What is Power BI

Power BI is a self-serviced, cloud-based reporting, Data visualization, Business Intelligence software

Self Service-if you are able to complete your work by doing Drag& Drop, Point& Click. That Kind of software you call it as Self-Service software.

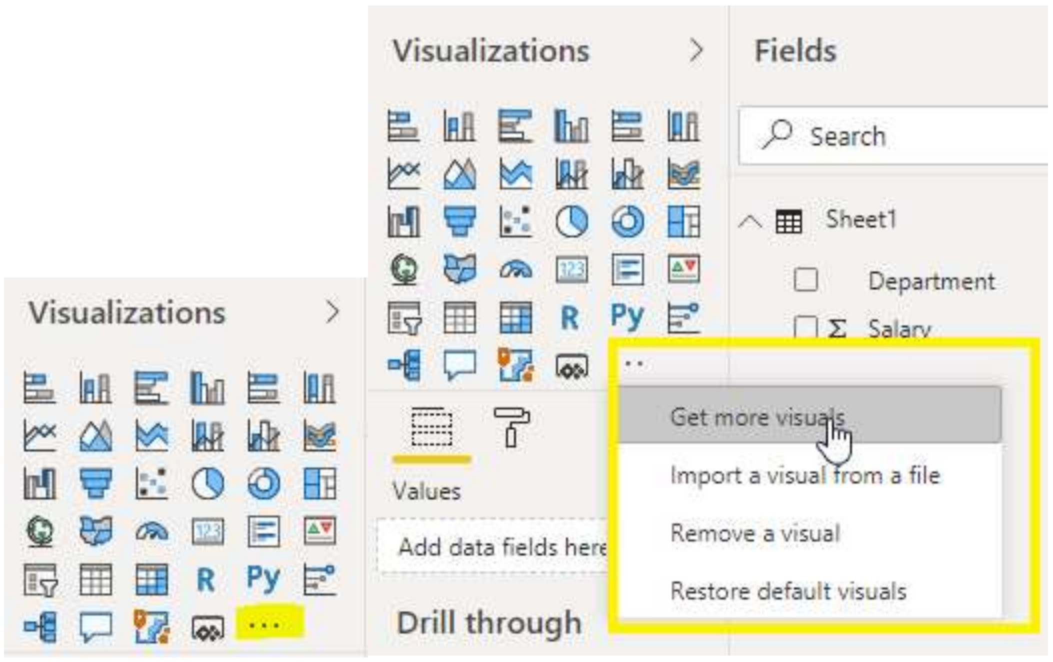
What is reporting/ Data Visualization:

Presenting the data in the form of image (Visualizations--There are 200 plus visualizations in Power BI) is called Data Visualization. (End users can understand the data within seconds)

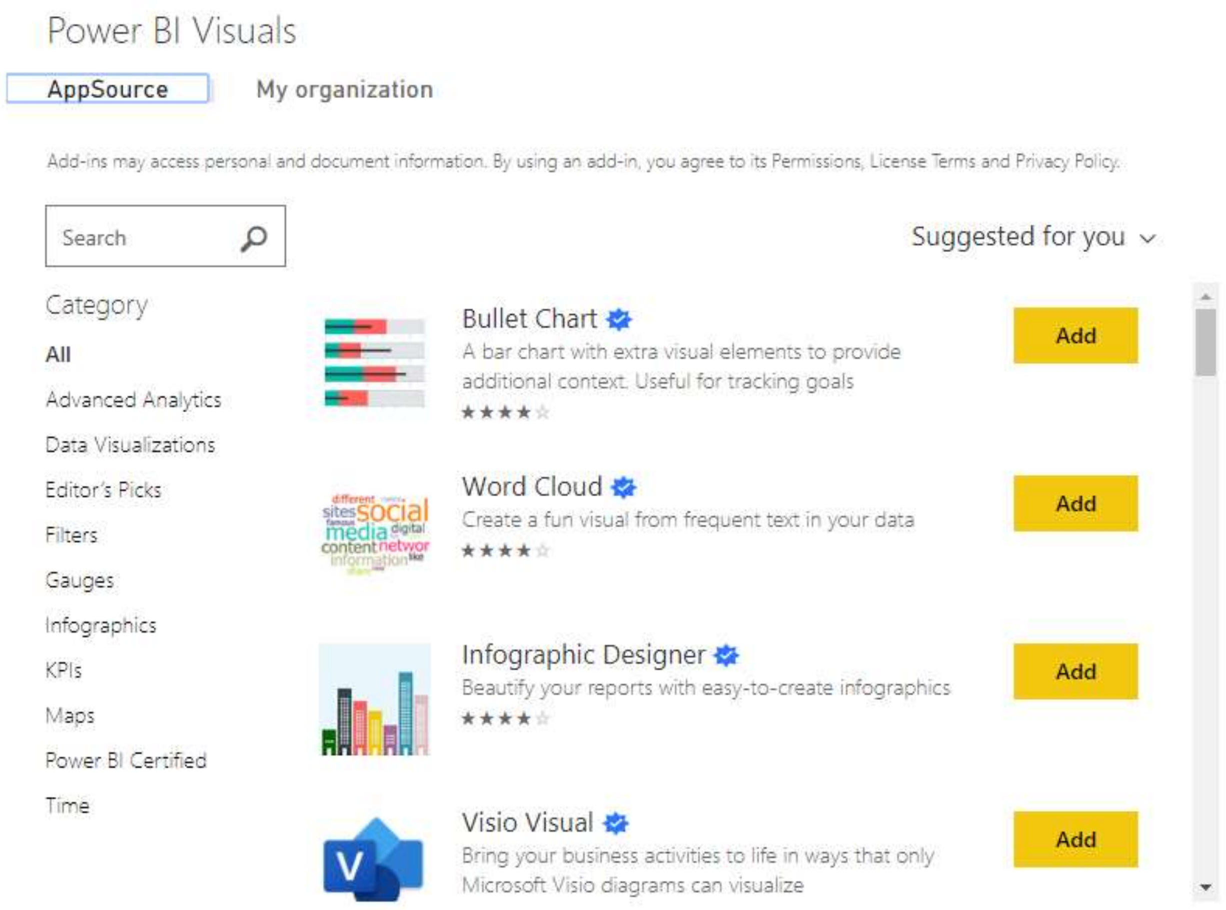
Presenting the data in a structured format as we can call as Reporting.

For ex: If customer wants the sales by region means you can take a tabular visual and drag Sales and Region columns to it. There are many visual are there. We can show data in many structured visual forms. Just I taken Table example here.

Here you can observe some visuals, then where are other visuals means you can right click on highliated part (...) of below screen after you will able to see like other screen



After clicking more visuals it will navigate to market place, there you can see lot of visuals. And you can add visuals to power bi desktop.



Add-ins may access personal and document information. By using an add-in, you agree to its Permissions, License Terms and Privacy Policy.

Bullet Chart
A bar chart with extra visual elements to provide additional context. Useful for tracking goals

Word Cloud
Create a fun visual from frequent text in your data

Infographic Designer
Beautify your reports with easy-to-create infographics

Visio Visual
Bring your business activities to life in ways that only Microsoft Visio diagrams can visualize

(Hardly tableau have 50plus visualizations. This is also one of the differences between the Power BI and tableau.)

Data analytics: if you have million of records and top of it you want to calculate some of sales, then how will you do you cannot manually do sum, in this case you need some data analytic software. this case power Bi will take care of it. Power Bi has a feature

When you call data analytic software—with the help of some software you are able to analyze huge volume of data. We call the software is called Data analytic software.

Power Bi is a tool where you can load millions of data and you can perform analysis on the data.

Business Intelligence: BI is the process which converts business data into actionable item.

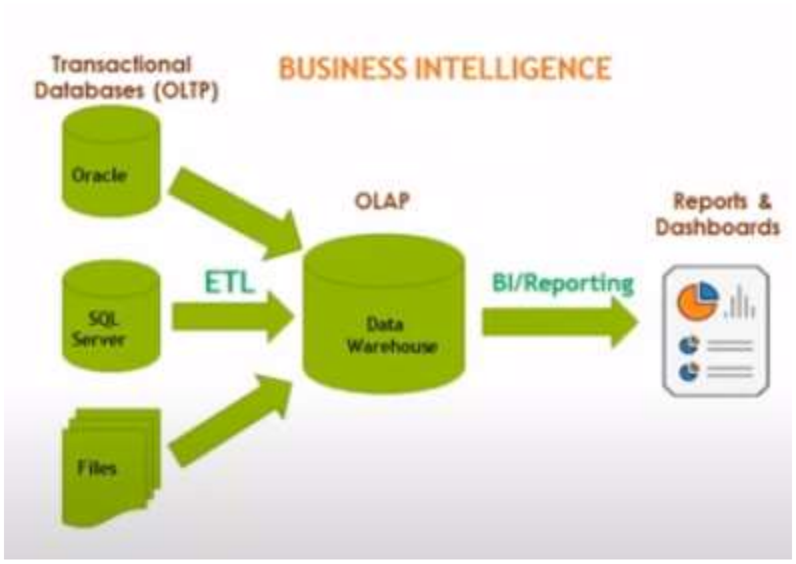
Business data means-=OLTP systems **Actionable Item= Reports/Dashboards**

Company-Amazon

Business-E commerce

Business data-Generally what type of business data E commerce company have

Like-how many sales are happened in last year, how many orders cancelled. And etc.



OLTP ETL OLAP Reports/Dashboard Sharing with client -Entire this process is called Business Intelligence.

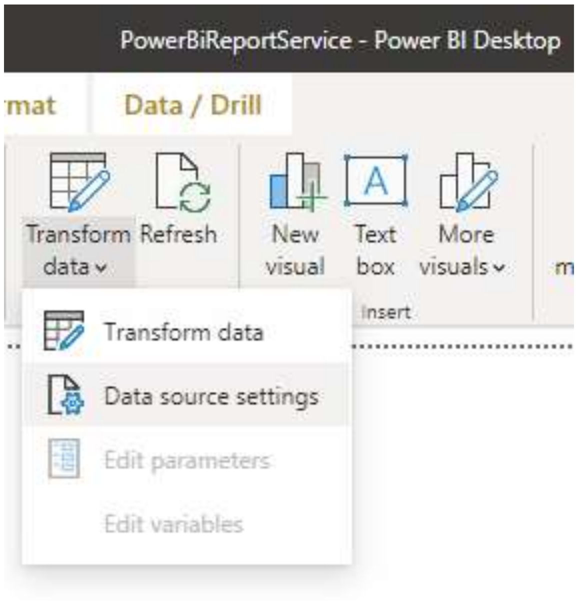
By using all Power Bi components, we can do BI activities.

Power Bi Components/Software's/Tools/Product

1) Power Bi Desktop- A Development Software (A Desktop-based authoring tool of choice for connecting, transforming, and modeling data for creating interactive reports)

- **Power Query-** ETL (Power Query Editor)
  How will you go to Power Query

In Power Bi desktop there is a option called Transform data/Edit queries. It will navigate to query editor. That part is called "**Power Query**"



There we Can Prepare the data for Reporting by Transform option (**Data preparation activities**). here We can do transformations (ETL Operations). For doing all these activities we have GUI options in Query Editor. Behind GUI options Microsoft are written in M language Code.
Once did close and apply means the data will load to Power Pivot.
When you load the table from power query to power pivot, you can see the tables in Model view.
Power Query uses M language do to transformations.

- **Power Pivot- OLAP. in this section you can perform DAX**

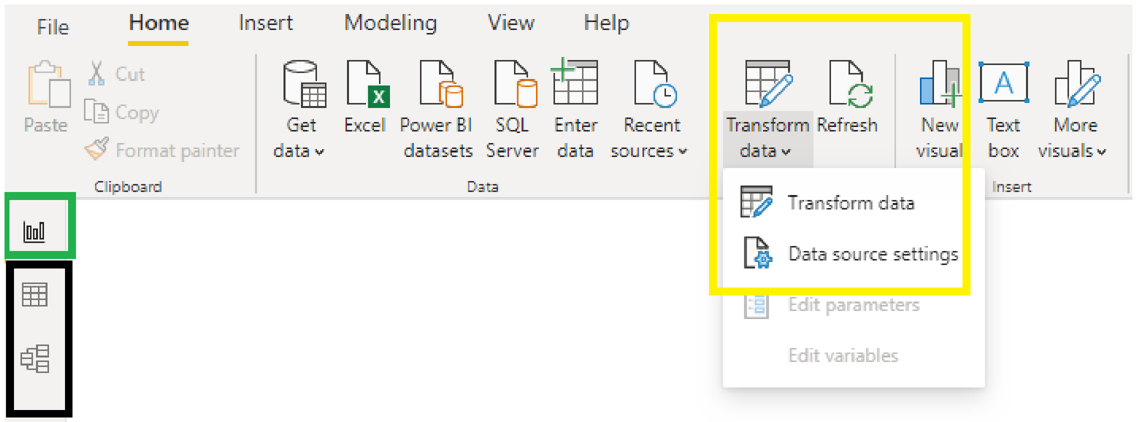
Power Pivot is a In-memory component(xvelocity)- In memory component means it is also database which contains tables (which we are loaded from power Query).
Power Pivot is a in memory database. It can store all the tables along with the data.
When compare to normal database, In-memory data base are much powerful.
Internally we have a software for Power Pivot, that software name is **vertipac** software.
(And similarly, we have SAP in memory database in market)
When you click on Close and Apply in Power Query, this **vertipac** software will come into the picture.
What vertipac does means- it will compress (Approx. 1/10th of source data) the data and stores the data in columnar format. This is completely is a internally mechanism. As a developer we no need do anything here. Power BI will take care of it.
If Normal source having 5 tables with 100GB memory means, in power pivot vertipac will compress into 10GB data. So, it compressing 1/10th part of original size.
Why in memories databases are Powerful

In Normal database (SQl or Oracle), they can store the data in row wise. When it comes to in memory database they can store the data column wise.
In memory means always they usa RAM memory.
Data will be loaded to application after some transformations.,
Data Modeling-establishing the relationship between tables is called **Data Modeling activities**.
creating schemas, we can create new information from existing info. Creating columns and Measures. DAX functions-Data analysis expression language-Functional language.
In power Pivot total we can perform 2 tasks:
1)Data modeling activities --will be done in **model view**
2)Enhancing data- Creating new information using existed information.—these will be done in Data view.
Ex: by using DAX functions we can do some operations here.
Power pivot uses DAX language to enhance the data.

- Online answer what will you do in power pivot is we **prepare a dataset**.
- **Power View- Create reports.**

We will use dataset(which was prepared in power pivot step) as datasource to create a report in power view.

Visualizing the data, Creating Visualizations (**Data Visualization activities**)



Based on color you can find which indicates to which one.
Power Query
Power Pivot
Power View

2)Power BI Service- Azure cloud-based server which is used to sharing the reports with End user. **https://app.powerbi.com**

A web-based service for collaboration, sharing, creation of dashboards, and additional functionality such as Q&A and alerts. It may also be referred to as the "Power Bi Site" or PowerBI.com or the Power Bi Web Portal

3)Power BI Reporting Server- On Premise server which is used to sharing the reports with End user.

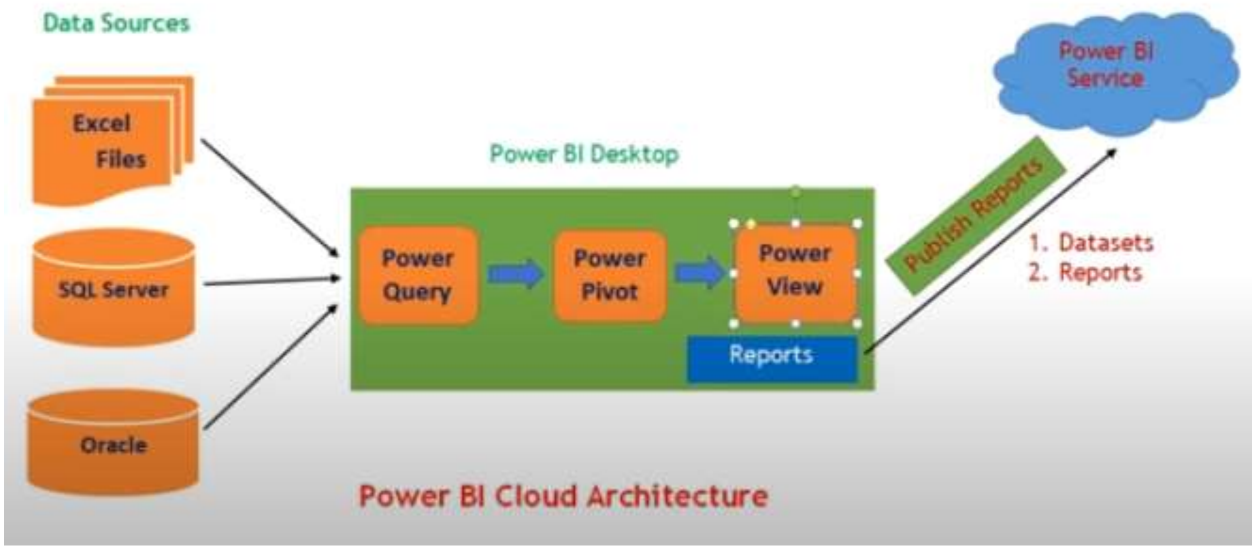
An alternative to the Power BI Service for deploying Power Bi Reports within an on-premises data center, as opposed to the cloud-based Power Bi Service. Power BI Report Server requires a Power BI Premium license.

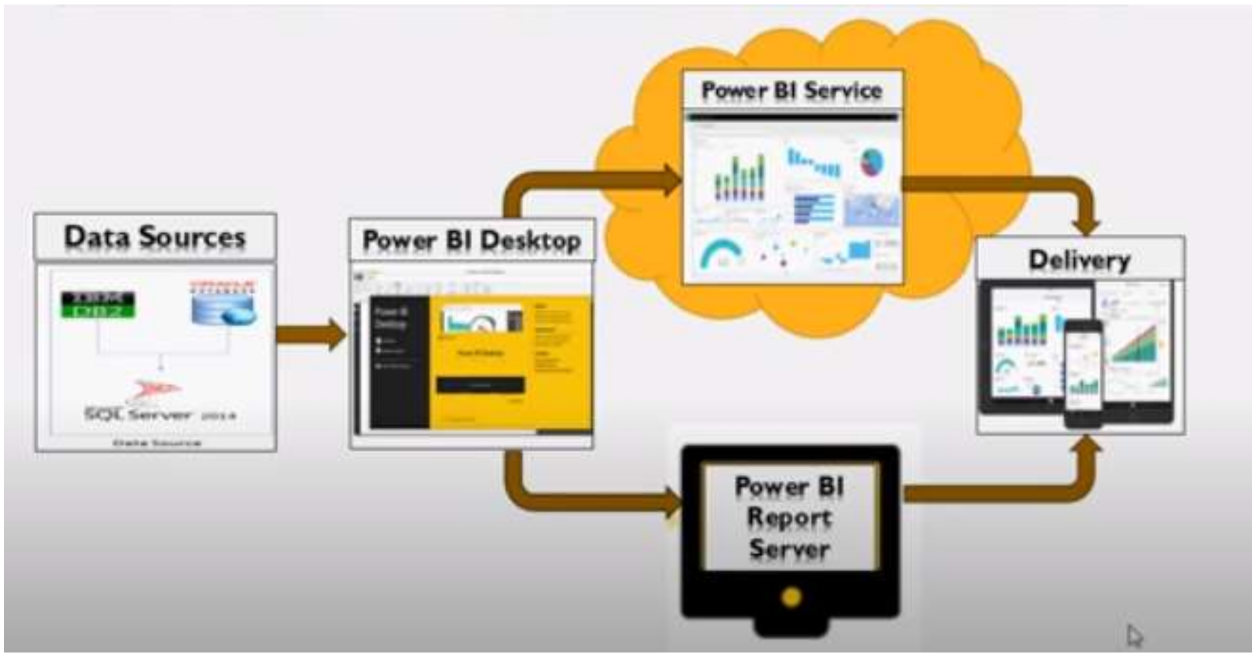
4) Power BI Mobile-Viewing the Reports and dashboards from anywhere in Mobile.

5)On-Premises Gateway- An agent installed within the corporate network to allow secure access to organizational data stored on-premises. There are two options: Enterprise and Personal

6)Power BI Embedded- APIs for embedding visuals into a custom application.

Power BI Architecture





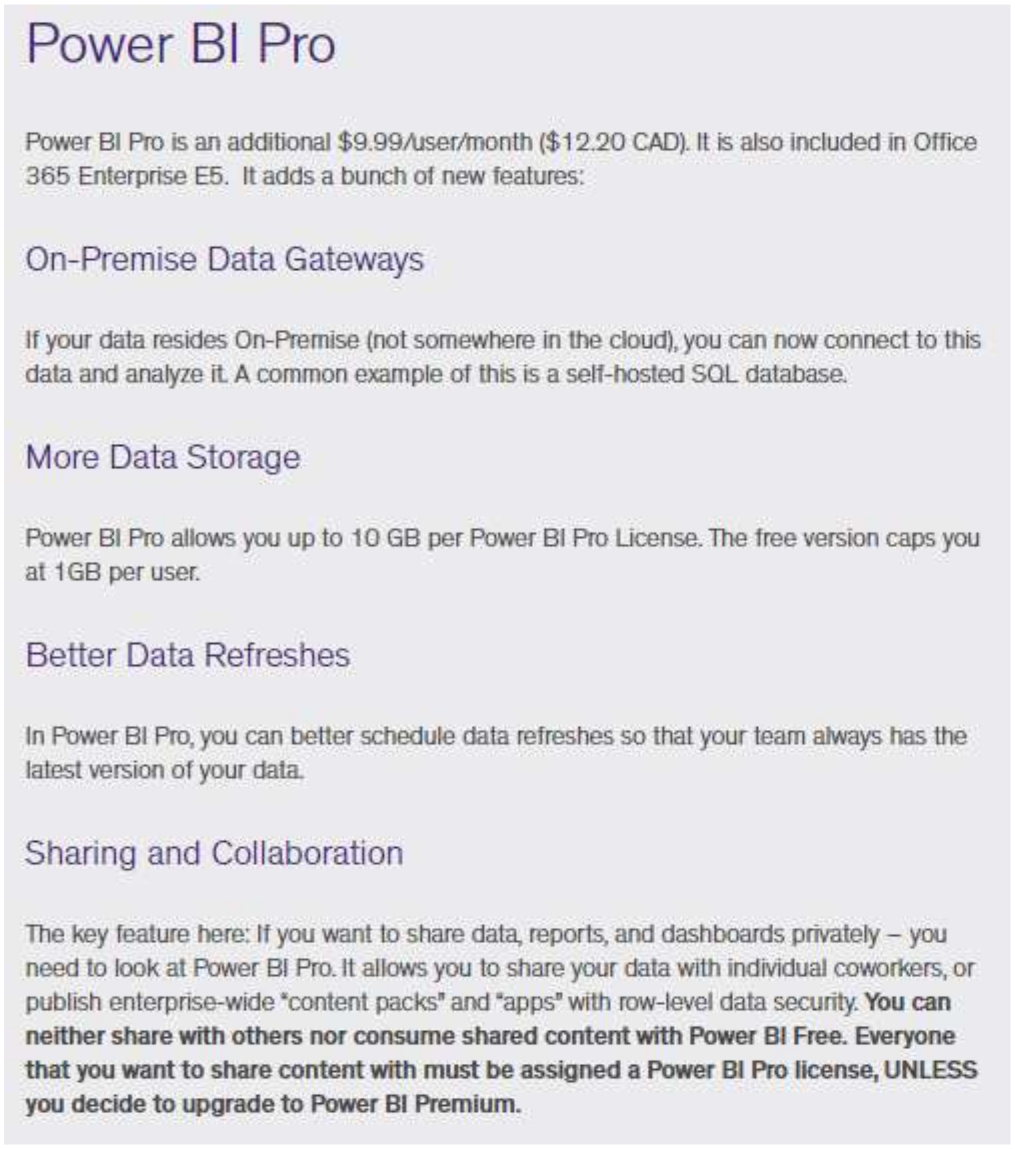
License

Power Bi Free:

- Incudes the ability to use the power bi Service for personal use.
- It allows you to connect hundreds of data sources (no limit/restriction other than the amount of data you pull in), clean and prepare your data, and build visualizations (no limit). All the types of visualizations options in Power BI Pro are available in Power BI free.

Power Bi Pro:

- Incudes all power BI free features.
- The Size of an imported dataset is limited to 2GB per individua dataset.
- A data refresh schedule for an imported dataset is limited to a maximum of 8 times per day with a Power Bi Pro license.


Power BI Pro

Power BI Pro is an additional $9.99/user/month ($12.20 CAD). It is also included in Office 365 Enterprise E5. It adds a bunch of new features:

On-Premise Data Gateways

If your data resides On-Premise (not somewhere in the cloud), you can now connect to this data and analyze it. A common example of this is a self-hosted SQL database.

More Data Storage

Power BI Pro allows you up to 10 GB per Power BI Pro License. The free version caps you at 1GB per user.

Better Data Refreshes

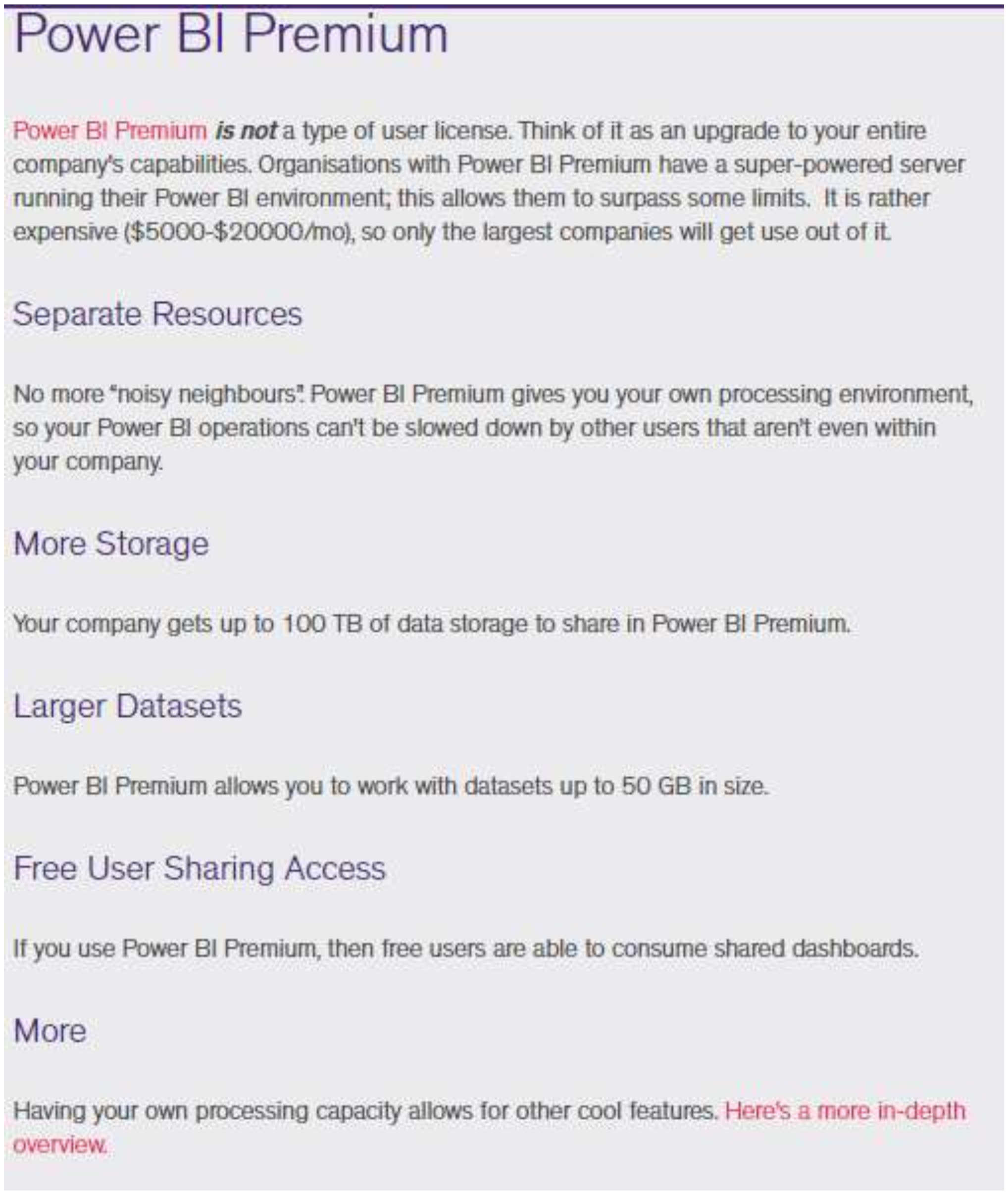
In Power BI Pro, you can better schedule data refreshes so that your team always has the latest version of your data.

