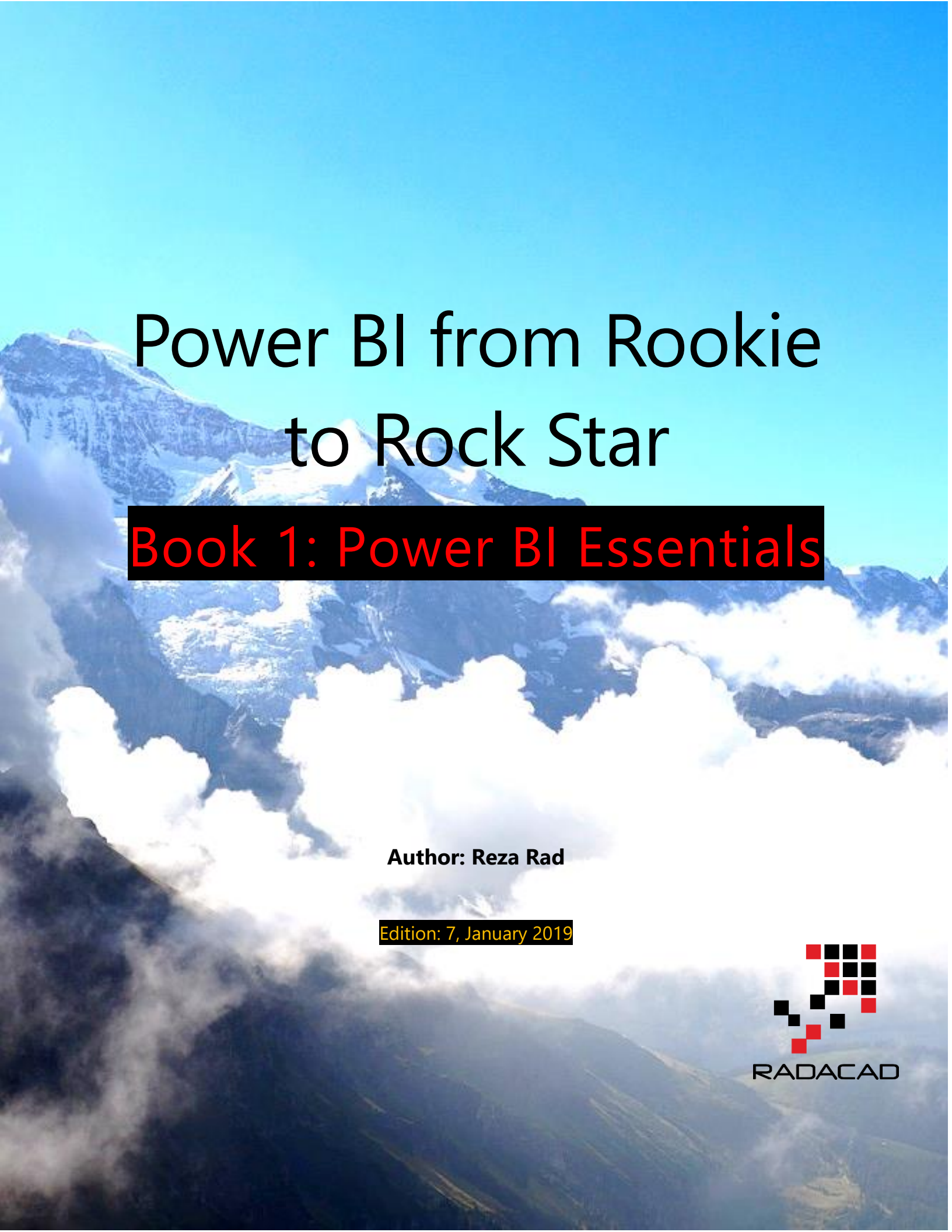


Power BI from Rookie to Rock Star

Book 1: Power BI Essentials

Author: Reza Rad

Edition: 7, January 2019

RADACAD

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

24 Riverhaven Drive,
Wade Heads,
Whangaparaoa 0932
New Zealand

Cover: Virgin Silberhorn Ebnefluh, Pixabay

About the New Edition and New Structure

The Power BI from Rookie to Rock Star been such a popular book from the time that it published, and I added content to it every single week. After edition 3 which released July 2017, there have been many contents added. The edition 3 itself was more than 1100 pages, and If I wanted to continue the book as an all-in-one, it would have been more than 2000 pages now. So I decided to break the book into a book series. Each book in this series is a complete book and can be read individually. However, each book covers a specific area of the Power BI, and if you want to learn Power BI from ground zero to sky hero, you would need to read them all. Here is the new structure:

- Book 1: Power BI Essentials
- Book 2: Visualization with Power BI
- Book 3: Power Query and Data Transformation in Power BI
- Book 4: Power BI Data Modelling and DAX
- Book 5: Pro Power BI Architecture

This book is the book one of the series. In this book, you will learn about all components of Power BI. You will learn each component in a beginner to intermediate level. You will learn from getting data, to do minor modeling, and visualizing the data. You will also learn about the Power BI website, and sharing reports through that environment. Book one is not a deep dive book. It is a book that rises you from the ground zero level to an intermediate level of understanding all Power BI components. If you are looking for deeper diver learning, your answer is within books 2 to 5.

About the book; Quick Intro from Author

In July 2015, after the first release of Power BI Desktop, I had 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with the fast pace of updates for Power BI Desktop, it is impossible to publish a paperback book because it will be outdated in few months.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 the 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 a 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 outdated with new changes in Power BI. I will do my best to update any changes in the 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 a Microsoft Regional Director, an Author, Trainer, Speaker and Consultant. He has a BSc in Computer engineering; he has more than 15 years' experience in data analysis, BI, databases, programming, and development mostly on Microsoft technologies. He is a Microsoft Data Platform MVP for eight continuous years (from 2011 till now) for his dedication in Microsoft BI. Reza is an active blogger and co-founder of RADACAD. Reza is also co-founder and co-organizer of Difinity conference in New Zealand.
His articles on different aspects of technologies, especially on MS BI, can be found on his blog: http://www.radacad.com/blog.
He wrote some books on MS SQL BI and also is writing some others, He was also an active member on online technical forums such as MSDN and Experts-Exchange, and was a moderator of MSDN SQL Server forums, and is an MCP, MCSE, and MCITP of BI.
He is the leader of the New Zealand Business Intelligence users group. He is also the author of very popular book Power BI from Rookie to Rock Star, which is free with more than 1100 pages of content.
He is an International Speaker in Microsoft Ignite, Microsoft Business Applications Summit, Data Insight Summit, PASS Summit, SQL Saturday and SQL user groups. And He is a Microsoft Certified Trainer.
Reza's passion is to help you find the best data solution; he is Data enthusiast.

Who should read this book?

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 a 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 a 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.

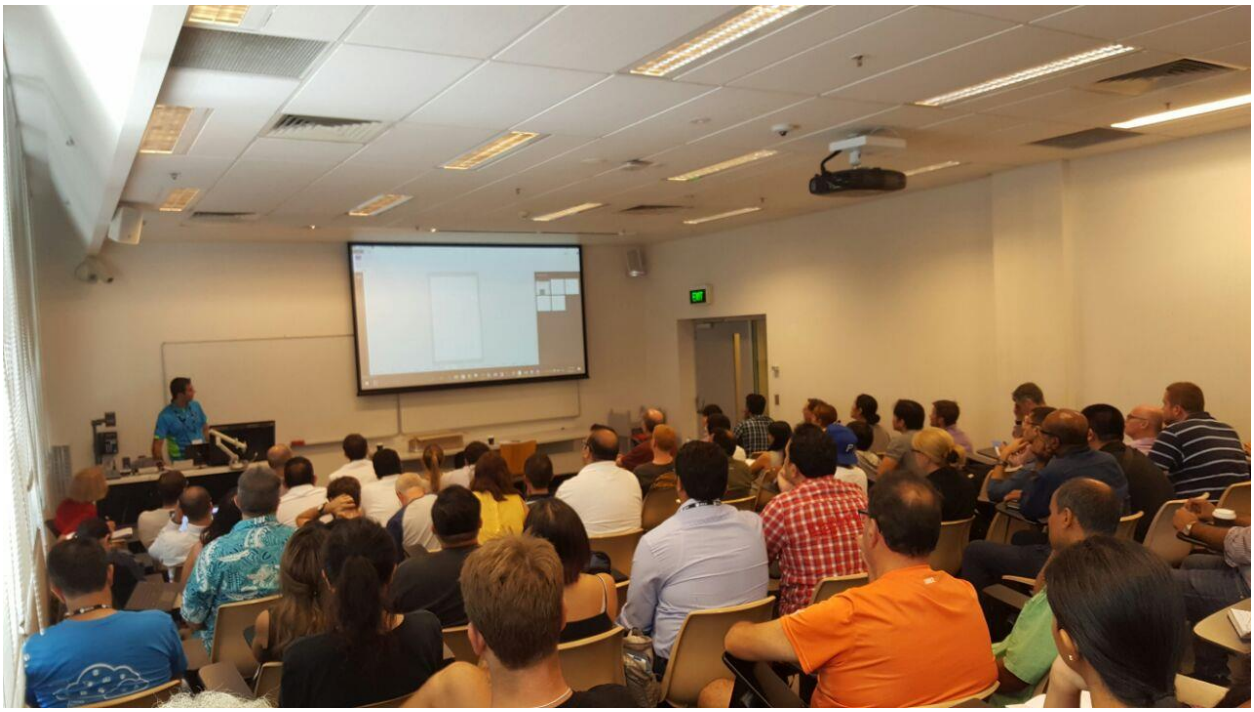
Upcoming Training Courses

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 the schedule of upcoming courses here:

http://radacad.com/events

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

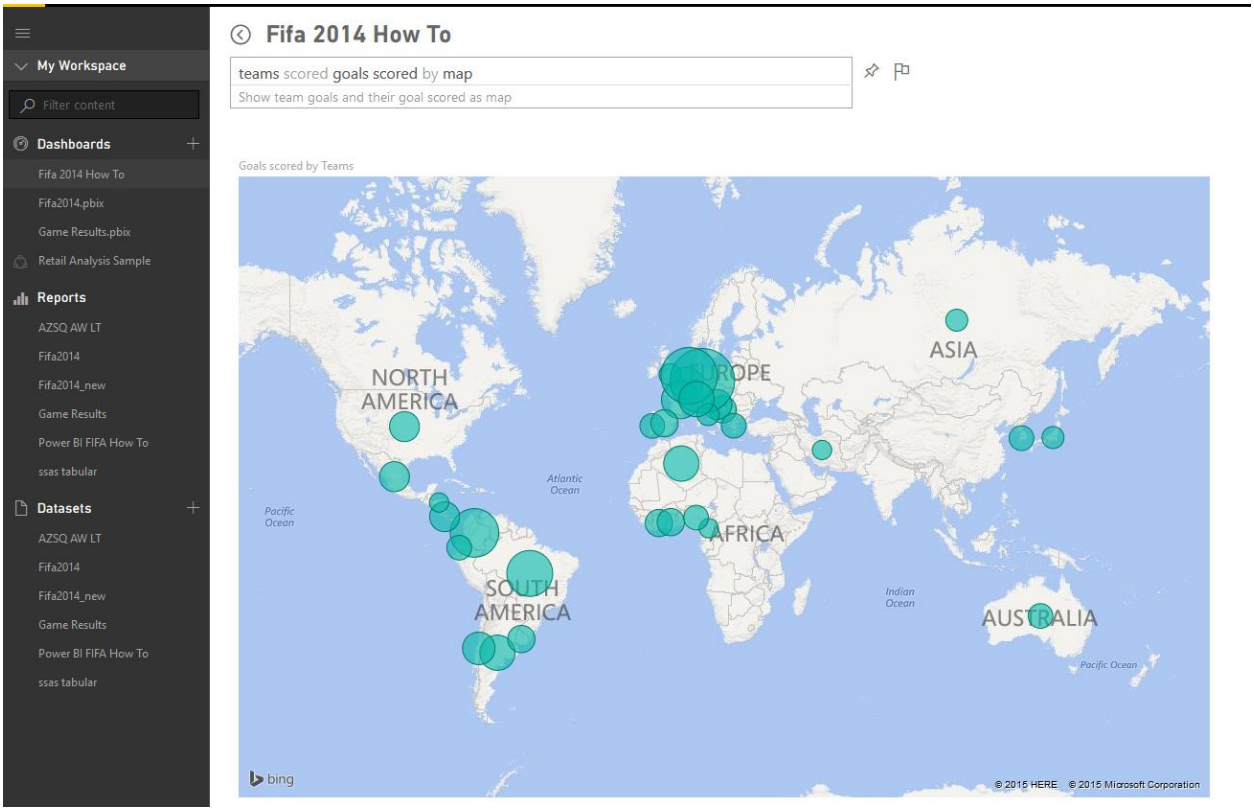


Heading Table of Content

Detailed Table of Content

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 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 in 2010 for the first time. Microsoft team worked through a long period 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 a 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 mashup 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 a 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 the cloud and share it with 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 a 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 modeling 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 modeling tool
- Power View: Data visualization tool
- Power Map: 3D Geospatial 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 the above components are strong and has been tested for a 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 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 mashup 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 the 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 a 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 a combined editor of Power Query, Power Pivot, and Power View. Power BI Desktop is available as a 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

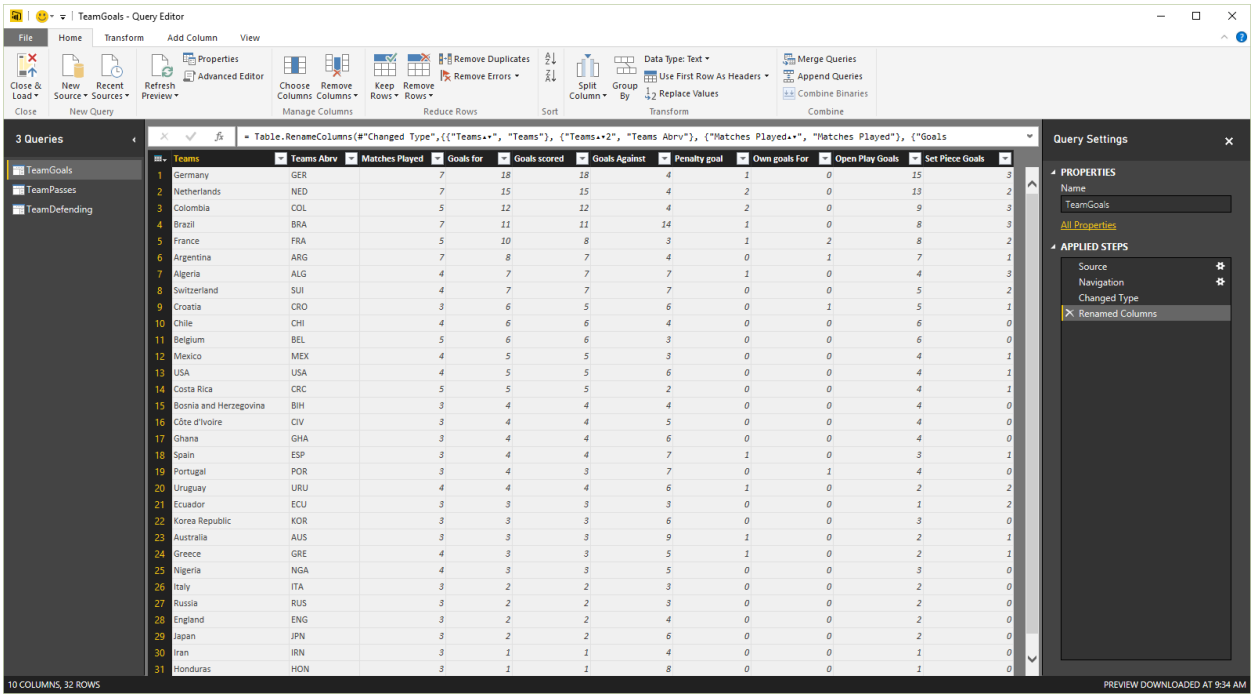
To give you an overall view of what you would expect to see in each component I've put a short explanation for each component here. There will be a detailed description of all components later on in future chapters.

Power Query

Power Query is data transformation and mashes up the 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 a 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

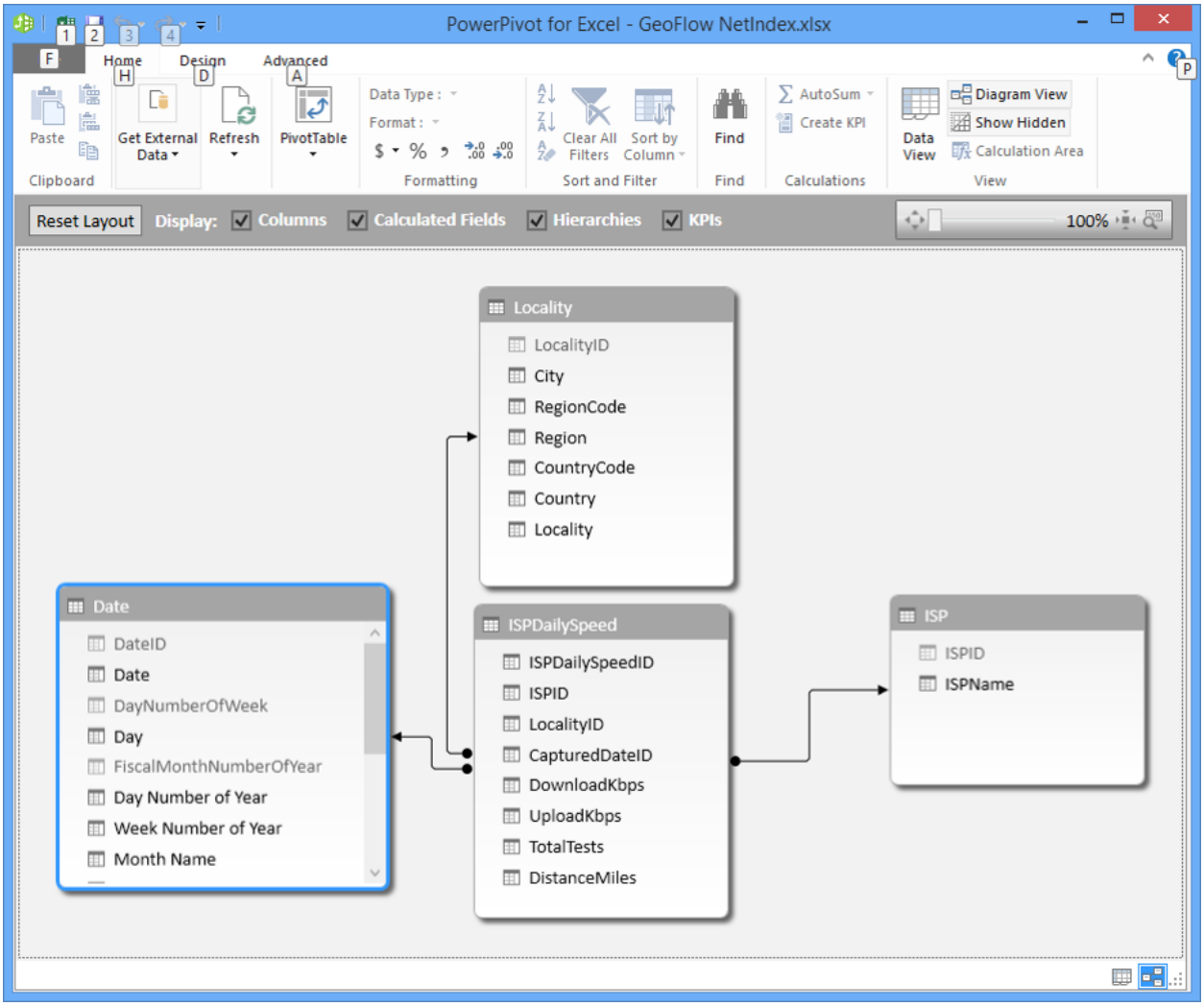
user interface to transform data as you need, adding columns, changing types, transformations for date and time, text, and many other operations are available. Power Query can load the result set into Excel or Power Pivot model.

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 functionalities in M that cannot be accessed through the 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. the screenshot below is a view of Power Query editor, and some of its transformations.



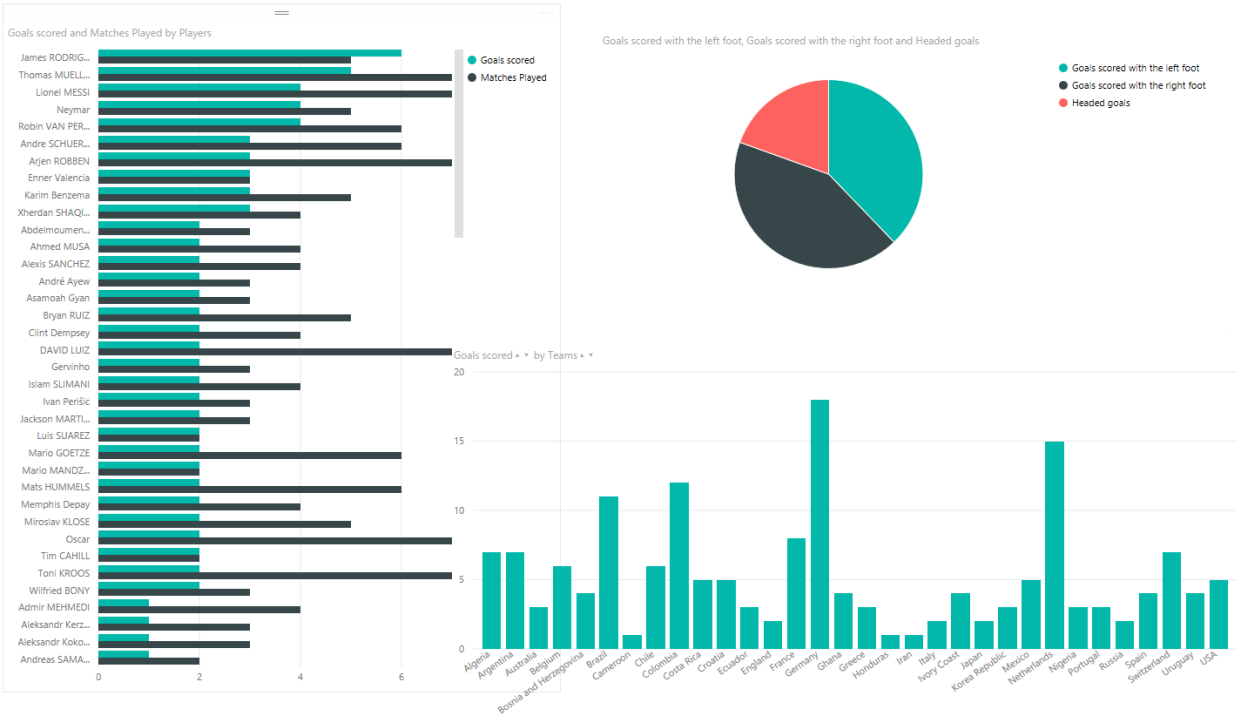
Power Pivot

Power Pivot is data modeling engine which works on xVelocity In-Memory based tabular engine. The In-Memory engine gives Power Pivot super fast response time, and the modeling engine would provide you with 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 the details of Power Pivot modeling and DAX in future chapters. The screenshot below shows the relationship diagram of Power Pivot



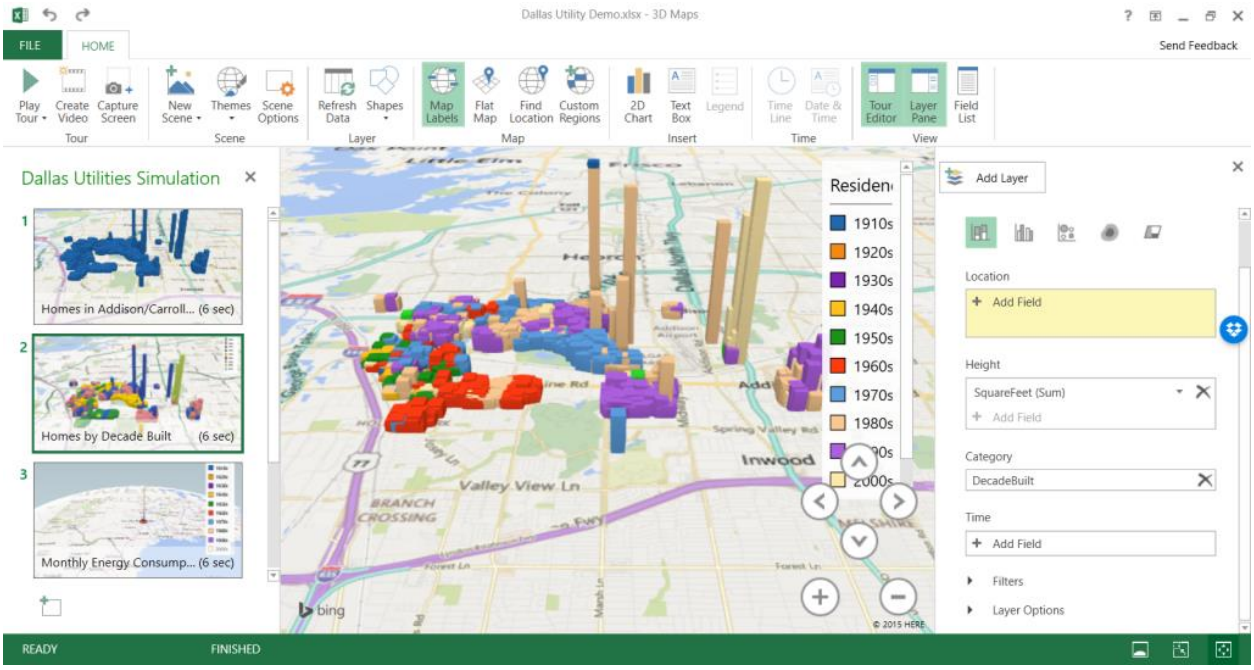
Power View

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 the ability to filter data for each data visualization element or 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.



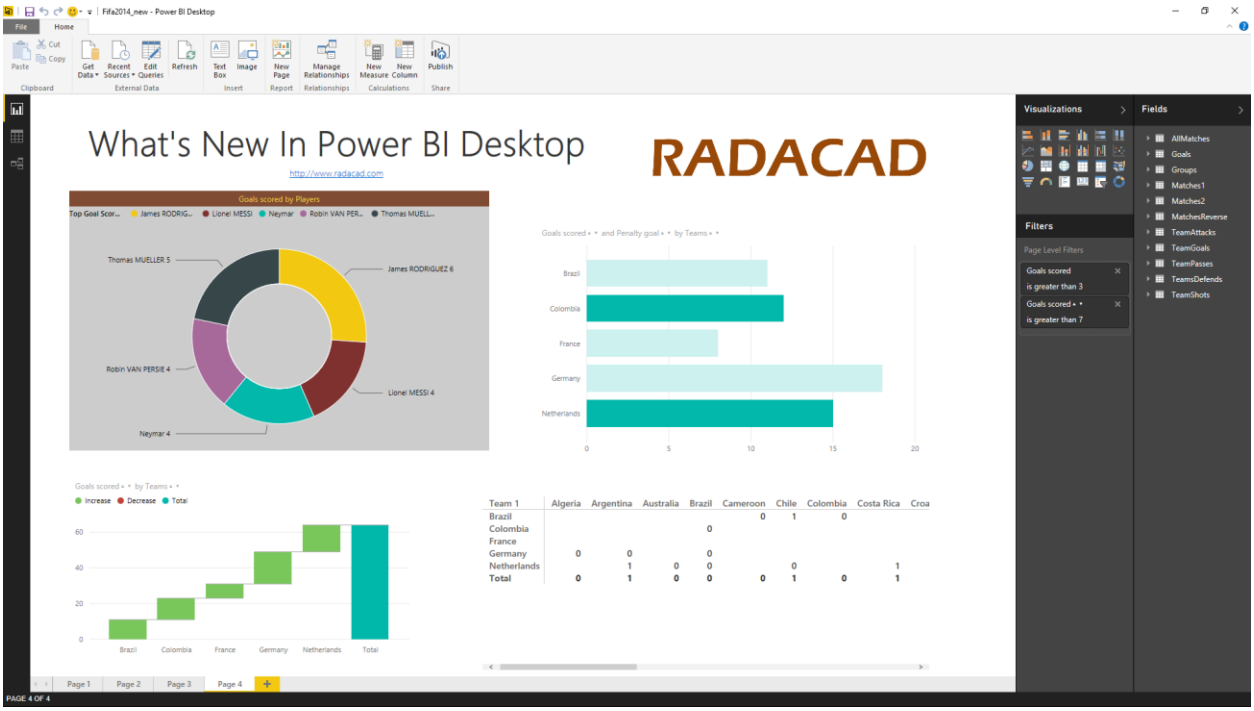
Power Map

Power Map is for visualizing Geospatial 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 the 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 the best visualization based on Geographical 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.



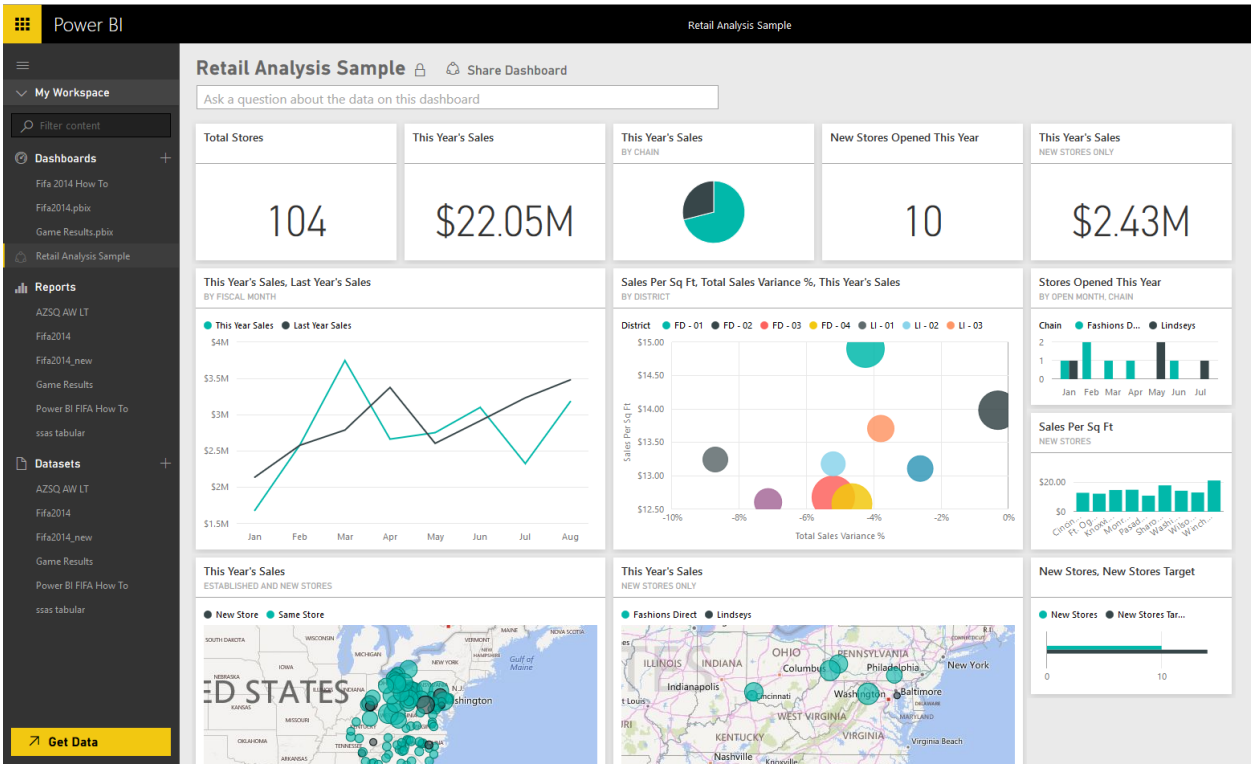
Power BI Desktop

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 the 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 some time with the 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 section "Power BI Hello World" with a demo of this product. You can have a better view of the newest features of Power BI Desktop here in this blog post. the screenshot below shows a view of this tool;



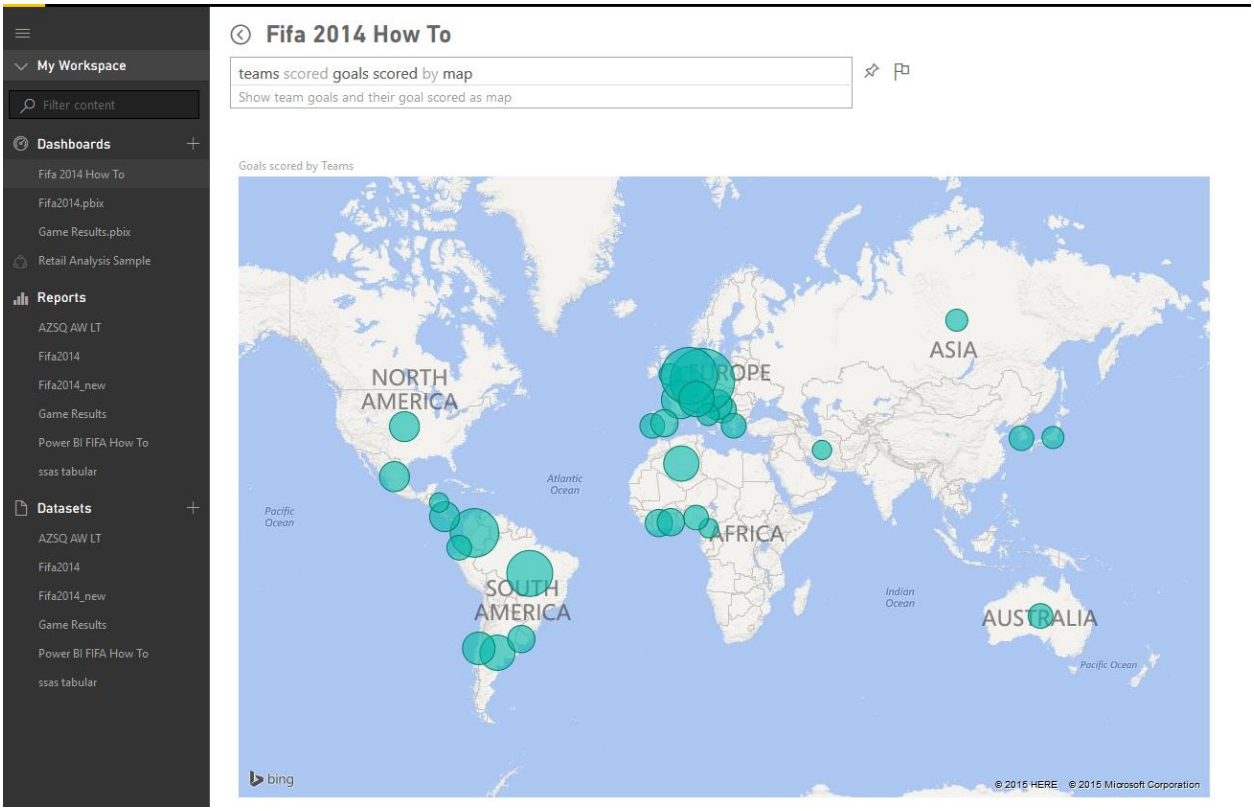
Power BI Website

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 build report and visualizations directly on Power BI site as well. the screenshot below shows a view of Power BI site and dashboards built there;



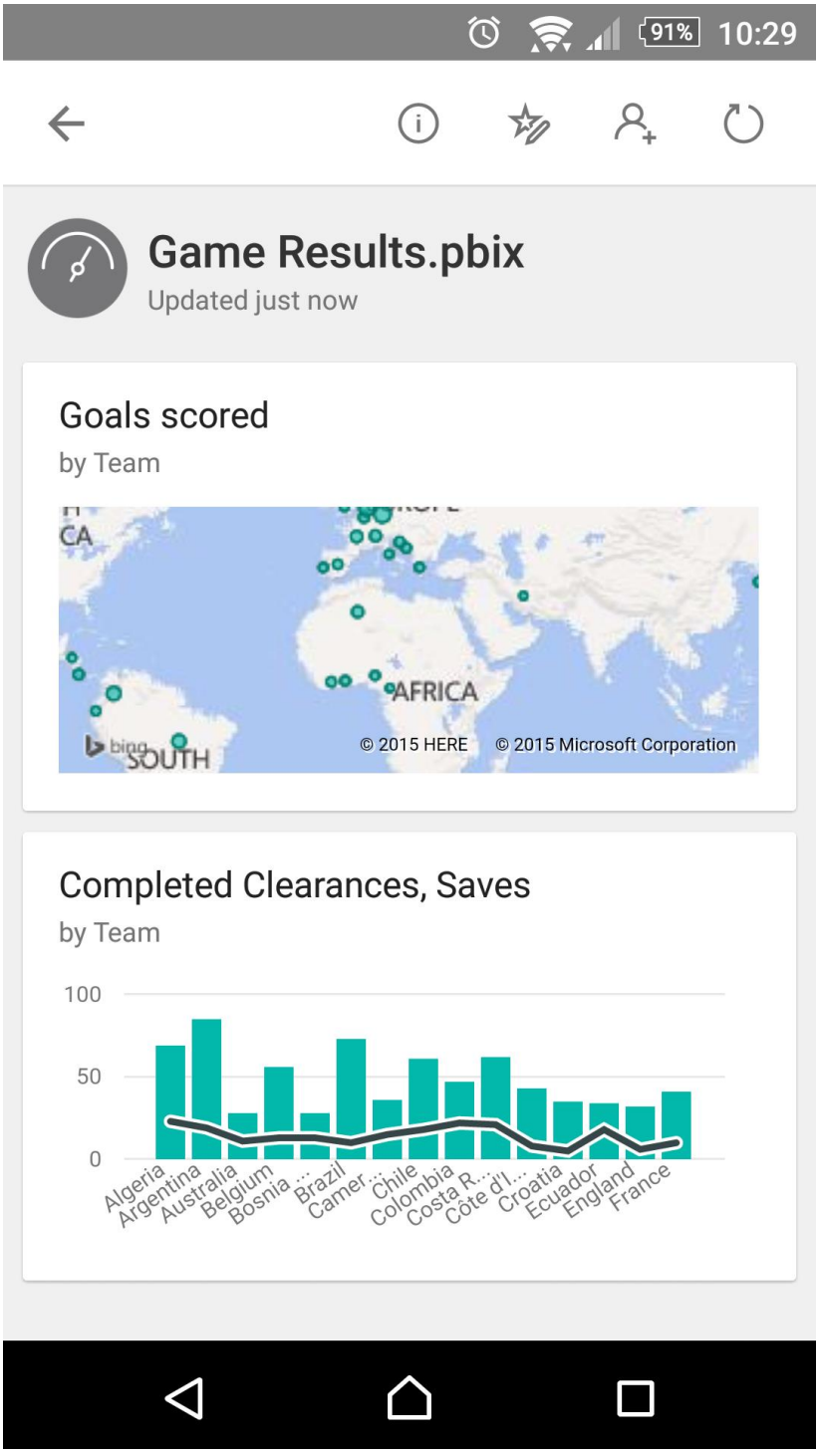
Power Q&A

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 and works with Power View for the data visualizations. So users can 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.



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 the 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 worth it.

Preparation

To follow examples of this book download the latest version of Power BI Desktop from here. Most of the 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 the 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, 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 power that can be used to answer complex BI requirements. In the 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 the 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 the Power BI website where users can see the report and interact with it from a 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 mashes 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 visualizes the data with different charts and visualization elements.

Power BI Desktop has everything in one editor, and this makes it a tool which is very easy to use, but don't be a 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 its 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:

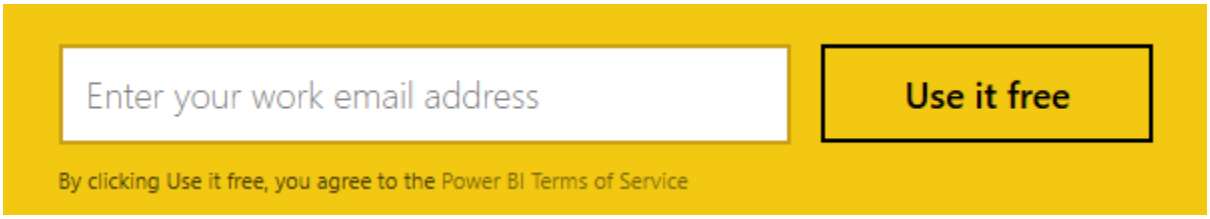
Download Power BI Desktop from this link. You can also install 64 or 32-bit options if you go to advanced download options.

Sample Database:

For this example, I used the old Pubs database (to create something different than AdventureWorks samples). You can download it here and install the Pubs database.

Power BI Account:

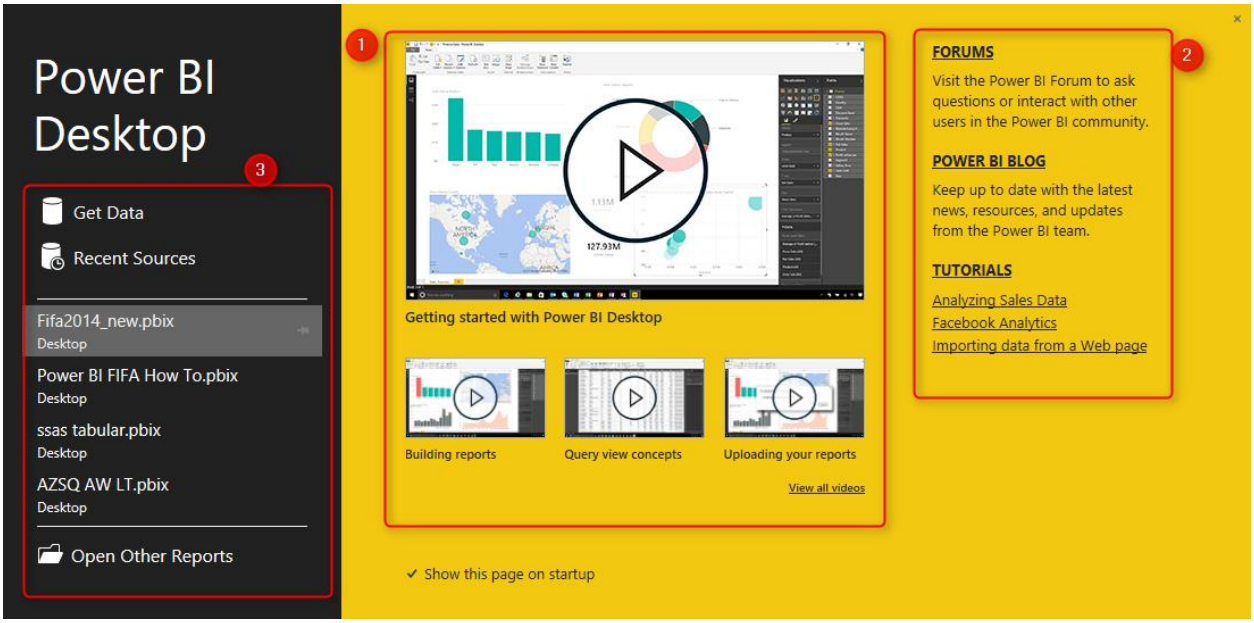
If you don't have a Power BI account, simply go to Power BI website, and enter your company mail address to get a free account. Note that you cannot use public email accounts such as Gmail and yahoo here



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:

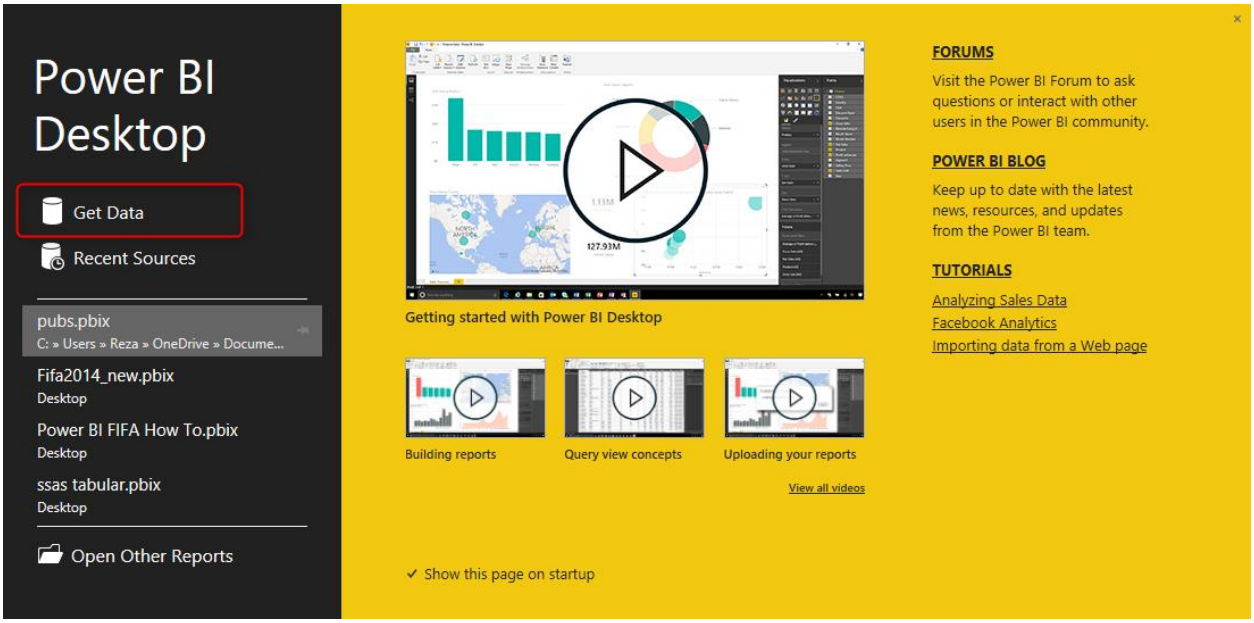
1. you can watch some Power BI Desktop tutorial videos.
2. You can also read the latest news about Power BI or some tutorials of Power BI, or you can go to Power BI forums to ask a question or search for some answers there.
3. You can open an existing report with Power BI, or start building a new report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



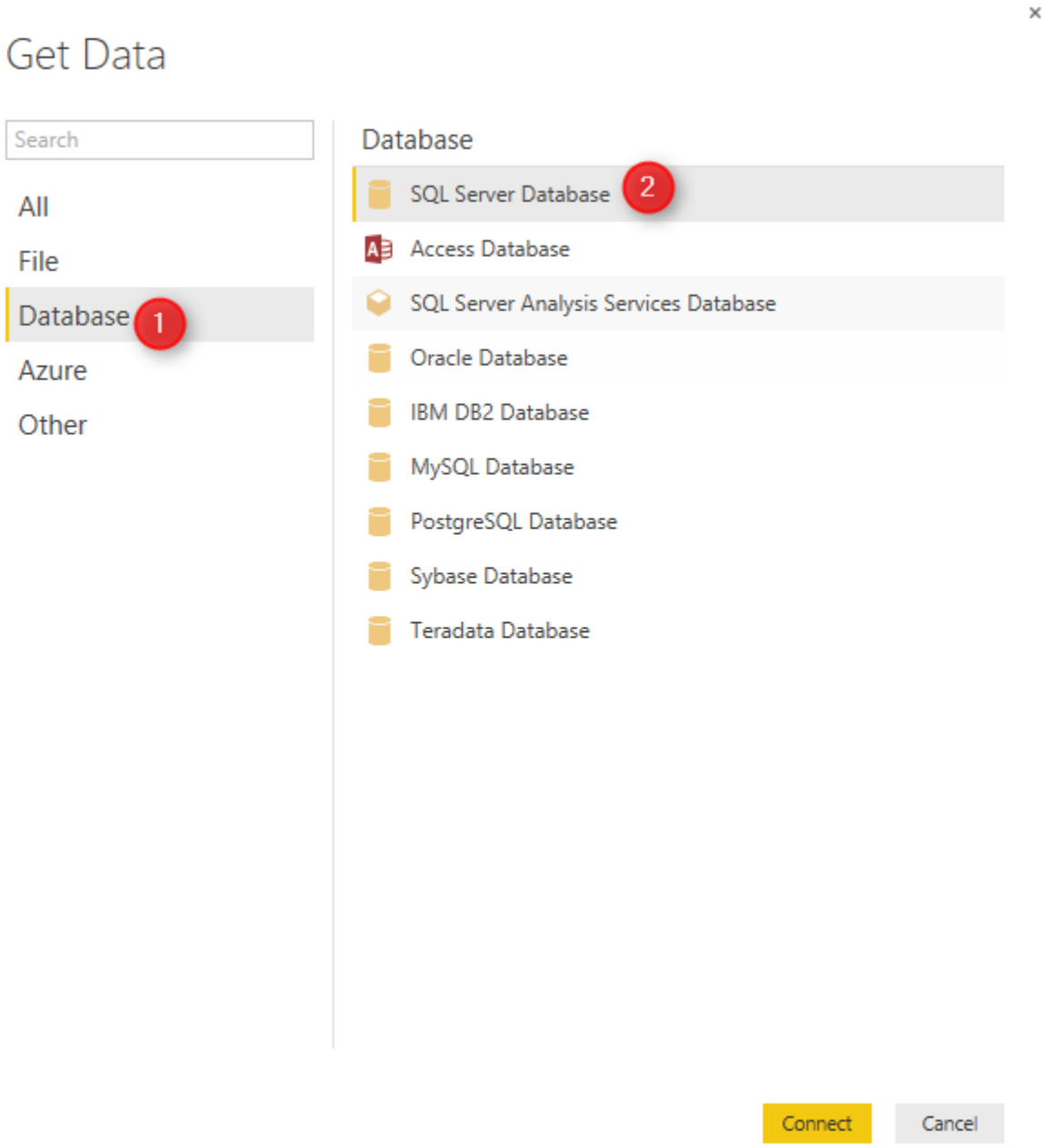
Get Data

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, 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 the 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 the Pubs database, and you'll apply a very simple transformation to it.

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

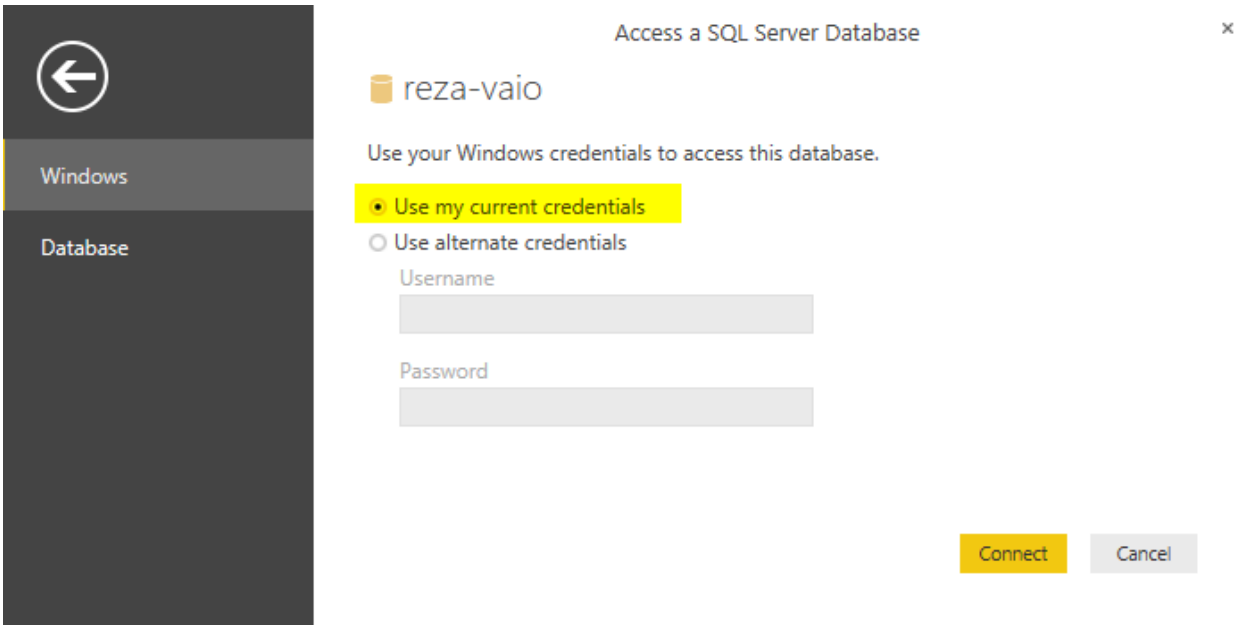


Get Data window will open. This window is the graphical interface to connect to many data sources. From the left options choose Database, and in the main section select SQL Server Database. (Remember we are connecting to SQL Server instance that has Pubs database installed)

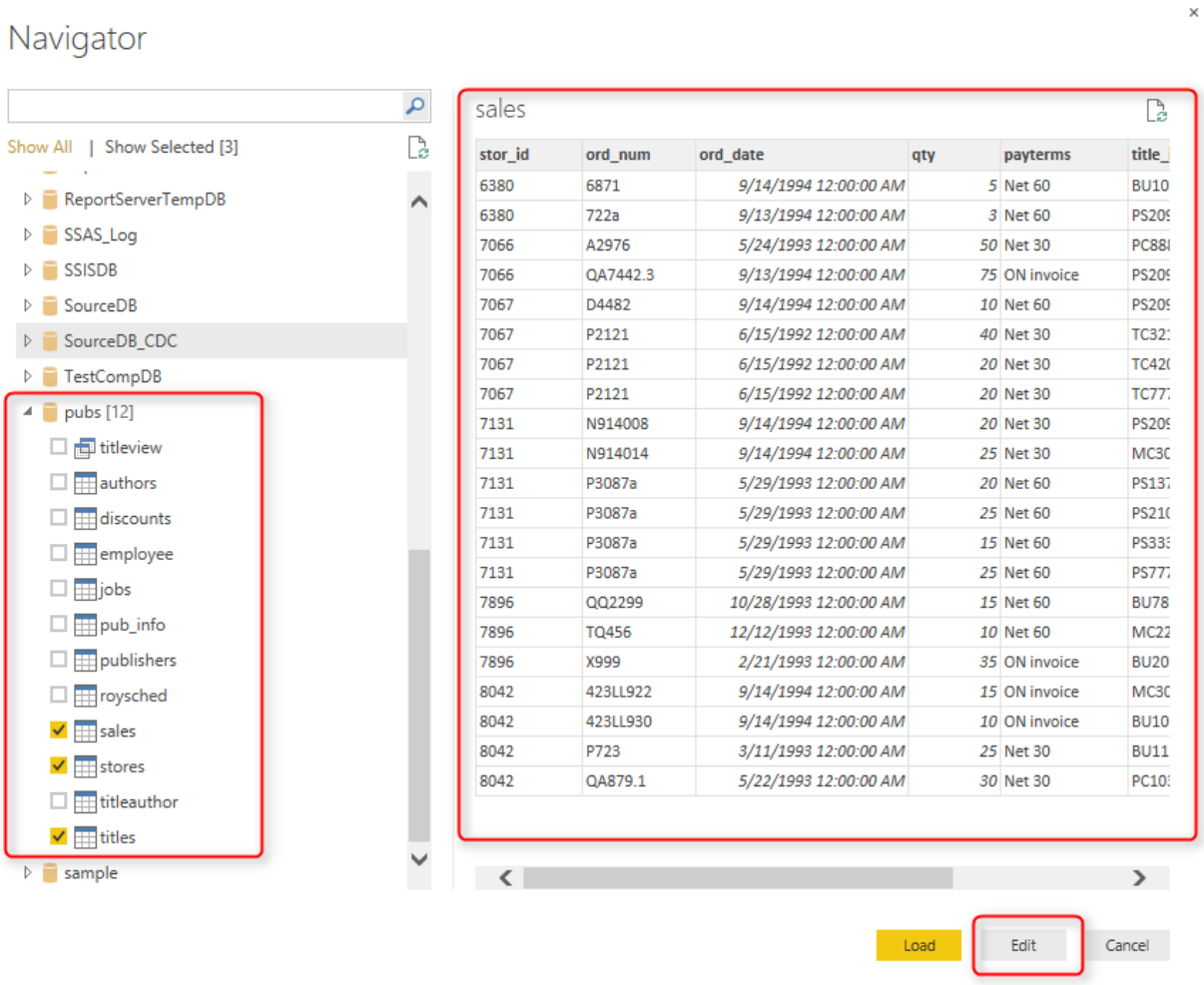


Next dialog box asks for server name and database name. The server name is necessary, but you can leave database name blank (blank will load all databases to choose from). You can also write a SQL statement here if you want to. because my server for the database is my local computer, so I enter my computer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

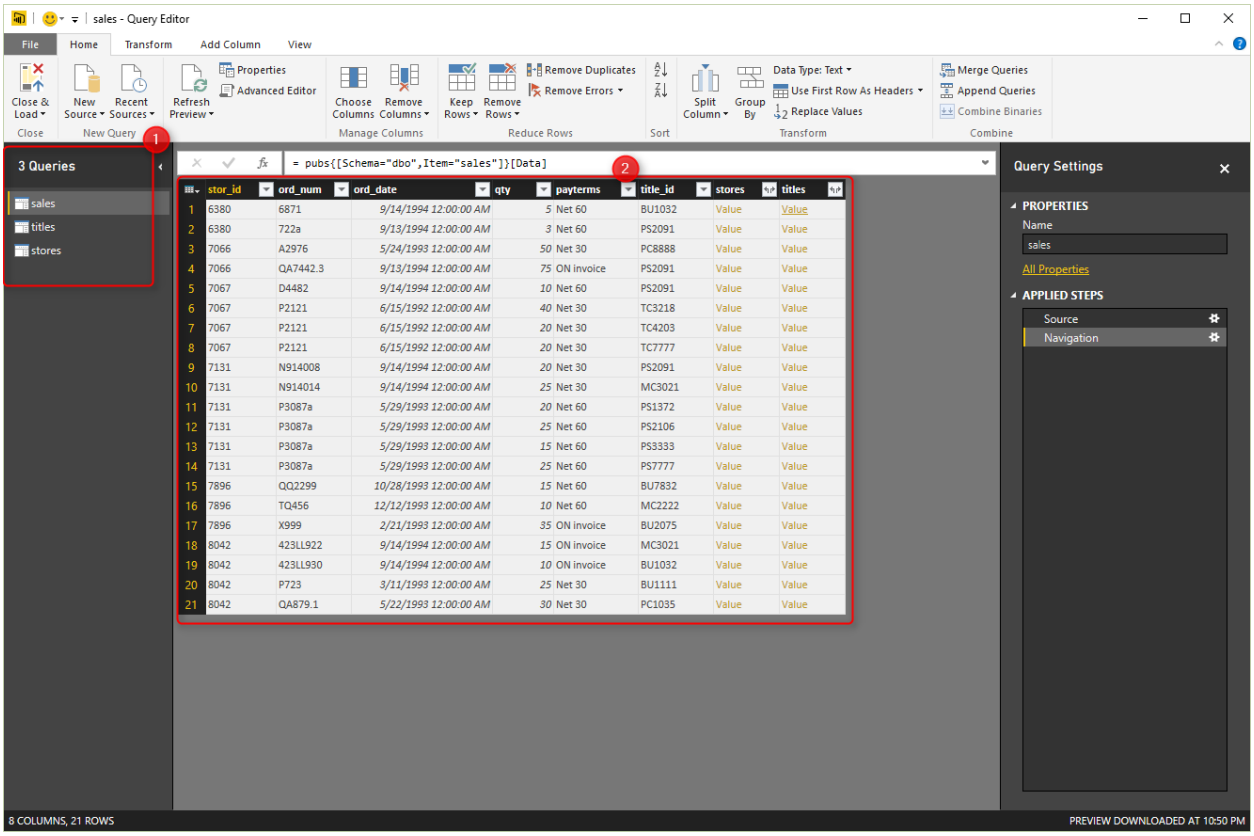
You can set up credential in the next window. I use my current credentials, but note that there are other options such as entering an alternate credential or using database (SQL Server authorization)



you might get a message about encryption support, 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.

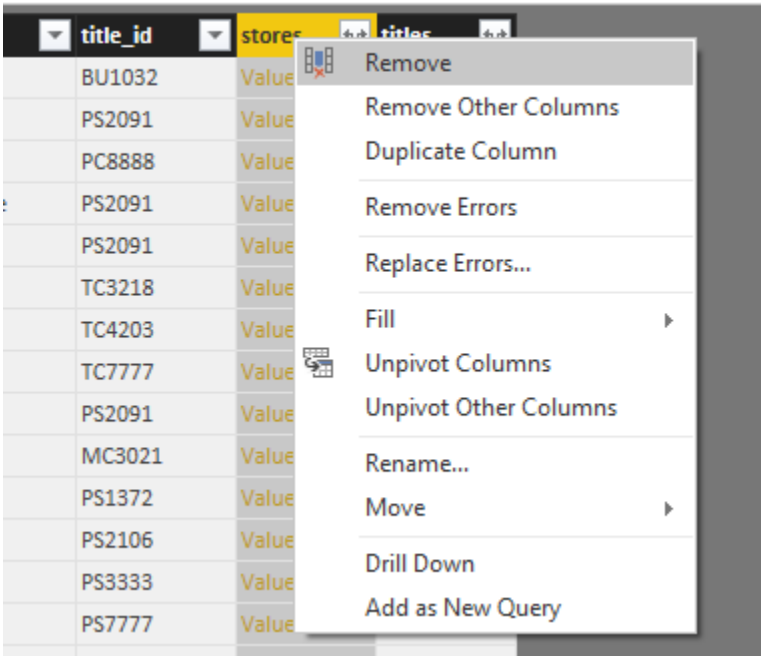


As you see in above screenshot when you click on tables, you'll see a sample data rows of that appearing in the main section. This helps you to have an understanding of what you are looking at. Click on Edit. The Query Editor window will open. This is the main Power Query Window that has all the tables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



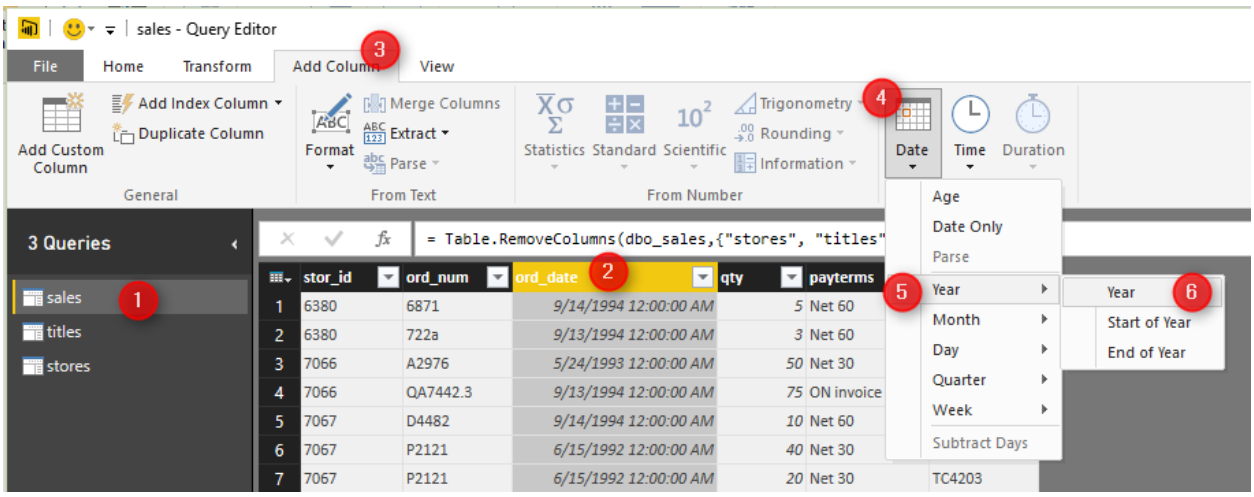
In Query Editor window in the left pane, you can see three tables that we've loaded (numbered as 1). Main pane (numbered as 2) shows the data set from a table in the left-hand side. Right-hand side shows the steps applied to the data set. The menu on top contains some transformations that you can apply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

Select sales table in the 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 the 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 select Remove.



Do the same in "titles" table, and remove columns: publishers, 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 has columns for year, months, week, etc, I 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 a sales query, then click on the ord_date column, and then from the main menu click on Add Column. In the Add Column section under Date, choose Year.

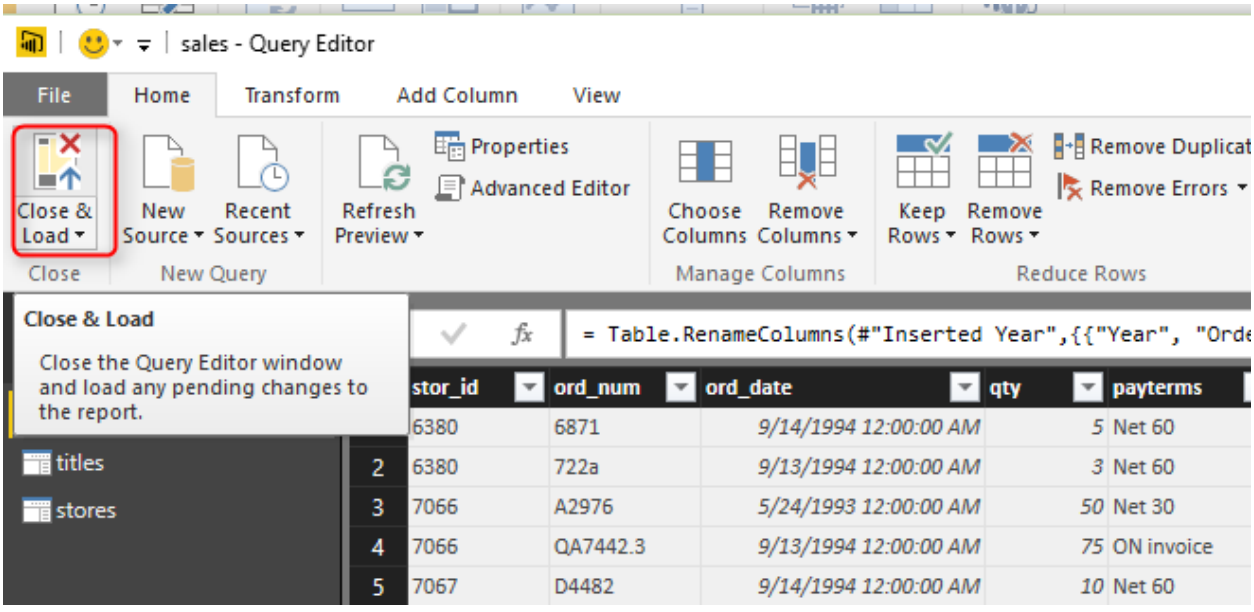


A new column with the name of Year will be added to the table, right click on the column and rename it to Order Year. Well done, you've done your first transformation with Power Query. It was so simple, wasn't it?

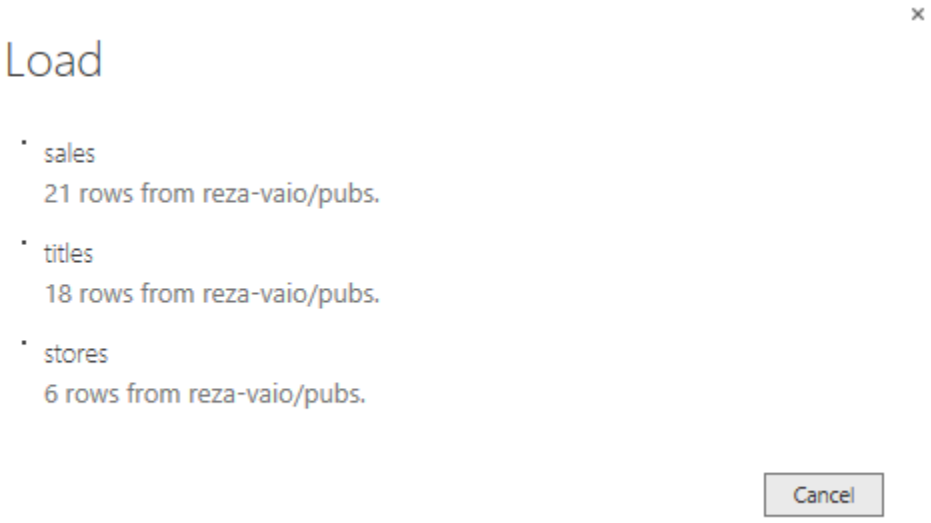
	stor_id	ord_num	ord_date	qty	payterms	title_id	Order Year
1	6380	6871	9/14/1994 12:00:00 AM	5	Net 60	BU1032	1994
2	6380	722a	9/13/1994 12:00:00 AM	3	Net 60	PS2091	1994
3	7066	A2976	5/24/1993 12:00:00 AM	50	Net 30	PC8888	1993
4	7066	QA7442.3	9/13/1994 12:00:00 AM	75	ON invoice	PS2091	1994
5	7067	D4482	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

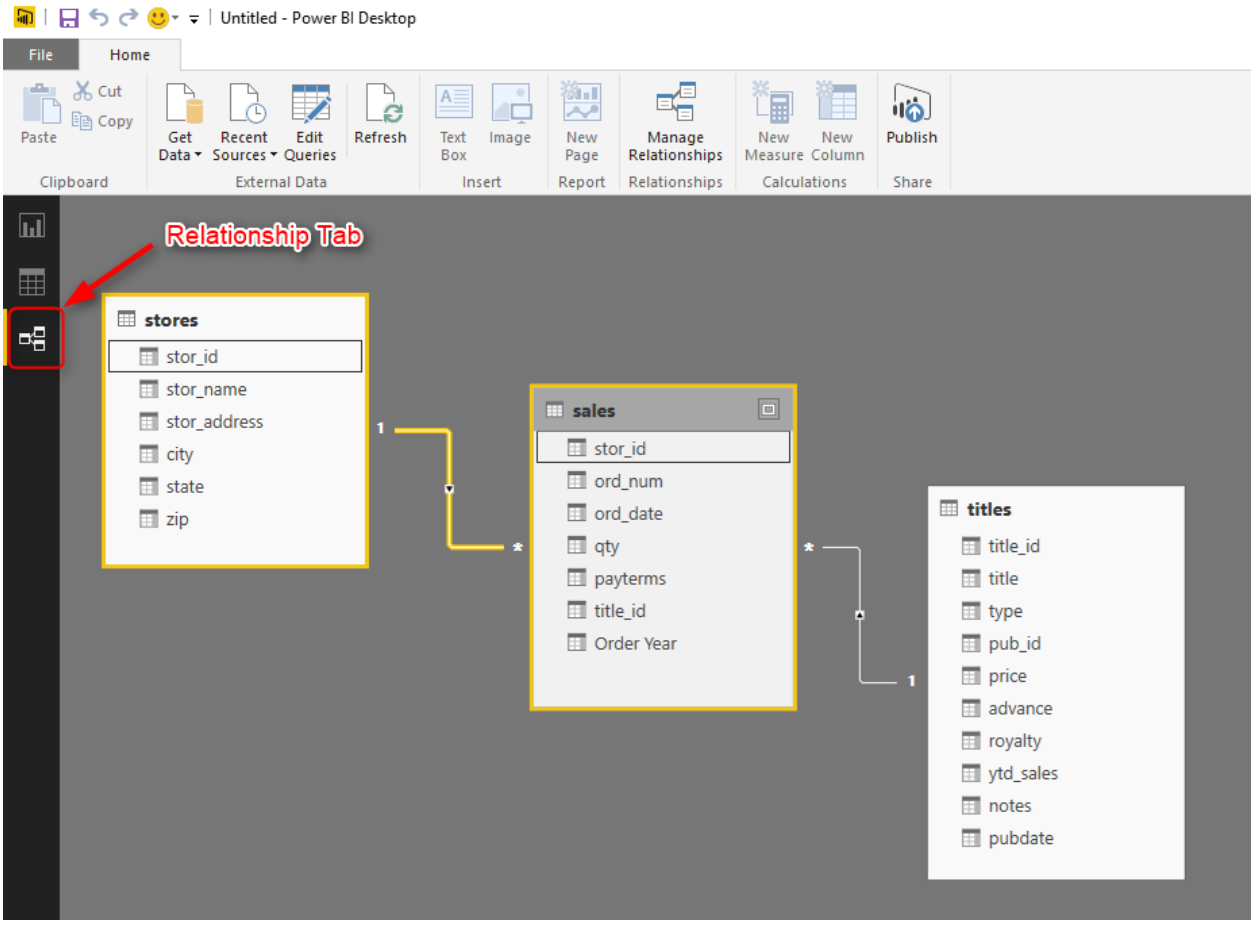
Now let's load data into a model, in Query Editor window under Home tab, click on Close & Load button. This will load all the queries with their data into the model.



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

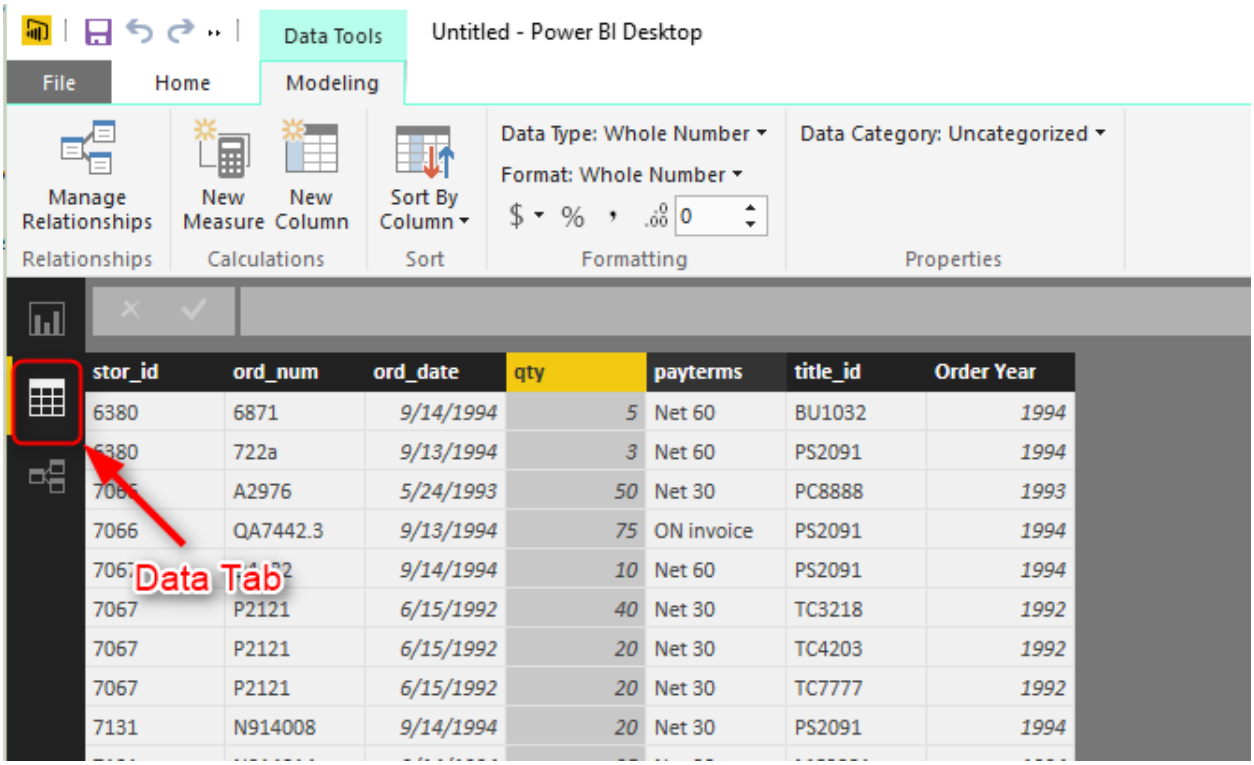


Now you will see the blank Power BI Desktop editor. From the left tabs, click on Relationship (last tab) as below to see the relationship diagram of your model



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

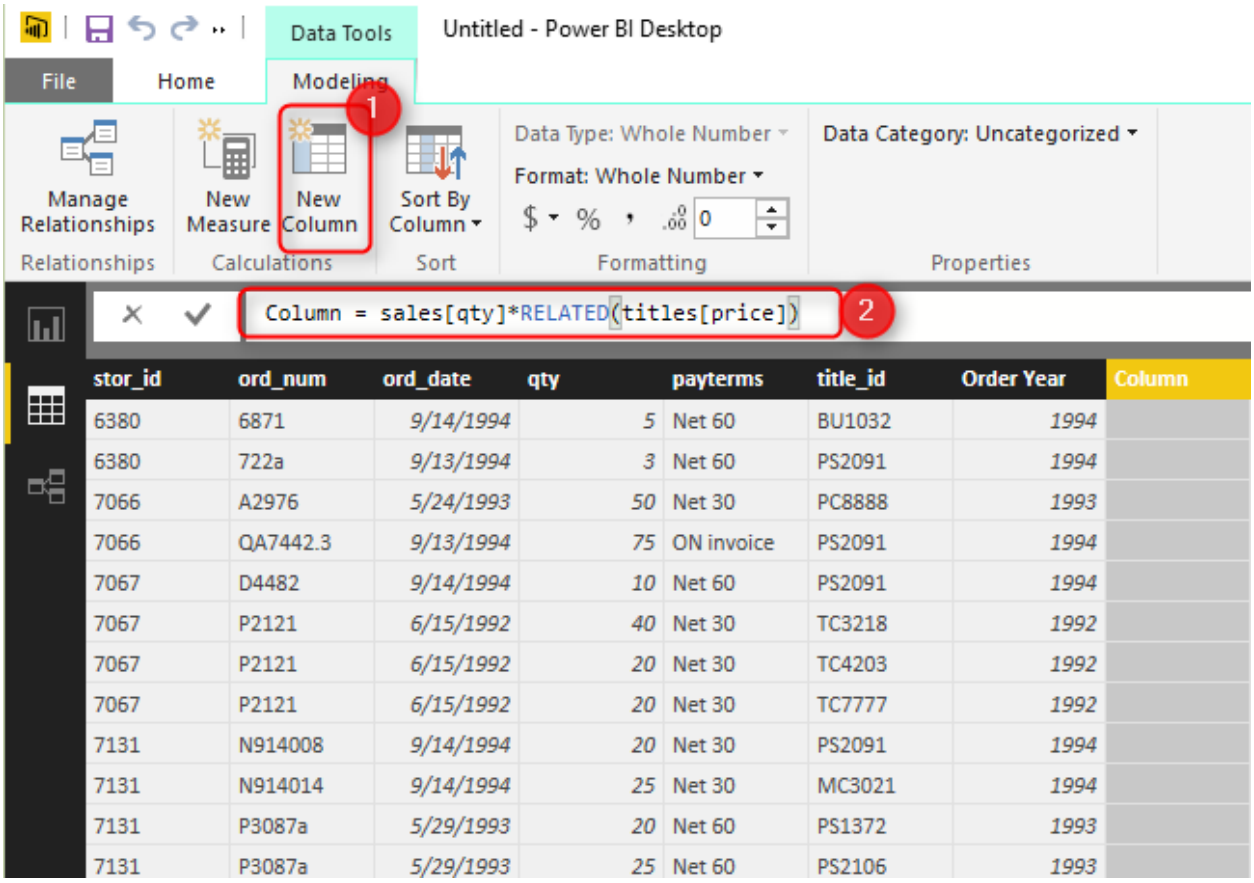
Click on Data Tab to see the structure of data in each table. You can see that qty exists in sales table. If you look at titles, you'll see price column there.



