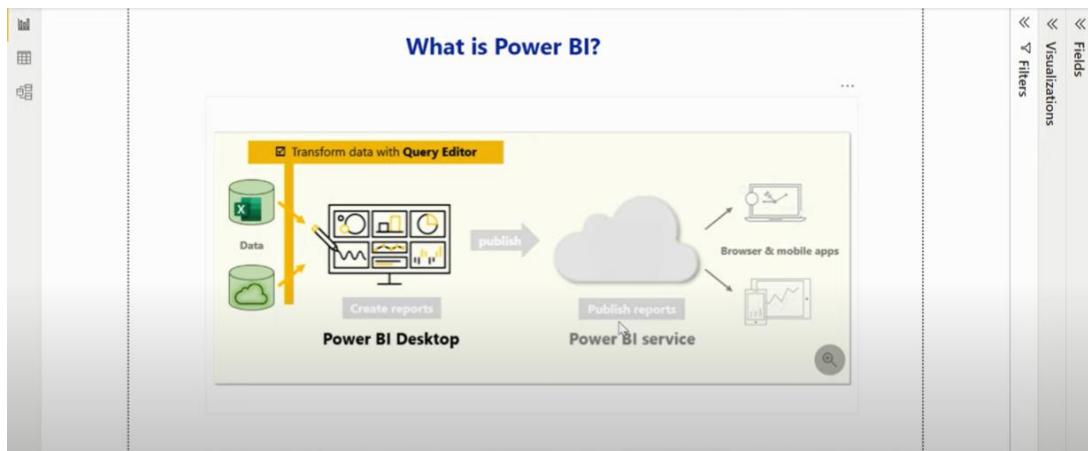
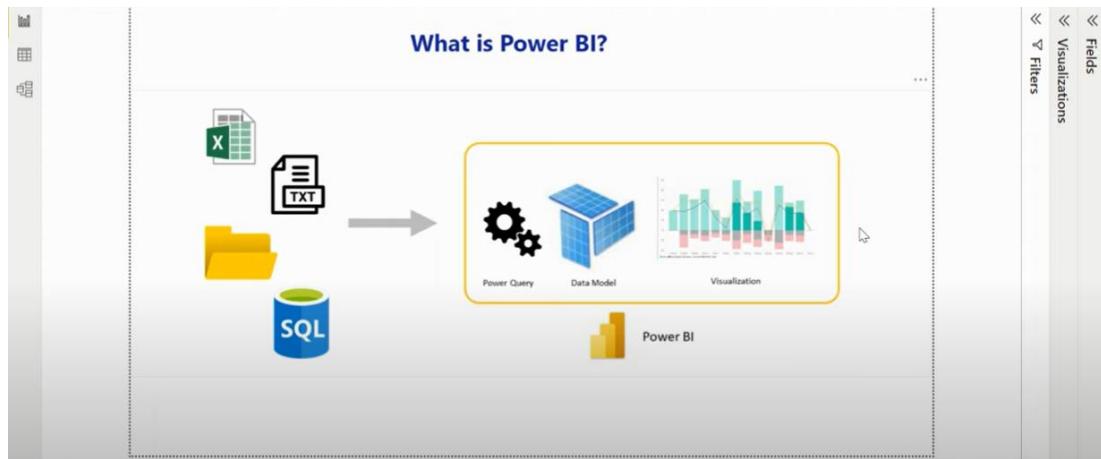
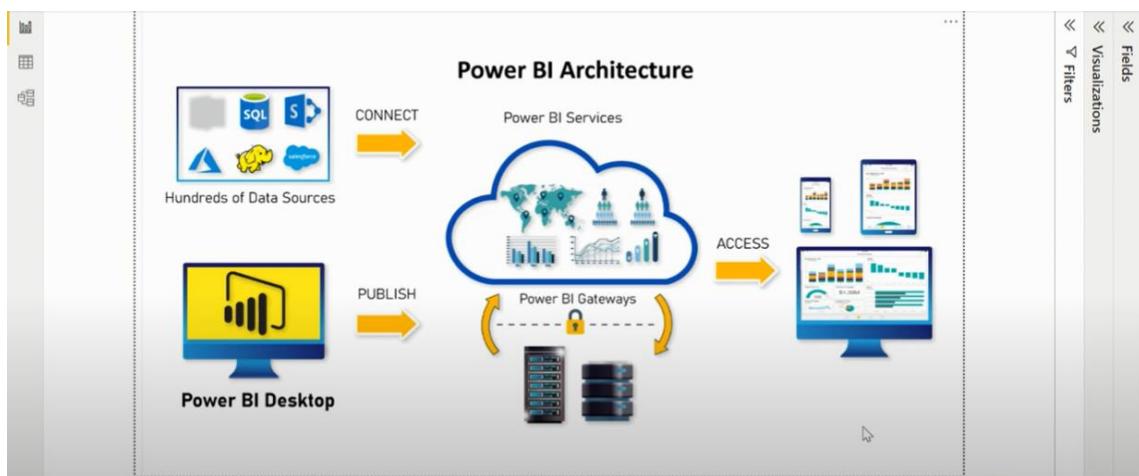