Sharing and Collaboration

The key feature here: If you want to share data, reports, and dashboards privately – you need to look at Power BI Pro. It allows you to share your data with individual coworkers, or publish enterprise-wide "content packs" and "apps" with row-level data security. **You can neither share with others nor consume shared content with Power BI Free. Everyone that you want to share content with must be assigned a Power BI Pro license, UNLESS you decide to upgrade to Power BI Premium.**

Power Bi Premium:

- Power Bi Premium also includes a Power Bi Report Server subscription.
- On-Premises report publishing. APIs for custom application integration.



Power BI Premium

Power BI Premium ***is not*** a type of user license. Think of it as an upgrade to your entire company's capabilities. Organisations with Power BI Premium have a super-powered server running their Power BI environment; this allows them to surpass some limits. It is rather expensive ($5000-$20000/mo), so only the largest companies will get use out of it.

Separate Resources

No more "noisy neighbours". Power BI Premium gives you your own processing environment, so your Power BI operations can't be slowed down by other users that aren't even within your company.

More Storage

Your company gets up to 100 TB of data storage to share in Power BI Premium.

Larger Datasets

Power BI Premium allows you to work with datasets up to 50 GB in size.

Free User Sharing Access

If you use Power BI Premium, then free users are able to consume shared dashboards.

More

Having your own processing capacity allows for other cool features. Here's a more in-depth overview.

OLTP and OLAP

OLTP-Online Transactional Processing

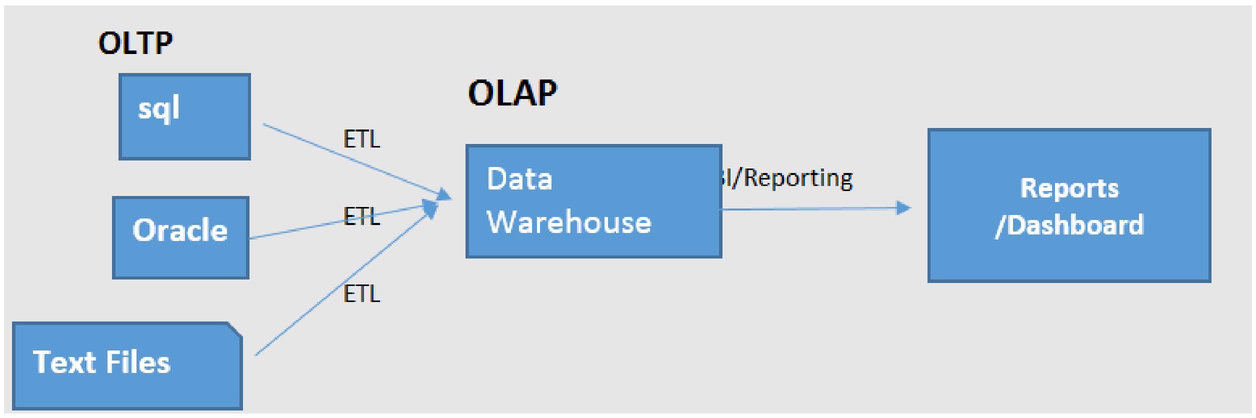
OLAP- Online Analytical Processing

OLTP Systems will store the day to day business data transactions. There we can perform insert/delete/update operations. OLTP systems are not recommended for Analysis/Reporting purpose.

OLAP Systems are Data Warehouse, there we can have extracted data from different sources to analyze the data.

What data we need to move From OLTP to OLAP is the Data that is required for Analysis Purpose. In below case we no need entire business data, we need only finance data for analyzing/creating the report.

OLTP(S)(Business Data)	OLAP(Analysis)
Cust	Fin
Emp	
Fin	



ETL-Extract, Transform, Load

Transform- converting the data from one format to another format is called Transforming.

Get Data

Is used to get the data from different sources. Power BI supportsmorethan100 different data sources.

Get Data GUI option is available in both Desktop and Query Editor Pane.

- When we are getting data from Desktop that time "**Load**", "**Transform**" ," **Cancel**" buttons are available. If Developer Click on **Transform** button, it will navigate to Query Editor pane. There we can do Transformations.

- If you are getting data in Query Editor pane, that time "Load", "Cancel" buttons will be available. Why because you are already in Transforms editor panel so here no need Transform button.

Different type of data source are:

1)File- Excel, Text/CSV, XML, JSON, Folder, PDF, SharePoint Folder.

2)Database-SQL Server Database, Oracle Database, MySQL Database, Postgre Database **etc...**

3)Power Platform-Power BI Datasets, Power BI data flows, **etc....**

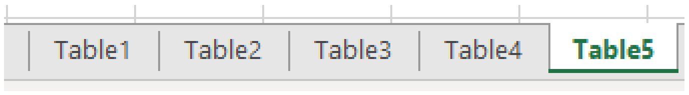
4)Azure-Azure SQL Database, Azure SQL Datawarehouse **etc....**

5)Online Services-Facebook, Google Analytics, Salesforce Objects, Azure Devops (Beta**) etc....**

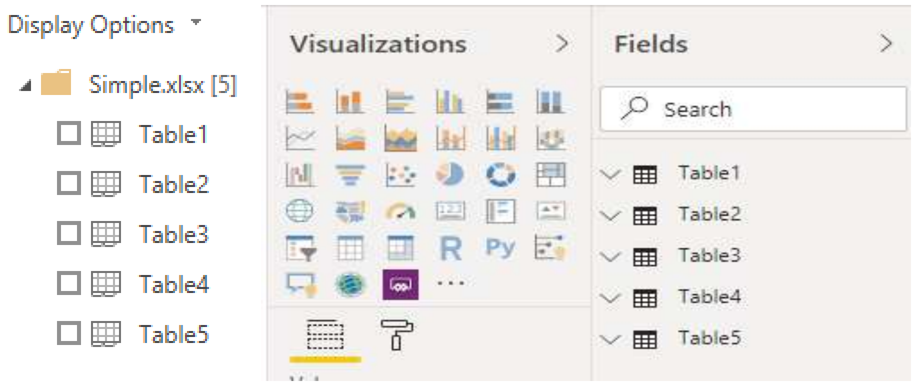
6)Others-Web, Python script, Spark, R Script **etc....**

Get Data from Excel:

If "Simple "Excel having 5 sheets means, Power Bi will convert in 5 pivot tables.



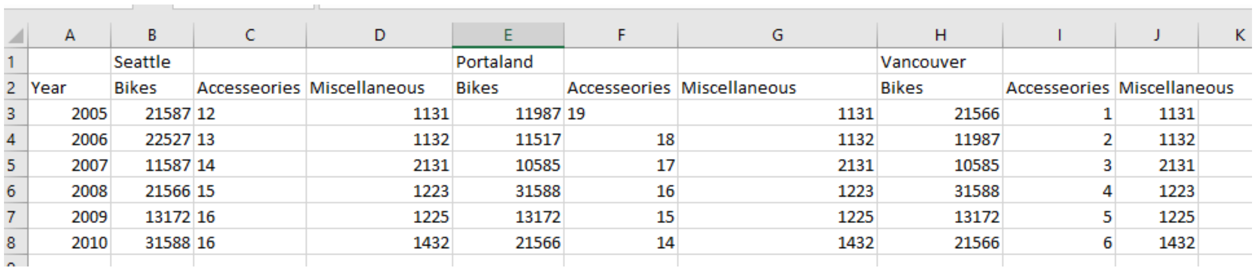
Below screenshots from Power BI Software



How to do data looks like structured way.

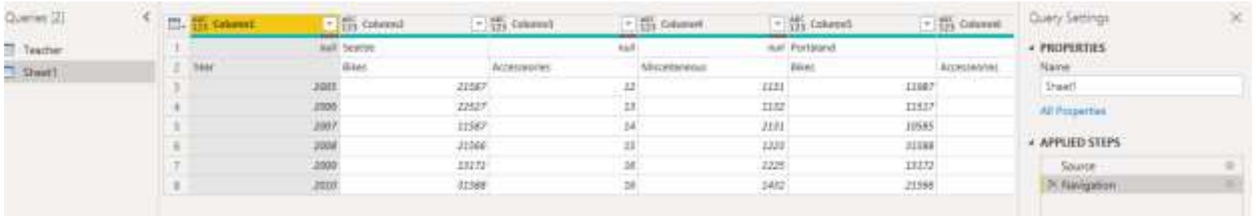
Here an Example by using Excel format.

Load below excel into Query Editor



	A	B	C	D	E	F	G	H	I	J	K
1		Seattle			Portaland			Vancouver			
2	Year	Bikes	Accesseories	Miscellaneous	Bikes	Accesseories	Miscellaneous	Bikes	Accesseories	Miscellaneous	
3	2005	21587	12	1131	11987	19	1131	21566	1	1131	
4	2006	22527	13	1132	11517	18	1132	11987	2	1132	
5	2007	11587	14	2131	10585	17	2131	10585	3	2131	
6	2008	21566	15	1223	31588	16	1223	31588	4	1223	
7	2009	13172	16	1225	13172	15	1225	13172	5	1225	
8	2010	31588	16	1432	21566	14	1432	21566	6	1432	

After loading. It shows like blow

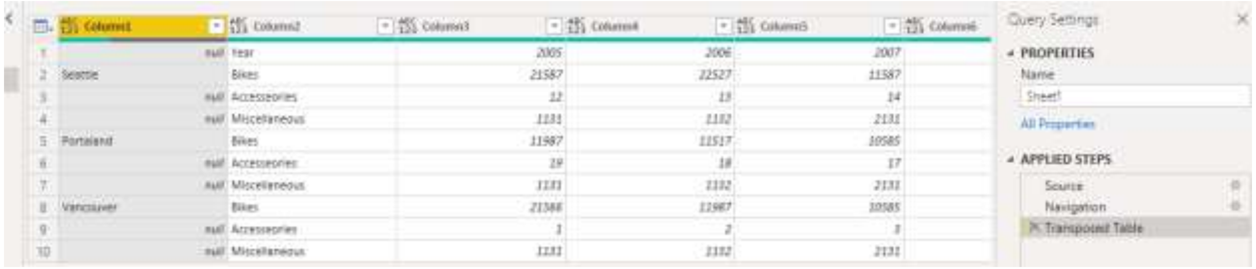


	Column1	Column2	Column3	Column4	Column5	Column6
1	null	Seattle	null	null	Portland	
2	Year	Bikes	Accesseories	Miscellaneous	Bikes	Accesseories
3	2005	21587	12	1131	11987	
4	2006	22527	13	1132	11517	
5	2007	11587	14	2131	10585	
6	2008	21566	15	1223	31588	
7	2009	13172	16	1225	13172	
8	2010	31588	16	1432	21566	

Queries [2]
Teacher
Sheet1

Query Settings
PROPERTIES
Name
Sheet1
All Properties
APPLIED STEPS
Source
Navigation

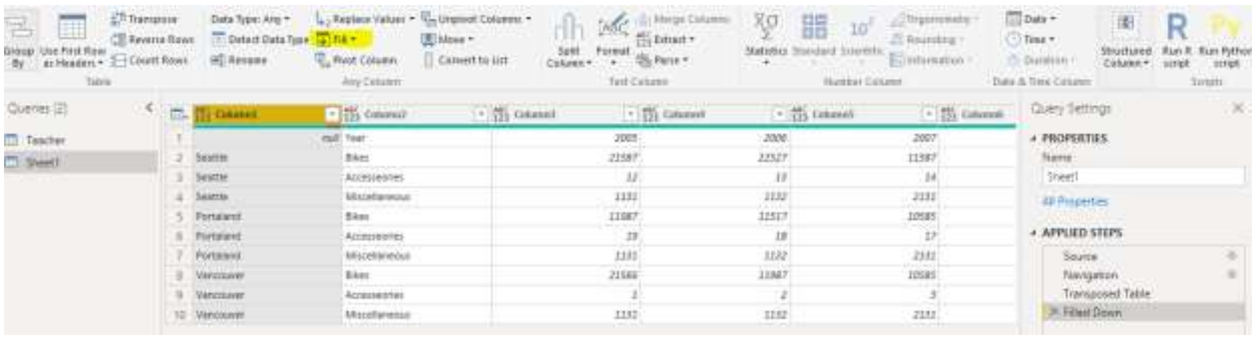
Step 1: Do Transpose -This is treats like Rows as Column and Column as Rows



	Column1	Column2	Column3	Column4	Column5	Column6
1	null	Year	2005	2006	2007	
2	Seattle	Bikes	21587	22527	11587	
3	null	Accesseories	12	13	14	
4	null	Miscellaneous	1131	1132	2131	
5	Portaland	Bikes	11987	11517	10585	
6	null	Accesseories	19	18	17	
7	null	Miscellaneous	1131	1132	2131	
8	Vancouver	Bikes	21566	11987	10585	
9	null	Accesseories	1	2	3	
10	null	Miscellaneous	1131	1132	2131	

Query Settings
PROPERTIES
Name
Sheet1
All Properties
APPLIED STEPS
Source
Navigation
Transposed Table

Step 2: Fill The data to remove nulls



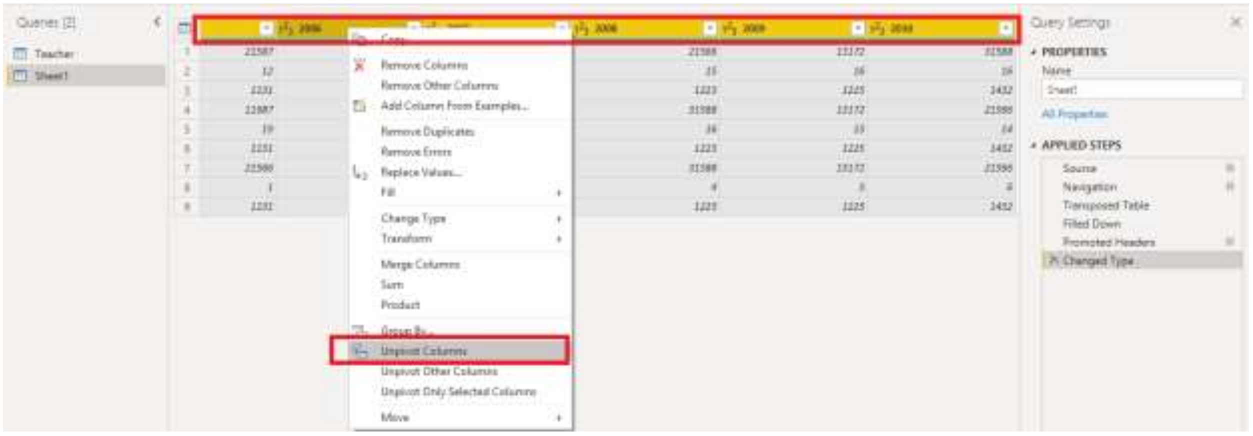
	Column1	Column2	Column3	Column4	Column5	Column6
1	null	Year	2005	2006	2007	
2	Seattle	Bikes	21587	22527	11587	
3	Seattle	Accesseories	12	13	14	
4	Seattle	Miscellaneous	1131	1132	2131	
5	Portaland	Bikes	11987	11517	10585	
6	Portaland	Accesseories	19	18	17	
7	Portaland	Miscellaneous	1131	1132	2131	
8	Vancouver	Bikes	21566	11987	10585	
9	Vancouver	Accesseories	1	2	3	
10	Vancouver	Miscellaneous	1132	1132	2131	

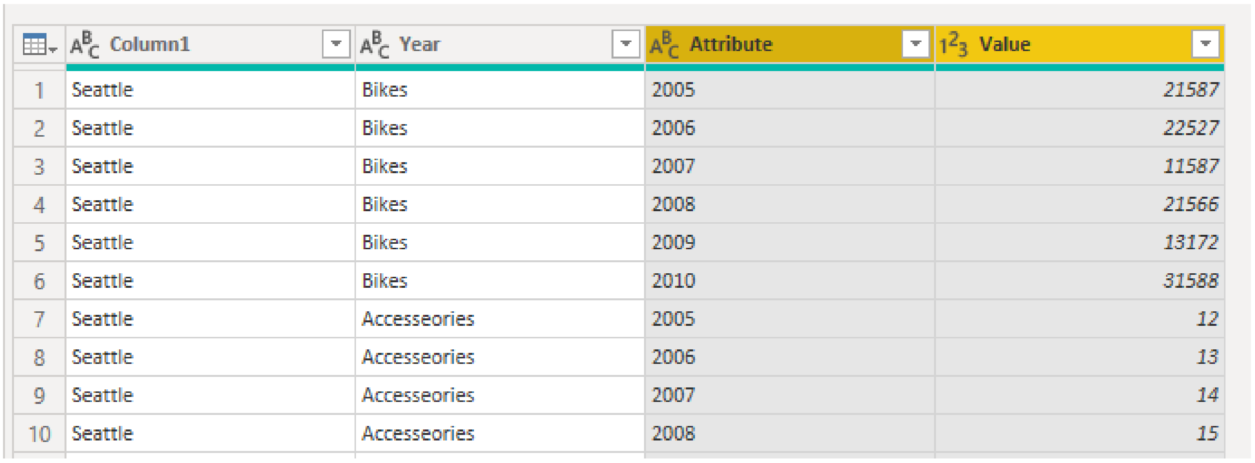
Queries [2]
Teacher
Sheet1

Query Settings
PROPERTIES
Name
Sheet1
All Properties
APPLIED STEPS
Source
Navigation
Transposed Table
Filled Down

Step 3: Make 1st column as Header

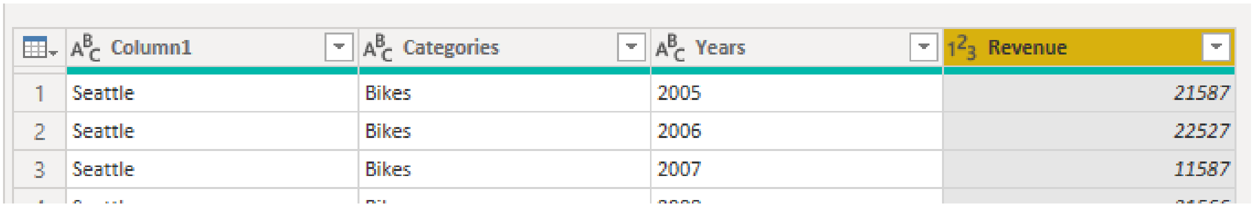
Step 4: Unpivot the years columns





	Column1	Year	Attribute	Value
1	Seattle	Bikes	2005	21587
2	Seattle	Bikes	2006	22527
3	Seattle	Bikes	2007	11587
4	Seattle	Bikes	2008	21566
5	Seattle	Bikes	2009	13172
6	Seattle	Bikes	2010	31588
7	Seattle	Accesseories	2005	12
8	Seattle	Accesseories	2006	13
9	Seattle	Accesseories	2007	14
10	Seattle	Accesseories	2008	15

Step 5: Rename the column names how you need it.



	Column1	Categories	Years	Revenue
1	Seattle	Bikes	2005	21587
2	Seattle	Bikes	2006	22527
3	Seattle	Bikes	2007	11587

Working with SQL Server Database

Import Mode

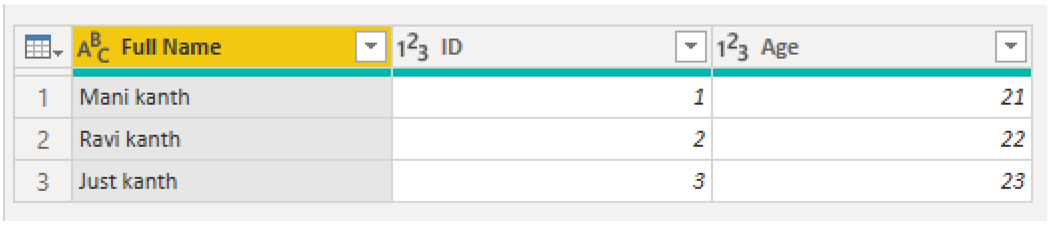
Direct Mode

Splitting Columns:

Power BI have an option to split the columns.

Ex: We have Table with the column “Full Name”.

Task: Split the FirstName column like first name and last name.

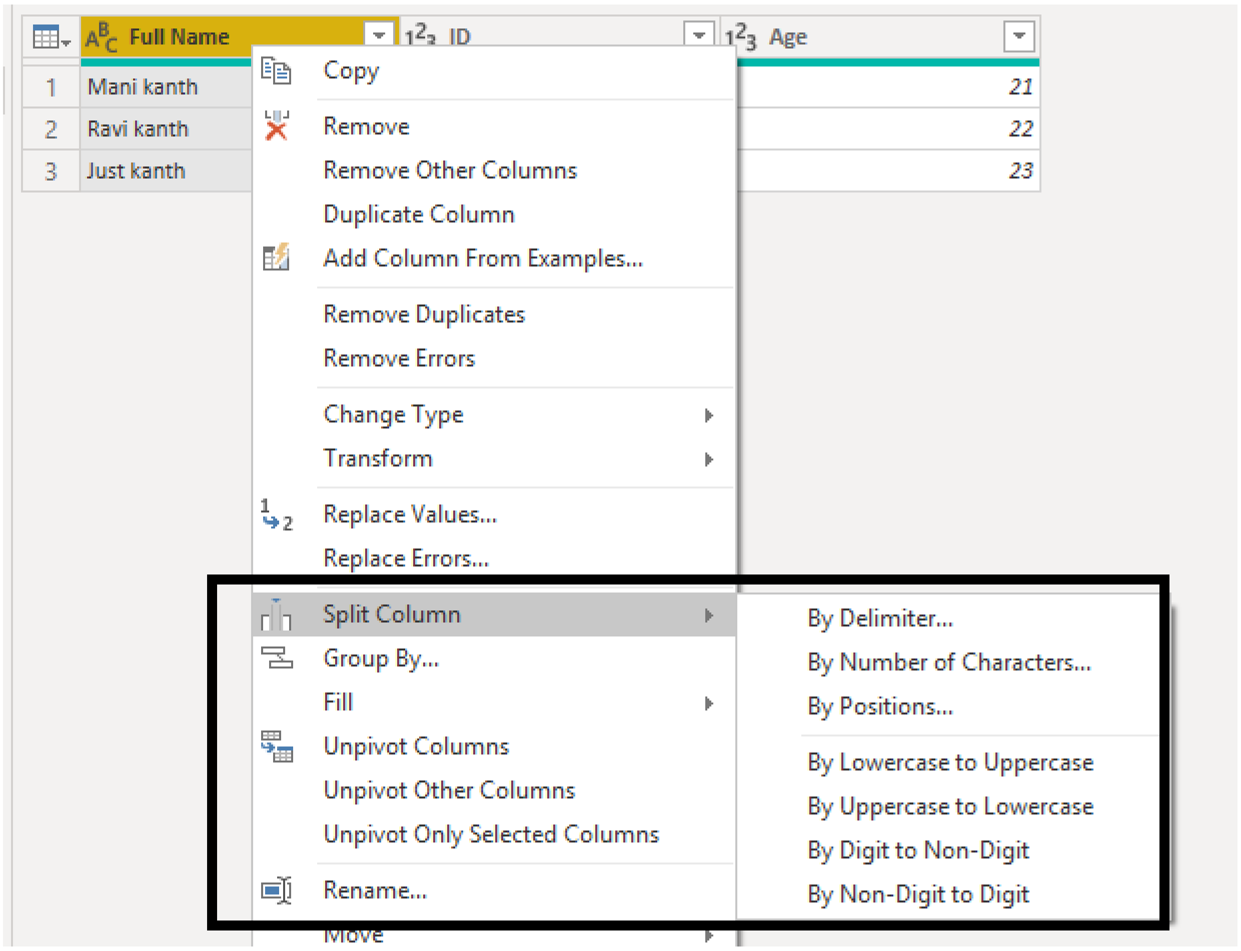


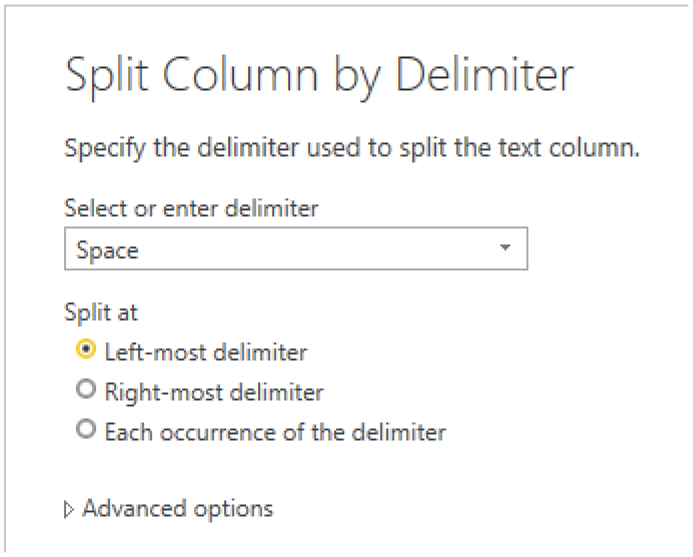
	Full Name	ID	Age
1	Mani kanth	1	21
2	Ravi kanth	2	22
3	Just kanth	3	23

Solution: There are few possibilities to split the column.

In Query Editor we can do that by Right click on required column.

Split columns→By delimiter (in our example we have spaces in between words so am using delimiter option, but there are different options available based on our scenarios)




Split Column by Delimiter

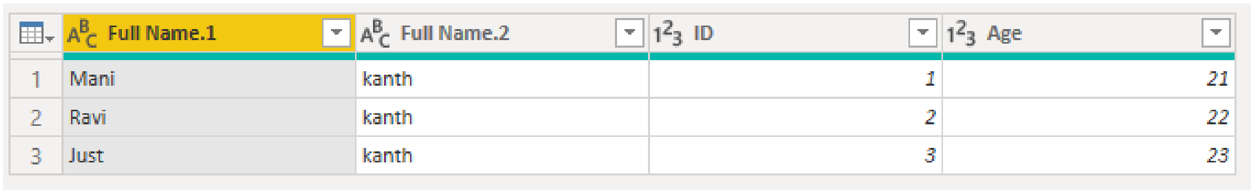
Specify the delimiter used to split the text column.

