentials (admin has granted access)

Schedule Refresh

Keep your data up to date

Yes ☒

Refresh frequency

Daily

Time zone

(UTC+12:00) Auckland, Wellington

Time

[Add another time](#)☒ Send refresh failure notification email to me

Apply

Discard

► Q&A and Cortana

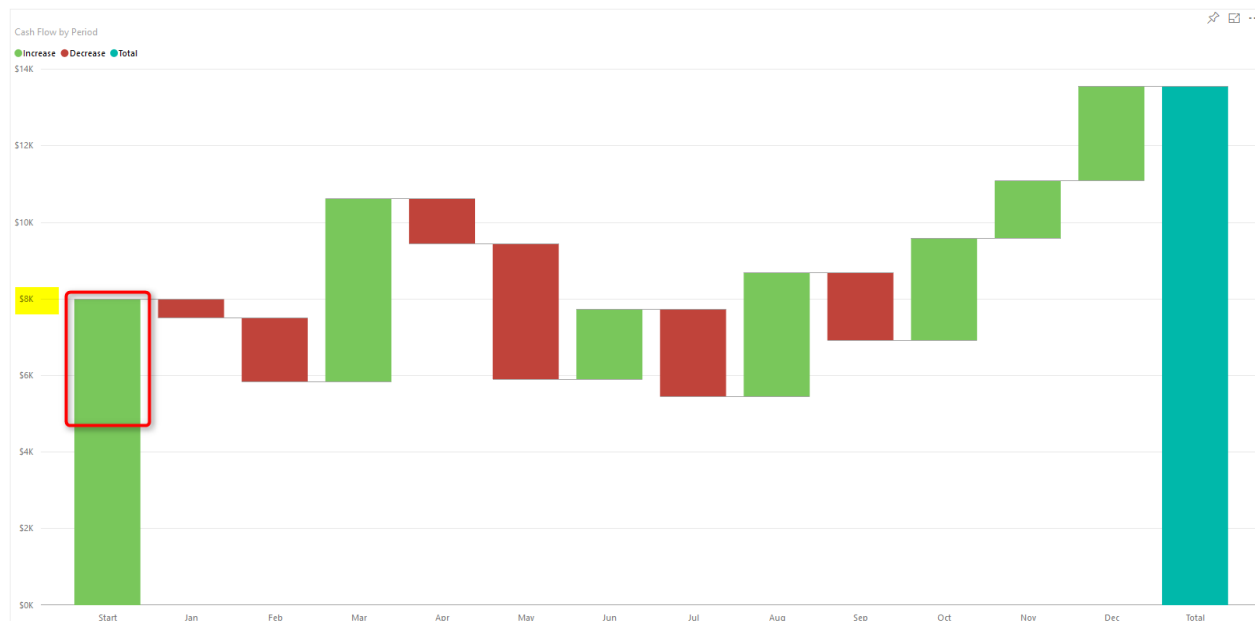
► Featured Q&A Questions

Testing the Refresh

Now If I change the Excel file in my local folder as below; start of my cash flow now changed to 8000\$ instead of 5000\$;

	A	B	C	D	E
1	Period ▾	Base ▾	Down ▾	Up ▾	Cash Flow ▾
2	Start		\$0	\$5,000	\$8,000
3	Jan	\$5,000	\$503	\$0	-\$503
4	Feb	\$4,497	\$1,670	\$0	-\$1,670
5	Mar	\$2,827	\$0	\$4,802	\$4,802
6	Apr	\$7,629	\$1,198	\$0	-\$1,198
7	May	\$6,431	\$3,526	\$0	-\$3,526
8	Jun	\$2,905	\$0	\$1,826	\$1,826
9	Jul	\$4,731	\$2,284	\$0	-\$2,284
10	Aug	\$2,447	\$0	\$3,250	\$3,250
11	Sep	\$5,697	\$1,780	\$0	-\$1,780
12	Oct	\$3,917	\$0	\$2,667	\$2,667
13	Nov	\$6,584	\$0	\$1,500	\$1,500
14	Dec	\$8,084	\$0	\$2,475	\$2,475
15	End	\$10,559			

After schedule refresh run (or even after manual refresh of the data set on Power BI site), I can see the report refreshed as below;



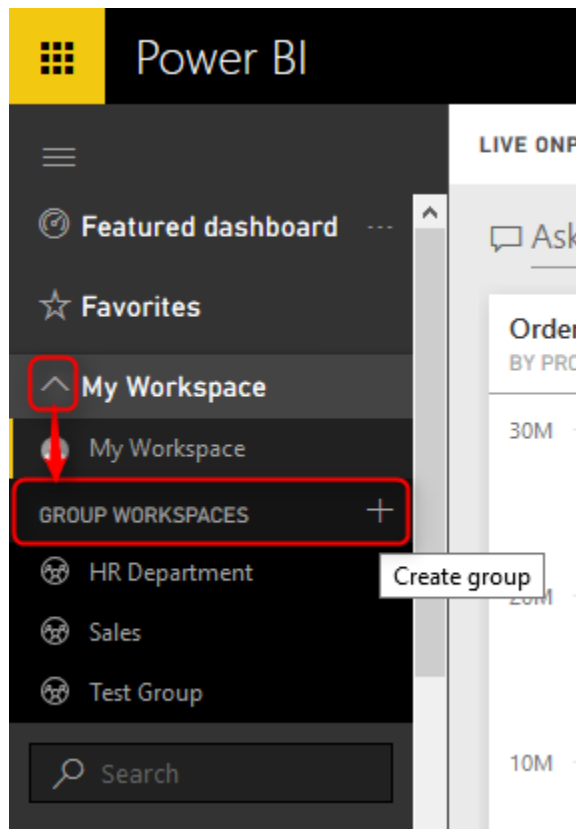
You can see that the waterfall chart starts with \$8K which is the new value from local Excel file.

Summary

In this post you've learned how to use gateway to create connection between on-premises source file and Power BI website, this functionality will expand your Power BI solution to use on-premises source files as source and schedule automatic refresh on those. Note that you can apply this on any files, it shouldn't be only Excel files. There are also options for using a folder as source which is useful when you have multiple files with same structure in a folder on-premises. Think about ways that this functionality can help your solutions.

Groups in Power BI

Published Date: June 13, 2016



You can share dashboards with other Power BI user accounts in your organizations, however you can leverage Power BI groups for creating a work group in your organization, and it works with Office 365 groups, contents will be shared with Power BI groups, and managing members are easily possible through Power BI service or Office 365 admin panel. In this post I get you through some features and basics about Power BI groups. If you like to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

Why Groups?

Sharing Contents with Team

You may have shared a dashboard with couple of your colleagues in your organization, after few weeks a need for new dashboard comes up, and you share that dashboard with them. couple of months later another member of your team asks for access to data

set in Power BI to be able to create his own report and share with some others. Power BI groups enables you to share content (dashboard, report, and data set) with all members of a group. You don't need to share each individual dashboard to each individual users, groups made it easy for you.

Multiple Work Spaces

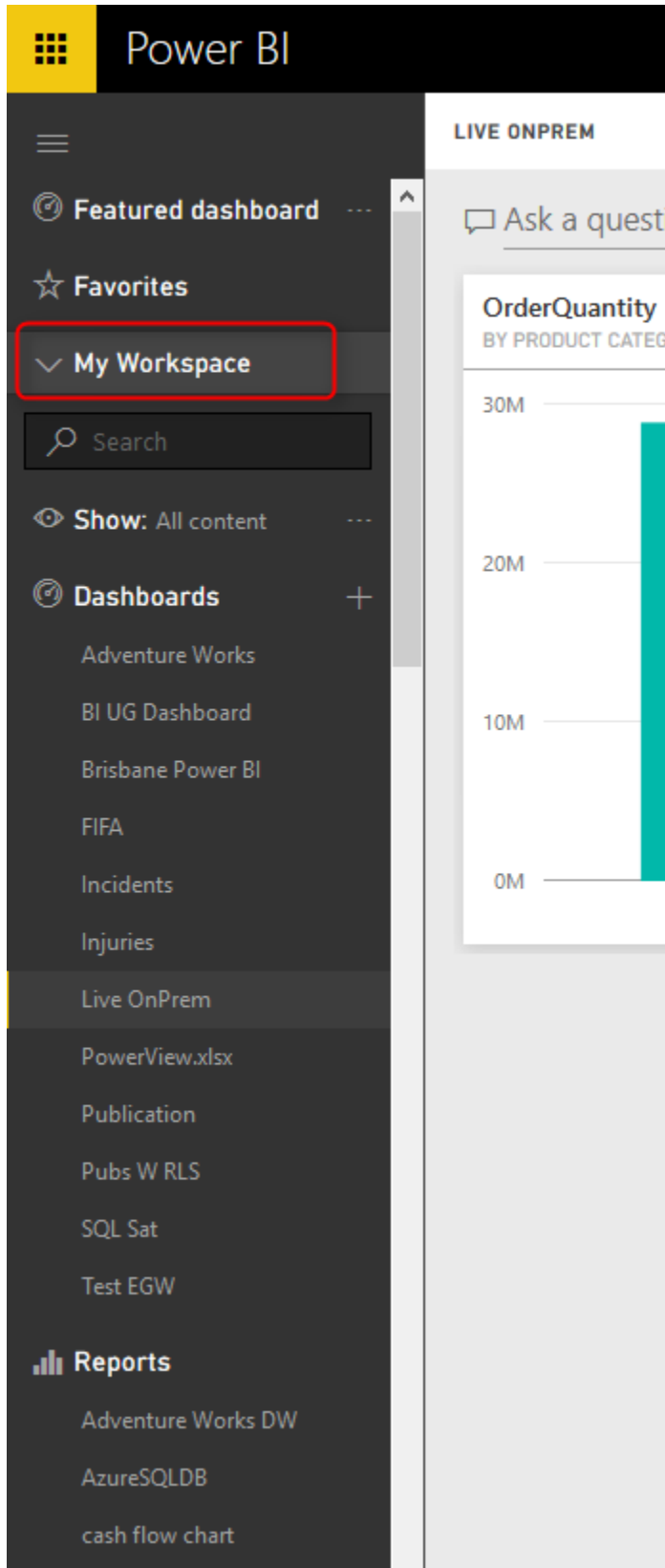
It is really hectic environment when you are part of multiple teams and each team have their own set of dashboards, reports, and data sets. Your work space in Power BI will be hundreds of items and finding something there would be a problem. Power BI groups creates separate work spaces for all members in the group. You can easily switch between work spaces in Power BI.

Isolated User/Group Administration

When you share a content with an individual in the organization, if that person leave the company, or be replaced by someone else from another team, then you have to remove sharing from previous user account, and assign it to the new user account. Best practice is to share content with groups. and members of Groups then easily can be managed by an administrator. Power BI groups are fully synchronized with Office 365 groups. once you used a group in Power BI, then it is only an admin's task to add/remove members from it.

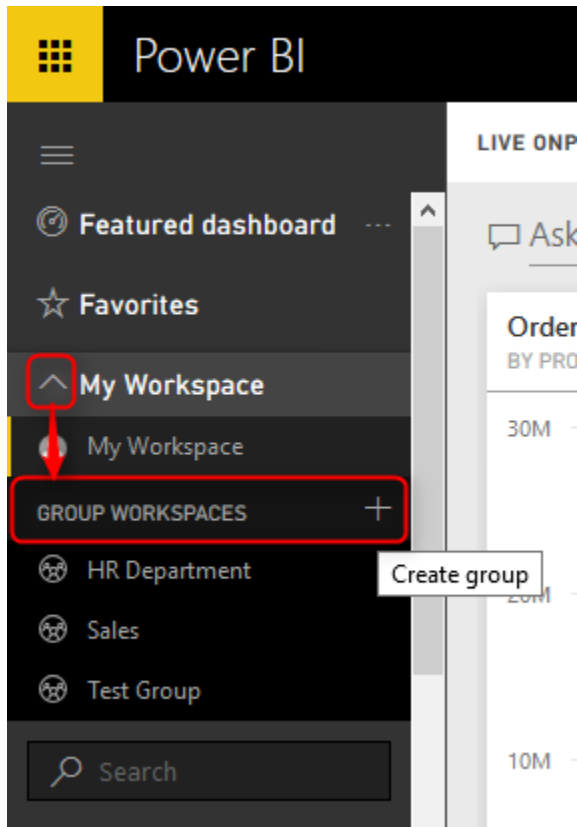
Creating Groups in Power BI

By default you see your own work space named as "My Workspace" in Power BI. all of your dashboards, reports, and data sets are listed there.



The screenshot shows the Power BI web interface. The left-hand navigation pane is visible, with the 'My Workspace' option highlighted by a red rectangular box. The main content area on the right displays a bar chart titled 'OrderQuantity BY PRODUCT CATEGORY'. The chart shows a single teal bar reaching approximately 28M on the y-axis, which ranges from 0M to 30M. Above the chart, there is a search bar and a 'Show: All content' filter. The top of the interface features the 'Power BI' logo and a 'LIVE ONPREM' status indicator.

You can easily create a Power BI group with expanding My Workspace, and click on Create Group



you need to set up a name, and privacy settings for the group. Once you set up the name the group ID would be automatically assigned based on that, you can edit that if you want.

Save


Cancel

Create a group

Name your group

Finance

Group ID

finance 

Available

Privacy

Private - Only approved members can see what's inside ▾

Members can edit Power BI content ▾

Add group members

Enter email addresses

Add

You can also choose if group is private for members only, or for everyone. Or you can choose if you want members to be able to edit Power BI content in the group, or just view it.

Privacy

Private - Only approved members can see what's inside ▾

Members can edit Power BI content ▾

Members can edit Power BI content

Members can only view Power BI content

Adding members in the group is also as easy as adding their email addresses. You can then decide if you want that user to be admin of the group or not.

Add group members

Enter email addresses 1

Add 2

leila@radacad.onmicrosoft.com 3

Member ▼

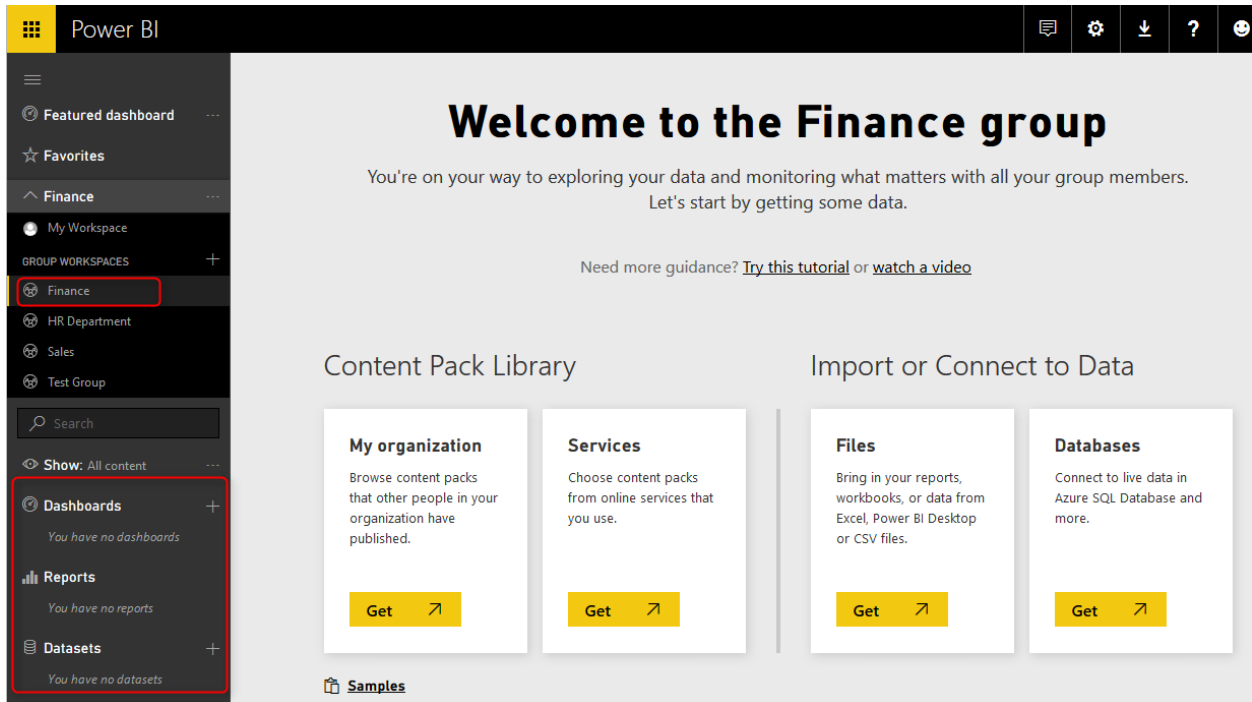
Admin

Member

×

A New Workspace

After creating the group, you will be redirected to a new workspace for that group with no dashboards, reports, and data sets.



Power BI

Welcome to the Finance group

You're on your way to exploring your data and monitoring what matters with all your group members. Let's start by getting some data.

Need more guidance? [Try this tutorial](#) or [watch a video](#)

Content Pack Library

My organization
Browse content packs that other people in your organization have published.
Get

Services
Choose content packs from online services that you use.
Get

Files
Bring in your reports, workbooks, or data from Excel, Power BI Desktop or CSV files.
Get

Databases
Connect to live data in Azure SQL Database and more.
Get

Samples

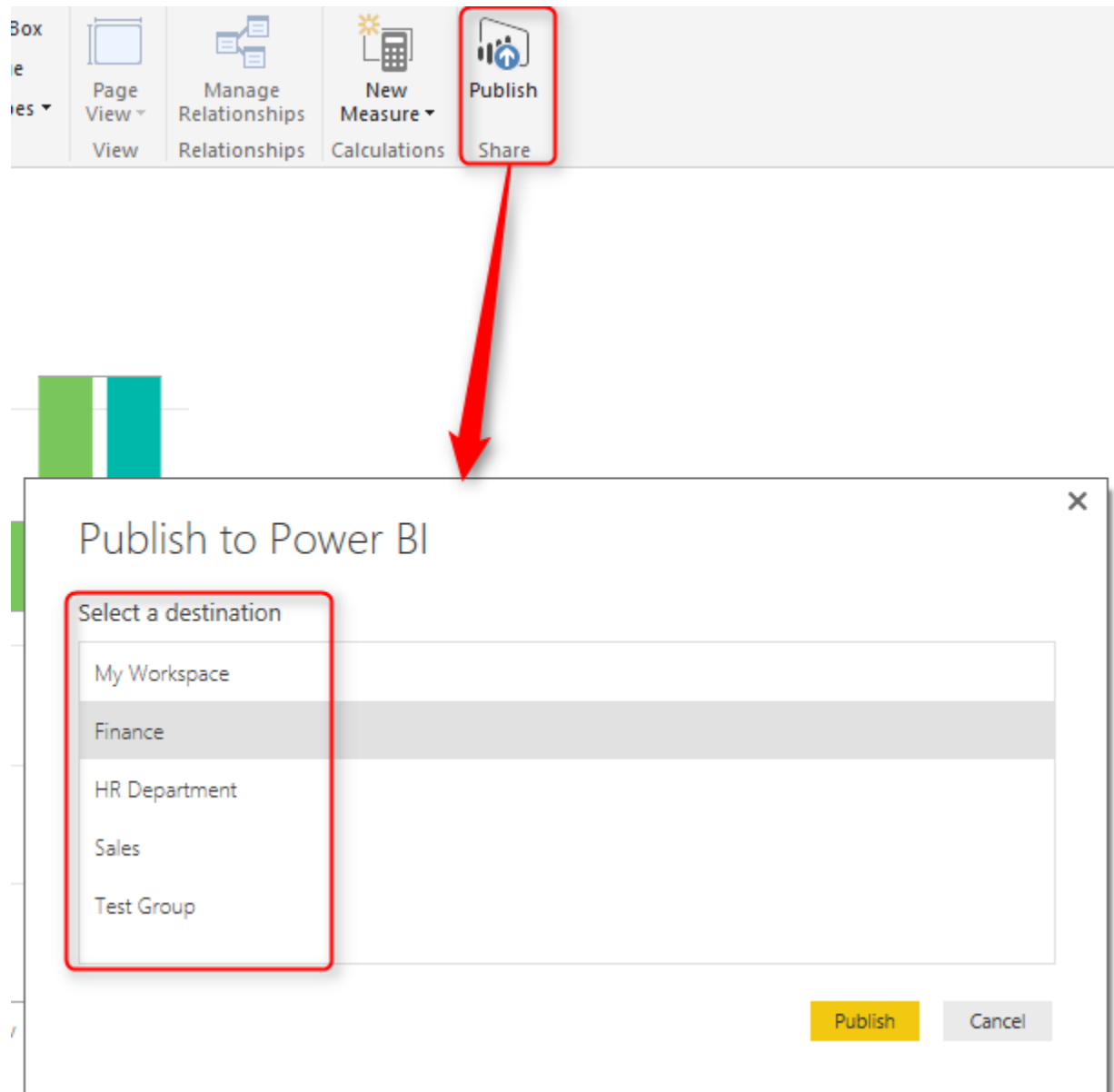
Left Sidebar:

- Featured dashboard
- Favorites
- Finance
- My Workspace
- GROUP WORKSPACES
 - Finance
 - HR Department
 - Sales
 - Test Group
- Show: All content
- Dashboards (You have no dashboards)
- Reports (You have no reports)
- Datasets (You have no datasets)

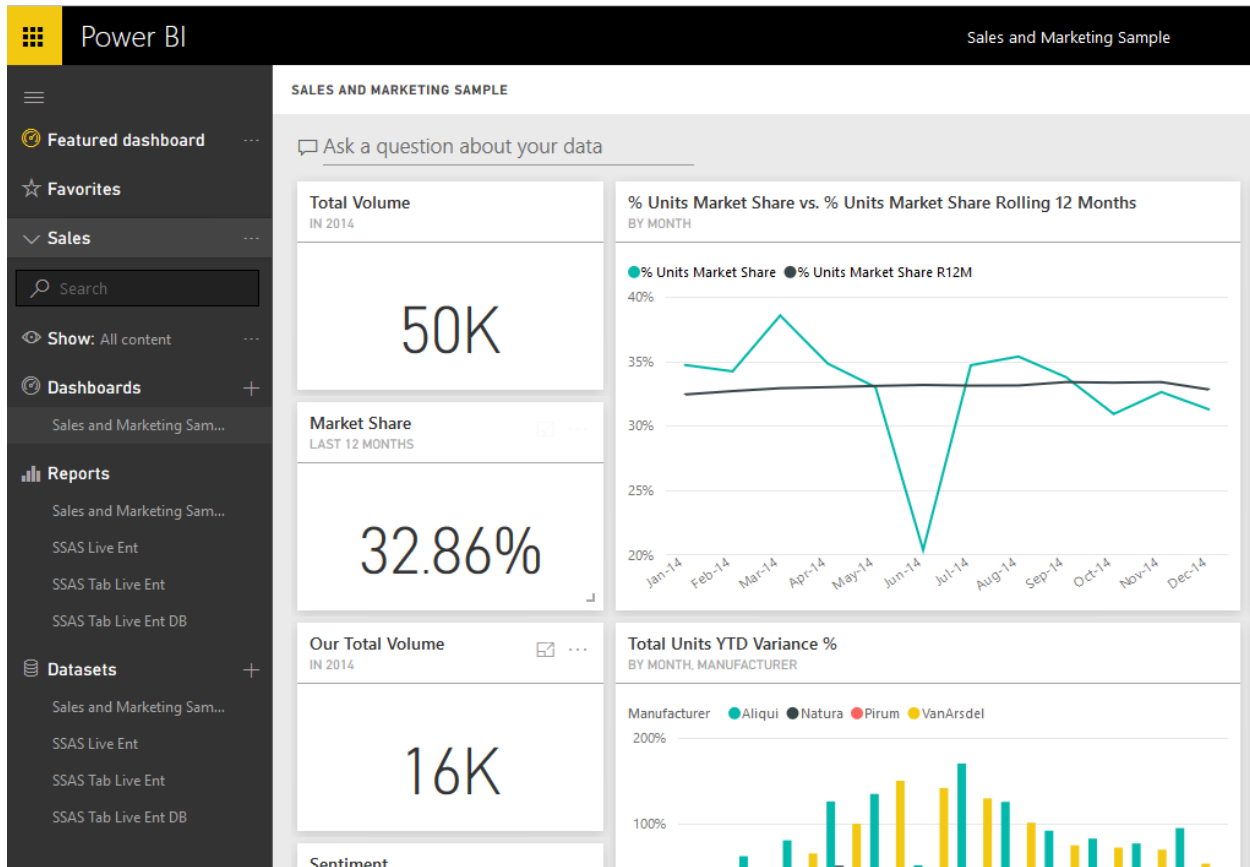
Every content in group is shared between all members of the group. Let's create the very first content in this group.

Publish Content to Group

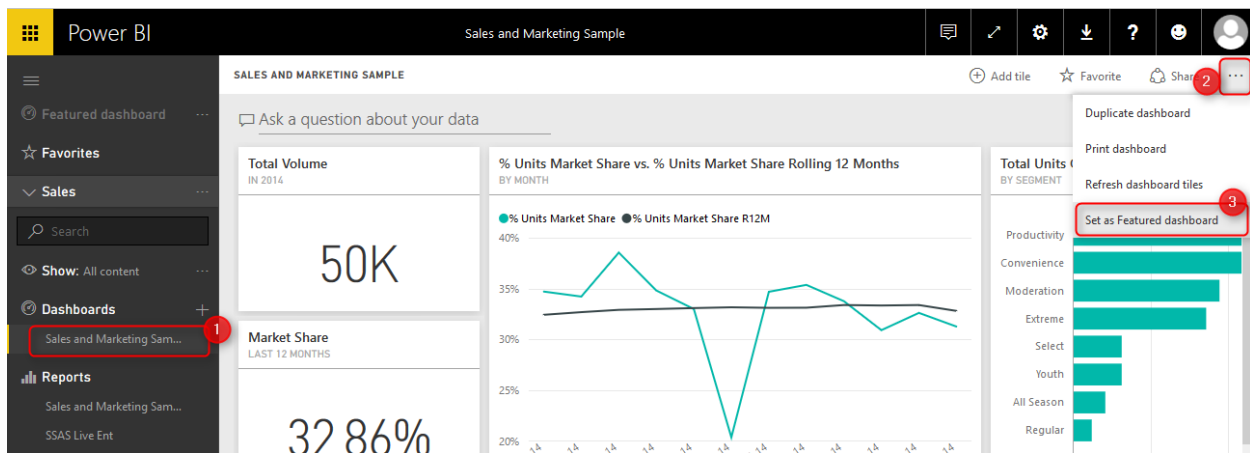
When you publish reports from Power BI Desktop, you will see all groups that you are member of, and you can choose where you want to publish. If you don't see any groups or options for Publish that means you are not part of a Power BI group.



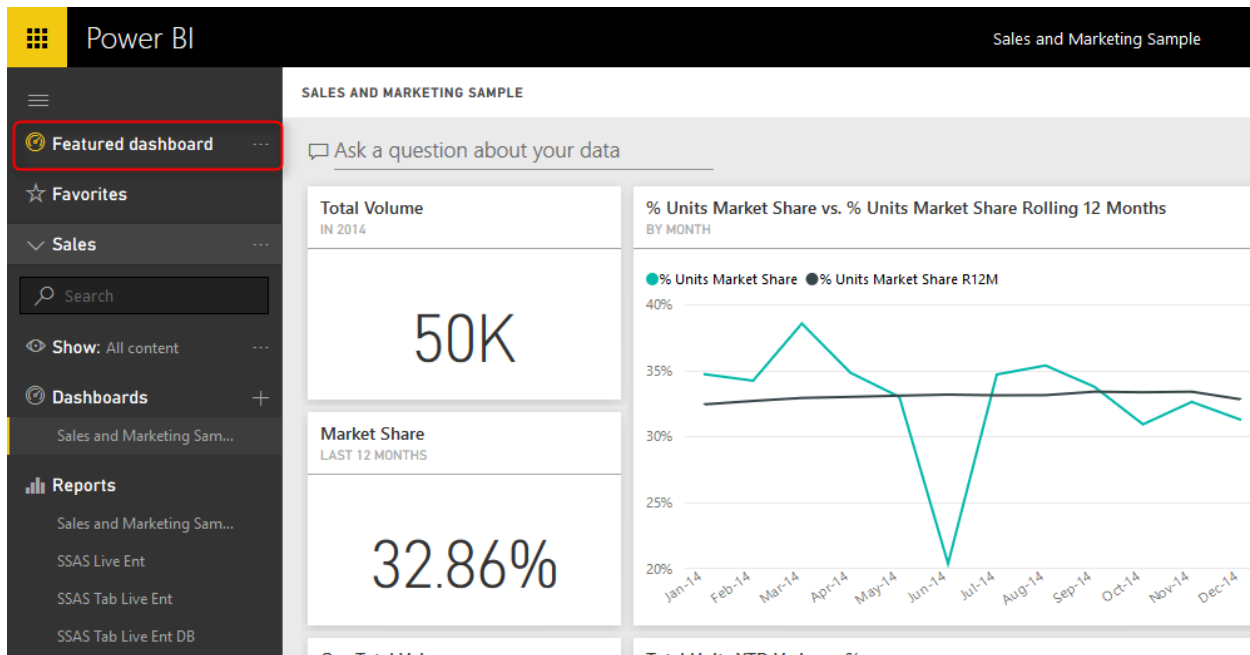
After publish your report and data set will be visible to all group members in Power BI services. A group can have multiple reports, dashboards, and data sets (such as screenshot below for Sales Group)



You can have a featured dashboard for your group, which is like a landing page in Power BI group. everyone from group would see this dashboard when they log in to the services. Featured dashboard can be easily set from each dashboard.

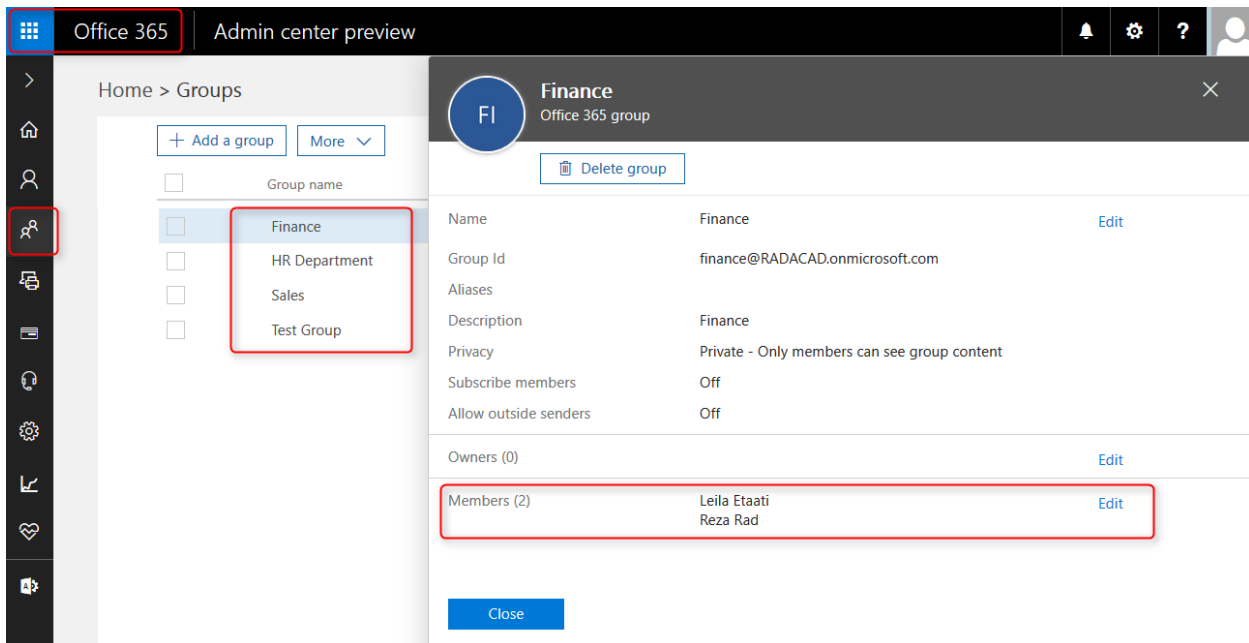


You can then see the featured dashboard anytime you login to Power BI services. Only one dashboard can be set as featured dashboard.



Office 365 Synchronization

Power BI groups are fully synchronized with Office 365 groups. If you go to O365 admin panel, you will see groups that you have created in Power BI. and you can manage groups and their members in this admin panel;




The image shows the Office 365 Admin center preview. The top navigation bar includes "Office 365" (highlighted with a red box) and "Admin center preview". The left sidebar contains navigation icons, with the "Groups" icon (highlighted with a red box) selected. The main content area displays the "Finance" Office 365 group. The group details include:

- Name:** Finance (with an "Edit" link)
- Group Id:** finance@RADACAD.onmicrosoft.com
- Aliases:**
- Description:** Finance
- Privacy:** Private - Only members can see group content
- Subscribe members:** Off
- Allow outside senders:** Off
- Owners (0):** (with an "Edit" link)
- Members (2):** Leila Etaati, Reza Rad (with an "Edit" link)

A "Close" button is located at the bottom of the group details panel.

This functionality is very helpful in enterprise solutions, where you share contents with a group in Power BI, and then members of group will be managed by an Office 365 admin.

Groups also can be created from Office 365 admin panel. for example creating a group such as below;

**IT Group**
Office 365 group

Add a group

Type

Office 365 group

Name *

IT Group

Group Id *

itgroup @ RADACAD.onmicrosoft.com

Available

Description

Privacy *


Public - Anyone can see group content

Language *

English (New Zealand)

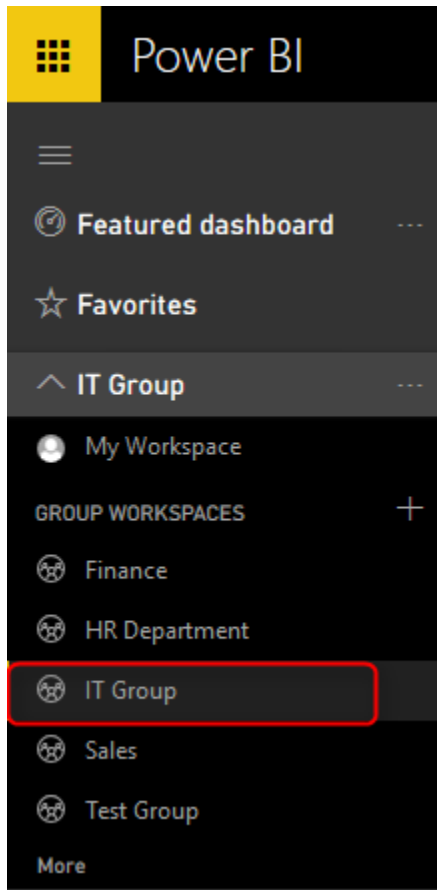
Subscribe members so they receive group conversations and calendar events in their inbox. Otherwise, they'll only see them in the group inbox. ☒ On

Owner *

**Reza Rad**
Reza@RADACAD.onmicrosoft.c... [Remove](#)

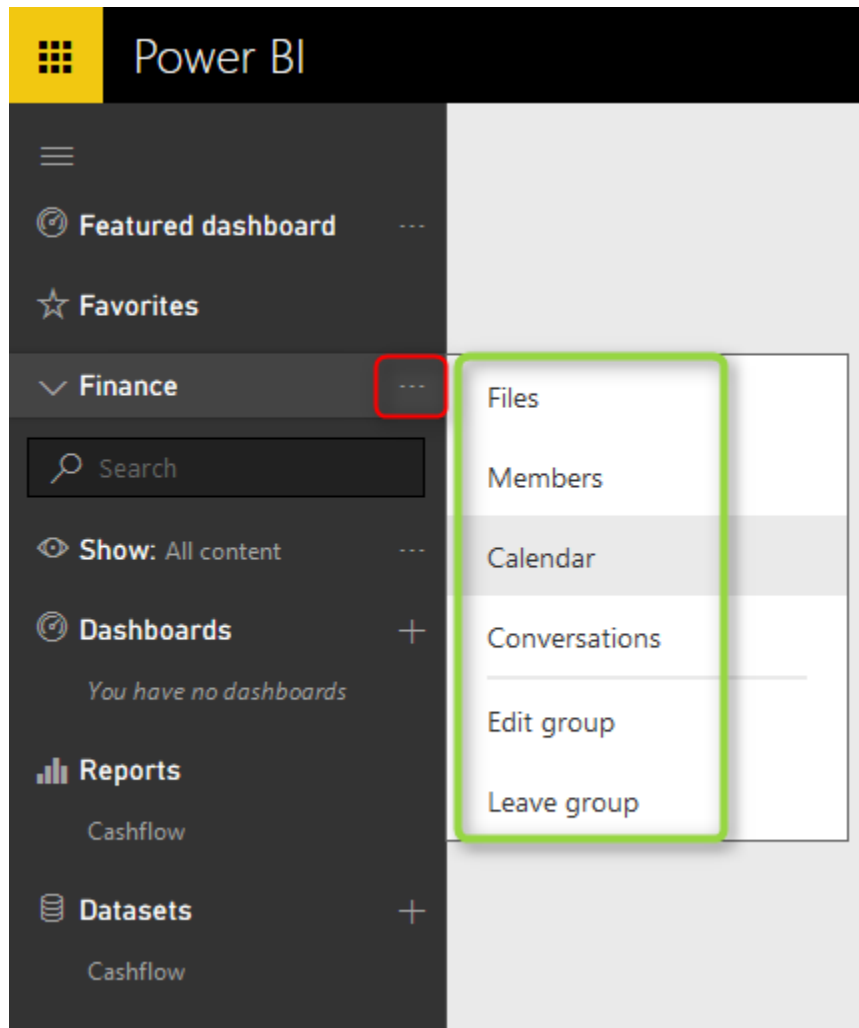
[Add](#) [Cancel](#)

Will be visible in Power BI service of all members (who have Power BI subscription added to their Office 365 subscription options);



That means if you already have Office 365 groups created for your organization, then you don't need to create Power BI groups for them again, they will be visible for users who have Power BI subscription.

You can also navigate to other group pages such as Calendar, Members, and Conversations (based on your permission level in the group)



If a group be deleted from Office 365, all of its content in Power BI with the workspace itself will be deleted.

Power BI Pro

At the time of writing this blog post, Power BI groups are a Pro option and are not part of Power BI free user account.

Need Power BI Pro for Groups



To access groups in Power BI, try Power BI Pro for 60 days, or contact your Office 365 tenant administrator to purchase Power BI Pro.

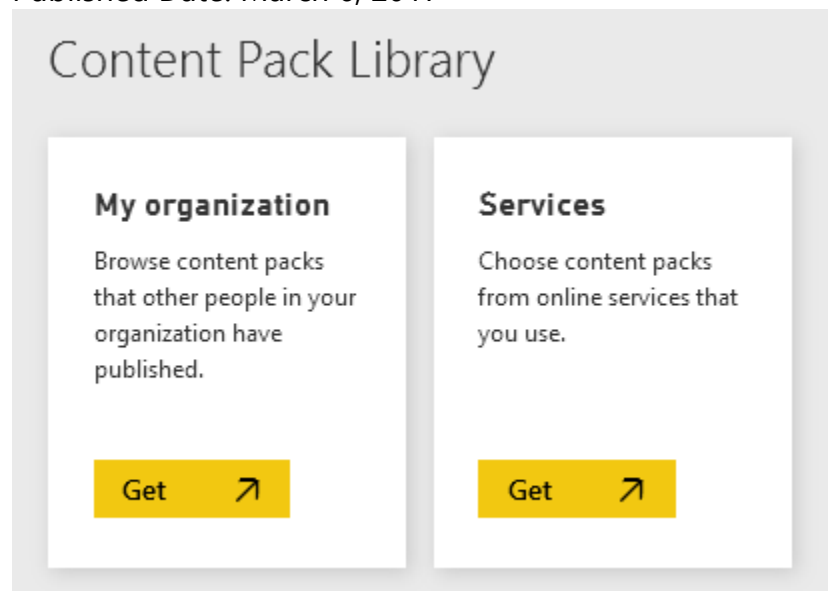
[Get help](#)

Try Pro for free

Cancel

Content Pack; Sharing, Self-Service and Governance Together

Published Date: March 6, 2017

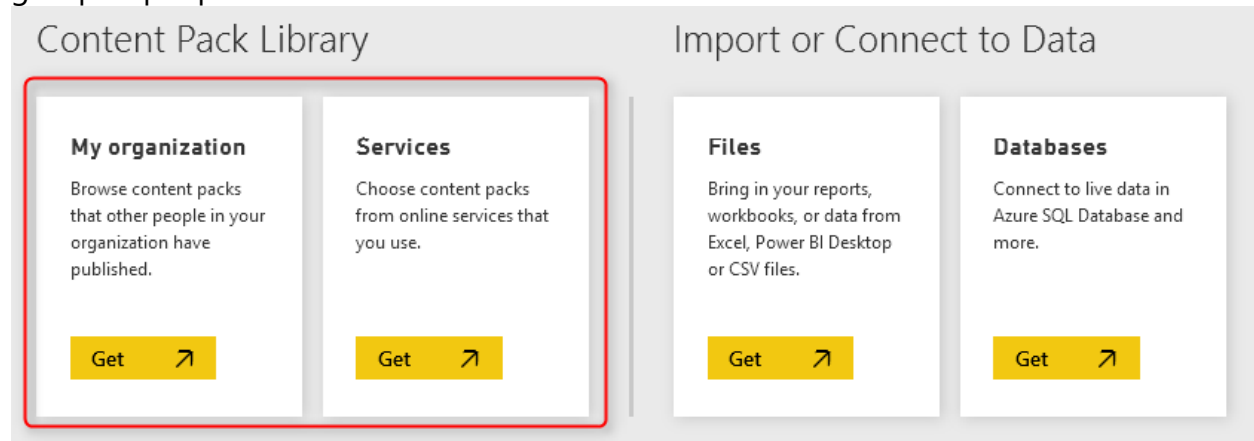


There are many ways to share Power BI content with others. I have previously written about some of these methods, including [Workspaces \(or groups\) in Power BI](#), [Publish to Web](#), and [Power BI Embedded](#). In this post, you will learn about organizational content pack which is another method of sharing. This method is a recommended approach for sharing your Power BI content with users. You will have full governance around your Power BI content, and users will be able to use self-service features of Power BI. This is the best method to share your content with others. Let's take a closer look at this method and see how does it work. If you are interested to learn more about Power BI, read the [Power BI book from Rookie to Rock Star](#).

Content Packs

Content Packs is a self-explanatory name, and it means a package of Power BI content (which can be Dashboard, Report, and Data set). There are two types of content packs; Organizational, and Service Content Packs. Service Content Packs are a set of Power BI content built by normally someone from a company who provided that service (such as Google Analytics, MailChimp, and etc) and you get access to it with your account information. We won't talk about service content pack in this post. Organizational Content Pack is a way to share your Power BI content with others easily. an

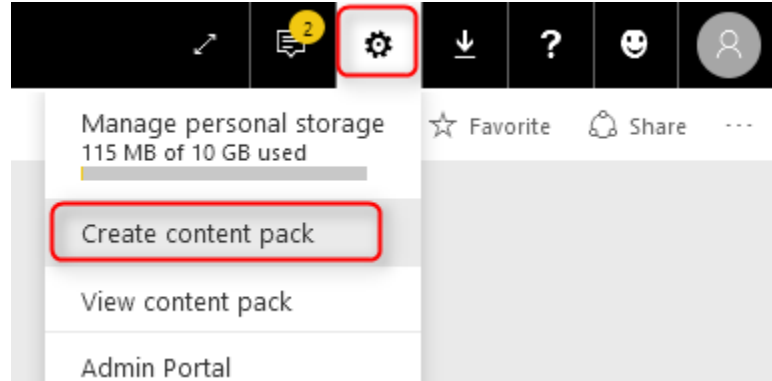
organizational content pack can be shared with your entire organization, or specific group of people.



How to Create Organizational Content Pack?

Creating content packs are easy, all you need to do is to go to Setting Icon in your Power BI account, and click on Create Content Pack.

** Content Packs are **Power BI Pro** features. Your account needs to be Power BI Pro account to use this feature. and everyone using this content also should have Power BI Pro account.



The create content pack dialog box is easy to use. Firstly, set the people whom this content pack will be shared with (Specific Groups, based on their O365 groups or Power BI accounts, or entire organization. Set a title, and description, and a logo. Then select contents that you want to be part of this package. you can always change this selection later on. If you select a higher level item, sub items will be automatically selected. For example, if you select a Dashboard, Report and Data set under that will be automatically selected. in picture below I have selected few items to share.

Choose who will have access to this content pack:

☐ Specific groups ☒ My entire organization

Title

Sample Content Pack 20170306

Description

This is ~~reza~~ rad's sample content pack



Upload an image or company logo

Image size: 45 KB or less, 4:3 aspect ratio, JPG or PNG format

[Use default](#)

Select items to publish

Dashboards

- ☐ Injuries
- ☐ Pubs W RLS
- ☐ Brisbane Power BI
- ☐ Incidents
- ☒ GPVC
- ☒ DotNetUG Dashboard
- ☐ Pubs Sales Data Analy...
- ☐ CIUG

Reports

- ☐ DAX Version 2
- ☐ Demo1
- ☐ DIF RLS
- ☒ DotNetUG
- ☐ DWCNZ
- ☐ DYN RLS 20161022
- ☐ Dynamic RLS 20170...
- ☐ Dvnamic RLS 20170...

Datasets

- ☐ Test Web Refresh2
- ☐ Test Web Refresh3
- ☐ Scatter
- ☒ GPVC Pubs Sample
- ☒ DotNetUG
- ☐ Pubs Sample Data Ana...
- ☐ CIUG
- ☐ PrivateFlight

The content pack will be available in your organization's content gallery. [Learn more](#)

☐ Make this a content pack template. This only applies to Power BI Desktop (.pbix) with imported data. [L...](#)

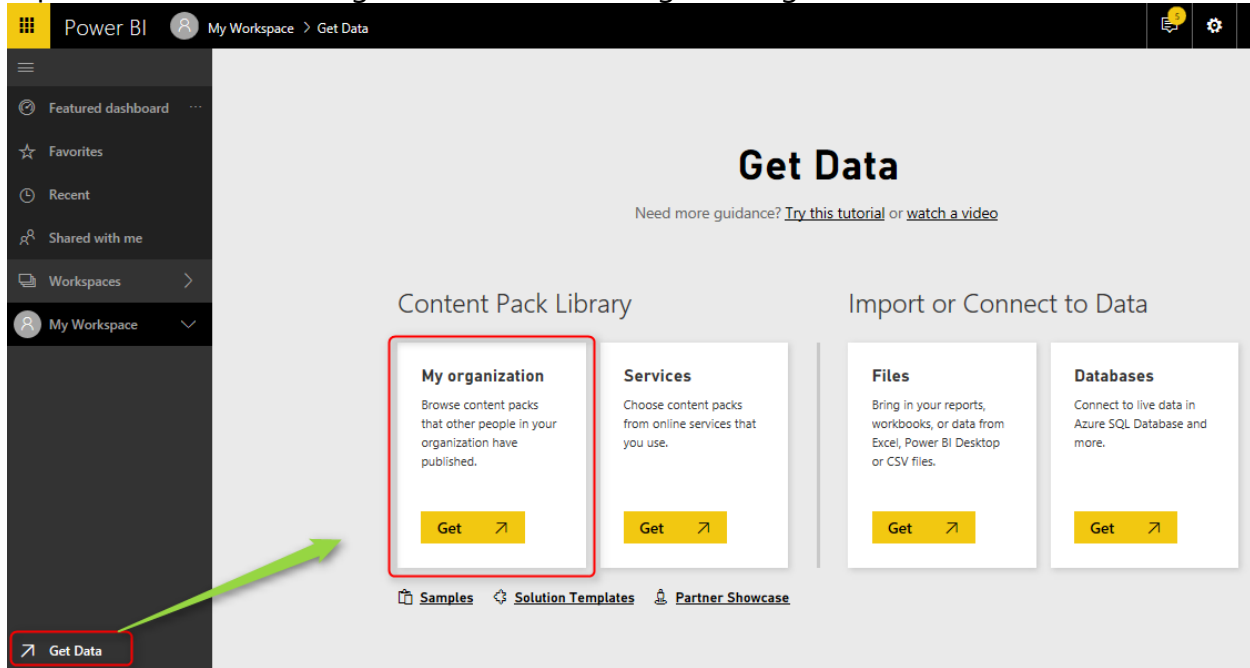
Publish

Cancel

After Publish, your content pack will be ready to use.

Consuming Content Packs

Now if someone from your organization (or the group of people that this content pack is published for them) logs into Power BI, and go through Get Data section;



Users will see content packs shared with them, and they can select the one that they want to install. and then click on Get It Now.

AppSource

Power BI

< Apps



Sample Content Pack 20170306

Reza Rad

This is reza rad's sample content pack

GET IT NOW

Products

Power BI

Publisher

Reza Rad

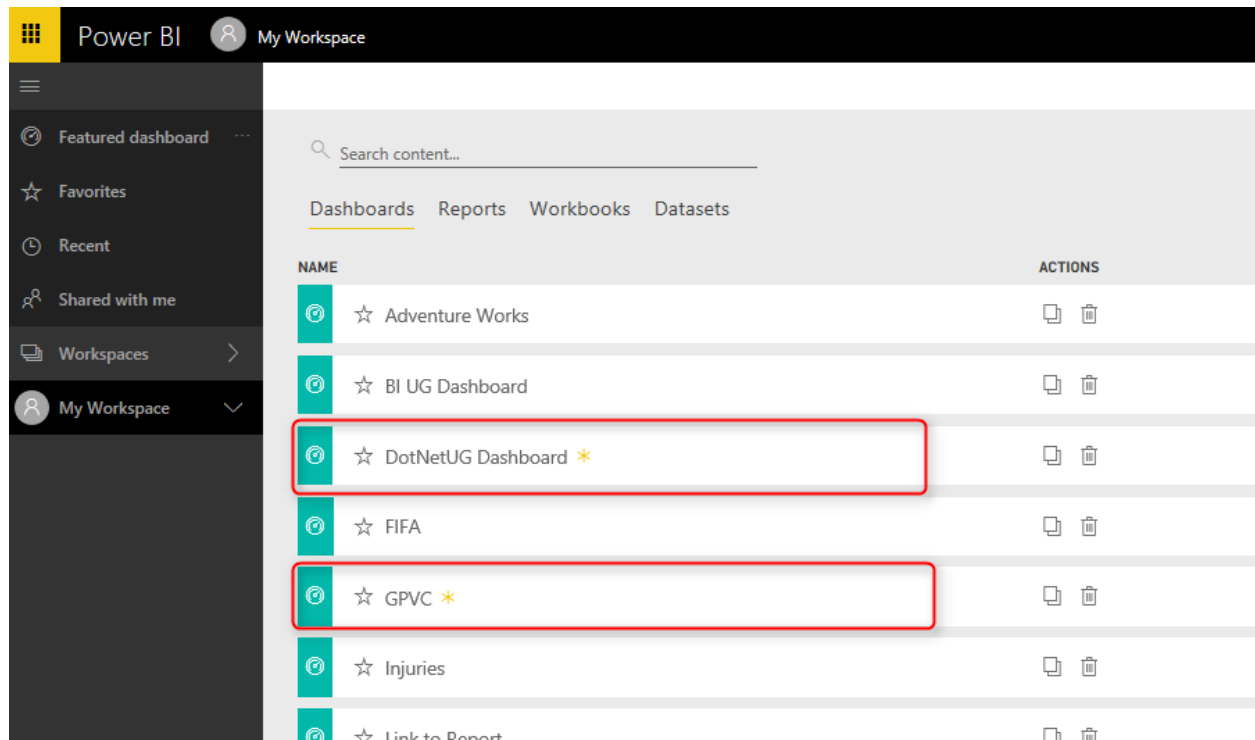
Version

1.0

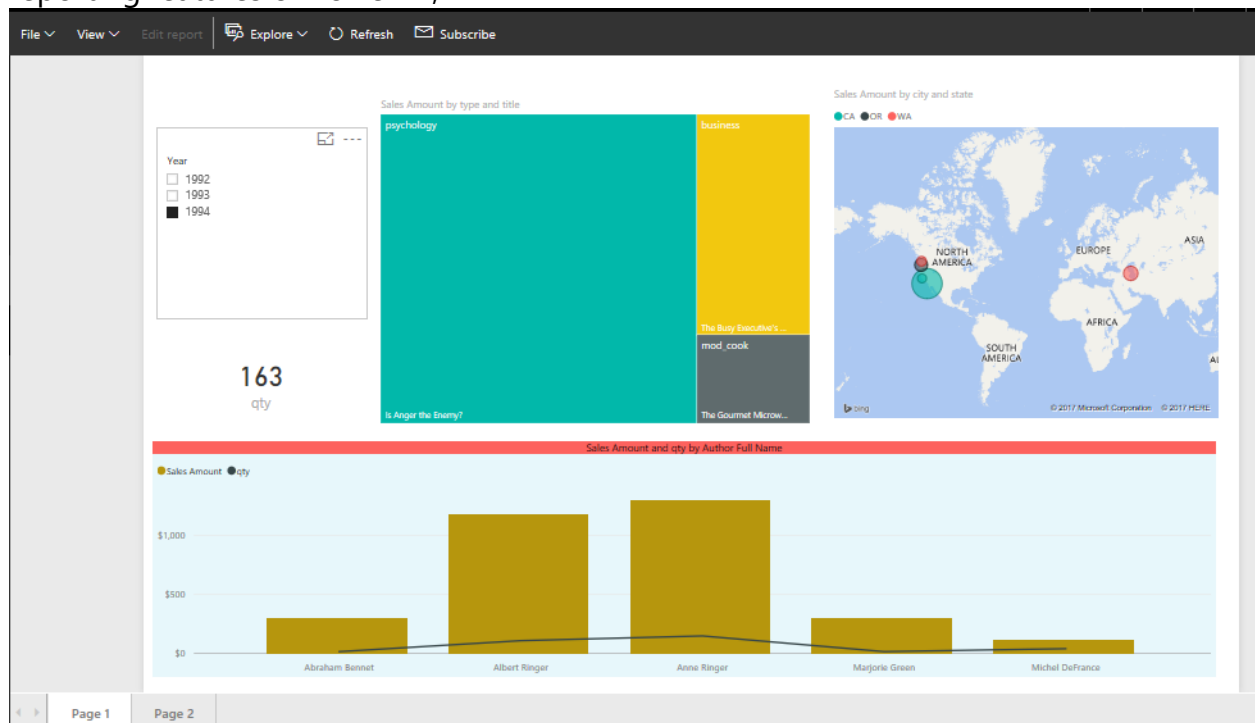
Updated

3/6/2017

After getting the content pack, all of its content (dashboards, reports, data sets) will be loaded in the Power BI account of the user.



User will be able to view dashboards, reports, and data sets with full interactive reporting features of Power BI;

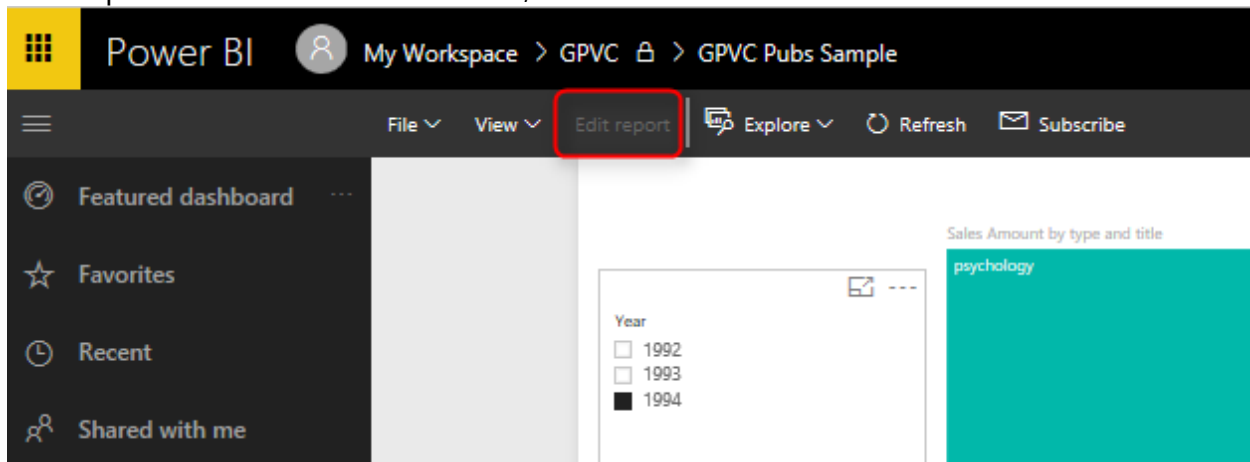


Advantages of Content Packs

Content Packs are easy way of sharing content, but their benefits aren't just this easiness. It is more because of other great features. In brief explanation, features are;

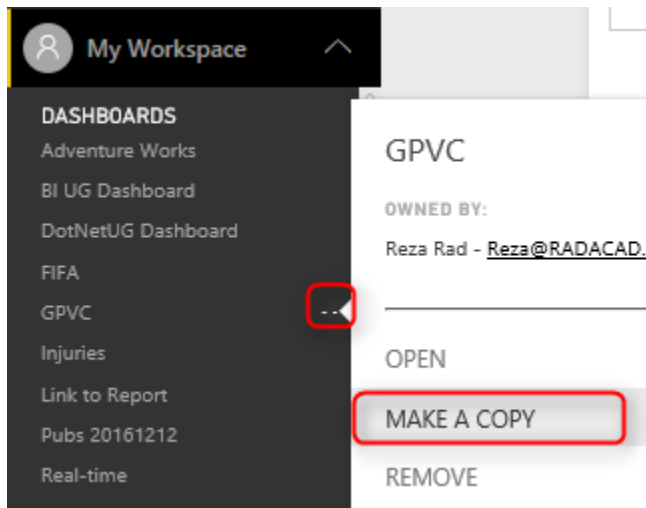
Original Content Will be Intact

None of users will be able to change the original contents in the content pack. In fact, users will have "Read-Only" access to the Power BI content shared with them. They cannot modify dashboard, Report, or data set. You can see in the screenshot below that Edit Report item is disabled for them;



Self-Service Is Possible

Although that users cannot edit the content and they have read only access. However, they can create COPY of the content, and make their own changes in their copy (which is for themselves only). This is a very important feature, because greatness of Power BI is the ability to build things self-service. And you never want to take that power from your users of course. To create copies, users just click on the content and create copies of that;



This action will generate copy version of all dashboards, reports, and data sets for the user.

Copy this content pack



















Click Save to make your own copy of this content pack. It will be a snapshot version that you can edit.

The following content is included:
DotNetUG Dashboard (Dashboard)
GPVC (Dashboard)
GPVC Pubs Sample (Report)
DotNetUG (Report)
DotNetUG (Dataset)

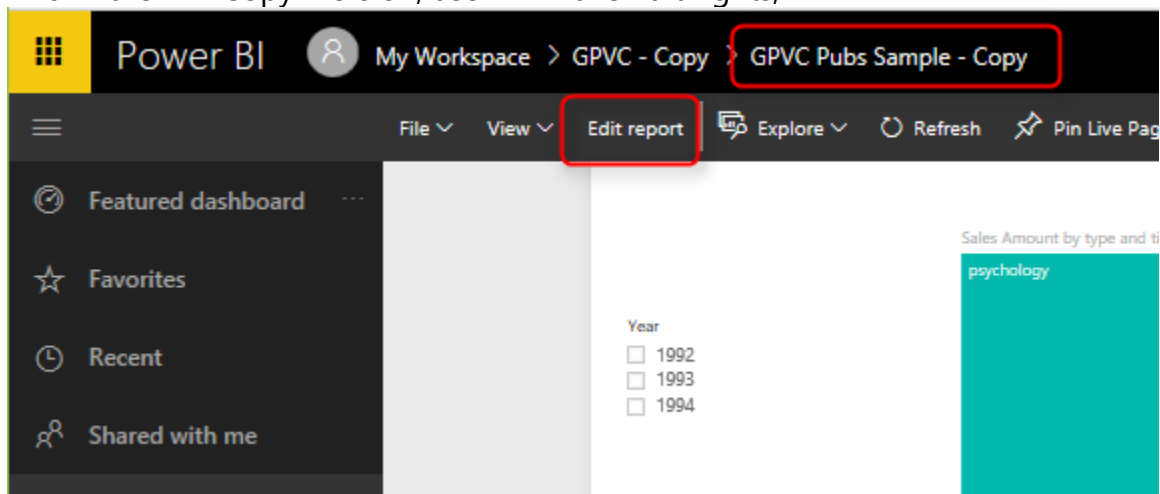
Save

Cancel

Everything will be post-fixed with – Copy.

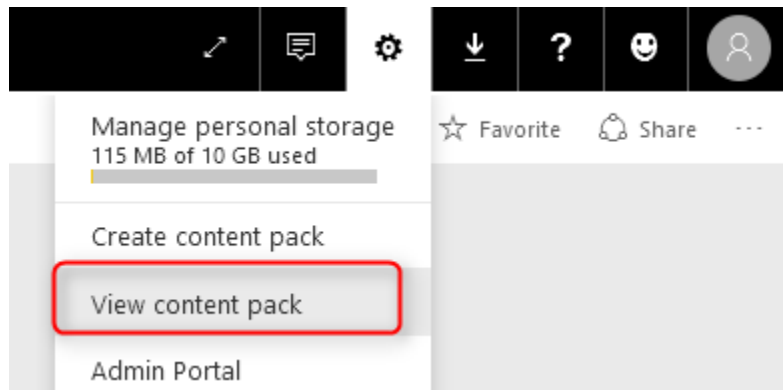
Dashboards Reports Workbooks Datasets			
NAME			ACTIONS
 ☆ BI UG Dashboard			 
 ☆ DotNetUG Dashboard *			 
 ☆ DotNetUG Dashboard - Copy *			 
 ☆ FIFA			 
 ☆ GPVC			 
 ☆ GPVC - Copy *			 

And in the “* – Copy” version, user will have Edit rights;



Managing Content Pack

As the owner of content pack, you can always to manage content pack, and configure the content you want to be shared. You can change the content, or even delete the content pack. Changes will affect immediately on user's side.



List of all content packs will be available for your in View Content Pack.

Name	Published To	Date published	Actions
Pubs 20161212	My Organization	Dec 12, 2016	Edit Delete
Difinity 2017 Content Pack	My Organization	Feb 9, 2017	Edit Delete
SQL Sat Melbourne	My Organization	Feb 11, 2017	Edit Delete
My New Content Pack	My Organization	Feb 15, 2017	Edit Delete
SQL Sat Sydney Content Pack	My Organization	Feb 18, 2017	Edit Delete
Sample Content Pack 20170306	My Organization	Mar 6, 2017	Edit Delete

Any changes will be possible, and will affect immediately on those who are using this content.

You Control the Refresh

One of the main questions I get when I reach to this part of my course, is that; Would users get refreshed data in their copy of content pack? the answer is: Yes. With using content packs, you manage the refresh on the main data set(s), and everyone using this content pack will get the refreshed data based on whatever schedule you have defined. There is no need to schedule data to be refreshed again for users.

Sharing Multiple Content All at Once

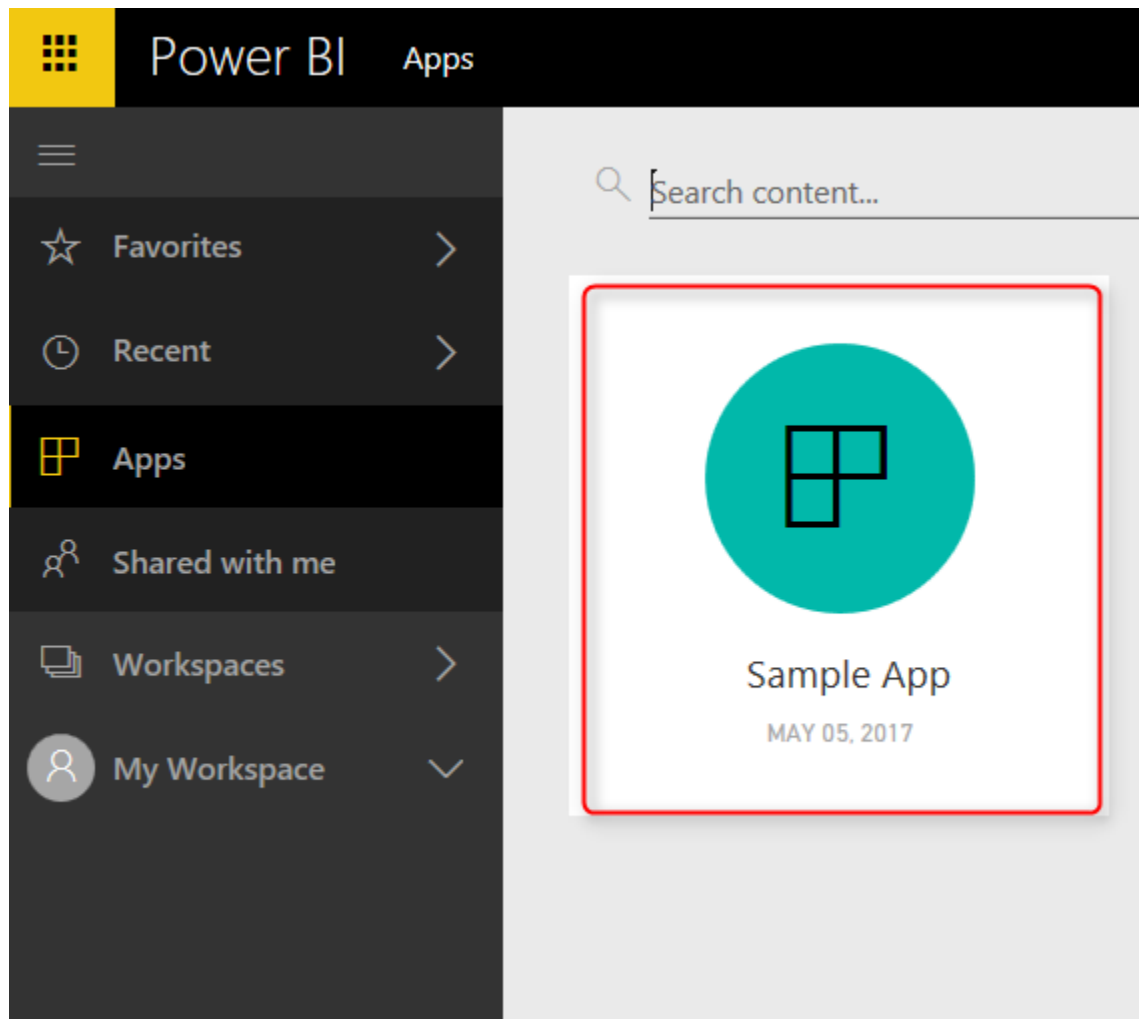
The last but not least advantage of this method, is that you can simply share 100s of reports, dashboards, and data sets with others with a single content pack.

Summary

In Summary; Organizational Content packs are best way to share Power BI content with users. You will have full governance for your content; you will set up refresh once, and no one would be able to change your original content. Users on the other hand will have access to all content as read-only, and they can make their own copy of the content, if they want to play with self-service features of Power BI. You can always control and change contents of the content pack from View Content Packs section in your Power BI account. This method is highly recommended for sharing content with end users.

Ultimate Sharing Strategy: Power BI Apps

Published Date: May 5, 2017



I have previously written about some of the methods of sharing content in Power BI, such as Work spaces, Publish To Web, and Content Pack. [Latest update of Power BI](#) includes stunning update for sharing mechanism which is the best of all for sharing. In this post I'll explain everything about new Power BI Apps; a mechanism to share the content in Power BI in a way that has security and governance together. If you want to learn more about Power BI; Read [Power BI book from Rookie to Rock Star](#).

Power BI Sharing Methods

Till now there were about 6 methods of sharing content in Power BI, including:

- Simple Dashboard Sharing
- [Publish To Web](#)
- Embed in SharePoint Online
- [Power BI Embedded](#)
- [Power BI Work Spaces](#)
- [Power BI Content Packs](#)

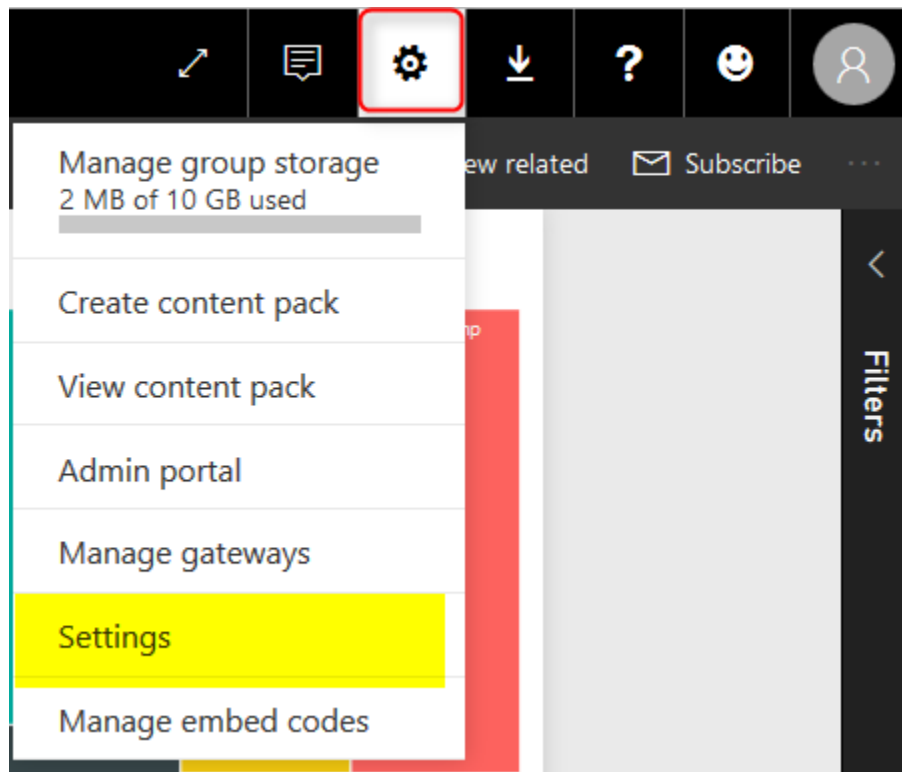
I have written about some of these already (follow links above), and will write about the rest soon. [Yesterday](#), Microsoft announced preview version of Power BI Apps, which is a new method of sharing. This is an enhancement version of two methods previously: Work spaces, and Content Packs together!

Power BI App

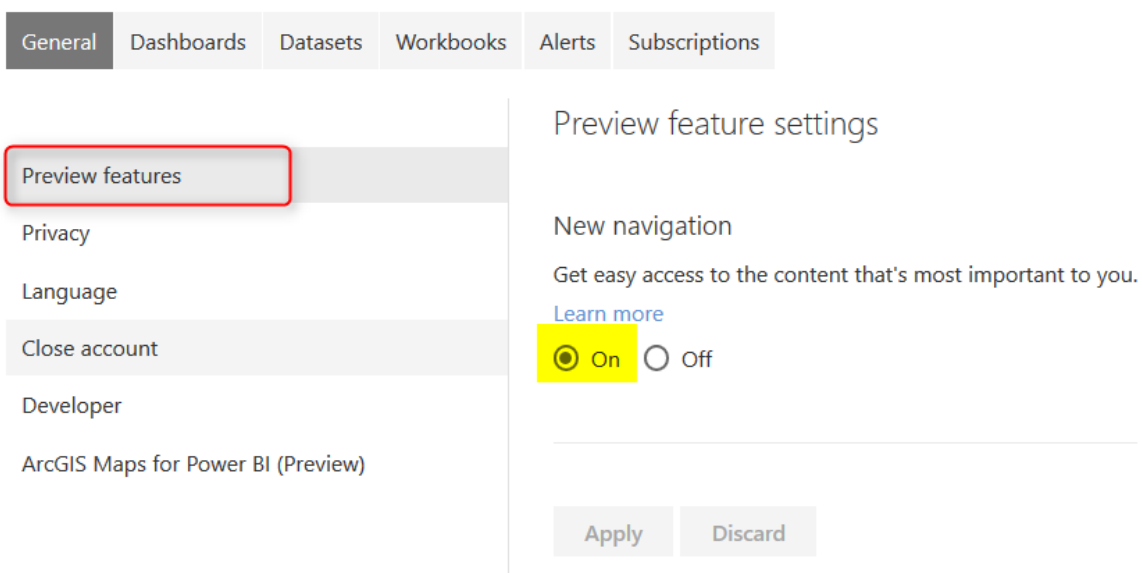
A Power BI app is a grouped content of Power BI (including Dashboards, Reports, Work books, and Data Sets), which is shared with organization or a group of people within organization. Let's go through creating and consuming an App to see how it works in action.

Power BI Service in Preview Mode

At the time of writing this blog post, Power BI App is just released as preview and is not yet available as general available. You need to turn on the Preview mode of Power BI service to be able to work with it. To turn Power BI in the preview mode, Go to Setting in Power BI website;



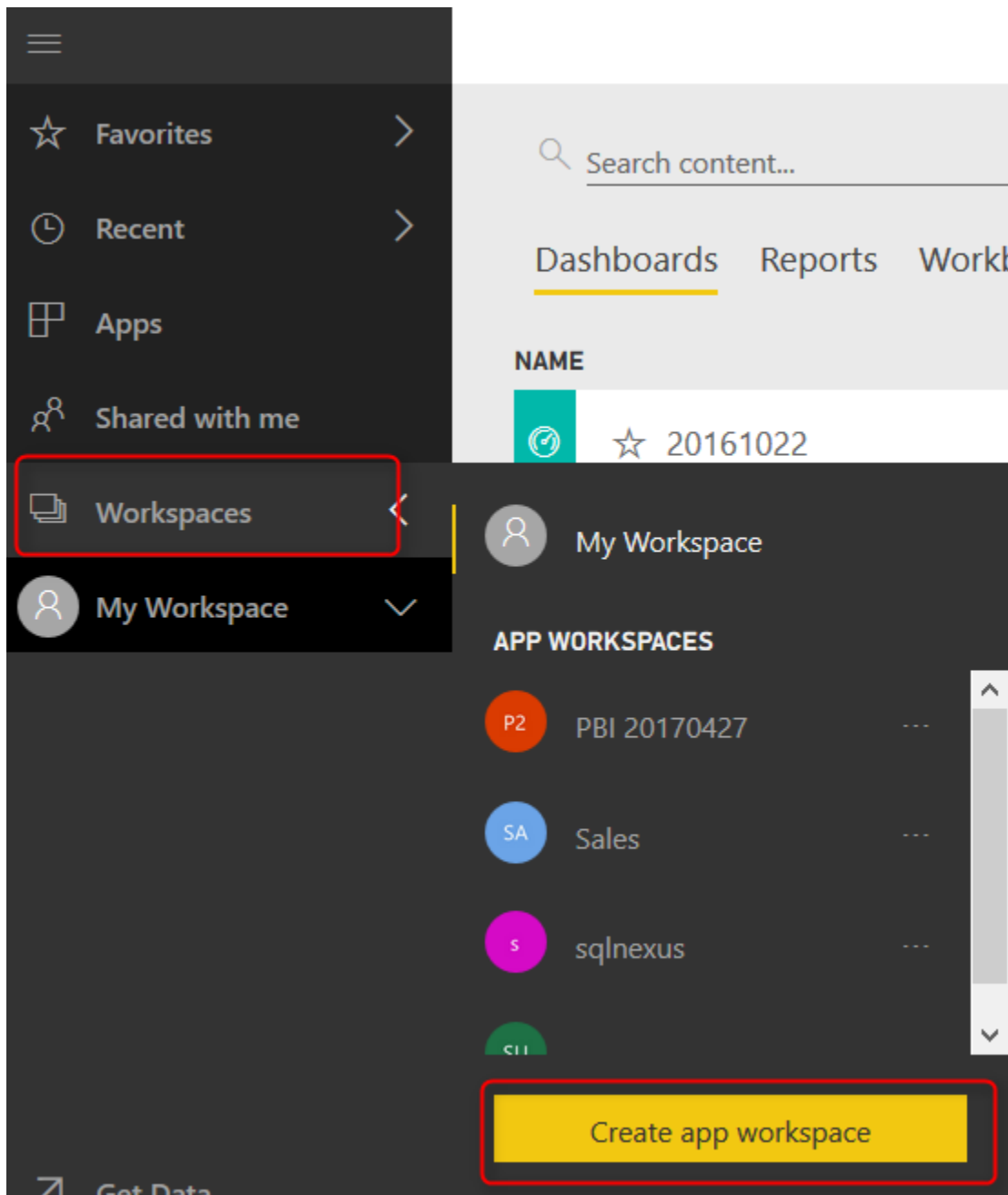
In Setting page, Go to Preview features section, and turn this feature on.



App Work Space

To start creating an app, you'll need an App Work Space. This is a new generation of work spaces in Power BI. content that you will have in this work space will be the entire

App for the work space. If you already have a [work space](#), you can use that as an App Work Space.



app work space is like a folder that is shared with a group of people. Set a name for the app work space. You can specify if users has access to this work space via edit or read only. Please note that this way of access is only for people that you add them in the group directly, not for users of this App.


Create a group

Name your group

Sample App

Group ID

sampleapp

 Available

Private - Only approved members can see what's inside

Members can edit Power BI content

Add group members

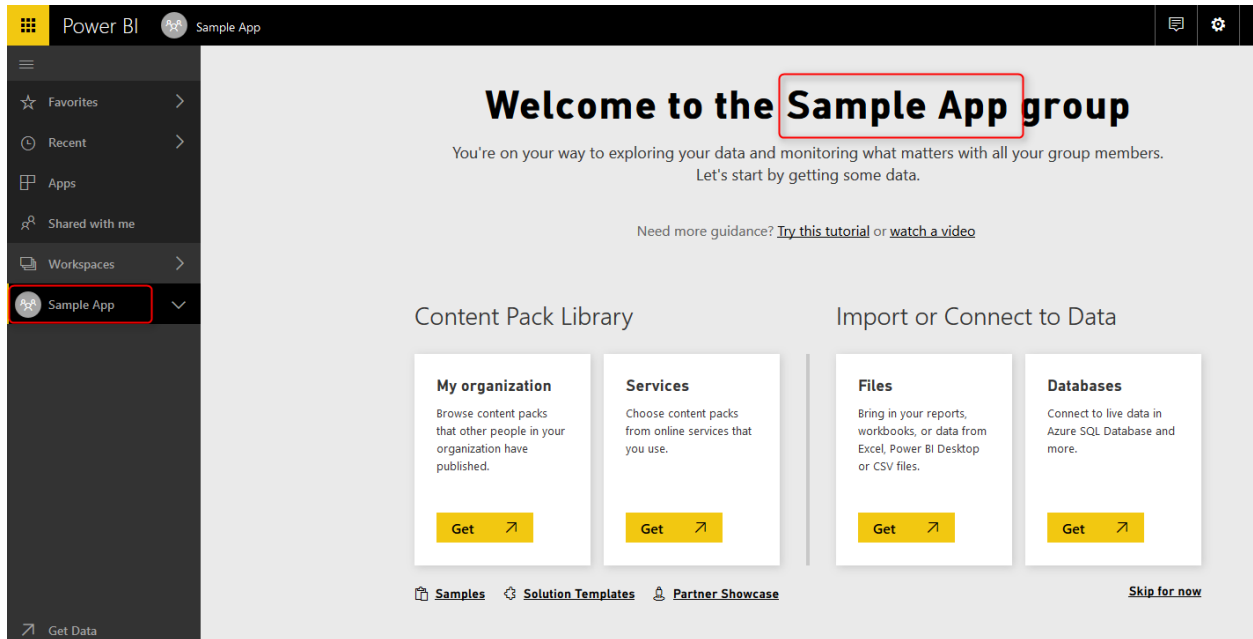
Enter email addresses

Add

Save

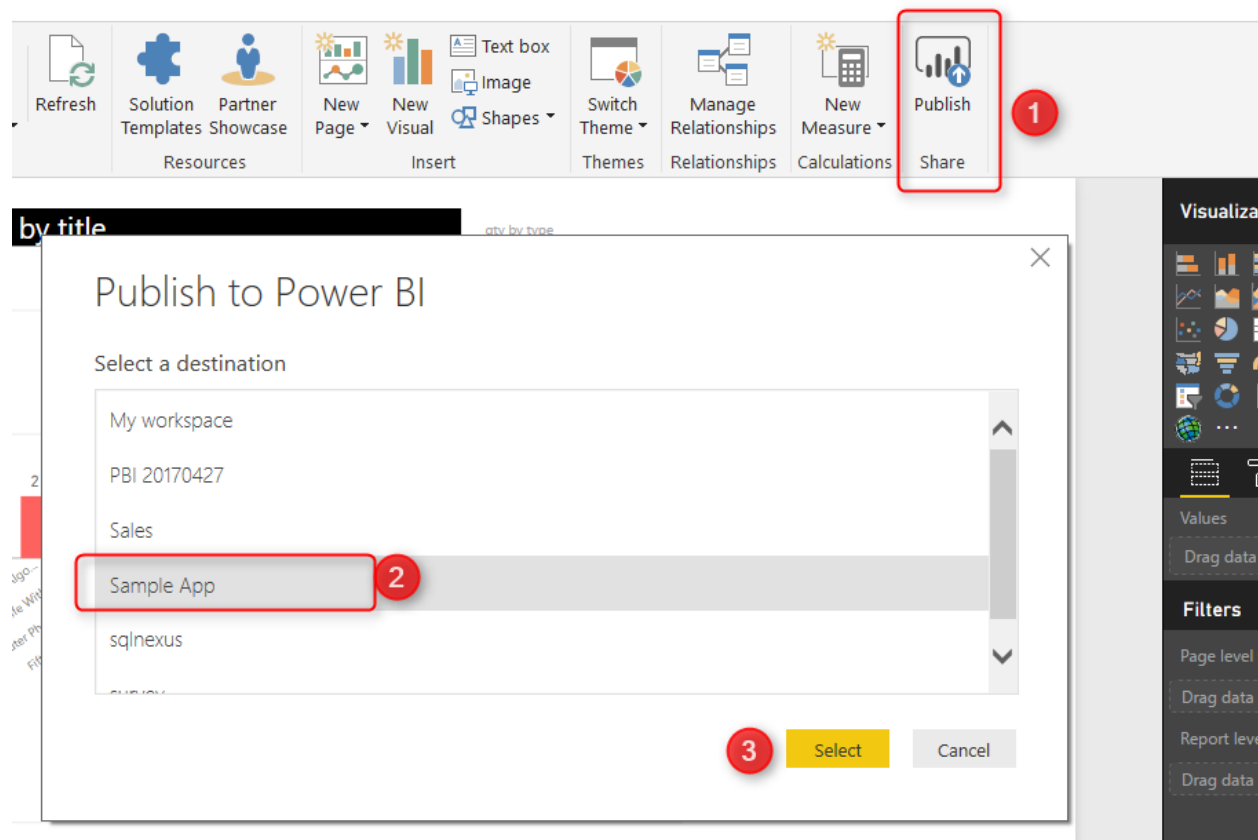
Cancel

After creating the app work space, you should see a new work space (folder) which is blank at first



Publish Content to App

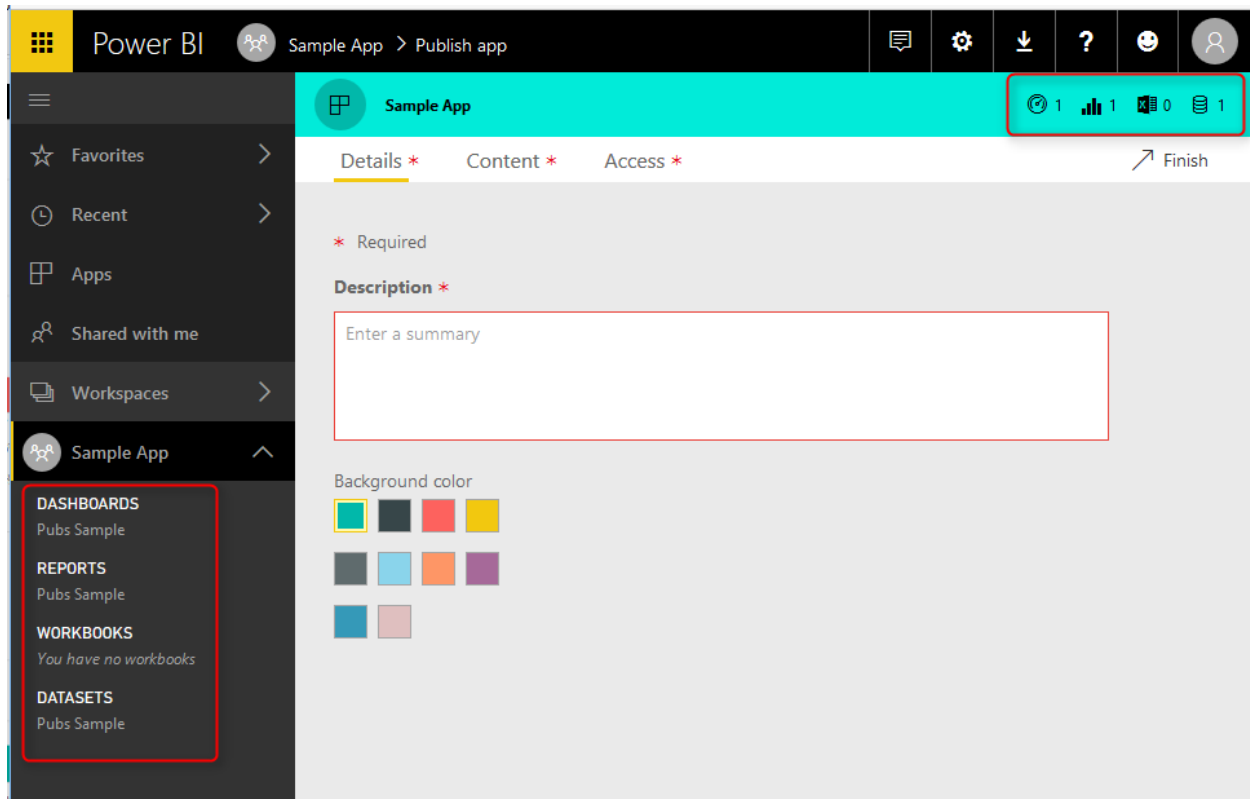
For publishing content to app, just simply publish it to the app work space. Open Power BI Desktop, go through Publish, and select the app work space created.



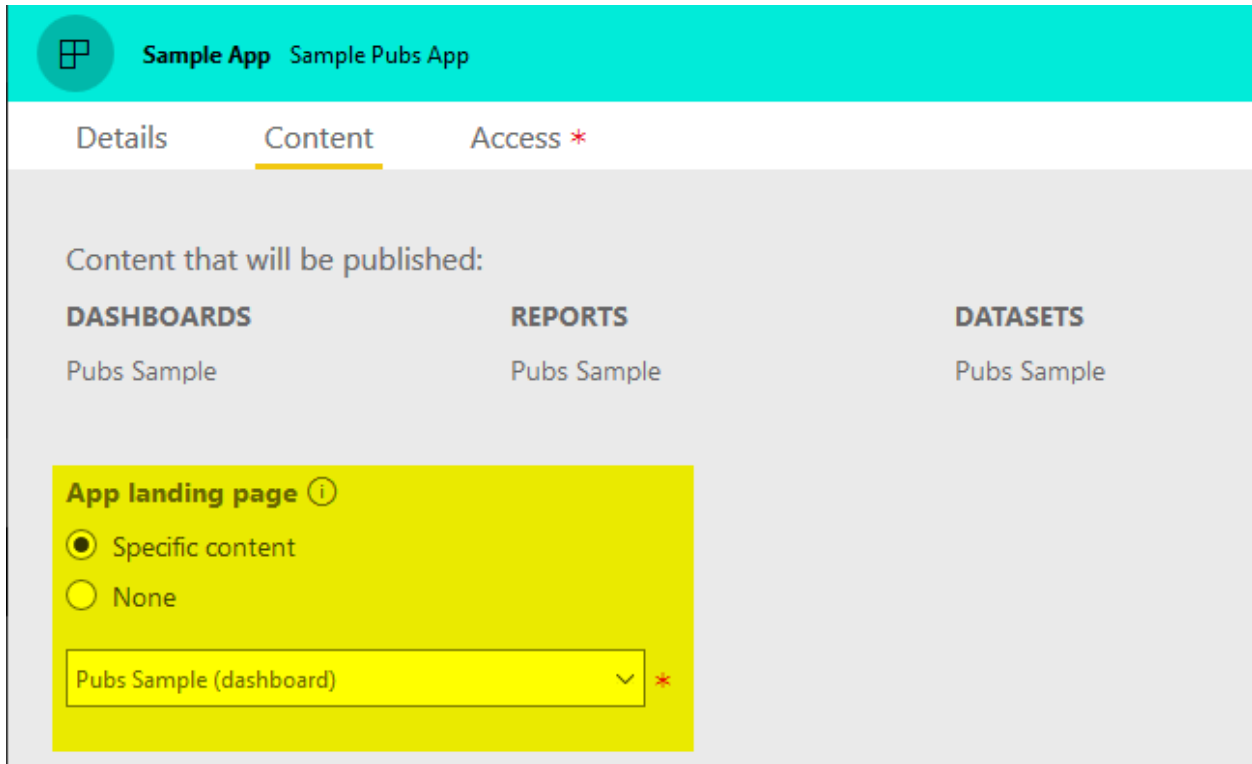
After publishing content into the new app work space, you should be able to publish App from there.

Publish App

You can easily publish an app from the app work space. Just click on the work space, and then from top right hand side, select Publish App. Set a description and color for the app. You can also see how many reports, dashboards, work books, and data sets are part of this app.

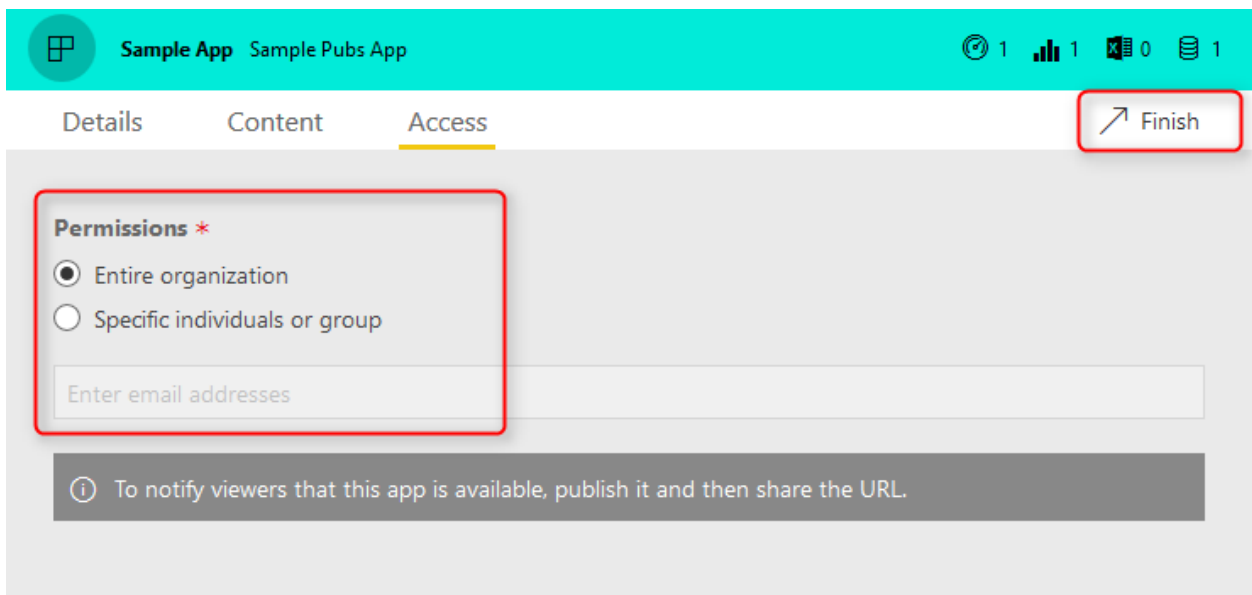


In the next tab, you will see the content. and you are also able to select the landing page for the content pack! This is a very good feature, that we haven't had before. Now with Power BI Apps you can set a landing page for people in the group.



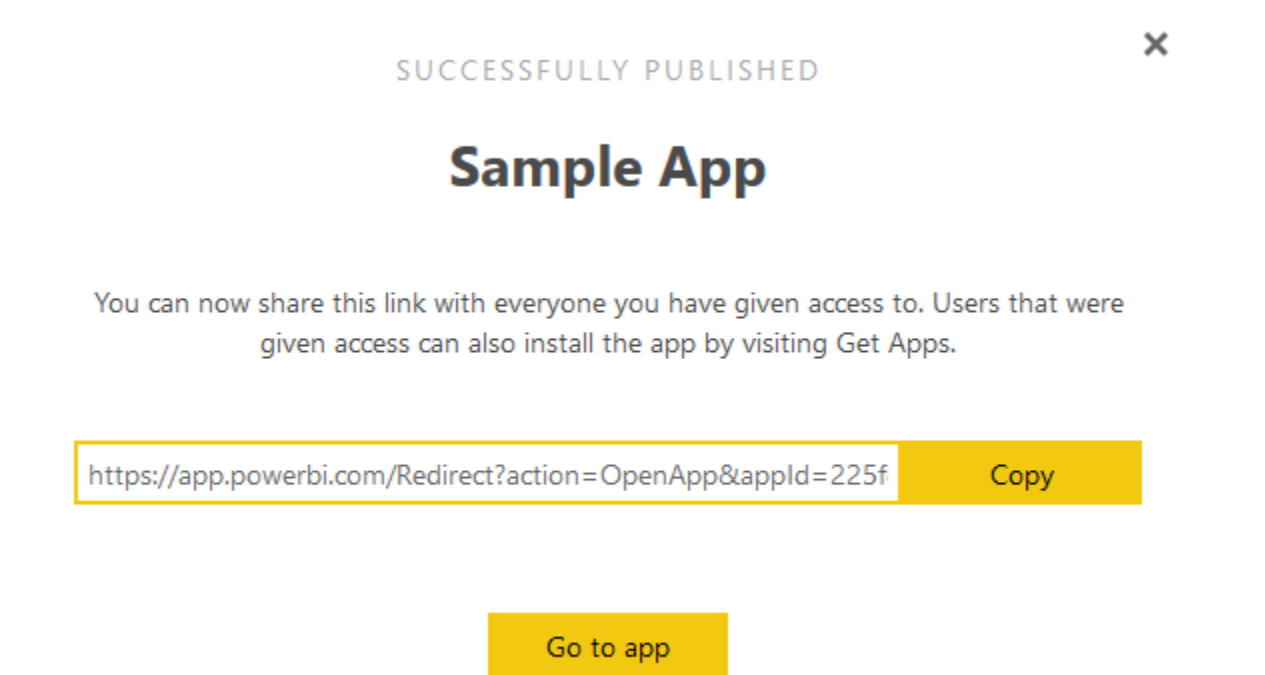
The screenshot shows the 'Content' tab of the 'Sample Pubs App' configuration page. The top navigation bar is teal with a grid icon and the text 'Sample App' and 'Sample Pubs App'. Below the bar are three tabs: 'Details', 'Content' (which is underlined), and 'Access *'. The main content area is light gray and contains the heading 'Content that will be published:'. Below this heading are three columns: 'DASHBOARDS', 'REPORTS', and 'DATASETS'. Each column contains the text 'Pubs Sample'. Below these columns is a yellow box titled 'App landing page ⓘ'. Inside this box are two radio buttons: 'Specific content' (which is selected) and 'None'. Below the radio buttons is a dropdown menu showing 'Pubs Sample (dashboard)' with a downward arrow and a red asterisk to its right.

You can also set group of people that you want the app to be shared with (entire organization or specific groups), and then click on Finish.



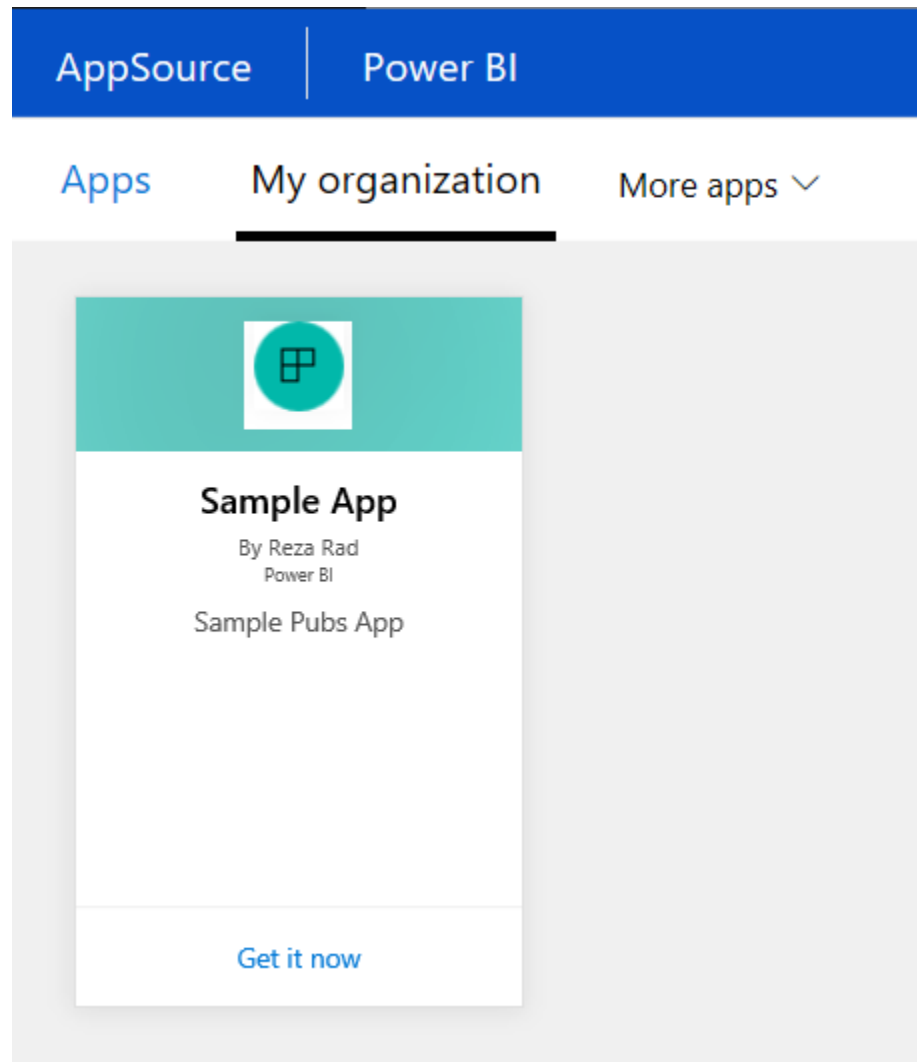
The screenshot shows the 'Access' tab of the 'Sample Pubs App' configuration page. The top navigation bar is teal with a grid icon and the text 'Sample App' and 'Sample Pubs App'. On the right side of the bar are four icons: a checkmark, a bar chart, a document with an 'X', and a list icon, each followed by the number '1'. Below the bar are three tabs: 'Details', 'Content', and 'Access' (which is underlined). On the right side of the 'Access' tab is a red-bordered button with an upward arrow and the text 'Finish'. The main content area is light gray and contains a red-bordered box titled 'Permissions *'. Inside this box are two radio buttons: 'Entire organization' (which is selected) and 'Specific individuals or group'. Below the radio buttons is a text input field with the placeholder text 'Enter email addresses'. Below the 'Permissions' box is a gray box with an information icon ⓘ and the text 'To notify viewers that this app is available, publish it and then share the URL.'

After publishing the app, users will be able to access it immediately. However, you can share the link with them to make it easier for them to access;

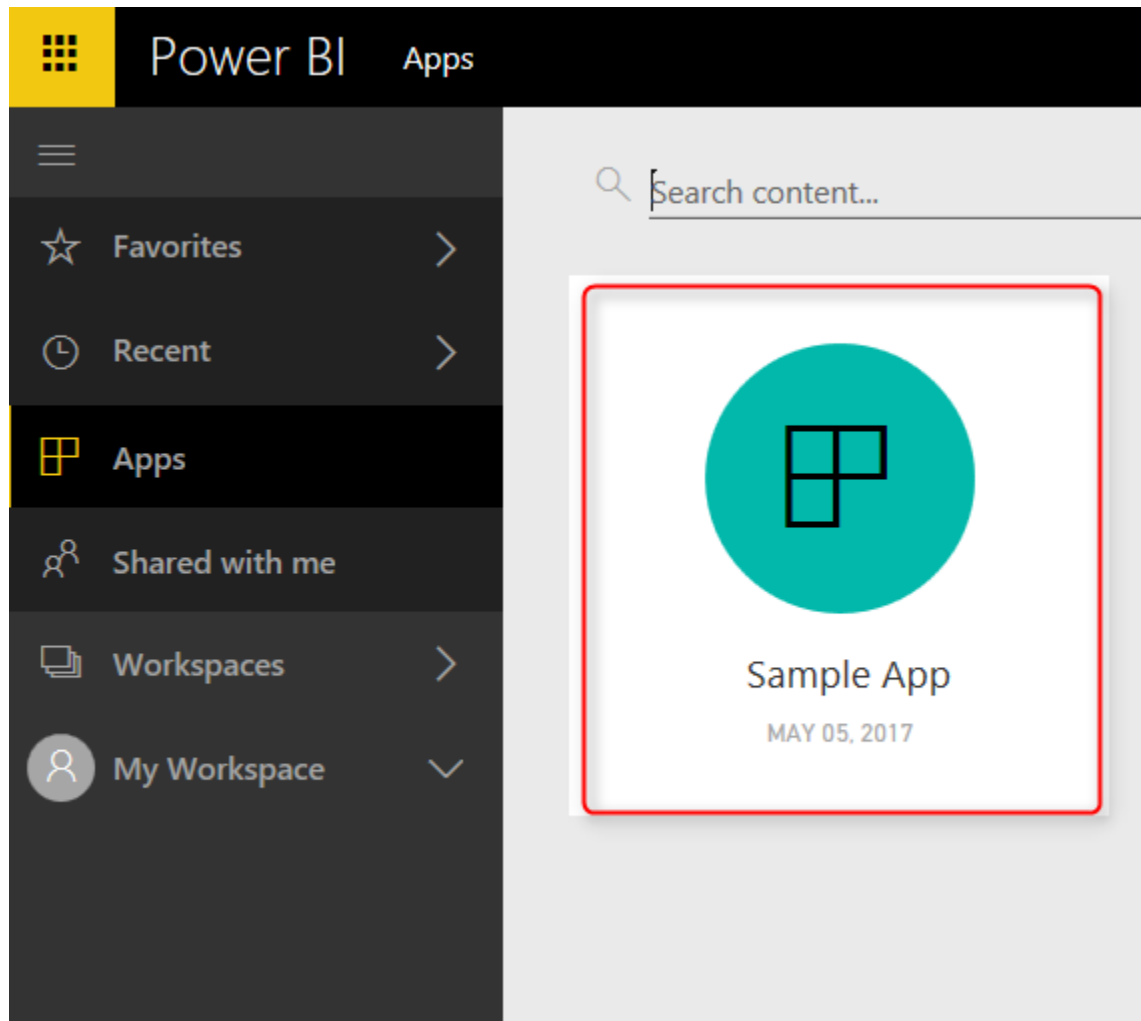


Getting App

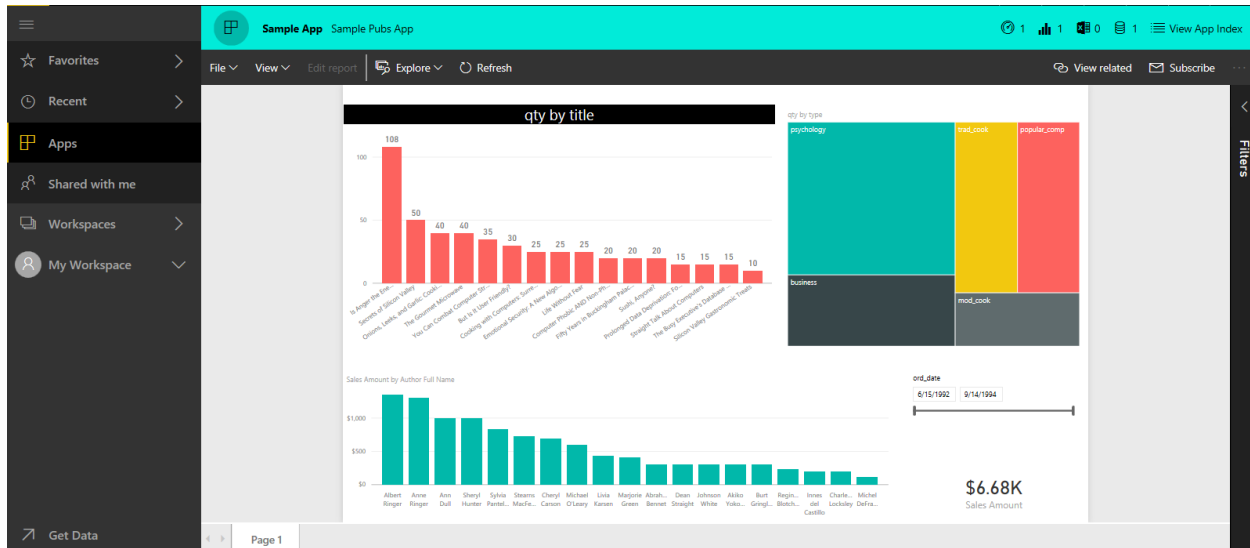
Users can go to their Power BI account page, and click on Apps, and then Get Apps. They will be able to see all apps shared with them.



Users can simply Get the app, and use it right away.

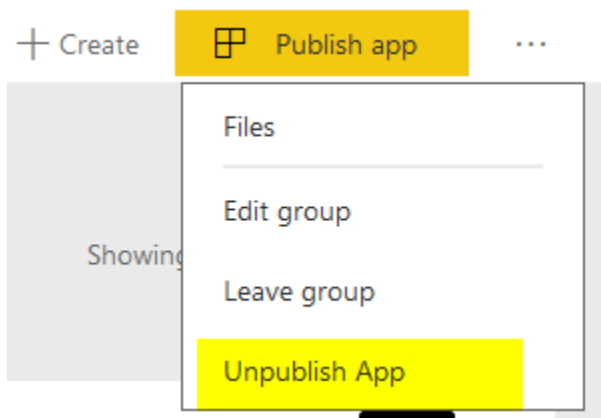


with clicking on the app, users simply will be redirected to the landing page (if specified), and can explore the content in the app.



Changes in the App

You can apply any changes you want on the content in the app work space. This will NOT affect users, until you "update app". With the new update they will get the updated content. You can also "Unpublish" the app.



My Thoughts on Power BI Apps

Power BI Apps comes to replace content pack and work spaces with one better solution, and it is very promising. With this new combination you will have power of both; You can use work spaces to group the content together, and give edit access to your power users. and you can also use content pack to share the content to the end users.