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
7066	A2976	5/24/1993	50	Net 30	PC8888	1993
7066	QA7442.3	9/13/1994	75	ON invoice	PS2091	1994
7067	[illegible]	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

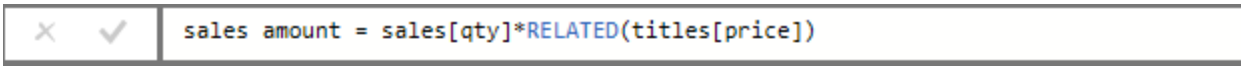
Go to sales table (from right pane choose sales), and then click on New Column. In the formula bar type in the formula below

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



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	

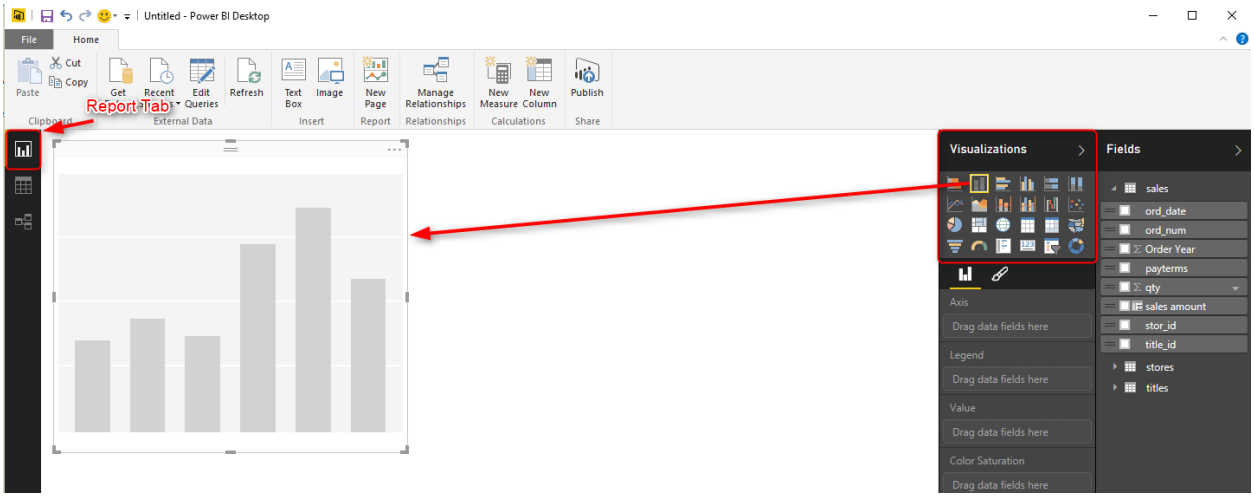
then press enter (you'll see that intellisense also helps you to write down the formula). Now you'll see the result in the new column. Right click on it, and rename it to sales amount. Congratulations! You've written your first DAX expression here. A related function is a function that brings the relative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



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

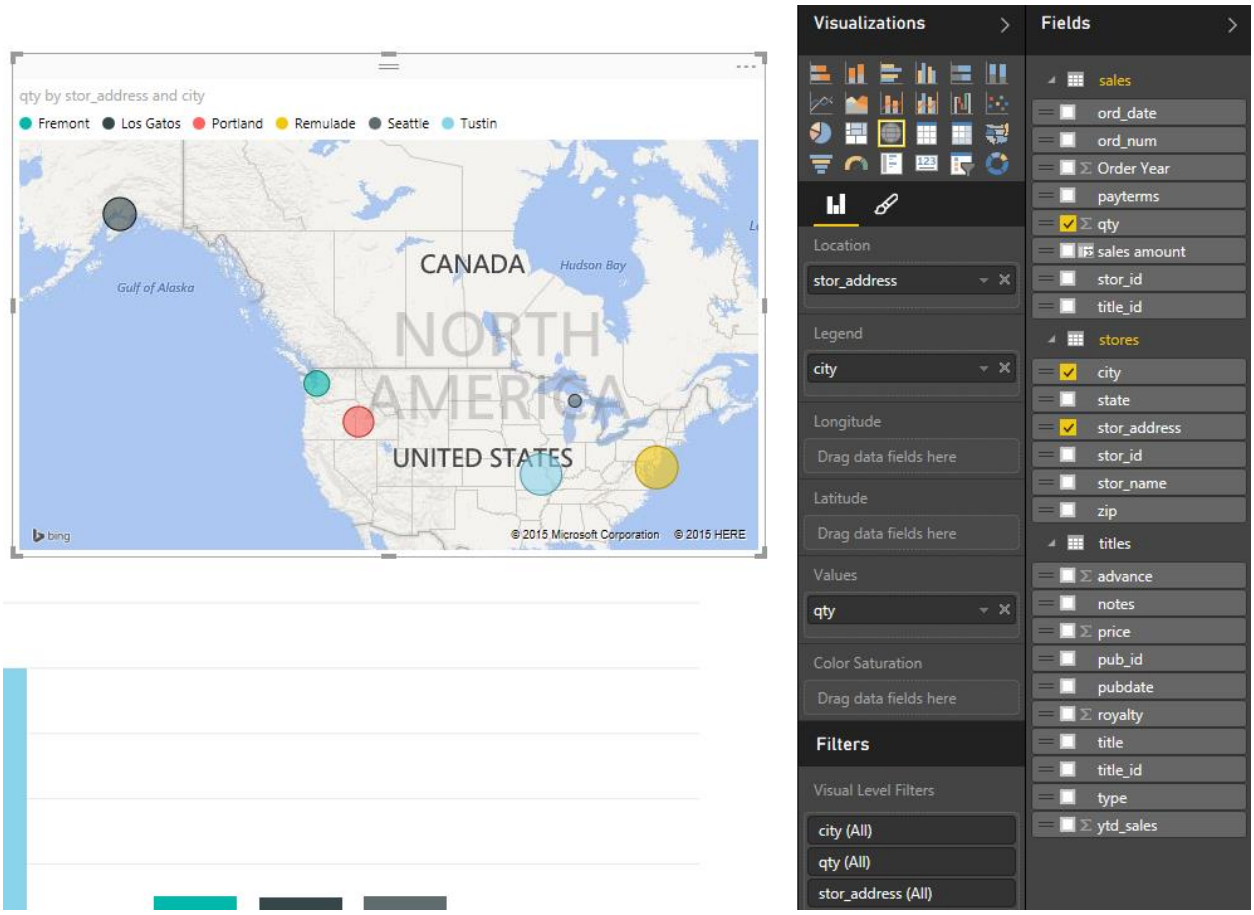
Visualization

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.

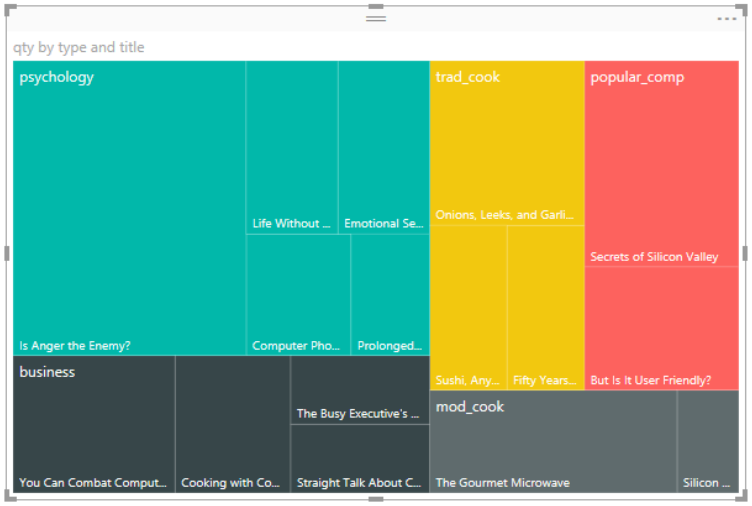


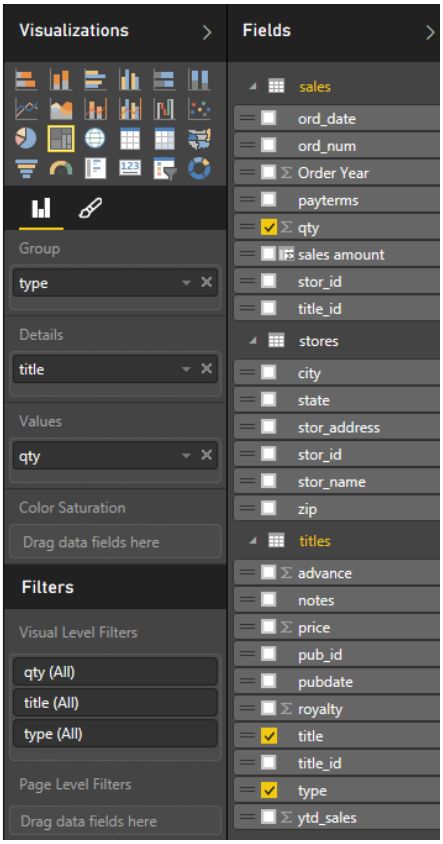
From the fields pane drag and drop sales amount into the Value section of the chart. Then drag and drop city (from stores table) into Legend, and then drag and drop title (from titles table) into Axis. Done! You have a clustered column chart already. you can change height and width of the chart simply (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 the city (from stores table). Your map is now ready.

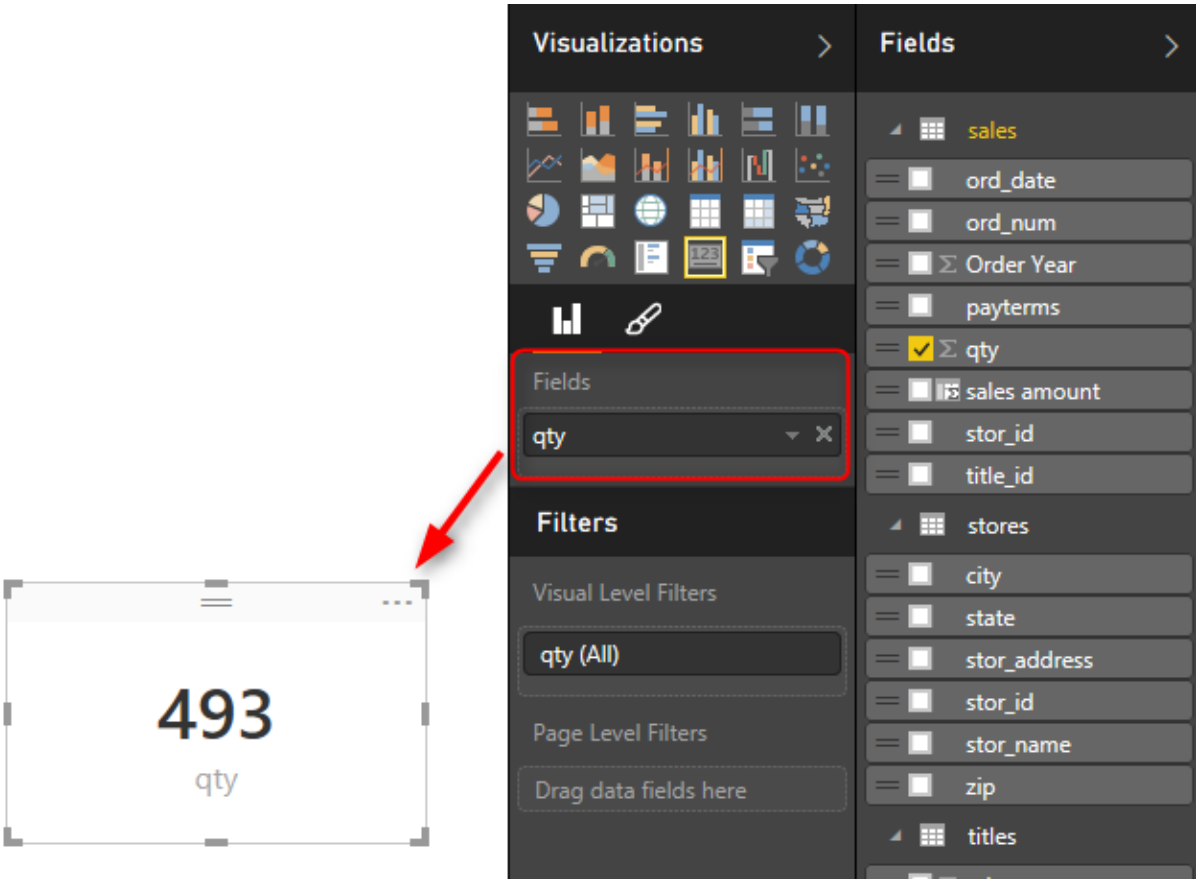


Let's add a treemap to the experience as well. Set the treemap with Group as type (from titles table), Details as the title (from titles table), and Values as qty (from sales table).

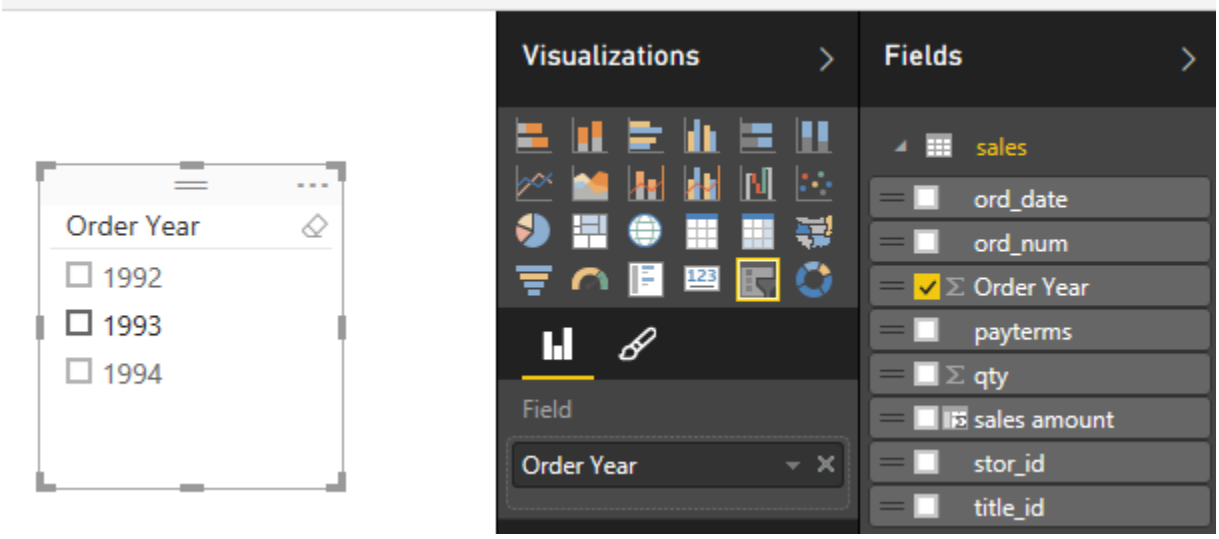




Let's add a total quantity sold as a text box in the report as well. Add a Card from visualization, and choose qty to be displayed there.

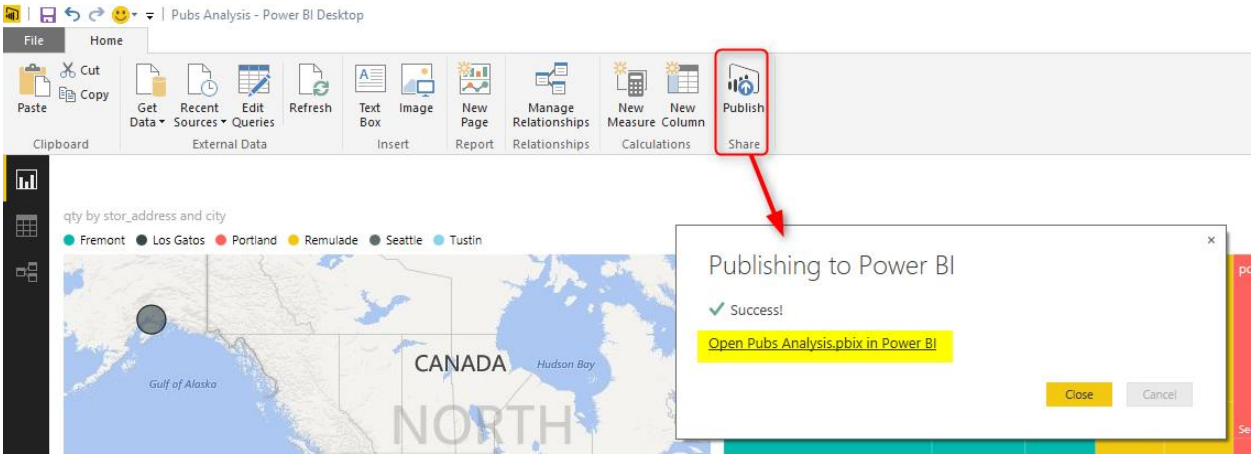


Now add a slicer as the last item for your report. Slicer now can slice and dice the data by Order Year (remember we've added this column in modeling part)

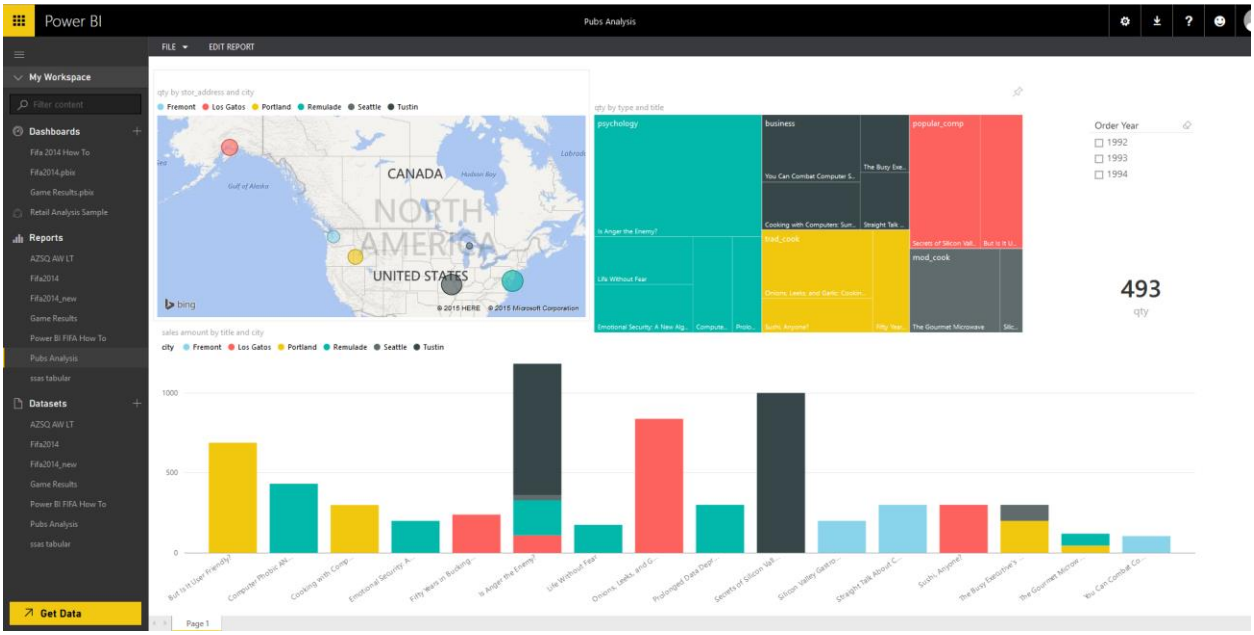


Publish

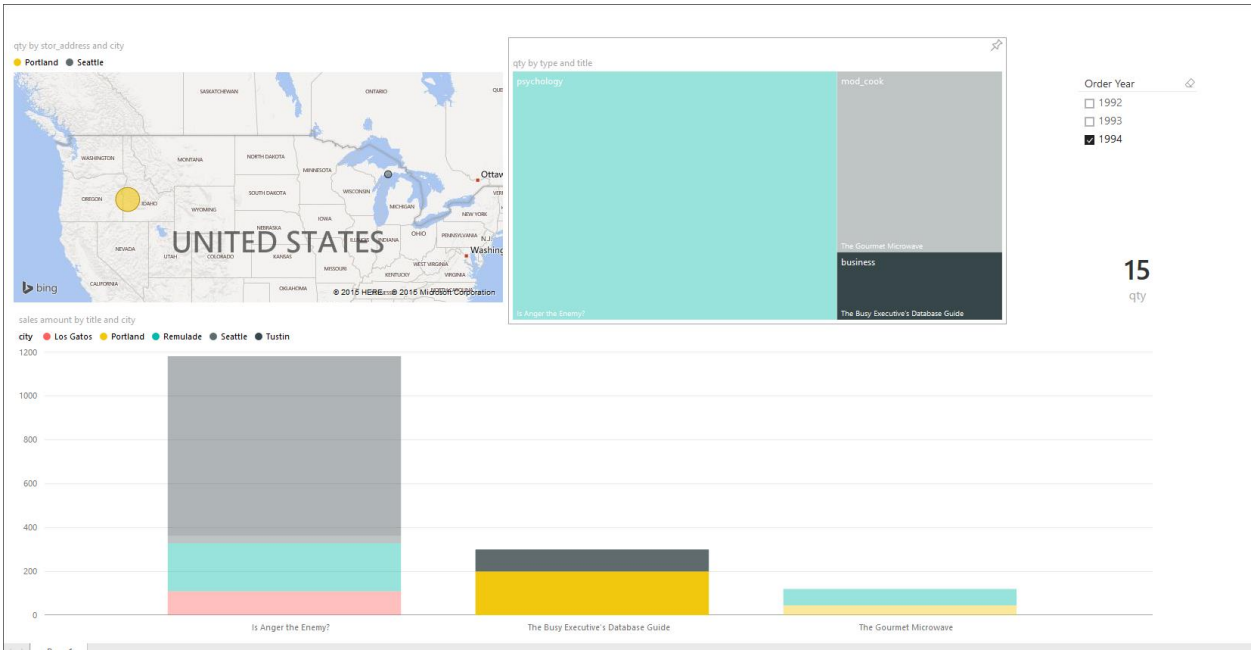
Save the Power BI Desktop file as Pubs Analysis. In the Report tab of the Power BI Desktop, click on Publish (you'll need to enter your account details here for the first time). You'll get a success message when publish is completed.



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



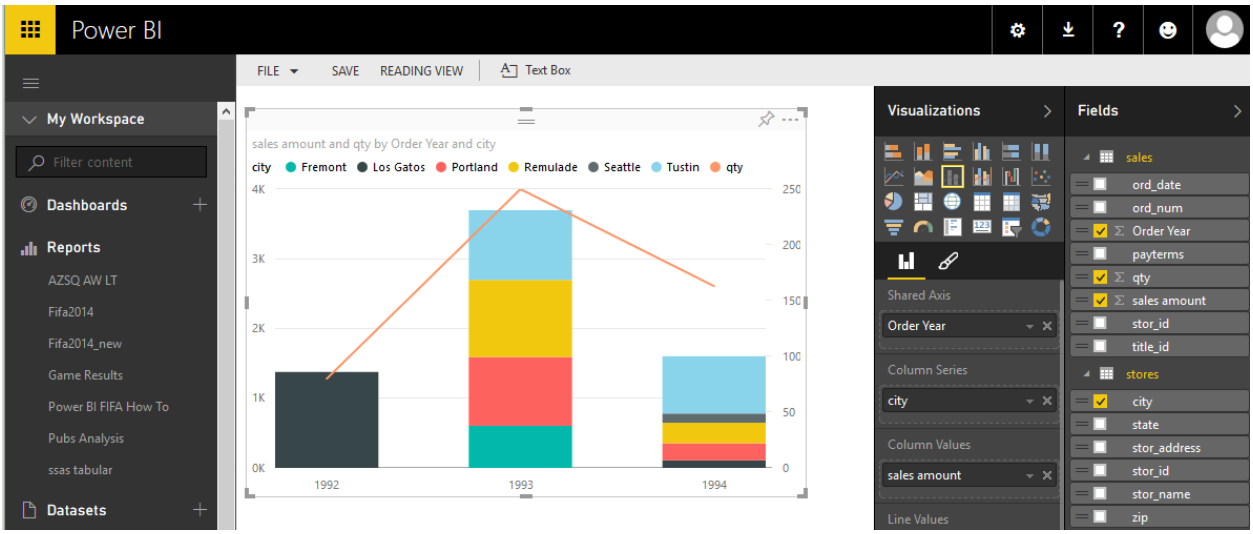
Your report is also fully interactive as well, click on charts and columns and maps to see how highlighting features works interactively.



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 Power BI Desktop is. You've connected to a SQL Server as the 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 on Power BI website. In the next section, we will go to Power BI website and Apps to see how they work.

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

Published Date: August 11, 2015



In the 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 a 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 the 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.

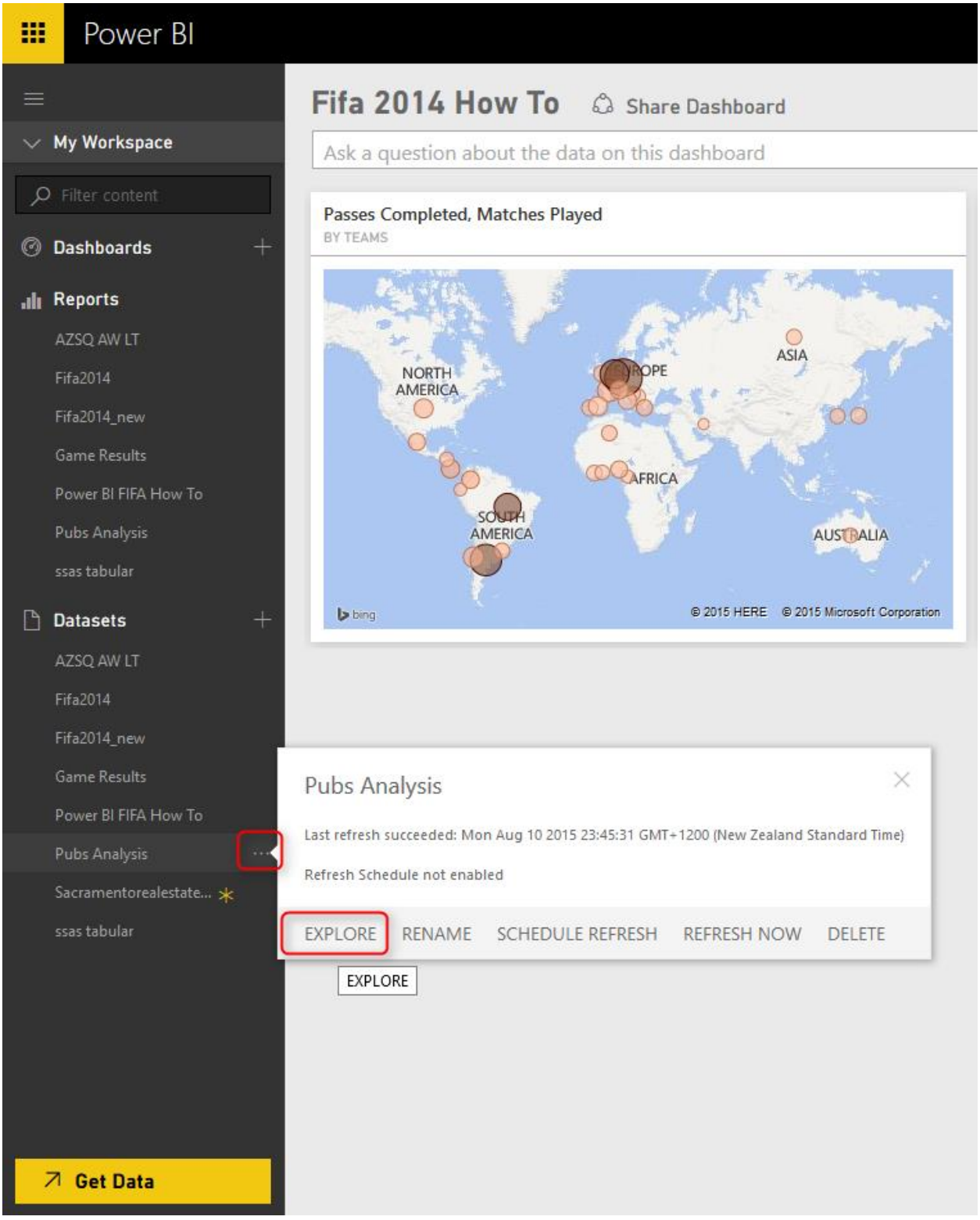
Power BI Website

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 the 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 supports 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

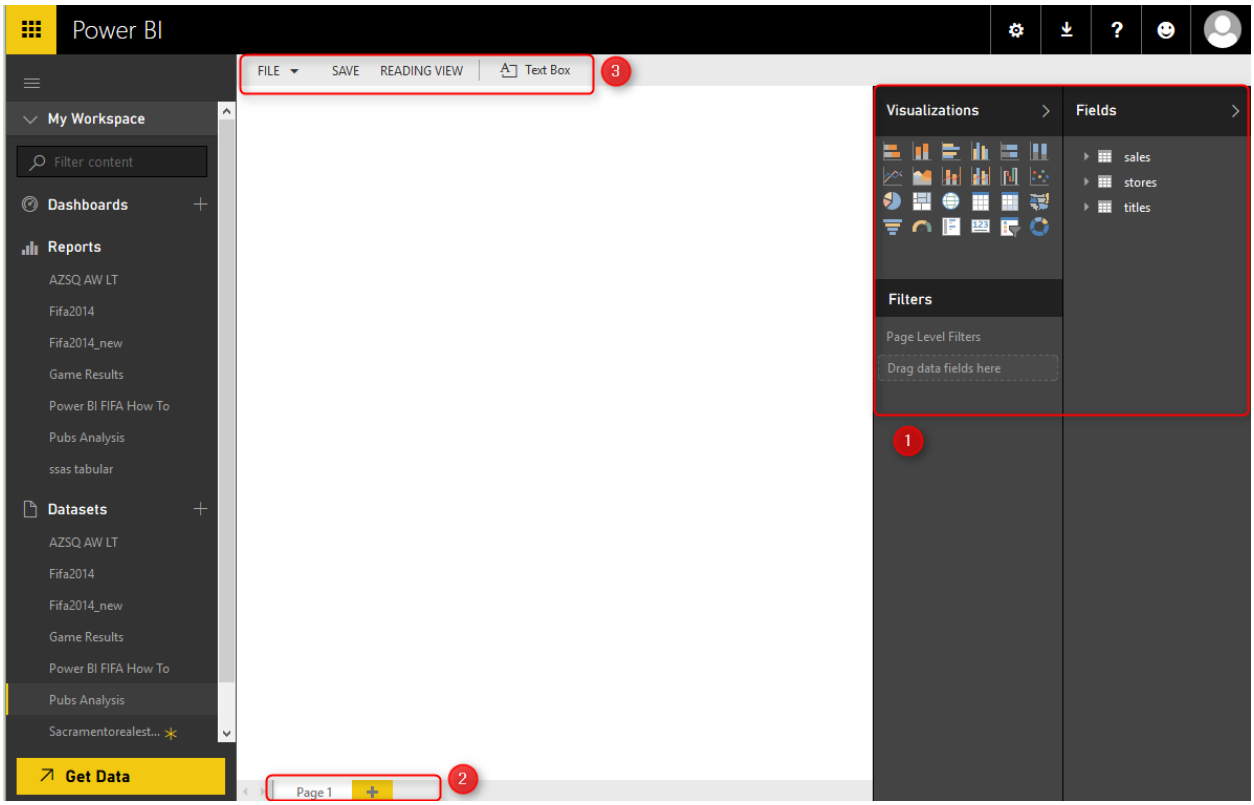
You can edit reports on the website and rewrite them, or you can save them as a new report. You can also create a new report from an existing or new data source. Let's use the data set that we've built in the previous section and use that as the source for our new report. Follow the instruction below to create a report on 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 the 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 contains metadata and the data. Click on ellipsis button near Pubs Analysis dataset and choose to explore from the popup menu.



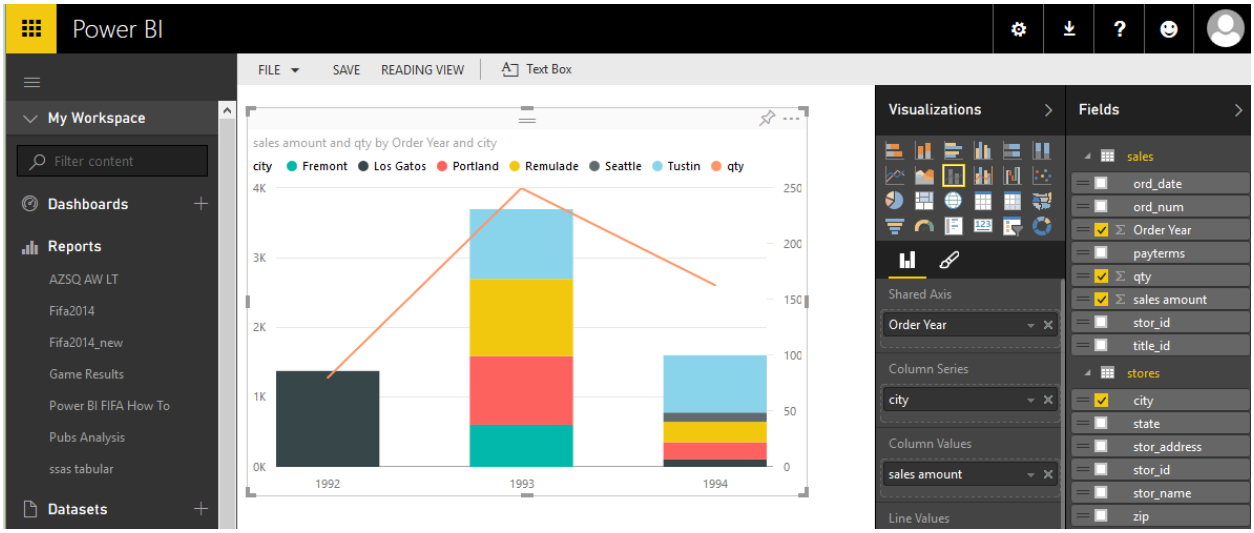
When you click explore, a report window will open in the main window. This is similar to the Report tab in Power BI Desktop. You'll have the Visualization, Fields, and Filters pane

in the right side (numbered as one 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 a screenshot of the report editor online

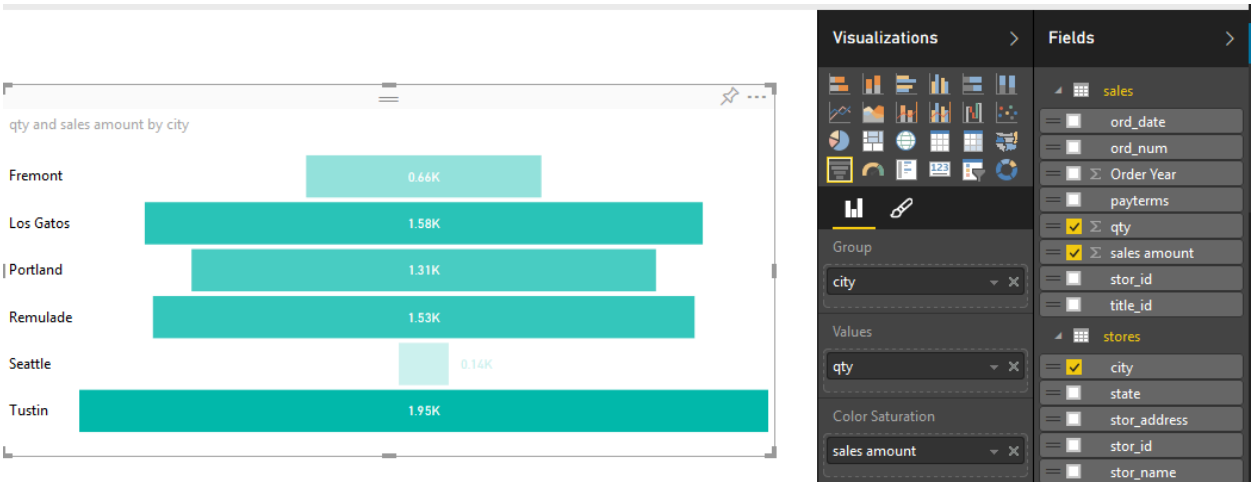


The report editor is not exactly similar to the report tab of Power BI Desktop, there are fewer functions here (For example you cannot add an image here). But most of the functionality is available through the online report editor. And it works perfectly with the web browser. Let's build a simple report;

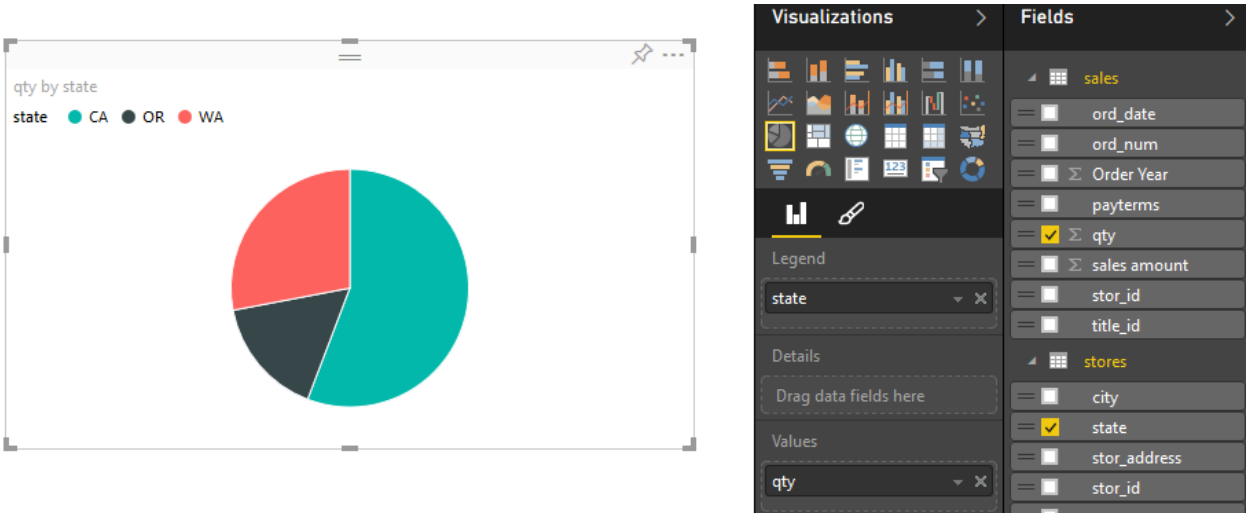
Create a "Line and Stacked Column Chart" from the Visualization panel. Ste Shared Axis as Order Year (from sales table), and Column Series as a city (from stores table), and Column Values as sales amount (from sales table), and finally Line Values as qty (from sales table). The report illustrated below;



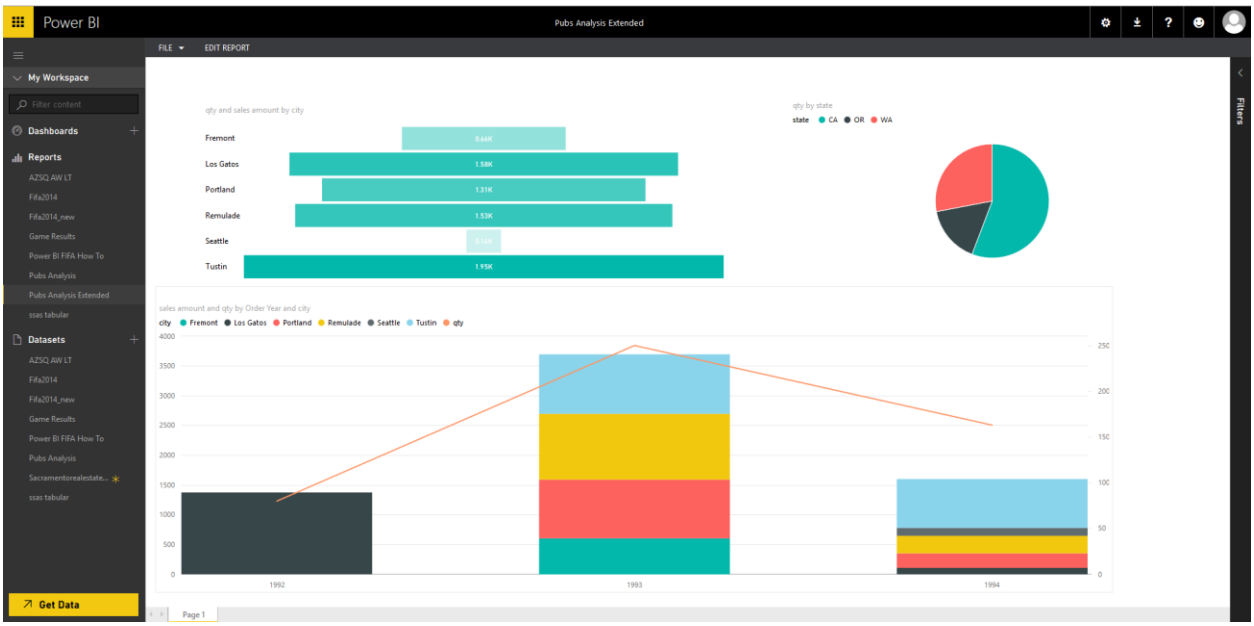
Let's create another visualization in the report. Add a Funnel to the report: Set Group to the city, Values to qty. Also set Color Saturation to sales amount. Color saturation will highlight the funnel bars based on the sum of sales amount.



For the last chart in this report, let's build a pie chart. I know that a pie chart is not a good chart for visualization, But I want to use it for three 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 a state (from stores), and Values as qty. below is the view of the pie chart

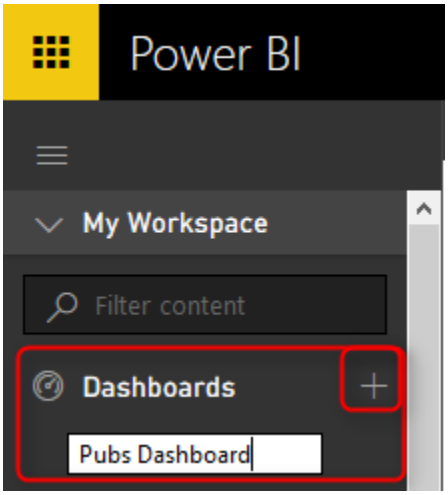


Now let's save the report with the name of Pubs Analysis Extended (save option is in the top left side of the report editor). You now see the new report under Reports. If you click on it, you'll see it in the viewer.

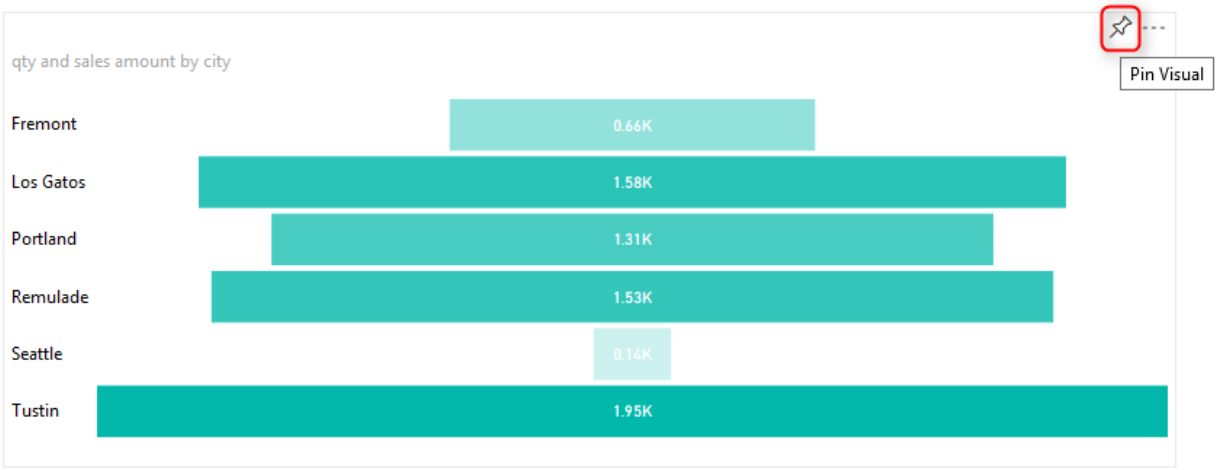


Build a Dashboard

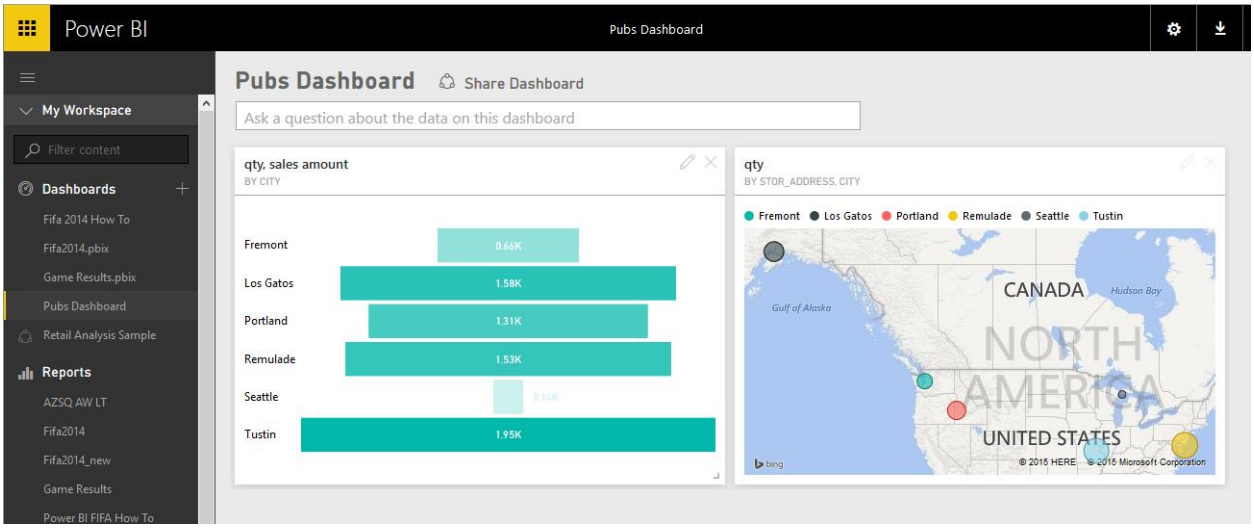
You can build dashboards from your existing reports. Dashboards will be primary viewpoint of you or users (from the website or mobile app). To create a dashboard, click on Add button close to Dashboards in the left pane, and create a new dashboard with the name of Pubs Dashboard.



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 the dashboard.



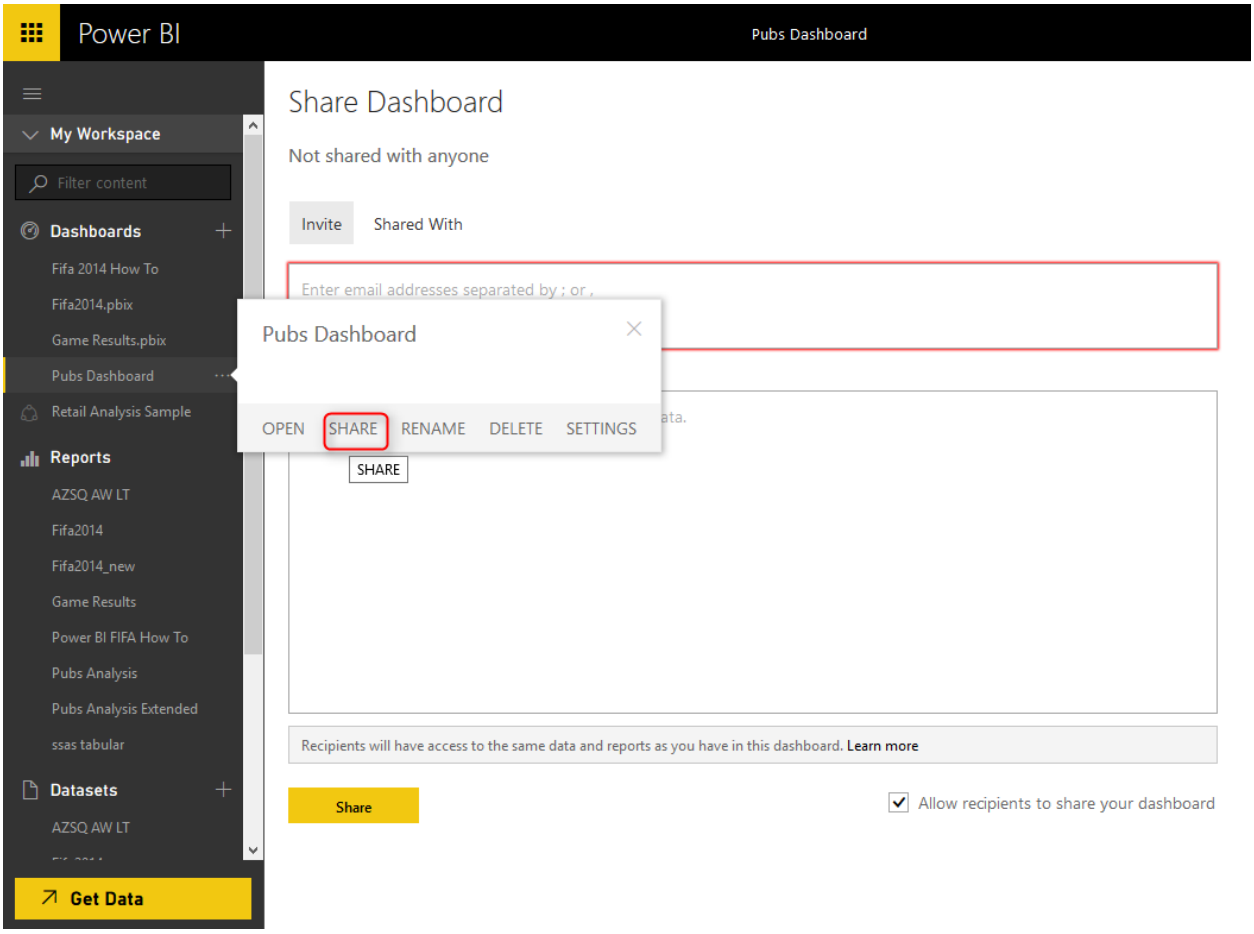
Go to Pubs Analysis report, and pin visual the Map visualization as well. Now the Pubs Dashboard looks like below screenshot;



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

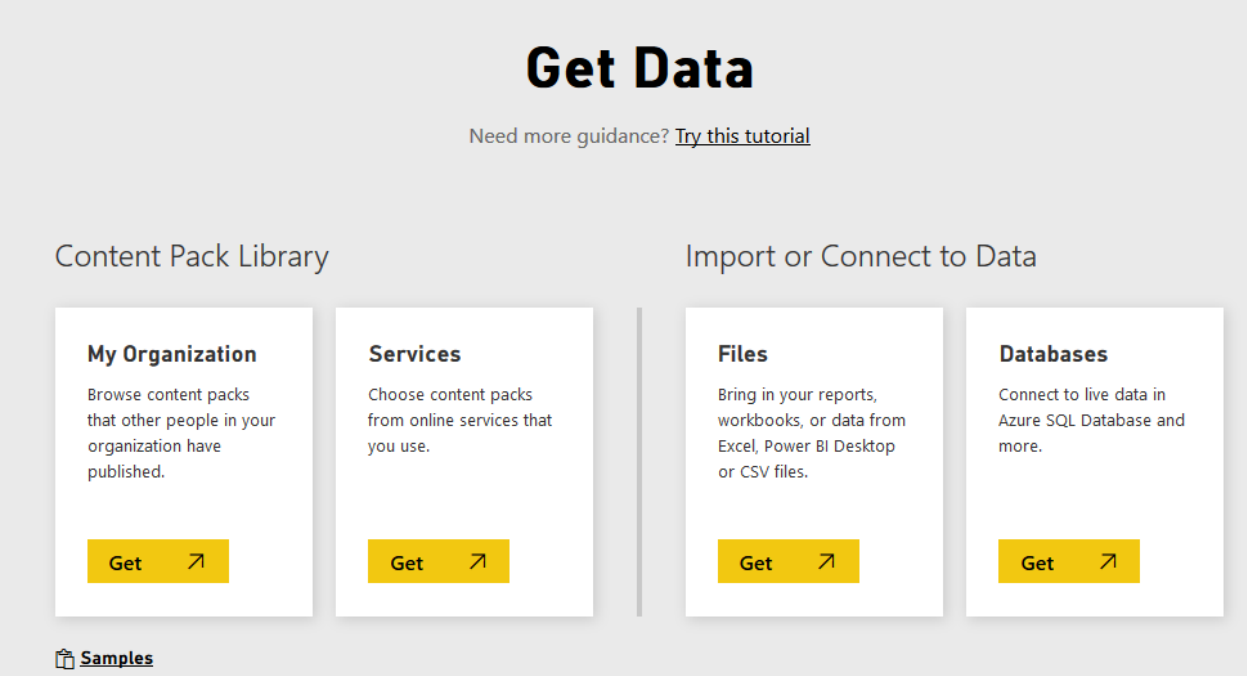
Sharing

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

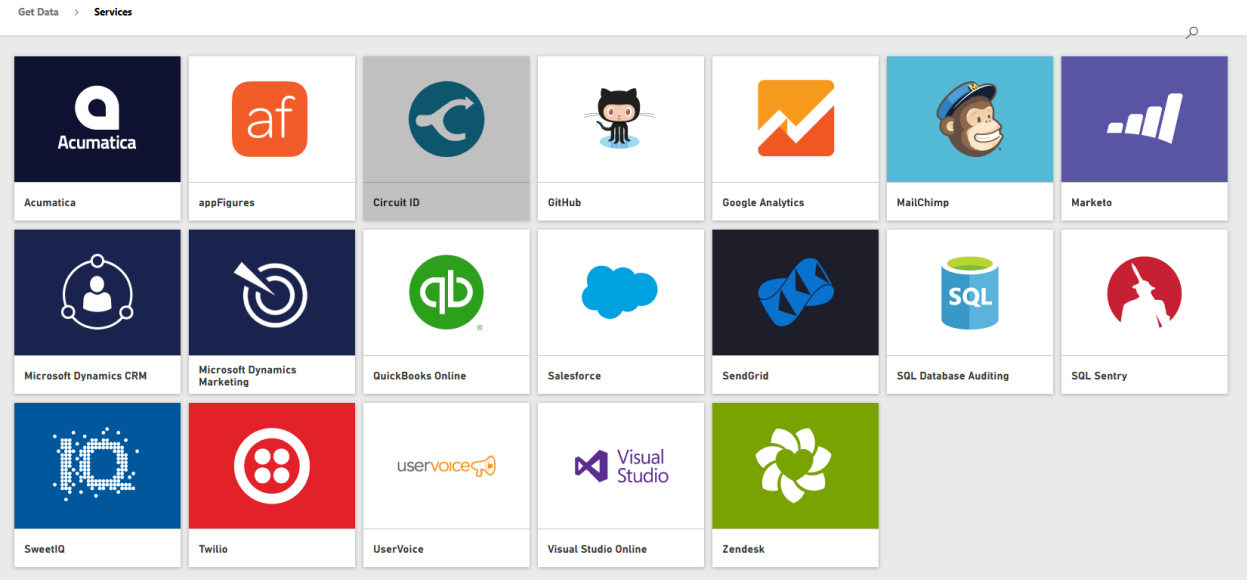


Get Data

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



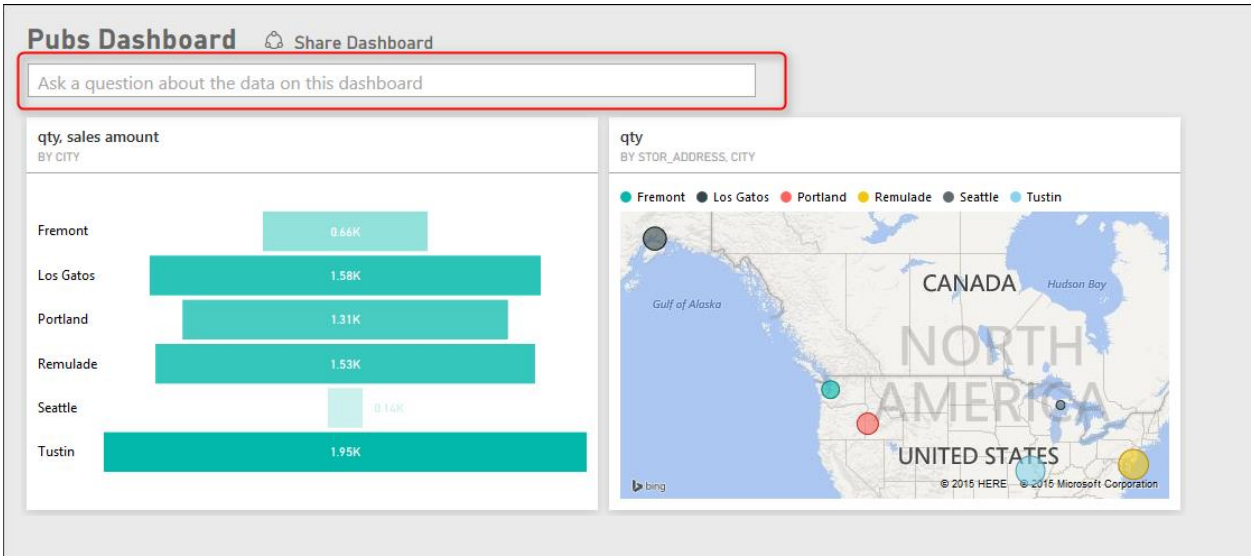
There are some data sources supported in online Get Data. You can get data from services (such as Google Analytics, MailChimp, CRM, Salesforce…), or get data from Files (such as Local File, OneDrive), or get from Databases (such as Azure SQL DB, Azure SQL DW…). Some of these options require Power BI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



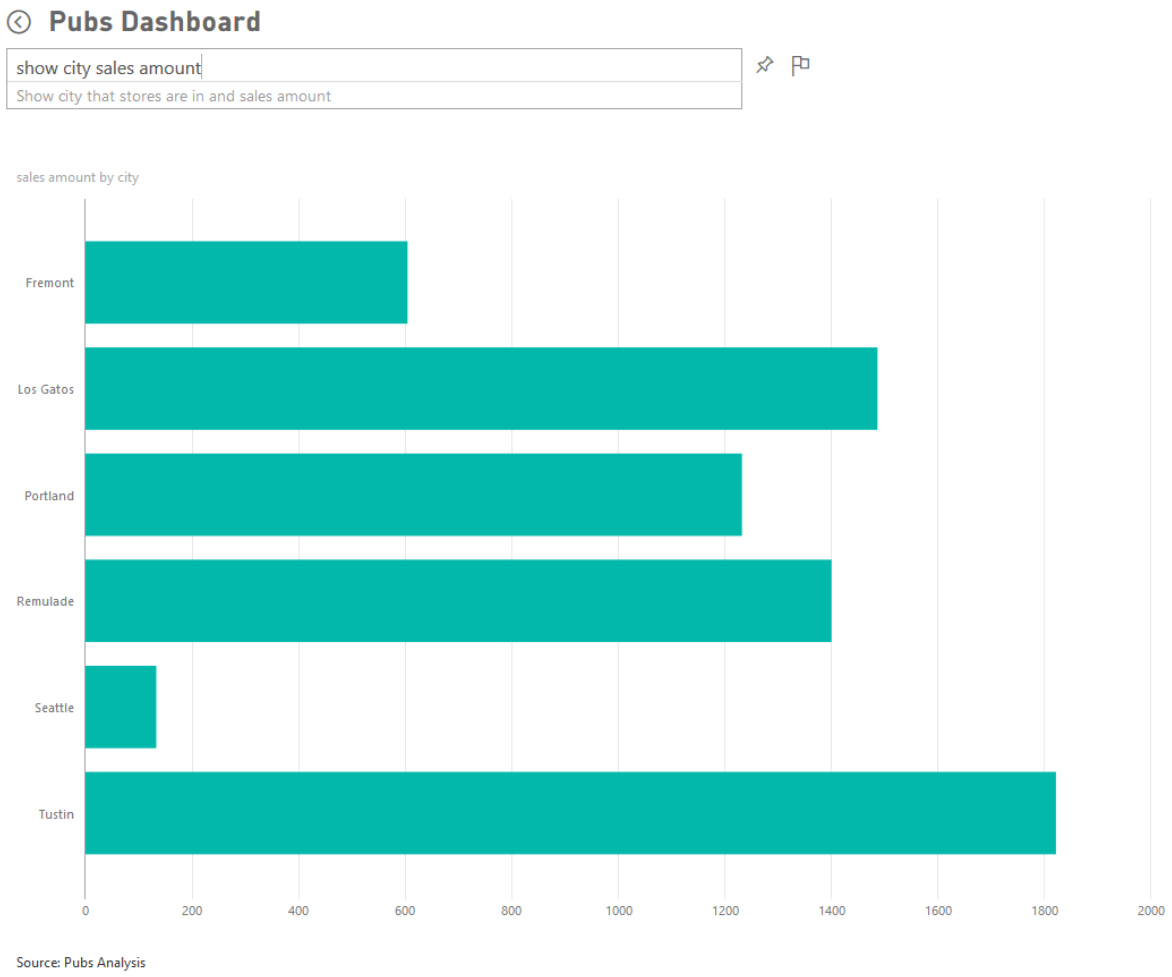
Power Q&A

Power Q&A is the question and answering engine of Power BI which works based on natural speaking language. You can ask questions in the 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.

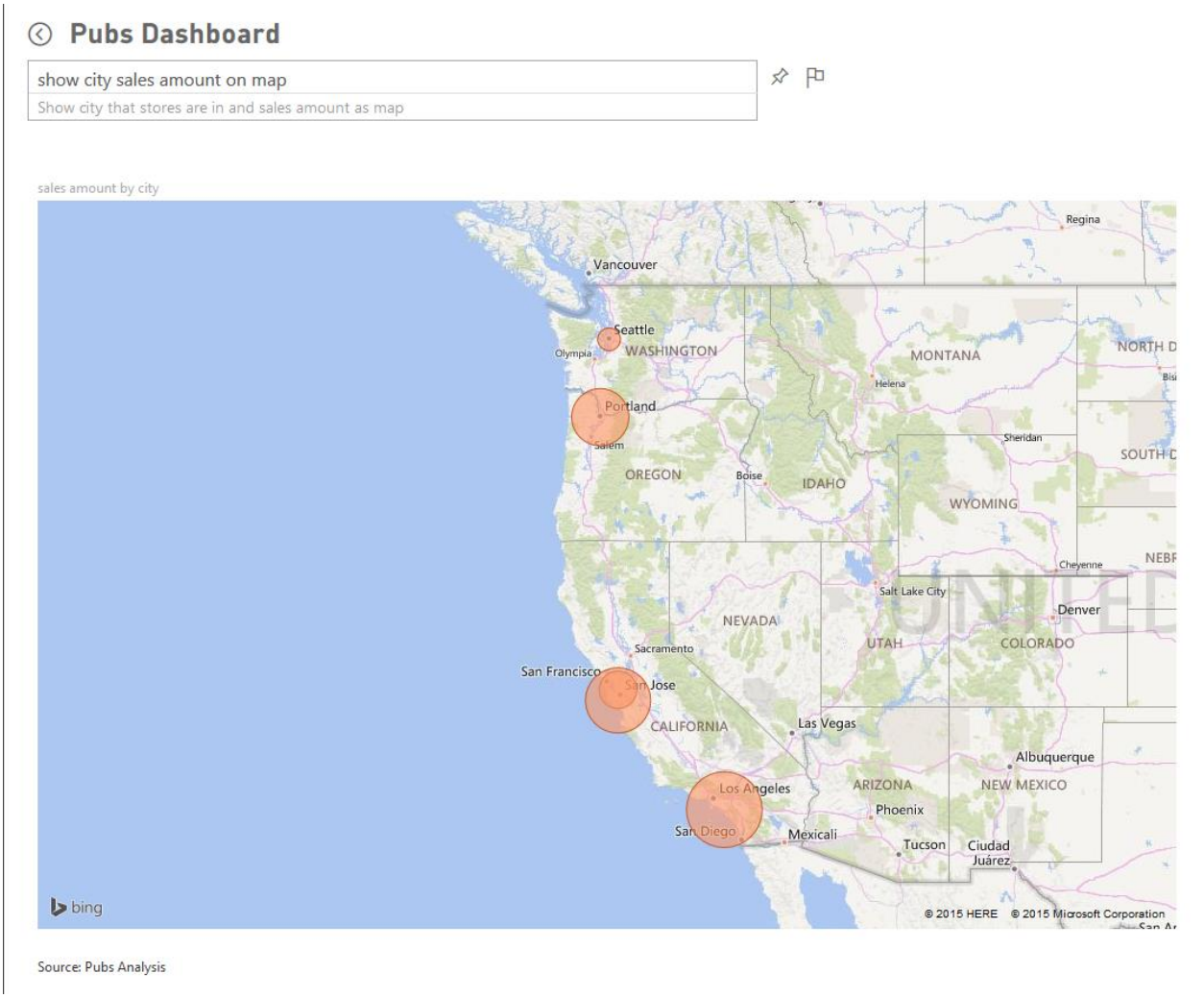
To ask a question simply use the question bar in the dashboard



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



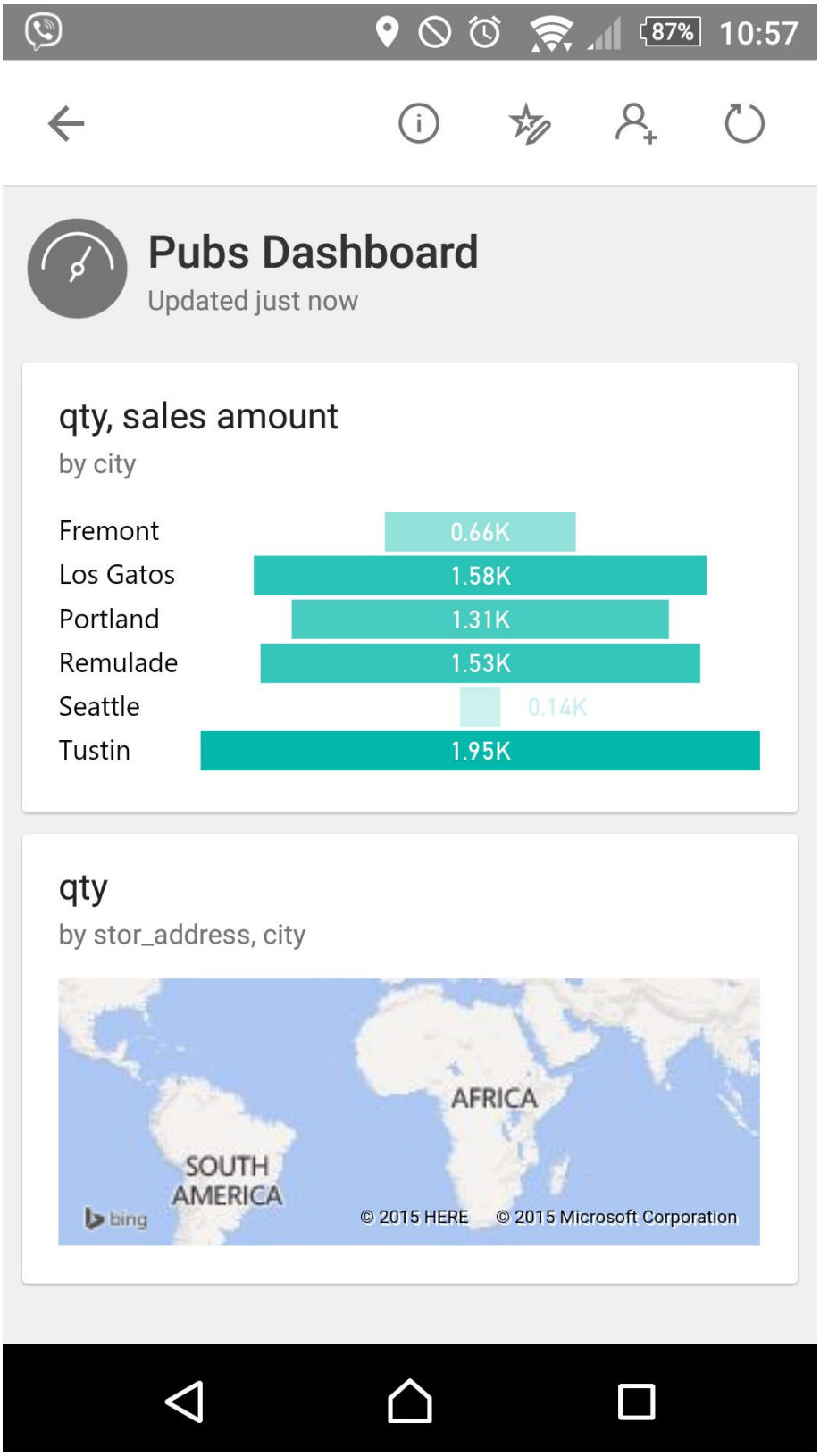
you can even change the visualization to map with this question: “Show city sales amount on map”



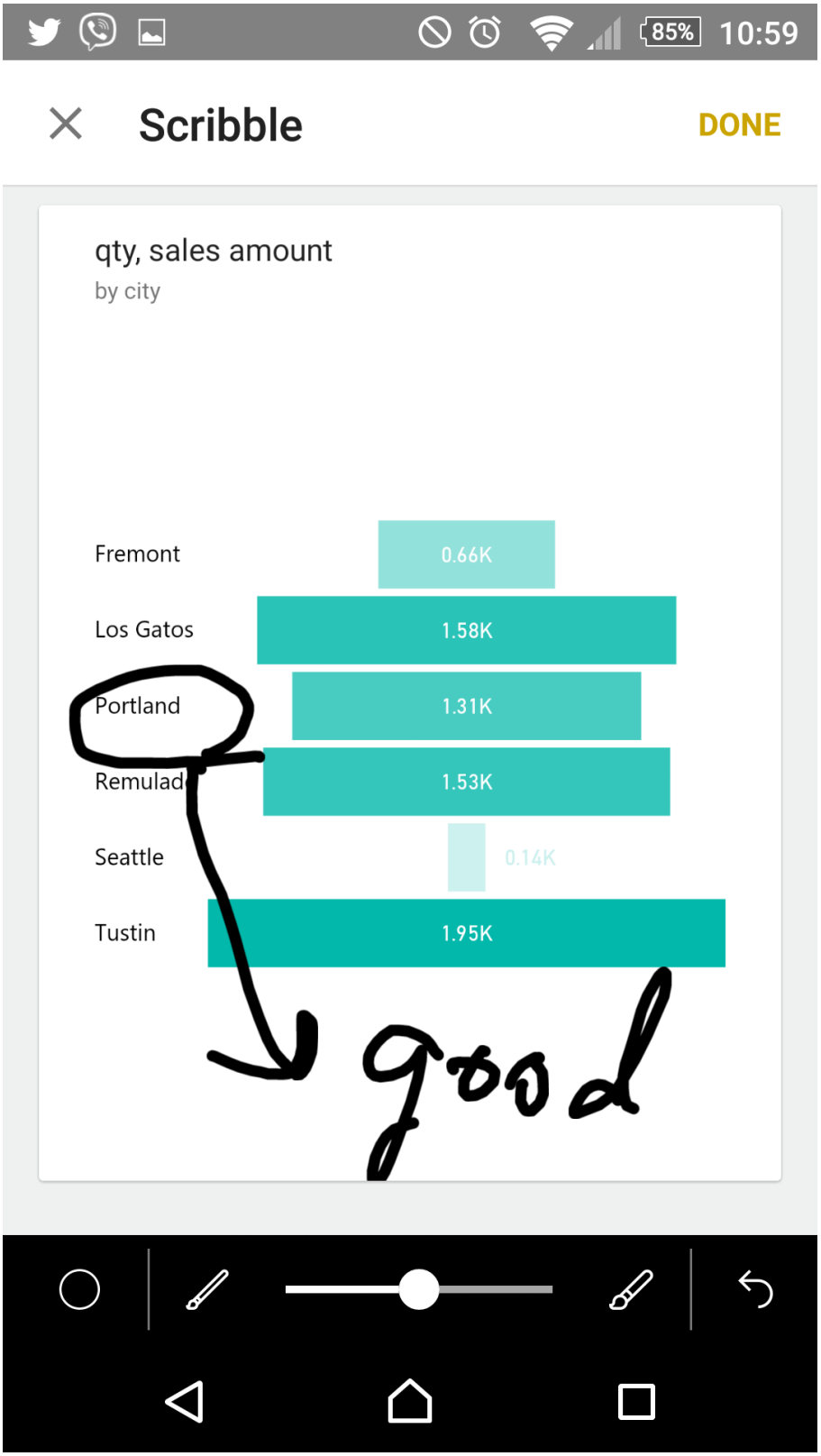
Power Q&A is a very powerful engine. There are some tips and tricks of how to develop your data model so Power Q&A could generate best responses to questions. I'll cover these tips and tricks in future chapters of this book.