Select or enter delimiter: Space

Split at
- [✓ Left-most delimiter]
- Right-most delimiter
- Each occurrence of the delimiter

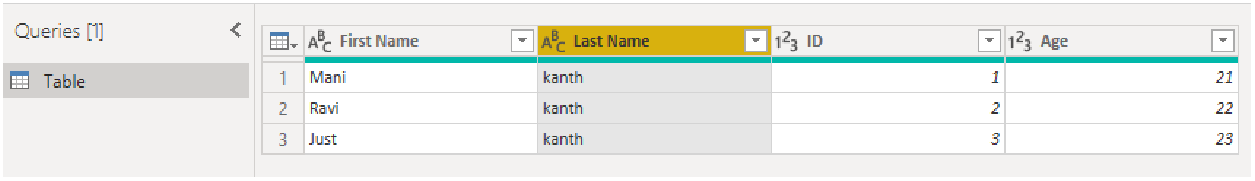
Advanced options

After clicked on OK on above screen, the data comes like below. You can observe that Full Name is spitted into Full Name1 and Full Name2.



	Full Name.1	Full Name.2	ID	Age
1	Mani	kanth	1	21
2	Ravi	kanth	2	22
3	Just	kanth	3	23

If you want change the column names you can change how you want.



Queries [1]

Table

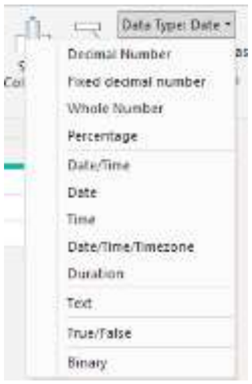
	First Name	Last Name	ID	Age
1	Mani	kanth	1	21
2	Ravi	kanth	2	22
3	Just	kanth	3	23

Keynote: You cannot split column when column data type is DATE.in this case change data type to text and do operations what you need and then later change datatype to DATE. And to do Date operations we have a “Date “option in Query Editor GUI.

Change Data Type

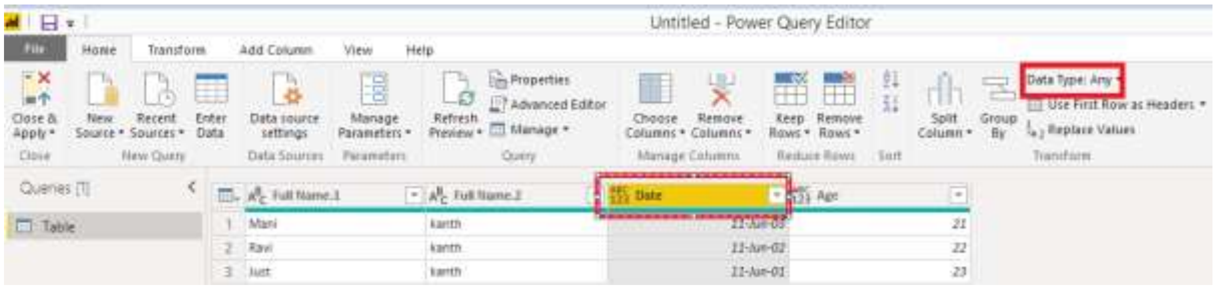
- Data types are in Power BI
- Decimal Number
- Fixed Decimal Number
- Whole Number
- Percentage
- Date/Time
- Date
- Time

- Date/Time/Time zone
- Duration
- Text
- True/False
- Binary



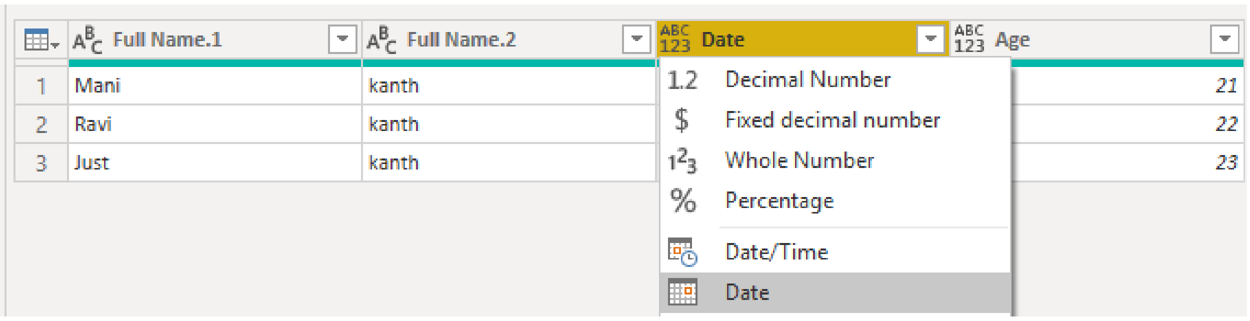
Once you have uploaded the data in your power bi desktop, we must make sure the data tables that we are uploaded should have some unique property to it.

Ex: A “Sample “table that contains a date values under “Date” column’s that data type of the column must be date.



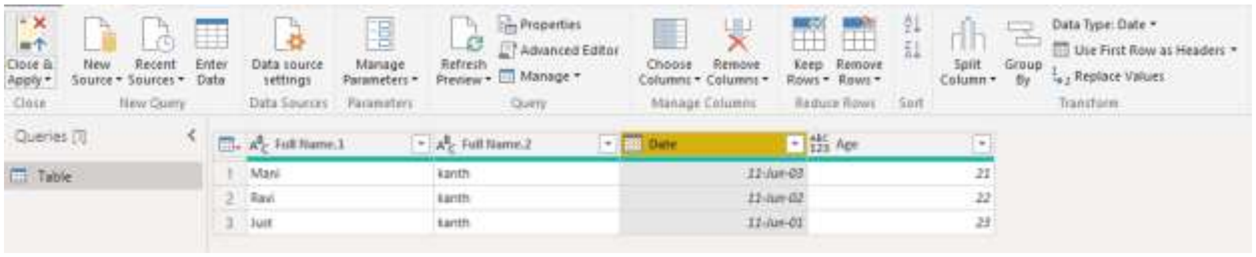
	Full Name.1	Full Name.2	Date	Age
1	Mani	kanth	11-Jun-03	21
2	Ravi	kanth	11-Jun-02	22
3	Just	kanth	11-Jun-01	23

In above image the data type of “Date” column is “Any”. So you have to change the type to “Date”



	Full Name.1	Full Name.2	Date	Age
1	Mani	kanth		21
2	Ravi	kanth		22
3	Just	kanth		23

After clicked “Date”. The data type changed to Date.



	Full Name.1	Full Name.2	Date	Age
1	Mani	kanth	11-Jun-03	21
2	Ravi	kanth	11-Jun-02	22
3	Just	kanth	11-Jun-01	23

Working with Dates:

We have a column with Dates.

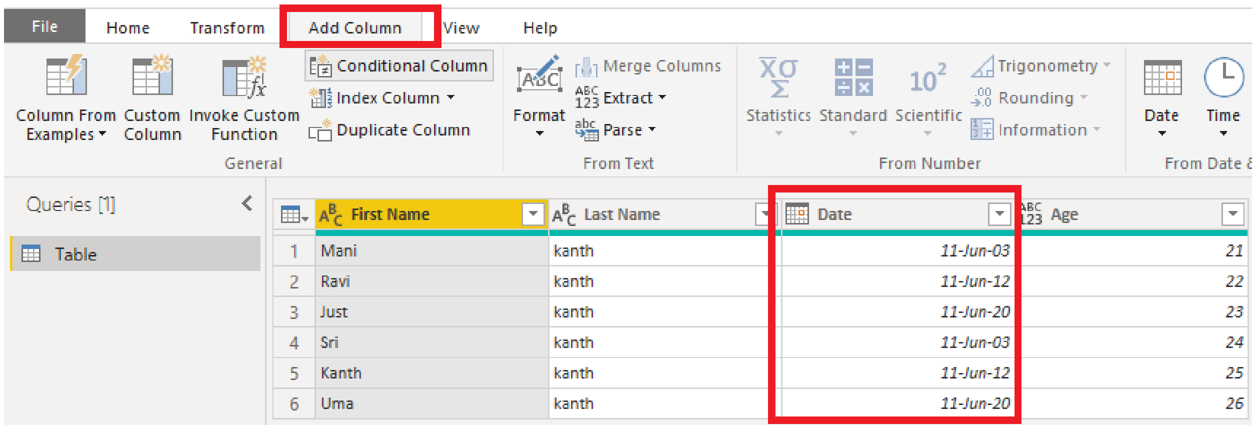
There you can see records are like this format under Date column 19-06-1989

But in report you want to show like below

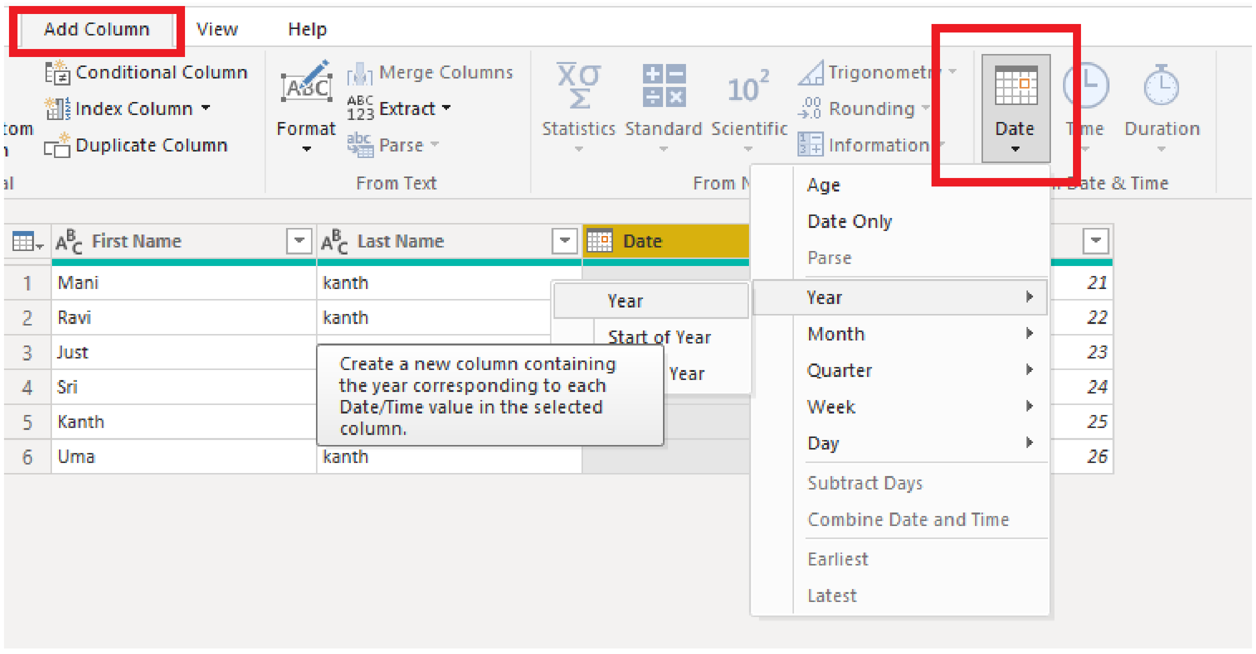
Case1: Add an extra column with Year values

Date	Year
19-06-1989	1989

In above case you must click on ADD Column option in query Editor.

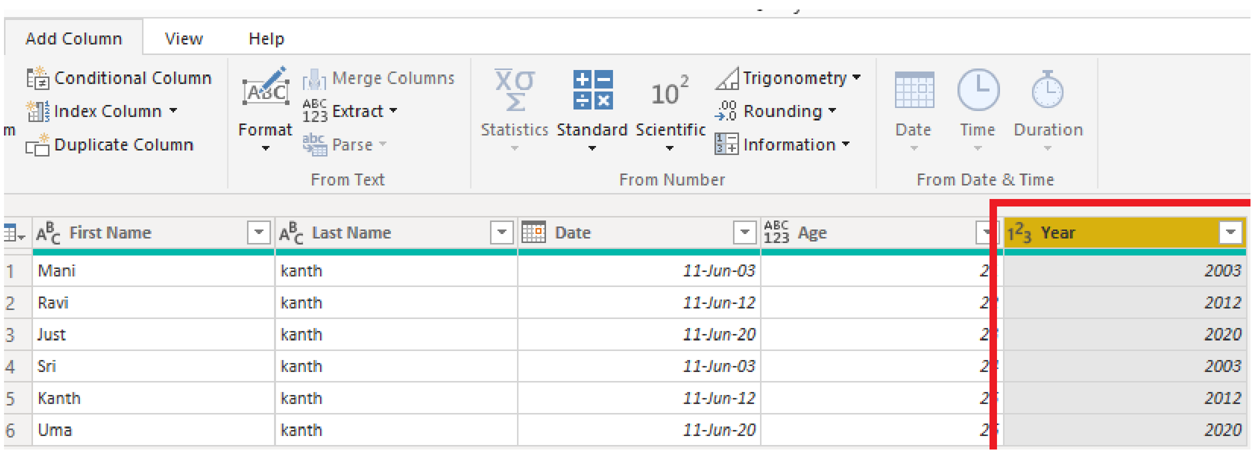


	First Name	Last Name	Date	Age
1	Mani	kanth	11-Jun-03	21
2	Ravi	kanth	11-Jun-12	22
3	Just	kanth	11-Jun-20	23
4	Sri	kanth	11-Jun-03	24
5	Kanth	kanth	11-Jun-12	25
6	Uma	kanth	11-Jun-20	26



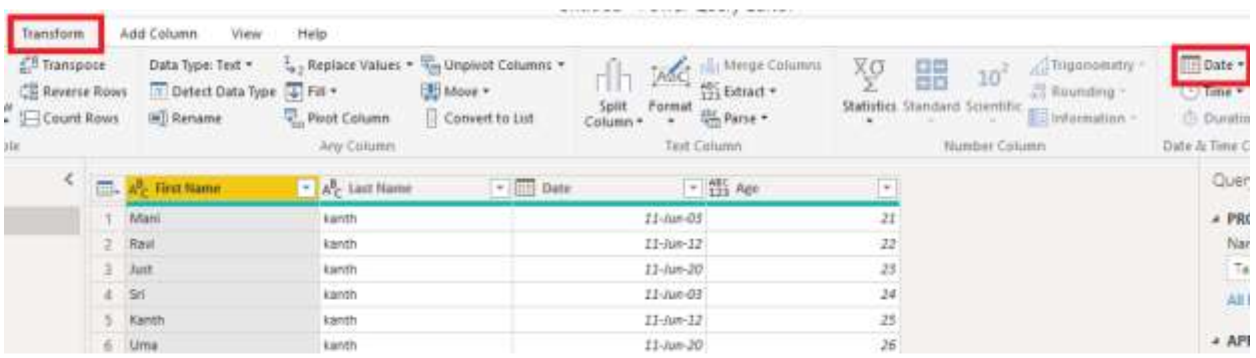
Create a new column containing the year corresponding to each Date/Time value in the selected column.

	First Name	Last Name	Date	
1	Mani	kanth		21
2	Ravi	kanth		22
3	Just			23
4	Sri			24
5	Kanth			25
6	Uma	kanth		26

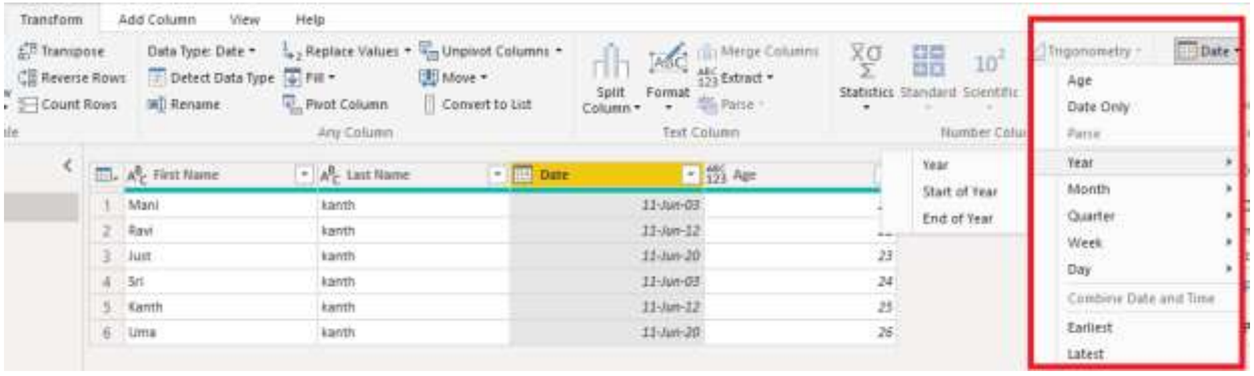


	First Name	Last Name	Date	Age	Year
1	Mani	kanth	11-Jun-03	2	2003
2	Ravi	kanth	11-Jun-12	2	2012
3	Just	kanth	11-Jun-20	2	2020
4	Sri	kanth	11-Jun-03	2	2003
5	Kanth	kanth	11-Jun-12	2	2012
6	Uma	kanth	11-Jun-20	2	2020

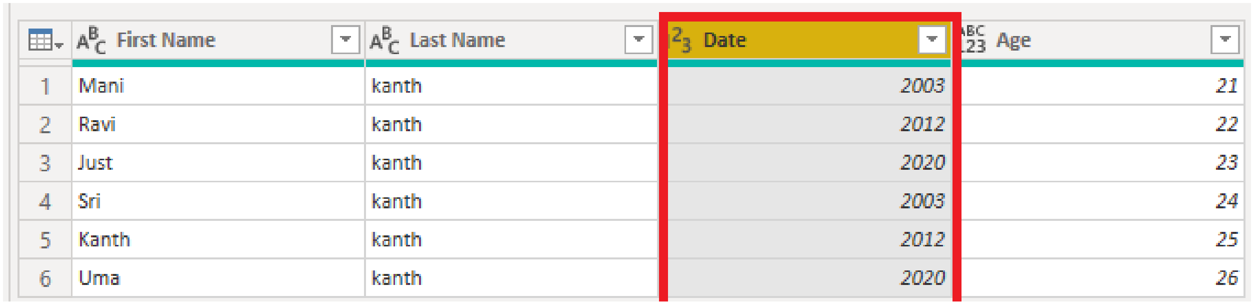
Case 2: In the same column you must show year instead of entire Date.



	First Name	Last Name	Date	Age
1	Mani	kanth	11-Jun-03	21
2	Ravi	kanth	11-Jun-12	22
3	Just	kanth	11-Jun-20	23
4	Sri	kanth	11-Jun-03	24
5	Kanth	kanth	11-Jun-12	25
6	Uma	kanth	11-Jun-20	26



	First Name	Last Name	Date	Age
1	Mani	kanth	11-Jun-03	
2	Ravi	kanth	11-Jun-12	
3	Just	kanth	11-Jun-20	23
4	Sri	kanth	11-Jun-03	24
5	Kanth	kanth	11-Jun-12	25
6	Uma	kanth	11-Jun-20	26



	First Name	Last Name	Date	Age
1	Mani	kanth	2003	21
2	Ravi	kanth	2012	22
3	Just	kanth	2020	23
4	Sri	kanth	2003	24
5	Kanth	kanth	2012	25
6	Uma	kanth	2020	26

If you clearly observe above 2 columns Complete Date records under Date column itself changed to Year wise.

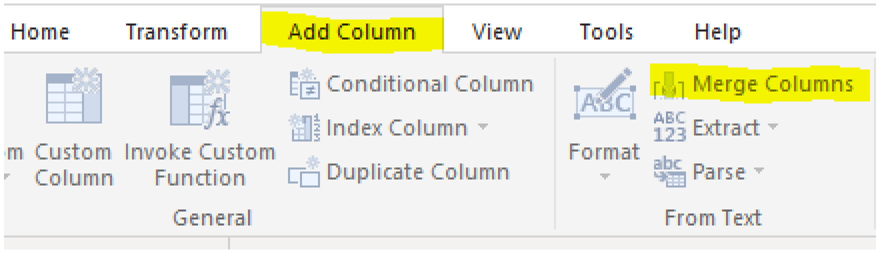
So Key points are:

- When you are doing under ADD column Tab means the extra column is adding for that table (Case 1).
- When you are doing same operation under Transform Tab means it will Transforming/Changing the same column (Case 2)
- Dimensions table to Fact table default relationship is One To Many.

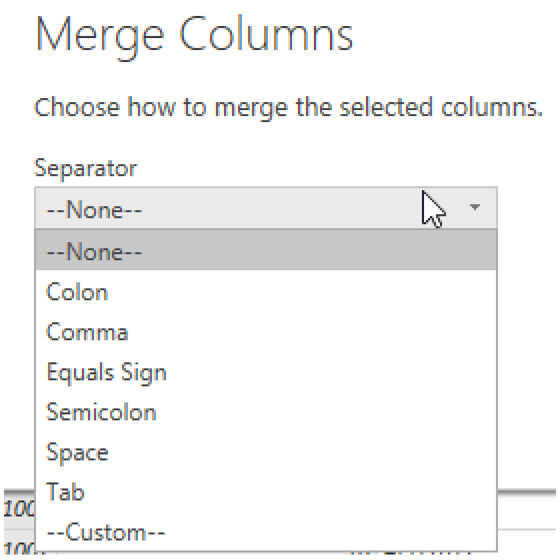
ADD Columns

- **Merge Columns**

We have a table with First Name and Last Name columns. But we want to show Full Name in a single column. In this case Power Query Editor>Add column>Merge Columns.



These are the separator options when we are Merging the selected columns.



Merge Columns

Choose how to merge the selected columns.

Separator

- **Conditional Columns**

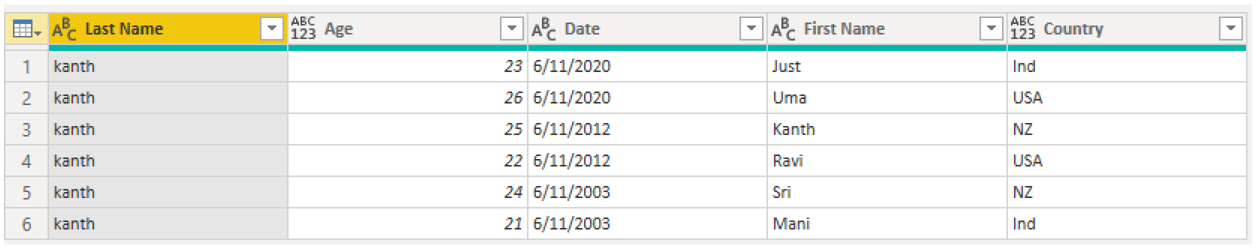
The heading itself says that a column which is based on certain condition.

Ex: Add a column if country=USA then it should be region

If Country=India it should be sub region

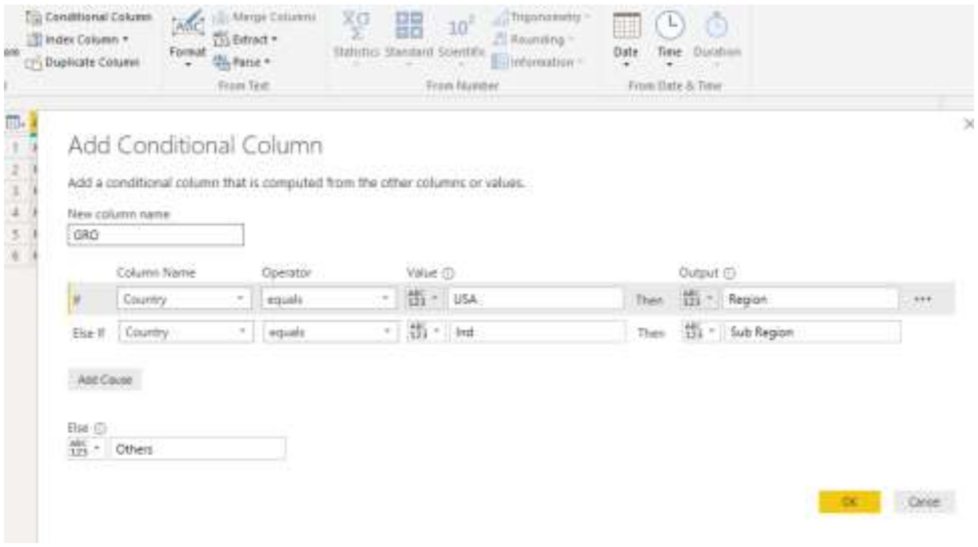
If Country not equal to India or USA, then it should be Others

Current Data:



	Last Name	Age	Date	First Name	Country
1	kanth	23	6/11/2020	Just	Ind
2	kanth	26	6/11/2020	Uma	USA
3	kanth	25	6/11/2012	Kanth	NZ
4	kanth	22	6/11/2012	Ravi	USA
5	kanth	24	6/11/2003	Sri	NZ
6	kanth	21	6/11/2003	Mani	Ind

After adding a condition for new column



Add Conditional Column

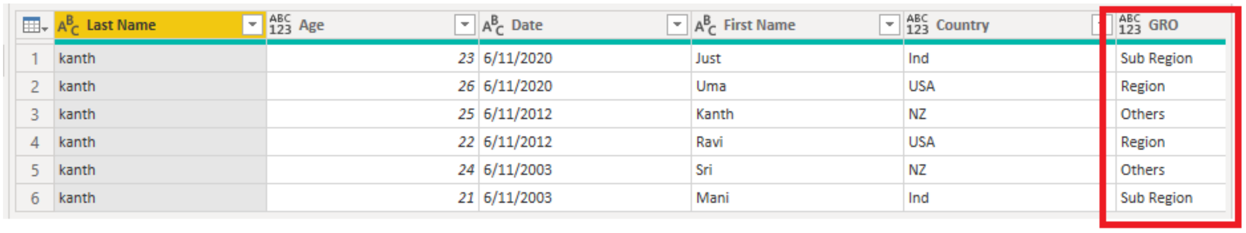
Add a conditional column that is computed from the other columns or values.

New column name: GRO

	Column Name	Operator	Value		Output
If	Country	equals	USA	Then	Region
Else If	Country	equals	Ind	Then	Sub Region

Add Clause

Else: Others



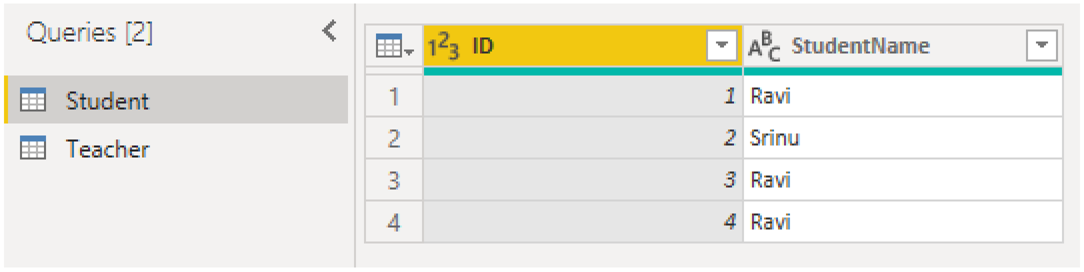
	Last Name	Age	Date	First Name	Country	GRO
1	kanth	23	6/11/2020	Just	Ind	Sub Region
2	kanth	26	6/11/2020	Uma	USA	Region
3	kanth	25	6/11/2012	Kanth	NZ	Others
4	kanth	22	6/11/2012	Ravi	USA	Region
5	kanth	24	6/11/2003	Sri	NZ	Others
6	kanth	21	6/11/2003	Mani	Ind	Sub Region

Merge and Append Queries

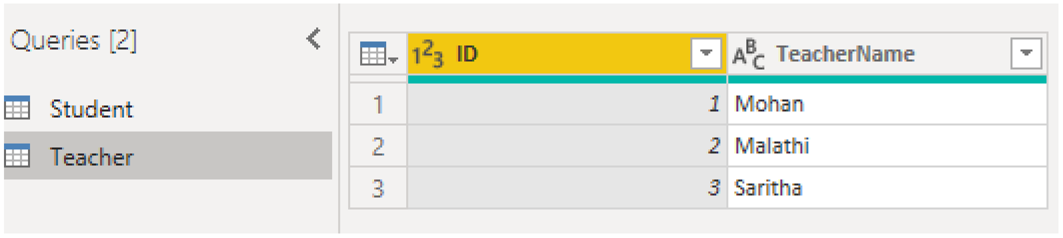
Merge:

We have a table with names Student and Teacher.