Power BI App is a very new feature, and many updates will come for this in the near future, including;

20. Disassociating from Office 365 groups.
21. Push app to users without the need for them to install (get) it.
22. Selective content publish from app work space.
23. Copying content between work spaces.
24. External Sharing

Above list is based on what Power BI team's blog post mentioned as the [road map for Power BI apps](#). However, I have some thoughts that I think would be great features for Power BI apps, and users will get lots of benefits from it. Here is my list:

34. Power BI Apps is not full replacement of Content Pack. Users cannot Make a Copy as they could in content pack. You might say; what is the benefit of copying? The answer is: Power BI is all about self-service. there are always some users who want to be able to create their own version of the report. With [Content Pack](#) that was simply possible. They could make a copy, and change their copy without touching the original report. ([read this blog post to learn how](#)). with the new app; they cannot change anything. they cannot make a copy. or if they are part of the group with edit access to the work space, then they CAN CHANGE the original! which is not good. It is either too much power for them, or nothing.
35. Immediate need for the app work space administrator. At the moment, users who are not admin of the group, but only have Edit access, are able to publish app! This is too much access. Consider a situation that users are using a published app, and suddenly someone by mistake updates the app! It is critical that the admin of app work space be the only person who can publish the app. Or at least can give this permission to specific group of people.

These are critical needs for Power BI Apps in my opinion. And I believe Microsoft Power BI Team will consider them and implement them quickly (as they've done already with many other suggestions of the community). What is your opinion about it? Do you think Apps would have been much better with some other features? name it in the comments area.

One Data Set, Multiple Reports; Power BI Next Big Step Forward

Published Date: April 10, 2017

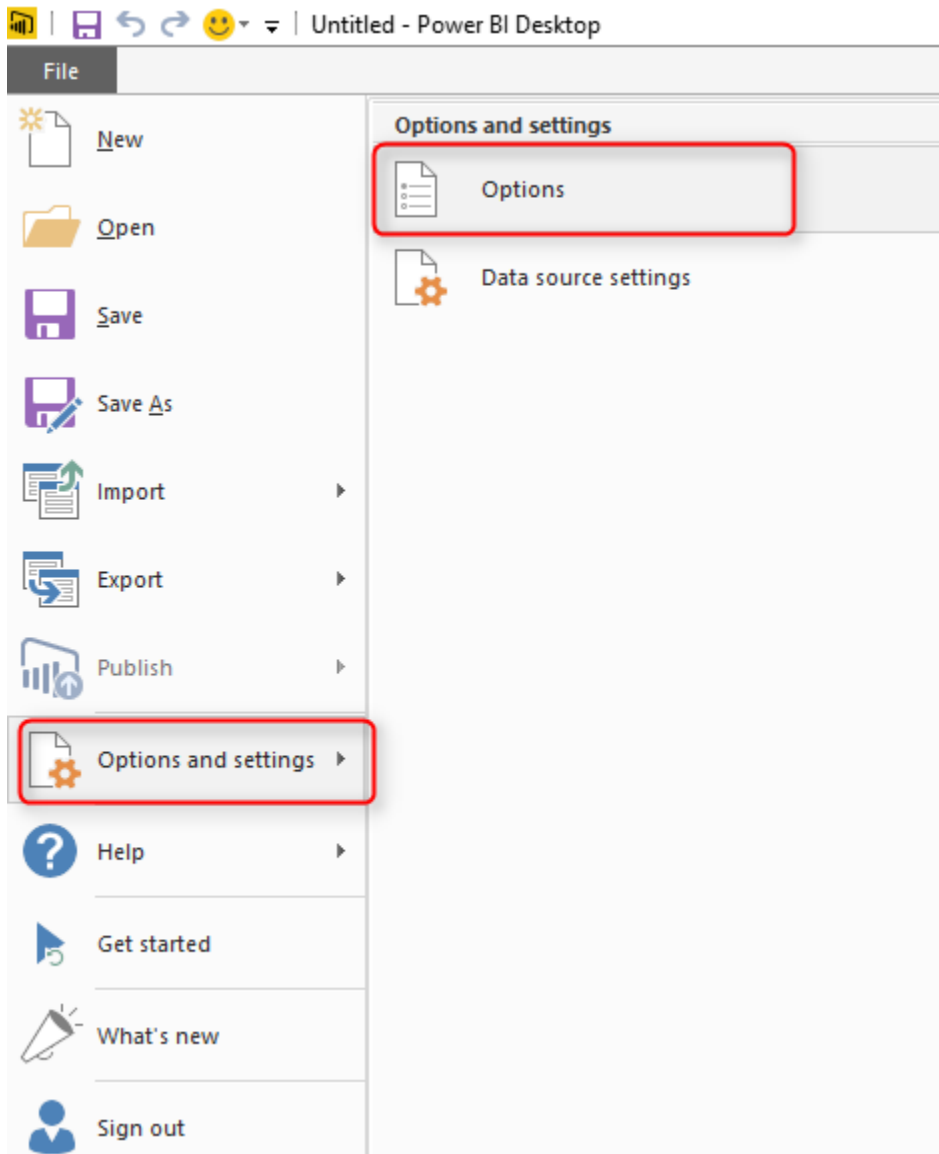


You can use an existing data set with LiveQuery or DirectQuery connection as the shared data source for multiple Power BI reports. You can also create multiple reports from Power BI Service (website) connecting to the same data set. However, this [new feature in Power BI Desktop version April 2017](#) is a big step forward; You can now create reports in Power BI Desktop pointing to a central data set. This is a great step towards making your data sets centralized and using it consistently from many reports. In this post I'll show you how this feature works. If you want to learn more about Power BI; read [Power BI book from Rookie to Rock Star](#).

Get Data from Power BI Service

You need to install Power BI Desktop version April 2017 or later versions to have this feature. At the time of writing this post this feature is still preview, and you need to enable it through going to path below:

File -> Options and Settings -> Options



In the Options window go to Preview Features, and enable this option;

Options

GLOBAL

Data Load
Query Editor
DirectQuery
R scripting
Security
Privacy
Updates
Usage Data
Diagnostics
Preview features
Auto recovery

CURRENT FILE

Data Load
Regional Settings
Privacy
Auto recovery

Preview features

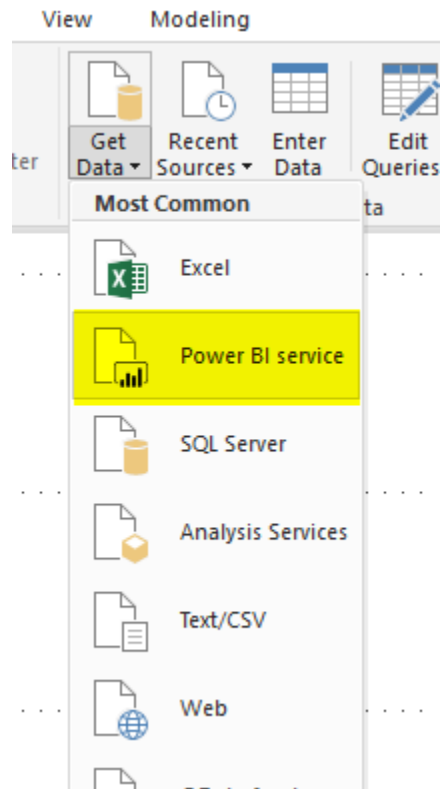
The following features are available for you to try in this release. Preview features might change or be removed in future releases.

- ☐ Impala [Learn more](#)
- ☐ Snowflake [Learn more](#)
- ☒ Shape map visual [Learn more](#)
- ☒ Custom report themes [Learn more](#)
- ☒ Enable cross filtering in both directions for DirectQuery [Learn more](#)
- ☒ ArcGIS Maps for Power BI [Learn more](#)
- ☒ New matrix visual [Learn more](#)
- ☒ Numeric range slicer [Learn more](#)
- ☐ Spanish language support for Power BI Q&A [Learn more](#)
- ☒ **Power BI Service Live Connection** [Learn more](#)
- ☒ Quick measures [Learn more](#)

OK


Cancel

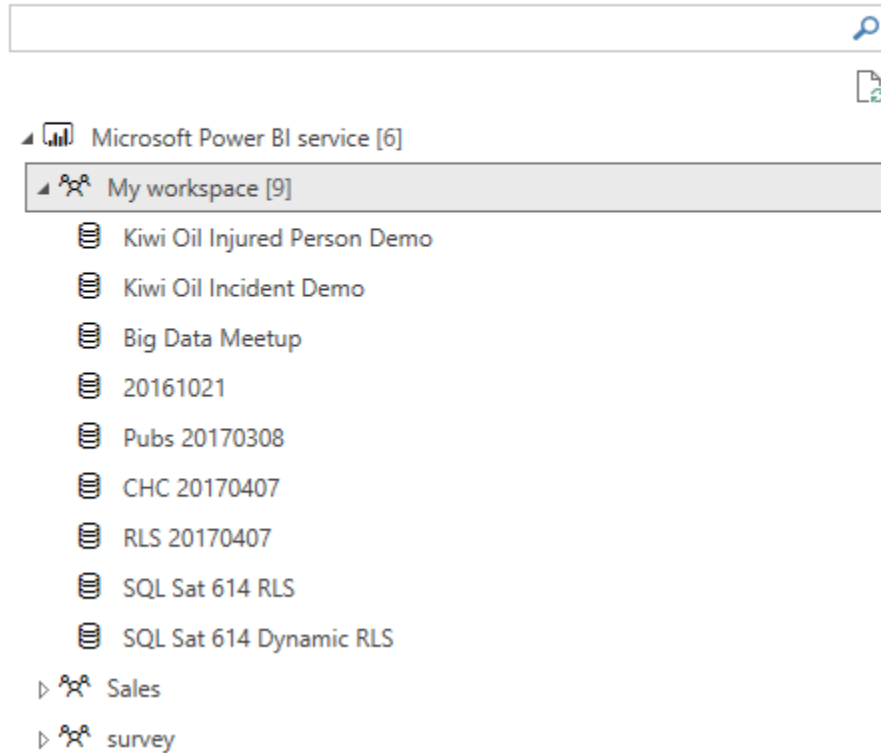
You will need to close Power BI Desktop and Open it. after re-opening Power BI Desktop, in Get Data section you will see the new option: from Power BI Service



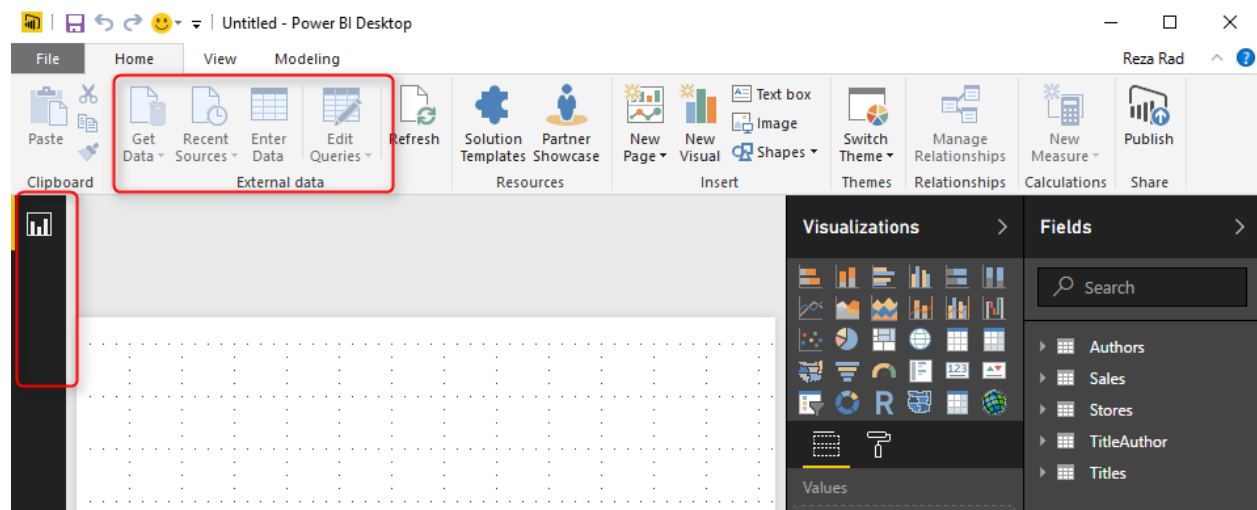
For accessing to Power BI service you need to login to your Power BI account from Power BI Desktop, and then you will see list of work spaces under your account and data sets under each.

Power BI service

Reza Rad 
Reza@RADACAD.onmicrosoft.com
[Switch account](#)



After selecting a data set, a LIVE QUERY connection will be created to that data set, and your Power BI Desktop will turn into a mere data visualization tool (exactly similar to having a LiveQuery connection to SSAS). You cannot edit the data source, you cannot change relationships, you cannot create new measures or calculations. However, you can create all visualizations you want based on this model.



As you can see in screenshot above Get Data from other data sources will be disabled, and you will have only report tab available in Power BI Desktop.

If you want to make any changes in the model, you need to open the Power BI Desktop file that contains the model, and make changes from there.

What This Change Means?

At first you might think; what is the big benefit of having this feature? the short answer is to have reports based on a centralized data set. the long answer is here;

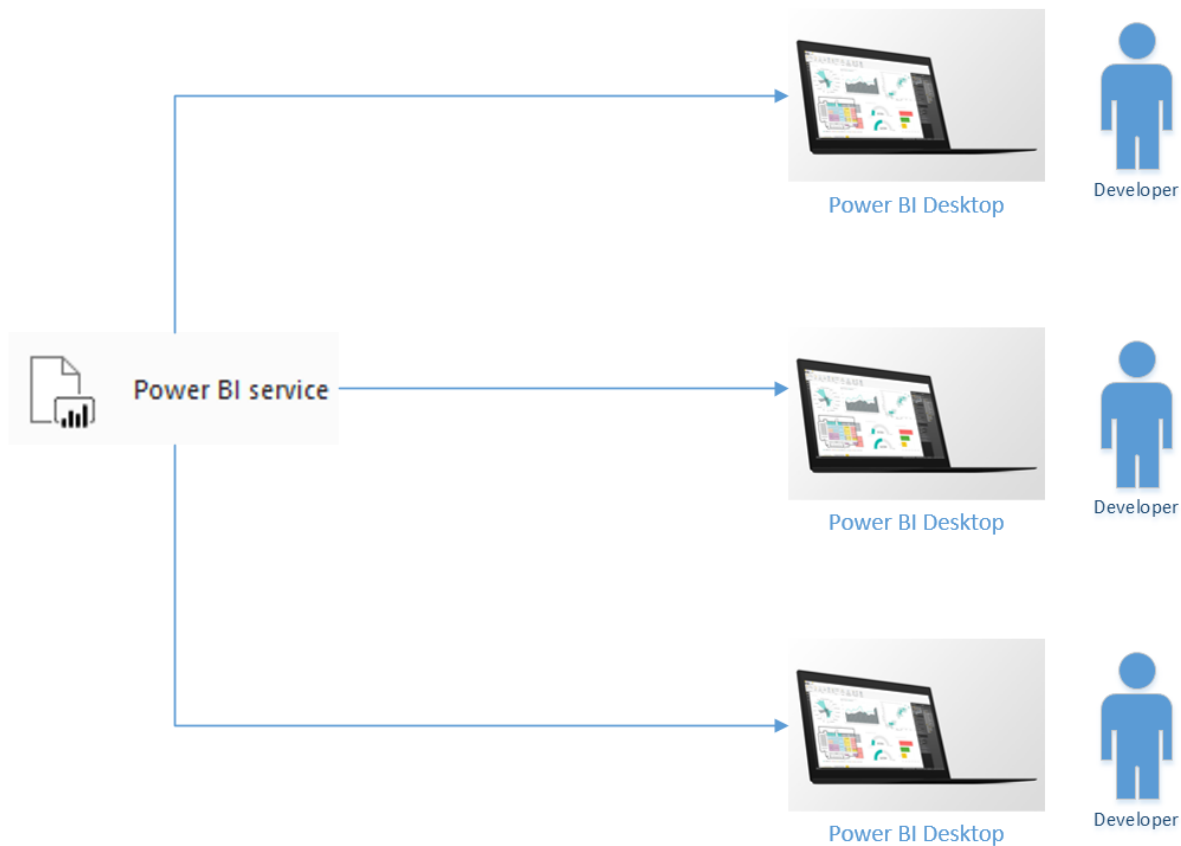
Central Data Model

Previously if you could use Power BI for getting data from multiple data sources, combining them, data mashup and transformation, building a great model with hierarchies and calculations, and build visualizations based on it, all in one file. If you wanted to create another file with the same model, you should copy the existing one, and make changes in it, and for maintaining you would have a headache of maintaining multiple models to be exactly similar! Now you can simply build a model once, and re-use it multiple times.



Multiple Development Environment

Developing everything in one Power BI file is good for one developer. However, when multiple developers want to work on the same file, then multiple files are needed. With this new feature the visualization layer can be easily separated from the model. Multiple Power BI files can be pointed to the same model, and multiple developers can work on it.



Report Level Sharing

You can share reports in multiple ways, but all of those methods share the whole report not particular pages. If you want to share some pages with some users, and others with other users, you will need to create multiple reports, or in other words; multiple Power BI files. With the new change you can easily switch to this method, it will not cost you anything. with the old method you had to copy everything, and redundancy problem will happen again with big maintenance task overhead.

Limitations

You've realized how good is this feature, and how the next generation of Power BI reports will shape in the future. You might think what is the limitation of this feature? This feature is still preview, and there will be lots of updates for it in the future, however at the time of writing this post, here are limitations:

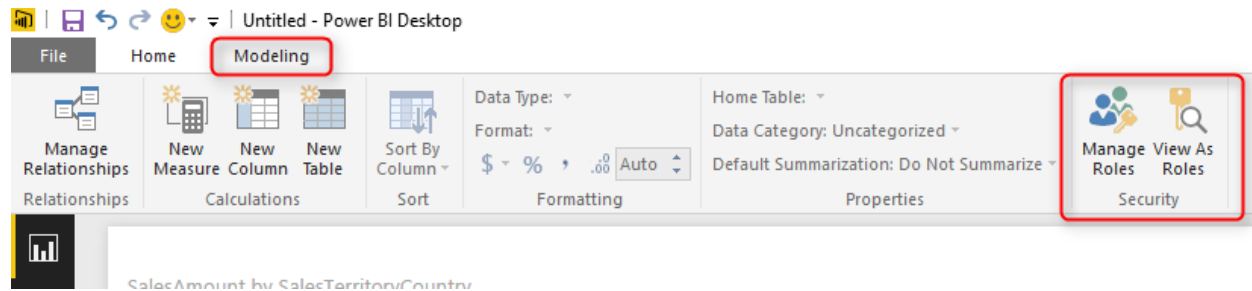
- One Data Source: You can not combine another data source with an existing model. If you want to do that, go back to the original model, and bring the data set there.
- Change of Model is not possible; You can change the model only through the original file. other new files will be just connection to that data source for visualization.
- LiveQuery Data Sources won't be listed; if you have SSAS LiveQuery connection, you won't be able to connect to it in this way. Why? because you can simply connect to SSAS itself with LiveQuery, you don't need an extra layer!
- Streaming Data Sets are not supported; If you have a streaming data set from REST API, Azure Stream Analytics, or PubNub, you cannot access it from here. I hope this feature be enabled very soon 😊 it would be the second big step forward in this area.

Summary

In summary this features is a game changer in the architecture of Power BI solutions. You can now use a centralized model, which is isolated from visualization layer. For very large data sets, and enterprise solutions, still SSAS live connection might be the best because of the size of data. However, if the size is not issue, then Power BI can be easily used as a centralized data model approach. Because of this big change I give 5-star rank to this month update of Power BI, well done team 😊

Row Level Security Configuration in Power BI Desktop

Published Date: July 2, 2016



Power BI team released the feature for creating Row Level Security in Power BI website in March, and I've written a [blog post](#) about how to use it. In June's update for Power BI Desktop, this feature took an step further, and is available in the Power BI Desktop. Yes, Row level security configuration is now part of your Power BI model, and will be deployed with your model. It won't be over-write with every publish from Power BI Desktop. In this post I'll show you this feature in details. If you like to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

What's Good About Row Level Security in Power BI Desktop?

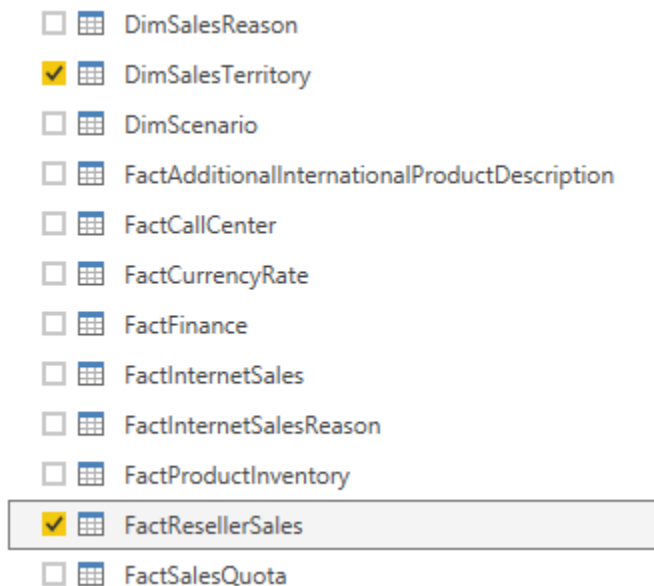
Row Level security is about applying security on a data row level. For example sales manager of united states, should only see data for United States and not for the Europe. Sales Manager of Europe won't be able to see sales of Australia or United States. And someone from board of directors can see everything. Row Level Security is a feature that is still in preview mode, and it was available in Power BI service, [here](#) I mentioned how to use it in the service. However big limitation that I mentioned in that post was that with every update of the report or data set from Power BI Desktop, or in other words with every publish from Power BI Desktop, the whole row level security will be wiped out. The reason was that Row Level Security wasn't part of Power BI model. Now in the new version of Power BI Desktop, the security configuration is part of the model, and will be deployed with the model.

Prerequisite

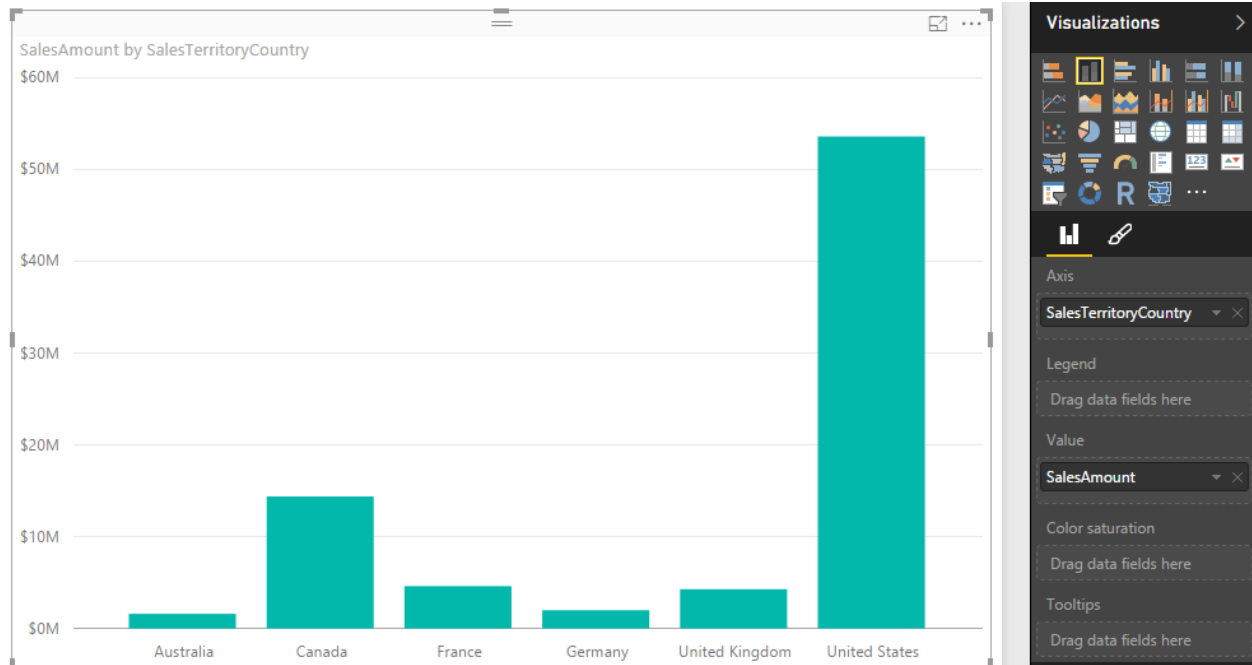
For this example I will use AdventureWorksDW2014 sample database. You can download it from [here](#) and install it. Also download June edition of Power BI Desktop.

Create Sample Report

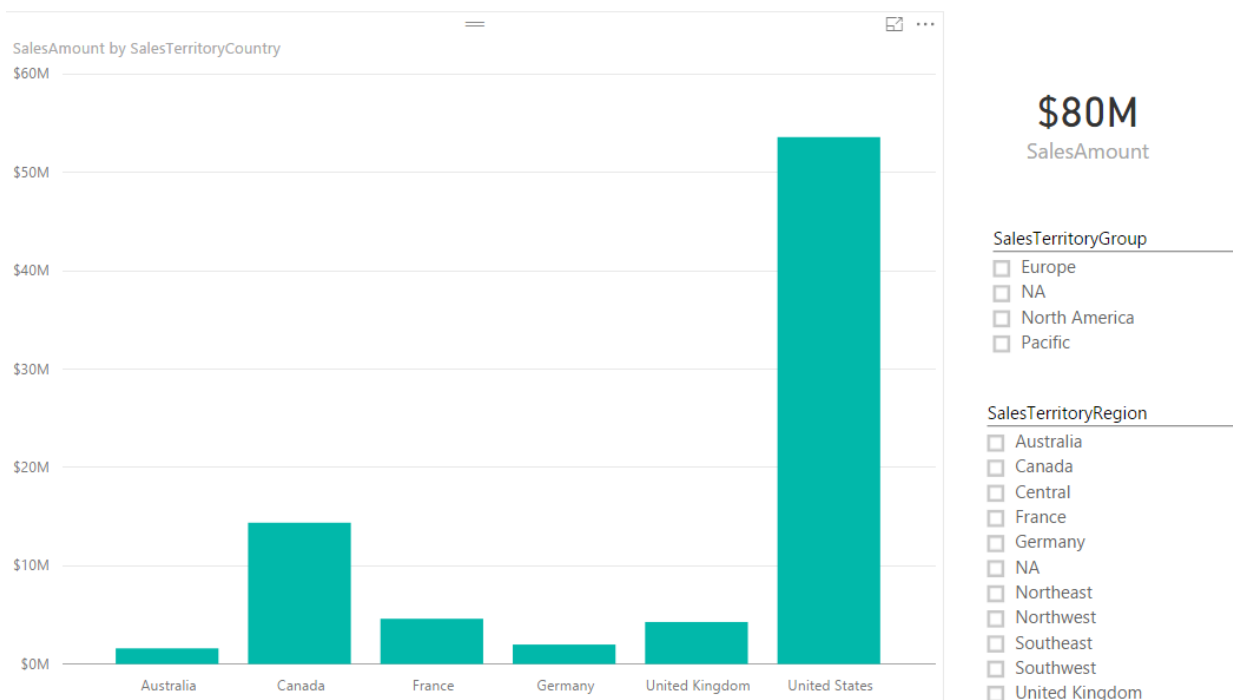
Let's start with creating a sample report in Power BI Desktop from AdventureWorksDW2014 database. I only select DimSalesTerritory, and FactResellerSales for this example;



without any changes in Power Query editor, let's load it in the report, and build a simple column chart with Sales Amount (from FactResellerSales), and Country (from DimSalesTerritory).



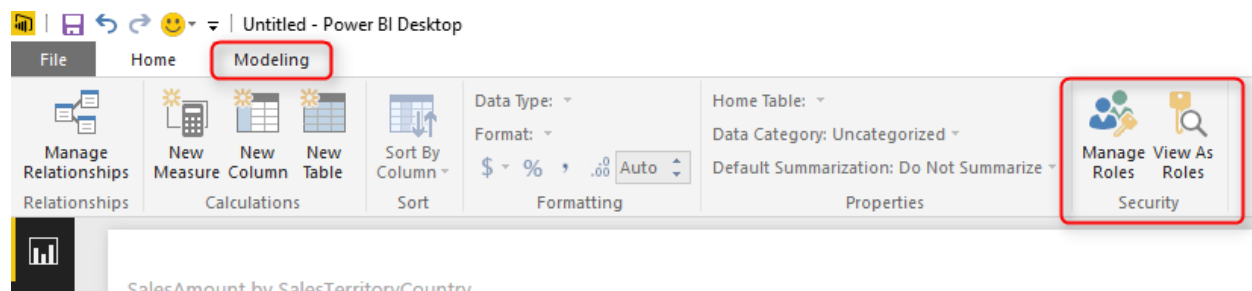
The chart shows sales amount by countries, which can be used for creating row level security on Geo-location information easily. Now let's add one card visualization for total Sales Amount. and two slicers (one for Sales Territory Group, and the other one for Sales Territory Region). This is the shape of this sample report now;



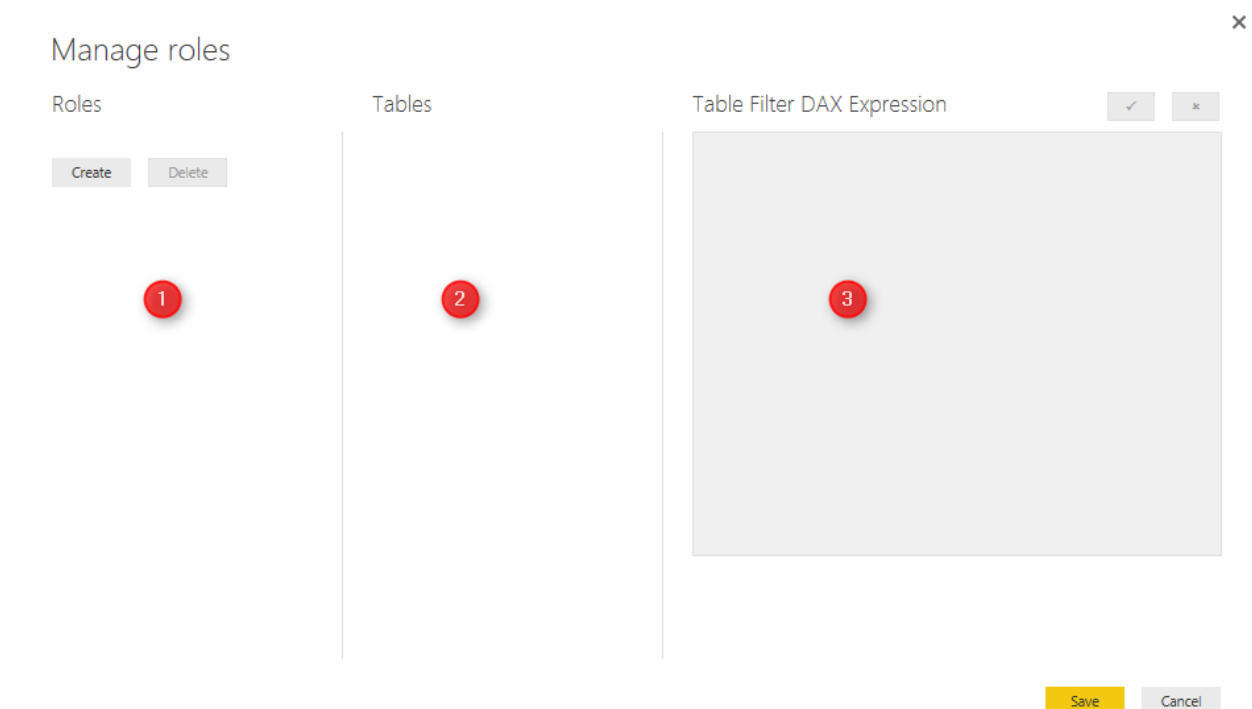
Our grand total Reseller sales amount in this view is \$80M, and we have sales values for Australia, Canada, France, Germany, UK, and USA. Now let's create roles.

Creating Roles

Now that we've build a visualization sample, let's create roles for that. our goal here is to build roles for sales manager of USA, and Europe. they should each only see their own group or country in the data set. For creating roles go to Modeling tab in Power BI Desktop. You will see a section named Security there (This is just added in June's update of this product);

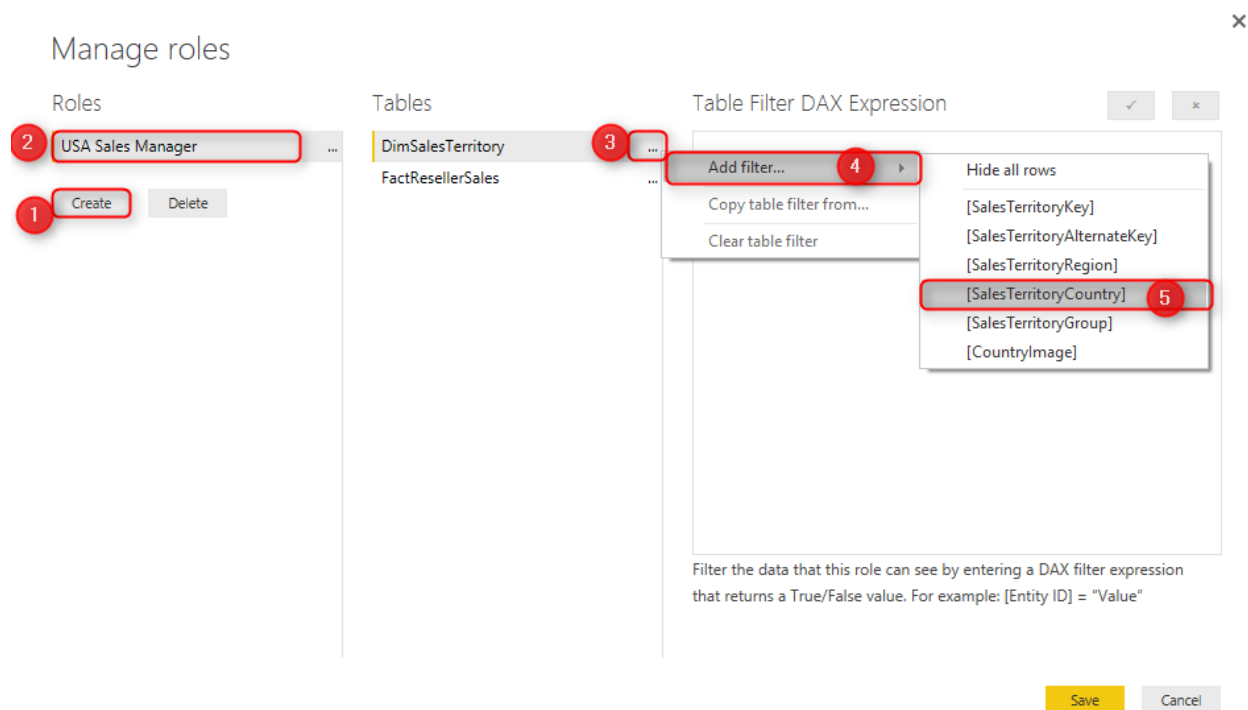


Click on Manage Roles to create a new role. You will see Manage Roles window which has three panes as below



You can create or delete roles in numbered 1 pane, You can see tables in your model in numbered 2 pane (for this example you will see two tables only, but not now, after creating the first role), and then you can write your DAX filtering expression in numbered 3 pane. Yes, you have to write DAX code to filter data for each role, but this can be very simple DAX expressions.

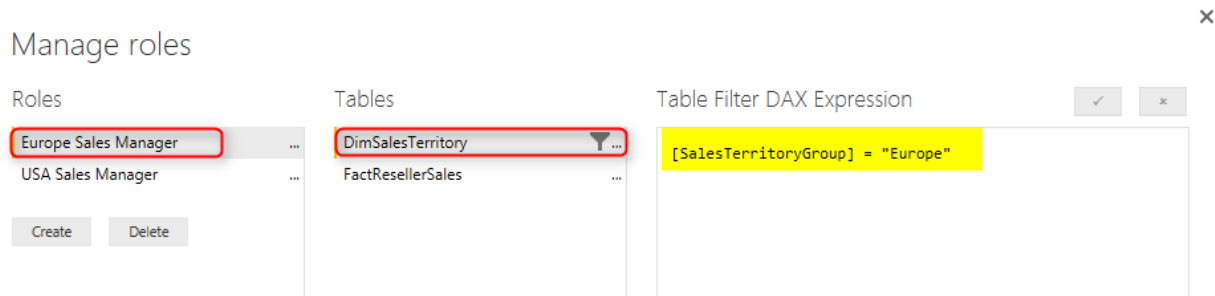
Now Create a Role, and name it as "USA Sales Manager", you will see two tables in the Tables section: FactResellerSales, and DimSalesTerritory. with click on ellipsis button of each table you can create DAX filters for each column. from DimSalesTerritory create a filter for Country.



Now in the DAX Filter expression you will see an expression created automatically as `[SalesTerritoryCountry] = "Value"`, change the value to United States, and apply.

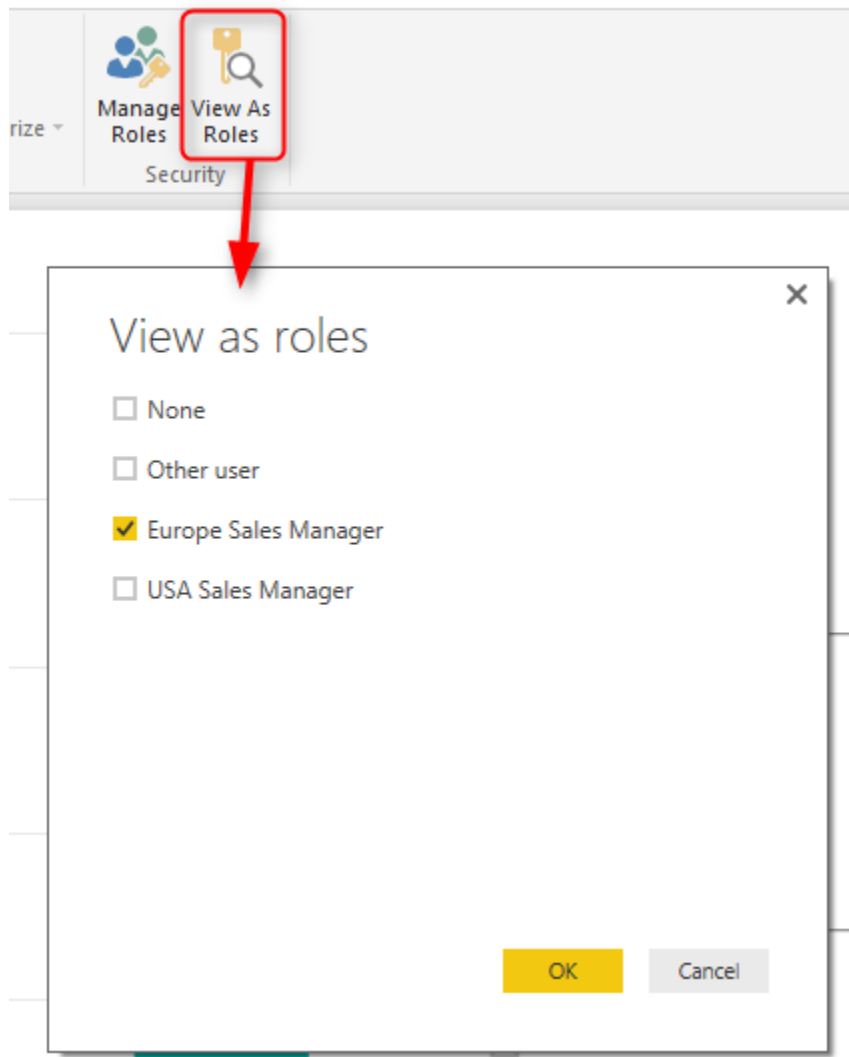


Now create another role, name it as Europe Sales Manager, put filter on SalesTerritoryGroup this time, and change Value to "Europe" as below;

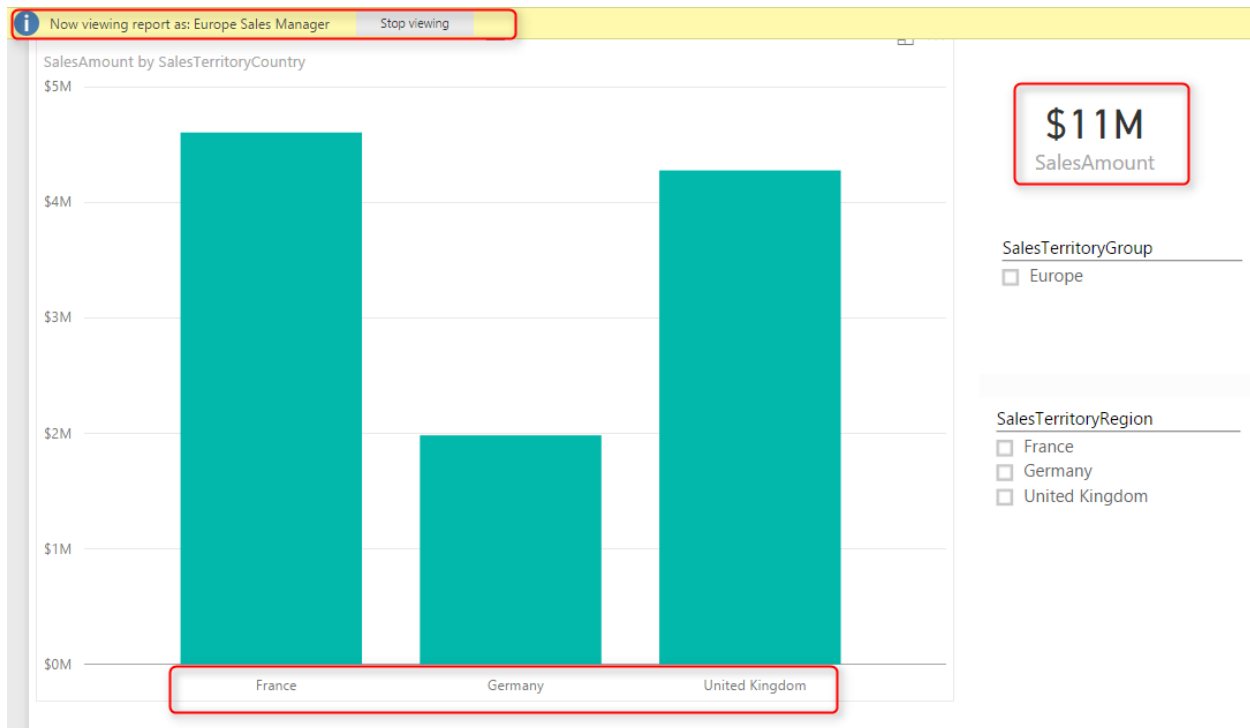


Testing Roles in Desktop

Great, we have created our two sample roles. now let's test them here. Yes, we can test them in Power BI Desktop with View As Roles menu option. This option allows us to view the report exactly as the user with this role will see. we can even combine multiple roles to see consolidated view of a person who has multiple roles. Go to Modeling tab, and choose View As Role option.



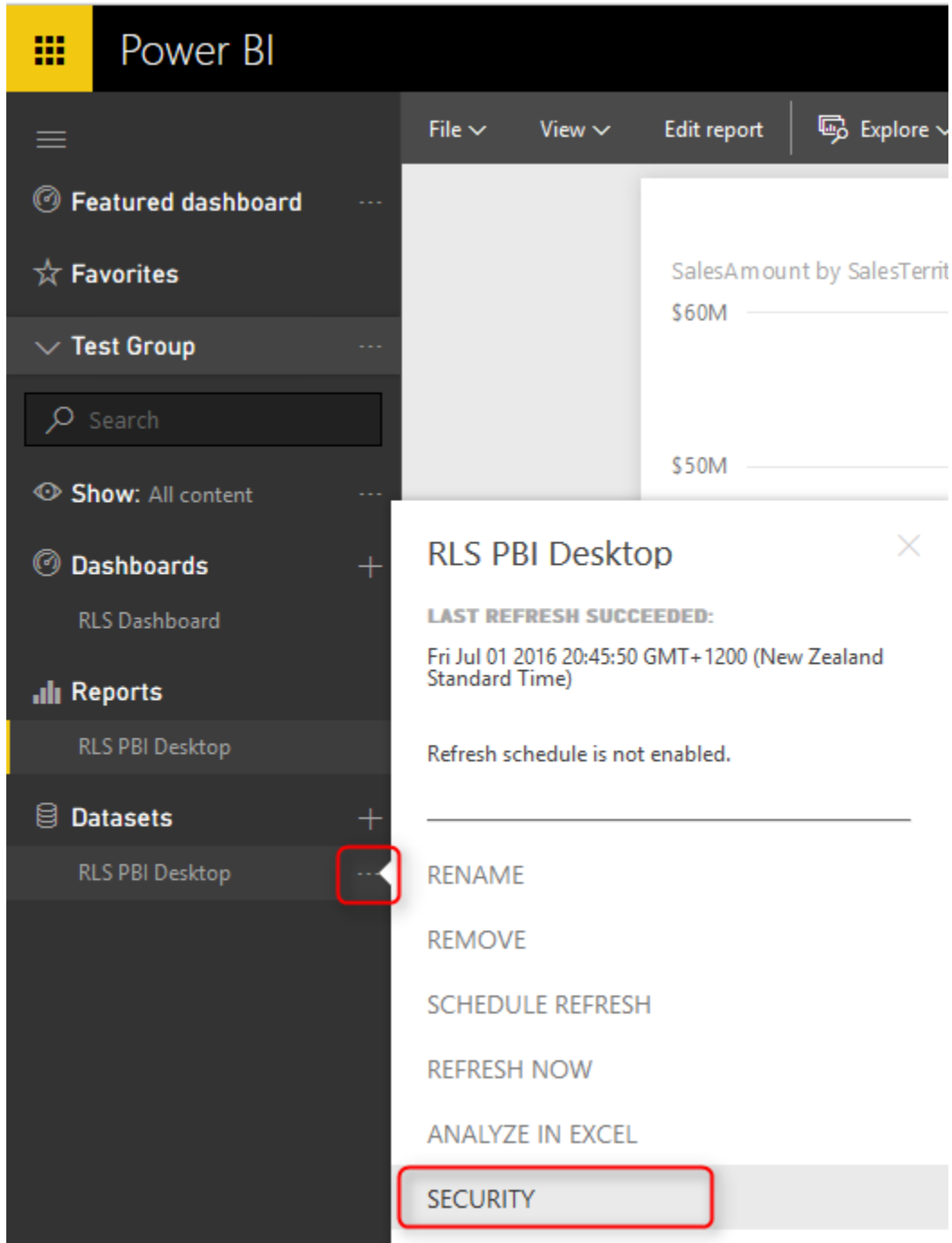
Choose Europe Sales Manager, and click on OK. You will see sales for Europe only showing with grand total of \$11M, and showing only countries Germany, UK, and France.



You can also see in the top of report there is an information line highlighted showing that the view is Europe Sales Manager. If you click stop viewing you will see the report as normal view (grand total view).

Power BI Service Configuration

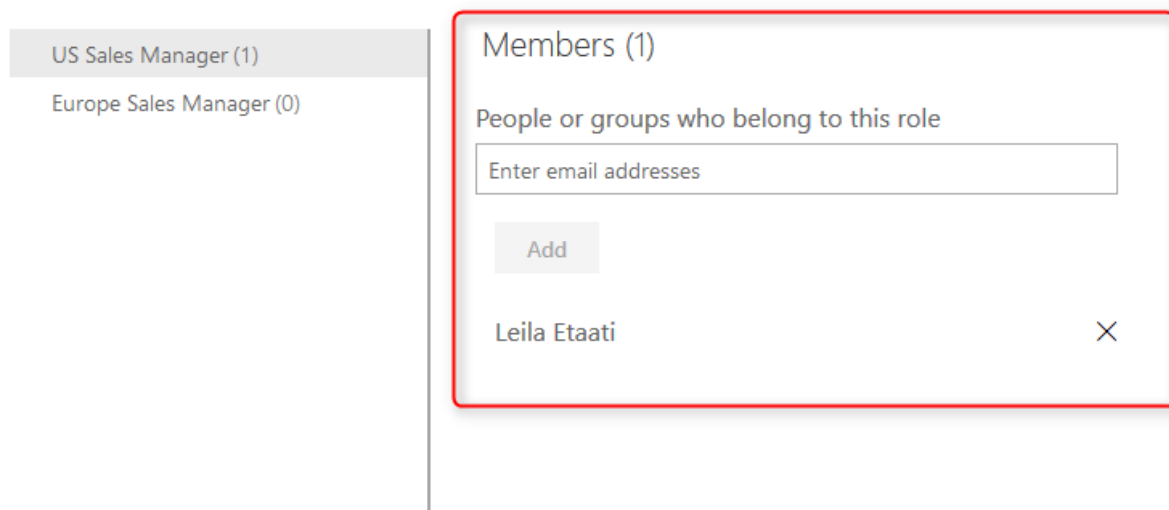
Roles should be assigned to Power BI users (or accounts in the other words), and this part should be done in Power BI Service. Save and publish the report into Power BI. I named this report as RLS PBI Desktop. You can name it whatever you want. After publish click on Security for the data set.



The screenshot shows the Power BI Desktop application. The left-hand navigation pane is visible, with the 'Datasets' section expanded. Under 'Datasets', the 'RLS PBI Desktop' dataset is selected. A context menu is open for this dataset, showing options: RENAME, REMOVE, SCHEDULE REFRESH, REFRESH NOW, ANALYZE IN EXCEL, and SECURITY. The 'SECURITY' option is highlighted with a red rectangle. The main area of the application shows a report titled 'SalesAmount by SalesTerrit' with a chart displaying sales amounts ranging from \$50M to \$60M.

Here you can see roles and assign them to Power BI accounts in your organization.

Row-Level Security (Preview)

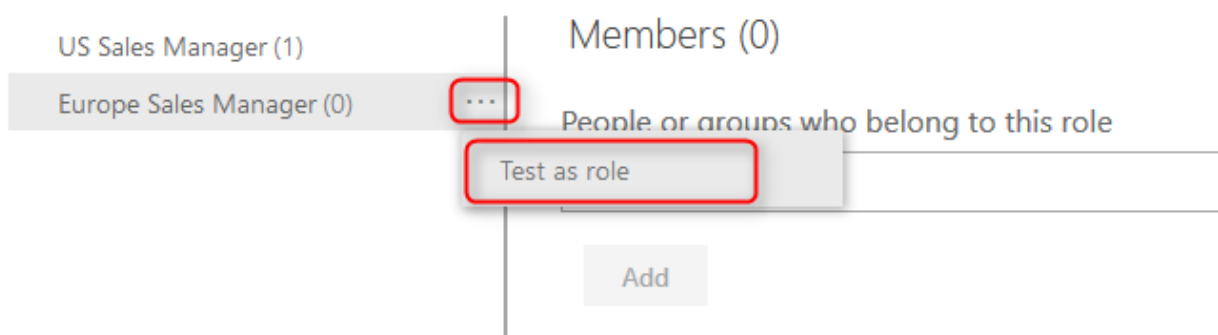


You can add each user to more than one role, and the user then will have a consolidated view of both roles. For example, a user with both roles for USA and Europe sales manager will see data from All Europe and USA.

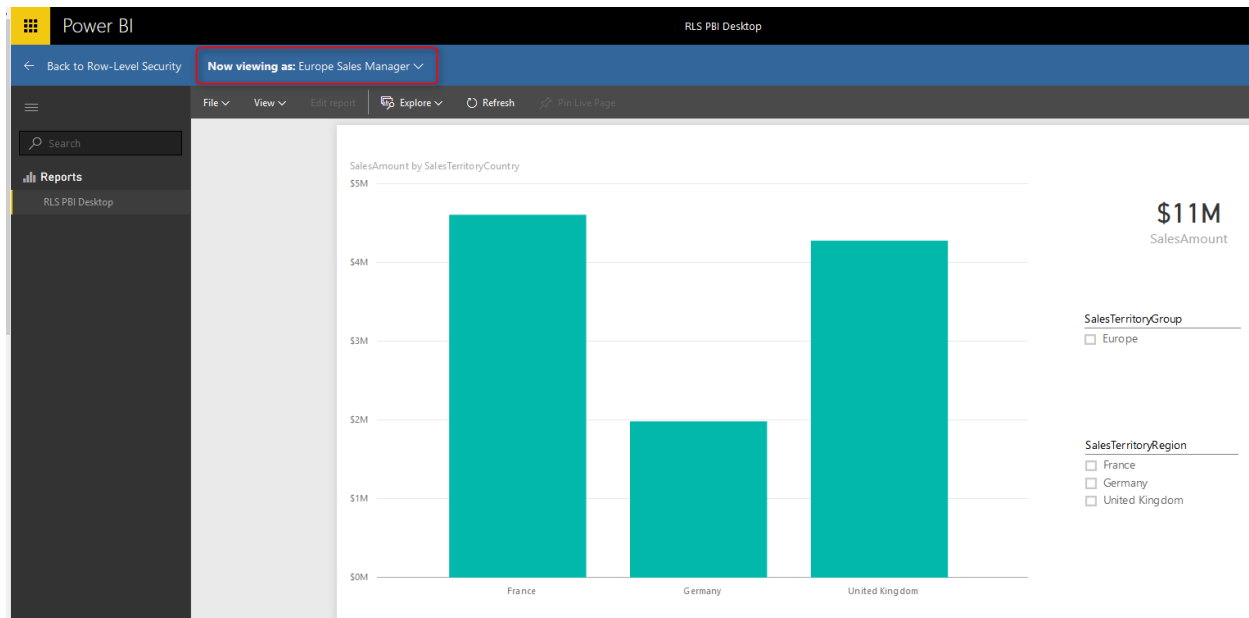
Test Roles in Power BI Service

You can also test each role here, just click on the ellipsis button beside each role, and click on Test as Role.

Row-Level Security (Preview)



This will show you the report in view mode for that role. As you see the blue bar shows that the report showed as the role of Europe Sales Manager. You can change it there if you like.



With setting users for each role, now your role level security is ready to work. If the user login with their account, they will only see data for their roles.

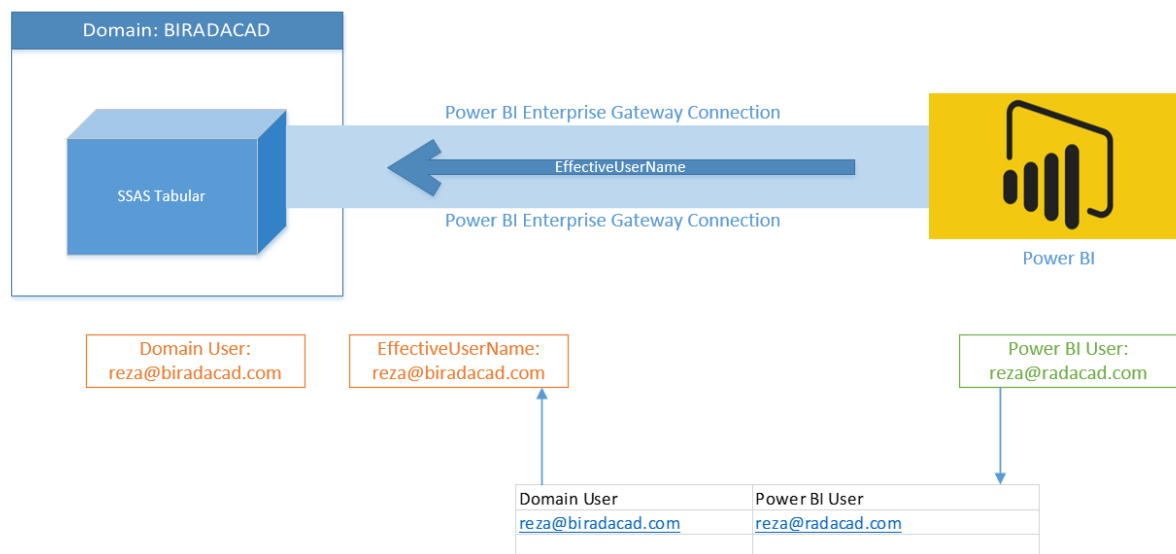
Re Publish Won't Hurt

As I mentioned in the very first paragraph of this post, the great thing about this new feature is that RLS is part of Power BI model. and if you publish your Power BI model again and again with changes, you won't lose configuration in the web. You also won't lose users assigned to each role, as long as you keep role names unchanged.

Don't forget that this feature is still in preview mode, and we will see more updates for this in the future.

Row Level Security with SSAS Tabular Live Connection in Power BI

Published Date: July 4, 2016



You can define [Row Level Security in Power BI itself](#). However sometimes you do use SQL Server Analysis Services Tabular as the source for Power BI through Live or DirectQuery connection. SSAS Tabular allows you to create same type of Row Level Security. So it would be much better to use Row Level Security defined in Tabular from Power BI, rather than duplicating that in Power BI. In this post I'll show you an end-to-end solution which contains elements below;

- Power BI Live Connect to SSAS Tabular through Enterprise Gateway
- Row Level Security Configuration in SSAS Tabular
- Viewing the result filtered by RLS in Power BI

Note that this method is different from defining Row Level Security in Power BI Desktop. In fact we won't use that method. If you like to learn more about that method read [this blog post](#). If you are interested to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

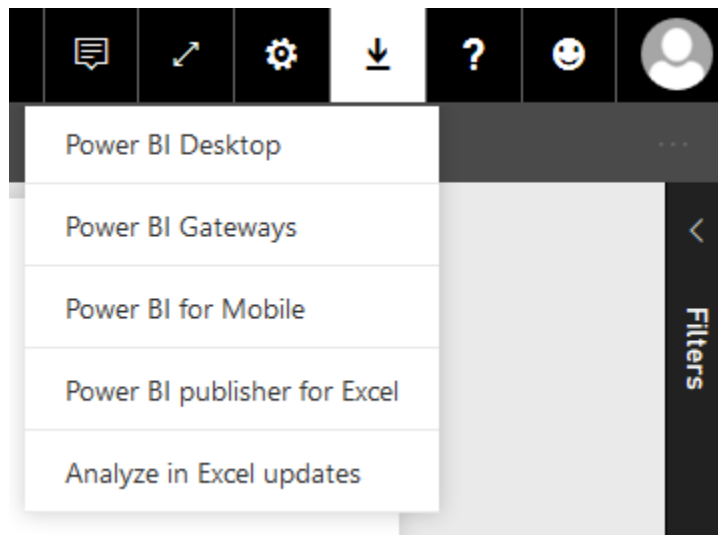
Prerequisite

For this demo I will be using my demo machine which has SSAS Tabular and Enterprise Gateway. In a separate post I'll explain how to create such demo VM. You also need to have SSAS Tabular database example AW Internet Sales Tabular Model 2014 from [here](#).

Install and Configure Enterprise Gateway

Purpose of gateways in Power BI is to create connection from Power BI cloud service to on-premises data sources. There are two types of gateways; Enterprise, and Personal. Naming of these gateways are a bit misleading. Personal doesn't mean you have to install gateway on your laptop only, and Enterprise doesn't mean only for organizations. There are more differences which I will talk in different post. In general Enterprise is built for more live connections specially to Analysis Services, and Personal is more for importing data from many data sources. Let's leave details of that for now. For this example we will be using Enterprise Gateway because we want to connect to SSAS Tabular on-premises through a Live/DirectQuery connection.

You can download latest version of Enterprise Gateway from [PowerBI.Microsoft.Com](https://powerbi.microsoft.com) website under downloads



Then from gateways download page, choose Enterprise gateway, and download it.

Choose the gateway that best fits your needs

For personal use

Refresh your on-premises data quickly without waiting for an IT Admin. Designed for use with personal data sets, there is no central monitoring capabilities for this gateway.

Power BI Gateway - Personal

[Download](#)[Learn more](#)

For enterprise deployments

This gateway is used by organizations to serve a large number of users. It also enables administrators to set up access control for individual data sources and monitor usage.

Power BI Gateway - Enterprise

[Download](#)[Learn more](#)

Installation of Enterprise Gateway is just following a wizard, so continue that till end of the wizard where it asks for Power BI login. Sign in to your Power BI account;



Power BI Gateway – Enterprise

[?](#) [x](#)


This wizard helps you configure a Power BI Gateway.

Setup was successful!

To configure the gateway, you must first sign in to Power BI.

[Sign in to Power BI](#)[Cancel](#)

After sign in, configure the Power BI Gateway with a name, and key, and then you should be good to go. If you see green checkbox and Connected means your configuration was successful.



Power BI Gateway – Enterprise

[?](#) [x](#)

The Power BI Gateway has been setup and configured.

✓ Connected: BIRADACAD is good to go.



Next Step:

Add data sources to this Gateway in the Power BI service.



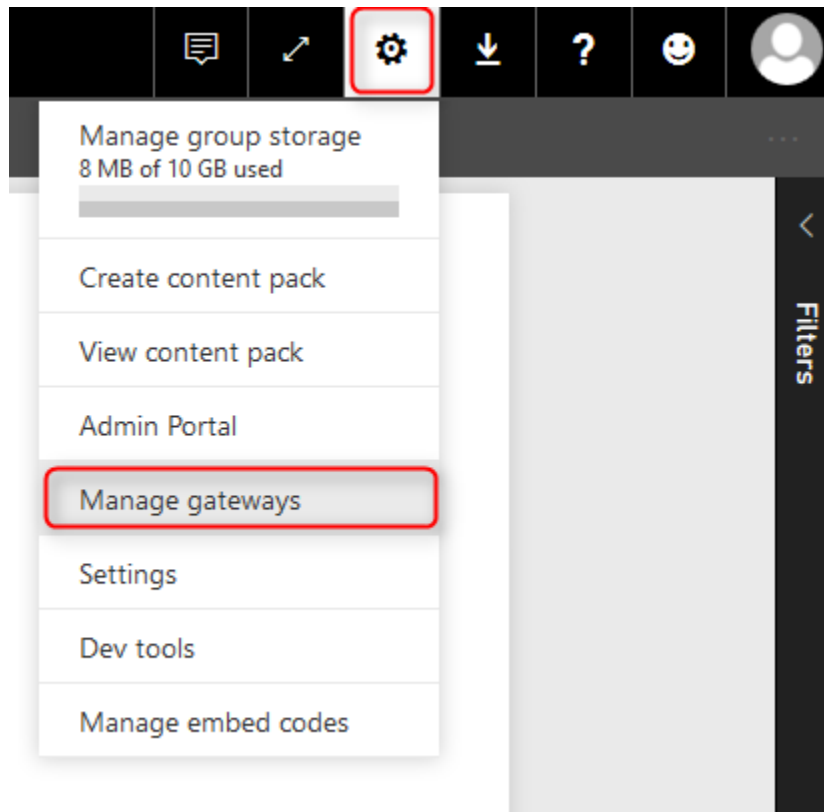
Help improve Power BI Gateway by sending usage information to Microsoft.

[Read the privacy statement online](#)

Close & Add data sources

As you can see after successful configuration, you can close the gateway (don't worry it is running behind the scene as a service), and add a data source to this gateway in Power BI Service.


Or alternatively you can directly go to Power BI website, after login go to setting menu option and choose Manage Gateways



In Gateway management window, you can see all gateways you have set up. My gateway here is named BIRADACAD, and I can see that it is successfully connected.

Gateways

+ ADD DATA SOURCE

> RezaSurface 

> BIRADACAD

Test all connections

Gateway Settings

Administrators

✓ Online: You are good to go.

Gateway Name

BIRADACAD

Department

Description

Contact Information

Reza@BIRADACAD.com

Apply

Discard

Create Data Source

Now Let's create Data Sources. You might think that one gateway is enough for connecting to all data sources in a domain. That is absolutely right, however you still need to add a data source under that gateway per each source. each source can be a SQL Server database, Analysis Services database and etc. For this example we are building a data source for SQL Server Analysis Tabular on premises. Before going through this step; I have installed AW Internet Sales Tabular Model 2014 on my SSAS Tabular, and want to connect to it. If you don't have this database, follow instruction in prerequisite section.

For creating a data source, click on Add Data Source in manage gateways window (you have to select the right gateway first)

Gateways

+ ADD DATA SOURCE

> RezaSurface ⚠


> BIRADACAD

Test all connections

Then enter details for the data source. I named this data source as AW Internet Sales Tabular Model 2014, I enter my server name, and database name. then I use Windows authentication with my domain user <domain>\username and the password. You should see a successful message after clicking on Apply. The domain name that I use is BIRADACAD (my SSAS Tabular domain), and the user is PBIGateway, which is a user of BIRADACAD domain (username: BIRADACAD\PBIGateway), and is an administrator for SSAS Tabular (explained in next few paragraphs).

Gateways

+ ADD DATA SOURCE

> RezaSurface 

✓ BIRADACAD

SSAS MD PBIRAD

SSAS Tab PBIRAD

AdventureWorksDW2014

Test all connections

Data Source Settings

Users

✓ Connection Successful

Data Source Name

SSAS Tab PBIRAD

Data Source Type

Analysis Services

Server

PBIRAD\Tabular

Database

AW Internet Sales Tabular Model 2014

The credentials are encrypted using the key stored on-premises on the gateway server. [Learn more](#)

Username

••••••••••

Password

••••••••••

> Advanced settings

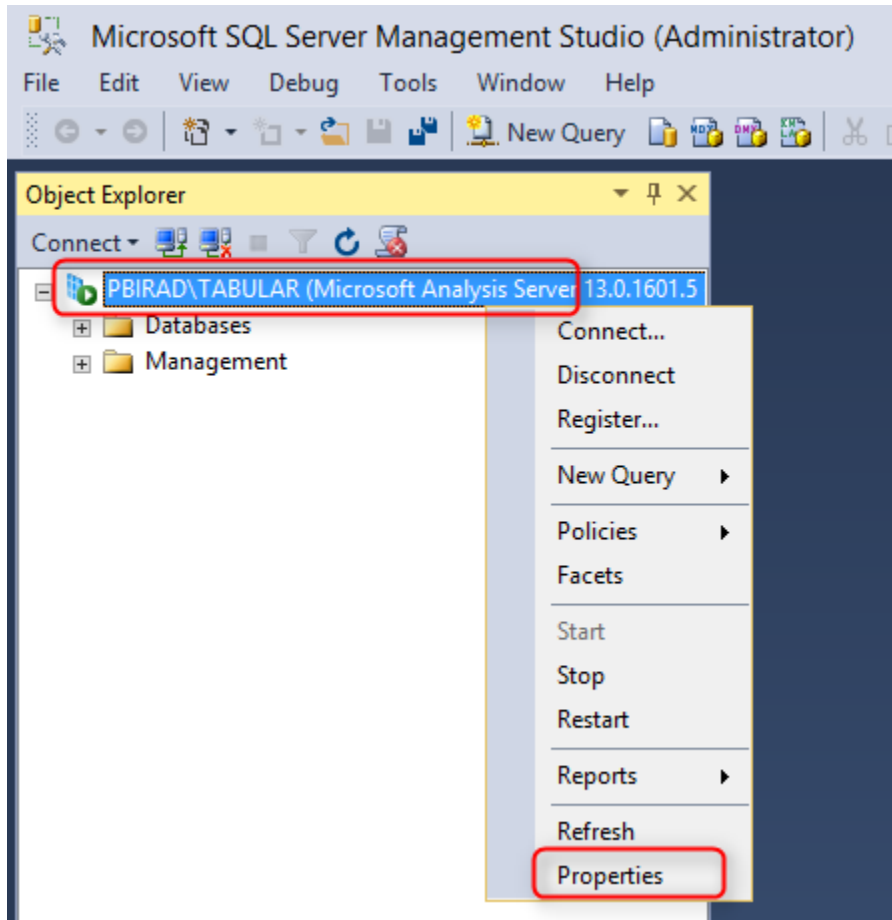
Apply

Discard

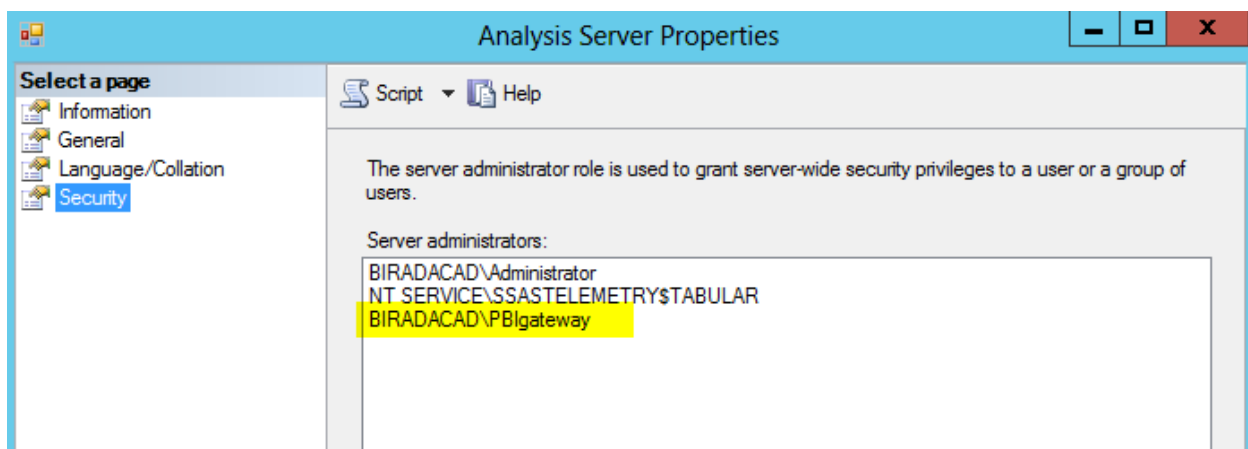
Note that the user account that you are using here should meet these conditions:

25. It should be a Domain User
26. the domain user should be an administrator in SSAS Tabular.

You can set administrator for SSAS Tabular with right clicking on SSAS Tabular instance in SSMS, and in Properties window,



in the Security setting add the user to administrators list.

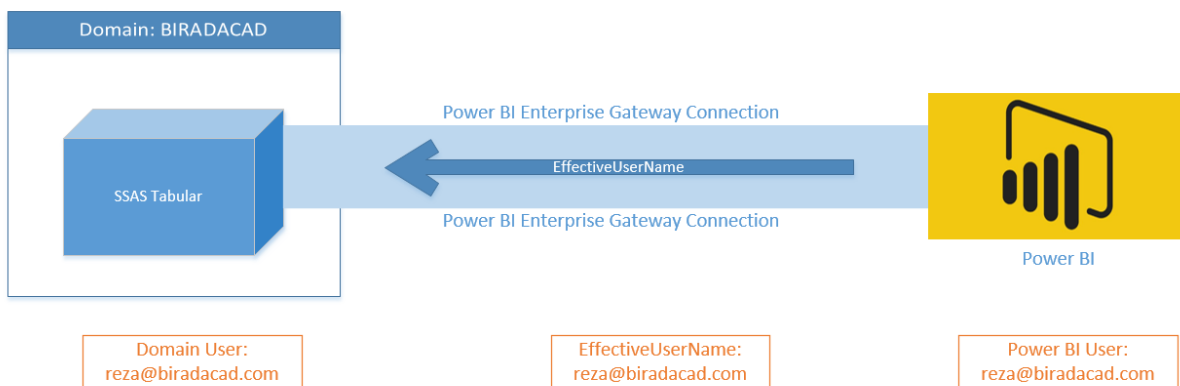


Effective User Name

Gateway account used for accessing Power BI cloud service to on-premises SSAS Tabular. However this account by itself isn't enough for the data retrieval. gateway then

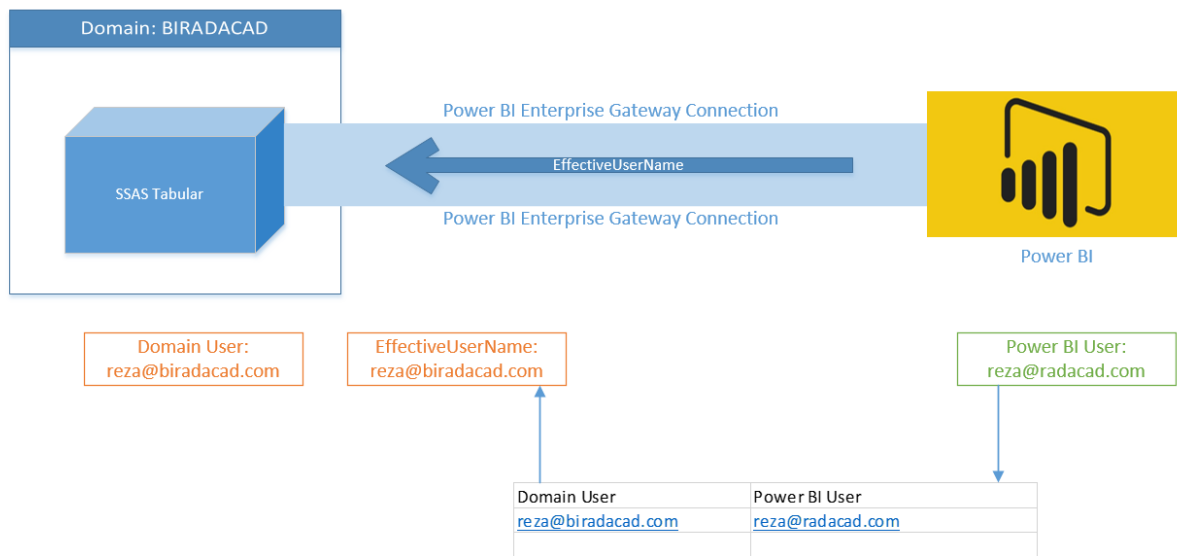
passes the EffectiveUserName from Power BI to on-premises SSAS Tabular, and result of query will be returned based on the access of EffectiveUserName account to SSAS Tabular database and model.

By default EffectiveUserName is the username of logged in user to Power BI, or in other words EffectiveUserName is the Power BI account. This means your Power BI account should have enough access to SSAS Tabular database to fetch required data. If Power BI account is the account from same domain as SSAS Tabular then there is no problem, and security configuration can be set in SSAS Tabular (explained later in this post). However if domains are different, then you have to do UPN mapping.



UPN Mapping

Your SSAS Tabular is part of a domain (it should be actually, because that's how Live connection works), and that domain might be the domain that your Power BI user account is. If you are using same domain user for Power BI account then you can skip this step. If you have separate Power BI user account than the domain account for SSAS Tabular, then you have to set the UPN Mapping.



UPN Mapping in simple definition will map Power BI accounts to your local on-premises SSAS Tabular domain accounts. Because in my example I don't use same domain account for my Power BI account, so I set up UPN as below.

Gateways

+ ADD DATA SOURCE

- > RezaSurface ⚠
- ✓ BIRADACAD
 - SSAS MD PBIRAD
 - SSAS Tab PBIRAD** 1
 - AdventureWorksDW2014

Test all connections

Data Source Settings **Users** 2

People who can publish reports that use this data source

Enter email addresses Add

☐ Reza Rad

Remove Map user names 3

Then in Mapping pane, I create a new mapping that map my Power BI user account to reza@biradacad.com, which is my local domain for SSAS Tabular server.

Map user names

Create rules to map user names to Analysis Services server user names or associate custom data with user names. [Learn more](#)

Select the type of rule for this data source

- ☒ Effective user names 1
☐ CustomData

	Replace	With
1	reza@biradacad.com	reza@biradacad.com
2	Original name	New name

3

Enter user name to see how the mapping rule will change it.

Original name

<input type="text" value="Original name"/>	<input type="button" value="Test rule"/>
--	--

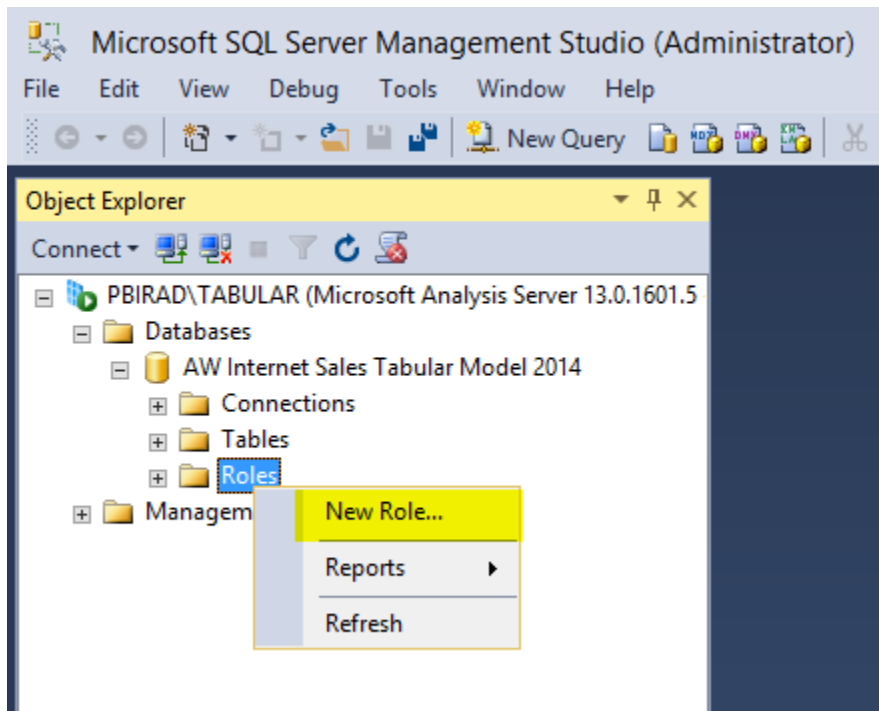
After rule applied

Result of applying mapping rule will appear here
--

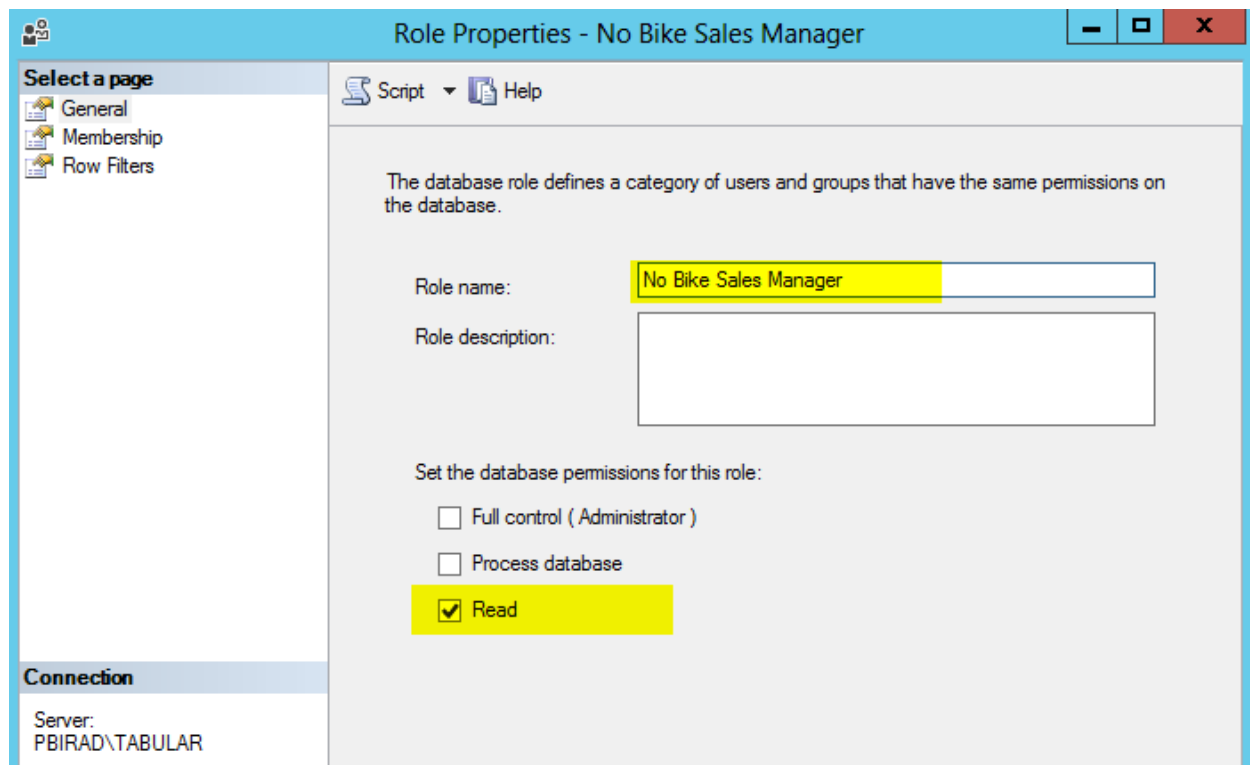
Now with this user name mapping, reza@biradacad.com will be passed as EffectiveUserName to the SSAS Tabular. If you want to learn more about UPN mapping, you can watch [Adam Saxton's great video](#) about it.

Configure Row Level Security in SSAS Tabular

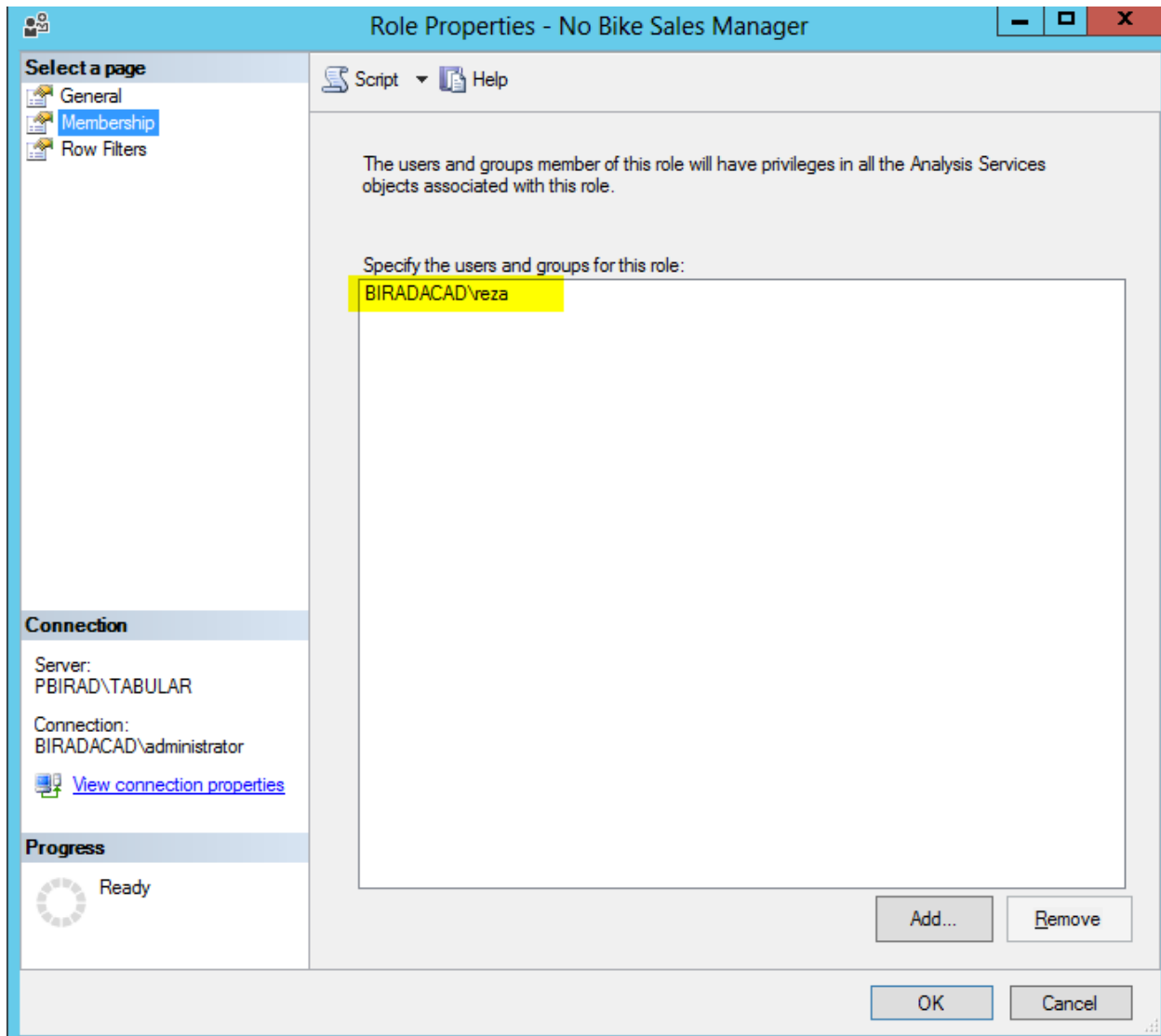
In SSAS Tabular opened in SSMS, expand AW Internet Sales Tabular Model 2014, and create a New Role



We want to create a role for users who don't have access to Bike sales across the database. so let's name the role as No Bike Sales Manager, this role has Read access to the database.

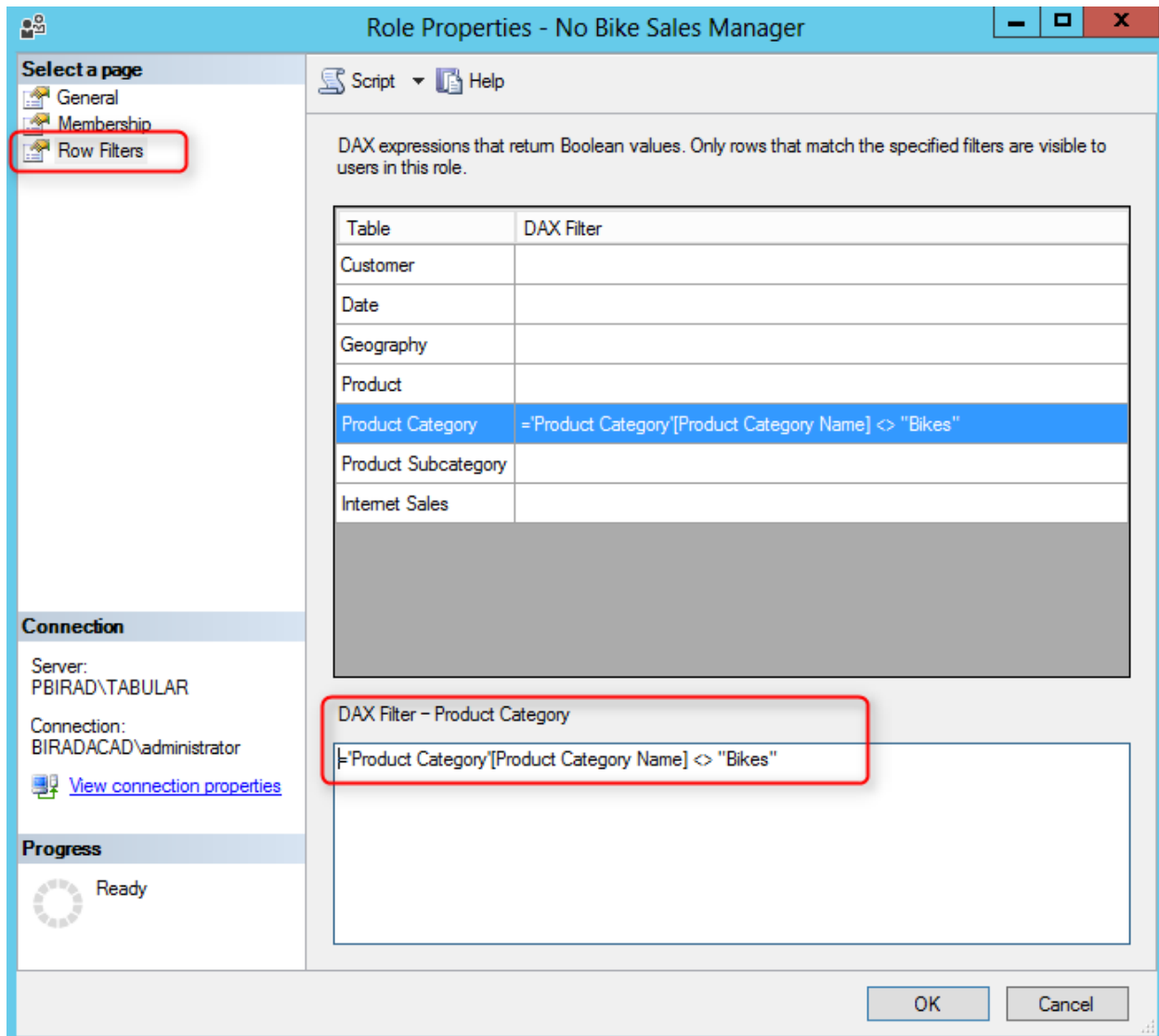


In the membership section you can add users to this role. Users should be domain users that you get through EffectiveUserName from Power BI (if Power BI accounts aren't domain users, then create UPN mapping for them as explained above in UPN mapping section). I add user BIRADACAD\Reza here. (Note that I've created a UPN mapping for this user. so each time Power BI user associated with this logs in, this domain account will be passed through EffectiveUserName to SSAS Tabular).



Now let's define Row Filters with a basic filter on Product Category as below;

= 'Product Category'[Product Category Name] <> 'Bikes'



Now that we defined row level security in SSAS Tabular, let's build a simple report to test it.

Create Sample Report

Our sample report here would only show a Pie chart of Product Categories Sales. So create a Live connection to SSAS Tabular from Power BI Desktop.

SQL Server Analysis Services Database

Connect live or import data from a SQL Server Analysis Services database.

Server

PBIRAD\Tabular

Database (optional)

AW Internet Sales Tabular Model 2014

☒ Connect live
 ☐ Import data

☐ MDX or DAX Query (optional)

OK

Cancel

Then choose the model

Navigator

PBIRAD\Tabular: AW Internet Sales Tabular Model

Internet Sales

Model

Model

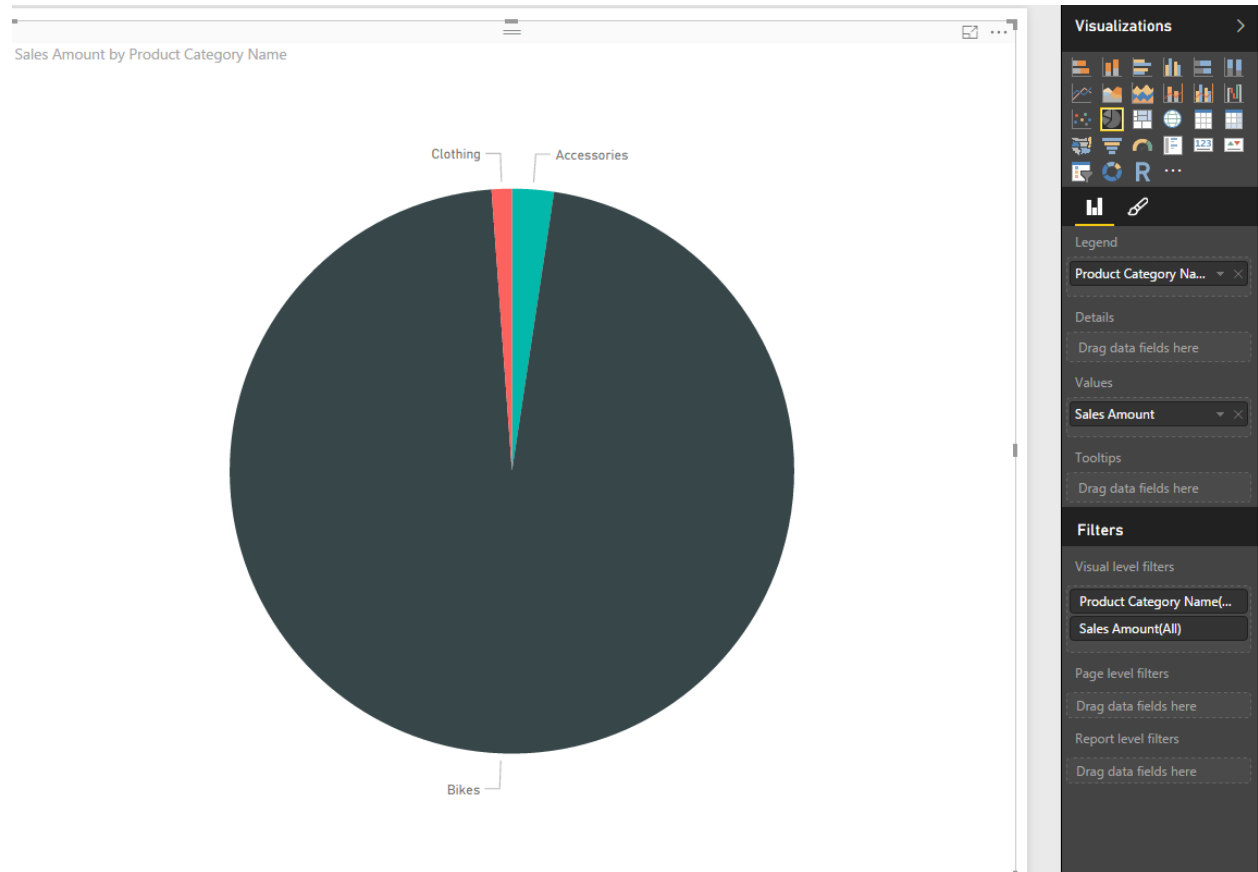
Last Modified: 07/18/2014 17:24:26

This model contains the following dimensions and measures

Customer, Date, Geography, Internet Sales, Product, Product Category, Product Subcategory, _Count of Models, Days in Current Quarter, Days Current Quarter to Date, Internet Distinct Count Sales Order, Internet Order Lines Count, Internet Total Discount Amount, Internet Total Product Cost, Internet Total Sales, Internet Total Tax Amt, Internet Total Freight, Internet Total Margin, Internet Previous Quarter Margin, Internet Current Quarter Margin, Internet Previous Quarter Margin Proportion to QTD, Internet Previous Quarter Sales, Internet Current Quarter Sales, Internet Previous Quarter Sales Proportion to QTD, Internet Current Quarter Sales Performance, Internet Current Quarter Margin Performance

After creating the connection you should see the Live Connection: Connected status in Power BI Desktop down right hand side corner.

Create a simple Pie chart with Product Category Name (from Product Category table) as Legend, and Sales Amount (from Internet Sales table) as Values.

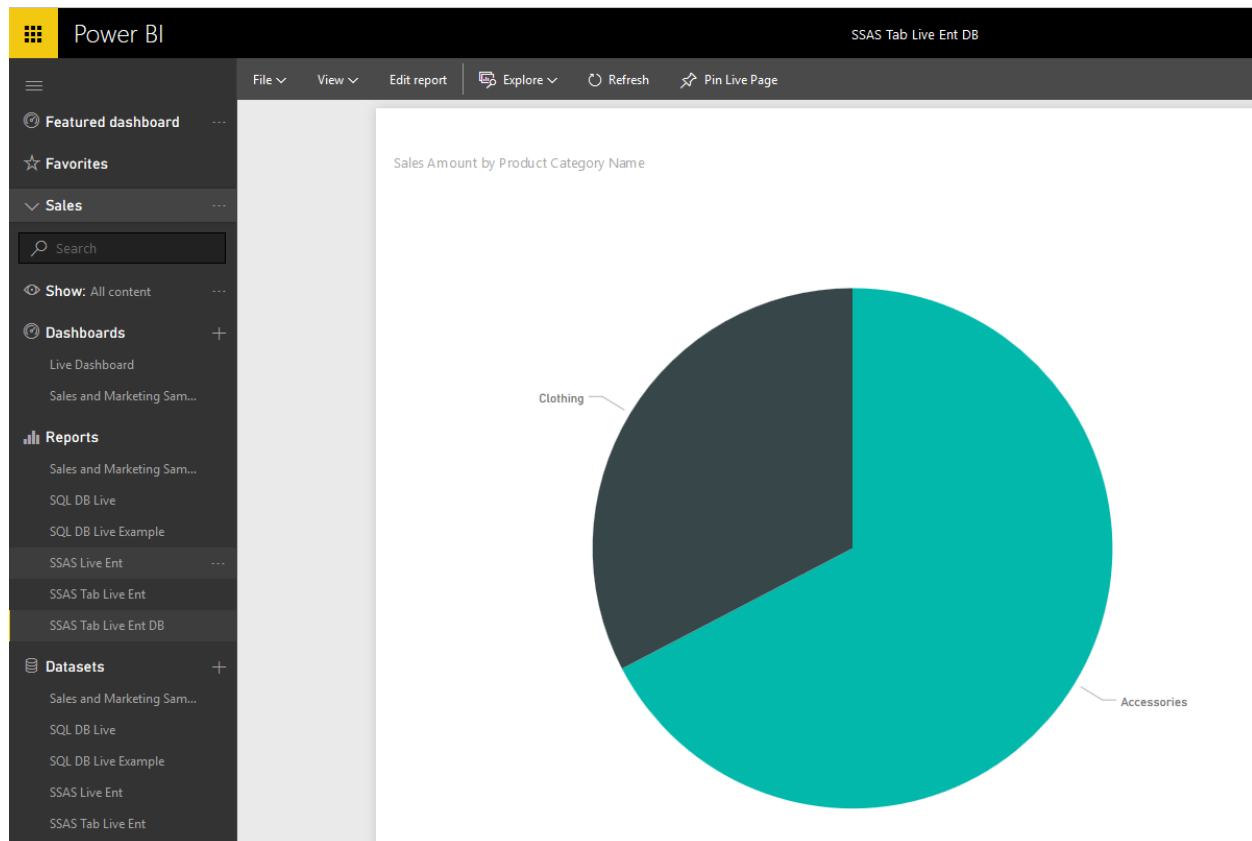


Save the Power BI file, for example as SSAS Tab Live Ent DB.pbix, and publish it.

Remember that you shouldn't set up Row Level Security in Power BI itself. Because we will be using RLS configuration from Live Tabular connection.

Test the Result

In Power BI website or service, when you login, and refresh the SSAS Tab Live Ent DB report. you won't see any sales from Bikes. you will only see sales of Clothing and Accessories.



How does it work?

What happened here is that my Power BI account mapped to reza@biradacad.com domain user, and this domain user passed through enterprise gateway as the EffectiveUserName to the SSAS Tabular on-premises. SSAS Tabular identified that this user has only one role which is No Bike Sales Manager, and this role has a restricted view of sales for everything but Bike. So the data returned from SSAS Tabular to Power BI report doesn't contain Bike's sales.

Summary

SSAS Tabular as a Live connection source for Power BI used in many enterprise solutions for Power BI. There are different reasons for that, for example;

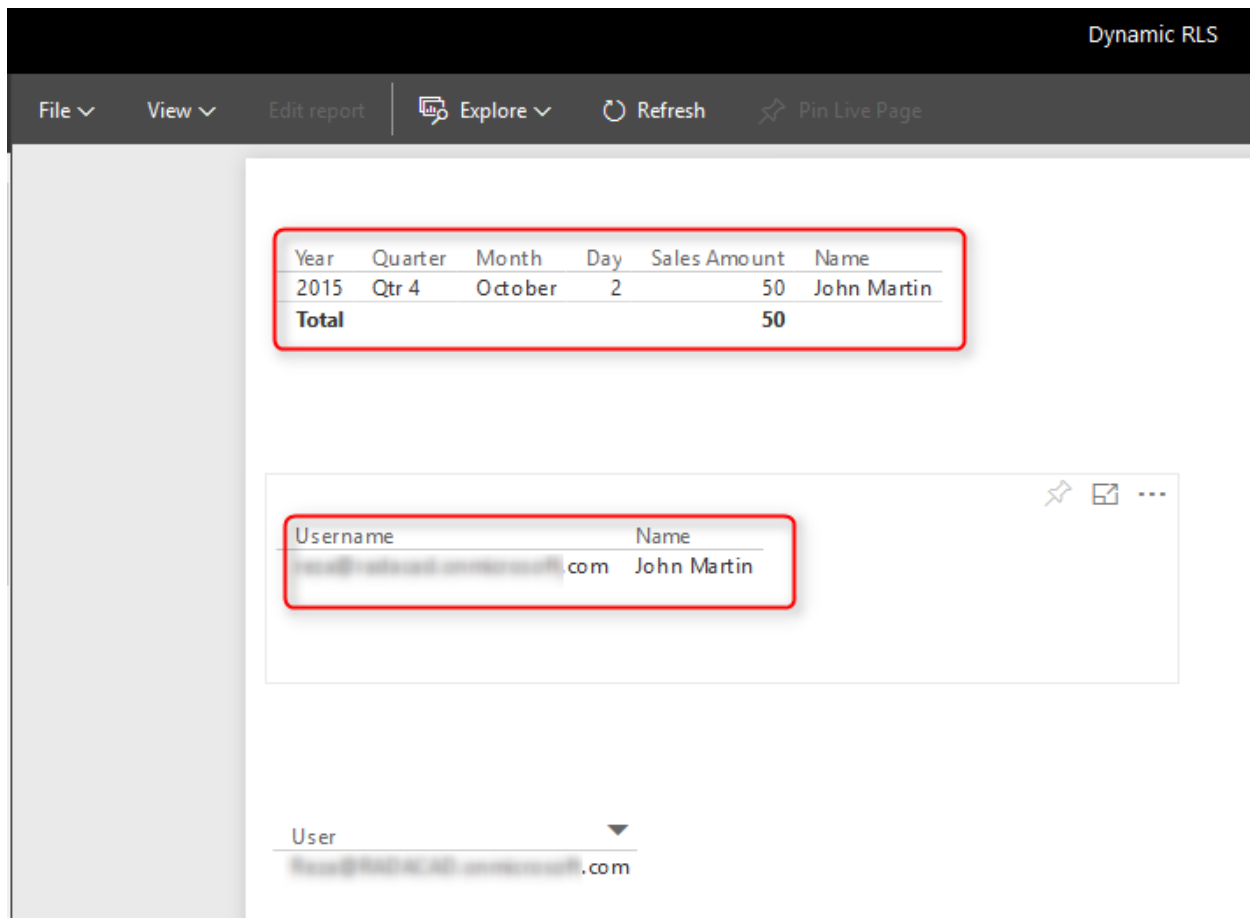
36. Some organization already have SSAS Tabular models ready, and they are using that for their on-premises reporting and data analysis. So they want to use the same source of truth.
37. Scale of data is larger than it fits into Power BI model. (Read [step beyond 10GB limitation for Power BI](#) as an example).

38. and many other reasons.

In this post you've seen how Row Level Security defined in SSAS Tabular will be passed through `EffectiveUserName` to Power BI accounts. This method will authorize users to only view part of the data that they are authorized to see.

Dynamic Row Level Security with Power BI Made Simple

Published Date: July 4, 2016



The screenshot shows a Power BI report titled "Dynamic RLS". The report has a table with columns: Year, Quarter, Month, Day, Sales Amount, and Name. The data shows sales for 2015 Qtr 4, October 2, with a sales amount of 50, attributed to John Martin. A "Total" row shows a sales amount of 50. Below the table, there is a section for "Username" and "Name", showing "john@radacad.com" and "John Martin". At the bottom, there is a "User" dropdown menu showing "john@radacad.com".

Year	Quarter	Month	Day	Sales Amount	Name
2015	Qtr 4	October	2	50	John Martin
Total				50	

Username	Name
john@radacad.com	John Martin

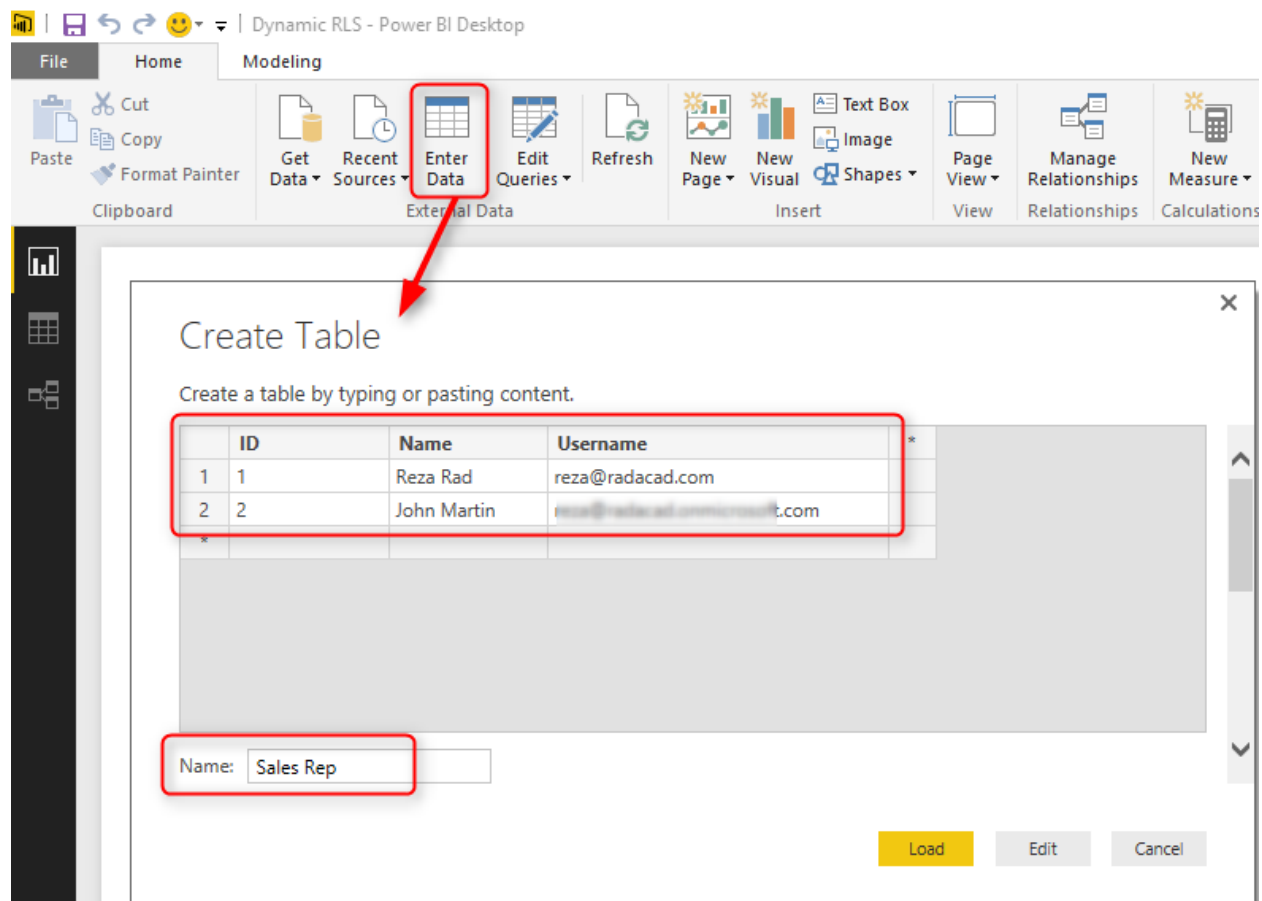
User: john@radacad.com

There are different methods to use row level security in Power BI; You can set up [Row Level Security in Power BI itself](#), or through a [live connection from a data source such as SSAS Tabular](#). However row level security defined in ways mentioned in blog posts above isn't dynamic. By dynamic row level security I mean definition of security be besides the user account information in the data source. For example when John log in to the system, based on data tables that shows John is sales manager for specific branch he should be able to see only those branch's data. This method is possible in Power BI using DAX USERNAME() function. In this blog post I'll show you an example of dynamic row level security with DAX USERNAME() function in Power BI. If you like to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

Sample Data

For this example I will use a data entered in Power BI itself. There won't be any external data sources. This doesn't mean that dynamic security has an issue with external data sources. Dynamic security works with any data sources as long as we have related data rows in the tables. However if I use on-premises data sources then half of this example should be explaining installation and configuration gateways, or if I use Azure data sources, then again I have to explain how to set up that example. So just for simplicity of this example I'll be using data source inside Power BI.