Power BI Mobile App

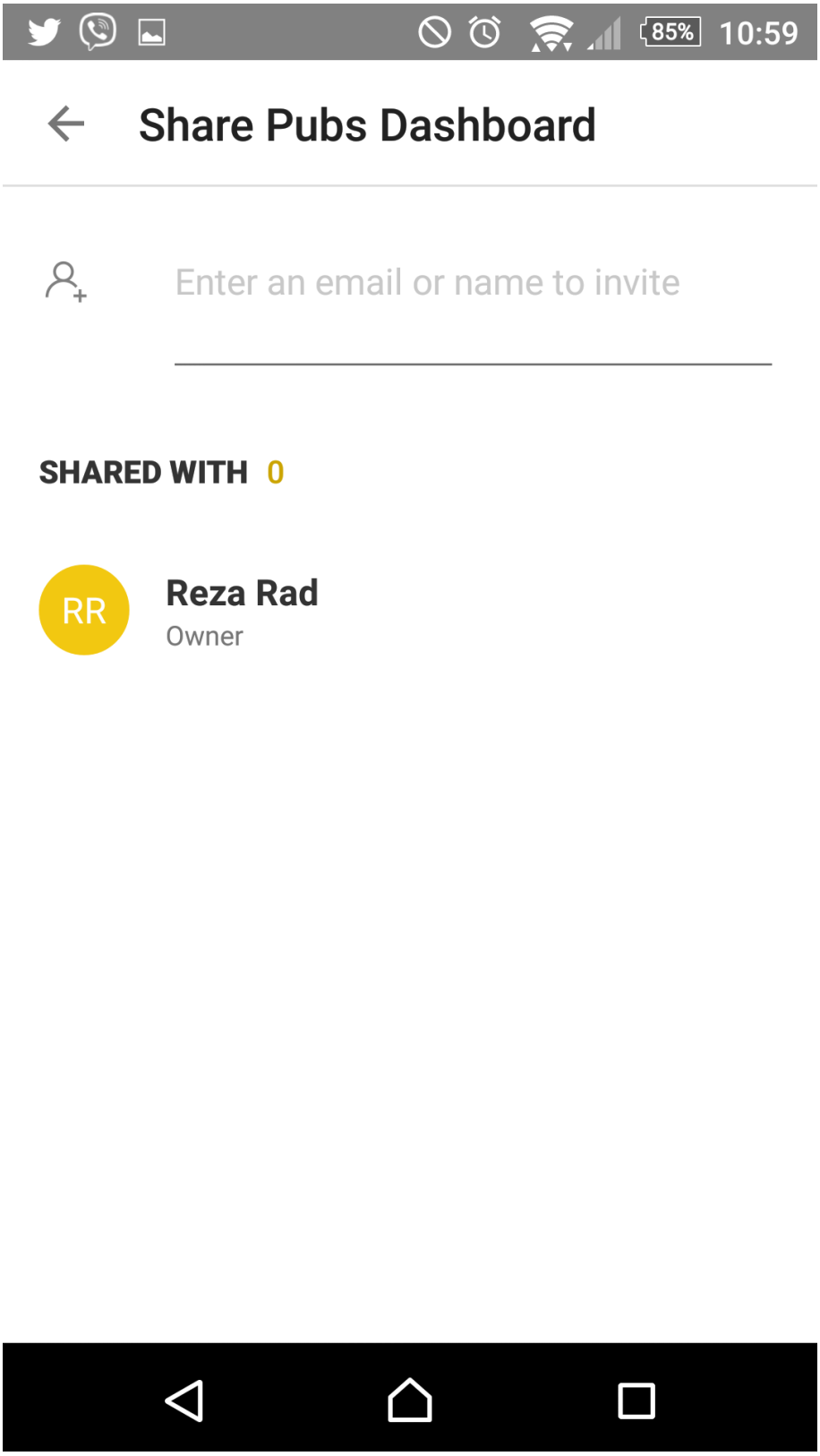
Power BI app is available for Android, Apple, and Windows Phone. simply download it from Google Play (Android), or AppStore (Apple), or Windows Store (Windows Phone). After the installation login with your username, and you'll see dashboards there. Screenshot below shows the dashboard in the Android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



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



You can even share it from Mobile App


85%
10:59
Share Pubs Dashboard
Enter an email or name to invite
SHARED WITH 0
RR
Reza Rad
Owner

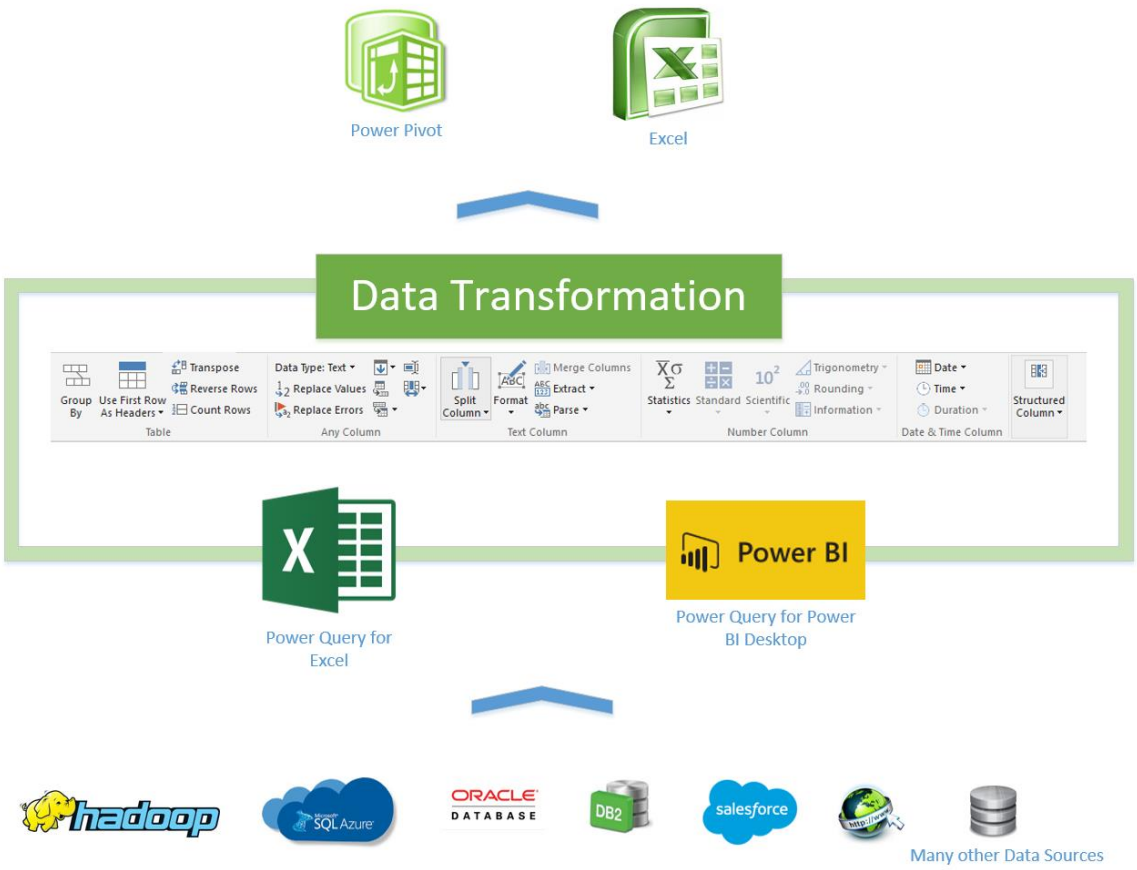
Summary

In summary, you've learned about options that are available to you from the 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 the 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 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 also has a 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 is Power Query?
- 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 a public preview for the first time in 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.

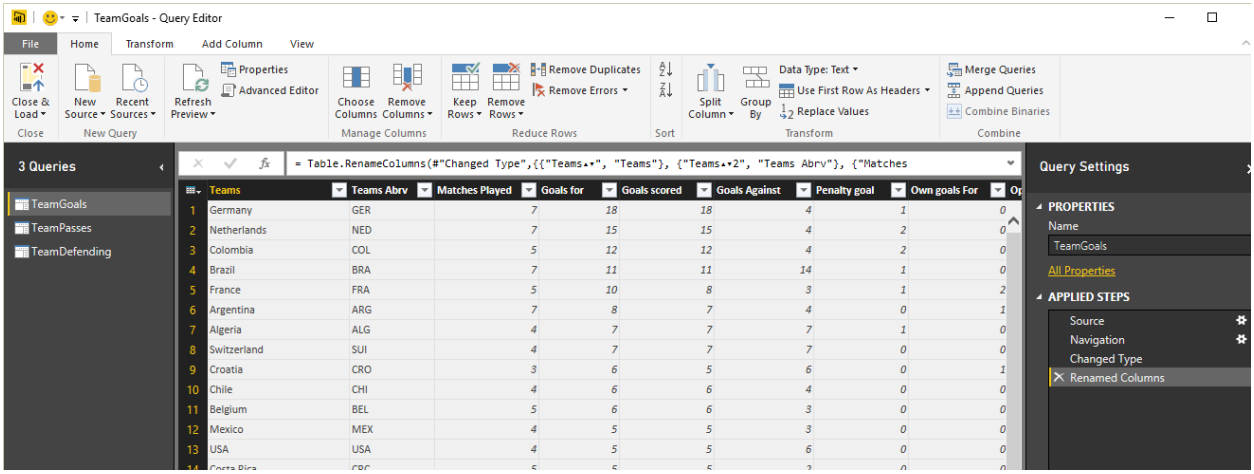
Power Query is a data extraction and transformation engine. The engine comes with a formula language and a graphical tool. The graphical tool has two major setup versions; one embedded in Power BI Desktop tool and the other one as an Add-In for Excel. The graphical tool has a list of transformations 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. 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 connect to a 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. A 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.

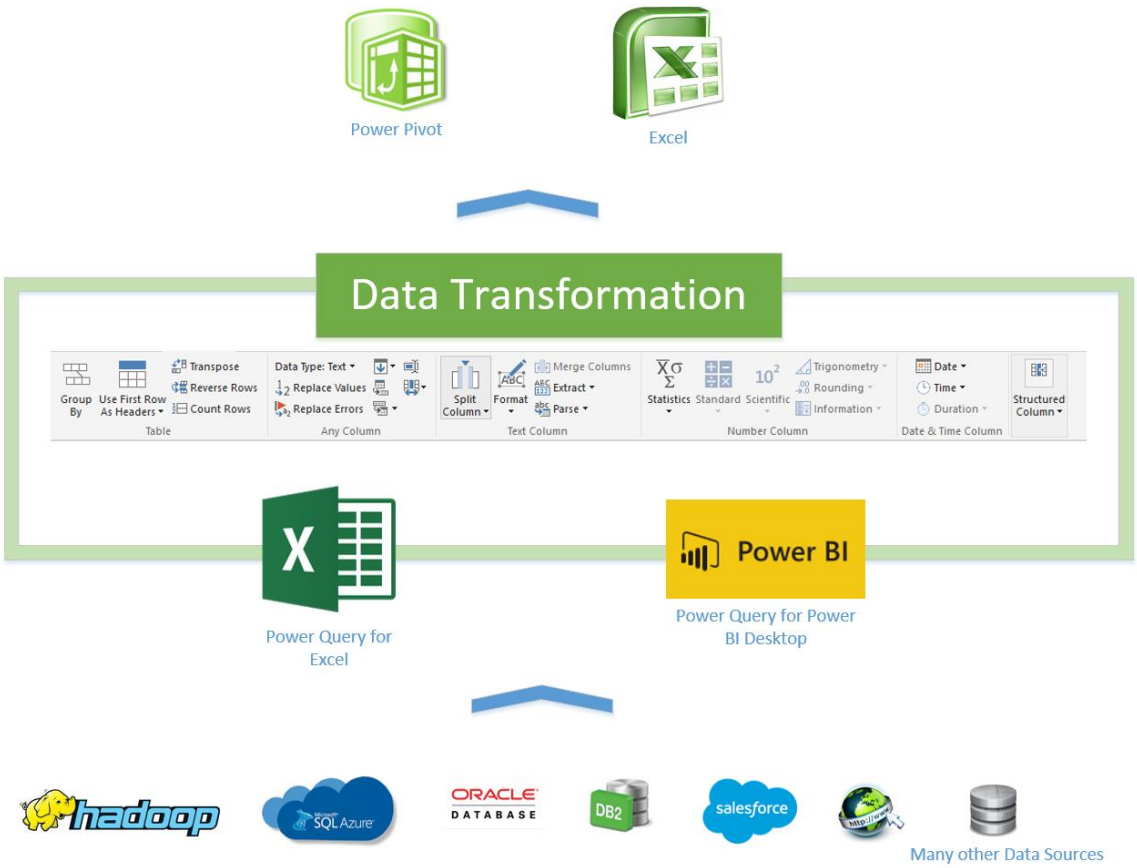
Power Query graphical interface is so easy to work with that even business analyst, or a power user can work with it, on the other hand, Power Query M language is so powerful that can be used for complex real-world challenges of data transformations. Power Query can load the result set into an Excel spreadsheet, or it can load it into Power Pivot for data modeling. The version of Power Query used in Power BI Desktop loads 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 a screenshot or the Query Editor window



	Teams	Teams Abrv	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

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 a 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 the preview stage, so the link is likely to change.

Power BI Desktop:

https://powerbi.microsoft.com/desktop

What Can You Do With Power Query?

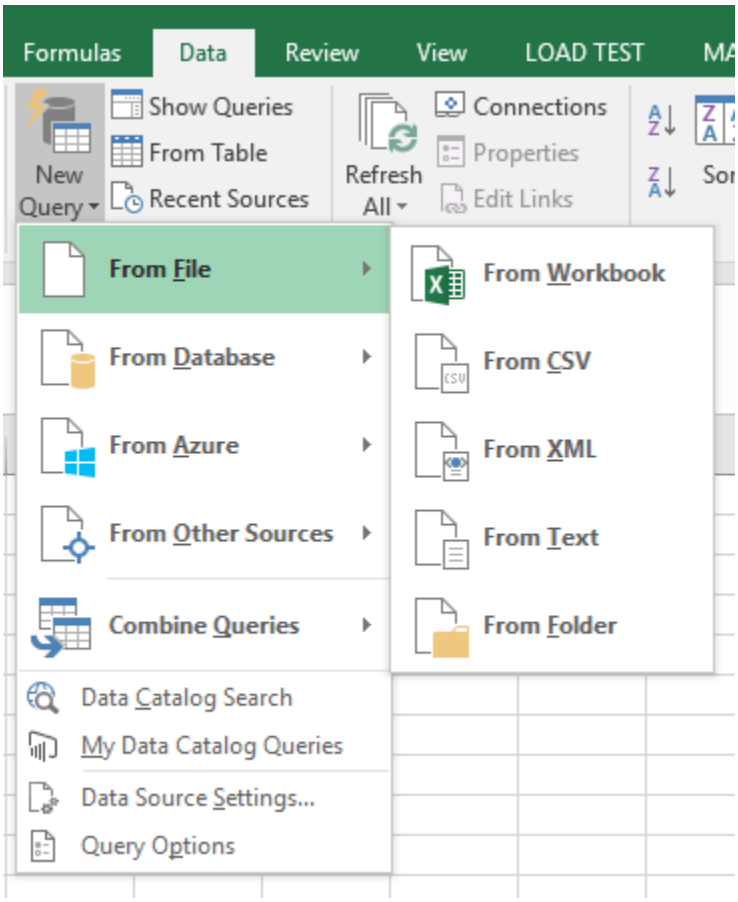
Get Data From Wide Range of Sources

With Power Query, you can connect to a wide range of data sources. SQL Server or DB2 or Oracle…. All of this 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 the list of files in a folder! You can connect to a range of applications such as Facebook, Salesforce, CRM Online, etc and get data from them. You can get data from Azure services such as Azure SQL

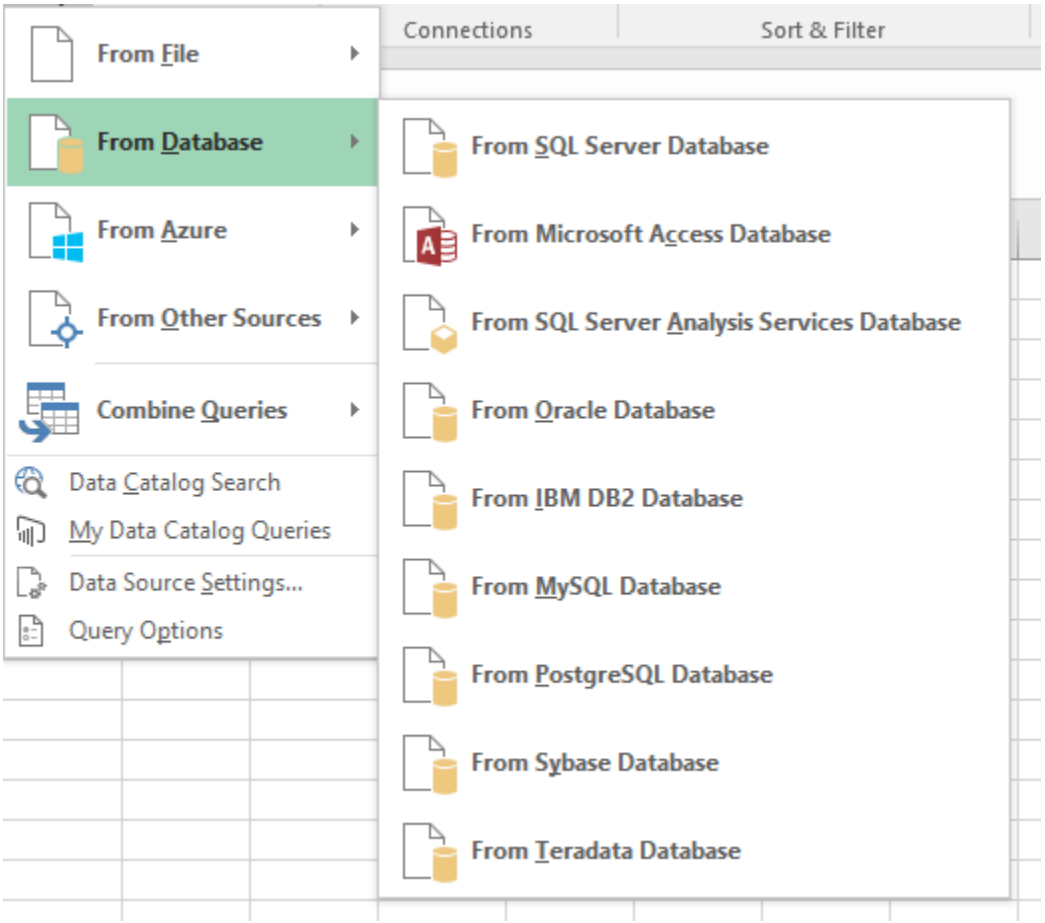
database, Azure HD Insight, Azure Blob storage, etc. There are many data sources supported for Power Query (and obviously for Power BI). Also more data sources will be available in every update of Power Query or Power BI.

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

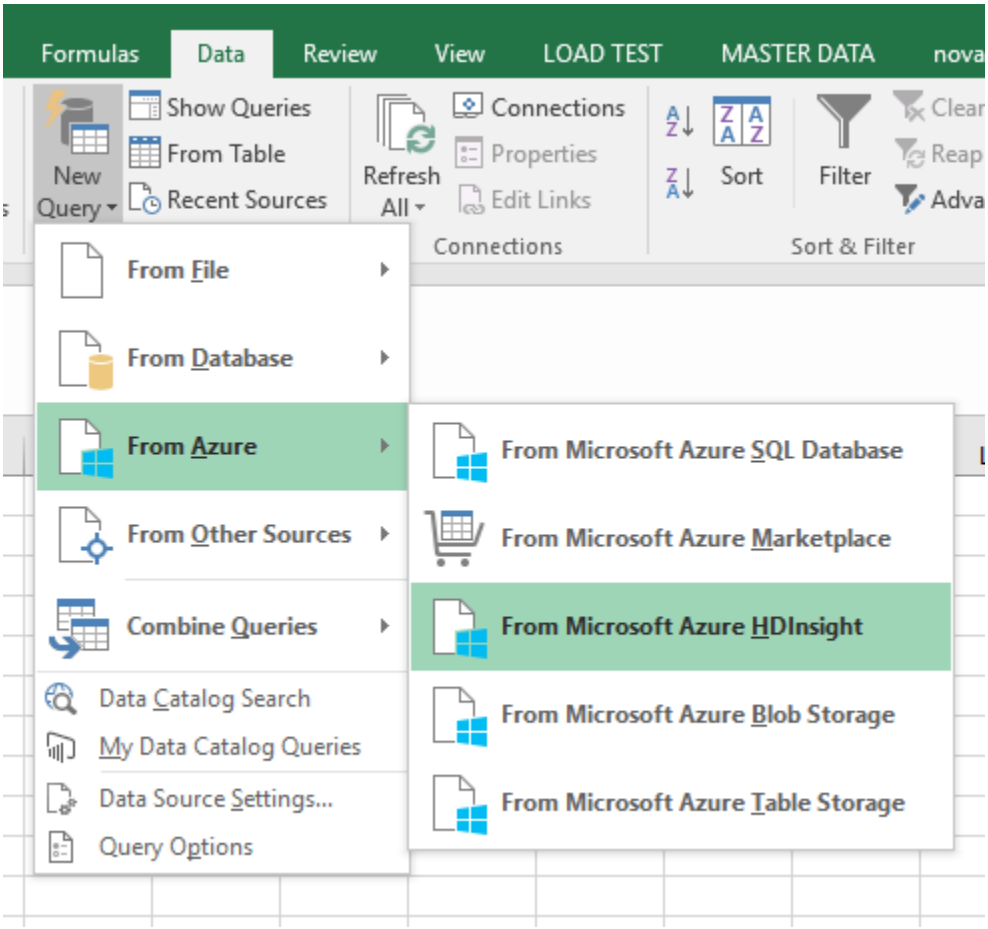
File Data Sources



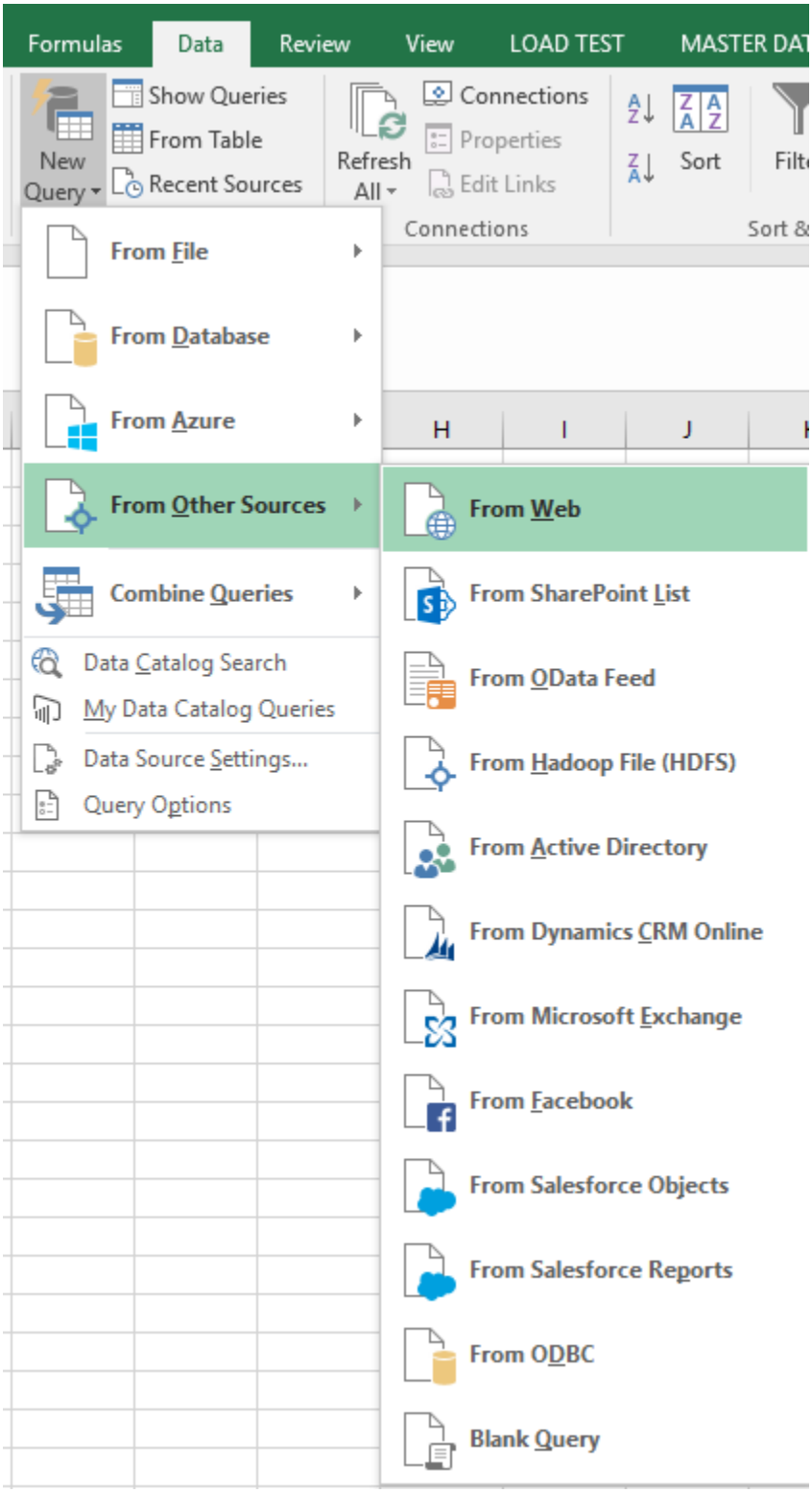
Databases


Connections
Sort & Filter
From File
From Database
From Azure
From Other Sources
Combine Queries
Data Catalog Search
My Data Catalog Queries
Data Source Settings...
Query Options
From SQL Server Database
From Microsoft Access Database
From SQL Server Analysis Services Database
From Oracle Database
From IBM DB2 Database
From MySQL Database
From PostgreSQL Database
From Sybase Database
From Teradata Database

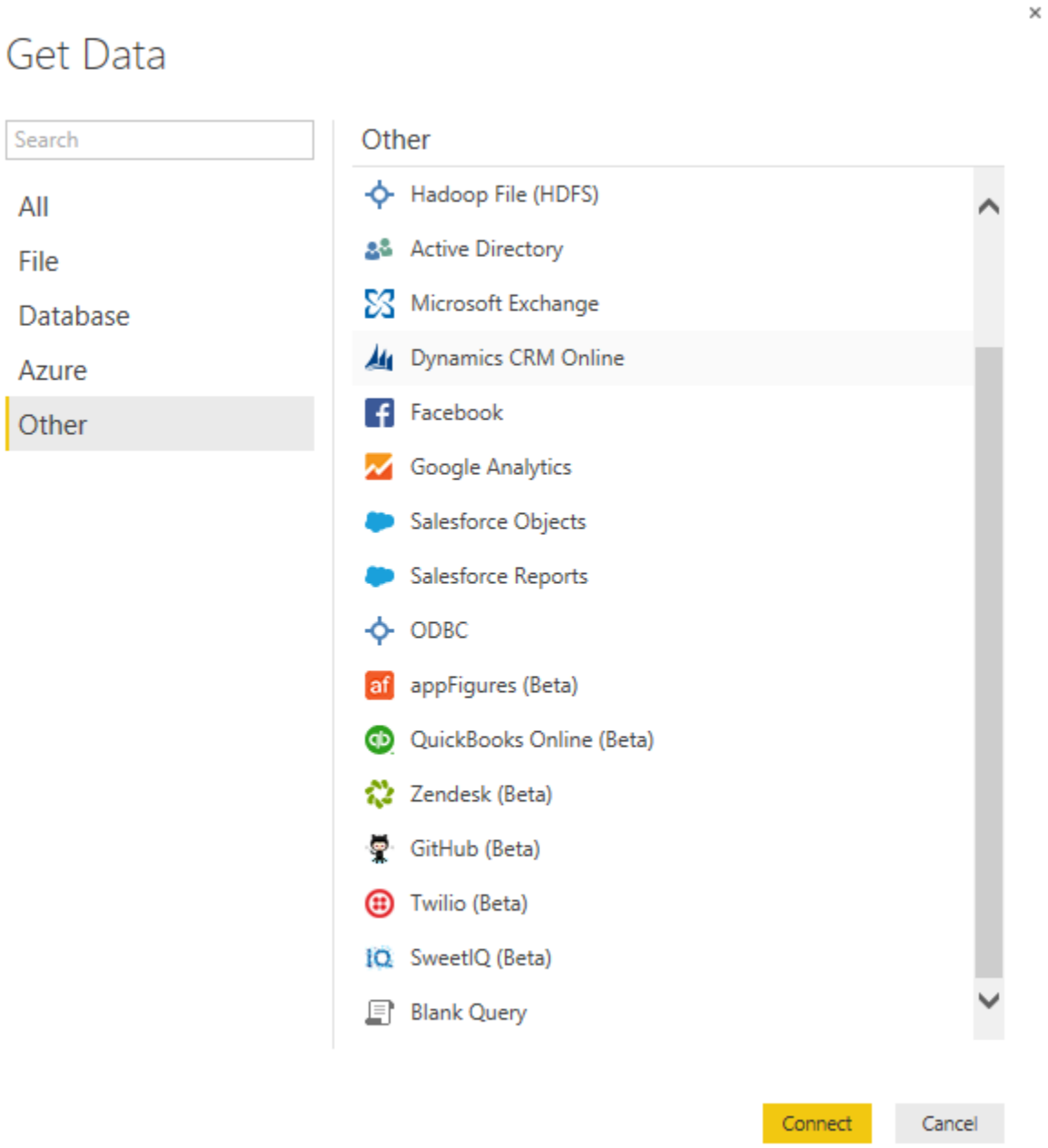
Azure



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:



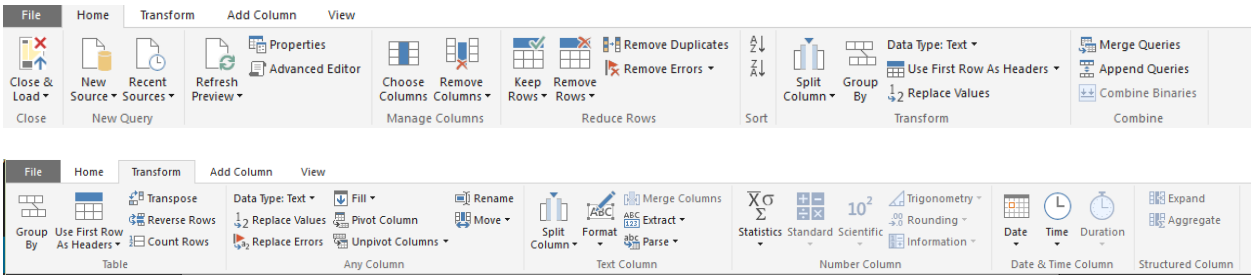
Apply Transformation In a Development Editor

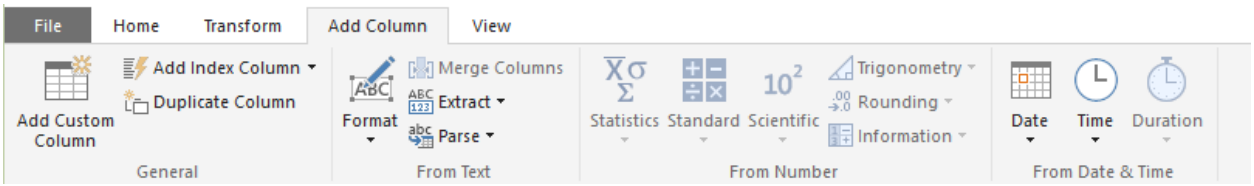
Power Query look at the data values with data types such as: Table, Record, List, DateTime, Text, Number, Boolean, etc. There are many data transformation functions for any of these data types. You can apply Merge (similar to join) or Append (similar to UNION) to two tables. You can apply text functions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

DateTime functions such as functions for the 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 the very easy user interface. You can apply most of the transformation with the 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:





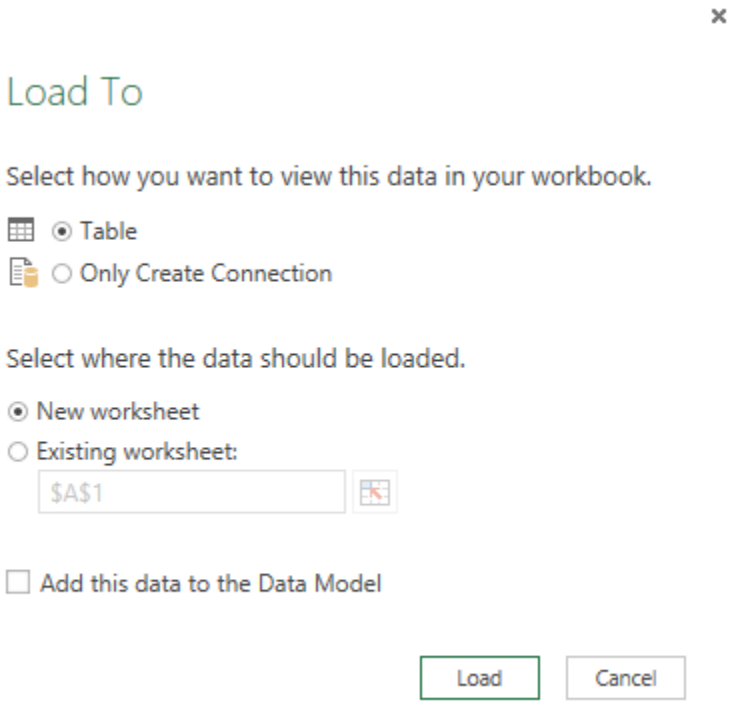
M is the formula language behind the scenes of Power Query. Everything you do in the Query Editor will be translated to an M script. M contains a full list of functions that you can use. So the powerful side of Power Query is M. I will go deep into details of M in this book because you would need it 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 the 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 a 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 the 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 the usage of it in scenarios and challenges that you might have right now! Good start, in next sections I will go through the experience of getting data with Power Query and Power BI Desktop.

Get Started with Power Query: Movies Data Mash-Up

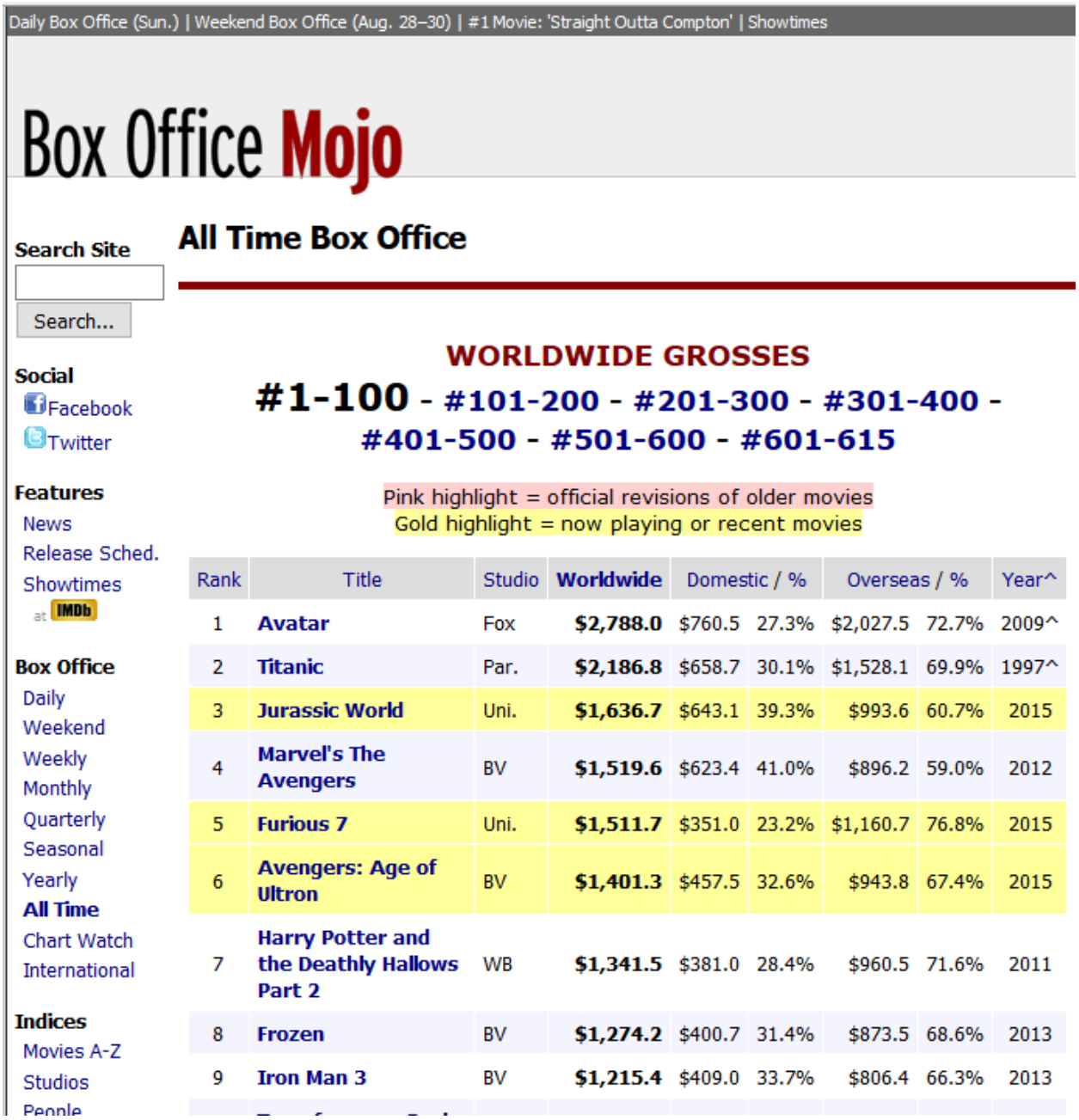
Published Date: September 1, 2015

As another section of the Power BI online book: from Rookie to Rockstar, I would like to get started 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 the movie's data to be mashed up. I used this example because the 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 Mashup 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 is 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

Search...

Social
Facebook
Twitter

Features
News
Release Sched.
Showtimes
at IMDb

Box Office
Daily
Weekend
Weekly
Monthly
Quarterly
Seasonal
Yearly
All Time
Chart Watch
International

Indices
Movies A-Z
Studios
People

All Time Box Office

WORLDWIDE GROSSES

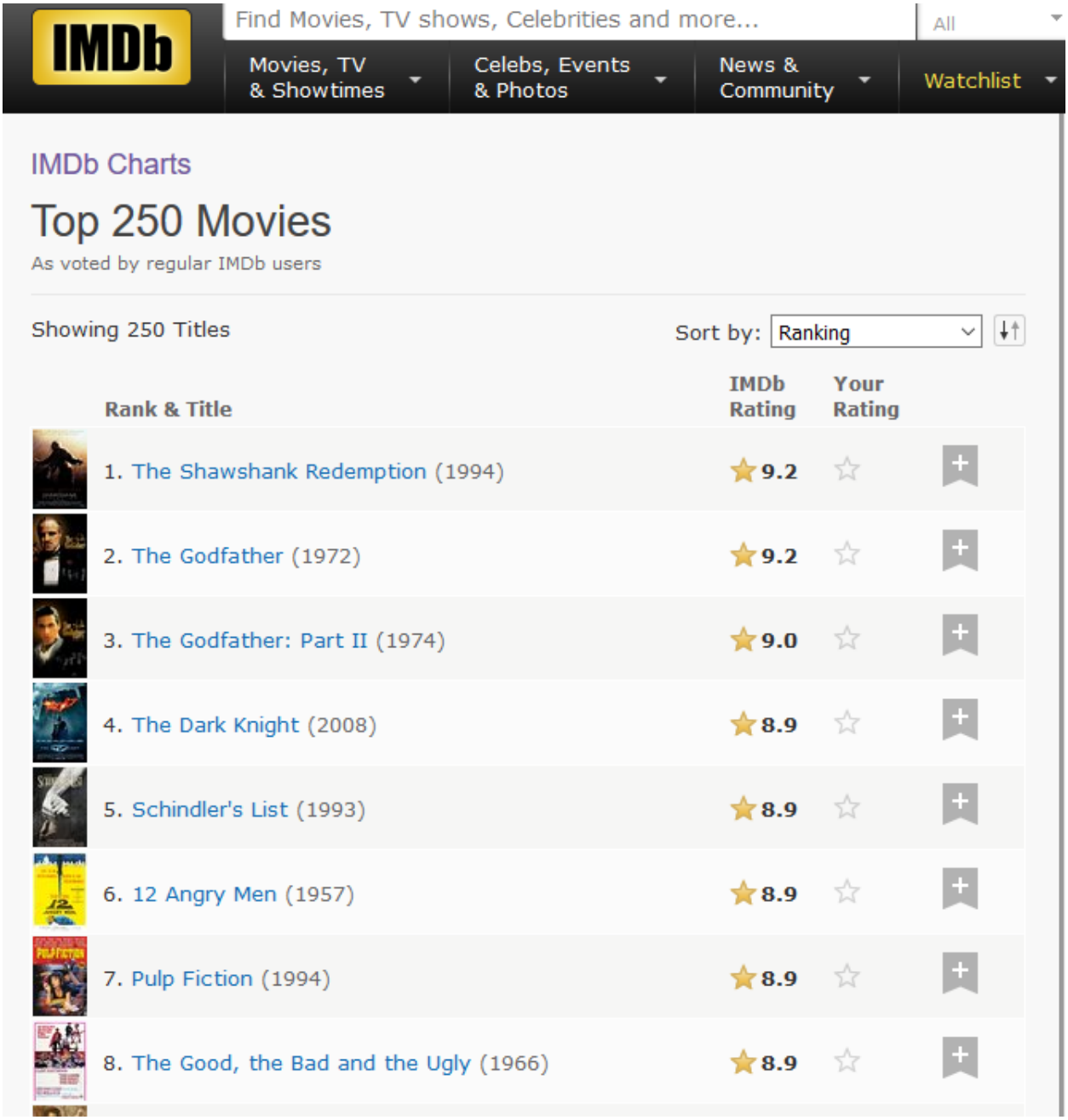
#1-100 - #101-200 - #201-300 - #301-400 - #401-500 - #501-600 - #601-615

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

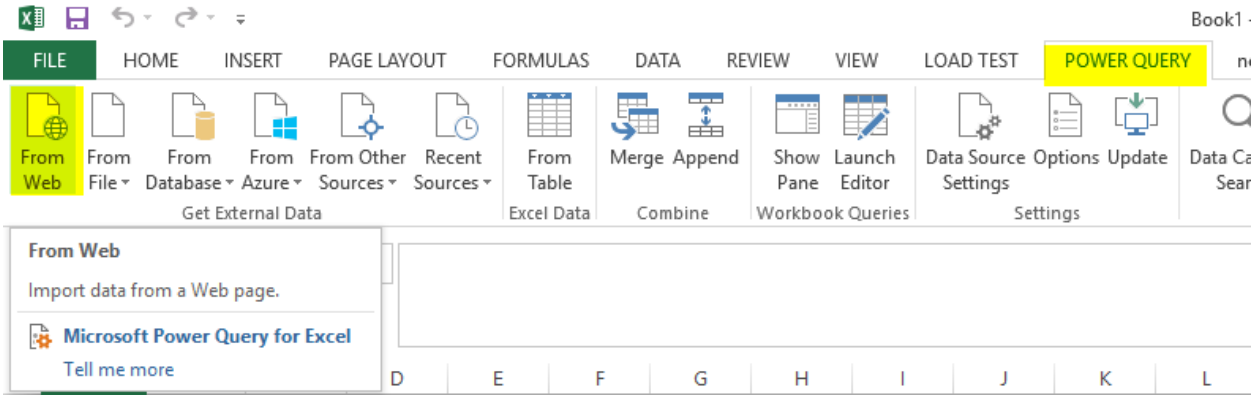
1. Top 250 movies ranked by people in IMDB website

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:

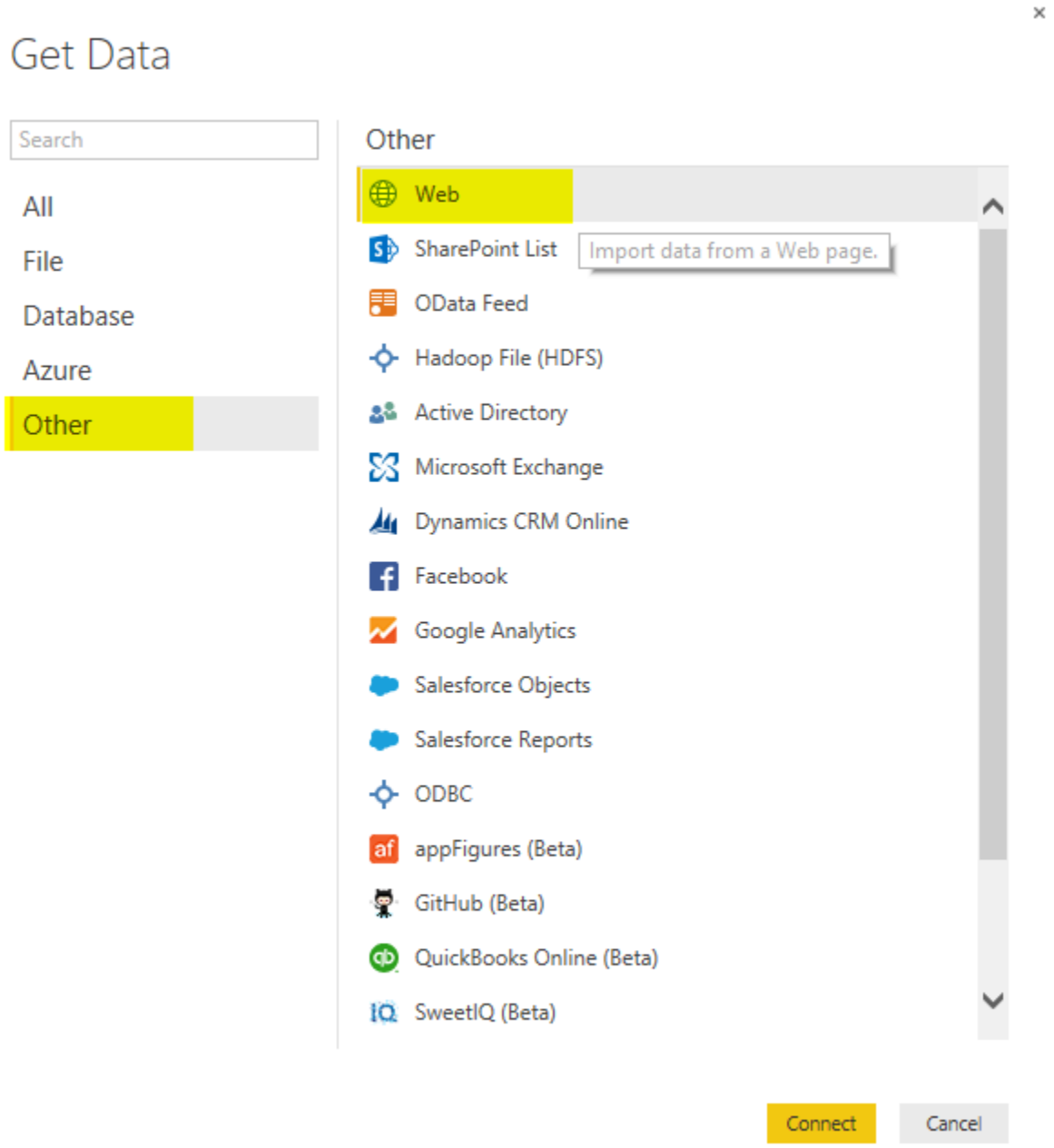


Let's Get Started

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



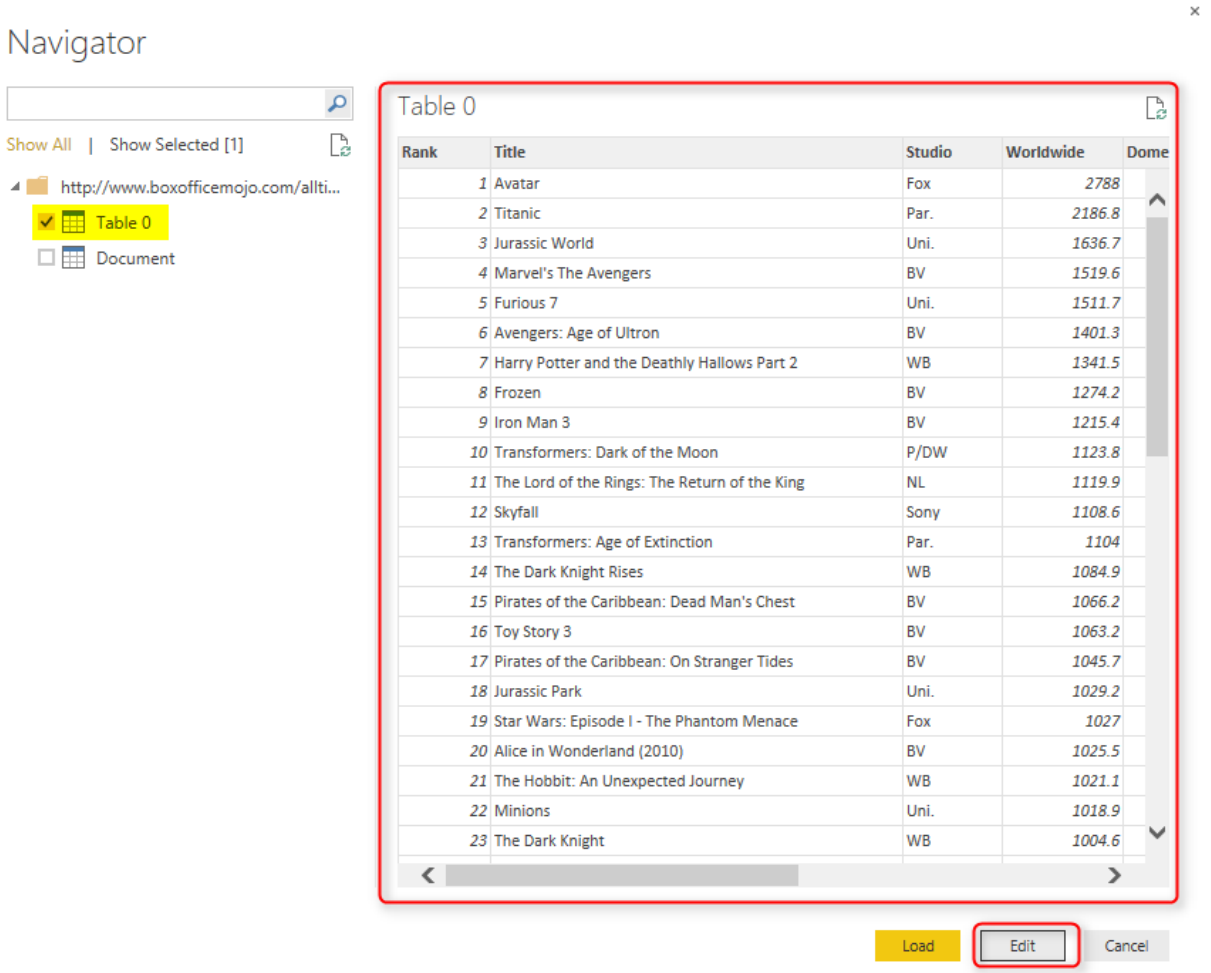
Or Open Power BI Desktop and Get Data from Web



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

Click OK, after 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;

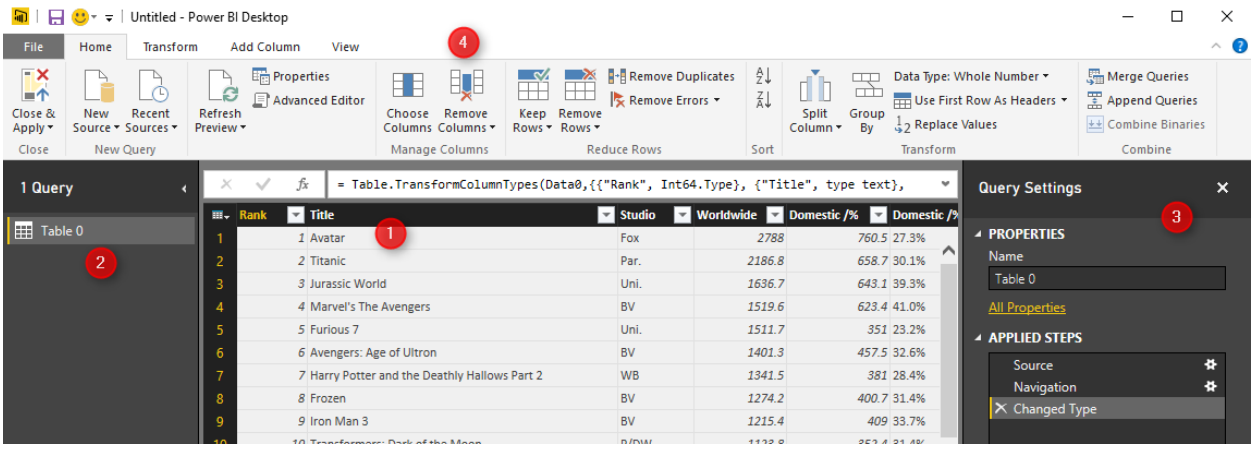
Click on Table 0. You will see a preview of data in the table in the main pane. Now tick the checkbox for Table 0 and click on Edit button in Navigator



Rank	Title	Studio	Worldwide	Dome
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	

Query Editor

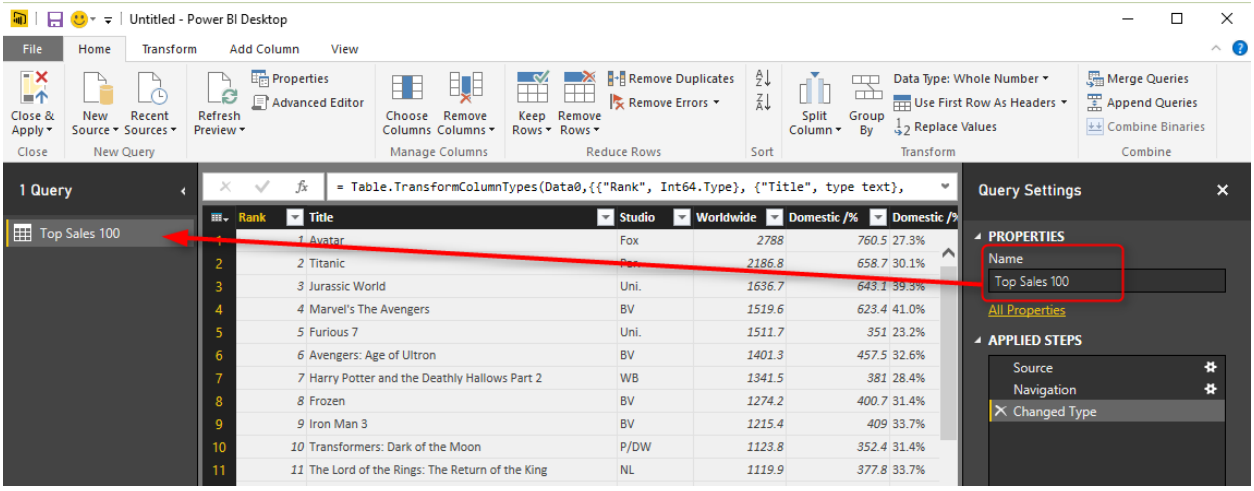
After clicking on Edit, you will see the Query Editor window opened. This is an editor that you will spend most of your time on data mash-up here.



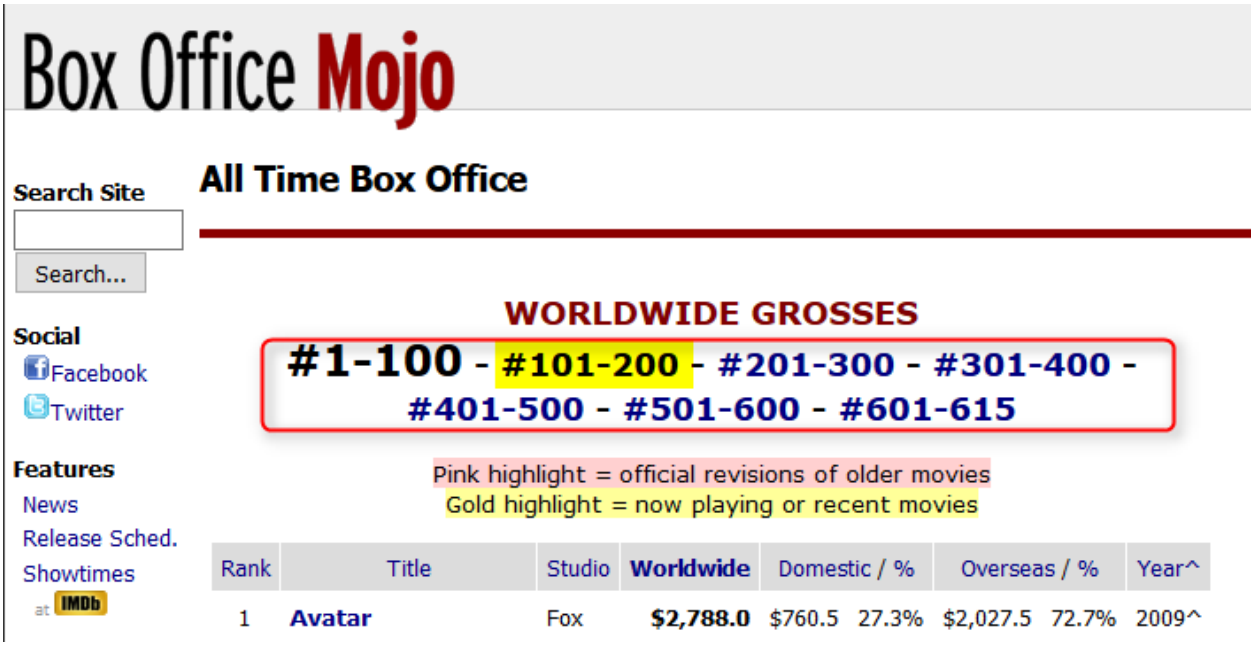
Query Editor has four main sections (numbers matched to screenshot above);

1. Main dataset pane; This is the central area that the result set will be displayed as a preview with a limited number of rows
2. List of Queries; Left-hand side pane will show a 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, a 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 the top section

Rename the existing query to Top Sales 100

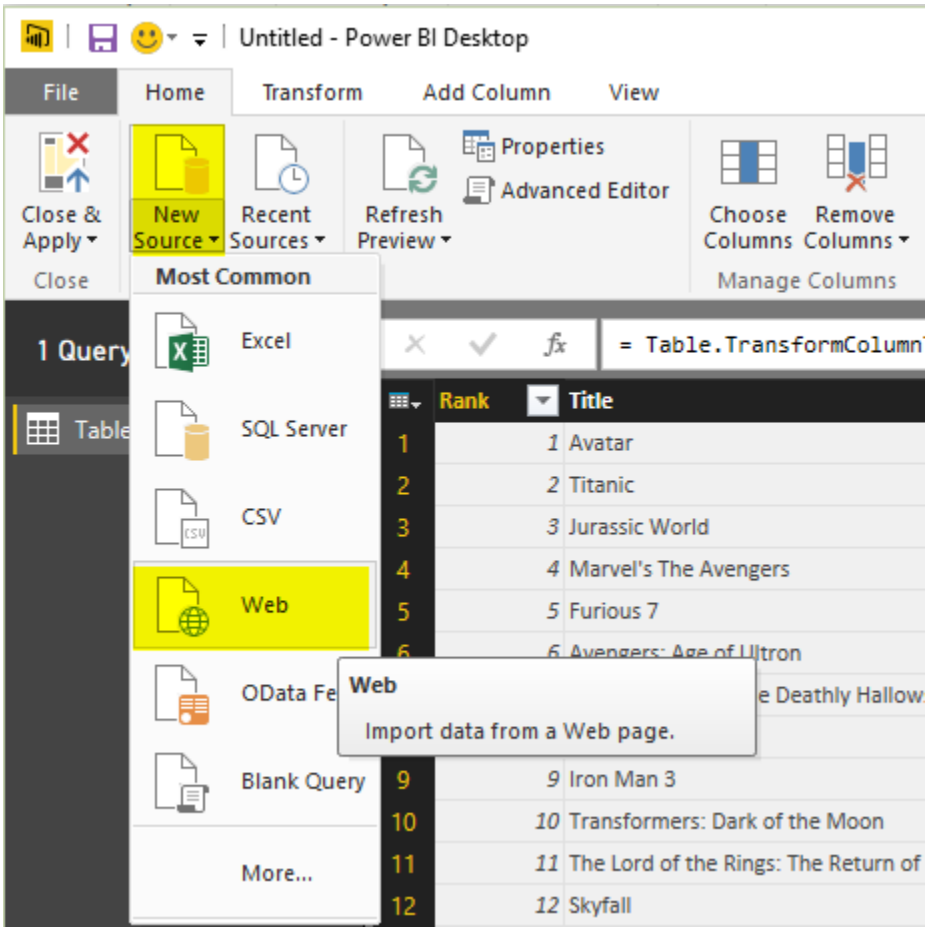


Our goal in this example is to join the data set of global gross sales with the IMDB user rating, and then analyze to see what are best sellers in movie titles among the best-rated movies or not? So the more data in gross sales we get would give us better analysis. The above URL only gives us top 100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

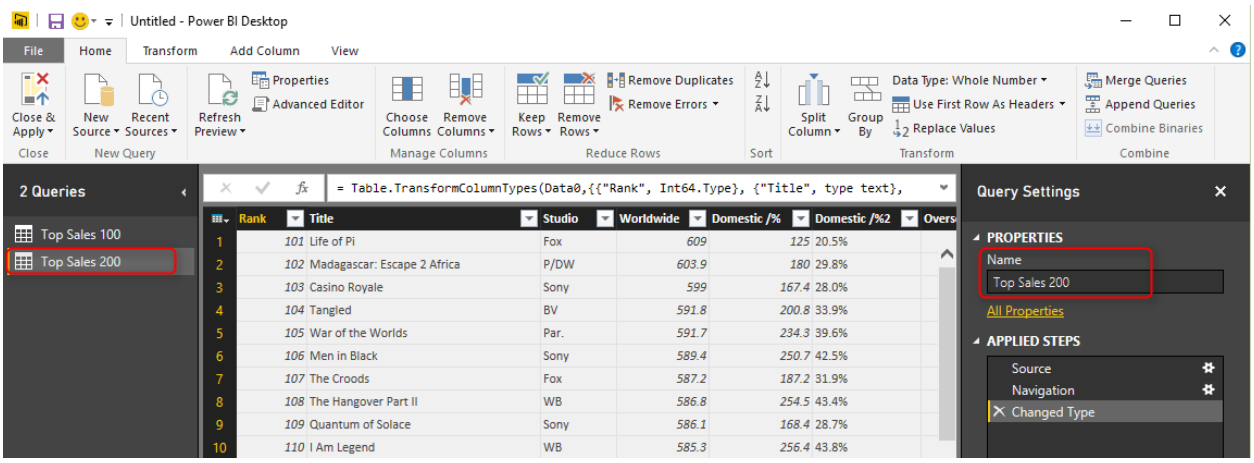


So Let's add the list of movies from 101 to 200 in best sellers;

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



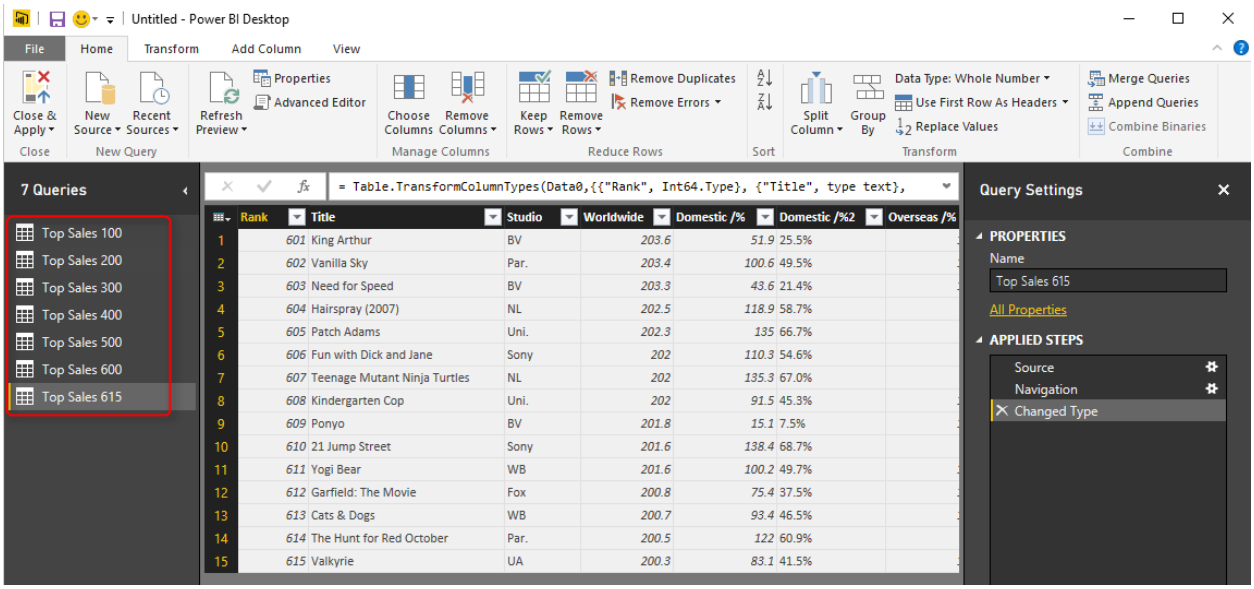
Follow this process for links below;

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

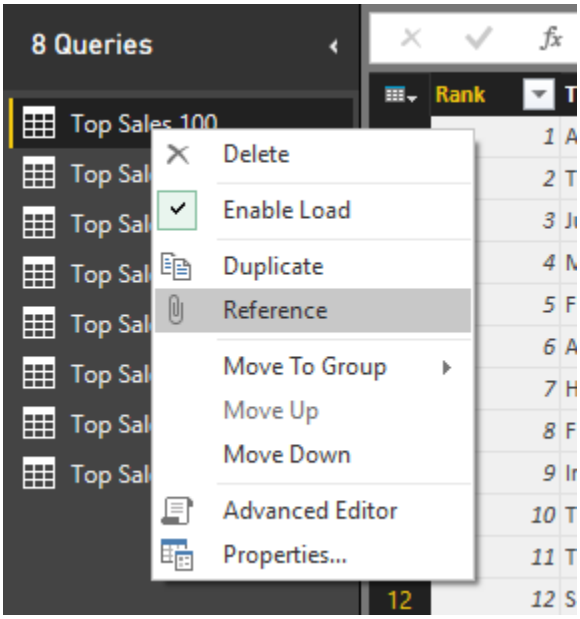


As you see in query editor, all of these queries are separate from each other. Let's combine them all together. In database and SQL world that can be done with UNION. Here in Power Query we can do Append Queries;

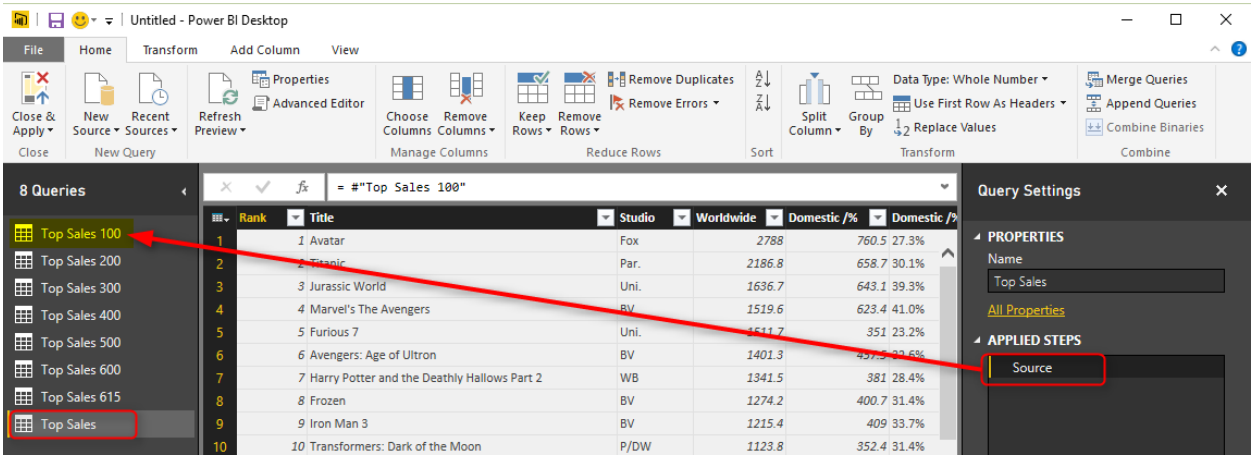
Use a Query as a Reference

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



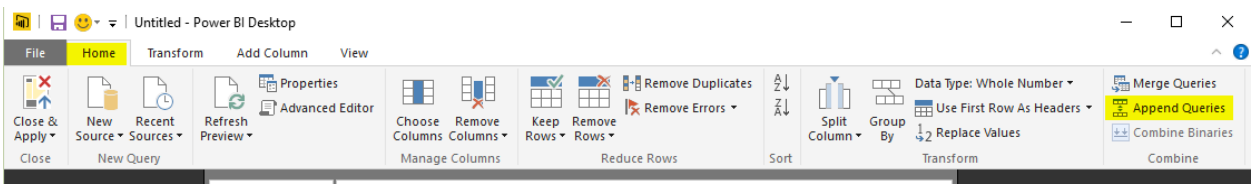
This will create a new query that users Top Sales 100 as the source (or reference). Rename this new query to be just "Top Sales."



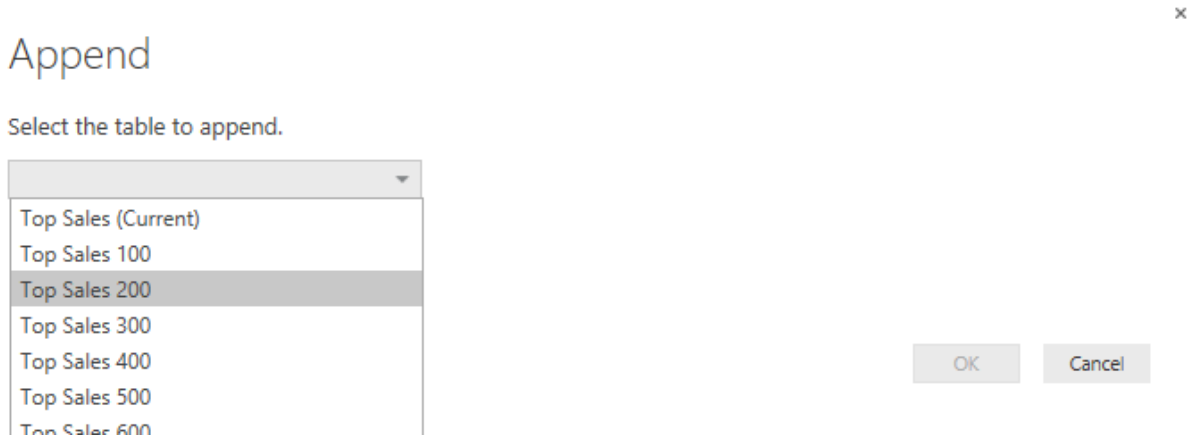
Append Queries

Now let's combine queries into this new query;

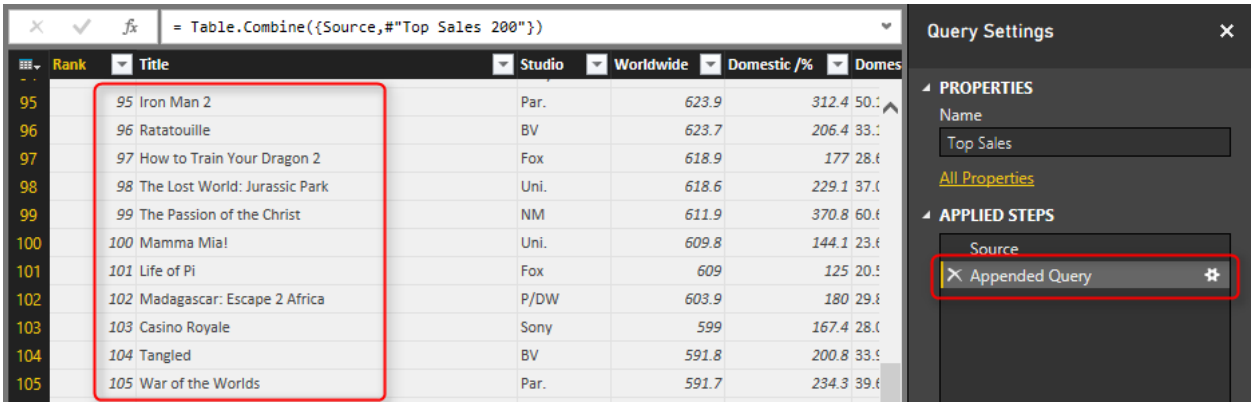
Click on Top Sales and then from the menu (Home) click on Append Queries



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;



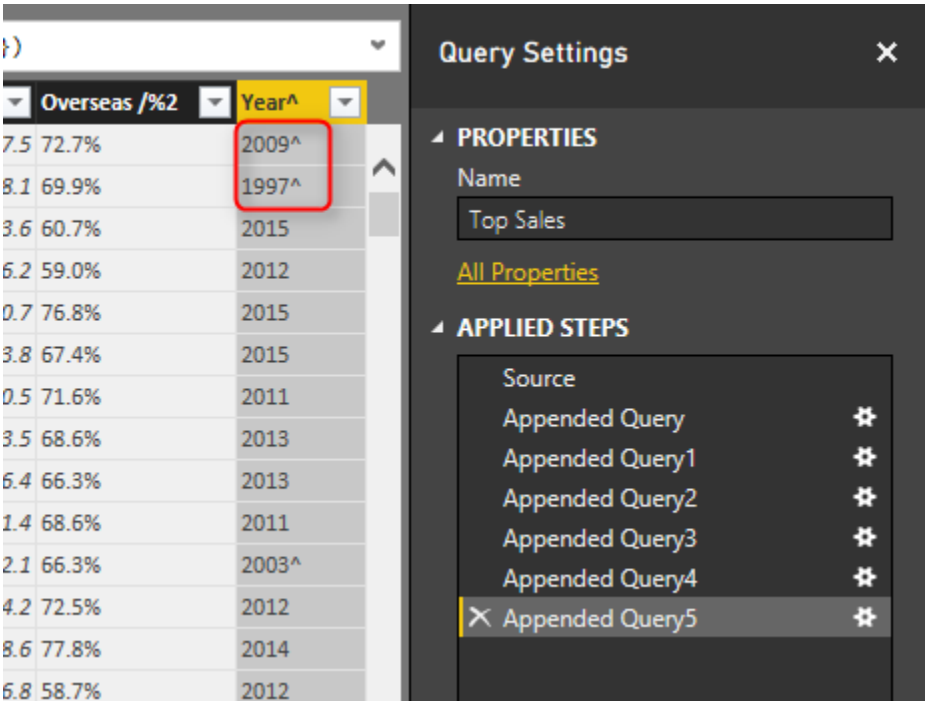
as you see in the screenshot above you can choose other queries. For append to works, best queries have to be in the same structure (number of columns, the order of columns, the data type of columns….). Choose Top Sales 200 in this window and click OK. This will create another step in the query setting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



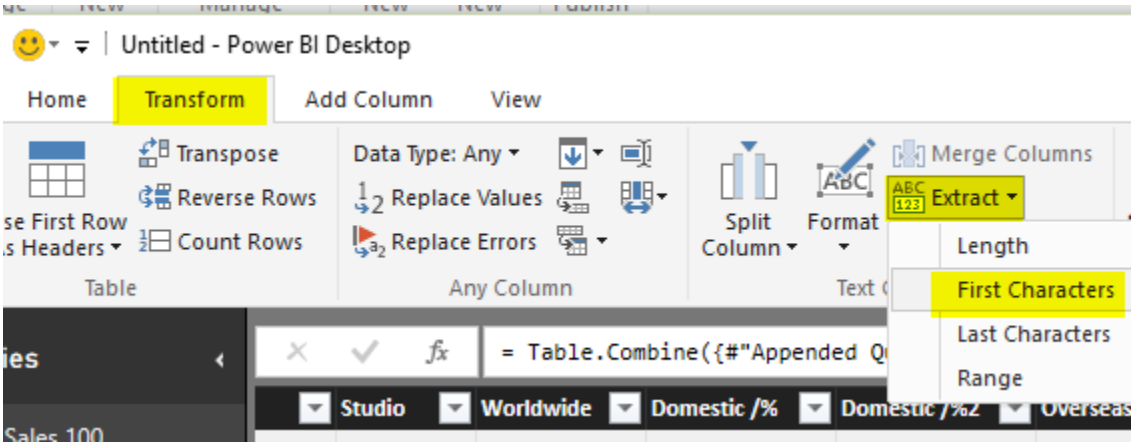
Continue this process to add all 615 top movies into Top Sales query.