There we have a common ID column.

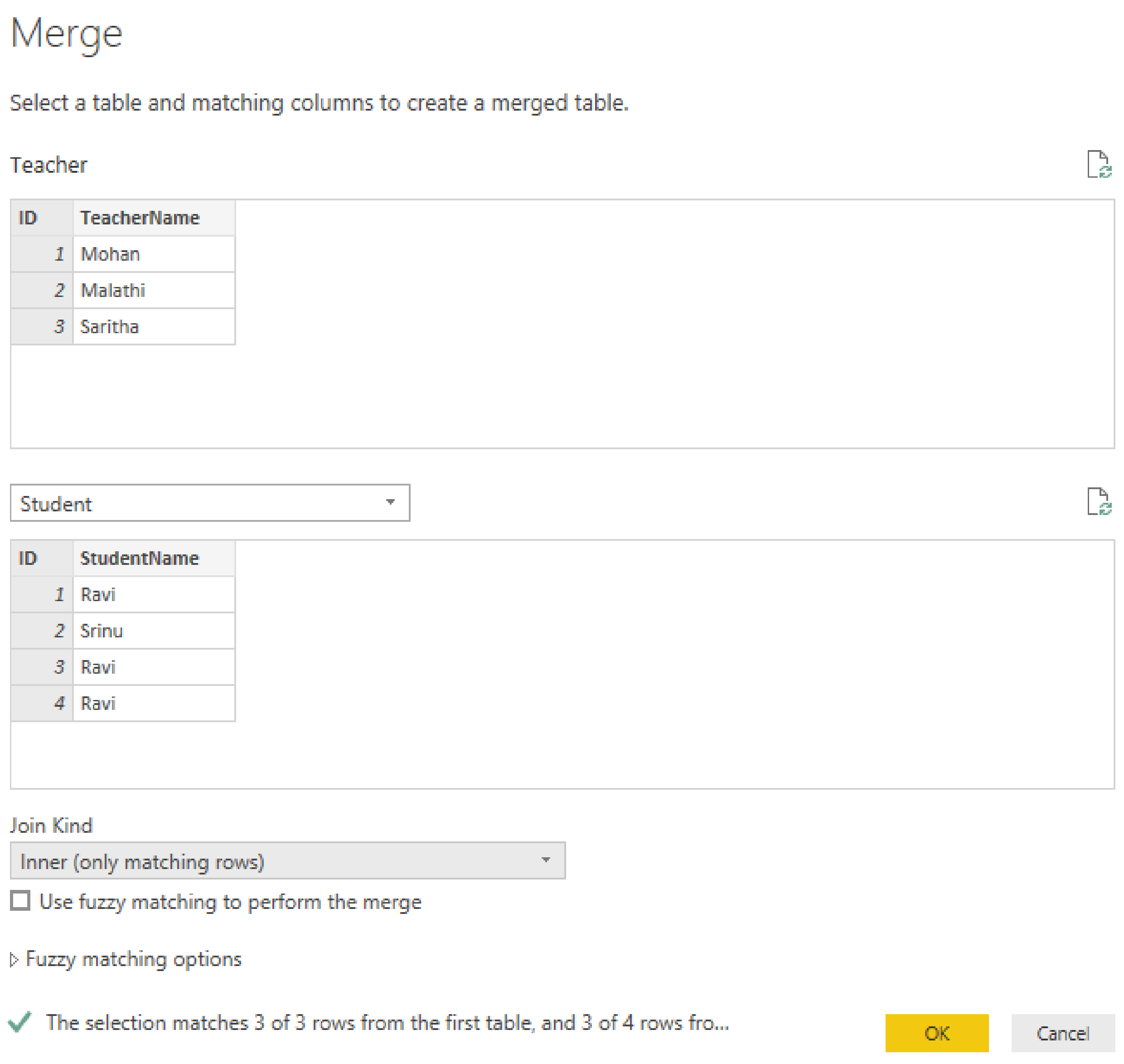


	ID	StudentName
1	1	Ravi
2	2	Srinu
3	3	Ravi
4	4	Ravi



	ID	TeacherName
1	1	Mohan
2	2	Malathi
3	3	Saritha

Power BI provides a feature to combine/merge the tables. (Joining conditions required)



Merge

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

Teacher

ID	TeacherName
1	Mohan
2	Malathi
3	Saritha

Student

ID	StudentName
1	Ravi
2	Srinu
3	Ravi
4	Ravi

Join Kind

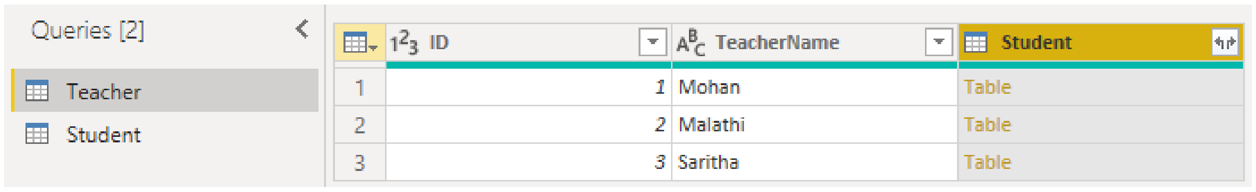
Inner (only matching rows)

☐ Use fuzzy matching to perform the merge

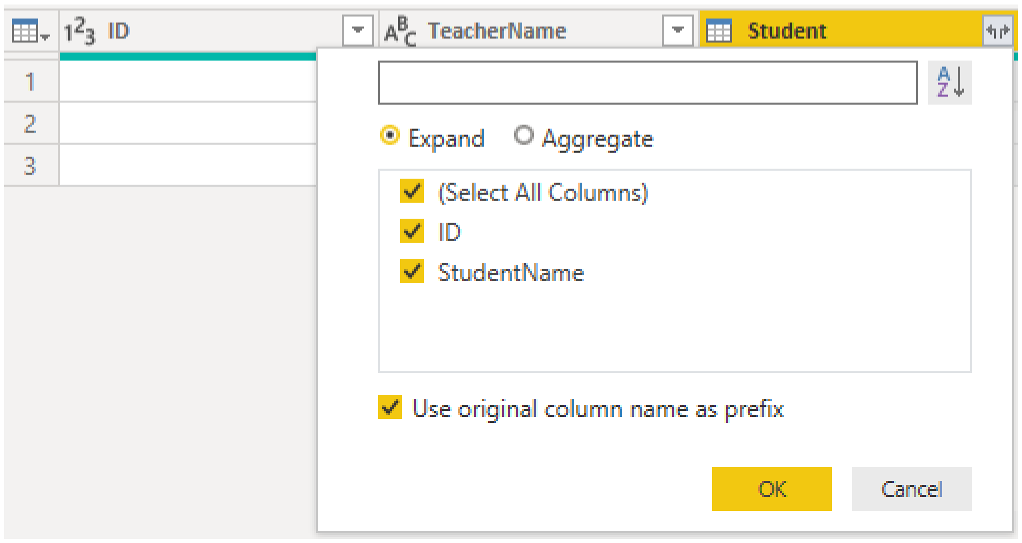
▷ Fuzzy matching options

✓ The selection matches 3 of 3 rows from the first table, and 3 of 4 rows fro...

After Clicked on OK



	ID	TeacherName	Student
1	1	Mohan	Table
2	2	Malathi	Table
3	3	Saritha	Table



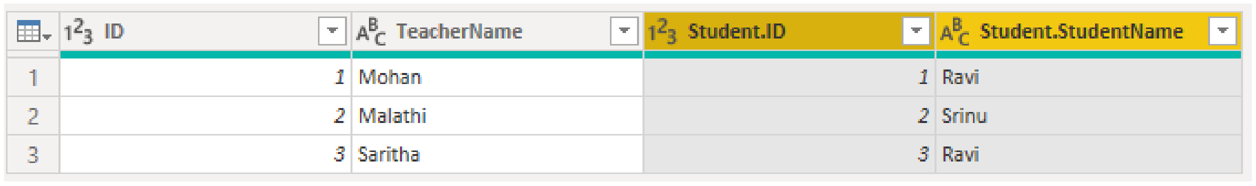
	ID	TeacherName	Student
1			
2			
3			

Expand [✓] Aggregate

- ✓ (Select All Columns)
- ✓ ID
- ✓ StudentName

✓ Use original column name as prefix

OK Cancel



	ID	TeacherName	Student.ID	Student.StudentName
1	1	Mohan	1	Ravi
2	2	Malathi	2	Srinu
3	3	Saritha	3	Ravi

When to go for Merge:

Here an Example:

We knew that In Snowflake Schema must have a relationship between 2 Dimension tables. In this case if we use merge these 2 dimensions table into a single table by using Merge Concept so that automatically schema also moved to Star Schema.

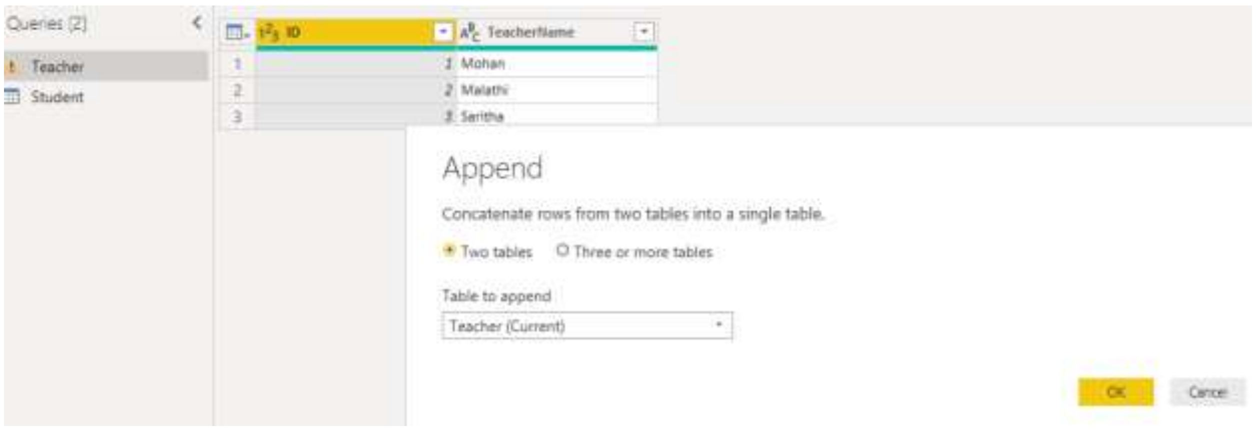
Merging the Tables using Multiple Join Conditions

How to Merge 3 Tables

Use Fuzzy Matching Option

Append:

It means appears another table data in the same table which we want to append it. (No Joining conditions required). We can call Append as a Union in Database world.



	ID	TeacherName
1	1	Mohan
2	2	Malathi
3	3	Saritha

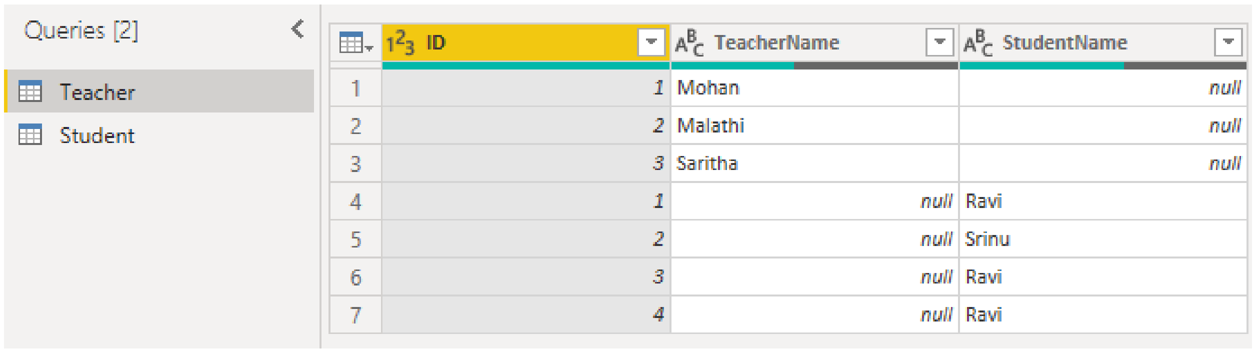
Append

Concatenate rows from two tables into a single table.

[✓ Two tables] Three or more tables

Table to append

Teacher (Current)



	ID	TeacherName	StudentName
1	1	Mohan	null
2	2	Malathi	null
3	3	Saritha	null
4	1	null	Ravi
5	2	null	Srinu
6	3	null	Ravi
7	4	null	Ravi

When to go for Append:

Here an Example:

Suppose we have 2 tables with same columns in our schema better to make a single table by using Append.

A simple explanation:

We have a “Prod” table in SQL server, Excel file. But we need these two tables in our requirement. In this case we will fetch Prod table from 2(SQL and Excel) Sources. So, after get Prod table in query editor we will have 2 mentions with prod table. So, we can union the both two Prod tables into a Single able by using Append functionality.

Query Options in Query Editor

- **a) Copy Query, Paste Query, Delete Query, Rename Query**
- **b) Enable Load, Include in report Refresh**
- **c) Duplicate Query, Reference Query**
- **d) Move to Group, Move Up, Move Down**
- **e) Create Function, Convert to Parameter**
- **f) Advanced Editor, Properties**

Data Modeling

Creating a Report is not called Data Modeling

1. **Dimension Column, Fact Column, Dimension Table, Fact Table**
   Fact column/Measure column:
   A column should satisfy 2 conditions,

- The data type of the column must be a number
- The number column needs to create an impact on the business

Dimension column.
The data type of the column may or may not be numeric.
All text columns and date columns are dimension columns.
In below example.
Channel Type and Customer Country are **Dimensions**
Sales Amount is a **Fact** column.

Channel type	Customer Country	Sales Amount
A	**india**	**100$**
B	**Usa**	**101$**

Fact (Data)= Data table contains a information about process. Data Table=Business Process

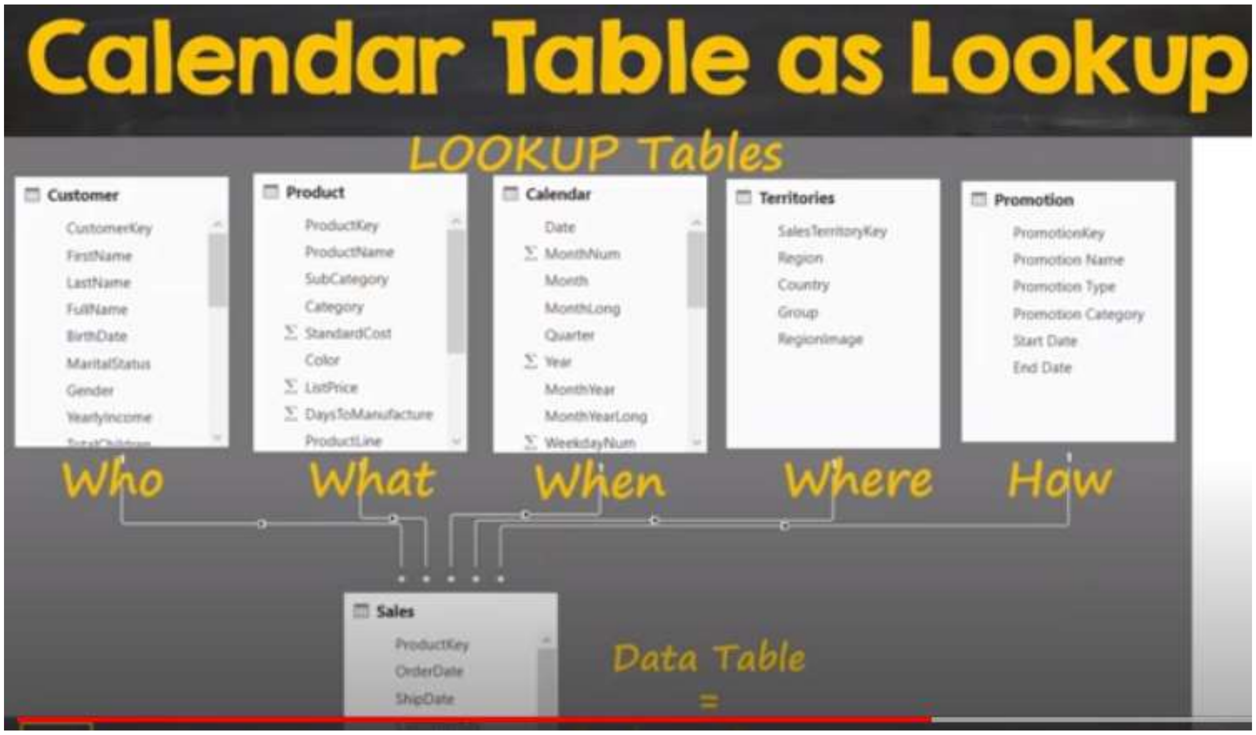
Dimension (Look-Up) =who what when where how

- (Calendar table also a lookup table)

Measures-Define ones use it everywhere

Data Table (Fact Table)	**Lookup Table (Dimensions)**
Business Process	Who, What, Where, When, How
Tall & Skinny	Shor & Squat
Numbers	Words

Relationships



One to Many Relationship:

Below one is an example of one to many. Why because one customer can purchase number of sales.

Customer		Sales
CustomerKey	$\longrightarrow$	CustomerKey
PurchasedDate		

One to One Relationship

Many to One Relationship

Many to Many Relationship

Schemas in Power BI

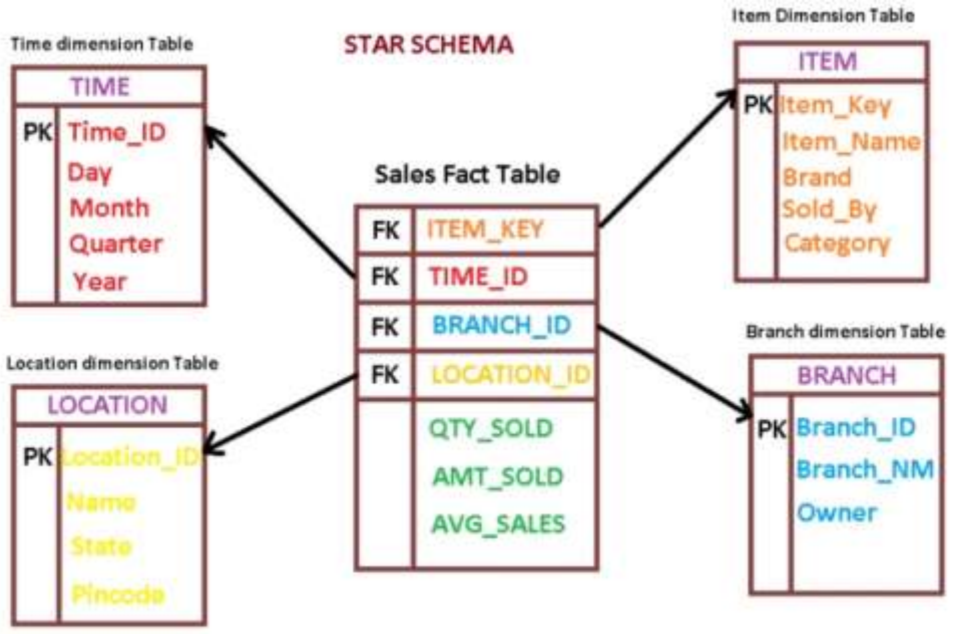
First, we need to understand Facts and Dimensions before we start Schemas.

Ex: Total Sales, Product Values these are numbers and I can say these are Measures. Measures are numbers.

Measures majorly stored in Fact Tables. And Dimensions describes what these measures mean.

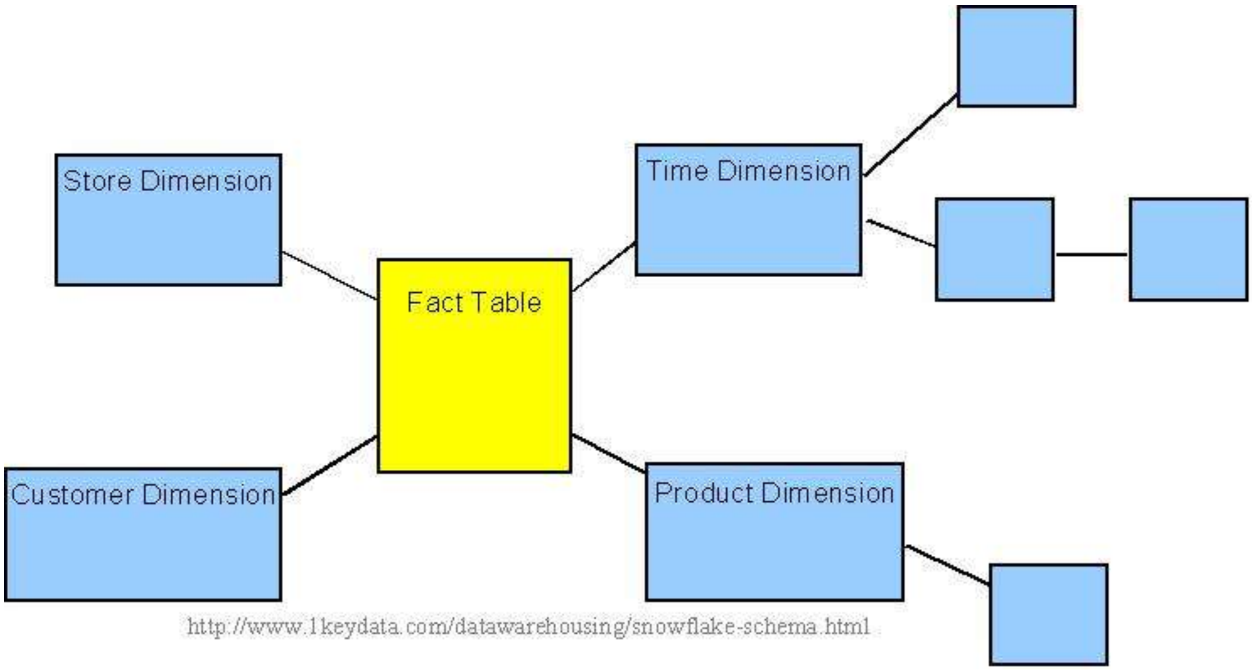
1)Star Schema

- Here Data is de-normalized. And Joining's are less. More Maintenance.



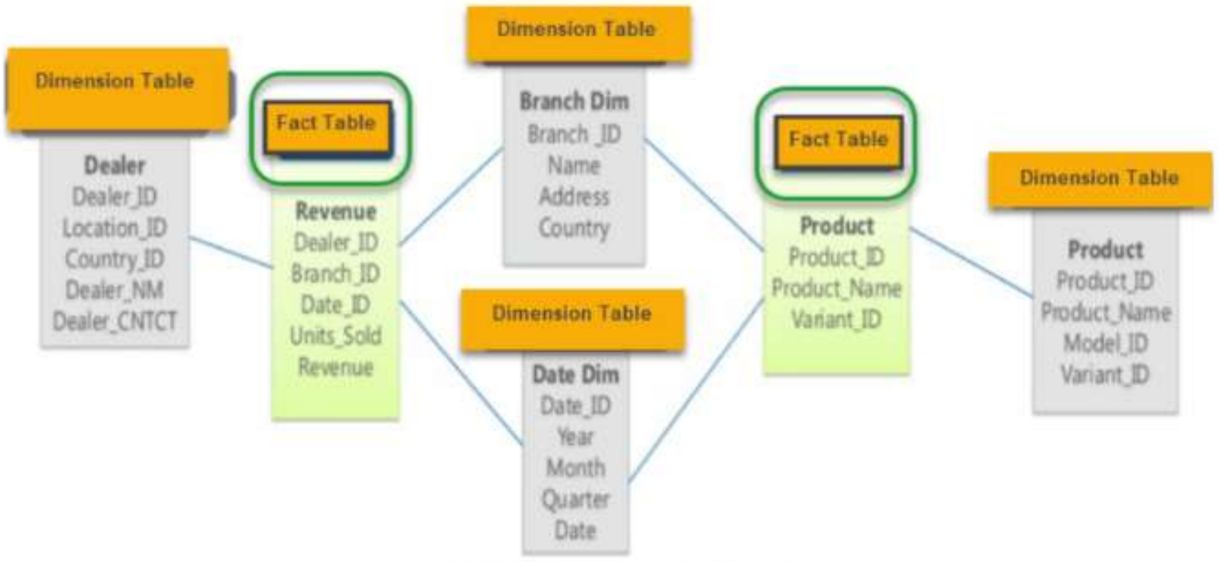
2)Snowflake Schema

- At least one of the dimension table should not have a relationship with Fact table.
- Here Data is normalized. And Joining's are more. Less Maintenance.
- Extended of star schema is called snowflake schema.



3) Galaxy Schema or Hybrid Schema or Fact Constellation Schema

A Galaxy Schema contains two fact table that share dimension tables between them. It is also called fact Constellation Schema. The schema is viewed as a collection of stars hence the name Galaxy Schema.



Example of Galaxy Schema

DAX Data Analysis Expression – Categories

Measures are 2 Types

- Implicit Measure-→SUM, MAX,MIN
- Explicit Measure→We have to write manually, by clicking new column, new table

Quick Measures

DAX is a functional language.

What DAX contain is Functions + Operators, with the help of functions and operators you will write the DAX.

- Text Functions
- Logical Functions
- Date and Time Functions
- Filter Functions
- Math & Statistical Functions
- Time Intelligence Functions
- Information functions

- Iterative Functions
- Parent and child functions
- Other functions
- Rank Functions and Variable

Text Functions:

CONCATENATE

Joins two text strings into one text string.

Syntax: CONCATENATE(<text1>, <text2>)

Scenario:new measure: =concatenate(sanjay, bi)

CONCATENATEX

Concatenates the result of an expression evaluated for each row in a table

Syntax: CONCATENATEX(<table>, <expression>, [delimiter])

CONCATENATEX(STUDENT, [FirstName] & " " & [LastName], ",")hlihkhi

EXACT

Compares two text strings and returns TRUE if they are exactly the same, FALSE otherwise. EXACT is case-sensitive but ignores formatting differences. You can use EXACT to test text being entered into a document.

Syntax:EXACT(<text1>,<text2>)

FIND

Returns the starting position of one text string within another text string. FIND is case-sensitive.

Syntax:FIND(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])

Example

The following formula finds the position of the first letter of the product designation, BMX, in the string that contains the product description.

=FIND("BMX","line of BMX racing goods")

LEFT

Returns the specified number of characters from the start of a text string.

Syntax: LEFT(<text>, <num_chars>)

Returns the number of characters in a text string.

RIGHT

RIGHT returns the last character or characters in a text string, based on the number of characters you specify.

Syntax: RIGHT(<text>, <num_chars>)

LEN

Returns the number of characters in a text string.

Syntax: LEN(<text>)

LOWER

Converts all letters in a text string to lowercase.