<https://docs.microsoft.com/en-us/dax/>

What is Power BI?



Power BI Architecture



Choose the right data connectivity mode in Power BI

Import option create a copy of the data (pbix file)

Direct Query option creates a connection to the data source

The screenshot shows the Power BI desktop application. On the left is the ribbon with icons for Home, Insert, Transform, Model, Page, and Share. The main area has a title 'Import vs Direct query'. Below it, under 'Import', there's a bulleted list: '.Faster', '.Without premium account, max data is 1GB', '.With premium, max 10GB data', '.With premium, >10 Gb cannot handle', '.Provides all functionalities of PBI', and '.Can connect to multiple data sources in the same report.' Under 'Direct Query:', there's a bulleted list: '.Slow', '.Every dataset in the report cannot exceed 1 million rows', '.Limited features are available', and '.Cannot connect to multiple data sources within same report.' To the right of the main content area are three vertical tabs: 'Fields', 'Visualizations', and 'Filters', each with a double arrow icon above it.

Different parts of Power BI:

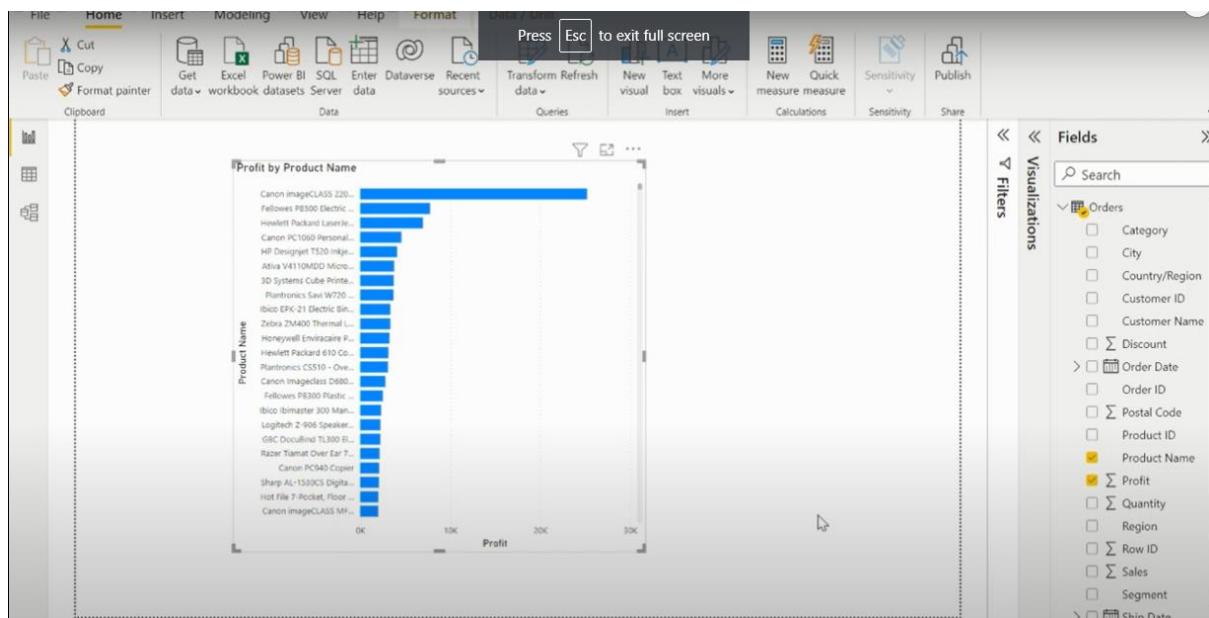
- 1) Power Query - Change Datatype, Renaming the column, deleting the column, new calculated column (M Language)
- 2) Power Pivot - To model the data, define relationship between multiple tables (DAX Language)
- 3) Power View - To build the visualization, where you can see the data visually