For this example let's create two simple tables; Sales Rep, and Transactions. Sales Rep has information of sales representatives, and transaction data is sales transactions, obviously each sales transaction handled by a sales rep. So let's create sample tables in Power BI. Open Power BI Desktop and from External Data section choose Enter Data.



Create a table as above with three columns, and data in it. You have to use usernames similar to Power BI accounts that you want to set up security for it. Name this table as Sales Rep.

Create another table for Transactions with structure below, and name it Transactions:

Create Table ×

Create a table by typing or pasting content.

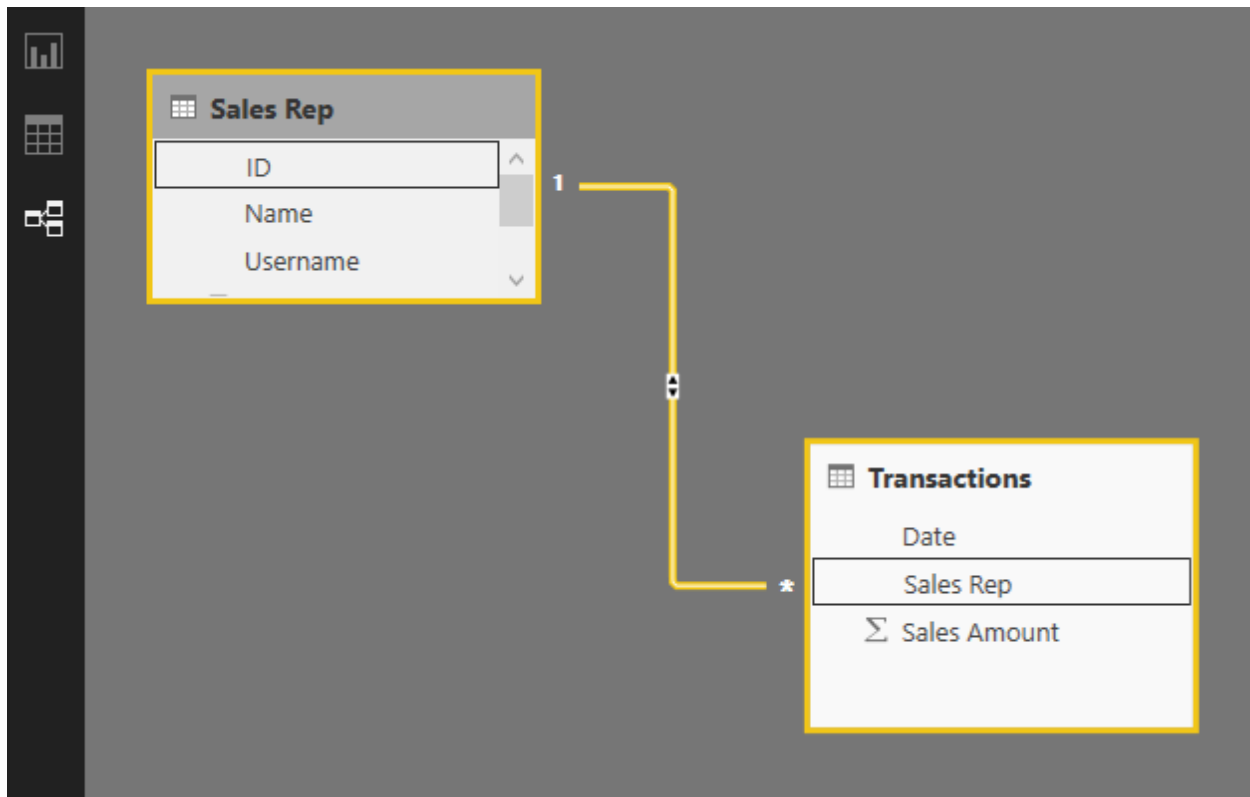
	Date	Sales Rep	Sales Amount	*
1	2015-10-1	1	100	
2	2015-10-2	2	50	
3	2015-10-3	1	200	
*				

Name:

Load Edit Cancel

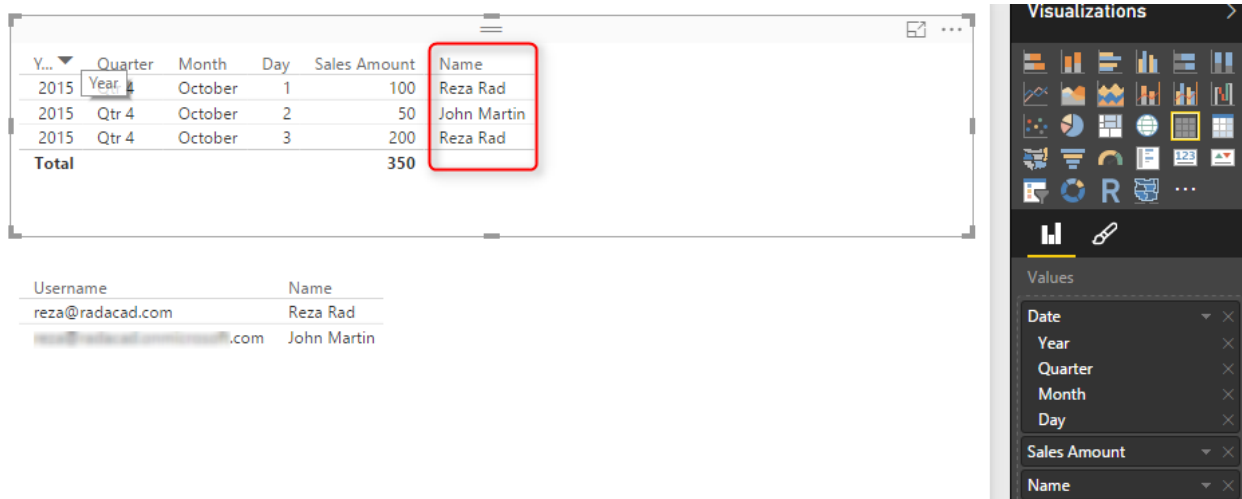
As you can see each sales transaction is handled by a sales rep. Again I mention that these tables are added inside Power BI just for simplicity of this example. Tables can come from everywhere.

load tables into Power BI, we don't need to do anything with Power Query at this stage. Go to Relationship tab and verify the relationship between Sales Rep (ID) and Transactions (Sales Rep) to be as below;



Sample Report

For this example I will be using basic table visualization. The table visualization will show Date, Sales Amount (from Transactions), and Name (from Sales Rep). I also added another table visualization under that to show username, and Name (both from Sales Rep);



The screenshot shows a Power BI report with two table visualizations. The top table displays transaction data, and the bottom table displays sales representative information. The Visualizations pane on the right shows the fields used in each table.

Y...	Quarter	Month	Day	Sales Amount	Name
2015	Year 4	October	1	100	Reza Rad
2015	Qtr 4	October	2	50	John Martin
2015	Qtr 4	October	3	200	Reza Rad
Total				350	

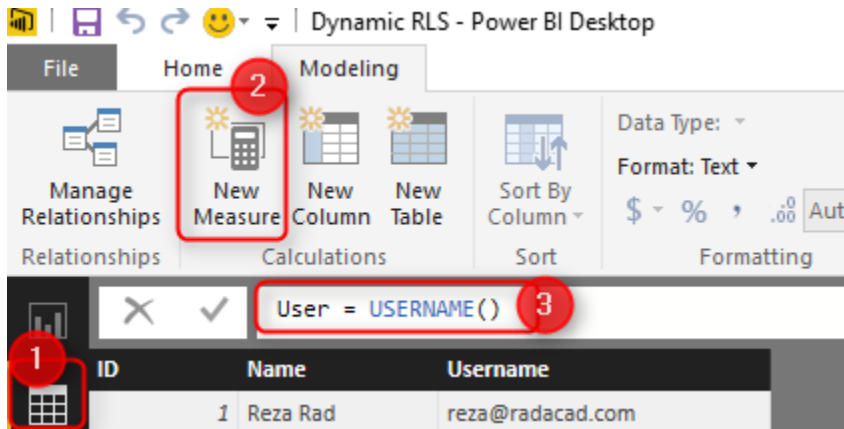
Username	Name
reza@radacad.com	Reza Rad
john@radacad.com	John Martin

Visualizations

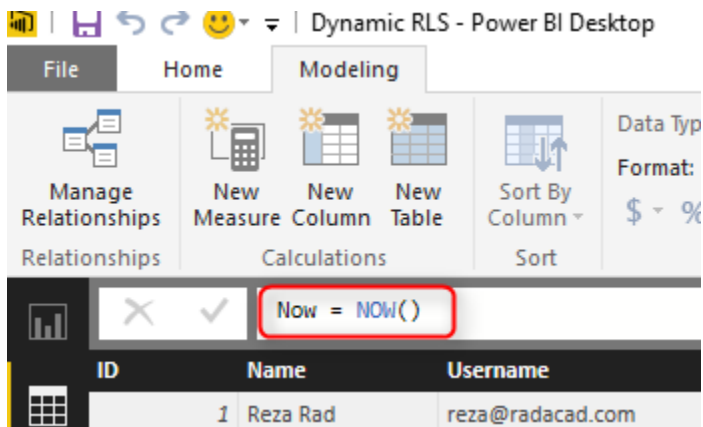
Values

- Date
- Year
- Quarter
- Month
- Day
- Sales Amount
- Name

The main reason for this visualization is to simply shows that each user will see only their own data rows from all tables. I also will add a measure for USERNAME() in DAX to see the user logged in from my report. So in Data Tab, create a new measure, and name it User, with value of USERNAME();



I also like to add date/time of refreshing the report with DAX NOW() function (note that NOW() function will return server's current time, not your local. if you are interested to learn to how fetch local's current time read [here](#)). So let's create new measure and name it Now;



Now let's add two other table visualizations to the report. one for User, and another for Now. Here is the report's final view;

Year	Quarter	Month	Day	Sales Amount	Name
2015	Qtr 4	October	1	100	Reza Rad
2015	Qtr 4	October	2	50	John Martin
2015	Qtr 4	October	3	200	Reza Rad
Total				350	

Username	Name
reza@radacad.com	Reza Rad
reza@radacad.com	John Martin

User ▼
REZA-VAIO\Reza

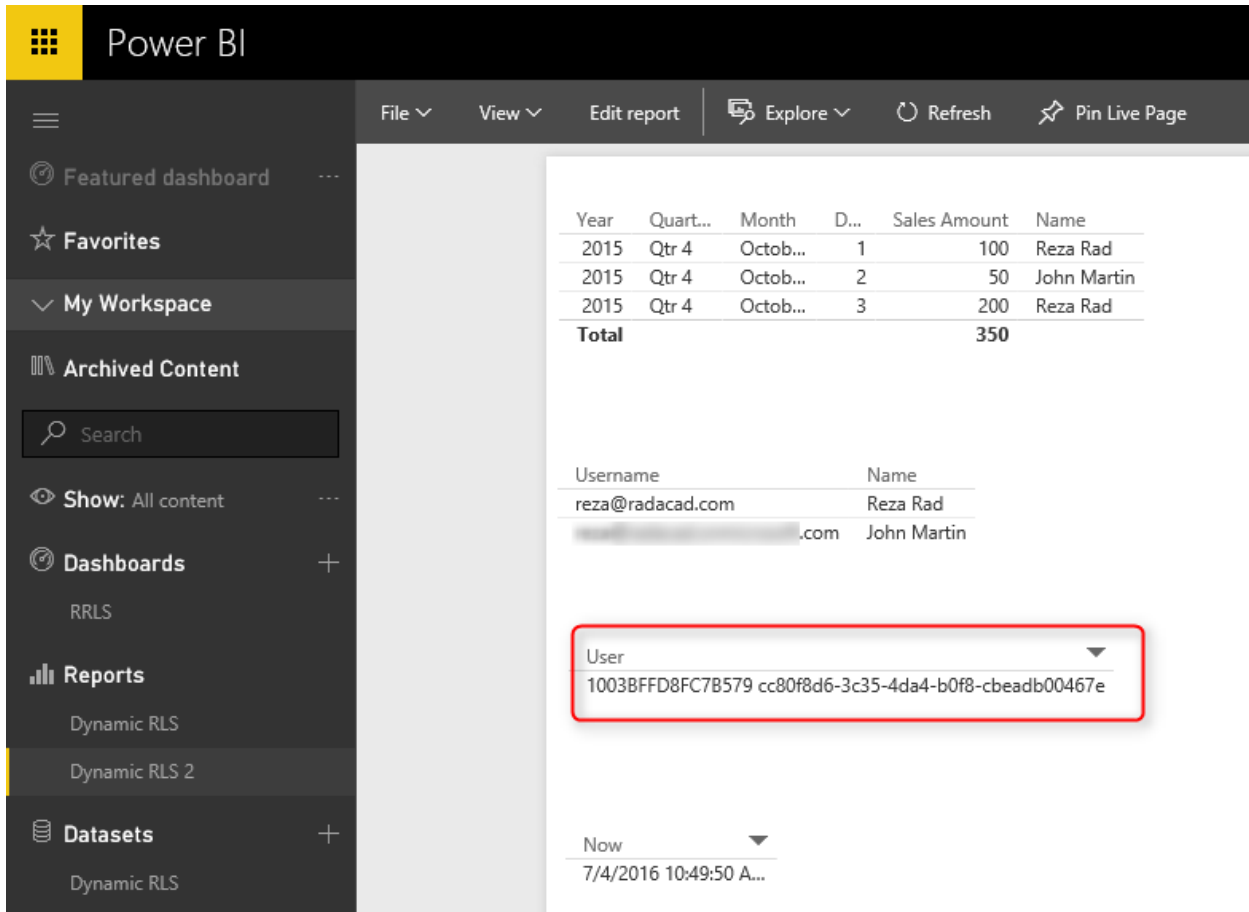
Logged In User

Now ▼
7/4/2016 10:42:33 PM

Refresh Date/Time

DAX USERNAME()

USERNAME() function in DAX returns the username of logged in user. However there is a small trick for it. If we don't set a row level security for our report. USERNAME() function will return user id which would be a unique identifier. To have an understanding of what I mean, publish your report to Power BI and browse it to see what you will see.



The screenshot shows the Power BI Desktop interface. On the left is the navigation pane with sections like 'Featured dashboard', 'Favorites', 'My Workspace', 'Archived Content', 'Dashboards', 'Reports', and 'Datasets'. The main area displays a report with two tables. The first table shows sales data by year, quarter, month, day, sales amount, and name. The second table shows user information. Below the tables, there is a 'User' dropdown menu with a red box highlighting the selected user ID: 1003BFFD8FC7B579 cc80f8d6-3c35-4da4-b0f8-cbeadb00467e. At the bottom, there is a 'Now' dropdown menu showing the current date and time: 7/4/2016 10:49:50 A...

Year	Quart...	Month	D...	Sales Amount	Name
2015	Qtr 4	Octob...	1	100	Reza Rad
2015	Qtr 4	Octob...	2	50	John Martin
2015	Qtr 4	Octob...	3	200	Reza Rad
Total				350	

Username	Name
reza@radacad.com	Reza Rad
john@radacad.com	John Martin

User
1003BFFD8FC7B579 cc80f8d6-3c35-4da4-b0f8-cbeadb00467e

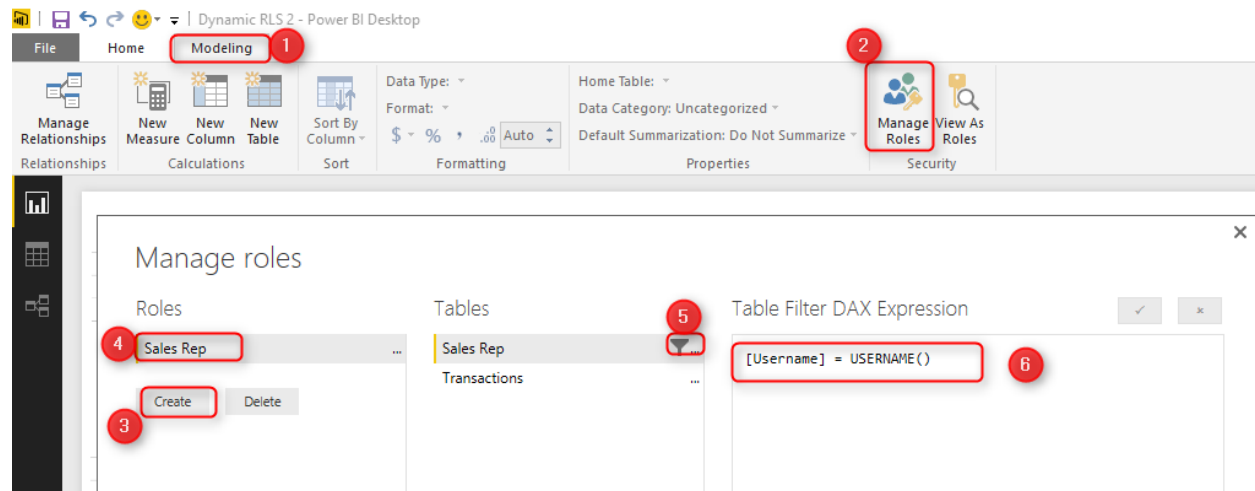
Now
7/4/2016 10:49:50 A...

Without a security configuration on your report you will see unique identifier for username which isn't useful. Now let's set up row level security and assign users to it to see how it works.

Row Level Security in Power BI Desktop

I have explained in another post how row level security in Power BI Desktop works, so if you like to learn it more in details read [this blog post](#). Here I will only use that technique to filter each role based on their username with DAX username() function. To create security go to Modeling tab (you need Power BI at least June 2016 update for this example), Manage Roles. Create a role and name it Sales Rep. and define a filter on Sales Rep table as below;

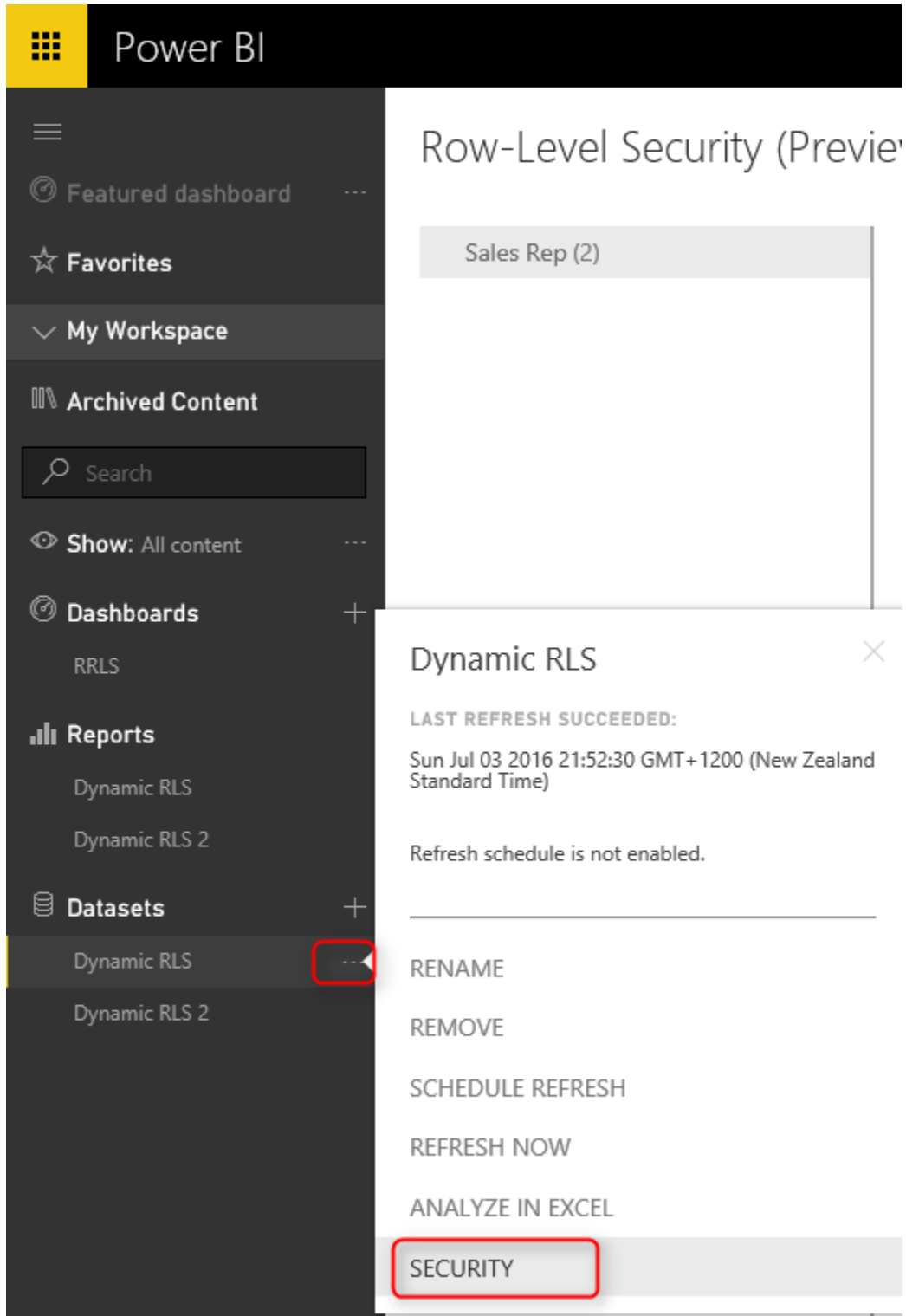
[Username] = USERNAME()



This filter simply means that logged in user will only see his/her own records in the whole data set. As you remember the username field in Sales Rep table defined exactly as usernames of Power BI accounts. and transactions table is also related to this table based on Sales Rep ID. so filtering one table will effect others. As a result this single line filter will enable dynamic row level security in the whole Power BI solution here.

Assign users to Power BI Security

Now Save, and publish your solution to Power BI. In Power BI service go to security setting of the data set you just published (I named this as Dynamic RLS).



The screenshot shows the Power BI web interface. On the left is a dark sidebar with navigation options: Featured dashboard, Favorites, My Workspace, Archived Content, a search bar, Show: All content, Dashboards (with a sub-item RRLS), Reports (with sub-items Dynamic RLS and Dynamic RLS 2), and Datasets (with sub-items Dynamic RLS and Dynamic RLS 2). The 'Dynamic RLS' item under Datasets is selected, and a red box highlights the three-dot menu icon to its right. A context menu is open, listing actions: RENAME, REMOVE, SCHEDULE REFRESH, REFRESH NOW, ANALYZE IN EXCEL, and SECURITY. The 'SECURITY' option at the bottom is highlighted with a red box. The main pane on the right is titled 'Row-Level Security (Preview)' and shows a table with one row labeled 'Sales Rep (2)'.

And in Security tab, add all users to the Sales Rep role.

Row-Level Security (Preview)

Sales Rep (2)

Members (2)

People or groups who belong to this role

Reza Rad X Enter email addresses

Add

Note that adding a user here doesn't mean that they will see data in the report. Remember that this security is dynamic, means that they will see their data rows ONLY if the underlying data set has a record for their username, and they will only see data rows related to their own username, not others.

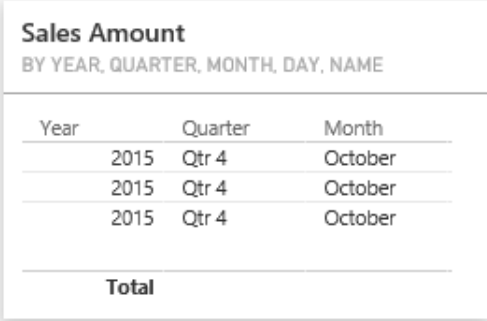
Now if you refresh the report in Power BI you will see actual usernames. because we already set up security for it, so it doesn't show unique identifiers anymore.

User ▼

reza@radacad.com

Share the Dashboard

Other users should have access to the dashboard and report first to see it, so create a dashboard from main table in the report, name the dashboard as RLS (or whatever you would like to call it);



Sales Amount		
BY YEAR, QUARTER, MONTH, DAY, NAME		
Year	Quarter	Month
2015	Qtr 4	October
2015	Qtr 4	October
2015	Qtr 4	October
Total		

Pin to dashboard ✕

Select an existing dashboard or create a new one.

Where would you like to pin to?

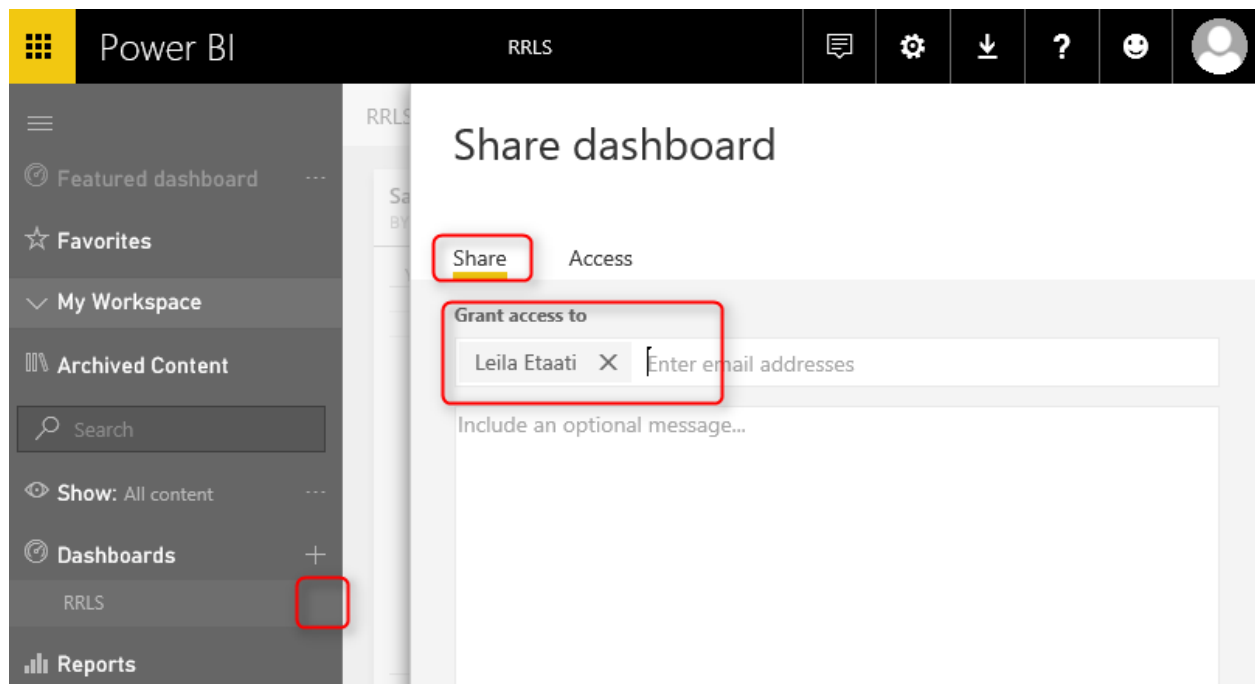
☐ Existing dashboard

☒ New dashboard

RLS

Pin
Cancel

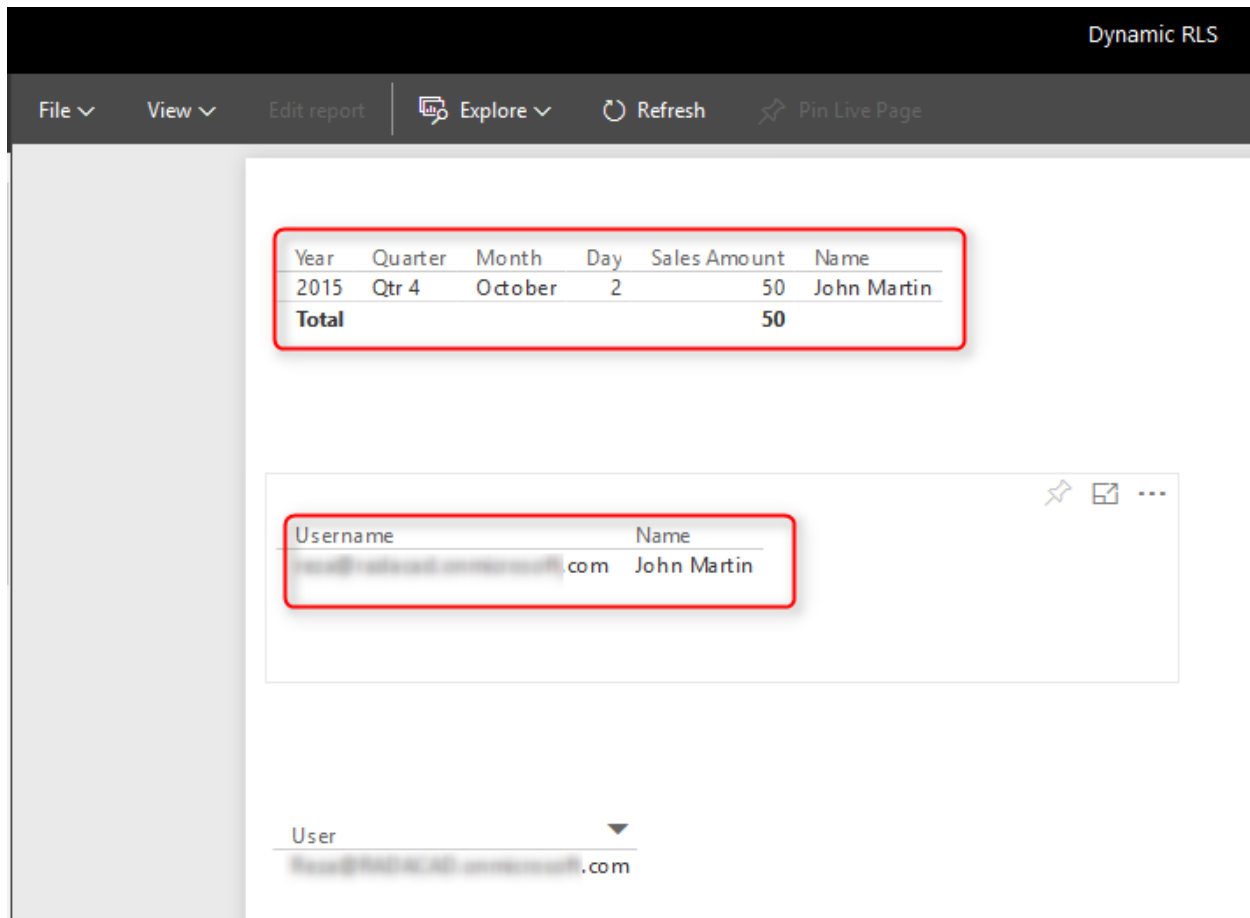
Now share the dashboard with other users



The screenshot shows the Power BI interface with the 'Share dashboard' dialog open. The 'Share' button is highlighted with a red box. The 'Access' tab is selected. The 'Grant access to' section shows a list of users, with 'Leila Etaati' highlighted and a red box around it. The 'Enter email addresses' input field is also visible. The 'Dashboards' section in the left sidebar is highlighted with a red box.

Test the Security

Now if other users open the report, and if their usernames matches one of the entries in Sales Rep table, they would see their own names, and data rows related to that in the report;



The screenshot shows a Power BI report interface with a dark header bar containing the text "Dynamic RLS". Below the header is a navigation bar with options: File, View, Edit report, Explore, Refresh, and Pin Live Page. The main content area displays a table with the following data:

Year	Quarter	Month	Day	Sales Amount	Name
2015	Qtr 4	October	2	50	John Martin
Total				50	

Below the table is a user profile card with the following information:

Username	Name
john@radacad.com	John Martin

At the bottom of the card, there is a "User" dropdown menu showing the current user: john@radacad.com.

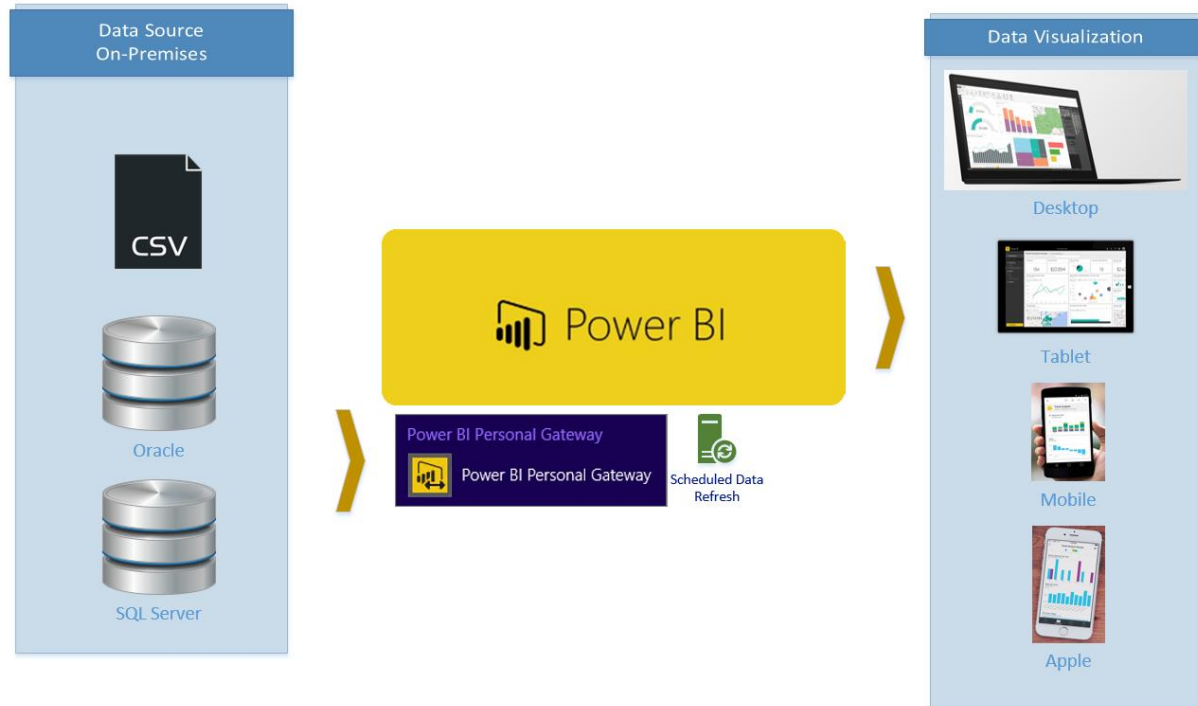
As you can see John Martin only see the transaction that he handled, and his own record in Sales Rep table. This is John's view of the Power BI report. While my view of this report would be totally different, I will see my two transactions and my own name under Sales Rep.

Summary

You have seen how easy is to use Dynamic row level security in Power BI using DAX USERNAME() function. With this method users will see their own view of the world. However you need to make sure that your Power BI model has relationship set up properly, otherwise people might see other table's data when there is no relationship between their profile table to those tables. Dynamic row level security is highly dependent to your data model, so keep your data model right.

Definitive Guide to Power BI Personal Gateway

Published Date: September 17, 2015



If you've followed my [Power BI from Rookie to Rockstar book](#) so far, you know that Power BI can reach on-premises data sources as well as cloud based. With Power BI Desktop you simply connect to on-premises data stores, however when the report published into Power BI website there should be a bus connection between Power BI (on cloud) and on-premises data stores (such as SQL Server, Oracle, SSAS Multi-Dimensional and so on). Here is where Power BI Personal Gateway come to play its role. Personal Gateway create the connection path from the data set in Power BI on cloud to the data store on -premises (on your organization server, or even your laptop!). Personal Gateway creates that connection line through Azure Service Bus.

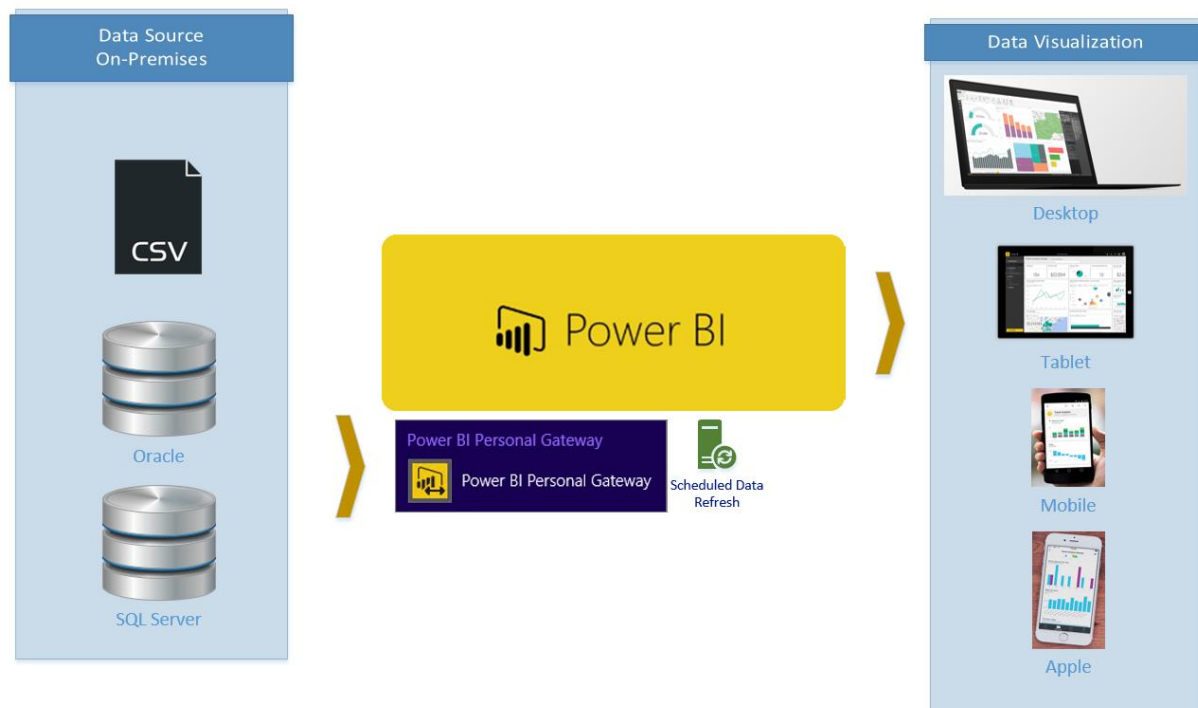
In this section you will learn

- In which scenarios Power BI Personal Gateway would be helpful?
- What are limitations of Personal Gateway?
- Myths and Misconceptions about Personal Gateway

We are in Get Data section of Power BI book and this is the best time to talk about Personal Gateway because most of the data sources in organizations I believe still are on-premises, and Power BI fortunately can connect to it. Let's start from the definition of Personal Gateway;

What Is Power BI Personal Gateway?

Power BI Personal Gateway is an application and service that creates the bus connection between Power BI data set on cloud to on-premises data store. This is a small application that you download, install and configure. You can use Personal Gateway to connect to any on-premises data stores that is listed in Power BI Get Data section except ODBC, Active Directory, and Microsoft Exchange. Screenshot below show a diagram of gateway role for Power BI and on-premises data stores;



Where Power BI Personal Gateway Would be Useful?

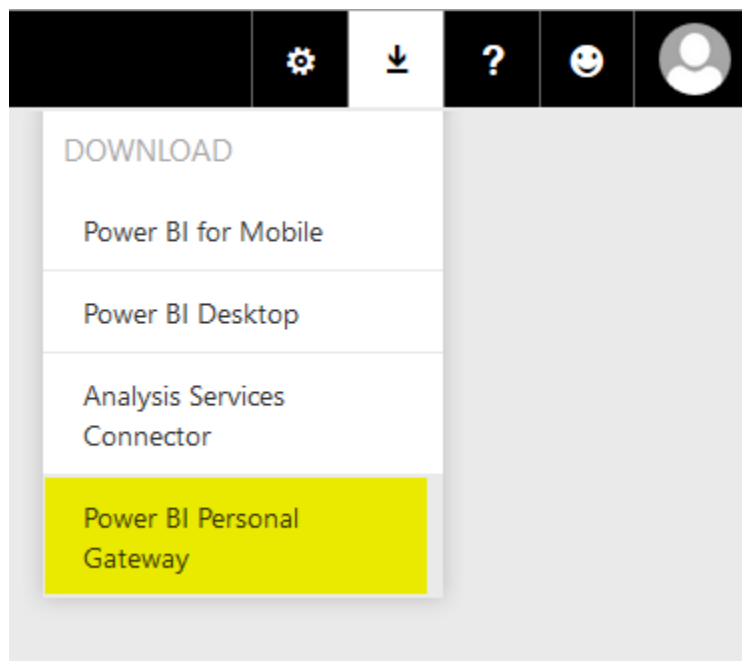
When you get data from an on-premises data store, and your on-premises data store is not ODBC, Active Directory, or Microsoft Exchange. So don't limit yourself, you can connect to SQL Server On-premises, SSAS multi-dimensional on-premises (You can do SSAS Tabular on-premises as well, but there is a better type of connection – live/direct connection – for it which would serve purposes much better).

What about Online Data?

If you get data from online data sources, such as CRM Online, Mailchimp and other Content packs of Power BI, data from a website and etc, then you don't need Personal Gateway. Online data can be refreshed as long as required credential is set up correctly. However remember if you are using combination of an online data store and on-premises, then you need personal gateway to get on-premises refreshed.

How to Install Power BI Personal Gateway?

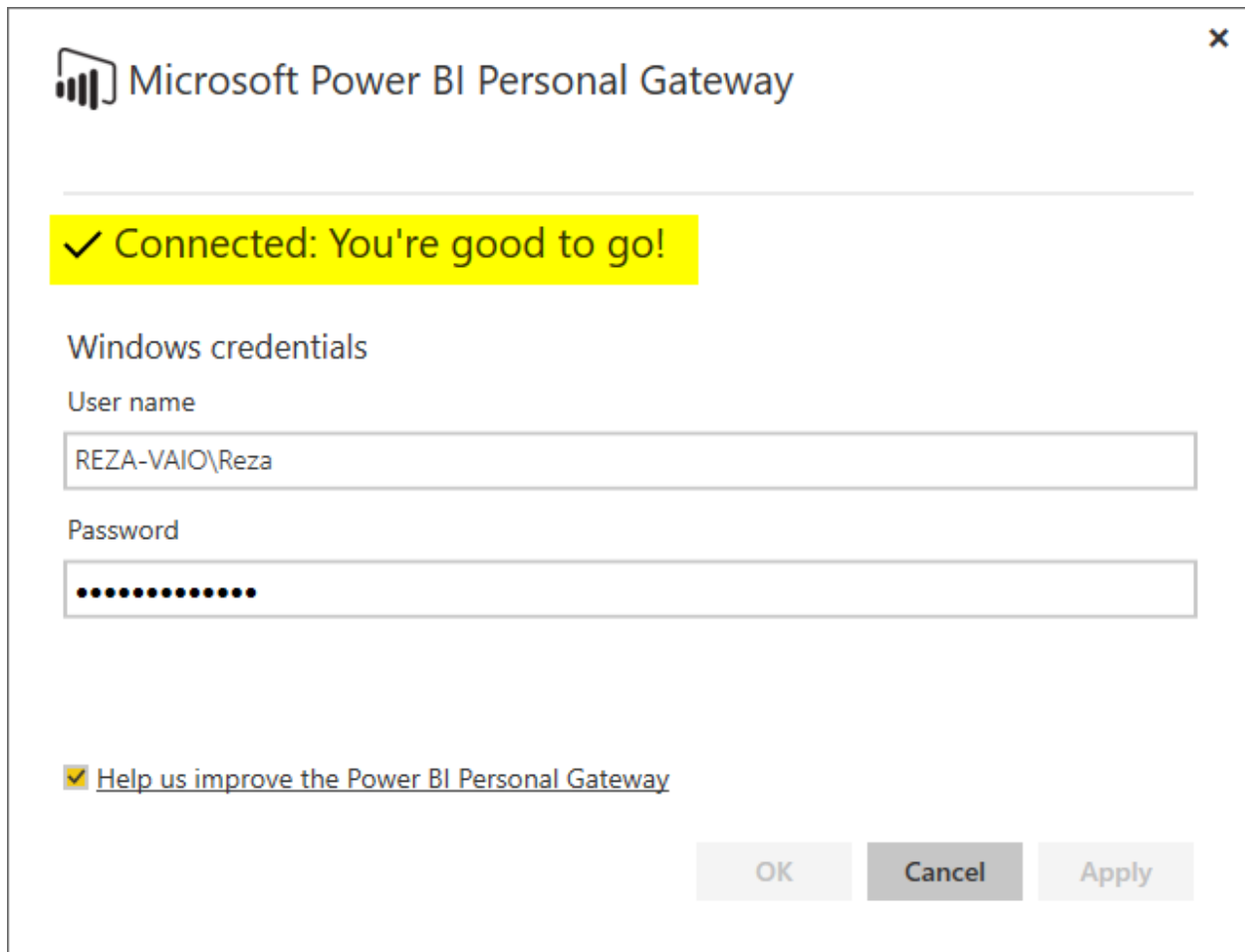
Installation is really easy and straight forward, you would be able to follow installation steps easily. Download it from the Power BI download menu option as screenshot below mentioned:



Or you can download it directly (Note that direct download link might not have the latest Power BI Personal Gateway version):

<http://go.microsoft.com/fwlink/?LinkId=534231>

After installation all you need to do is to configure it with a windows/Active Directory user and password. The Gateway then connected and running.



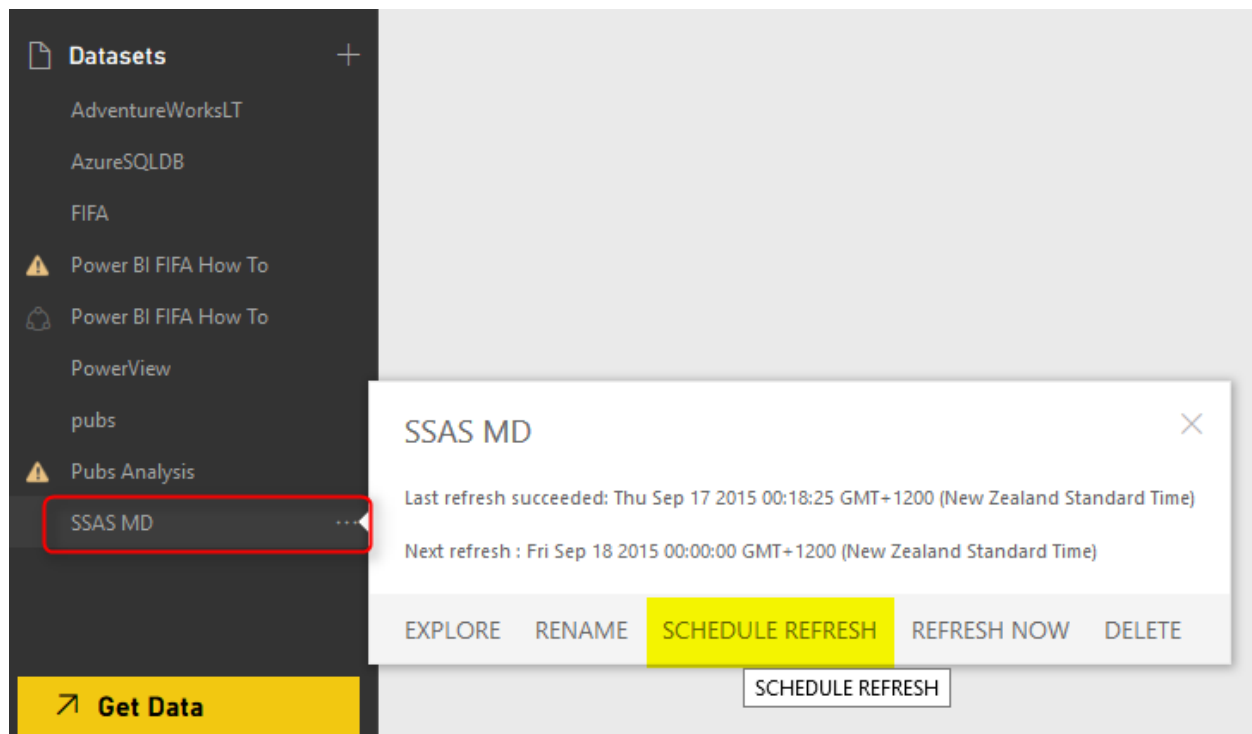
Do I Need More than One Instance of Gateway Installed?

No. One Gateway will provide service bus connection to all supported on-premises data stores on your machine. In other words; **One Gateway to Rule Them All** 😊

How to Set Scheduled Refresh in Power BI Website?

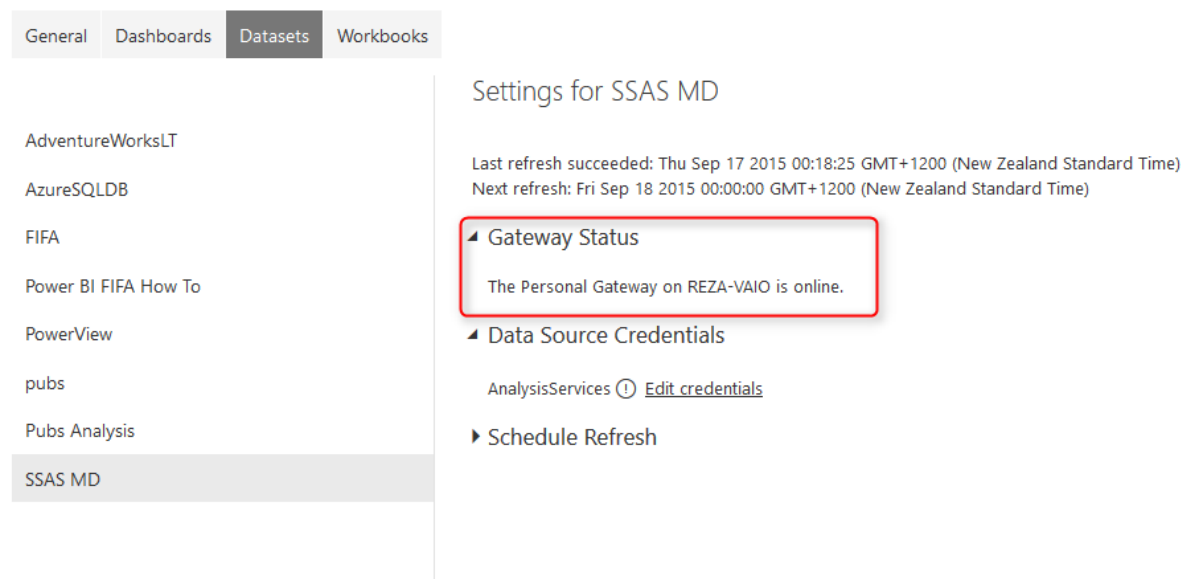
The whole purpose of Gateway is for being able to schedule a data refresh or be able to refresh the data set on Power BI website with Refresh Now button. You can Set a Scheduled Refresh in this way:

Click on ellipsis button on the right hand side of the data set in Power BI website



In the dataset setting you can see if Gateway is online or not (if there is no gateway installed, you can install it from here as well)

Settings



You can set credentials. Example below belongs to an Analysis Services Multi Dimensional data store. SSAS MD doesn't support basic authentication, but don't worry this bug has been reported, and will be fixed.

Configure SSAS MD ×

Server

Database

Authentication Method:

Windows ▼

For Windows authentication, you do not need to provide your Windows credentials

Sign In Cancel

Then you can simply schedule refresh as you want

Settings for SSAS MD

Last refresh succeeded: Thu Sep 17 2015 00:18:25 GMT+1200 (New Zealand Standard Time)

Next refresh: Fri Sep 18 2015 00:00:00 GMT+1200 (New Zealand Standard Time)

Gateway Status

The Personal Gateway on REZA-VAIO is online.

Data Source Credentials

AnalysisServices ⓘ [Edit credentials](#)

Schedule Refresh

Keep your data up-to-date

Yes



Refresh frequency

Daily ▼

Time Zone

(UTC+12:00) Auckland, Wellington ▼

Time

12 ▼ 00 ▼ AM ▼ ✕

[Add another time](#)

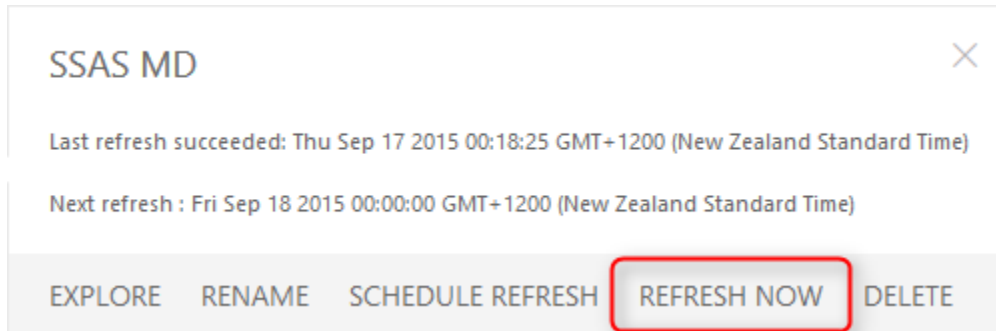
☒ Send refresh failure notification email to me

[Refresh History](#)

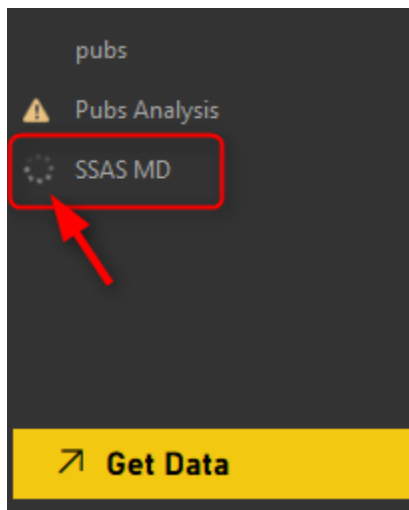
Apply

Discard

Not that you can also Refresh dataset anytime you want with Refresh Now option



When refresh happens you will see a refresh icon besides the data set



Limitations of Power BI Personal Gateway



This Gateway in its early life stages, suffer some limitations, which most of them will hopefully go away soon. Here are list of existing limitations;

- Limitation On Data Sources; As mentioned above, all data sources on-premises and supported by Power BI are supported except ODBC, Active Directory, and Microsoft Exchange.
- Gateway can be only installed on 64 bit machine.
- Gateway cannot be installed on a machine that has Power BI Analysis Services Connector Installed. (SSAS Connector will be discussed in another section)
- The machine that has gateway running, should not be turned off, shut down, or stand by. (At the time of scheduled refresh happening at least)
- Power BI Personal Gateway would be only available for Power BI Pro. (You can try trial version of it for free for 60 days I reckon)

Myths and Misconceptions About Power BI Personal Gateway

Gateway doesn't pass user credential from Power BI website

Gateway creates the service bus for cloud based Power BI data set to connect to the on-premises data stores. However the connection won't be dynamic. What I mean by Dynamic? I'll explain it here:

Let's consider that you have a SQL Server Analysis Services Multi-Dimensional set up. Let's assume that you have set up cubes for HR data marts. As you probably now HR contains sensitive data that needs to be filtered for each user/department based on their access level. someone from one department shouldn't be allowed to see someone else's information from another department. Fortunately in SSAS multi-dimensional this is supported through defining Roles and row level security applied on them. Then users and groups on active directory will be mapped to roles, and anytime they login through a client (let's say Excel PivotTable for example) their credential will pass through down to SSAS engine, and data will be returned only based on their authorized permissions defined in their role in SSAS.

This scenario works perfectly when client tool or user report is on-premises, such as Excel, or even SSRS reports on your organizational servers can leverage this method

using Kerberos installed and configured. However when client tool is on cloud then things should be set up differently. Power BI report is a cloud based tool which connects to on-premises data store (In this example SSAS multi-dimensional) through Personal Gateway. So you might end up with this conclusion that the user credential from the person who logged in to Power BI website will pass through the gateway down to SSAS on-premises and he/she would access only to their allowed data portion defined in SSAS Roles. This is a misconception however.

Unfortunately Power BI Personal Gateway doesn't pass user credential from cloud to on-premises. Personal Gateway actually works with a SAVED credential (the one that you defined at the time of configuring Personal Gateway), and users will have access to any data that the saved credential does. So as a result data level Role-based security with personal gateway is not possible (I mean for on-premises data stores, and only with Personal Gateway). For SSAS Tabular things are not similar however, Fortunately SSAS Tabular supported through direct connection with Power BI Analysis Services Connector. SSAS Connector passes credentials through, and credential works with Active Directory and Azure Active Directory (I'll explain that in separate section).

Same principal applies on any on-premises data stores that Power BI connects to it through personal gateway. In Nutshell Power BI Personal Gateway doesn't pass credentials to data stores.

Personal Gateway is not a Live Connection

The other misconception that you might have is that the connection to on-premises with Personal Gateway is Live. The actual fact is that connection is off-line, and is not live. the data set needs to be refreshed anytime for the data to be updated.

You Can Read Data From On-Premises Data Stores Not Only Databases But Also Files

Personal Gateway is not just for databases. You can connect to a CSV file stored on-premises shared folder on your organization server or even your laptop. As long as the saved credential in personal gateway has access to it, Power BI can read it.

Personal Gateway and Analysis Services Connector Are Different!

I've heard this question a lot! Is Personal Gateway same as AS connector? The answer is No. You probably concluded that so far after reading this section. But they are two separate applications, here is their purpose:

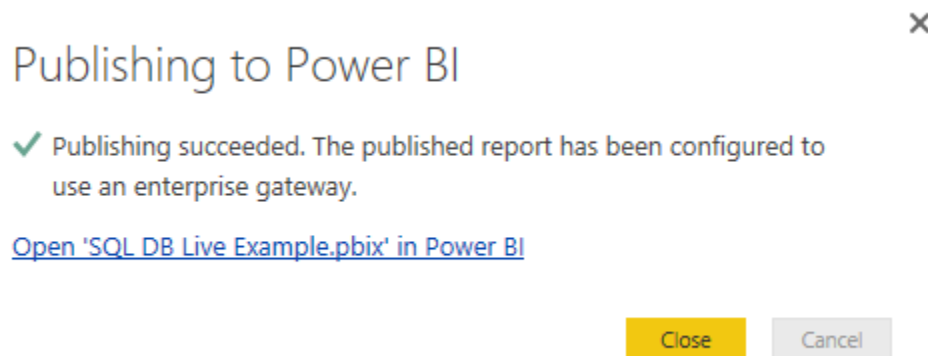
6. **Power BI Personal Gateway** is for connecting on-premises data stores (except those that mentioned above) in an off-line connection to Power BI
7. **Power BI Analysis Services Connector** is for connection only and only SQL Server Analysis Services Tabular on-premises LIVE and Directly to Power BI. (stay tuned for a whole section about Power BI AS Connector)

I would like to mention the Power BI site's article about Personal Gateway as well, where you can study more about it;

<https://support.powerbi.com/knowledgebase/articles/649846-power-bi-personal-gateway>

Step by Step Walk through: On-Premises Live SQL Server Connection with Power BI Enterprise Gateway

Published Date: July 1, 2016



I have mentioned previously that one of the ways to [step beyond Power BI 10GB limitation](#) is to use Power BI with Live Connection. In this post I will explain in a full step by step walk through how to use SQL Server as a DirectQuery or Live source for Power BI, and how to set up Enterprise Gateway to work with that. If you like to learn more about Power BI, read Power BI online book; [Power BI from Rookie to Rock Star](#). There are many resources explaining Power BI live connection, or how to use Enterprise Gateway. In this post I'll explain all in one; set up Enterprise Gateway and create a live Power BI report from On-premises SQL Server.

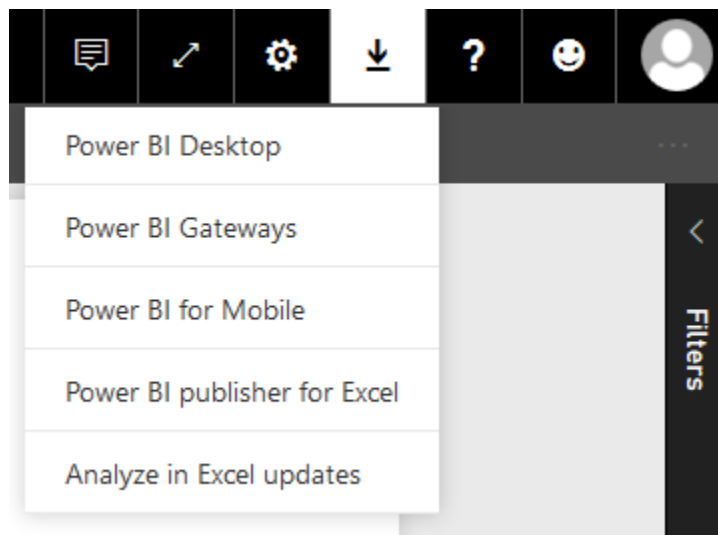
Prerequisite

For this example I will be using AdventureWorksDW2014 sample database. You can download this sample database from [URL here](#). and restore the database backup. [here](#) is the link to backup file.

Install and Configure Enterprise Gateway

Purpose of gateways in Power BI is to create connection from Power BI cloud service to on-premises data sources. There are two types of gateways; Enterprise, and Personal. Naming of these gateways are a bit misleading. Personal doesn't mean you have to install gateway on your laptop only, and Enterprise doesn't mean only for organizations. There are more differences which I will talk in different post. In general Enterprise is built for more live connections specially to Analysis Services, and Personal is more for importing data from many data sources. Let's leave details of that for now. For this example we will be using Enterprise Gateway.

You can download latest version of Enterprise Gateway from [PowerBI.Microsoft.Com](https://powerbi.microsoft.com) website under downloads



Then from gateways download page, choose Enterprise gateway, and download it.

Choose the gateway that best fits your needs

For personal use

Refresh your on-premises data quickly without waiting for an IT Admin. Designed for use with personal data sets, there is no central monitoring capabilities for this gateway.

Power BI Gateway - Personal

[Download](#)[Learn more](#)

For enterprise deployments

This gateway is used by organizations to serve a large number of users. It also enables administrators to set up access control for individual data sources and monitor usage.

Power BI Gateway - Enterprise

[Download](#)[Learn more](#)

Installation of Enterprise Gateway is just following a wizard, so continue that till end of the wizard where it asks for Power BI login. Sign in to your Power BI account;



Power BI Gateway – Enterprise

[?](#) [x](#)

This wizard helps you configure a Power BI Gateway.

Setup was successful!

To configure the gateway, you must first sign in to Power BI.

[Sign in to Power BI](#)[Cancel](#)

After sign in, configure the Power BI Gateway with a name, and key, and then you should be good to go. If you see green checkbox and Connected means your configuration was successful.



Power BI Gateway – Enterprise

[?](#) [x](#)

The Power BI Gateway has been setup and configured.

✓ Connected: BIRADACAD is good to go.



Next Step:

Add data sources to this Gateway in the Power BI service.



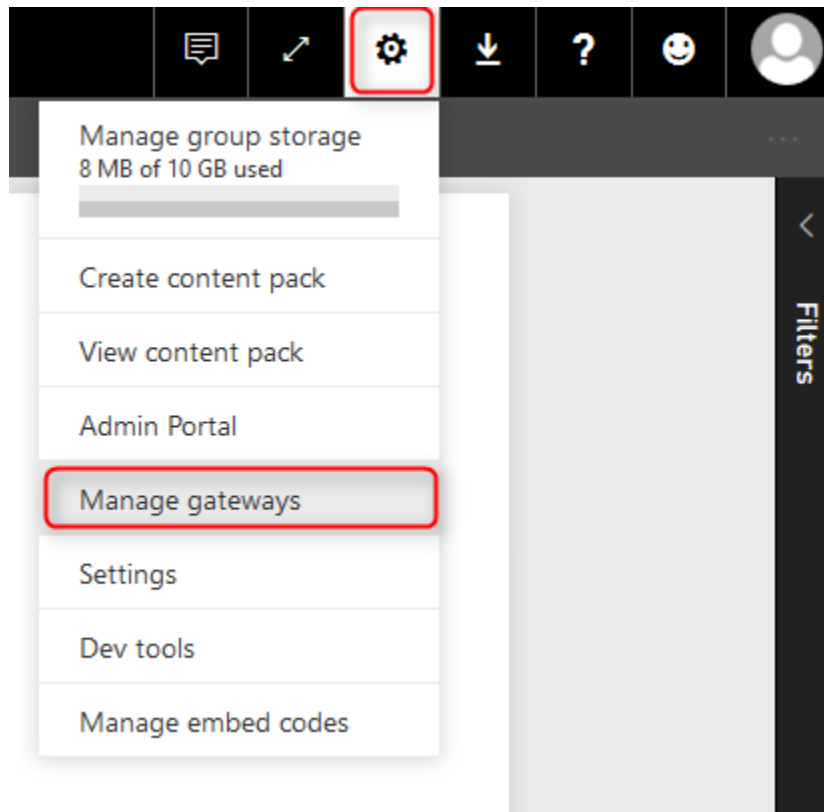
Help improve Power BI Gateway by sending usage information to Microsoft.

[Read the privacy statement online](#)

Close & Add data sources

As you can see after successful configuration, you can close the gateway (don't worry it is running behind the scene as a service), and add a data source to this gateway in Power BI Service.


Or alternatively you can directly go to Power BI website, after login go to setting menu option and choose Manage Gateways



In Gateway management window, you can see all gateways you have set up. My gateway here is named BIRADACAD, and I can see that it is successfully connected.

Gateways

+ ADD DATA SOURCE

> RezaSurface 

> BIRADACAD

Test all connections

Gateway Settings

Administrators

✓ Online: You are good to go.

Gateway Name

BIRADACAD

Department

Description

Contact Information

Reza@BIRADACAD.com

Apply

Discard

Create Data Source

Now Let's create Data Sources. You might think that one gateway is enough for connecting to all data sources in a domain. That is absolutely right, however you still need to add a data source under that gateway per each source. each source can be a SQL Server database, Analysis Services database and etc. For this example we are building a data source for SQL Server on premises. Before going through this step; I have installed AdventureWorksDW2014 on my SQL Server, and want to connect to it. If you don't have this database, follow instruction in prerequisite section.

For creating a data source, click on Add Data Source in manage gateways window (you have to select the right gateway first)

Gateways

+ ADD DATA SOURCE

> RezaSurface ⚠


> BIRADACAD


Test all connections


Then enter details for the data source. I named this data source as AdventureWorksDW2014, I enter my server name, and database name. then I use Windows authentication with my domain user <domain>\username and the password. You should see a successful message after clicking on Apply.

Gateways

+ ADD DATA SOURCE


RezaSurface 

New data source 

Excel On Prem 

BIRADACAD

SSAS MD PBIRAD

SSAS Tab PBIRAD 

AdventureWorksDW2014

Test all connections

Data Source Settings

Users

✓ Connection Successful

Data Source Name

AdventureWorksDW2014

Data Source Type

SQL Server

Server

PBIRAD

Database

AdventureWorksDW2014

Authentication Method

Windows

The credentials are encrypted using the key stored on-premises on the gateway server. [Learn more](#)

Username

.....

Password

.....

>Advanced settings

Apply Discard

Now that we have data source ready, let's build a very simple Power BI report and publish it to the service.

Building Sample Power BI Report

Open Power BI Desktop, and Get Data from SQL Server Database

Get Data

All


File


Database


Azure


Online Services


Database

 SQL Server Database

 Access Database

 SQL Server Analysis Services Database

 Oracle Database

 IBM DB2 Database

Enter server name, and database name, and choose the connection type as DirectQuery. DirectQuery connection is a live connection to the source. this type of connection is not available for all data sources. Fortunately it is available for SQL Server. Live Connection won't import data into Power BI. It will only bring metadata, and the report every time for refreshing the content will query the data source directly.

SQL Server Database

Import data from a SQL Server database.

Server

PBIRAD

Database (optional)

AdventureWorksDW2014

☐ Import

☒ DirectQuery

▶ Advanced options

OK

Cancel

You will be then asked for credentials to connect to the database if it is for the first time. use your windows authentication there, and then you will be redirected to Navigator window. For this example let's only select DimDate, and FactInternetSales. Click on Load.

Navigator

Display Options ▾

▲ PBIRAD: AdventureWorksDW2014 [39]

☐ vAssocSeqLineItems

☐ vAssocSeqOrders

☐ vDMPrep

☐ vTargetMail

☐ vTimeSeries

☐ AdventureWorksDWBuildVersion

☐ DatabaseLog

☐ DimAccount

☐ DimCurrency

☐ DimCustomer

☒ DimDate

☐ DimDepartmentGroup

☐ DimEmployee

☐ DimGeography

☐ DimOrganization

☐ DimProduct

☐ DimProductCategory

☐ DimProductSubcategory

☐ DimPromotion

☐ DimReseller

☐ DimSalesReason

☐ DimSalesTerritory

☐ DimScenario

☐ FactAdditionalInternationalProductDescription

☐ FactCallCenter

☐ FactCurrencyRate

☐ FactFinance

☒ FactInternetSales

☐ FactInternetSalesReason

FactInternetSales

Preview downloaded on Monday

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	ProductSubcategoryKey
310	20101229	20110110	20110105	21768	
346	20101229	20110110	20110105	28389	
346	20101229	20110110	20110105	25863	
336	20101229	20110110	20110105	14501	
346	20101229	20110110	20110105	11003	
311	20101230	20110111	20110106	27645	
310	20101230	20110111	20110106	16624	
351	20101230	20110111	20110106	11005	
344	20101230	20110111	20110106	11011	
312	20101231	20110112	20110107	27621	
312	20101231	20110112	20110107	27616	
330	20101231	20110112	20110107	20042	
313	20101231	20110112	20110107	16351	
314	20101231	20110112	20110107	16517	

The data in the preview has been truncated due to size limits.

<

>

Select Related Tables

Load

Edit

Cancel

As you can see Live connection will be created instead of importing data

Create Connections

- DimDate

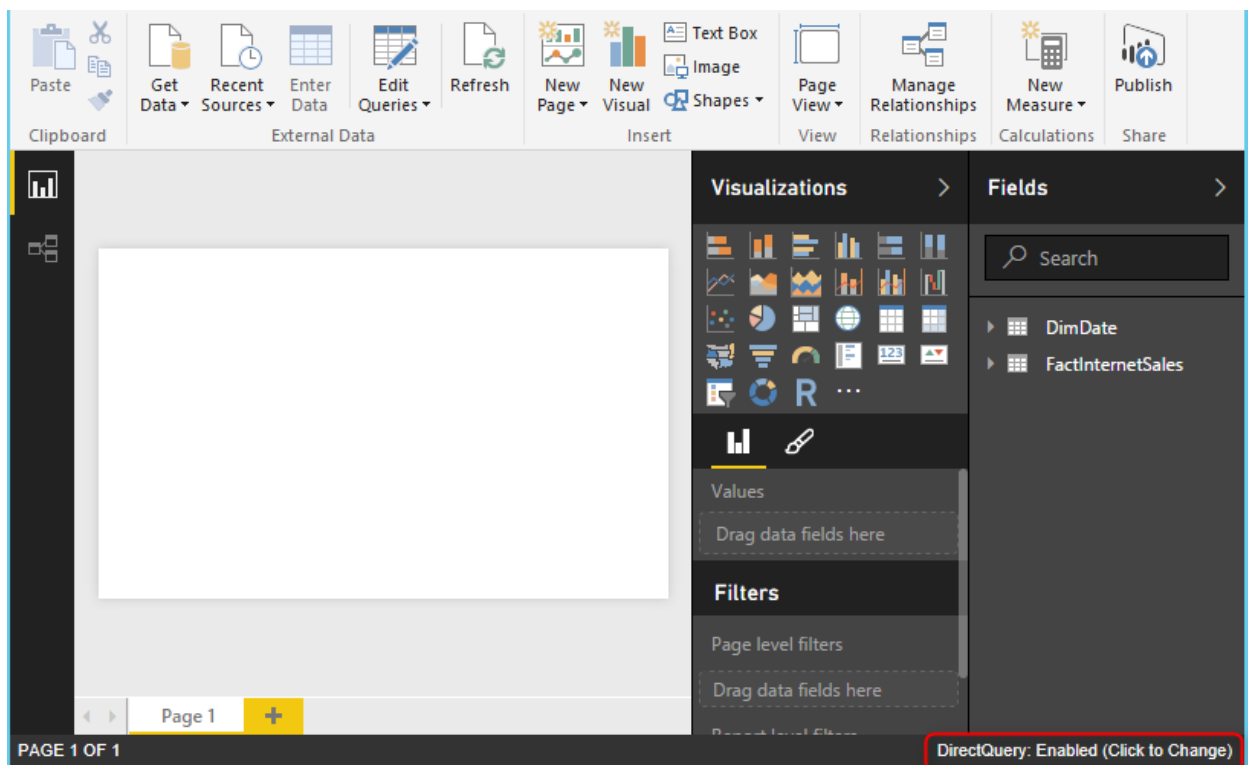
Creating live connections...

- FactInternetSales

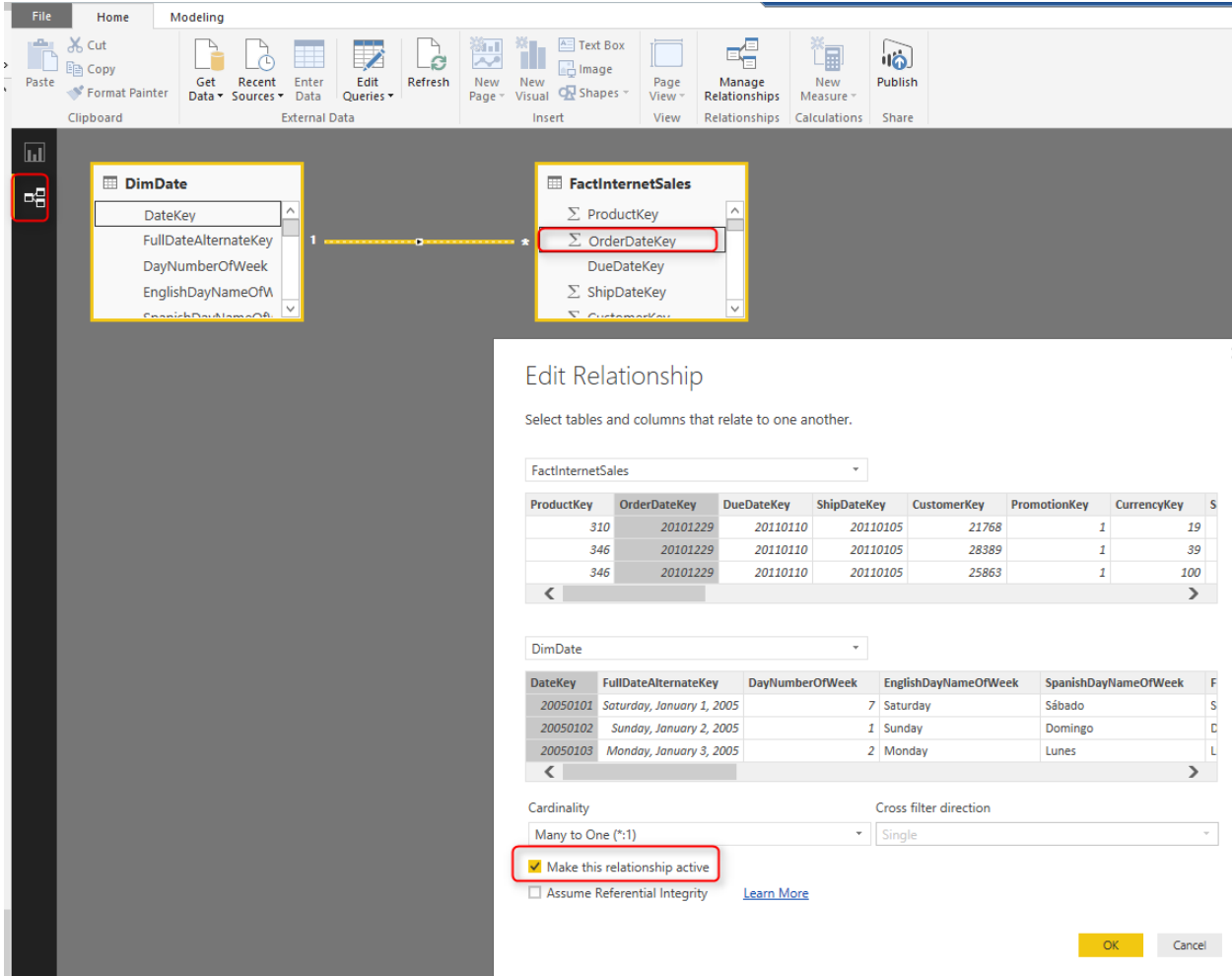
Evaluating...

Cancel

You can also see in the down right hand side of the Power BI that it mentioned the connection is DirectQuery: Enabled



Now go to Relationship tab. Here you can see DimDate and FactInternetSales are connected to each other with three relations. Because there are three date fields in FactInternetSales; DueDate, OrderDate, and ShipDate. For this example remove relationship for DueDate and ShipDate, and only keep the relationship for OrderDate, and change it to be Active Relationship.



DimDate

- DateKey
- FullDateAlternateKey
- DayNumberOfWeek
- EnglishDayNameOfW
- SpanishDayNameOfW

FactInternetSales

- ProductKey
- OrderDateKey
- DueDateKey
- ShipDateKey
- CustomerKey

Edit Relationship

Select tables and columns that relate to one another.

FactInternetSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	S
310	20101229	20110110	20110105	21768	1		19
346	20101229	20110110	20110105	28389	1		39
346	20101229	20110110	20110105	25863	1		100

DimDate

DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	F
20050101	Saturday, January 1, 2005	7	Saturday	Sábado	S
20050102	Sunday, January 2, 2005	1	Sunday	Domingo	D
20050103	Monday, January 3, 2005	2	Monday	Lunes	L

Cardinality: Many to One (*:1)

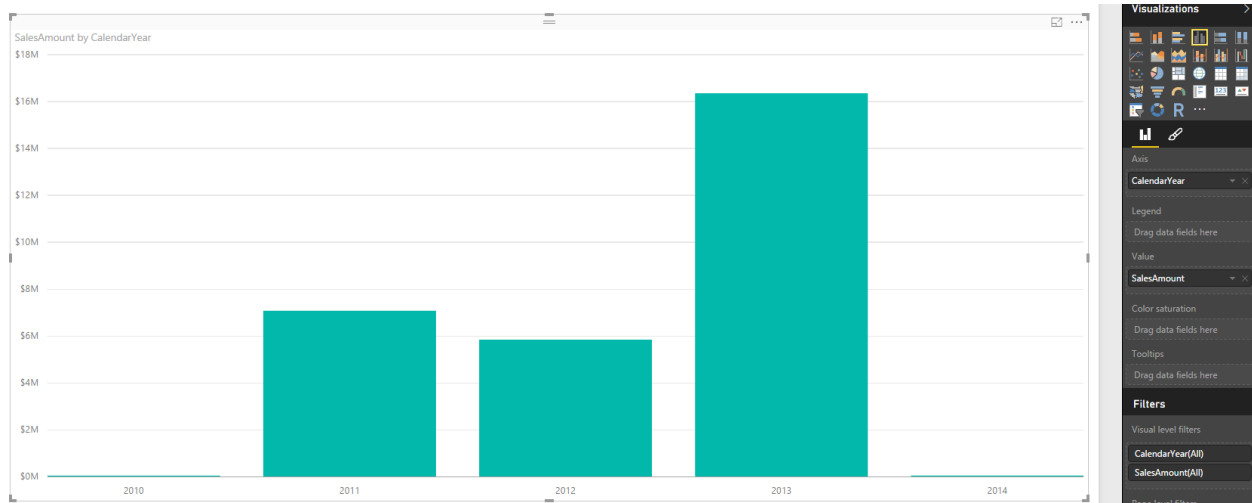
Cross filter direction: Single

☒ Make this relationship active

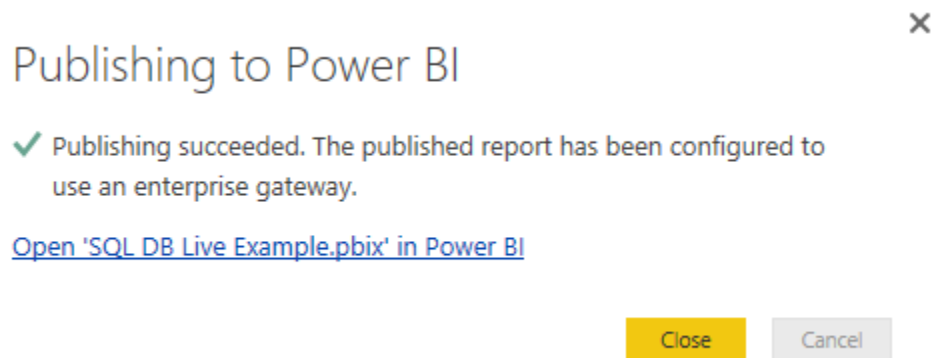
☐ Assume Referential Integrity [Learn More](#)

OK Cancel

Now build a very simple report with a column chart, that has Sales Amount (from FactInternetSales) as the Value, and CalendarYear (from DimDate) as Axis.

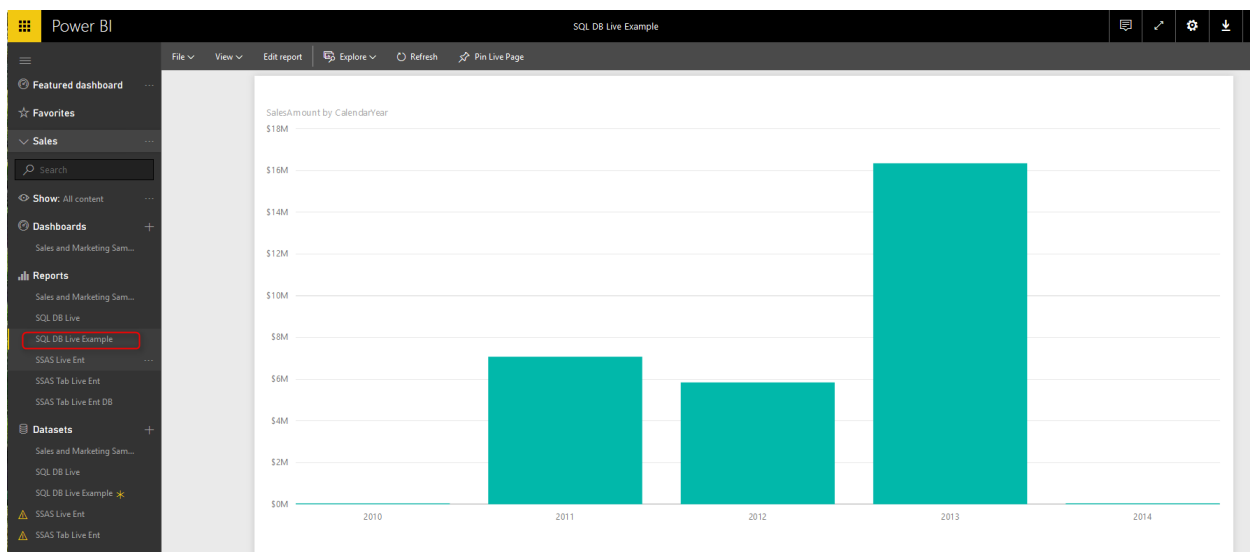


This is all we want to build for this example. One chart is enough to show the functionality. Save the report as something, for example; SQL DB Live Example, and Publish it. When you publish the report you will see a message saying that publishing succeeded, The Published report has been configured to use an enterprise gateway. This is good message showing that report's data source, and enterprise gateway data source are able to find each other 😊 Nothing more is required then.



Test the Live Report

To check the Enterprise gateway connection to on-premises SQL Server, the last step is to browse the report from Power BI website. You should be able to see the column chart. If you get any message regarding configuration issues, check enterprise gateway, and data source configurations to be set up correctly. Here is the report refreshed successfully;



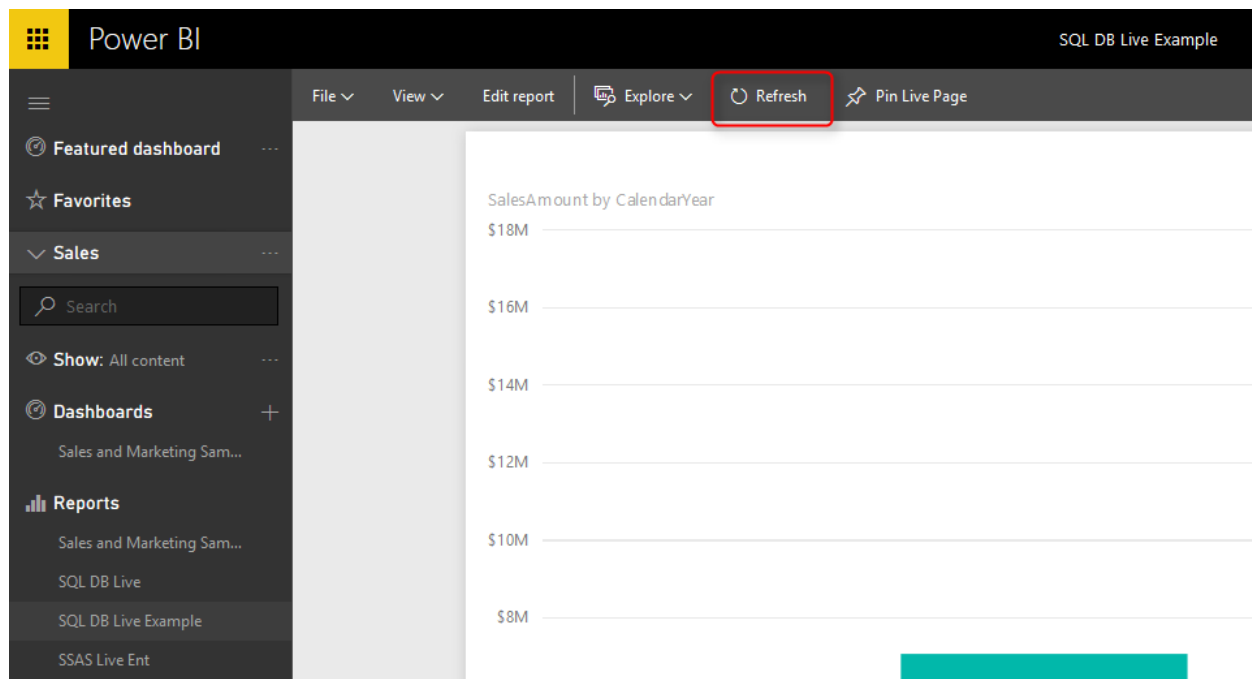
Live Refresh

This report is live which means you don't need to schedule a refresh for the data set. report directly connected to the data source, and with every change in data source there will be a change in the report. Let's do a small change by adding a record in fact table which increase sales amount of year 2014 about \$3M. here is the script for the change:

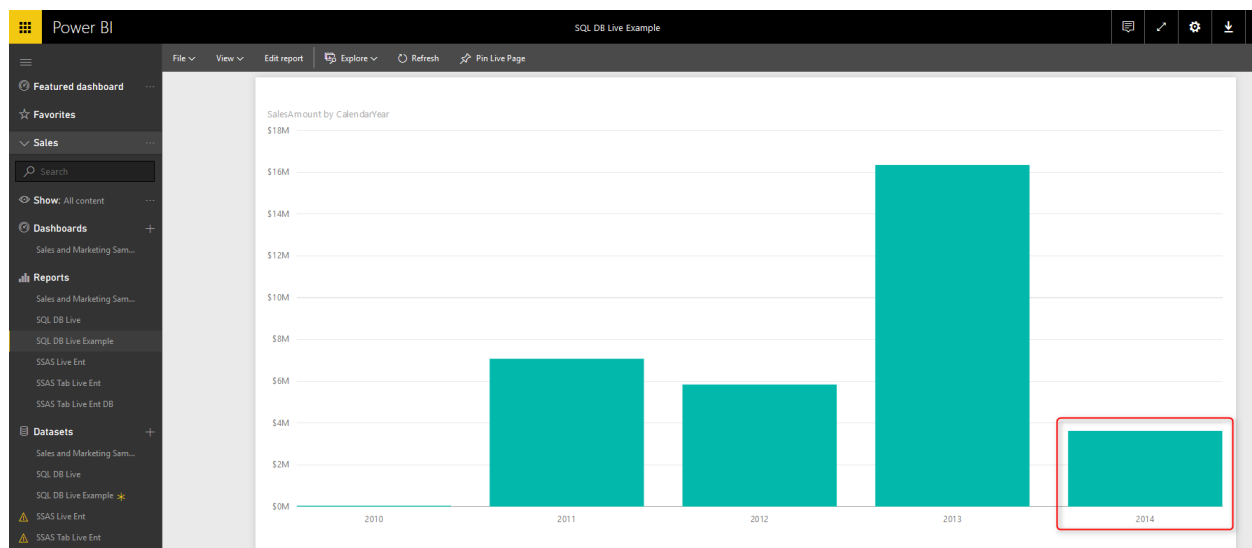
```
USE [AdventureWorksDW2014]
GO
```

```
INSERT INTO [dbo].[FactInternetSales]
    ([ProductKey]
    ,[OrderDateKey]
    ,[DueDateKey]
    ,[ShipDateKey]
    ,[CustomerKey]
    ,[PromotionKey]
    ,[CurrencyKey]
    ,[SalesTerritoryKey]
    ,[SalesOrderNumber]
    ,[SalesOrderLineNumber]
    ,[RevisionNumber]
    ,[OrderQuantity]
    ,[UnitPrice]
    ,[ExtendedAmount]
    ,[UnitPriceDiscountPct]
    ,[DiscountAmount]
    ,[ProductStandardCost]
    ,[TotalProductCost]
    ,[SalesAmount]
    ,[TaxAmt]
    ,[Freight]
    ,[CarrierTrackingNumber]
    ,[CustomerPONumber]
    ,[OrderDate]
    ,[DueDate]
    ,[ShipDate])
VALUES
    (314, 20140208, 20110220, 20110215, 16657, 1, 6, 9, 'SO4397133', 1, 1, 1,
3578.2700, 3578.2700, 0, 0, 2171.2942, 2171.2942, 3578000.2700, 286.2616, 89.4568,
NULL, NULL, '2011-02-08 00:00:00.000', '2011-02-20 00:00:00.000', '2011-02-15 00:00:00.000')
GO
```

Right after adding the record, without any delay, go and refresh the report (Not even the data set, just refresh the report in Power BI website),



And you will see the change there.



As you can see the column height for year 2014 changed now. Fantastic live report, isn't it?

Advantages of Live Connection

As I've mentioned previously, one of the biggest advantages of live connection is that you [step beyond limitation of Power BI](#) file size which is 10GB. You can have databases with Terabytes of data, and create live connection for it. That said, you have to take care

of performance of your database, because live queries are expensive (performance-wise I mean). As an example have a look [here](#) at two queries on same size of data, but different response time (one of them only performance tuned a bit with clustered column-store index).

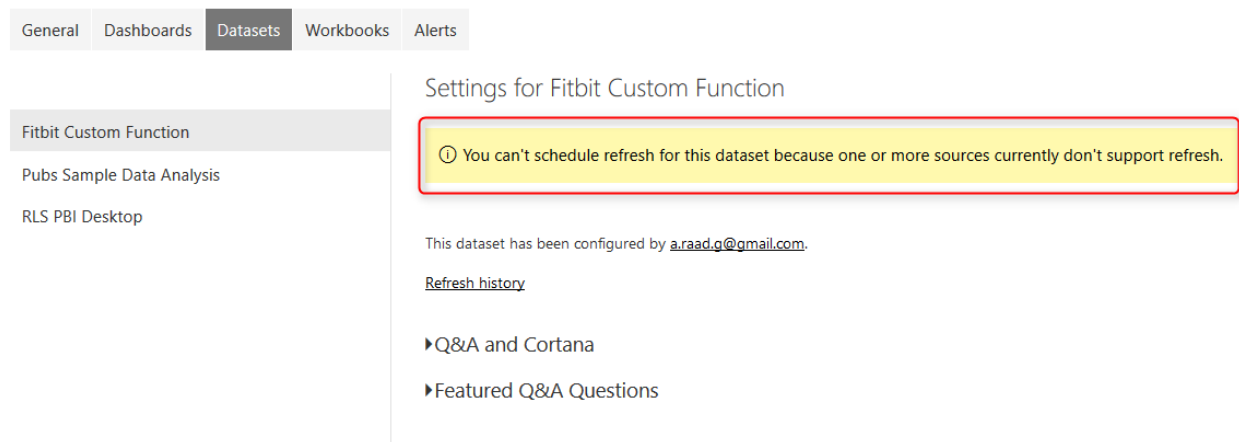
DirectQuery or Live SQL Server On-Premises Limitations

With live connection you won't have the Modeling (or Data) tab in Power BI, you can't create DAX calculated columns or measures, You cannot do formatting for columns. However you have Power Query, and all transformations in Power Query will be translated into T-SQL when queries sent to database.

You cannot have multiple data sources in your Power BI report. Live connection only works with one database.

Loop Through On-Premises Files with Power BI and Schedule it to Refresh Automatically

Published Date: October 7, 2016



Previously I've explained a method of [looping through number of files in a folder](#) which used Custom Functions. This method is preferred method but not always working with Schedule Refresh and gateway setup, because scheduled refresh doesn't always support custom functions in the code. So in this post I show you a second method that loops through the files dynamically, and it doesn't use custom function, and you will see at the end of this post that schedule refresh works with this method. For better clarification of the scenario I will first do custom function approach, show you the problem with scheduled refresh, and then I'll explain about the second method. If you would like to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

Prerequisite

For running example of this post you need to download set of CSV files from here: [fitbitsourcefilesradacad](#)

Defining the Challenge

In this example we have number of CSV files which has list of my Fitbit activities, one file for each month. All under the same folder, all same structure. We want to apply number of similar steps to each files and loop through them and fetch the data all into single

data set. After building a report on top of that we want to publish it to Power BI service, and get that refreshed through a gateway configuration. As files has similar structure we are looking for a mechanism to apply same steps on all files.

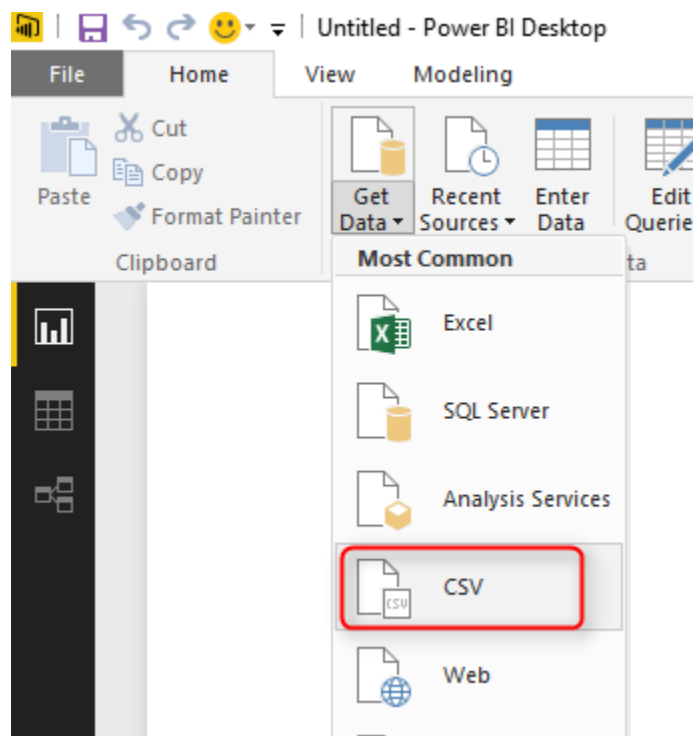
You will see two approaches with this, first one is preferred approach because it uses custom functions, however because scheduled refresh doesn't support custom functions always, so there is a second approach which will work with automatic refresh. If you are visiting this page at the time that problem of scheduled refresh and custom function resolved, then use the first approach, otherwise use the second.

First Approach: Custom Function

This is Preferred Approach but is not working with Schedule Refresh at the time of writing this post.

Process one File

Open a New Power BI Desktop, Start by Get Data, and Select CSV as a source.



Choose of the files in Fitbit Source folder and continue;

fitbit_export_201507.csv

File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10
Activities									
Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Minutes Extremely Active
01-07-2015	3,953	13,361	10.64	16	1,058	196	71	87	2,354
02-07-2015	3,432	10,818	8.76	10	726	224	19	45	1,875
03-07-2015	4,102	15,537	12.38	16	701	237	55	88	2,567
04-07-2015	3,959	14,490	11.54	6	685	255	46	69	2,445
05-07-2015	4,028	14,652	11.68	3	1,003	283	93	61	2,567
06-07-2015	3,538	12,161	9.84	16	1,144	168	76	52	1,954
07-07-2015	3,550	11,103	8.99	17	785	250	24	41	1,954
08-07-2015	3,238	10,029	8.22	10	1,204	152	36	48	1,567
09-07-2015	3,994	13,951	11.29	10	733	183	85	96	2,445
10-07-2015	3,998	12,516	9.5	9	650	194	92	92	2,567
11-07-2015	2,802	6,750	5.18	14	594	164	14	29	1,175
12-07-2015	4,156	19,471	15.39	21	472	272	92	96	2,754
13-07-2015	3,523	14,730	11.73	17	654	208	39	73	1,954
14-07-2015	3,039	9,543	7.24	16	712	195	10	38	1,475
15-07-2015	2,750	6,663	5.06	15	1,222	206	6	6	1,075
16-07-2015	3,373	14,603	11.46	12	745	198	22	69	1,875
17-07-2015	3,948	15,787	11.98	29	649	252	23	115	2,445
18-07-2015	3,917	16,652	13.35	19	543	332	52	55	2,567

Buttons: Load, Edit, Cancel

Go to Edit Queries Window with clicking on Edit button, and You will see data previewed there.

Untitled - Query Editor

File Home Transform Add Column View

Close & Apply, New Source, Recent Sources, Enter Data, Data Source Settings, Manage Parameters, Refresh Preview, Advanced Editor, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, Data Type: Text, Merge Queries, Append Queries, Combine Binaries

Queries [1]

fitbit_export_201507

Table.TransformColumnTypes(Source,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}, {"Column5", type text}, {"Column6", type text}, {"Column7", type text}, {"Column8", type text}, {"Column9", type text}, {"Column10", type text}})

Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10
Activities									
Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Minutes Extremely Active
01-07-2015	3,953	13,361	10.64	16	1,058	196	71	87	2,354
02-07-2015	3,432	10,818	8.76	10	726	224	19	45	1,875
03-07-2015	4,102	15,537	12.38	16	701	237	55	88	2,567
04-07-2015	3,959	14,490	11.54	6	685	255	46	69	2,445
05-07-2015	4,028	14,652	11.68	3	1,003	283	93	61	2,567
06-07-2015	3,538	12,161	9.84	16	1,144	168	76	52	1,954
07-07-2015	3,550	11,103	8.99	17	785	250	24	41	1,954
08-07-2015	3,238	10,029	8.22	10	1,204	152	36	48	1,567
09-07-2015	3,994	13,951	11.29	10	733	183	85	96	2,445
10-07-2015	3,998	12,516	9.5	9	650	194	92	92	2,567
11-07-2015	2,802	6,750	5.18	14	594	164	14	29	1,175

Query Settings

PROPERTIES

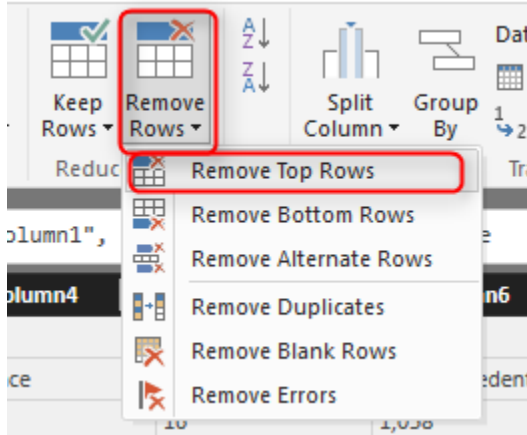
Name: fitbit_export_201507

APPLIED STEPS

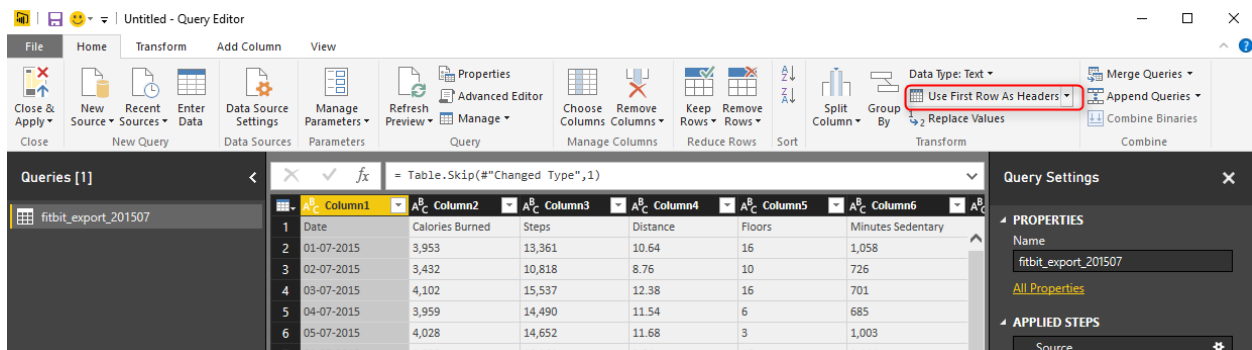
Source

Changed Type

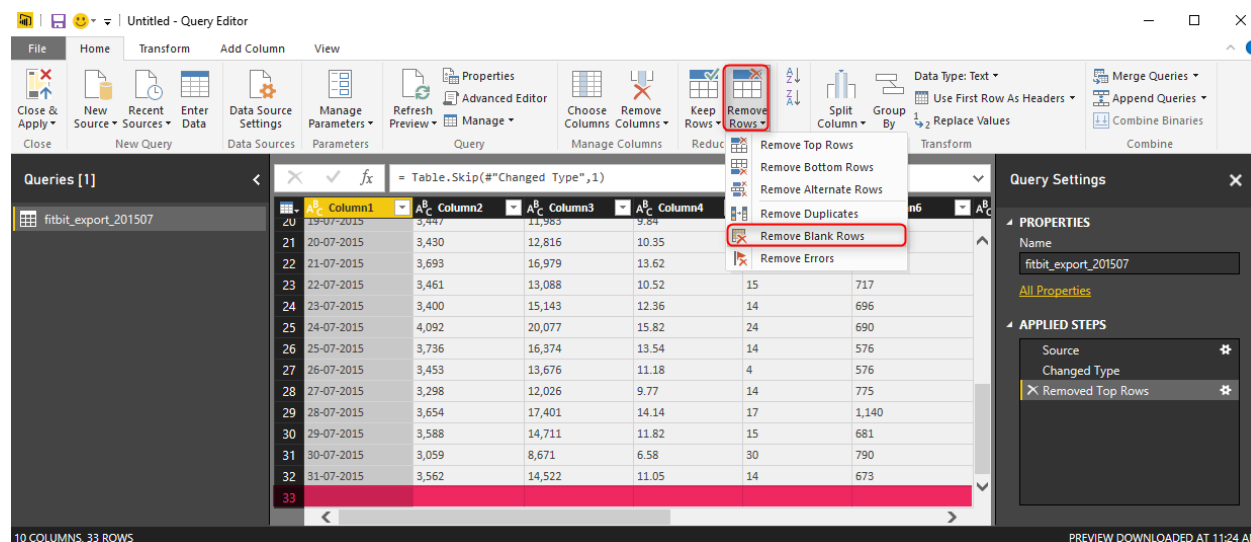
As you can see in the preview the first record isn't useful. It has just a title "Activities", so let's remove it with going to Remove Rows, Remove Top Rows. and enter 1 in the dialog box;



Now the first row only contains column headers. So let's promote it to column headers with clicking on Use First Row As Headers.



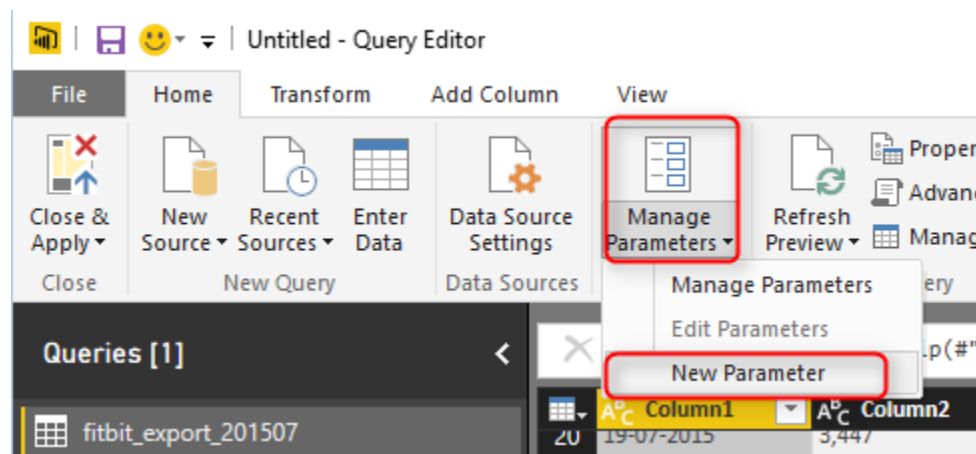
There are also some blank rows, so let's remove them with Remove Rows -> Remove Blank Rows.



Now we've done all clean up work for this file. We have one row per each day with details of activities in that day. Next step is to loop through all files in the source folder, and apply same number of steps on them. Fortunately we can use Custom functions for this. Previously [I've explained how Custom Functions](#) are helpful in code re-usability, better maintenance, and consistency. Instead of repeating same steps for each file is it much wiser to create a function from these steps, and call the function for each file. So let's do it;

Creating Parameter

For creating the function through GUI, the first step is to create a parameter. You can create Parameter from Menu option or with right click on the blank area in Queries section.



Create the Parameter with name of FullPath, of type Text, and set a default value to be full path (including file name, and extension) of one of CSV files in your folder;

Parameters

Parameters

New

FilePath

Name
FilePath

Description

☒ Required

Type
Text

Suggested Values
Any value

Current Value
refresh\FitbitSourceFilesRADACAD\fitbit_export_201507.csv

After creating the function you should be able to see that with Parameter icon in Queries pane;

Untitled - Query Editor

Text Tools

Transform

File Home Transform Add Column View Transform

Convert Transform

Queries [2]

fitbit_export_201507

FilePath (C:\Users\Reza\SkyDrive\Blog\L...