Syntax: LOWER(<TEXT>)

UPPER

Converts a text string to all uppercase letters

Syntax: UPPER(<TEXT>)

MID

Returns a string of characters from the middle of a text string, given a starting position and length.

Syntax: MID(<text>, <start_num>, <num_chars>)

REPLACE

REPLACE replaces part of a text string, based on the number of characters you specify, with a different text string.

Syntax: REPLACE(<old_text>, <start_num>, <num_chars>, <new_text>)

REPITITION

Repeats text a given number of times. Use REPT to fill a cell with a number of instances of a text string.

Syntax: REPT(<text>, <num_times>)

SEARCH

Returns the number of the character at which a specific character or text string is first found, reading left to right. Search is case-insensitive and accent sensitive.

Syntax: SEARCH(<find_text>, <within_text>[, [<start_num>][, <NotFoundValue>]])

TRIM

Removes all spaces from text except for single spaces between words.

Syntax: TRIM(<text>)

VALUE

Converts a text string that represents a number to a number.

Syntax: VALUE(<text>)

Logical Functions

Logical functions act upon an expression to return information about the values or sets in the expression. For example, you can use the IF function to check the result of an expression and create conditional results.

AND

Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE. Otherwise returns false.

Syntax: AND(<logical1>,<logical2>)

Scenario: New measure=IF(AND(4>3,3<4),"BOTH TRUE","ANYONE FALSE")

OR

Checks whether one of the arguments is TRUE to return TRUE. The function returns FALSE if both arguments are FALSE.

Syntax OR(<logical1>,<logical2>)

Scenario:New measure= IF(OR(1>3,3<4),"ONE TRUE","BOTH FALSE")

TRUE

Returns the logical value TRUE.

Syntax: TRUE()

Scenario:New measure=IF(SUM(FactPayments[Discount_Fee])=1900000, TRUE(),FALSE())

IF

Checks if a condition provided as the first argument is met. Returns one value if the condition is TRUE, and returns another value if the condition is FALSE.

Syntax: IF(logical_test>,<value_if_true>, value_if_false)

Scenario:New Column=IF(FactPayments[Tax amount]<200,"LOW",IF(FactPayments[Tax amount]<250,"MED","HIGH"))

IFERROR

Evaluates an expression and returns a specified value if the expression returns an error; otherwise returns the value of the expression itself.

Syntax: IFERROR(value, value_if_error)

Scenario:New Measure=IFERROR(24/0,9999)

IN

Returns True if the scalar value shows up in at least one row of the input relation.

Syntax: IN

Scenario:New Measure=CALCULATE(SUM(FactPayments[Discount_Fee]),DimCourse[CourseID] IN {"MSBI-F","MSBI-N"})

SWITCH (Applies multiple conditions)

Evaluates an expression against a list of values and returns one of multiple possible result expressions.

Syntax: SWITCH(<expression>, <value>, <result>[, <value>, <result>]…[, <else>])

Scenario: New column=Increment
disc_fee=switch(FactPayments[ModeID],"Online",FactPayments[Discount_Fee] *10/100,"Classroom",FactPayments[Discount_Fee] * 20/100,"Customized", FactPayments[Discount_Fee]* 30/100)

Date and Time Functions

CALENDAR

Returns a table with a single column named "Date" that contains a contiguous set of dates. The range of dates is from the specified start date to the specified end date, inclusive of those two dates.

Syntax: CALENDAR(<start_date>, <end_date>)

Examples

The following formula returns a table with dates between January 1st, 2020 and December 31st, 2020.

=CALENDAR (DATE (2020, 1, 1), DATE (2020, 12, 31))

DATE

Returns the specified date in datetime format.

Syntax: DATE(<year>, <month>, <day>)

Example: Returning a Simple Date

Code

=DATE (2020,3,28)

CALENDAR AUTO

Returns a table with a single column named "Date" that contains a contiguous set of dates. The range of dates is calculated automatically based on data in the model.

Syntax: CALENDARAUTO()

TODAY

Returns the current date.

Syntax: TODAY()

UTCToday

Returns the current UTC

Syntax: UTCTODAY()

UTCNow

Returns the current UTC date and time

Syntax: UTCNOW()

NOW

Returns the current date and time in datetime format.

Syntax:NOW()

DAY

Returns the day of the month, a number from 1 to 31.

Syntax:DAY(DATE)

WEEKDAY

Returns a number 2. By default the day ranges from 1 (Sunday) to 7 (Saturday).

Syntax: WEEKDAY(<date>, <return_type>)

MONTH

Returns the month as a number from 1 (January) to 12 (December).

Syntax: MONTH(<datetime>)

YEAR

Returns the year of a date as a four digit integer in the range 1900-9999.

Syntax: YEAR(<date>)

DATEDIFF

Returns the count of interval boundaries crossed between two dates.

Syntax: DATEDIFF(<start_date>, <end_date>, <interval>)

YEARFRAC

Calculates the fraction of the year represented by the number of whole days between two dates.

Syntax: YEARFRAC(<start_date>, <end_date>, <basis>).

EOMONTH

Returns the date in datetime format of the last day of the month, before or after a specified number of months.

Syntax : EOMONTH(<start_date>, <months>)

WEEKNUM

Returns the week number for the given date and year according to the return_type value. The week number indicates where the week falls numerically within a year.

Syntax: WEEKNUM(<date>, <return_type>)

EDATE

Returns the date that is the indicated number of months before or after the start date.

Syntax: EDATE(<start_date>, <months>)

DATEVALUE

Converts a date in the form of text to a date in datetime format.

Syntax:DATEVALUE(date_text)

TIMEVALUE

Converts a time in text format to a time in datetime format.

Syntax:TIMEVALUE(time_text)

HOUR

Returns the hour as a number from 0 (12:00 A.M.) to 23 (11:00 P.M.)

Syntax: HOUR(<datetime>)

MINUTE

Returns the minute as a number from 0 to 59, given a date and time value.

Syntax: MINUTE(<datetime>)

Filter Functions

Filtering operations to manipulate the data context.

Context

Context is one of the most important DAX concepts to understand. There are two types of context in DAX: row context and filter context. We'll first look at row context.

Row context

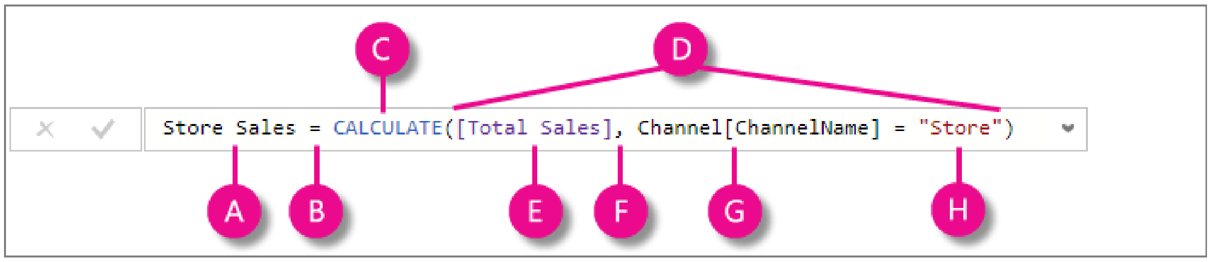
Row context is most easily thought of as the current row. It applies whenever a formula has a function that applies filters to identify a single row in a table. The function will inherently apply a row context for each row of the table over which it is filtering. This type of row context most often applies to measures.

Filter context

Filter context is a little more difficult to understand than row context. You can most easily think of filter context as: One or more filters applied in a calculation that determines a result or value.

Filter context doesn't exist in place of row context; rather, it applies in addition to row context. For example, to further narrow down the values to include in a calculation, you can apply a filter context, which not only specifies the row context, but also specifies a particular value (filter) in that row context.

Why is filter context so important to DAX? Because while filter context can most easily be applied by adding fields to a visualization, filter context can also be applied in a DAX formula by defining a filter using functions such as ALL, RELATED, FILTER, CALCULATE, by relationships, and by other measures and columns. For example, let's look at the following formula in a measure named Store Sales:



```
Store Sales = CALCULATE([Total Sales], Channel[ChannelName] = "Store")
```

This formula includes the following syntax elements:

A. The measure name, **Store Sales**.

B. The equals sign operator (=), which indicates the beginning of the formula.

C. The **CALCULATE** function, which evaluates an expression, as an argument, in a context that is modified by the specified filters.

D. Parenthesis **()**, which surround an expression containing one or more arguments.

E. A measure **[Total Sales]** in the same table as an expression. The Total Sales measure has the formula: =SUM(Sales[SalesAmount]).

F. A comma (**,**), which separates the first expression argument from the filter argument.

G. The fully qualified referenced column, **Channel[ChannelName]**. This is our row context. Each row in this column specifies a channel, such as Store or Online.

H. The particular value, **Store**, as a filter. This is our filter context.

This formula ensures only sales values defined by the Total Sales measure are calculated only for rows in the Channel[ChannelName] column, with the value *Store* used as a filter.

Filter

Returns a table that represents a subset of another table or expression.

Syntax:FILTER(<table>,<filter>)

Returns-A table containing only the filtered rows.(Returns a table that has been filtered.)

Scenario: Goto New Table=Filter(Dimcourse,Dimcourse[Coursename]="Azure Fast Track")

Filters

Returns the values that are directly applied as filters to columnName. i.e.,Returns a table of the filter values applied directly to the specified column.

Syntax:FILTERS(<columnName>)

Scenario: Goto New Table=Filters(Dimcourse[Coursename]='"Azure Fast Track")

All

Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied.This function is useful for clearing filters and creating calculations on all the rows in a table.

Syntax: ALL({<table> | <column>[, <column>[, <column>[,…]]]})

Scenario:

Take an year slicer and see the result, when year changes Discount total 1 changes, where as Discount total 2 does not change. Take two cards and show the result.

Discount total 2 = sumx(all(FactPayments),FactPayments[Discount_Fee])

Total discount fees = SUMX(FactPayments,FactPayments[Discount_Fee])

All Except

Removes all context filters in the table except filters that have been applied to the specified columns.

Syntax: ALLEXCEPT(<table>,<column>[,<column>[,…]])

Scenario: Take year filter,change year, it does not change the values.Where as Monthname if you change, it will affect.

Discount total 2 = calculate(sum(FactPayments[Discount_Fee]),allexcept(DimDate,DimDate[Month])

Calculate

Evaluates an expression in a context that is modified by the specified filters.

Syntax: CALCULATE(<expression>,<filter1>,<filter2>…)

Scenario:

Discount total 2 = calculate(sum(FactPayments[Discount_Fee]),allexcept(DimDate,DimDate[Month])

Calculate Table(we can create table of rows n columns)

CALCULATE function requires as its first argument an expression that returns a single value, the CALCULATETABLE function takes a table of values. Evaluates a table expression in a context modified by the given filters.

Syntax: CALCULATETABLE(<expression>,<filter1>,<filter2>,…)

Scenario: Display 2017 year data

New Table: Caltab2017 rows= CALCULATETABLE(FactPayments,DimDate[Year]=2017)

Scenario: Total Discount fee in 2017 using calculate table

New Measure:Caltab_2017_TDF=
sumx(CALCULATETABLE(FactPayments,DimDate[Year]=2017),FactPayments[Discount_Fee])

Count Rows and Distincts

Distinct rows with count of rows.

Syntax: DISTINCT(<table>)<column>

COUNTROWS(<column>)

Scenario: Count rows of distinct Discount fee in fact table.

DISTINCTROWCOUNT = COUNTROWS(DISTINCT(FactPayments[Discount_Fee]))

<u>Related</u>

Returns a related value from another table.

Syntax: RELATED (<COLUMN>)

Scenario: Create new column Coursedetail in Factpayments using Related function which joins fact table with DimCourse table with new column retrieving with values against each courseid(joining column) with course name details.

Goto Factpayments→NewColumn→Coursedetail = RELATED(DimCourse[Coursename])

Process: Each Courseid value from Factpayments passed to the model, join with Dimcourse table and then matched row Coursename it gets from Dimcourse table.

Join concept: One value pass and get one value against each matched courseid(1:1) —Join and return a single value against each matched id.

<u>Related Table</u>

Evaluates a table expression in a context modified by the given filters and return a table of values.

Syntax: Syntax RELATEDTABLE(<tableName>)

Scenario: Findout in Coursetable for each courseid total discount fee generated.

Goto Coursetable→TotalDF = SUMX(RELATEDTABLE(FactPayments),FactPayments[Discount_Fee])

Process: Each Courseid from DimCourse passed to Factpayments, compare and get table of rows and then sum of all the retrieved rows total created.

Join concept: One value passed and many values retrieval (1:many) —join and return table data.

<u>Values</u>

Returns a one-column table that contains the distinct values from the specified table or column. In other words, duplicate values are removed and only unique values are returned.

Syntax: VALUES(<TableNameOrColumnName>)

Scenario: Display only coursenames. Syntax: Course_values=Values(Dimcourse[coursename])

Scenario:Display count of coursename.

Syntax: goto new column=Count rows(Values(Dimcourse[coursename]))

<u>HasoneValue</u>

Returns TRUE when the context for columnName has been filtered down to one distinct value only. Otherwise is FALSE.

Syntax: HASONEVALUE(<columnName>)

Scenario: If courseID has one value, then it will show you count. Otherwise returns BLANK.

Goto New column
→Hasonevalue=IF(HASONEVALUE(FactPayments[CourseID]),FILTERS(FactPayments[CourseID]),BLANK())

HasoneFilter

Returns TRUE when the number of directly filtered values on columnName is one; otherwise returns FALSE.

Syntax: HASONEFILTER(<columnName>)

Scenario: If courseID has one value, then it will show you count. Otherwise returns BLANK.

Goto New column
→HasoneFilter=IF(HASONEFILTER(FactPayments[CourseID]),FILTERS(FactPayments[CourseID]),BLANK())

IsCrossFiltered

Returns TRUE when columnName or another column in the same or related table is being filtered.

Syntax: ISCROSSFILTERED(<columnName>)

Scenario: Goto New column→ Iscrossfilter = ISFILTERED(DimDate[Quarter])

IsFiltered

Returns TRUE when columnName is being filtered directly. If there is no filter on the column or if the filtering happens because a different column in the same table or in a related table is being filtered then the function returns FALSE.

Syntax: ISFILTERED(<columnName>)

Scenario: Goto New column→ Isfiltered = ISFILTERED(DimDate[Quarter])

Keepfilters

Modifies how filters are applied while evaluating a CALCULATE or CALCULATETABLE function.

Syntax: KEEPFILTERS(<expression>)

SelectedValue

Returns the value when the context for columnName has been filtered down to one distinct value only. Otherwise returns alternateResult.

Syntax: SELECTEDVALUE(<columnName>[, <alternateResult>])

Scenario: Goto→New Measure→= SELECTEDVALUE(DimCourse[Coursename],"XYZ"), use card for the measure values and coursename in slicer, if no value selected in slicer→card displays alternate value XYZ.

UseRelationship

Specifies the relationship to be used in a specific calculation as the one that exists between columnName1 and columnName2.

Syntax: USERELATIONSHIP(<columnName1>,<columnName2>)

Scenario:

DiscountFee_Relationship = CALCULATE(sum(FactPayments[Discount_Fee]), USERELATIONSHIP(FactPayments[Date],DimDate[Date]))

No value returned, but the relationship is utilized.

Math & Statistical Functions

The mathematical functions in Data Analysis Expressions (DAX) are very similar to the Excel mathematical and trigonometric functions.

SUM

Adds all the numbers in a column.

Syntax: SUM(<COLUMN>)

Example: The following example adds all the numbers that are contained in the column, Amt, from the table. **Sales=SUM(Sales[Amt])**

SUMX

Returns the sum of an expression evaluated for each row in a table.

Syntax: SUMX(<table>, <expression>)

The following example first filters the table, Factpayments, on the expression and then returns the sum of all values in the column, Discount_Fee.

Total discount fees = SUMX(FactPayments,FactPayments[Discount_Fee])

CEILING

Rounds a number up, to the nearest integer or to the nearest multiple of significance.

Syntax: CEILING(<number>, <significance>)

Example

The following formula returns 4.45. This might be useful if you want to avoid using smaller units in your pricing. If an existing product is priced at $4.42, you can use CEILING to round prices up to the nearest unit of five cents.

=CEILING(4.42,0.05)

CURRENCY

Evaluates the argument and returns the result as currency data type.

Syntax: CURRENCY(<value>)

Example Convert number 1234.56 to currency data type. =CURRENCY(1234.56) Returns the value $1234.5600.

<u>DIVIDE</u>

Performs division and returns alternate result or BLANK() on division by 0.

Syntax: DIVIDE(<numerator>, <denominator> [,<alternateresult>])

Example: The following example returns 2.5. =DIVIDE(5,2)

Example: The following example returns BLANK. =DIVIDE(5,0)

Example: The following example returns 1. =DIVIDE(5,0,1)

Time Intelligence Functions

These functions work on running dates of current and past.

<u>Sameperiod last year</u>

Returns a table that contains a column of dates shifted one year back in time from the dates in the specified dates column, in the current context.

Syntax : SAMEPERIODLASTYEAR(<dates>)

<u>Parallelperiod</u>

Returns a table that contains a column of dates that represents a period parallel to the dates in the specified dates column, in the current context, with the dates shifted a number of intervals either forward in time or back in time.

Syntax: PARALLELPERIOD(<dates>,<number_of_intervals>,<interval>)

<u>Previous year</u>

Returns a table that contains a column of all dates from the previous year, given the last date in the dates column, in the current context.

Syntax: PREVIOUSYEAR(<dates>[,<year_end_date>])

<u>Previous quarter</u>

Returns a table that contains a column of all dates from the previous quarter, based on the first date in the dates column, in the current context.

Syntax: PREVIOUSQUARTER(<dates>)

<u>Previous month</u>

Returns a table that contains a column of all dates from the previous month, based on the first date in the dates column, in the current context.

Syntax: PREVIOUSMONTH(<dates>)

Previous day

Returns a table that contains a column of all dates representing the day that is previous to the first date in the dates column, in the current context.

Syntax : PREVIOUSDAY(<dates>)

Datesbetween

Returns a table that contains a column of dates that begins with the start_date and continues until the end_date.

Syntax: DATESBETWEEN(<dates>,<start_date>,<end_date>)

Datesinperiod

Returns a table that contains a column of dates that begins with the start_date and continues for the specified number_of_intervals.

Syntax: DATESINPERIOD(<dates>,<start_date>,<number_of_intervals>,<interval>)

Start of Month

Returns the first date of the month in the current context for the specified column of dates.

Syntax: STARTOFMONTH(<dates>)

Start of Quarter

Returns the first date of the quarter in the current context for the specified column of dates.

Syntax: STARTOFQUARTER(<dates>)

Start of Year

Returns the first date of the year in the current context for the specified column of dates.

Syntax: STARTOFYEAR(<dates>)

End of Month

Returns the last date of the month in the current context for the specified column of dates.

Syntax :ENDOFMONTH(<dates>)

End of Quarter

Returns the last date of the quarter in the current context for the specified column of dates

Syntax: ENDOFQUARTER(<dates>)

End of Year

Returns the last date of the year in the current context for the specified column of dates.

Syntax: ENDOFYEAR(<dates> [,<year_end_date>])

Dates YTD

Returns a table that contains a Column of the dates for the year to date, in the current context.

Syntax: DATESYTD(<dates> [,<year_end_date>])

Dates QTD

Returns a table that contains a column of the dates for the quarter to date, in the current context.

Syntax: DATESQTD(<dates>)

Dates MTD

Returns a table that contains a column of the dates for the month to date, in the current context.

Syntax: DATESMTD(<dates>)

Total YTD

Evaluates the year-to-date value of the expression in the current context.

Syntax: TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])

Total QTD

Evaluates the value of the expression for the dates in the quarter to date, in the current context.

Syntax: TOTALQTD(<expression>,<dates>[,<filter>])

Total MTD

Evaluates the value of the expression for the month to date, in the current context.

Syntax: TOTALMTD(<expression>,<dates>[,<filter>])

Firstdate

Returns the first date in the current context for the specified column of dates.

Syntax: FIRSTDATE(<dates>)

FirstNonBlank

Returns the first value in the column, column, filtered by the current context, where the expression is not blank.

Syntax: FIRSTNONBLANK(<column>,<expression>)

Lastdate

Returns the last date in the current context for the specified column of dates.

Syntax: LASTDATE(<dates>)

LastNonblankdate

Returns the last value in the column, column, filtered by the current context, where the expression is not blank.

Syntax: LASTNONBLANK(<column>,<expression>)

Nextday

Returns a table that contains a column of all dates from the next day, based on the first date specified in the dates column in the current context

Syntax: NEXTDAY(<dates>)

Nextmonth

Returns a table that contains a column of all dates from the next month, based on the first date specified in the dates column in the current context

Syntax: NEXTMONTH(<dates>)

NextQuarter

Returns a table that contains a column of all dates from the next quarter, based on the first date specified in the dates column in the current context

Syntax: NEXTQUARTER(<dates>)

Information functions

DAX information functions look at the cell or row that is provided as an argument and tells you whether the value matches the expected type. For example, the ISERROR function returns TRUE if the value that you reference contains an error.

CONTAINS

Returns true if values for all referred columns exist, or are contained, in those columns; otherwise, the function returns false.

Syntax: CONTAINS(<table>, <columnName>, <value>[, <columnName>, <value>]…)

Scenario: Return true or false, if courseID=”MSBI-F” and ModeID=”Online” available in fact table

Course_Mode_Exists = CONTAINS(FactPayments,FactPayments[CourseID],"MSBI-F", FactPayments[ModeID],"Online")

IN Operator / CONTAINSROW

Returns TRUE if a row of values exists or contained in a table, otherwise returns FALSE. Except syntax, the IN operator and CONTAINSROW function are functionally equivalent.

IN Operator Syntax :

<scalarExpr> IN <tableExpr>

(<scalarExpr1>, <scalarExpr2>, …) IN <tableExpr>

ContainsRow

syntax:

CONTAINSROW(<tableExpr>, <scalarExpr>[, <scalarExpr>, …])

Scenario: Goto→New Table→ ContainsRow = FILTER(ALL(DimCourseMode[ModeID]), CONTAINSROW({ "Online", "Classroom" }, [ModeID]))

Query Execution:Bring All modeids irrespective of flter.

Within the list filter function applies condition {containsrow}.

Note: Contains gives single values only as true/false, against single column.

Where as Contains row gives multiple values against multiple columns in a row.

IsError

Checks whether a value is an error, and returns TRUE or FALSE.

Syntax: ISERROR(<value>)

Iserror_val = iserror([Columname]/0)

Scenario: If there is an error substistuing with 99999 if(iserror(FactPayments[M1]/0),99999)

IsBlank

Checks whether a value is blank, and returns TRUE or FALSE.

Syntax: ISBLANK(<value>)

Scenario: Goto→New column→ = ISBLANK(DimCourse[CourseID])

Checks the value against courseid column and returns true/false.

IsEven & IsOdd

If values is Even result as True and if value is Odd result as false.

Isblank = ISBLANK(DimCourse[CourseID])

Syntax=ISEVEN(number)

Scenario: = ISEVEN(DimCourse[Duration]) returns result as True, duration values are all even.

= ISOdd(DimCourse[Duration]) returns result as False, duration values are all even.

IsNumber

Checks whether a value is a number, and returns TRUE or FALSE.

Syntax: ISNUMBER(<value>)

IsText

Checks if a value is text, and returns TRUE OR FALSE

Syntax: ISTEXT(<VALUE>)

IsLogical

Checks whether a value is a logical value, (TRUE or FALSE/Boolean values) and returns TRUE or FALSE.

Syntax: ISLOGICAL(<value>)

IsNontext

Checks if a value is not text (blank cells are not text), and returns TRUE or FALSE.

Syntax: ISNONTEXT(<value>)

IsonOrafter

A boolean function that emulates the behavior of a 'Start At' clause and returns true for a row that meets all of the condition parameters.

This function takes a variable number of triples, the first two values in a triple are the expressions to be compared, and the third parameter indicates the sort order. The sort order can be ascending (default) or descending.

Based on the sort order, the first parameter is compared with the second parameter. **If the sort order is ascending, the comparison to be done is first parameter greater than or equal to second parameter.** If the sort order is descending, the comparison to be done is second parameter less than or equal to first parameter.

Syntax: ISONORAFTER(<scalar_expression>, <scalar_expression>sort_order] [,scalar_expression>, <scalar_expression>, [sort_order][,…])

Scenario: ison = ISONORAFTER(DimDate[Date],DimDate[Month],ASC)→results as True

ison = ISONORAFTER(DimDate[Date],DimDate[Month],DESC)→results as False

LookupValue

Returns the value in result_columnName for the row that meets all criteria specified by search_columnName and search_value.

Syntax:LOOKUPVALUE(<result_columnName>, <search_columnName>, <search_value>[, <search_columnName>, <search_value>]…)

Scenario: Goto→New Measure→ = LOOKUPVALUE(DimCourse[Coursename],DimCourse[CourseID],"msbi-c")

UserName

Returns the domain name and username from the credentials given to the system at connection time

Syntax: USERNAME()

Scenario: Goto→Measure→=Username()

Iterative Functions

Parent and child functions

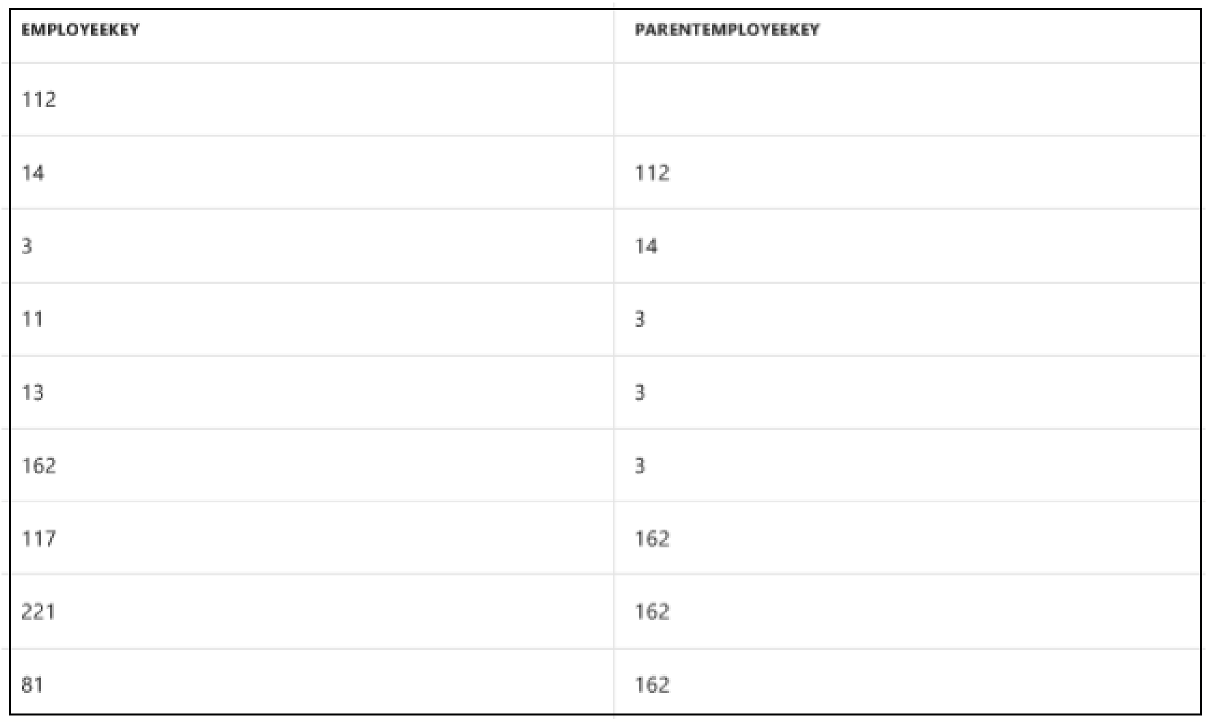
These Data Analysis Expressions (DAX) functions help users manage data that is presented as a parent/child hierarchy in their data models.