Power Query Editor - Inbuild ETL (Extract, Transform, Load) Tool, used for data transformations

Data Transformations in Power BI:

Interpret the missing value

Get Data -> Search & open the file -> Transform data



This screenshot shows the 'Transform Data' step in the Power BI 'Get Data' process. On the left, the 'Navigator' pane shows a tree view of files: 'Sample - Superstore.xlsx' is expanded, showing 'Orders', 'Orders2', 'People', 'People2', 'Returns', and 'Returns2'. 'People' and 'People2' are selected. The main area displays a preview of the 'People' table, which was last updated on 23 March 2022. The table has two columns: 'Column1' and 'Column2'. The data is as follows:

Column1	Column2
Regional Manager	Region
Sadic Pawthorne	West
Chuck Magee	East
Roxanne Rodriguez	Central
Fred Suzuki	South

At the bottom of the preview area are three buttons: 'Load', 'Transform Data', and 'Cancel'.

Transformations Options Available:

Choose columns/rows

remove columns/rows

split column

groupby operation

use first row as headers

The screenshot shows the Power BI Query Editor interface. In the center, there is a preview window displaying a table with two columns: 'Column1' and 'Column2'. The data in the table is as follows:

Column1	Column2
1 Regional Manager	Region
2 Sadie Pawthorne	West
3 Chuck Magee	East
4 Roxanne Rodriguez	Central
5 Fred Suzuki	South

The top ribbon has tabs for Close & Apply, New Source, Recent Sources, Data source settings, Manage Parameters, Refresh, Advanced Editor, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Split Column, Group By, Sort, Data Type: Text, Use First Row as Headers, Transform, Merge Queries, Append Queries, Vision, Combine Files, Azure Machine Learning, and AI Insights.

The left sidebar shows 'Queries [3]' with 'Orders', 'Returns2', and 'People2' listed. The right sidebar shows 'Properties' with 'Name: People2' and 'Applied Steps' which includes 'Source' and 'Navigation'.

Query Settings

Table Names can be changed

Also shows applied steps till now, non relevant steps can be removed, removing steps may undo the work

Merge Queries – same as SQL merge (to join the multiple tables)

Append Queries – same as union in SQL (to append similar tables together)