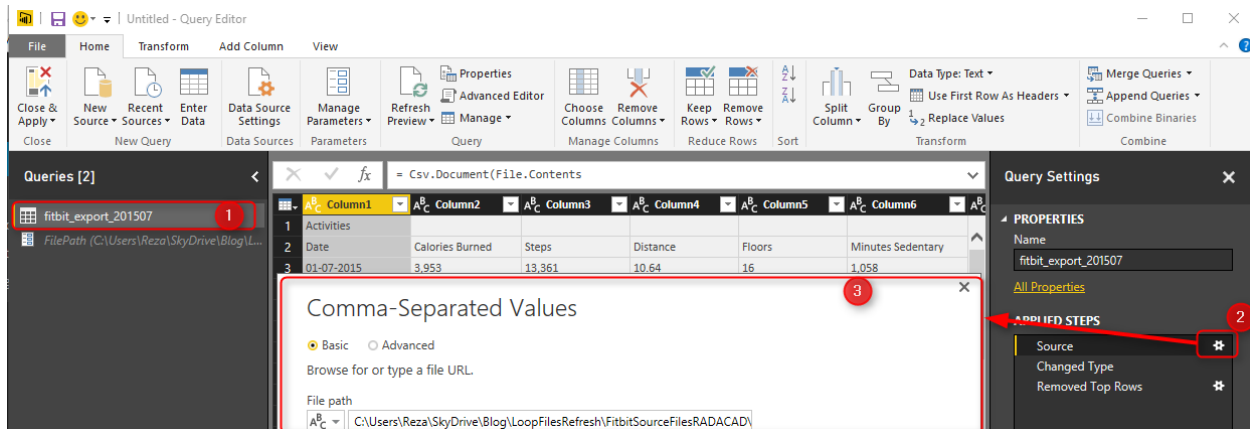
Current Value

C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSource

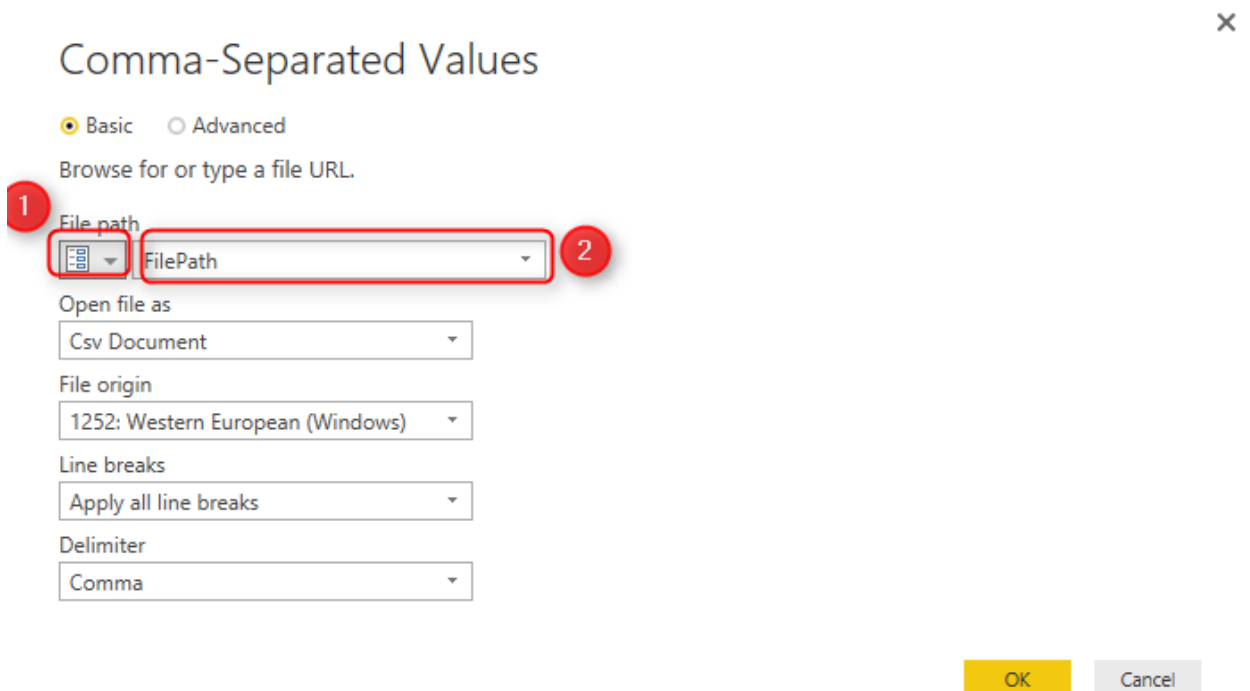
Manage Parameter

Creating Function

Next step is to get the current query (in the screenshot named as fitbit_export_201507) sourced from the FilePath parameter instead of static path. Click on the query, and choose settings from steps in right hand side pane;



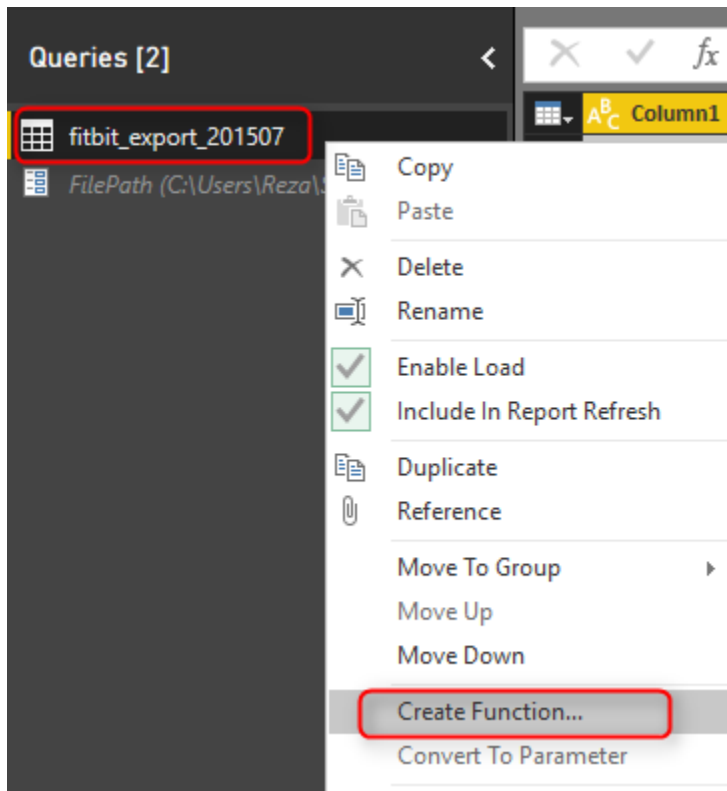
Change the File path in the setting window to be sourced from Parameter, and because you have only one parameter FilePath parameter will be selected automatically.



After clicking on OK, you will see result of first step before applying your changes, so simply click on the last step to see the final result. if the file path you have provided as

the default value of parameter be wrong you will see an error, otherwise you will see the data of that particular file. Now let's create the function;

Right click on the query that we just changed the source of it (fitbit_export_201705), and select Create Function.



Set the function name as ProcessFile, and leave the parameter as is.

Create Function

Enter names for the new function and its parameters.

Function name

ProcessFile

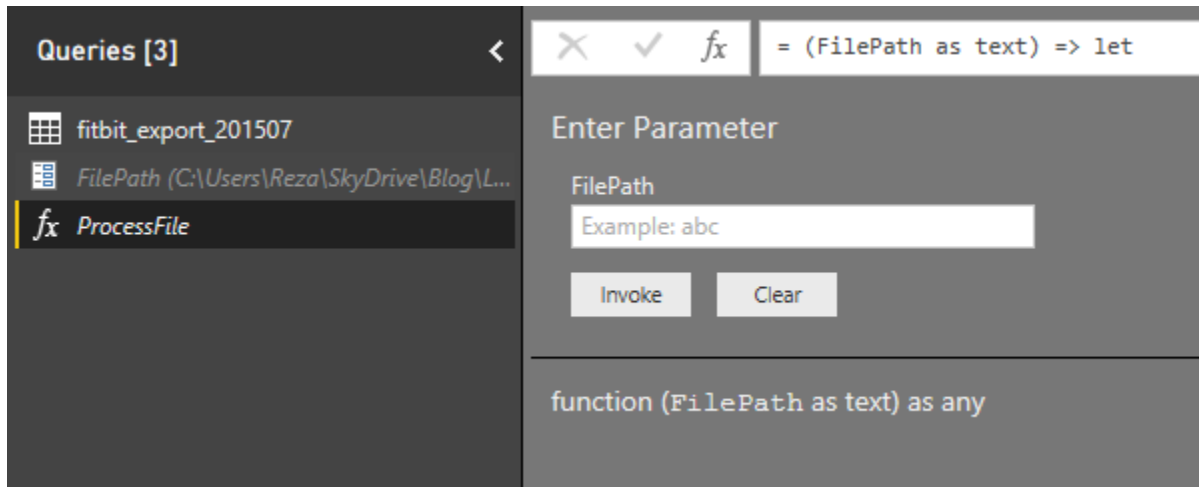
Parameter name (FilePath)

FilePath

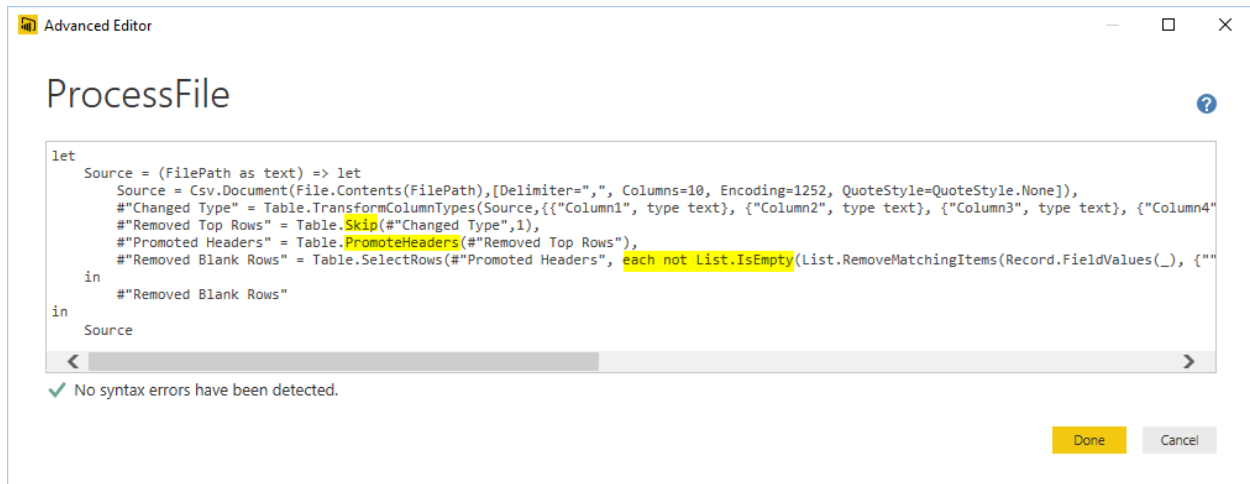
OK

Cancel

Because the query used parameter as a source so it can be simply used to create a function. Now You will see a function listed in Queries pane as well.

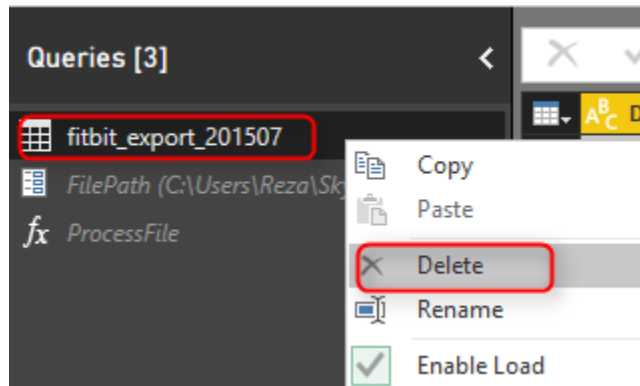


Note that the function uses M (Power Query Formula language) with Lambda syntax. But we have created it all just with graphical user interface option. If you are interested to learn more about code side of it, [read this post](#). This function now has all the steps generated in the first query. To see the steps you can simply click on Advanced Editor while you have ProcessFile function selected in Queries pane.



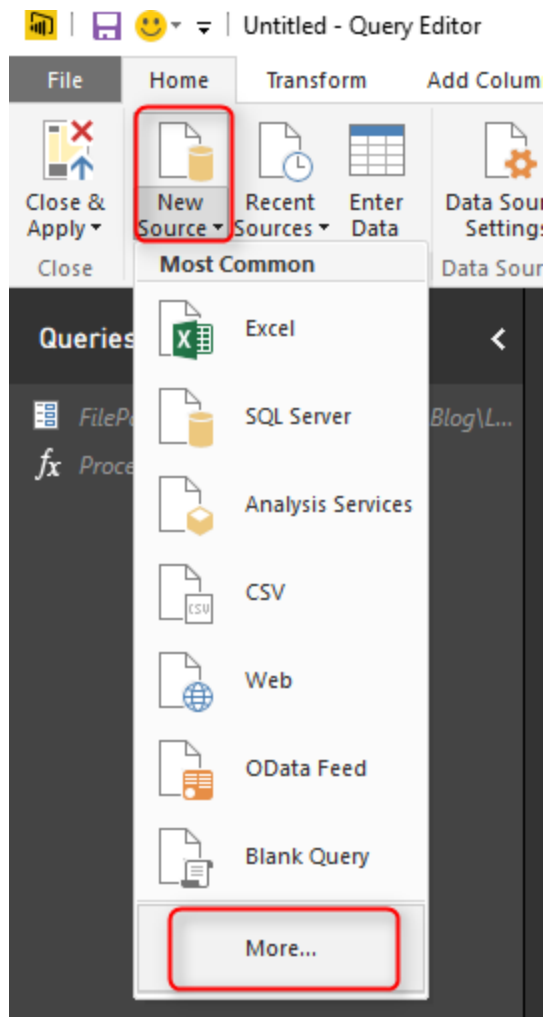
Don't be afraid of this code! you don't need to change anything here. It was just for letting you know that we have same steps copied into the function.

Now you can remove the first query, we don't need that anymore, steps are already in the custom function.



Get Data From Folder

Now that we have the function ready, let's get list of files and apply it on all files. Get Data from a New Source, and More.



Choose Folder this time;

Get Data

All

File


Database


Azure


Online Services


Other


File


 Excel


 CSV

 XML

 Text

 JSON

 Folder



 SharePoint Folder

And select the folder which contains all CSV files;

Folder

Choose a folder.

Folder Path

 C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD 

OK

Cancel

After clicking OK here and also on the preview, you will see list of all files in this folder with their details such as file name, extension, date created.... and folder path. You also see a column as Content, which is the Binary format of this files (the actual data in each file), I will talk about this column more in second approach.

Queries [3] ✕ ✓ fx = Folder.Files

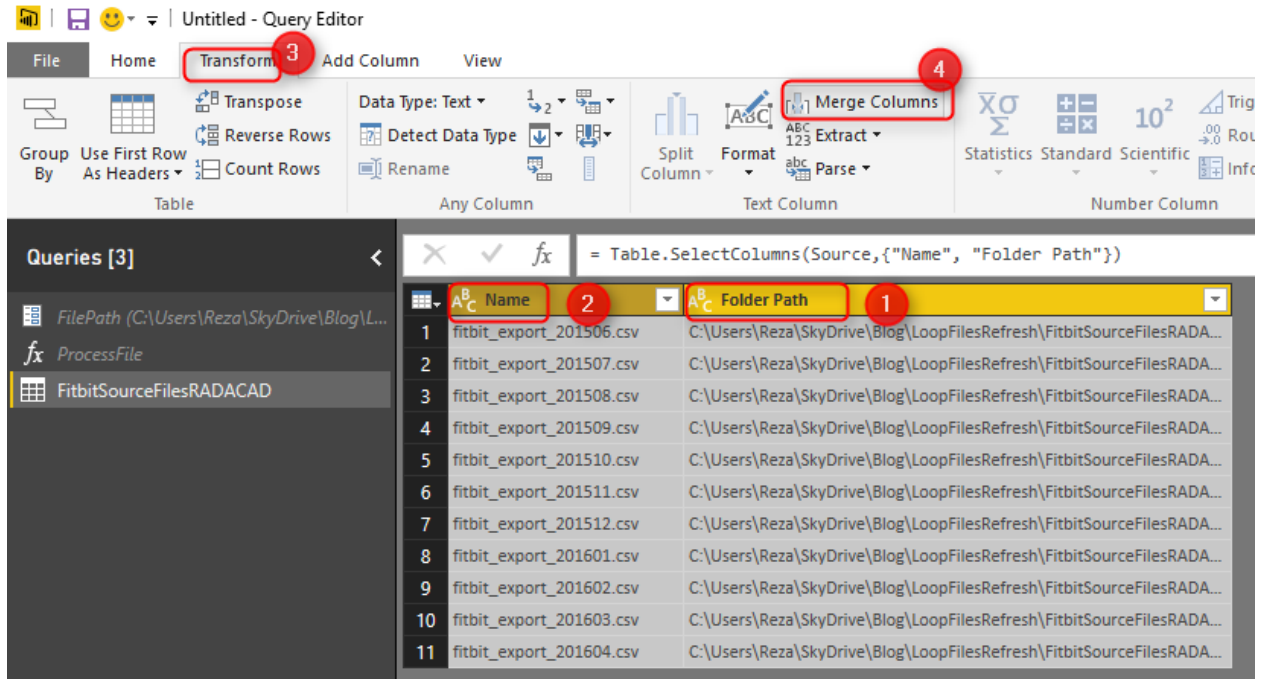
	Content	Name	Extension	Date accessed	Date modified
1	Binary	fitbit_export_201506.csv	.csv	10/7/2016 11:16:29 AM	9/24/2015 12:29:43 AM
2	Binary	fitbit_export_201507.csv	.csv	10/7/2016 11:16:29 AM	11/11/2015 10:49:37 PM
3	Binary	fitbit_export_201508.csv	.csv	10/7/2016 11:16:29 AM	11/11/2015 10:48:51 PM
4	Binary	fitbit_export_201509.csv	.csv	10/7/2016 11:16:29 AM	11/11/2015 10:48:11 PM
5	Binary	fitbit_export_201510.csv	.csv	10/7/2016 11:16:29 AM	11/11/2015 10:47:09 PM
6	Binary	fitbit_export_201511.csv	.csv	10/7/2016 11:16:29 AM	12/1/2015 3:07:38 PM
7	Binary	fitbit_export_201512.csv	.csv	10/7/2016 11:16:29 AM	4/28/2016 9:40:59 PM
8	Binary	fitbit_export_201601.csv	.csv	10/7/2016 11:16:29 AM	4/28/2016 9:42:16 PM
9	Binary	fitbit_export_201602.csv	.csv	10/7/2016 11:16:29 AM	4/28/2016 9:42:55 PM
10	Binary	fitbit_export_201603.csv	.csv	10/7/2016 11:16:29 AM	4/28/2016 9:43:46 PM
11	Binary	fitbit_export_201604.csv	.csv	10/7/2016 11:16:29 AM	4/28/2016 9:35:49 PM

For this approach we just need a full file path which we can generate from Folder Path and Name , so only select these two columns and remove everything else.

✕ ✓ fx = Table.SelectColumns(Source,{"Name", "Folder Path"})

	Name	Folder Path
1	fitbit_export_201506.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
2	fitbit_export_201507.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
3	fitbit_export_201508.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
4	fitbit_export_201509.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
5	fitbit_export_201510.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
6	fitbit_export_201511.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
7	fitbit_export_201512.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
8	fitbit_export_201601.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
9	fitbit_export_201602.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
10	fitbit_export_201603.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
11	fitbit_export_201604.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...

Now let's concatenate these two and create a full file path, Select Folder Path first, and then hold ctrl key and select Name column then. Go to Transform Tab and Choose Merge Columns



Queries [3]

- FilePath (C:\Users\Reza\SkyDrive\Blog\L...
- ProcessFile
- FitbitSourceFilesRADACAD

	Name	Folder Path
1	fitbit_export_201506.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
2	fitbit_export_201507.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
3	fitbit_export_201508.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
4	fitbit_export_201509.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
5	fitbit_export_201510.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
6	fitbit_export_201511.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
7	fitbit_export_201512.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
8	fitbit_export_201601.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
9	fitbit_export_201602.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
10	fitbit_export_201603.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...
11	fitbit_export_201604.csv	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADA...

Name the new concatenated column as FullFilePath, You don't need to put a separator for it, because Folder path already has an ending backslash in the string.

Merge Columns

Choose how to merge the selected columns.

Separator

--None--

New column name (optional)

FullFilePath

OK

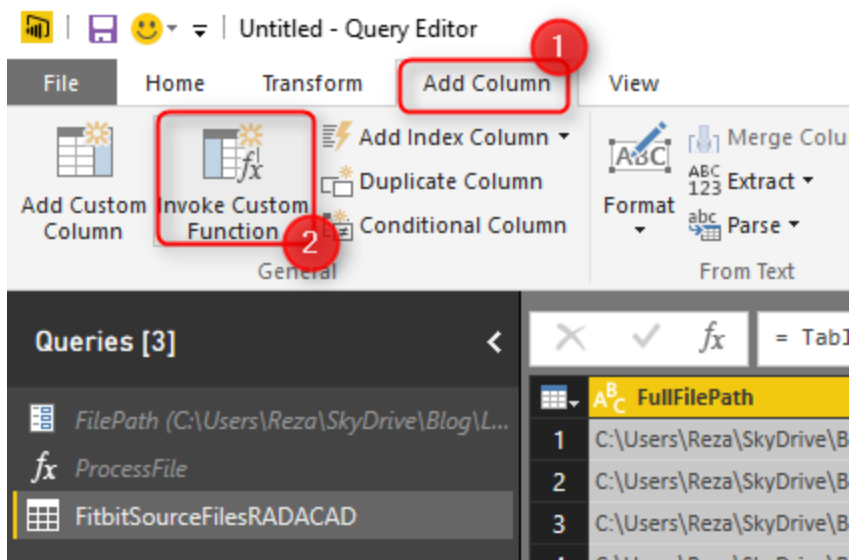
Cancel

This will replace existing two columns with the new FullFilePath concatenated column;

	<code>= Table.CombineColumns(#"Removed Other Columns",{"Folder</code>
FullFilePath	
1	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201506.csv
2	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201507.csv
3	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201508.csv
4	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201509.csv
5	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201510.csv
6	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201511.csv
7	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201512.csv
8	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201601.csv
9	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201602.csv
10	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201603.csv
11	C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201604.csv

Call Custom Function For Each File

Now we have everything ready for the custom function to be called for every record of this table. Go to Add Column tab, and click on Invoke Custom Function;



In the Invoke Custom function dialog box, select the function as ProcessFile, and the input parameter this time comes from the FullFilePath column that we created few minutes ago. Name this column as Content.



Invoke Custom Function

Invoke a custom function defined in this file for each row.

New column name

Content

Function query

ProcessFile

FilePath



FullFilePath

OK

Cancel

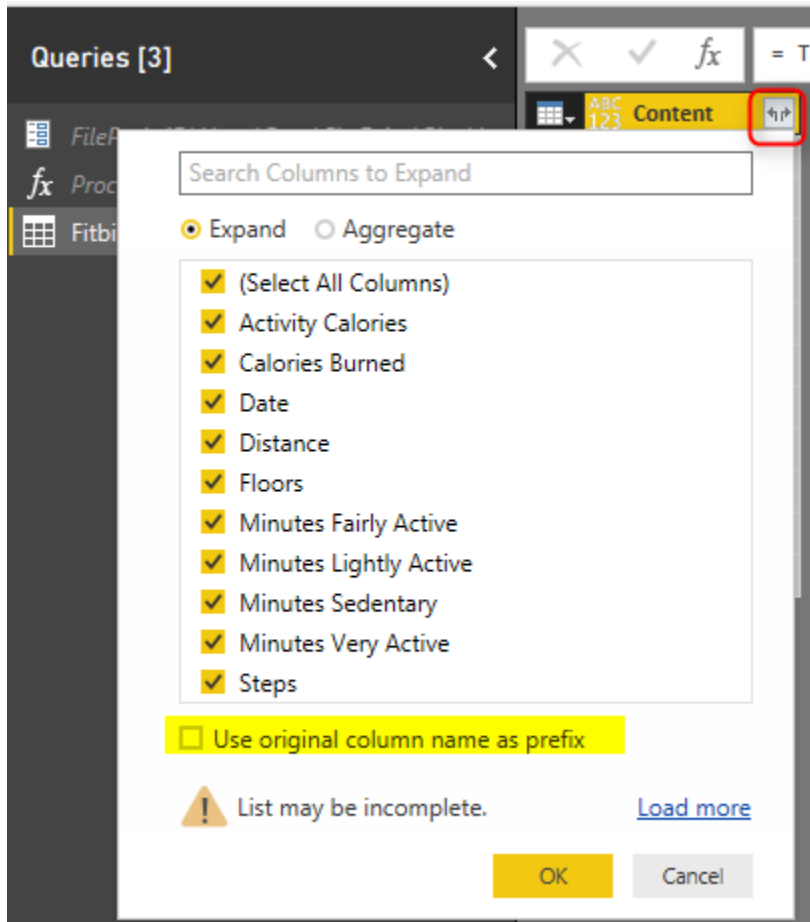
After clicking on OK, in an instant, you will see the new column appears which has a Table in each cell. This is the data table processed for each file. If you click on a blank area of one of table's cells you will see the data preview of that in a details view pane underneath preview pane.

fx = Table.AddColumn("#Merged Columns", "Content", each ProcessFile([FullFilePath]))

FullFilePath	Content
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201506.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201507.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201508.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201509.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201510.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201511.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201512.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201601.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201602.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201603.csv	Table
C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD\fitbit_export_201604.csv	Table

Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories
01-10-2015	3,422	9,793	7.46	13	1,121	233	43	43	1,851
02-10-2015	3,729	11,396	8.65	20	1,060	236	69	75	2,243
03-10-2015	4,508	13,502	10.28	41	900	308	127	105	3,204
04-10-2015	4,536	17,167	13.86	10	972	261	67	140	3,178
05-10-2015	3,830	12,806	10.64	20	1,086	267	16	71	2,310
06-10-2015	3,332	9,218	7	26	1,156	171	46	67	1,728
07-10-2015	4,022	13,746	11.28	24	1,058	228	59	95	2,534
08-10-2015	4,659	16,311	13.25	44	927	296	82	135	3,358
09-10-2015	4,332	17,027	12.92	26	1,024	207	80	129	2,900
10-10-2015	4,840	17,264	13.96	20	868	301	159	112	3,632
11-10-2015	3,753	13,958	11.57	11	1,083	237	48	72	2,280
12-10-2015	3,815	15,177	12.32	35	1,071	218	69	82	2,341
13-10-2015	3,344	8,452	6.42	23	1,130	224	23	63	1,783
14-10-2015	3,836	13,774	10.51	24	1,058	226	56	100	2,323
15-10-2015	3,808	16,526	13.34	24	1,091	200	53	96	2,292
16-10-2015	4,292	19,965	15.15	54	1,043	145	73	179	2,815
17-10-2015	3,907	13,473	11.14	4	1,020	307	41	72	2,473
18-10-2015	3,627	11,077	9.17	6	1,096	235	44	65	2,114
19-10-2015	3,518	13,746	11.35	29	1,145	164	35	96	1,949
20-10-2015	2,657	6,214	4.72	29	1,276	122	18	24	936

We don't need FullFilePath column anymore, so Remove that column. Now in the table we have just one column, which is a structured column. Means it has a structured value in each cell (in this case a table in each cell). We can now expand this structured column to all underlying columns with clicking on Expand button. Also remember to uncheck the box for Use Original Column Name as prefix.



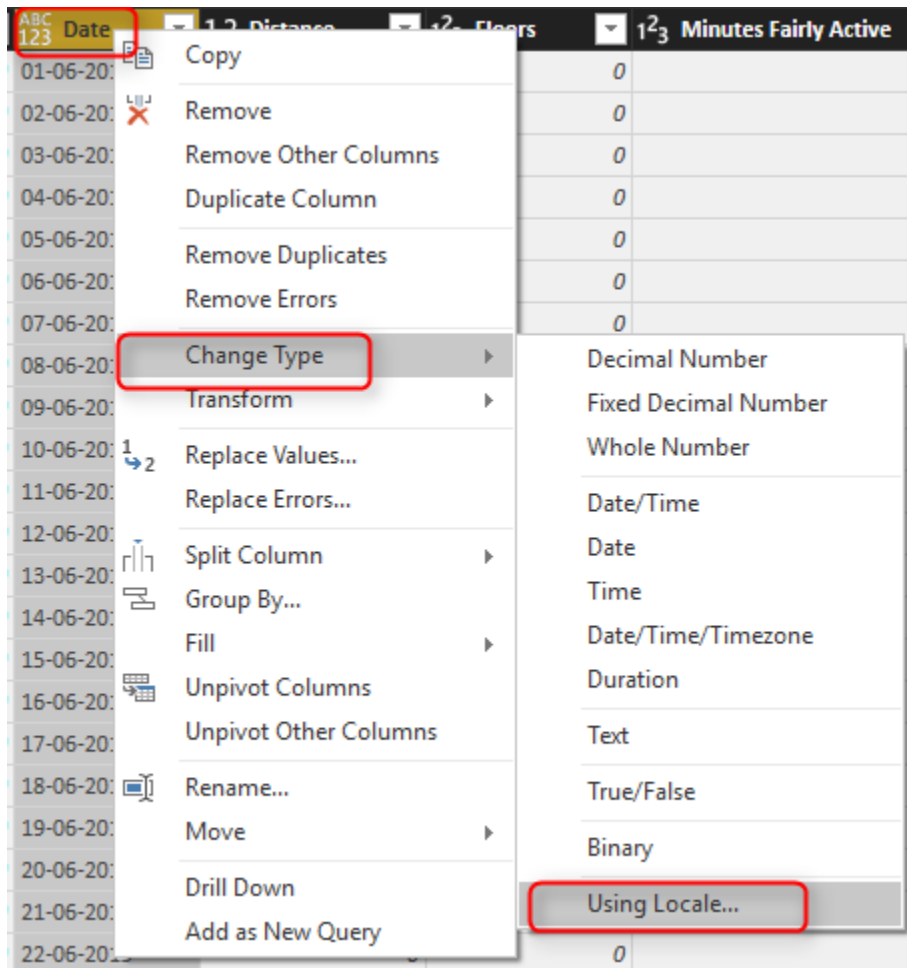
This expand button (or you might call it Magic button now, as it reveals the magic!) combines all data tables into a single big data table, including data rows from all CSV files.

= Table.ExpandTableColumn(*Removed Columns", "Content", {"Activity Calories", "Calories Burned", "Date", "Distance", "Floors", "Minutes Fairly Active", "Minutes Lightly Active", "Minutes Sedentary", "Minutes Very Active", "Steps"											
	Activity Calories	Calories Burned	Date	Distance	Floors	Minutes Fairly Active	Minutes Lightly Active	Minutes Sedentary	Minutes Very Active	Steps	
1	0	1,850	01-06-2015	0	0	0	0	1,440	0	0	
2	0	1,850	02-06-2015	0	0	0	0	1,440	0	0	
3	0	1,850	03-06-2015	0	0	0	0	1,440	0	0	
4	0	1,850	04-06-2015	0	0	0	0	1,440	0	0	
5	0	1,850	05-06-2015	0	0	0	0	1,440	0	0	
6	0	1,850	06-06-2015	0	0	0	0	1,440	0	0	
7	0	1,850	07-06-2015	0	0	0	0	1,440	0	0	
8	0	1,850	08-06-2015	0	0	0	0	1,440	0	0	
9	0	1,850	09-06-2015	0	0	0	0	1,440	0	0	
10	0	1,850	10-06-2015	0	0	0	0	1,440	0	0	
11	0	1,850	11-06-2015	0	0	0	0	1,440	0	0	
12	0	1,850	12-06-2015	0	0	0	0	1,440	0	0	
13	0	1,850	13-06-2015	0	0	0	0	1,440	0	0	
14	0	1,850	14-06-2015	0	0	0	0	1,440	0	0	
15	0	1,850	15-06-2015	0	0	0	0	1,440	0	0	
16	0	1,850	16-06-2015	0	0	0	0	1,440	0	0	
17	0	1,850	17-06-2015	0	0	0	0	1,440	0	0	
18	0	1,850	18-06-2015	0	0	0	0	1,440	0	0	
19	0	1,850	19-06-2015	0	0	0	0	1,440	0	0	
20	0	1,850	20-06-2015	0	0	0	0	1,440	0	0	
21	0	1,850	21-06-2015	0	0	0	0	1,440	0	0	
22	0	1,850	22-06-2015	0	0	0	0	1,440	0	0	
23	0	1,850	23-06-2015	0	0	0	0	1,440	0	0	
24	0	1,850	24-06-2015	0	0	0	0	1,440	0	0	
25	0	1,850	25-06-2015	0	0	0	0	1,440	0	0	
26	0	1,850	26-06-2015	0	0	0	0	1,440	0	0	
27	0	1,850	27-06-2015	0	0	0	0	1,440	0	0	
28	0	1,850	28-06-2015	0	0	0	0	1,440	0	0	
29	0	1,850	29-06-2015	0	0	0	0	1,440	0	0	

All we need to do now is some data type changing;

Data Type Conversion

Change data type of all columns except Date and Distance to Whole Number. Change data type of Distance column to Decimal. and Finally change data type Date field to Date. For the data type of Date field to be changed you need to choose locale. because the exported files has date values in New Zealand Date format, and if you change it with normal data type conversion, Power Query try to convert their data type based on your local system date format, which might not be New Zealand's format. So right click on Date Column. under Data Type choose Using Locale.



In the new dialog box, choose the data type to convert to as Date, and set locale as English New Zealand



Change Type with Locale

Change the data type and select the locale of origin.

Data Type

Date

Locale

English (New Zealand)

i Sample input values:

29/03/2016

Tuesday, 29 March 2016

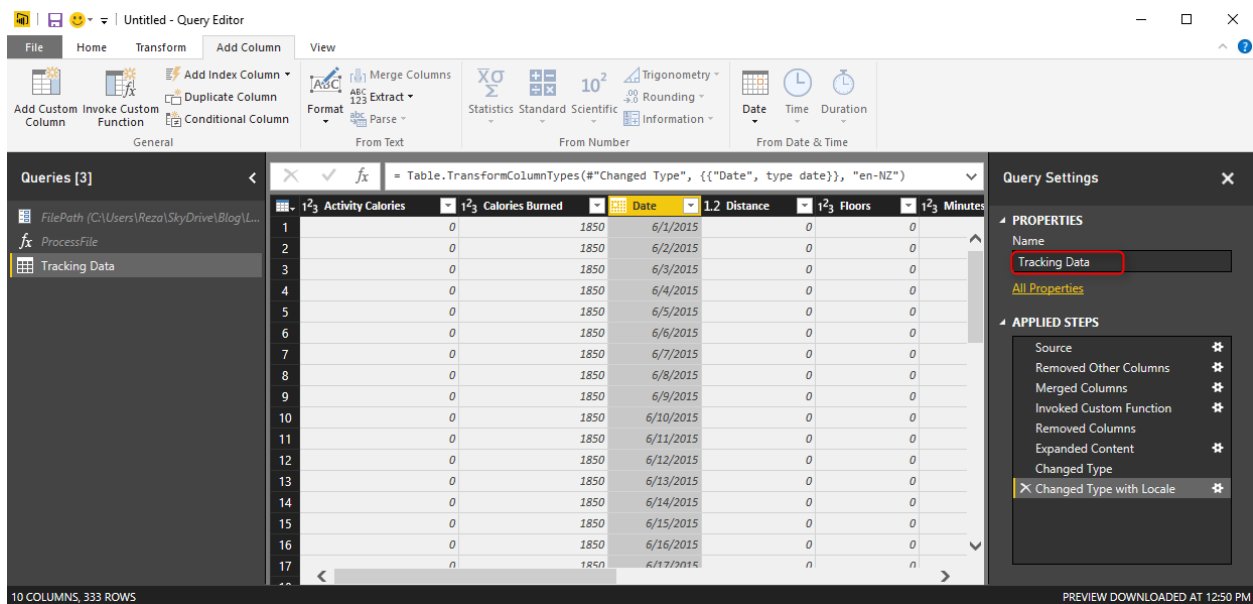
29/March

March/2016

OK

Cancel

After clicking on OK you will see data types converted correctly without any error. As the final step now change the query name to Tracking Data;



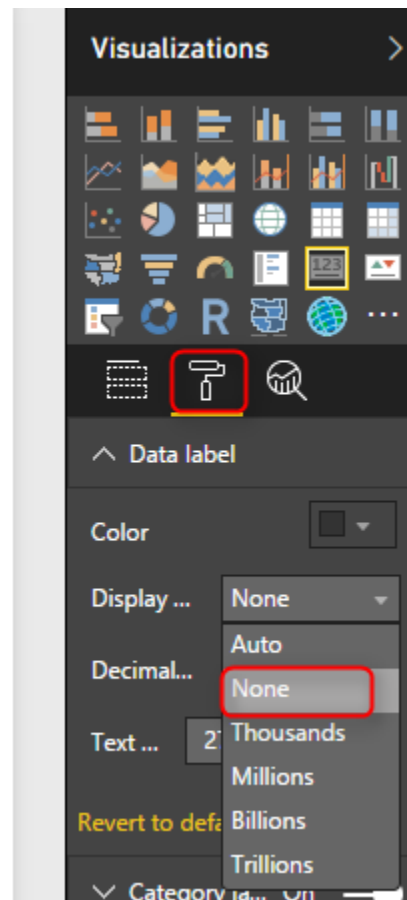
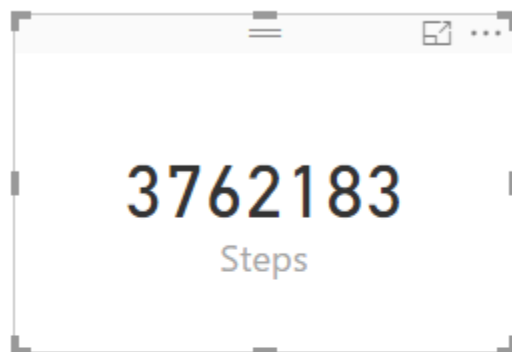
The screenshot shows the Power BI Query Editor interface. The 'Query Settings' pane on the right shows the query name 'Tracking Data' highlighted in red. The main data view shows a table with columns: Activity Calories, Calories Burned, Date, Distance, Floors, and Minutes. The 'Date' column is highlighted with a date icon in the header. The 'Query Settings' pane on the right shows the 'APPLIED STEPS' list, which includes 'Changed Type with Locale'.

	Activity Calories	Calories Burned	Date	Distance	Floors	Minutes
1	0	1850	6/1/2015	0	0	0
2	0	1850	6/2/2015	0	0	0
3	0	1850	6/3/2015	0	0	0
4	0	1850	6/4/2015	0	0	0
5	0	1850	6/5/2015	0	0	0
6	0	1850	6/6/2015	0	0	0
7	0	1850	6/7/2015	0	0	0
8	0	1850	6/8/2015	0	0	0
9	0	1850	6/9/2015	0	0	0
10	0	1850	6/10/2015	0	0	0
11	0	1850	6/11/2015	0	0	0
12	0	1850	6/12/2015	0	0	0
13	0	1850	6/13/2015	0	0	0
14	0	1850	6/14/2015	0	0	0
15	0	1850	6/15/2015	0	0	0
16	0	1850	6/16/2015	0	0	0
17	0	1850	6/17/2015	0	0	0

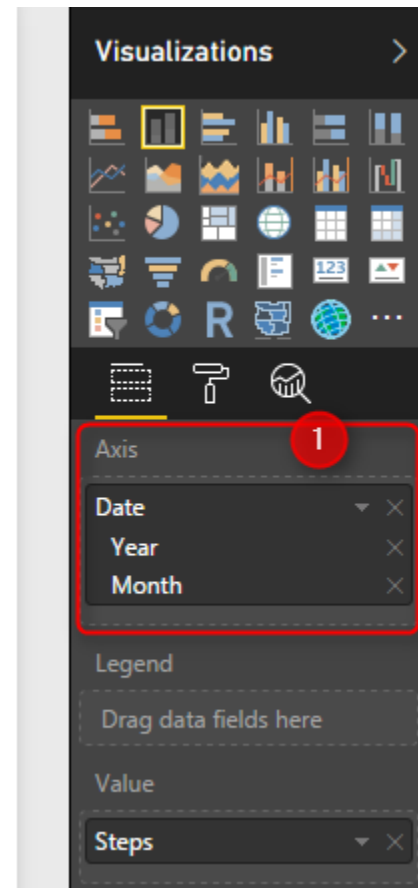
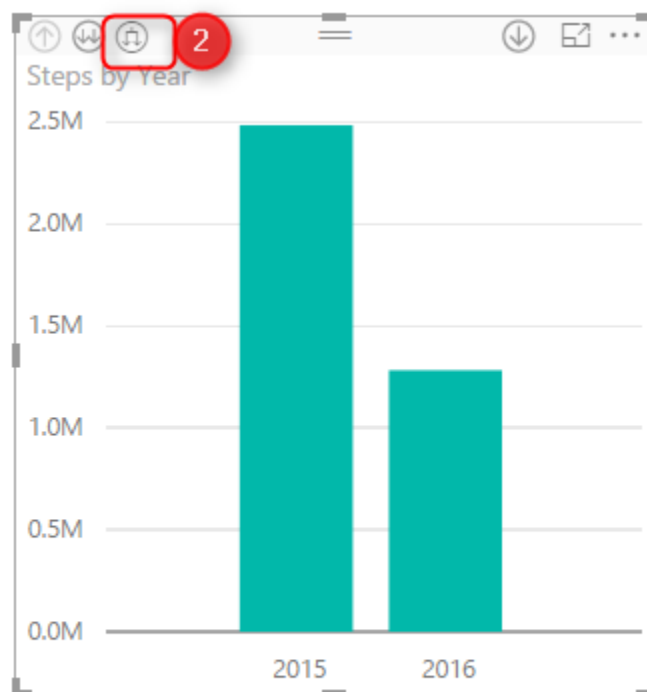
Notice that Date column showed in above screenshot is converted to a Date format (you can see the date format icon in the column header), but it shows in this window to me in my computer locale format which is English (United States). The way that it shows doesn't matter. All matters is that this is of Date format.

Build Sample Report

Now let's build a very simple Report with the data. Close and Apply Power Query window, and in the Report pane, add Steps as Total value in a Card Visualization. Also in the formatting section set the Format of data label to display as None (means without any millions, or K, or anything like that).

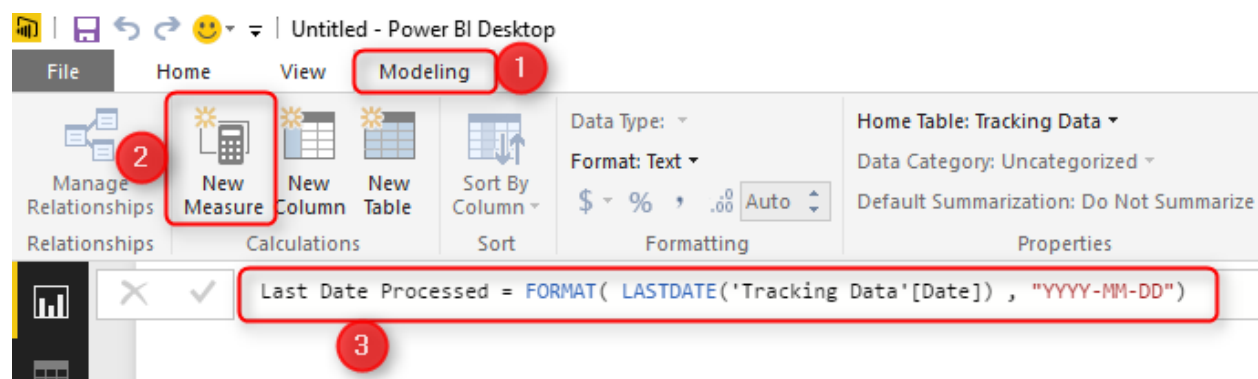


The other visualization to add is a Column Chart, which Steps as Value, Date as Axis. It will automatically bring Date Hierarchy, So remove Quarter and Day from the hierarchy.



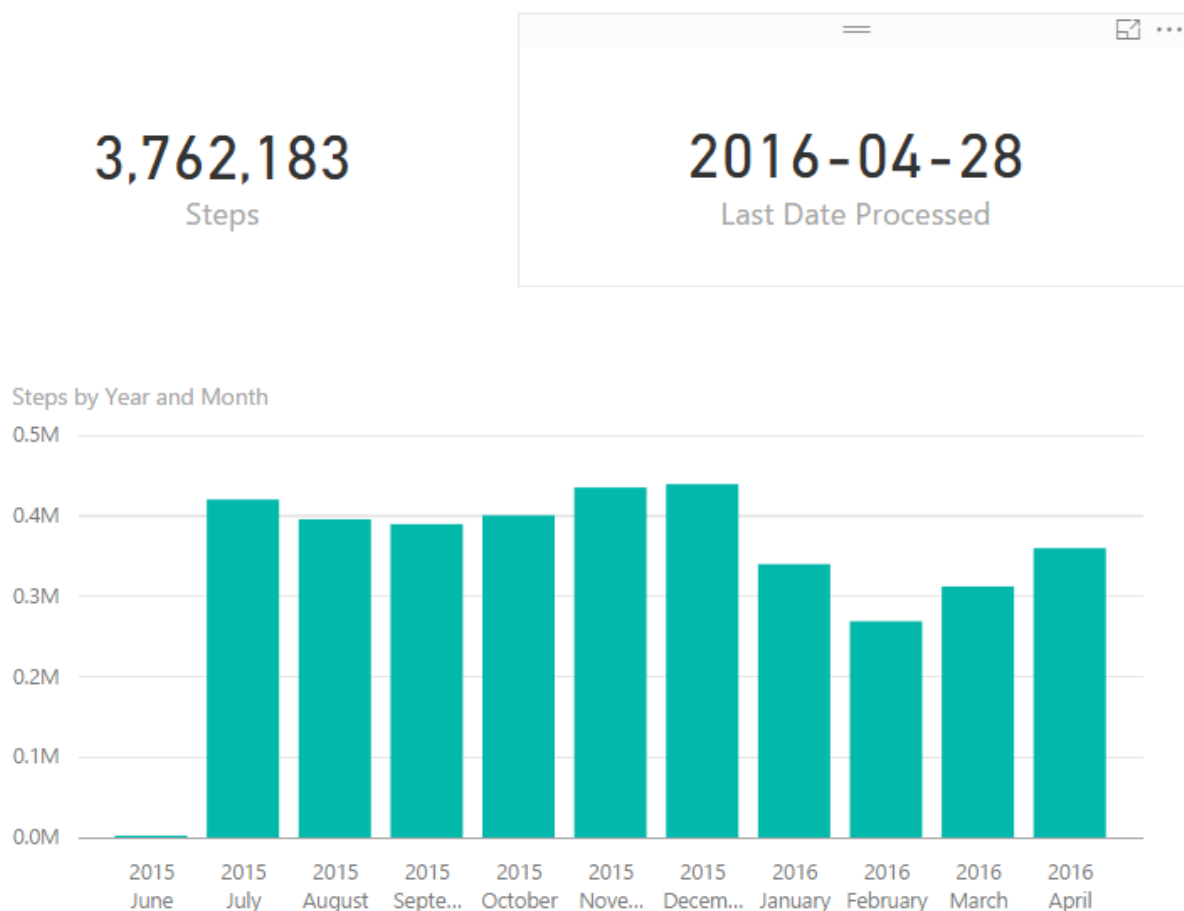
After removing Quarter and Day, click on Expand all down one level in the hierarchy in the top left corner of the column chart (screenshot above numbered 2);

Also create a measure to show the last date. Go to Modeling tab, New Measure, and use LastDate function to get the latest date of the date field in the Tracking Data table;



Last Date Processed = FORMAT(LASTDATE('Tracking Data'[Date]) , "YYYY-MM-DD")

Show this new measure as a Card visual in the report.



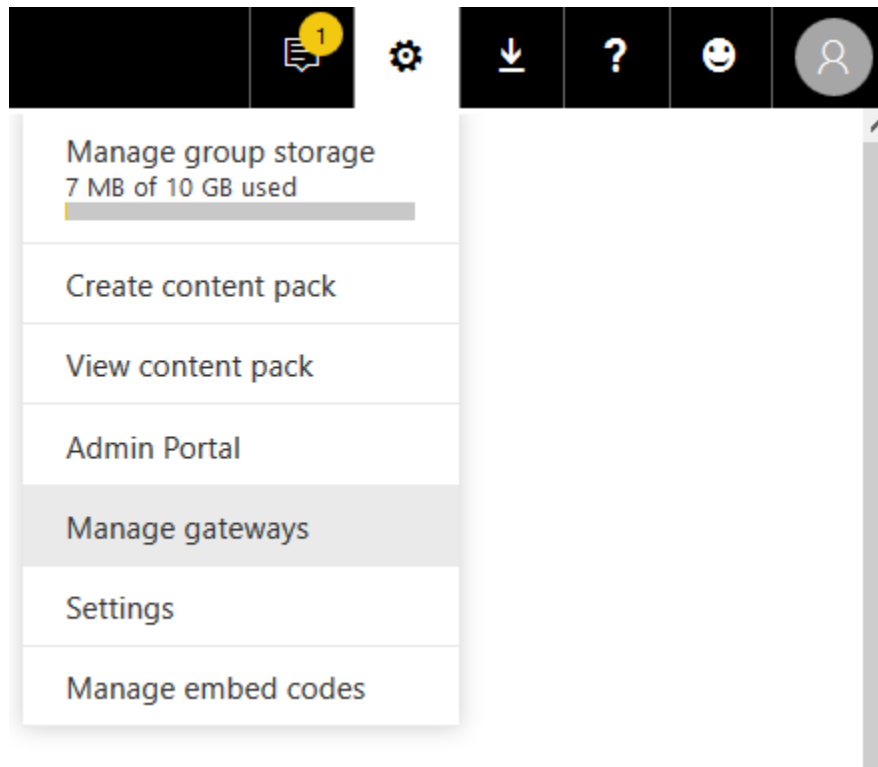
We can make this report much richer with adding more insights and visualizations, but let's keep that simple for now. Save the report and Publish. You can save it with a name like Fitbit Custom Function Method.

Set up Gateway

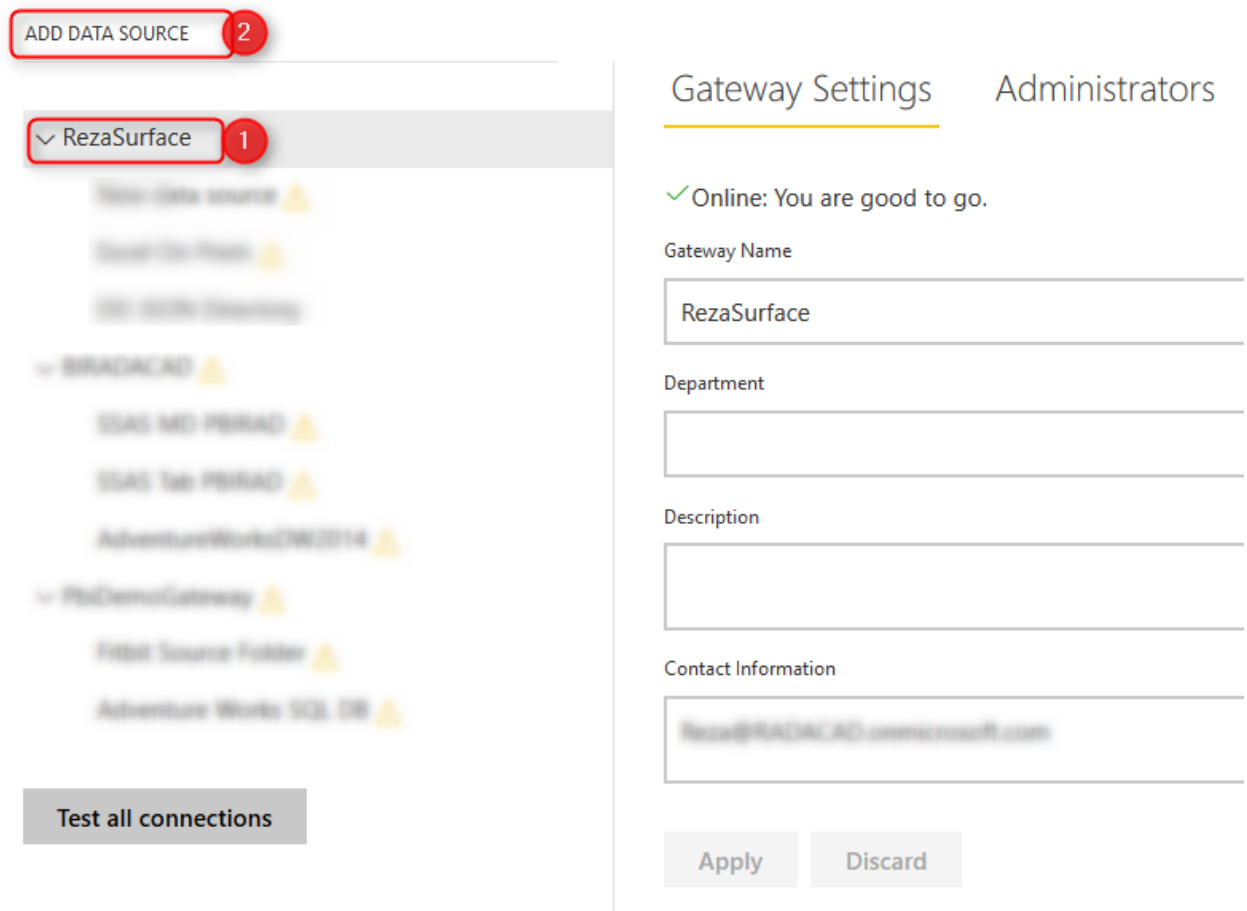
Because we want the report to be refreshed from on-premises folder we need to set up a gateway. Both Personal and On-Premises gateways can do this. Explanation of how to install and configure gateway is out of scope of this post, If I want to explain that I will make this post VERY LONG! If you want to learn more about gateways read this post about [Personal gateway](#), or this one about [On-Premises Gateway](#). I explain everything from after successful setup of gateway here, where we need to add a new data source.

New Data Source Configuration

We have to create a data source under the gateway for our local folder containing CSV files. Go to Manage Gateways in Power BI Service



Now select your gateway, and add a new data source under it.



ADD DATA SOURCE 2

▼ RezaSurface 1

- New data source
- Excel On-Prem
- SQL Azure Gateway
- ▼ BRADACAD
 - SSAS MD PBRAD
 - SSAS Tab PBRAD
 - AdventureWorks/DW2014
- ▼ PbxDemoGateway
 - Fabrik Source Folder
 - Adventure Works SQL DB

Test all connections

Gateway Settings Administrators

✓ Online: You are good to go.

Gateway Name

RezaSurface

Department

Description

Contact Information

Reza@RADACAD.com@microsoft.com

Apply Discard

Set up the new data source for Folder and set the local folder path there;

Data Source Settings

Data Source Name

Fitbit Source Directory

Data Source Type

Folder

Full path

C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD

The credentials are encrypted using the key stored on-premises on the gateway server. [Learn more](#)

Windows username

reza-vaio\reza

Windows password

.....

>Advanced settings

Add

Discard

Note that the local path should be accessible from the machine that has gateway installed on it. otherwise it should be a shared path. The windows username used here also should be for an account that has read access on the folder we are trying to access.

After successful setup you will see a message clarifying that;

Data Source Settings

[Users](#)

✓ Connection Successful

ⓘ Next Step: Go to the [Users tab](#) above and add users to this Data Source

Data Source Name

Fitbit Source Directory

Data Source Type

Folder

Full path

C:\Users\Reza\SkyDrive\Blog\LoopFilesRefresh\FitbitSourceFilesRADACAD

The credentials are encrypted using the key stored on-premises on the gateway.

Windows username

●●●●●●●●●●●●●●●●

Windows password

●●●●●●●●●●●●●●●●

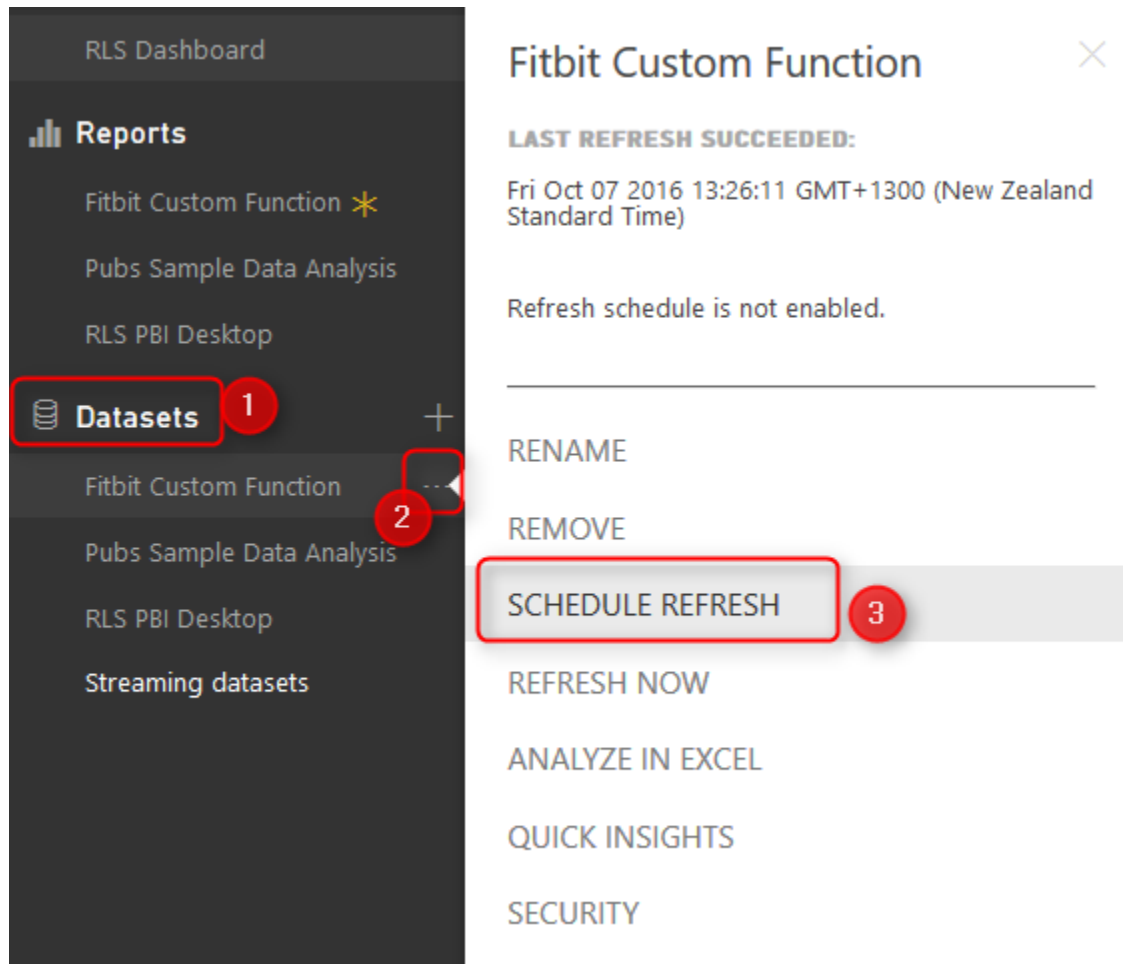
>Advanced settings

Apply

Discard

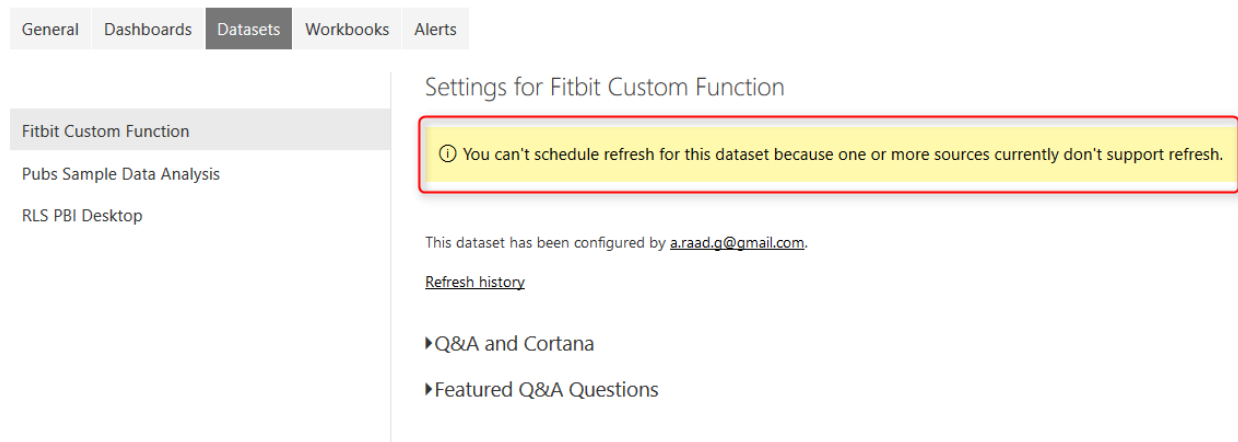
Scheduled Refresh

We've published the report earlier. Now it's the time to Schedule it for refresh, and configure it to get its data from gateway. Under Datasets find the Fitbit Custom Function dataset, and click on ellipsis button and choose Schedule Refresh.



In the Schedule Refresh Setting section, You will see an error message that says you can not configure gateway for this data set. That's because of custom function used here. (Note that this happens at the time of writing this blog post, It might be resolved by the time that you are reading this post)

The error message is: **You can't schedule refresh for this dataset because one or more sources currently don't support refresh.**



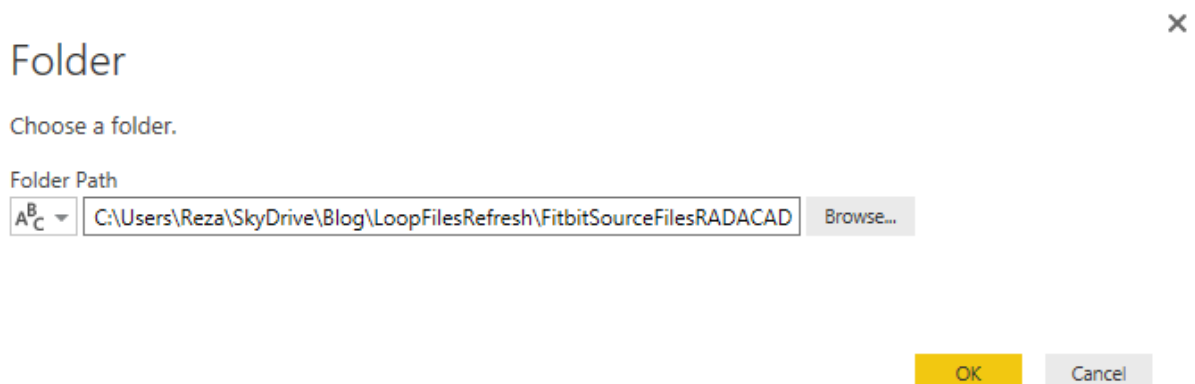
Now you see the problem, It was a long way to show you the issue! but you need to know it before using many custom functions in your code. Let's go through the second approach to resolve it.

Second Approach: Combine Files

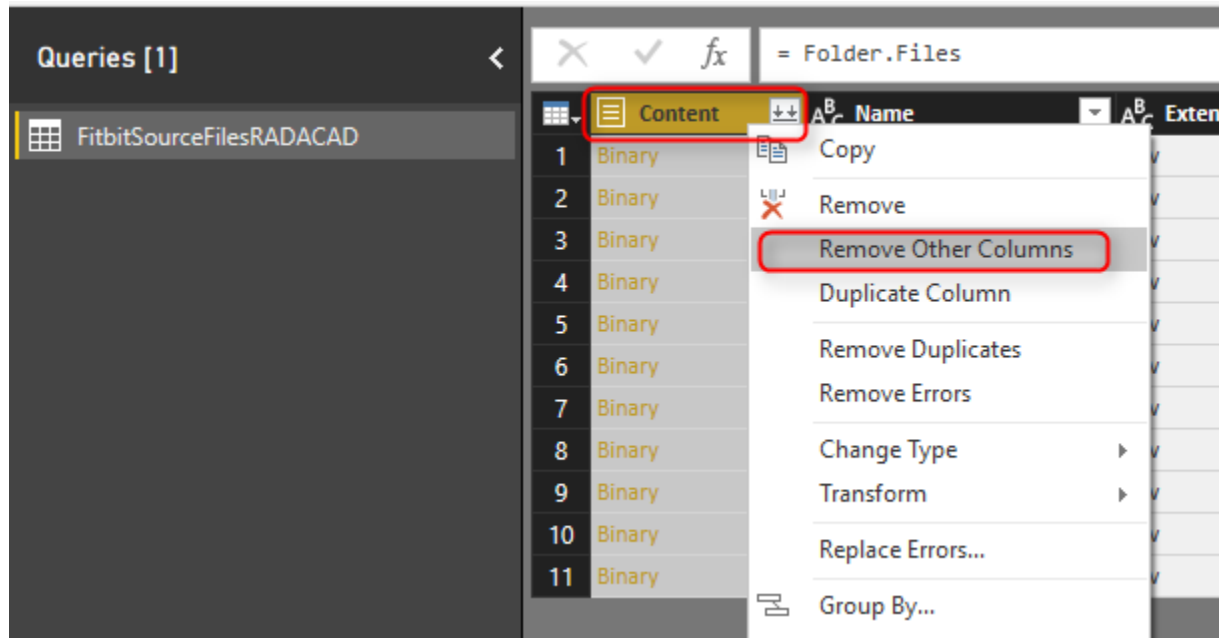
This approach has some limitations and difficulties involved, However at the time of writing this post, this is the only way of getting the scheduled refresh working.

Get Data From Folder

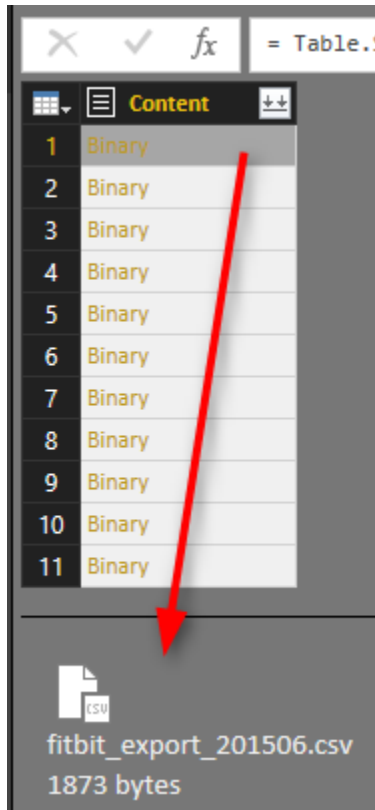
With this method we start with getting data from Folder. Open a new Power BI Desktop solution, and Get Data From Folder, exactly as you've seen in the Get Data From Folder section earlier in this post.



This time however we don't use custom functions, we want to use binary data of the file content. Content column includes the Binary Data. Content column is all we need, so remove all other columns.

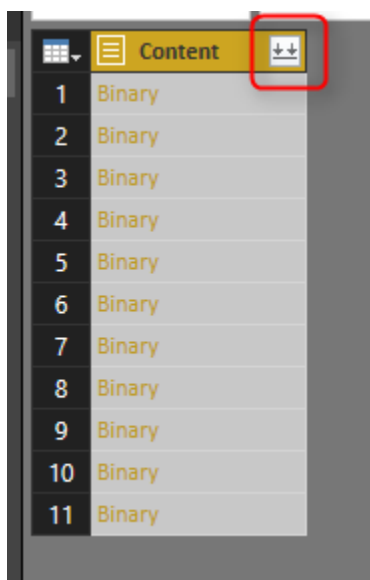


If you click on a blank area in any Binary cell in the content column, you will see the CSV file in preview pane under it.

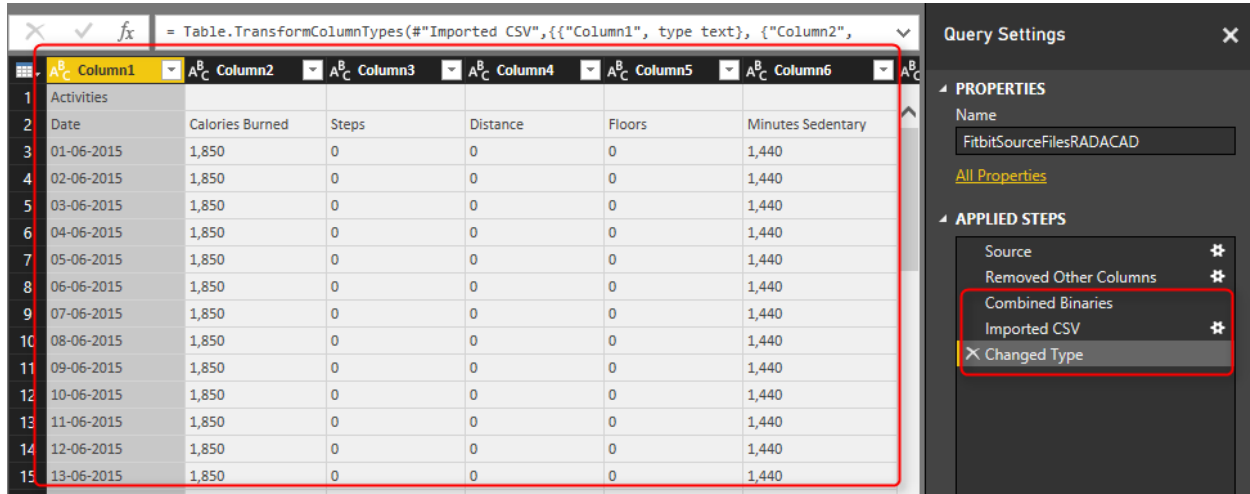


Combine Binaries

This Content column is a structured column, and any structured column can be expanded. Expanding a structured column with Binary values Will combine them together. Click on Combine Binaries button on the top right hand side of the column.



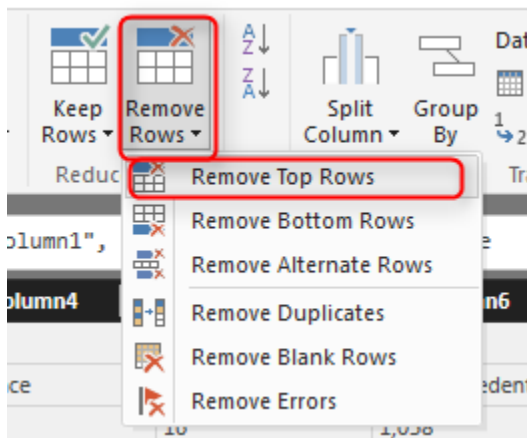
Combine Binaries will combine content of all files together, and will generate a big list of all data rows. You can also see automatic steps applied after combining binaries in the steps pane.



The screenshot shows the Power BI Desktop interface. On the left, a data table is displayed with columns: Activities, Date, Calories Burned, Steps, Distance, Floors, and Minutes Sedentary. The first row is labeled 'Activities' and the subsequent rows contain dates and numerical values. On the right, the 'Query Settings' pane is open, showing the 'APPLIED STEPS' section. The steps listed are: Source, Removed Other Columns, Combined Binaries, Imported CSV, and Changed Type. The 'Changed Type' step is highlighted with a red box.

Now that we have all data rows. Let's do some cleanup;

As you can see in the preview the first record isn't useful. it has just a title "Activities", so let's remove it with going to Remove Rows, Remove Top Rows. and enter 1 in the dialog box;



The screenshot shows the 'Remove Rows' dropdown menu in Power BI Desktop. The menu options are: Remove Top Rows, Remove Bottom Rows, Remove Alternate Rows, Remove Duplicates, Remove Blank Rows, and Remove Errors. The 'Remove Top Rows' option is highlighted with a red box.

Now the first row only contains column headers. So let's promote it to column headers with clicking on Use First Row As Headers. There are also some blank rows, so let's remove them with Remove Rows -> Remove Blank Rows.

There are still some additional rows with "Activities" or "Date" labels in their first column. These are heading rows in each CSV file. After combining content of all files these appears in the middle of the data set.

Table.SelectRows(*Promoted Headers", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), ("", null))))

	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories
1	01-06-2015	1,850	0	0	0	1,440	0	0	0	0
2	02-06-2015	1,850	0	0	0	1,440	0	0	0	0
3	03-06-2015	1,850	0	0	0	1,440	0	0	0	0
4	04-06-2015	1,850	0	0	0	1,440	0	0	0	0
5	05-06-2015	1,850	0	0	0	1,440	0	0	0	0
6	06-06-2015	1,850	0	0	0	1,440	0	0	0	0
7	07-06-2015	1,850	0	0	0	1,440	0	0	0	0
8	08-06-2015	1,850	0	0	0	1,440	0	0	0	0
9	09-06-2015	1,850	0	0	0	1,440	0	0	0	0
10	10-06-2015	1,850	0	0	0	1,440	0	0	0	0
11	11-06-2015	1,850	0	0	0	1,440	0	0	0	0
12	12-06-2015	1,850	0	0	0	1,440	0	0	0	0
13	13-06-2015	1,850	0	0	0	1,440	0	0	0	0
14	14-06-2015	1,850	0	0	0	1,440	0	0	0	0
15	15-06-2015	1,850	0	0	0	1,440	0	0	0	0
16	16-06-2015	1,850	0	0	0	1,440	0	0	0	0
17	17-06-2015	1,850	0	0	0	1,440	0	0	0	0
18	18-06-2015	1,850	0	0	0	1,440	0	0	0	0
19	19-06-2015	1,850	0	0	0	1,440	0	0	0	0
20	20-06-2015	1,850	0	0	0	1,440	0	0	0	0
21	21-06-2015	1,850	0	0	0	1,440	0	0	0	0
22	22-06-2015	1,850	0	0	0	1,440	0	0	0	0
23	23-06-2015	1,850	0	0	0	1,440	0	0	0	0
24	24-06-2015	1,850	0	0	0	1,440	0	0	0	0
25	25-06-2015	1,850	0	0	0	1,440	0	0	0	0
26	26-06-2015	1,850	0	0	0	1,440	0	0	0	0
27	27-06-2015	1,850	0	0	0	1,440	0	0	0	0
28	28-06-2015	1,850	0	0	0	1,440	0	0	0	0
29	29-06-2015	1,850	0	0	0	1,440	0	0	0	0
30	30-06-2015	1,892	305	0.22	0	1,414	13	0	0	48
31	Activities	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories
32	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories
33	01-07-2015	3,953	13,361	10.64	16	1,058	196	71	87	2,393
34	02-07-2015	3,432	10,818	8.76	10	726	224	19	45	1,803
35	03-07-2015	4,102	15,537	12.38	16	701	237	55	88	2,550
36	04-07-2015	3,999	14,490	11.54	6	685	255	46	69	2,443
37	05-07-2015	4,028	14,652	11.68	3	1,003	283	93	61	2,598
38	06-07-2015	3,538	12,161	9.84	16	1,144	168	76	52	1,968

Query Settings

PROPERTIES

Name
FitbitSourceFilesRADACAD

Source
Removed Other Columns
Combined Binaries
Imported CSV
Changed Type
Removed Top Rows
Promoted Headers
Removed Blank Rows

We can remove them by applying a filter on Date column. Click on Filter icon for Date column and from Text Filters choose Does Not Equal.

Queries [1]

Table.SelectRows(*Promoted Headers", each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), ("", null))))

1

2

3

Sort Ascending

Sort Descending

Clear Sort

Clear Filter

Remove Empty

Text Filters

Search

(Select All)

01-01-2016

01-02-2016

01-03-2016

01-04-2016

01-06-2015

01-07-2015

01-08-2015

Equals...

Does Not Equal...

Begins With...

Does Not Begin With...

Ends With...

Does Not End With...

Contains...

Does Not Contain...

Choose Does Not Equal, and with an AND condition type in both Activities and Date there.

Filter Rows

☒ Basic ☐ Advanced

Show rows where: Date

does not equal

Activities

☒ And ☐ Or

does not equal

Date

OK

Cancel

This will filter all extra labeled rows, and you will only see actual data rows.

	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories
1	01-06-2015	1,850	0	0	0	1,440	0	0	0	0
2	02-06-2015	1,850	0	0	0	1,440	0	0	0	0
3	03-06-2015	1,850	0	0	0	1,440	0	0	0	0
4	04-06-2015	1,850	0	0	0	1,440	0	0	0	0
5	05-06-2015	1,850	0	0	0	1,440	0	0	0	0
6	06-06-2015	1,850	0	0	0	1,440	0	0	0	0
7	07-06-2015	1,850	0	0	0	1,440	0	0	0	0
8	08-06-2015	1,850	0	0	0	1,440	0	0	0	0
9	09-06-2015	1,850	0	0	0	1,440	0	0	0	0
10	10-06-2015	1,850	0	0	0	1,440	0	0	0	0
11	11-06-2015	1,850	0	0	0	1,440	0	0	0	0
12	12-06-2015	1,850	0	0	0	1,440	0	0	0	0
13	13-06-2015	1,850	0	0	0	1,440	0	0	0	0
14	14-06-2015	1,850	0	0	0	1,440	0	0	0	0
15	15-06-2015	1,850	0	0	0	1,440	0	0	0	0
16	16-06-2015	1,850	0	0	0	1,440	0	0	0	0
17	17-06-2015	1,850	0	0	0	1,440	0	0	0	0
18	18-06-2015	1,850	0	0	0	1,440	0	0	0	0
19	19-06-2015	1,850	0	0	0	1,440	0	0	0	0
20	20-06-2015	1,850	0	0	0	1,440	0	0	0	0
21	21-06-2015	1,850	0	0	0	1,440	0	0	0	0
22	22-06-2015	1,850	0	0	0	1,440	0	0	0	0
23	23-06-2015	1,850	0	0	0	1,440	0	0	0	0
24	24-06-2015	1,850	0	0	0	1,440	0	0	0	0
25	25-06-2015	1,850	0	0	0	1,440	0	0	0	0
26	26-06-2015	1,850	0	0	0	1,440	0	0	0	0
27	27-06-2015	1,850	0	0	0	1,440	0	0	0	0
28	28-06-2015	1,850	0	0	0	1,440	0	0	0	0
29	29-06-2015	1,850	0	0	0	1,440	0	0	0	0
30	30-06-2015	1,892	305	0.22	0	1,414	13	0	0	48
31	01-07-2015	3,953	13,361	10.64	16	1,058	196	71	87	2,393
32	02-07-2015	3,432	10,818	8.76	10	726	224	19	45	1,803
33	03-07-2015	4,102	15,537	12.38	16	701	237	55	88	2,550
34	04-07-2015	3,959	14,490	11.54	6	685	255	46	69	2,443
35	05-07-2015	4,028	14,652	11.68	3	1,003	283	93	61	2,598
36	06-07-2015	3,538	12,161	9.84	16	1,144	168	76	52	1,968
37	07-07-2015	3,550	11,103	8.99	17	785	250	24	41	1,950
38	08-07-2015	3,238	10,029	8.22	10	1,204	152	36	48	1,587

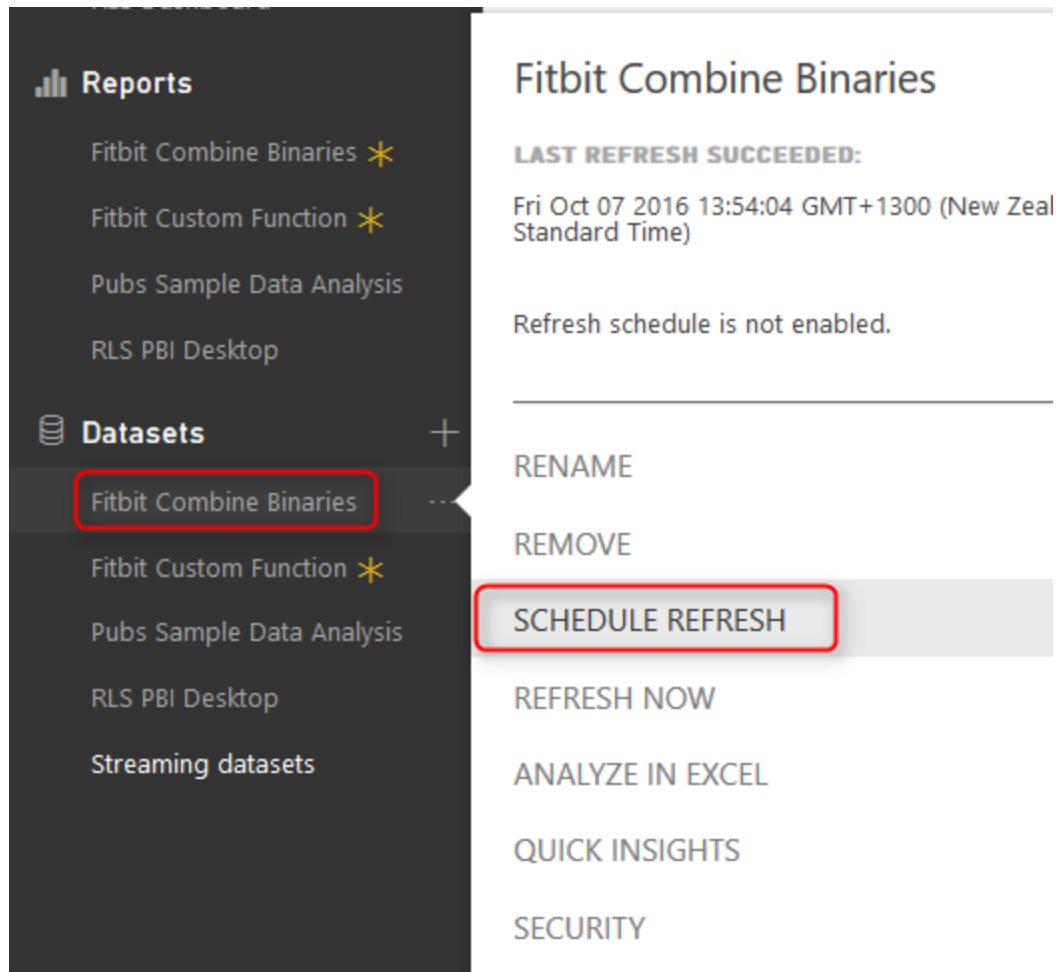
Now Apply Same data type conversion steps and build the same report as we've done earlier. So continue again from Step Data Type Conversion.

Apply the same steps from Step **Data Type Conversion**, and **Build the Sample Report** till saving the report.

Save this Power BI file as Fitbit Combine Binaries. and Publish.

Schedule Refresh

If you have already set up the Data Source for this folder, you don't need to do it again. Just go directly to Schedule Refresh under Fitbit Combine Binaries dataset.



This time in Schedule Refresh setting you won't see any error, and you can simply choose the Gateway and apply. Because we haven't used any custom functions here, so no error message anymore. All good to go.

General

Dashboards

Datasets

Workbooks

Alerts

Fitbit Combine Binaries

Fitbit Custom Function

Pubs Sample Data Analysis

RLS PBI Desktop

Settings for Fitbit Combine Binaries

This dataset has been configured by a.raad.g@gmail.com.

[Refresh history](#)

Gateway connection

To use a data gateway, make sure the computer is online and the data source is added in [Manage Gateways](#). [Learn More](#)

☐ Use your data gateway (Power BI – personal)

(offline, running on REZA-VAIO)

☒ Use a data gateway

Status	Department	Gateway	Contact information	Description
online		RezaSurface	Reza@RADACAD.onmicros...	

Apply

Discard

Under this section there is another section that you can Schedule report to be refreshed, daily or weekly depends on how you want it.

Schedule Refresh

Keep your data up to date

Yes



Refresh frequency

Weekly



Time zone

(UTC+12:00) Auckland, Wellington



Sunday



Monday



Tuesday



Wednesday



Thursday



Friday



Saturday

Time

[Add another time](#)



Send refresh failure notification email to me

Apply

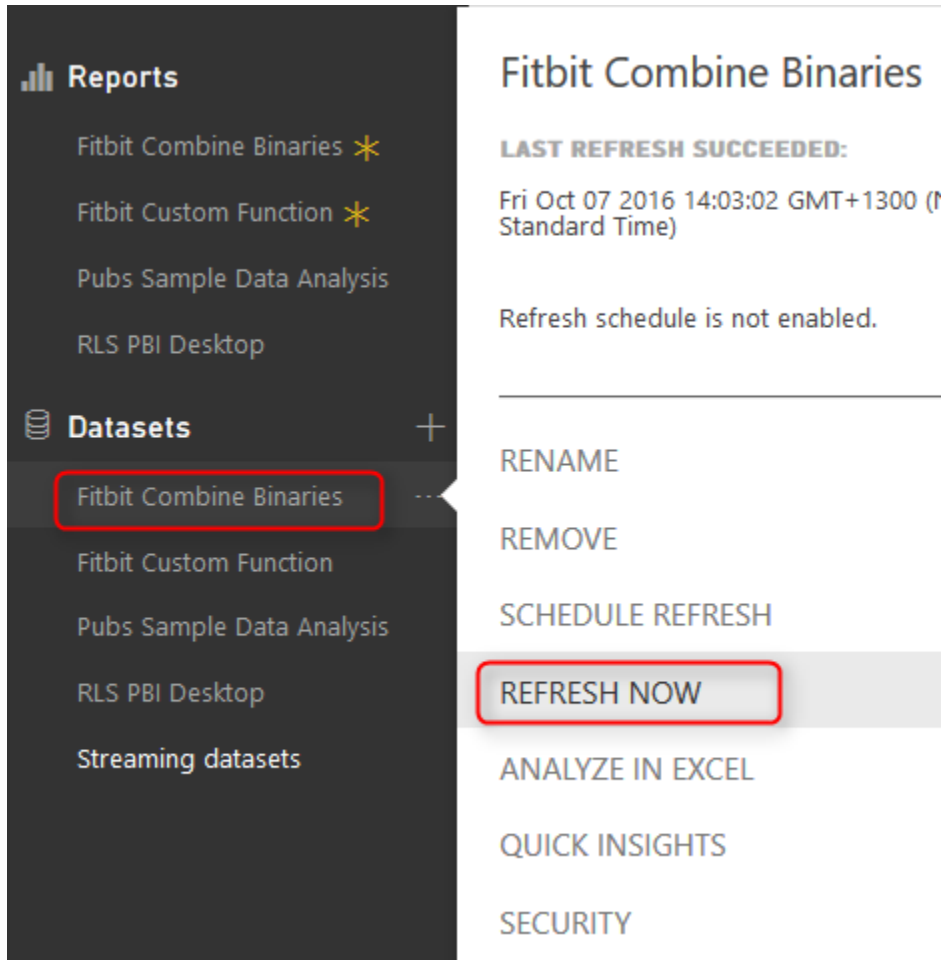
Discard

Testing Automatic Refresh

Now let's test the refresh. Download this new CSV file and paste it in the same folder of other CSV files (the folder that you've set up for gateway as a data source);

[fitbit export 201605](#)

Refresh the dataset with Refresh Now (Or wait until schedule refresh happens! I'm not that patient, so I'll do refresh now 😊)



Reports

- Fitbit Combine Binaries *
- Fitbit Custom Function *
- Pubs Sample Data Analysis
- RLS PBI Desktop

Datasets

- Fitbit Combine Binaries
- Fitbit Custom Function
- Pubs Sample Data Analysis
- RLS PBI Desktop
- Streaming datasets

Fitbit Combine Binaries

LAST REFRESH SUCCEEDED:

Fri Oct 07 2016 14:03:02 GMT+1300 (Standard Time)

Refresh schedule is not enabled.

RENAME

REMOVE

SCHEDULE REFRESH

REFRESH NOW

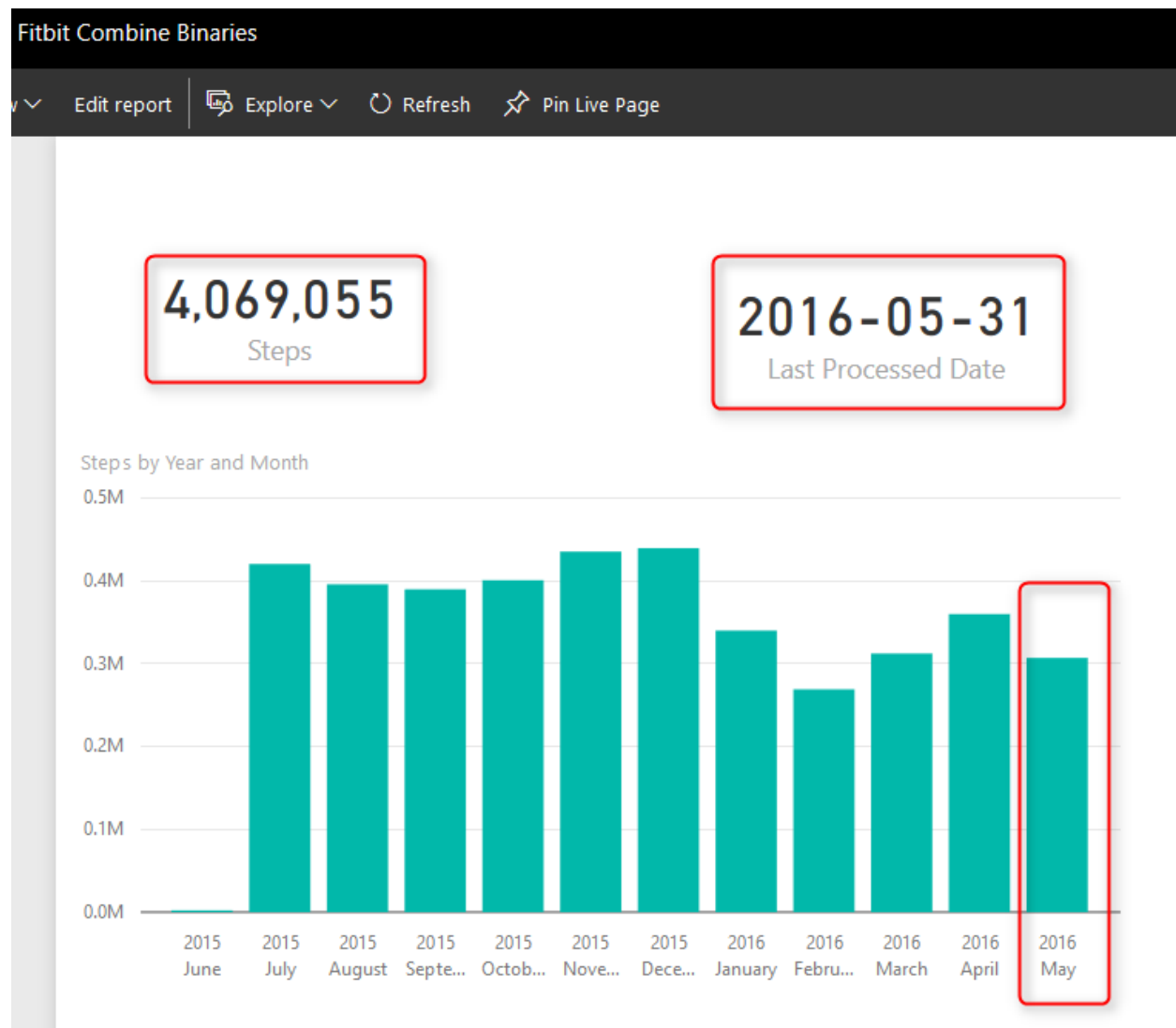
ANALYZE IN EXCEL

QUICK INSIGHTS

SECURITY

If there be no errors in connection and gateway setup and access to the source folder, then refresh finish successfully. Go to report and refresh the report to see the new data.

You will notice that Data of month May is added. Total number of steps is now over 4 million, and last processed Date is 31st of May 2016.





Congratulations, You have done the example successfully, and much harder; You have finished this very long post 😊

Which Method is Best?

If the Gateway issue with custom functions resolves, I'd say first method with custom function is definitely the best method. Because you have vast flexibility with custom function. Combine Binaries doesn't always produce an easy output, with some excel files it might generate some data rows which needs many more steps for cleansing the final data. But for now that gateway has this issue with custom functions, the only possible way is Combine Binaries.

Power BI Q&A; How to Ask Questions?

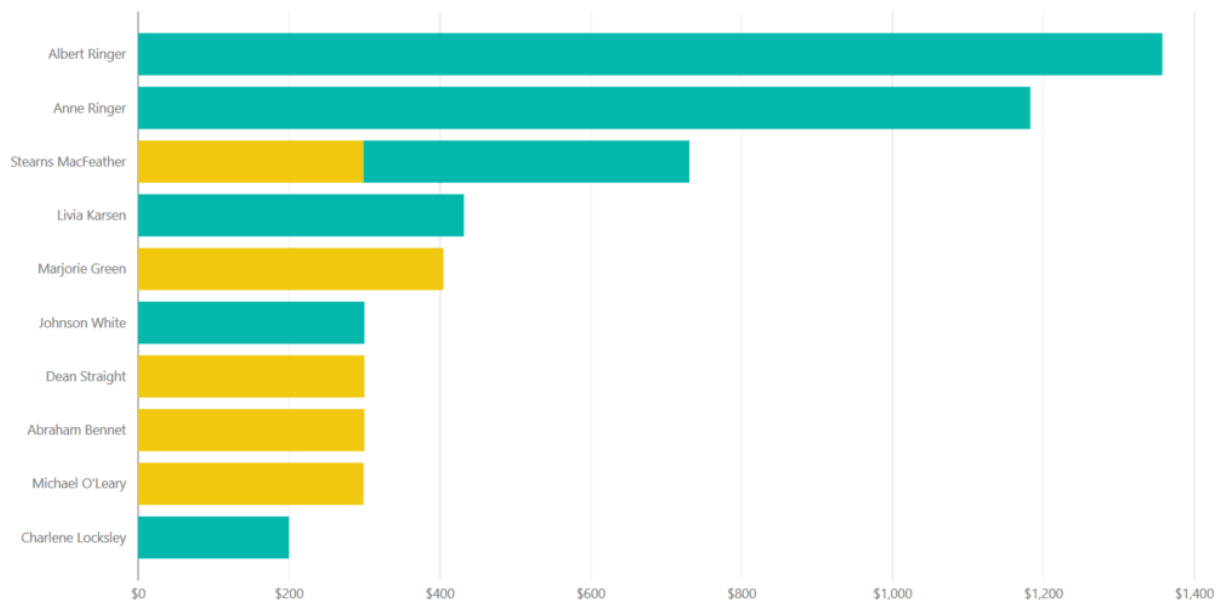
Published Date: March 11, 2016

gend type psychology and business by stacked bar chart sorted sales amount descending  

Show type where type is psychology or business sorted by sales amount and author full name descending as stacked bar chart

Sales_Amount by Author Full Name and type

type ● business ● psychology



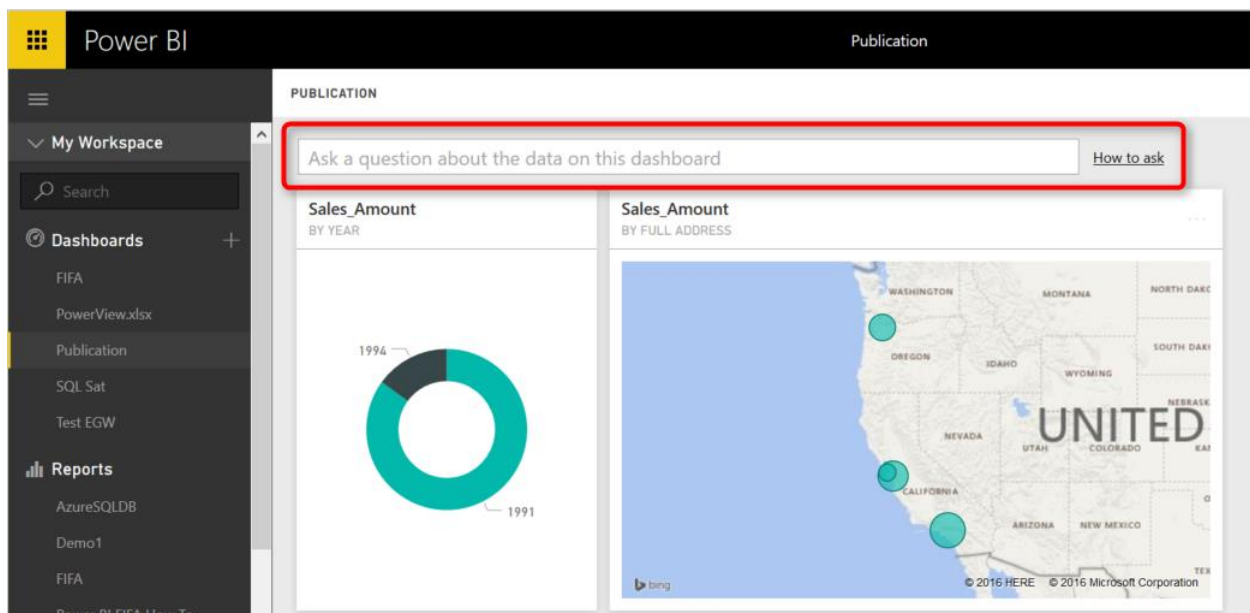
In previous chapters of [Power BI from Rookie to Rock Star](#) you've learned that Power BI reports and models can be published in Power BI website for sharing, scheduling and some other purposes. One of the features in Power BI website which is unique to Power BI and is not available in many BI tools in the market is Power Q&A engine. Power Q&A is a natural question and answering engine in Power BI. With Q&A you can ask your questions from the existing model with natural language and get response with visualization elements! In this post I'll show you some of intelligence questions that you can ask from Q&A to get a very useful response.

Prerequisite for Running Samples of this Post

If you want to run sample of this post and see how they works in action you need to create the Publication dashboard from the Power BI report built on top of Pubs database. In one of my earlier posts in the book I explained [how to create that report](#). You can also ask similar questions from any other model in your Power BI dashboard with just changing name of fields and values to whatever you have in your dataset.

Explore the Data

When you published your Power BI file into the Power BI desktop, usually you create a dashboard for it. For Power Q&A to work (the version of Power Q&A at the time of writing this post) you should create a dashboard for your report. After creating the dashboard you will see the question bar of Q&A on the top of your dashboard.



Let's start by the most basic question:

Explore a Dataset

by typing name of a dataset you can get it in a table view. You can either type in name of the table; such as "sales" or put a "show" keyword before it like "show sales"

sales

Show sales

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount
6380	6871	14/09/1994 00:00:00	5	Net 60	BU1032	\$99.950000000000003
6380	722a	13/09/1994 00:00:00	3	Net 60	PS2091	\$32.850000000000001
7066	A2976	24/05/1993 00:00:00	50	Net 30	PC8888	\$1,000
7066	QA7442.3	13/09/1994 00:00:00	75	ON invoice	PS2091	\$821.25
7067	D4482	14/09/1994 00:00:00	10	Net 60	PS2091	\$109.5
7067	P2121	15/06/1992 00:00:00	20	Net 30	TC4203	\$239
7067	P2121	15/06/1992 00:00:00	20	Net 30	TC7777	\$299.80000000000001
7067	P2121	15/06/1992 00:00:00	40	Net 30	TC3218	\$838
7131	N914008	14/09/1994 00:00:00	20	Net 30	PS2091	\$219
7131	N914014	14/09/1994 00:00:00	25	Net 30	MC3021	\$74.75
7131	P3087a	29/05/1993 00:00:00	15	Net 60	PS3333	\$299.85000000000002
7131	P3087a	29/05/1993 00:00:00	20	Net 60	PS1372	\$431.80000000000001
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS2106	\$175
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS7777	\$199.75
7896	QQ2299	28/10/1993 00:00:00	15	Net 60	BU7832	\$299.85000000000002
7896	TQ456	12/12/1993 00:00:00	10	Net 60	MC2222	\$199.90000000000001
7896	X999	21/02/1993 00:00:00	35	ON invoice	BU2075	\$104.65000000000001
8042	423LL922	14/09/1994 00:00:00	15	ON invoice	MC3021	\$44.850000000000001
8042	423LL930	14/09/1994 00:00:00	10	ON invoice	BU1032	\$199.90000000000001
8042	P723	11/03/1993 00:00:00	25	Net 30	BU1111	\$298.75
8042	QA879.1	22/05/1993 00:00:00	30	Net 30	PC1035	\$688.5

As you can see the Q&A is also smart enough to suggest you a sentence to ask the question.

Filter Values

You can filter values with a WHERE keyword in your sentence. example below fetch all sales for the book titled "Is Anger Enemy?"

show sales where title "Is Anger the Enemy?"

Show sales and titles named "Is Anger the Enemy?"

ord_num	stor_id	ord_date	title_id	title	type
722a	6380	13/09/1994 00:00:00	PS2091	Is Anger the Enemy?	psychology
D4482	7067	14/09/1994 00:00:00	PS2091	Is Anger the Enemy?	psychology
N914008	7131	14/09/1994 00:00:00	PS2091	Is Anger the Enemy?	psychology
QA7442.3	7066	13/09/1994 00:00:00	PS2091	Is Anger the Enemy?	psychology

You can do date filters as well simply by mentioning the date. below sample shows sales for year 1994:

1994 sales

Show sales where ord date is in 1994

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount
6380	6871	14/09/1994 00:00:00	5	Net 60	BU1032	\$99.950000000000003
6380	722a	13/09/1994 00:00:00	3	Net 60	PS2091	\$32.850000000000001
7066	QA7442.3	13/09/1994 00:00:00	75	ON invoice	PS2091	\$821.25
7067	D4482	14/09/1994 00:00:00	10	Net 60	PS2091	\$109.5
7131	N914008	14/09/1994 00:00:00	20	Net 30	PS2091	\$219
7131	N914014	14/09/1994 00:00:00	25	Net 30	MC3021	\$74.75
8042	423LL922	14/09/1994 00:00:00	15	ON invoice	MC3021	\$44.850000000000001
8042	423LL930	14/09/1994 00:00:00	10	ON invoice	BU1032	\$199.90000000000001

Explore Related Datasets

You can explore data from related datasets. For example to get titles and authors (which are in two separate datasets) you can ask:

title and author full name

Show titles and author full name

title_id	title	type	Author Full Name
BU1032	The Busy Executive's Database Guide	business	Abraham Bennet
BU1032	The Busy Executive's Database Guide	business	Marjorie Green
BU1111	Cooking with Computers: Surreptitious Balance Sheets	business	Michael O'Leary
BU1111	Cooking with Computers: Surreptitious Balance Sheets	business	Stearns MacFeather
BU2075	You Can Combat Computer Stress!	business	Marjorie Green
BU7832	Straight Talk About Computers	business	Dean Straight
MC2222	Silicon Valley Gastronomic Treats	mod_cook	Innes del Castillo
MC3021	The Gourmet Microwave	mod_cook	Anne Ringer
MC3021	The Gourmet Microwave	mod_cook	Michel DeFrance
PC1035	But Is It User Friendly?	popular_comp	Cheryl Carson
PC8888	Secrets of Silicon Valley	popular_comp	Ann Dull
PC8888	Secrets of Silicon Valley	popular_comp	Sheryl Hunter
PC9999	Net Etiquette	popular_comp	Charlene Locksley
PS1372	Computer Phobic AND Non-Phobic Individuals: Behavior Variations	psychology	Livia Karsen
PS1372	Computer Phobic AND Non-Phobic Individuals: Behavior Variations	psychology	Stearns MacFeather
PS2091	Is Anger the Enemy?	psychology	Albert Ringer
PS2091	Is Anger the Enemy?	psychology	Anne Ringer
PS2106	Life Without Fear	psychology	Albert Ringer
PS3333	Prolonged Data Deprivation: Four Case Studies	psychology	Johnson White
PS7777	Emotional Security: A New Algorithm	psychology	Charlene Locksley
TC3218	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean	trad_cook	Sylvia Panteley
TC4203	Fifty Years in Buckingham Palace Kitchens	trad_cook	Reginald Blotchet-Halls
TC7777	Sushi, Anyone?	trad_cook	Akiko Yokomoto
TC7777	Sushi, Anyone?	trad_cook	Burt Gringlesby
TC7777	Sushi, Anyone?	trad_cook	Michael O'Leary

Sorting Data

You can use Ascending or Descending commands in your question.

show sales sorted by sales amount descending as table

Show sales sorted by sales amount descending as table

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount	
7066	A2976	24/05/1993 00:00:00	50	Net 30	PC8888	\$1,000	▼
7067	P2121	15/06/1992 00:00:00	40	Net 30	TC3218	\$838	
7066	QA7442.3	13/09/1994 00:00:00	75	ON invoice	PS2091	\$821.25	
8042	QA879.1	22/05/1993 00:00:00	30	Net 30	PC1035	\$688.5	
7131	P3087a	29/05/1993 00:00:00	20	Net 60	PS1372	\$431.80000000000001	
7131	P3087a	29/05/1993 00:00:00	15	Net 60	PS3333	\$299.85000000000002	
7896	QQ2299	28/10/1993 00:00:00	15	Net 60	BU7832	\$299.85000000000002	
7067	P2121	15/06/1992 00:00:00	20	Net 30	TC7777	\$299.80000000000001	
8042	P723	11/03/1993 00:00:00	25	Net 30	BU1111	\$298.75	
7067	P2121	15/06/1992 00:00:00	20	Net 30	TC4203	\$239	
7131	N914008	14/09/1994 00:00:00	20	Net 30	PS2091	\$219	
7896	TQ456	12/12/1993 00:00:00	10	Net 60	MC2222	\$199.90000000000001	
8042	423LL930	14/09/1994 00:00:00	10	ON invoice	BU1032	\$199.90000000000001	
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS7777	\$199.75	
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS2106	\$175	
7067	D4482	14/09/1994 00:00:00	10	Net 60	PS2091	\$109.5	
7896	X999	21/02/1993 00:00:00	35	ON invoice	BU2075	\$104.65000000000001	
6380	6871	14/09/1994 00:00:00	5	Net 60	BU1032	\$99.95000000000003	
7131	N914014	14/09/1994 00:00:00	25	Net 30	MC3021	\$74.75	
8042	423LL922	14/09/1994 00:00:00	15	ON invoice	MC3021	\$44.85000000000001	
6380	722a	13/09/1994 00:00:00	3	Net 60	PS2091	\$32.85000000000001	

Also notice that you can change the order with simply hovering your mouse on each column and clicking on sort icon that appears there. I also have the “As Table” at the end of question above that force result set to be viewed as a table visualization.