With these functions:

- ✓ A user can obtain the entire lineage of parents a row has,
- ✓ How many levels has the lineage to the top parent,
- ✓ Who is the parent n-levels above the current row,
- ✓ Who is the n-descendant from the top of the current row hierarchy
- ✓ And is certain parent a parent in the cursent row hierarchy?

Parent-Child functions in DAX

The following table contains a Parent-Child hierarchy on the columns: EmployeeKey and ParentEmployeeKey that is used in all the functions examples.

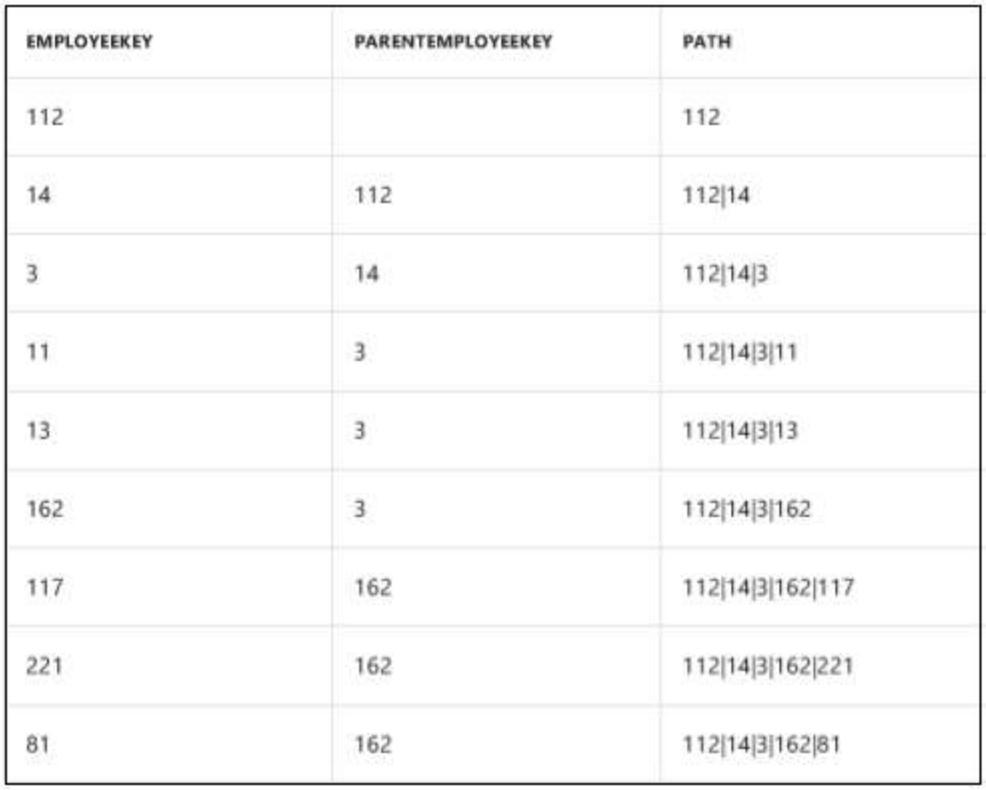


EMPLOYEEKEY	PARENTEMPLOYEEKEY
112	
14	112
3	14
11	3
13	3
162	3
117	162
221	162
81	162

In the above table you can see that employee 112 has no parent defined, employee 14 has employee 112 as manager (ParentEmployeeKey), employee 3 has employee 14 as manager and employees 11, 13, and 162 have employee 3 as manager. The above helps to understand that employee 112 has no manager above her/him and she/he is the top manager for all employees shown here; also, employee 3 reports to employee 14 and employees 11, 13, 162 report to 3.

The following table presents the available functions, a brief description of the function and an example of the function over the same data shown above.

PATH function (DAX) - Returns a delimited text with the identifiers of all the parents to the current row, starting with the oldest or top most until current.



EMPLOYEEKEY	PARENTEMPLOYEEKEY	PATH
112		112
14	112	112|14
3	14	112|14|3
11	3	112|14|3|11
13	3	112|14|3|13
162	3	112|14|3|162
117	162	112|14|3|162|117
221	162	112|14|3|162|221
81	162	112|14|3|162|81

Remarks

Returns a delimited text string with the identifiers of all the parents of the current identifier, starting with the oldest and continuing until current.

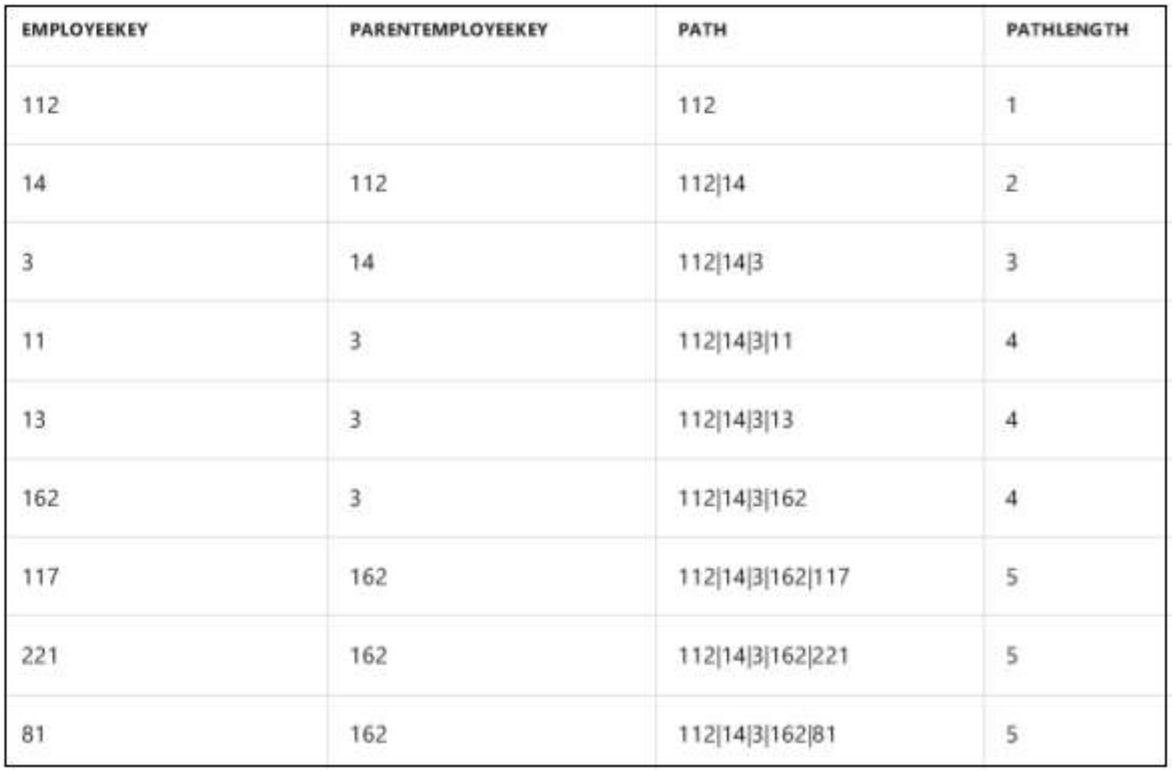
A delimited text string containing the identifiers of all the parents to the current identifier.

This function is used in tables that have some kind of internal hierarchy, to return the items that are related to the current row value. For example, in an Employees table that contains employees, the managers of employees, and the managers of the managers, you can return the path that connects an employee to his or her manager. The path is not constrained to a single level of parent-child relationships; it can return related rows that are several levels up from the specified starting row.

- ✓ The delimiter used to separate the ascendants is the vertical bar, '|'.
- ✓ The values in ID_columnName and parent_columnName must have the same data type, text or integer.
- ✓ Values in parent_columnName must be present in ID_columnName. That is, you cannot look up a parent if there is no value at the child level.
- ✓ If parent_columnName is BLANK then PATH() returns ID_columnName value. In other words, if you look for the manager of an employee but the parent_columnName column has no data, the PATH function returns just the employee ID.
- ✓ If ID_columnName has duplicates and parent_columnName is the same for those duplicates then PATH() returns the common parent_columnName value; however, if parent_columnName value is different for those duplicates then PATH() returns an error. In other words, if you have two listings for the same employee ID and they have the same manager ID, the PATH function returns the ID for that manager. However, if there are two identical employee IDs that have different manager IDs, the PATH function returns an error.

- ✓ If ID_columnName is BLANK then PATH() returns BLANK.
- ✓ If ID_columnName contains a vertical bar '|' then PATH() returns an error.

PATHLENGTH function (DAX) - Returns the number of levels in a given PATH(), starting at current level until the oldest or top most parent level. In the following example column PathLength is defined as ' =PATHLENGTH([Path]) '; the example includes all data from the Path() example to help understand how this function works.



EMPLOYEEKEY	PARENTEMPLOYEEKEY	PATH	PATHLENGTH
112		112	1
14	112	112|14	2
3	14	112|14|3	3
11	3	112|14|3|11	4
13	3	112|14|3|13	4
162	3	112|14|3|162	4
117	162	112|14|3|162|117	5
221	162	112|14|3|162|221	5
81	162	112|14|3|162|81	5

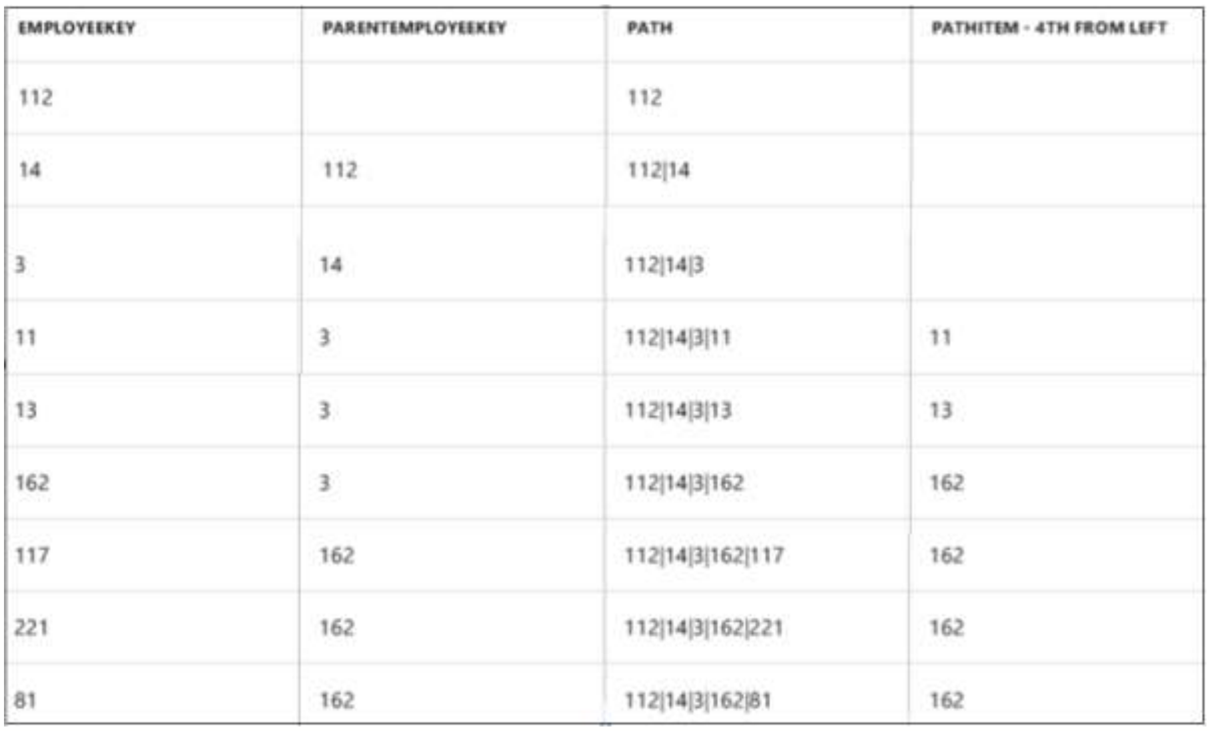
Remarks

This DAX function is not supported for use in DirectQuery mode.

PATHITEM function (DAX) - Returns the item at the specified position from a PATH() like result, counting from left to right. In the following example column PathItem - 4th from left is defined as ' =PATHITEM([Path], 4) '; this example returns the EmployeKey at fourth position in the Path string from the left, using the same sample data from the Path() example.

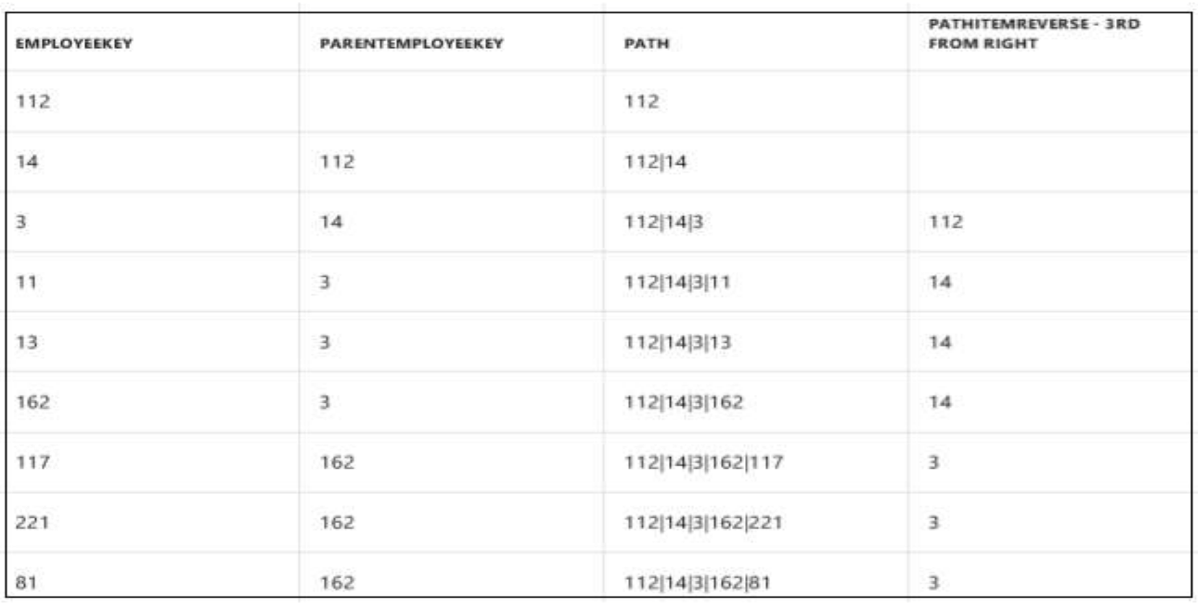
Remarks

- ✓ This function can be used to return a specific level from a hierarchy returned by a PATH function. For example, you could return just the skip-level managers for all employees.
- ✓ If you specify a number for position that is less than one (1) or greater than the number of elements in path, the PATHITEM function returns BLANK
- ✓ If type is not a valid enumeration element an error is returned.



EMPLOYEEKEY	PARENTEMPLOYEEKEY	PATH	PATHITEM - 4TH FROM LEFT
112		112	
14	112	112|14	
3	14	112|14|3	
11	3	112|14|3|11	11
13	3	112|14|3|13	13
162	3	112|14|3|162	162
117	162	112|14|3|162|117	162
221	162	112|14|3|162|221	162
81	162	112|14|3|162|81	162

<u>PATHITEMREVERSE function (DAX)</u> - Returns the item at position from a PATH() like function result, counting backwards from right to left. In the following example column PathItemReverse - 3rd from right is defined as ‘=PATHITEMREVERSE([Path], 3) ’; this example returns the EmployeKey at third position in the Path string from the right, using the same sample data from the Path() example.



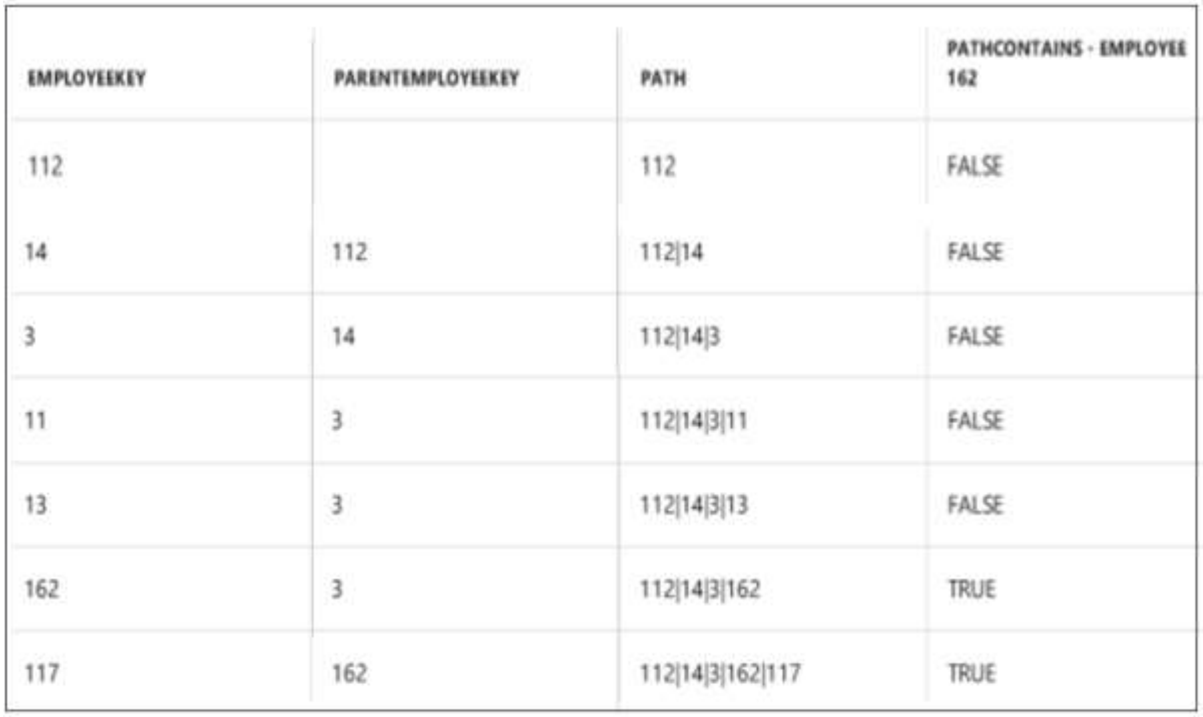
EMPLOYEEKEY	PARENTEMPLOYEEKEY	PATH	PATHITEMREVERSE - 3RD FROM RIGHT
112		112	
14	112	112|14	
3	14	112|14|3	112
11	3	112|14|3|11	14
13	3	112|14|3|13	14
162	3	112|14|3|162	14
117	162	112|14|3|162|117	3
221	162	112|14|3|162|221	3
81	162	112|14|3|162|81	3

Remarks

- ✓ This function can be used to get an individual item from a hierarchy resulting from a PATH function.

- ✓ This function reverses the standard order of the hierarchy, so that closest items are listed first, For example, if the PATh function returns a list of managers above an employee in a hierarchy, the PATHITEMREVERSE function returns the employee's immediate manager in position 2 because position 1 contains the employee's id.
- ✓ If the number specified for position is less than one (1) or greater than the number of elements in path, the PATHITEM function returns BLANK.
- ✓ If type is not a valid enumeration element an error is returned.

PATHCONTAINS function (DAX) - Returns TRUE if the specified item exists within the specified path. In the following example column PathContains - employee 162 is defined as ' =PATHCONTAINS([Path], "162") '; this example returns TRUE if the given path contains employee 162. This example uses the results from the Path() example above.



EMPLOYEEKEY	PARENTEMPLOYEEKEY	PATH	PATHCONTAINS - EMPLOYEE 162
112		112	FALSE
14	112	112|14	FALSE
3	14	112|14|3	FALSE
11	3	112|14|3|11	FALSE
13	3	112|14|3|13	FALSE
162	3	112|14|3|162	TRUE
117	162	112|14|3|162|117	TRUE

Remarks

If item is an integer number it is converted to text and then the function is evaluate d. If conversion fails then the function returns an error.

Rank Functions and Variable

Other functions

Provides a mechanism for declaring an inline set of data values. (Similar to Create Table - Helps to create a dynamic table with rows.

Syntax: DATATABLE (ColumnName1, DataType1, ColumnName2, DataType2..., {{Value1, Value2...}, {ValueN, ValueN+1...}...})

Scenario: NEW TABLE=STUDENT TABLE=DataTable("IDNO", STRING, "NAME", STRING ,{ {"STU1","AKASH"}, {" STU2","VISHU"}, {" STU3","RAGHU"}, {"STU4","KIRAN"}, {" STU5","KRISH"} })

EXCEPT

Returns the rows of one table which do not appear in another table. Syntax EXCEPT(<table_expression1>, <table_expression2>)

Scenario:

NEW TABLE ST1 = DataTable("IDNO", STRING, "NAME", STRING ,{ {"STU1","AKASH"}, {" STU2","VISHU"}})

NEW TABLE ST2 = DataTable("IDNO", STRING, "NAME", STRING ,{ {"STU1","AKASH"}, {"STU3","KIRAN"}})

NOW: EXSTU1STU2 = EXCEPT('NEW TABLE ST1','NEW TABLE ST2')

UNION

Creates a union (join) table from a pair of tables.

Syntax: UNION(<table_expression1>, <table_expression2> [,<table_expression>]…)

Scenario: USTU1_STU2= UNION('NEW TABLE ST1','NEW TABLE ST2')

INTERSECT

Returns the row intersection of two tables, retaining duplicates.

Syntax : INTERSECT(<table_expression1>, <table_expression2>)

Scenario:INTSTU1_STU2= INTERSECT('NEW TABLE ST1','NEW TABLE ST2')

GENERATESERIES

Returns a single column table containing the values of an arithmetic series, that is, a sequence of values in which each differs from the preceding by a constant quantity. The name of the column returned is Value.

Syntax: GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])

Scenario:New Table= GENERATESERIES(1,10,2)

Returns a table with a single column, starting with 1 and increment by 2.

GROUP BY

The GROUPBY function is similar to the SUMMARIZE function. However, GROUPBY does not do an implicit CALCULATE for any extension columns that it adds. GROUPBY permits a new function, CURRENTGROUP(), to be used inside aggregation functions in the extension columns that it adds. GROUPBY attempts to reuse the data that has been grouped making it highly performant.

Syntax: GROUPBY (<table>, [<groupBy_columnName1>], [<name>, <expression>]…)

Scenario: Inst_Mode_Total = GROUPBY(FactPayments,FactPayments[InstituteID],FactPayments[ModeID],"Total Fee",SUMx(currentgroup(),FactPayments[Discount_Fee]))

SUMMARIZECOLUMNS

Strict comparison semantics are used during join. There is no type coercion; for example, 1 does not equal 1.0. Returns a summary table over a set of groups.

Syntax: SUMMARIZECOLUMNS(<groupBy_columnName> [, < groupBy_columnName >]…, [<filterTable>]…[, <name>, <expression>]…)

Scenario: Inst_Mode_Total = SUMMARIZECOLUMNS(FactPayments[InstituteID],FactPayments[ModeID],FactPayments,"Total",SUM(FactPayments[Discount_Fee]))

TREATAS

Applies the result of a table expression as filters to columns from an unrelated table.

Syntax: TREATAS(table_expression, <column>[, <column>[, <column>[,…]]]})

Scenario:Finding the total from Dimcourse and Dimcourse_new based on selection from Dimcourse query.

1.create another table Dimcourse_new with similar few courseid values.

2. create connection with fact table.

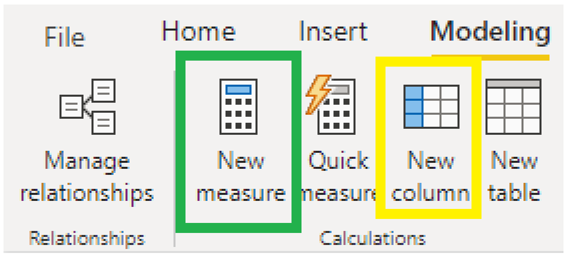
3. Take slicer and use courseid

4. create measure like below and use in card.

New Measure=Diff_Tablesum= calculate(sum(FactPayments[Discount_Fee]), treatas(values(DimCourse[CourseID]),DimCourse_New[CourseID]))

There are 2 types of calculations in DAX.

- Calculated Measures
- Calculated column



Calculated column	Calculated Measures
Calculated column are created during the data load (That means during refresh of the data). It is stored as part of the data table. Calculated column is always a row level. It calculates row by row.	New measure is created is on the fly that means it is not stored as part of the data model, but it is created based on the context of visualization.
Calculated columns are faster because the values are be stored	Slower
Do not prefer Calculated column for Ratio(coloumn1/coloumn2) activities.	For Ratio, you need to create a calculated measure
When you are creating make sure that click on the particular table which going to you are creating calculated column.	When creating Calculated measures, no need to click on a table why because we can drag measure to a required table.

implicit Measure

Q) Why Explicit measures are better than Implicit measures

There are 3 reasons

- Control
- Re-Use
- Connected Reports

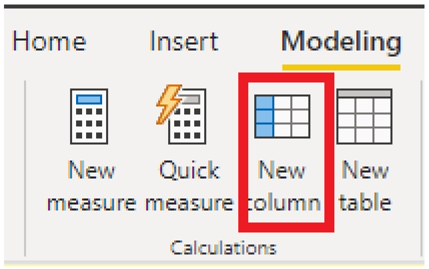
Calculated Columns

A calculated column is a new column that you can created by defining of a calculation that transforms or combines two or more elements of existing Data. Creating a calculate column is a simple way to enrich and enhance your data.

For example, you can create a new column by combing two columns into one.

One useful reason for creating calculated column is to a establish a relationship between two tables, when no unique field exist that can be used to establish a relationship.

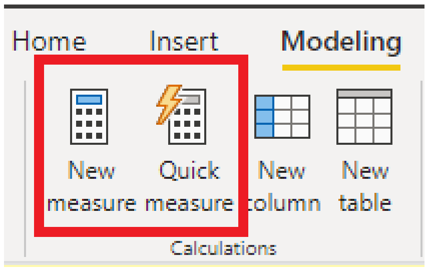
To create a measure in a report view →Modeling Tab→New Column



Calculated Measure

Is a calculation that exists in your Power BI data model.

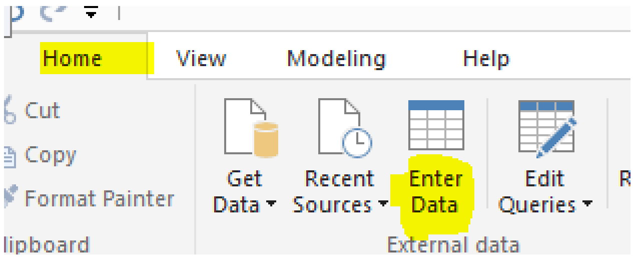
To create a measure in a report view →Modeling Tab→New Measure



Drill Through

Entering Data

Here we can manually create a table also we can copy the data from excel to here.



Calendar Table

I can say that if there is a Date field in table, you better have to create a calendar table.