Extract First Characters

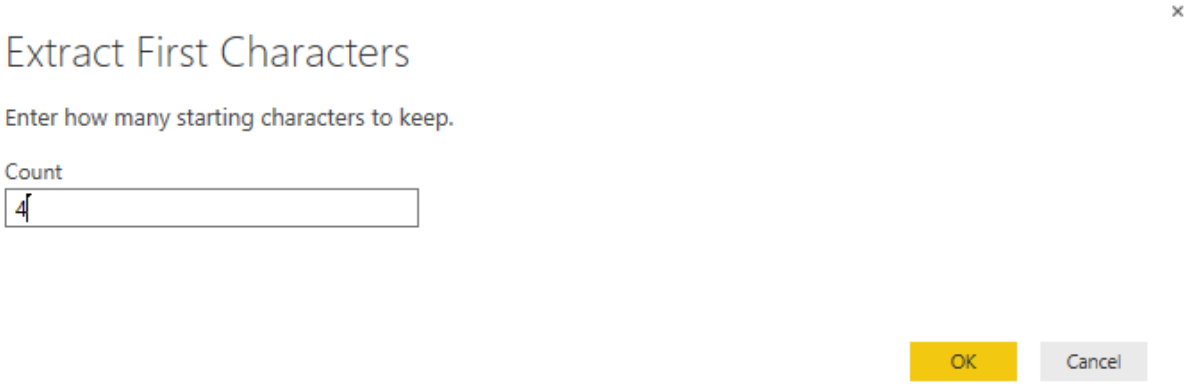
After doing this change Let's clean the Year column data; Year column has a special character in some values as below;



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

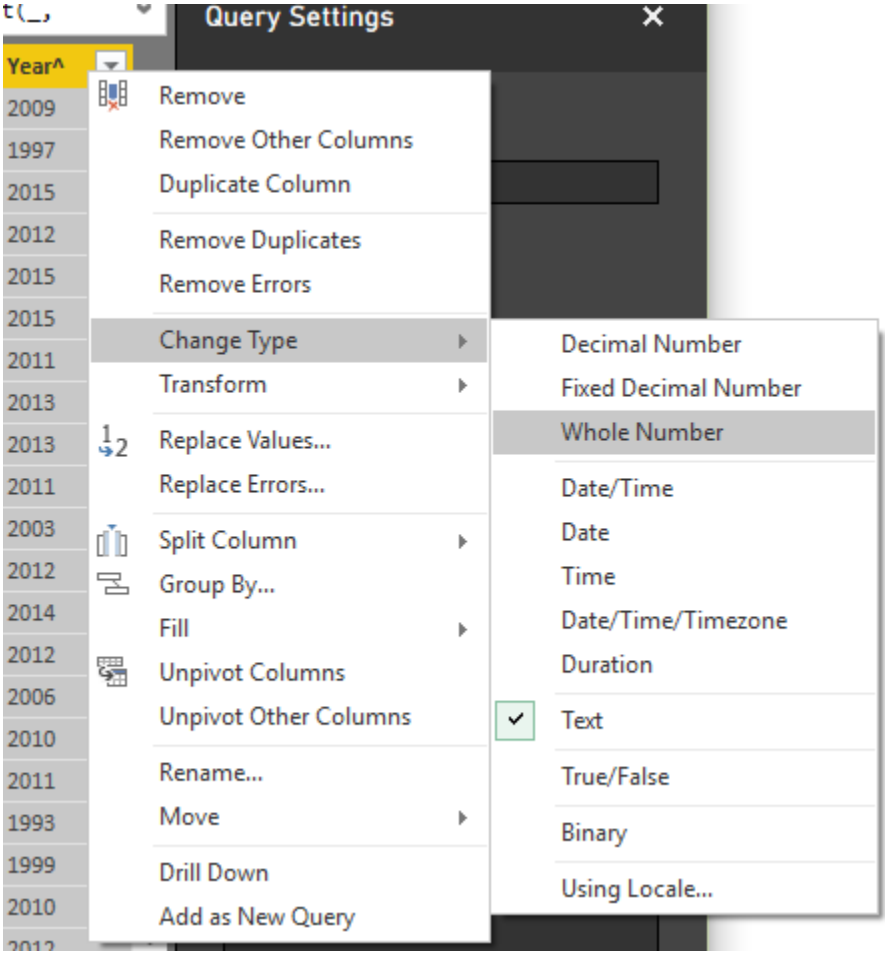


Enter 4 in the Extract First Characters dialog box (because the year isn't more than four characters). Then click on OK.



You will see that year column is clean now without any extra characters. That was easy data transform. This option in the transformation menu (Extract First Characters) has been added recently in Power BI Desktop.

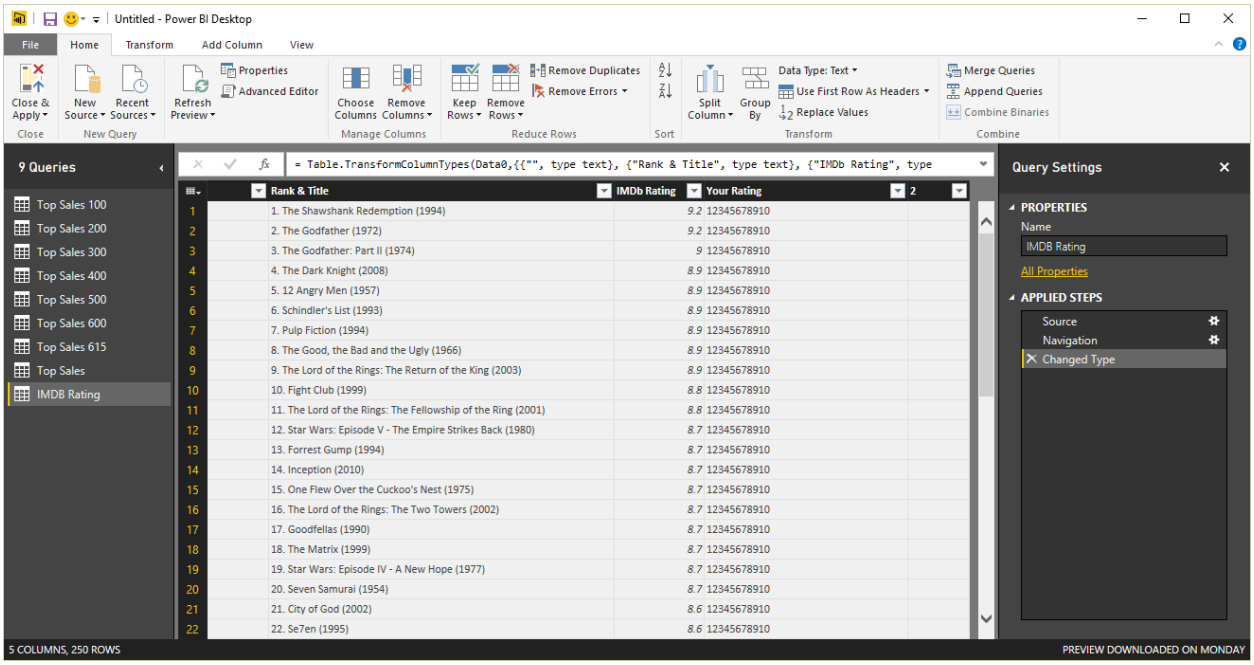
You can even now change the data type of this column to the whole number. Right click on Year column and then under Change Type choose the Whole Number.



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 Web. Enter the URL as http://www.imdb.com/chart/top

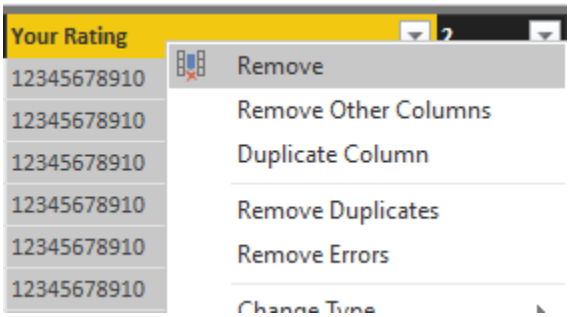
In the Navigator window, Table 0 contains the data that we want, so load it with clicking on OK. the data loads into Query Editor as the screenshot below illustrates



Rename the query to IMDB Rating.

Remove Columns

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

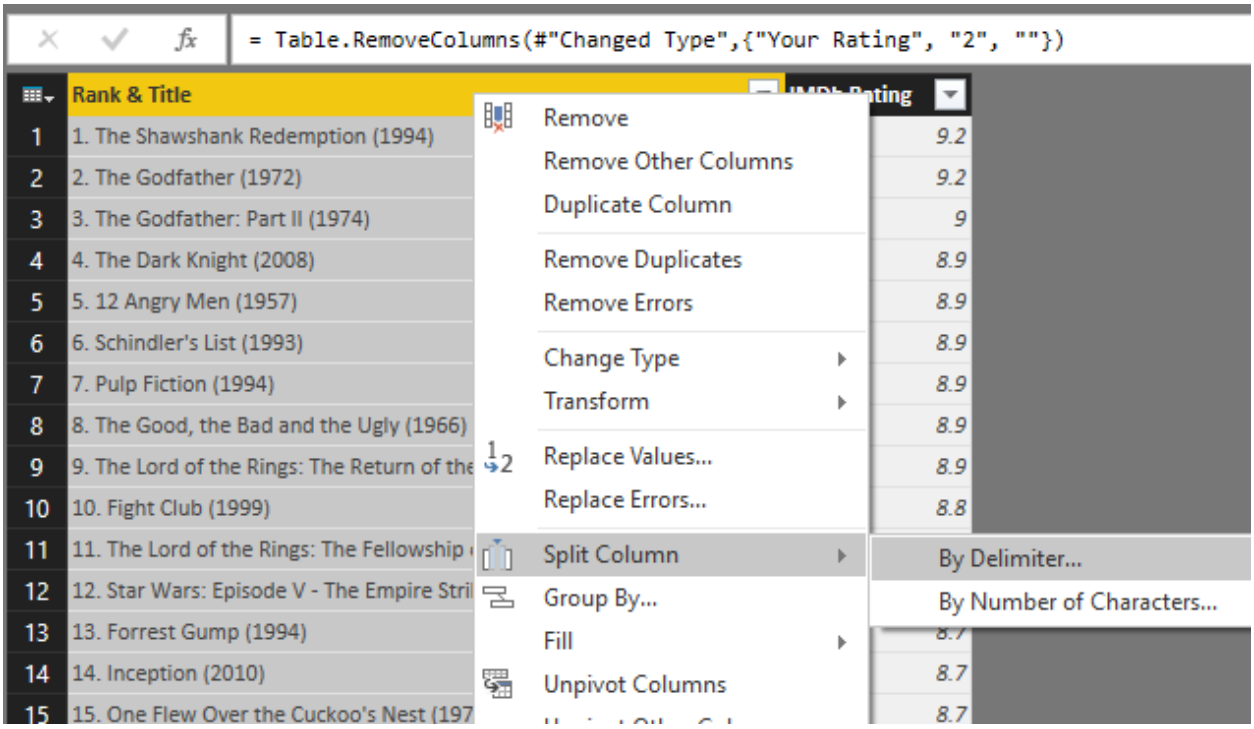


Split Column

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 a 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



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 a 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.

after the 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

Rename the Rank & Title.1 column to Rank.

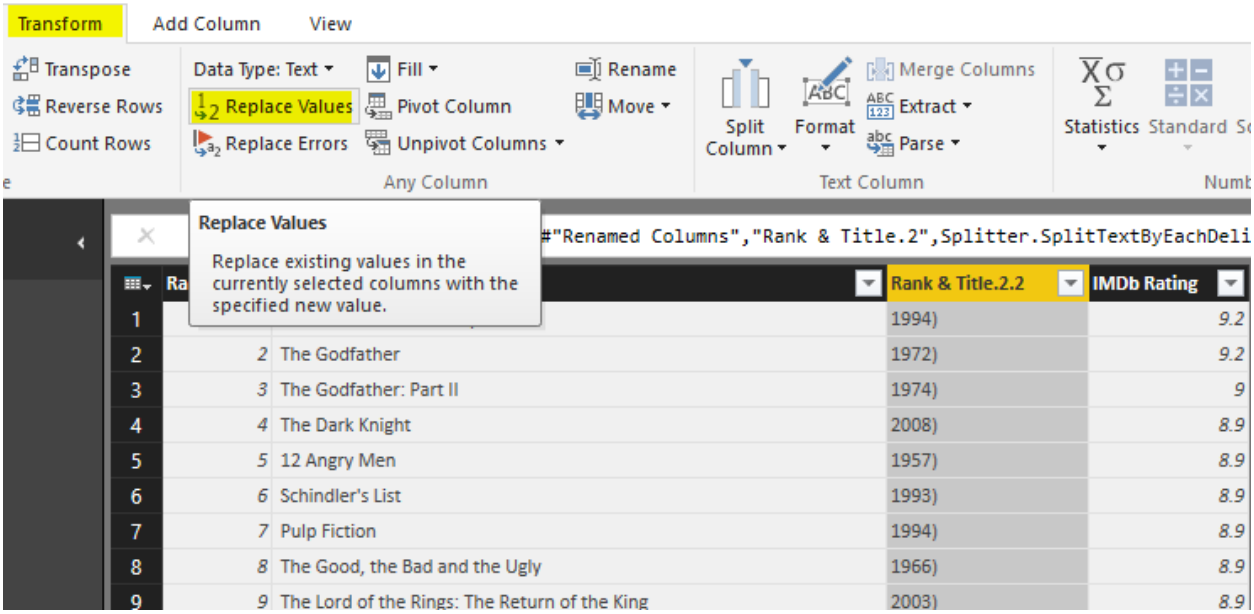
Now Let's split title and year. Year value is surrounded between brackets, so we can use the same split column method, this time using open bracket as below;

The result set looks like below screenshot;

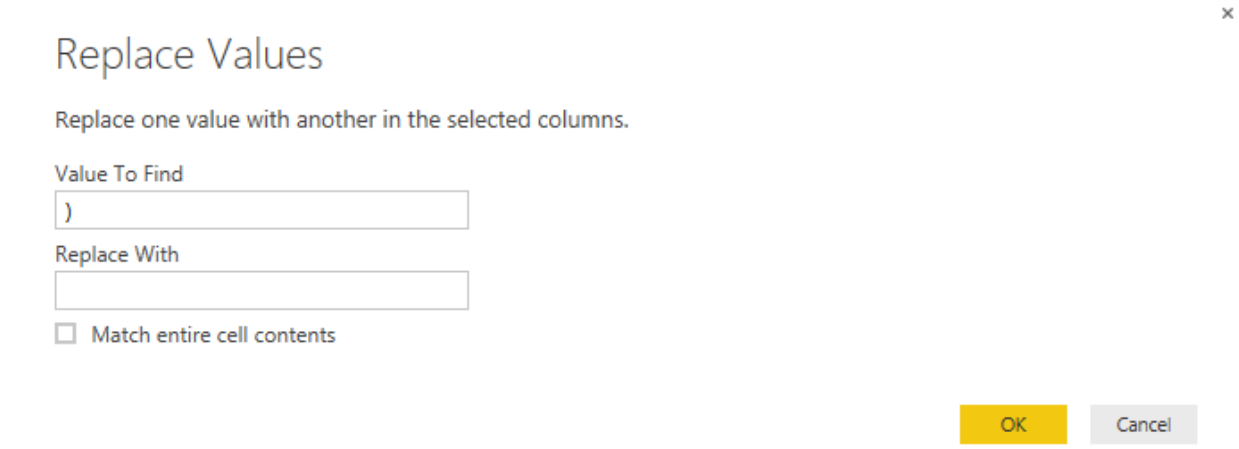
	Rank	Rank & Title.2.1	Rank & Title.2.2	IMDb Rating
1	1	The Shawshank Redemption	1994)	9.2
2	2	The Godfather	1972)	9.2
3	3	The Godfather: Part II	1974)	9
4	4	The Dark Knight	2008)	8.9
5	5	12 Angry Men	1957)	8.9
6	6	Schindler's List	1993)	8.9
7	7	Pulp Fiction	1994)	8.9
8	8	The Good, the Bad and the Ugly	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



Replace close bracket with an empty string as below;



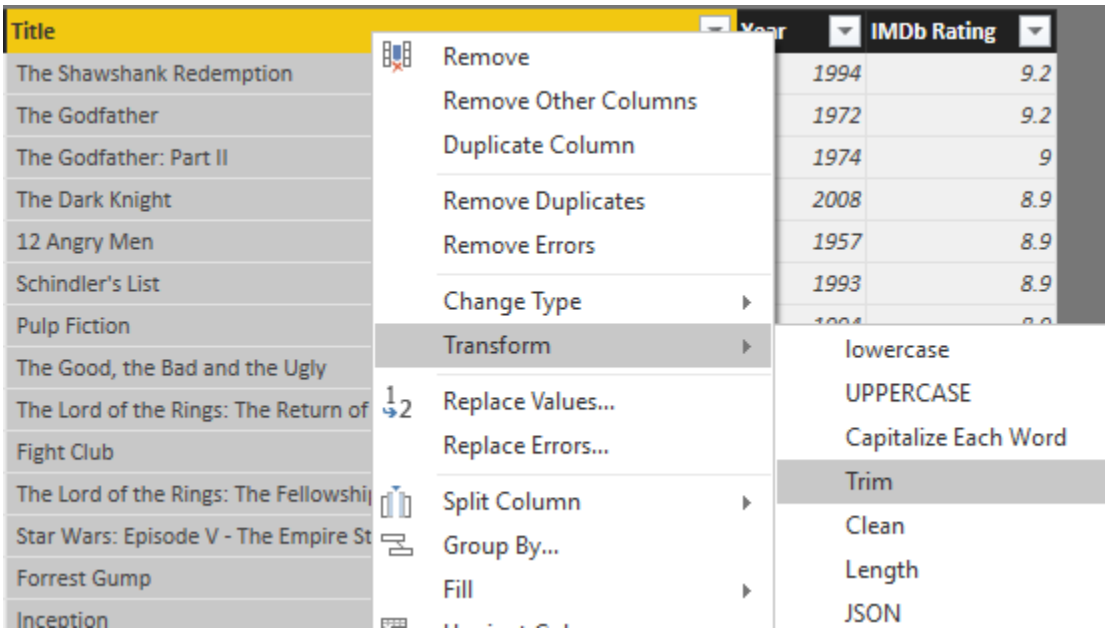
Result set would have the close bracket removed. rename the column to Year, and change its data type to Whole number (change data type with right click on the column)

	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;

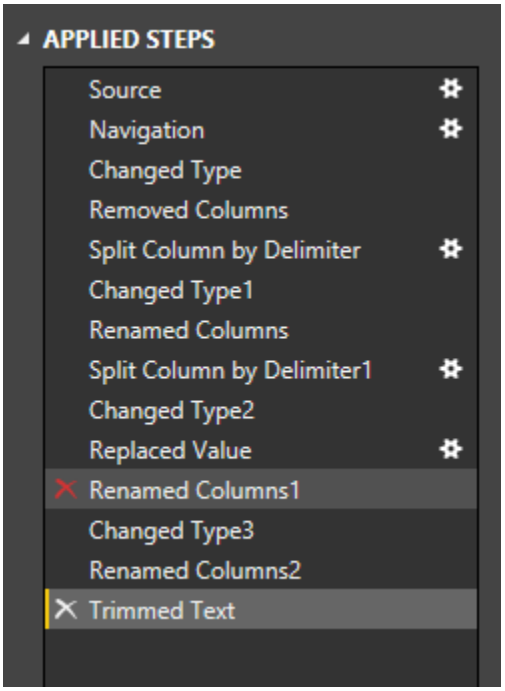
Right click on this column and then under transform choose Trim. This will remove all heading and trailing spaces from values in this column.



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

Applied Steps

One of the most useful sections of the Query Editor window is Applied Steps in the Query Settings Pane. This section of the 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 a step, and the main pane will show you the data at that step! Such an awesome way of keeping track of steps.

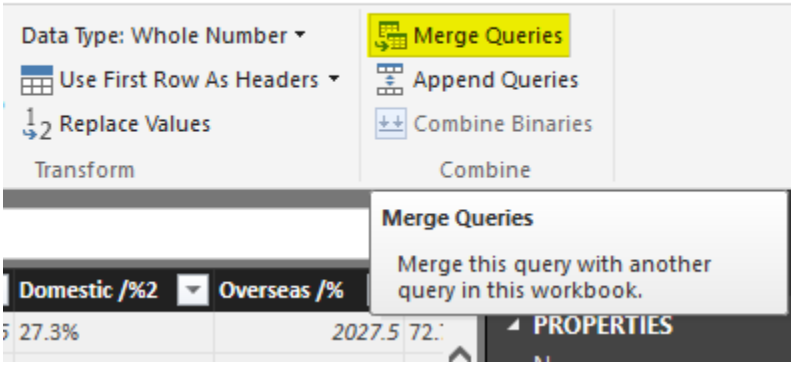


You can even remove a step, or you can change settings of a step with clicking on remove icon (on the left side of step) or setting icon (on the right side of the step, but only for steps that settings are applicable to them).

Final Merge

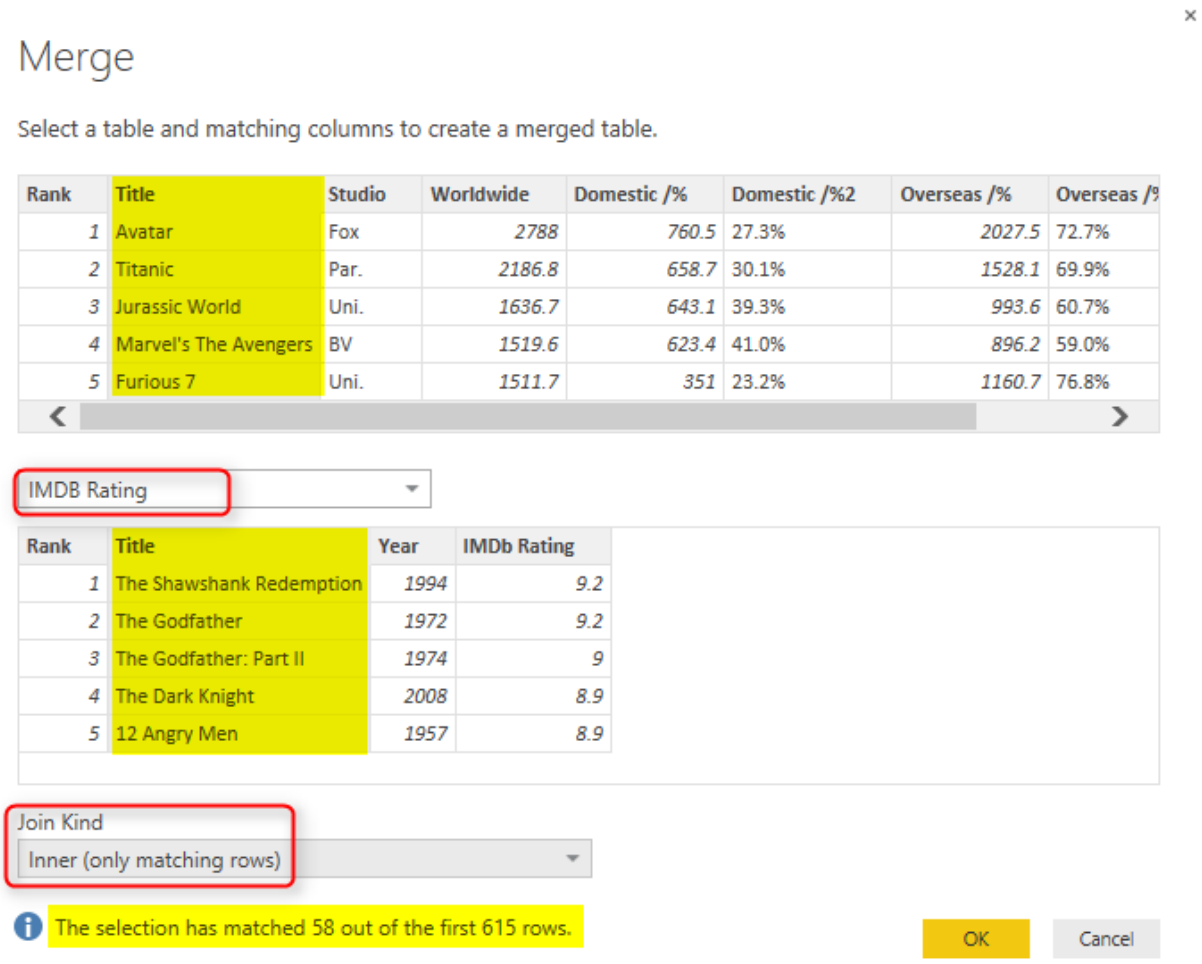
We've prepared both data sets for a final merge together to see how best-selling movies are among top user rated films. So we are one step away from this result. We have to merge these two data sets or Join them in another word.

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



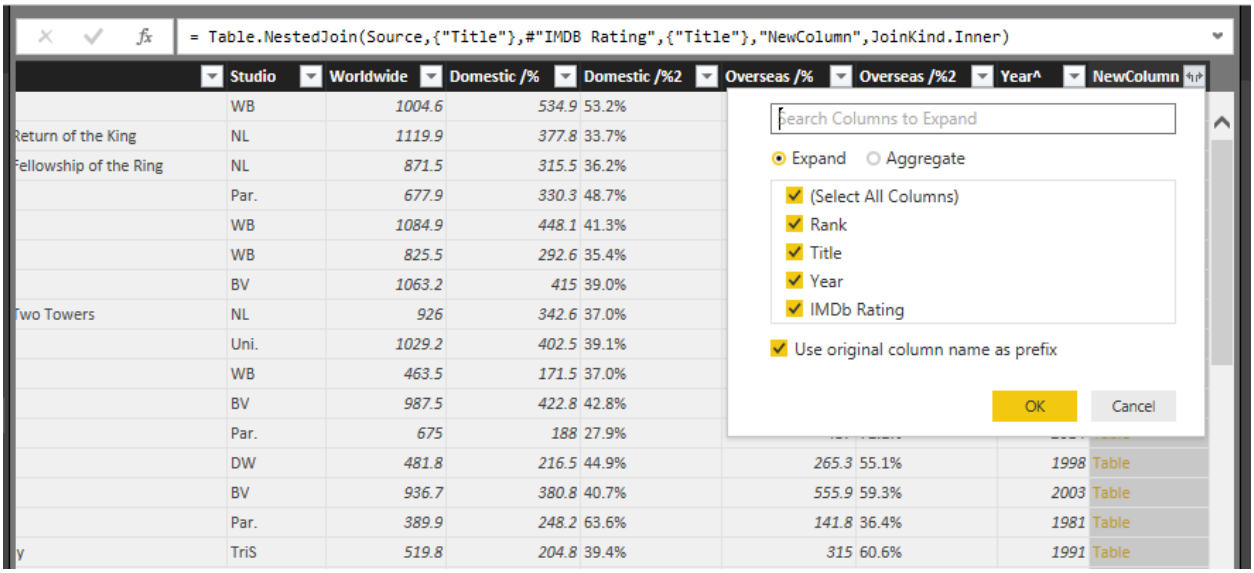
Merge Queries is equivalent to Join in SQL or database terminology.

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 the 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 kindly to Left outer join (this will only select all records from the first query with matched rows of that from the second query)



Notice in the 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. The 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 the first table with a new column for the new table. This new column holds table values which need to be expanded. If you click on the column header icon, you can choose which columns of the nested table you want to expand.



	Studio	Worldwide	Domestic /%	Domestic /%2	Overseas /%	Overseas /%2	Year^	NewColumn
	WB	1004.6	534.9	53.2%				
Return of the King	NL	1119.9	377.8	33.7%				
Fellowship of the Ring	NL	871.5	315.5	36.2%				
	Par.	677.9	330.3	48.7%				
	WB	1084.9	448.1	41.3%				
	WB	825.5	292.6	35.4%				
	BV	1063.2	415	39.0%				
Two Towers	NL	926	342.6	37.0%				
	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

Let's keep all columns and click OK. You can now see some movies that are among best sellers but not in top 250 users rated list of IMDB; There are movies name such 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	null	null
20	2	Titanic	Par.	2186.8	null	null	null	null
21	3	Jurassic World	Uni.	1636.7	null	null	null	null
22	4	Marvel's The Avengers	BV	1519.6	null	null	null	null
23	5	Furious 7	Uni.	1511.7	null	null	null	null
24	6	Avengers: Age of Ultron	BV	1401.3	null	null	null	null
25	7	Harry Potter and the Deathly Hallows Part 2	WB	1341.5	null	null	null	null
26	8	Frozen	BV	1274.2	null	null	null	null
27	9	Iron Man 3	BV	1215.4	null	null	null	null
28	10	Transformers: Dark of the Moon	P/DW	1123.8	null	null	null	null
29	12	Skyfall	Sony	1108.6	null	null	null	null
30	13	Transformers: Age of Extinction	Par.	1104	null	null	null	null
31	15	Pirates of the Caribbean: Dead Man's Chest	BV	1066.2	null	null	null	null
32	17	Pirates of the Caribbean: On Stranger Tides	BV	1045.7	null	null	null	null
33	19	Star Wars: Episode I - The Phantom Menace	Fox	1027	null	null	null	null
34	20	Alice in Wonderland (2010)	BV	1025.5	null	null	null	null
35	21	The Hobbit: An Unexpected Journey	WB	1021.1	null	null	null	null

Summary

In this section, you've learned the 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 a split column, replace values, change the data type, and extract part of 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: How to Analyze FIFA 2014 Worldcup

Published Date: July 26, 2015

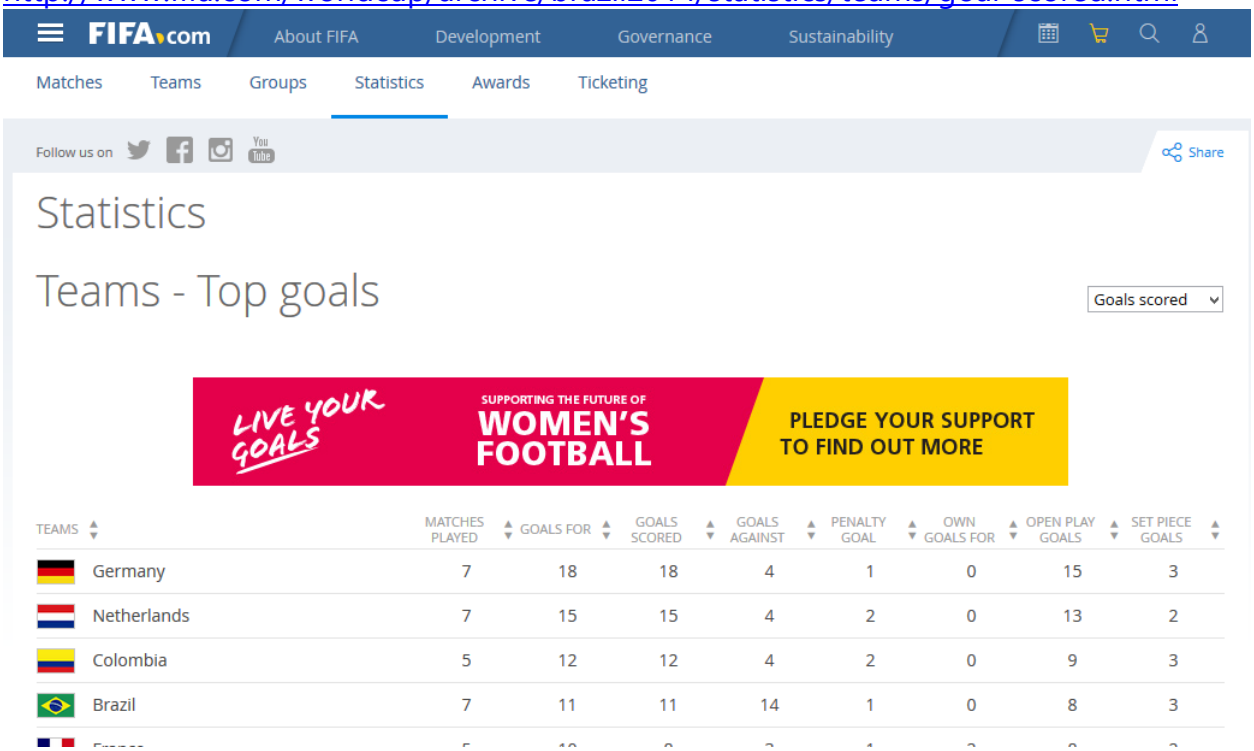
It was some time ago that I'd posted some data analysis that I've done with Power BI Designer on FIFA 2014 Worldcup results. In this post, I want to take you through the process of how to achieve that analysis. This post is an introduction to Power BI features with the example of FIFA 2014 Worldcup result set. Through this post, you will learn how to connect to web-based data sets and mash up the data with Power Query component of Power BI Desktop. And finally how to model it and visualize it in the Power BI report.
Most of the data for this example comes from datasets in FIFA 2014 Worldcup official website: http://www.fifa.com/worldcup/archive/brazil2014/index.html
For this example I use Power BI Desktop which has been announced yesterday as General Availability, you can download Power BI Desktop (32 or 64 bit) from the link below:
http://www.microsoft.com/en-us/download/details.aspx?id=45331

Data Extraction

I use more than one data set for analysis, so let's start with some basic data sets;

Goals Scored by Teams

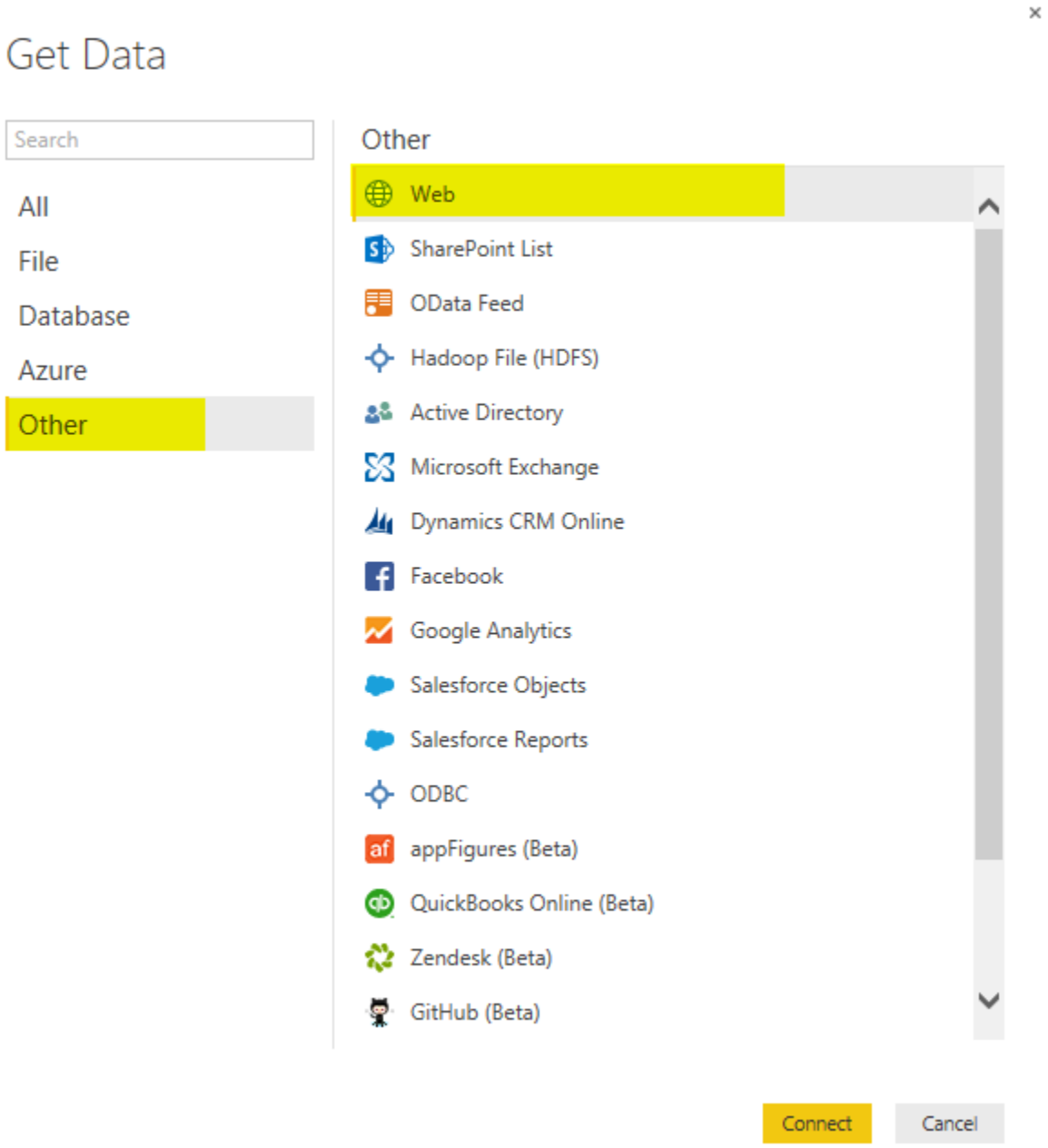
Number of goals scored by each team listed in this URL:
http://www.fifa.com/worldcup/archive/brazil2014/statistics/teams/goal-scored.html



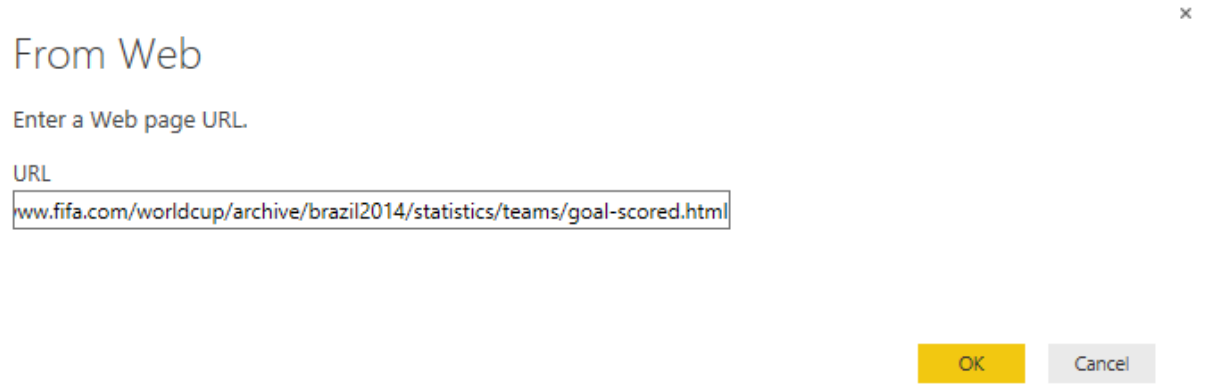
TEAMS	MATCHES PLAYED	GOALS FOR	GOALS SCORED	GOALS AGAINST	PENALTY GOAL	OWN GOALS FOR	OPEN PLAY GOALS	SET PIECE GOALS
Germany	7	18	18	4	1	0	15	3
Netherlands	7	15	15	4	2	0	13	2
Colombia	5	12	12	4	2	0	9	3
Brazil	7	11	11	14	1	0	8	3

Steps to get this dataset into Power BI is;

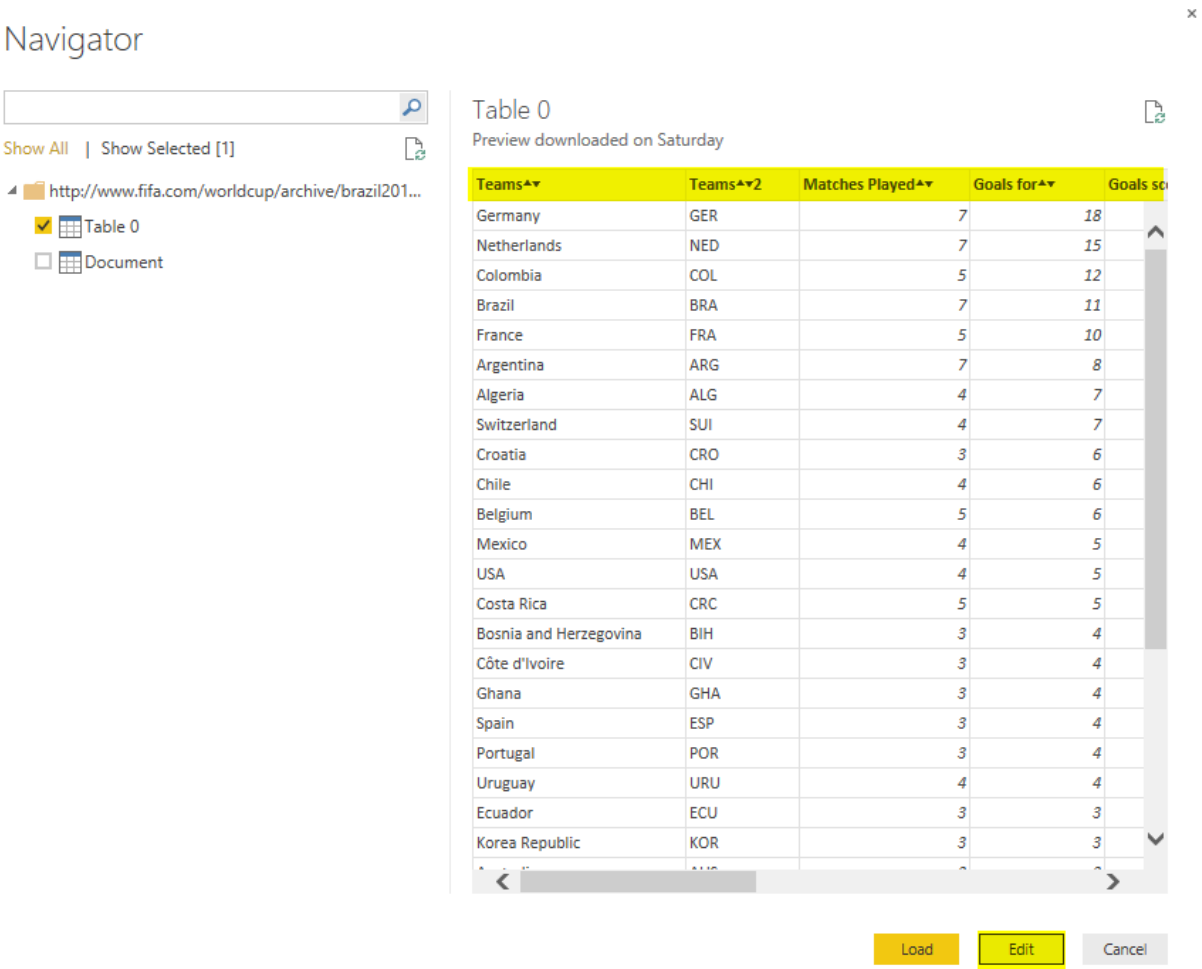
- Open Power BI Desktop
- Choose Get Data
- In Get Data Window, under Other, click on Web



4. In "From Web" dialog box, Enter the URL and click OK; http://www.fifa.com/worldcup/archive/brazil2014/statistics/teams/goal-scored.html

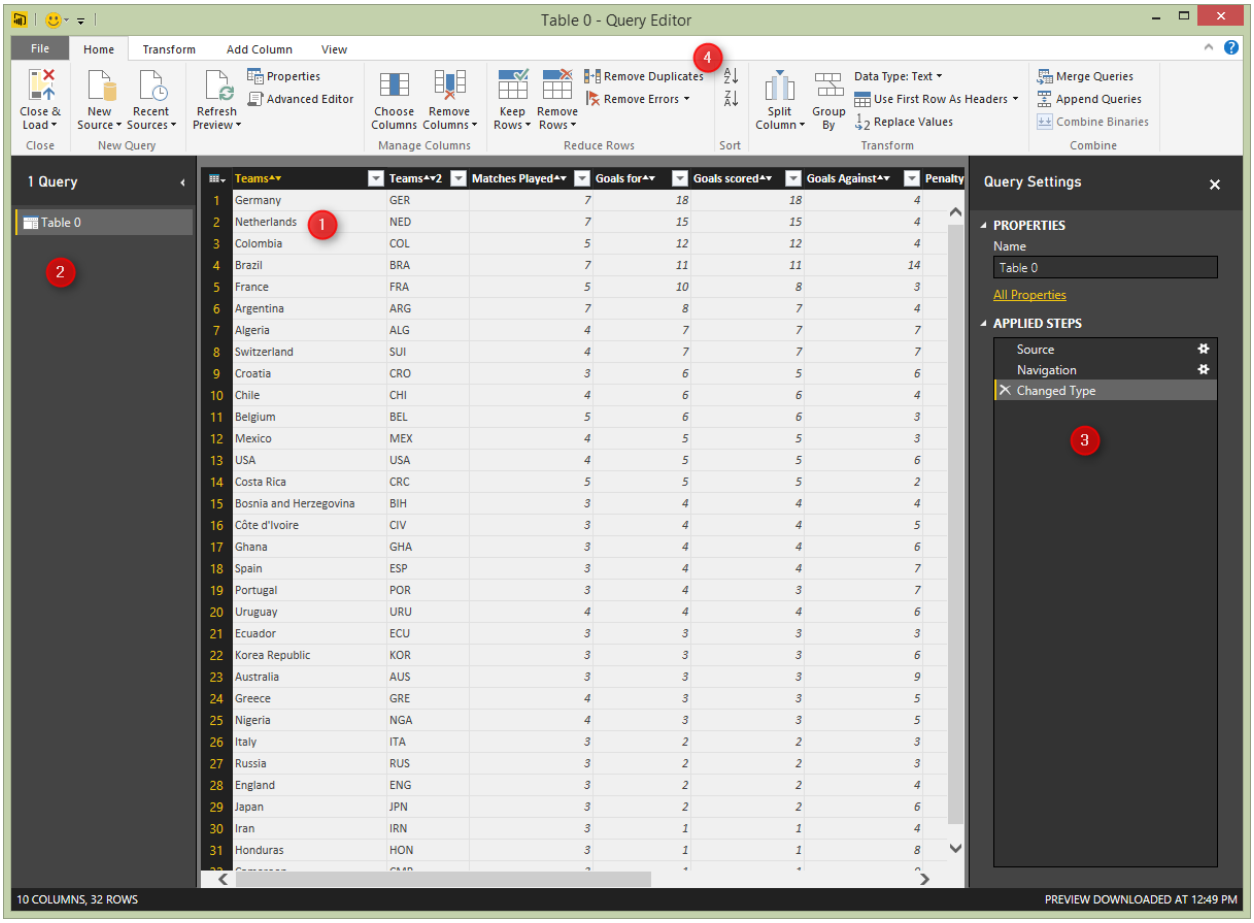


3. a Navigator window will open as below, click on the table that contains dataset of goal scorers and then click on Edit to open a Query Editor window



Teams	Teams2	Matches Played	Goals for	Goals sc
Germany	GER	7	18	
Netherlands	NED	7	15	
Colombia	COL	5	12	
Brazil	BRA	7	11	
France	FRA	5	10	
Argentina	ARG	7	8	
Algeria	ALG	4	7	
Switzerland	SUI	4	7	
Croatia	CRO	3	6	
Chile	CHI	4	6	
Belgium	BEL	5	6	
Mexico	MEX	4	5	
USA	USA	4	5	
Costa Rica	CRC	5	5	
Bosnia and Herzegovina	BIH	3	4	
Côte d'Ivoire	CIV	3	4	
Ghana	GHA	3	4	
Spain	ESP	3	4	
Portugal	POR	3	4	
Uruguay	URU	4	4	
Ecuador	ECU	3	3	
Korea Republic	KOR	3	3	

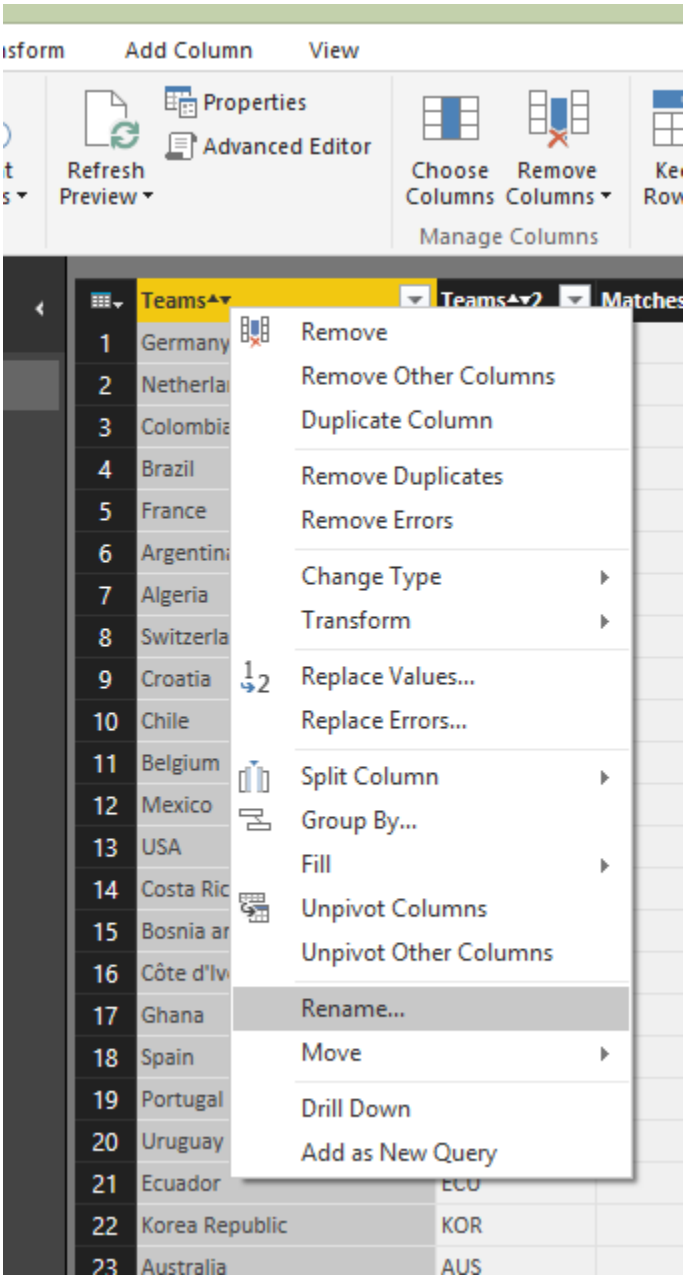
- The Query Editor window will appear with the data set loaded and displayed in the main pane.



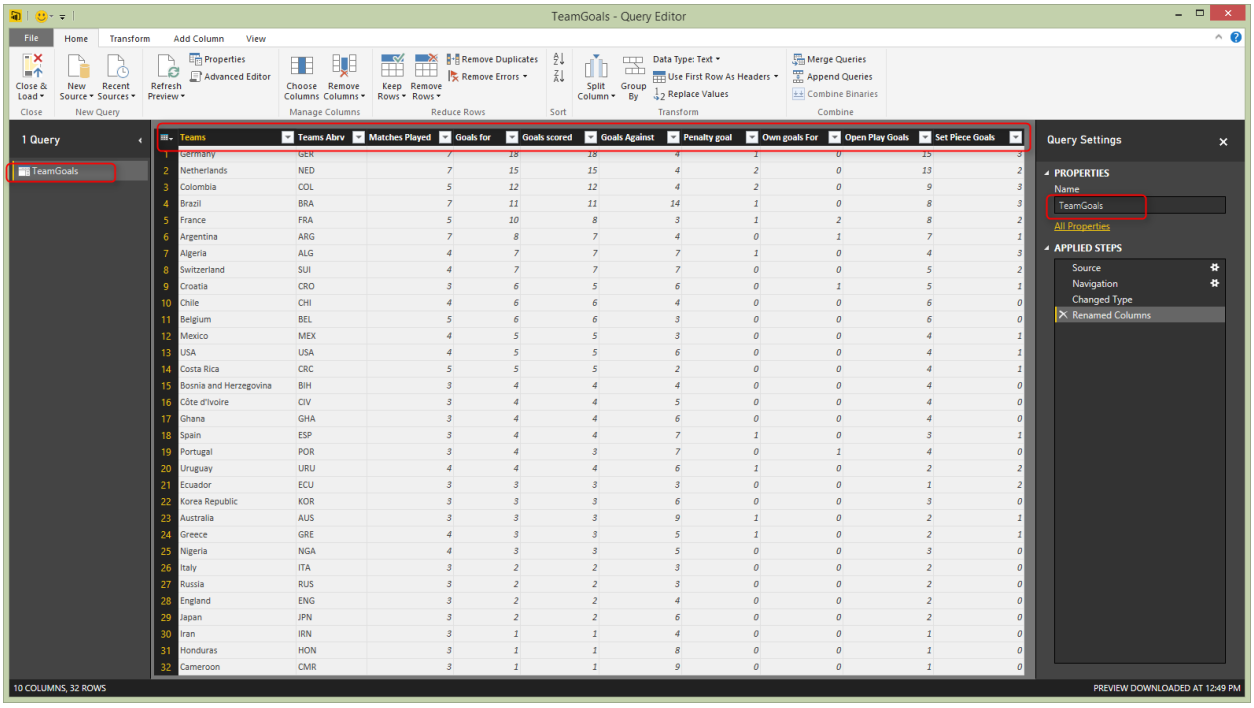
Query Editor is the Power Query editor window, which gives you the ability to transform data and then load it into the model that Power BI can use. There are different panes in Power Query Editor window here;

4. The main pane, that shows a preview of the dataset after applying transformation listed in the Query Setting pane.
5. Query pane, which shows all queries used
6. Query setting pane; here you can see steps applied on the dataset, you can delete them or change them as you want.
7. Transformation Pane; all transformation options listed in top menu here, you can choose to work with any of them.

This data set is almost ready to use, the only transformation I want to make here is to remove extra two characters at the end of each column name that came from icons used in the FIFA website. So rename every column with right click on that and choose "rename column" option from the menu.



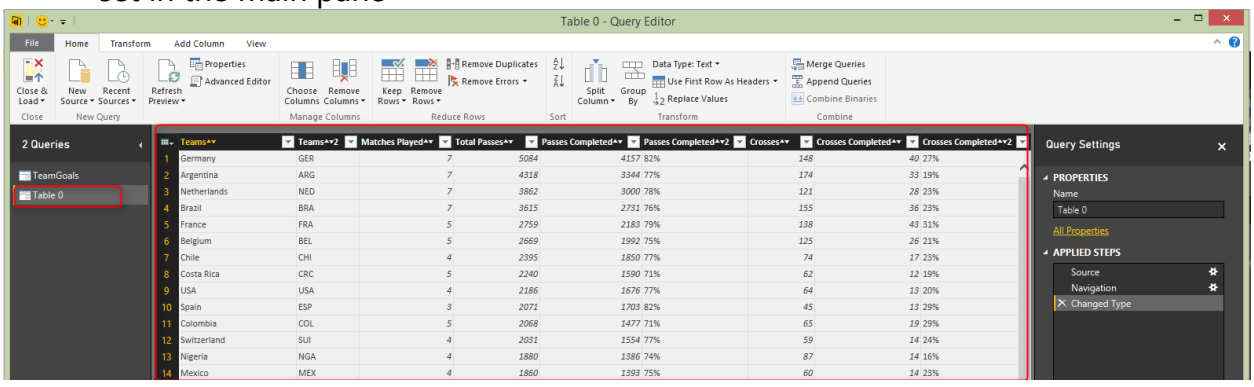
Also, rename the Query name to TeamGoals in the right-hand side pane in Query Setting section. Here is the result set after the above changes;



Passes by Teams

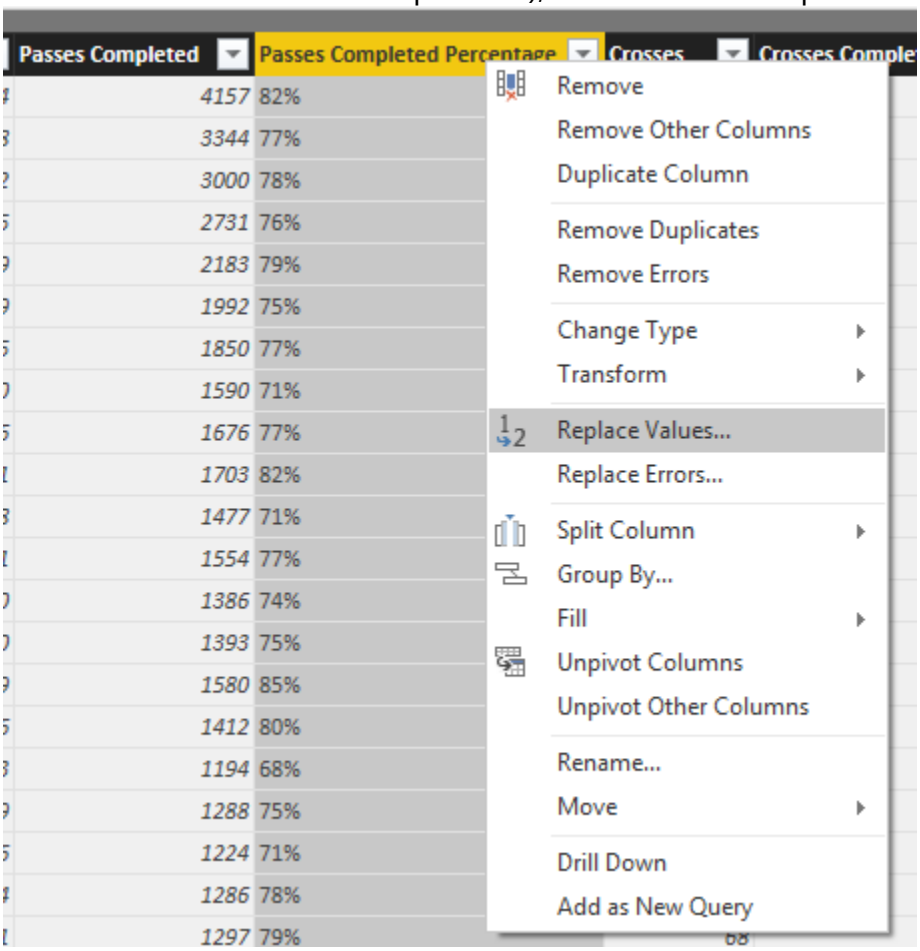
The second data set is passed by teams which we get from this URL: http://www.fifa.com/worldcup/archive/brazil2014/statistics/teams/passes.html

- Without closing Query Editor window, on the Menu (under Home, New Query) click on New Source, and choose from Web. In the From Web paste URL above and press OK.
- In Navigator window choose the table that contains passes by teams data set, and click OK.
- Now in the same Query Editor window, you will see Table 0 loaded with its data set in the main pane

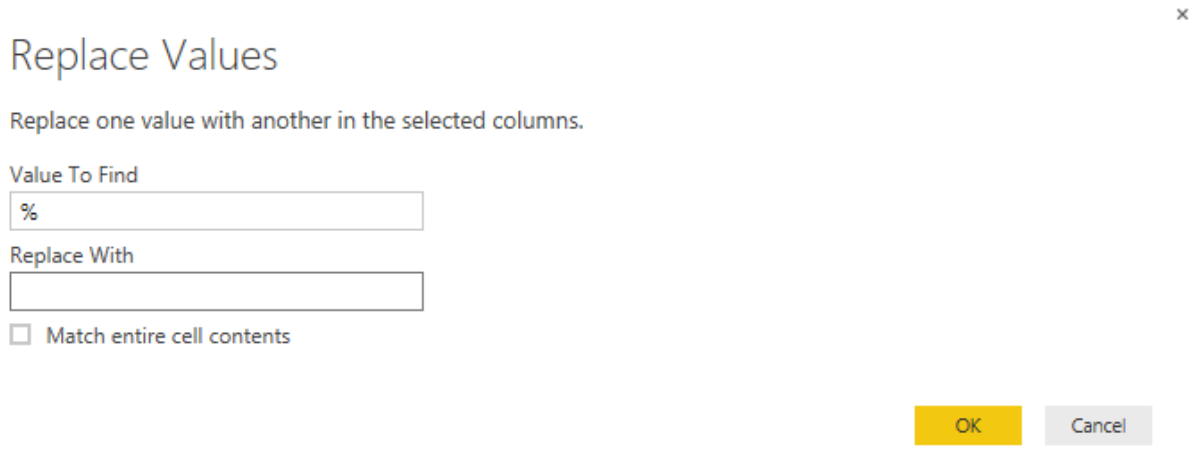


- For this dataset again we have to change column names and remove two extra characters from each column name. Rename Team2 column name to Teams Abrv.

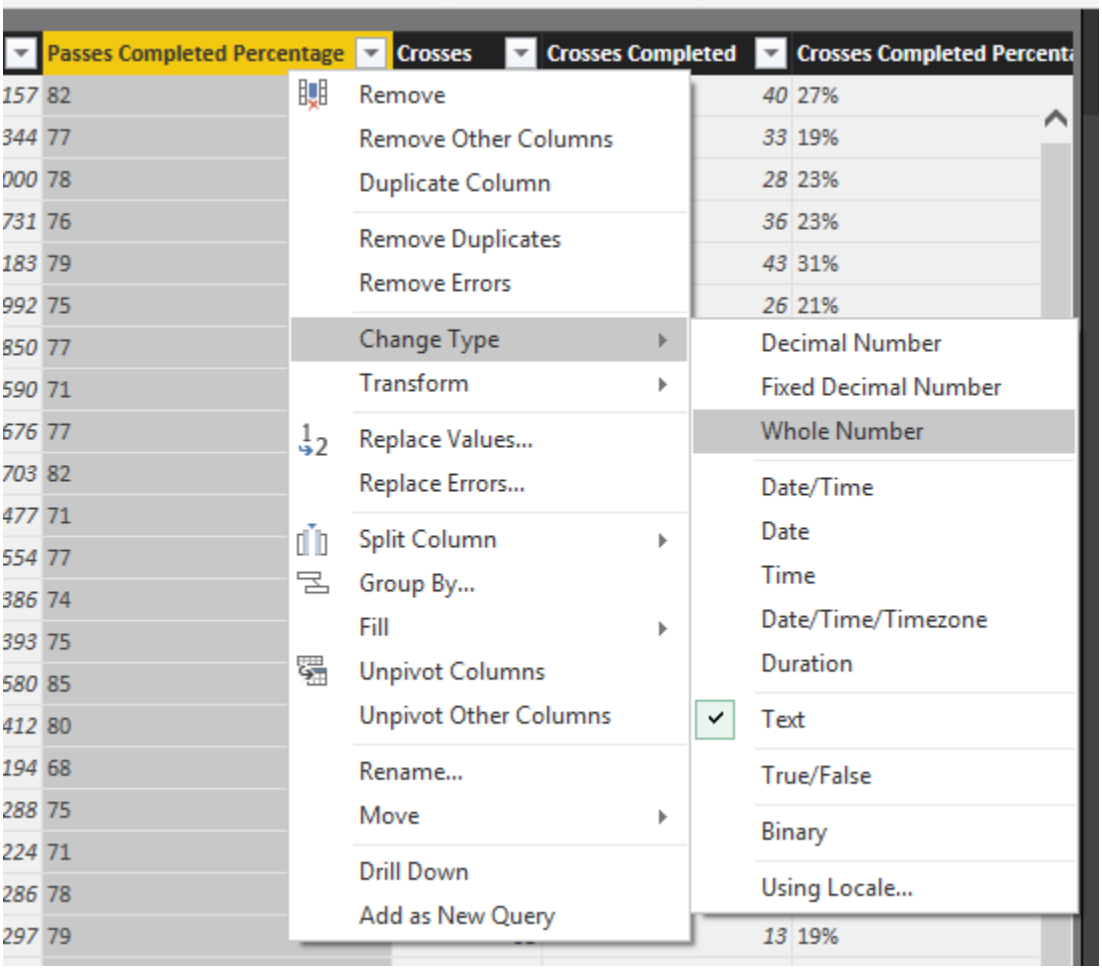
- There are some percentage columns in this data set, which considered as text column (because of % character in the value), we want to remove the character and change the data type to be Decimal number.
- Right click on Passes Completed Percentage column (this is renamed column version of Passes Completed 2), and choose to Replace Values



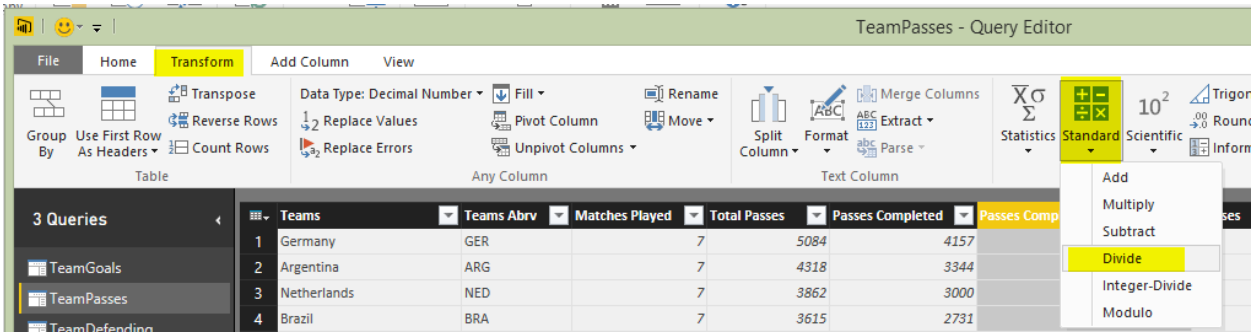
1. Replace the % character with an empty string as mentioned below



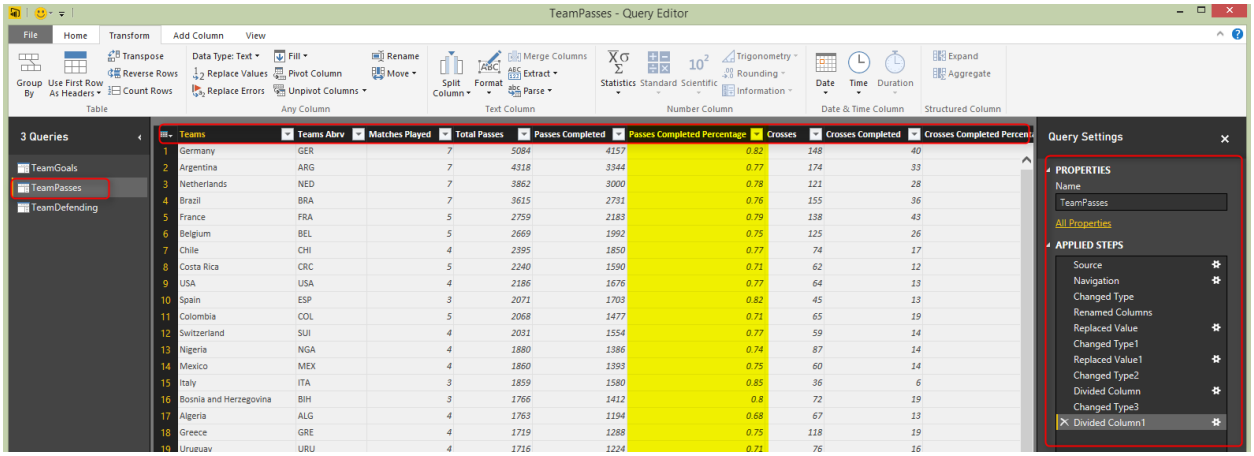
1. After applying the above transformation, you will see % character removed. Now change the data type of this column to Decimal number



2. Now we have to divide the value of this column by 100 (to have a real percentage value). Click on the Passes Completed Percentage Column. and from Transform Menu, Under Number Column, Under Standard, select Divide



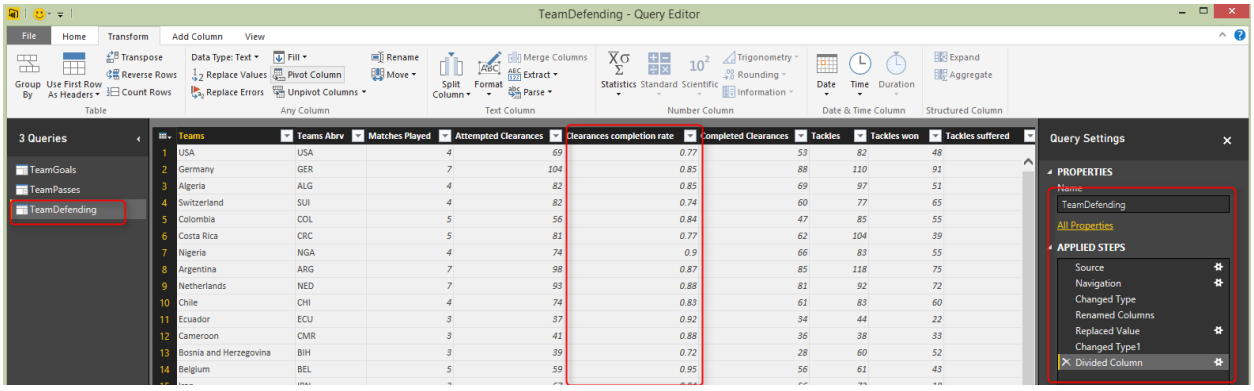
2. Enter the value as 100 in the Divide dialog box, and click OK.
3. Do the same for Crosses Completed Percentage column (renamed from Crosses Completed 2)
4. Rename the query to TeamPasses. The result should look like below screenshot



Teams Defending

I would also like to add another data set to this analysis which is Teams Defending result set. URL:
http://www.fifa.com/worldcup/archive/brazil2014/statistics/teams/defending.html

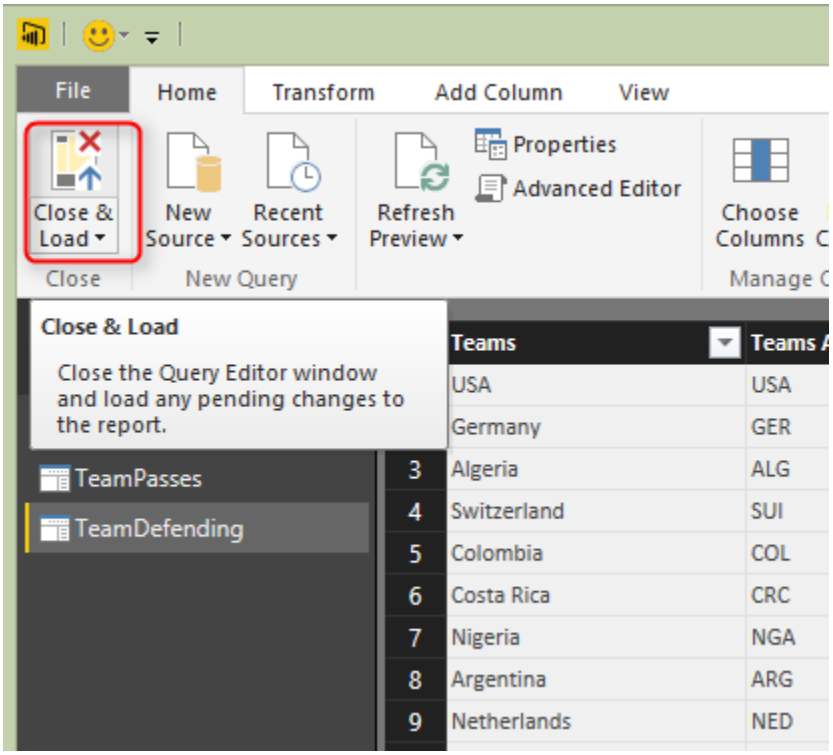
5. Get Data again, from New Source, From Web, and enter URL above.
6. After loading the data set in Query Editor apply below changes (as illustrated above)
7. Rename columns and remove extra two characters
8. remove % character and change the data type of percentage columns to the Decimal number
9. Divide column by 100 to get an actual percentage value.
10. rename the query to TeamDefending
11. The result set should look like below:



Data Modelling

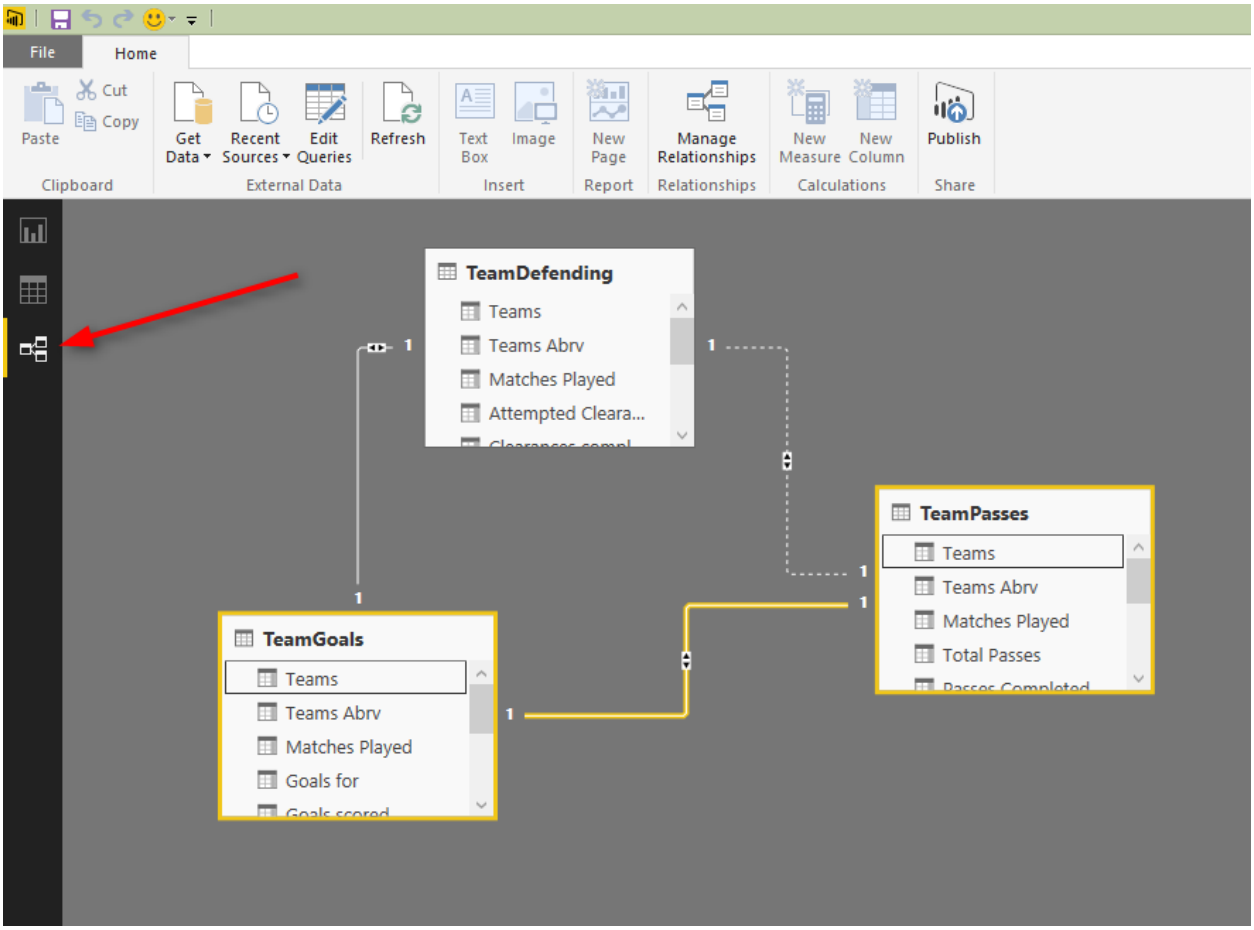
We can add much more data sets from the FIFA official website. But for this example let's keep this simple and stick to three data sets above. In this section, we would model the relationship between these data sets, and model it to be used in the reporting section.

in Query Editor window click on Close & Load icon in the Home tab, under Close section

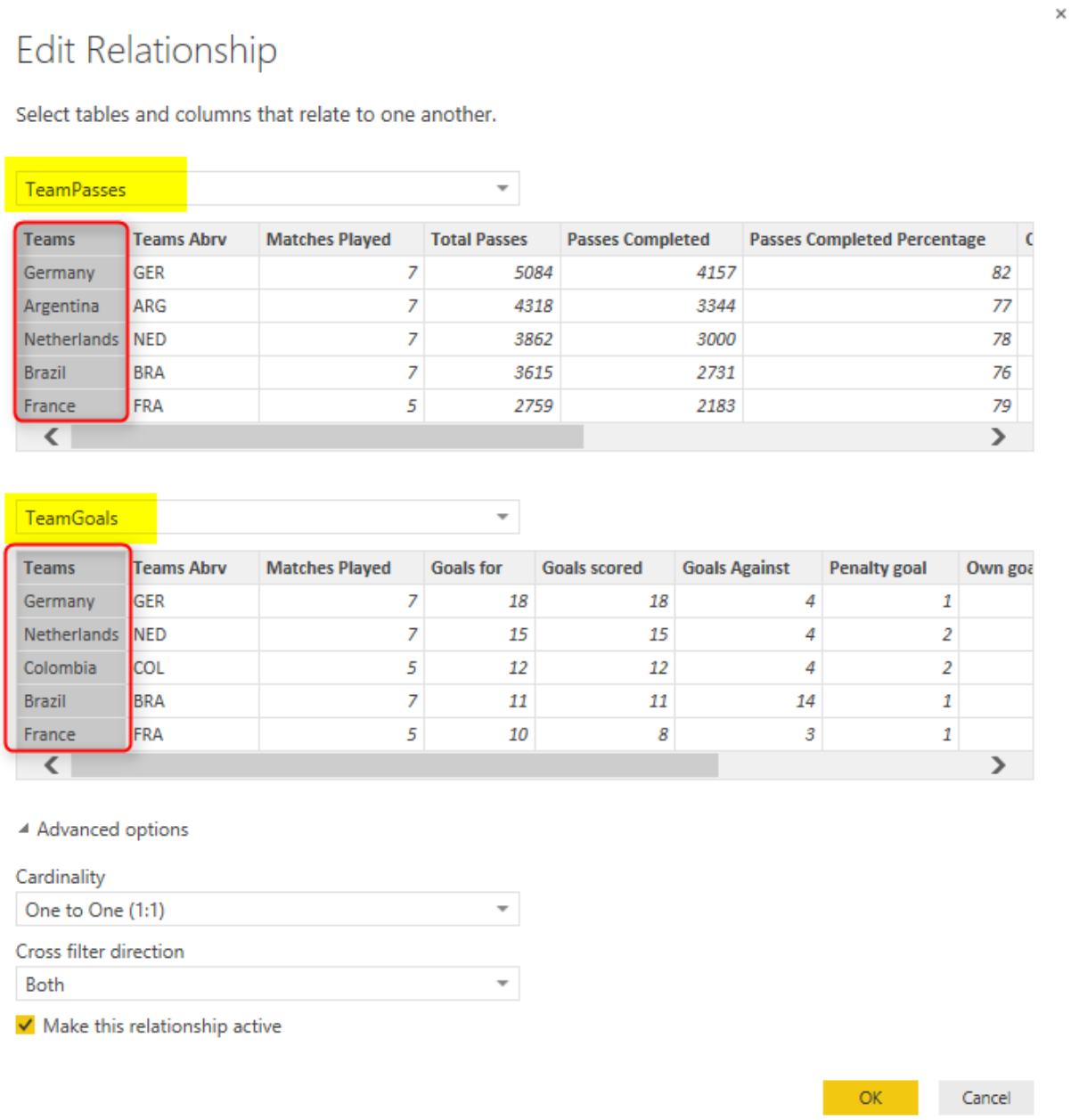


Relationship Tab

In the Power BI Desktop from left pane click on the Relationships tab and you will see the relationship created itself by the similarity of column name between queries.