More Filters

Equity Filter

You can filter on equity of values. such as sales on the payerms Net 60

sales with payterms Net 60

Show Net 60 sales

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount
6380	6871	14/09/1994 00:00:00	5	Net 60	BU1032	\$99.95000000000003
6380	722a	13/09/1994 00:00:00	3	Net 60	PS2091	\$32.850000000000001
7067	D4482	14/09/1994 00:00:00	10	Net 60	PS2091	\$109.5
7131	P3087a	29/05/1993 00:00:00	15	Net 60	PS3333	\$299.85000000000002
7131	P3087a	29/05/1993 00:00:00	20	Net 60	PS1372	\$431.80000000000001
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS2106	\$175
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS7777	\$199.75
7896	QQ2299	28/10/1993 00:00:00	15	Net 60	BU7832	\$299.85000000000002
7896	TQ456	12/12/1993 00:00:00	10	Net 60	MC2222	\$199.90000000000001

Range Filter

Use "between" keyword to filter for a range

sales where qty between 10 and 30 as table

Show sales and their qty where qty is between 10 and 30 as table

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount
7067	D4482	14/09/1994 00:00:00	10	Net 60	PS2091	\$109.5
7067	P2121	15/06/1992 00:00:00	20	Net 30	TC4203	\$239
7067	P2121	15/06/1992 00:00:00	20	Net 30	TC7777	\$299.80000000000001
7131	N914008	14/09/1994 00:00:00	20	Net 30	PS2091	\$219
7131	N914014	14/09/1994 00:00:00	25	Net 30	MC3021	\$74.75
7131	P3087a	29/05/1993 00:00:00	15	Net 60	PS3333	\$299.85000000000002
7131	P3087a	29/05/1993 00:00:00	20	Net 60	PS1372	\$431.80000000000001
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS2106	\$175
7131	P3087a	29/05/1993 00:00:00	25	Net 60	PS7777	\$199.75
7896	QQ2299	28/10/1993 00:00:00	15	Net 60	BU7832	\$299.85000000000002
7896	TQ456	12/12/1993 00:00:00	10	Net 60	MC2222	\$199.90000000000001
8042	423LL922	14/09/1994 00:00:00	15	ON invoice	MC3021	\$44.850000000000001
8042	423LL930	14/09/1994 00:00:00	10	ON invoice	BU1032	\$199.90000000000001
8042	P723	11/03/1993 00:00:00	25	Net 30	BU1111	\$298.75
8042	QA879.1	22/05/1993 00:00:00	30	Net 30	PC1035	\$688.5

Exact Date Filter

You can mention the exact date simply like sales October 28, 1993

sales october 28 1993
Show sales where ord date is 10/28/1993

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount
7896	QQ2299	28/10/1993 00:00:00	15	Net 60	BU7832	\$299.85000000000002

Date Range Filter

You can filter dataset for dates before or after a date as below

sales after october 28 1993
Show sales where ord date is after 10/28/1993

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount
6380	6871	14/09/1994 00:00:00	5	Net 60	BU1032	\$99.95000000000003
6380	722a	13/09/1994 00:00:00	3	Net 60	PS2091	\$32.850000000000001
7066	QA7442.3	13/09/1994 00:00:00	75	ON invoice	PS2091	\$821.25
7067	D4482	14/09/1994 00:00:00	10	Net 60	PS2091	\$109.5
7131	N914008	14/09/1994 00:00:00	20	Net 30	PS2091	\$219
7131	N914014	14/09/1994 00:00:00	25	Net 30	MC3021	\$74.75
7896	TQ456	12/12/1993 00:00:00	10	Net 60	MC2222	\$199.90000000000001
8042	423LL922	14/09/1994 00:00:00	15	ON invoice	MC3021	\$44.850000000000001
8042	423LL930	14/09/1994 00:00:00	10	ON invoice	BU1032	\$199.90000000000001

Filter Relative to Today's Date

Fetch data with filters such as past or next periods. for example fetch dataset information for past 10 years or so.

sales whitin the past 22 years

Show sales where ord date is from 22 years before now to last year

stor_id	ord_num	ord_date	qty	payterms	title_id	Sales_Amount
6380	6871	14/09/1994 00:00:00	5	Net 60	BU1032	\$99.9500000000000003
6380	722a	13/09/1994 00:00:00	3	Net 60	PS2091	\$32.8500000000000001
7066	QA7442.3	13/09/1994 00:00:00	75	ON invoice	PS2091	\$821.25
7067	D4482	14/09/1994 00:00:00	10	Net 60	PS2091	\$109.5
7131	N914008	14/09/1994 00:00:00	20	Net 30	PS2091	\$219
7131	N914014	14/09/1994 00:00:00	25	Net 30	MC3021	\$74.75
8042	423LL922	14/09/1994 00:00:00	15	ON invoice	MC3021	\$44.8500000000000001
8042	423LL930	14/09/1994 00:00:00	10	ON invoice	BU1032	\$199.9000000000000001

Aggregation

You can even do calulations in your question, how impresive is that. For example ask Total Sales amount or Average Sales Amount

total sales amount

Show total sales amount

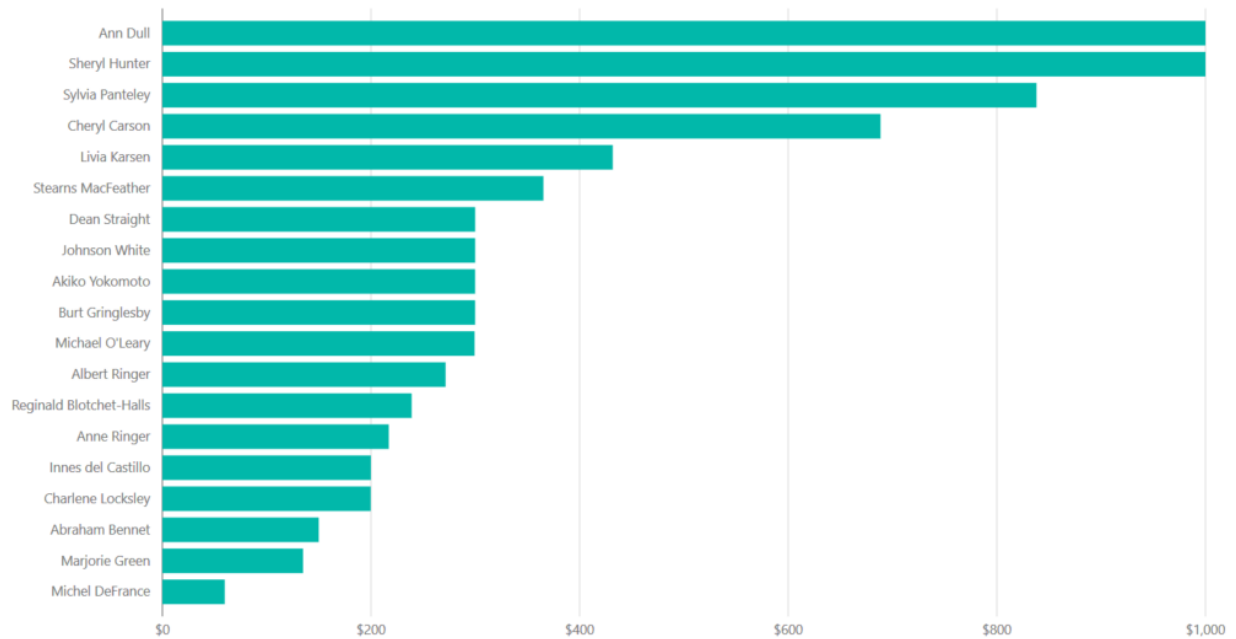
\$6,676.899999999999600
Sales_Amount

You can then aggregate across attribute values. such as; average sales amount per author

average sales amount per author full name
Show author full name and average sales amount



Average of Sales_Amount by Author Full Name



Source: pubs

Visualization Customization

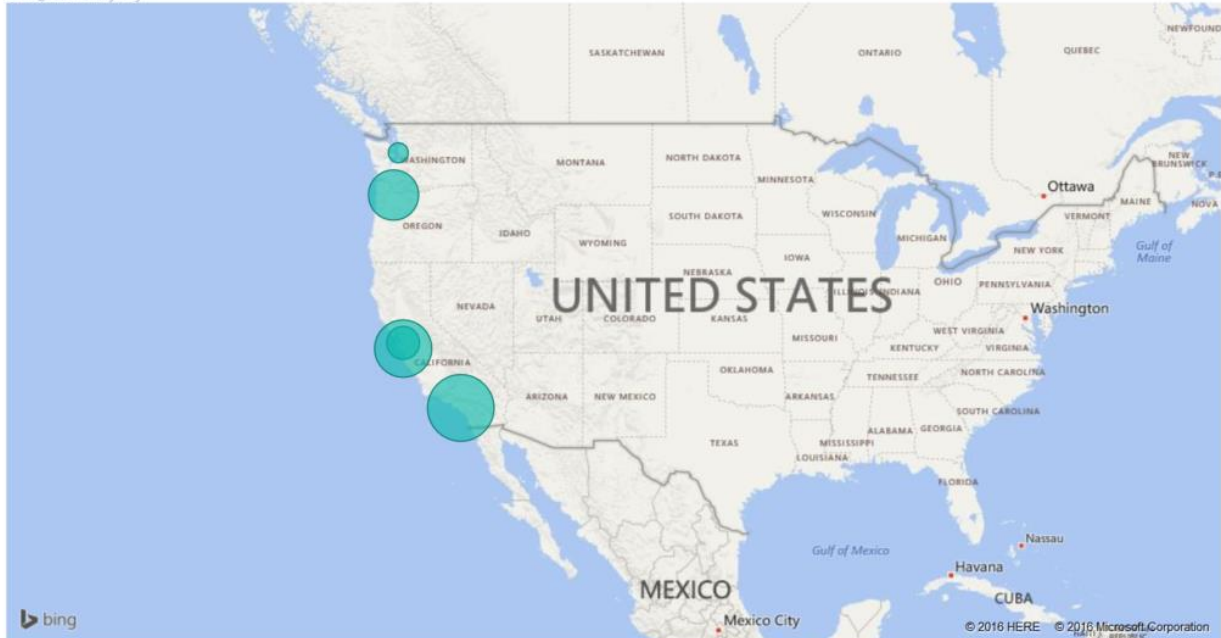
You can ask what type of visual you want to see the result set to be shown.

total sales amount by city by map

Show city that stores are in and total sales amount as map





Sales_Amount by city



Combination of some above

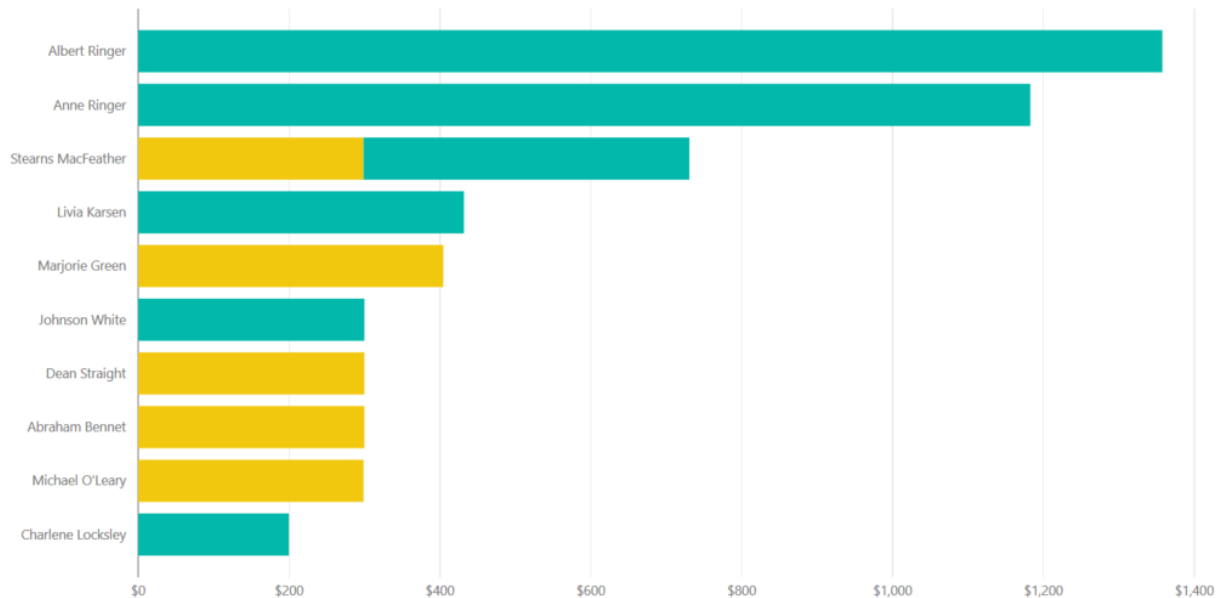
Now that you know some keywords to put in your question, you can ask questions that have combined items to create more useful reports such as:

gend type psychology and business by stacked bar chart sorted sales amount descending  

Show type where type is psychology or business sorted by sales amount and author full name descending as stacked bar chart

Sales_Amount by Author Full Name and type

type ● business ● psychology



You can pin every answer you want to your dashboard to save it for later.

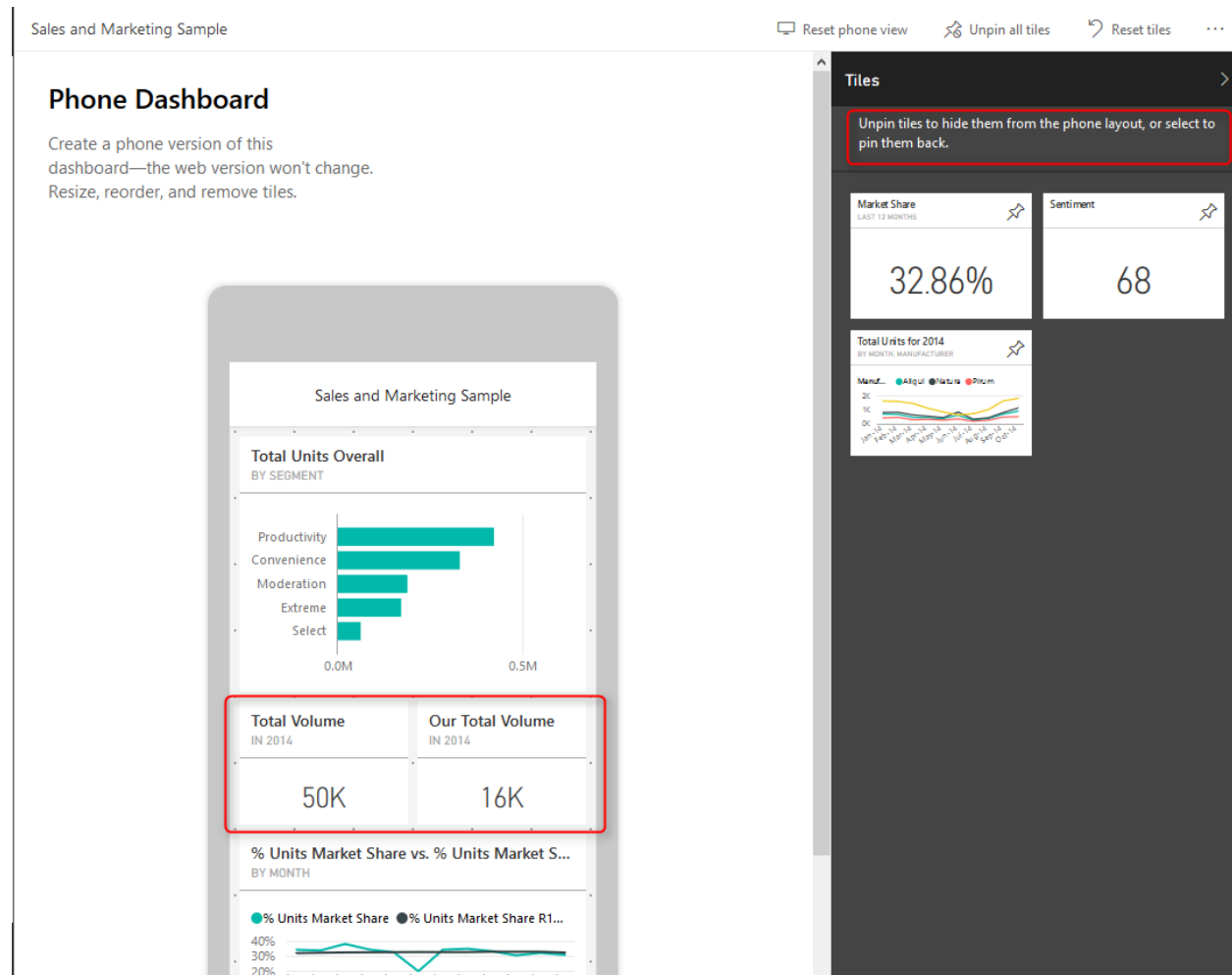
There are some other keywords that helps you to ask better questions and get better answers from Power BI, but let's keep this post simple for now. In future posts I'll explain how to build you Power BI model in a way that responds best to question from Q&A. tips in creating relationship, naming columns, data types and etc.

Your Turn

Now go to your Power BI dashboard and have fun with Q&A and bring some of examples you've done. I'd love to see what you will find 😊

Dashboard Design for Mobile Power BI

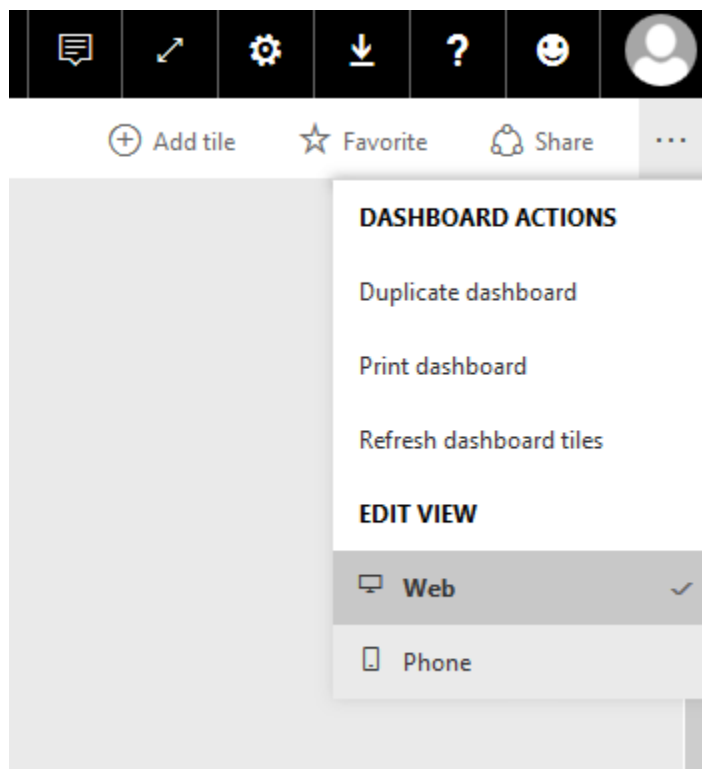
Published Date: July 18, 2016



In this blog post of [Mobile friendly Power BI report development](#) series I would explain how to design dashboards for mobile devices. Fortunately this has come right at the time of the recent update on Power BI service with ability to design dashboard page specifically for mobile device. The dashboard design for mobile does work with all mobile sizes and shapes as long as it is one of the supporting mobile operating systems; IOS, Windows Phone, and Android. If you have worked with Mobile report publisher of SSRS 2016 you will see very similar experience here. If you are interested to learn more about Power BI, read [Power BI online book: from Rookie to Rock Star](#).

Change Dashboard View

With the latest update of Power BI service, you can now change the view of Dashboard for mobile device. For using this feature you should login to Power BI, and then on a selected dashboard click on ellipsis button on the top right hand side of the dashboard (below your account's profile photo), then choose Phone under Edit View.



The default view is Web view, and when you change it to Phone view you will see a message about switching to phone version.

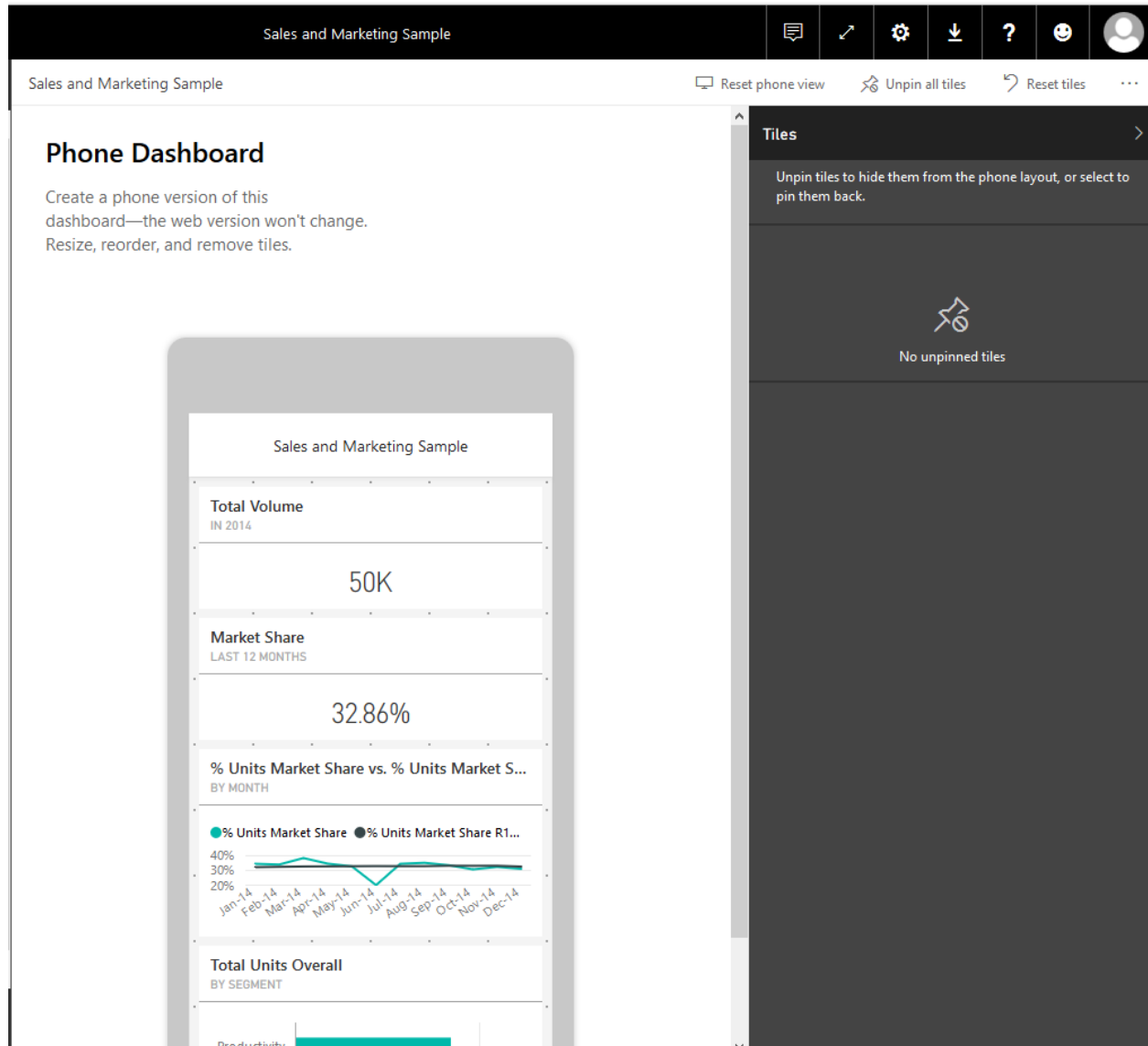
Phone Version



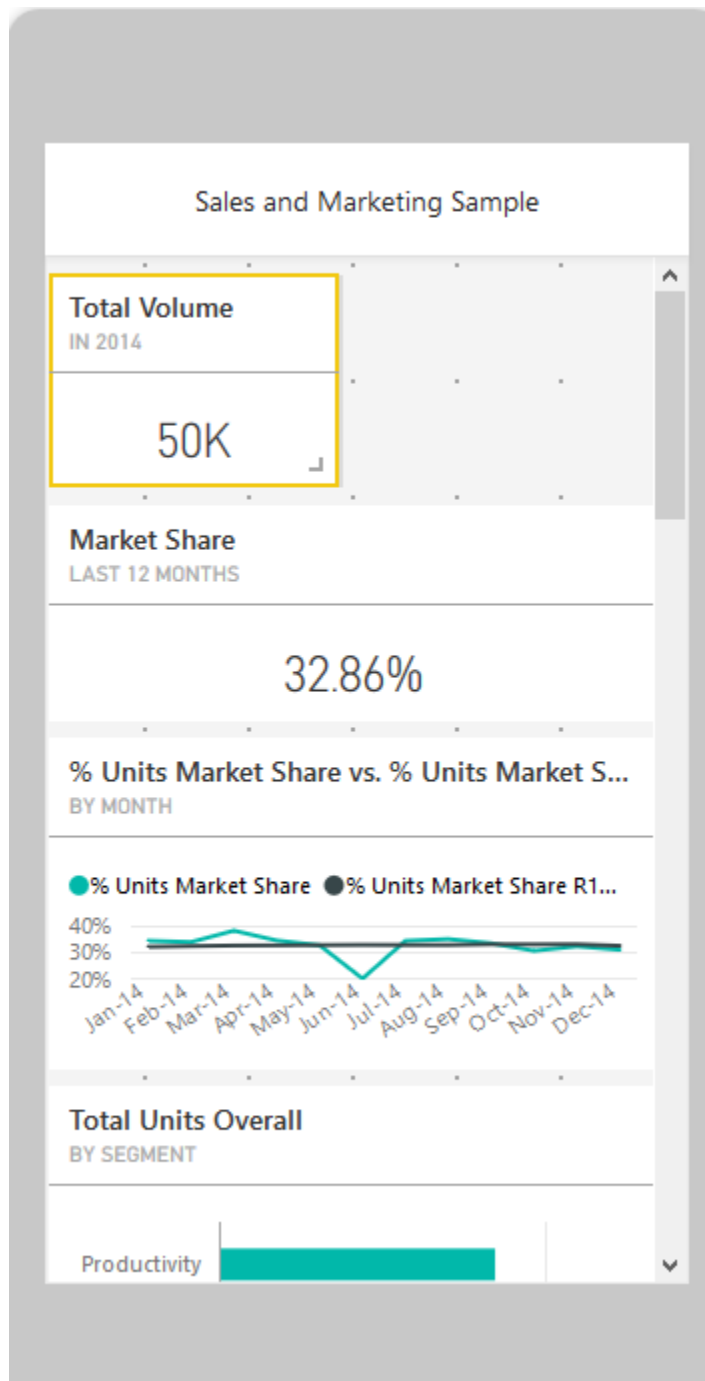
When switching to phone version, your dashboard tiles will re-arrange to phone customization on all mobile devices. You may revert to web version at any time.

[Continue](#)[Cancel](#)

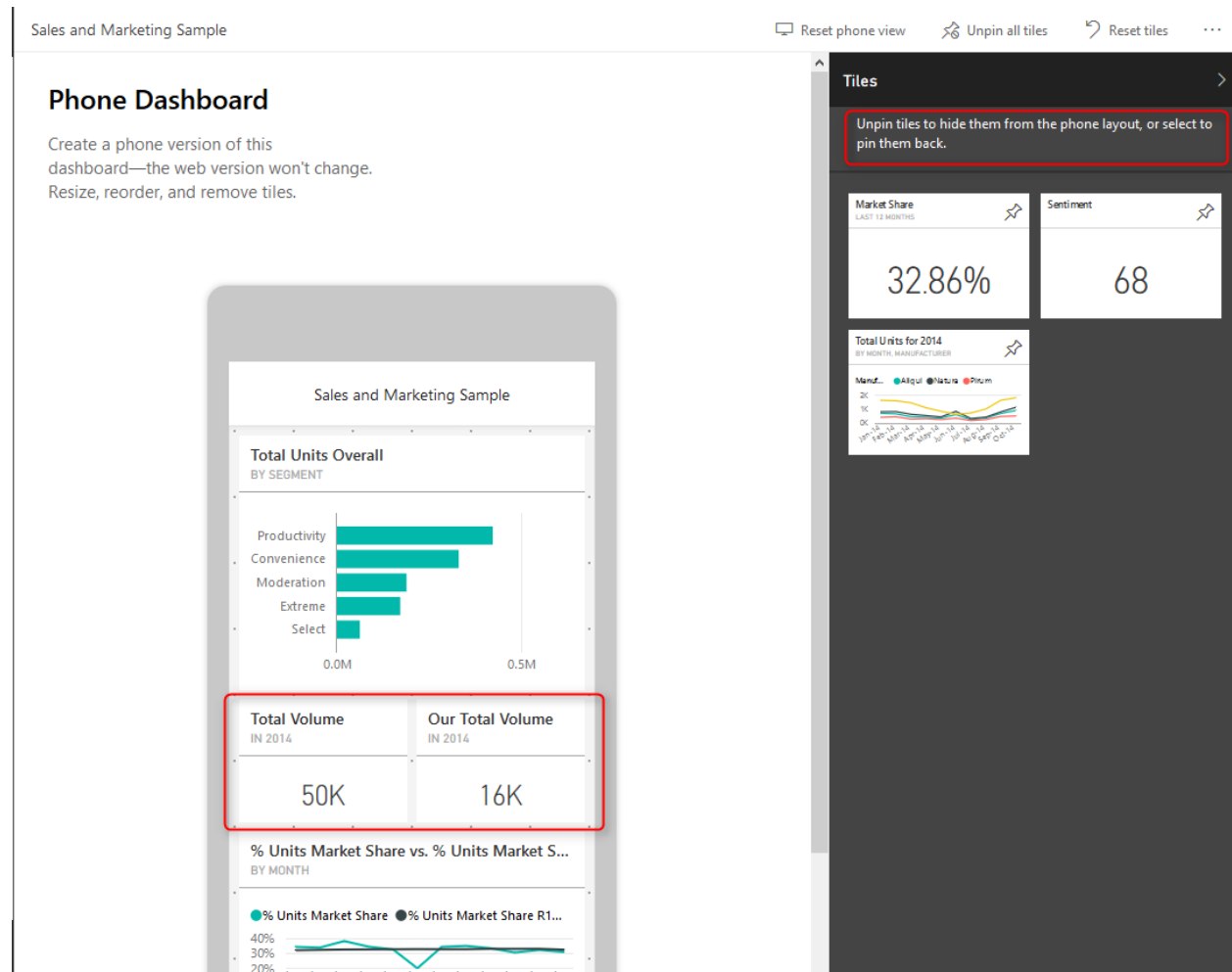
Don't worry about this message, you won't lose any configuration on your web version of dashboard. In fact your dashboard would have two different views; Web view, and Phone view. After entering to Phone view you will see a phone view with tiles of dashboard added to it automatically.



As you can see this is a view of smart phone, and the good thing about it is that this is size independent. so works with all phone sizes. You can change size or order of a tile as you want with drag and drop or moving their borders to resize it.

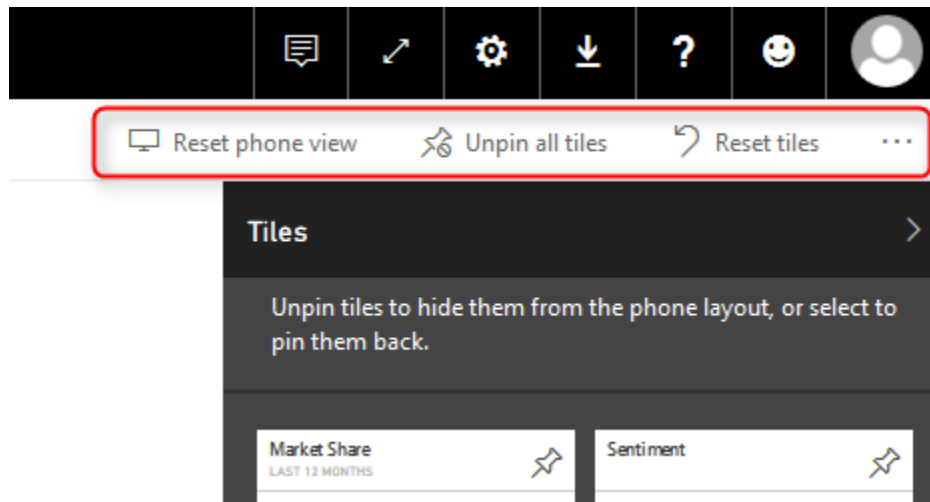


This feature is a great step forward for mobile dashboards, because previously dashboards only designed for web, and they automatically set for phone view. Now we have the ability to design exactly what we want to show. We can event unpin some of tiles from phone view's dashboard. unpinned tiles will stay in the Tiles pane in the right hand side, and you can add them to the dashboard whenever you like.



As you can see in above screenshot now you can arrange the view to have two or more (if you can fit in the screen of phone) tiles in same horizontal position. This is great view specially when you have only numbers to show, and uses whole width of the phone rather than having blank spaces around.

You can reset the view anytime you want and design a new dashboard from the beginning



View Reports on Mobile Device

The reports view on mobile devices still needs some more tips and tricks to work with. If you like to learn more how to design reports which are mobile friendly read my [blog post](#) here.

Mobile Reports with Power BI Gets Better and Better

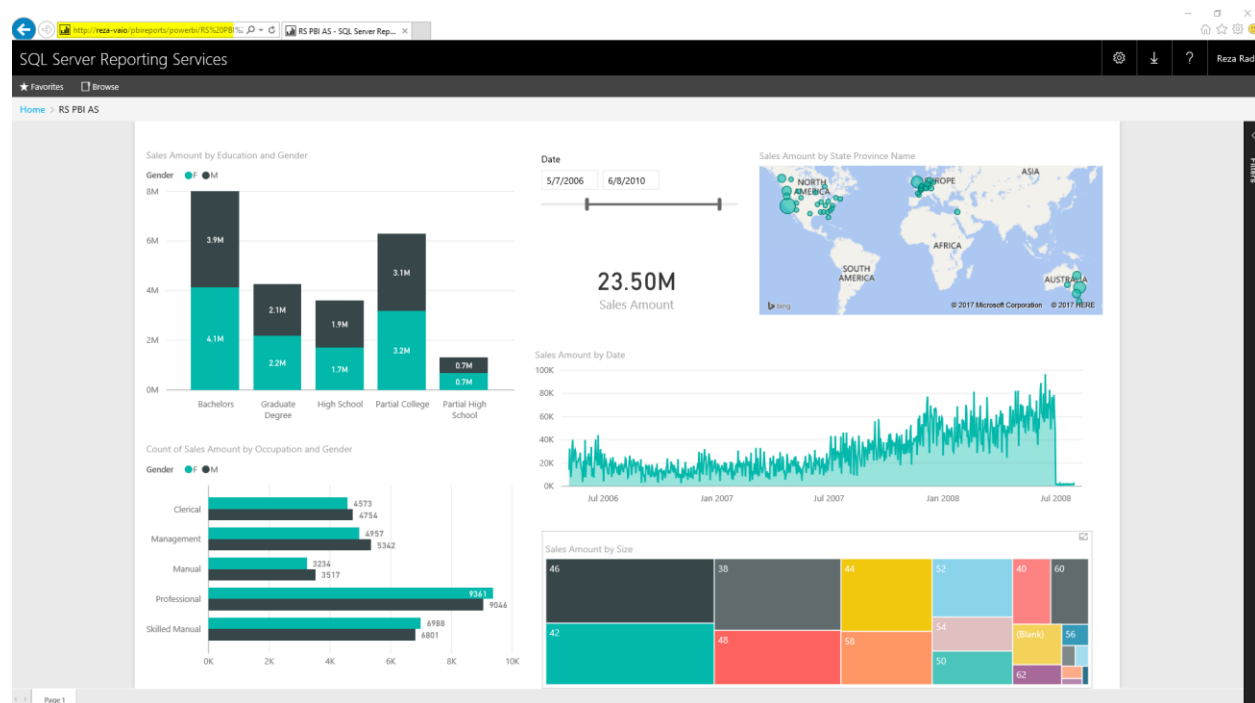
For those of you who are using mobile/phone reports for Power BI the great news is that many awesome new features are coming. Thanks to efforts of Microsoft Power BI team, you will have much better view for mobile reports. Stay tuned for updates.

Power Q&A for Mobile Dashboard?

As you probably noticed, there is no Power Q&A for Phone dashboard. I personally think this would be great to have such feature even on a phone device. However Microsoft team would like to have a feedback from users about this. So your turn now, would you like to have Q&A for mobile dashboard? if yes, please comment below, and also mention possible use case for your mobile/phone users to use Q&A feature.

Power BI On-Premises; Hello World!

Published Date: January 18, 2017



It has been a very long time request from users around the world to make Power BI available for on-premises, not for using on-premises data sources (this is available from long time ago with gateways), but for publishing reports into on-premises server. With the fast pace of development of Power BI this feature was looking so impossible to achieve, however the [great news](#) is that Power BI can be published to on-premises now. This feature is still preview at the time of writing this post, but I believe it will be soon generally available. So the wait is over, you can now host Power BI reports in your on-premises SSRS Server. In this quick post I'll show you how easy is to set up this with Technical Preview of Reporting Services for Power BI. If you would like to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

Technical Preview of SQL Server Reporting Services

For using this preview version, you need to download and install Technical Preview of SQL Server Reporting Services (SSRS) from here:

<https://www.microsoft.com/en-us/download/details.aspx?id=54610>

This download includes the SSRS Technical Preview version where you can publish your Power BI Desktop files to it, and also specific version of Power BI Desktop, which you can build your Power BI reports with it (Yes, for this preview version, you need to create reports with this specific version of Power BI Desktop named as SQL Server Reporting Service Power BI Desktop);

Power BI reports in SQL Server Reporting Services - Technical Preview

Language: English	Download
--------------------------	-----------------

Create reports with Power BI Desktop and publish them to an on-premises report server with this Technical Preview.

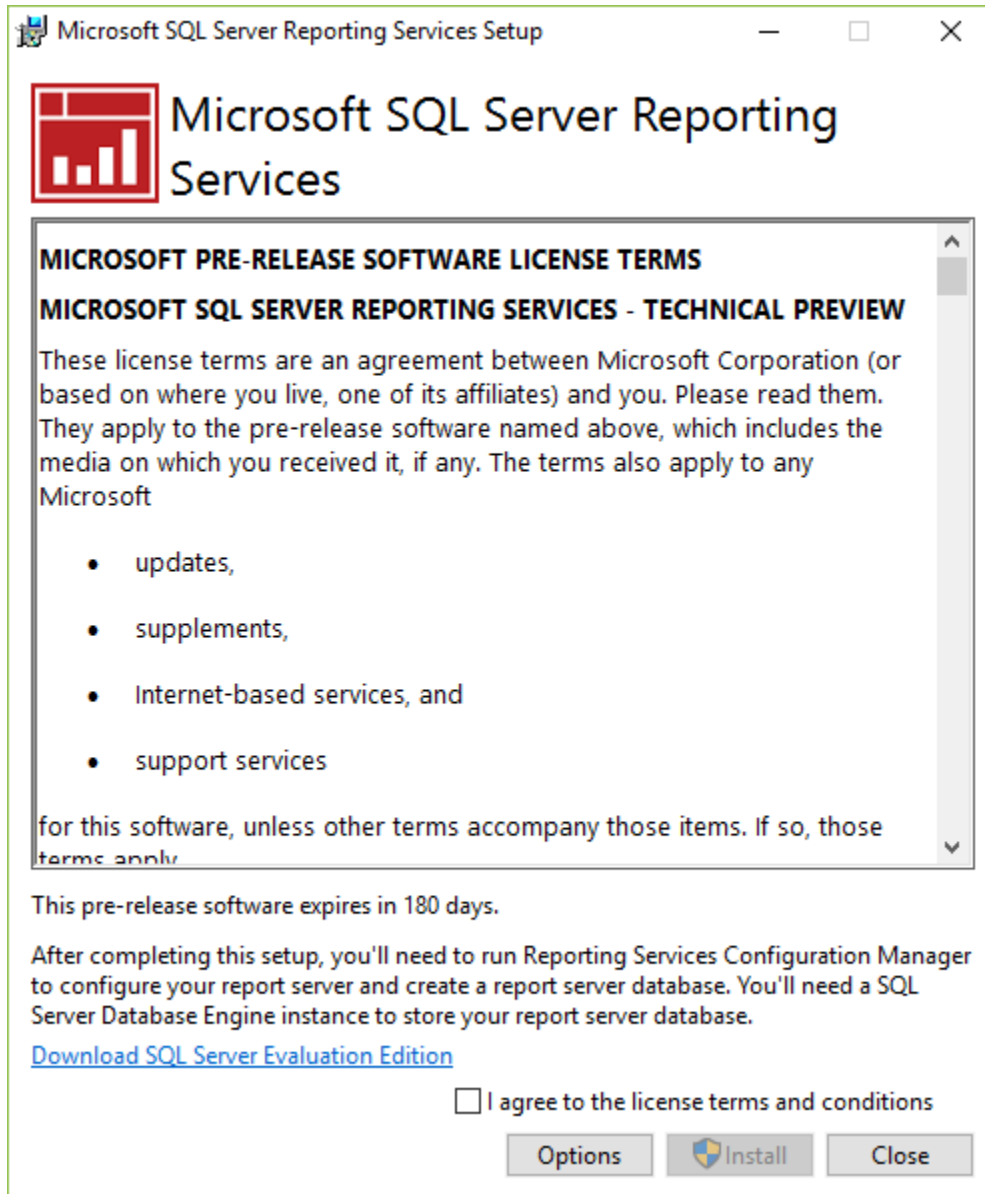
Details

Note: There are multiple files available for this download. Once you click on the "Download" button, you will be prompted to select the files you need.

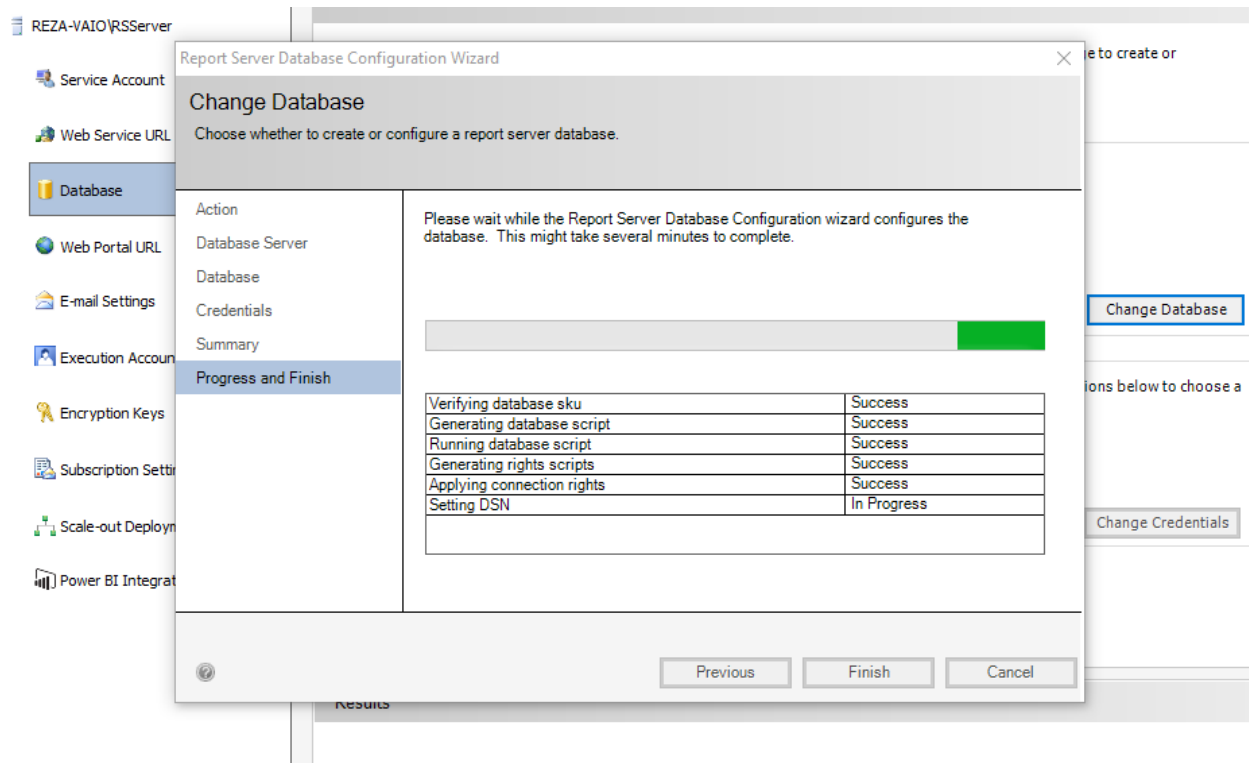
Version:	Date Published:
0.1	1/11/2017
File Name:	File Size:
PBIDesktopRS.msi	102.2 MB
PBIDesktopRS_x64.msi	118.3 MB
SQLServerReportingServices.exe	49.6 MB

Installation

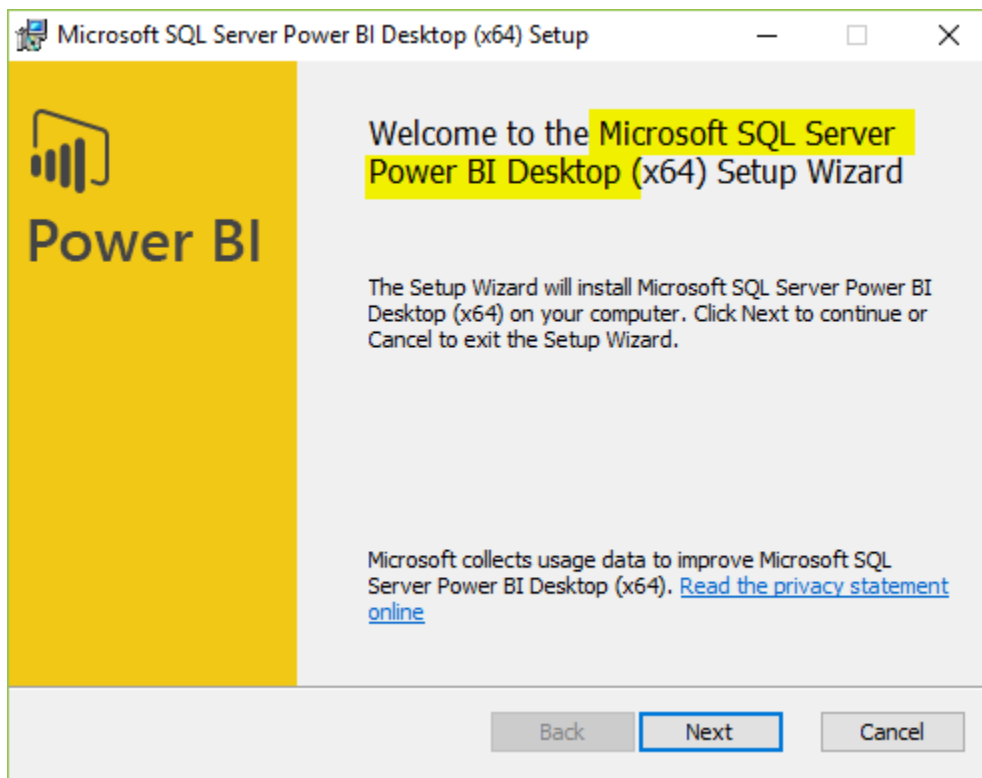
Installation of SSRS Technical Preview is easy, just continue the setup which is only for SSRS (You don't need to install whole SQL Server package for it);



After installing SSRS, you need to set up Report Server in Configuration Manager (Mainly you need to set up a database for report server);



Next step is to install Power BI Desktop version for SSRS;



Now you are ready to build the report;

Building a Report

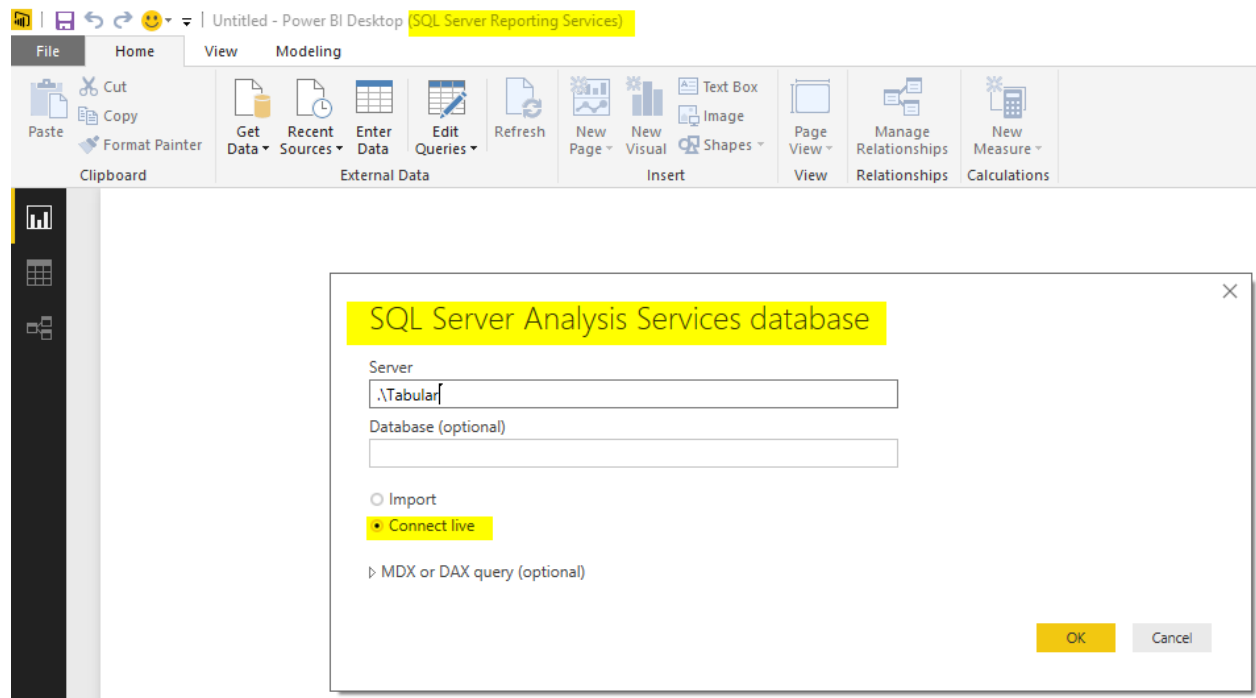
Note that with this preview version you can only use SQL Server Analysis Services (SSAS) Live connection as the source, nothing else. If you use other sources, you will get this error:

Can't upload this report

We can't open this Power BI Desktop report because it contains an unsupported data source connection or embedded data. This preview version supports only Power BI Desktop reports that connect live to SQL Server Analysis Services (on-premises).

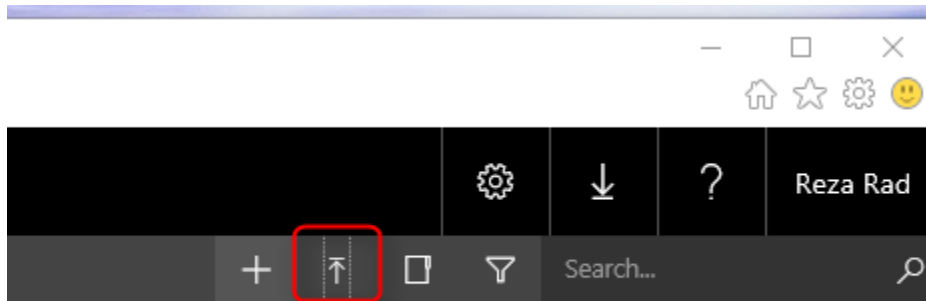
OK

So simply select a SSAS Live query and build a report;

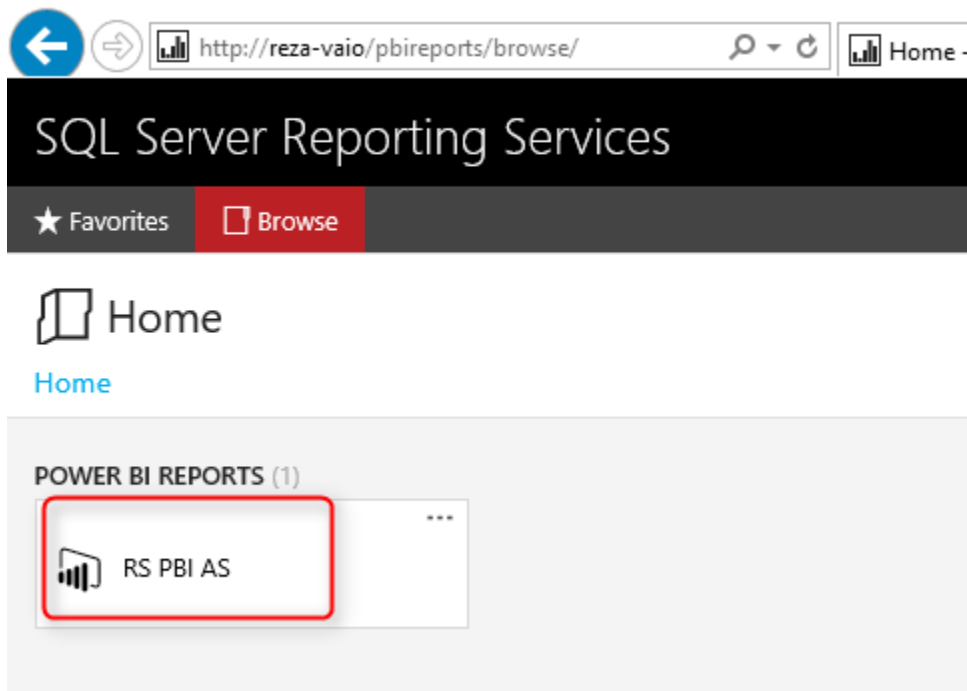


Publish to On-Premises

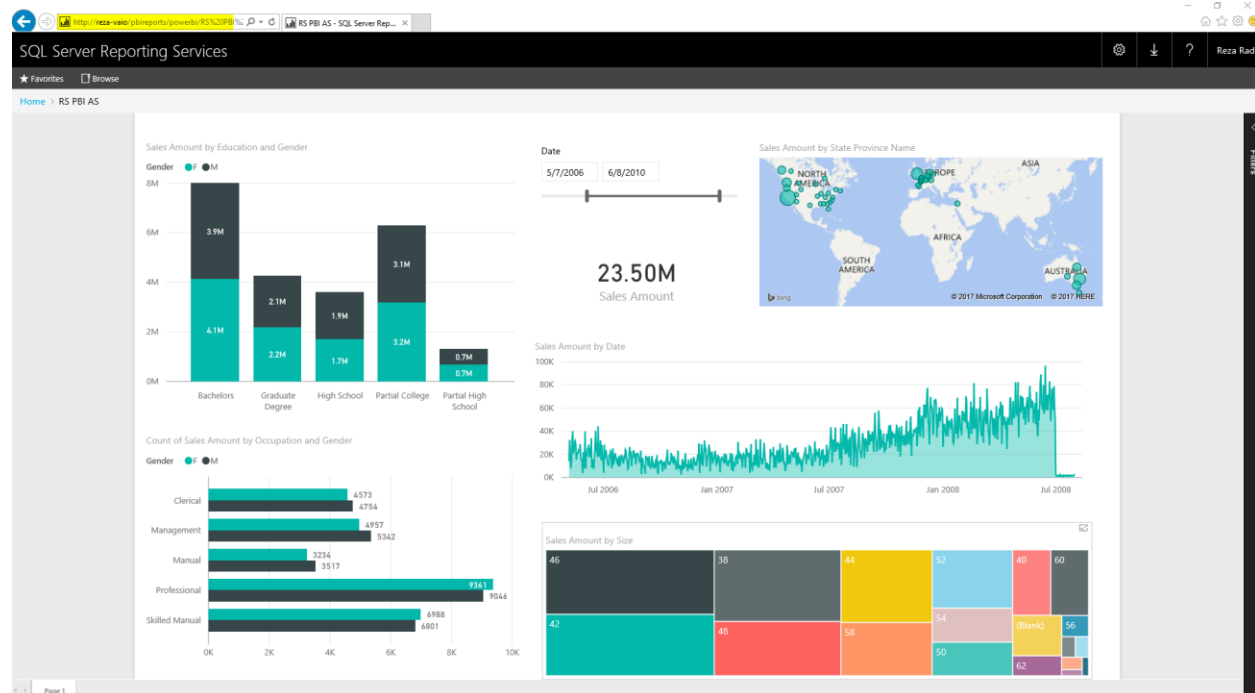
After building the report, just save it, and then in SSRS Report Manager, upload the file;



Your Power BI report will appear in report manager then as a Power BI Report object



Click on the Power BI report to open, and Yaaay! You've got Power BI on-premises! Fully interactive.



My Thoughts

Why Power BI On-Premises?

We are in a world that rapidly running towards cloud. Your files are in Dropbox, or OneDrive these days, Your photos uploaded to a cloud storage, your emails are all backed up in a cloud backup media, and I'm in this thinking that in next few years, we might eat our food from a cloud kitchen! However there are still businesses and companies who require some on-premises solutions, and as long as a requirement exists, there should be an answer for it. Power BI for On-Premises bring the power of self-service, interactive reports of Power BI to these businesses. Power BI for On-premises is a great big step towards utilizing better data insight in all environments.

How the Costing Works?

To be honest, I don't know yet. This is not yet fully released! However the costing and licensing would be definitely different from Power BI in cloud. Probably instead of per user, it might be through Enterprise or licensing editions like that. We need to wait to see that.

Would it be Similar to Power BI in Cloud?

I don't think so. Power BI on cloud (or Power BI Service), has so many features in it; Dashboards, Power Q&A, Security, Sharing, Administration, data streaming input, and many other great features. These features will take time to implement in SSRS, and by the time that these be implemented in SSRS, some new features will be implemented in Power BI Service. My gut feeling is that Power BI Service would be the full experience of Power BI.

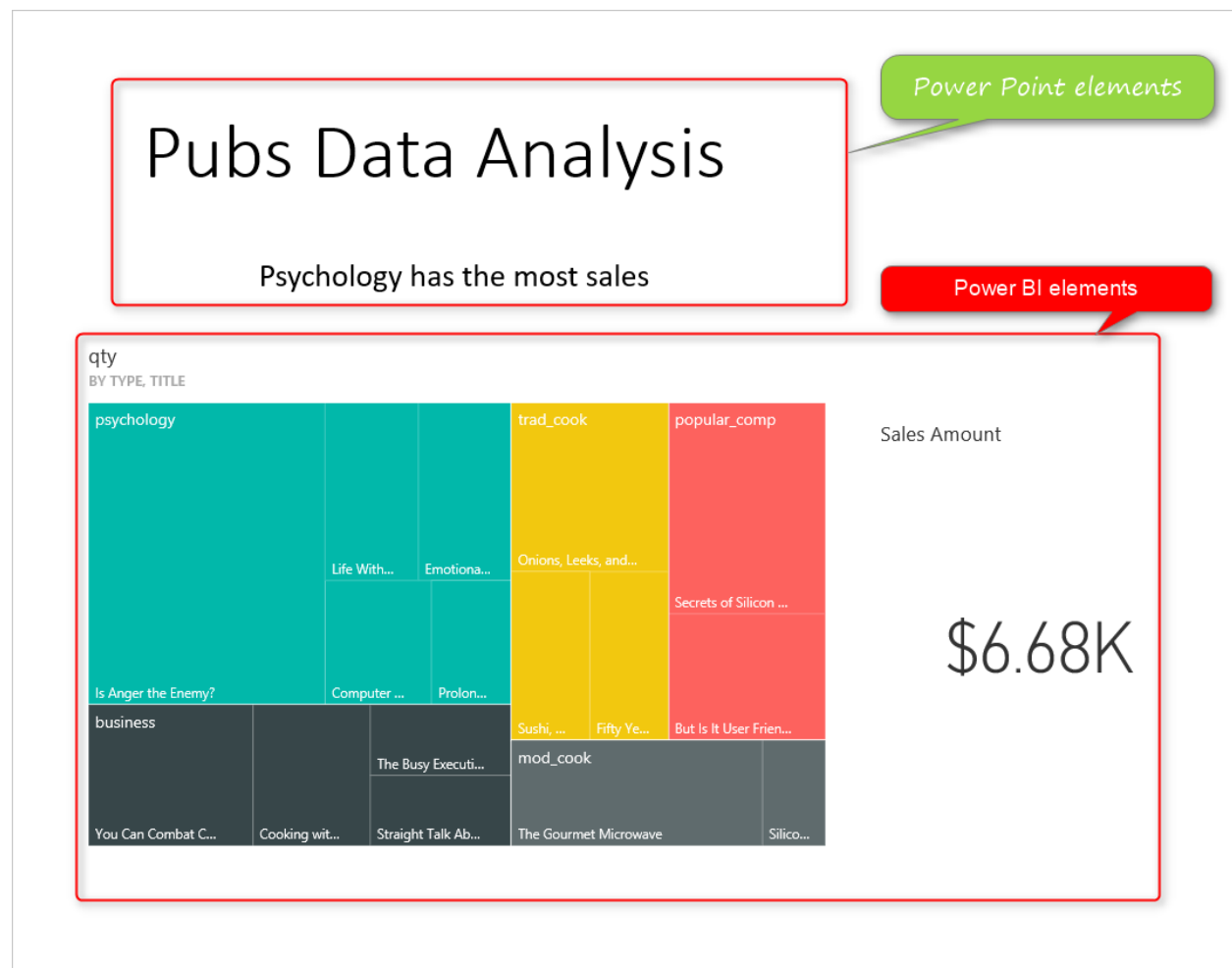
In overall I believe this is a great step towards better world, using data insight in every environment. Thanks Microsoft team for this great enhancement.

Your thoughts?

Let me know what do you think of this change? I'd love to hear your thinking 😊

Power BI Story in PPT Slides with Comments

Published Date: September 5, 2016



This is not story of Power BI in Power Point slides. This post is about how to use Power Point integration with Power BI for story telling of Power BI reports and dashboards through slides, and also give some users features such as commentary. This features is not a new feature, it has been released almost a year ago! but because many people still are not aware of such feature, I intended to write about it in a post. In this post you will learn;

- Enabling Power BI Power Point Integration
- Pin Live reports and dashboards elements in Power Point Slides

- Using Power Point and Power BI from storytelling with commentary feature

If you want to learn Power BI, read [Power BI online book; from Rookie to Rock Star](#).

Prerequisite

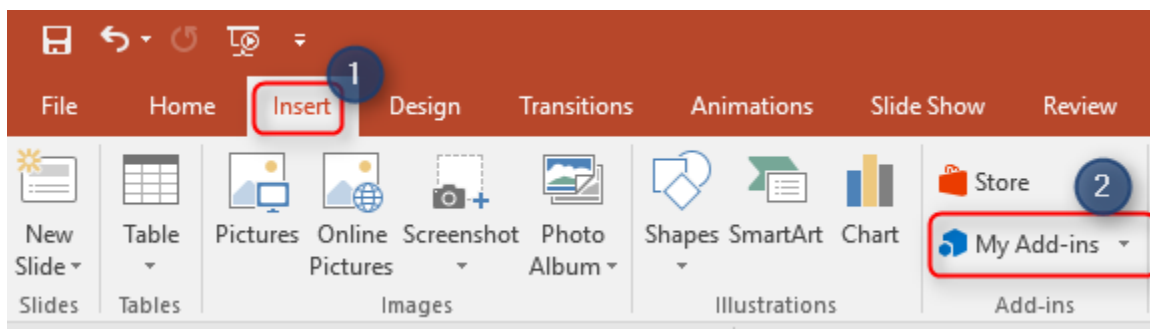
Dashboard and reports used in this example are sourced from [Power BI Desktop; The First Experience](#).

Power BI Tiles

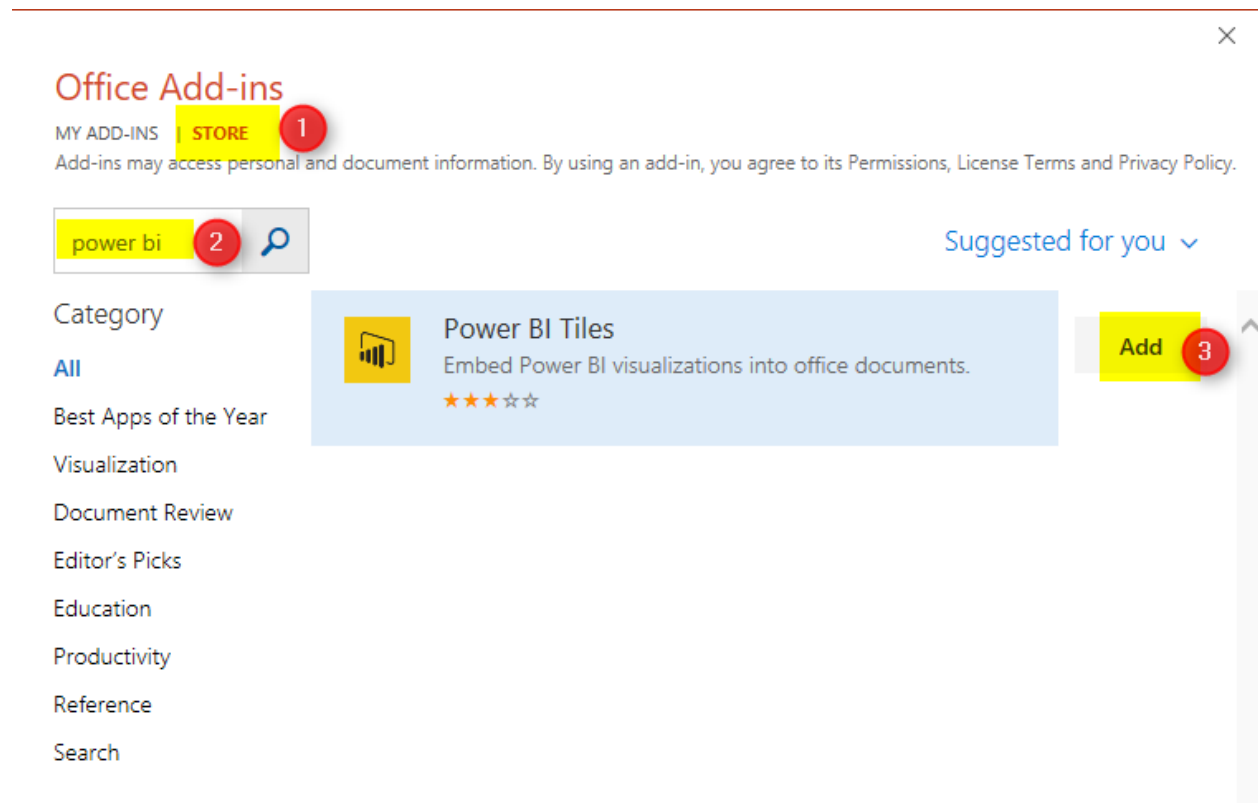
[Power BI Tiles](#) is an add-in for Excel, Power Point and few other Microsoft Office products version 2013 SP 1 or later. This component is developed by [DevScope](#). Please note that this is not Microsoft built-in feature. However it can be used easily with Microsoft Power BI and Power Point. This component enables Power BI reports and dashboard elements to be pinned in Power Point slides. Download this component and install it from there, or from your office project.

Installation

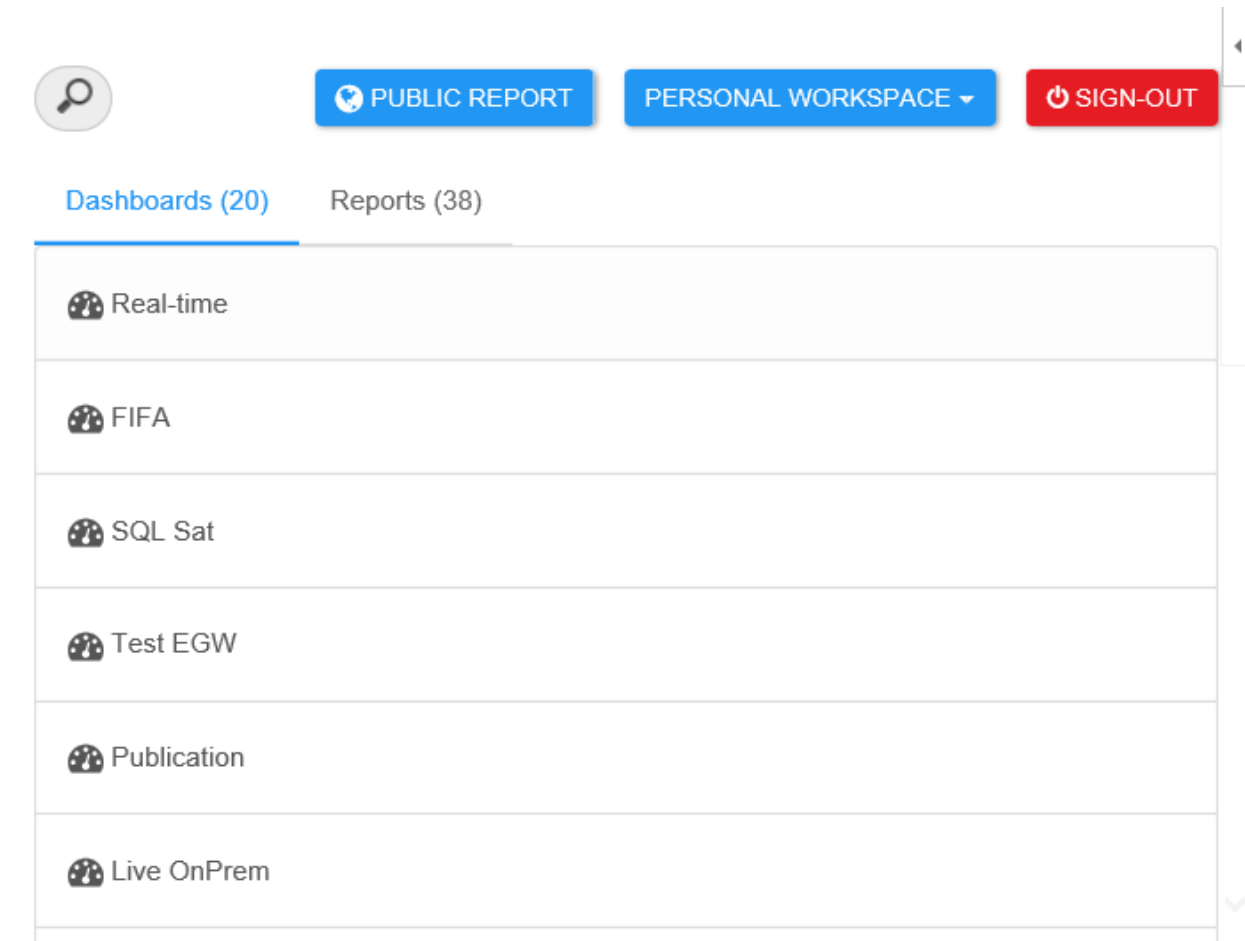
Open Power Point, create a blank slide deck, then insert a BLANK slide, and from Insert menu option, click on My Add-ins



From My Add-ins you might see Power BI Tiles if you have installed that before. otherwise go to Store tab, and search for Power BI

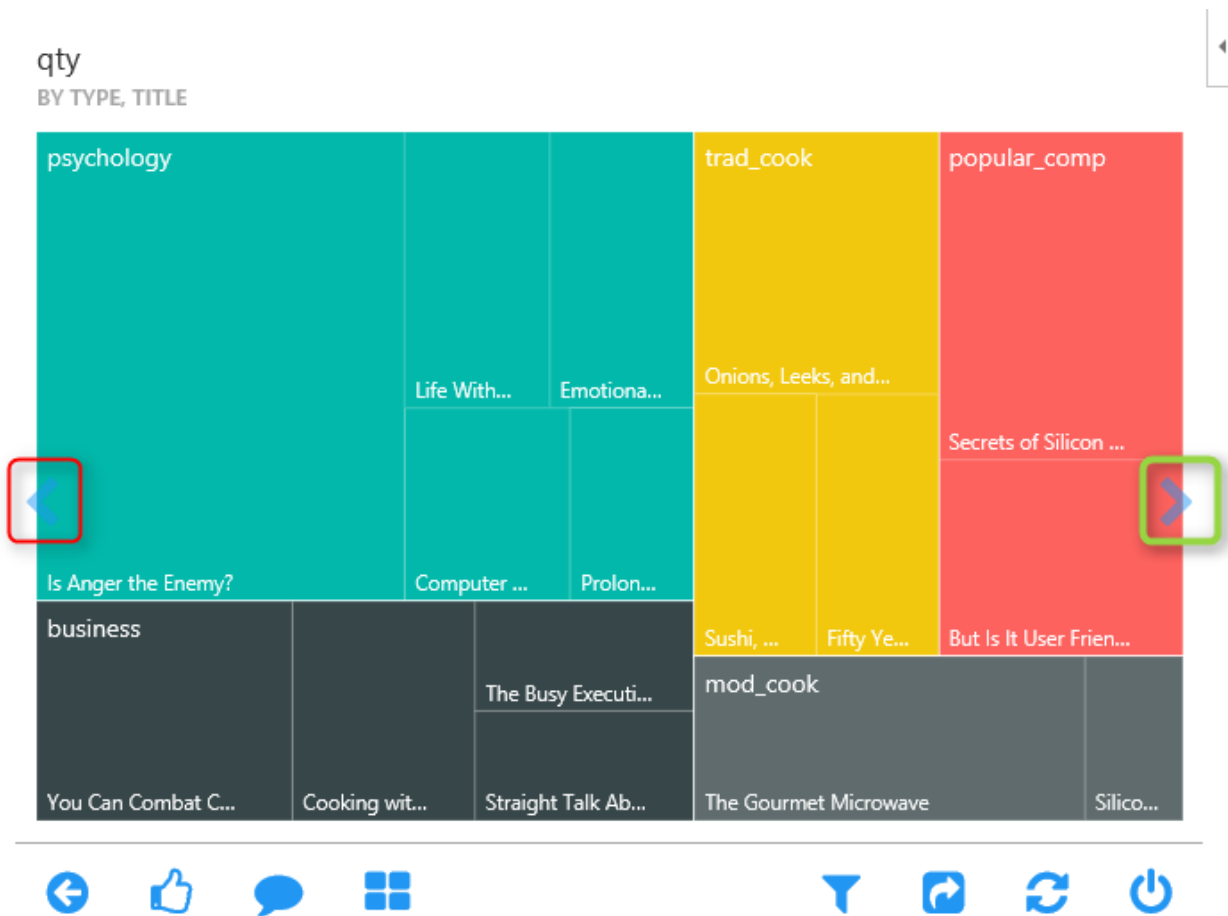


After adding this add-in you will be asked to enter Power BI account related to this, and also authorizing Power BI tiles to access reports and dashboards in your Power BI cloud storage. After authorization you will see a box including list of dashboards in your work space will appear.



Adding Power BI Tiles to Slides

This box is Power BI Tile added to your slide. You can select any of dashboards you want. for this example I use the dashboard which is related to Pubs Data Analysis Demo I've done before. After choosing a dashboard you will see one of its tiles appear here.



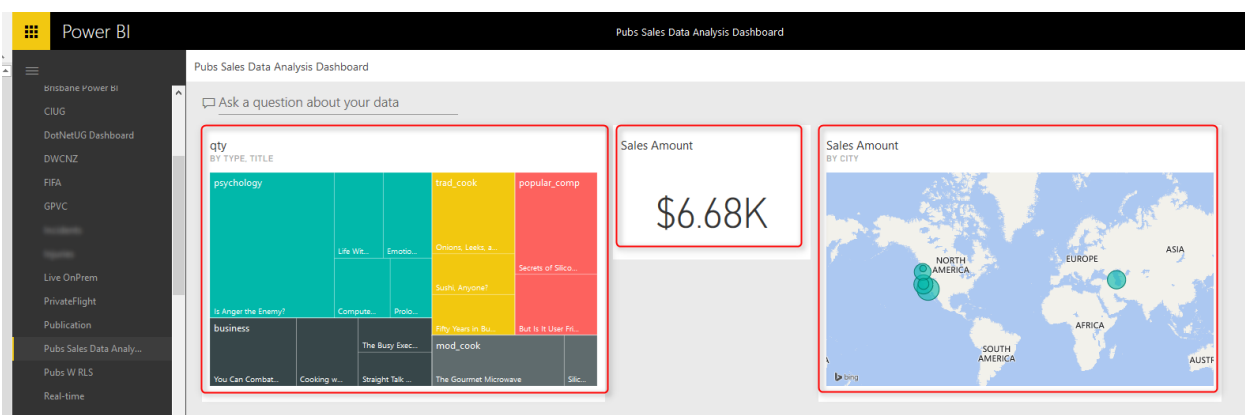
This is only one of the tiles in the existing dashboard, You can navigate between tiles by two navigator arrow buttons highlighted above.

Sales Amount

< \$6.68K >

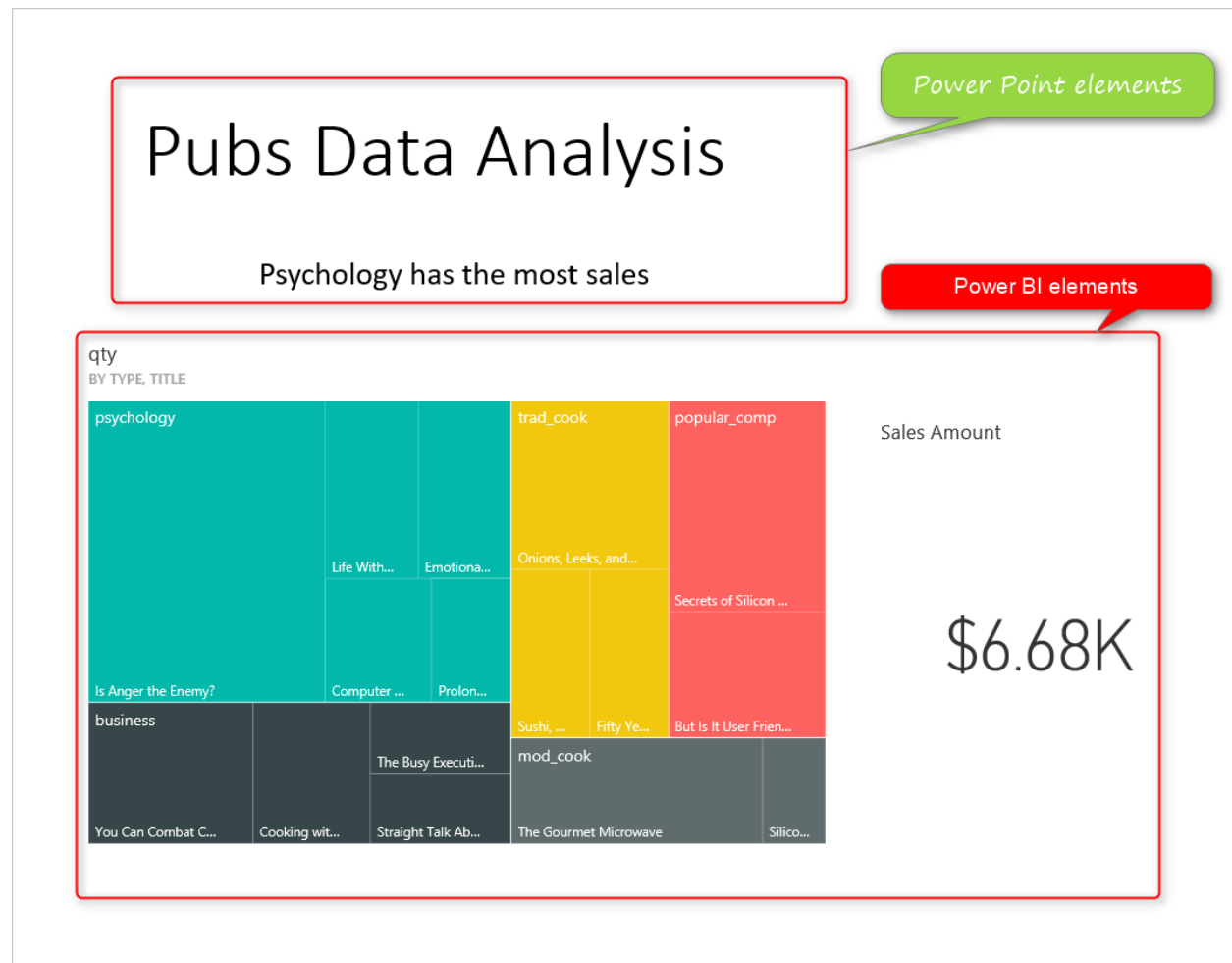


Here is the dashboard in full view in Power BI service. as you can see in Power BI Tiles add-in we can navigate between tiles individually.



Once you chose the tile, you can leave it there. It is already embedded into your Power BI slide. You can now add another Power BI Tile, again from My Add-ins and this time Power BI Tiles is there. Note that you can build similar dashboard here in Power Point with adding each item as a tile.

Storytelling with Power BI and Power Point

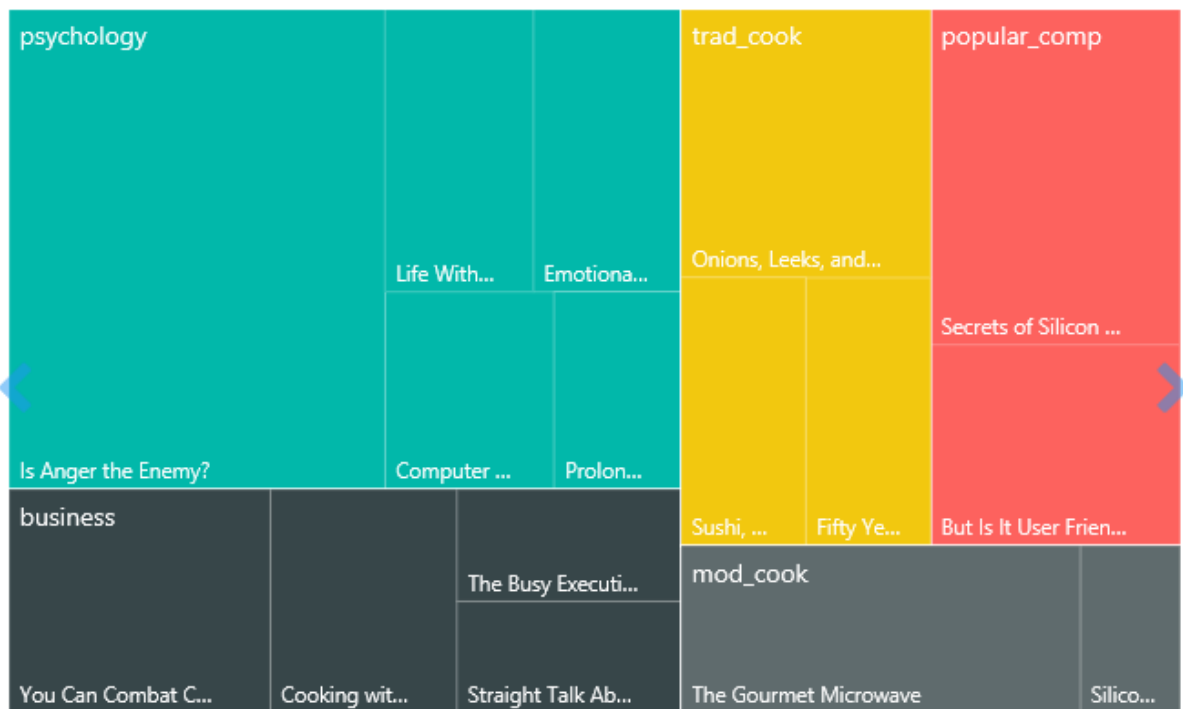


You can see here that I've added a title for this slide in Power Point, and also a Textbox with text "Psychology has the most sales". Yes, with this method you can add commentary to Power BI reports and dashboards, you can tell the story behind the data with integration of Power BI and Power Point. very simple feature but really useful.

Under each Power BI tile in Power Point there will be a link to Power BI

qty

BY TYPE, TITLE

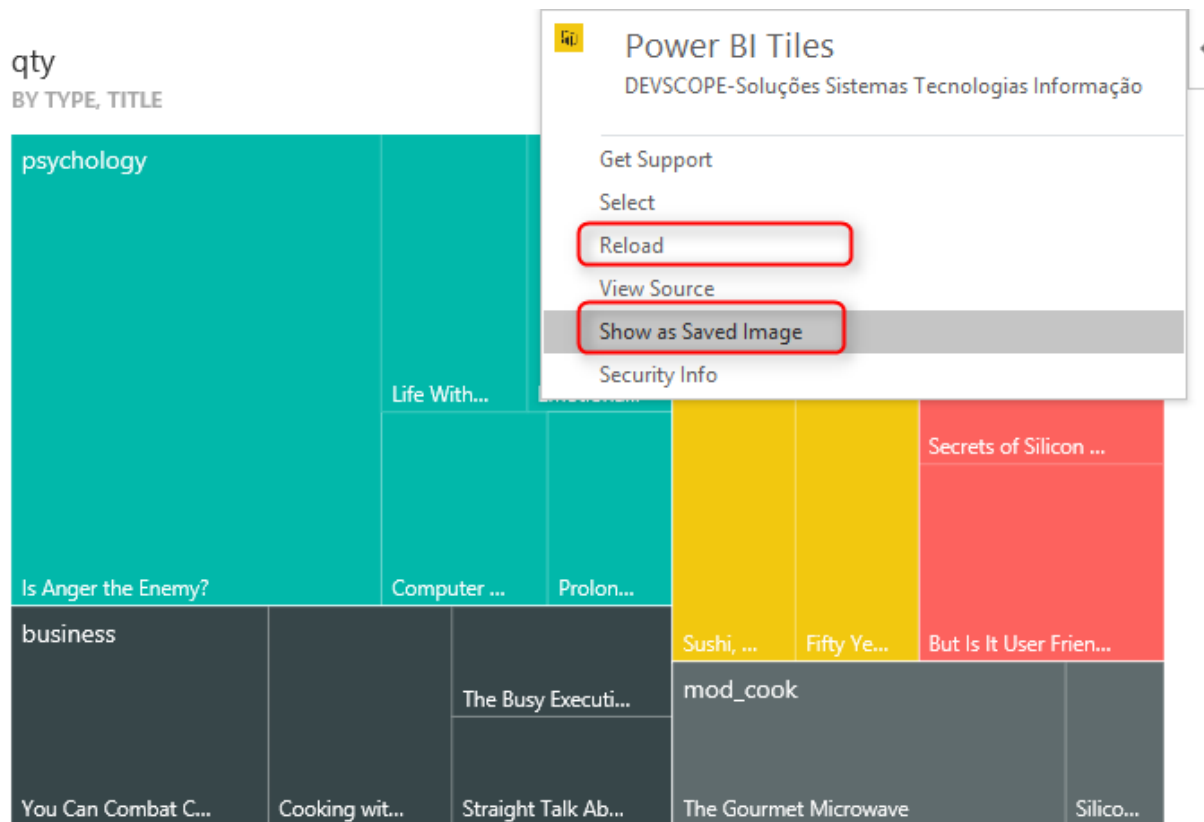


Open in PowerBI

When you click on this link, it will open that Power BI dashboard or report in a browser. This provides you the navigation from a high level story point to the detailed report in Power BI service.

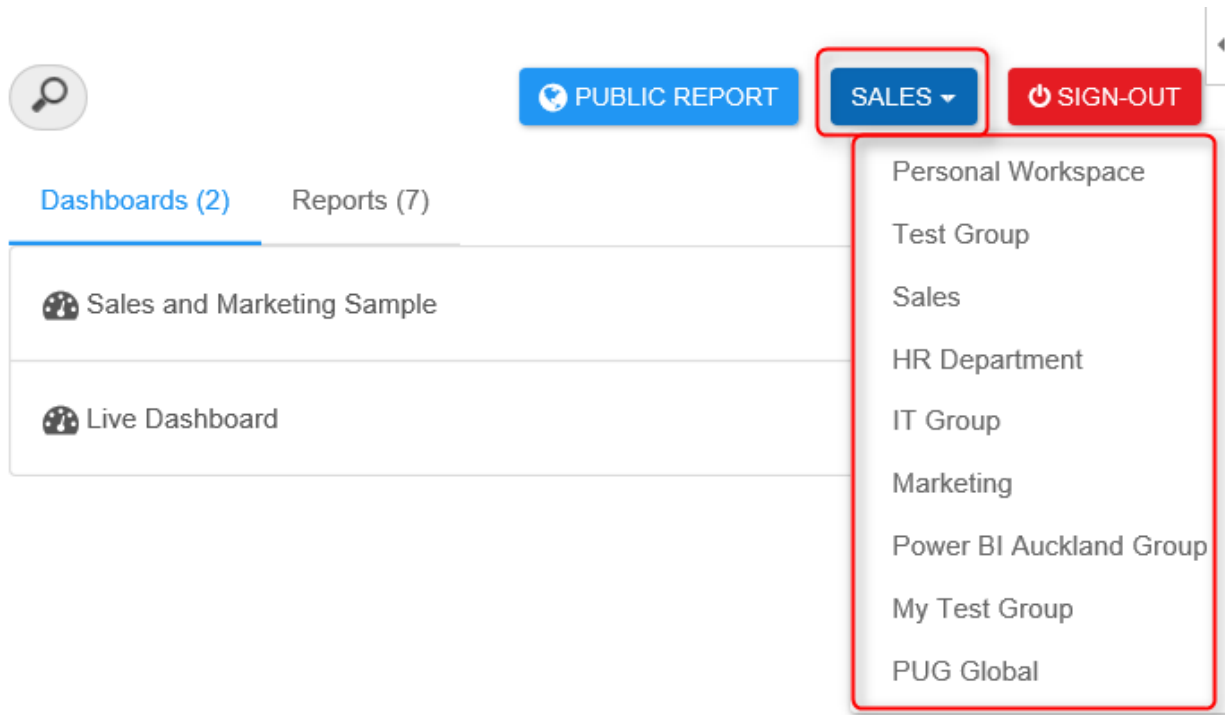
Images or Re-Loaded Tile Feature

By default Power BI Tiles are interactive tiles that can be refreshed and clicked, highlighted, and interactive. However if you want them as an image only tile, you can choose it from top right hand side of Power BI tile.



Working with other Work Spaces

You can navigate to all other work spaces that you are part of, and see all reports and dashboards under it. by default Power BI Tiles will look into "My Workspace" (named as Personal Workspace), but you can change it from the top side of Power BI Tile.



You can also bring a Power BI public report (a report that is [published to web](#)) into a tile

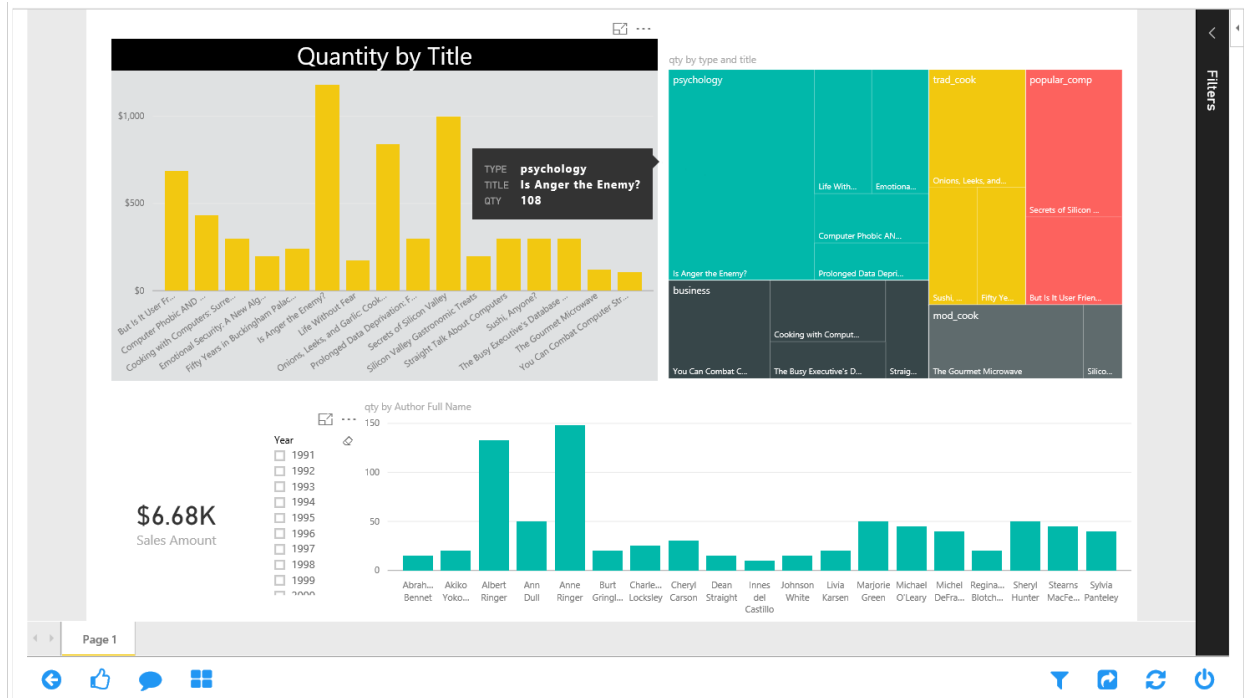
▼ Show Power BI Public Report

<https://app.powerbi.com/>
(Ex: <https://app.powerbi.com/view?r=eyJrljoiNGlzMjk1OD>)

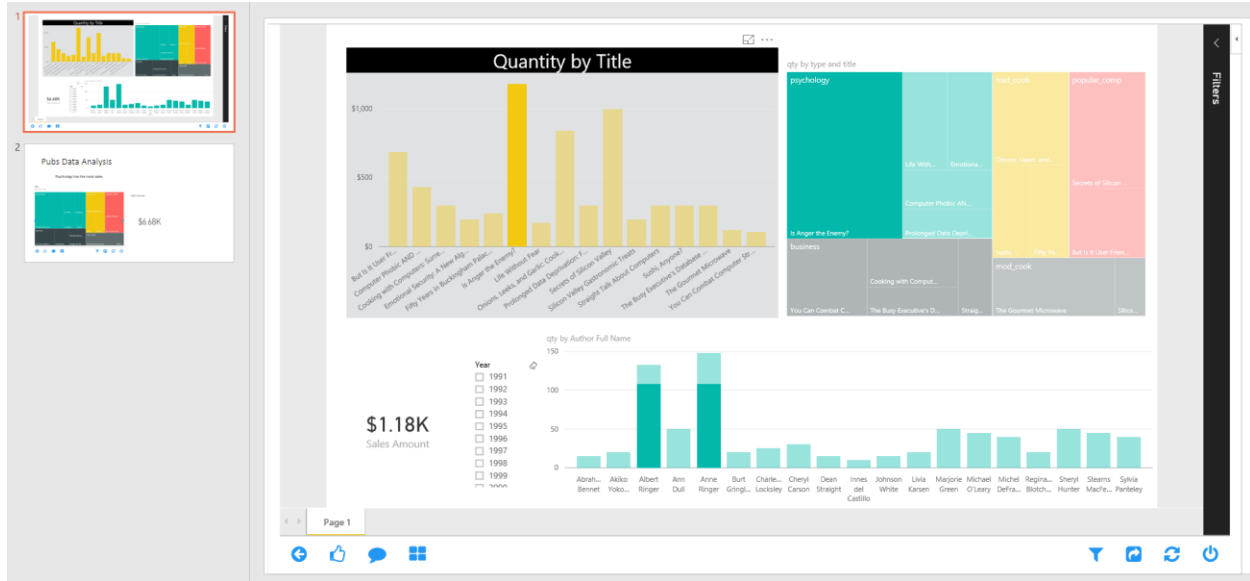
CLEARAPPLYCANCEL

Reports in Tile

If you choose dashboard, then you can pick one element of dashboard in each tile. but if you choose a report. For example “Pubs Analysis” report we’ve built in [another example](#), then you will see the whole report in tile, with all report pages, and filter pane.



The view of report is fully interactive here. For example if you hover on something or click on something you will see it responds like any other Power BI reports (but in Power Point slide)

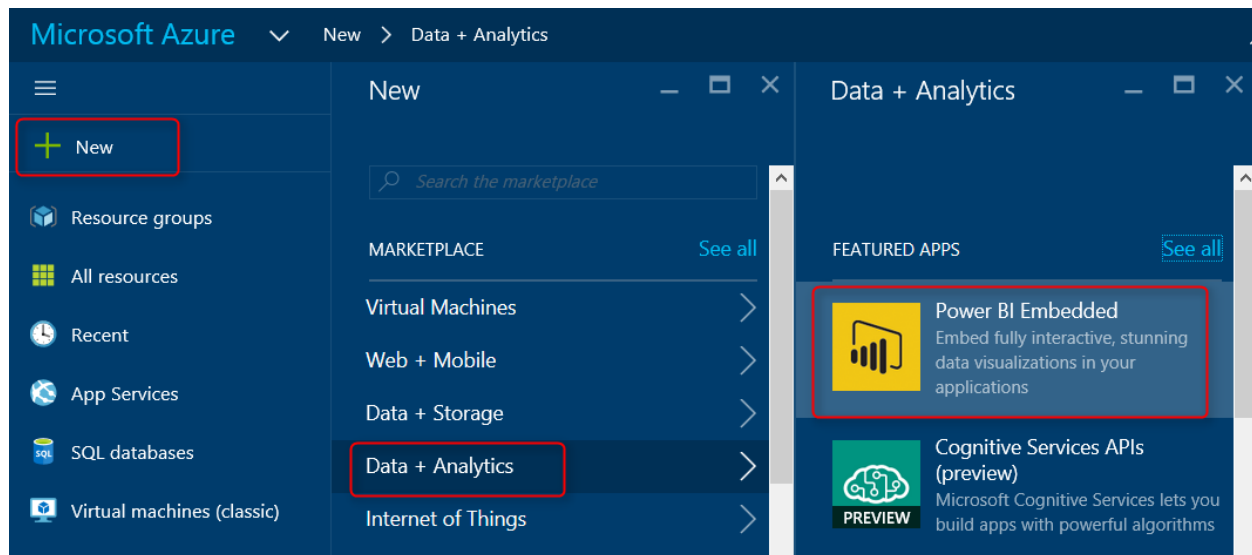


Summary

Power BI and Power Point integration helps to tell story behind the data much easier. You have presentation and story telling features of Power Point, plus interactive visual, refresh-able from cloud features of Power BI. Good news is that [Microsoft team is working on another integration feature](#) named **"Export to Power Point"** which will be released very soon I believe. I'll put together another post when that feature comes live.

Bring the Power into Your Application; Power BI Embedded

Published Date: August 5, 2016



Power BI which released at July 2015 targeted business analysts and power users. From that time Power BI team made massive enhancements, and with each enhancement the user market of Power BI expanded. Recently Power BI take a big step forward with bringing developers as users. Power BI Embedded is the Power BI reporting solution for application developers. In this post I'll explain how to get started with Power BI Embedded. If you want to learn more about Power BI read [Power BI online book; from Rookie to Rock Star](#).

What is Power BI Embedded?

Power BI Embedded is the reporting and analysis solution for mobile and web applications. Power BI Embedded is an Azure service that integrates Power BI solution into mobile and web applications. The report still has to be authored and created in Power BI Desktop. After creating the report it can be published into Power BI workspace in Azure, and using API Keys of Power BI workspace and embedding Power BI report frame into the web/mobile application it will be integrated into the application.

Key Advantages of Power BI Embedded

6. Integrating Power BI into web and mobile applications
7. Application users don't need Power BI account to view the report
8. Security integration with the application

There are other advantages of using Power BI embedded, but these three are the major two reasons why people will use Power BI embedded. One of them brings the Power of reporting and analysis of Power BI into an application. On the other hand if your application has some users, then they will be able to view Power BI reports without having Power BI account. Security then can be applied easily through workspaces in Power BI Embedded.

Prerequisites For the Sample Report

Power BI embedded is an Azure service, so the very first thing you need in Azure subscription. If you don't have it you can apply for [free trial](#) version.

Also for this example I will show you a Power BI solution with connection to sample Azure SQL Database AdventureWorksLT. If you don't know how to set it up, follow the [instructions in this post](#) to set up AdventureWorksLT Azure SQL DB.

Getting Started with Power BI Embedded

For this example I will use a Live/DirectQuery connection to Azure SQL DB AdventureWorksLT. The main reason is that Power BI embedded right now at the time of writing this post doesn't support scheduled refresh, I believe this feature will be available very soon. So let's create a sample Power BI report in Power BI Desktop. If you don't know what Power BI Desktop is; read this blog post: [Get Started with Power BI Desktop](#).

Open Power BI Desktop, Start with Get Data, and Choose SQL Server. Make sure you have checked Live Connection;

SQL Server Database

Import data from a SQL Server database.

Server

rezasql.database.windows.net

Database (optional)

AdventureWorksLT

☐ Import

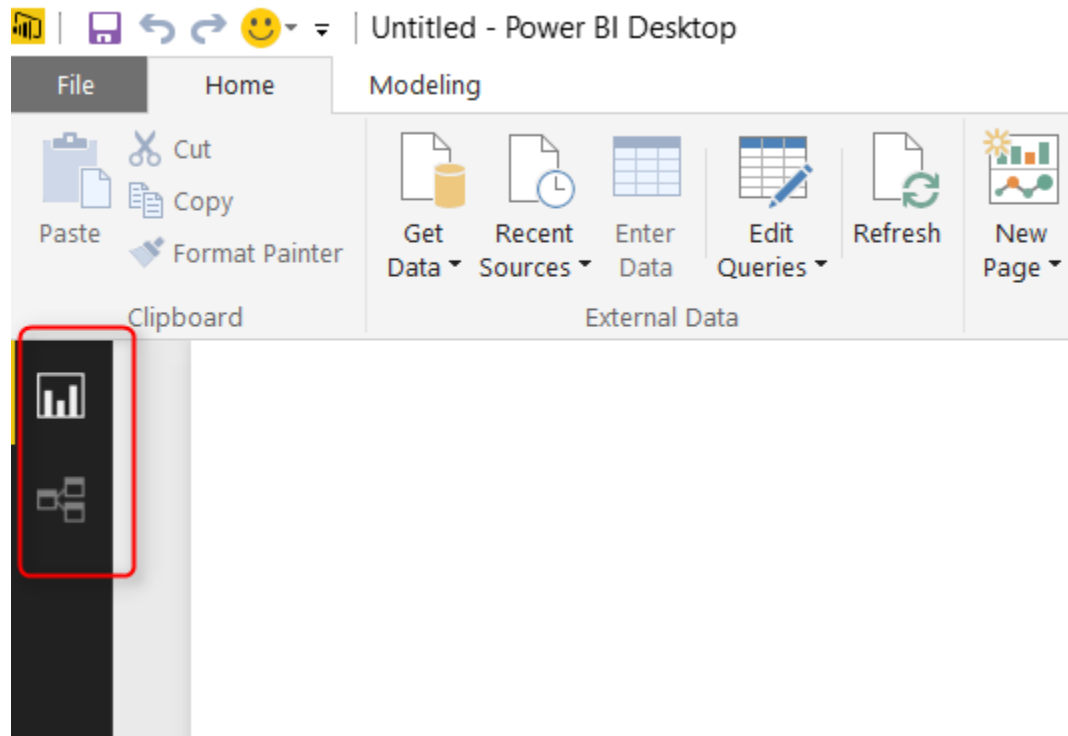
☒ DirectQuery

▸ Advanced options

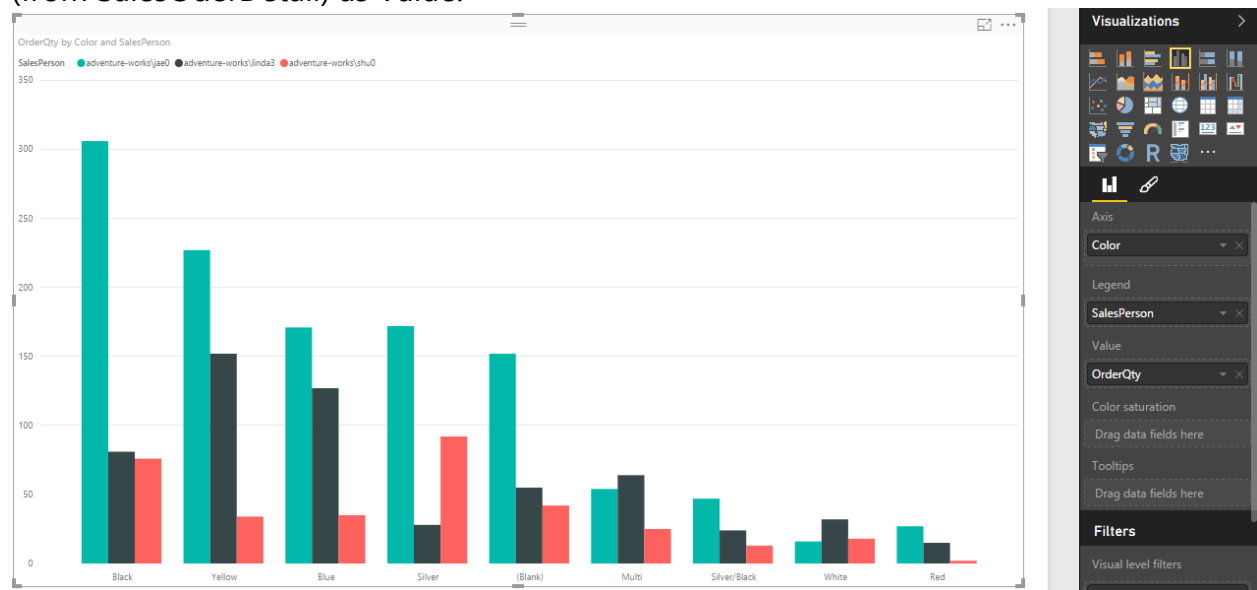
After entering credentials for connecting to Azure SQL DB you should be able to see list of tables. Note that you might need to set up firewall in Azure SQL server to allow your computer IP to pass, if you don't know how to set it up. read my [blog post](#) about connecting to Azure SQL DB from Power BI. You only need to read the set up firewall section of it.

From list of tables, choose these: Customer, Product, ProductCategory, ProductModel, SalesOrderDetail, and SalesOrderHeader. and click on Load.

After load you will notice that there is no Data tab in Power BI, and the reason is that we've used DirectQuery/Live connection. with this option we won't be able to use DAX or calculations from it in Data tab.



Now let's create a very simple chart, the chart is a clustered column chart with Color (from Product table) as Axis, and SalesPerson (from Customer) as Legend, and OrderQty (from SalesOrderDetail) as Value.

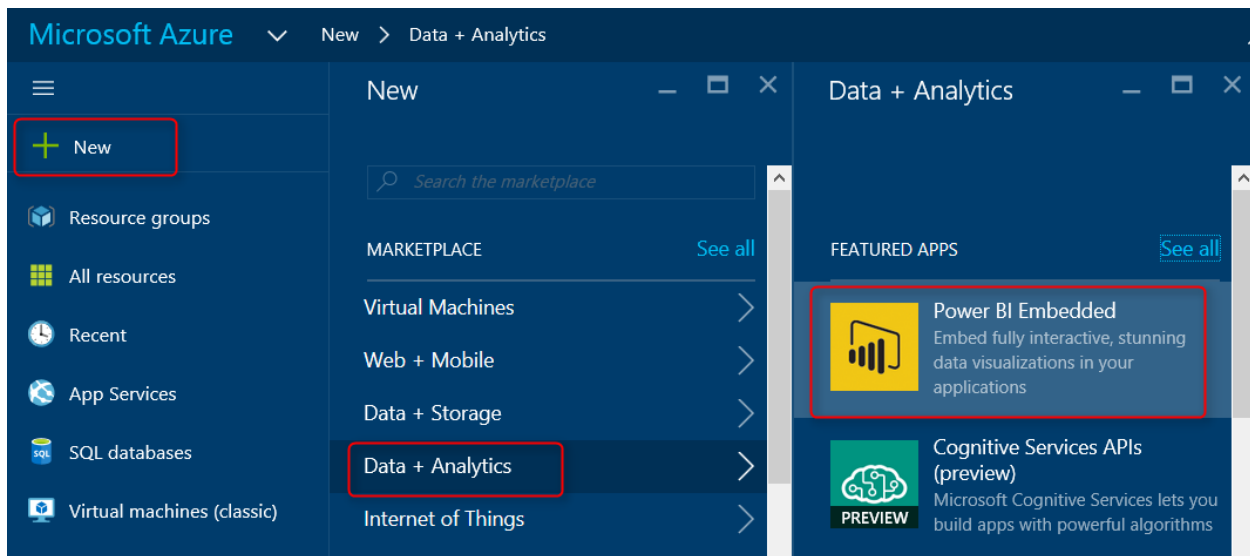


Now save the report as SampleReport, or any other names you like. We will publish this report into a Power BI workspace in next few steps.