Power BI Basic Building Blocks

Report Section – shows the visualizations

Data Section – shows the data in form of rows & columns

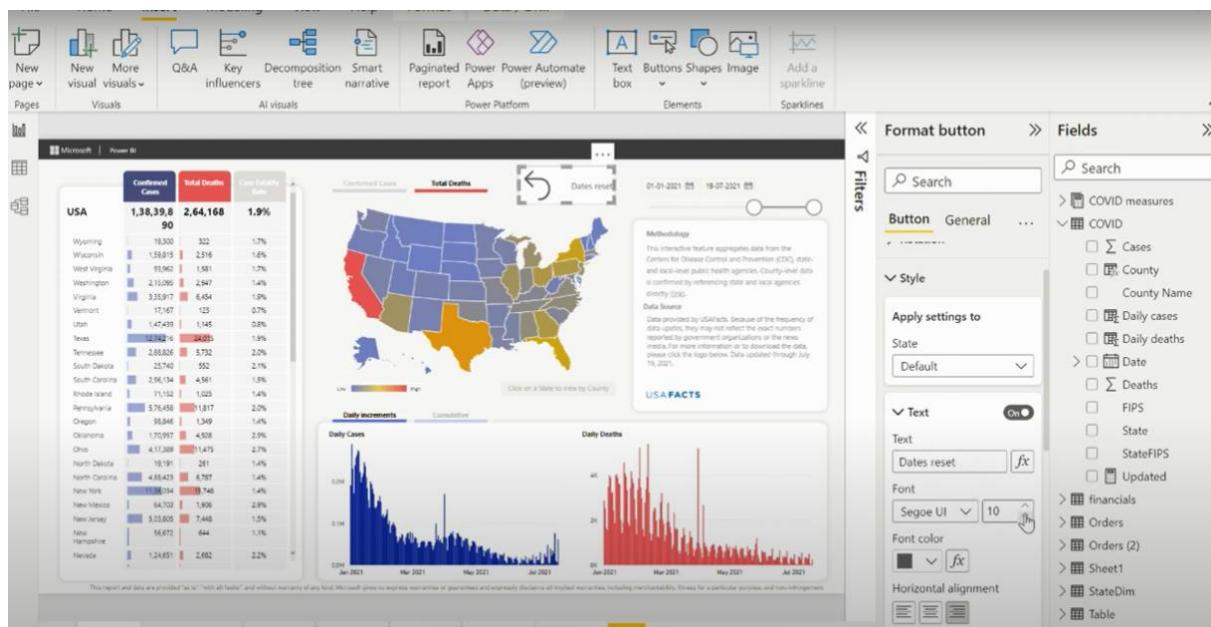
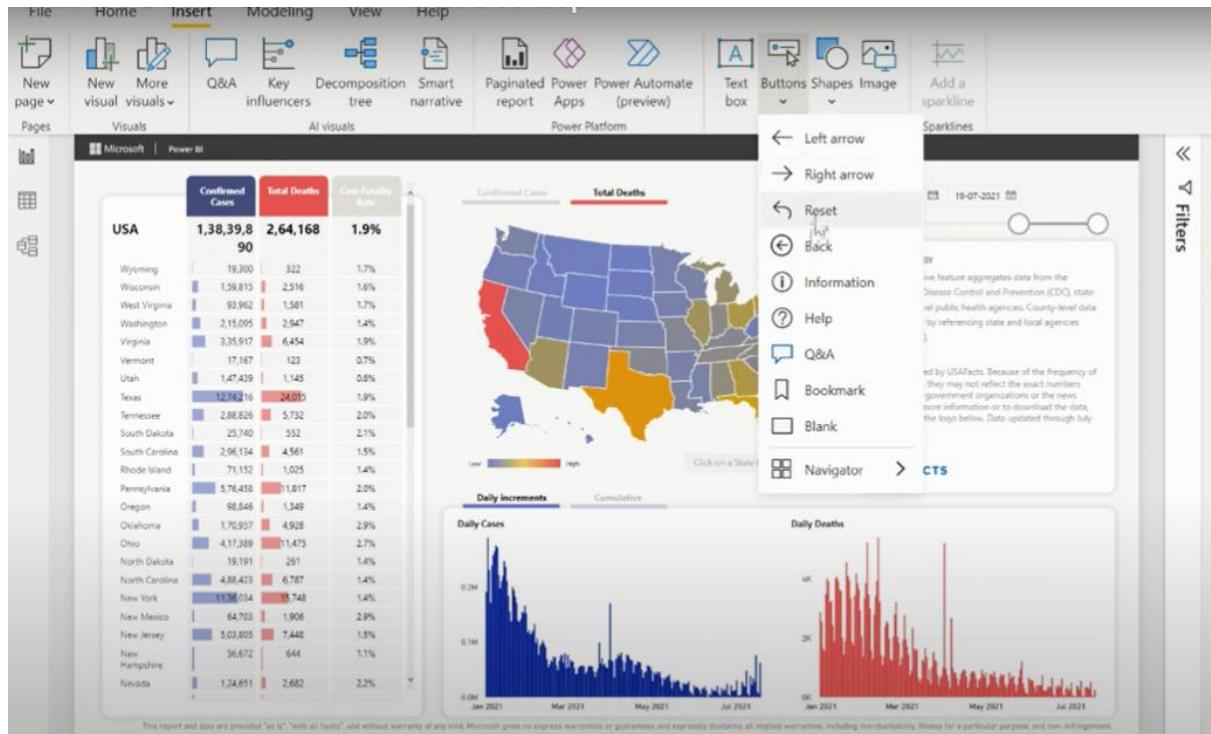
Model Section – allows to define relationships between multiple tables

Bookmarks in Power BI?