If a sales table start date 2014 and end date is 2016, the calendar table range must cover 2014 to 2017.then only we can cover all the data.

There are two benefits of creating a calendar table.

1)Time Saving—We can perform Date functions easily.

2)Look up table-We can take Date/Month as keys in this lookup table.

There are many ways to create a calendar table.

- 1)Excel
- 2)SQL
- 3)Azure
- 4)DAX (Calendar Auto)

But my favorite method is Using Power Bi Query Editor by using DAX function.

Power BI Desktop>Modeling tab>New Table>Enter below DAX function

```
D_Calendar = ADDCOLUMNS ( CALENDAR (DATE(2000,1,1), DATE(2049,12,31)), "Date_Int",
FORMAT ( [Date], "YYYYMMDD" ), "Year", YEAR ( [Date] ), "Month", FORMAT ( [Date], "MM" ),
"Year_Month", FORMAT ( [Date], "YYYY-MM" ), "Year_Month_Name", FORMAT ( [Date], "YYYY-
mmm" ), "Month_Name", FORMAT ( [Date], "mmm" ), "Week", WEEKDAY ( [Date] ), "Week_Day",
FORMAT ( [Date], "dddd" ), "Qtr", "Q" & FORMAT ( [Date], "Q" ), "Year_Qtr", FORMAT ( [Date],
"YYYY" ) & "-Q" & FORMAT ( [Date], "Q" ))
```

Creating and Managing Hierarchies

To help your users navigate through your data, you can create hierarchies to help them understand the different levels in your data and how can they move up and down between them.

Time Intelligence

Do exercise for Date field by using Drill Up and Drill down options

Manually typing Data in Data table

Yes, we can able to add data manually for table that table we created in query editor by using Enter Data option in query editor.

Include and Exclude

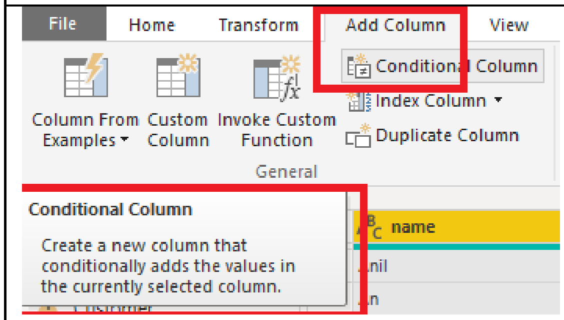
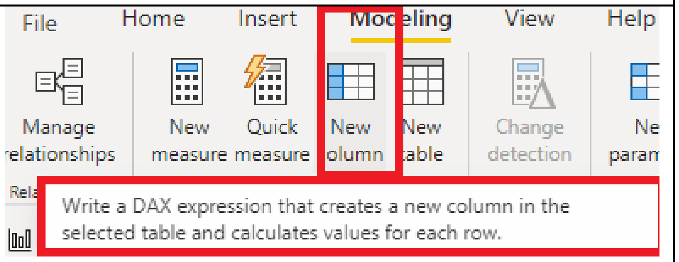
Difference between Copy, Duplicate, Reference in Query Editor

When to use:

Copy	**Duplicate**	**Reference**
	After creating a Duplicate table, it will be shown all applied steps from source table.	After creating a Reference table, it will not be shown all applied steps from source table.
	If we change anything in source table those changes never reflected in Duplicate table	If we change anything in source table those changes also reflected in Reference table

	If we change anything in Duplicate table, it will not affect Source table	If we change anything in Reference table, it will not affect Source table

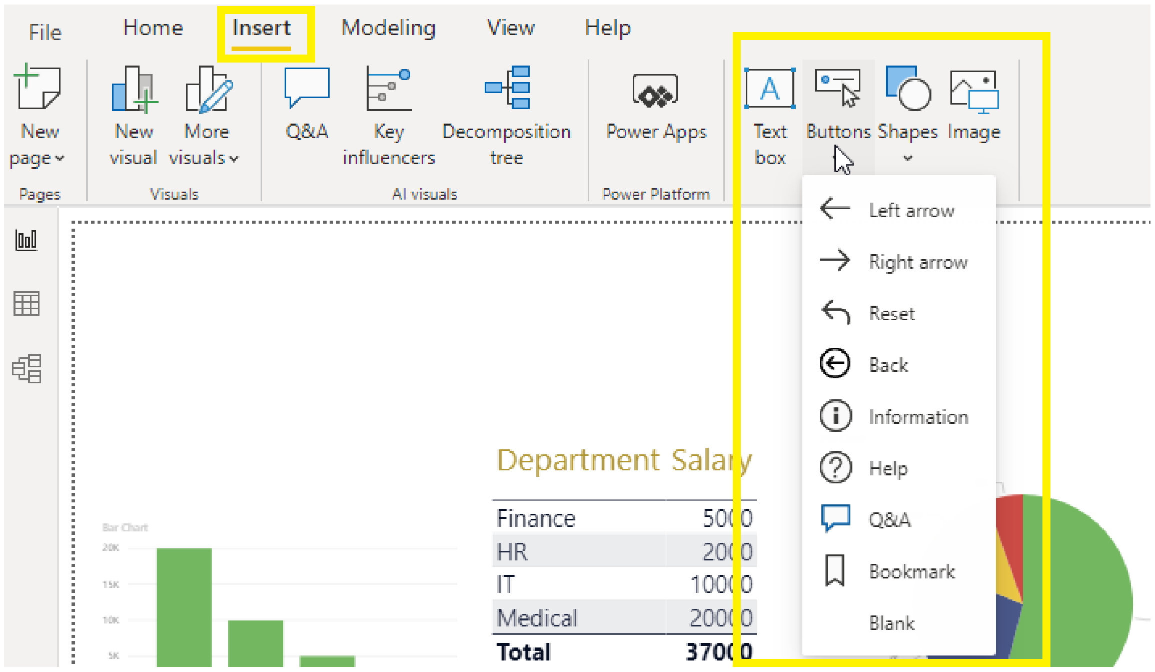
Diff b/w Added column (Conditional Column) and Calculated column

Add column- (Conditional column)	Calculated column
Conditional column you can find in Query Editor.	Calculated Column you can find in Power Bi Desktop
Hear Conditions are used.	DAX expressions are used hear.
 File Home Transform Add Column View Column From Examples Custom Column Invoke Custom Function Conditional Column Index Column Duplicate Column General name nil n Conditional Column Create a new column that conditionally adds the values in the currently selected column.	 File Home Insert Modeling View Help Manage relationships New measure Quick measure New column New table Change detection New param Write a DAX expression that creates a new column in the selected table and calculates values for each row.

Buttons, Selection, Bookmarks and Toggle Button.

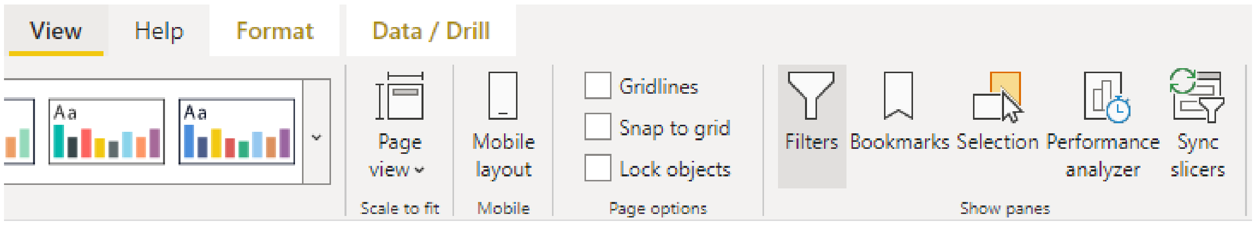
Buttons

- There is list of buttons available in Power BI desktop.
- If you want to create a Button with Some text go to last option **BLANK**. After clicking that button an empty button will be appear in your Desktop. You can give text to that Button by using button properties.



Bookmarks and Selection

For creating Bookmarks and Selection you just navigate to View tab. There you find list of options seen in below.



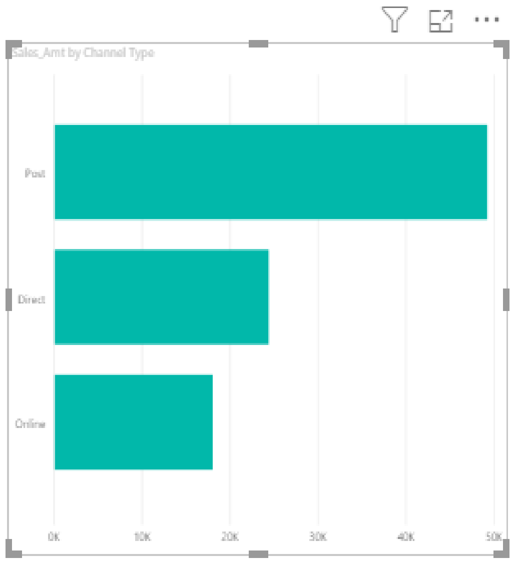
Row Level Security

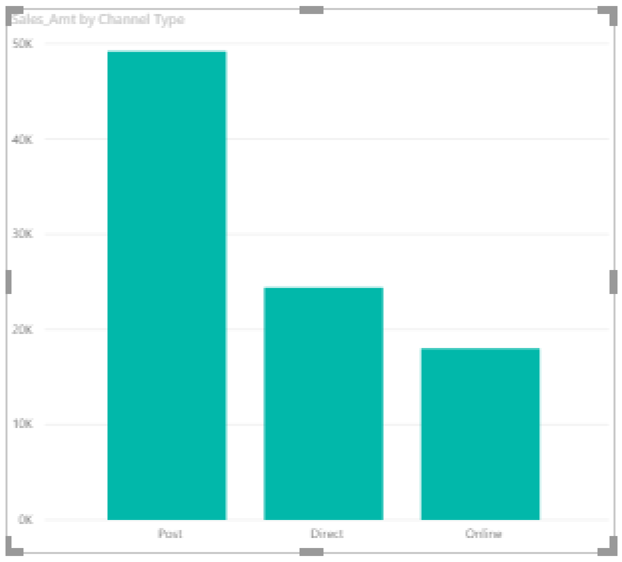
Row-Level security allows you to secure your data at the row level with security filters based on roles you create. Using the Manage roles button in the Modeling Tab, you an create roles then use DAX expressions to control what each role sees. Then, once you publish it up to the web, you can assign users to each role.

Power BI Visualizations

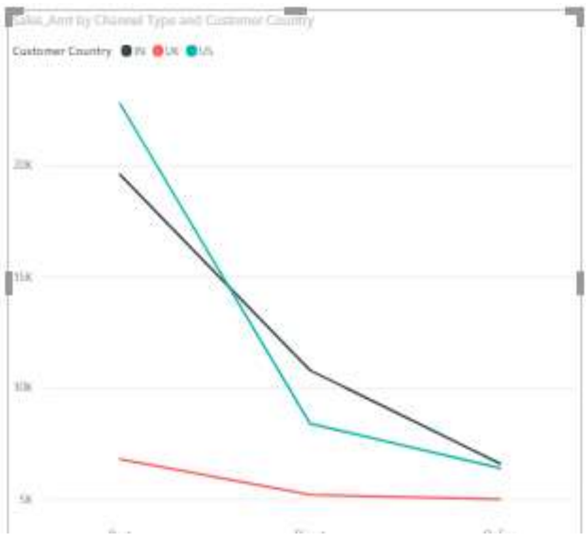
Stacked bar chart

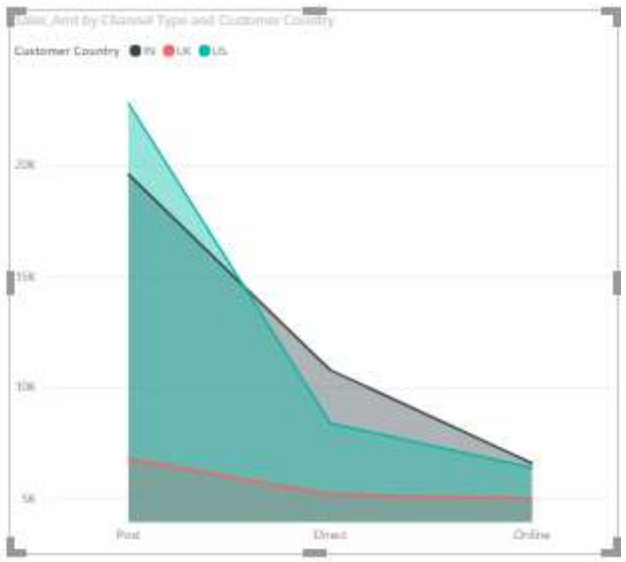
Stacked Column Chart



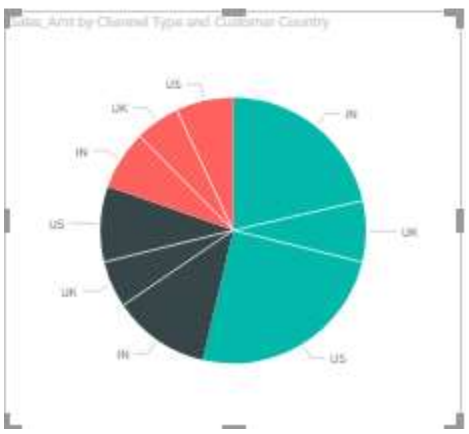


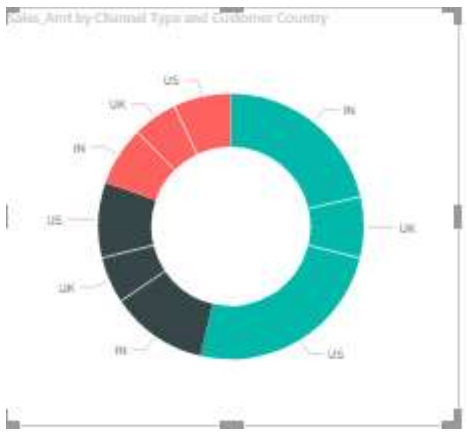
Line Chart **Area Chart**



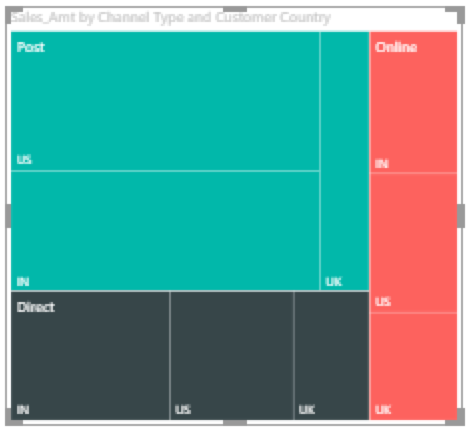


Pie Chart **Donut Chart**



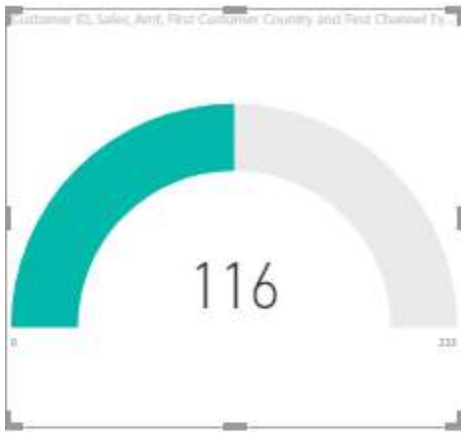


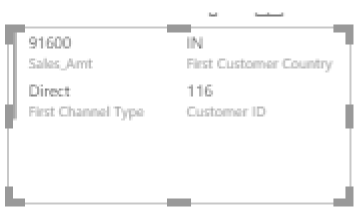
Tree Map **Map** **Card**



Sales_Amt by Customer Country and Channel Type
Channel Type ●Direct ●Online ●Post
NORTH AMERICA
EUROPE
ASIA
Atlantic Ocean
AFRICA
SOUTH AMERICA
Indian Ocean
Bing

Gauge **Multi Card** **KPI**





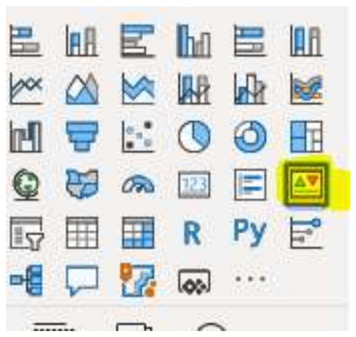
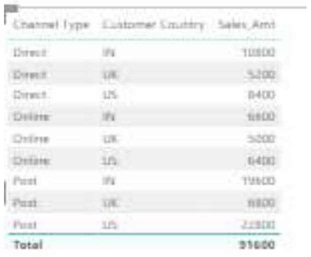
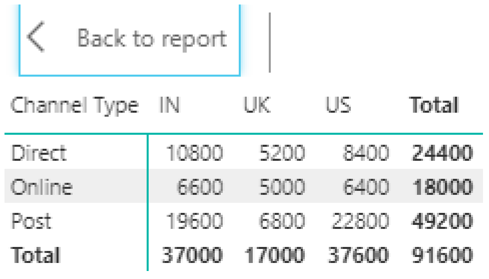


Table **Matrix**



Channel Type	Customer Country	Sales_Amt
Direct	IN	10800
Direct	UK	5200
Direct	US	8400
Online	IN	6600
Online	UK	5200
Online	US	6400
Post	IN	19600
Post	UK	6800
Post	US	22800
Total		**91600**



Channel Type	IN	UK	US	Total
Direct	10800	5200	8400	**24400**
Online	6600	5000	6400	**18000**
Post	19600	6800	22800	**49200**
Total	**37000**	**17000**	**37600**	**91600**

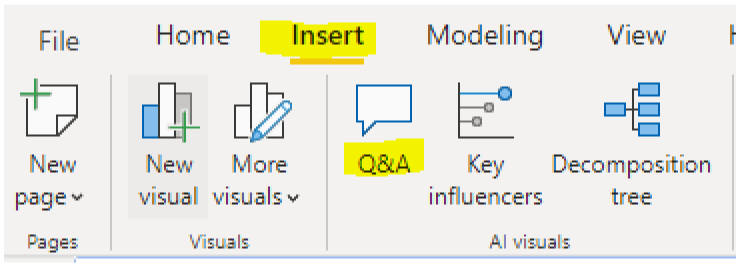
KPI Traffice Light Visulizations

Display Tool Tip Visual

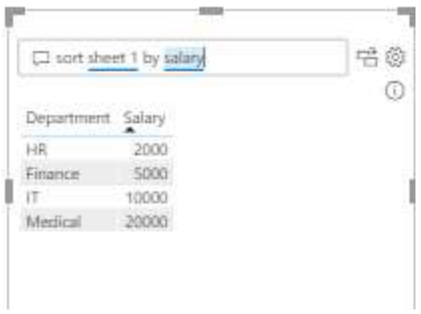
Templates and Themes

Ask a Question Functionality

Which is very interesting functionality which has given by in Power Bi.



If you ask any question of top of any visual it will create new visual(The visual is based on a requested question)



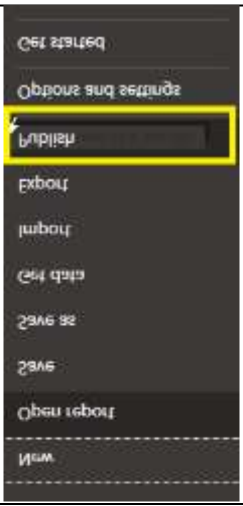
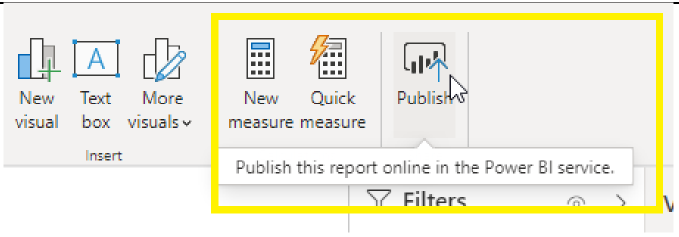
sort sheet 1 by salary

Department	Salary
HR	2000
Finance	5000
IT	10000
Medical	20000

Understanding the Mash Up language behind GUI Options.

Publish a Report to the Web (Power BI cloud Service)

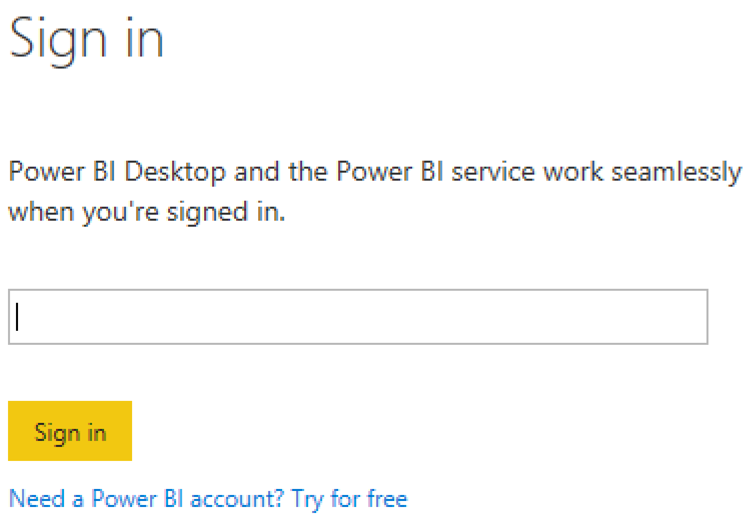
1. There are 2 ways to Publish a report, but both are do same operations.

Menu, Select File>Publish to Web	• Use GUI option
 Get started Options and settings Publish Export Import Get data Save as Save Open report New	 New visual Text box More visuals Insert New measure Quick measure Publish Publish this report online in the Power BI service. Filters

2. Click on Publish
   - After clicked it will open the dialog box to Enter Credentials.

(If you already login with credentials while working on Report means it will not ask again)

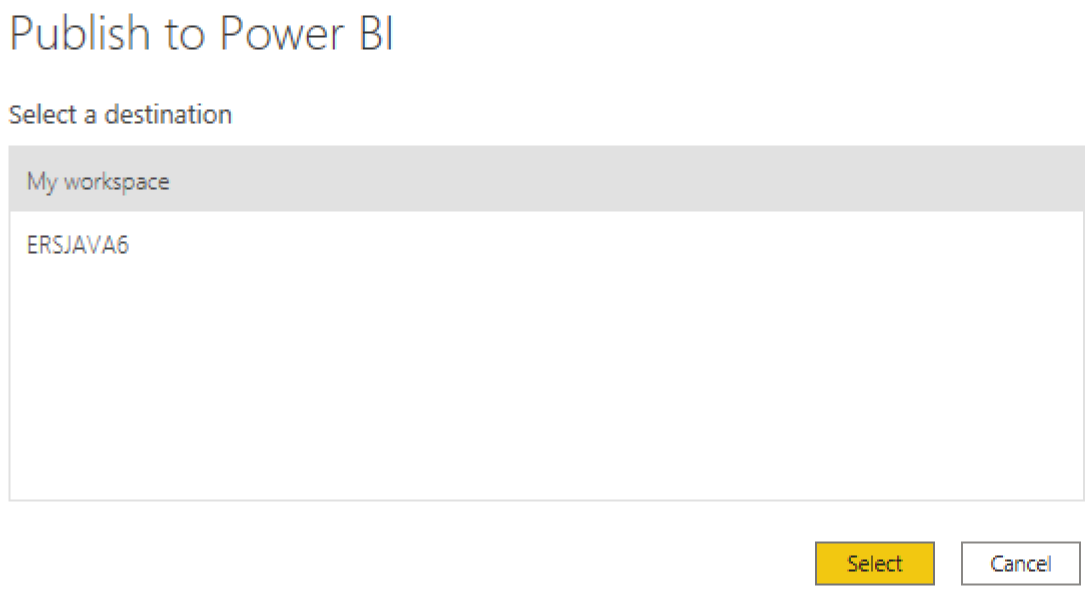
- Give Office or any type of organizational accounts. Like @hcl.com.
  (@gmail, @outllok, @yahoo will not work here)



Sign in

Power BI Desktop and the Power BI service work seamlessly when you're signed in.

And It will ask you to Select a destination. (List of workspaces will be appeared).



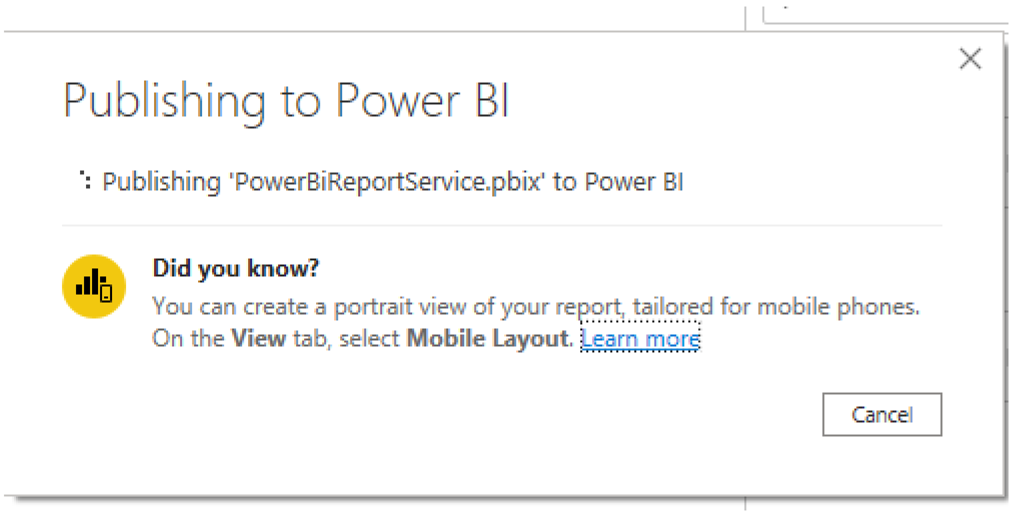
Publish to Power BI

Select a destination

My workspace

ERSJAVA6

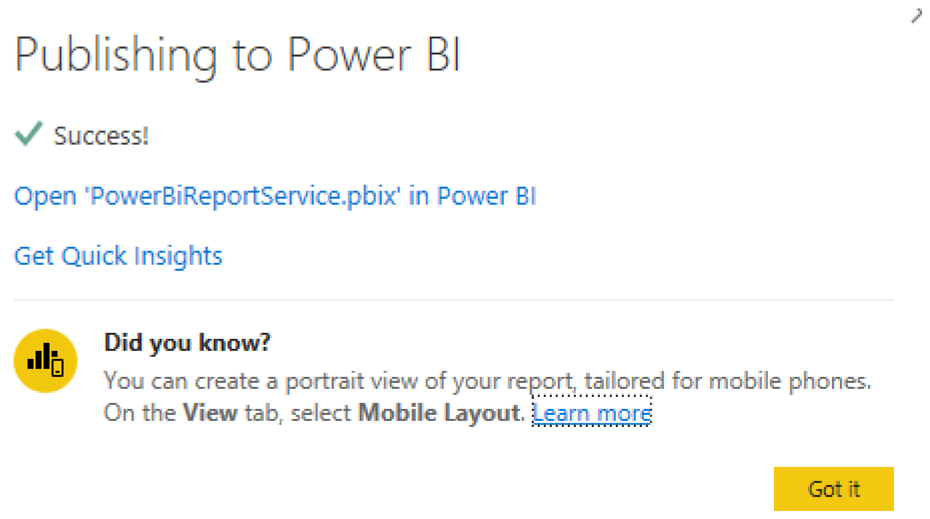
After selected destination, you report will be published.