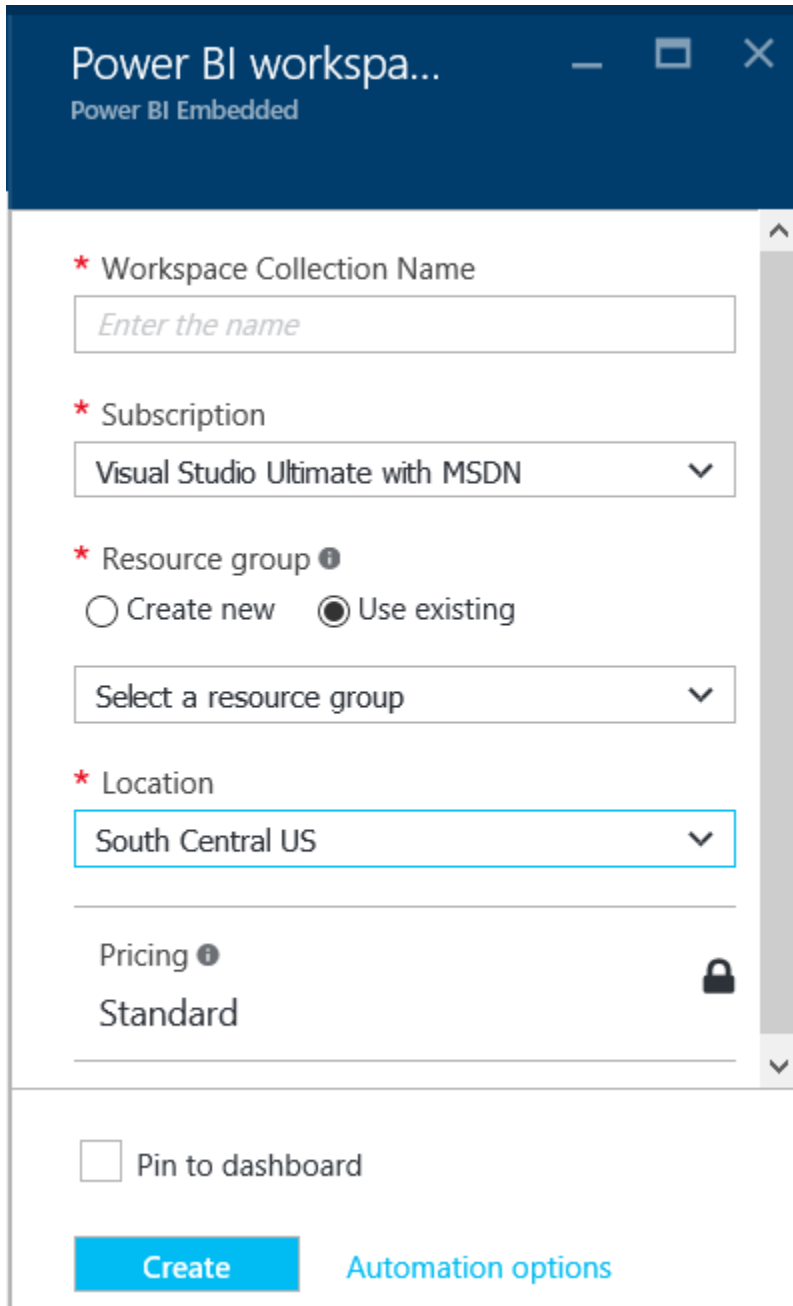
Power BI Workspace Collection

Power BI reports can not be published from Power BI Desktop for using with Power BI Embedded. In fact for using Power BI Embedded, reports should be published through the application itself. and it would be published to a container or folder under your Azure subscription. This container or folder is Power BI Workspace Collection.

To create the workspace collection, go to Azure portal (<https://portal.azure.com>), and create a New Power BI Embedded under Data and Analytics section.



Now set a name for the Power BI workspace collection, and either use an existing resource group, or create a new one for it.



The screenshot shows a 'Power BI workspace...' dialog box with a dark blue header. Below the header, there are several fields for configuration. The 'Workspace Collection Name' field has a placeholder 'Enter the name'. The 'Subscription' dropdown is set to 'Visual Studio Ultimate with MSDN'. The 'Resource group' section has radio buttons for 'Create new' and 'Use existing', with 'Use existing' selected. Below this is a dropdown for 'Select a resource group'. The 'Location' dropdown is set to 'South Central US'. A 'Pricing' section shows 'Standard' with a lock icon. At the bottom, there is a 'Pin to dashboard' checkbox and a 'Create' button. A link for 'Automation options' is also present.

Power BI workspace...
Power BI Embedded

* Workspace Collection Name

* Subscription
Visual Studio Ultimate with MSDN

* Resource group ⓘ
☐ Create new ☒ Use existing
Select a resource group

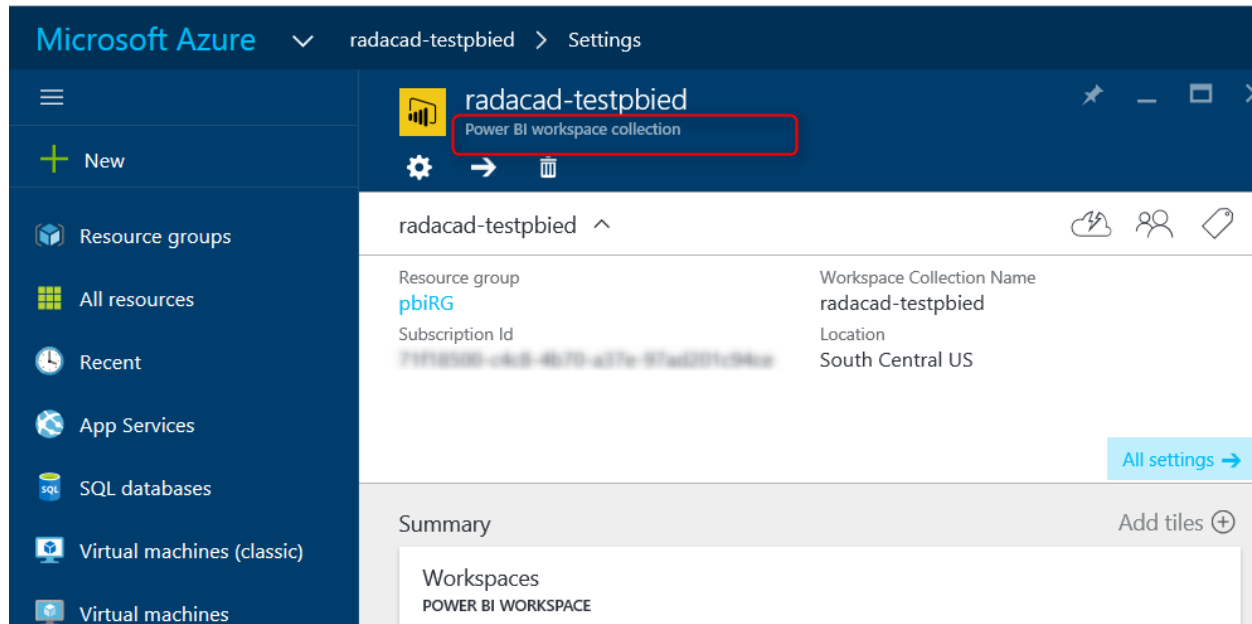
* Location
South Central US

Pricing ⓘ
Standard

☐ Pin to dashboard

Create Automation options

As you can see creating the workspace collection itself is simple and easy. After creation of Power BI embedded workspace collection, you will see the workspace collection blade opening as below;

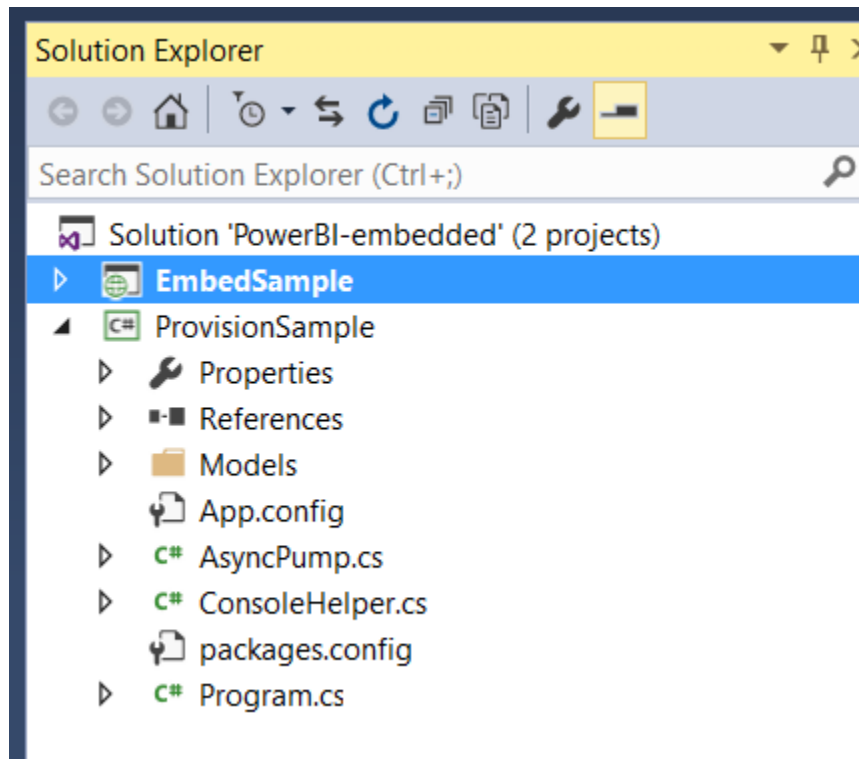


Under each workspace collection there can be one or more workspaces. like folders that has sub folders. Power BI reports will be published to workspaces. I will explain in future posts how you can leverage using multiple workspaces, for now let's keep it simple and work with only one workspace.

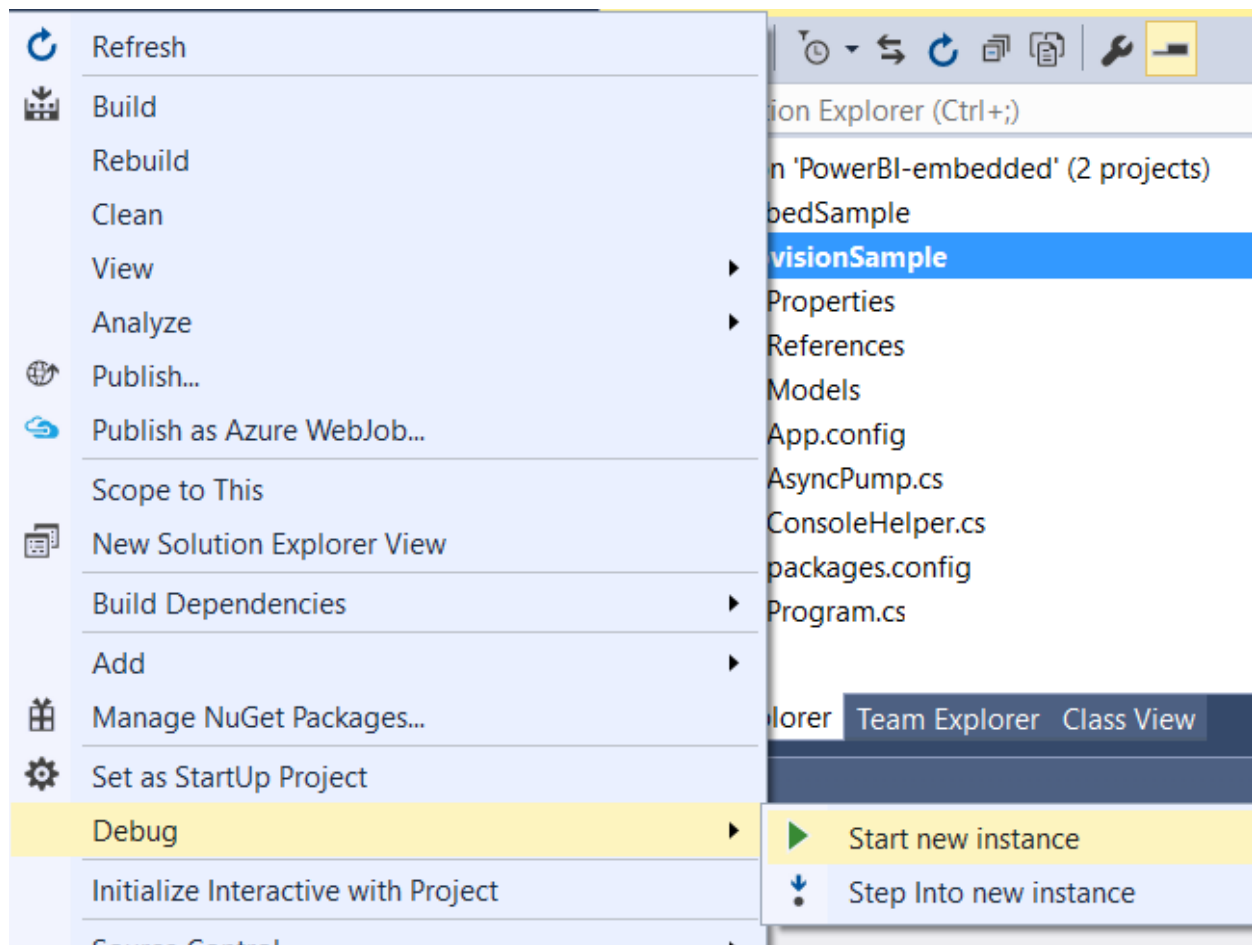
Sample .NET Projects

At this moment there is no graphical user interface option in Azure portal to create a workspace, so it needs to be created through an application. because this is the getting started sample and I don't want to dive into coding here, I'll be using sample Microsoft .NET projects from [here](#).


Download the code above. Note that you need Visual Studio installed to open this project. There are two projects under Github all under same solution.



For creating a workspace use ProvisionSample project, and right click on it, and from Debug, choose start new instance



This project has the code for creating workspace, and also publishing the report.

 file:///C:/Users/Reza/SkyDrive/Blog/embedded/power-bi-embedded-integrate-report-into-web-app-master/Prov...

What do you want to do?

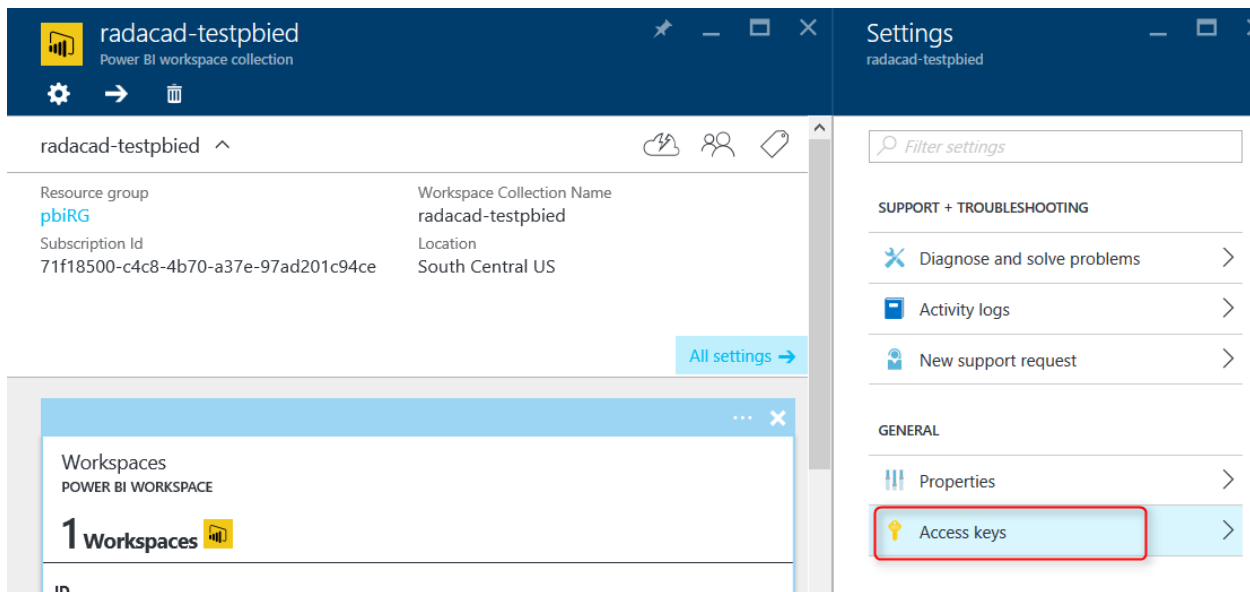
- ```
=====
```
1. Provision a new workspace collection
  2. Get workspace collection metadata
  3. Retrieve a workspace collection's API keys
  4. Get list of workspaces within a collection
  5. Provision a new workspace in an existing workspace collection
  6. Import PBIX Desktop file into an existing workspace
  7. Update connection string info for an existing dataset
  8. Retrieve a list of Datasets published to a workspace
  9. Delete a published dataset from a workspace



\*\* Note that this project is not a utility for deploying Power BI reports, this part can be embedded into your .NET application. This is just an example code for doing the deployment.

## Create Workspace

For creating a workspace choose option 5. it will asks some information about your existing Azure subscription key, and work space collection and will create the workspace for you. It will also asks for your Power BI workspace collection Access Keys, which you can find them under Power BI embedded workspace collection blade. Be careful that don't share these keys to others. this would be the access to Power BI embedded.



After creating the workspace successfully you should be able to see it under workspace collections in Azure Power BI embedded blade.

## Publish Report

To publish report again there is no graphical interface. and you can use same ProvisionSample project for publishing. (Number 6 in the list of options to choose);

```
file:///C:/Users/Reza/SkyDrive/Blog/embedded/power-bi-embedded-integrate-report-into-web-app-master/Prov...
=====
```

1. Provision a new workspace collection
2. Get workspace collection metadata
3. Retrieve a workspace collection's API keys
4. Get list of workspaces within a collection
5. Provision a new workspace in an existing workspace collection
6. Import PBIX Desktop file into an existing workspace
7. Update connection string info for an existing dataset
8. Retrieve a list of Datasets published to a workspace
9. Delete a published dataset from a workspace

```
Workspace Collection Name:radacad-testpbied
Workspace ID:8e075414-cc86-4473-ba98-418c76f776af
Dataset Name:AdventureWorks Report
File Path:C:\Users\Reza\SkyDrive\Blog\embedded\SampleReport.pbix
Access Key: mm6ePhv6y9y8ODU/1wPpR7LsVpTQVjI10v+27nd14wnc18y0mCgm8d7p8e+7kx00w0+8-
2kx9yc7dMLCZck8vhg==
Checking import state... Publishing
Checking import state... Publishing
Checking import state... Succeeded
Import: fff147dd-174f-43d6-89d3-b20276cb4043
What do you want to do?
```

For publishing the report you need to provide the access key and name of workspace collection, the id for workspace, then name for data set (the name that you want this to be stored), the full path of \*.pbix file. Now you see that Power BI report is published.

## Update Connection String of the Report

Because we used an Azure SQL DB as the source of connection, we have to update the connection string after deployment. Again you can use same ProvisionSample project with option 7 this time.

```
file:///C:/Users/Reza/SkyDrive/Blog/embedded/power-bi-embedded-integrate-report-into-web-app-master/Prov... What do you want to do?
=====
1. Provision a new workspace collection
2. Get workspace collection metadata
3. Retrieve a workspace collection's API keys
4. Get list of workspaces within a collection
5. Provision a new workspace in an existing workspace collection
6. Import PBIX Desktop file into an existing workspace
7. Update connection string info for an existing dataset
8. Retrieve a list of Datasets published to a workspace
9. Delete a published dataset from a workspace

Workspace Collection Name:radacad-testpbied

Workspace ID:b0875d1d-cc86-4473-ba98-418c76f776af

Username: reza

Password: *****

Connection String (enter to skip):

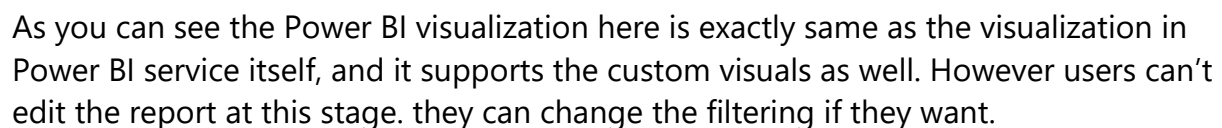
Access Key: b0875d1d-cc86-4473-ba98-418c76f776af

Connection information updated successfully.
```

## Testing the Embedded Report

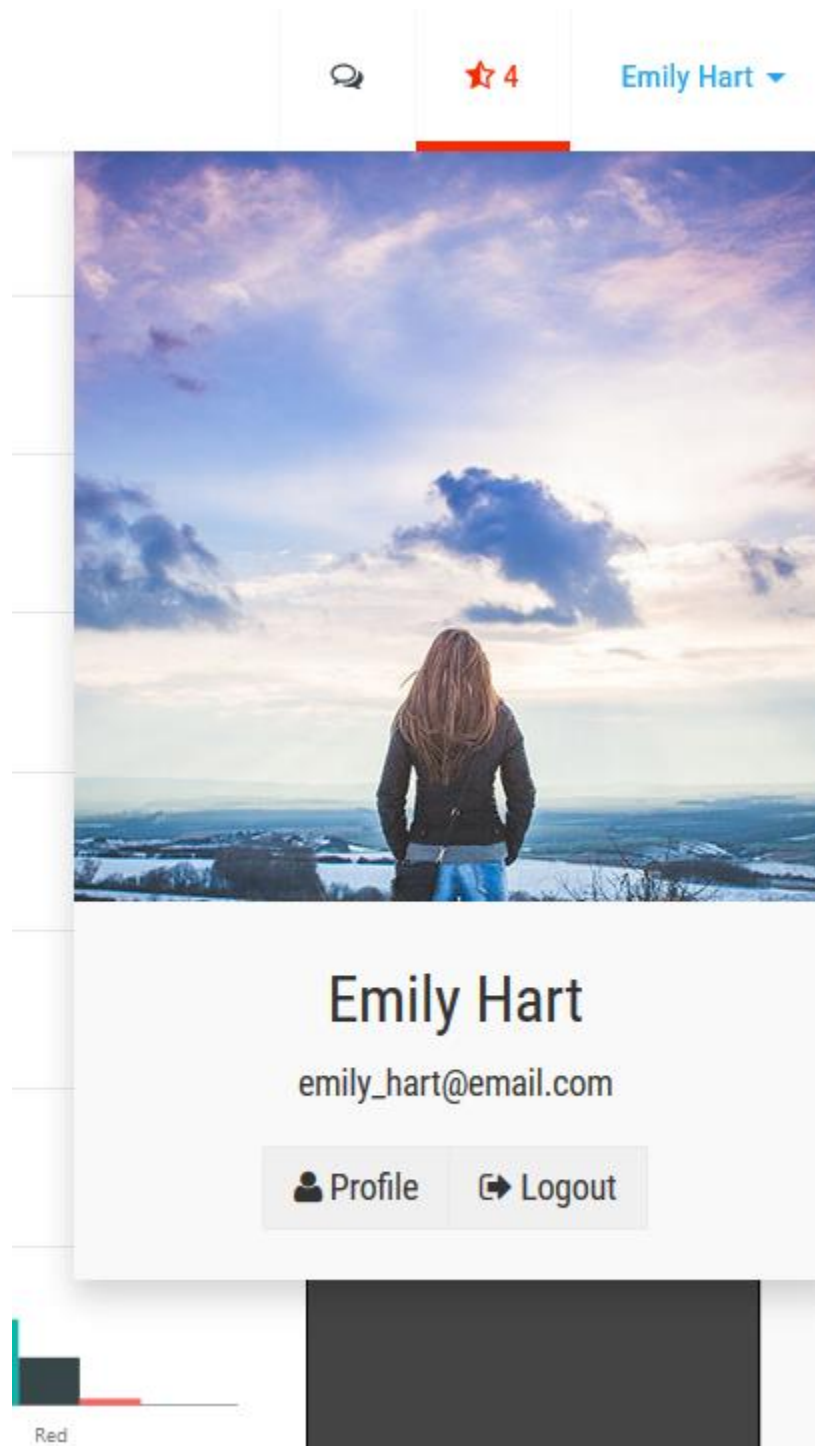
After updating the connection information, now we can test the report. There is another project under this solution named EmbedSample. This project is an ASP.NET MVC solution which already has the Power BI embedded code in the application (I will discuss about details of that code in future posts). All you need to do is to set up the Web.Config of this project to use your workspace collection. Open Web.config and configure lines below;

After entering the information for Access Key and workspace collection and workspace id, then you can run this project and in the web application opened, under reports you will see the AdventureWorks Report which will show you the live data as result of DirectQuery connection.



## Security Thoughts?

One of the big advantages of Power BI Embedded over [Power BI publish](#) to web is that you can set up a security in embedded, while the publish to web is available for everyone. Workspaces can be a way to set up the security. I will allocate a separate post to this, because this post is already more than long enough. You can choose which users of your application can access which report pages. right now in the sample application the user who logged in is an application user which doesn't have any Power BI account. However the security for this user can be set specifically.



## Pricing

Because with Power BI Embedded users won't have the Power BI account, so the pricing plan changed from user based to session based. there is a current pricing plan up to 1st of September based on renders, which will be changed to Per Session. free option

allows up to 100 sessions per month, and with standard option each session costs only 5 cents. Each session will be the user/browser that has report opened in the application.

## Limitations of Power BI Embedded

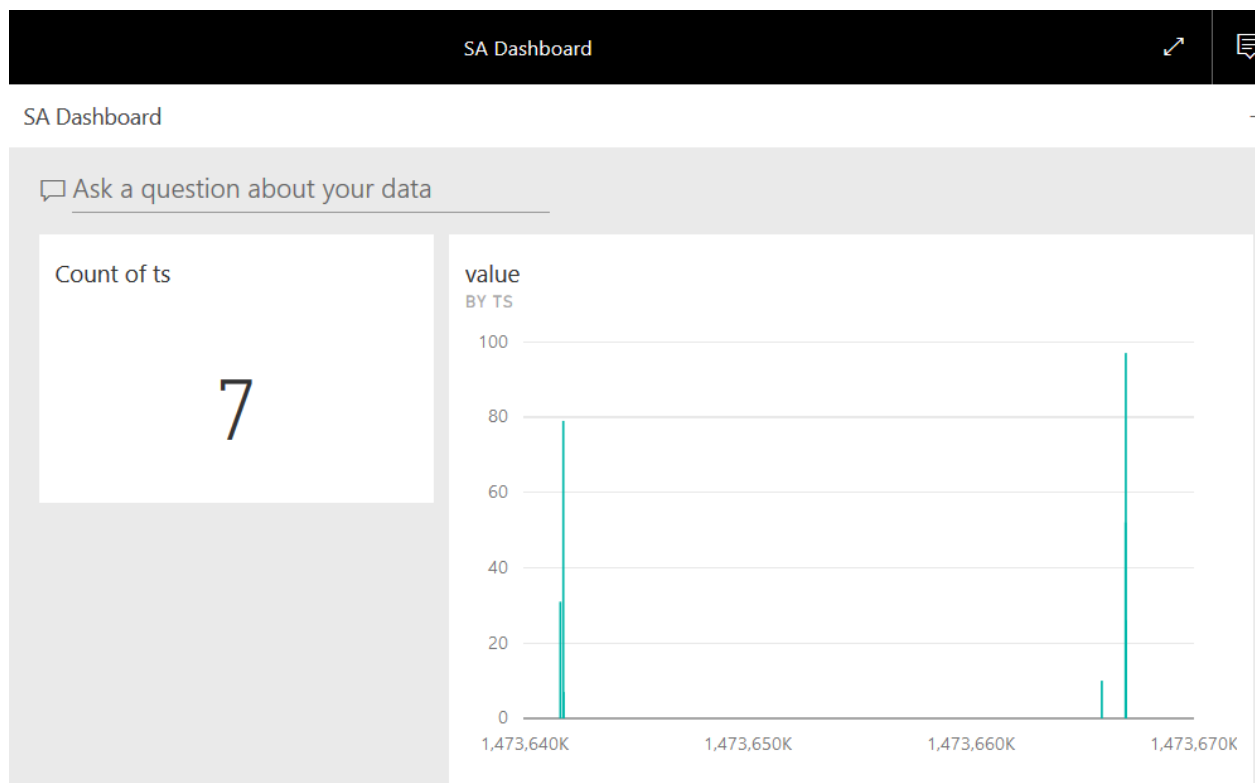
Power BI Embedded is a new feature, and as expected has some limitations for now. but most of these limitations will be removed in near future I believe. here are biggest limitations so far;

10. Source is limited to Azure SQL DB and Azure SQL Data Warehouse through DirectQuery and Live Connection
11. Import Data is not supporting scheduled Refresh; That means, you CAN connect to other data sources such as Excel and import the data into Power BI. build the report, publish it to Power BI embedded. you can see the report without any issue in the application. However there is no support for scheduled refresh yet.
12. On premises data sources is not supported.
13. Because of limitation to Live connection DAX and Data tab is not available.

Stay tuned for next posts about Power BI embedded where I will explain more about code behind of Power BI embedded, Row Level Security with Embedded, and using workspaces more in details.

# Stream Analytics and Power BI Join Forces to Real-time Dashboard

Published Date: September 12, 2016



Few weeks ago, I've written how [REST API can be used for real-time dashboard in Power BI](#). Another method for real-time dashboard is using Azure Stream Analytics. This can be combined with Azure IOT and Event Hub to get data from IOT devices. In this post I'll explain to you all steps necessary to push output from a .NET application through Azure Event Hub, and Azure Stream Analytics to Power BI. You will see how to create a real-time dashboard with this method. If you want to learn more about Power BI; read [Power BI online book; from Rookie to Rock Star](#).

## Prerequisite

You need an Azure account for running sample of this post. Azure account can be created with a free trial 1 month of credit. To create an Azure Account go to [Portal.Azure.com](https://portal.azure.com).



You also need [Visual Studio 2015 professional or Community](#), because a .NET application will be used as the source of the data.

## Architecture

Diagram below shows the architecture of this sample.



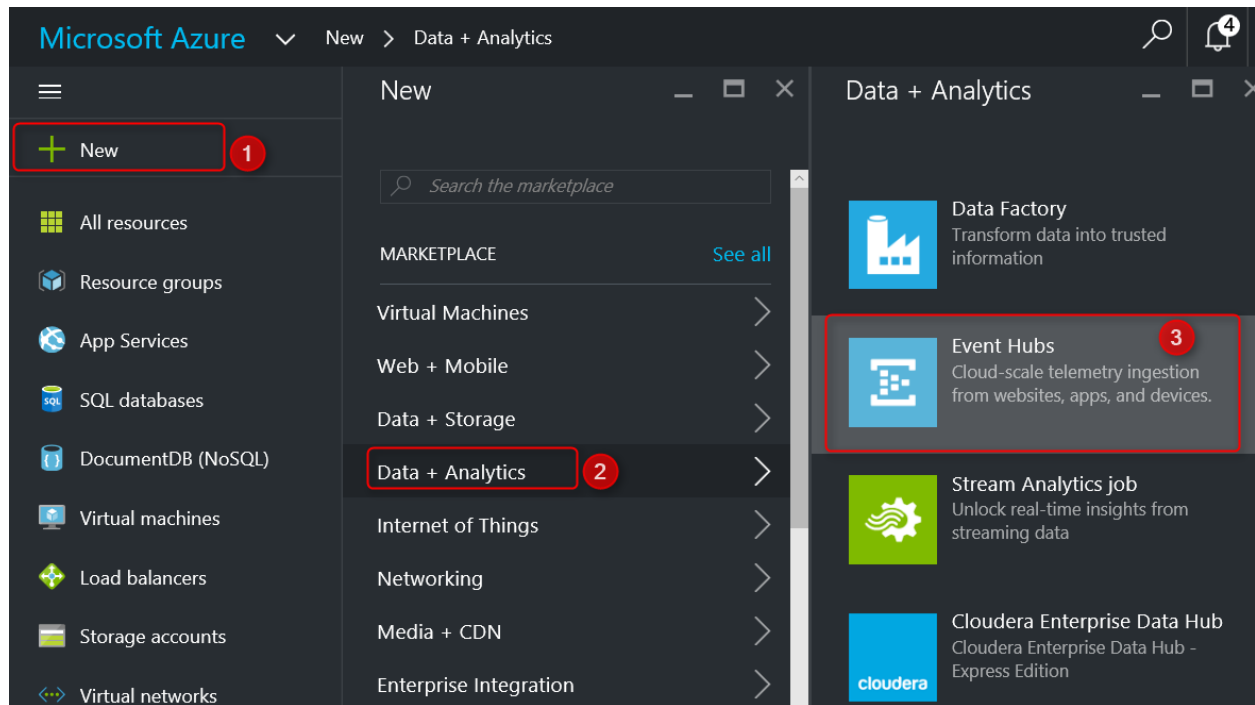
IoT Devices or Applications can pass their data to Azure Event Hub, and Azure Event hub can be used as an input to Azure Stream Analytics (which is a data streaming Azure service). Then Azure stream analytics can pass the data from input based on queries to outputs. If Power BI be used as an output then a dataset in Power BI will be generated that can be used for real-time dashboard.

As a result anytime a new data point from application or IoT device comes through Event hubs, and then Stream Analytics, Power BI dashboard will automatically update with new information.

## Azure Event Hub

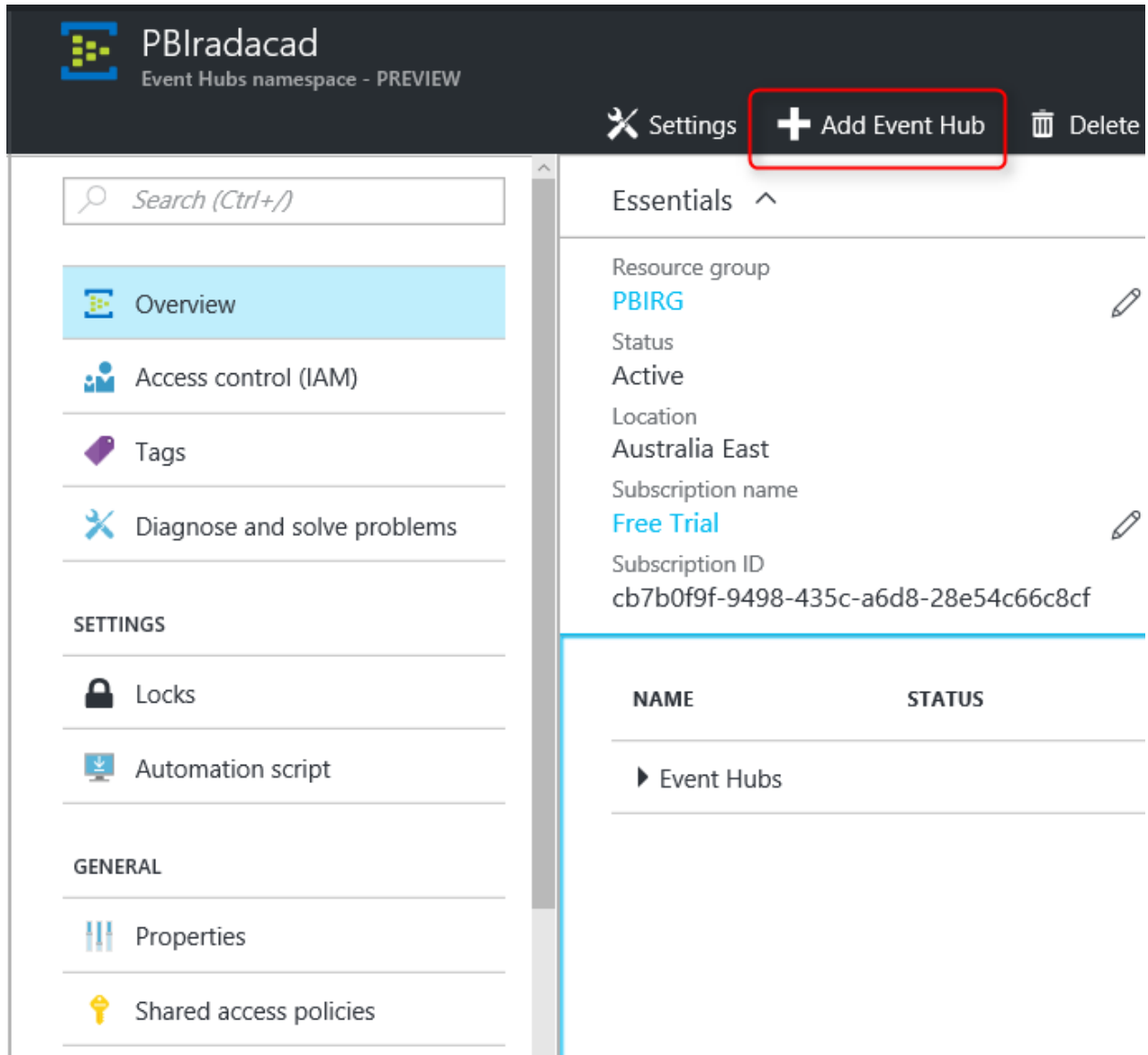
Azure Event Hub is the entry point of the data from applications. Start by creating an Event Hub by going to [Portal.Azure.com](#)

In Azure portal, create a New Event Hub from New, and then Data + Analytics



Event Hub needs a Service Bus, this first step here creates the namespace for service bus. Set a namespace in the new blade. for example PBIRadacad. choose a pricing tier. for this example you can choose Basic. Select a resource group (or create if you don't have one), and set a location.

After creating you will be redirected to Event Hubs Namespace Blade. Your Event Hubs namespace can include multiple Event Hubs. Create your first Event Hub here, by [Adding an Event Hub](#).



The screenshot shows the Azure Event Hubs console. The top header bar is dark blue with the 'PBIRadacad' logo and text 'Event Hubs namespace - PREVIEW'. On the right of the header are three buttons: 'Settings' (wrench icon), '+ Add Event Hub' (highlighted with a red box), and 'Delete' (trash icon). The left sidebar contains a search bar and a list of navigation items: 'Overview' (selected), 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'SETTINGS' (with sub-items 'Locks' and 'Automation script'), and 'GENERAL' (with sub-items 'Properties' and 'Shared access policies'). The main content area is titled 'Essentials' and displays the following details:

- Resource group: **PBIRG** (with edit icon)
- Status: **Active**
- Location: **Australia East**
- Subscription name: **Free Trial** (with edit icon)
- Subscription ID: **cb7b0f9f-9498-435c-a6d8-28e54c66c8cf**

Below these details is a table with two columns: 'NAME' and 'STATUS'. It contains one entry: 'Event Hubs' with a right-pointing arrow icon.

For the new Event Hub just set a name. like pbi for example. and leave other settings as is. You can change these settings if you want to change partitions, and number of days to retention.

## Create Event Hub

PBIRadacad - PREVIEW

\*

Name

pbi

✓

Partition Count ⓘ

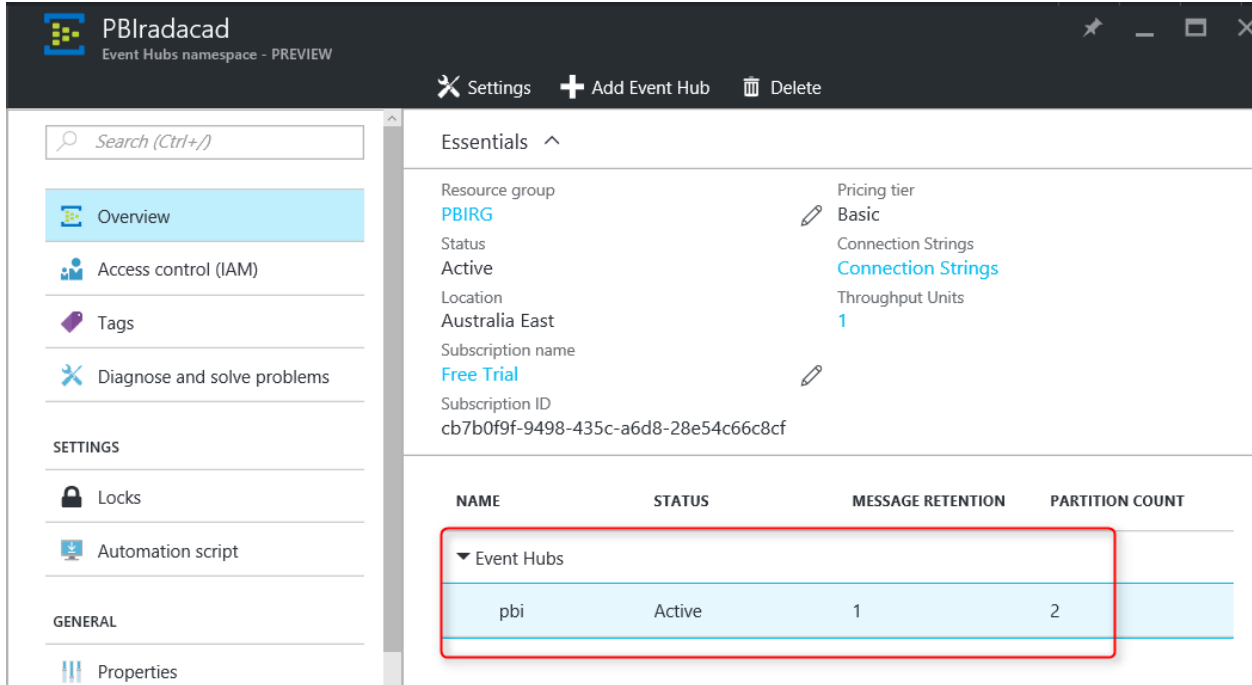
2

Message Retention ⓘ

1

Create

After creation the Event Hub should be listed under PBIRadacad Event Hubs (or whatever name you set for the namespace).



The screenshot shows the Azure Event Hubs console for a resource group named 'PBIRG'. The left sidebar contains navigation links for Overview, Access control (IAM), Tags, Diagnose and solve problems, Locks, Automation script, and Properties. The main content area displays the 'Essentials' section with various configuration details. A table at the bottom lists the Event Hubs, with one entry 'pbi' highlighted and enclosed in a red box.

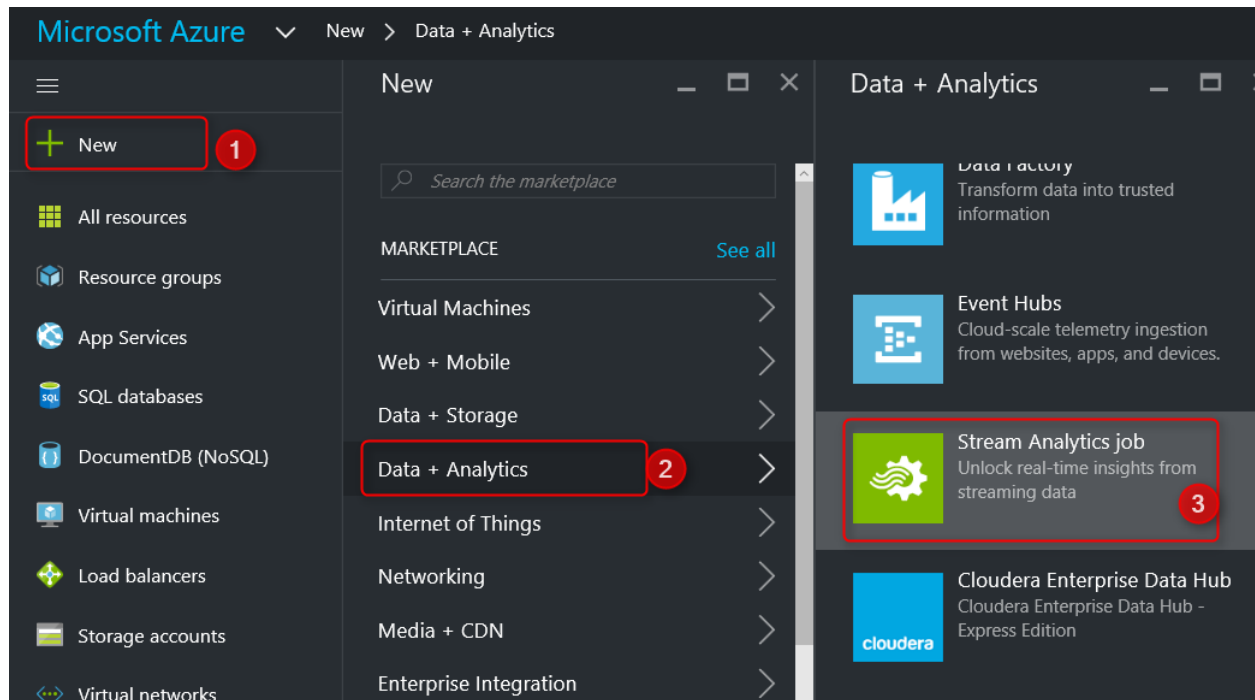
| NAME         | STATUS | MESSAGE RETENTION | PARTITION COUNT |
|--------------|--------|-------------------|-----------------|
| ▼ Event Hubs |        |                   |                 |
| pbi          | Active | 1                 | 2               |

For simplicity we keep all configuration and setting as is. So far, you have created the Event Hub which can be used as the data entry input point. Now Let's create an Azure Stream Analytics Job.

## Azure Stream Analytics

Stream Analytics is a data streaming technology under Azure list of services. With this technology you can pass data from input (such as Event Hub) to one or more outputs. Note that Stream Analytics is not a data storage. If you want the data to be stored you have to pass that to an storage output, such as Azure Blob Storage, or Azure SQL Database or Azure SQL Data Warehouse.

Create an Azure Stream Analytics Job from Azure portal under New, and Data + Analytics.



Set a Name, Resource Group, and Location for Stream Analytics Job.

New Stream Analytics J... — □ ×

\* Job name  
pbiStream

\* Subscription  
Free Trial ▼

\* Resource group ⓘ  
☐ Create new ☒ Use existing  
PBIRG ▼

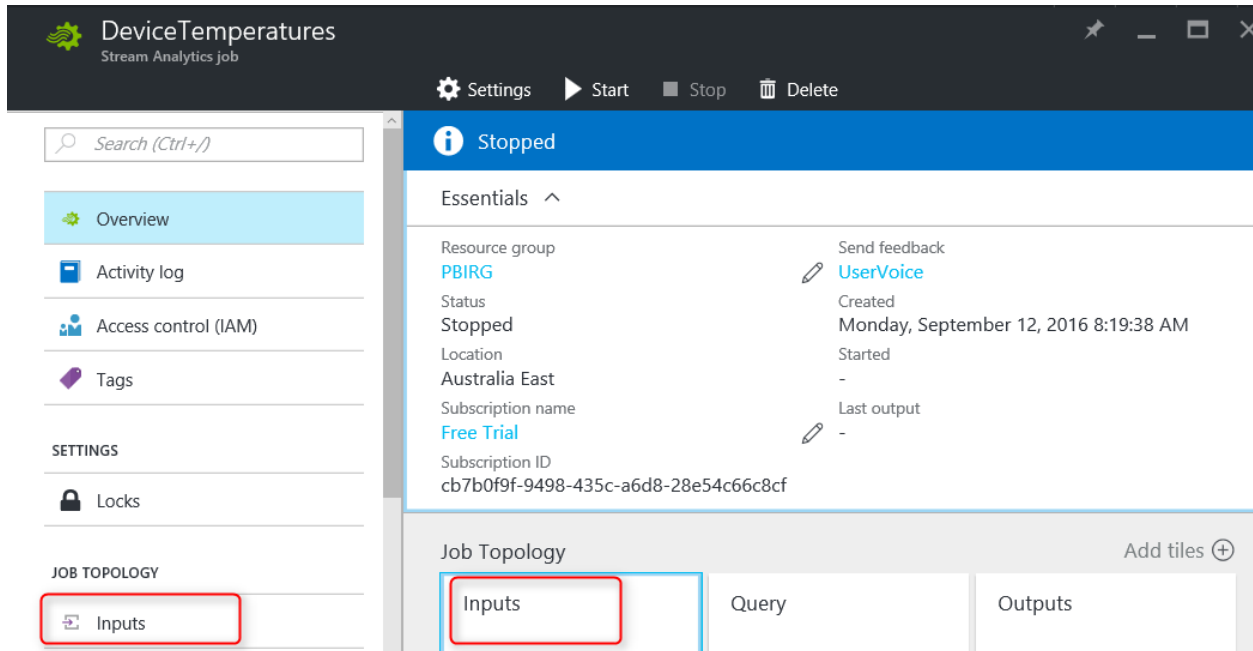
Location  
Australia East

☒ Pin to dashboard

Create [Automation options](#)

After creating the job, you will be redirected to the job's blade.





Job above named as DeviceTempratures, yours might be named something else.

## Input from Event Hub

From Job Topology click on Inputs, and then Add an Input in the new blade. In the New Input blade, set a name. set Source to Event Hub, and choose the service bus namespace as the namespace you entered earlier in this post (PBIRadacad). and select the event hub as pbi.

New input

\* Input alias

PBIEventHub

!

\* Source Type ⓘ

Data stream

▼

\* Source ⓘ

Event hub

▼

\* Subscription

Use event hub from current subscription

▼

\* Service bus namespace

PBIradacad

▼

\* Event hub name

pbi

▼

\* Event hub policy name

RootManageSharedAccessKey

▼

Event hub consumer group ⓘ

\* Event serialization format ⓘ

JSON

▼

Encoding ⓘ

UTF-8

▼

Create

Leave event serialization format as JSON and encoding as UTF-8. With setting above, now Stream Analytics reads the input from Event Hub.

## Output to Power BI

Go to Stream Analytics main blade and create an Output this time. Set a name such as PBIOutput, and set Sink to Power BI. This means Stream Analytics will output to Power BI as a data set.

New output

\* Output alias

PBIOutput

\* Sink ⓘ

Power BI

**Authorize Connection**

You'll need to authorize with Power BI to configure your output settings.

Authorize

Don't have a Microsoft Power BI account yet?

[Sign Up](#)

i

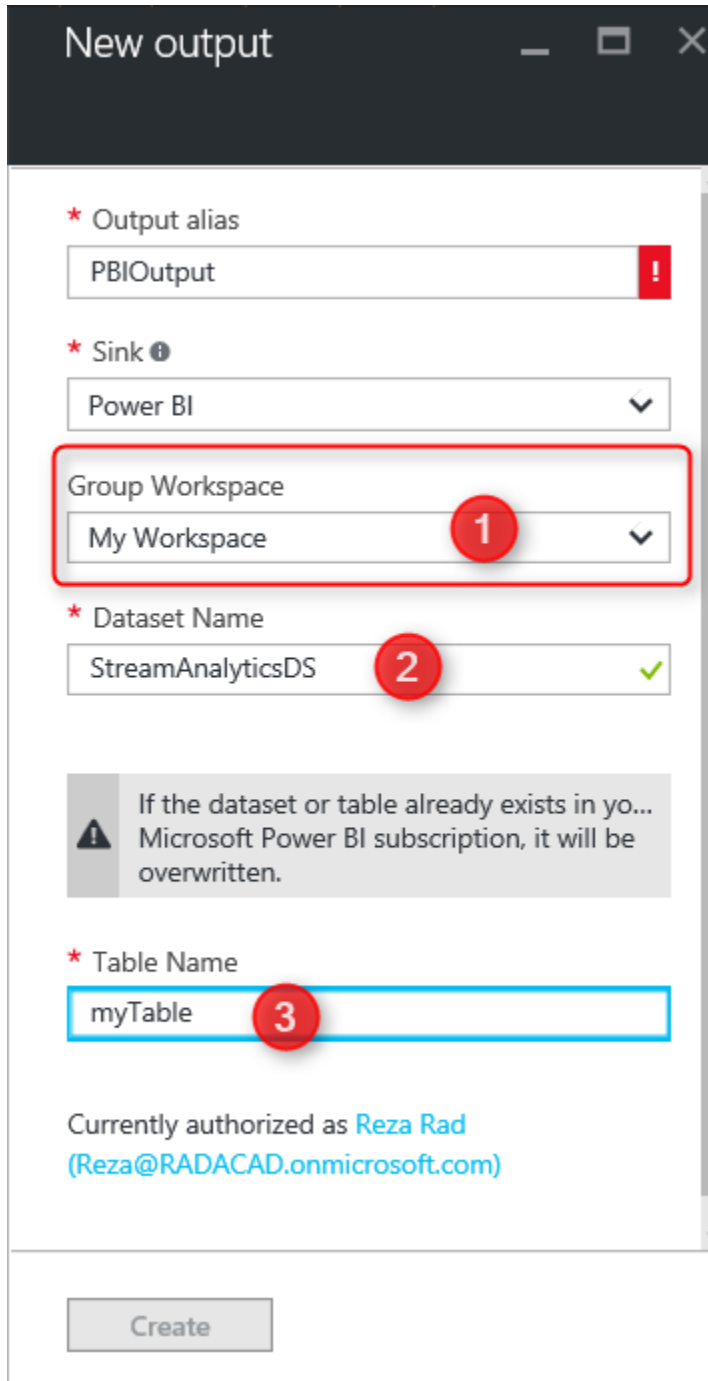
Note: You are granting this output permanent access to your Power BI dashboard. Should you need to revoke this access in the future you can do one of the following:

1. Change the user account password.
2. Delete this output.
3. Delete this job.

Create

Then Click on Authorize to map the Power BI account that you want to use here. You will be prompted to login with your Power BI username and password. This account can be different from your Azure account.

After authorization, you will see list of all groups in the Power BI account, and you can choose where the data set will be published to. (by default it goes to My Workspace, but you can change it to other groups). If you want to learn more about groups in Power BI, read [this post](#). Set a name for data set, for example StreamAnalyticsDS. Notice that if a data set with this name exists, data set will be overwritten. set a name for table, for example myTable. and then create the output.



New output

\* Output alias  
PBIOutput

\* Sink  
Power BI

Group Workspace  
My Workspace

\* Dataset Name  
StreamAnalyticsDS

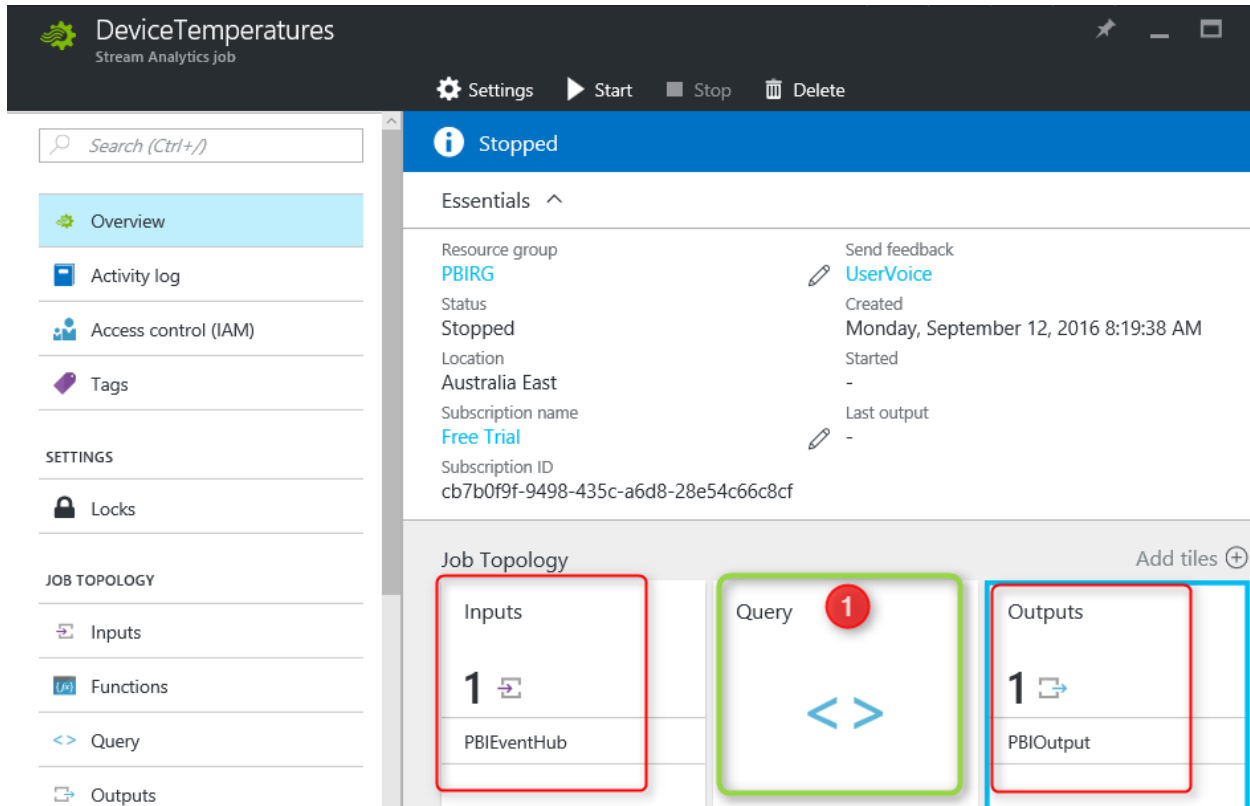
If the dataset or table already exists in yo...  
Microsoft Power BI subscription, it will be overwritten.

\* Table Name  
myTable

Currently authorized as [Reza Rad](#)  
([Reza@RADACAD.onmicrosoft.com](#))

Create

After creating output and input you will see them both under Stream Analytics blade. but you can't start the job yet. there is one step more.

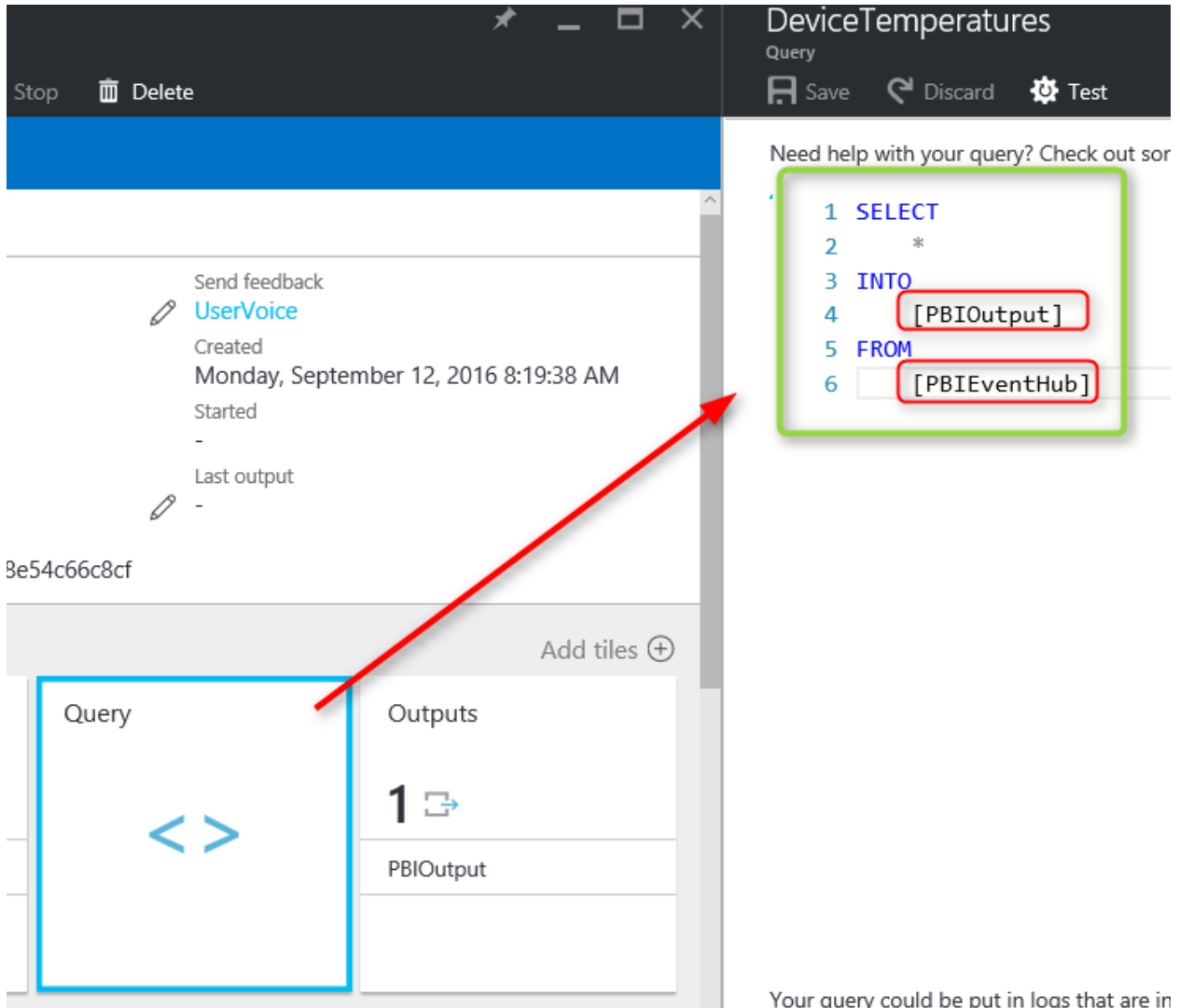


## Stream Analytics Query

Stream Analytics passes the data from input to output through query. Without a query you cannot start Stream Analytics job. For this example we use a very simple query which select everything from input and pass that to output. Click on Query item in the Stream Analytics main blade (as numbered 1 in above screenshot).

Change the default query to;

```
SELECT
*
INTO
[PBIOutput]
FROM
[PBIEventHub]
```



DeviceTemperatures  
Query

Stop Delete Save Discard Test

Need help with your query? Check out sor

```
1 SELECT
2 *
3 INTO
4 [PBIOutput]
5 FROM
6 [PBIEventHub]
```

Send feedback  
UserVoice  
Created  
Monday, September 12, 2016 8:19:38 AM  
Started  
-  
Last output  
-

3e54c66c8cf

Add tiles (+)

Query

Outputs

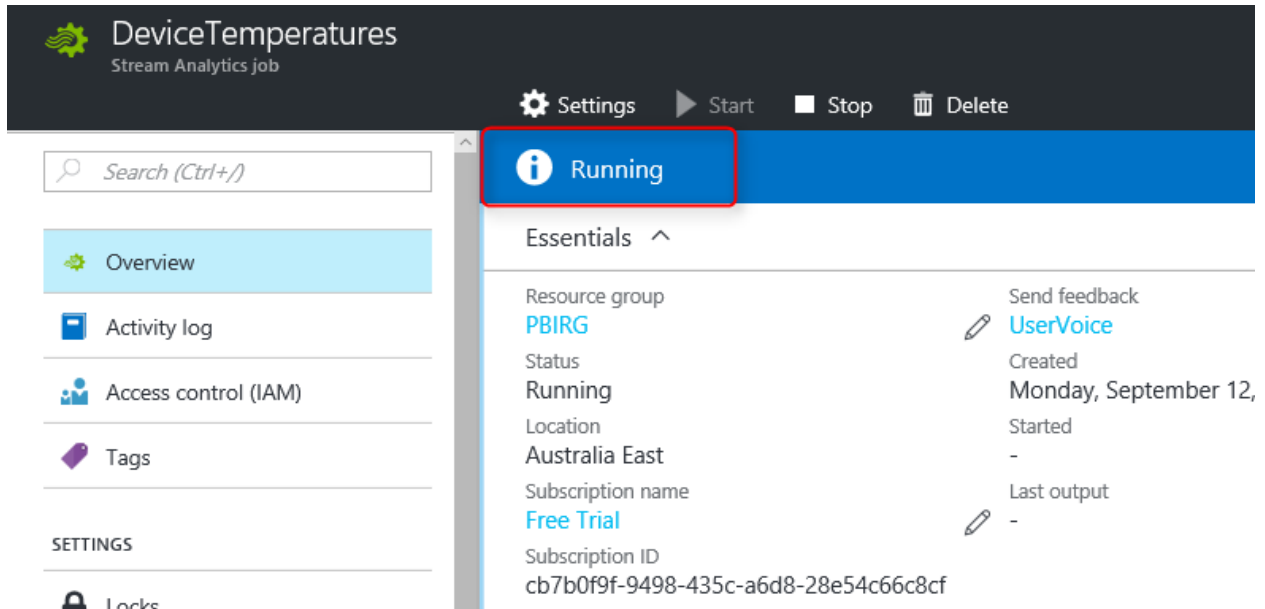
1 ➡

PBIOutput

Your query could be put in logs that are in

Notice that the query select everything from the input. so you have to put your input name after FROM clause. and it will insert result into output. So you need to put your output name after INTO clause. if names doesn't match you get failure error message at the time of starting the job.

Don't test the query. Just Save it and Start the job. It will take few minutes for job to start. Wait and check if the job started successfully. Otherwise failure might be related to input, output, or the query. You have to see job in RUNNING state as below.



**DeviceTemperatures**  
Stream Analytics job

Settings Start Stop Delete

Search (Ctrl+/)

Overview

Activity log

Access control (IAM)

Tags

SETTINGS

Locks

**Running**

Essentials ^

Resource group  
**PBIRG**

Status  
**Running**

Location  
**Australia East**

Subscription name  
**Free Trial**

Subscription ID  
**cb7b0f9f-9498-435c-a6d8-28e54c66c8cf**

Send feedback  
[UserVoice](#)

Created  
**Monday, September 12,**

Started  
-

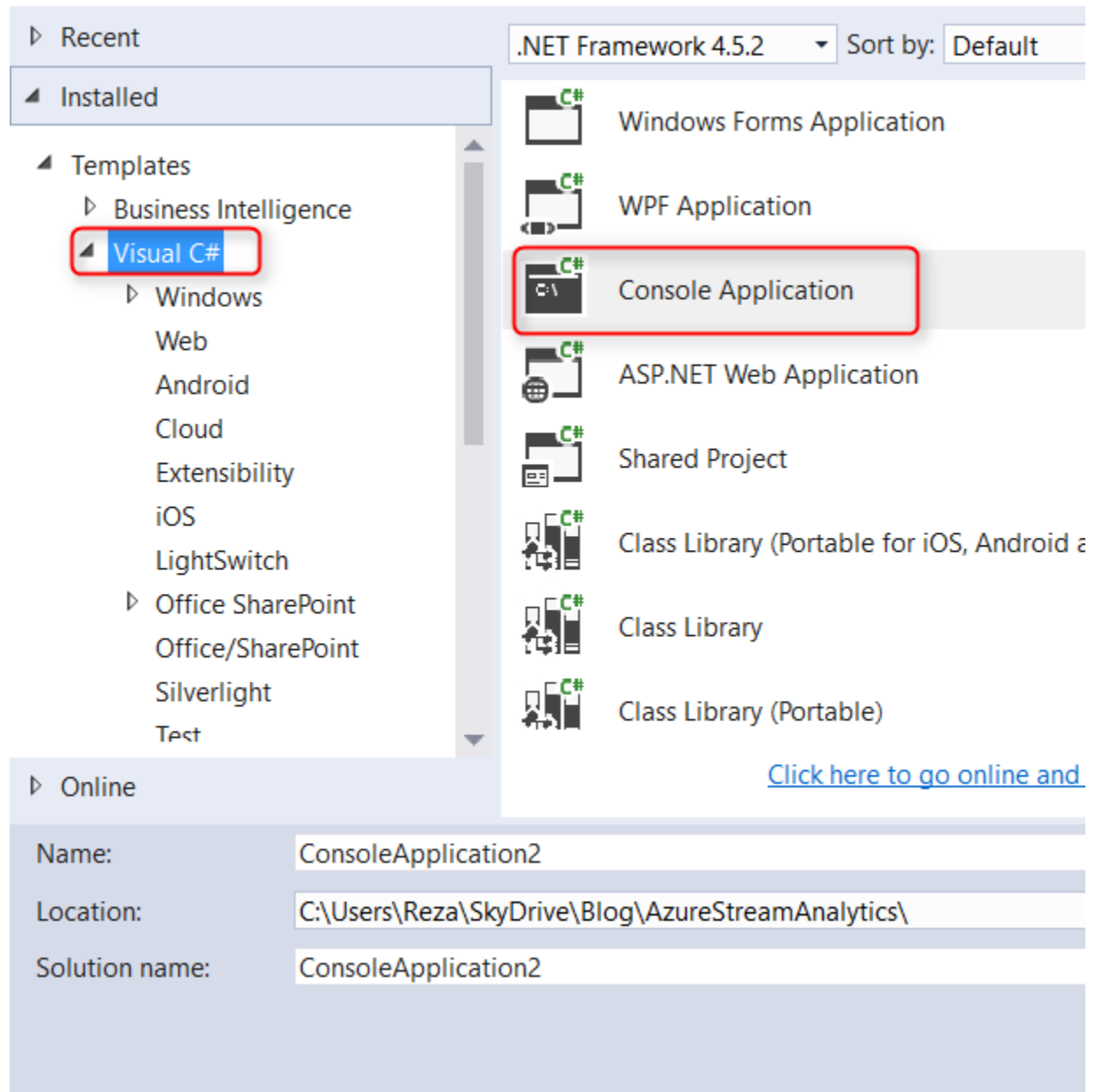
Last output  
-

## Push Data Application

As I mentioned earlier in this post, I'll be using a .NET C# console application to pass the data to Event Hub. Let's create the sample application by opening Visual Studio 2015 Professional. Create a New C# Console Application.



## New Project



Recent

Installed

Templates

- Business Intelligence
  - Visual C#**
  - Windows
  - Web
  - Android
  - Cloud
  - Extensibility
  - iOS
  - LightSwitch
- Office SharePoint
  - Office/SharePoint
  - Silverlight
  - Test

Online

.NET Framework 4.5.2 Sort by: Default

- Windows Forms Application
- WPF Application
- Console Application**
- ASP.NET Web Application
- Shared Project
- Class Library (Portable for iOS, Android a
- Class Library
- Class Library (Portable)

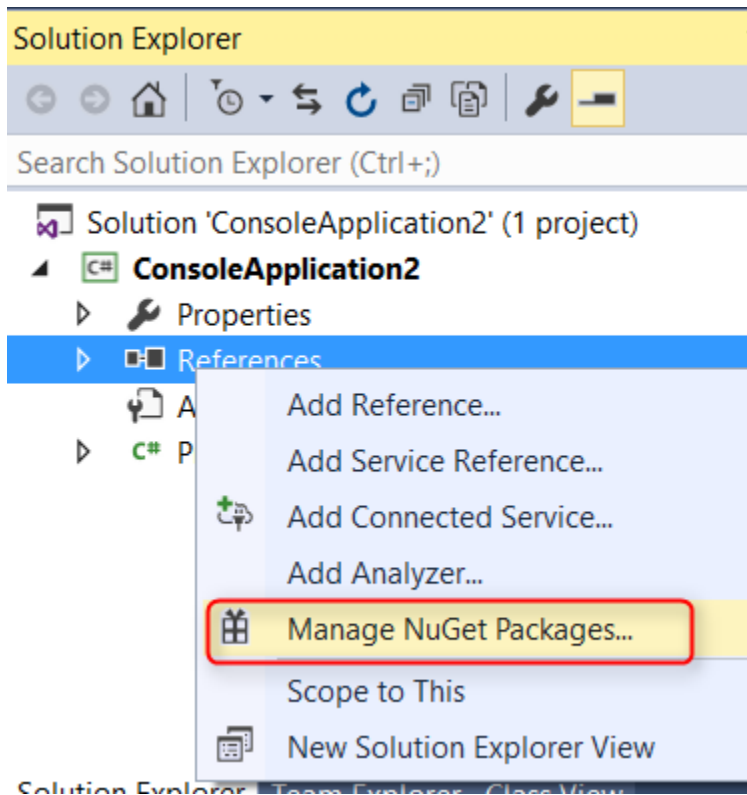
[Click here to go online and](#)

Name: ConsoleApplication2

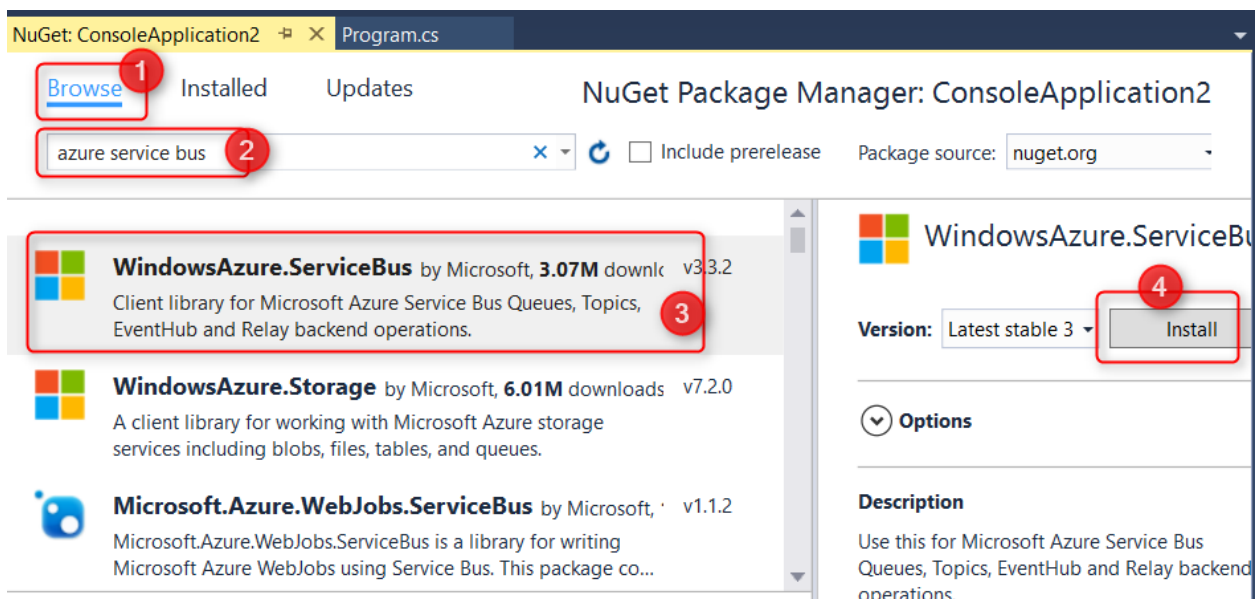
Location: C:\Users\Reza\SkyDrive\Blog\AzureStreamAnalytics\

Solution name: ConsoleApplication2

Name the application something, and when it is opened in Solution Explorer, go to References, right click and select Manage NuGet Packages



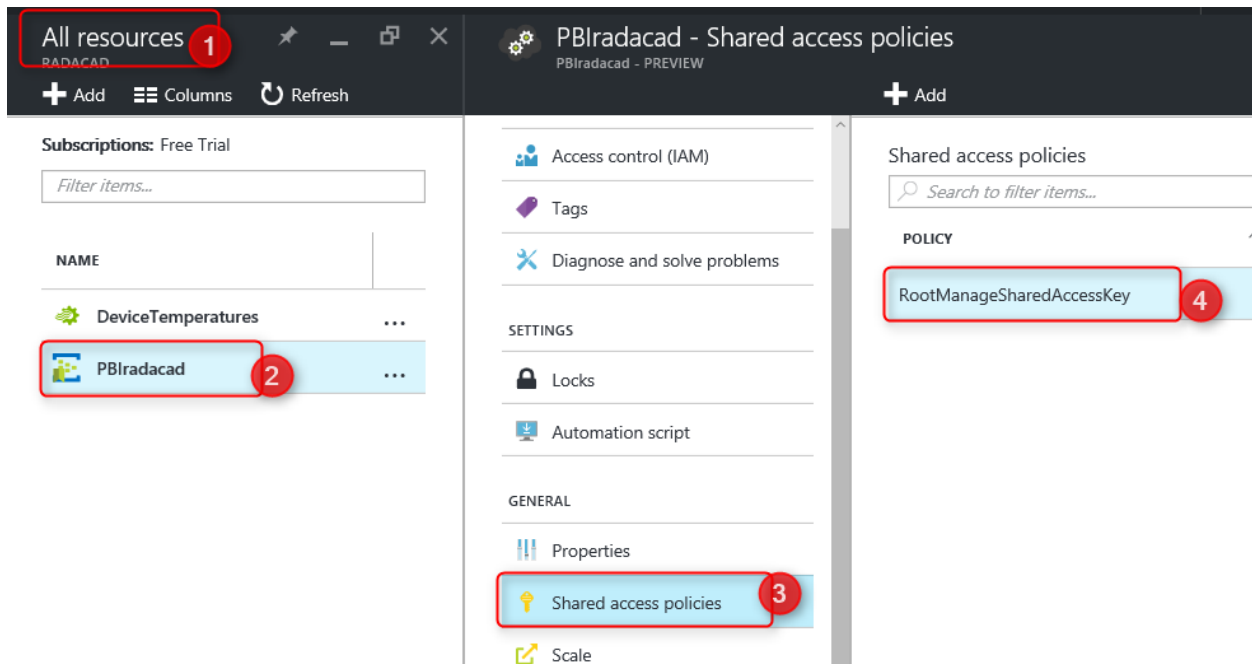
Click on Browse, Search for Azure Service Bus, and install it.



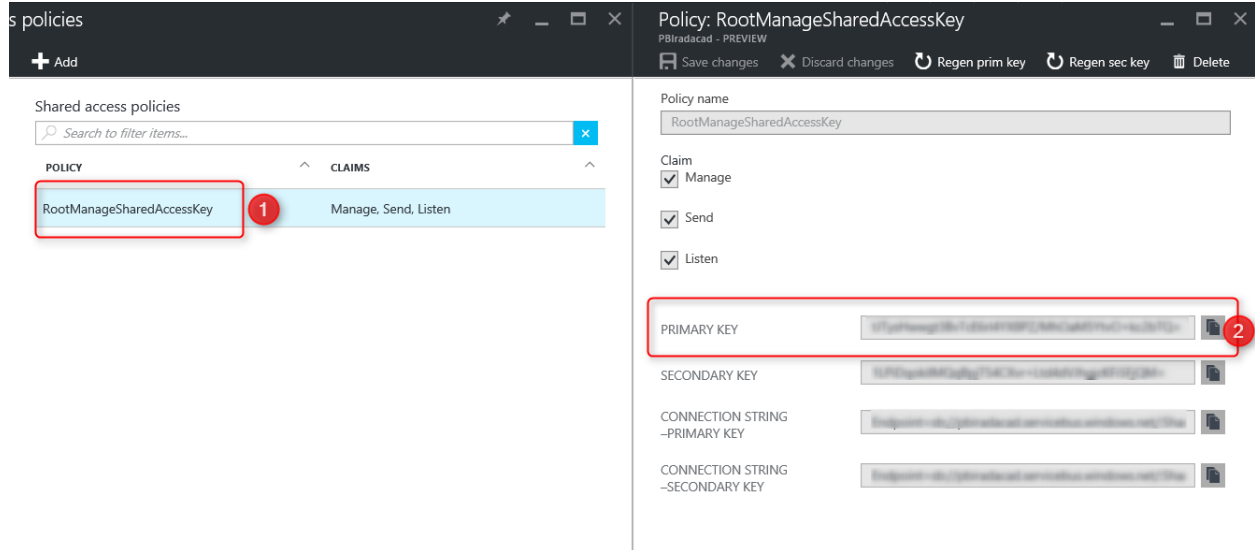
After installation, open App.Config file and at the end of the file search for appSettings. The connection string to Azure Service Bus is in the value beneath this section.

```
App.config NuGet: ConsoleApplication2 Program.cs
1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3 <startup>
4 <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2"/>
5 </startup>
6 <system.serviceModel>...</system.serviceModel>
47 <appSettings>
48 <!-- Service Bus specific app settings for messaging connections -->
49 <add key="Microsoft.ServiceBus.ConnectionString"
50 value="Endpoint=sb://[your namespace].servicebus.windows.net;SharedAccessKey=
51 </appSettings>
52 </configuration>
```

The highlighted line should be changed to your service bus connection string. To find your service bus connection string go to Azure Portal, find the service bus (we've created it in Event Hub section) under All Resources.

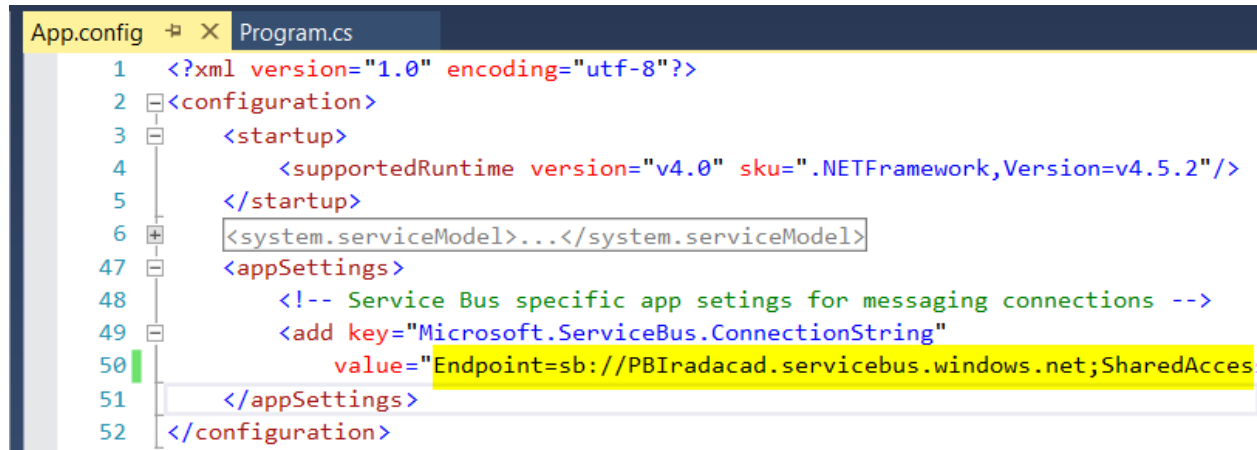


Under RootManageSharedAccessKey you'll find Primary Key;



The screenshot shows the Azure portal interface for managing policies. On the left, the 'Shared access policies' list shows a policy named 'RootManageSharedAccessKey' with a red box labeled '1' around it. On the right, the 'Policy: RootManageSharedAccessKey' details pane shows the 'PRIMARY KEY' field with a red box labeled '2' around it.

Copy the value of primary key here and replace the value in app.Config file with it. Save the App.Config after change and close it.

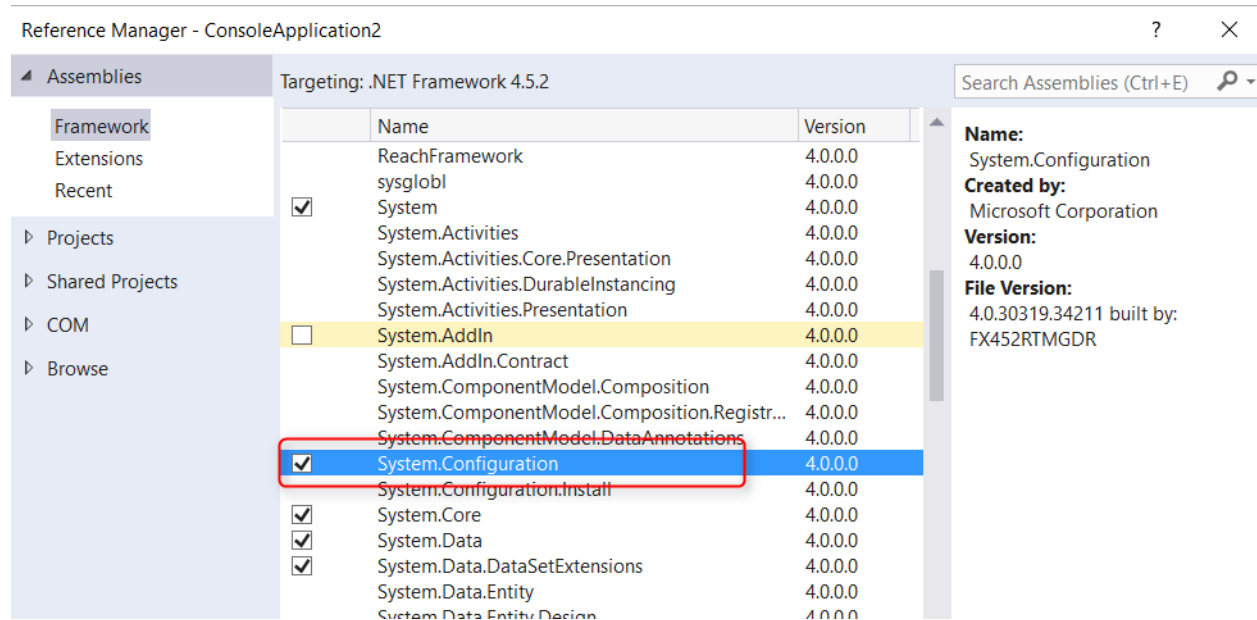


```

1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3 <startup>
4 <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2"/>
5 </startup>
6 <system.serviceModel>...</system.serviceModel>
47 <appSettings>
48 <!-- Service Bus specific app settings for messaging connections -->
49 <add key="Microsoft.ServiceBus.ConnectionString"
50 value="Endpoint=sb://PBiradacad.servicebus.windows.net;SharedAccessKey=
51 "
52 </appSettings>
53 </configuration>

```

Then right click on the references folder in solution explorer and add System.Configuration reference.



Now open Program.CS file and add these two namespaces using;

```
using System.Configuration;
using Microsoft.ServiceBus.Messaging;
```

Also add code below under Main method;

```
string connectionString = ConfigurationManager.AppSettings["Microsoft.ServiceBus.ConnectionString"];
EventHubClient client = EventHubClient.CreateFromConnectionString(connectionString, "pbi");

// Declare values that we're about to send
Int32 unixTimestamp = (Int32)(DateTime.UtcNow.Subtract(new DateTime(1970, 1,
1))).TotalSeconds;
Random r = new Random();
int currentValue = r.Next(0, 100);

// Generate string in JSON format
string postData = String.Format("{{ \"ts\": {0}, \"value\":{1} }}", unixTimestamp, currentValue);

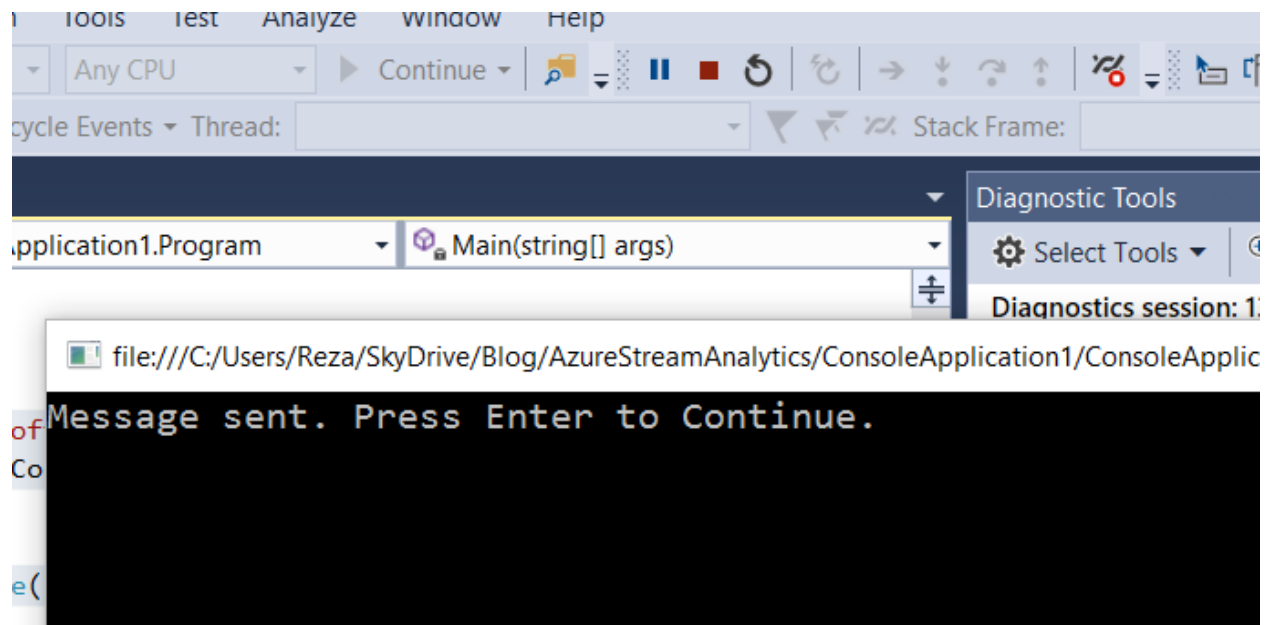
// Send the Message
client.Send(new EventData(Encoding.UTF8.GetBytes(postData)));

Console.WriteLine("Message sent. Press Enter to Continue.");
Console.ReadLine();
```

First Line in the code reads the connection string from app.config. Second line create an instance of EventHubClient with the connection string. Note that in this line connection created to Event Hub with name "pbi". if you named your Event Hub differently enter your own name.

Next three lines are creating random integer value with time stamp. and line with `String.Format` will generate a JSON formatted string. and `Client.Send` send the string in UTF-8 encoding to Service Bus (Event Hub).

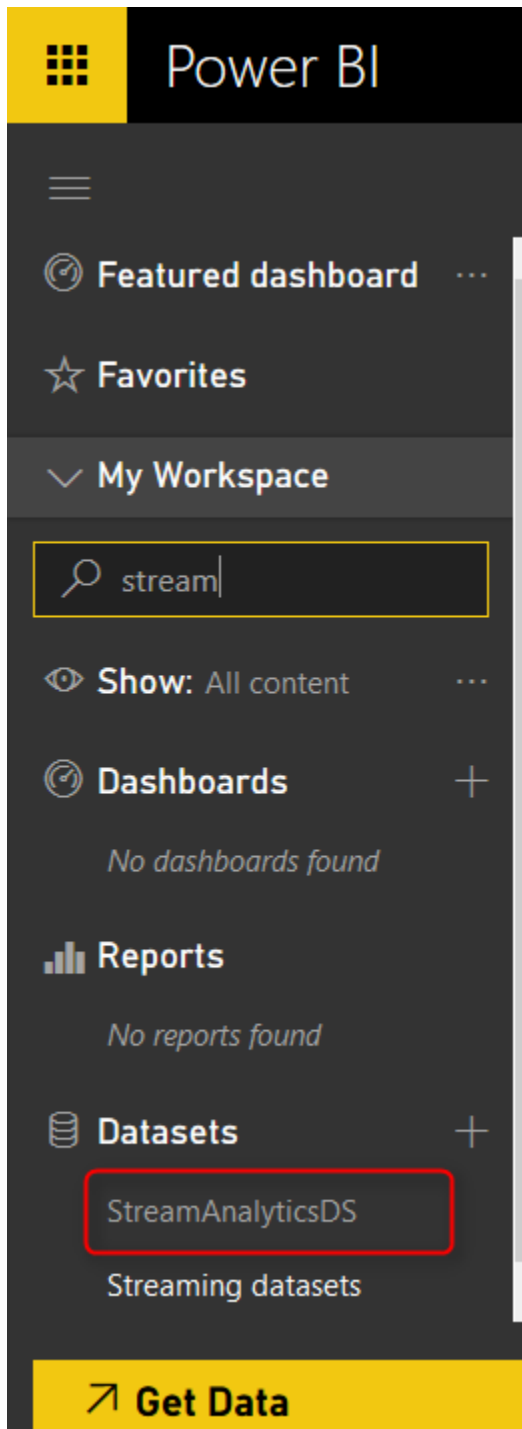
Execution of this application will pass only one data point to Event Hub. For adding another entry you have to run it again. Now the application is ready to run. Let's run it for the first time.



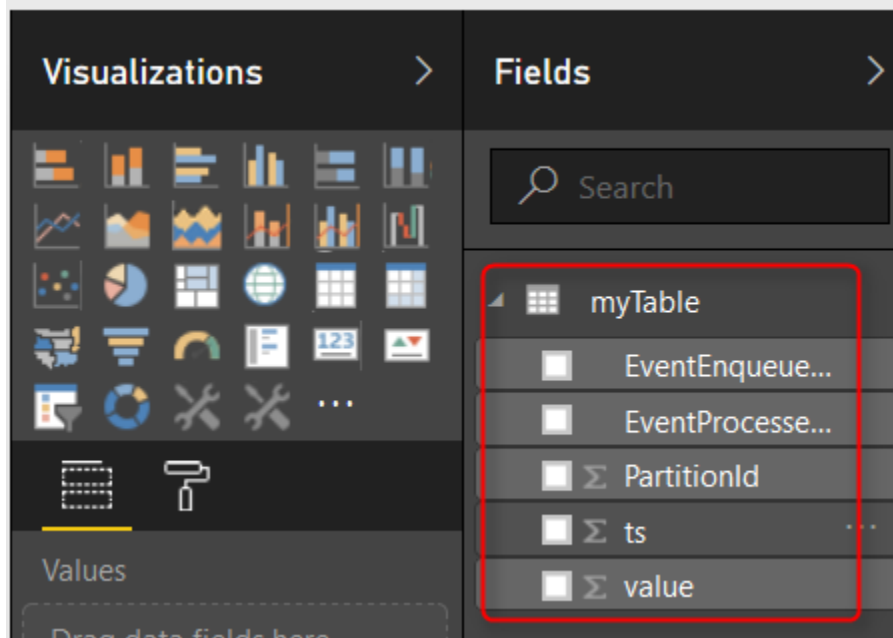
After the execution we expect Power BI to show something. Let's see.

## Power BI Real-time Dashboard

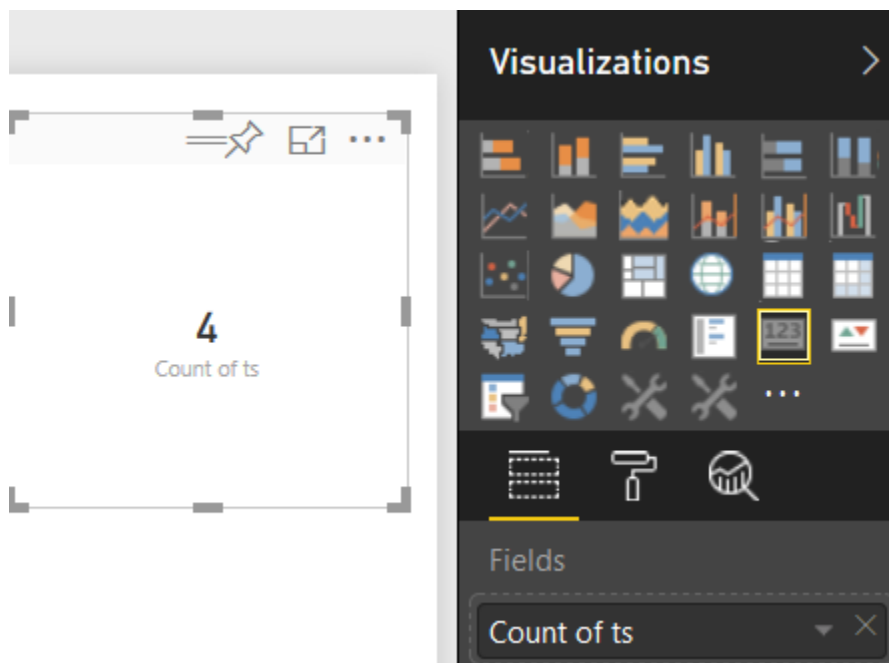
Login to Power BI service with the account you have used in the Stream Analytics Authorization section. In My Workspace (or any other group you've selected in Stream Analytics Output section) you'll find the new data set. if you followed the naming I suggested it should be named `StreamAnalyticsDS`.



Open the data set and you'll see the table under data set with two fields we've passed (ts, and value) plus other fields from Event Hub (EventEnqueuedUtcTime, EventProcessedUtcTime, and PartitionId).

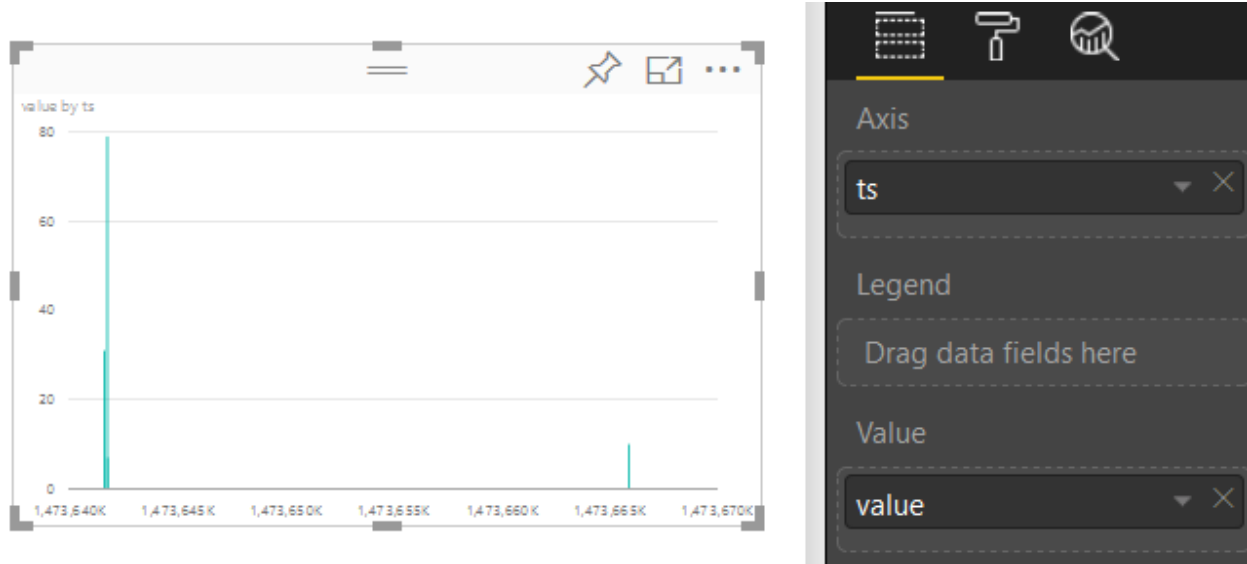


As you can see the structure can be anything sent from Event Hub and Stream Analytics. Now create two simple visualizations. One Card Visualization with distinct count of ts. This will show number of times we send data point (execute the console application) to Event Hub.

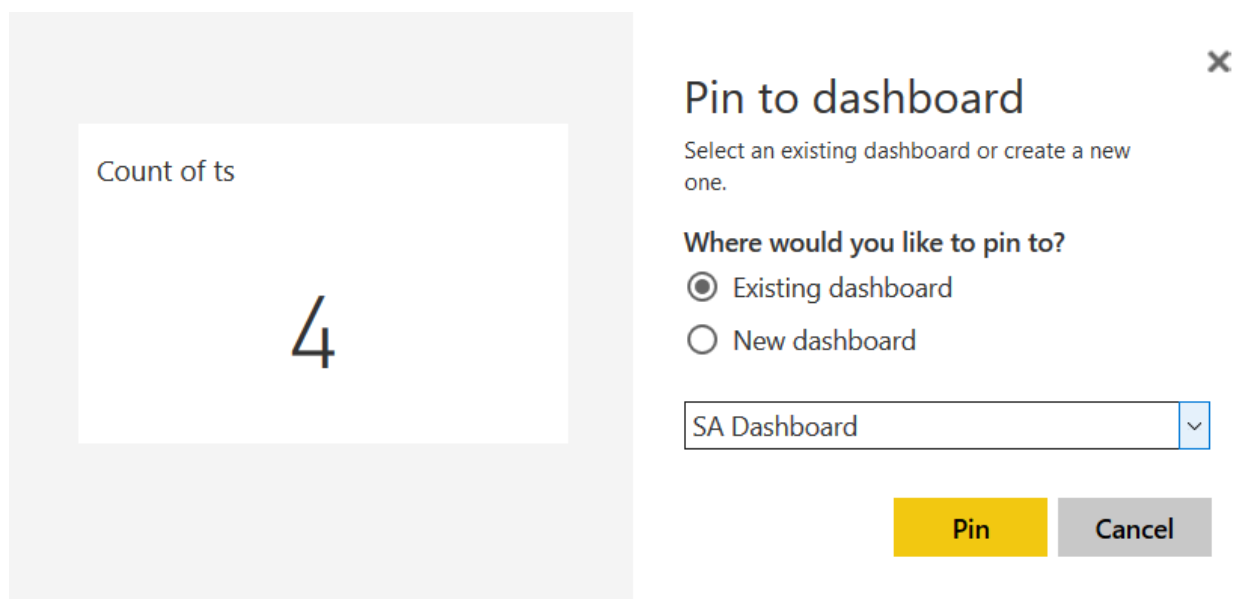


Also a Column chart with TS at Axis, and Value as the Value of chart.

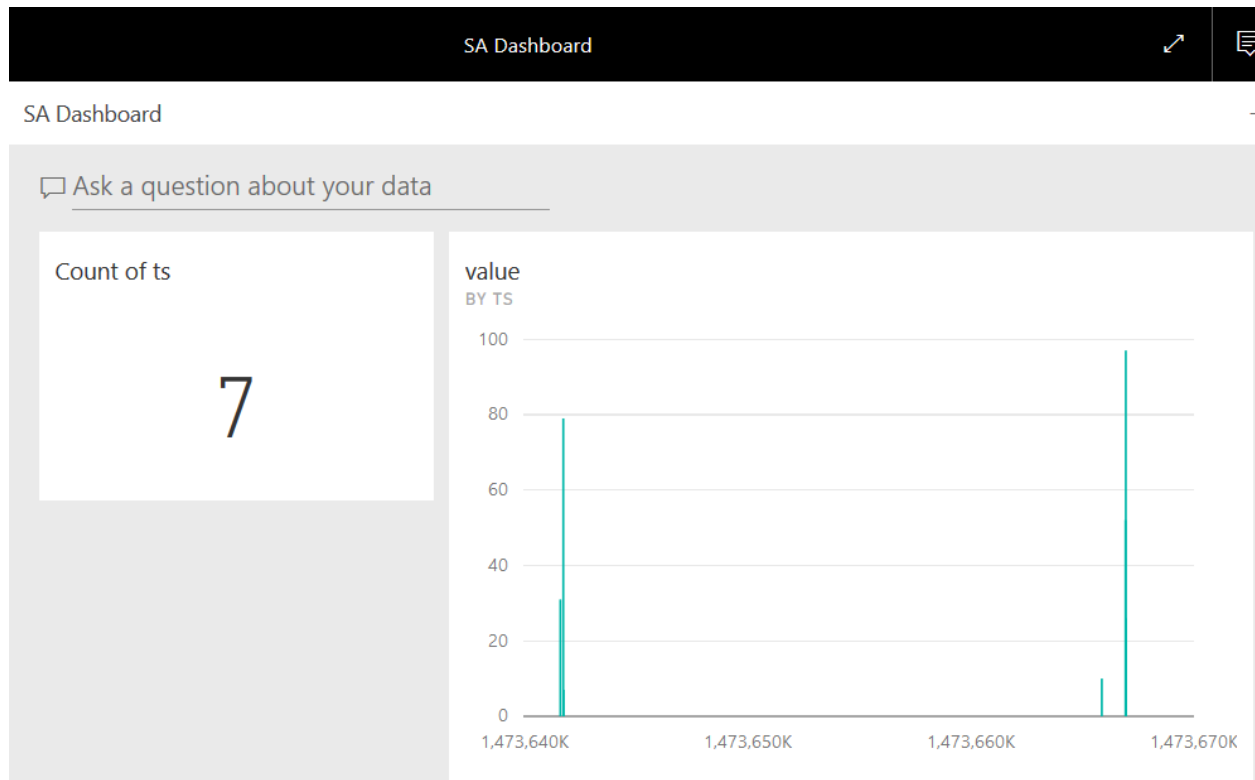




Save this report, and Pin these two visualizations into a new dashboard.



Now go to the dashboard. While dashboard is open execute the console application. after few seconds of execution and sending message you will see the dashboard shows result. Fantastic, You've build your first Azure Stream Analytics and Power BI real-time dashboard now.

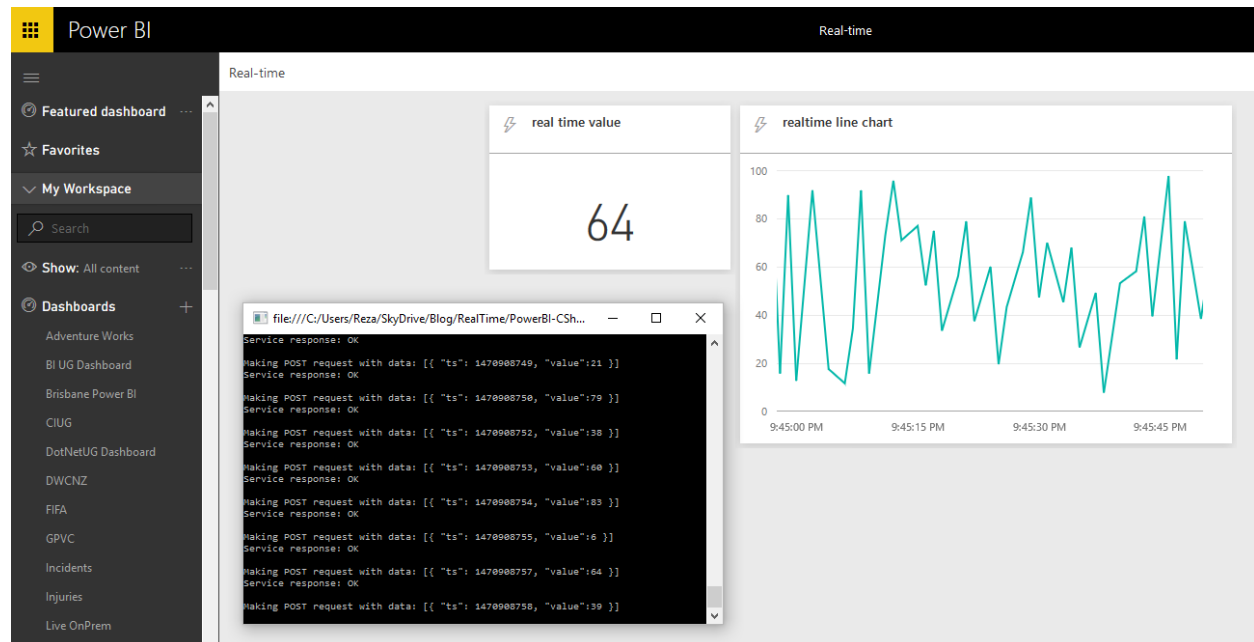


## Think Further

Real-time dashboard with Power BI, Azure Stream Analytics, and Event Hub open the path to many possible solutions. You can analyze Twitter data real-time. Or you can pass data from an IoT device with IoT Hub into a real-time dashboard. Come with ideas of what real-time dashboard with Stream Analytics can do? I like to hear your ideas.