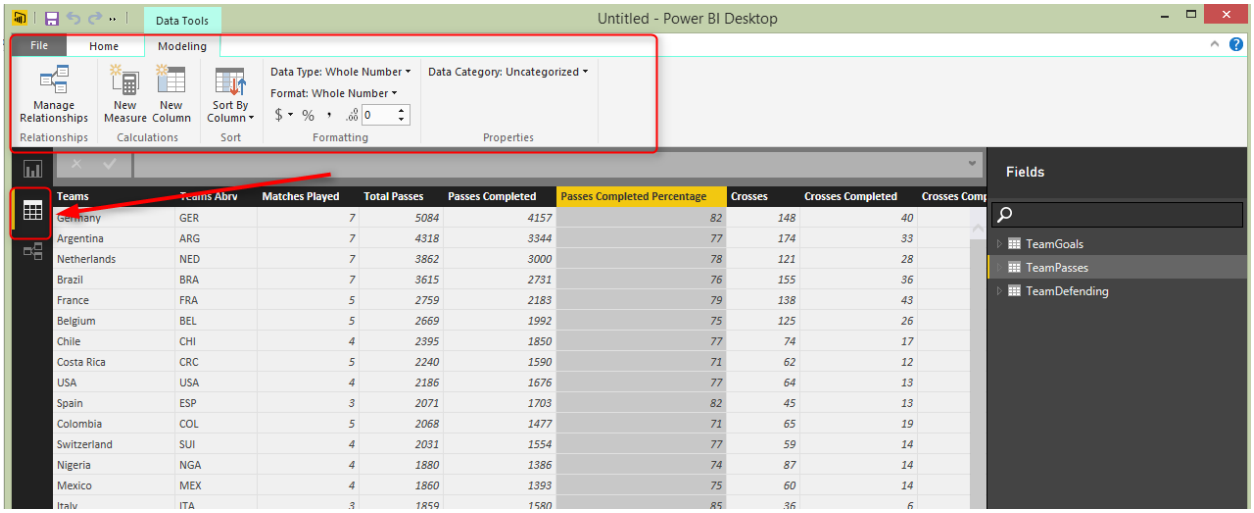
You can also change the relationship if you want. The current relationship is based on Teams column as the key column.



I'll keep the relationship as it is.

Data Tab

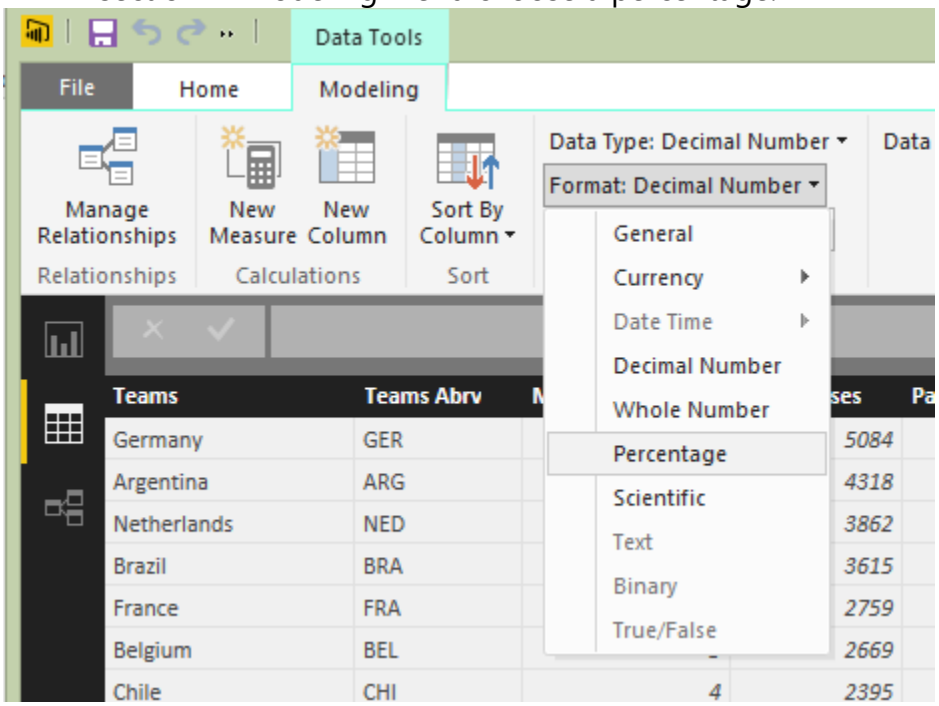
In the data tab, you can view the data set, and set the formatting of the columns or add new measures (DAX measures) if you want. Screenshot below shows how to access the data tab from the left pane. You can also see the Modeling menu that gives you the ability to create and edit measures, as well as appropriately format them. You can also access managing relationships from this menu.



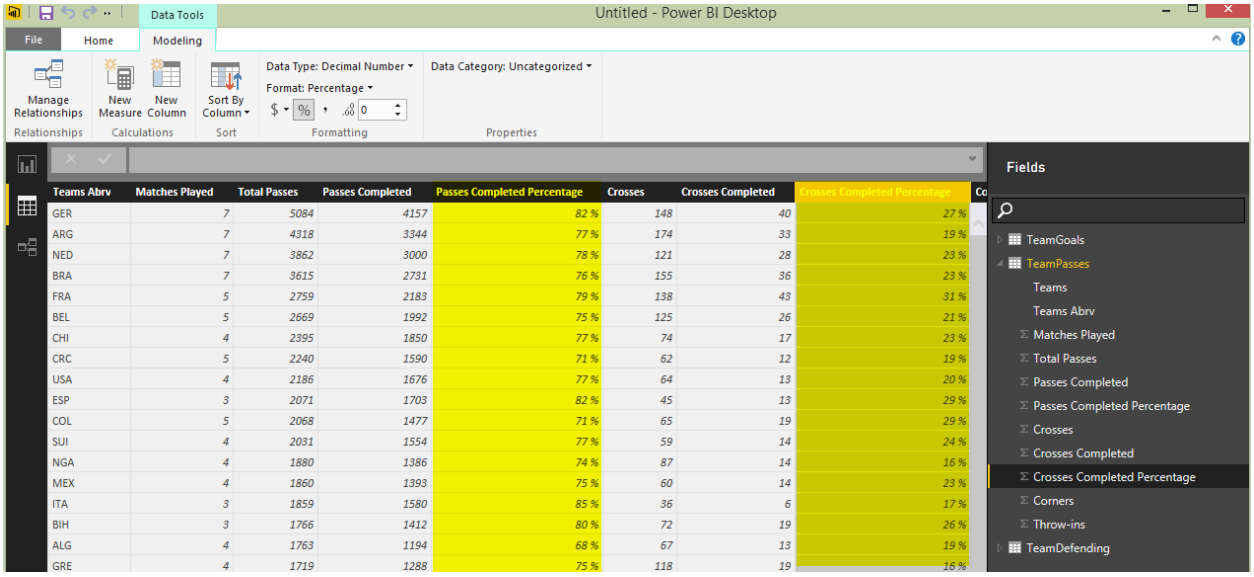
Teams	Teams Abrv	Matches Played	Total Passes	Passes Completed	Passes Completed Percentage	Crosses	Crosses Completed	Crosses Comp
Germany	GER	7	5084	4157	82	148	40	
Argentina	ARG	7	4318	3344	77	174	33	
Netherlands	NED	7	3862	3000	78	121	28	
Brazil	BRA	7	3615	2731	76	155	36	
France	FRA	5	2759	2183	79	138	43	
Belgium	BEL	5	2669	1992	75	125	26	
Chile	CHI	4	2395	1850	77	74	17	
Costa Rica	CRC	5	2240	1590	71	62	12	
USA	USA	4	2186	1676	77	64	13	
Spain	ESP	3	2071	1703	82	45	13	
Colombia	COL	5	2068	1477	71	65	19	
Switzerland	SUI	4	2031	1554	77	59	14	
Nigeria	NGA	4	1880	1386	74	87	14	
Mexico	MEX	4	1860	1393	75	60	14	
Italy	ITA	3	1859	1580	85	36	6	

Let's set the formatting of percentage columns appropriate;

1. in the Fields pane in right-hand side click on TeamPasses.
2. In the data set click on Passes Completed Percentage, and from Formatting section in Modeling menu choose a percentage.



1. Do this for all percentage columns in TeamPasses and TeamDefending



Teams Abrv	Matches Played	Total Passes	Passes Completed	Passes Completed Percentage	Crosses	Crosses Completed	Crosses Completed Percentage
GER	7	5084	4157	82 %	148	40	27 %
ARG	7	4318	3344	77 %	174	33	19 %
NED	7	3862	3000	78 %	121	28	23 %
BRA	7	3615	2731	76 %	155	36	23 %
FRA	5	2759	2183	79 %	138	43	31 %
BEL	5	2669	1992	75 %	125	26	21 %
CHI	4	2395	1850	77 %	74	17	23 %
CRC	5	2240	1590	71 %	62	12	19 %
USA	4	2186	1676	77 %	64	13	20 %
ESP	3	2071	1703	82 %	45	13	29 %
COL	5	2068	1477	71 %	65	19	29 %
SUI	4	2031	1554	77 %	59	14	24 %
NGA	4	1880	1386	74 %	87	14	16 %
MEX	4	1860	1393	75 %	60	14	23 %
ITA	3	1859	1580	85 %	36	6	17 %
BIH	3	1766	1412	80 %	72	19	26 %
ALG	4	1763	1194	68 %	67	13	19 %
GRE	4	1719	1288	75 %	118	19	16 %

Now our model is ready enough to be used in the reporting section

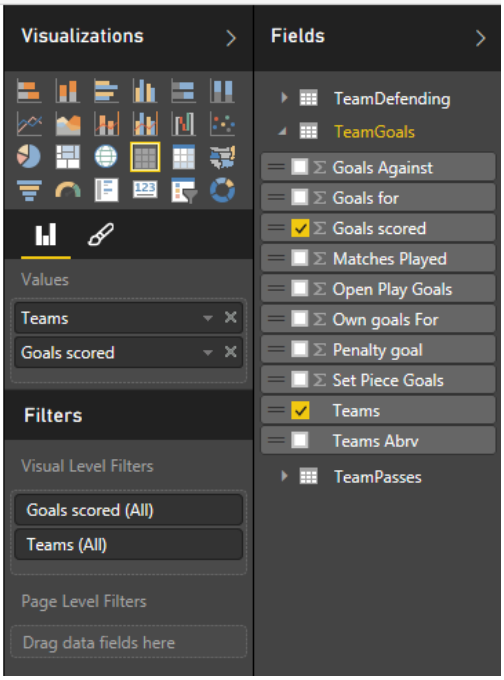
Visualization

Now let's make some visualization with the Report tab based on the data in the model we've built so far.

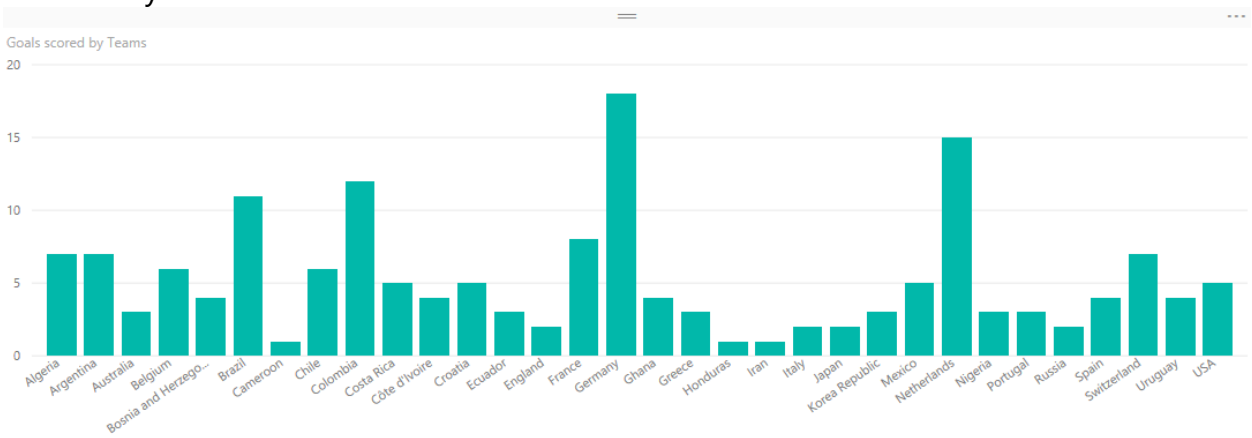
Goals Scored Column Chart

From the Fields section under TeamGoals, drag and drop teams into the main pane, then drag and drop Goals Scored on it. you will have a table showing values as below

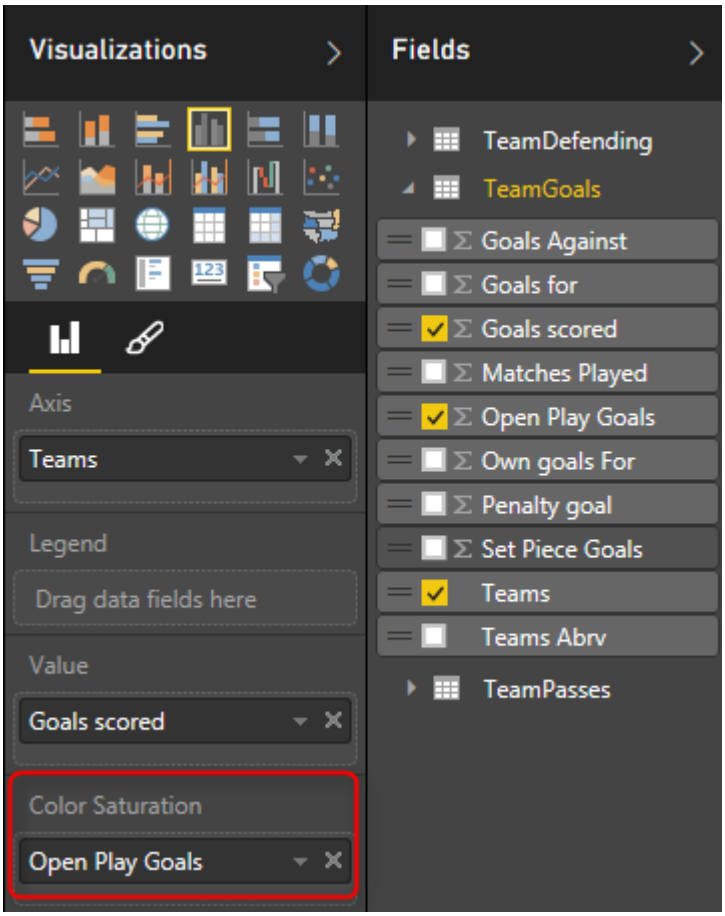
Teams	Goals scored
Algeria	7
Argentina	7
Australia	3
Belgium	6
Bosnia and Herzegovina	4
Brazil	11
Cameroon	1
Chile	6
Colombia	12
Costa Rica	5
Côte d'Ivoire	4
Croatia	5
Ecuador	3
England	2
France	8
Total	166



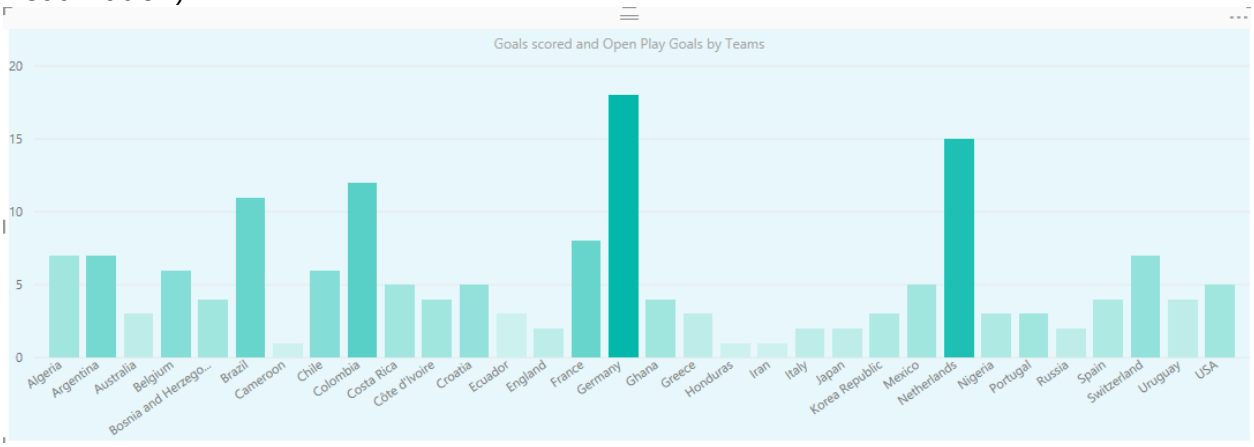
Now from Visualizations click on Column chart, and you will see the result set in the chart easily.



Let's highlight teams that scored their goals in open play, rather than a penalty or own goals. We can achieve that with the fantastic new feature of Color Saturation in the Power BI as mentioned below

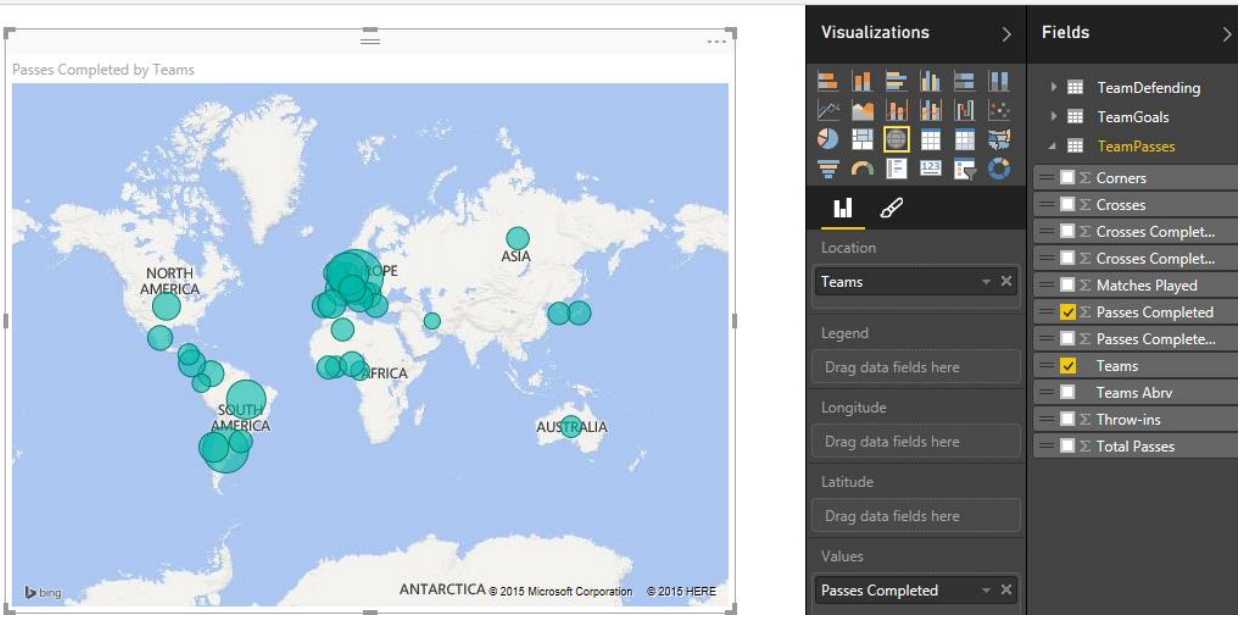


The result set also can be modified with some color them for background and moving the title to the center (all of these can be achieved in the format section of the visualization)

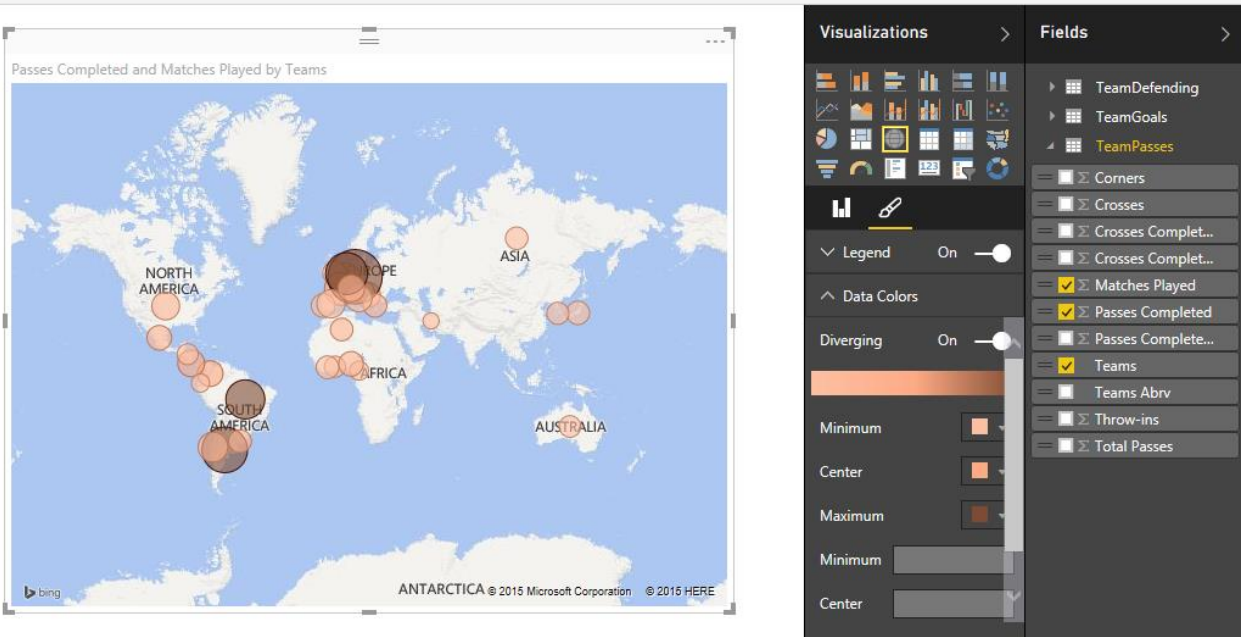


Map

Double click on the map from the visualization section. Then from TeamPasses Drag and drop Teams into the Location section of the map setting. Bring Passes Completed into the values section

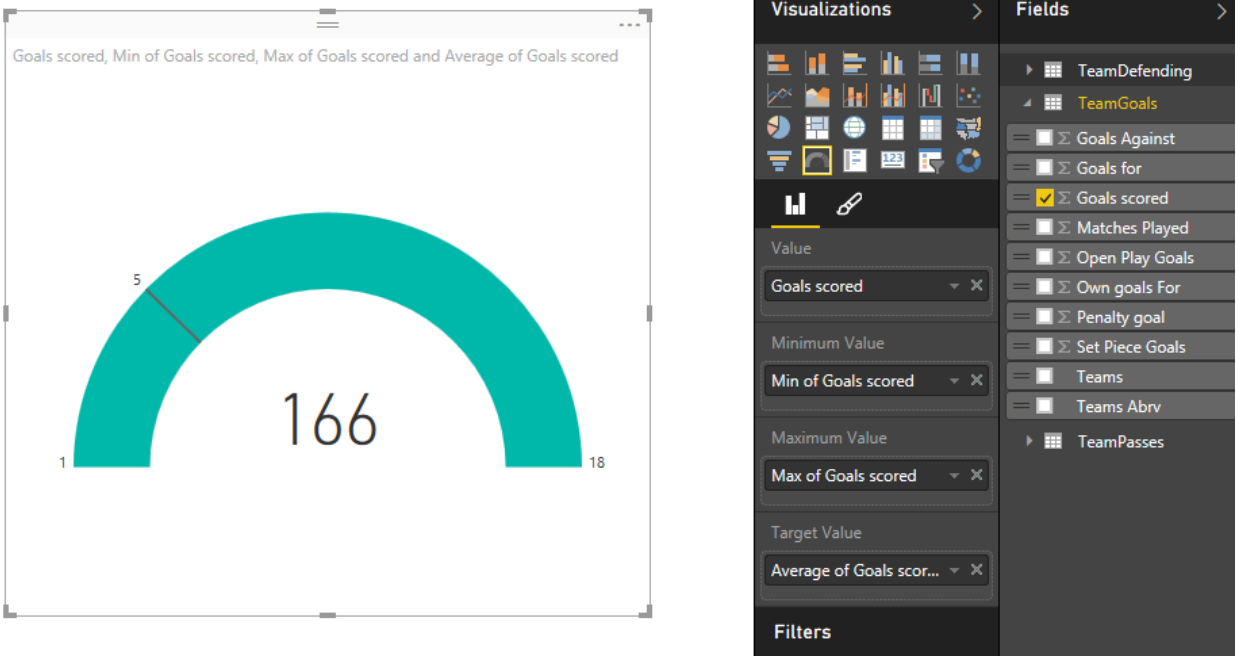


Let's change the color, and saturate the color by a number of matches each team played.



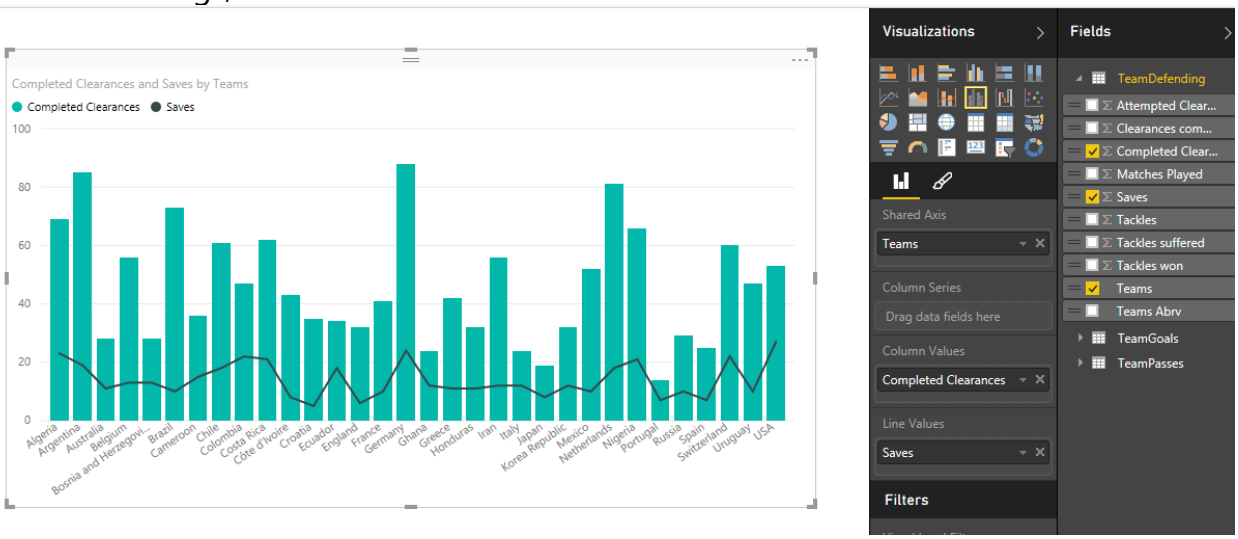
Gauge

Now let's create a gauge for goals scored. Set the value as Goals Scored. Minimum value as Min of Goals Scored, Maximum value as Max of Goals scored, and the Target as Average of Goals Scored.

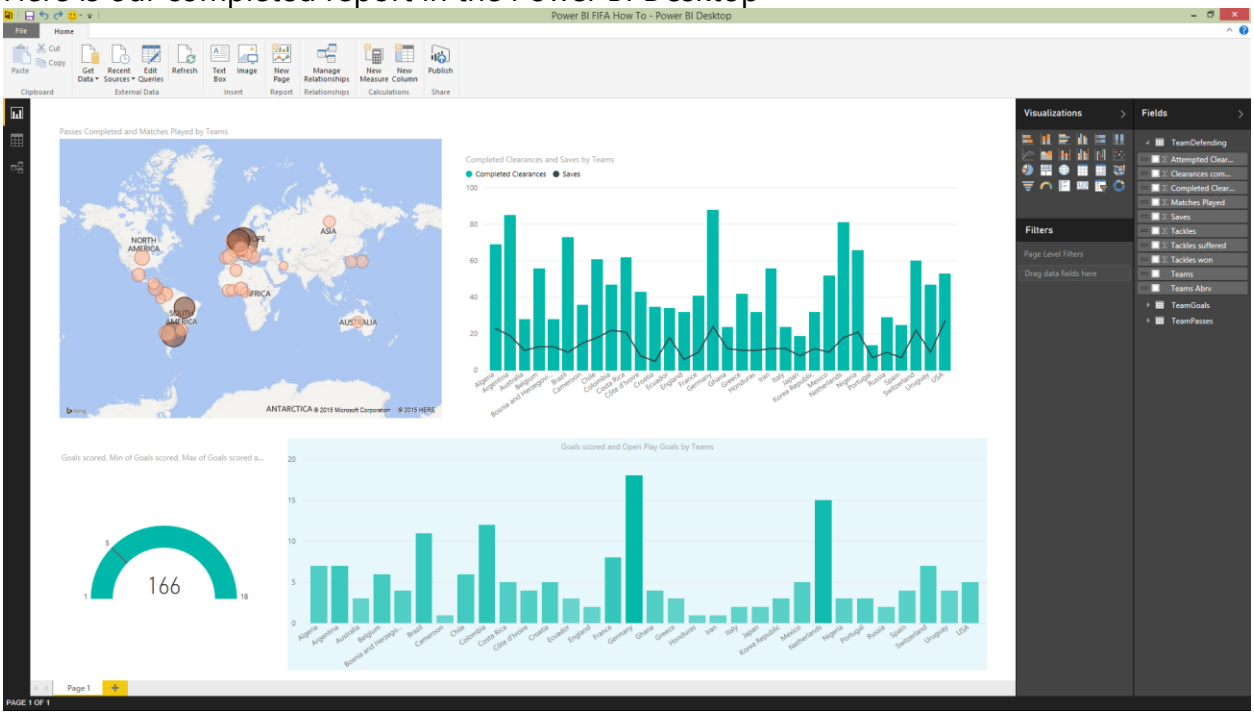


Clustered Column and Line Chart

As the last visualization item in this report, I would like to show a column chart combined with a line chart. The column values show Completed Clearance from TeamDefendings, and the line value shows the number of Saves. Here is the chart



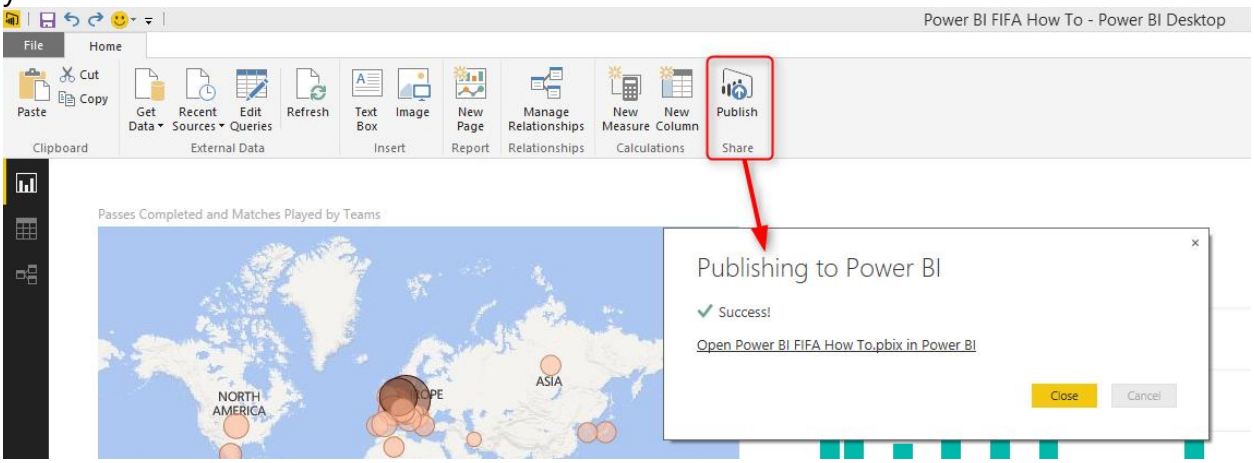
Here is our completed report in the Power BI Desktop



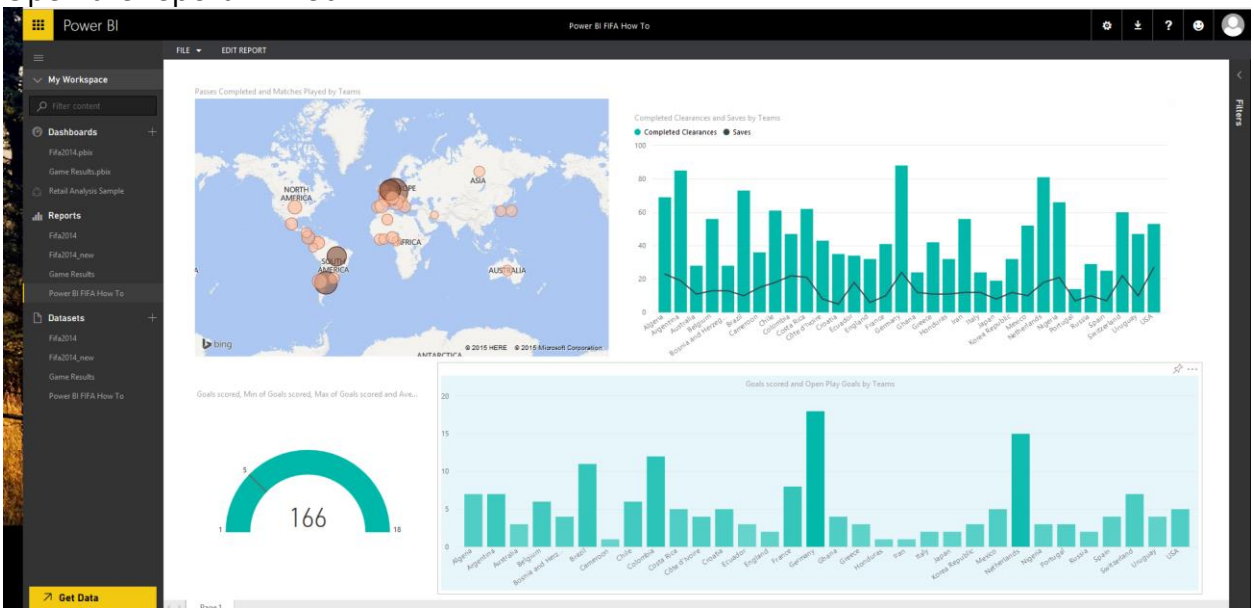
Publish

Now let's publish this report into Power BI website. For this section you have to create an account in Power BI website, if you don't have it, create it, it is free!

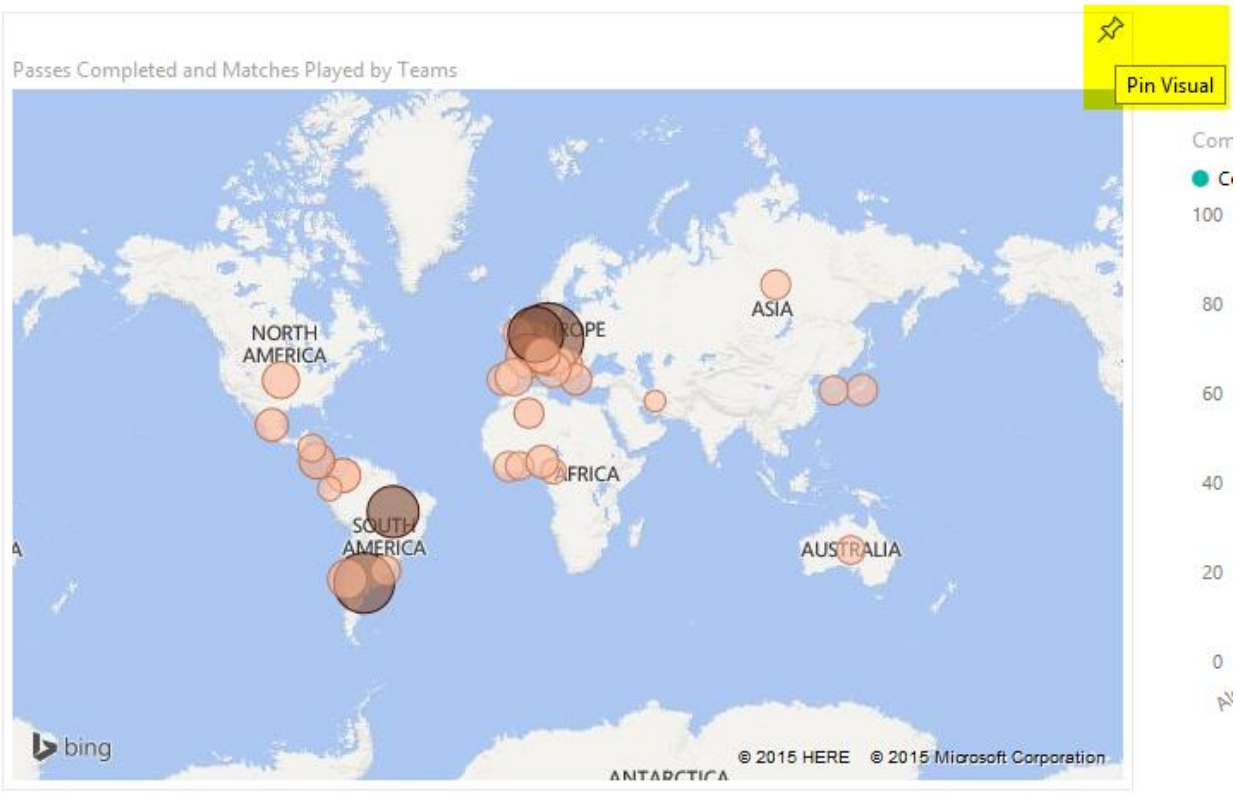
Click on Publish option in the menu above, and your report simply will be uploaded into your Power BI account.



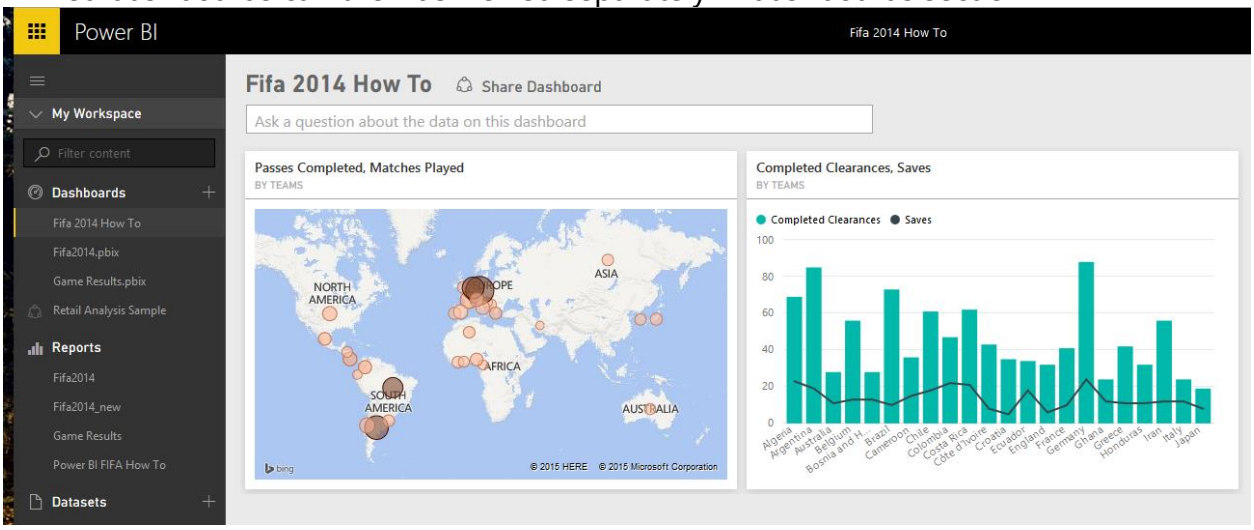
Open the report in Web



You can even pin some visualization items as dashboards as illustrated below

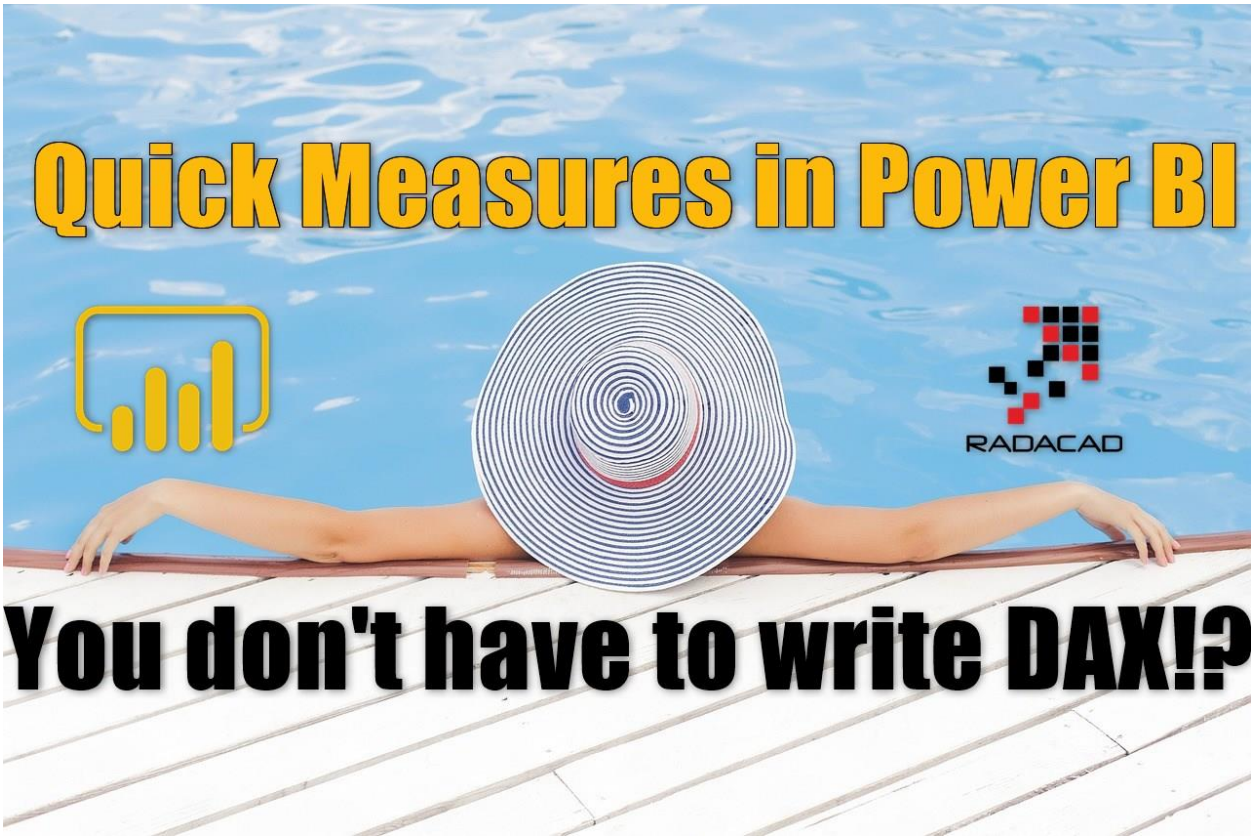


Pinned dashboards can then be viewed separately in dashboards section



Quick Measures in Power BI: You don’t have to write DAX!?

Published Date: November 22, 2018



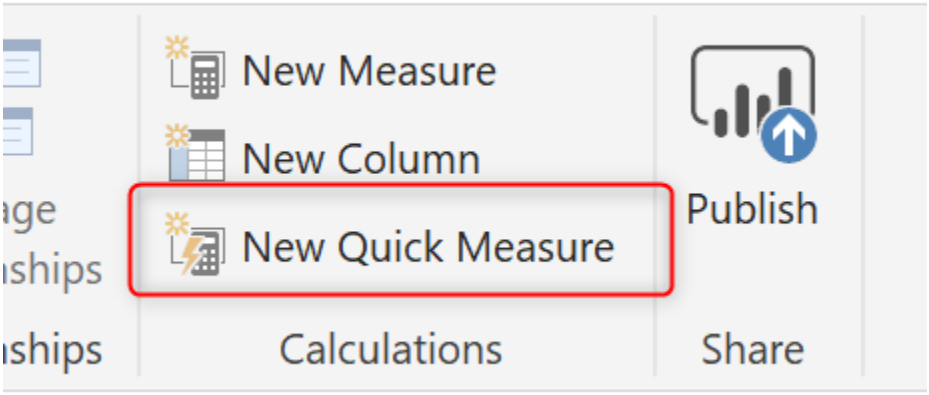
You read it right; you don’t have to write DAX. If you are calculating a year to date, running total, or every rolling average, you may be able to do it without writing a single line of DAX code. How is this possible? With Quick Measures. Quick measures are an easy way to create DAX measures. This functionality is easy but still not very known or used by many users of Power BI. I believe this is a good starting point for business users to get their way into analytics powers of Power BI. If you want to learn more about Power BI, read Power BI book, from Rookie to Rock Star.

What is Quick Measure?

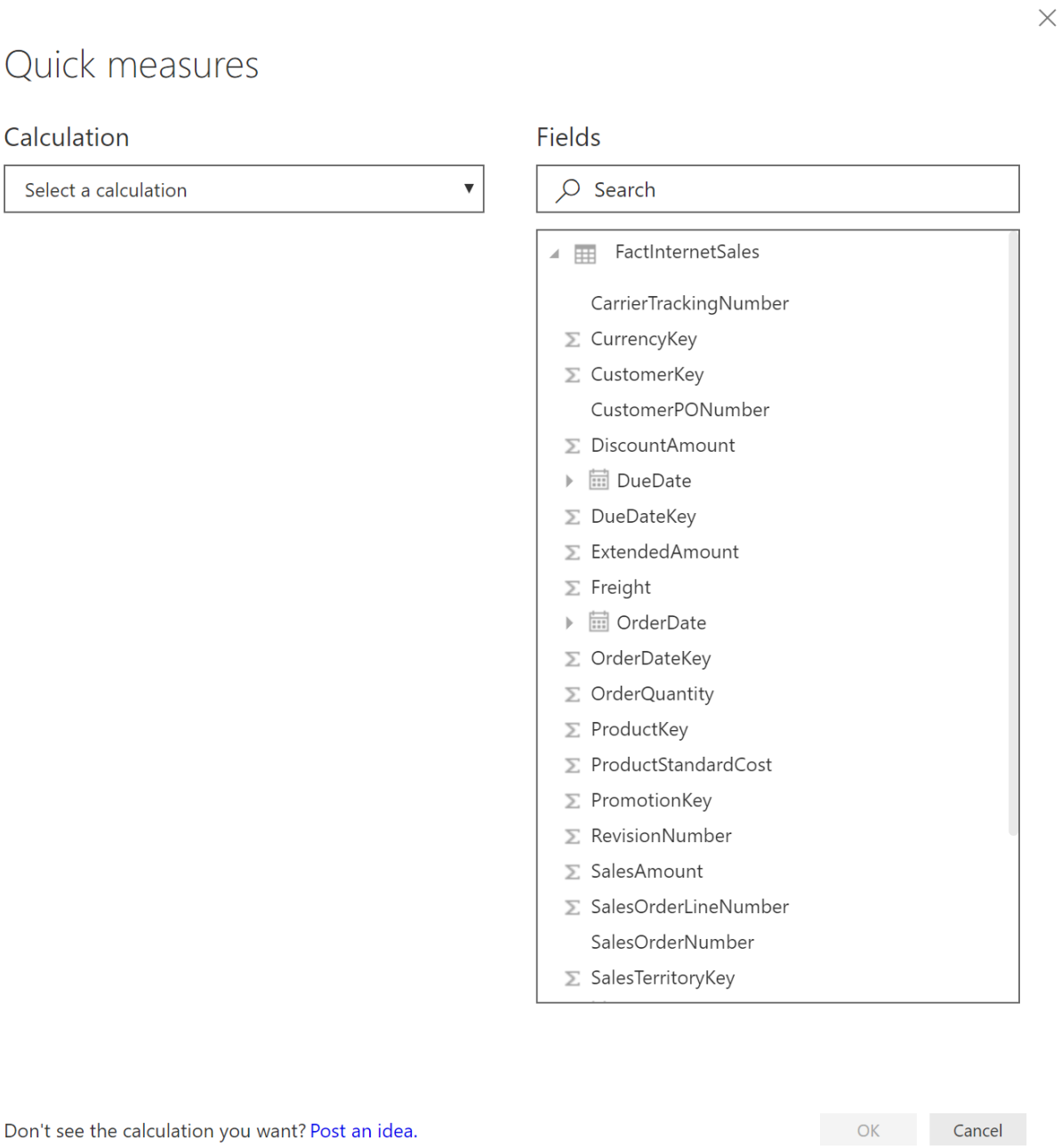
A Quick Measure is a quick way to create a DAX measure! That is why it is called so! This is I believe one of the best scenarios of naming a feature; it is self-explanatory. Do you need to have a special setup to use quick measures? No. let’s see how it works.

Let's start with a Year to Date example. You want to create a year to date calculation, but don't know how? Here is a quick measure built for you;

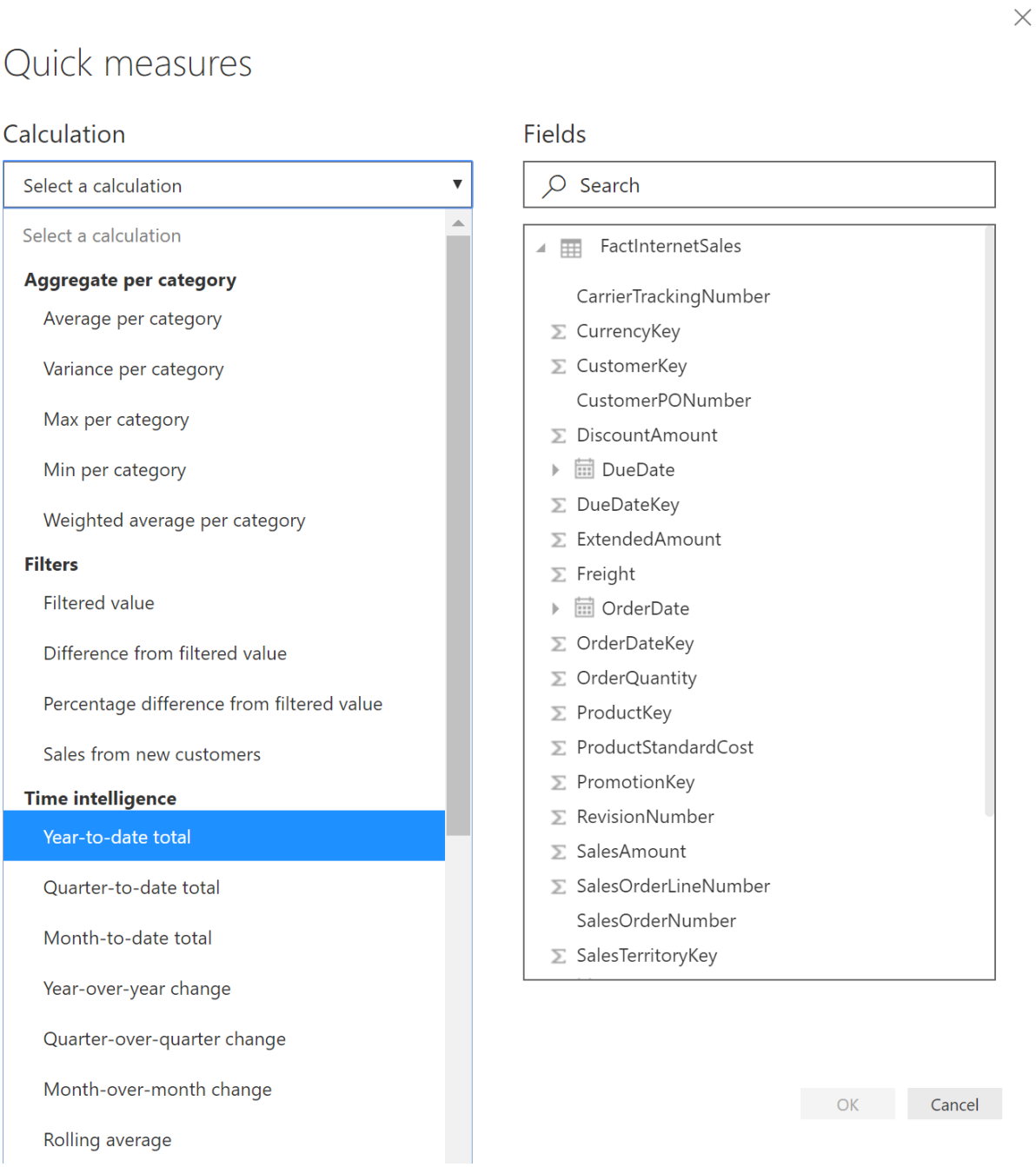
In the home tab of Power BI Desktop, select New Quick Measure.



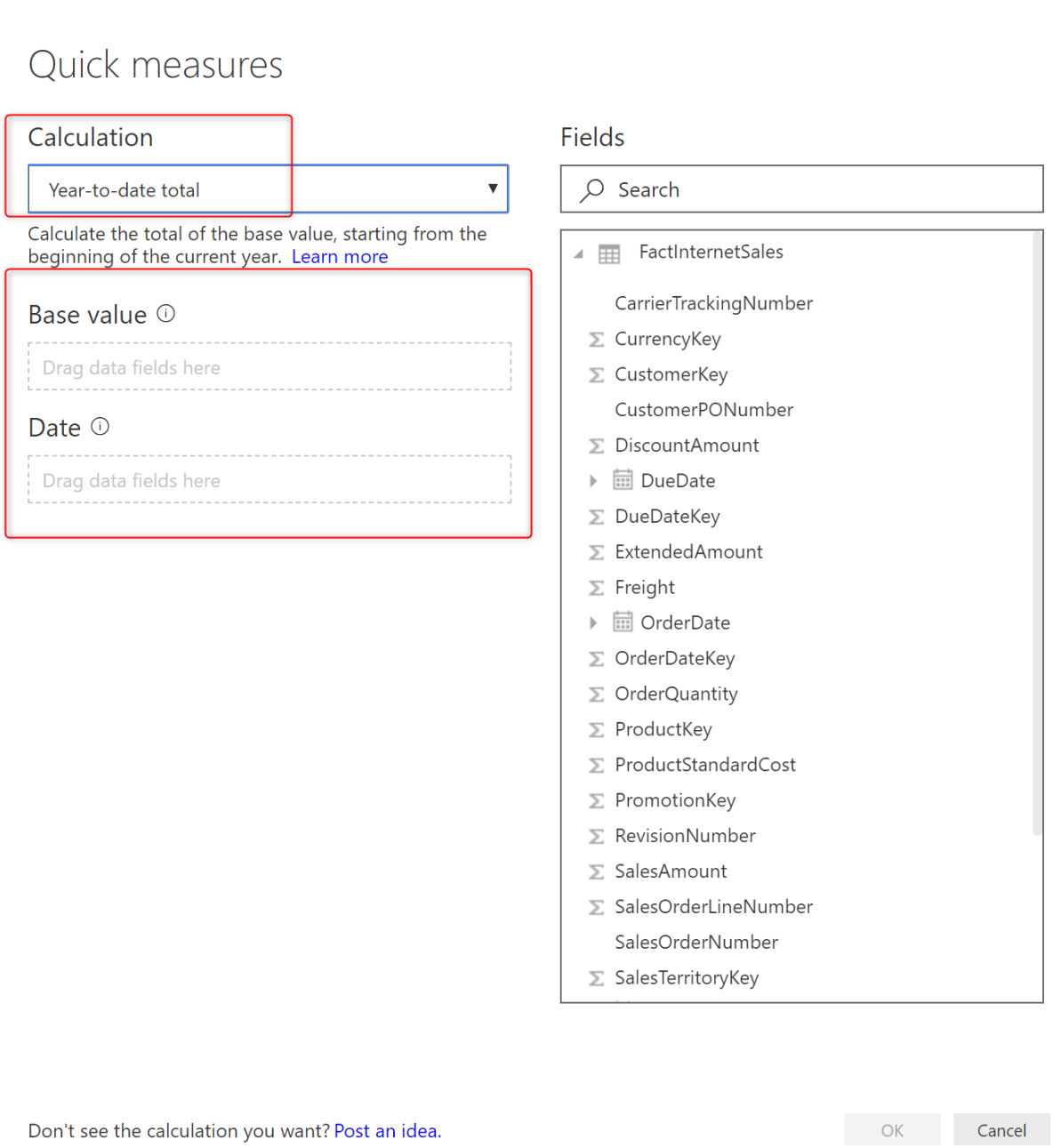
You'll see the Quick Measure dialog which asks for the calculation and shows the list of tables and fields in the right-hand side.



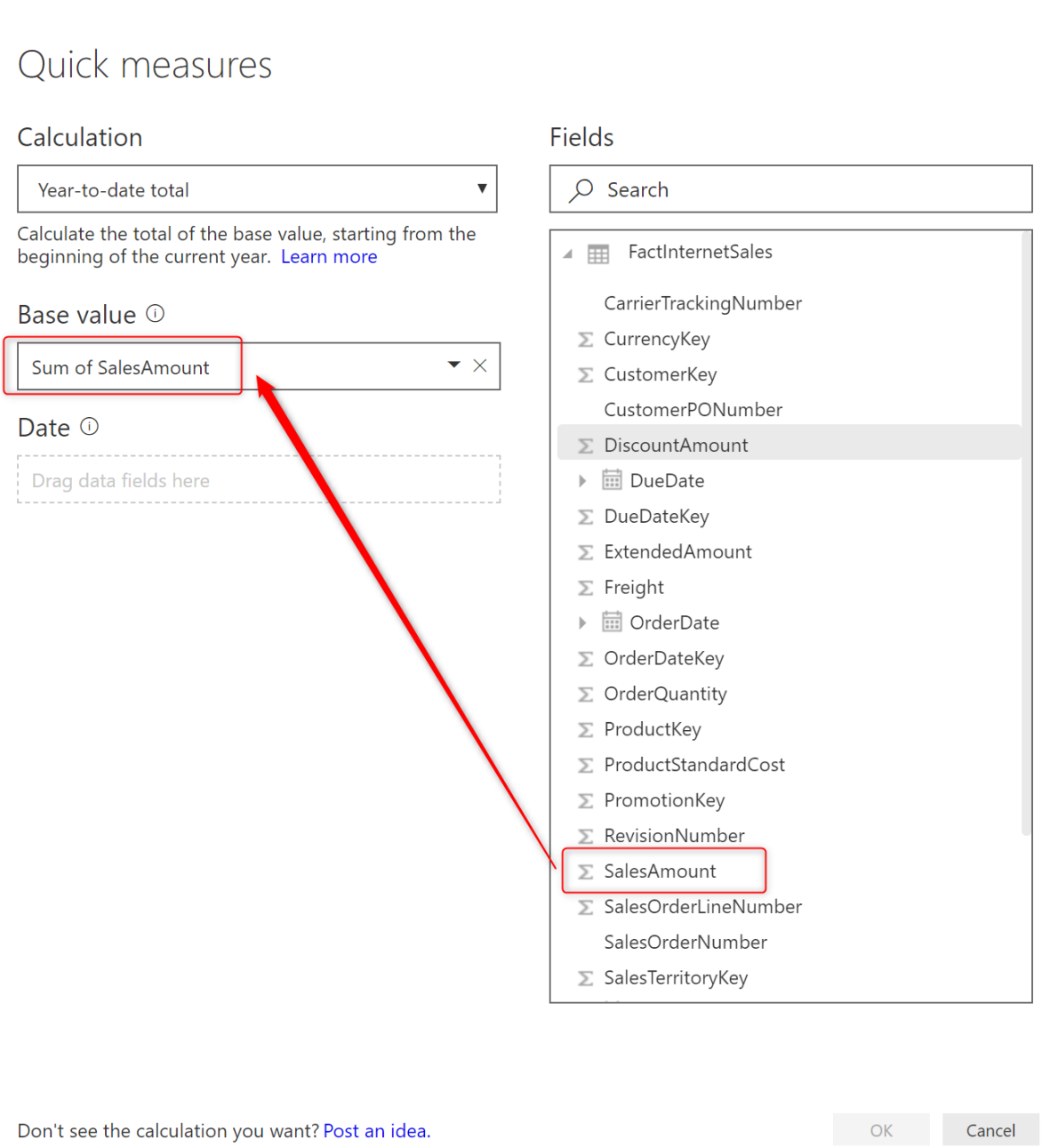
Select year-to-date total from the list of calculations.



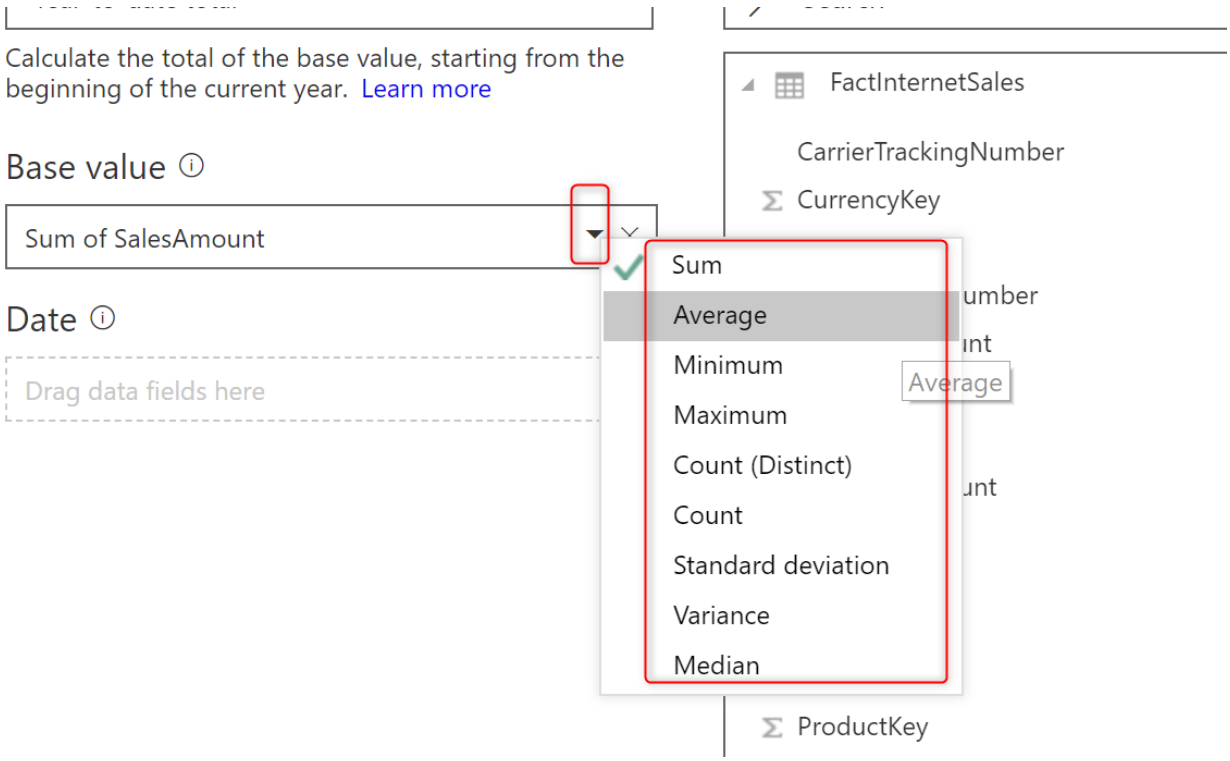
For every calculation, there will be input parameters to set. For year-to-date, the input parameters are Date field and expression as Base Value:



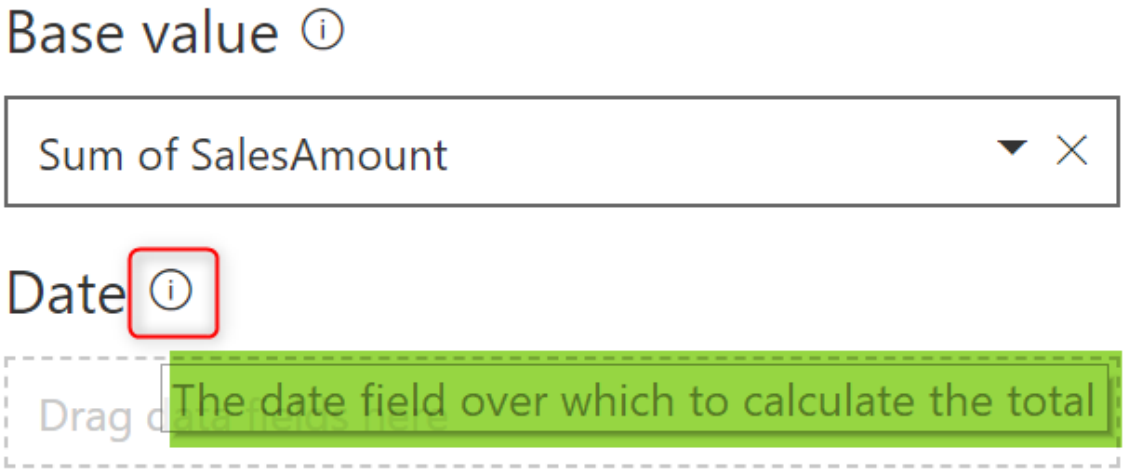
Drag and drop SalesAmount from the list of fields in the right-hand side to the Base Value section. You'll see it comes as Sum of SalesAmount (You can change it to other aggregations if you want to);



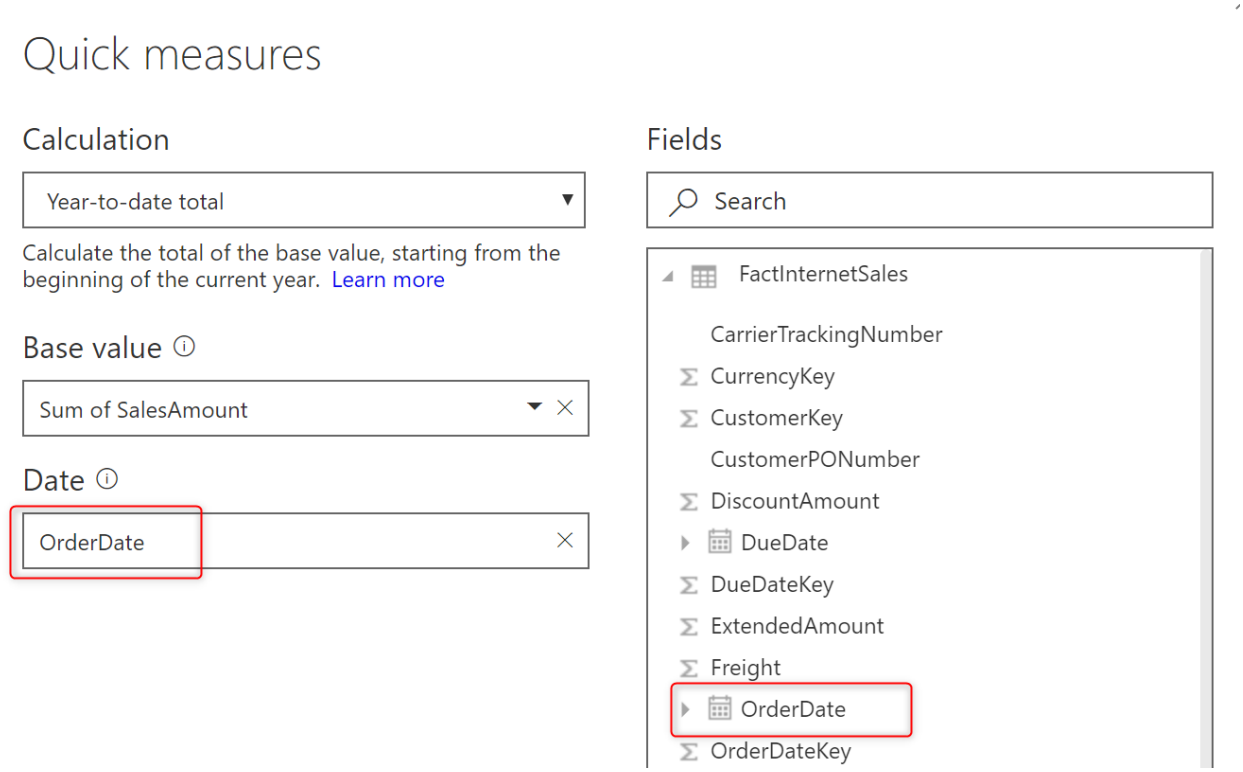
If you want to change the aggregation, you can do it this way;



For this example, I'll keep the aggregation as Sum. The Date field is the next input parameter. Even if you don't know what those parameters are, you can hover your mouse on the information button beside the parameter name and see a description about it.



For the Date field, I pass the OrderDate field from FactInternetSales. Please note that for this example, I'm using the default/built-in Date dimension in Power BI. If you want to learn what the default Date dimension in Power BI is, [read my blog post here](#).



Now click on OK. Your DAX expression is ready!

```
SalesAmount YTD =
IF(
ISFILTERED('FactInternetSales'[OrderDate]),
ERROR("Time intelligence quick measures can only be grouped or filtered by the Power BI-provided date hierarchy or primary date column."),
TOTALYTD(
SUM('FactInternetSales'[SalesAmount]),
'FactInternetSales'[OrderDate].[Date]
)
)
```

The whole expression created by the Quick measure wizard. You did not write even a single line of code for it. Let's now use it in a visual. For this visual, I used OrderDate, SalesAmount, and the new quick measure created named SalesAmount YTD.

```
SalesAmount YTD =
IF(
    ISFILTERED('FactInternetSales'[OrderDate]),
    ERROR("Time intelligence quick measures can only be grouped or filtered by the Power BI-provided date hierarchy or primary date column."),
    TOTALYTD(
        SUM('FactInternetSales'[SalesAmount]),
        'FactInternetSales'[OrderDate].[Date]
    )
)
```

Year	Quarter	Month	Day	SalesAmount	SalesAmount YTD
2005	Qtr 3	July	1	14,477.34	14,477.34
2005	Qtr 3	July	2	13,931.52	28,408.86
2005	Qtr 3	July	3	15,012.18	43,421.04
2005	Qtr 3	July	4	7,156.54	50,577.58
2005	Qtr 3	July	5	15,012.18	65,589.75
2005	Qtr 3	July	6	14,313.08	79,902.83
2005	Qtr 3	July	7	7,855.64	87,758.47
2005	Qtr 3	July	8	7,855.64	95,614.11
2005	Qtr 3	July	9	20,909.78	116,523.89
2005	Qtr 3	July	10	10,556.53	127,080.42
Total				**29,358,677.22**	**9,770,899.74**

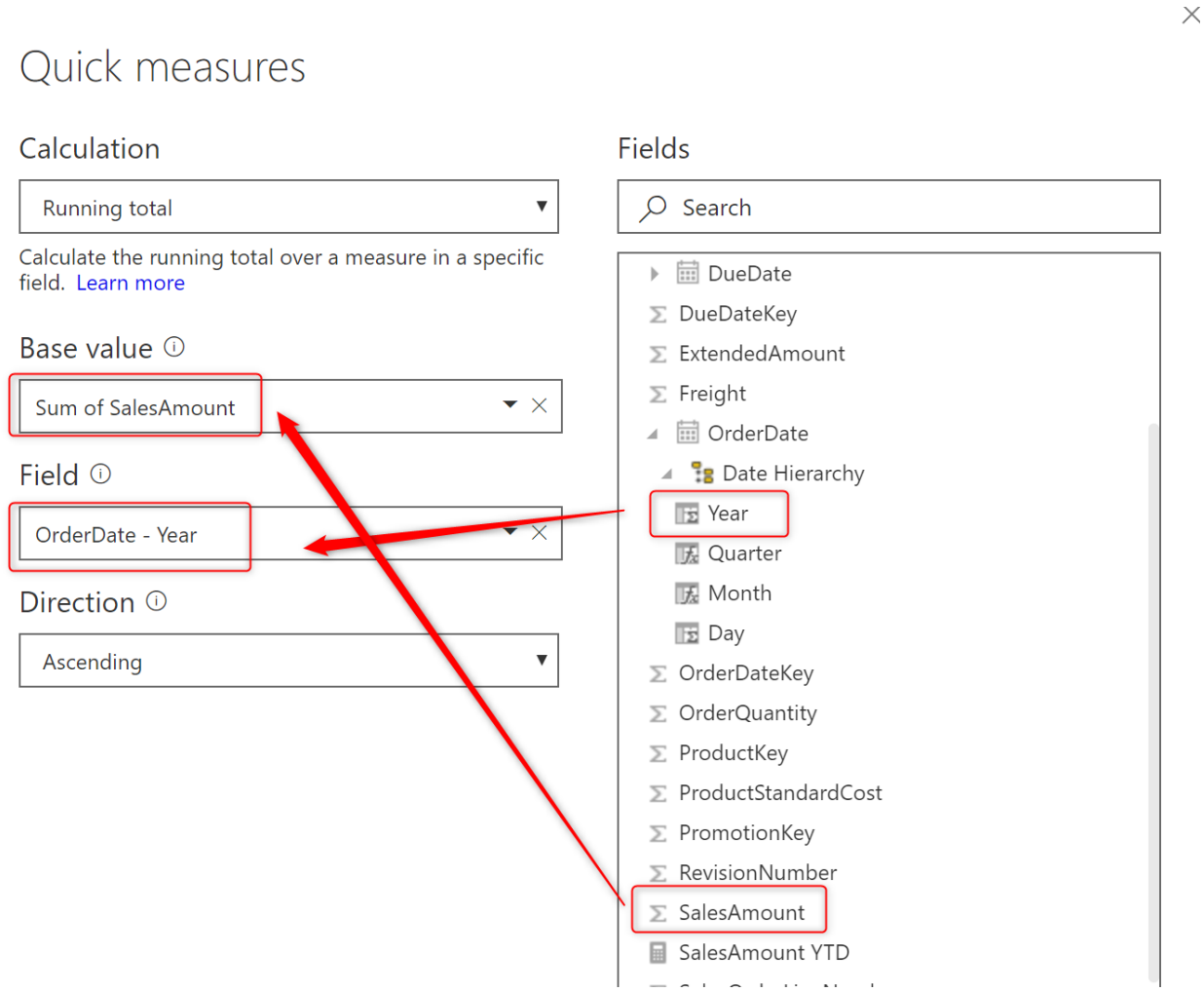
You can see in the above screenshot how beautifully this measure works with no need to write DAX statements.