Publishing to Power BI

Publishing 'PowerBiReportService.pbix' to Power BI

Did you know?
You can create a portrait view of your report, tailored for mobile phones. On the **View** tab, select **Mobile Layout**. Learn more



Publishing to Power BI

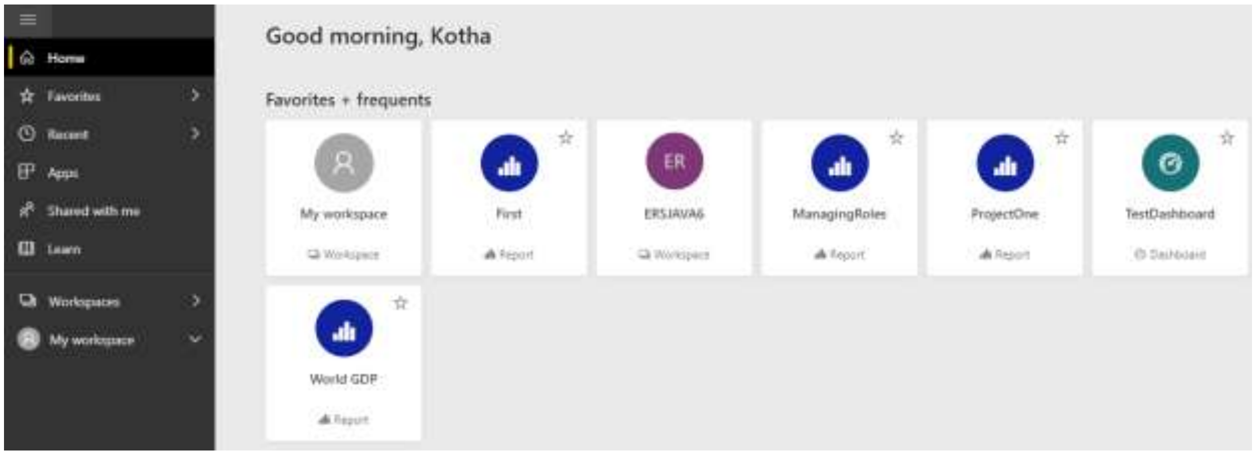
Success!

Open 'PowerBiReportService.pbix' in Power BI

Get Quick Insights

Did you know?
You can create a portrait view of your report, tailored for mobile phones. On the **View** tab, select **Mobile Layout**. Learn more

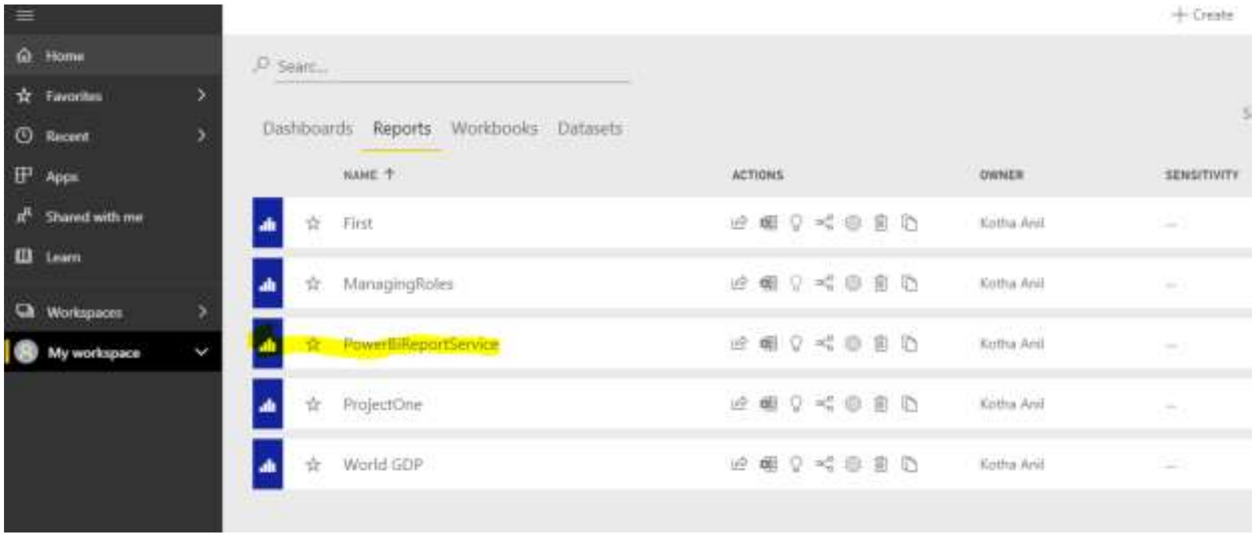
After published you can see your report in **https://app.powerbi.com/home**



Good morning, Kotha

Favorites + frequents

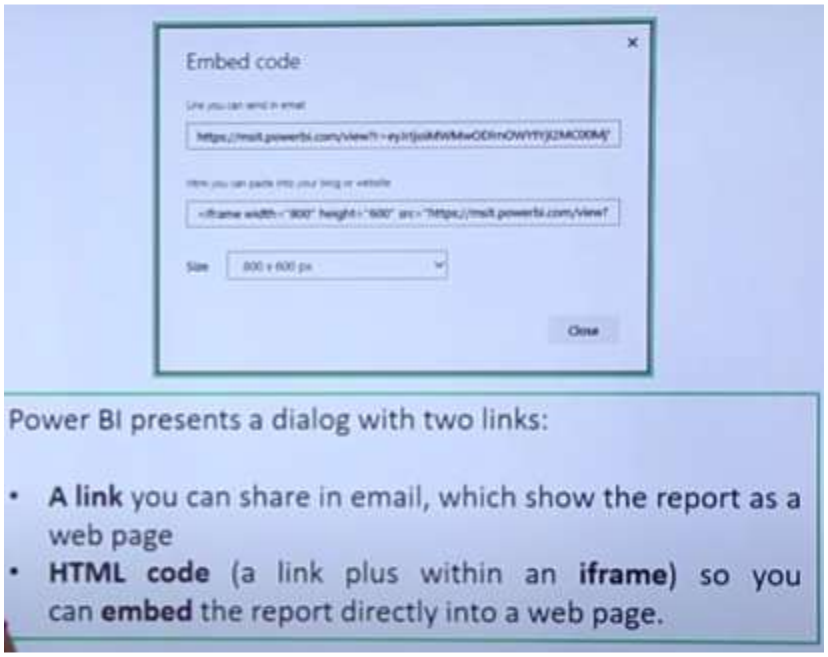
Go to workspace where you published. I published in My Workspace so am clicking on My workspace, after you clicked you can be able to see your report.



NAME	ACTIONS	OWNER	SENSITIVITY
First		Kotha Anil	—
ManagingRoles		Kotha Anil	—
PowerBIReportService		Kotha Anil	—
ProjectOne		Kotha Anil	—
World GDP		Kotha Anil	—

We can also Edit the Report

- A dialog that explains you will get an embed code that lets you share the report on a website or in mail
- When you select create embed code, Power BI presents another dialog, telling you again that you are about to share your data with everyone on the internet. Make sure that is okay!


Embed code
Link you can send in email
https://msit.powerbi.com/view?r=eyJrIjoiMWMwODRnOWYtYjI2MC00Mj
Html you can paste into your blog or website
<iframe width="800" height="600" src="https://msit.powerbi.com/view?
Size
800 x 600 px
Close

Power BI presents a dialog with two links:

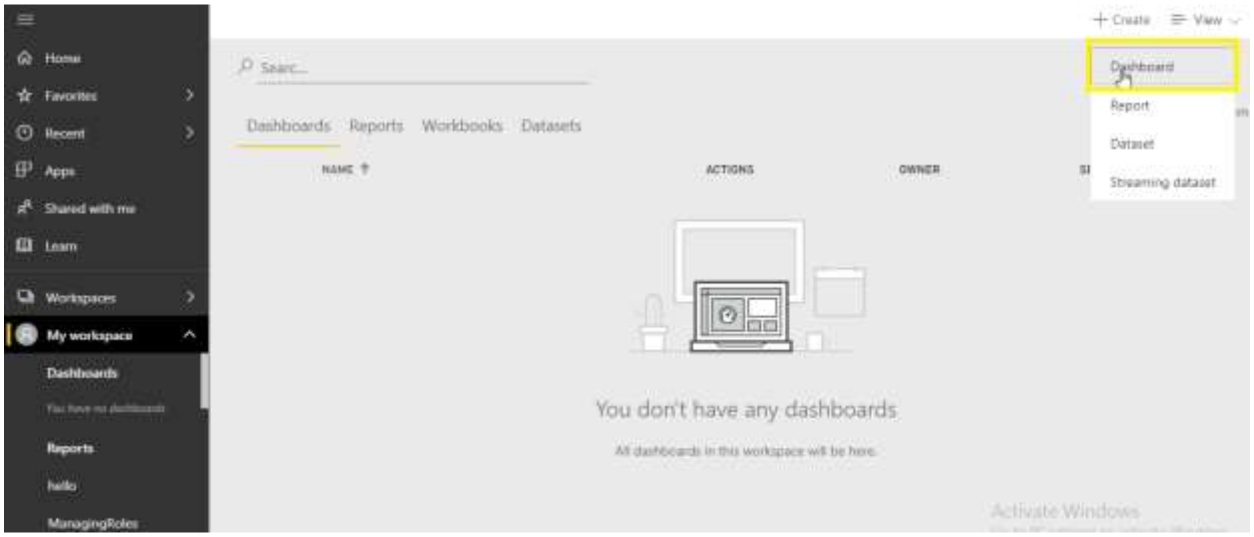
- **A link** you can share in email, which show the report as a web page
- **HTML code** (a link plus within an **iframe**) so you can **embed** the report directly into a web page.

Dashboard

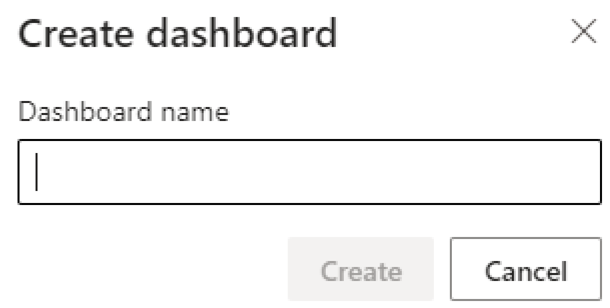
We can a pic multiple visual from different reports to a Dashboard.

How to create a Dashboard

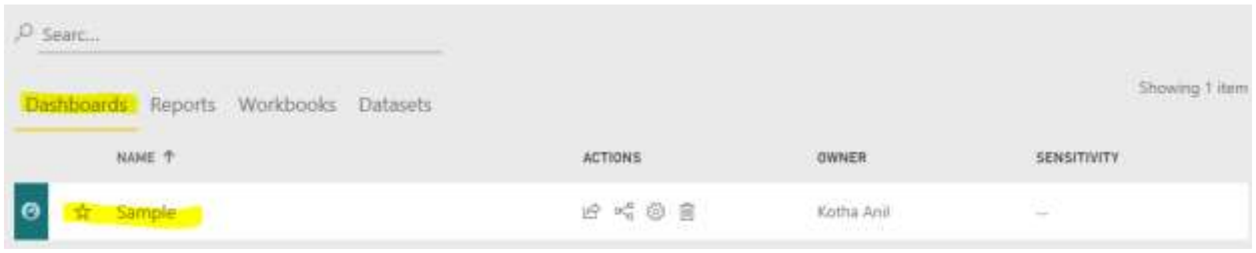
Click on highliated part to create a new dashboard.



After clicked, Give the Name

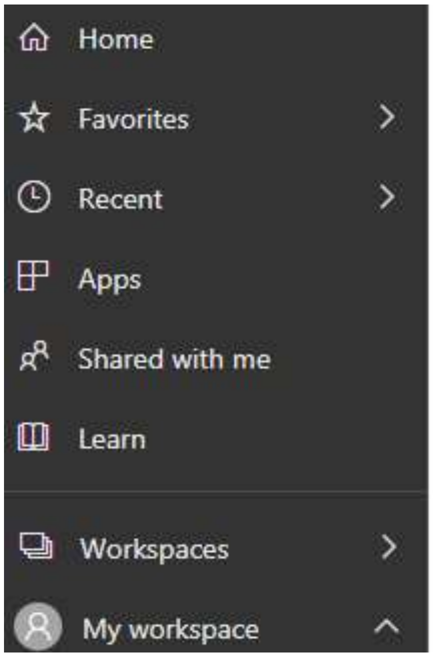


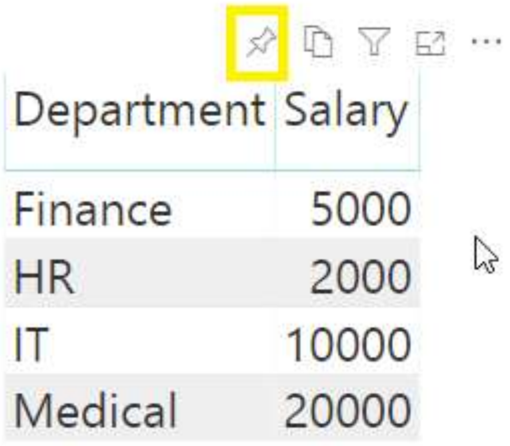
Am giving name as Sample



Now you can add reports to this dashboard.

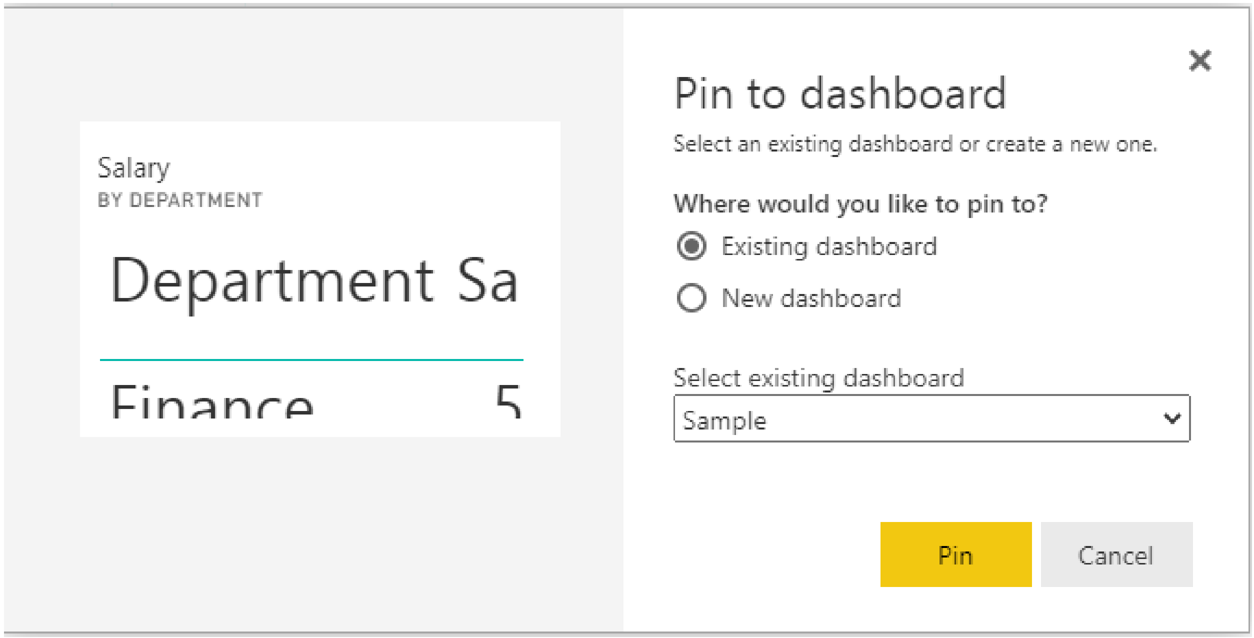
First go to a report and pin a visual which you want to add to a dashboard.





Department	Salary
Finance	5000
HR	2000
IT	10000
Medical	20000

Click on that pin symbol, it will open a window there you can map to existing dashoard or new one. here also you have a chance to create a new dashboard.



Salary
BY DEPARTMENT

Pin to dashboard

Select an existing dashboard or create a new one.

Where would you like to pin to?

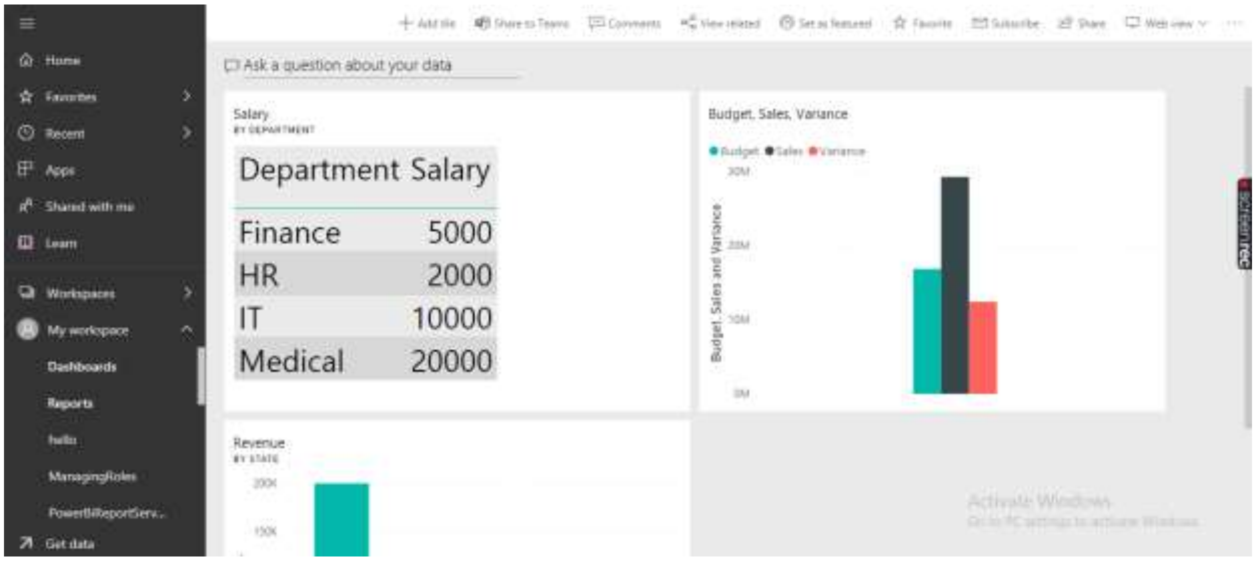
[✓ Existing dashboard] New dashboard

Select existing dashboard: Sample

Similarly, am adding two more visuals to tis dashboard.

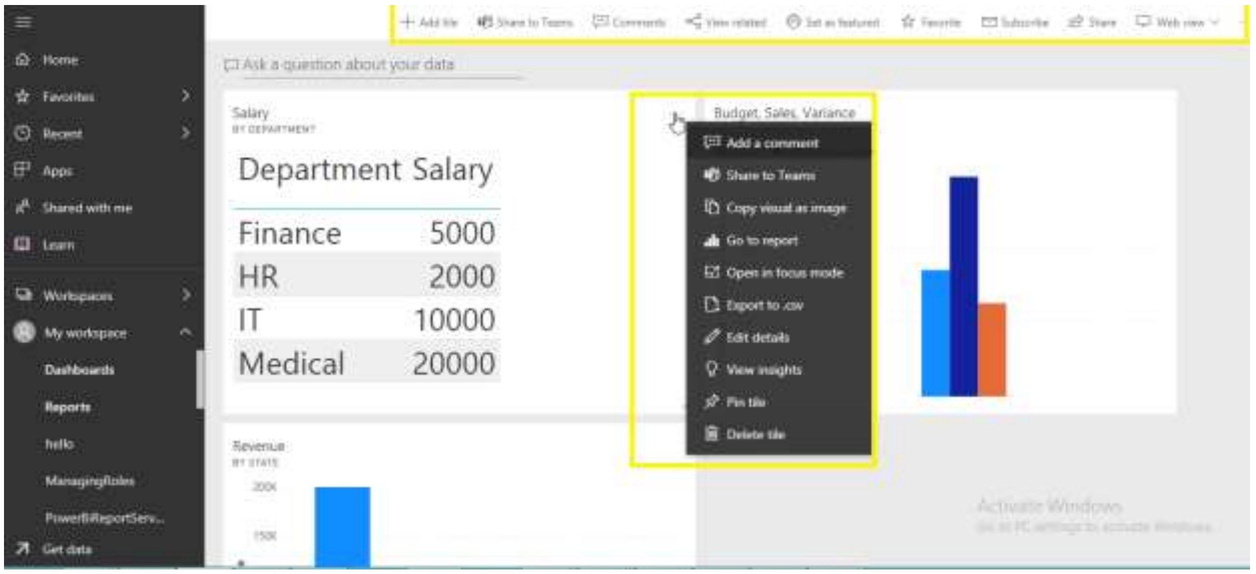
After adding 3 visuals into a SAMPLE dashboard. Now check below screen.

(If you click on any visual it will navigate to that actual report)



Department	Salary
Finance	5000
HR	2000
IT	10000
Medical	20000

We can perform below highliated things in a dashboard for a any visual.



Department	Salary
Finance	5000
HR	2000
IT	10000
Medical	20000

Dataset

Power bi gateway

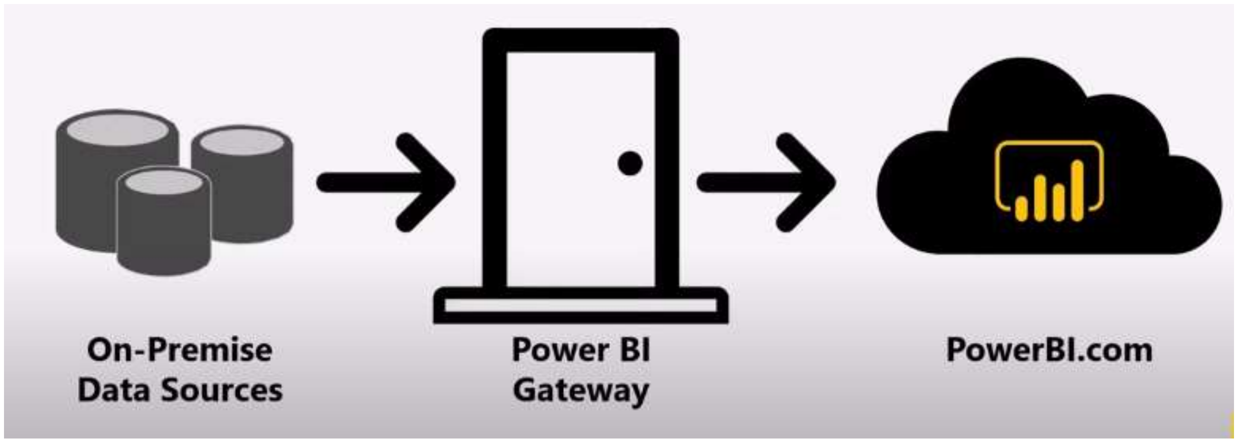
The system that lets you connect your on-premises data sources to the power BI service is called the **Data Gateway**

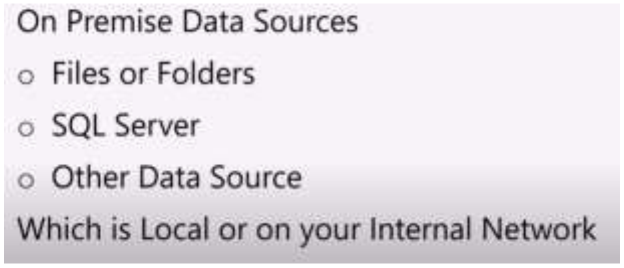
It is small application that runs on your computer and uses a pre-arranged schedule to connect to your data, gather any updates, and push them up to the Power BI service.

Gateways are categorized into 2 types.

- Personal Gateway
- Enterprise Gateway

The Personal Gateway is a version of the Data Gateway that can be used without any administrator Configuration.




On Premise Data Sources
- Files or Folders
- SQL Server
- Other Data Source

Which is Local or on your Internal Network

Note: personal mode gateways will not appear there, only Enterprise mode gateways will be appeared. So, if you use enterprise gateway you will see it under Manage Gateways.

Free license allows us to schedule 8 times per a day.

Paid license which is more.

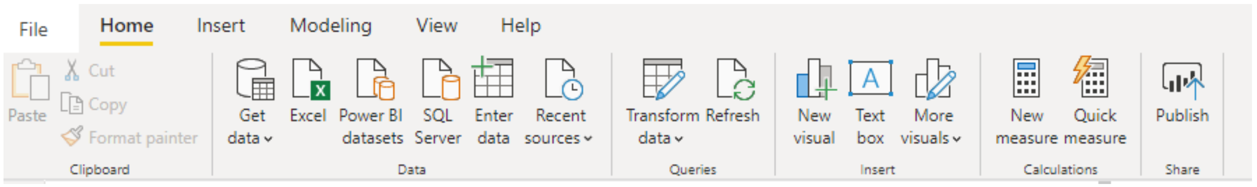
Key Points

- We are using that Sales(5000KB) excel file and developed a report and saved in local machine with name Sales. pbix. If you compare size of .pbix file is lesser than the original excels. pbix file size may be 2000KB. **.pbix size is always lesser than the size of original source**

- If values are like 20.00,25.00,26.00 but you want like 20,25,26 so in this case change data type of column to Text. So automatically values will come like without .00
- If you Want to hide a table in a Report view, you go to modeling part and do right click(…) on the table and click on Hide in Report View.

Few Concepts in Power Bi

1) **Ribbons in Power Bi**
   What is meant by Ribbon-Graphical control element is a graphical control element in the form of a set of toolbars placed on several tabs. ... The usage of the term "ribbon"
   Below example is a Ribbon of power Bi desktop in a Report View.



2) **Advance Editor and Query Dependence in Query Editor.**

3) **Filters in Power Query and their Purpose in Query Editor**
4) **Types of Filters in Power Query (Basic Filtering, Advanced Filtering) in Query Editor**
5) **Auto Filter / Basic Filtering in Query Editor and Filter Multiple Columns**
6) **Group Rows / Group by in built Row Transformations in Query Editor**
7) **Manage Parameters (Manage Parameters, Edit Parameters, New Parameter) in Query Editor**
8) **Suggested Values (Any Value, List of Values, Query), Convert to List**
9) **Query Parameters and Power BI Templates**
10) **.pbix(Power BI File) and .pbit(Power BI Template File)**
11) **Refresh Preview, Refresh All, Cancel Refresh**
12) **Sorting Data**

Questions and Answers

1) **Difference between Dashboard and Report?**
2) **What is DAX Studio?**
3) **Explain about Power Bi service and Power Bi Report server?**
4) **Difference between Fact, Dimension, Attribute in Power BI?**
5) **Difference between Measure and Calculated Column?**
6) **Difference between Copy, Duplicate and Reference?**
7) **What Is meant by Calculated table in Power BI, and how many ways to create calculate table?**
8) **Scheduling in Power BI?**
9) **Do you work on ALLEXCEPT function-DAX?**
10) **How to do filter to show the records by year wise or quarter wise or monthly wise?**

11) **What is Power Bi?**
    Power Bi is a self-serviced cloud-based Business Intelligence software.
12) **Why can't we directly create report on OLTP systems? Why it was not recommended?**
13) **Current version of Power Bi desktop?**
    2.83 64 bit released in July 2020

Project Info:

- Project Name?
- Which sources are using in your project?
- How you are using Gateways?
- SSAS ETL tool is using or not in your project?
- How will you share the Reports to Clients and With In your organization?
- Which cloud service is using in project?
- How to give access to your dashboard for a set of people?
- Scheduling intervals?
- How you are getting tasks-means how they are allocating work?
- How will you send the reports to testing team?