# Monitor Real-time Data with Power BI Dashboards

Published Date: August 11, 2016



I've hear this a lot in my presentations that "Does Power BI support real-time data?", And I answer yes. In fact this feature was available long time ago with Azure Stream Analytics and Power BI, however [recently some new changes happened](#) in this area which allows Power BI dashboards to monitor real-time data changes on the dashboard from a data coming from API, and also PubNub. In this post I'll show you how easy is to create a dashboard that shows real-time data coming from API. If you want to learn more about Power BI, read [Power BI online book; from Rookie to Rock Star](#).

## Real-Time Data, Instant Monitoring

One of the key features of monitoring tools is the ability to monitor the real-time data instantly when the data refreshes. That means as a user you don't need to click on refresh button to see the latest data. This single key feature brings heaps of value to Power BI as a monitoring tool. This ability isn't new, in fact [Amir NetZ showed an awesome demo](#) of an IoT device working with Azure Stream Analytics and Power BI as a real-time monitoring. So what is new is that we can now have this feature in the new

generation of Power BI with a dashboard tile connected to streaming data. Streaming data sets can be now either API, Azure Stream Analytics (coming soon), and PubNub. More streaming services will come very soon I believe.

In this post I want to show you only streaming data through API, and showing that in a dashboard tile in Power BI. This single streaming option by itself opens many gates to Power BI as a real-time data monitoring tool. So let's start.

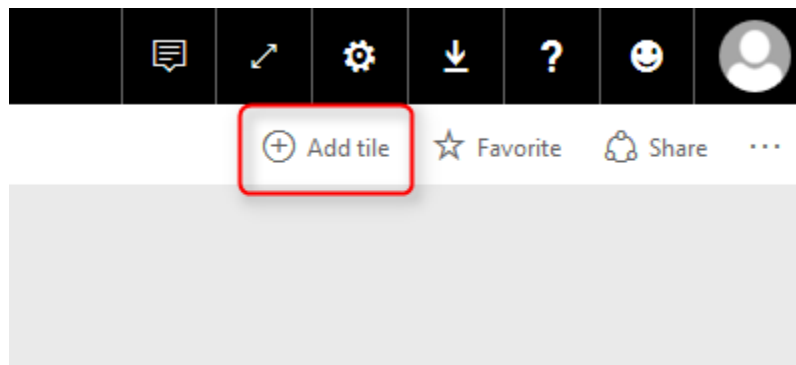
## Prerequisite

For this example I will be using Microsoft sample .NET application named PBIRealTimeStreaming. You can download it from [here](#).

You need Visual Studio (Not SSDT) for running this example solution.

## Real-time Data Set

Start with logging in to Power BI service, create a new dashboard (or open an existing dashboard). And create a New Tile.



In the New Tile pane you now can see an option to add a tile for real-time data.

## Add tile

Select source

### MEDIA



WEB CONTENT



IMAGE

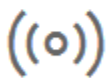


TEXT BOX



VIDEO

### REAL-TIME DATA



CUSTOM  
STREAMING DATA

Next

Cancel

Select this option and click Next. if this is the first time you are using this service, then you probably don't have any streaming data set existing there. So go to Manage Data to create one;

## Add a custom streaming data tile

Choose a streaming dataset

rtTestDataset

Not finding your streaming dataset? Go to [manage data](#)

Back

Next

Cancel

Now you are in streaming data sets window. you can see list of existing data sets, or create a new one with click on Add Streaming Dataset option on top right hand side.

Sales > Streaming data

[+ Add streaming dataset](#)

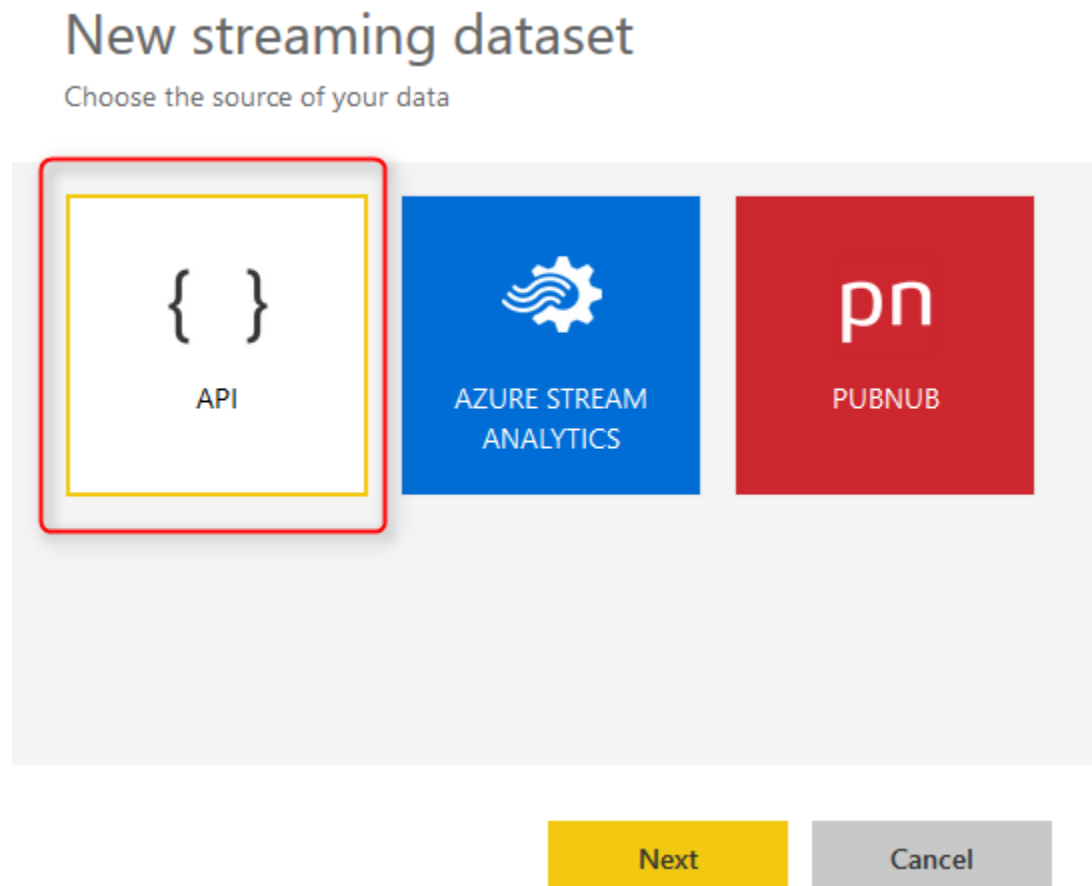
Streaming data

Search streaming datasets...

| NAME          | TYPE | USED IN DASHBOARDS | HISTORICAL | ACTIONS                                                                                                                                                                                                                                                           |
|---------------|------|--------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rtTestDataset | API  |                    | Disabled   |    |

## Create Sample Streaming Data Set

As I've mentioned before at the time of writing this post only these three options for streaming data is available: API, Azure Stream Analytics, and PubNub (which is a data streaming service).



Because for this example we are using API as a streaming option, choose API and continue. Now create a new data set with a name such as mySampleRTDS or any other names you like. You have to create two columns as below;

1. ts column with DateTime data type
2. value column with Number data type

These are columns that will be created and passed through a .NET application through Power BI API to this data set.

## New streaming dataset

Create a streaming dataset and integrate our API into your device or application to send data. [Learn more about the API.](#)

Dataset name \*

mysampleRTDS

Values from stream \*

ts

DateTime

value

Number

Enter a new value name

Text

```
[
 {
 "ts" : "2016-08-11T09:11:28.446Z",
 "value" : 98.6
 }
]
```

Historic data analysis

☐ Off

Back

Create

Cancel



Now click on create. This will create the data set with a Push URL that can be used in the application through API. copy this URL. We will be using that in the application next step.

## ✓ Streaming dataset created

The schema for mysampleRTDS is created.

Push URL

<https://api.powerbi.com/beta/cc80f8d6-3c35-4da4-b0f8-cbeadb00467e/data>

Raw

cURL

PowerShell

```
[
 {
 "ts" : "2016-08-11T09:15:13.139Z",
 "value" : 98.6
 }
]
```

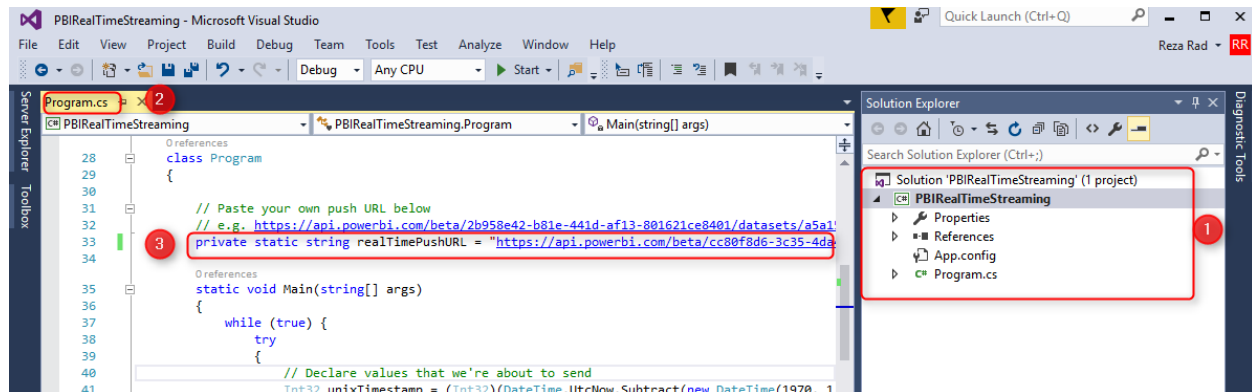
## Sample Streaming Data API Application

As I mentioned before, [download](#) the sample project. and then select the project under this path:

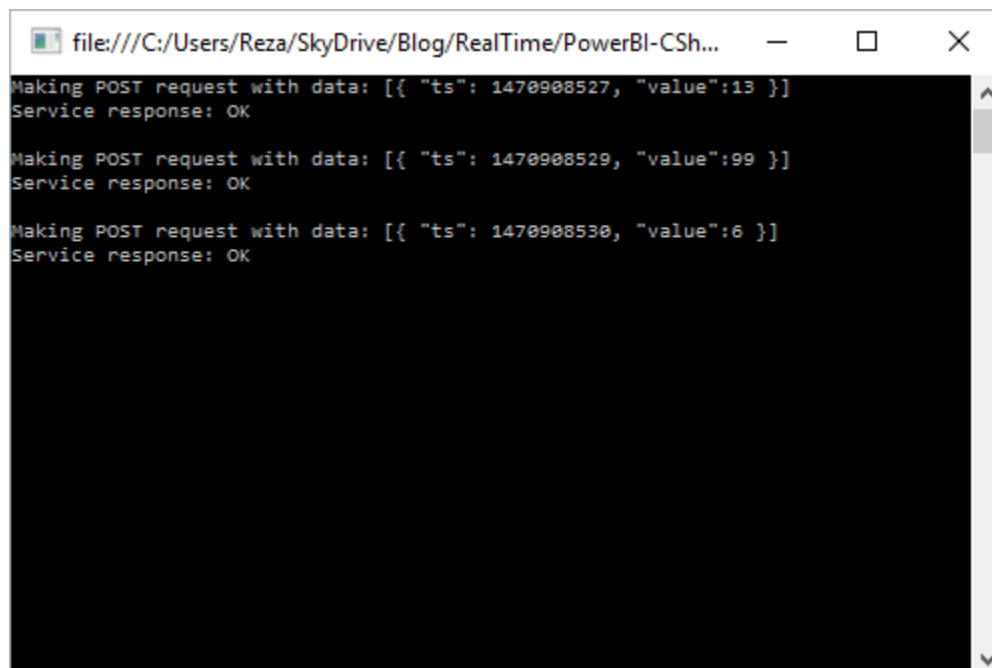
samples/consoleapp/getting-started-for-dotnet-real-time/PBIRealTimeStreaming

And open the PBIRealTimeStreaming solution in Visual Studio. This project creates a data set including a datetime value and a numeric value and will pass that through Power BI API (which will be discussed later in another post) to Power BI service.

Open Program.cs file and in line 33 change the value of realTimePushURL to the URL which we've copied from above step.



Run the application with pressing F5, or right click on project, Debug, Start New instance. The application will generate a data point in each second and send it through API.



## Create Dashboard Tile

Now, let's create the dashboard tile for this streaming data set. Follow the steps and choose the mySampleRTDS as the data streaming source for the tile.

There are number of charts/visualizations that you can use with streaming data set. Choose Card for the first example. and set the value of that to "value" data field

## Add a custom streaming data tile

Choose a streaming dataset > Visualization design

Visualization Type

Card

Fields

value

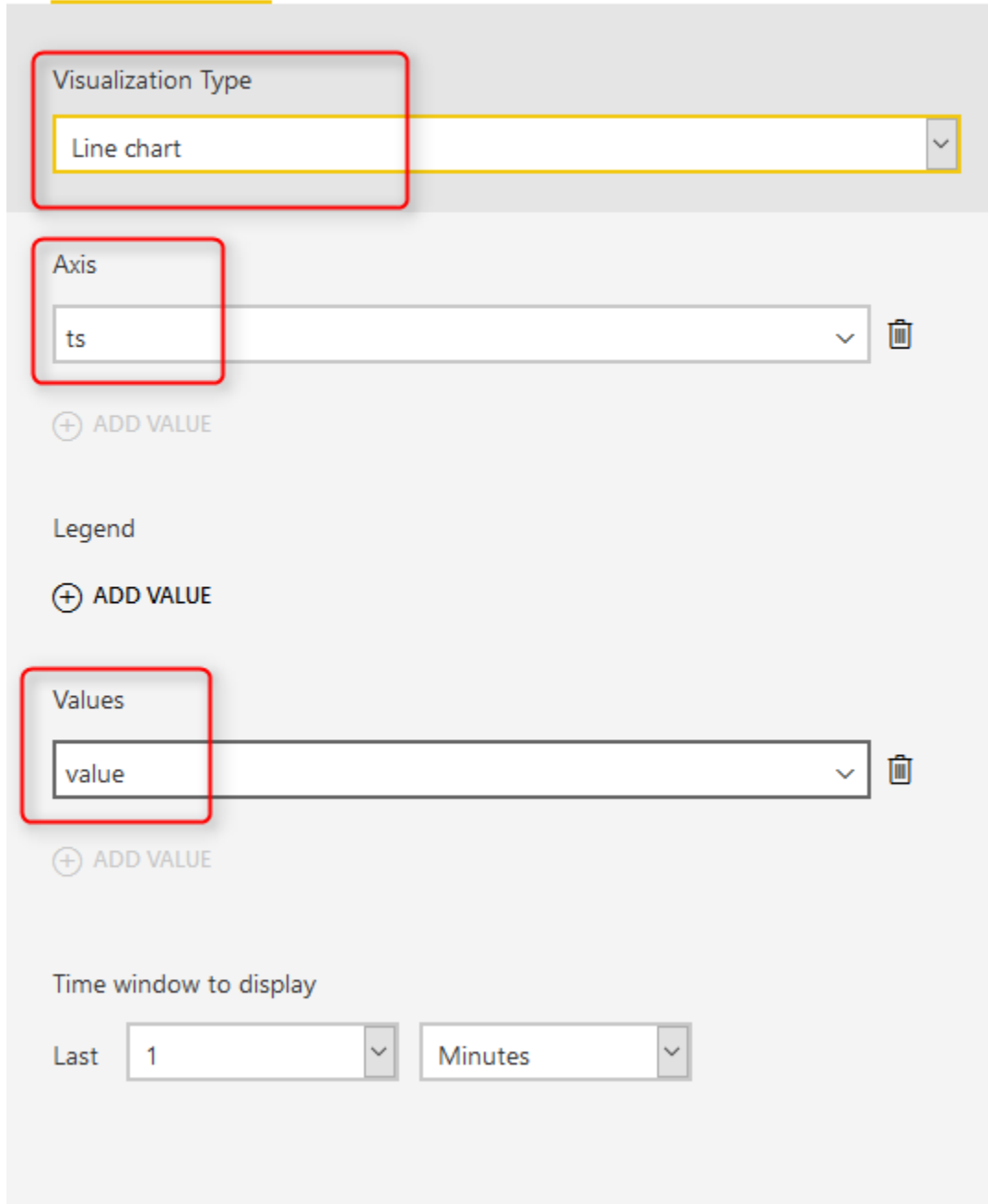
+ ADD VALUE

Add another dashboard tile for the same data set, and set this to be a Line chart with ts field in Axis, and value field in Values.

## Tile details

Visualization design

Tile details



Visualization Type

Line chart

Axis

ts

⊕ ADD VALUE

Legend

⊕ ADD VALUE

Values

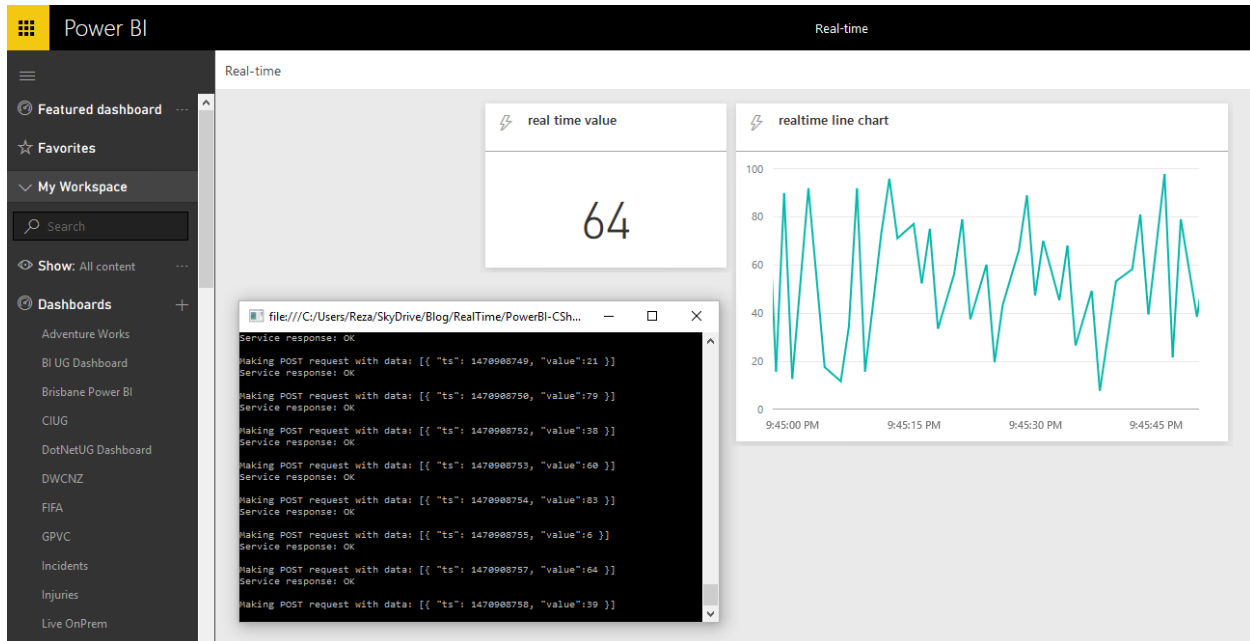
value

⊕ ADD VALUE

Time window to display

Last 1 Minutes

After adding each chart you will immediately see the real-time dashboard working and changes every second (because streaming data from application updates every second);

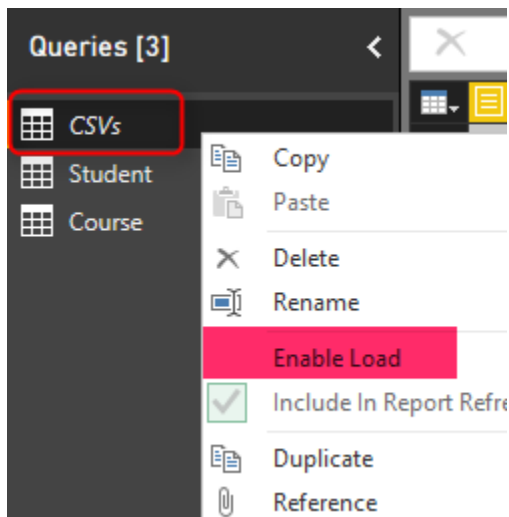


## API Data Can Be Anything

The sample used here had only one datetime field and a value field. However you can create your own data stream from any event, notifications, or data set that you want to be monitored real time. With a simple application the stream of data can be generated. Or it can come from Azure Stream Analytics or PubNub. Don't limit your options, think about where you can use this feature to solve a real-time monitoring challenge.

# Performance Tip for Power BI; Enable Load Sucks Memory Up

Published Date: November 14, 2016



In the area of performance tuning a Power BI model many things has to be considered, most of them around consumption of the CPU and RAM. One of the most basic but important consideration is minimizing the usage of memory. By default all queries from Query Editor will be loaded into the memory of Power BI Model. In this post I'll show you an example to disable the load for some queries, especially queries that used as intermediate transformation to produce the final query for the model. This is a very basic tip but very important when your model grows big. If you want to learn more about Power BI read the [Power BI online book from Rookie to Rock Star](#).

## Prerequisite

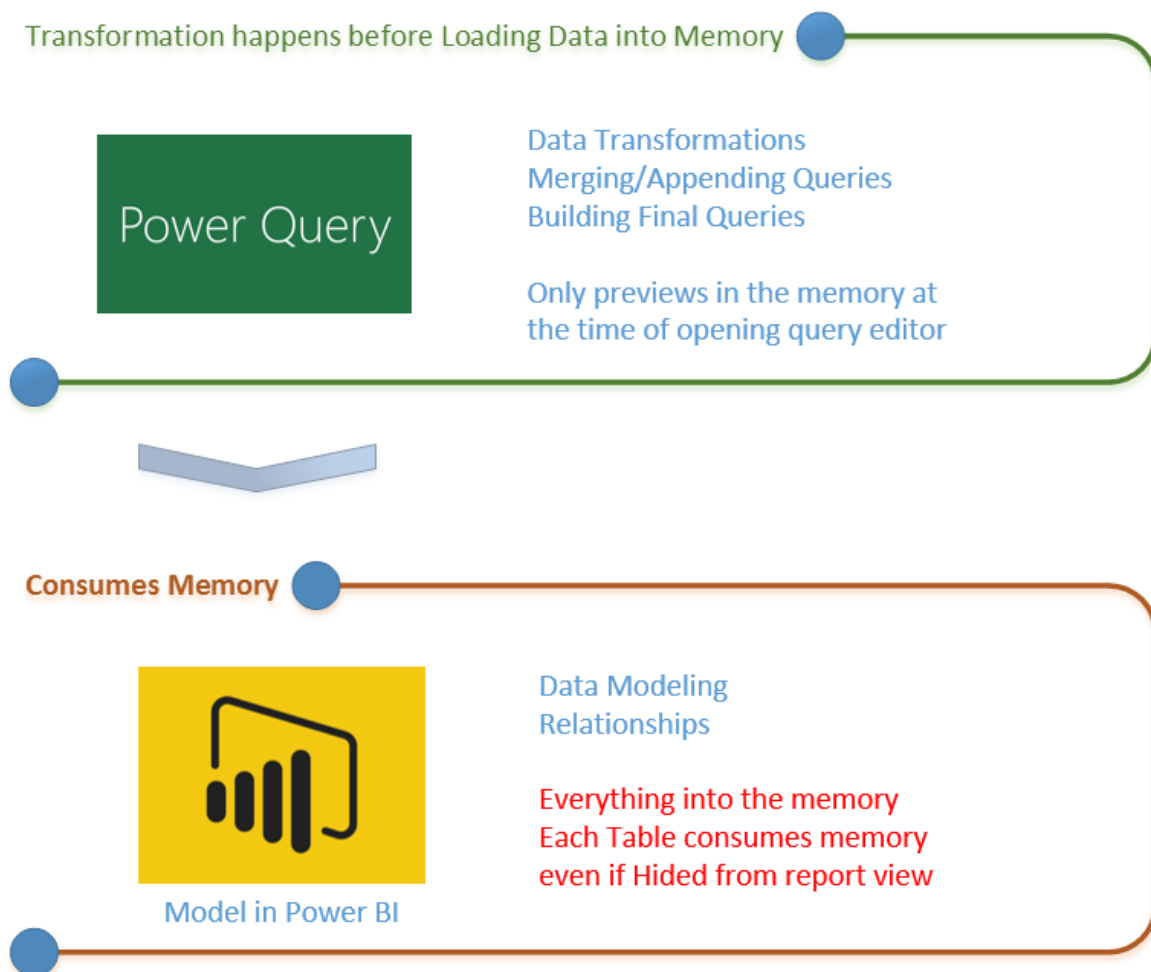
For running examples of this book you need to download the ZIP file here;

[Download ZIP file for CSVs](#)

## Load Mechanism for Power Query

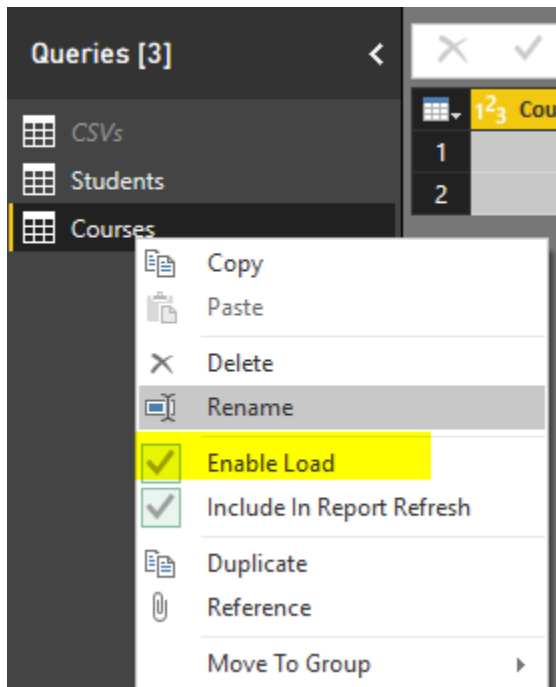
By Default all queries in Power Query will be loaded into the Power BI model. This behavior might be a desired behavior if you are connecting to a proper star schema modeled data warehouse, because normally you don't need to make many changes in the structure of queries. However this brings some issues if you are connected to a

transactional data store, some files, web source, and many other non-star schema modeled data sources. Normally when you get data from a data source, you apply transformations for rows and columns, and merge queries or append them, and you might end up to have 5 tables out of 10 queries as final queries. By default when you close and apply your query editor window all queries will be loaded into the model no matter if you want to use them in your final model or not.



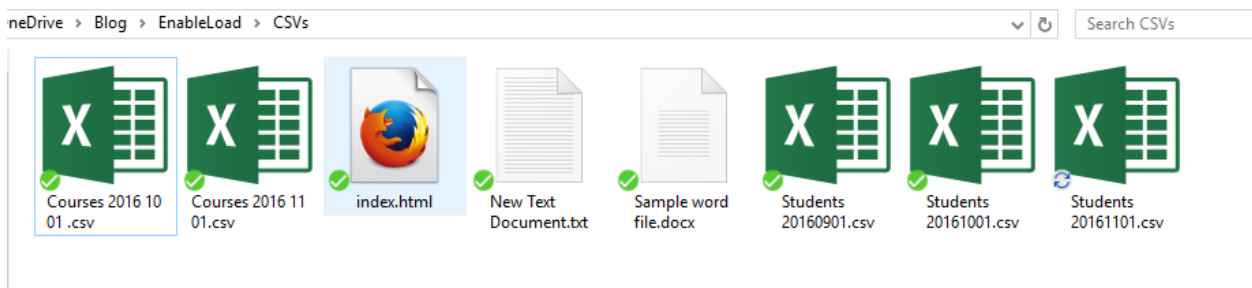
For every query that loads into model memory will be consumed. and Memory is our asset in the Model, less memory consumption leads to better performance in most of the cases. I have seen lots of models that people Hide the unwanted query from the model, This approach doesn't help to the performance because hided query will still consume the memory. The best approach is to disable loading before closing query editor. Disabling Load doesn't mean the query won't be refreshed, it only means the

query won't be loaded into the memory. When you click on Refresh model in Power BI, or when a scheduled refresh happens even queries marked as Disable Load will be refreshed, but their data will be used as intermediate source for other queries instead of loading directly into the model. This is a very basic performance tuning tip, but very important when your Power BI model grows bigger and bigger. Let's look at this through an example.



## Example Scenario

In this example Scenario I want to get list of all files from a directory. There are two types of CSV files in the directory, some Students files, and some course files, both are CSV files, but different structured. Also there might be some other files in the directory, such as Word files or HTML files which I don't want to process.



Aim is to load data rows of all students and courses as two separate tables into the model. Instead of fetching files from the folder twice, we use one query to fetch the files,

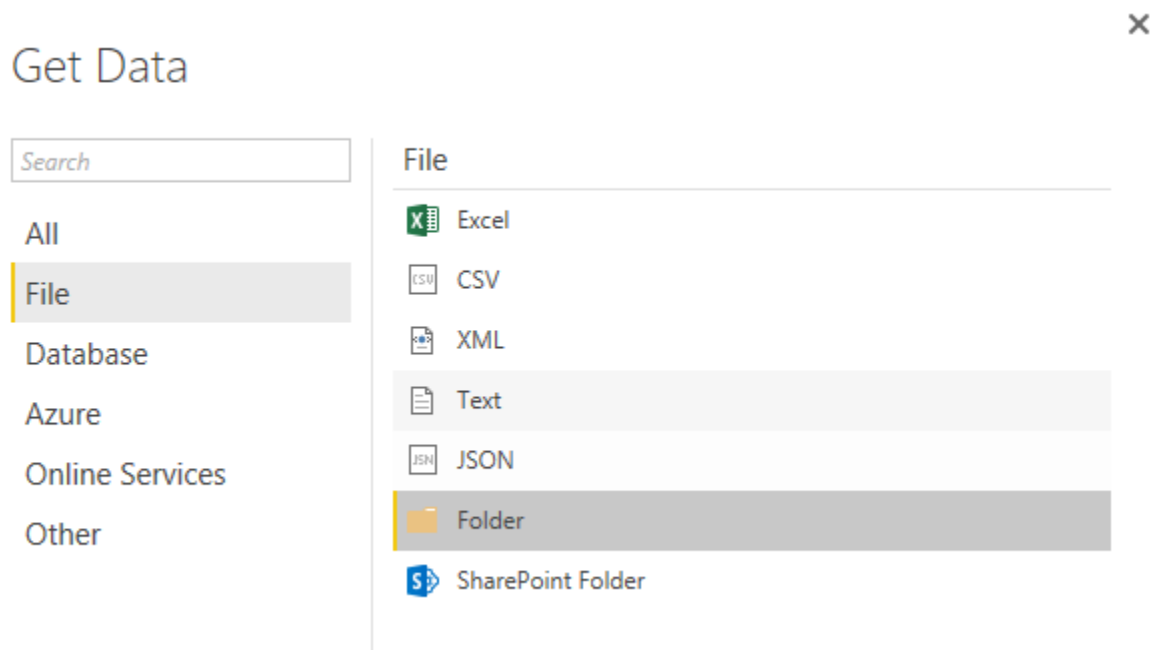


and then use it as a reference for other queries. The referenced query itself doesn't need to be loaded into the model. Let's build the solution and see how it works in action.

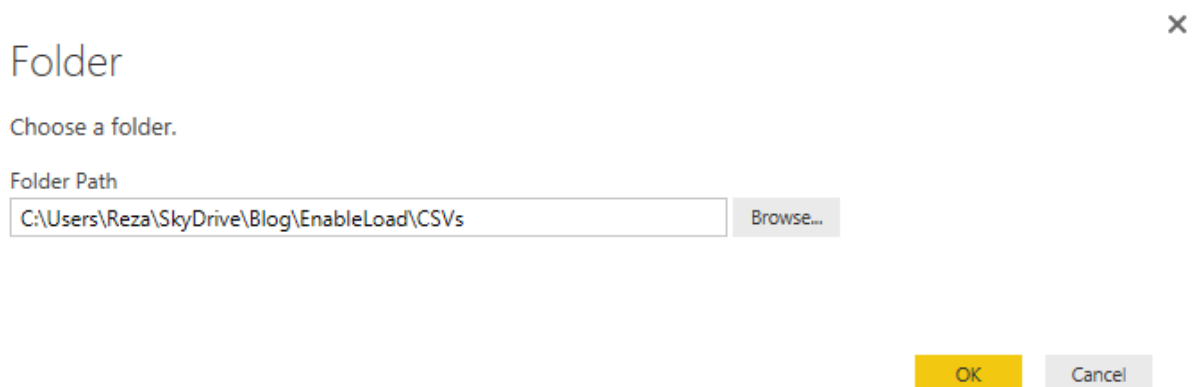
## Build the Transformations

### Get Data from Folder

Open a new Power BI file, and start by Getting Data from Folder



Enter the path of folder that contains all files (Files in this folder can be downloaded from ZIP file up in the prerequisite section of this post)



Click on Edit in the preview showed in the navigator window to open the folder content in Query Editor. As you see there are number of different files in the folder.

Untitled - Query Editor

File Home Transform Add Column View

Close & Apply Close New Source Recent Sources Enter Data Data Source Settings Manage Parameters Refresh Preview Properties Advanced Editor Manage Choose Columns Remove Columns Keep Rows Remove Rows Sort Split Column Group By

Queries [1] CSVs

fx = Folder.Files("C:\Users\Reza\SkyDrive\Blog\EnableLoad\CSVs")

|   | Content | Name                    | Extension | Date accessed         | Date modified        |
|---|---------|-------------------------|-----------|-----------------------|----------------------|
| 1 | Binary  | Courses 2016 10 01 .csv | .csv      | 11/14/2016 9:10:41 AM | 11/14/2016 9:11:36 A |
| 2 | Binary  | Courses 2016 11 01.csv  | .csv      | 11/14/2016 8:13:15 AM | 11/14/2016 8:13:15 A |
| 3 | Binary  | index.html              | .html     | 11/14/2016 9:09:39 AM | 11/14/2016 9:09:39 A |
| 4 | Binary  | New Text Document.txt   | .txt      | 11/14/2016 8:26:37 AM | 11/14/2016 8:26:37 A |
| 5 | Binary  | Sample word file.docx   | .docx     | 11/14/2016 9:10:55 AM | 11/14/2016 9:10:24 A |
| 6 | Binary  | Students 20160901.csv   | .csv      | 11/14/2016 7:59:49 AM | 11/14/2016 8:03:28 A |
| 7 | Binary  | Students 20161001.csv   | .csv      | 11/14/2016 7:59:35 AM | 11/14/2016 8:08:49 A |
| 8 | Binary  | Students 20161101.csv   | .csv      | 11/14/2016 7:59:17 AM | 11/14/2016 7:59:17 A |

Filter the Extension to only .csv. Note that both Course and Student files are .CSV files which is what we need.

Content Name Extension

1 Binary Sort Ascending

2 Binary Sort Descending

3 Binary Clear Sort

4 Binary Clear Filter

5 Binary Remove Empty

6 Binary Text Filters

7 Binary

8 Binary

Search

☒ (Select All)

☒ .csv

☐ .docx

☐ .html

☐ .txt

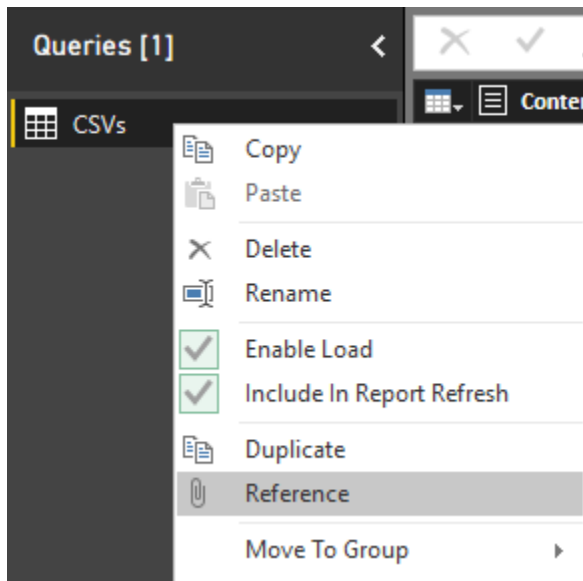
OK Cancel

Now the subset includes Course files and Students files which are different in the structure. We have to apply transformations on each set individually.

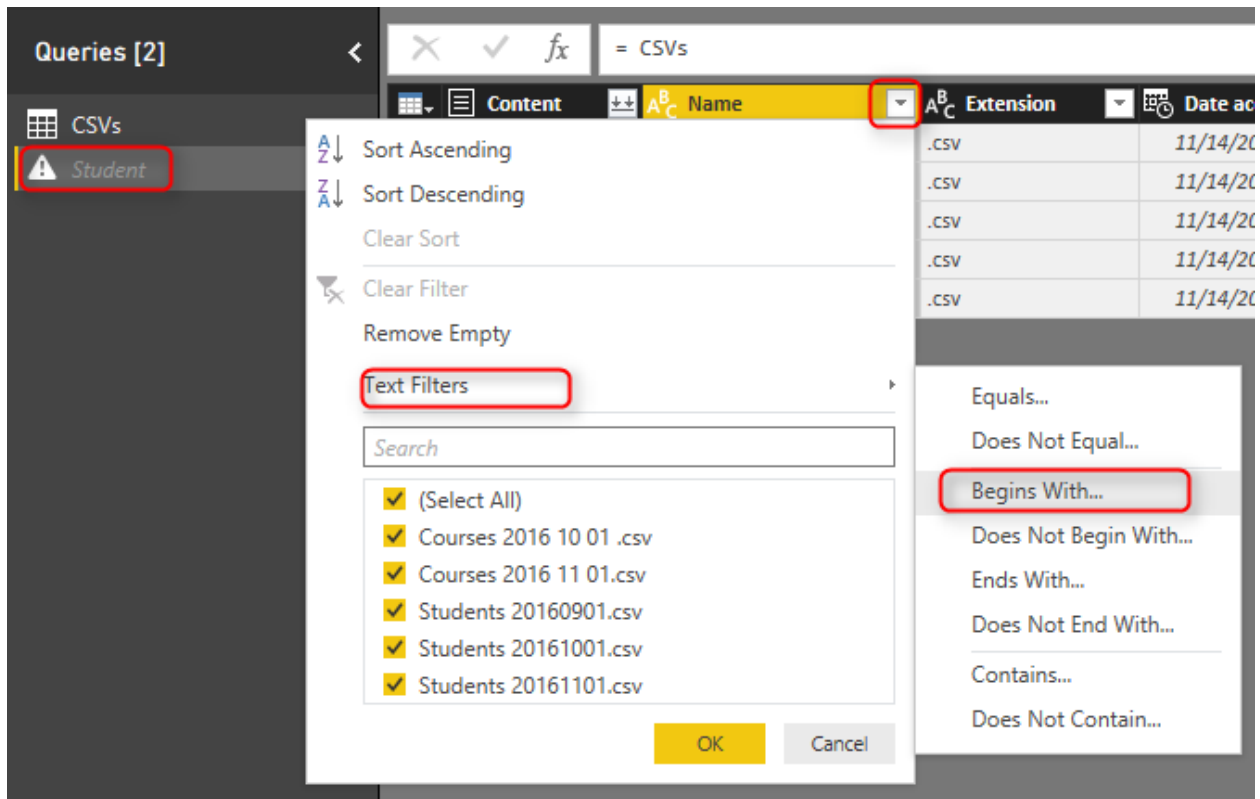
| ✕ ✓ fx |         | = Table.SelectRows(Source, each ([Extension] = ".csv")) |                                       |                       |                       |
|--------|---------|---------------------------------------------------------|---------------------------------------|-----------------------|-----------------------|
|        | Content | A <sup>B</sup> <sub>C</sub> Name                        | A <sup>B</sup> <sub>C</sub> Extension | Date accessed         | Date modified         |
| 1      | Binary  | Courses 2016 10 01 .csv                                 | .csv                                  | 11/14/2016 9:10:41 AM | 11/14/2016 9:11:36 AM |
| 2      | Binary  | Courses 2016 11 01.csv                                  | .csv                                  | 11/14/2016 8:13:15 AM | 11/14/2016 8:13:15 AM |
| 3      | Binary  | Students 20160901.csv                                   | .csv                                  | 11/14/2016 7:59:49 AM | 11/14/2016 8:03:28 AM |
| 4      | Binary  | Students 20161001.csv                                   | .csv                                  | 11/14/2016 7:59:35 AM | 11/14/2016 8:08:49 AM |
| 5      | Binary  | Students 20161101.csv                                   | .csv                                  | 11/14/2016 7:59:17 AM | 11/14/2016 7:59:17 AM |

## Students Query

Because I don't want to repeat the process of getting all files from a folder, and I want to split the data into two data sets; one for Students and another one for Courses. I'll generate a REFERENCE from the main query. Right click on the query (which called CSVs in my example), and select Reference.



This will generate a new query named as CSVs (2). This new query is NOT A COPY of first query. This is only a reference from first query. which means if the first query changes, the source for this query will also change. Rename this new query to be Student. Filter the Name column to everything starting (or begins with) "Student".



The reason that I don't type in in the search box and use Text Filters specifically is that the search box will filter the data statistically based on values that exists in the current data set. If in the future new values (file names) comes in the folder this won't consider that. However the Text Filter will apply on any data set because it will be filtered dynamically. (I'll write a post later to explain that in details). In the Filter Rows window type in Student as filter for begins with.

## Filter Rows

☒ Basic ☐ Advanced

Show rows where: Name

begins with

☒ And ☐ Or

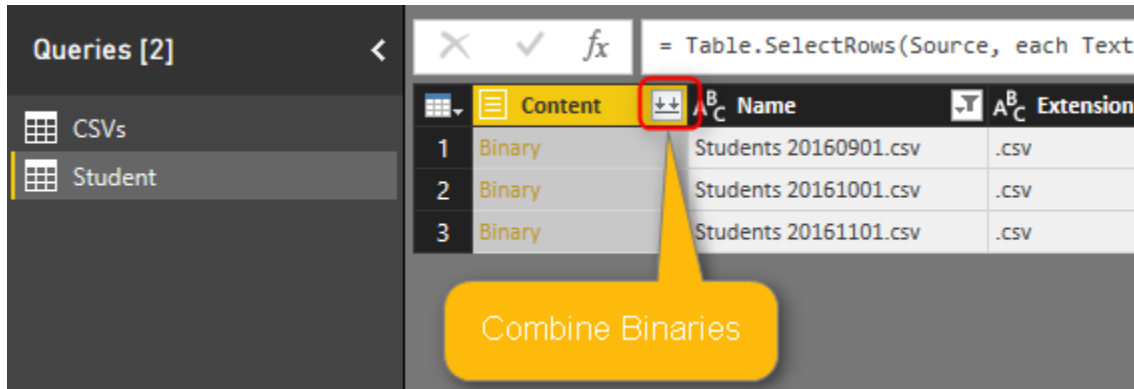
OK

Cancel

Now you will see only Students files.

|   |         | = Table.SelectRows(Source, each Text.StartsWith([Name], "Student")) |           |                       |                       |             |
|---|---------|---------------------------------------------------------------------|-----------|-----------------------|-----------------------|-------------|
|   | Content | Name                                                                | Extension | Date accessed         | Date modified         | Date create |
| 1 | Binary  | Students 20160901.csv                                               | .csv      | 11/14/2016 7:59:49 AM | 11/14/2016 8:03:28 AM | 11/14/2016  |
| 2 | Binary  | Students 20161001.csv                                               | .csv      | 11/14/2016 7:59:35 AM | 11/14/2016 8:08:49 AM | 11/14/2016  |
| 3 | Binary  | Students 20161101.csv                                               | .csv      | 11/14/2016 7:59:17 AM | 11/14/2016 7:59:17 AM | 11/14/2016  |

Click on the Combine Binaries icon on the header of Content column to combine all CSV files into one.



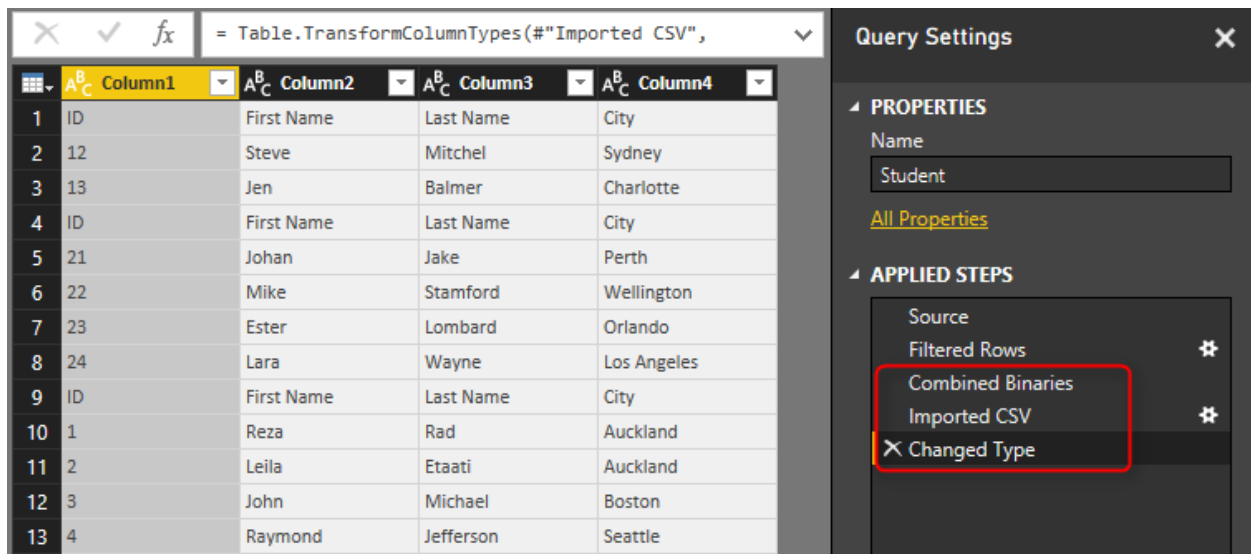
Queries [2]

- CSVs
- Student

|   |         |                       |           |
|---|---------|-----------------------|-----------|
|   | Content | Name                  | Extension |
| 1 | Binary  | Students 20160901.csv | .csv      |
| 2 | Binary  | Students 20161001.csv | .csv      |
| 3 | Binary  | Students 20161101.csv | .csv      |

Combine Binaries

After combining binaries, Power Query will also import the CSV into the table and do automatic data type conversion. You can see all three steps in the Query Editor.



|    |         | = Table.TransformColumnTypes(#"Imported CSV", |           |             |  |
|----|---------|-----------------------------------------------|-----------|-------------|--|
|    | Column1 | Column2                                       | Column3   | Column4     |  |
| 1  | ID      | First Name                                    | Last Name | City        |  |
| 2  | 12      | Steve                                         | Mitchel   | Sydney      |  |
| 3  | 13      | Jen                                           | Balmer    | Charlotte   |  |
| 4  | ID      | First Name                                    | Last Name | City        |  |
| 5  | 21      | Johan                                         | Jake      | Perth       |  |
| 6  | 22      | Mike                                          | Stamford  | Wellington  |  |
| 7  | 23      | Ester                                         | Lombard   | Orlando     |  |
| 8  | 24      | Lara                                          | Wayne     | Los Angeles |  |
| 9  | ID      | First Name                                    | Last Name | City        |  |
| 10 | 1       | Reza                                          | Rad       | Auckland    |  |
| 11 | 2       | Leila                                         | Etaati    | Auckland    |  |
| 12 | 3       | John                                          | Michael   | Boston      |  |
| 13 | 4       | Raymond                                       | Jefferson | Seattle     |  |

Query Settings

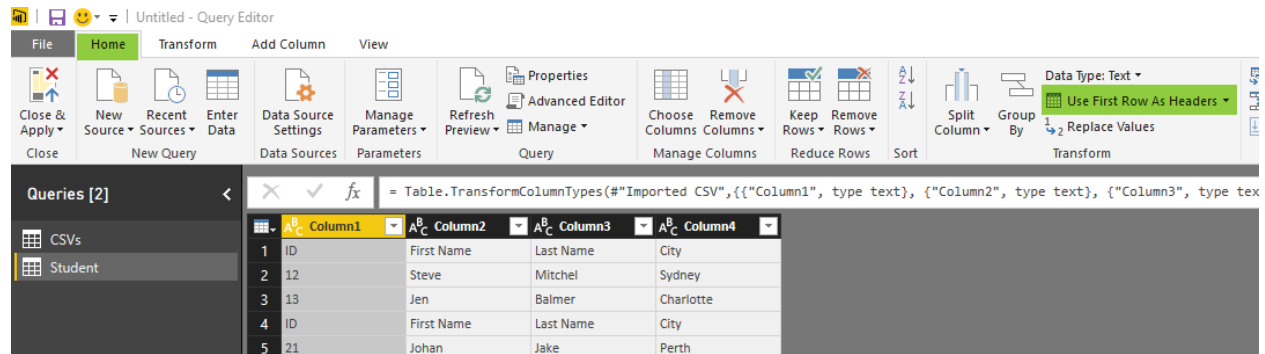
PROPERTIES

Name: Student

APPLIED STEPS

- Source
- Filtered Rows
- Combined Binaries
- Imported CSV
- Changed Type

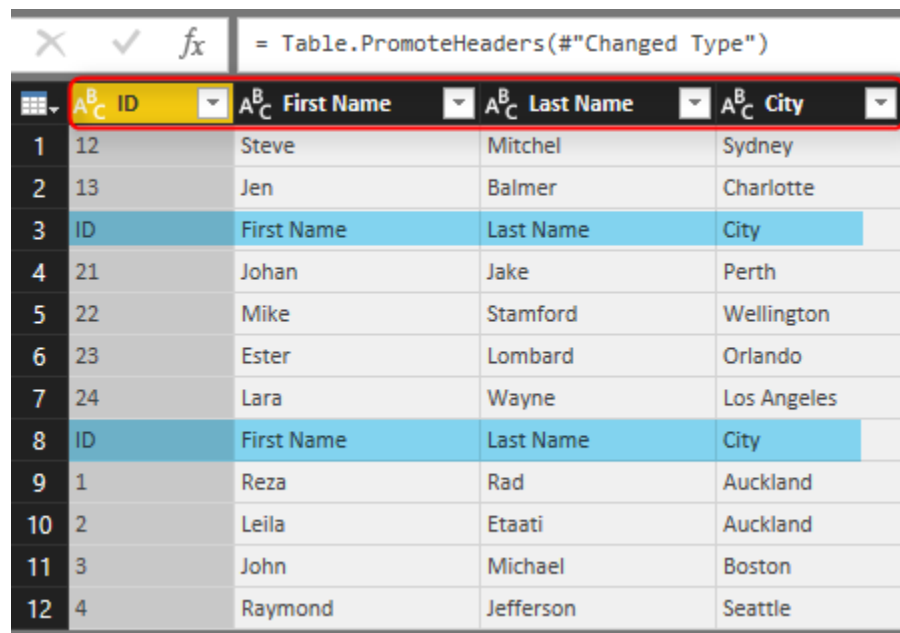
The data table needs couple of changes before be ready. First; set the column names. The first row has column names. so use the menu option of "Use First Row As Headers" to promote the first row to be column headers.



Queries [2] < X ✓ fx = Table.TransformColumnTypes(#"Imported CSV",{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}})

|   | Column1 | Column2    | Column3   | Column4   |
|---|---------|------------|-----------|-----------|
| 1 | ID      | First Name | Last Name | City      |
| 2 | 12      | Steve      | Mitchel   | Sydney    |
| 3 | 13      | Jen        | Balmer    | Charlotte |
| 4 | ID      | First Name | Last Name | City      |
| 5 | 21      | Johan      | Jake      | Perth     |

This simply brings column headers;



X ✓ fx = Table.PromoteHeaders(#"Changed Type")

|    | ID | First Name | Last Name | City        |
|----|----|------------|-----------|-------------|
| 1  | 12 | Steve      | Mitchel   | Sydney      |
| 2  | 13 | Jen        | Balmer    | Charlotte   |
| 3  | ID | First Name | Last Name | City        |
| 4  | 21 | Johan      | Jake      | Perth       |
| 5  | 22 | Mike       | Stamford  | Wellington  |
| 6  | 23 | Ester      | Lombard   | Orlando     |
| 7  | 24 | Lara       | Wayne     | Los Angeles |
| 8  | ID | First Name | Last Name | City        |
| 9  | 1  | Reza       | Rad       | Auckland    |
| 10 | 2  | Leila      | Etaati    | Auckland    |
| 11 | 3  | John       | Michael   | Boston      |
| 12 | 4  | Raymond    | Jefferson | Seattle     |

Also you need to remove all extra header rows from the combined data set. There are some rows with "ID,First Name, Last Name, and City" as their values which should be removed from the data set. You can simply remove them with a Text Filter of Does Not Equal to on the ID column. values should not be equal to "ID" because every row with "ID" in the first column's value is a header row and should be removed.



## Filter Rows

☒ Basic ☐ Advanced

Show rows where: ID

does not equal ID

☒ And ☐ Or

Type or select a value

OK

Cancel

Now you have your cleaned data set for Students. Note that I have also applied a data type conversion for column ID to type whole number;

|    | ID | First Name | Last Name | City        |
|----|----|------------|-----------|-------------|
| 1  | 12 | Steve      | Mitchel   | Sydney      |
| 2  | 13 | Jen        | Balmer    | Charlotte   |
| 3  | 21 | Johan      | Jake      | Perth       |
| 4  | 22 | Mike       | Stamford  | Wellington  |
| 5  | 23 | Ester      | Lombard   | Orlando     |
| 6  | 24 | Lara       | Wayne     | Los Angeles |
| 7  | 1  | Reza       | Rad       | Auckland    |
| 8  | 2  | Leila      | Etaati    | Auckland    |
| 9  | 3  | John       | Michael   | Boston      |
| 10 | 4  | Raymond    | Jefferson | Seattle     |

## Courses Query

The other entity is Course which we need to create as a reference from the CSVs query. Create another reference from CSVs, and name it Course.

| Queries [3] |  | = CSVs   |                         |
|-------------|--|----------|-------------------------|
| CSVs        |  | Content  | Name                    |
| Student     |  | 1 Binary | Courses 2016 10 01 .csv |
| Course      |  | 2 Binary | Courses 2016 11 01 .csv |
|             |  | 3 Binary | Students 20160901.csv   |
|             |  | 4 Binary | Students 20161001.csv   |
|             |  | 5 Binary | Students 20161101.csv   |

Create a Text Filter for Begins with on the Name Column with the value of: "Course"

## Filter Rows

☒ Basic ☐ Advanced

Show rows where: Name

begins with

And the you'll have only Course files. Combine Binaries on that.

|   | A <sub>C</sub> Column1 | A <sub>C</sub> Column2 | A <sub>C</sub> Column3     | A <sub>C</sub> Column4 | A <sub>C</sub> Column5 |
|---|------------------------|------------------------|----------------------------|------------------------|------------------------|
| 1 | Course Number          | Department             | title                      | Capacity               | Date                   |
| 2 | 2                      | BI                     | SSIS Training              | 30                     | 2016/10/1              |
| 3 | 3                      | Database               | Execution Plans            | 30                     | 2016/10/15             |
| 4 | Course Number          | Department             | title                      | Capacity               | Date                   |
| 5 | 12                     | BI                     | Power BI Training          | 30                     | 2016/11/1              |
| 6 | 13                     | Database               | High Availability in a Day | 30                     | 2016/12/1              |

Same as Student query, apply transformations in this order;

- Promote headers with "Use First Row As Headers"
- Create a Text Filter on the Course Number Column that values does not equal to "Course Number".
- Apply data type conversion on Course Number and Capacity to Whole Number and Date to Date.

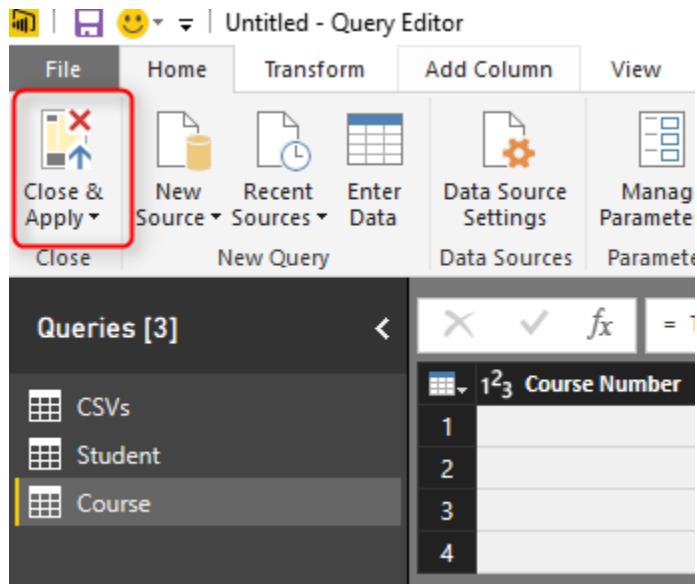
Here is the final data set for course;

|   | 1 <sup>2</sup> <sub>3</sub> Course Number | A <sub>C</sub> Department | A <sub>C</sub> title       | 1 <sup>2</sup> <sub>3</sub> Capacity | Date       |
|---|-------------------------------------------|---------------------------|----------------------------|--------------------------------------|------------|
| 1 | 2                                         | BI                        | SSIS Training              | 30                                   | 10/1/2016  |
| 2 | 3                                         | Database                  | Execution Plans            | 30                                   | 10/15/2016 |
| 3 | 12                                        | BI                        | Power BI Training          | 30                                   | 11/1/2016  |
| 4 | 13                                        | Database                  | High Availability in a Day | 30                                   | 12/1/2016  |

## Default Behavior: Enable Load

Now to see the problem, without any changes in the default load behavior, Close the Query Editor and Apply changes.



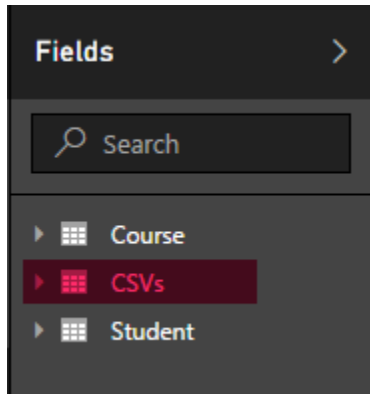


You will see that after the refresh, three queries loads in the model; Student, Course, and CSVs.

## Apply Query Changes

- CSVs  
Creating connection in model...
- Student  
Creating connection in model...
- Course  
Creating connection in model...

Student and Course are expected tables in the model. However CSVs is not useful. we already fetched everything we wanted from the query, and this is used as an intermediate query for loading Student and Course. Having CSVs as a separate table in our model has two problems;

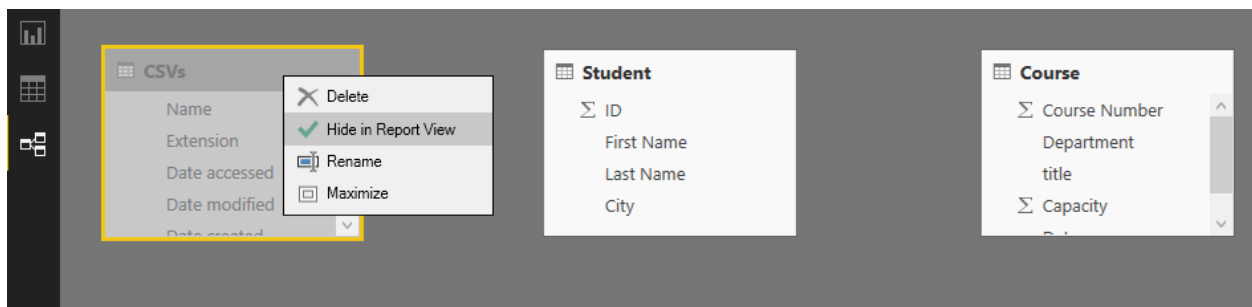


- 27. one extra table; confusion for users
- 28. extra memory consumption

The main issue is the memory consumption for this extra table. and normally memory consumption will reduce the performance and brings heavier load to the model. In this example CSVs table only has few rows. but this is just a sample data, in real world examples you might have intermediate tables with millions of rows. You need to remove every unused table from the Power BI model to enhance the memory consumption.

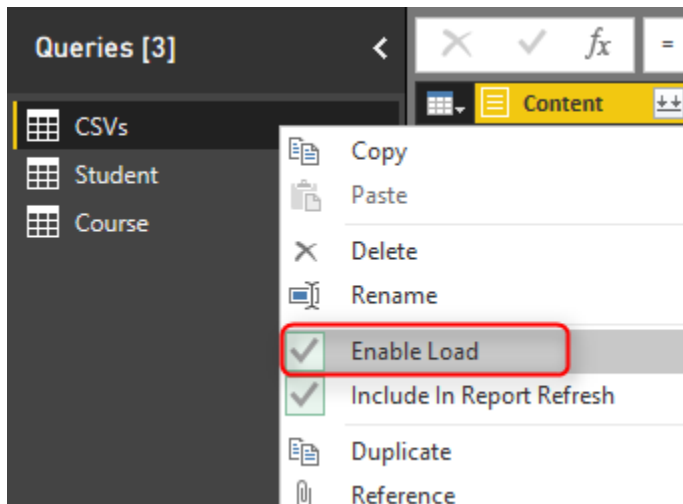
## What about Hiding from Report?

The first issue “Confusion for users” can be solved by hiding the table from report view. You can do that in Relationship tab of the model, right click on the CSVs table, and click on Hide from report view. This method HIDEs the table from report view. However the table still exists, and it consumes memory! It is just hidden. Hiding a table from report view is good for tables that you need for the modeling. For example a relationship table (that creates many to many relationship) should be kept in the model but be hidden. because it creates the relationship, but it is not useful for the report viewer. Any other tables that is not used for relationship and is not used for reporting should be removed before loading into the model.

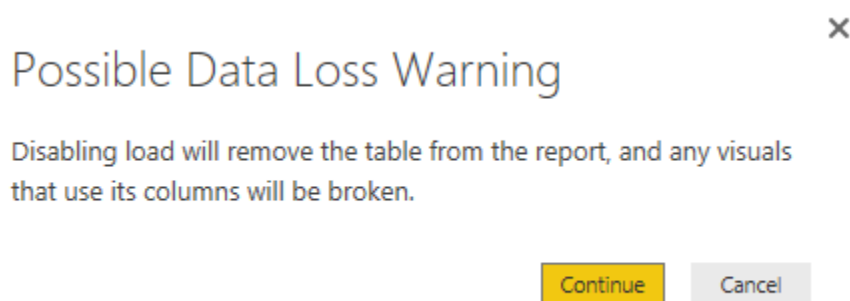


## Disable Load to Save Memory

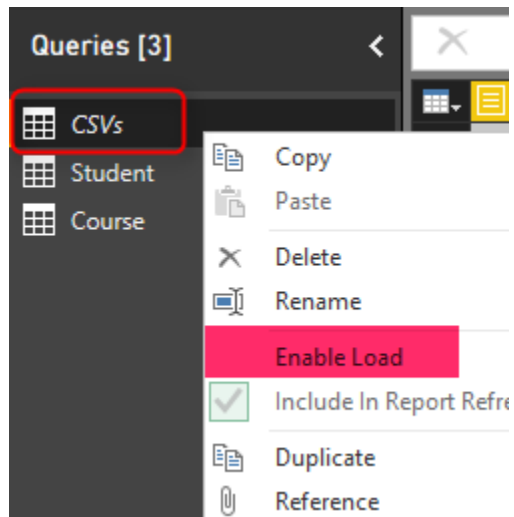
Go back to Query Editor, and right click on CSVs query. You will see that by default every query is checked as Enable Load.



Click on the Enable Load to disable it. You will see a message saying that any visuals attached to this query will not work after this change. Which is totally fine, because this query is not used (and won't be) for any visualization. The message is: "Disabling load will remove the table from the report, and any visuals that use its columns will be broken."

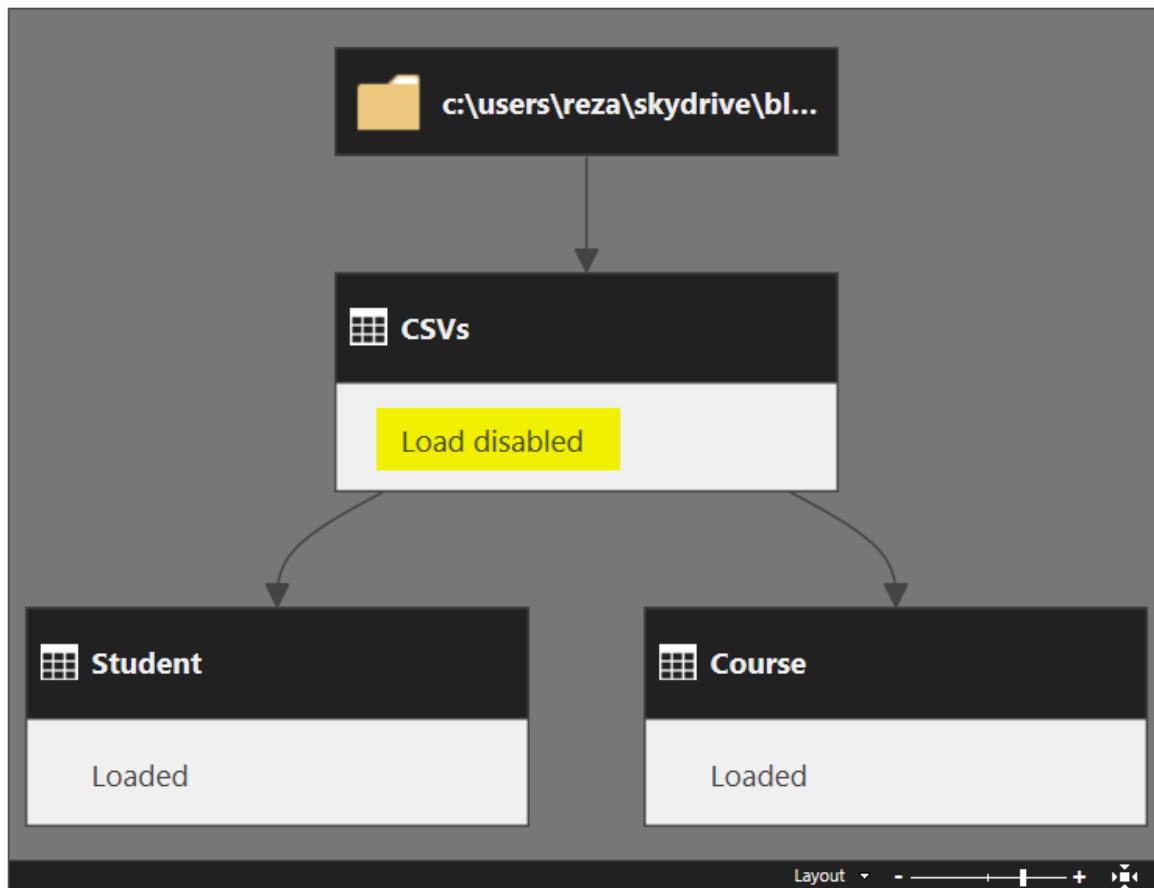


Click on Continue, and you will see the load is disabled for the query now. the query name will be also in *Italic* font illustrating that it won't be loaded into the model. Note that query will be refreshed and if new files comes to the directory it will pass it on to Course and Student queries. It just won't be loaded into the model itself.

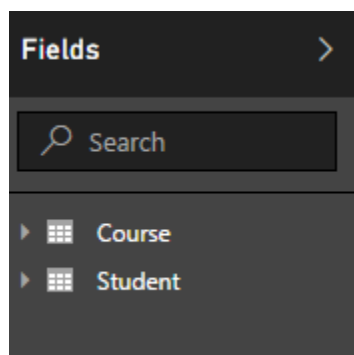


You can also see in the View tab of Query Editor, on the Query Dependencies that this query is the source of the other two queries. and in that view it also shows which query is load disabled.

## Query Dependencies

[Close](#)

Now close and apply the Query Editor. This time you see that CSVs query won't be loaded into the model. Your relationship tab and the report tab only contains two tables; Course, and Student.

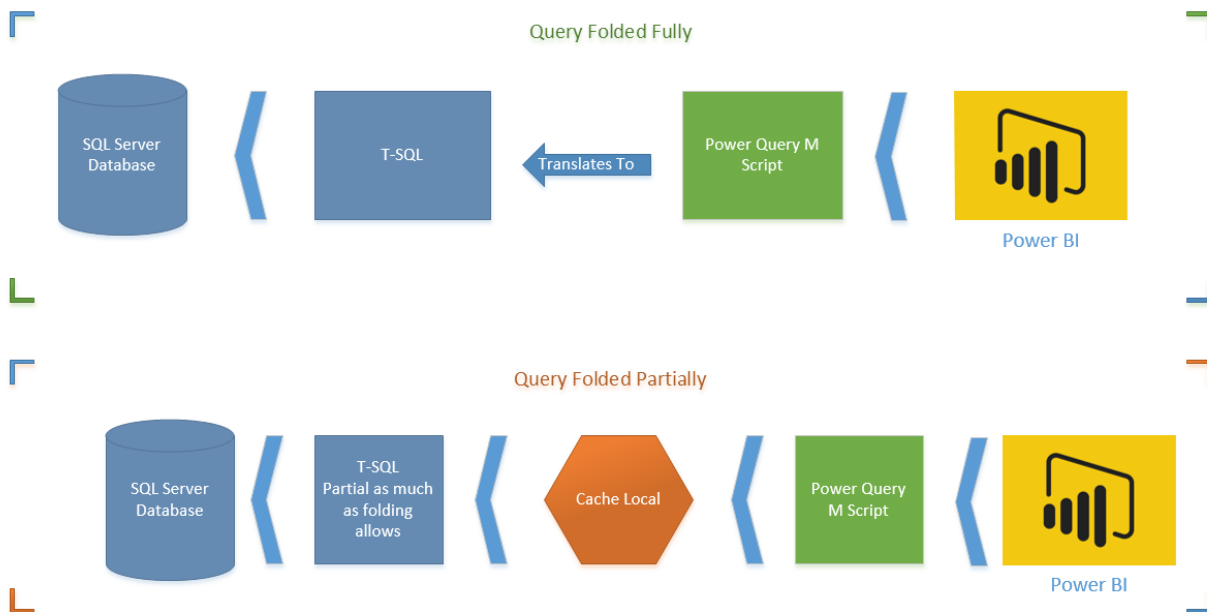


## Summary

In summary you've seen how an easy tip can save the memory and enhance the performance of Power BI model. Always remember that Query Editor is your ETL (Extract, Transform, Load) engine. It will apply all transformations before loading the data into the model, but once it finished the transformation all queries will be loaded into the model and they take memory. By default all queries are Enabled to Load into the model. Simply change that for queries that is not required in the model.

# Not Folding; the Black Hole of Power Query Performance

Published Date: November 15, 2016



Have you ever thought does your Power Query Transformations happens in the data source side (server side), or on the local memory (client side)? When you use a relational data source, query can be run on the data source, but it depends on transformations. Some transformations can be translated to query language of the data source, some not. as an example; recently (I believe from last few releases) Power BI Desktop added a feature called Merge Columns. Merge Columns concatenate columns to each other to either create a new column, or replace them with the new concatenated result. Previously you could do the concatenation with adding concatenation simply with **&** character, what you've done was adding a new custom column, and writing M expression to concatenate columns. Now with the new Merge Column this are much easier, you select columns and apply Merge Columns. This easiness does come with a price, a high price I'd say, price of reducing the performance of Power Query and as a result Power BI! Merge Columns doesn't support query folding and it means it will affect performance badly. In this post I'll show you how this cause the performance issue, and how it can be solved. Note that Merge Columns is an example here ,this situation might happens with some other transformations as well. If you like to learn more about Power BI; read the [Power BI online book from Rookie to Rock Star](#).

## Query Folding

I can't start talking about the issue without explaining what Query Folding is, so let's start with that. Query Folding means translating Power Query (M) transformations into native query language of the data source (for example T-SQL). In other words; when you run Power Query script on top of a SQL Server database, query folding will translate the M script into T-SQL statements, and fetch the final results.

Here is an example of M Script:

```
let
 Source = Sql.Databases("."),
 AdventureWorks2012 = Source{[Name="AdventureWorks2012"]}[Data],
 Sales_SalesOrderHeader = AdventureWorks2012{[Schema="Sales",Item="SalesOrderHeader"]}[Data],
 #"Added Conditional Column" = Table.AddColumn(Sales_SalesOrderHeader, "Custom", each if
[SubTotal] >= 100 then "0" else "1")
in
 #"Added Conditional Column"
```

And here is the folded version of that translated to native T-SQL query:

```
select [].[SalesOrderID] as [SalesOrderID],
 [].[RevisionNumber] as [RevisionNumber],
 [].[OrderDate] as [OrderDate],
 [].[DueDate] as [DueDate],
 [].[ShipDate] as [ShipDate],
 [].[Status] as [Status],
 [].[OnlineOrderFlag] as [OnlineOrderFlag],
 [].[SalesOrderNumber] as [SalesOrderNumber],
 [].[PurchaseOrderNumber] as [PurchaseOrderNumber],
 [].[AccountNumber] as [AccountNumber],
 [].[CustomerID] as [CustomerID],
 [].[SalesPersonID] as [SalesPersonID],
 [].[TerritoryID] as [TerritoryID],
 [].[BillToAddressID] as [BillToAddressID],
 [].[ShipToAddressID] as [ShipToAddressID],
 [].[ShipMethodID] as [ShipMethodID],
 [].[CreditCardID] as [CreditCardID],
 [].[CreditCardApprovalCode] as [CreditCardApprovalCode],
 [].[CurrencyRateID] as [CurrencyRateID],
 [].[SubTotal] as [SubTotal],
 [].[TaxAmt] as [TaxAmt],
 [].[Freight] as [Freight],
 [].[TotalDue] as [TotalDue],
 [].[Comment] as [Comment],
 [].[rowguid] as [rowguid],
```



```

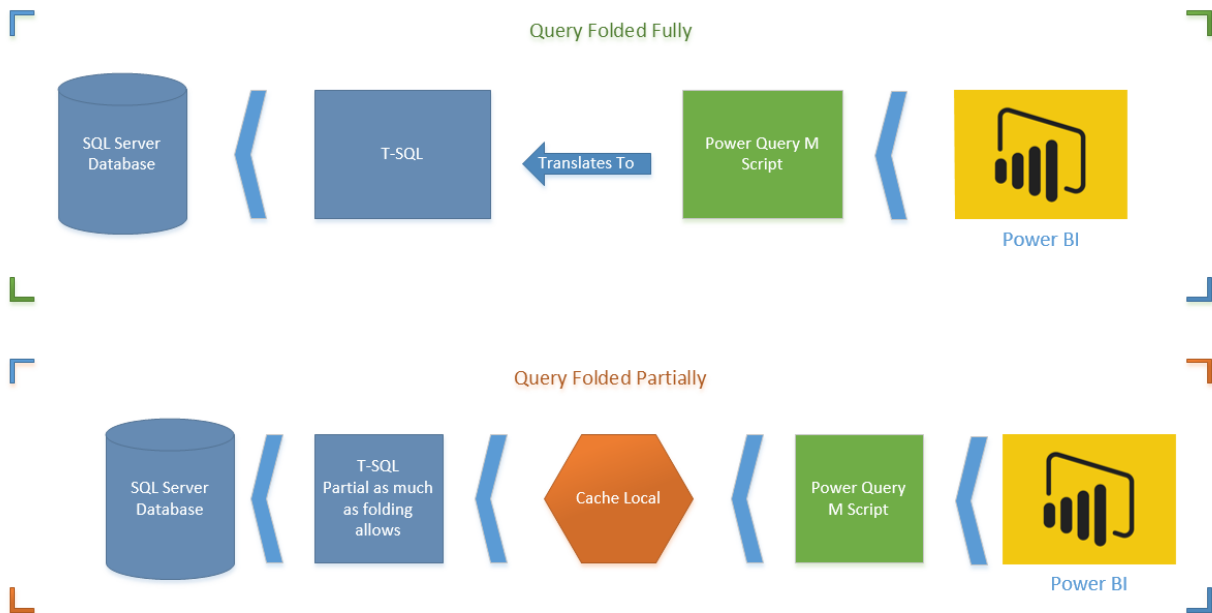
[].[ModifiedDate] as [ModifiedDate],
case
 when [].[SubTotal] >= 100
 then '0'
 else '1'
end as [Custom]
from
(
 select [$Table].[SalesOrderID] as [SalesOrderID],
 [$Table].[RevisionNumber] as [RevisionNumber],
 [$Table].[OrderDate] as [OrderDate],
 [$Table].[DueDate] as [DueDate],
 [$Table].[ShipDate] as [ShipDate],
 [$Table].[Status] as [Status],
 [$Table].[OnlineOrderFlag] as [OnlineOrderFlag],
 [$Table].[SalesOrderNumber] as [SalesOrderNumber],
 [$Table].[PurchaseOrderNumber] as [PurchaseOrderNumber],
 [$Table].[AccountNumber] as [AccountNumber],
 [$Table].[CustomerID] as [CustomerID],
 [$Table].[SalesPersonID] as [SalesPersonID],
 [$Table].[TerritoryID] as [TerritoryID],
 [$Table].[BillToAddressID] as [BillToAddressID],
 [$Table].[ShipToAddressID] as [ShipToAddressID],
 [$Table].[ShipMethodID] as [ShipMethodID],
 [$Table].[CreditCardID] as [CreditCardID],
 [$Table].[CreditCardApprovalCode] as [CreditCardApprovalCode],
 [$Table].[CurrencyRateID] as [CurrencyRateID],
 [$Table].[SubTotal] as [SubTotal],
 [$Table].[TaxAmt] as [TaxAmt],
 [$Table].[Freight] as [Freight],
 [$Table].[TotalDue] as [TotalDue],
 [$Table].[Comment] as [Comment],
 convert(nvarchar(max), [$Table].[rowguid]) as [rowguid],
 [$Table].[ModifiedDate] as [ModifiedDate]
 from [Sales].[SalesOrderHeader] as [$Table]
) as [.]

```

You can see as an example how the conditional column script in M translated to Case statement in T-SQL.

## Is Query Folding Good or Bad?

Good obviously. Why? because performance is much higher to run transformations on billions of records in the data source, rather than bringing millions of records into cache and applying some transformations on it.

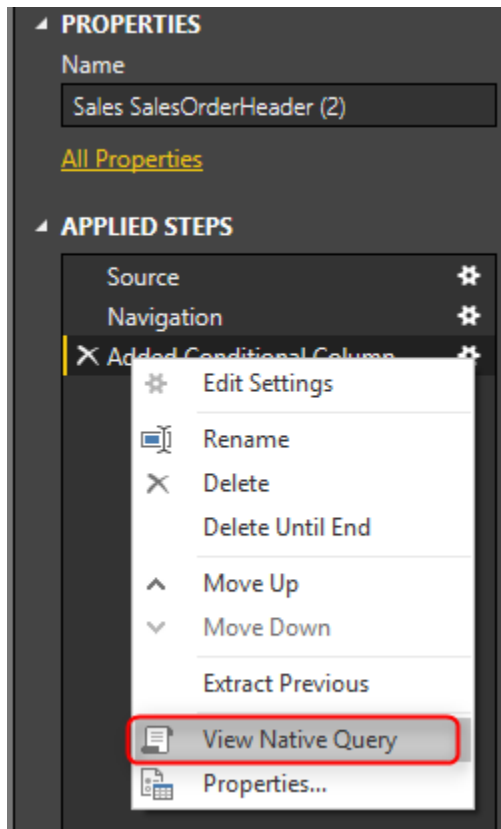


The first diagram shows a M script fully folded (translated to T-SQL). This is the best situation. Server side operation apply all transformations on the data set and returns only desired result set.

The second diagram shows Query Folding partially supported (only up to specific step). In this case T-SQL brings the data before that step. and data will be loaded in the local cache, and rest of transformations happens on M engine side. You have to avoid this option as much as possible.

## Can I see the Native Query?

The question that might comes into your mind right now is that; Can I see the Native Query that M script translates to it? The answer is Yes. If Query Folding is supported on a step, you can right click on that step and click on View Native Query.



## So Why Not Query Folding?

Query Folding is enabled by default. However in some cases it is not supported. For example if you are doing some transformations on a SQL Server table in Power Query and then join it with a web query query folding stops from the time you bring the external data source. That means transformations will happen on the data of SQL Server table. then before joining to web query it will be fetched into cache and then rest of steps happens by M engine. You would need to bring data from different data sources in Power BI, and this is the ability that Power Query gives to you. So sometimes you have to step beyond query folding, and there might be no better way of doing that.

There are also some Transformations in Power Query that Query Folding doesn't support them. Example? Merge Columns! Fortunately there are workarounds for this situation. Let's dig into this more in details.

## Example: Merge Columns

Merge Columns concatenates columns to each other in the order of selection of columns. You can also specify the delimiter character(s). So simply if you want to create

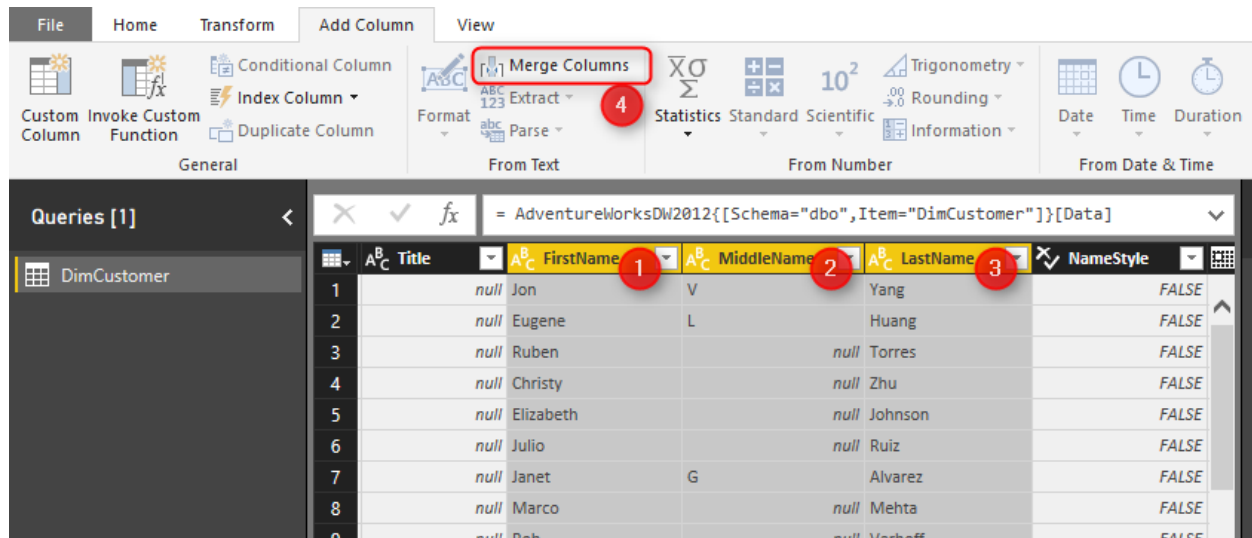
a full name from first name and last name, you can select them in right order, and from either Transform tab, or Add Column tab choose Merge Columns. Let's see this through an example;

## Prerequisite for running the example

You need to have AdvanetureWorksDW database installed on SQL Server. or alternatively you can use any table in SQL Server that has two string columns which can be concatenated.

## Merge Column Transformation

Create a new Power BI file in Power BI Desktop, and Get Data from SQL Server with Import Data Mode. Get Data from DimCustomer table only, and click on Edit. When in Query Editor. Select First Name, Middle Name, and Last Name in the correct order, and then from Add Column Select Merge Columns.



|   | Title | 1<br>FirstName | 2<br>MiddleName | 3<br>LastName | NameStyle |
|---|-------|----------------|-----------------|---------------|-----------|
| 1 | null  | Jon            | V               | Yang          | FALSE     |
| 2 | null  | Eugene         | L               | Huang         | FALSE     |
| 3 | null  | Ruben          | null            | Torres        | FALSE     |
| 4 | null  | Christy        | null            | Zhu           | FALSE     |
| 5 | null  | Elizabeth      | null            | Johnson       | FALSE     |
| 6 | null  | Julio          | null            | Ruiz          | FALSE     |
| 7 | null  | Janet          | G               | Alvarez       | FALSE     |
| 8 | null  | Marco          | null            | Mehta         | FALSE     |
| 9 | null  | Bob            | null            | Verhoff       | FALSE     |

In the Merge Columns Specify the Separator to be space, and name the new column to be Full Name.

## Merge Columns

Choose how to merge the selected columns.

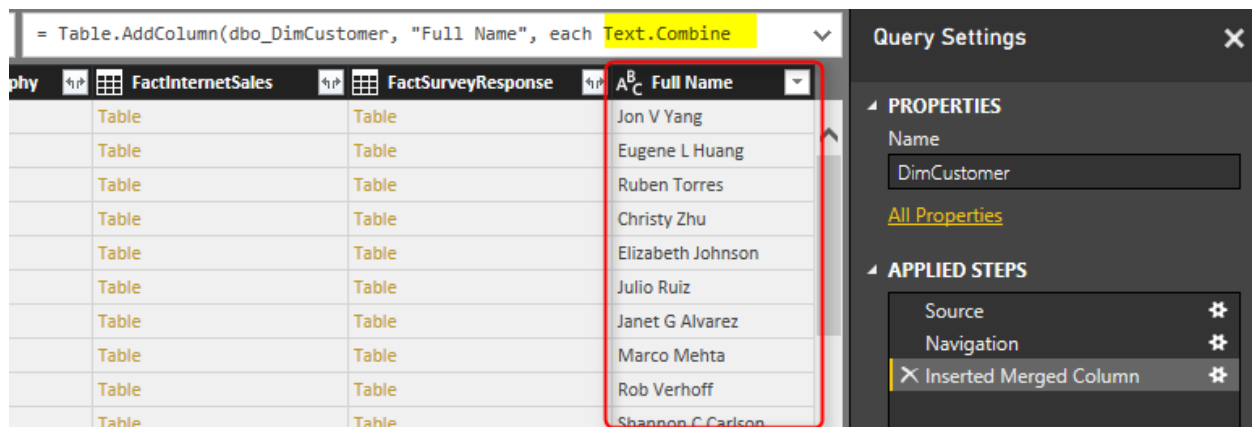
Separator

Space

New column name (optional)

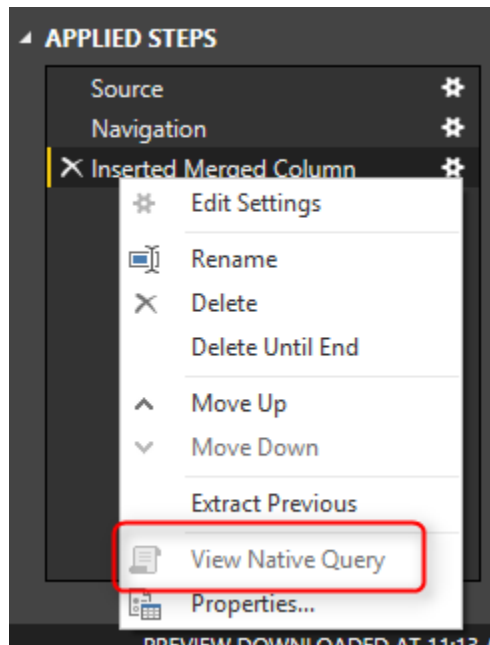
Full Name

You will see the new column generates simply and adds to the end of all columns. You also may notice that Merge Columns uses Text.Combine Power Query function to concatenate columns to each other.



The screenshot shows the Power BI Merge Columns interface. The formula bar at the top displays the M code: `= Table.AddColumn(dbo_DimCustomer, "Full Name", each Text.Combine`. Below the formula bar, a table with columns `FactInternetSales` and `FactSurveyResponse` is shown. A new column, `Full Name`, has been added to the end of the table. The `Full Name` column contains the concatenated values of the two source columns, such as "Jon V Yang", "Eugene L Huang", "Ruben Torres", "Christy Zhu", "Elizabeth Johnson", "Julio Ruiz", "Janet G Alvarez", "Marco Mehta", "Rob Verhoff", and "Shannon C Carlson". The `Full Name` column is highlighted with a red box. On the right side, the `Query Settings` pane is open, showing the `PROPERTIES` section with the `Name` set to `DimCustomer` and the `APPLIED STEPS` section with the `Inserted Merged Column` step selected.

Now to see the problem with Query Folding, right click on Inserted Merge Column step in Applied Steps section. You will see that View Native Query is disabled.

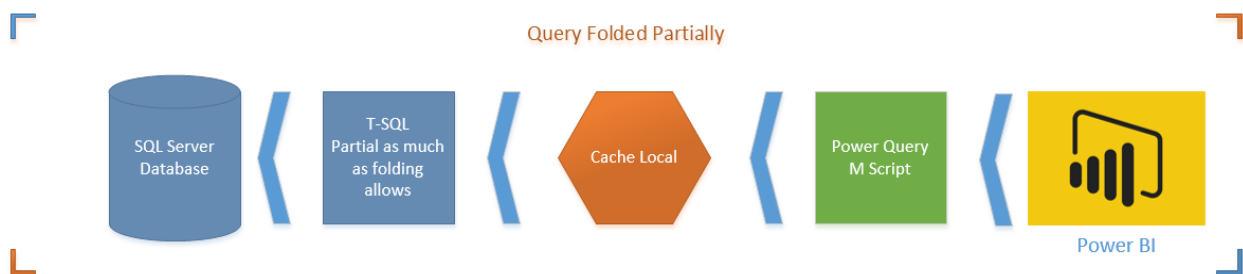


Rule of thumb is that: When View Native Query is not enabled, that step won't be folded! that means the data will be loaded into cache up the step before this step. and then rest of the operation will happen locally. To understand how it works, right click on the step before which was Navigation. You will see the View Native Query. Click on that, and you can see the T-SQL query for that which is as below;

```
select [$Table].[CustomerKey] as [CustomerKey],
 [$Table].[GeographyKey] as [GeographyKey],
 [$Table].[CustomerAlternateKey] as [CustomerAlternateKey],
 [$Table].[Title] as [Title],
 [$Table].[FirstName] as [FirstName],
 [$Table].[MiddleName] as [MiddleName],
 [$Table].[LastName] as [LastName],
 [$Table].[NameStyle] as [NameStyle],
 [$Table].[BirthDate] as [BirthDate],
 [$Table].[MaritalStatus] as [MaritalStatus],
 [$Table].[Suffix] as [Suffix],
 [$Table].[Gender] as [Gender],
 [$Table].[EmailAddress] as [EmailAddress],
 [$Table].[YearlyIncome] as [YearlyIncome],
 [$Table].[TotalChildren] as [TotalChildren],
 [$Table].[NumberChildrenAtHome] as [NumberChildrenAtHome],
 [$Table].[EnglishEducation] as [EnglishEducation],
 [$Table].[SpanishEducation] as [SpanishEducation],
 [$Table].[FrenchEducation] as [FrenchEducation],
 [$Table].[EnglishOccupation] as [EnglishOccupation],
 [$Table].[SpanishOccupation] as [SpanishOccupation],
 [$Table].[FrenchOccupation] as [FrenchOccupation],
```

```
[<table>.<table>] as [<table>],
[<table>].[NumberCarsOwned] as [<table>],
[<table>].[AddressLine1] as [<table>],
[<table>].[AddressLine2] as [<table>],
[<table>].[Phone] as [<table>],
[<table>].[DateFirstPurchase] as [<table>],
[<table>].[CommuteDistance] as [<table>],
[<table>].[FullName] as [<table>]
from [dbo].[DimCustomer] as [<table>]
```

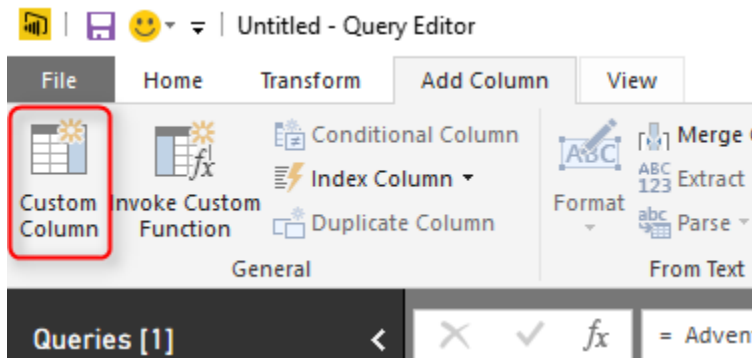
You can see that this is simple query from DimCustomer Table. What will happen here in this scenario is that Power Query cannot translate Text.Combine to T-SQL Query. So data up to step before will be loaded into the cache. It means the query for step before (which is above query) will run on the database server, the result will come to the cache, and then Text.Combine will happen on the local memory in cache. Here is a diagram of how it works;



In this example the data set is so small, but if data set be big, then not folding cause performance issues. It is taking much longer to load the whole data set into cache and then apply transformations, rather than doing transformations in the data source, and just loading the result into Power BI.

## Solution: Simple Concatenate with Add Column

Now remove the step for Inserted Merged Column, and go to Add Column Tab, and select Custom Column



In the Add Custom Column write below expression to generate Full Name;

```
=
[FirstName]
&
" "
&
(if [MiddleName]=null then "" else [MiddleName])
&
" "
&[LastName]
```

## Add Custom Column

New column name

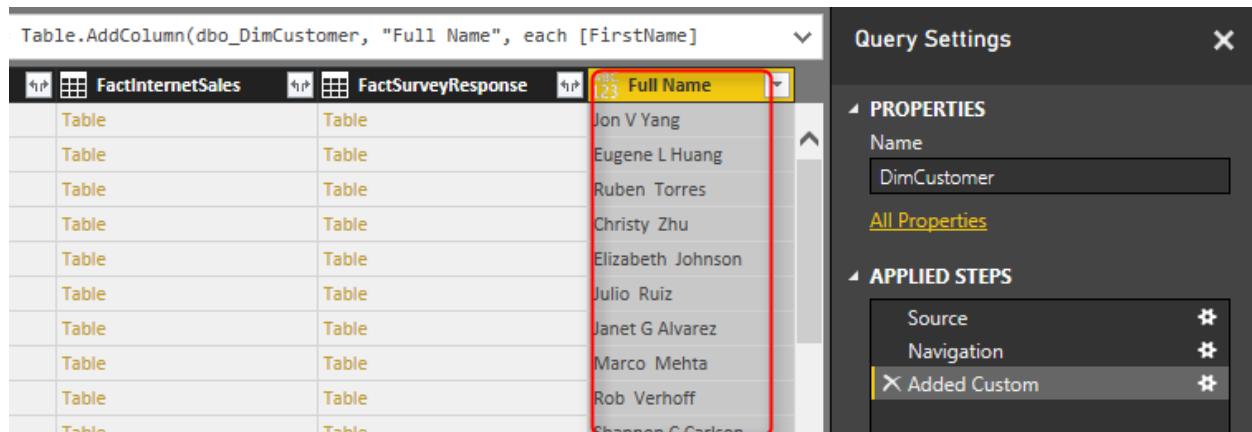
Full Name

Custom column formula:

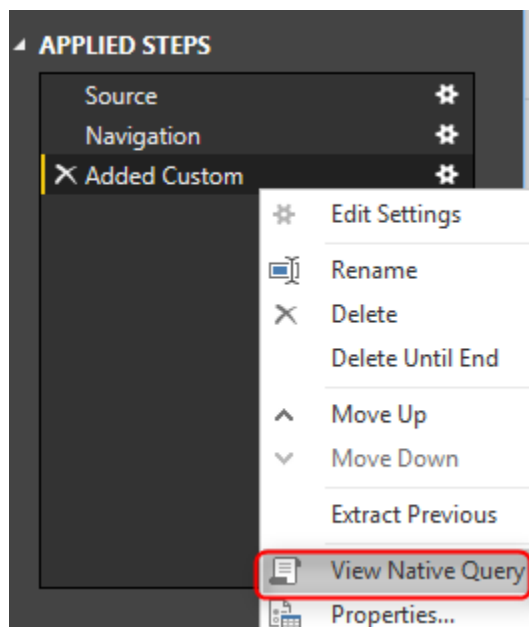
```
=
[FirstName]
&
" "
&
(if [MiddleName]=null then "" else [MiddleName])
&
" "
&[LastName]
```

This expression used the concatenation character which is **&**. and also checked if Middle Name is null or not. Result in Power Query side is the same, and it generates the Full Name column like the previous example;





However it is different for Query Folding. Right Click on the Added Custom step, and you will see the Native Query this time.



Query is simply the same with a new concatenated column added.

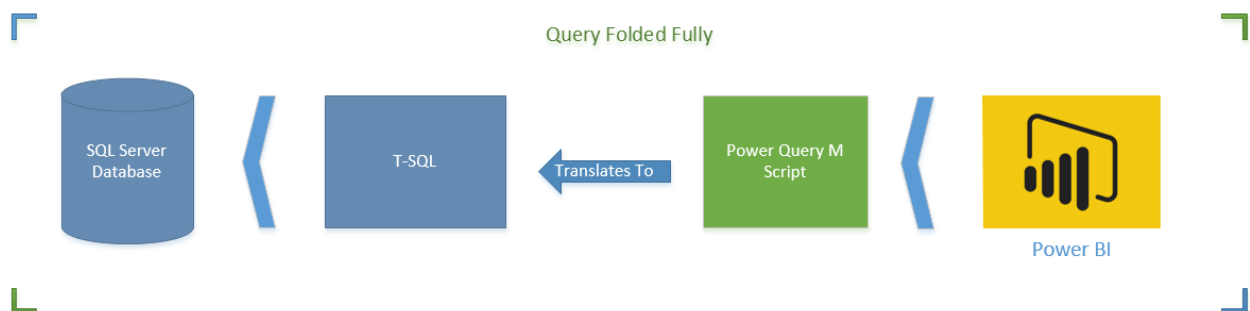
```
select [].[CustomerKey] as [CustomerKey],
 [].[GeographyKey] as [GeographyKey],
 [].[CustomerAlternateKey] as [CustomerAlternateKey],
 [].[Title] as [Title],
 [].[FirstName] as [FirstName],
 [].[MiddleName] as [MiddleName],
 [].[LastName] as [LastName],
 [].[NameStyle] as [NameStyle],
 [].[BirthDate] as [BirthDate],
 [].[MaritalStatus] as [MaritalStatus],
 [].[Suffix] as [Suffix],
```

```

[].[Gender] as [Gender],
[].[EmailAddress] as [EmailAddress],
[].[YearlyIncome] as [YearlyIncome],
[].[TotalChildren] as [TotalChildren],
[].[NumberChildrenAtHome] as [NumberChildrenAtHome],
[].[EnglishEducation] as [EnglishEducation],
[].[SpanishEducation] as [SpanishEducation],
[].[FrenchEducation] as [FrenchEducation],
[].[EnglishOccupation] as [EnglishOccupation],
[].[SpanishOccupation] as [SpanishOccupation],
[].[FrenchOccupation] as [FrenchOccupation],
[].[HouseOwnerFlag] as [HouseOwnerFlag],
[].[NumberCarsOwned] as [NumberCarsOwned],
[].[AddressLine1] as [AddressLine1],
[].[AddressLine2] as [AddressLine2],
[].[Phone] as [Phone],
[].[DateFirstPurchase] as [DateFirstPurchase],
[].[CommuteDistance] as [CommuteDistance],
[].[FullName] as [FullName],
((([].[FirstName] + ' ') + (case
 when [].[MiddleName] is null
 then ''
 else [].[MiddleName]
end)) + ' ') + [].[LastName] as [Full Name]
from [dbo].[DimCustomer] as [.]

```

This time there won't be an intermediate cache. transformation happens in the data source with the T-SQL query, and result will be loaded into Power BI.



## How Do I know Which Transformations Folds?

Great Question. It is important to understand which step/transformation folds and which doesn't. To understand that simply right click on every step and see if the View Native Query is enabled or not. If it is enabled Query Folding is supported for that step, otherwise not. Also Note that Query Folding is not supported for data sources such as web query, or CSV or things like that. Query Folding at the moment is only supported

for data stores that supports a native query language. For Web, Folder, CSV... there is no native query language, so you don't need to be worry about Query Folding.

\*Important Note: At the time of writing this post Merge Columns doesn't support Query Folding. I have reported this to Power Query Team, and they are working on it, to solve the issue. The Merge Columns is very likely to support query folding very soon as the result of bug fix. However there are always some other transformations that doesn't support query folding. This post is written to give you an understanding what kind of issue might happen, and how to resolve it.

My advice to you as a performance best practice is that when working with a relational data source (such as SQL Server). always check the query folding. Sometimes it is not supported, so use another approach for the transformation. Don't fall into the black hole of not folding, otherwise your query might take ages to run.

# Power BI Premium. Is It For You or Not?

Published Date: May 15, 2017





There were many announcements earlier this month about Power BI. One of them which was and still is the breaking news of all is Power BI Premium. Many of you already know that it is a new licensing plan, but how it works? Is this a licensing plan that works for you or not? What is the break-even point with this plan? In this post I'll answer all of these questions with a detailed explanation about Power BI Premium. If you want to learn more about Power BI; read [Power BI book; from Rookie to Rock Star](#).

## Power BI Licensing Before Premium

To look at the new licensing plan, it is best to first understand how the previous licensing worked. Before this change, licensing plan for Microsoft were simple, it had only two plans: Free, and Power BI Pro. Free was free of charge obviously. All you needed was just your company email address to get Power BI account. Pro, however provided some additional features with a cost of about \$9.99 per user per month.

Difference between these two plans was not about development work. You can do almost anything you want with Power BI free for doing the development work. But, when it comes to contribution, sharing, security, and using Power BI in production environment, most of features needed Pro account. Here are the details in a table;

| Feature                            | Power BI Free            | Power BI Pro        |
|------------------------------------|--------------------------|---------------------|
| Space allocation                   | 1GB                      | 10GB *              |
| Power BI Desktop                   | Yes                      | Yes                 |
| Development features               | Yes                      | Yes                 |
| DirectQuery or LiveQuery           | No                       | Yes                 |
| Data Set Refresh Frequency         | up to once a day         | up to 8 times a day |
| Gateways – On-premises data source | No                       | Yes                 |
| Row Level Security                 | No                       | Yes                 |
| Content Pack                       | No                       | Yes                 |
| Work Spaces                        | No                       | Yes                 |
| Data Streaming                     | up to 10K rows per hours | unlimited           |



| Feature                            | Power BI Free            | Power BI Pro        |
|------------------------------------|--------------------------|---------------------|
| Space allocation                   | 1GB                      | 10GB *              |
| Power BI Desktop                   | Yes                      | Yes                 |
| Development features               | Yes                      | Yes                 |
| DirectQuery or LiveQuery           | No                       | Yes                 |
| Data Set Refresh Frequency         | up to once a day         | up to 8 times a day |
| Gateways - On-premises data source | No                       | Yes                 |
| Row Level Security                 | No                       | Yes                 |
| Content Pack                       | No                       | Yes                 |
| Work Spaces                        | No                       | Yes                 |
| Data Streaming                     | up to 10K rows per hours | unlimited           |

As you can see anything regarding to Developing a Power BI solution is available for free. However, for using Power BI in a production environment, most of the time you require Pro account. Here is more details;

20. If you want to use a proper mechanism to share Power BI content with others, in most of the cases you need to use either Power BI [work spaces](#), or [Content Packs](#), which both are Pro features.
21. If you want to use [Row Level Security](#), which is giving access to users based on their part of the data set, not the whole, then you need Pro account.
22. If you are using [Gateways](#) (doesn't matter [Personal](#) or On-Premises) to connect to on-premises data source, then you need Pro account.
23. For LiveQuery or DirectQuery connection to data sources you will need Pro account. LiveQuery or DirectQuery option happens a lot in enterprise environment when the size of data and scale is so high, that Power BI 1GB model cannot host it.
24. and many more as you see in the table above

As you can see, in production environment of using Power BI, you will hit one of the limitations above for Free account, and you have to use Pro Account. This is very important to understand that;

Pro account is not only for developer, but also for consumer! If Power BI Content is Pro, then whoever uses and consumes that content should have Pro account.

Yes, you've read it correctly. Any content, that has one of below features considered as Pro content. and everyone who uses or consumes that content, should have Power BI pro account;

- Data from a DirectQuery dataset, such as [SQL Server Analysis Services](#) tabular data, [Azure SQL Database](#), [Azure SQL Data Warehouse](#), or Apache [Spark for HDInsight](#).
- Data from a [dataset that refreshes more frequently than daily](#).
- Data from a dataset that connects to on-premises data using the [Power BI Gateway – Personal](#) or the [On-premises Data Gateway](#), and for which a scheduled refresh is set.
- Data from a live connection to Azure Analysis Services.
- Data (including reports, dashboards or tiles) from a dataset that uses [Row-level security \(RLS\)](#).
- A dashboard or report that's installed from [an app](#) or an [organizational content pack](#).
- A dashboard, report, or dataset that's contained in an [app workspace](#).
- A dashboard that contains data streamed at a rate above 10k rows/hour.

Source from: <https://powerbi.microsoft.com/en-us/documentation/powerbi-power-bi-pro-content-what-is-it/>

## Why Premium?

With list of limitations for Power BI free and also pro account, it is almost obvious why we need another licensing plan (Premium). I just point few of them here for more elaboration;

### Large User Base

Limitations in Pro Content as above means that if you are working in a large organization with 10,000 users, and you have to pay 10,000 \* \$10 per month, which would be \$100K per month! or \$1.2 million per year! This sounds scary!

### Power BI Model Size Limitation

Also as you might already know; with Pro account you get 10GB space in your Power BI account. However, each Power BI Model (or let's say file), cannot be more than 1GB! This means you have to combine then Power BI with other technologies such as SQL Server Analysis Services for LiveQuery connection, or a premier database provider such as SQL Server, Oracle, Teradata... for DirectQuery Connection. And that means paying for licenses for those products too. Because in enterprise environment it is very likely that size of Power BI model rise to more than 1GB.



## Readers or Consumers

In most of the implementations of Power BI, majority of users are just readers or consumers of the report. Readers or consumers CAN interact with the report, they can use charts and visuals interactively and analyze the data, but they won't create or update reports. There is always small amount of developers who do the development work. In this licensing plan, every user who is using Pro content, regardless of the role (developer, or consumer), should be pro account. This sounds like paying more than what you should pay!



## What is Power BI Premium?

Now that you know about previous licensing plan of Power BI, I can start writing about Premium. What is the premium? Power BI Premium, is simply a licensing plan that covers limitations of Power BI pro account. It is a licensing plan that helps overcome what you couldn't easily achieve with Power BI pro account. The license start at higher ground, at the time of writing this post it is \$5K, but don't be scared from this high cost, we will get into details of that shortly.



## **Is Power BI Pro and Free Gone Now?**

No! Power BI Free and Pro are still there. However, there will be some changes in features supported by each type of account. I assure you here that all you can do with Power BI pro, you can still do it with Pro, so you don't need to upgrade to Premium if you don't want to.