> Quick measures are pre-defined DAX templates for the most common calculations. It is an easy way to start the analytics with DAX and Power BI with no knowledge of the DAX language.

I'm not saying that you do not need to learn DAX, you would need it eventually later in your path towards being a Power BI developer. There is no way around it; if you want to be good at doing analytics in Power BI, you need to learn DAX. All I'm saying is that if you are a business user, with zero experience in DAX, you can still start building a model and create calculations for it easily. It is obvious that for making custom changes or some complex scenarios, you do need to write DAX statement yourself, or modify the created quick measure.

Running Total

Let's say another example is to get the running total Sales through all years. Here is what the quick measure looks like for it;



Year part of the OrderDate will be used as the field to reset the running total if the value repeats, and the sum of SalesAmount would be our expression. Here is the result;

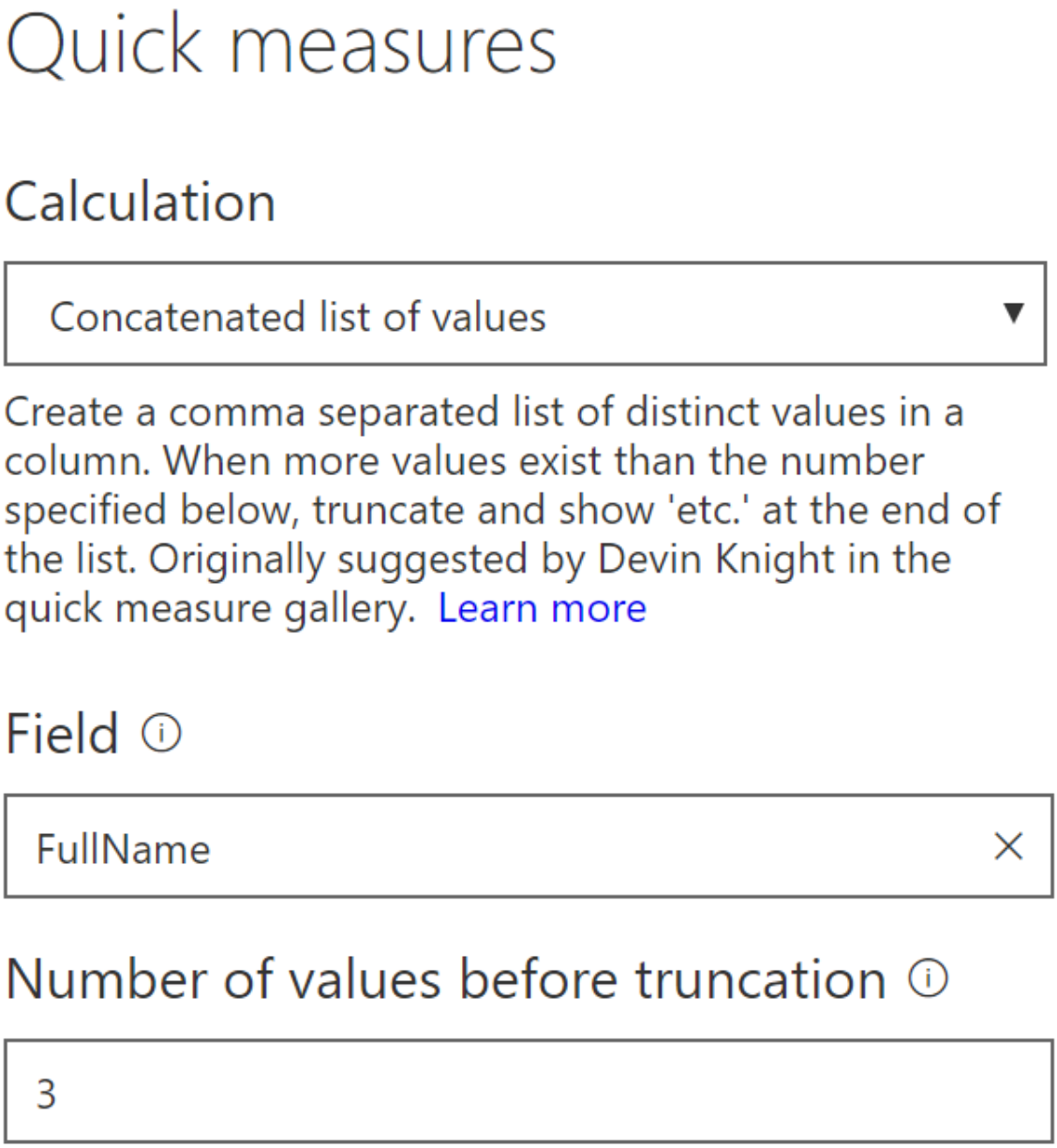
```
SalesAmount running total in Year =
CALCULATE(
    SUM('FactInternetSales'[SalesAmount]),
    FILTER(
        ALLSELECTED('FactInternetSales'[OrderDate].[Year]),
        ISONORAFTER('FactInternetSales'[OrderDate].[Year], MAX('FactInternetSales'[OrderDate].[Year]), DESC)
    )
)
```

Year	SalesAmount	SalesAmount running total in Year
2005	3,266,373.66	3,266,373.66
2006	6,530,343.53	9,796,717.18
2007	9,791,060.30	19,587,777.48
2008	9,770,899.74	29,358,677.22
Total	**29,358,677.22**	**29,358,677.22**

As you can see the quick measure simply provide the output needed.

Concatenated List of Values

Another example of quick measures is the Concatenated List of values, which can be helpful to see some details about a specific category. Here is measure configuration for that;

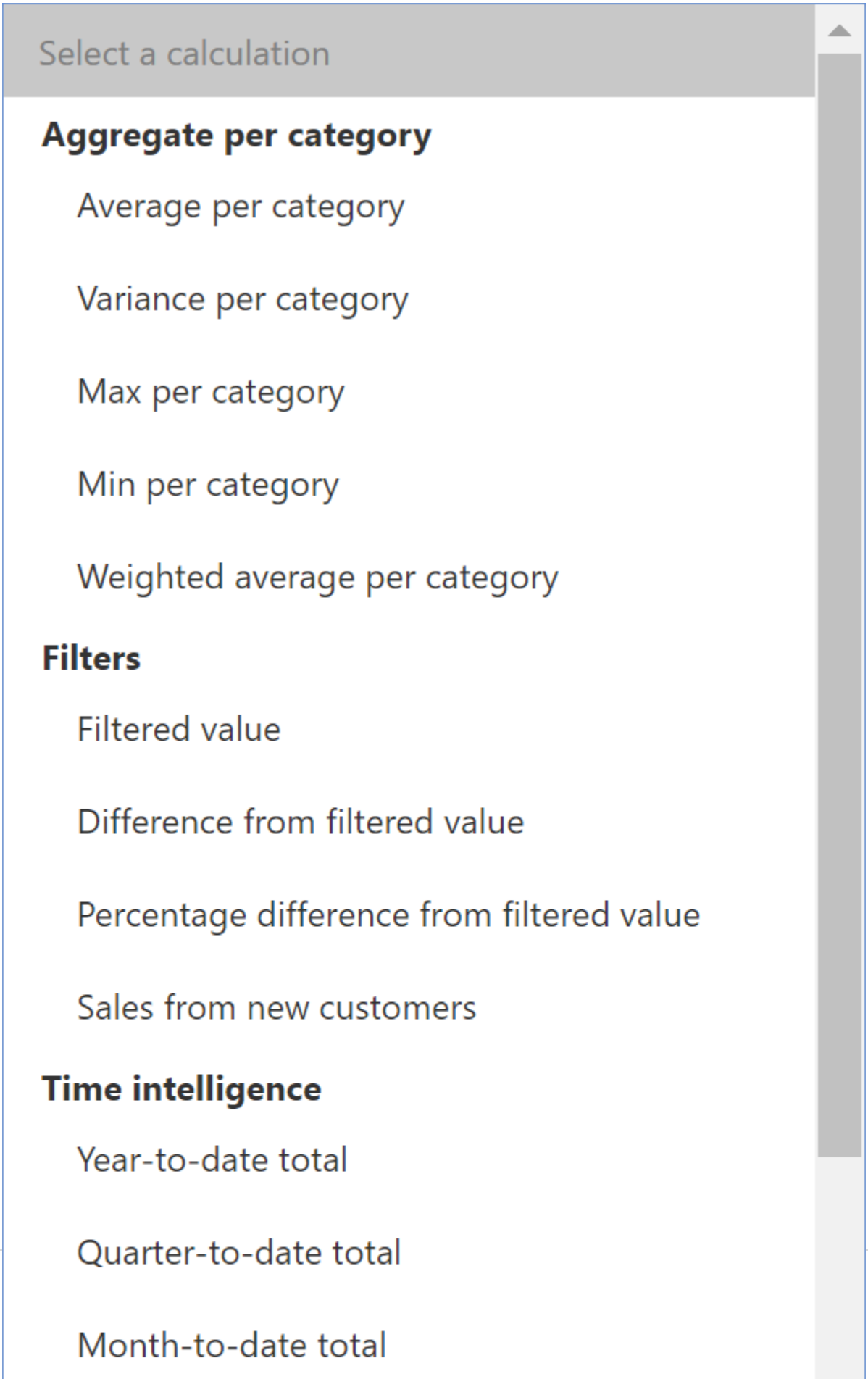


And the output, which shows 3 Full names from each category concatenated together;

EnglishEducation	Gender	List of FullName values
Bachelors	F	Abby Arthur, Abby Gonzalez, Abby Lopez, etc.
Bachelors	M	Aaron Bryant, Aaron Chen, Aaron Diaz, etc.
Graduate Degree	F	Abigail Coleman, Abigail Flores, Abigail Garcia, etc.
Graduate Degree	M	Aaron Campbell, Aaron Collins, Aaron Evans, etc.
High School	F	Abby Fernandez, Abby Kapoor, Abby Madan, etc.
High School	M	Aaron Alexander, Aaron Allen, Aaron Baker, etc.
Partial College	F	Abby Chandra, Abby Garcia, Abby Kovár, etc.
Partial College	M	Aaron Adams, Aaron Butler, Aaron Carter, etc.
Partial High School	F	Abby Mehta, Abigail Morgan, Abigail Morris, etc.
Partial High School	M	Aaron Green, Aaron Griffin, Aaron Jenkins, etc.

Gallery of Quick Measures

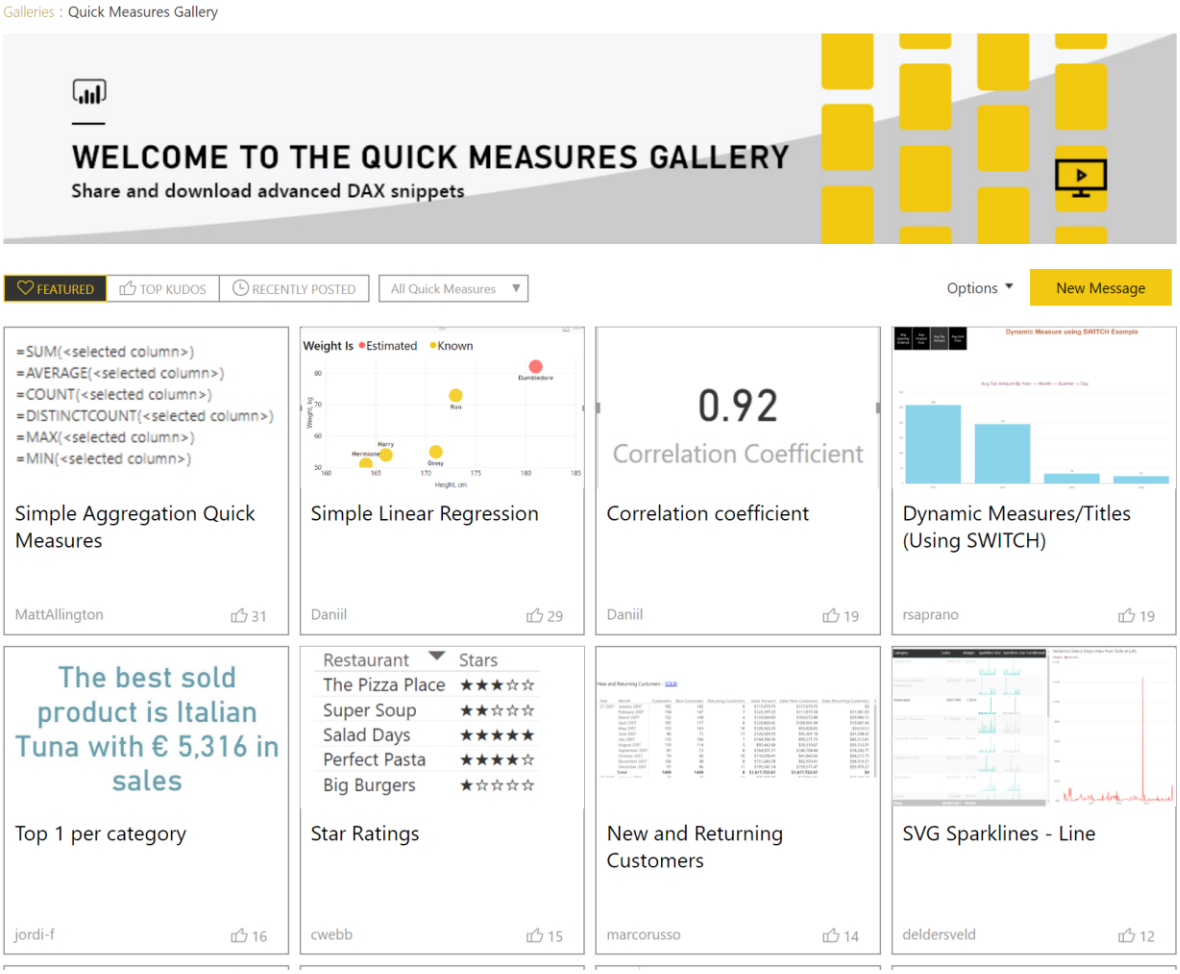
The existing list of quick measures is limited, here are some of the existing options;


Select a calculation
Aggregate per category
Average per category
Variance per category
Max per category
Min per category
Weighted average per category
Filters
Filtered value
Difference from filtered value
Percentage difference from filtered value
Sales from new customers
Time intelligence
Year-to-date total
Quarter-to-date total
Month-to-date total

However, the list is getting updated every month with the new version of Power BI Desktop. If you are interested in a quick measure that is not in the list, you can submit an idea about what you want:

Don't see the calculation you want? Post an idea.

There is also a [Gallery of quick measures in the Power BI community website] which can be a good source of extra quick measures. Usually, measures will be added from that gallery into the Quick Measures list of Power BI Desktop gradually.



Misconceptions

Quick Measures work with Default or Custom Date Dimension

There is a common misconception that says Quick measures only work with the custom date dimension, not with the default one. This is not true. As you see in above, it works perfectly with the default Date dimension, and it also works perfectly with a custom Date dimension too.

DAX is not Obsolete

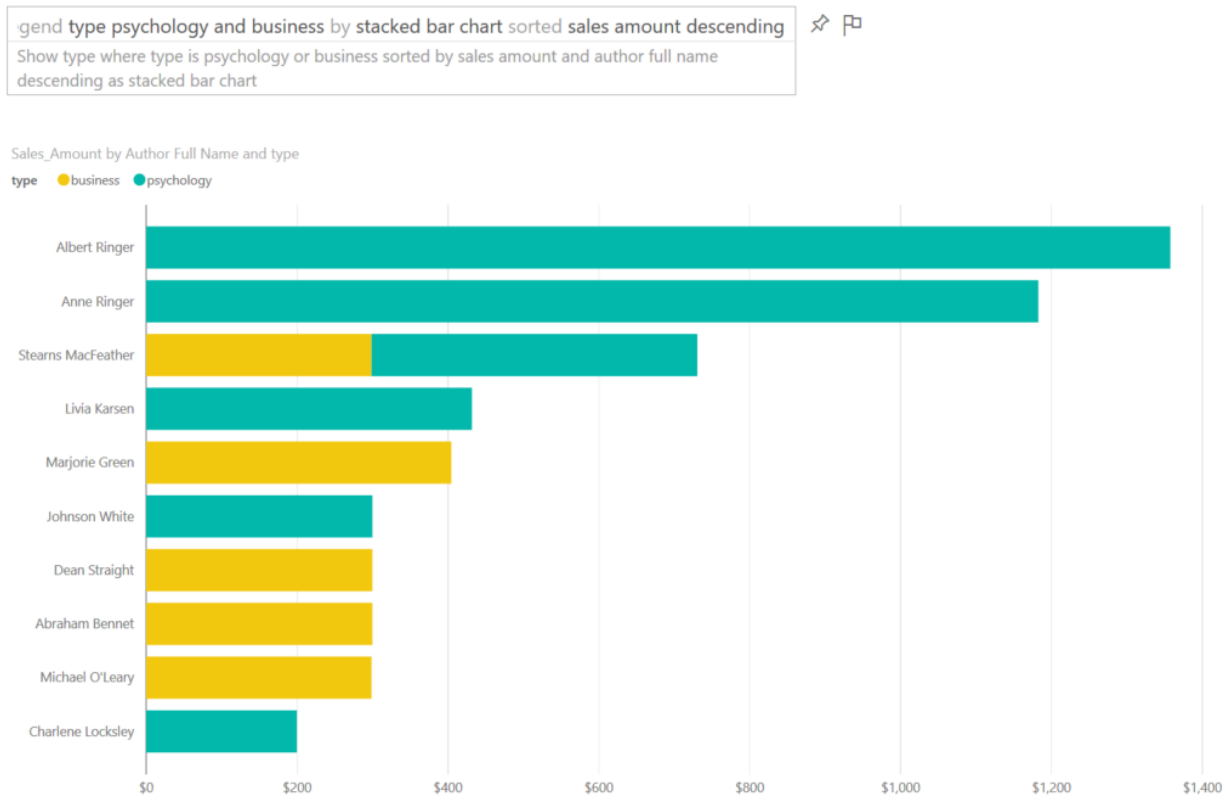
Another misconception is saying that now that we have a gallery of quick measures, then we don't need to learn DAX! That is not true, and would never be. You have to learn writing DAX statements because, for every business and requirement, there are always challenging part that might be different from the generic solution suggested in the quick measure. The quick measure can be a good start towards creating your analytics calculation, but it is not the final answer. You need to write DAX to get the maximum ultimate analytics power in the Power BI toolset.

Summary

Quick measures are a simple way of using pre-defined templates of common DAX calculations in Power BI. If you have a requirement that can be solved with that, you may not need to write DAX statement. Further, in the path, you will need DAX for your specific business case scenarios. However, the quick measures can be a good start for stepping towards that.

Power BI Q&A; How to Ask Questions?

Published Date: March 11, 2016



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 a response with visualization elements! In this post, I'll show you some of the intelligence questions that you can ask from Q&A to get a very useful response.

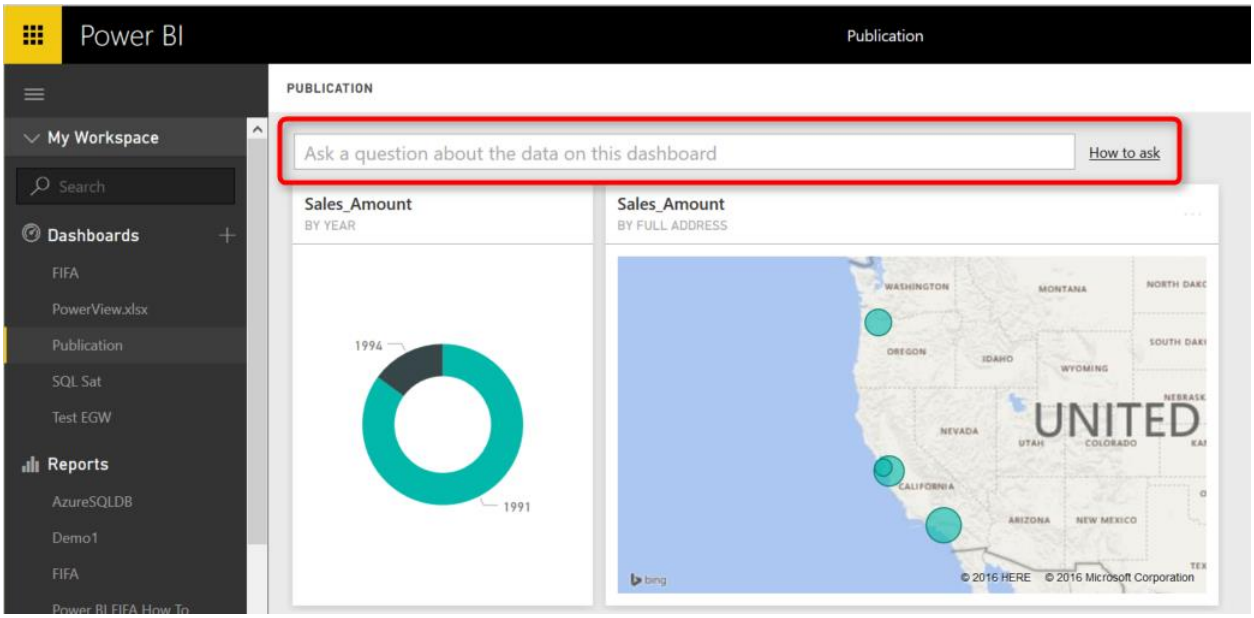
Prerequisite for Running Samples of this Post

If you want to run a sample of this post and see how they work 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 the 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 with the most basic question:

Explore a Dataset

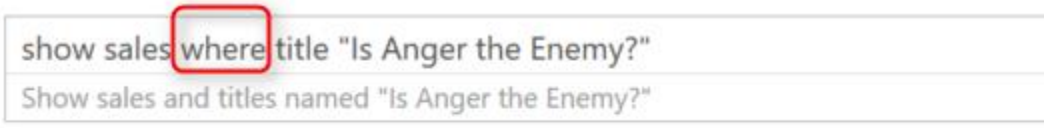
By typing the name of a dataset, you can get it in a table view. You can either type name of the table; such as "sales" or put a "show" keyword before it like "show sales"

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.

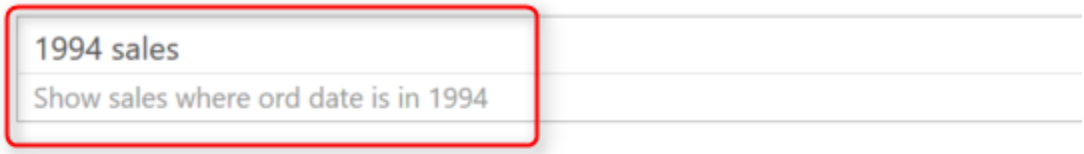
Filter Values

You can filter values with a WHERE keyword in your sentence. example below fetch all sales for the book titled “Is Anger Enemy?”



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 the year 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

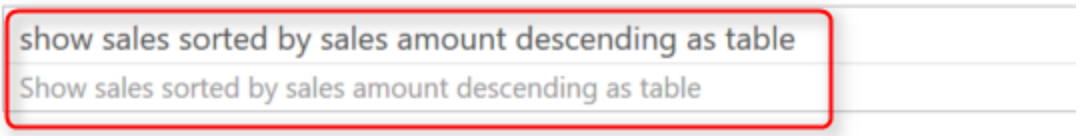
Explore Related Datasets

You can explore data from related datasets. For example to get titles and authors (which are in two separate datasets) you can ask:

title_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.



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	$99.950000000000003
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
6380	722a	13/09/1994 00:00:00	3	Net 60	PS2091	$32.850000000000001

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 the 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

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
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

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

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

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
7896	TQ456	12/12/1993 00:00:00	10	Net 60	MC2222	$199.900000000000001
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	$199.900000000000001

Filter Relative to Today's Date

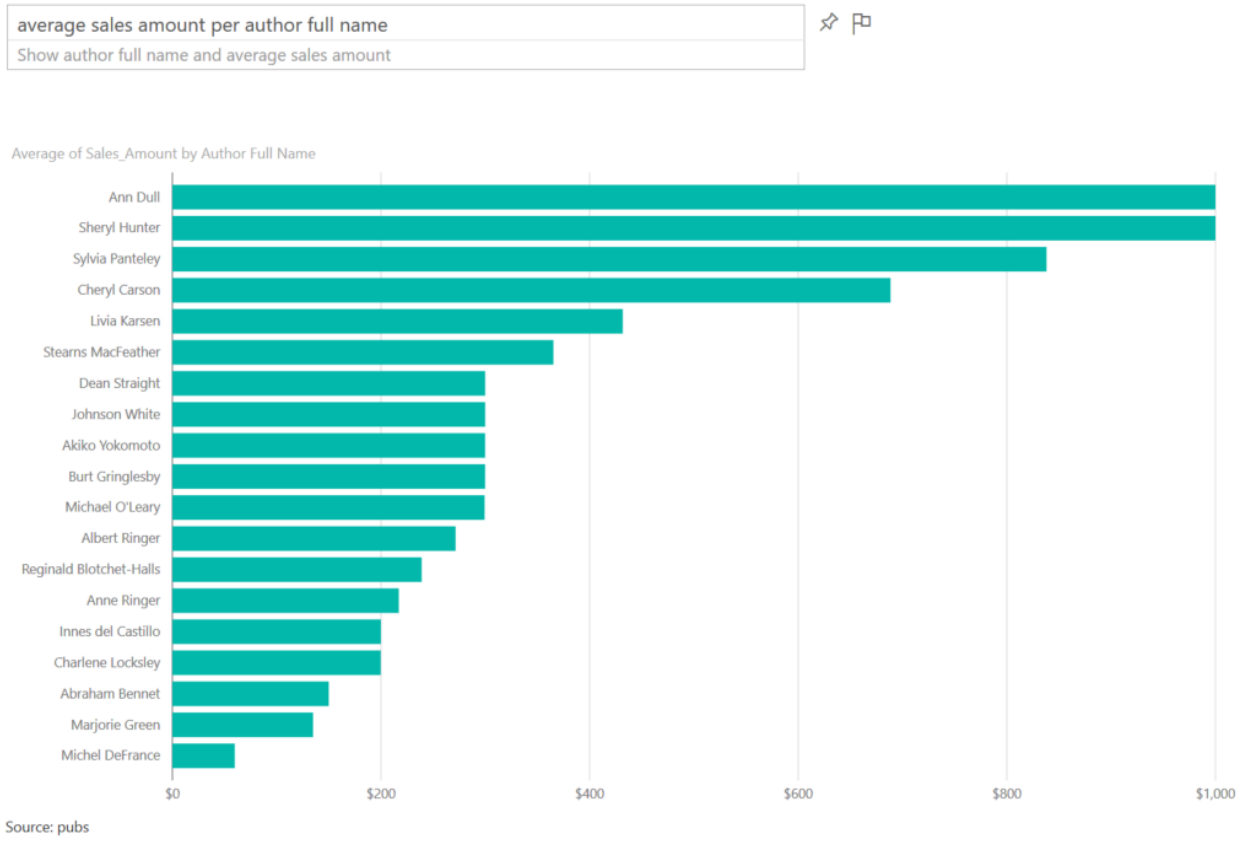
Fetch data with filters such as past or next periods. For example, fetch dataset information for the past ten years or so.

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

Aggregation

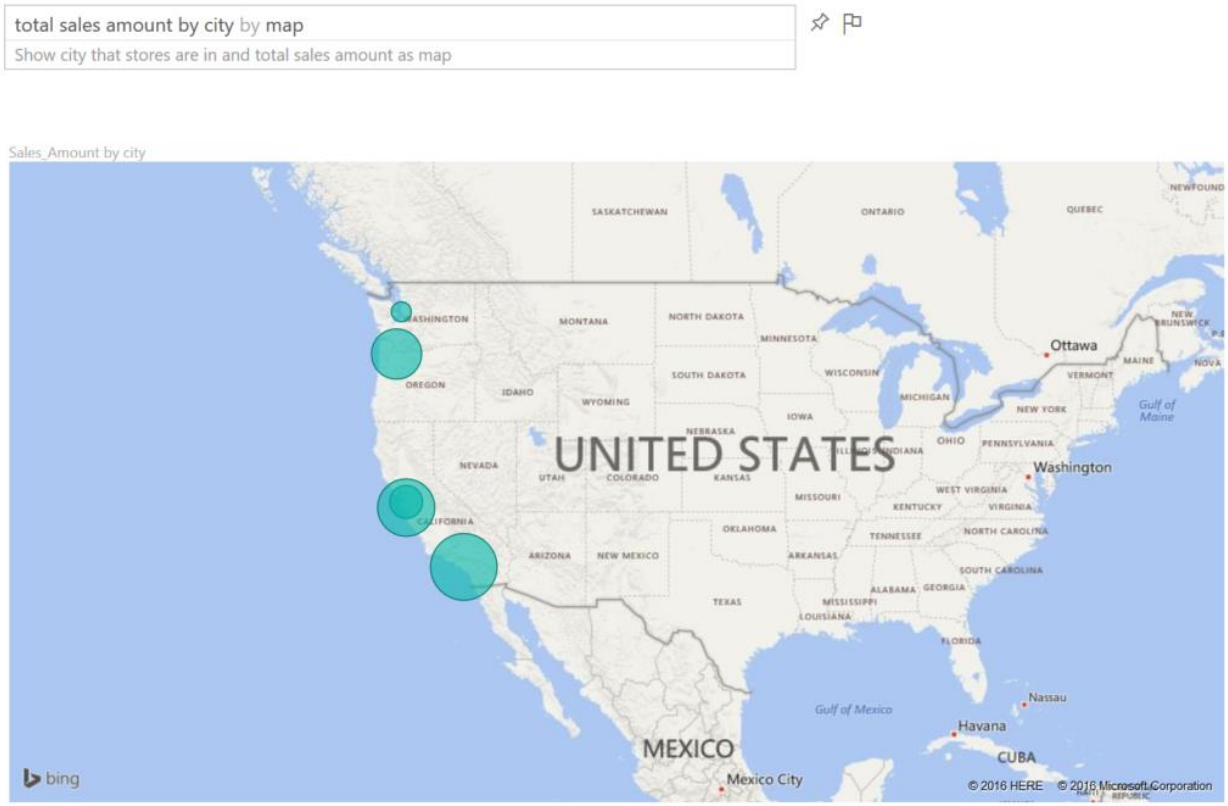
You can even do calculations in your question, how impressive is that. For example, ask the Total Sales amount or Average Sales Amount

You can then aggregate across attribute values. such as; average sales amount per author



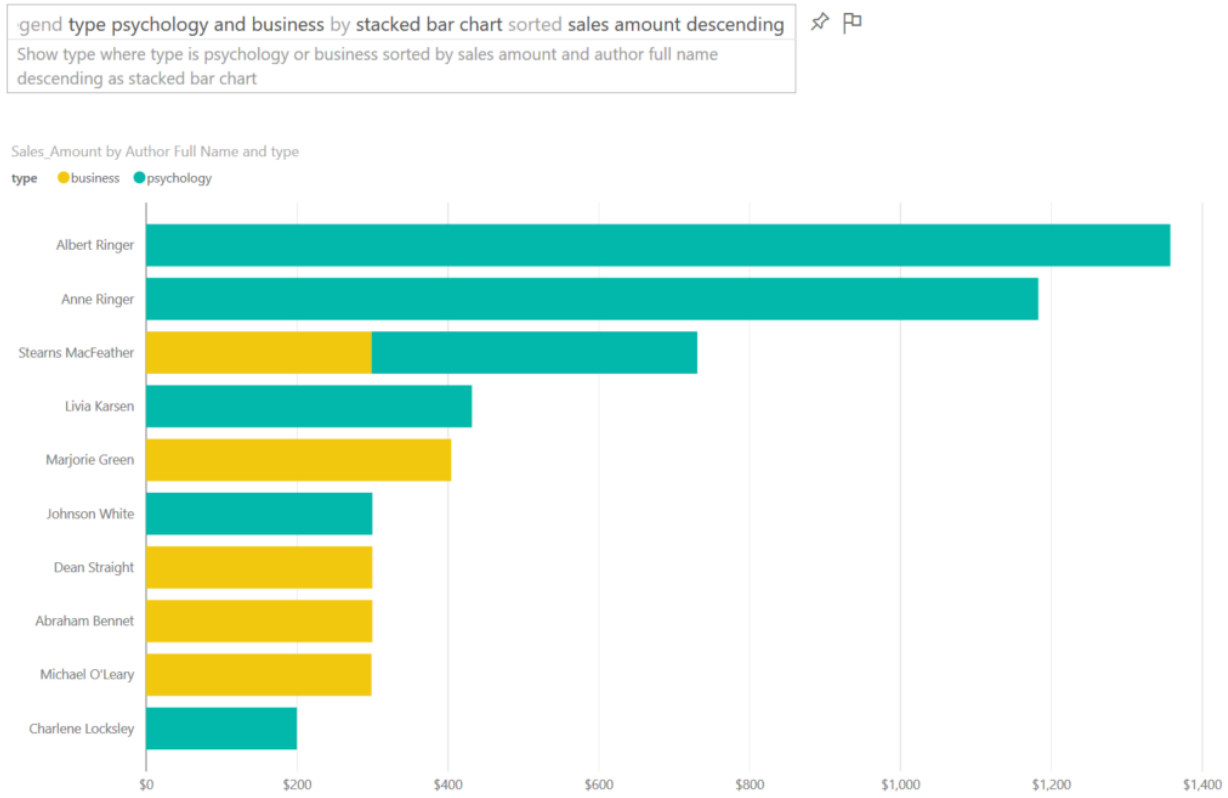
Visualization Customization

You can ask what type of visual you want to see the result set to be shown.



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:



You can pin every answer you want to your dashboard to save it for later.

Some other keywords help 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 your Power BI model in a way that responds best to question from Q&A. Tips in creating a relationship, naming columns, data types, etc.

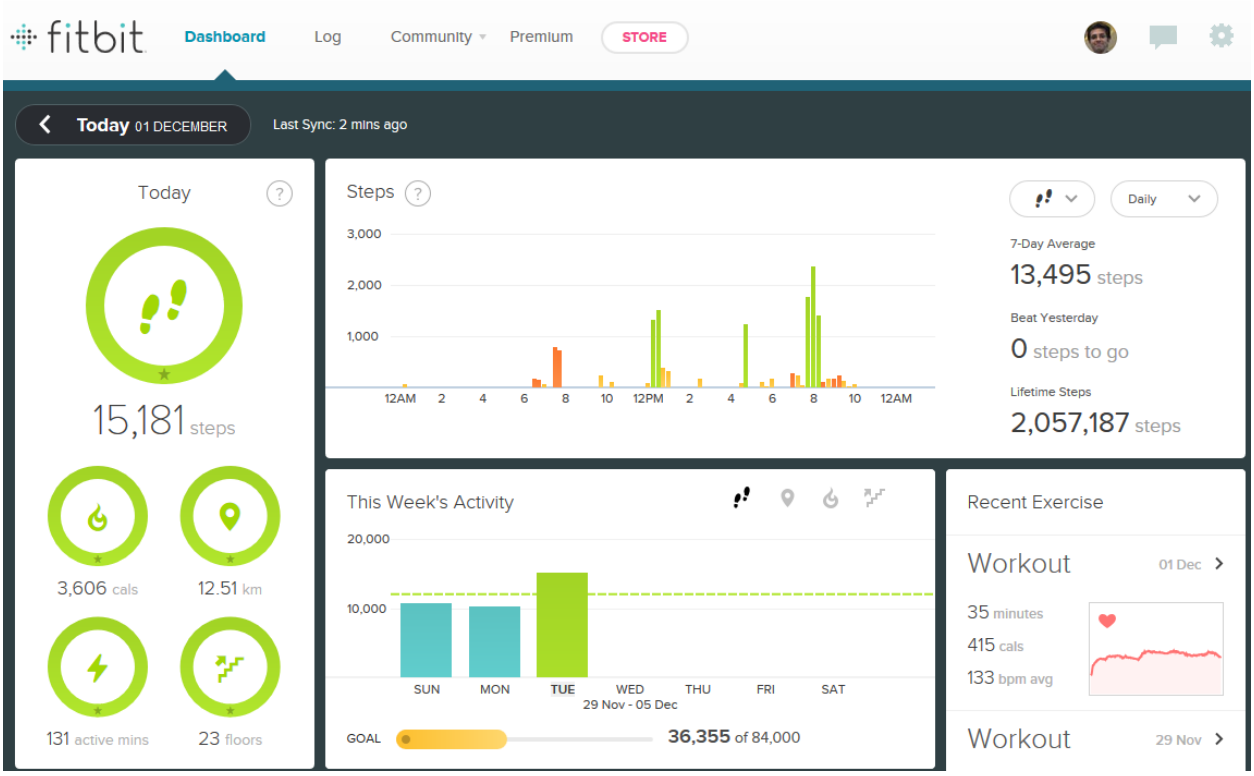
Your Turn

Now go to your Power BI dashboard and have fun with Q&A and bring some of the examples you've done. I'd love to see what you will find ☺

Be Fitbit BI Developer in Few Steps: the First Step Get Data from CSV

Published Date: December 1, 2015

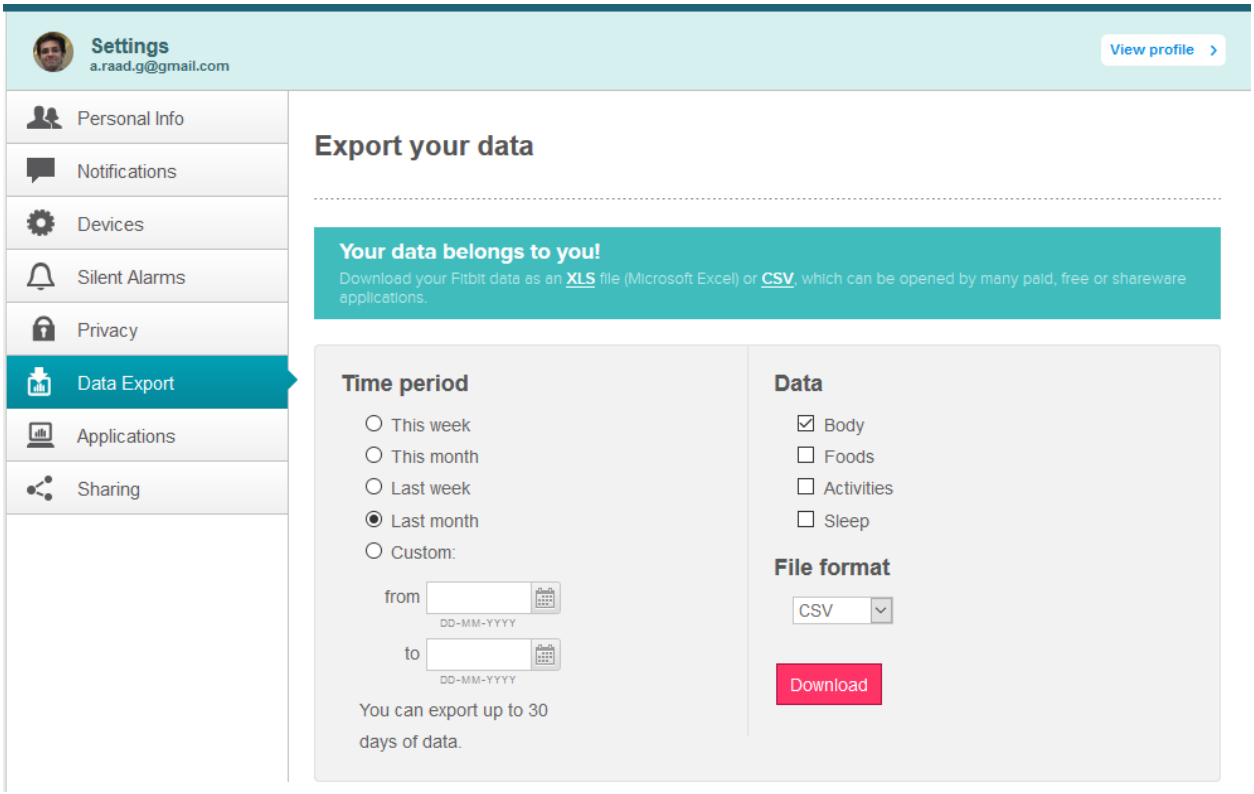
Power BI made things easy; it isn't hard to build dashboards and data analysis on almost any types of data sources. Previously in other sections of the online book; Power BI from Rookie to Rockstar I've mentioned a few data sources and ways of connecting to them. In this post, I like to show you how 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 a great demo. The reason is that you probably have seen your Fitbit dashboard in the website as below:



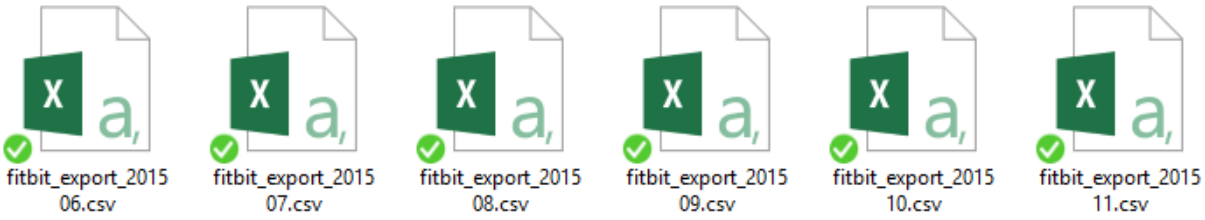
So for this post, we are going to build that dashboard (not all of that obviously, because we don't have the data required for all of that), but most of it with Power BI. You will see how easy and powerful is Power BI in this kind of scenarios, and you will see how you can be the BI Developer of Fitbit in a few steps of building this demo.

Before the 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. They are doing a great job, and most of this great job is hidden from our view, they are mashing up the data to build it in a way that can be visualized best. So I admire them because of their efforts. In this post, I want to show how easy it 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 months, you will end up with two CSV files.



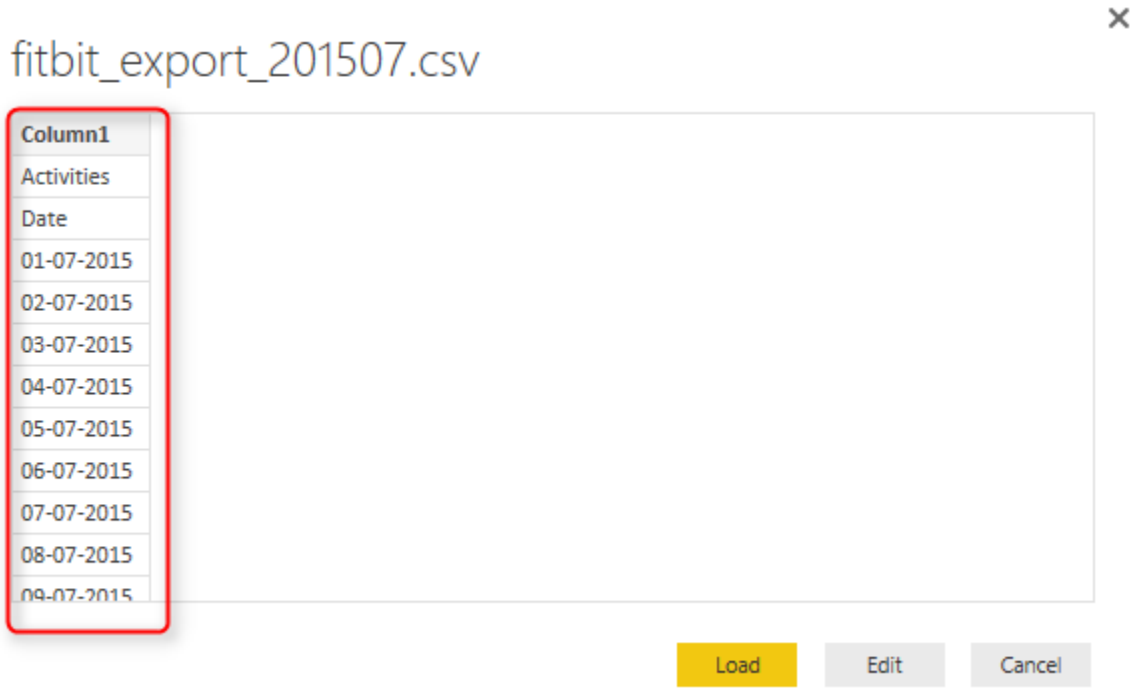
I have exported my Fitbit activities history from the time that I got it as a gift (Great gift I have to say ☺) till now, which is 5 months.



Here is sample data set in files:

```
Activities
Date,Calories Burned,Steps,Distance,Floors,Minutes Sedentary,Minutes Lightly Active,Minu
"01-07-2015","3,953","13,361","10.64","16","1,058","196","71","87","2,393"
"02-07-2015","3,432","10,818","8.76","10","726","224","19","45","1,803"
"03-07-2015","4,102","15,537","12.38","16","701","237","55","88","2,550"
"04-07-2015","3,959","14,490","11.54","6","685","255","46","69","2,443"
"05-07-2015","4,028","14,652","11.68","3","1,003","283","93","61","2,598"
"06-07-2015","3,538","12,161","9.84","16","1,144","168","76","52","1,968"
"07-07-2015","3,550","11,103","8.99","17","785","250","24","41","1,950"
"08-07-2015","3,238","10,029","8.22","10","1,204","152","36","48","1,587"
"09-07-2015","3,994","13,951","11.29","10","733","183","85","96","2,489"
"10-07-2015","3,998","12,516","9.5","9","650","194","92","92","2,508"
"11-07-2015","2,802","6,750","5.18","14","594","164","14","29","1,115"
"12-07-2015","4,156","19,471","15.39","21","472","272","92","96","2,781"
"13-07-2015","3,523","14,730","11.73","17","654","208","39","73","1,959"
"14-07-2015","3,039","9,543","7.24","16","712","195","10","38","1,409"
"15-07-2015","2,750","6,663","5.06","15","1,222","206","6","6","1,085"
"16-07-2015","3,373","14,603","11.46","12","745","198","22","69","1,813"
"17-07-2015","3,948","15,787","11.98","29","649","252","23","115","2,471"
"18-07-2015","3,917","16,652","13.35","19","543","332","52","55","2,512"
"19-07-2015","3,447","11,983","9.84","3","561","255","48","37","1,945"
"20-07-2015","3,430","12,816","10.35","15","709","174","37","74","1,834"
"21-07-2015","3,693","16,979","13.62","39","744","162","64","100","2,167"
"22-07-2015","3,461","13,088","10.52","15","717","211","29","59","1,863"
"23-07-2015","3,400","15,143","12.36","14","696","185","31","69","1,827"
"24-07-2015","4,092","20,077","15.82","24","690","172","70","127","2,602"
"25-07-2015","3,736","16,374","13.54","14","576","248","30","68","2,208"
"26-07-2015","3,453","13,676","11.18","4","576","232","45","55","1,933"
"27-07-2015","3,298","12,026","9.77","14","775","173","44","52","1,684"
"28-07-2015","3,654","17,401","14.14","17","1,140","163","37","100","2,115"
"29-07-2015","3,588","14,711","11.82","15","681","241","40","58","2,062"
"30-07-2015","3,059","8,671","6.58","30","790","231","7","20","1,417"
"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 getting Data from CSV. But this is the result I get:



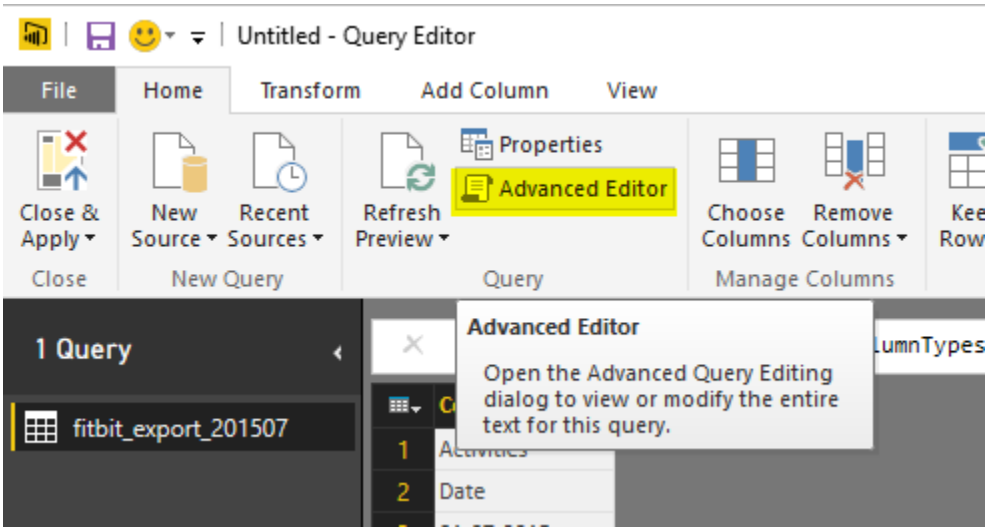
Well, this is happening because of the very first record in the CSV file which is the header row, as you see below;

```
Activities
Date,Calories Burned,Steps,Distance,Floors,Minutes Sedentary,Minutes Lightly Active,Minu
"01-07-2015","3,953","13,361","10.64","16","1,058","196","71","87","2,393"
"02-07-2015","3,432","10,818","8.76","10","726","224","19","45","1,803"
"03-07-2015","4,102","15,537","12.38","16","701","237","55","88","2,550"
"04-07-2015","3,959","14,490","11.54","6","685","255","46","69","2,443"
"05-07-2015","4,028","14,652","11.68","3","1,003","283","93","61","2,598"
"06-07-2015","3,538","12,161","9.84","16","1,144","168","76","52","1,968"
"07-07-2015","3,550","11,103","8.99","17","785","250","24","41","1,950"
"08-07-2015","3,238","10,029","8.22","10","1,204","152","36","48","1,587"
"09-07-2015","3,994","13,951","11.29","10","733","183","85","96","2,489"
"10-07-2015","3,998","12,516","9.5","9","650","194","92","92","2,508"
"11-07-2015","2,802","6,750","5.18","14","594","164","14","29","1,115"
"12-07-2015","4,156","19,471","15.39","21","472","272","92","96","2,781"
"13-07-2015","3,523","14,730","11.73","17","654","208","39","73","1,959"
"14-07-2015","3,039","9,543","7.24","16","712","195","10","38","1,409"
"15-07-2015","2,750","6,663","5.06","15","1,222","206","6","6","1,085"
"16-07-2015","3,373","14,603","11.46","12","745","198","22","69","1,813"
"17-07-2015","3,948","15,787","11.98","29","649","252","23","115","2,471"
"18-07-2015","3,917","16,652","13.35","19","543","332","52","55","2,512"
"19-07-2015","3,447","11,983","9.84","3","561","255","48","37","1,945"
"20-07-2015","3,430","12,816","10.35","15","709","174","37","74","1,834"
"21-07-2015","3,693","16,979","13.62","39","744","162","64","100","2,167"
"22-07-2015","3,461","13,088","10.52","15","717","211","29","59","1,863"
"23-07-2015","3,400","15,143","12.36","14","696","185","31","69","1,827"
"24-07-2015","4,092","20,077","15.82","24","690","172","70","127","2,602"
"25-07-2015","3,736","16,374","13.54","14","576","248","30","68","2,208"
"26-07-2015","3,453","13,676","11.18","4","576","232","45","55","1,933"
"27-07-2015","3,298","12,026","9.77","14","775","173","44","52","1,684"
"28-07-2015","3,654","17,401","14.14","17","1,140","163","37","100","2,115"
"29-07-2015","3,588","14,711","11.82","15","681","241","40","58","2,062"
"30-07-2015","3,059","8,671","6.58","30","790","231","7","20","1,417"
"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.

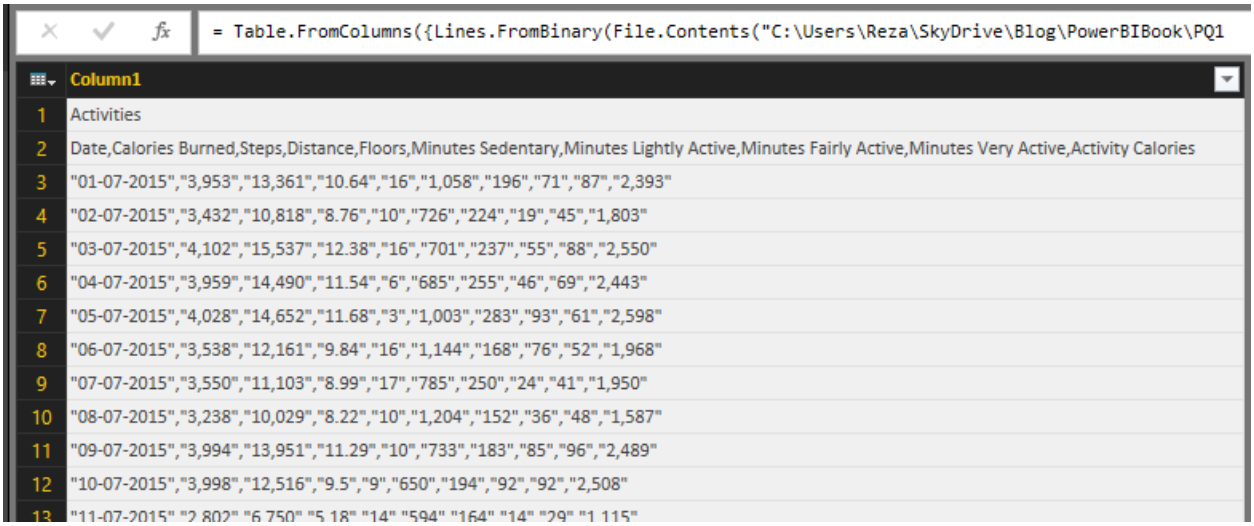


In the Advanced Editor window, you see the M script that generates the output.

Change the Code below;

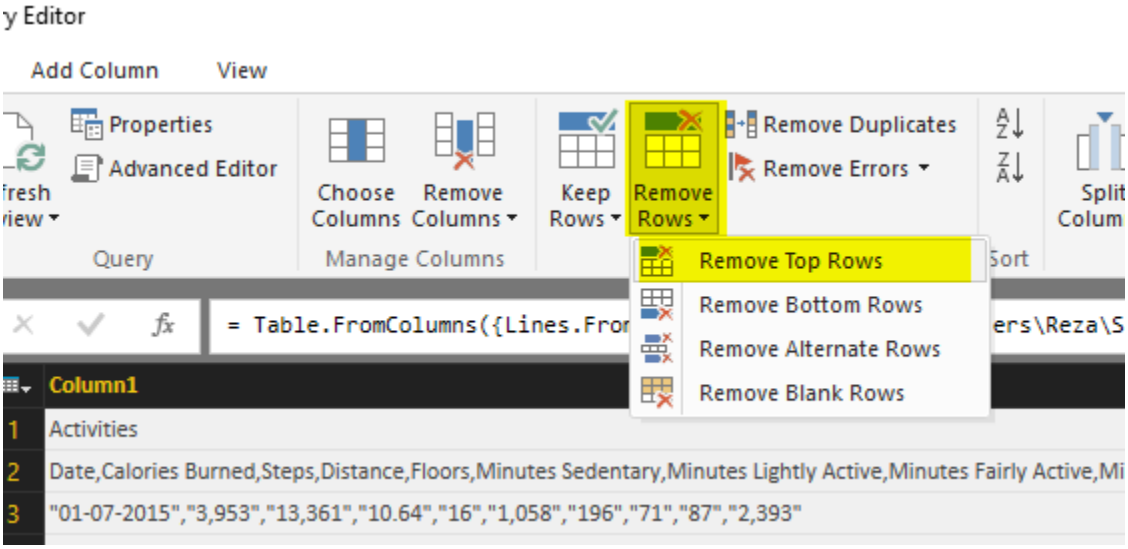
```
let
    Source =
Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV
\SourceFiles\fitbit_export_201507.csv"),null,null,1252)})
in
    Source
```

Now you can see the result as full data in one column rather than an only first column with date values.

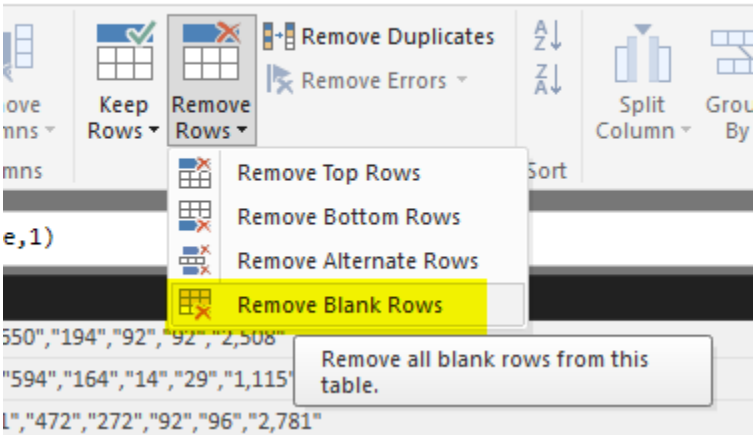


Follow the steps below to shape the data for using the best in the report;

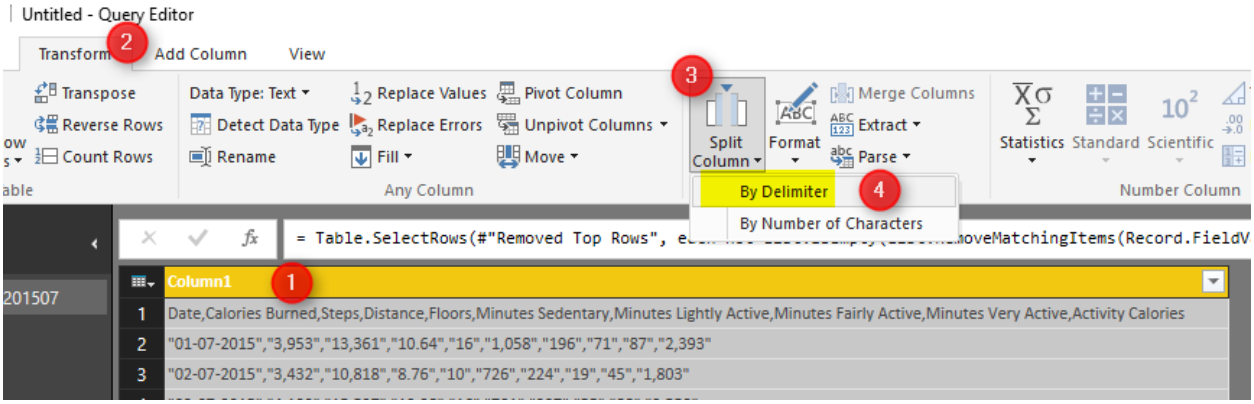
Get rid of the first record with Remove Rows, and then choosing 1 top row to remove;



You can also remove any blank rows at the end of the report



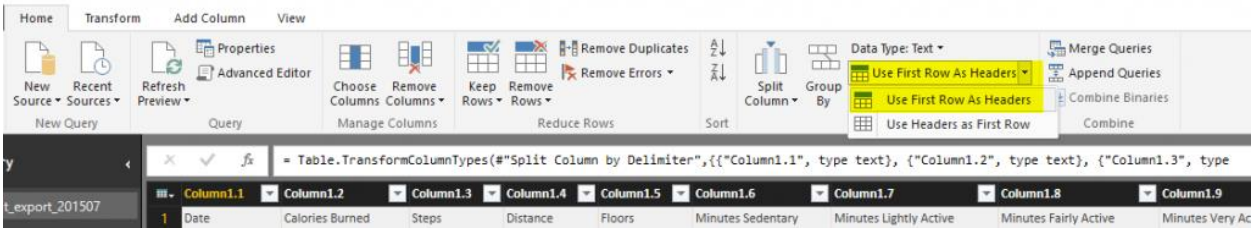
Then select Column1 and Got to Transform Tab, Choose Split Column, then By Delimiter



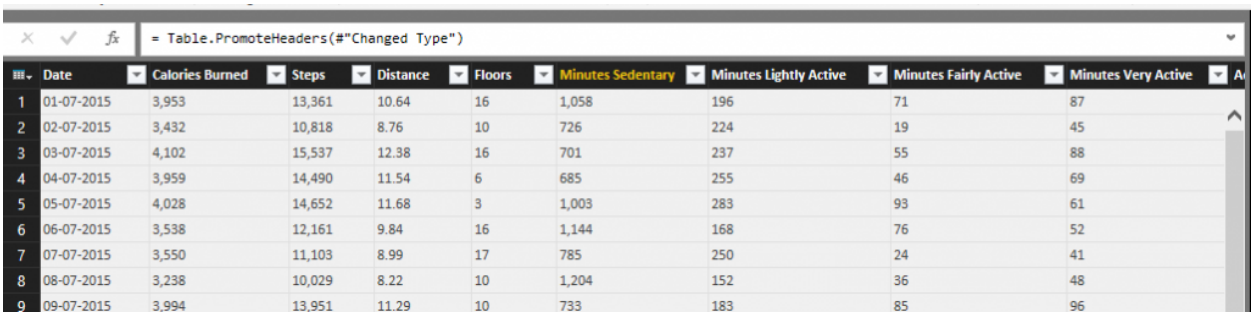
Leave the options as default (which is delimiter Comma, and Split at each occurrence of the delimiter) in Split Column by Delimiter dialog box and continue. You will see how columns all split as below;

	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 Acti
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 the header, which going to Home Tab in Query Editor window and clicks on Use First Row As Headers option.

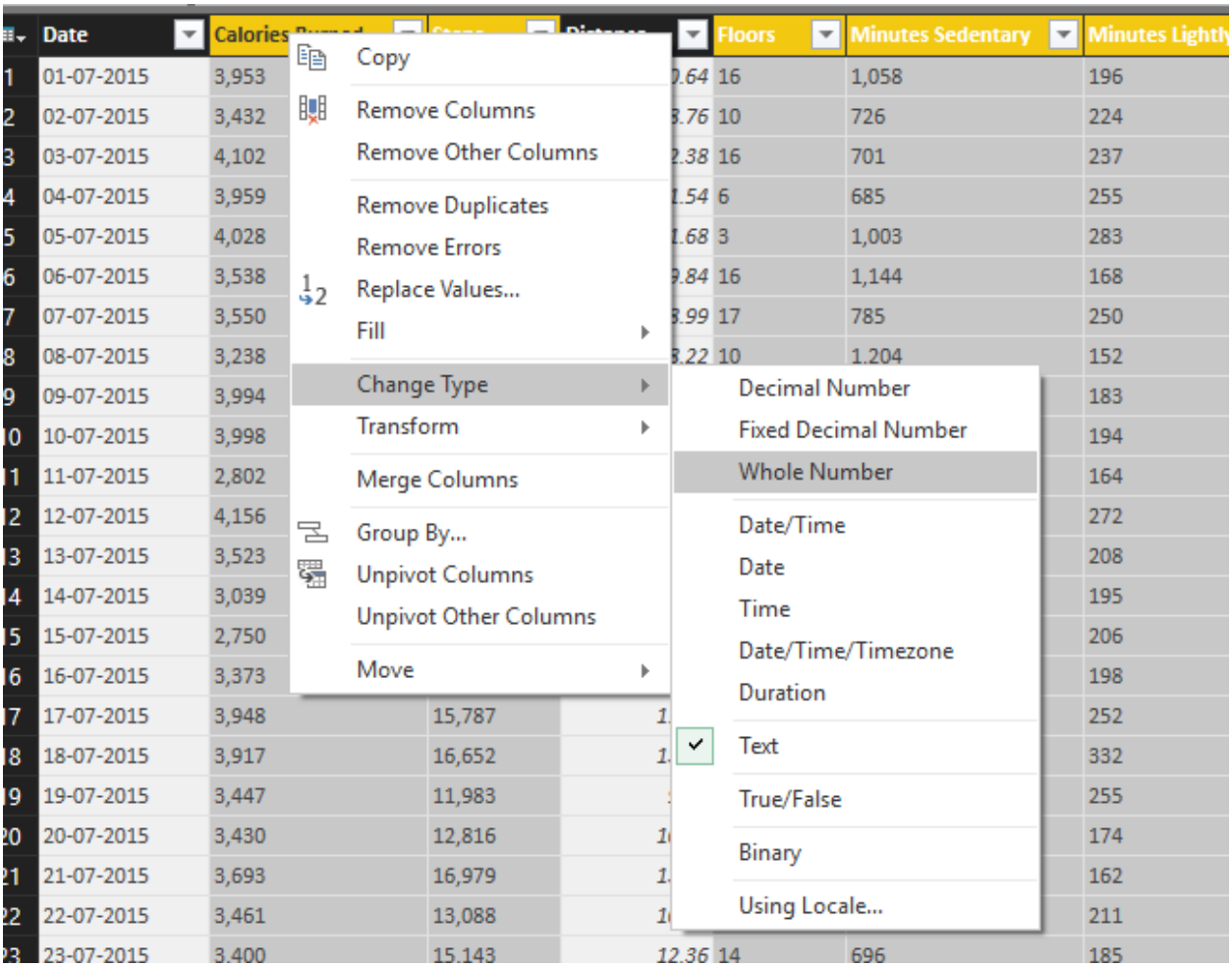


This option will rename all column headers to the values from the first row;

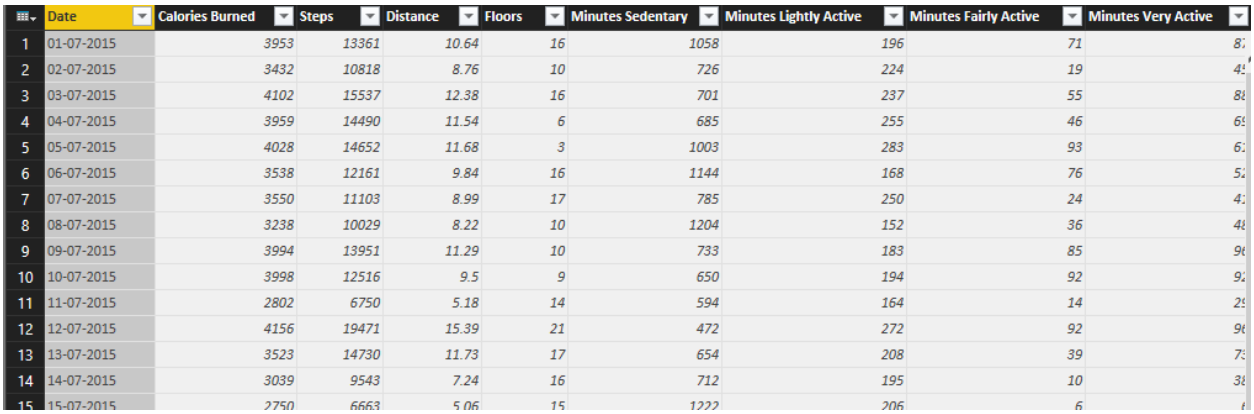


	Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active
1	01-07-2015	3,953	13,361	10.64	16	1,058	196	71	87
2	02-07-2015	3,432	10,818	8.76	10	726	224	19	45
3	03-07-2015	4,102	15,537	12.38	16	701	237	55	88
4	04-07-2015	3,959	14,490	11.54	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 the data type of Distance column to Decimal and all other columns (except Date) to the Whole Number

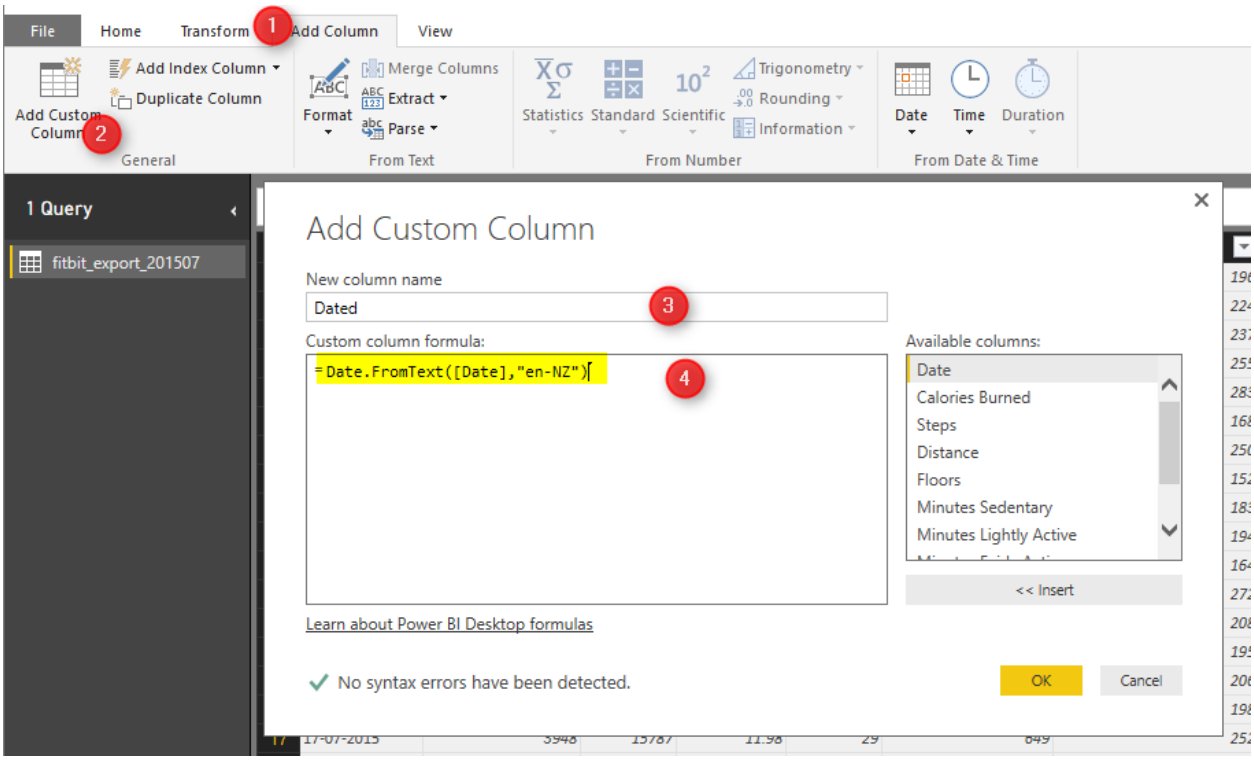


For the Date Column itself, you can't always change the type that easily, because the Date conversion works with the locale of your system. As you see in the screenshot below the Date format is DD-MM-YYYY which not always equal to the locale of all systems.



	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	8
2	02-07-2015	3432	10818	8.76	10	726	224	19	4
3	03-07-2015	4102	15537	12.38	16	701	237	55	8
4	04-07-2015	3959	14490	11.54	6	685	255	46	6
5	05-07-2015	4028	14652	11.68	3	1003	283	93	6
6	06-07-2015	3538	12161	9.84	16	1144	168	76	5
7	07-07-2015	3550	11103	8.99	17	785	250	24	4
8	08-07-2015	3238	10029	8.22	10	1204	152	36	4
9	09-07-2015	3994	13951	11.29	10	733	183	85	9
10	10-07-2015	3998	12516	9.5	9	650	194	92	9
11	11-07-2015	2802	6750	5.18	14	594	164	14	2
12	12-07-2015	4156	19471	15.39	21	472	272	92	9
13	13-07-2015	3523	14730	11.73	17	654	208	39	7
14	14-07-2015	3039	9543	7.24	16	712	195	10	3
15	15-07-2015	2750	6663	5.06	15	1222	206	6	

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 the list of culture and locale information. After doing that you will see Dated Column added to the end of the table with the 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

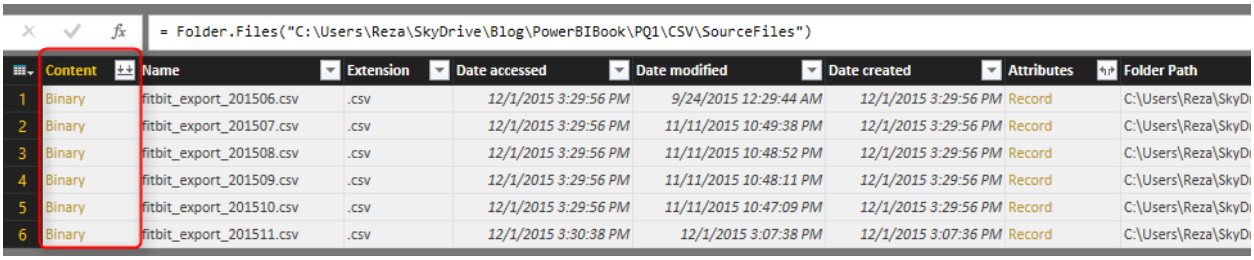
```
    Source =
Table.FromColumns({Lines.FromBinary(File.Contents("C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV
\SourceFiles\fitbit_export_201507.csv"),null,null,1252)}),
    #"Removed Top Rows" = Table.Skip(Source,1),
    #"Split Column by Delimiter" = Table.SplitColumn(#"Removed Top
Rows","Column1",Splitter.SplitTextByDelimiter(",", 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 the 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 also requires 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);


420K
Steps
Minutes Very Active, Minutes Sedentary, Minutes Lightly Active and Minutes Fairly Active
Minutes Very Active
Minutes Sedentary
Minutes Lightly Active
Minutes Fairly Active
Average of Calories Burned by IsWeekDay
Distance by Day of Week Name
100%
Friday
60.73
Monday
41.69
Saturday
43.61
Sunday
48.09
Thursday
50.45
Tuesday
43.99
Wednesday
46.26
76.2%
Floors and Steps by Date
Floors
Steps

Be Fitbit BI Developer in Few Steps: Step 2 Loop Through All CSV Files

Published Date: December 2, 2015



	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\SkyD
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\SkyD
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\SkyD
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\SkyD
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\SkyD
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\SkyD

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 the 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 contain 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 the 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 tthe ime 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 big large table. You will also learn some Power Query M functions through this process.

Where We've been in the Previous Step

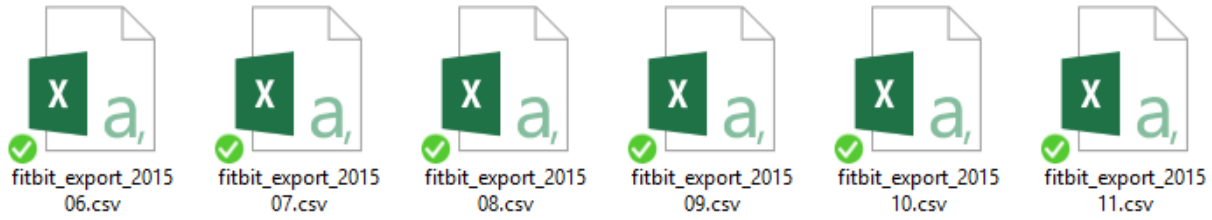
In the previous step, we ended up with below script

```
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

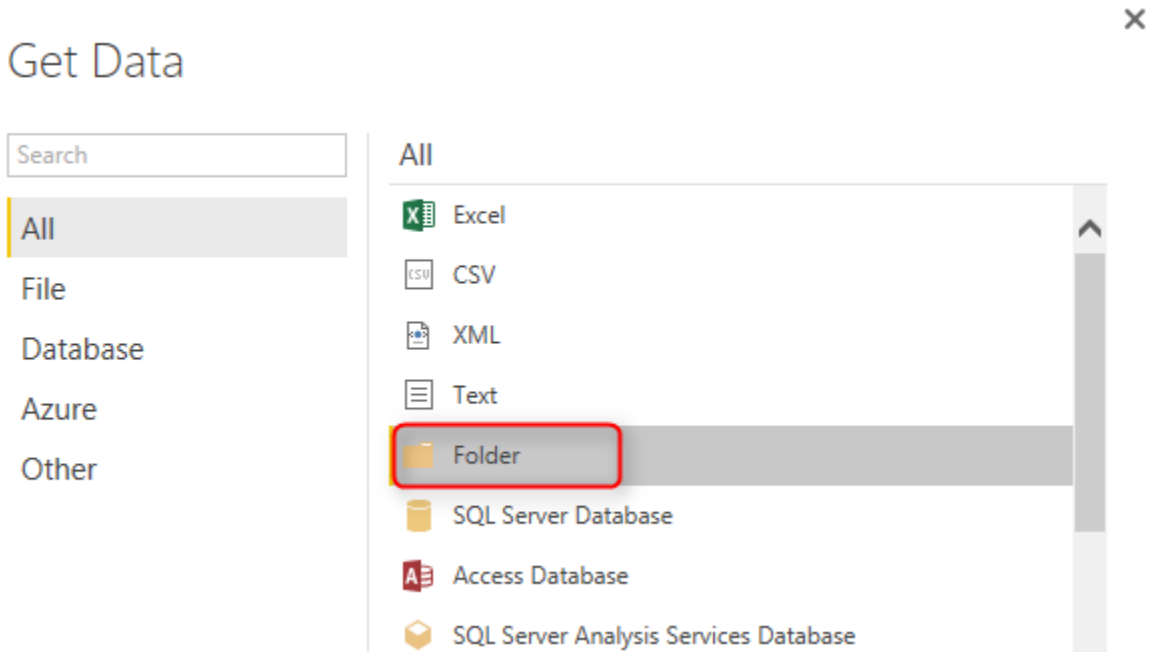
Dated	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Acti
7/1/2015	3953	13361	10.64	16	1058	196	71	87	
7/2/2015	3432	10818	8.76	10	726	224	19	45	
7/3/2015	4102	15537	12.38	16	701	237	55	88	
7/4/2015	3959	14490	11.54	6	685	255	46	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	

Now There are multiple files in the source folder that I want to repeat the process for all of them;

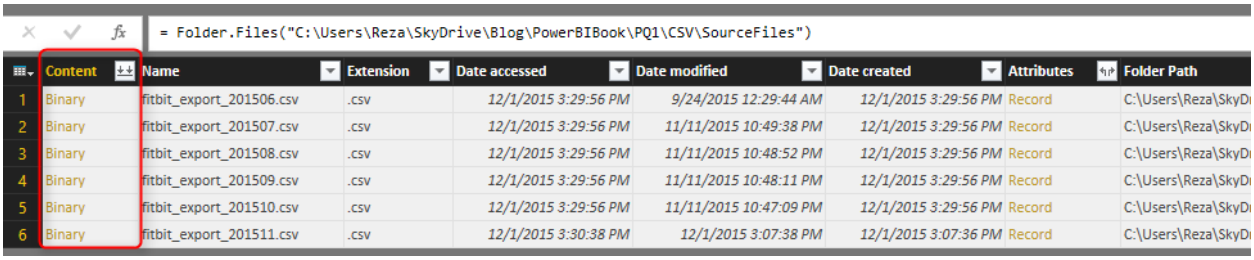


Get Data From Folder

Let's start the process with reading all files in a folder. Power BI Desktop can do that easily. Through getting Data options to choose from Folder and then 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 the name of the file, folder path, date created, modified, 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\SkyDr
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\SkyDr
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\SkyDr
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\SkyDr
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\SkyDr
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\SkyDr

Custom Function

So far we have a list of all files and the script to convert their content into the 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 part 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 a below 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 the code above is the code for the custom function. The first line of function (line 2 in above script) contains the name of the function; **FetchTable**, and the input parameter; **table**. From line 3 to line 16 is all the script we've copied from the [first step](#) of this exercise.

To Make sure that the function works properly you can call it below;

```
FetchTable(Files{[#"Folder
Path"="C:\Users\Reza\SkyDrive\Blog\PowerBIBook\PQ1\CSV\SourceFiles\",Name="fitbit_export_201509.cs
v"]}[Content])
```

Add above a single line of code in the IN section of the 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 to 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 especially 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 the 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 gets 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 see in the above example, 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 extra manual 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!

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 it 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 help building visualization elements.

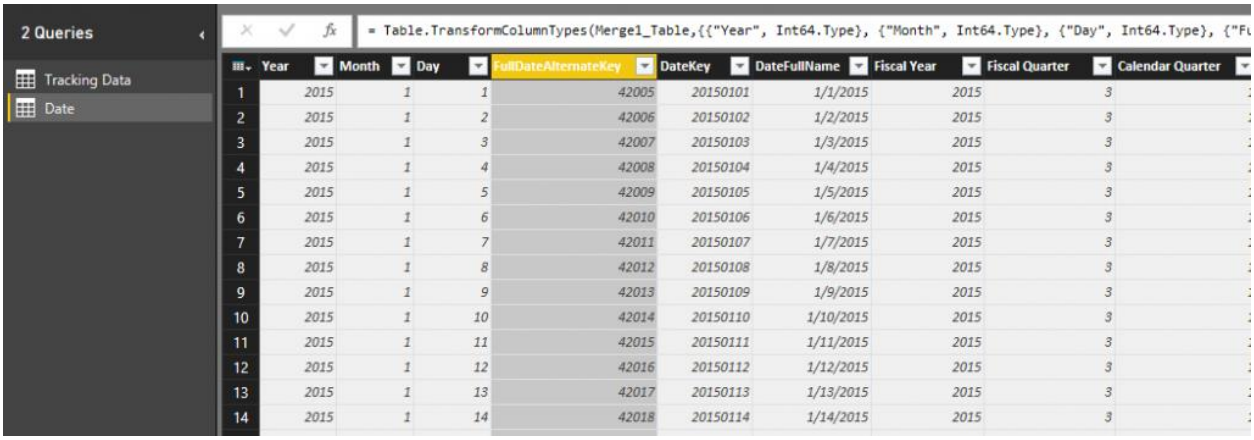
Adding Date Dimension

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 a result:

Year	Month	Day	FullDateAlt	DateKey	DateFullName	Fiscal Year	Fiscal Qu	Calendar Q	IsWeek	DayOfV	Month Nam	Day of Week Na	HolidayDescription	IsPubli
2015	1	1	42005	20150101	01 January 2015	2015	3	1	1	4	January	Thursday	New Year's Day	
2015	1	2	42006	20150102	02 January 2015	2015	3	1	1	5	January	Friday	Day after New Year's Day	
2015	1	3	42007	20150103	03 January 2015	2015	3	1	0	6	January	Saturday		
2015	1	4	42008	20150104	04 January 2015	2015	3	1	0	0	January	Sunday		
2015	1	5	42009	20150105	05 January 2015	2015	3	1	1	1	January	Monday		
2015	1	6	42010	20150106	06 January 2015	2015	3	1	1	2	January	Tuesday		
2015	1	7	42011	20150107	07 January 2015	2015	3	1	1	3	January	Wednesday		
2015	1	8	42012	20150108	08 January 2015	2015	3	1	1	4	January	Thursday		
2015	1	9	42013	20150109	09 January 2015	2015	3	1	1	5	January	Friday		
2015	1	10	42014	20150110	10 January 2015	2015	3	1	0	6	January	Saturday		
2015	1	11	42015	20150111	11 January 2015	2015	3	1	0	0	January	Sunday		
2015	1	12	42016	20150112	12 January 2015	2015	3	1	1	1	January	Monday		
2015	1	13	42017	20150113	13 January 2015	2015	3	1	1	2	January	Tuesday		
2015	1	14	42018	20150114	14 January 2015	2015	3	1	1	3	January	Wednesday		
2015	1	15	42019	20150115	15 January 2015	2015	3	1	1	4	January	Thursday		
2015	1	16	42020	20150116	16 January 2015	2015	3	1	1	5	January	Friday		
2015	1	17	42021	20150117	17 January 2015	2015	3	1	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 the previous step of this exercise);



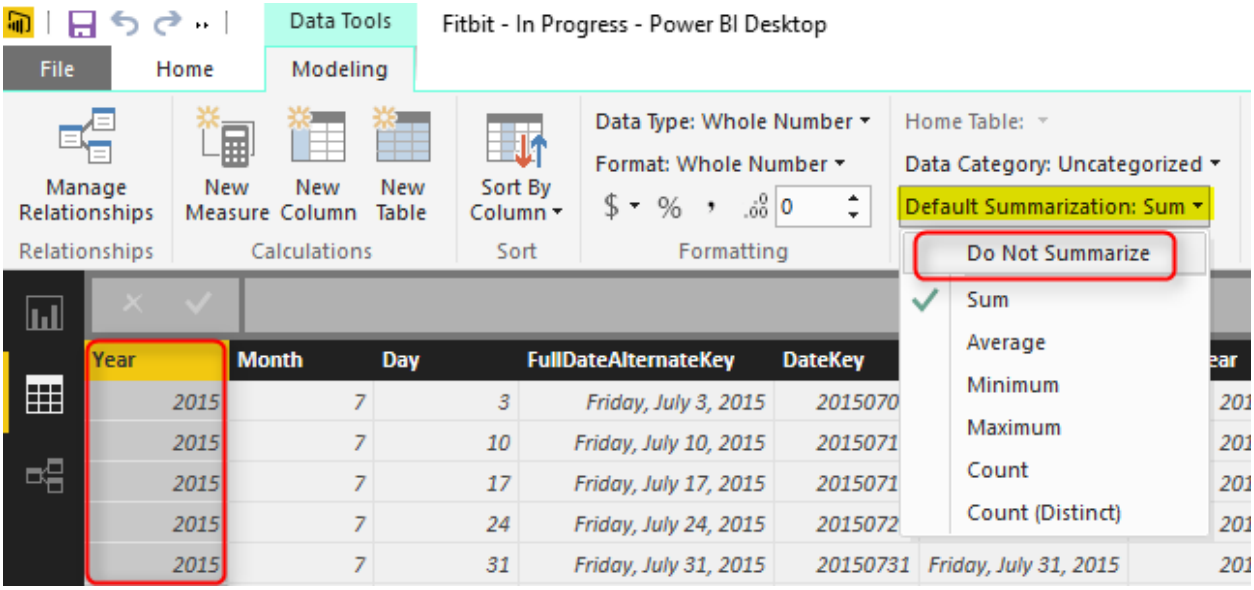
	Year	Month	Day	FullDateAlternateKey	DateKey	DateFullName	Fiscal Year	Fiscal Quarter	Calendar Quarter
1	2015	1	1	42005	20150101	1/1/2015	2015	3	
2	2015	1	2	42006	20150102	1/2/2015	2015	3	
3	2015	1	3	42007	20150103	1/3/2015	2015	3	
4	2015	1	4	42008	20150104	1/4/2015	2015	3	
5	2015	1	5	42009	20150105	1/5/2015	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 right click on it and change it to Date format, so I have proper date field there;

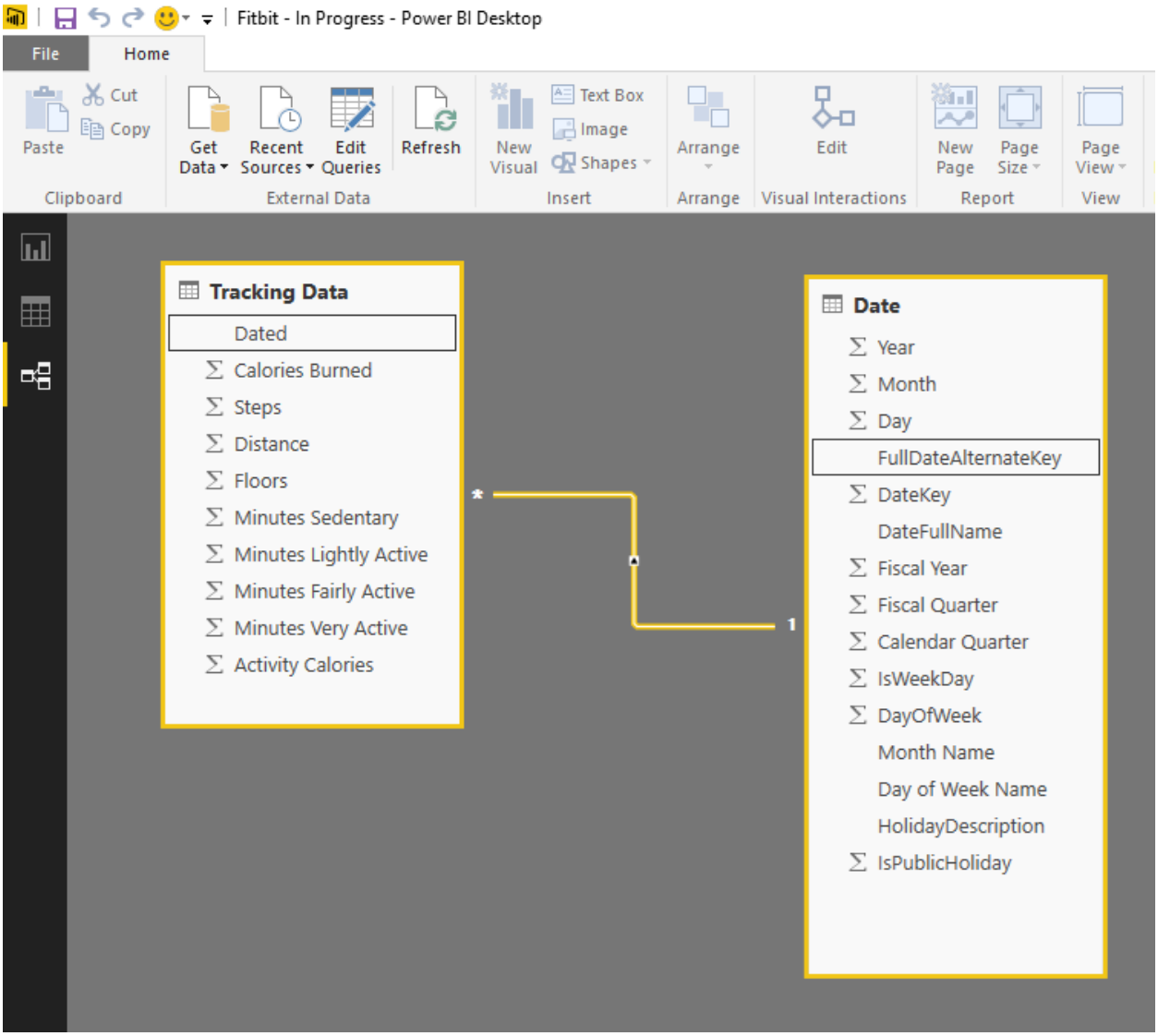
	Year	Month	Day	FullDateAlternateKey	DateKey	DateFullName	Fiscal Year	Fiscal Quarter
1	2015	1	1	1/1/2015	20150101	1/1/2015	2015	
2	2015	1	2	1/2/2015	20150102	1/2/2015	2015	
3	2015	1	3	1/3/2015	20150103	1/3/2015	2015	
4	2015	1	4	1/4/2015	20150104	1/4/2015	2015	
5	2015	1	5	1/5/2015	20150105	1/5/2015	2015	
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 the 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.

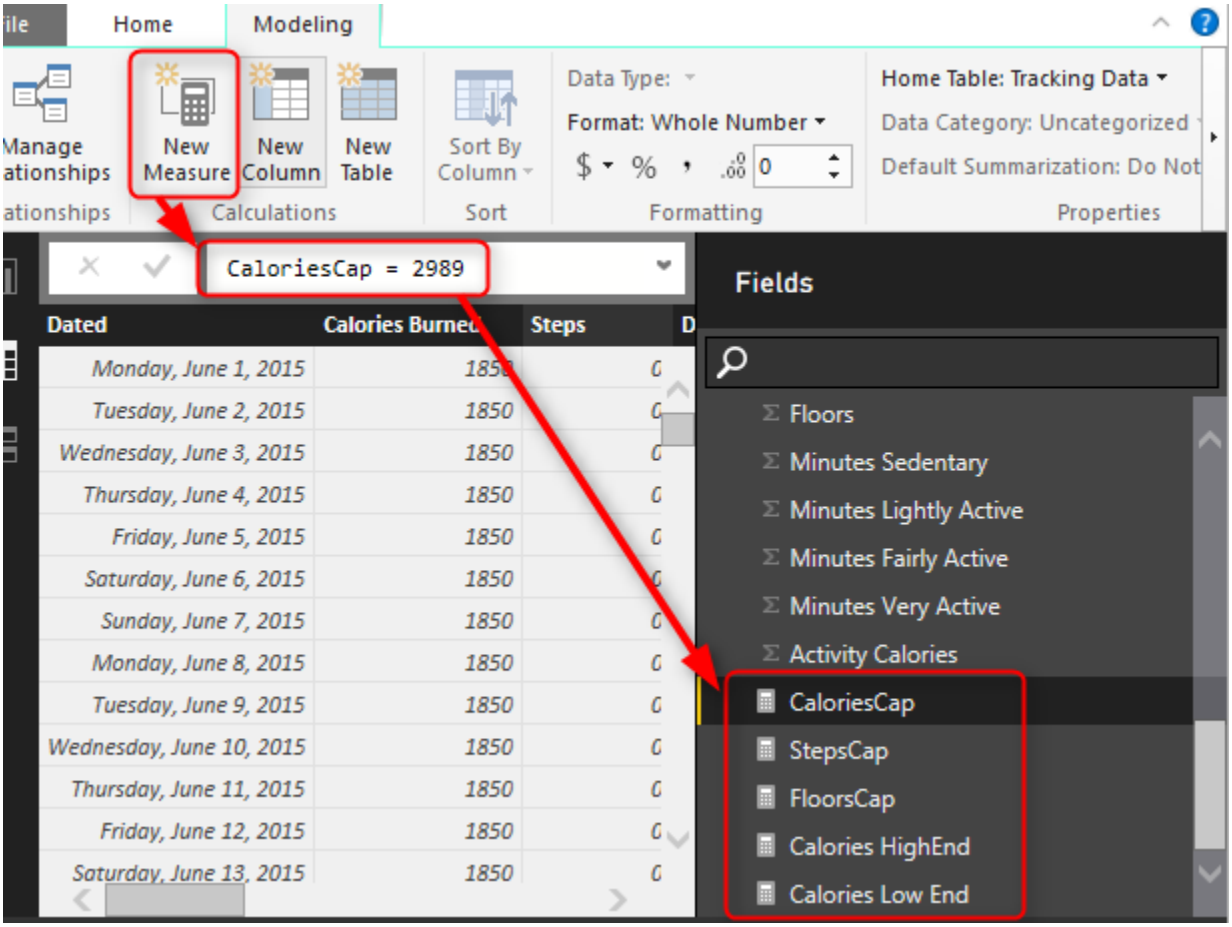


Now go to the Relationship tab and create a relationship between two tables based on their full date columns which are; FullDateAlternateKey Column from Date, and Dated Column in Tracking Data table.



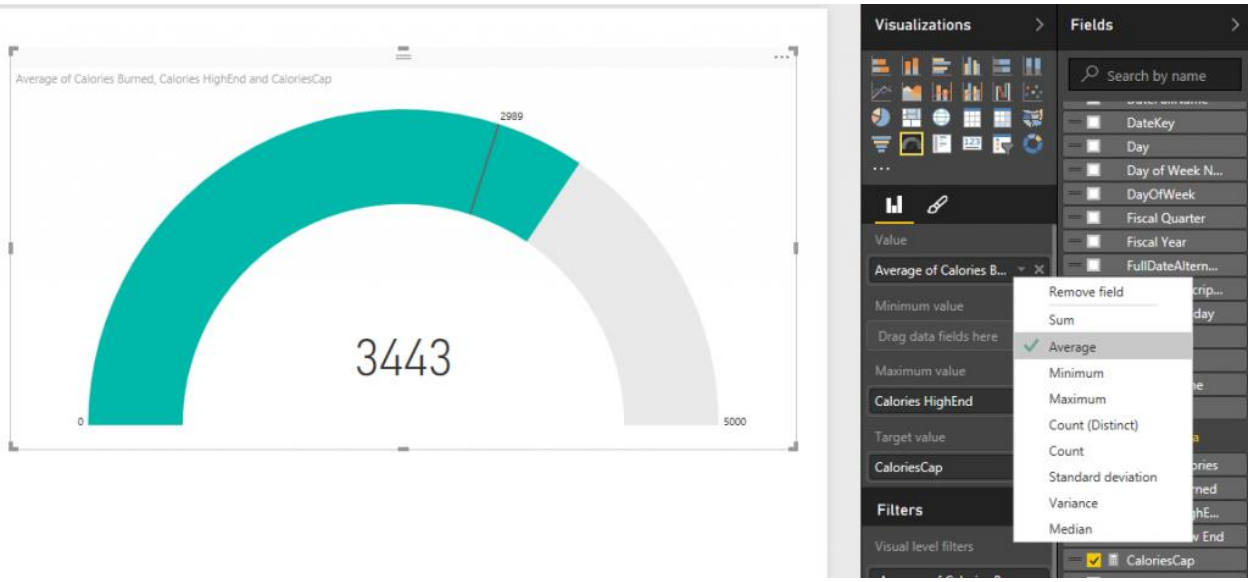
Adding Few Static Measures

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 number 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 the Data tab.

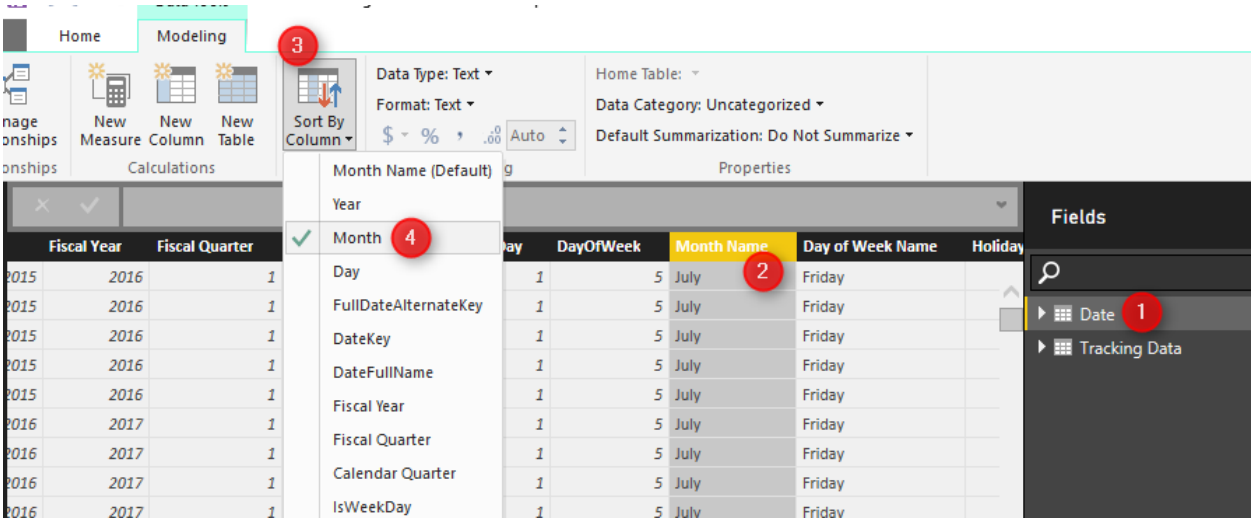


Let's Visualize

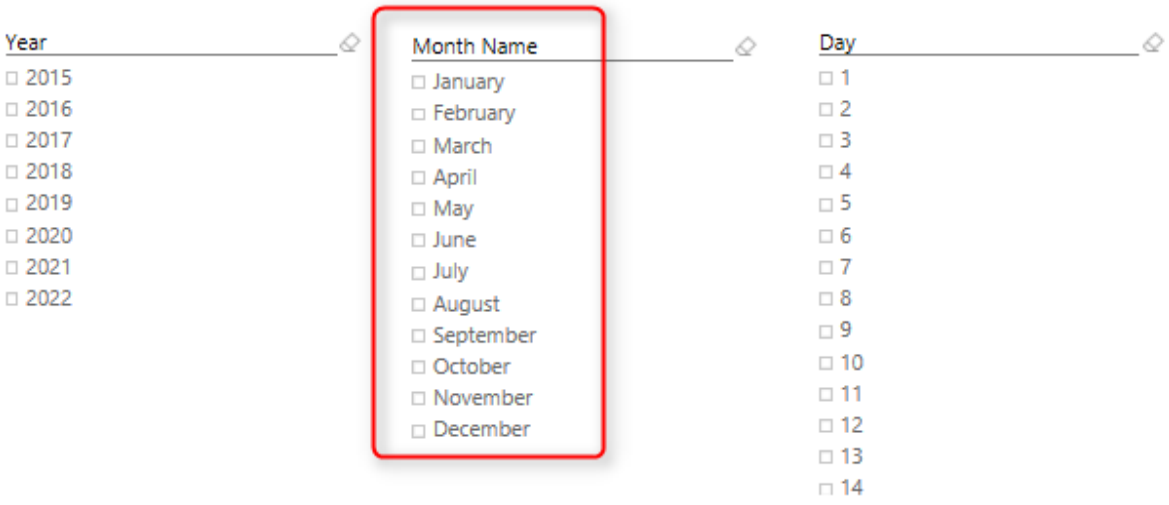
It's time to visualize what we have so far in the data set. Go to the Report tab, and start with building a Gauge. Then drag and drop CaloriesCap as Target Value. You can also put Calories HighEnd measure as Maximum. And then drag and drop Calories measure into the value. Change the aggregation from the sum to Average as it showed in below image.



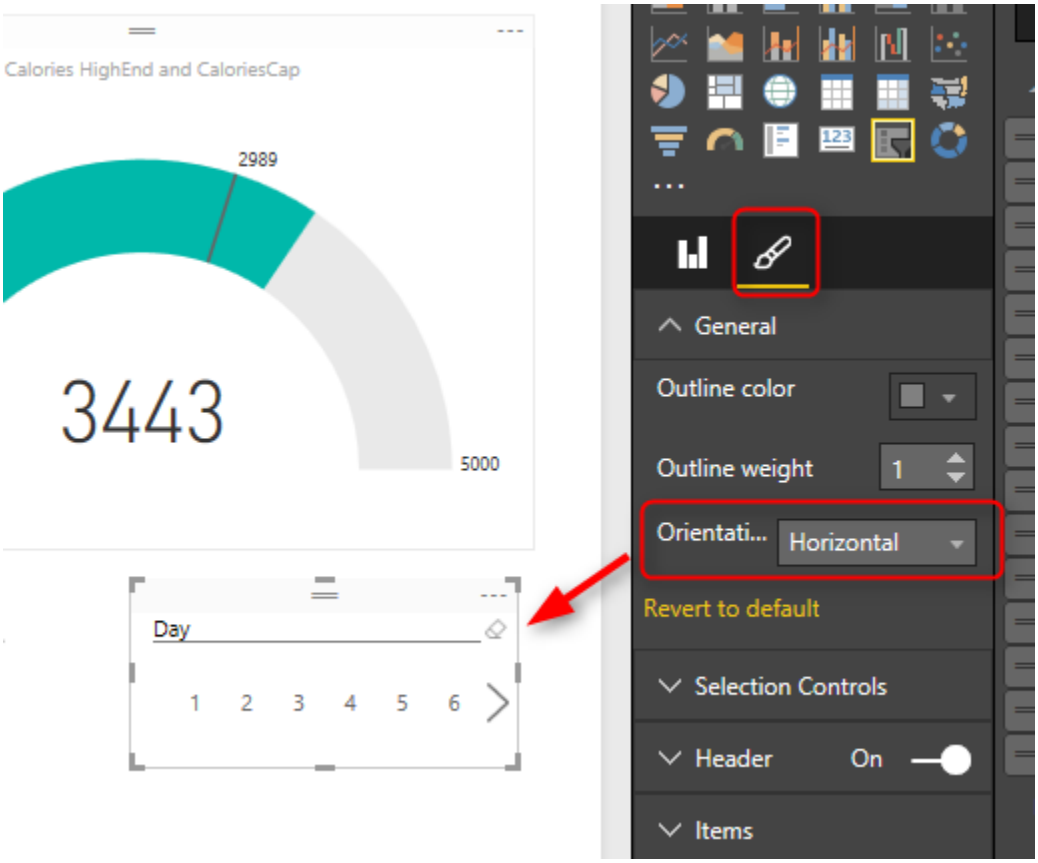
Easy! and lucky me who spent (on average) more than the calories I have to spend each day. Do the same for Floor with FloorCap and Floors measure. Now Add Three Slicers for; Year, Month Name, and Day. When you create a slicer for Month Name, you will see month names are ordered by their alphabetic order of names which is not right. Go to the 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 the year);



Now that you've set ordering you will be able to see Month names in the correct order in the slicer as below;

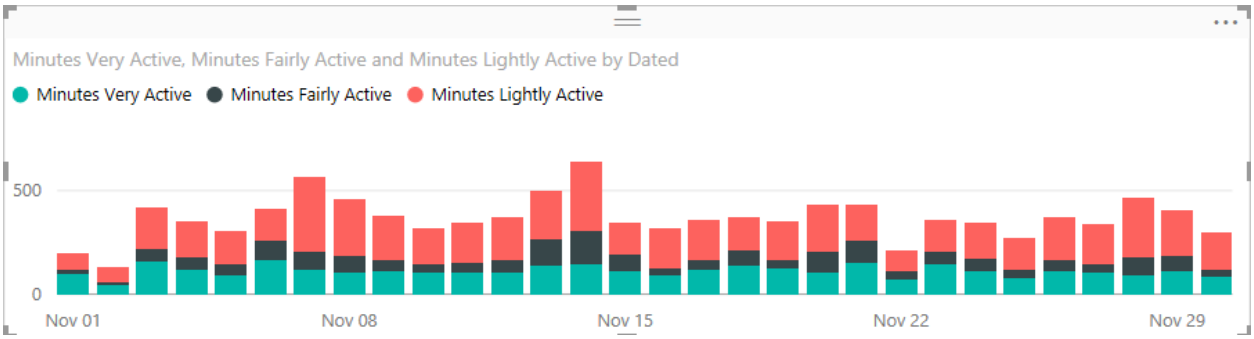


Change the orientation of Day slicer to be horizontal rather than vertical in the formatting option of the slicer



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 the value

Add a stacked column chart with Dated as the axis, and three measures as value: Minutes Very Active, Minutes Fairly Active, and Minutes Lightly Active.

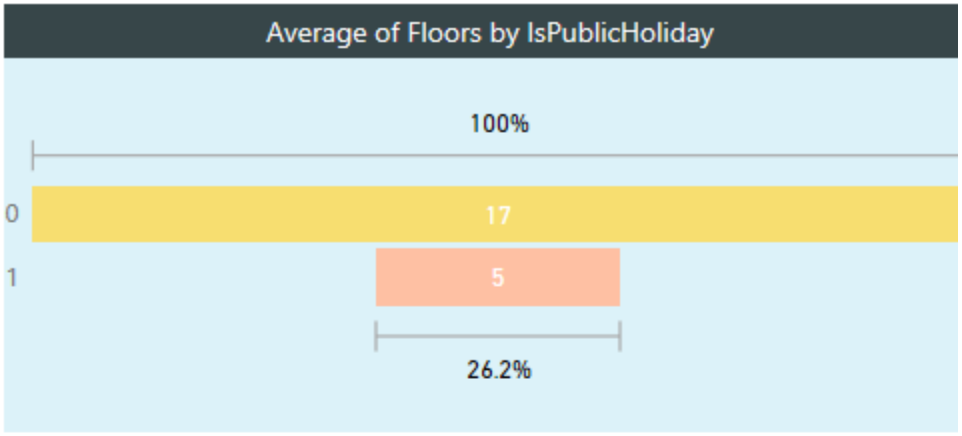


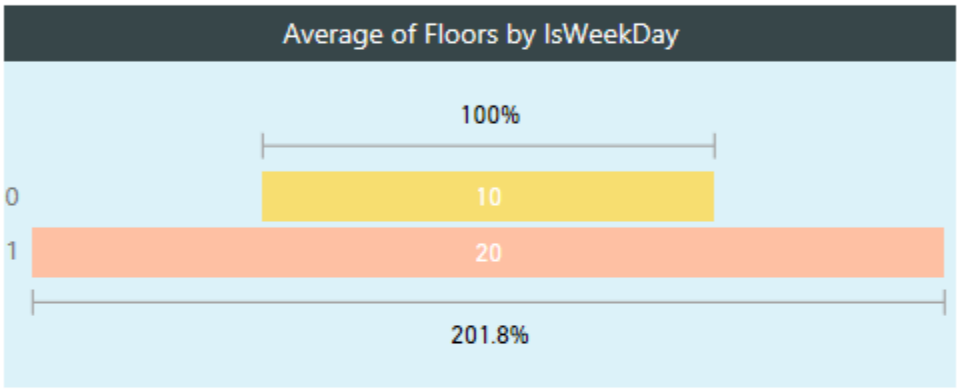
Add Also an average of all of three measure above in a pie chart to show the total average. And Add two Card visualization one for Average of steps, and another for an average of distance.

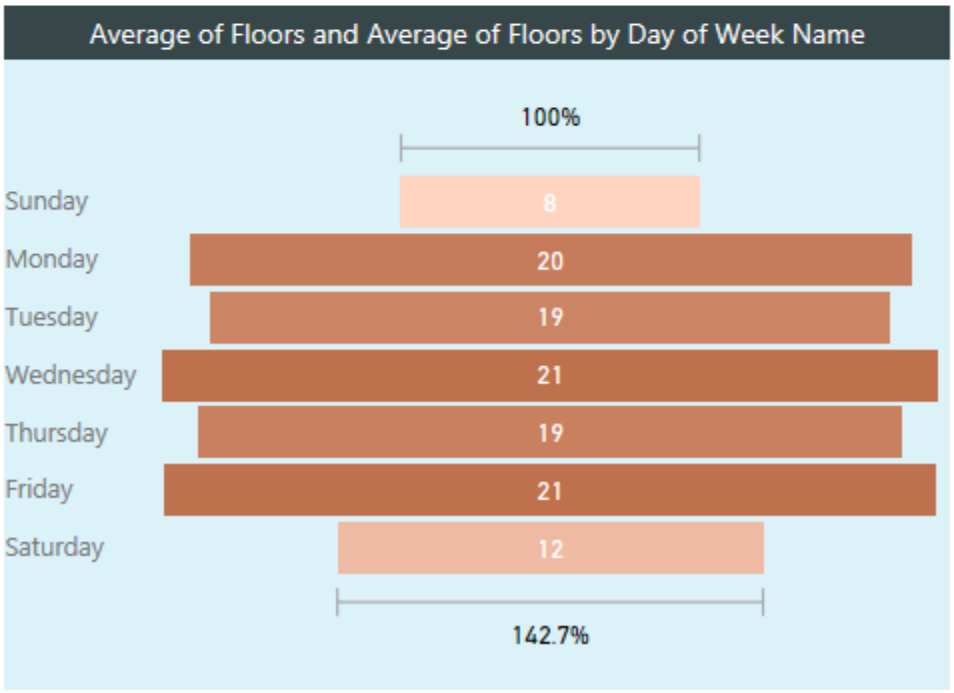
Well, you can customize the visualization as you want in the formatting option of each chart or element. Have a play with it and create something remarkable. I'm sure you design it better than me, here is what I built in a few minutes;

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 the week usually I perform best in walking! Here we go;

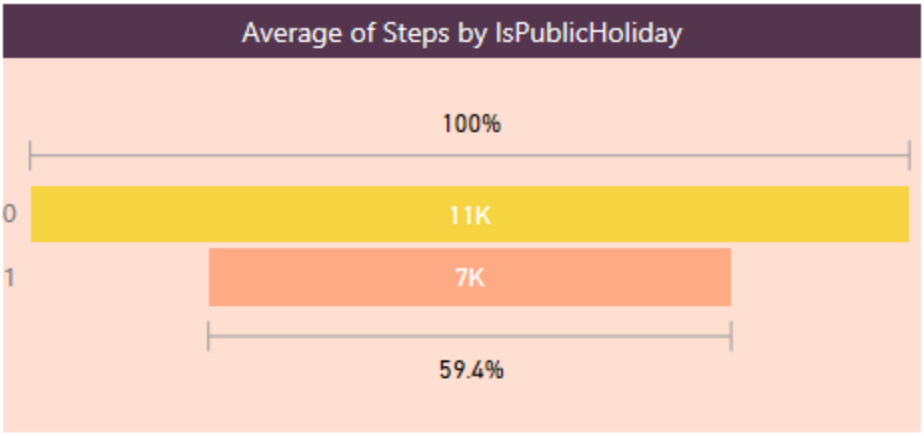
Interesting! Isn't it? I can get more insights out of this visualization than what I see in Fitbit dashboard. I can see that I'm performing badly in taking floors in public holidays! Well, our home is the flat single floor. However, I'm doing a little bit better at weekends. My best days for Floors is always weekdays, and there isn't so much difference between them.

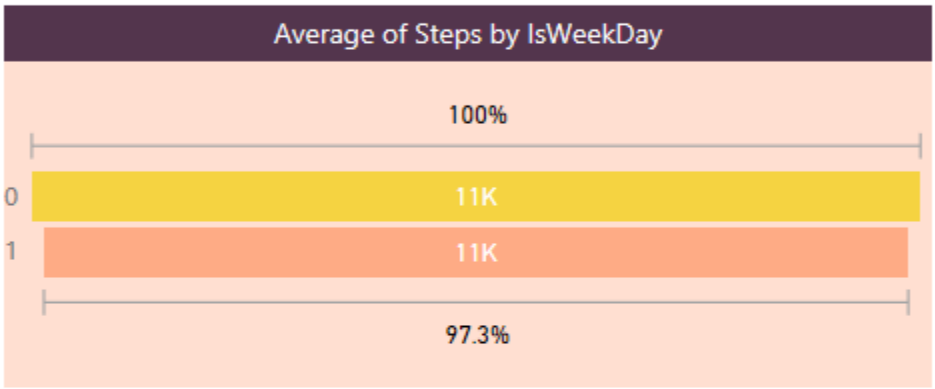

Average of Floors by IsPublicHoliday
100%
0
17
1
5
26.2%

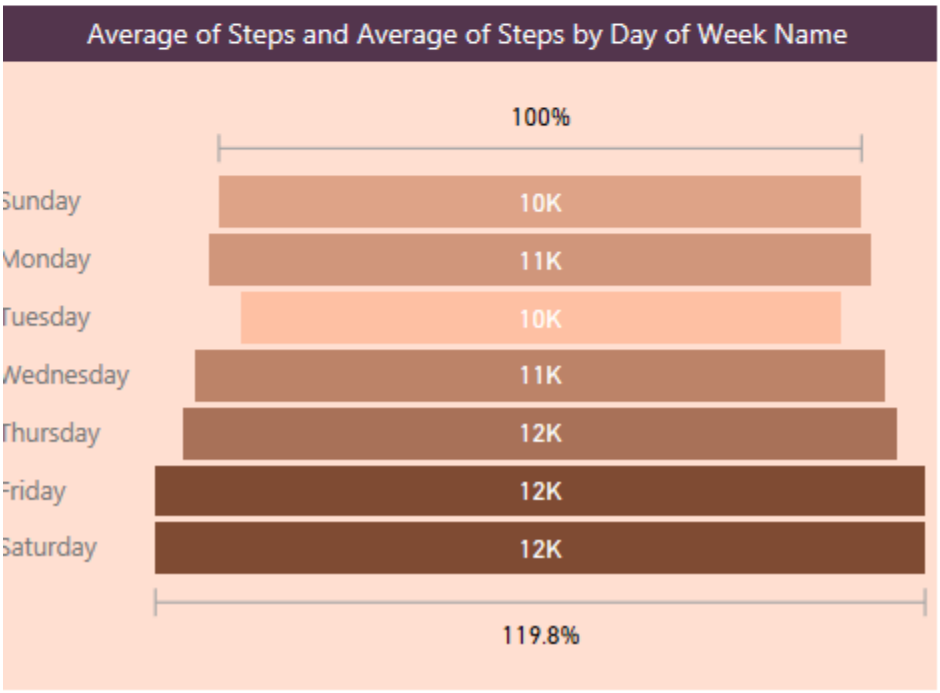

Average of Floors by IsWeekDay
100%
0
10
1
20
201.8%


Average of Floors and Average of Floors by Day of Week Name
100%
Sunday
8
Monday
20
Tuesday
19
Wednesday
21
Thursday
19
Friday
21
Saturday
12
142.7%

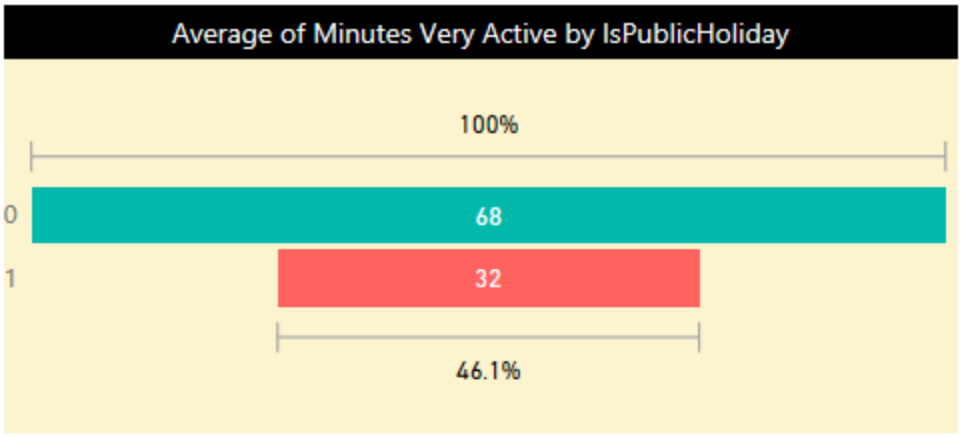
Now if I check my steps; I'm not doing that bad in public holidays, 7K in public holidays in comparison with 11K is other days is fine for me 😉 A, nd I'm doing even very close in weekends and weekdays. If I look at each day's average I can see that I'm a bit tired in Sundays and prefer to rest more, 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.

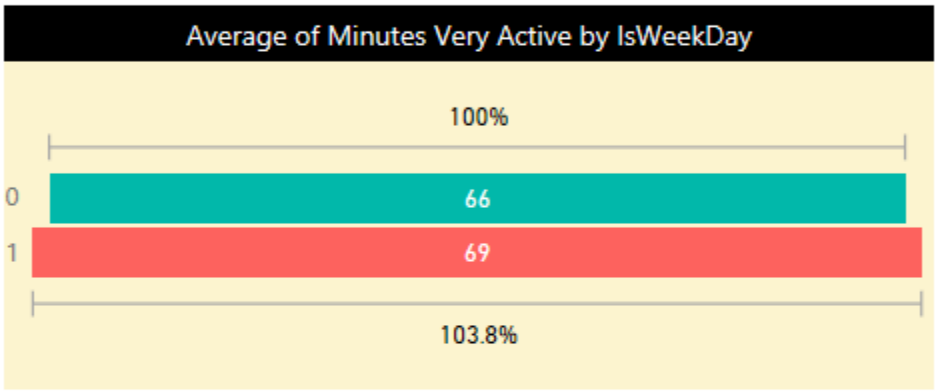

Average of Steps by IsPublicHoliday
100%
0
11K
1
7K
59.4%

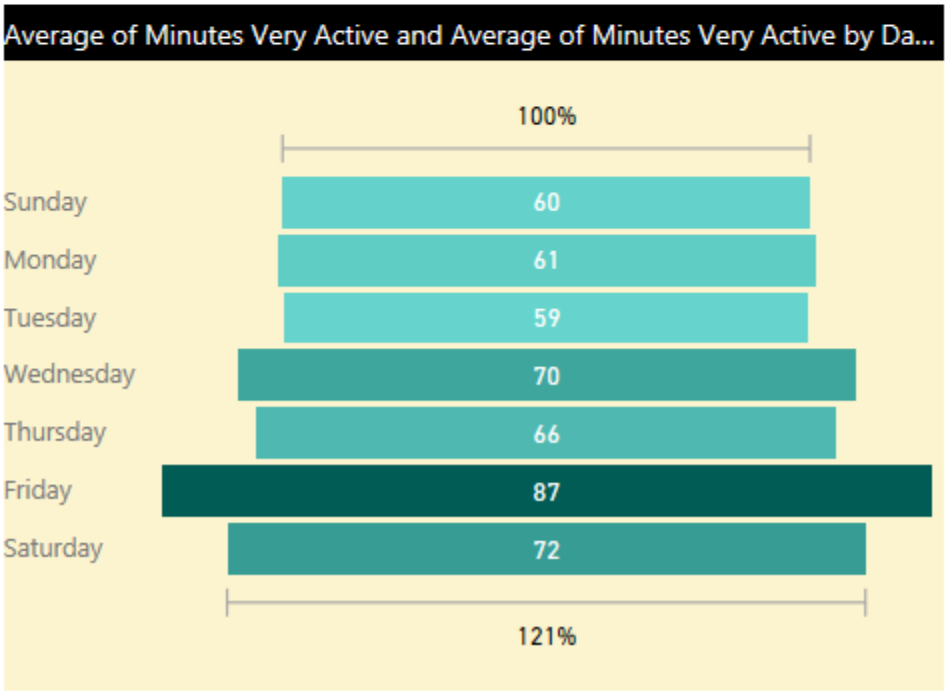

Average of Steps by IsWeekDay
100%
0
11K
1
11K
97.3%


Average of Steps and Average of Steps by Day of Week Name
100%
Sunday
10K
Monday
11K
Tuesday
10K
Wednesday
11K
Thursday
12K
Friday
12K
Saturday
12K
119.8%

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 come Fridays. There is an obvious reason for it. Fridays I play indoor soccer with some friends, and I feel my heart is coming out of my chest at some minutes in the game! Here is the evidence; I got the most active minutes on Fridays!


Average of Minutes Very Active by IsPublicHoliday
100%
0
68
1
32
46.1%

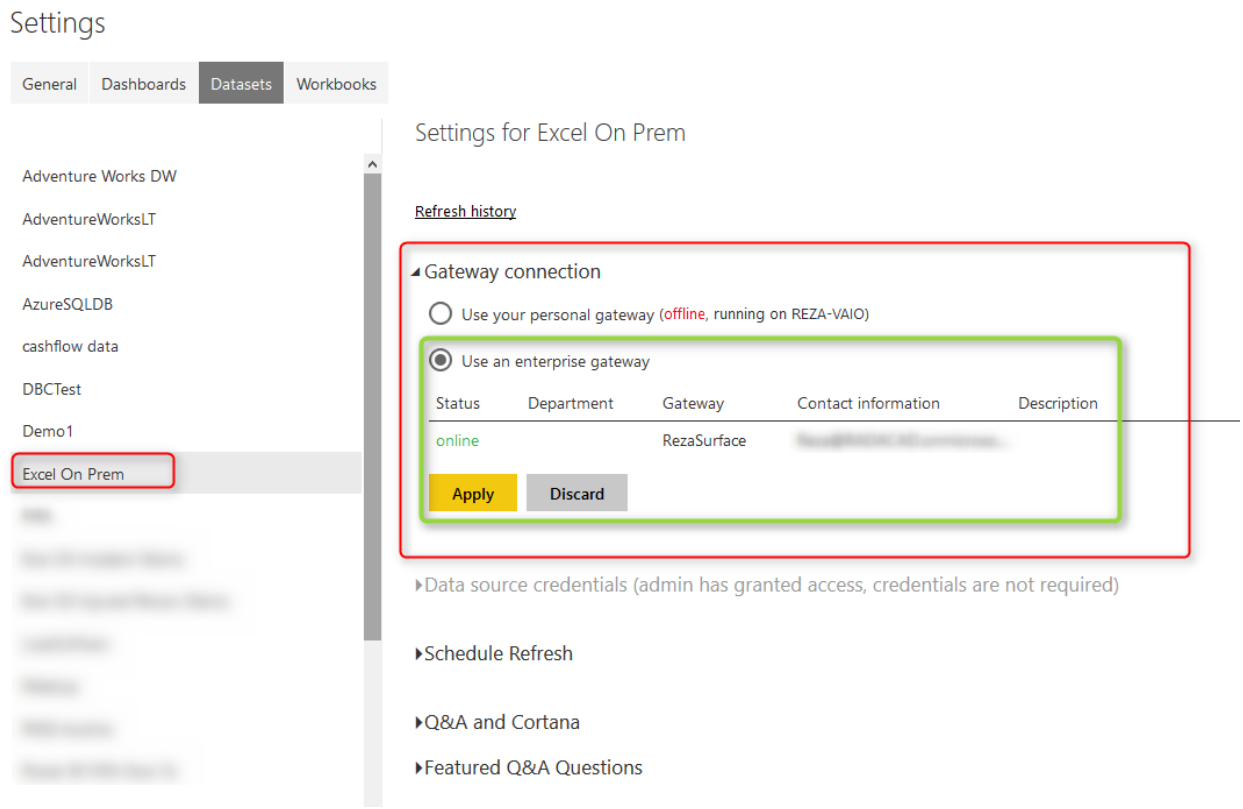

Average of Minutes Very Active by IsWeekDay
100%
0
66
1
69
103.8%


Average of Minutes Very Active and Average of Minutes Very Active by Da...
100%
Sunday
60
Monday
61
Tuesday
59
Wednesday
70
Thursday
66
Friday
87
Saturday
72
121%

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 creativity and build dashboards and charts that give you more insights. You've seen so far how easy it is to be a BI developer if you use Power BI 😉

Schedule Refresh Local Files on Power BI Web Site

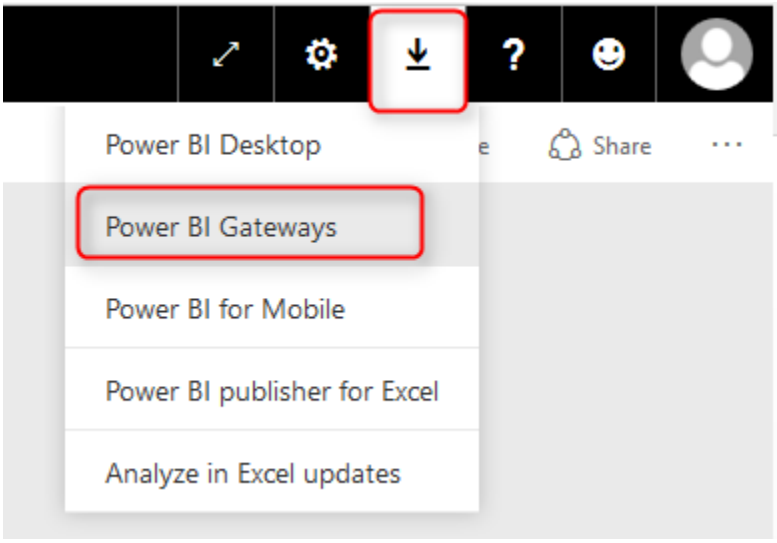
Published Date: June 8, 2016



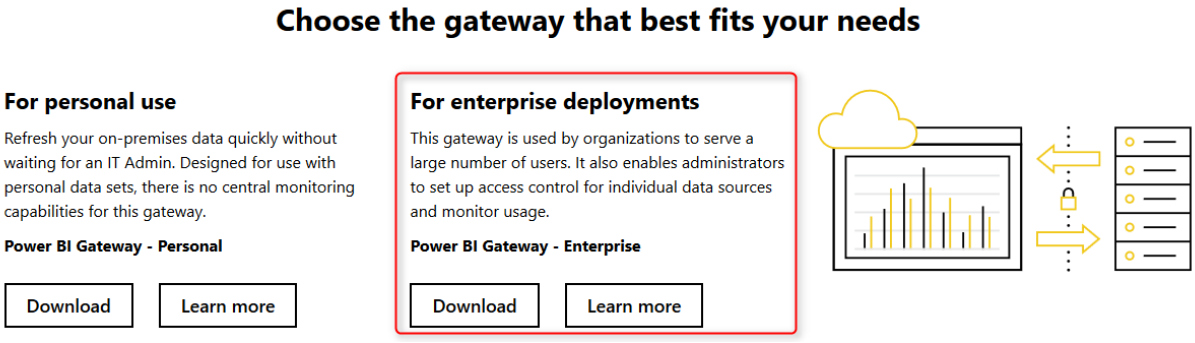
I get this question asked in webinars, speakings, and comments on my blog posts that Is it possible to use a local file as a source in Power BI and schedule that to be refreshed? The answer is Yes, of Course. You can host your files locally or on a server on premises, and then use gateways to create 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 walkthrough 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 a gateway, or how to build a sample report in Power BI if you are interested in learning more on those subjects read posts on Power BI online book; From Rookie to Rock Star.

Install and Configure Gateway

Two gateways allows us to connect from Power BI service or website (which is a cloud service) to on-premises data sources; Personal and Enterprise gateway. There are some differences between these two gateways which I will explain in a 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 the screenshot below;

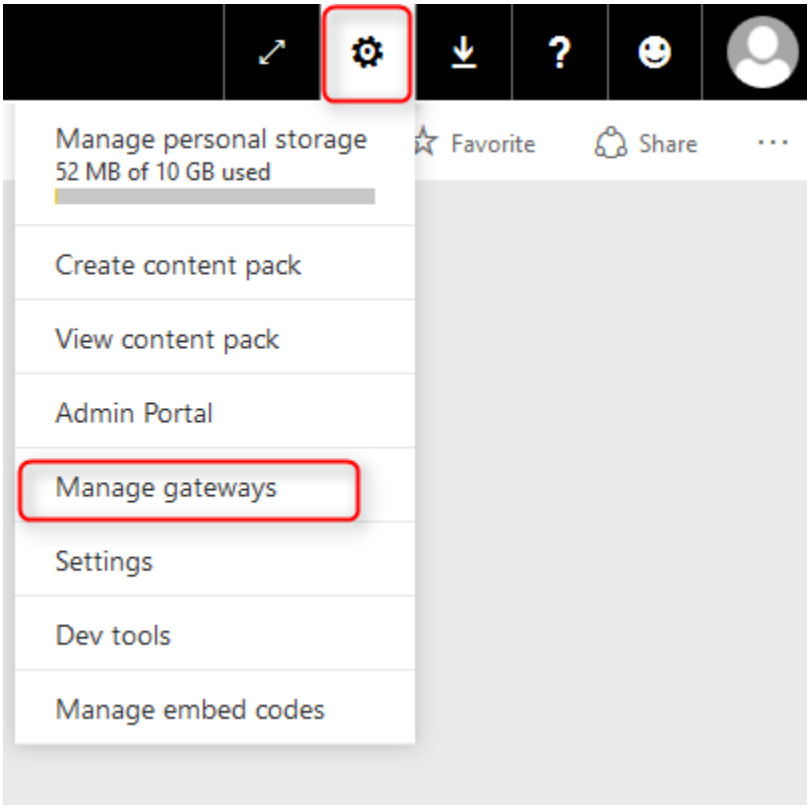


Then choose Enterprise Gateway for this example

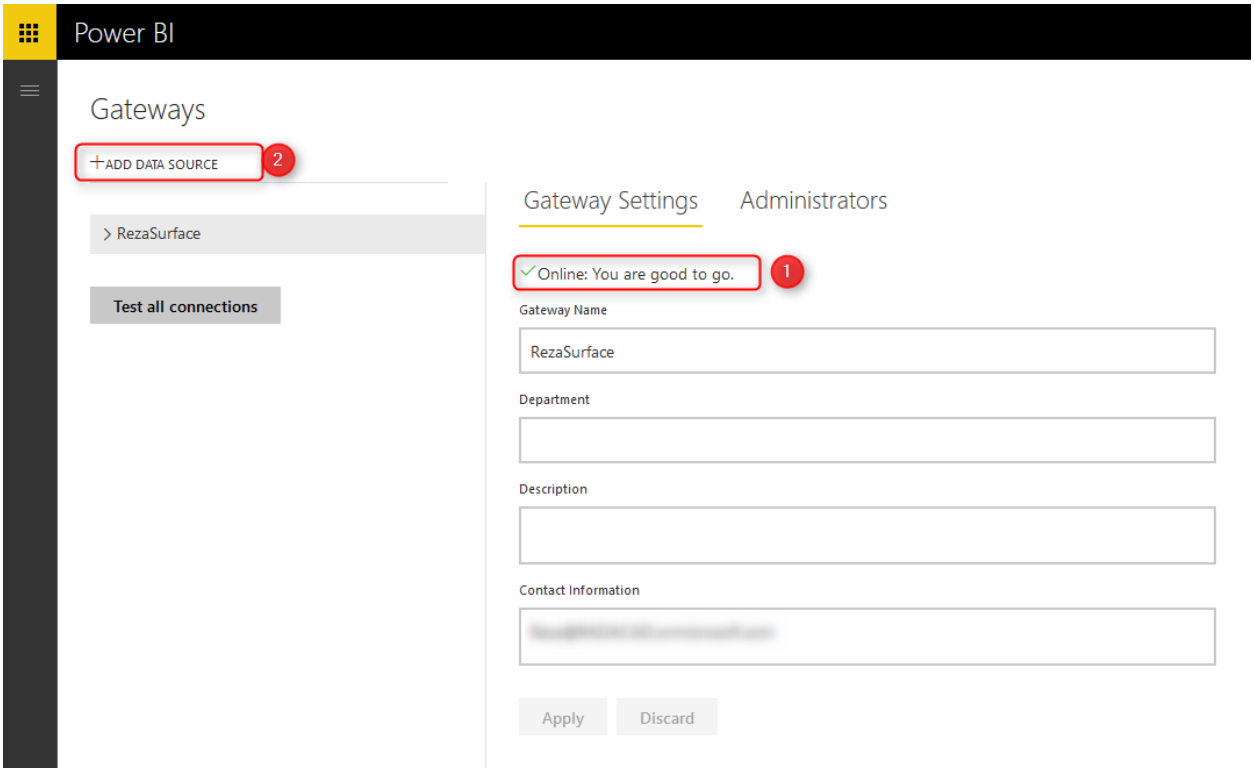


After downloading follow the setup wizard with few steps and you will have Enterprise gateway installed with few clicks. There are some limitations though; You cannot have Enterprise and Personal Gateway on the same machine. I won't talk about their differences, that would be a whole separate post which I will write later. In this post, we only go through an example of using this gateway.

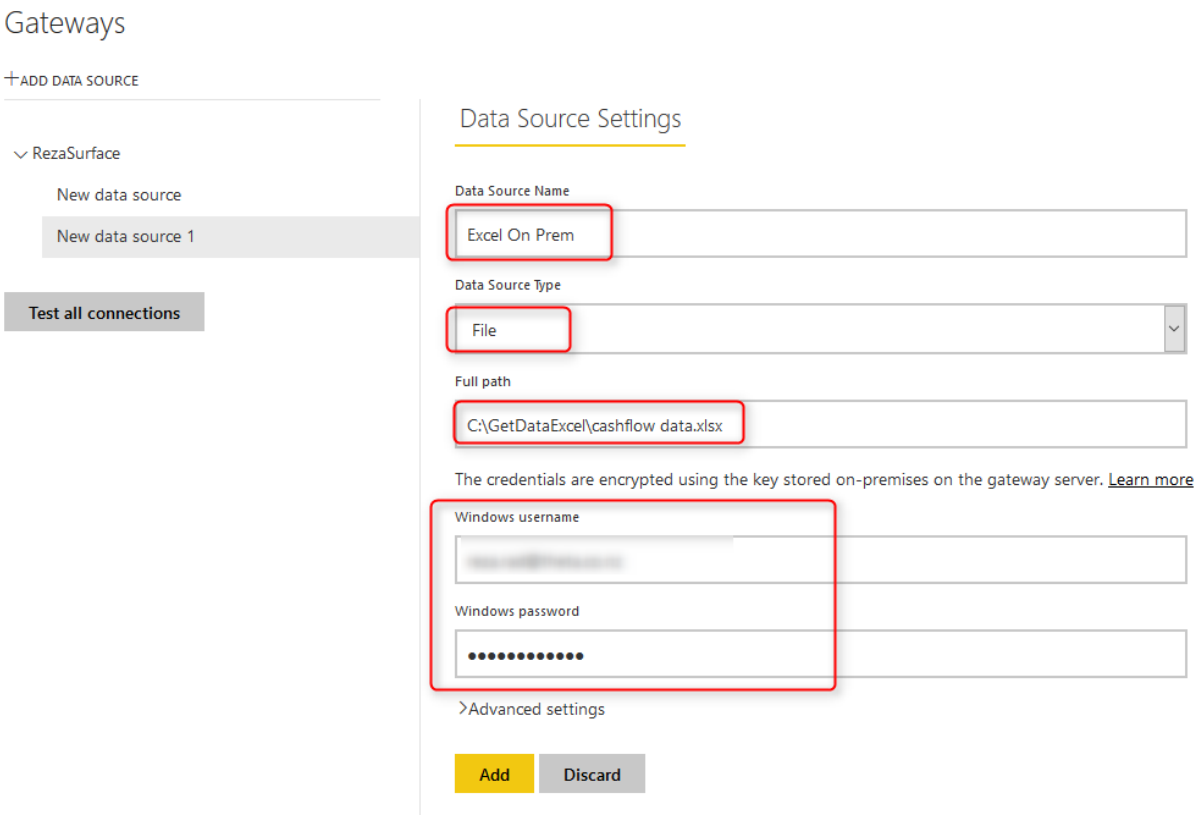
After installation, you can configure your gateway. All you have to do is to sign in to your Power BI account from gateway, set a name for your gateway and a code or recovery key. Then you will be good to go. I've already set up my gateway and named it as RezaSurface. Now I can manage my gateway through Power BI website, through a Setting icon, and then Manage Gateways.



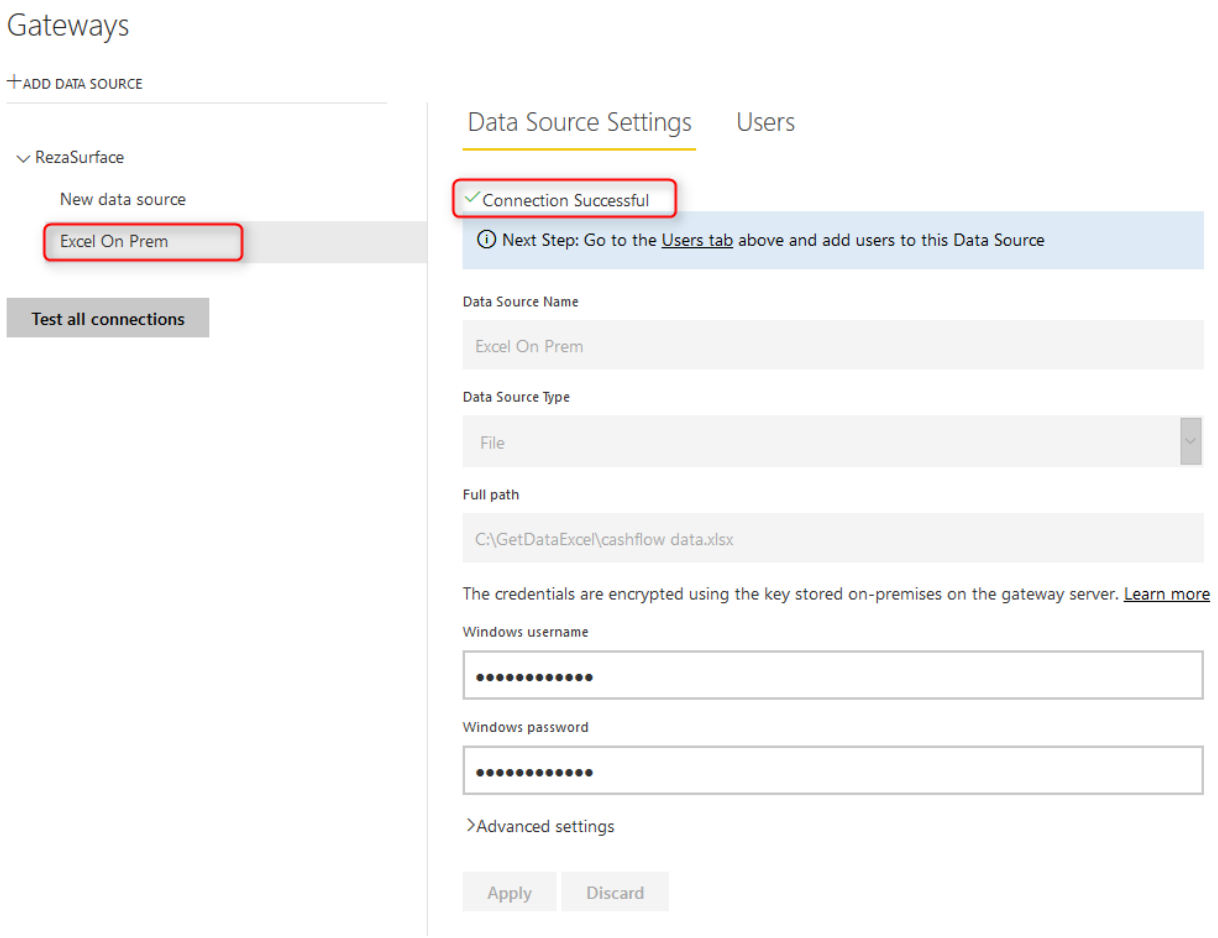
In the Manage Gateways page, I can see the gateway I've configured named RezaSurface. And I can also see that it is online with a green checkbox close to it. Which means everything works correctly.



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 the screenshot above). As you can see in the screenshot below, I name the new data source as Excel On-Prem, and choose data source type as File, and set full path of the file. This should be the local path of the 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.



Note that for this example I will be using the 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.



Deploy Power BI Solution

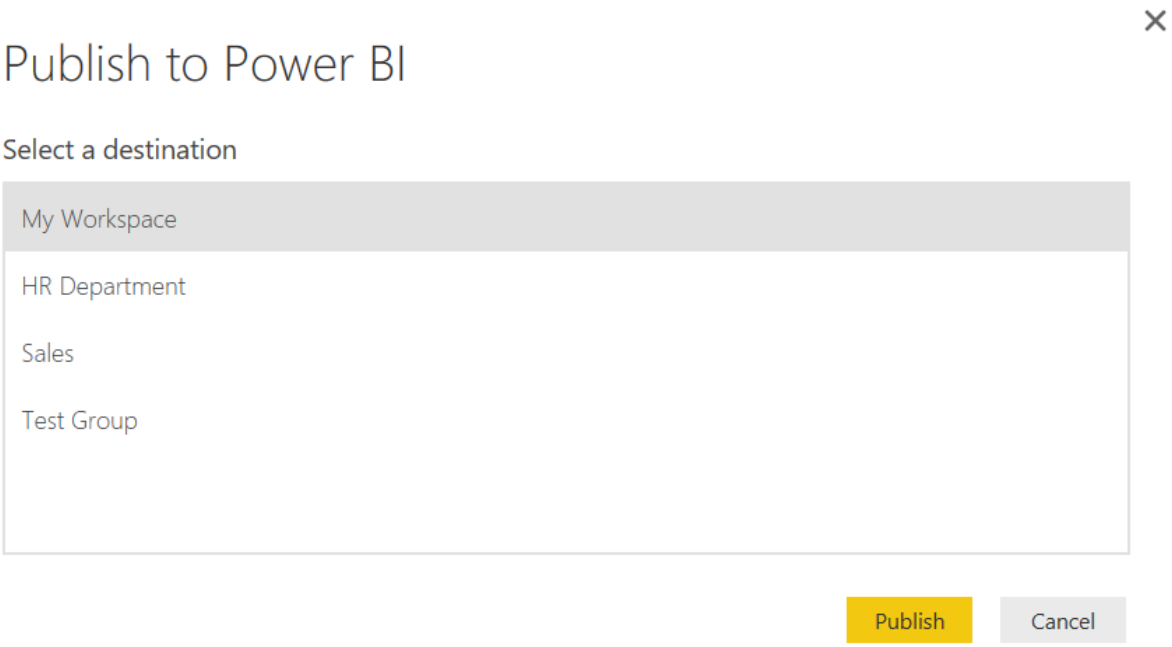
Power BI solution that I have is a simple waterfall chart on a cash flow data table in an Excel source. Here is my sample excel source table:

	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			

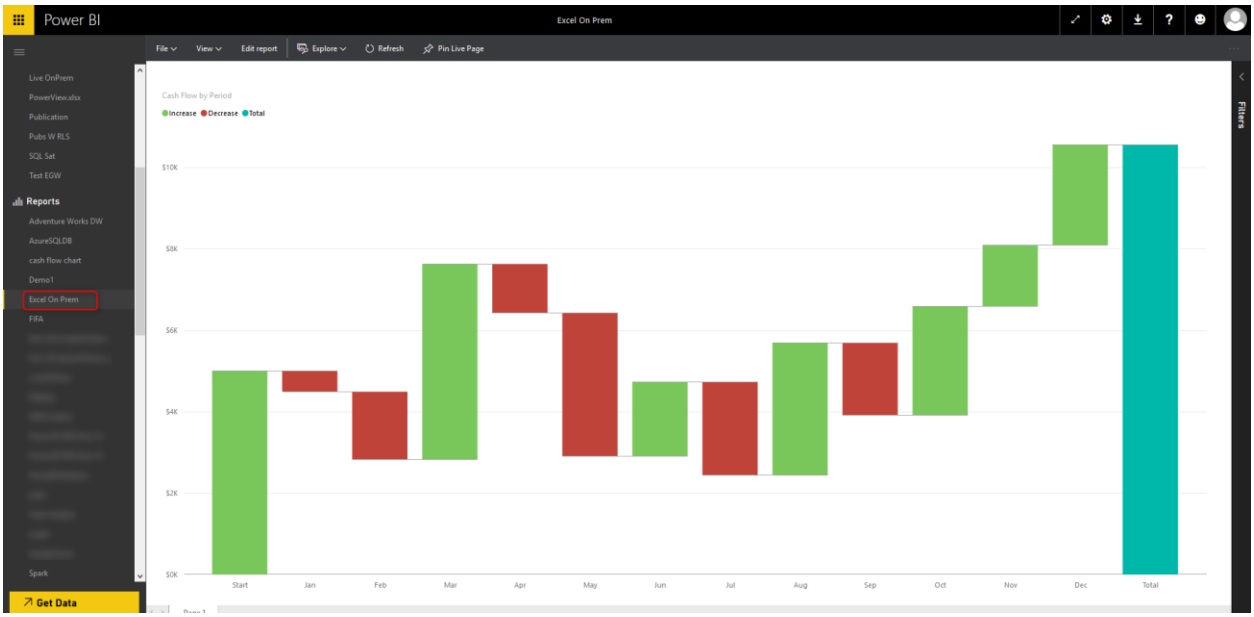
and here is the waterfall chart I've built for the data set above in Power BI Desktop

If you need more details information about how to build report above reading [Waterfall Chart blog post here](#).

I named this Power BI file as Excel On-Prem, and publish it to my workspace in Power BI



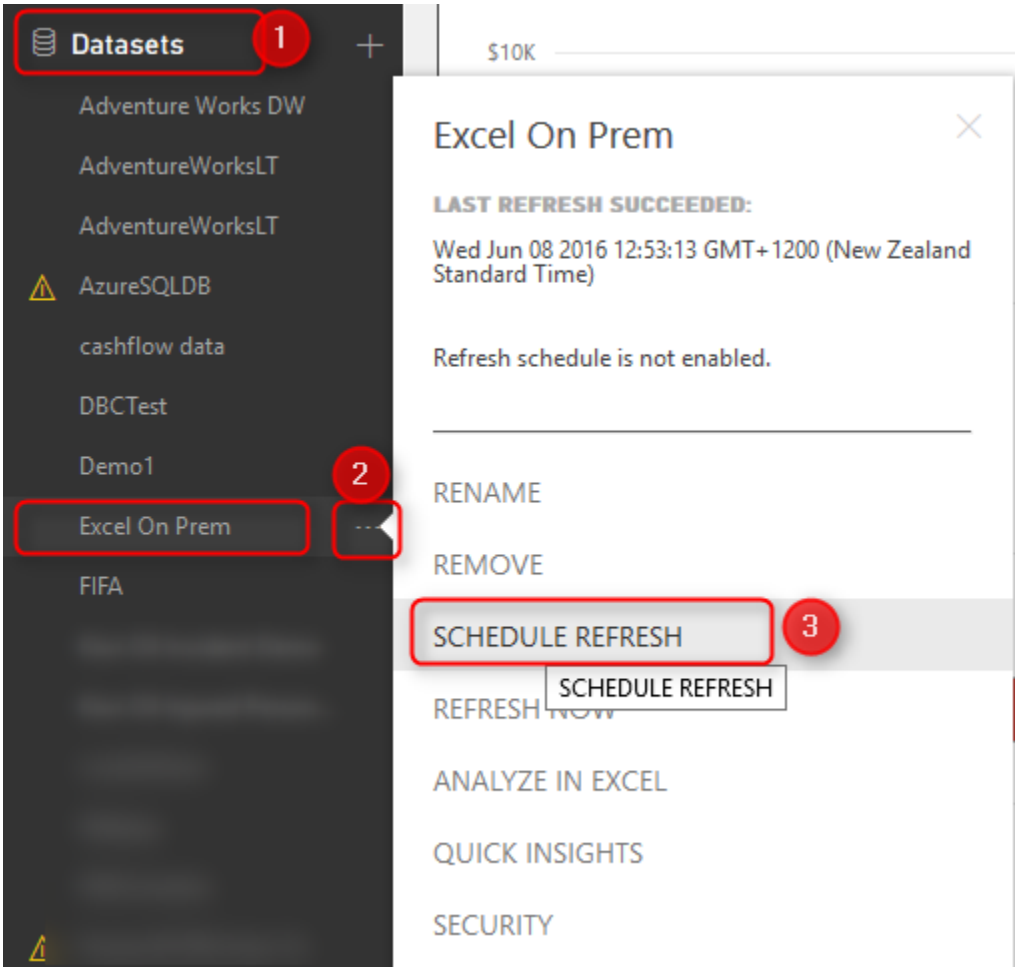
Now I can see my report on Power BI website;



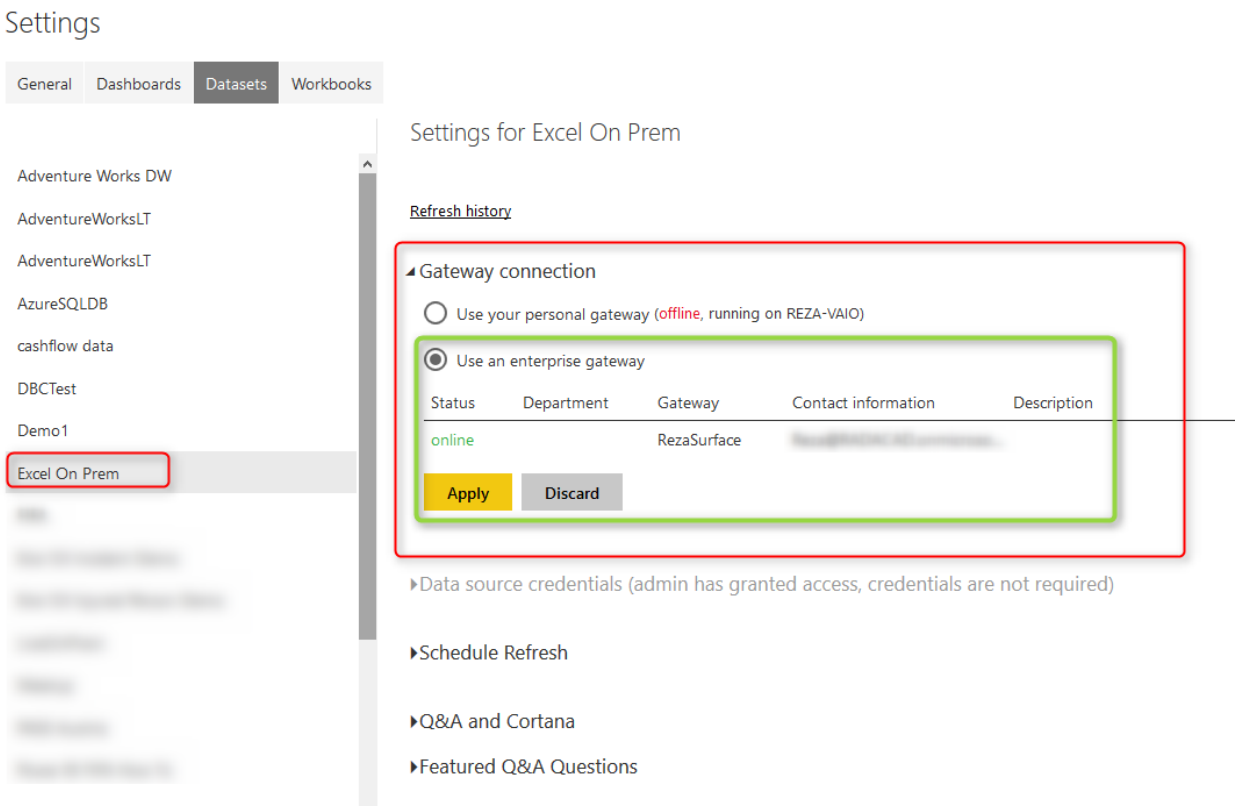
Schedule Refresh

Now that I have my report and dataset published in Power BI service, and a gateway configured with the data source, I can connect these two and schedule refresh.