Features that might be taken away from free are more about collaboration, such as sharing. So, in the feature for sharing Power BI content, you might need at least Pro account.

## **I Am Currently Using Power BI Pro, Is Premium The Only Option Now?**

No! You can continue using Power BI Pro. Pro gives you all you have already, you won't lose anything. and you don't have to upgrade to Premium. However, using Premium gives you some more options and features, let's look at them as below.

## **What Are Benefits of Power BI Premium?**

Power BI Premium is an additional licensing plan. Additional means that you can use it or not, if you use this plan, then you would get some benefits that helps you to reduce costs of your BI solution sometimes. But, first you need to know what these features are. These features might be superb features for some companies, but not essentials for others.

## Dedicated Power BI Resources

First and one of the most important features of Premium is that you get dedicated resources, such as capacity, cores and processing units. Normal Power BI Free or Pro accounts are publishing contents into SHARED Power BI servers. These servers are definitely high performing servers with great cores and capacity, however it is not dedicated.

With Power BI Premium, you can choose type of node that you want, and configure your own Power BI node, here are nodes as for now;

| CAPACITY NODE | CORES      | BACKEND CORES        | FRONTEND CORES |
|---------------|------------|----------------------|----------------|
| P1            | 8 v-Cores  | 4 cores, 25 GB RAM   | 4 cores        |
| P2            | 16 v-Cores | 8 cores, 50 GB RAM   | 8 cores        |
| P3            | 32 v-Cores | 16 cores, 100 GB RAM | 16 cores       |

## Larger Data Sets Supported

As you already know from Pro limitation; even with Pro account you cannot have a model with more than 1GB of size, and your total space is about 10GB. With Power BI Premium you can have up to 100TB space to fill in! Your model can simply be sized 50GB, or more in near future. This means you might not need paying for additional SQL Server or another database technology for developing large scale BI solution.

## More Frequent Data Set Refresh Rate

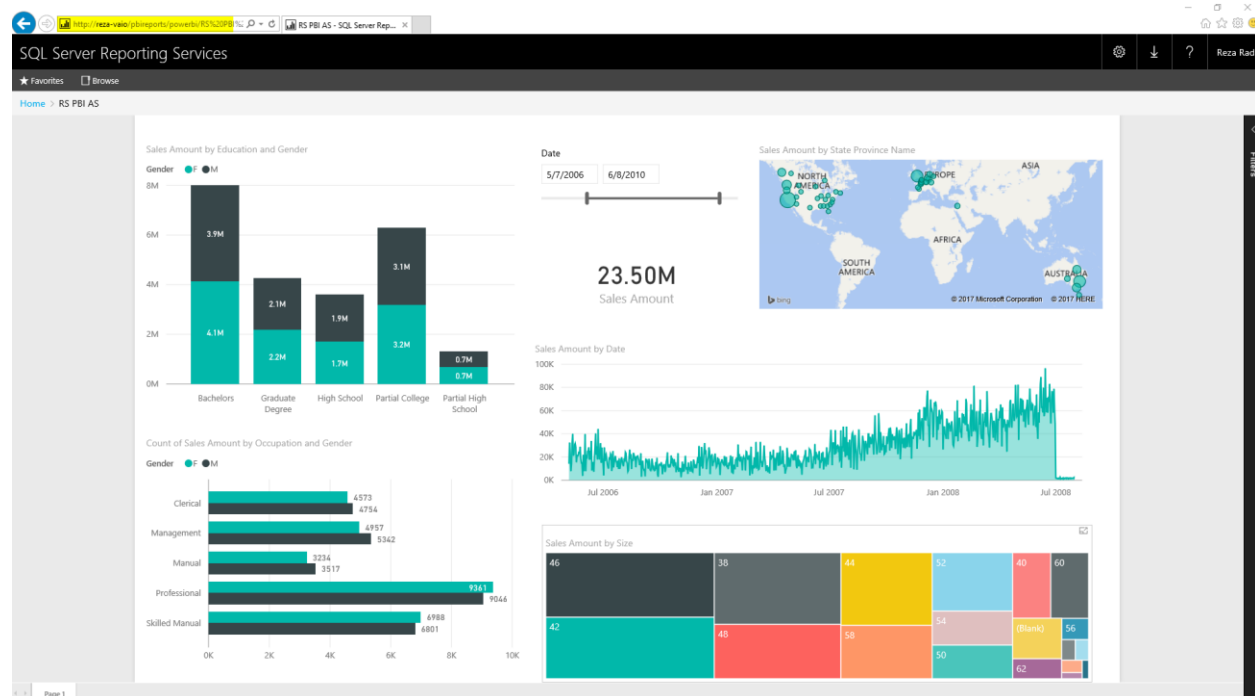
With Free you can refresh up to once a day. With Pro you can refresh up to 8 times a day. With Premium you can refresh your data set up to 48 times a day. The minimum frequency for refreshing data was 30 minutes, which is lifted in Premium. You can schedule on a minute level now with this new plan.

## Readers or Consumers Plan

Power BI Premium is a licensing plan for readers and consumers. Readers can enjoy consuming a content that is still a Pro content, but with a licensing that costs less than Pro per user. So, if your company has 10K users, you don't need to pay \$100K per month, you would pay far less than that. I will get into the details of costs shortly.

## Power BI Report Server

This part would need its own blog post of course, and is without a doubt one of the best offerings of Power BI Premium. Previously I have written about [Power BI reports on Premises with Reporting Services](#). This functionality will be available as a new feature called Power BI Report Server. Power BI Report Server will be part of Premium licensing. So with purchasing Premium licensing you would be able to host your reports on-premises. More details about costs will come. But the good news is that you will have Power BI on-premises. To understand how that might work, read [my blog post about hosting Power BI report on-premises](#).



## Upcoming Features for Premium

Power BI Premium would be the best offering of Power BI, and there are lots of awesome features coming in for it in near future, such as below. Note that at the time of writing this post, features below are not available yet. These are in the road map.





### **Incremental Refresh**

When we are talking about data set more than 1GB of the size, then incremental refresh matters. You don't want to populate that amount of data in every refresh. With Premium you will have this option in near future to use incremental load and update only changed set.

### **Pin Data Set to Memory**

Power BI by default allocate memory to models based on their usages. This automatic allocation sometimes might cause users to wait for their old report to load. In the feature you will have option to pin those data sets that you think are important and critical for business to memory and configure performance of Power BI based on your requirements.

### **Dedicated Data Refresh Node**

You would be able to dedicate specific nodes for data refresh, while other nodes are providing response to report. This will help performance of report loading to be high while data set is refreshing.

## Geo-Replica and Read-Only Replicas

You will be able to distribute replicas of Power BI model geographically and in other ways, so your users get the best performance.

## The Most Important Topic! Costs of Premium

Cost of Power BI Premium is the most important topic to discuss. When Premium announce first time, I have seen lots of people scared from the \$5K entry point of it. Let's get into details of costing and see how it is working.



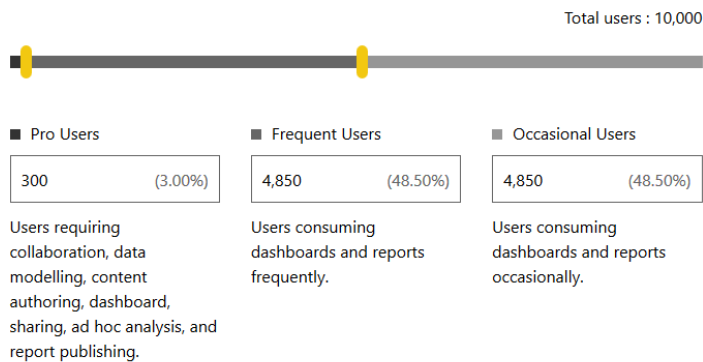
There is a [calculator](#) that helps you understand the costing based on your requirement. Basically the costing is based on how many developers (Pro) accounts you have, how many frequent users, and how many readers. The reason to separate frequent users and readers are mainly to understand how many cores you will need for Power BI Premium.

Let's go through an example. Consider an organization with 10K users. out of these 10K uses, only 300 of them are developers (they will create or update reports), rest are just consuming reports (Let's say 4.8K frequent consumers, and 4.8K occasional). Based on

the calculator for such requirement, you will need to have 300 Pro users (this is obvious), and 6 P1 nodes to cover readers. Each P1 Node costs \$5K per month.

So as a result it would costs about \$32K;

**Please tell us how your users will interact with your reports and dashboards**



## Cost estimate

**Total: \$32,967**

### Power BI Pro licenses

| Units | (×) Monthly fees      |   |
|-------|-----------------------|---|
| 300   | \$9.99 / User / Month | ? |

(=) Total fees **\$2,997 / Month**

### Power BI Premium

| Units                  | (×) Monthly fees               |   |
|------------------------|--------------------------------|---|
| 6 P1 nodes             | \$4,995/ Node / Month          | ? |
| Power BI Report Server | Included with Power BI Premium | ? |

(=) Total fees **\$29,970 / Month**

The costs is high of course, but if you compare it with purchasing 10K Power BI Pro plans (\$100K) it is almost one third of that cost! As I mentioned before, you don't need pay price of Pro for readers, so that makes your costs lower.

Also you will notice in the calculator that this costing is including Power BI Report Server. so with this cost you are also able to host reports on-premises.

## But Would This Premium Cost Be Always Better?

No! Like many other things, it depends. If you are working for a large enterprise with thousands of users, and most of users are consumers, then as you saw in the example above it would be much more cost effective to purchase Power BI Premium licensing.

However, if you are medium size or small business with few hundreds of users or even less, then maybe Power BI Pro is better option for you. Here is an example;

If you have 50 users for Power BI, even if you purchase Pro account for them all, you have to pay \$500 per months. However for premium you would need to pay at least \$5K which is ten times more expensive. So with the current entry point of nodes (which is \$5K), it is not cost effective to pay for Premium if you have such a small user base. simply continue using Power BI Pro.

If you need to host Power BI on-prem, you might still need to consider Premium even with small user base, because Power BI Report Server is available within Premium, or with SQL Server licensing (more details to come in the future).

## **What is Break-Even Point?**

One of the most common questions in this area is that when I need to use Premium, and when pro? what is the break-even point?





The answer to this question can be calculated easily. I cannot give you a precise number, because there are many different situations to consider details: Example: Two companies that have 5K users, and one of them might have 2K developers, the other one 200 only. One of them might have all users as frequent users, the other one 80% occasional users. Depends on your requirements number of nodes for Power BI Premium is different. and number of nodes has a direct affect on the costs.

I advise you to use the calculator and enter your total number of users, frequent users, and occasional users, then check if the total price with that requirement is lower than purchasing Pro account for all users, then you're break-even point is passed. If not, then you can simply use Power BI Pro.

I've had two examples in my post already. here are summary of those again;

- A large enterprise with 10K users, which only 300 of these users are developers, and the rest are half frequent, and half occasional users. 6 p1 nodes of Power BI would cover this user base, and so the cost would be:  $300 * \$10$  (per user Pro account) +  $6 * \$4,995$  (per node Premium) = \$32,967. This cost is far less than  $10K * \$10$  (per user Pro account) = \$100,000. So this organization should use Premium obviously.
- A small/medium size business with 50 users, even if pay \$10 for pro account for everyone, it is paying \$500 per month. However, if this company purchases premium. The minimum entry point (At the time of writing this blog) is \$4,995 which is ten times more expensive. So this company can simply continue using Power BI Pro.

## Summary

In Summary Power BI Premium is a great offering. It comes with lower pricing for enterprise scale customers, and with many features such as dedicated server, higher data set size, higher frequency of data refresh, Power BI on-premises and many other options. However, not all companies need these features, neither they have large user base. they can simply use Power BI Pro license as they are using already and enjoy the current offering. If you like to read more about details of Power BI Premium, I recommend reading [Power BI Premium Whitepaper](#) written by Amir Netz.

## Your Thoughts?



With Power BI Premium offering there were a lot of noises, questions, rumours, and comments about the offering. Some people said it is perfect, some said this is too expensive (the entry pricing), some said some of features should be available for Pro, and many other suggestions. Please note that this licensing plan is available to help in areas that other two plans couldn't (you already know it if you read my post fully). However, I would like to use this post as a message to Microsoft and let them know about what do YOU THINK about this change? What is good or bad about it? Do you have any suggestions that can make this offering better? Please let us know in comments area, I'm sure Microsoft Power BI team will consider every comments here (I'll pass these to them). Thanks in advance for your collaboration.

# Integrate Power BI into Your Application:

## Part 1 - Register your App

Published Date: June 26, 2017



Power BI for Developers

## Register an Application for Power BI

Register a new application that can be used to call Power BI APIs

### Step 1 Login to your Power BI account

Welcome, **Reza Rad!** (Wrong account? No problem, [logout](#) and try again.)

### Step 2 Tell us about your app

Let's start with some basic details.

App Name:

RADACAD Sample App

App Type:

Specify the type of app. Use 'Server-side Web app' for web apps or Web APIs, or 'Native app' for apps that run on client devices (Android, iOS, Wi

Server-side Web app

Server-side Web app

Native app

[Power BI Embedded](#), which I have written about it [previously](#), brings the power of Power BI into your application. Power BI Embedded Recently had major changes. If you remember from [my previous post](#), there were many limitations in the type of content you can embed into your application and the ways of interacting with it, and even in licensing. Now with new version of REST API for Power BI, everything changed. However, changes are in a good way; you gain more, lose nothing! This blog post series is about new way of integrating Power BI report into your application and reveals some of the questions you might have in this path. if you like to learn more about Power BI; read [Power BI book; from Rookie to Rock Star](#).

## Power BI Embedded – Previous Version

Previously, if you wanted to embed Power BI content into your application, you could. However, you could only do that for Power BI reports with DirectQuery connection to Azure SQL Database. Every other type of data source was not supported! This was the main barrier for everyone wanted to use this service. Difficulty in provisioning the content was another problem. With previous version of Power BI embedded, you couldn't use the content published in Power BI website. You needed to publish content separately into different location (Azure Power BI Embedded Workspace). With previous method you could not embed dashboards or tiles, it was just reports. Licensing plan for Power BI Embedded was another issue; totally unpredictable.

## Integrate Power BI content into your Application – New Way

New way of integrating Power BI into your application is so powerful and fascinating, that it doesn't even bring the name of embedded with it. It is more like integration. That's why many people call it Integrating Content into Power BI. With the new way; you can bring any content (Power BI Dashboard, Tile, and Report) into your application. Content can be sourced from ANY type of data sources, it can be import data or LiveQuery or DirectQuery, it can use Gateways or get it from cloud data sources. You can easily integrate the same content which is published into Power BI service. And the licensing plan is different, more predictable I'd say.

Let's take a look at this new way in a blog post series, starting from this one.

### Sample Code

The sample code for this example can be downloaded from [here](#). This sample code won't be used in this particular post, but next posts after this will need it.

## Power BI REST API

In this post I will be using Power BI REST API which is at the moment version 2. If you download the sample code above, then you don't need to download API separately. However, if you are starting a new project, you can go to NuGet Manager in Visual Studio and search for Power BI REST API.


The new version of Power BI REST API has MANY features, which I will talk about them in a series of blog posts. In fact, features are so much that cannot fit into few posts. So, stay tuned for lots of upcoming posts about it. In this post I'll focus on few functionalities that is useful for integrating report into your application.

## Register Your Application

First step is to register your application. What does that mean? Any application which wants to interact with Azure should be registered, and authorized through a process. To register your application there are two ways; one is simpler to start, the other one is simpler to maintain 😊

### Register through App Registration website

Go to this [website](#), and register your application from there. it is a very easy process, let's have a look. Login with your Power BI account first;

 Power BI for Developers

## Register an Application for Power BI

Register a new application that can be used to call Power BI APIs

**Step 1** Login to your Power BI account

Create a new application with just a few simple steps. Once created, you can make any subsequent changes to your app details via the [Azure Management Portal](#).

Sign in with your existing account

Don't have an account? Get one for free.

After signing in, you can specify a name for your application (just to be able to find it easier in a list of applications). also select the application type.





## Power BI for Developers

# Register an Application for Power BI

Register a new application that can be used to call Power BI APIs

### Step 1 Login to your Power BI account

Welcome, **Reza Rad!** (Wrong account? No problem, [logout](#) and try again.)

### Step 2 Tell us about your app

Let's start with some basic details.

App Name:

RADACAD Sample App

App Type:

Specify the type of app. Use 'Server-side Web app' for web apps or Web APIs, or 'Native app' for apps that run on client devices (Android, iOS, Wi

Server-side Web app

Server-side Web app

Native app

There are two types of applications: Server-side web app, and Native app. what is the difference?

29. Server-side Web App; An application that runs on a server, and browsed through clients. Such as web applications and mobile applications in most of the cases.
30. Native app; an application that build to run on a specific environment. such as Console application.

### Types of Integrating Power BI Content into your application

Before selecting the right application type, you need to understand that there are two types of integrating Power BI content into an application:

39. Integrating without a Token: You are integrating content into your application, and you will be using Power BI users for access to the content. This means everyone who wants to look at the content should have Power BI account.
40. Integrating with a Token: You are integrating content into your application, and you are NOT using Power BI users. You will be managing user authentication through your own application. Users may not even have Power BI accounts.

In this example we are talking about integration WITH a Token.

Very Important: If you are using the method of integration "With a Token", then no matter what type of application you are writing. You have to choose Native App from the type of application above, even if you have a web/mobile application.

After selecting Native app, then set a redirect URL. For this example, you can simply set it to be: `http://localhost:13526`

This redirect URL is not that much important, later in future posts I'll explain about it. You can even set it to be other URLs.

Next step is to decide what permissions you want for this application. As you can see in below image, there are different levels of permissions you can apply. It is not usually good idea to give full permission to an application, however for the test purpose you can select all permissions in below list. For this example you might only need Read All Reports.

Redirect URL:  
A valid URL

`http://localhost:13526`

---

### Step 3 Choose APIs to access

Select the APIs and the level of access your app needs.

| Dataset APIs                                                    | Report and Dashboard APIs                                      | Other APIs                                          |
|-----------------------------------------------------------------|----------------------------------------------------------------|-----------------------------------------------------|
| <input checked="" type="checkbox"/> Read All Datasets           | <input checked="" type="checkbox"/> Read All Dashboards        | <input checked="" type="checkbox"/> Read All Groups |
| <input checked="" type="checkbox"/> Read and Write All Datasets | <input checked="" type="checkbox"/> Read All Reports           | <input checked="" type="checkbox"/> Create Content  |
|                                                                 | <input checked="" type="checkbox"/> Read and Write All Reports |                                                     |

Register your application



Last step in this section is to click on Register your application, you will get a Client ID.

#### Step 4 Register your app

Once you've set everything the way you want it, click the button below and we'll register your app. Your client ID and secret (for web apps only) will appear below. Be sure to copy the values into your app. By clicking the Register App button, you have accepted the [terms of use](#).

Register App

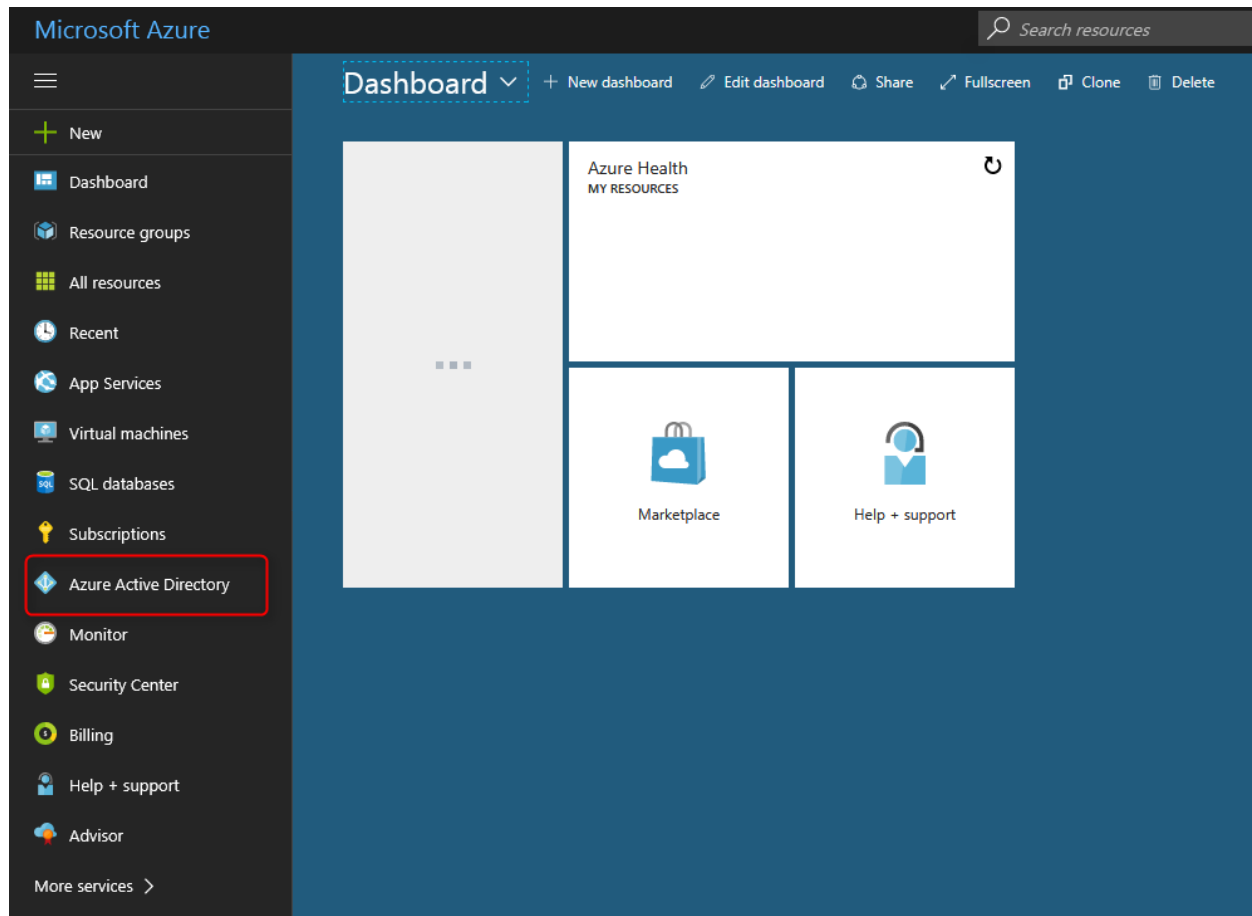
Client ID:

If you register a server-side app, you will also receive a client Secret along with client ID. Client ID (and Client Secret), are important factors for your application to be able to interact with Azure and Power BI service.

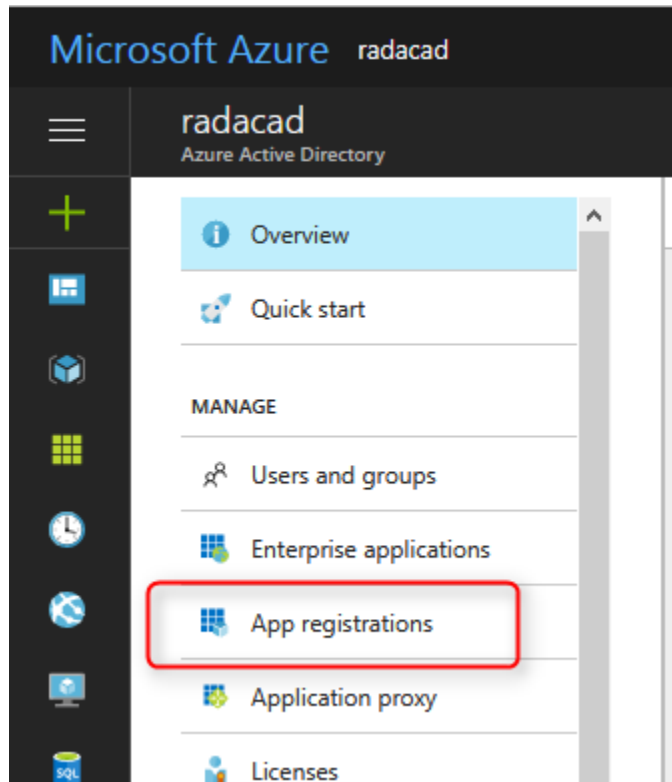
## Register Application through Azure Portal

Another method of registering application is through Azure Portal. Go to <http://portal.azure.com>, for this way of registration. The previous method of registration is easier and more user friendly. This method is much better for maintaining an application. With previous method, there is no way of maintaining an application. you can only initiate registration from there. If you want to change permissions or anything related to the application later on, the only way is through this method: Azure Portal.

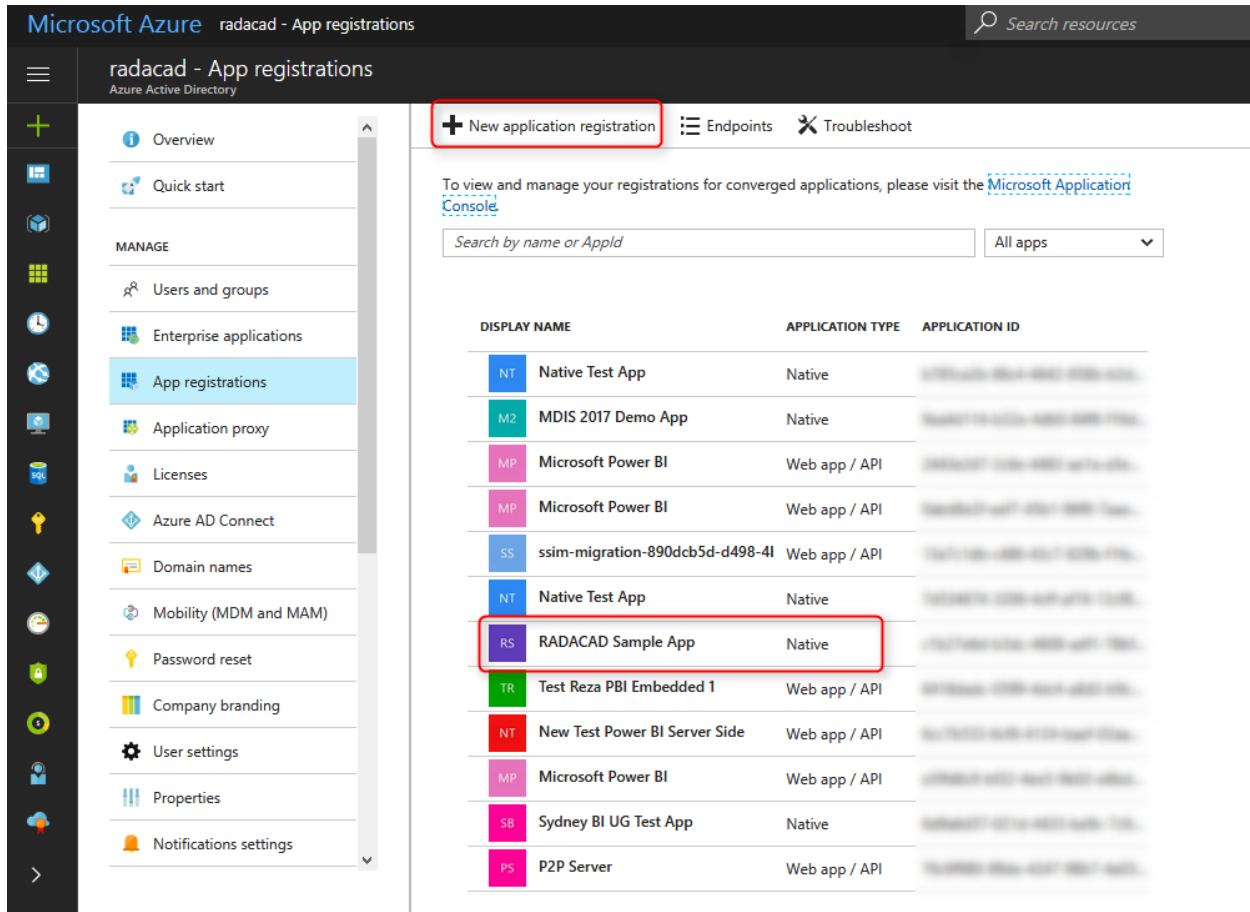
In Azure portal, go to Azure Active Directory;



Then go to App Registrations



Here you can register your application. or you can even find your existing application. screenshot below shows the RADACAD Sample Application I have registered above;



Microsoft Azure radacad - App registrations

radacad - App registrations  
Azure Active Directory

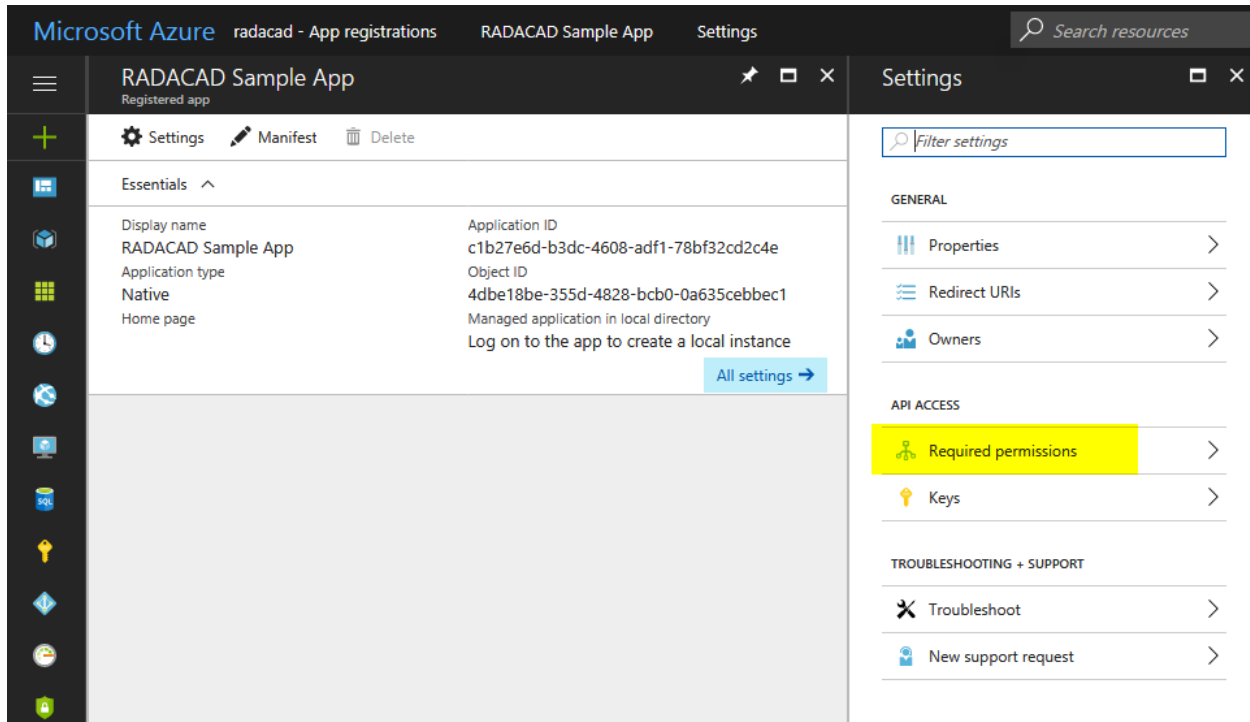
+ New application registration Endpoints Troubleshoot

To view and manage your registrations for converged applications, please visit the [Microsoft Application Console](#)

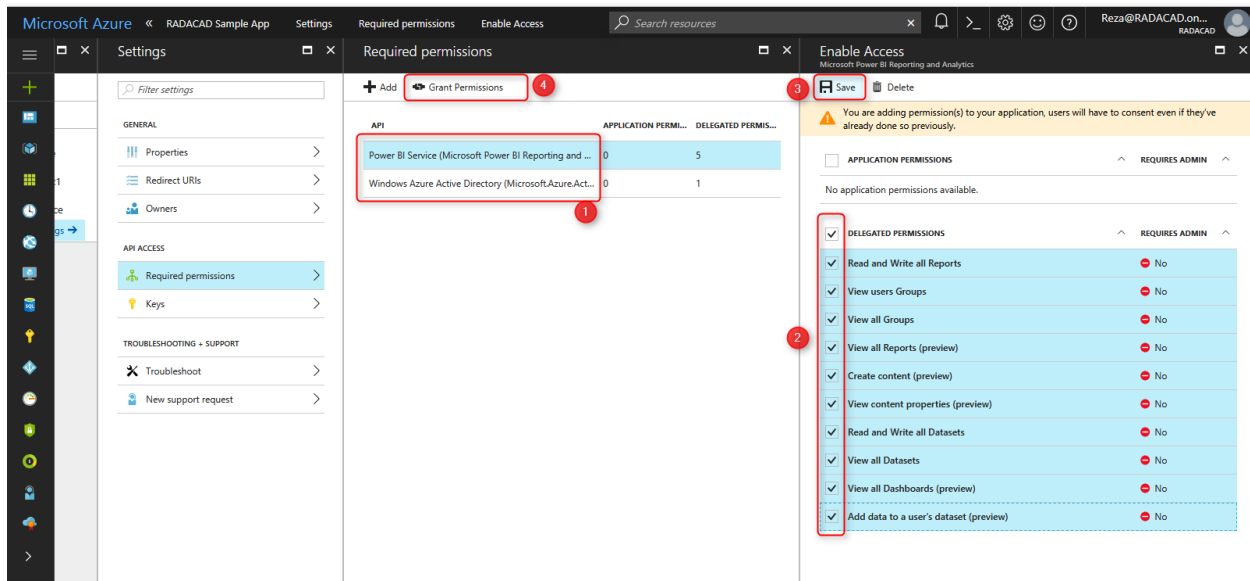
Search by name or Appld All apps

|    | DISPLAY NAME                       | APPLICATION TYPE | APPLICATION ID                  |
|----|------------------------------------|------------------|---------------------------------|
| NT | Native Test App                    | Native           | 1775c5b5-8b5a-4882-b386-455b... |
| M2 | MDIS 2017 Demo App                 | Native           | 8a6d7141-425a-4882-b386-455b... |
| MP | Microsoft Power BI                 | Web app / API    | 2895a271-125a-4882-b386-455b... |
| MP | Microsoft Power BI                 | Web app / API    | 2895a271-125a-4882-b386-455b... |
| SS | ssim-migration-890dcb5d-d498-4f... | Web app / API    | 7a7112b5-4882-b386-455b-715a... |
| NT | Native Test App                    | Native           | 7a7112b5-4882-b386-455b-715a... |
| RS | RADACAD Sample App                 | Native           | 7a7112b5-4882-b386-455b-715a... |
| TR | Test Reza PBI Embedded 1           | Web app / API    | 8f70a6a1-125a-4882-b386-455b... |
| NT | New Test Power BI Server Side      | Web app / API    | 8f70a6a1-125a-4882-b386-455b... |
| MP | Microsoft Power BI                 | Web app / API    | 4798b211-125a-4882-b386-455b... |
| SB | Sydney BI UG Test App              | Native           | 4798b211-125a-4882-b386-455b... |
| PS | P2P Server                         | Web app / API    | 7a7112b5-4882-b386-455b-715a... |

After selecting the application, you can go to Required Permissions to set up permissions for this app.



For Power BI apps you will find two sets of Permissions; for Power BI, and for Azure Active Directory. after selecting the set, then you can select permissions required, then save. Don't forget to click on Grant Permissions after that, otherwise permissions won't be granted!



You can check permissions assigned then in the list later if you want.

Required permissions

+ Add

Grant Permissions

| API                                                    | APPLICATION PERMI... | DELEGATED PERMIS... |
|--------------------------------------------------------|----------------------|---------------------|
| Power BI Service (Microsoft Power BI Reporting and ... | 0                    | 10                  |
| Windows Azure Active Directory (Microsoft.Azure.Act... | 0                    | 9                   |

Enable Access

Microsoft Azure ActiveDirectory

Save

Delete

☐ APPLICATION PERMISSIONS

REQUIRES ADMIN

No application permissions available.

☒ DELEGATED PERMISSIONS

REQUIRES ADMIN

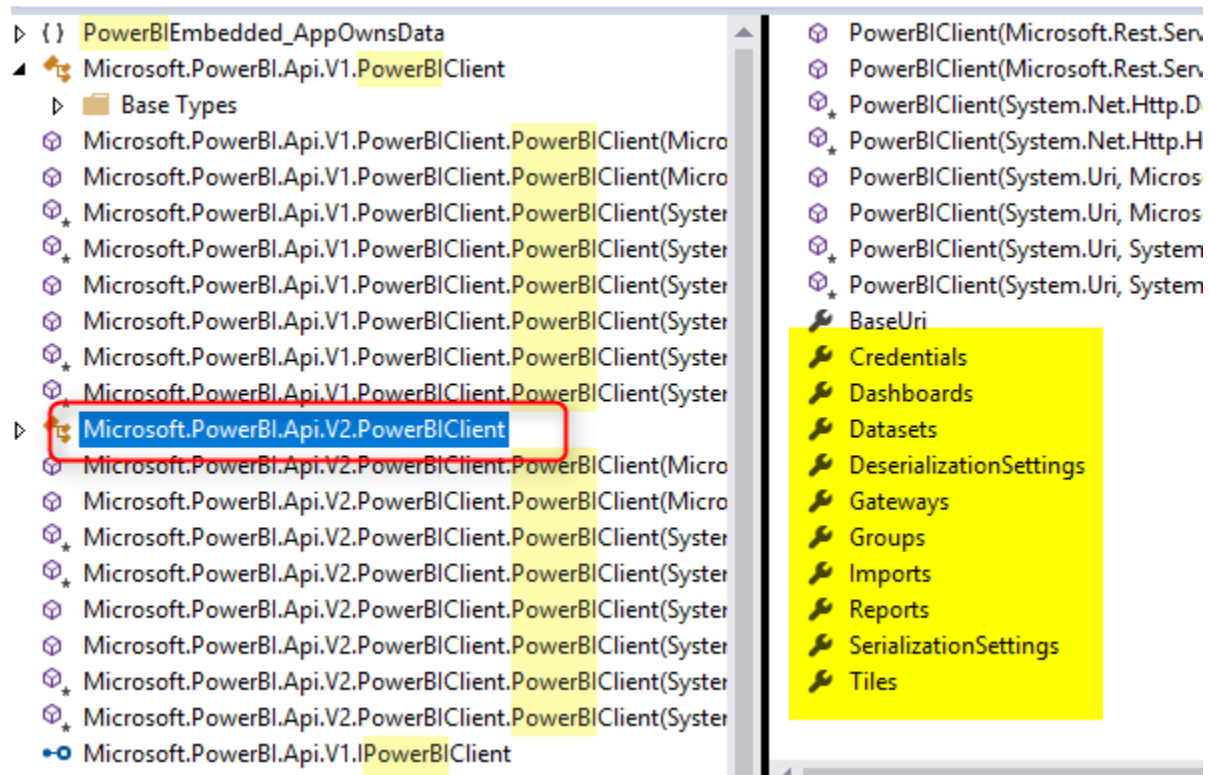
|                                                                                |     |
|--------------------------------------------------------------------------------|-----|
| <input checked="" type="checkbox"/> Read hidden memberships                    | Yes |
| <input checked="" type="checkbox"/> Sign in and read user profile              | No  |
| <input checked="" type="checkbox"/> Read all users' basic profiles             | No  |
| <input checked="" type="checkbox"/> Read all users' full profiles              | Yes |
| <input checked="" type="checkbox"/> Read all groups                            | Yes |
| <input checked="" type="checkbox"/> Read and write all groups                  | Yes |
| <input checked="" type="checkbox"/> Read and write directory data              | Yes |
| <input checked="" type="checkbox"/> Read directory data                        | Yes |
| <input checked="" type="checkbox"/> Access the directory as the signed-in user | No  |

Process of registering your application is done, now the next step is to start scripting and coding of your application with Power BI REST API.

# Integrate Power BI into Your Application:

## Part 2 - Authenticate

Published Date: June 27, 2017



In the [part 1](#), you learned how to register an application to be able to interact with Power BI service and Azure environment. In this part, I'm going to explain what is the process involved after that in order to authenticate through your application into Azure Active Directory. This step is important step before doing any work with Power BI REST API, after completing this step, you can then create a Power BI client object that can be used for embedding Power BI content, data source and gateway management, and all other operations with Power BI objects. if you like to learn more about Power BI; read [Power BI book; from Rookie to Rock Star](#).

### Step 1: Register your Application

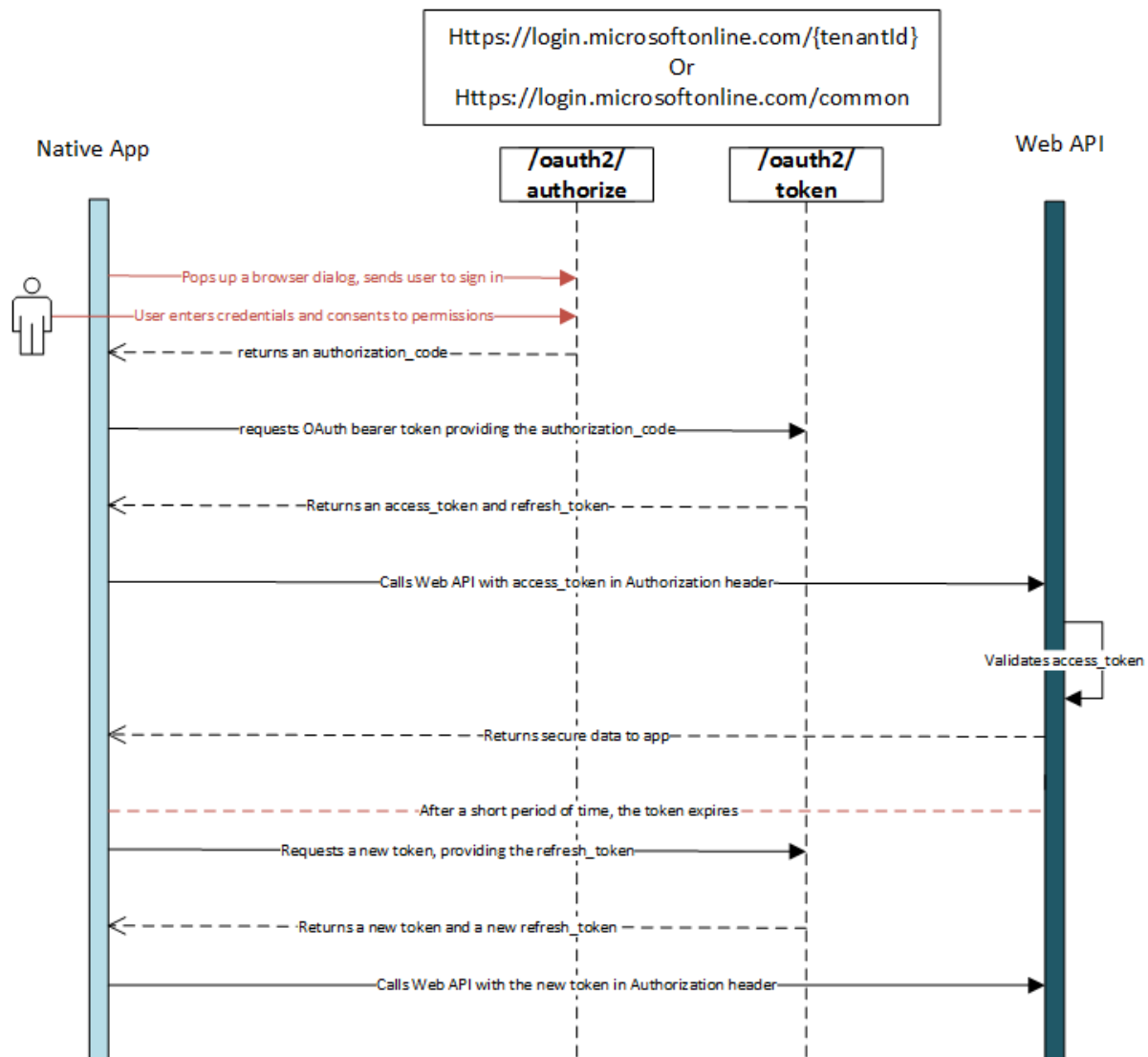
## Step 2: Authenticate

Every application which want to interact with Azure environment, should go through an authentication process. This authentication process is passing Client ID (and sometimes Client Secret) to Azure, and getting an authentication code. From this authentication code, and access Token can be fetched. This access token is a property that should be involved in every request sent from your application to Power BI and Azure afterwards. Without this access token your requests will fail.

### Get Access Token

As mentioned above, you need to get the access token in order to interact with Power BI REST API. This access token can be fetched through a process of authentication with Azure. the process itself, requires a full blog post at least to explain. below is the flow diagram of the process;





I'm not going to explain details of this flow in this post. If you are interested in that details, read this [document](#).

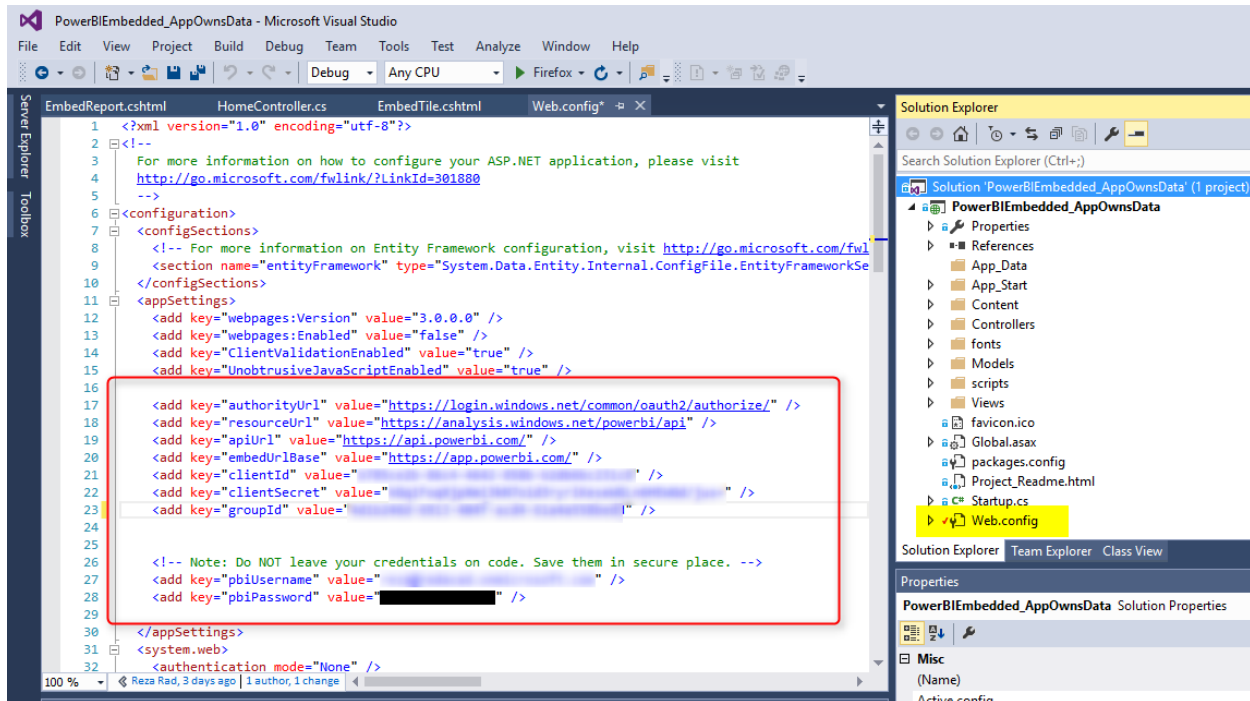
In this post, I only show you few lines of code, that do this for you.

## Sample Code

The sample code for this example can be downloaded from [here](#).

## Get Access Token from C# Code

After opening the sample code above, go to Web.Config file



In the web.config file, you will see some settings such as ApiURL, and EmbedURL, and some other configurations. Leave these as is. Just put your CLIENT ID there (the one you've got from previous step of registering your application). For this example you don't need Client Secret. Also you need to specify your Power BI username and password, make sure that these are saved from a secure place, and fetched from there.

## Sample Application Structure

The sample application used here is an MVC application which has three layers; Model, View, and Controller. I'm not going to explain details of that, you need to read a book to understand how MVC works. In a very basic definition; View is presentation layer, Controller is business layer and code logic, and Model is the content passed from Controller to View. Authentication process and getting Access Token in this scenario happens in controller.

## Authentication Process

If you open HomeController.cs under Controllers folder, then you will find the first method EmbedReport(), which is doing the authentication process in very first few lines;



```

29
30
31
32 public async Task<ActionResult> EmbedReport()
33 {
34 var error = GetWebConfigErrors();
35 if (error != null)
36 {
37 return View(new EmbedConfig()
38 {
39 ErrorMessage = error
40 });
41 }
42
43 // Create a user password credentials.
44 var credential = new UserPasswordCredential(Username, Password);
45
46 // Authenticate using created credentials
47 var authenticationContext = new AuthenticationContext(AuthorityUrl);
48 var authenticationResult = await authenticationContext.AcquireTokenAsync(ResourceUrl, ClientId, credential);
49
50 if (authenticationResult == null)
51 {
52 return View(new EmbedConfig()
53 {
54 ErrorMessage = "Authentication Failed."
55 });
56 }
57
58 var tokenCredentials = new TokenCredentials(authenticationResult.AccessToken, "Bearer");
59
60 // Create a Power BI Client object. It will be used to call Power BI APIs.
61 using (var client = new PowerBIClient(new Uri(ApiUrl), tokenCredentials))
62 {
63 // Get a list of reports
64 }

```

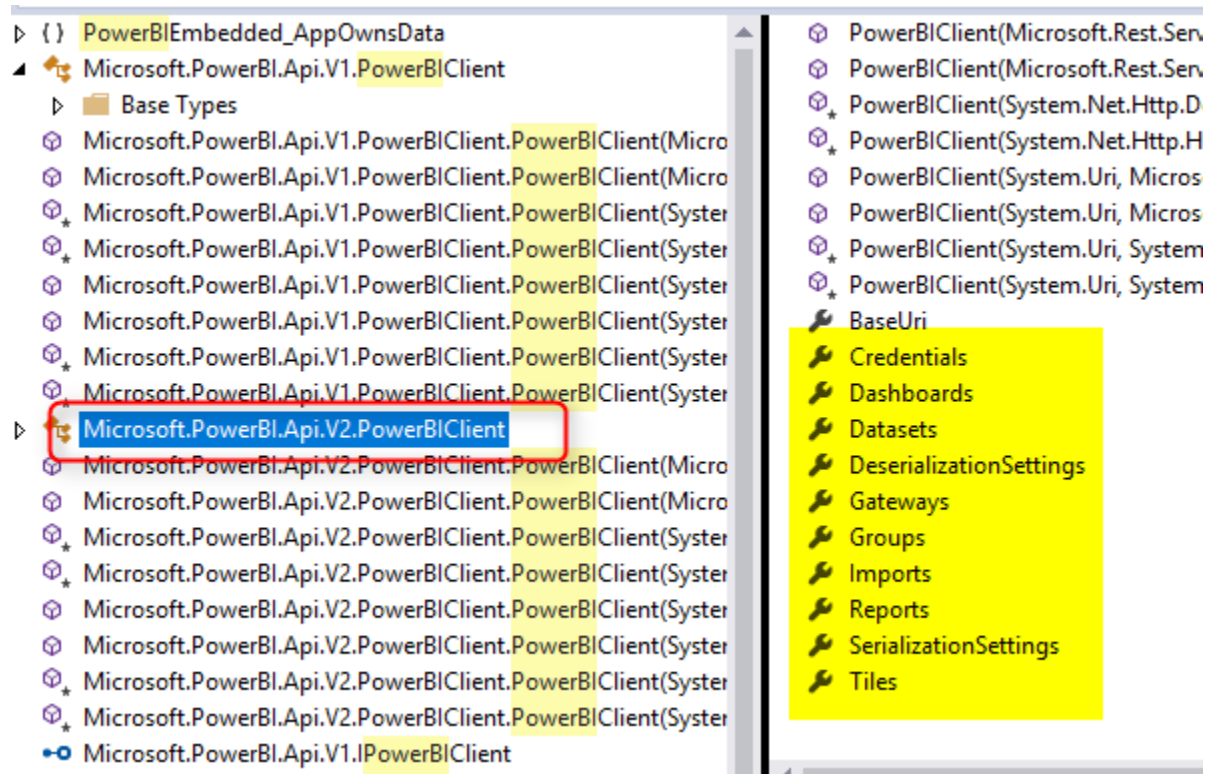
Authentication process in general includes, passing credentials to AuthorityURL, and getting the result back, if application has the access, and authentication goes through correctly, then you will get a result which you can fetch Token from it. The token then will be used in creating a new instance of Power BI Client. Power BI Client is the main object that should be created from REST API in order to interact with Power BI service.

## Power BI Client Object

Power BI Client object is the main object needs to be created in order to interact with Power BI Service. As you've learned above, you need to have the access token for this step. You just need to parameters to create a Power BI Client object; ApiURL (which is static), and token (which is gained through authentication process above).

```
var client = new PowerBIClient(new Uri(ApiUrl), tokenCredentials);
```

After creating this object, you have access to massive amount of information from Power BI Service, and you can implement many actions through it (Which I explain many of these actions in other posts).



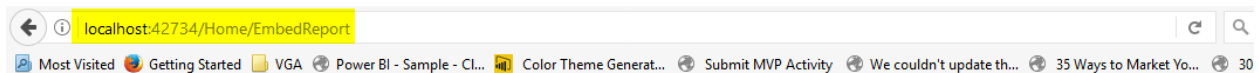
## Next Steps

After creating Power BI client object, you can do massive operations with it; embedding content into an application, refreshing data set, data source and gateway management, and many other operations. Read next upcoming posts to learn more details about it.

# Integrate Power BI into Your Application:

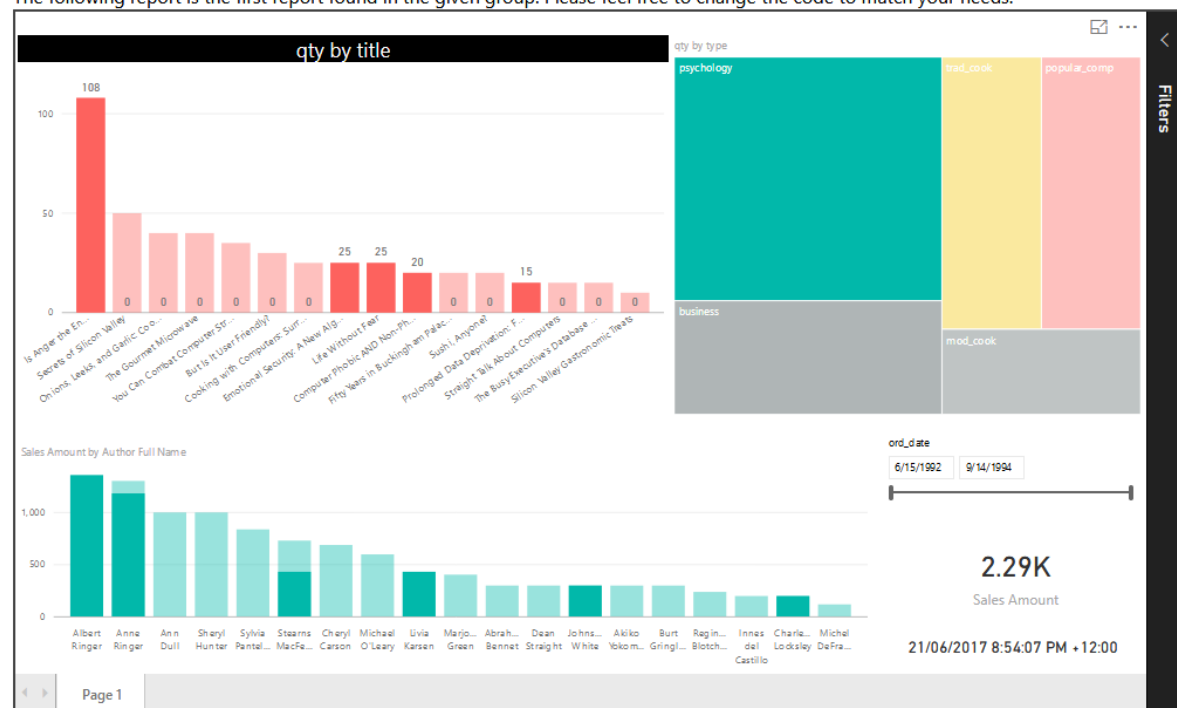
## Part 3 - Embed Content

Published Date: June 28, 2017



### Embedded Report

The following report is the first report found in the given group. Please feel free to change the code to match your needs.



In the [part 1](#), you learned how to register an application to be able to interact with Power BI service and Azure environment. In the [part 2](#), you learned how the authentication process works and how you can get the access token. In this part, I'm going to explain what is the process of embedding content (Dashboard, Report, and Tile) into a web page. you will learn about JavaScript part of it, and also the REST API section of it. if you like to learn more about Power BI; read [Power BI book; from Rookie to Rock Star](#).

### [Step 1: Register your Application](#)

## [Step 2: Authenticate](#)

## Step 3: Embed Content

in this step you will learn about codes and the method to embed the content into your application after registration and authentication of the application.

### Sample Code

The sample code for this example can be downloaded from [here](#).

## Power BI Client Object

Power BI Client object is the main object needs to be created in order to interact with Power BI Service. As you've learned above, you need to have the access token for this step. You just need to parameters to create a Power BI Client object; ApiURL (which is static), and token (which is gained through authentication process mentioned in [previous post](#)).

```
var client = new PowerBIClient(new Uri(ApiUrl), tokenCredentials);
```

After creating this object, you have access to massive amount of information from Power BI Service, and you can implement many actions through it (Which I explain many of these actions in other posts).

## Get List of Reports

For this example; I'm going to Embed a report into web page, so I will get list of reports first (you can even directly mention the report Id).

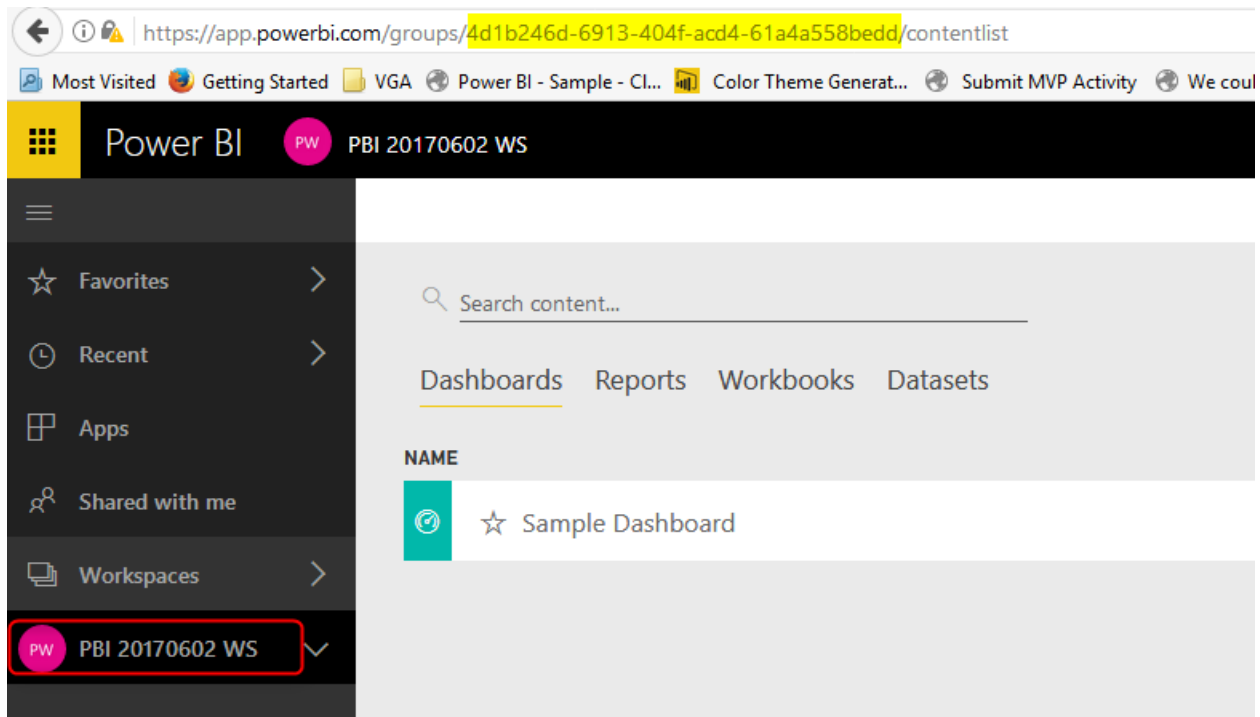
```
// Get a list of reports.
var reports = await client.Reports.GetReportsInGroupAsync(GroupId);

// Get the first report in the group.
var report = reports.Value.FirstOrDefault();

if (report == null)
{
 return View(new EmbedConfig()
 {
 ErrorMessage = "Group has no reports."
 });
}
```

```
}
```

As you can see the function used is `GetReportsInGroupAsync`. This function simply returns all reports under a group. Group Id is specified in `web.config` file. Notice that Id is not name of the group. the name that you see for group, dashboard, report, and data set, are just display labels. Each object in Power BI has an ID which is a unique identifier (GUID). you can easily find it when you browse the group in Power BI Service through URL;



These Ids can then be used for Power BI REST API functions.

Function above will return first report on that list, because I've used `FirstOrDefault()` function. You can easily mention any other reports if you want to.

## Creating Embed Content

In the MVC application the next step is to create the embed content and push it to the view. lines below generating the embed content;

```
// Generate Embed Token.
var generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "view");
var tokenResponse = await client.Reports.GenerateTokenInGroupAsync(GroupId, report.Id,
generateTokenRequestParameters);
```

```
if (tokenResponse == null)
{
 return View(new EmbedConfig()
 {
 ErrorMessage = "Failed to generate embed token."
 });
}

// Generate Embed Configuration.
var embedConfig = new EmbedConfig()
{
 EmbedToken = tokenResponse,
 EmbedUrl = report.EmbedUrl,
 Id = report.Id
};

return View(embedConfig);
```

## JAVA Script in the Web page

In the EmbedReport.Chtml page which is view web page, you need a JavaScript section to embed the content into the page. here is the JavaScript code;

```
// Read embed application token from Model
var accessToken = "@Model.EmbedToken.Token";

// Read embed URL from Model
var embedUrl = "@Html.Raw(Model.EmbedUrl)";

// Read report Id from Model
var embedReportId = "@Model.Id";

// Get models. models contains enums that can be used.
var models = window['powerbi-client'].models;

// Embed configuration used to describe the what and how to embed.
// This object is used when calling powerbi.embed.
// This also includes settings and options such as filters.
// You can find more information at https://github.com/Microsoft/PowerBI-JavaScript/wiki/Embed-Configuration-Details.
var config = {
 type: 'report',
 tokenType: models.TokenType.Embed,
 accessToken: accessToken,
 embedUrl: embedUrl,
 id: embedReportId,
 permissions: models.Permissions.All,
```



```

settings: {
 filterPaneEnabled: true,
 navContentPaneEnabled: true
}
};

// Get a reference to the embedded report HTML element
var reportContainer = $('#reportContainer')[0];

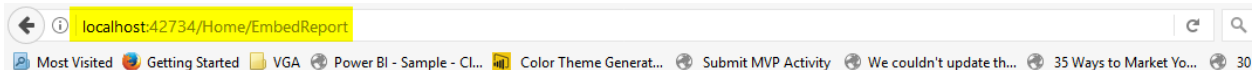
// Embed the report and display it within the div container.
var report = powerbi.embed(reportContainer, config);

```

As you can see in the above section; Embed Token is required for embedding part again, plus embed content coming from the controller.

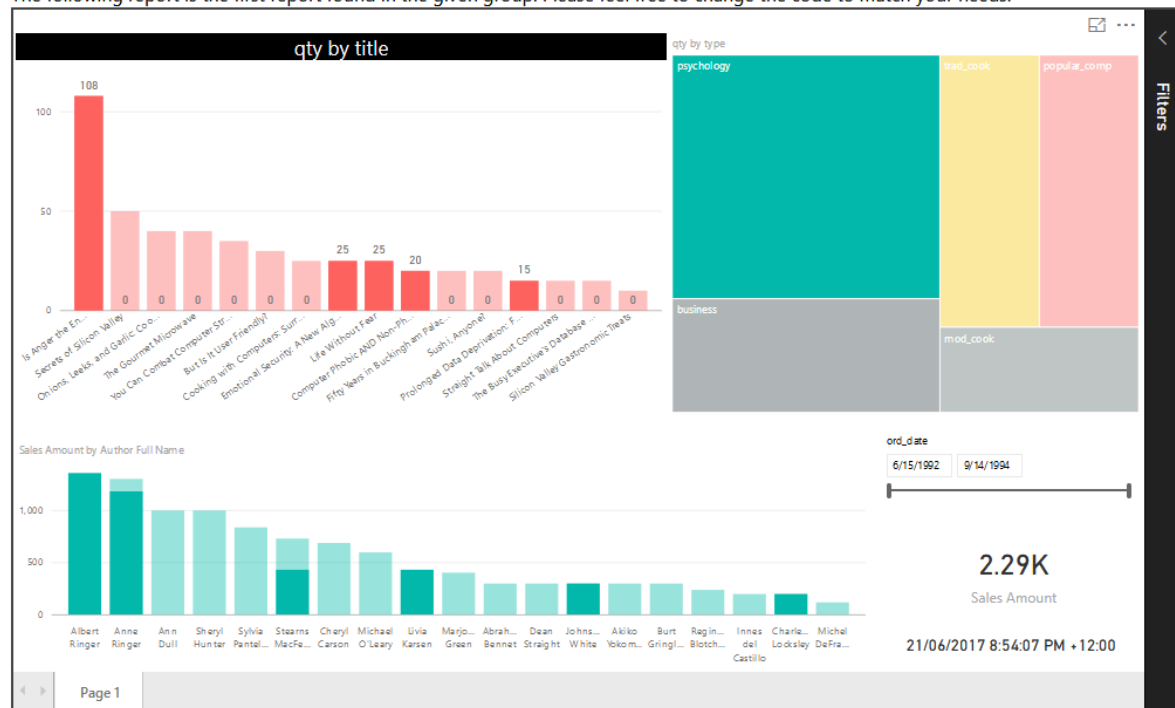
## Testing Application

If you run the sample application, you can check and see how embedding Report in application works perfectly;



### Embedded Report

The following report is the first report found in the given group. Please feel free to change the code to match your needs.

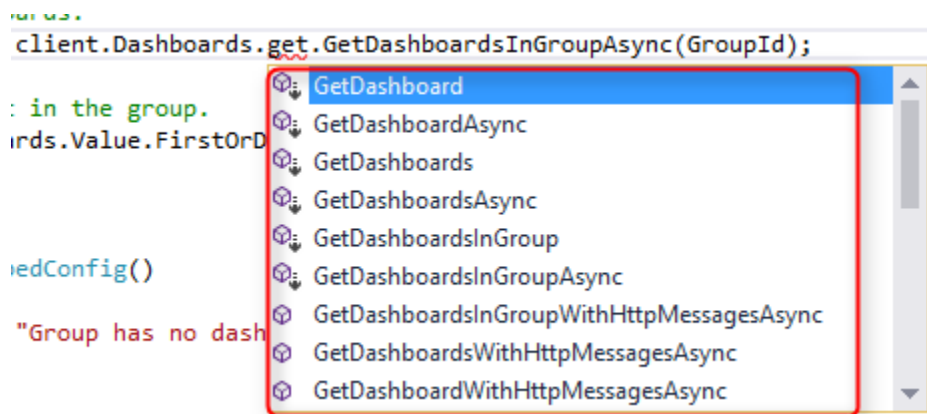


As you can see, the report is integrated into my application which is running on my localhost. This method of embedding works with any type of report, sourced from any data source with any connection type.

## Getting List of Dashboards

There is a function that returns list of dashboards for you, and you can easily select dashboards you want.

GetDashboardsInGroupAsync, or GetDashboardAsync and many methods as listed below can give you information about dashboards;



After fetching list of dashboards it can then be passed to view with script below;

```
// Get a list of dashboards.
var dashboards = await client.Dashboards.get.GetDashboardsInGroupAsync(GroupId);

// Get the first report in the group.
var dashboard = dashboards.Value.FirstOrDefault();

if (dashboard == null)
{
 return View(new EmbedConfig()
 {
 ErrorMessage = "Group has no dashboards."
 });
}

// Generate Embed Token.
var generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "view");
var tokenResponse = await client.Dashboards.GenerateTokenInGroupAsync(GroupId,
dashboard.Id, generateTokenRequestParameters);
```

```
if (tokenResponse == null)
{
 return View(new EmbedConfig()
 {
 ErrorMessage = "Failed to generate embed token."
 });
}

// Generate Embed Configuration.
var embedConfig = new EmbedConfig()
{
 EmbedToken = tokenResponse,
 EmbedUrl = dashboard.EmbedUrl,
 Id = dashboard.Id
};

return View(embedConfig);
```

## Embed Dashboard

Finally embedding dashboard is similar to embedding report, with slight difference;

```
// Read embed application token from Model
var accessToken = "@Model.EmbedToken.Token";

// Read embed URL from Model
var embedUrl = "@Html.Raw(Model.EmbedUrl)";

// Read dashboard Id from Model
var embedDashboardId = "@Model.Id";

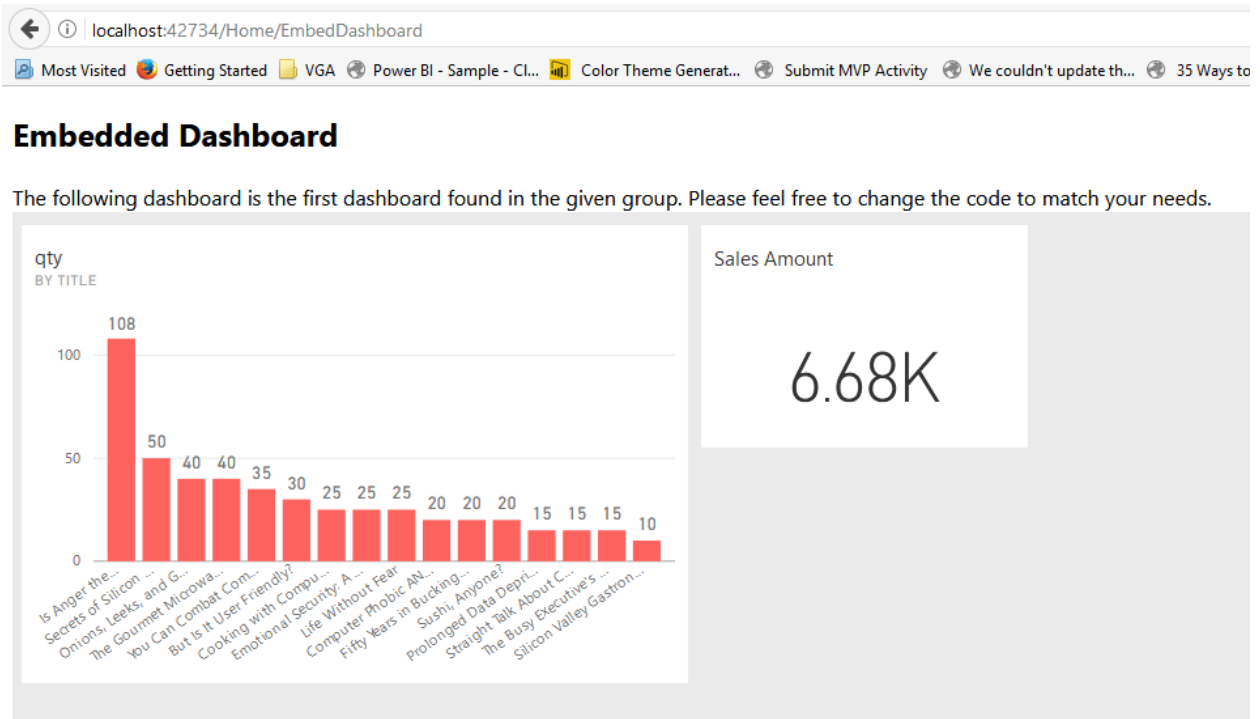
// Get models. models contains enums that can be used.
var models = window['powerbi-client'].models;

// Embed configuration used to describe the what and how to embed.
// This object is used when calling powerbi.embed.
// This also includes settings and options such as filters.
// You can find more information at https://github.com/Microsoft/PowerBI-JavaScript/wiki/Embed-Configuration-Details.
var config = {
 type: 'dashboard',
 tokenType: models.TokenType.Embed,
 accessToken: accessToken,
 embedUrl: embedUrl,
 id: embedDashboardId
};
```

```
// Get a reference to the embedded dashboard HTML element
var dashboardContainer = $('#dashboardContainer')[0];
```

```
// Embed the dashboard and display it within the div container.
var dashboard = powerbi.embed(dashboardContainer, config);
```

so as a result this is the embedded dashboard;



## Getting Tiles

You first need to have the dashboard object in order to get tiles under it.

```
it client.Dashboards.gettiles | GetTilesInGroupAsync(GroupId, dashboard.

: tile in the group.
: .Value.FirstOrDefault

:d Token for a tile.
: nRequestParameters =
: e = await client.Til
```

- GetTiles
- GetTilesAsync
- GetTilesInGroup
- GetTilesInGroupAsync
- GetTilesInGroupWithHttpMessagesAsync
- GetTilesWithHttpMessagesAsync

As you can see above there are some functions that fetch Tiles from Dashboard and group. Code below will fetch first tile in a dashboard and pass it to the view page.

```
var tiles = await client.Dashboards.GetTilesInGroupAsync(GroupId, dashboard.Id);
```

```
// Get the first tile in the group.
var tile = tiles.Value.FirstOrDefault();

// Generate Embed Token for a tile.
var generateTokenRequestParameters = new GenerateTokenRequest(accessLevel: "view");
var tokenResponse = await client.Tiles.GenerateTokenInGroupAsync(GroupId, dashboard.Id,
tile.Id, generateTokenRequestParameters);

if (tokenResponse == null)
{
 return View(new TileEmbedConfig()
 {
 ErrorMessage = "Failed to generate embed token."
 });
}

// Generate Embed Configuration.
var embedConfig = new TileEmbedConfig()
{
 EmbedToken = tokenResponse,
 EmbedUrl = tile.EmbedUrl,
 Id = tile.Id,
 dashboardId = dashboard.Id
};

return View(embedConfig);
```

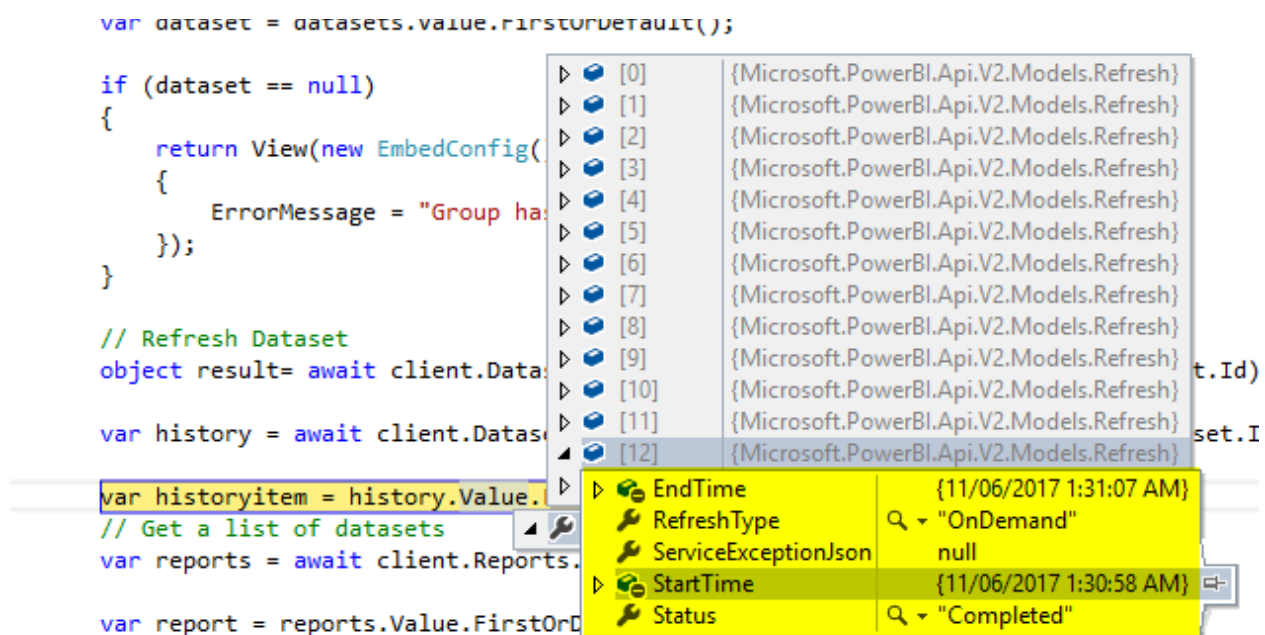
## Embedding Content Limitless

Very important consideration of the new REST API way of integrating Power BI content is that, now you can embed any type of content: Dashboard, Report, and Tile. Your data set can be connected to any data set, online or on-premises, import data or DirectQuery. With this method of integration, you can integrate anything that you have in Power BI account easily into your application. I will explain another post later about FAQs of such integration plan. Stay tuned for next blog posts from this series.

# Integrate Power BI into Your Application:

## Part 4 - Refresh Data Set

Published Date: June 29, 2017



Power BI REST API is not just for embedding content or getting list of dashboards and reports. It also has many functions to work with data sets, gateways, and data sources. As one of the many exciting features of that; you can easily refresh a data set from an application. you can refresh it as many as times you want. with any frequency you want. You can refresh your data set after ETL run through a console application. or you can have a service application that kick of refresh even every single minute! There is no limitation for your data refresh anymore! Let's check it out. if you like to learn more about Power BI; read [Power BI book; from Rookie to Rock Star](#).

### Step 1: Register your Application

### Step 2: Authenticate

### Sample Code

The sample code for this example can be downloaded from [here](#).

## Step 3: Get Data Set

With Power BI REST API version 2, you can access many objects from Power BI service. One of these object types is data set. You can get list of data sets, and you can then apply some operations on it, such as refresh. After authenticating, code below gets list of data sets;

```
var datasets = await client.Datasets.GetDatasetsInGroupAsync(GroupId);
```

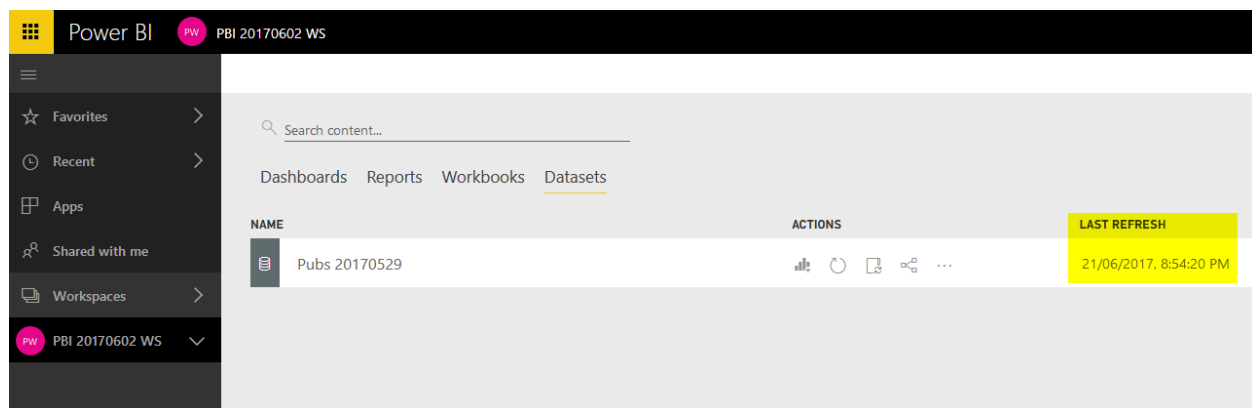
GetDatasetsInGroupAsync simply gives you a list of all data sets, and you can then iterate through them. As I've mentioned before, group id should be the guid of the group. You can then find the particular data set you want. code below just gets the very first data set in the list:

```
var datasets = await client.Datasets.GetDatasetsInGroupAsync(GroupId);
```

```
var dataset = datasets.Value.FirstOrDefault();

if (dataset == null)
{
 return View(new EmbedConfig()
 {
 ErrorMessage = "Group has no datasets."
 });
}
```

This data set is already in a group in my Power BI account;



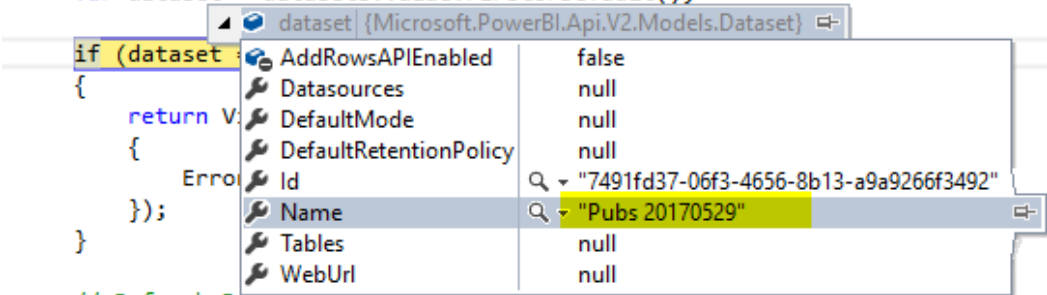
As you can see the data set above, is named Pubs 20170529, and it refreshed at 21st of June 2017 as the last time.

Now with running the code, I can easily find this data set;

```
// Create a Power BI Client object. It will be used to call Power BI APIs.
using (var client = new PowerBIClient(new Uri(ApiUrl), tokenCredentials))
{
 // Get a list of datasets
 var datasets = await client.Datasets.GetDatasetsInGroupAsync(GroupId);

 var dataset = datasets.Value.FirstOrDefault();

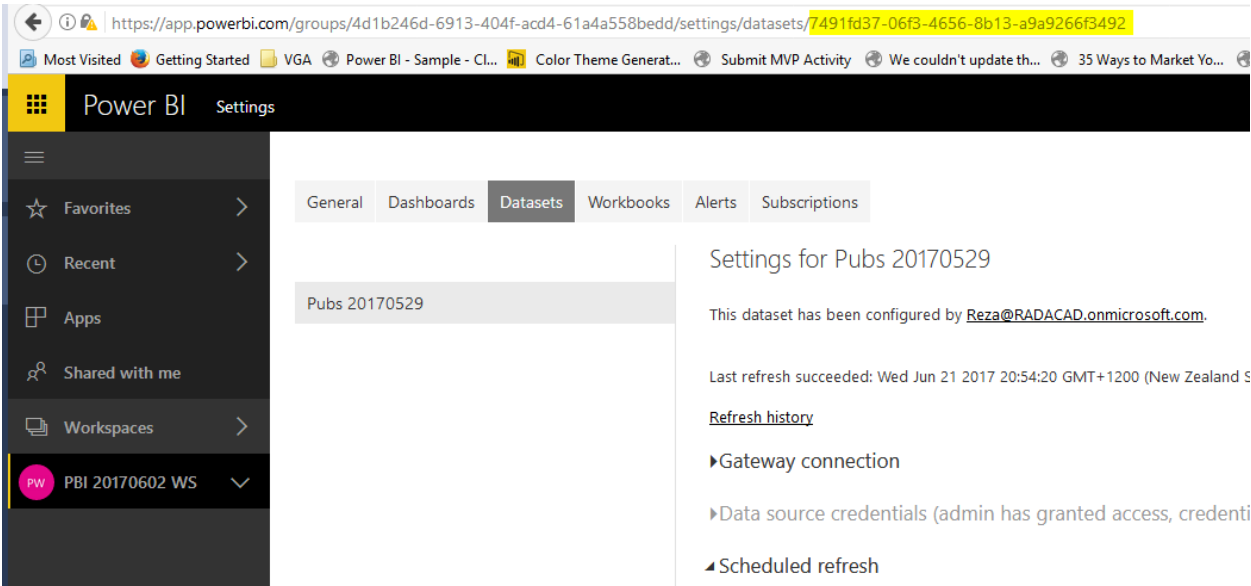
 if (dataset != null)
 {
 return V
 {
 Error
 });
 }
}
```



After finding the data set, you can easily refresh it.

## Refresh Data Set

You can even directly refresh a data set, you just need Id of that data set. one way to find the data set ID is through application. another way is from URL of power bi service. If you go to setting of that data set, you can find the ID of it easily;

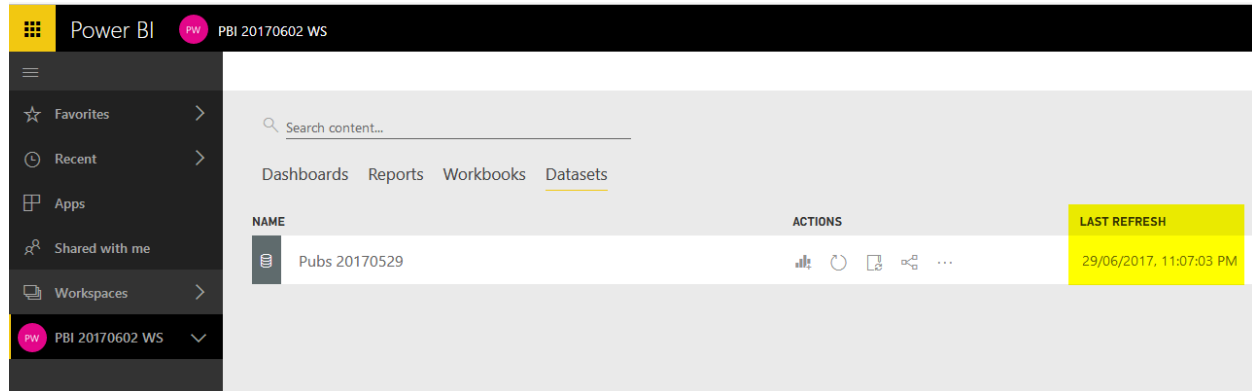


the code to refresh a data set is very simple:

```
object result= await client.Datasets.RefreshDatasetInGroupAsync(GroupId, dataset.Id);
```

As a result of refresh, then you can see the data set is refreshed successfully;





Last refresh time is now 29th of June 2017, 11pm, which is the time of writing this blog post 😊

Important note to consider here is that this data set is an IMPORT DATA data set, and it is working through gateway. so I need the gateway to be up and running. However, even if you are connected to a cloud data source through import data, then even without gateway you can refresh the data set. so, any data set can be easily refreshed. the fascinating fact is:

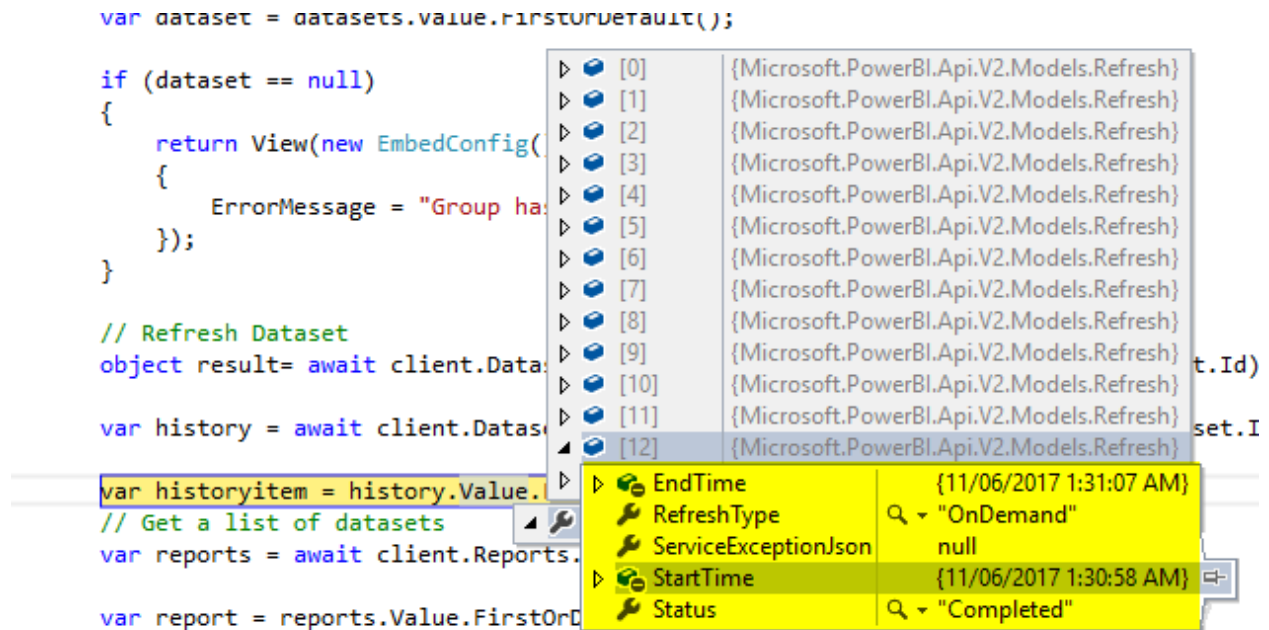
You can refresh your data set from an application. Any time you like, at any frequency you like. After ETL run, or on a schedule basis. This is Refreshing data beyond limits.

## Refresh History

The new REST API is an evolution in Power BI integration. You can even read the history of refresh for a data set. Here is the code for it:

```
var history = await client.Datasets.GetRefreshHistoryInGroupAsync(GroupId, dataset.Id);
```

Running this code will return the whole history of refresh for that data set. You can easily understand the type of refresh, and time and status of each.



Same refresh history in Power BI service:

## Refresh history

Scheduled OneDrive

| Details | Type      | Start                   | End                     | Status    | Message                                                |
|---------|-----------|-------------------------|-------------------------|-----------|--------------------------------------------------------|
|         | On demand | 29/06/2017, 11:15:27 PM | 29/06/2017, 11:15:31 PM | Completed |                                                        |
|         | On demand | 29/06/2017, 11:06:46 PM | 29/06/2017, 11:07:03 PM | Completed |                                                        |
| Show    | On demand | 29/06/2017, 11:04:41 PM | 29/06/2017, 11:04:44 PM | Failed    | The gateway is either offline or could not be reached. |
|         | On demand | 21/06/2017, 8:53:57 PM  | 21/06/2017, 8:54:20 PM  | Completed |                                                        |
|         | On demand | 14/06/2017, 7:15:08 AM  | 14/06/2017, 7:15:11 AM  | Completed |                                                        |
|         | On demand | 14/06/2017, 7:11:37 AM  | 14/06/2017, 7:11:42 AM  | Completed |                                                        |
|         | On demand | 13/06/2017, 8:08:36 AM  | 13/06/2017, 8:08:49 AM  | Completed |                                                        |
| Show    | On demand | 13/06/2017, 8:06:51 AM  | 13/06/2017, 8:06:54 AM  | Failed    | The gateway is either offline or could not be reached. |

Close

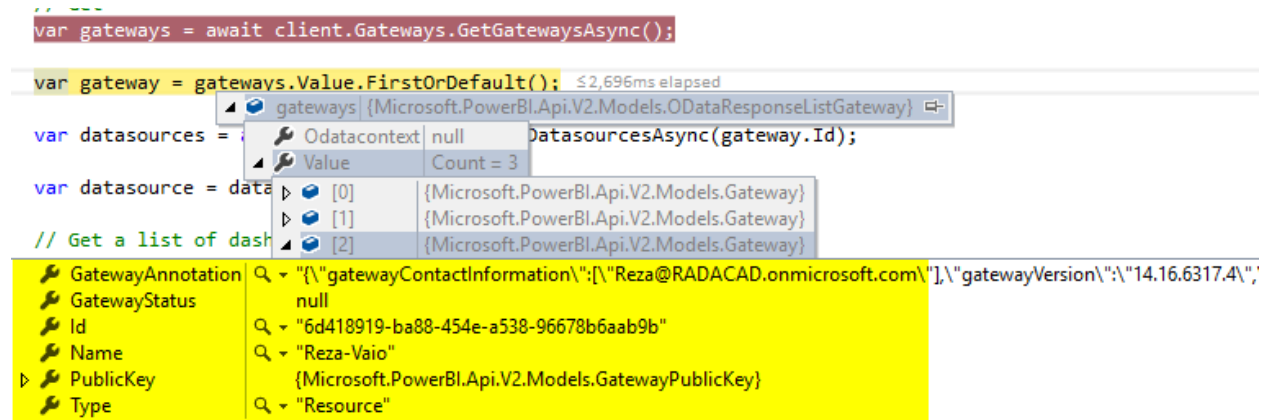
This means you can easily write an application that refresh the data set anytime, and also check the history of refresh for troubleshooting.

In the next blog post I'll explain some of other great features of Power BI REST API, such as managing gateways and data sources.

# Integrate Power BI into Your Application:

## Part 5 – Data Source Management

Published Date: July 6, 2017



New REST API of Power BI is revolutionary. In addition to embedding content in Power BI, ability to refresh data set from API, it also gives you many functions to work with Gateways and Data Sources. With this API, you can set up new data sources, clone data sets, check the credentials of a data source, get list of all data sources under a gateway, and do many other operations. In other words; Data source management can be fully automated with REST API. Continue reading this post if you like to learn more about this. If you like to learn more about Power BI; read [Power BI book; from Rookie to Rock Star](#).

### [Step 1: Register your Application](#)

### [Step 2: Authenticate](#)

### Sample Code

The sample code for this example can be downloaded from [here](#).

## Get List of Gateways

List of gateways are accessible through a function under Gateways, called `GetGatewaysAsync`. you can easily fetch list of gateways (on-premises gateways, not personal), and then loop through these.

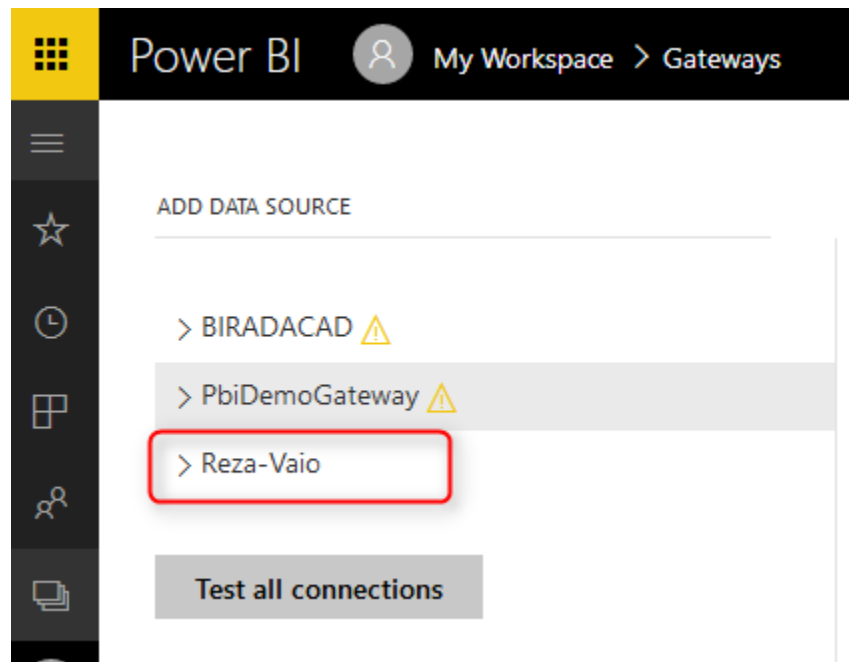
```
var gateways = await client.Gateways.GetGatewaysAsync();
```

```
// ...
var gateways = await client.Gateways.GetGatewaysAsync();
var gateway = gateways.Value.FirstOrDefault(); // 2,696ms elapsed
var datasources = await client.Gateways.DatasourcesAsync(gateway.Id);
var datasource = data[0];
// Get a list of dashboards
```

| Property          | Value                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------|
| GatewayAnnotation | Q - {"gatewayContactInformation":{"Reza@RADACAD.onmicrosoft.com"},"gatewayVersion":{"14.16.6317.4","14.16.6317.4"}} |
| GatewayStatus     | null                                                                                                                |
| Id                | Q - "6d418919-ba88-454e-a538-96678b6aab9b"                                                                          |
| Name              | Q - "Reza-Vaio"                                                                                                     |
| PublicKey         | {Microsoft.PowerBI.Api.V2.Models.GatewayPublicKey}                                                                  |
| Type              | Q - "Resource"                                                                                                      |

This function as you can see in above code screenshot will give you list of all on-premises data gateways defined under your account. then you can loop through each gateway and get the information of that gateway, for example; name of the gateway, Id of it, and some information such as version.

Here are list of gateways under this Power BI account;



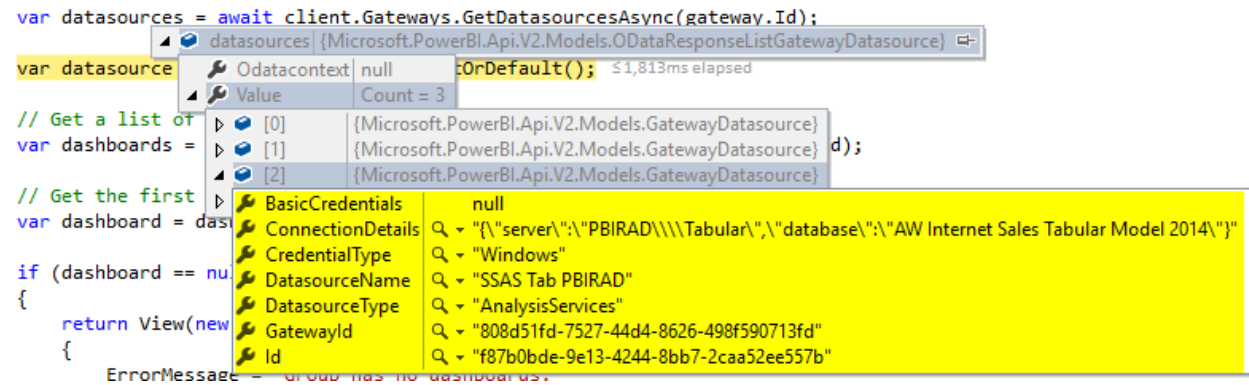
As you can see all of above gateways can be accessed through REST API.

## Get List of Data Sources

Under a gateway, you can access all data sources with a GetDataSourcesAsync function.

```
var dashboards = await client.Dashboards.GetDashboardsInGroupAsync(GroupId);
```

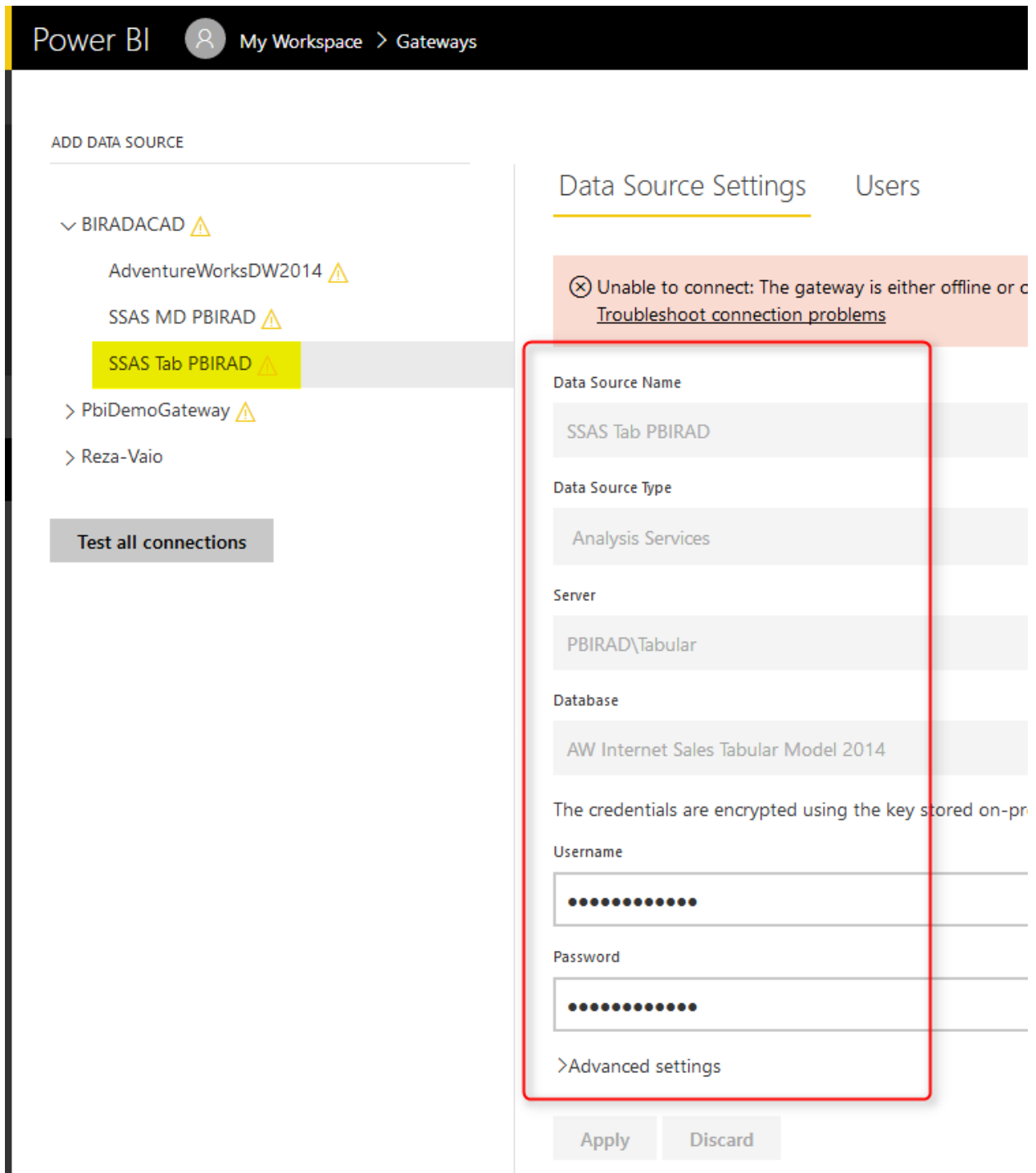
```
var datasources = await client.Gateways.GetDatasourcesAsync(gateway.Id);
var datasource = datasources.FirstOrDefault();
// Get a list of dashboards
var dashboards = client.Dashboards.GetDashboardsInGroupAsync(GroupId);
// Get the first dashboard
var dashboard = dashboards.FirstOrDefault();
if (dashboard == null)
{
 return View(new ErrorMessage { Message = "Group has no dashboards." });
}
```



| Id                                                  | GatewayId                                           |
|-----------------------------------------------------|-----------------------------------------------------|
| {Microsoft.PowerBI.Api.V2.Models.GatewayDataSource} | {Microsoft.PowerBI.Api.V2.Models.GatewayDataSource} |
| {Microsoft.PowerBI.Api.V2.Models.GatewayDataSource} | {Microsoft.PowerBI.Api.V2.Models.GatewayDataSource} |
| {Microsoft.PowerBI.Api.V2.Models.GatewayDataSource} | {Microsoft.PowerBI.Api.V2.Models.GatewayDataSource} |

| Property          | Value                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------|
| BasicCredentials  | null                                                                                                     |
| ConnectionDetails | Q - "{\\"server\\":\\"PBIRAD\\\\\\Tabular\\",\\"database\\":\\"AW Internet Sales Tabular Model 2014\\"}" |
| CredentialType    | Q - "Windows"                                                                                            |
| DatasourceName    | Q - "SSAS Tab PBIRAD"                                                                                    |
| DatasourceType    | Q - "AnalysisServices"                                                                                   |
| GatewayId         | Q - "808d51fd-7527-44d4-8626-498f590713fd"                                                               |
| Id                | Q - "f87b0bde-9e13-4244-8bb7-2caa52ee557b"                                                               |

which match exactly the configuration of that data source in the Power BI service



Power BI My Workspace > Gateways

ADD DATA SOURCE

✓ BIRADACAD ⚠

- AdventureWorksDW2014 ⚠
- SSAS MD PBIRAD ⚠
- SSAS Tab PBIRAD ⚠**
- > PbiDemoGateway ⚠
- > Reza-Vaio

Test all connections

Data Source Settings Users

⊗ Unable to connect: The gateway is either offline or c  
[Troubleshoot connection problems](#)

Data Source Name  
SSAS Tab PBIRAD

Data Source Type  
Analysis Services

Server  
PBIRAD\Tabular

Database  
AW Internet Sales Tabular Model 2014

The credentials are encrypted using the key stored on-pr

Username  
●●●●●●●●

Password  
●●●●●●●●

>Advanced settings

Apply Discard

As you can see; there are also properties for accessing the credentials. or even changing it.

## Data Source Management

In addition to reading list of data sources, you can do some other operations such as creating a data source, setting up credentials for it. and even binding a data source to a gateway.

## Create Data Source

There is a function to create data source, called `CreateDatasourceAsync`. You need to specify data source through a new object of `PublishDatasourceToGatewayRequest` for that. this object can contain type of data set, and connection information about it including credentials.

```
var newdatasource = await client.Gateways.CreateDatasourceAsync(gateway.Id,
 new PublishDatasourceToGatewayRequest(
 "Sql",
 "{ \"server\": \".\", \"database\": \"AdventureWorksDW2014\" }",
 new CredentialDetails(
 "Basic"),
 "AdventureWorksDW2014"
)
);
```

## Binding Data Set to a Gateway

after creating the data source, then you can bind a data set to that gateway. For finding the particular data set, please read the [previous blog post](#). when you have the data set key, then you can easily use BindToGatewayInGroupAsync function;

```
client.Datasets.BindToGatewayInGroupAsync(GroupId, datasetKey, new
BindToGatewayRequest(gateway.Id));
```

## Summary

There are many other data source and gateway functions that you can use, make sure you explore the REST API and find out more about all functions. Next post will be about posting data rows to a real-time streaming data set through REST API. Stay tuned.





# Power BI Training

Reza runs Power BI training courses both online and in-person. RADACAD also runs Advanced Analytics with R, Power BI, Azure Machine Learning and SQL Server courses ran by Dr. Leila Etaati. Our courses run both online, and in-person in major cities and countries around the world.

Check schedule of upcoming courses here:

<http://radacad.com/events>

<http://radacad.com/power-bi-training>

some of upcoming events in next few months:

31<sup>st</sup> of July 2017 – Power BI Live 5-days course – Europe  
11<sup>th</sup> of August 2017 – Advanced Data Modeling and DAX – Colombo, Sri Lanka  
14<sup>th</sup> of August 2017 – Power BI Live 3-days course – Bangalore, India  
28<sup>th</sup> of August 2017 – Power BI Live 5-days course – US East  
11<sup>th</sup> of September 2017 – Power BI Live 5-days course – Asia and Australia West  
25<sup>th</sup> of September 2017 – Power BI Live 5-days course – US West  
9<sup>th</sup> of October 2017 – Power BI Live 5-days course – Australia East  
16<sup>th</sup> of October 2017 – Power BI Live 5-days course – Auckland, New Zealand  
20<sup>th</sup> of November 2017 – Power BI Live 5-days course – Wellington, New Zealand