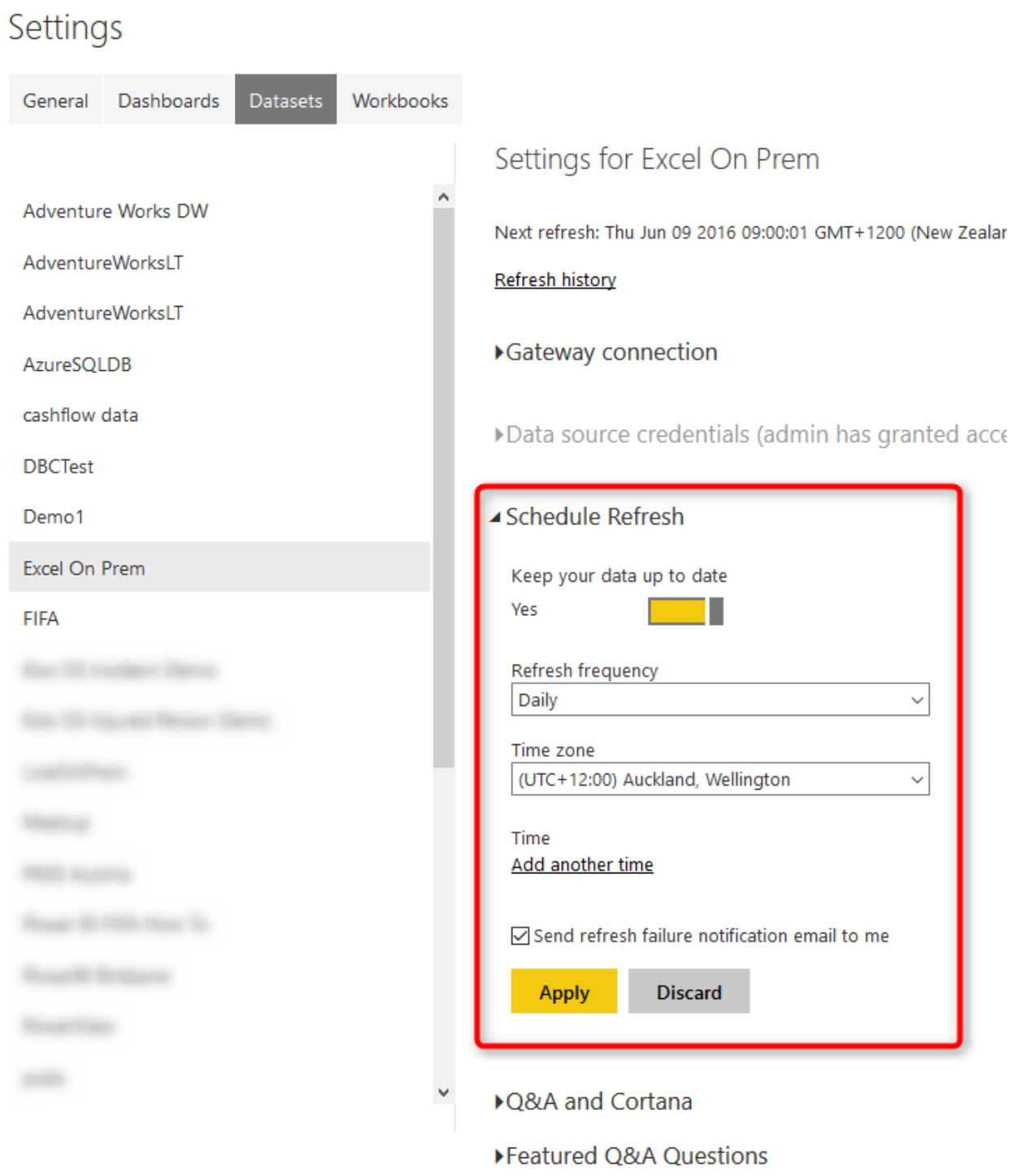
Click on ellipsis button beside the Excel On-Prem dataset and choose Schedule Refresh.



Then I will be redirected to the Settings page where I can configure settings for this data set. As you see in the screenshot below, I expand the 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.



The connection between the data set in my Power BI report and the data source in the gateway is now created. And I can Schedule Refresh based on the schedule I would like daily or weekly…



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 a manual refresh of the data set on Power BI site), I can see the report refreshed as below;

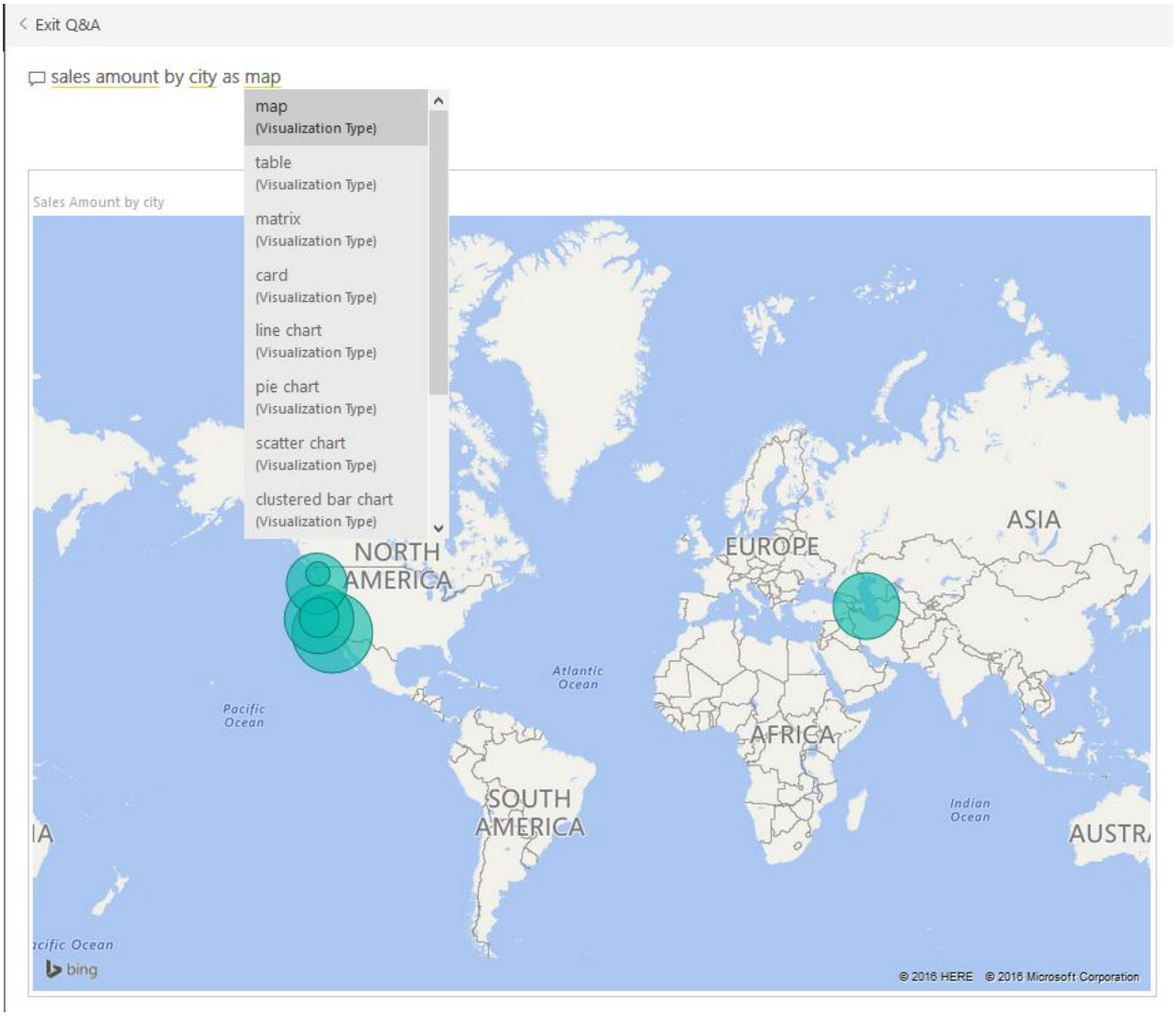
You can see that the waterfall chart starts with $8K which is the new value from the local Excel file.

Summary

In this post you've learned how to use the gateway to create a connection between the 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 to any files, it shouldn't be only Excel files. There are also options for using a folder as a source which is useful when you have multiple files with the same structure in a folder on-premises. Think about ways that this functionality can help your solutions.

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 the dashboard is interactive visualizations with the 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 the 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 in learning more about Power BI; read Power BI online book from Rookie to Rock Star.

Definition

Dashboard: General

Stephen Few's definitions 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 a 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 . 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. The 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 this 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 are 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 the other month? Or Which product is selling best, and how it is distributed through branches. Here is an example of a Power BI report;

If you like to know how to create this report, please read this post.

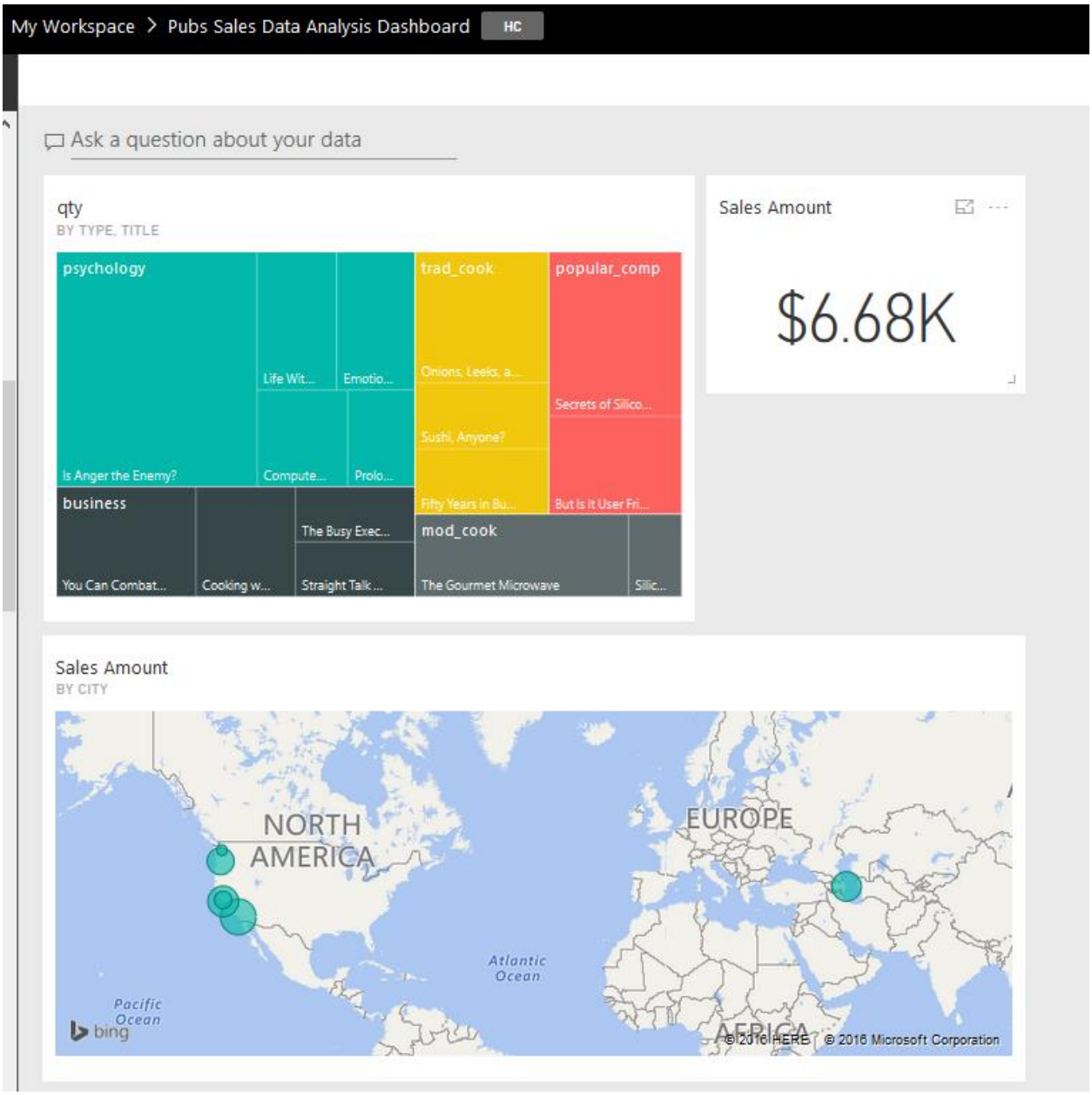
Unique Features of a Power BI Report

- Slicers
- Multi-Pages
- Interactivity
- Drill Down/Up
- Publish to Web
- 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

5. Automatic Refresh
6. Sharing
7. Customization by User is Easy
8. Featured Dashboard
9. Power Q&A

10. Real-time Monitoring
11. Alerts
12. Related Insights

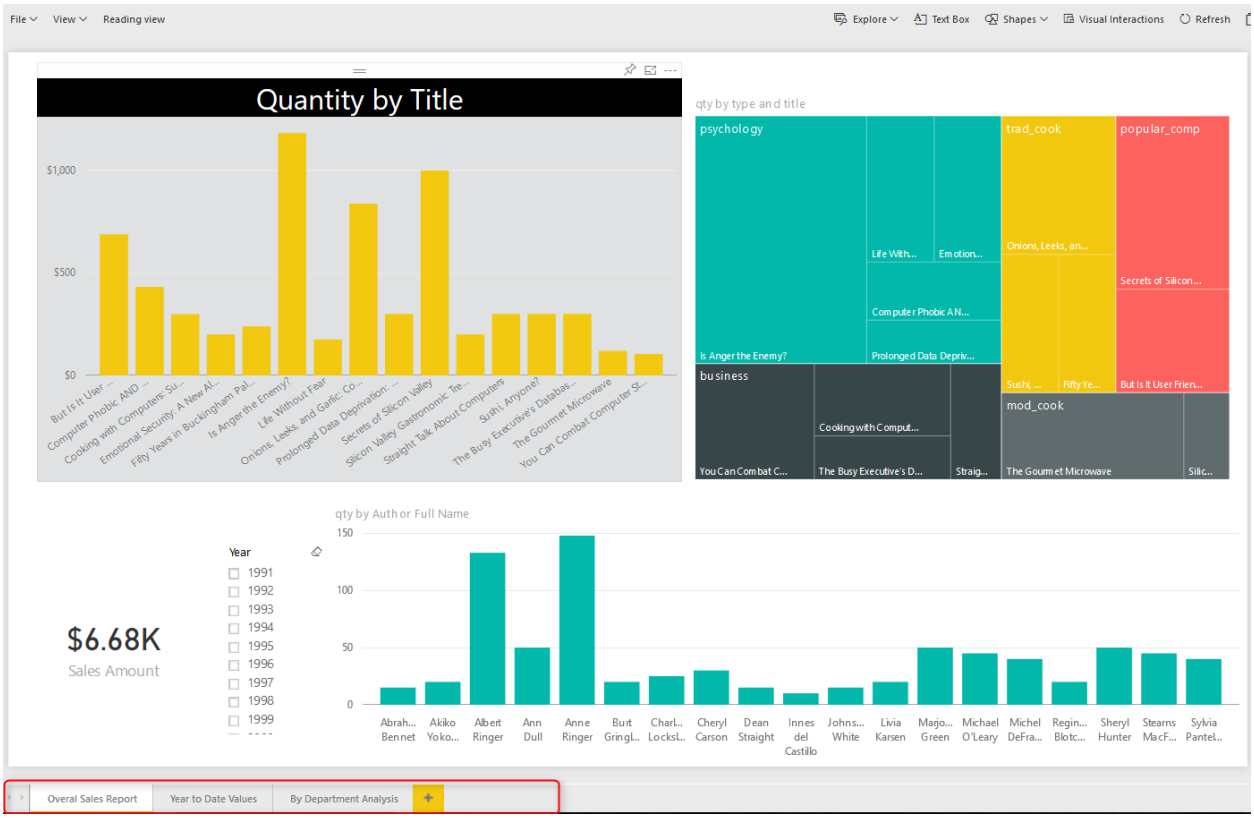
Unique Features of a Power BI Report

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)

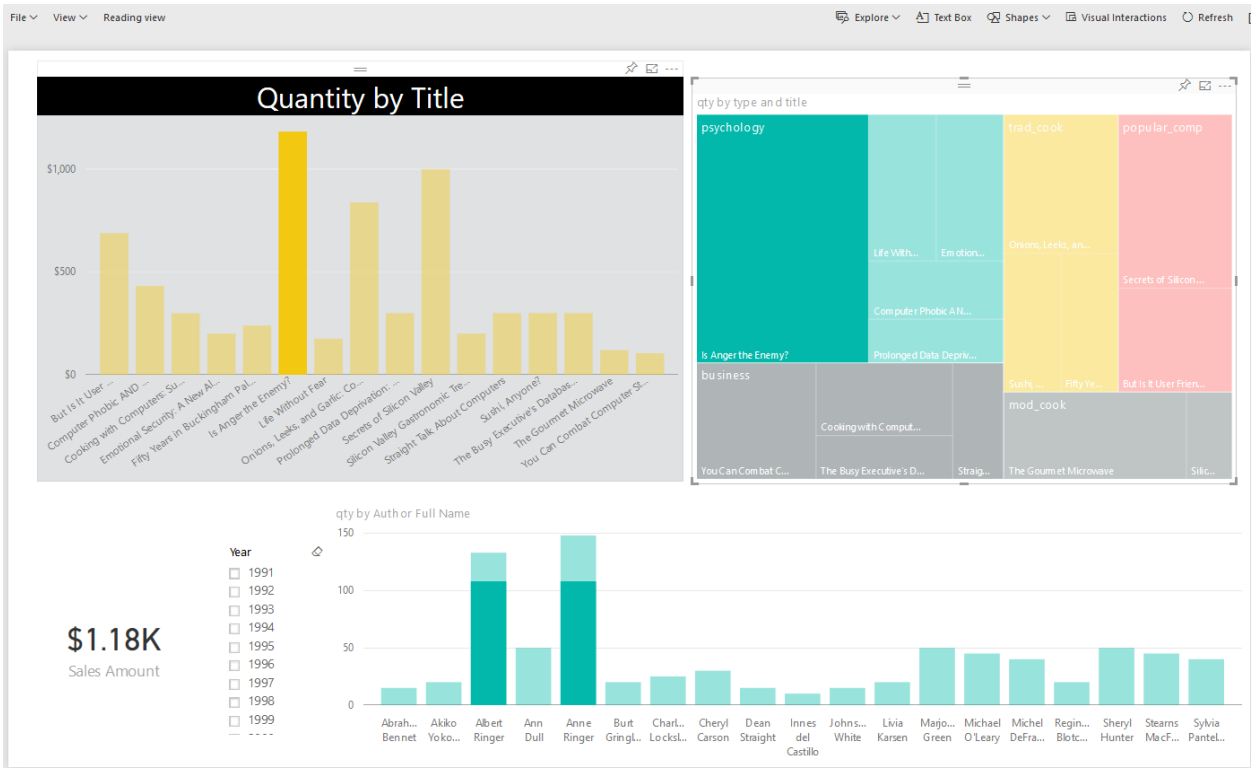
Multi-Pages

You can have multiple pages in a report. Navigation between pages is possible through the navigation pane at the bottom of the report. In the dashboard you can have as many as tiles you want. But there is no concept of pages. everything is on one page, If content doesn't fit into one page, scrollbars 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. The user can simply interact with report elements and get more insight from it. In the dashboard with clicking on a tile, you will be redirected to the report/page that built the element.

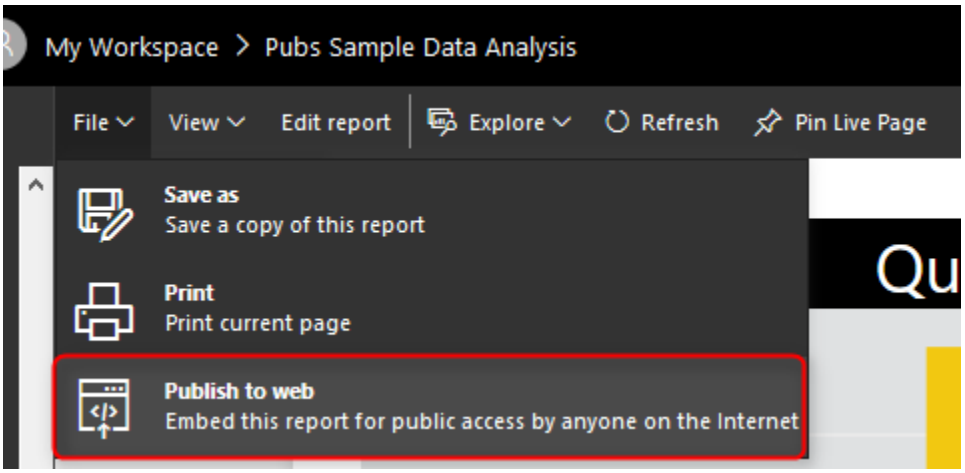


Drill Down/Up

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 the 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.

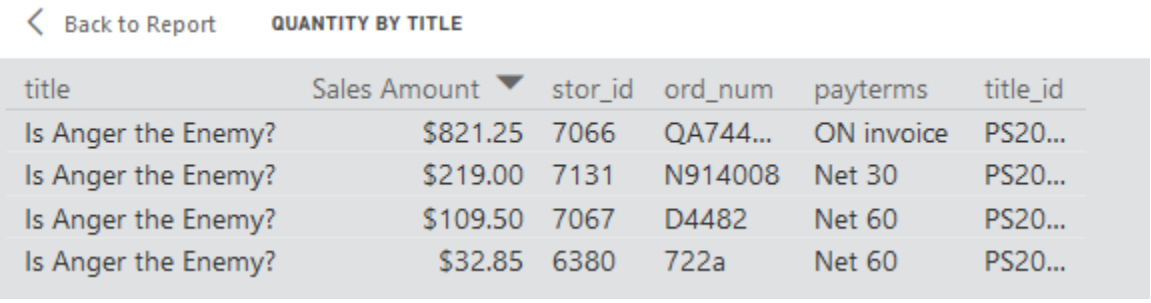
Publish to Web

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 particular value in a chart. In dashboard, only the data of dashboard (not data records) can be exported.



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 gets refreshed, you need to refresh your report manually otherwise your report won't be refreshed automatically.

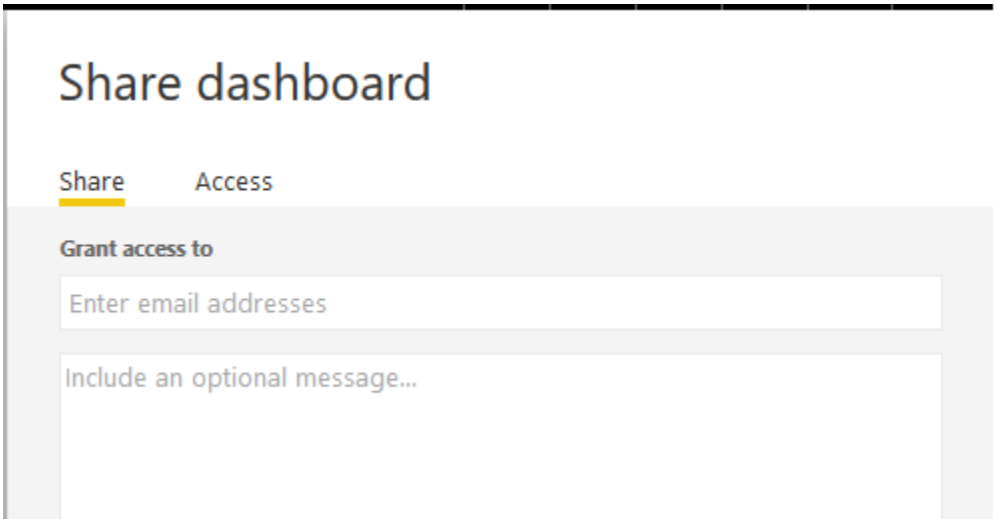
Well when I get to this point, many students in my courses asks why this behaves like that?! Why it is not getting refreshed automatically like the dashboard. I believe the main reason for this is scenarios like this: Consider that an analyst has opened an inventory report, and is checking a number of 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 is 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.

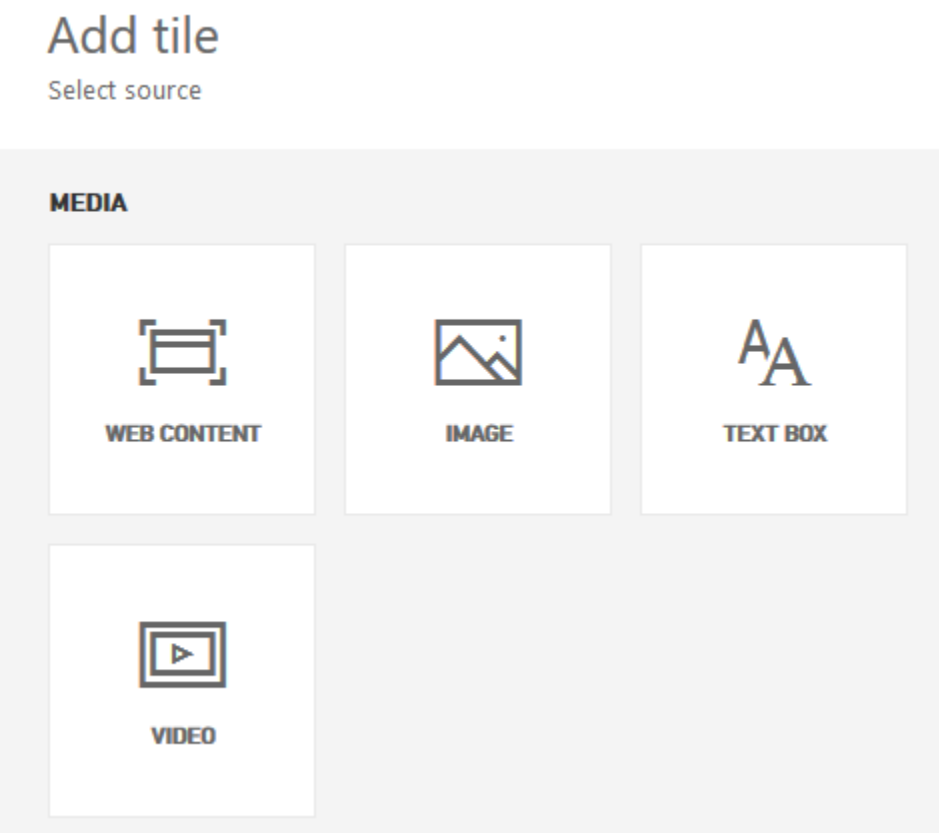
Sharing

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.



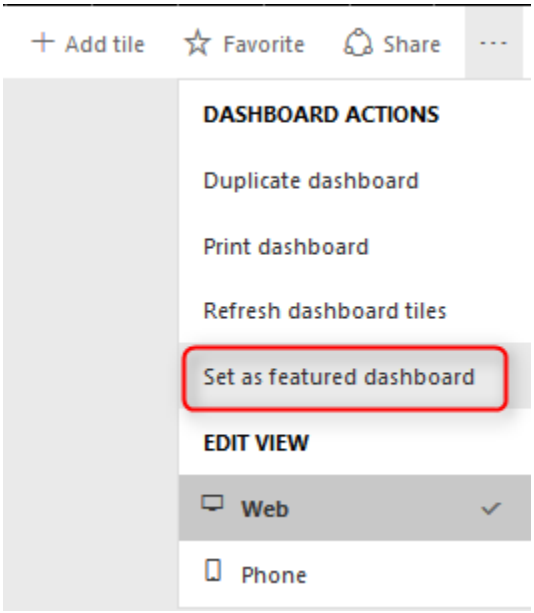
Customization by User is Easy

It is easy to change the size of dashboard tiles, and change order of them, also to add new tiles (Image, Text, Web, and Video) in the 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. The dashboard is like a user's view of the world, so simply can be adjusted to what he/she wants to see.



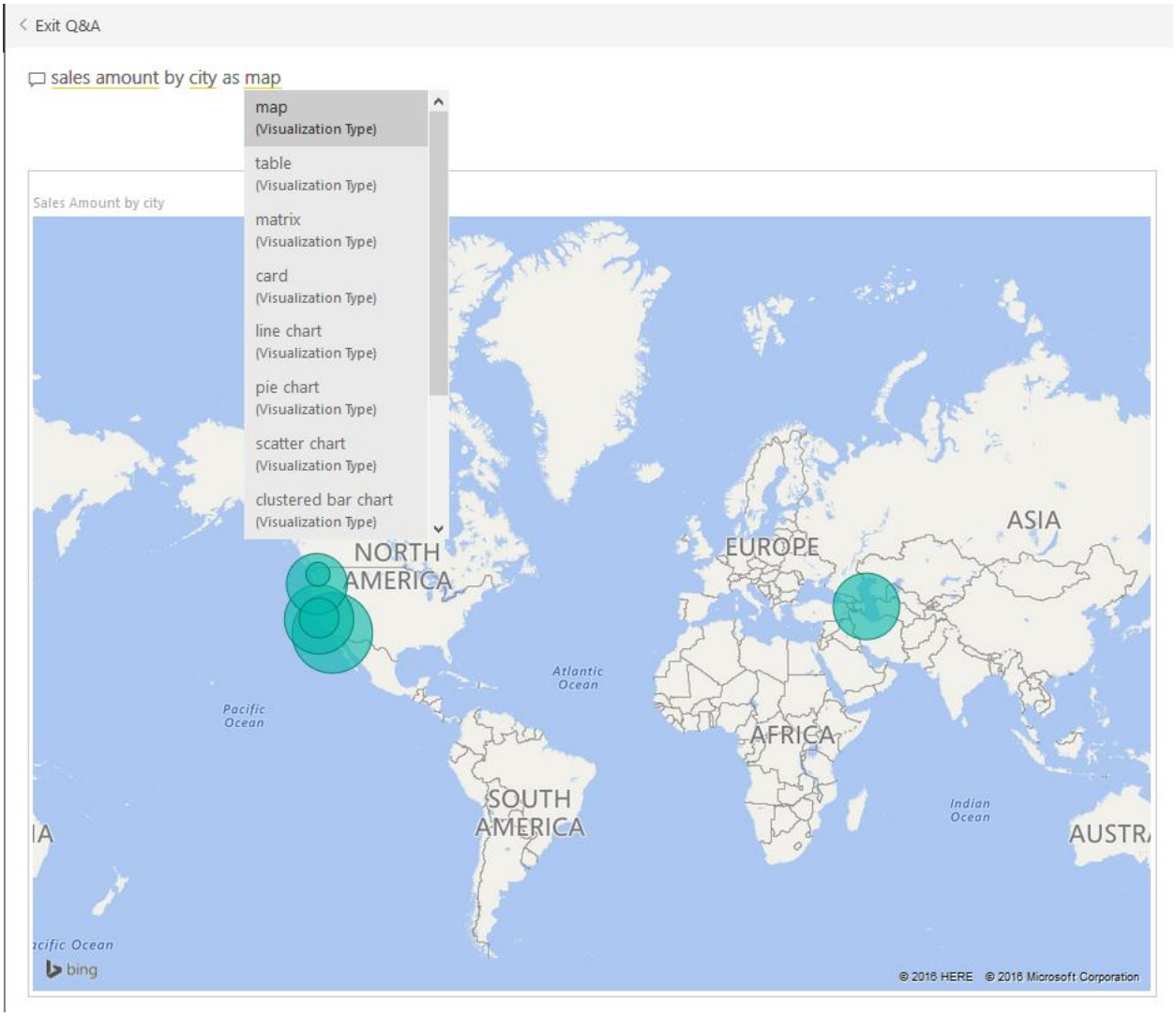
Featured Dashboard

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 previously featured dashboard will be replaced with this new one.



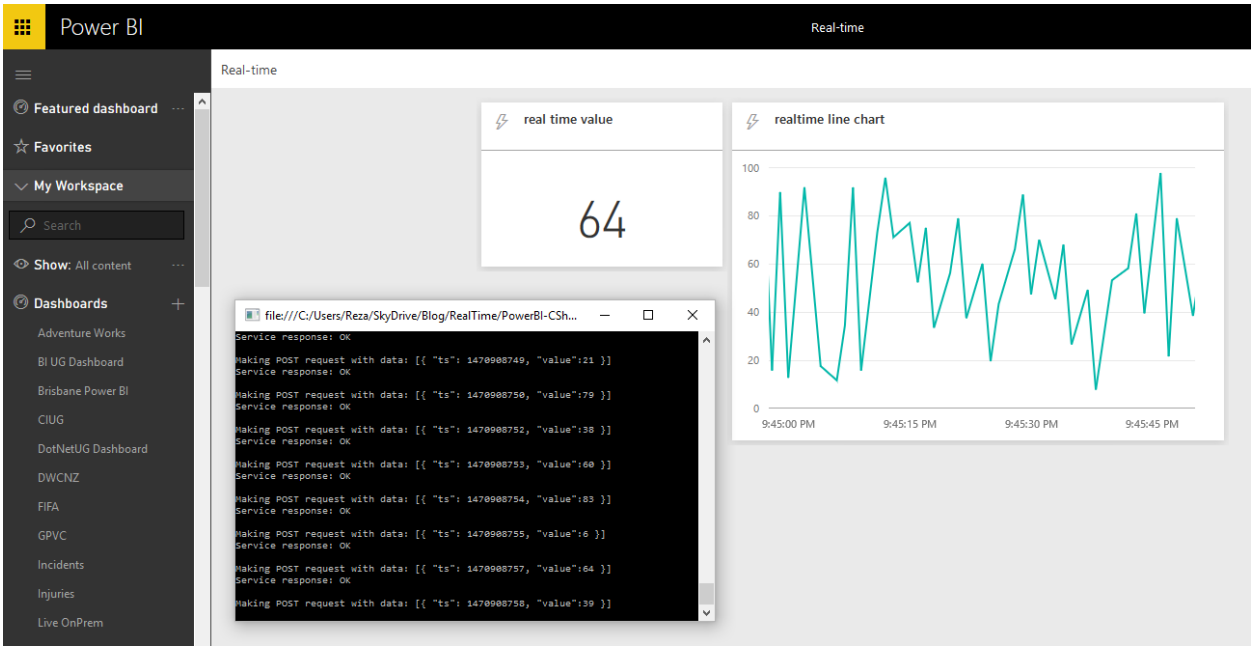
Power Q&A

If you haven't heard about Q&A, I encourage you to read this blog post. Q&A is an engine of the 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).



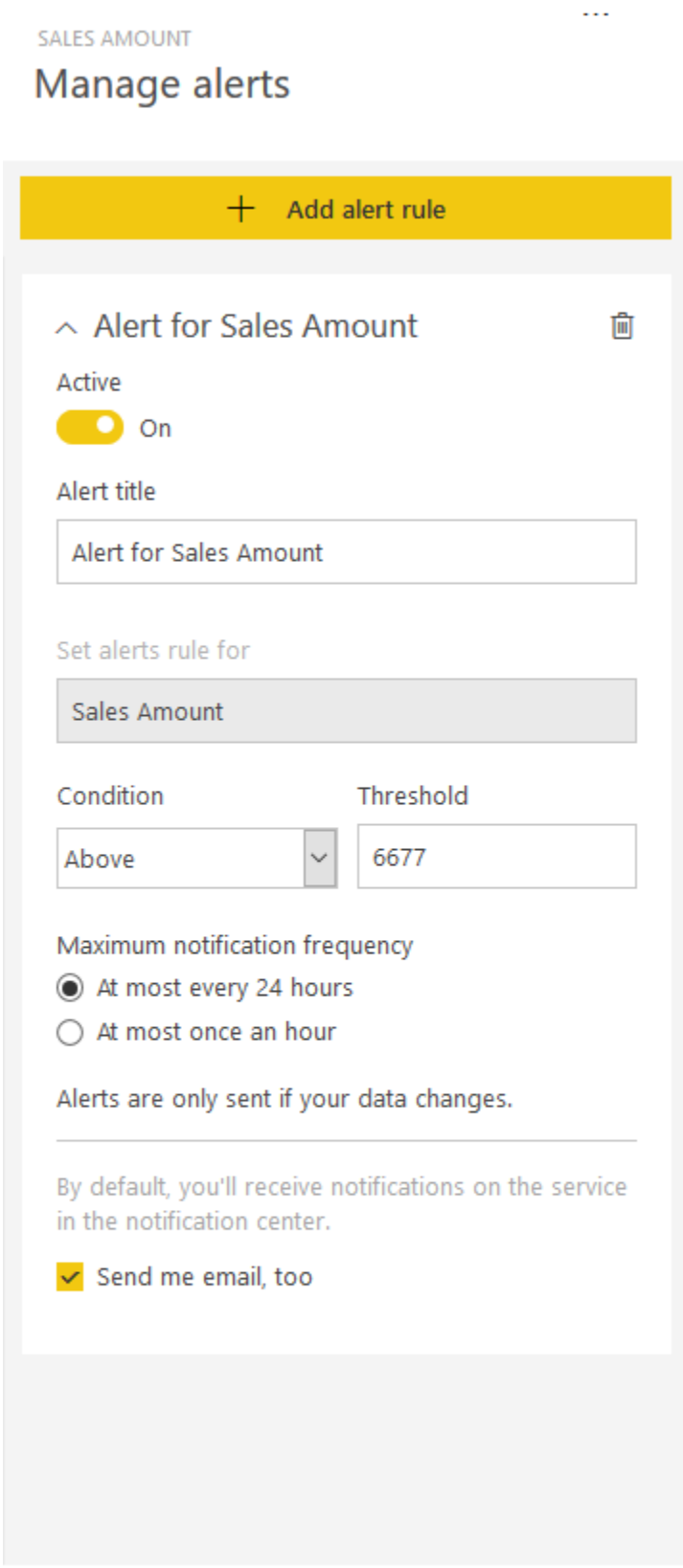
Real-time Monitoring

I've mentioned that dashboards refresh 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. Having real-time monitoring of events with Power BI Dashboards is possible. The dataset 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.



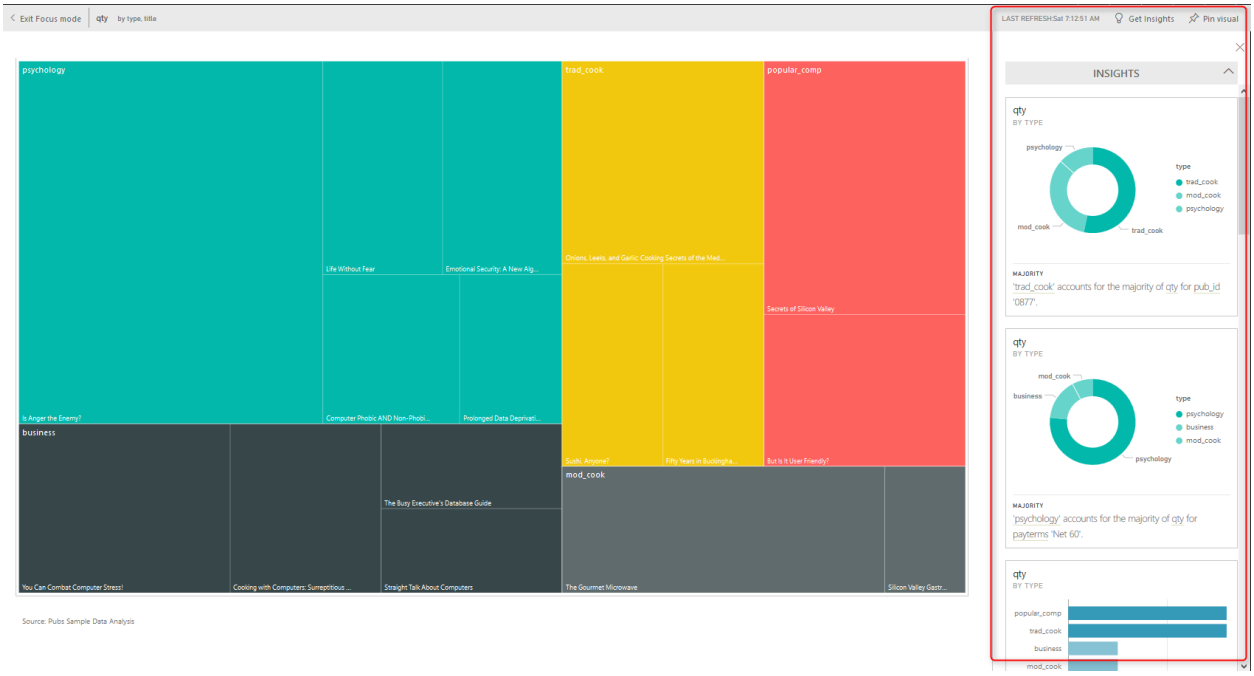
Alerts

You can define alerts for each data driven tile in the report. Alerts can be as simple as if the 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
Threshold
Above
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.

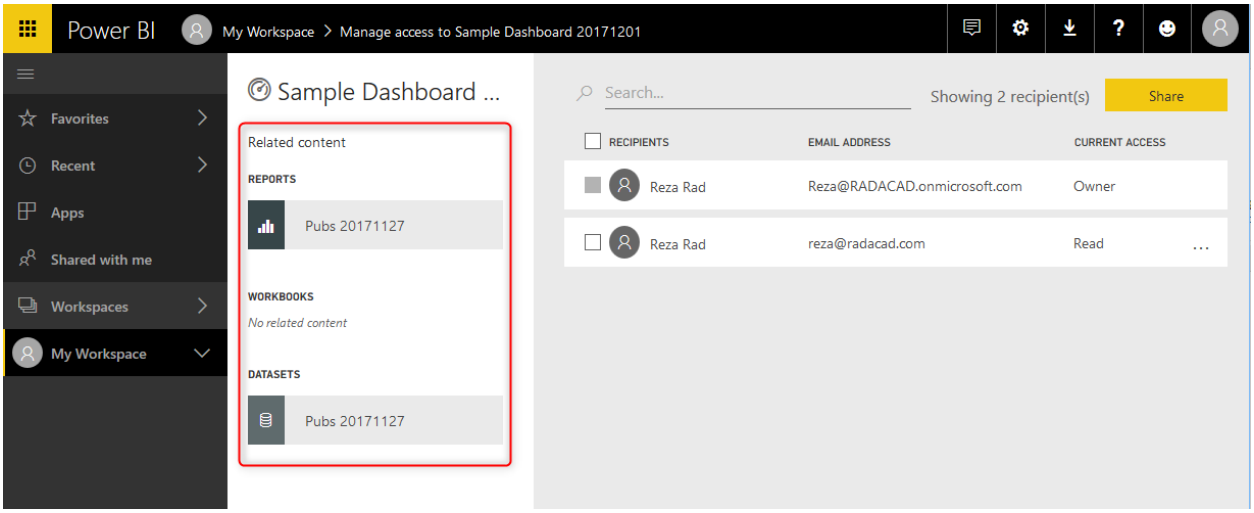


Differences At a Glance

As you see dashboards and reports are not 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. The 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 compliments of each other, not replacements.

Dashboard Sharing, and Manage Permissions in Power BI; Simple, but Useful?

Published Date: December 21, 2017



Power BI provides multiple ways of sharing the content with users. Each sharing method has pros and cons and can be used for specific scenarios. Some of the sharing methods can be used together to build a framework for sharing. In this post, I will talk about the most basic way of sharing Power BI content. This method is called Dashboard Sharing. Dashboard sharing is the easiest way of sharing; however, it is always the best way of sharing. In this post, you'll learn how this method works, you will learn about the pros and cons of this method, and scenarios of using it.

Power BI Content Owner

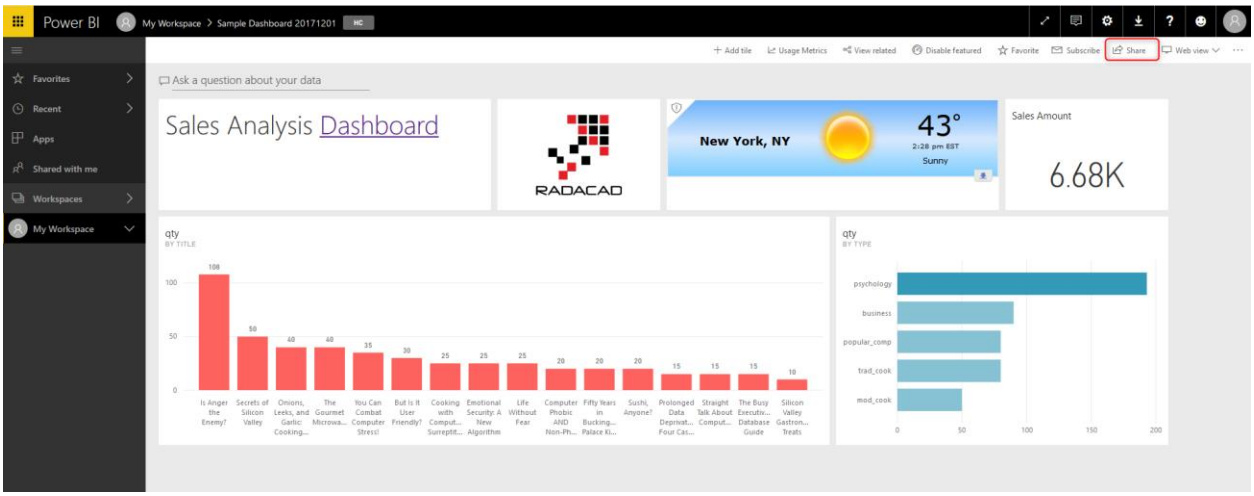
Before going through the dashboard sharing, you need to understand the content security in Power BI. When you publish a *.pbix report into Power BI website, especially when you publish it under "My workspace", no one else will see or have access to your report. It would be only you who has access to it. Then you can decide whom you want to share this report.

Every Power BI content (report, dashboard, or dataset) has an owner; content owner is the person who created and published that content into Power BI. The owner has full

access to the content of Power BI. One of the accesses that the owner has is to share the content with others.

How Dashboard Sharing works?

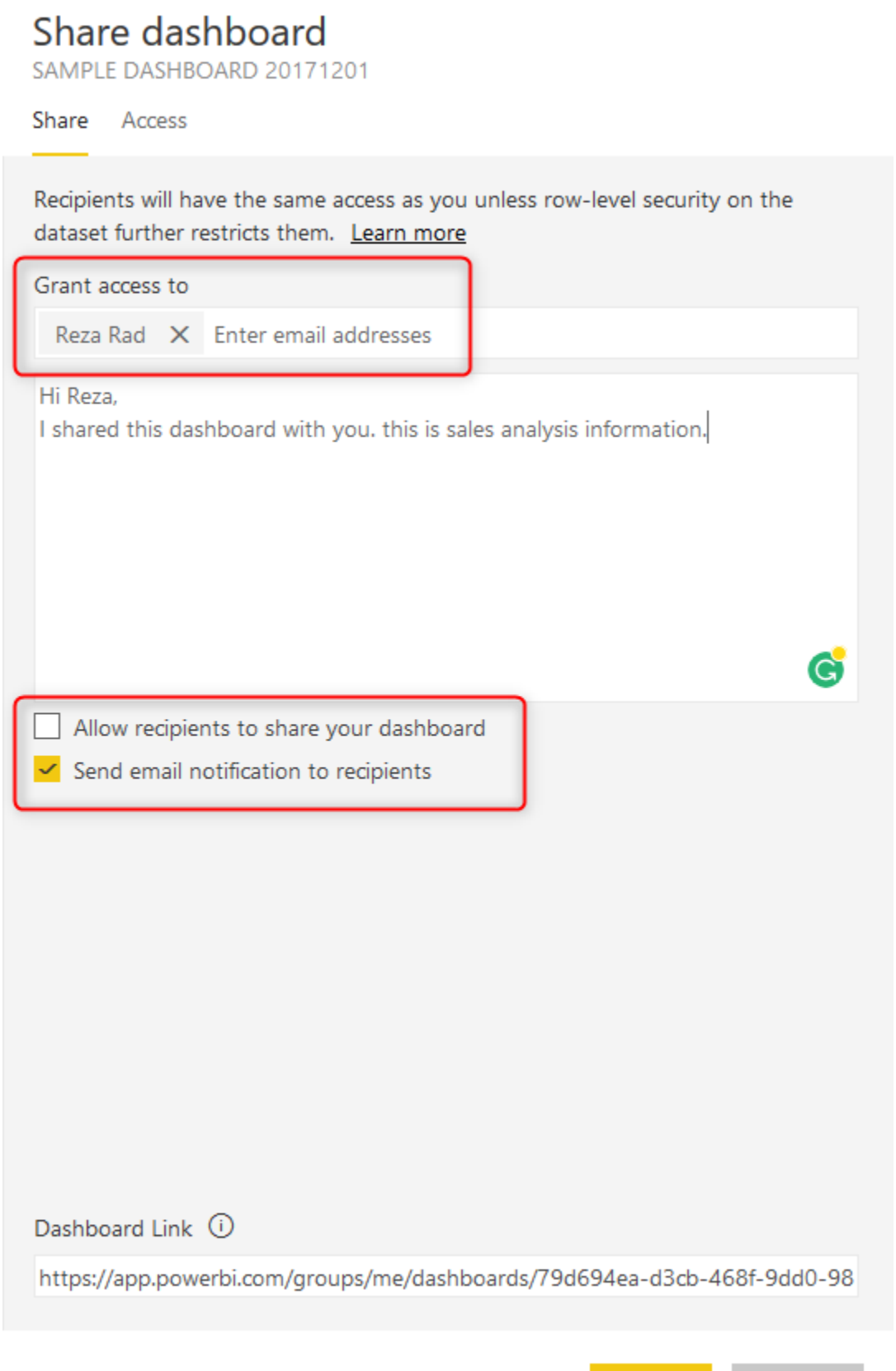
What dashboard sharing as the name of it explains is based on a dashboard. You can only share a dashboard with this method, not a report. Consider that you have a dashboard like below screenshot, and you want to share it. There is a share link at the top right corner of the dashboard.



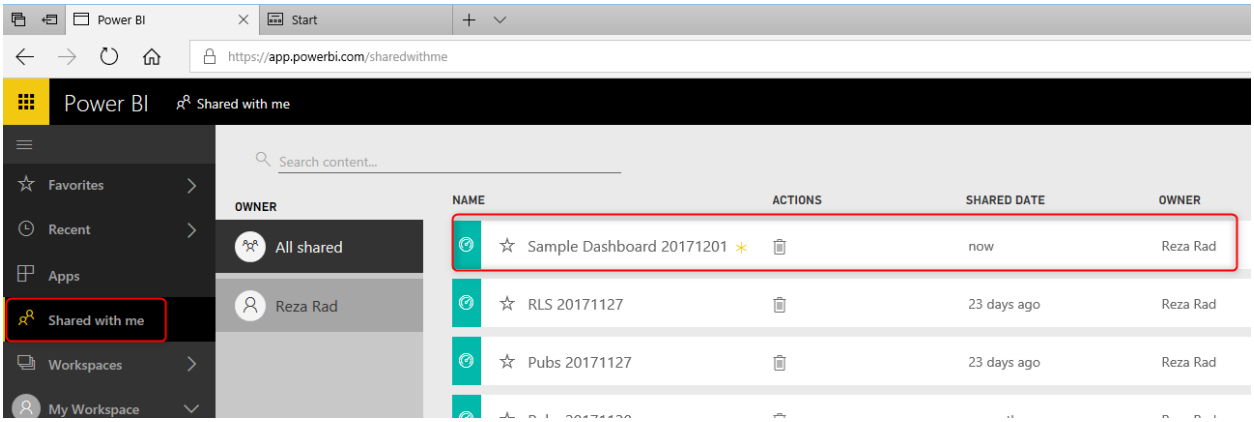
Dashboard sharing have very few options to set and is very simple to configure. You need to add the email address of people whom you want to share this report. You can also write a message for them to know that this report is shared with them. There are two options to set;

- Allow recipients to share your dashboard
- Send email notification to recipients

You can decide if people that you shared this dashboard with are also allowed to share it with others or not? And you can also choose if you want them to receive an email notification when you shared the dashboard with them or not.


Share dashboard
SAMPLE DASHBOARD 20171201
Share
Access
Recipients will have the same access as you unless row-level security on the dataset further restricts them. Learn more
Grant access to
Reza Rad
Enter email addresses
Hi Reza,
I shared this dashboard with you. this is sales analysis information.
Allow recipients to share your dashboard
Send email notification to recipients
Dashboard Link
https://app.powerbi.com/groups/me/dashboards/79d694ea-d3cb-468f-9dd0-98
Share
Cancel

After configuration, then you can click on the Share button. The recipient will immediately have access to the report. If you selected “Send email notification to recipients”, they will receive an email. Otherwise, they get a notification in Power BI itself. When they login to the service (http://powerbi.microsoft.com), they can find this dashboard under “Shared with me” section.

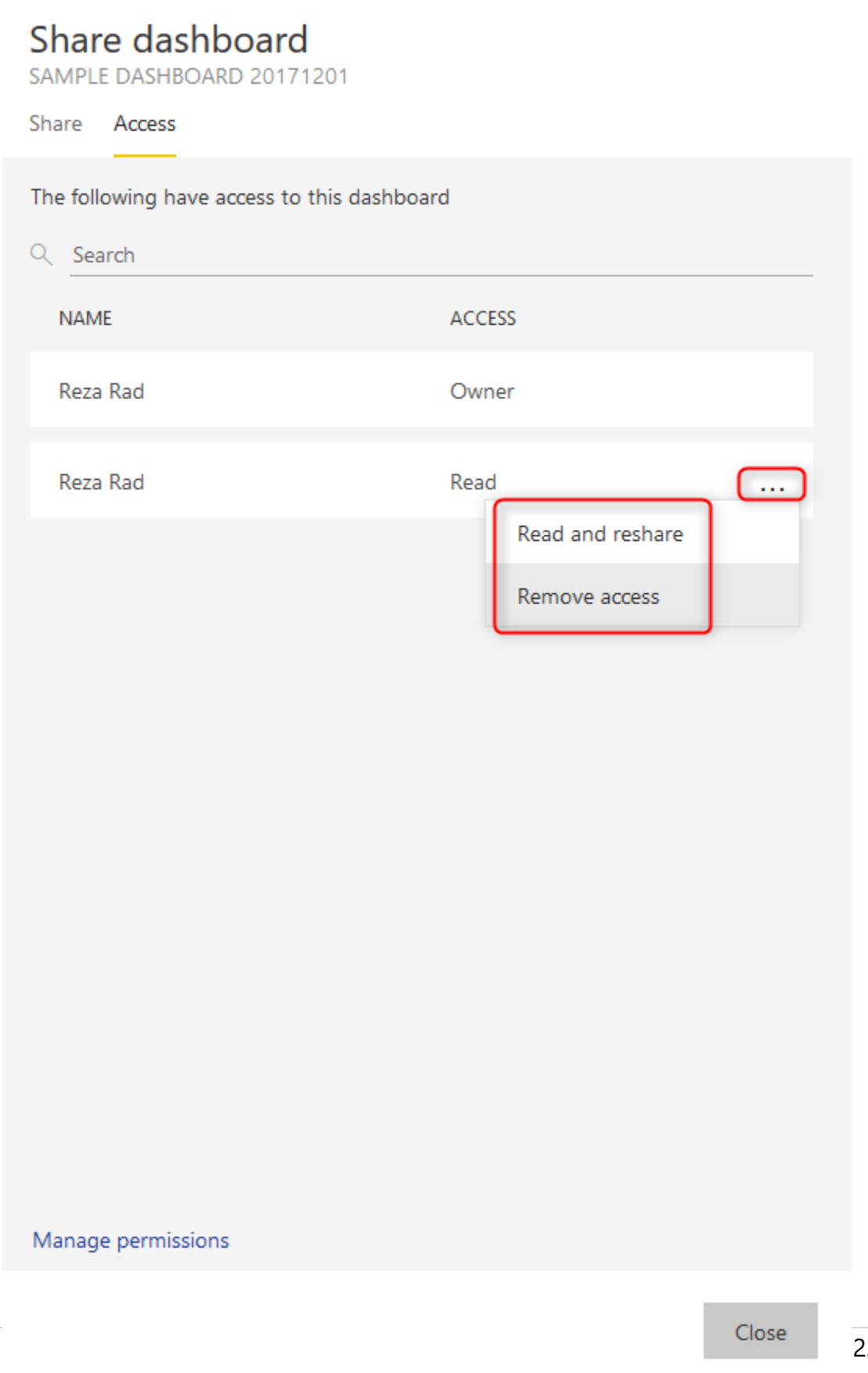


The recipient can click on the dashboard to view it.

Two Levels of access

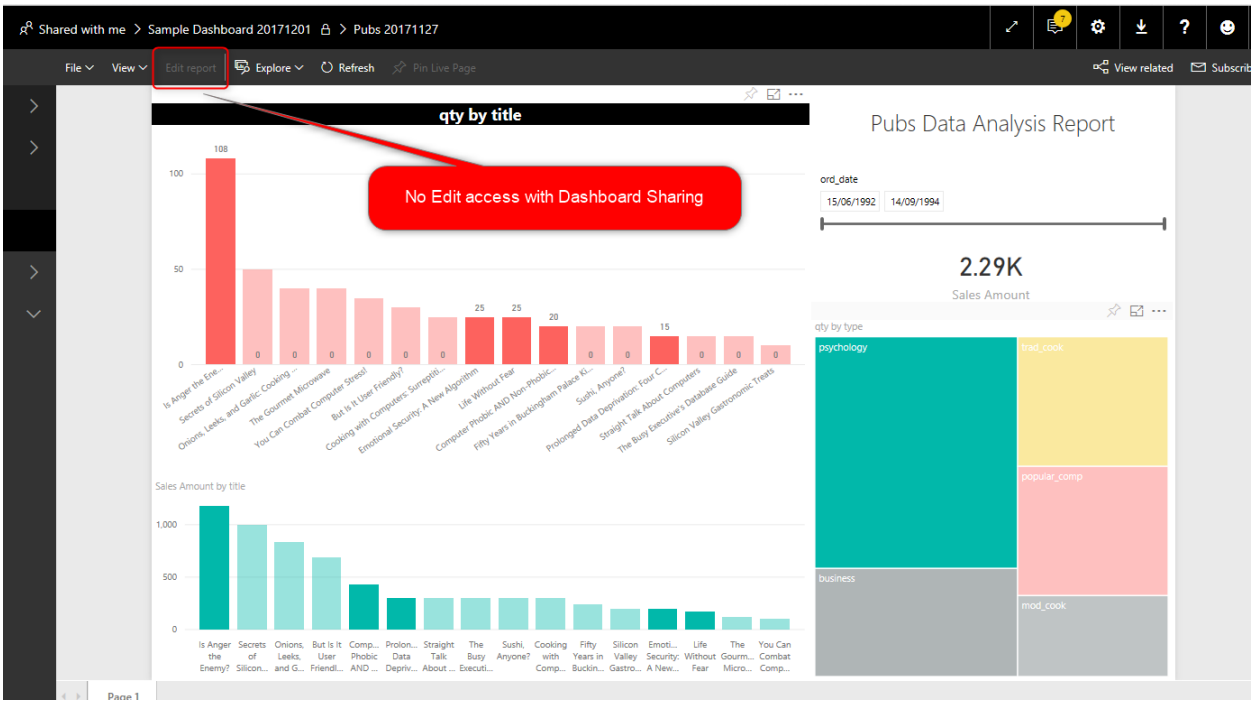
With dashboard sharing, users will have two levels of access; Read, or Read and reshare. If you give them access, without selecting the option “Allow recipients to share your dashboard”, then this access is Read. If you choose the option mentioned above, then the access is Read and reshare.

You can also remove this access anytime you want, by going to the Share option in the same dashboard, and click on the Access tab. You will see a list of all users who have access to this dashboard, and their access level (Owner, Read, Read and reshare), and then you can click on more option (…) and change it.

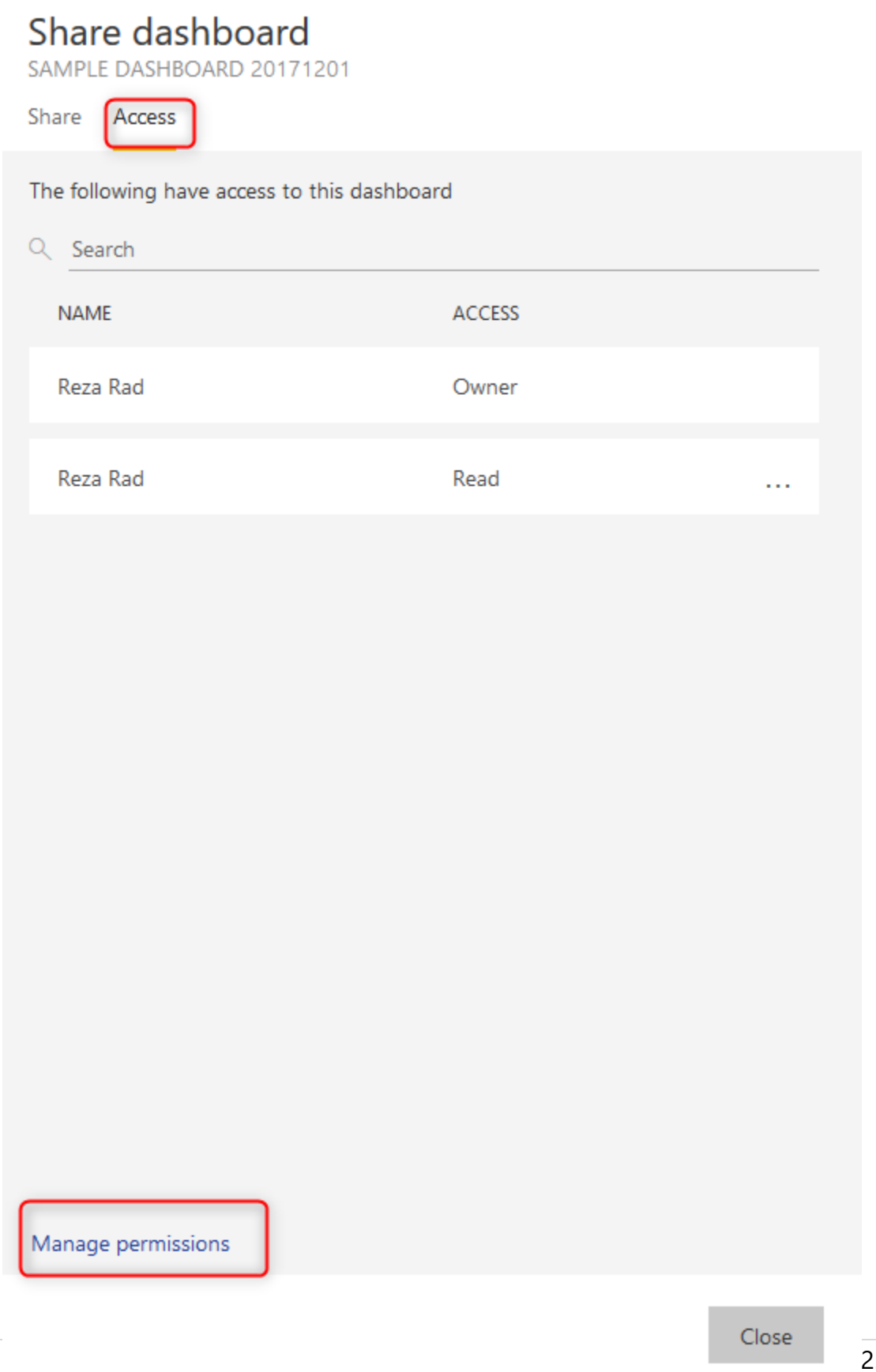

Share dashboard
SAMPLE DASHBOARD 20171201
Share
Access
The following have access to this dashboard
Search
NAME
ACCESS
Reza Rad
Owner
Reza Rad
Read
...
Read and reshare
Remove access
Manage permissions
Close

Manage Permissions

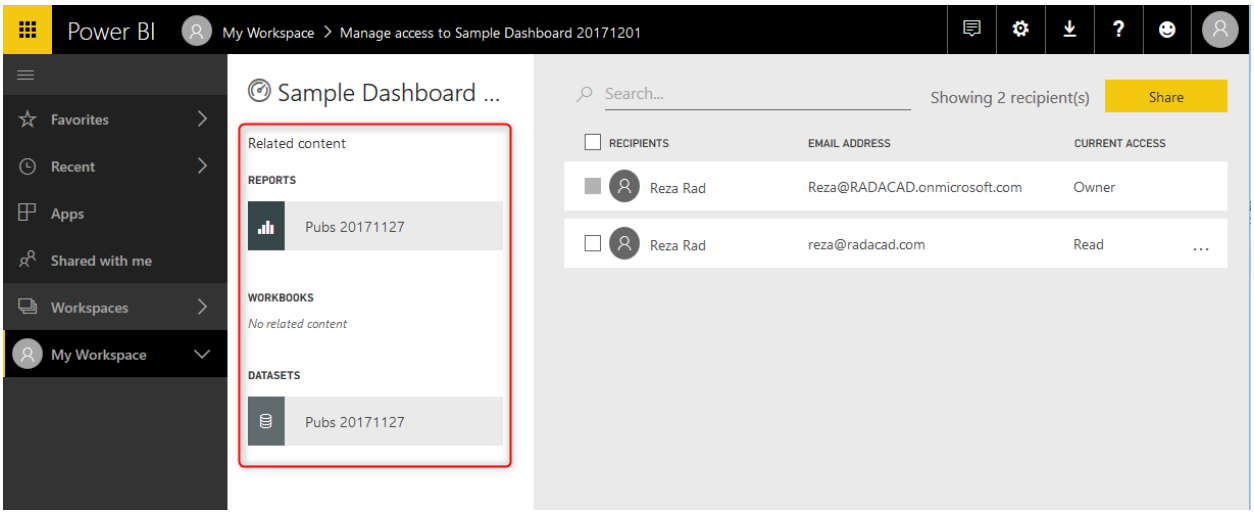
Another way of setting access is through manage permission in the dashboard, report, or dataset. If you share a dashboard, by default the report and the dataset will also be shared as read-only for users. Users can click on the dashboard and go to the report; they can interact with the report quickly. However, they cannot Edit the report. The access to edit report cannot be provided through this method.



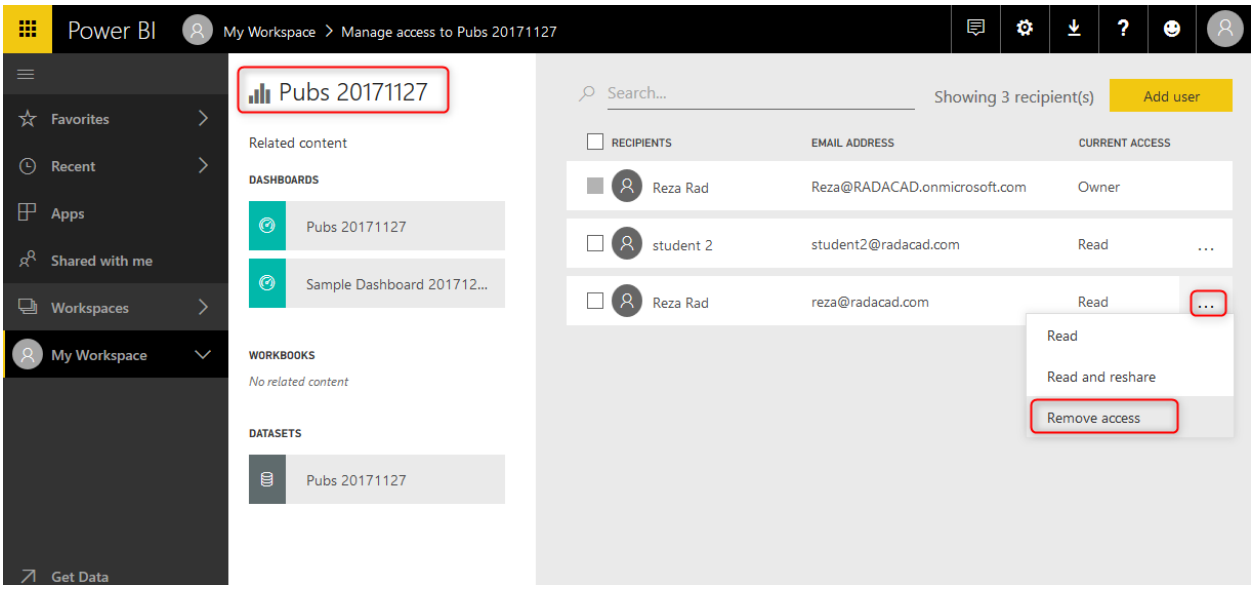
To manage permission on every item (dashboard, report, or dataset) individually, you can go to Manage Permission in the Access tab of Share window for the dashboard.


Share dashboard
SAMPLE DASHBOARD 20171201
Share
Access
The following have access to this dashboard
Search
NAME
ACCESS
Reza Rad
Owner
Reza Rad
Read
...
Manage permissions
Close

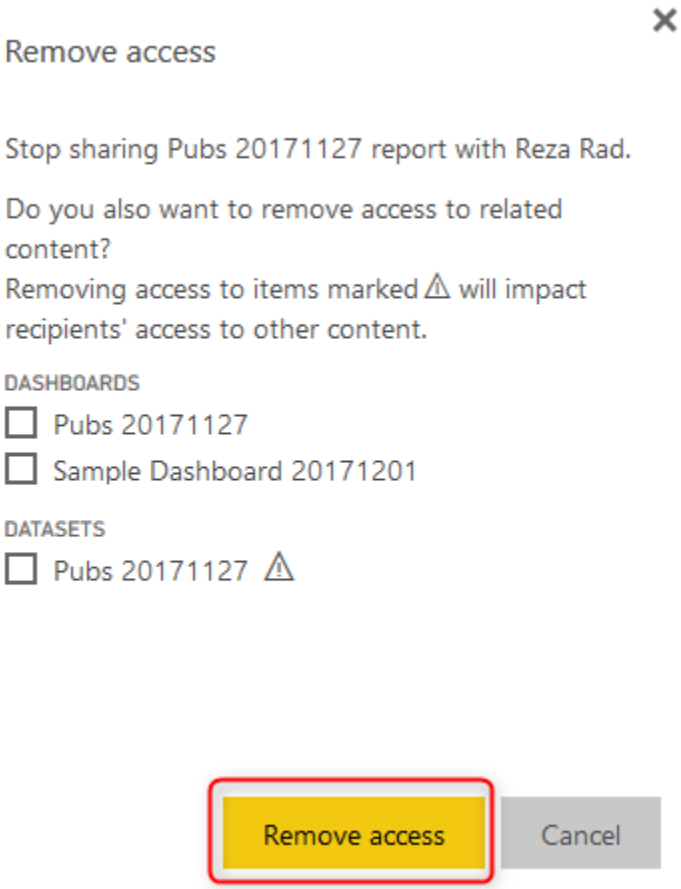
Manage permissions will show you a detailed list of access to the dashboard, reports, and datasets. In the left-hand side of the Manage Permissions section, you will see related reports and datasets. You can click on the report.



By clicking on a report or dataset, you will see the permission specified for that object. And you can change it. For example, user reza@radacad.com has access as Read to the report in below screenshot (because we shared this dashboard with him, so the report sharing happened automatically after that). You can remove that access by clicking on more options.

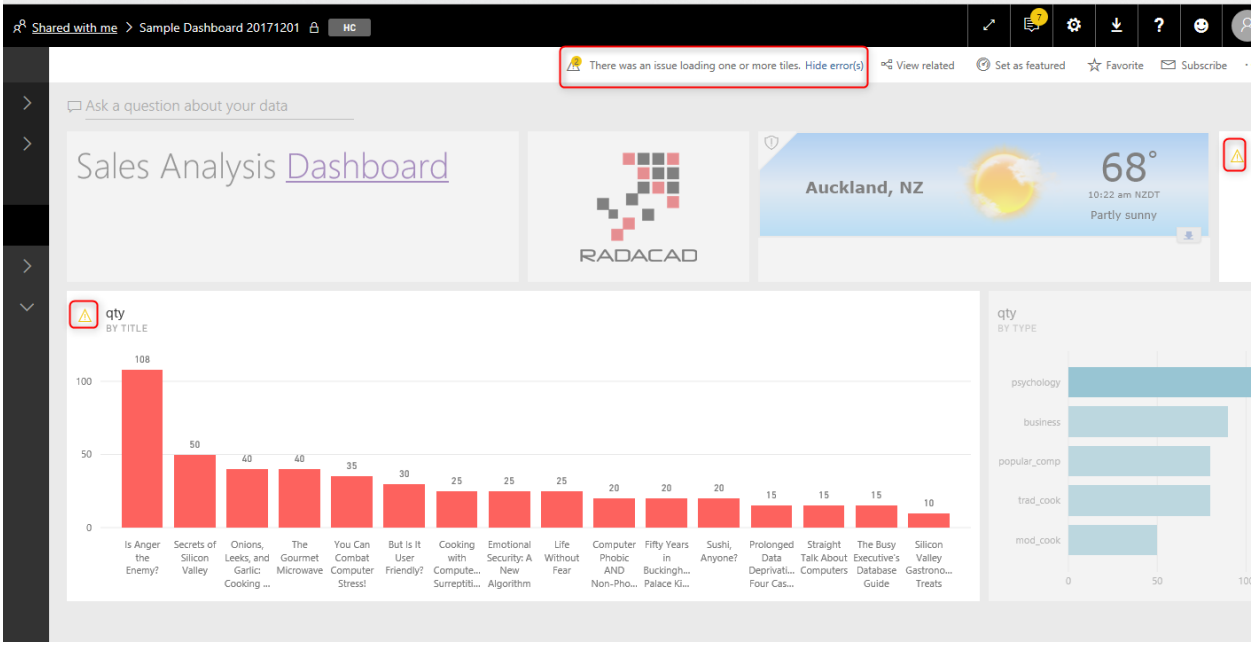


You will see the Remove access window, which asks do you also want to remove the access to some of the related content as well or not. Let's remove access to this report now.



If you are removing access to some of the other items, you should be careful, because that item might be used in multiple other objects. For example, if you remove the access to the dataset, that dataset might be used in multiple reports.

If you shared a dashboard with a user but removed the access to the report or dataset, the user when logged in and accessing the dashboard will see the error message for tiles that are coming from that report. Users cannot drill into the report, because they don't have access to it.



Licensing part of this sharing

Dashboard sharing like many other methods of sharing in Power BI is a paid feature. The account which is sharing the content should be Power BI pro account, and people who are using the shared content should be part of a paid account (Power BI Pro accounts, of Power BI free accounts under a Power BI premium capacity). I'll talk about licensing more in future posts. Free users with no connection to Power BI premium cannot leverage content shared with this method of sharing.

Advantages of Dashboard Sharing

Dashboard sharing is the most basic way of sharing content in Power BI. This method is quick and easy to set up. You don't need to have a lot of steps to set up sharing of the dashboard. The ability to share it very quickly makes this method the most common method of sharing for testing.

> If you have created a Power BI content and want to share it with others easily just for Testing, one of your first options in Dashboard sharing.

Disadvantages of Dashboard Sharing

Dashboard sharing is simple; however, it has many drawbacks, which makes it hard to be used in production. I do not recommend using this method to share Power BI content with users in a production environment because of reasons mentioned below;

No Edit Access

With Dashboard sharing, you cannot specify edit access. For end users, you never want to give edit access, however, if you are working with a team of developers, and you want to provide them with access to edit the content, you cannot do that with dashboard sharing. You have to use other methods of sharing, which will come in the next few posts.

Share Objects one at a time

You can only share one dashboard at a time. What if you wanted to share hundreds of dashboards? You must go to each dashboard, and share items individually. Sharing every single dashboard would add a lot of maintenance overhead to your work. Best would be having all contents under a group and sharing it with others at once.

Summary

Dashboard sharing is straightforward; it has two levels of access, Read, or Read and reshare. You can use this method efficiently for test scenarios. When you want to share a dashboard with a user for testing, Dashboard sharing can be one of the best options to choose.

Dashboard sharing, however, has some disadvantages. There is no Edit access to this way of sharing, and on the other hand, if you want to share multiple items, you have to go to each dashboard and share individually from there. Because of these two significant limitations, dashboard sharing is never used in development or production environment of Power BI implementation. Other methods which I'll write about them in the next few posts, can cover these limitations.

Other modules of the book

Congratulations on completing the first book of Power BI from Rookie to Rock Star series. You are in the right track, but still more to do. Here are other modules that you can read:

- Book 1: Power BI Essentials
- **Book 2: Visualization with Power BI**
- **Book 3: Power Query and Data Transformation in Power BI**
- **Book 4: Power BI Data Modelling and DAX**
- **Book 5: Pro Power BI Architecture**

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 the schedule of upcoming courses here:

http://radacad.com/events

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