Bookmark reset the data in the form it is created

View -> Bookmarks -> Add -> Rename

Insert -> Button -> Reset -> place it on the report view -> format button -> style -> text on -> rename text



To link the button with the bookmark created

Click on created button -> Action -> enable it -> type (bookmark) -> bookmark (bookmark name)

Ctrl + click : Bookmark will be activated

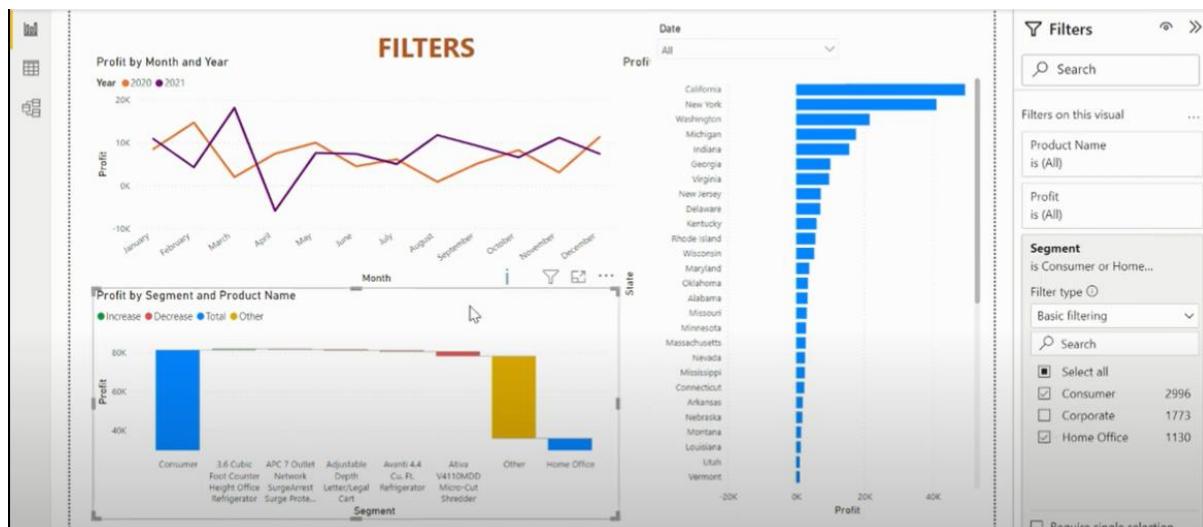


Filters and Slicers :

Both are used to filter the data

Filter :

Filter on Visualization: to be applied on the visualization in report view



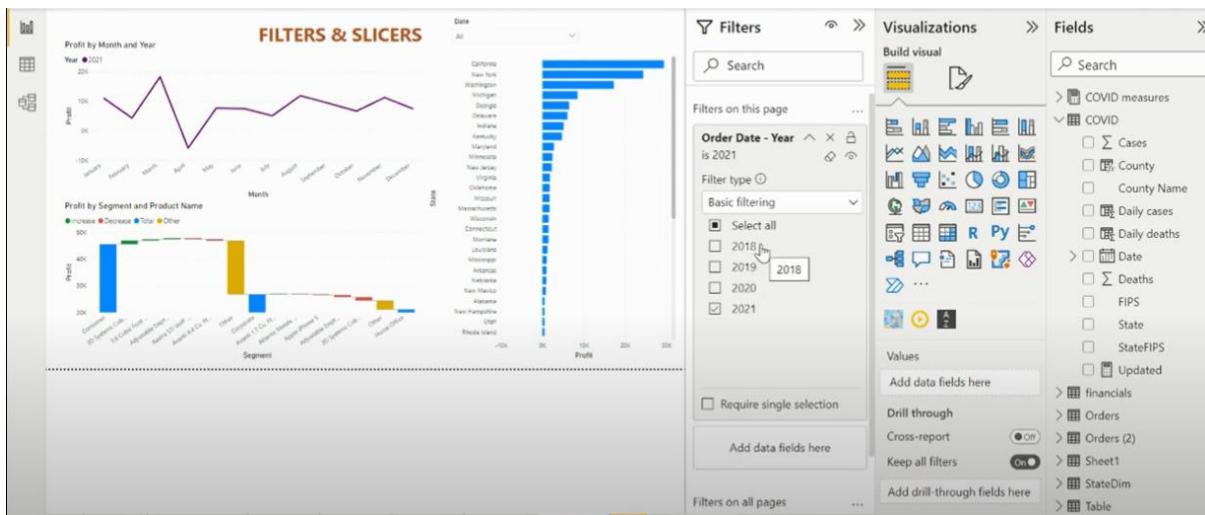
Filter on this page: will filter the current page only

Filter on all pages: all pages will be filtered in one go

Drag & drop the field on required filter type

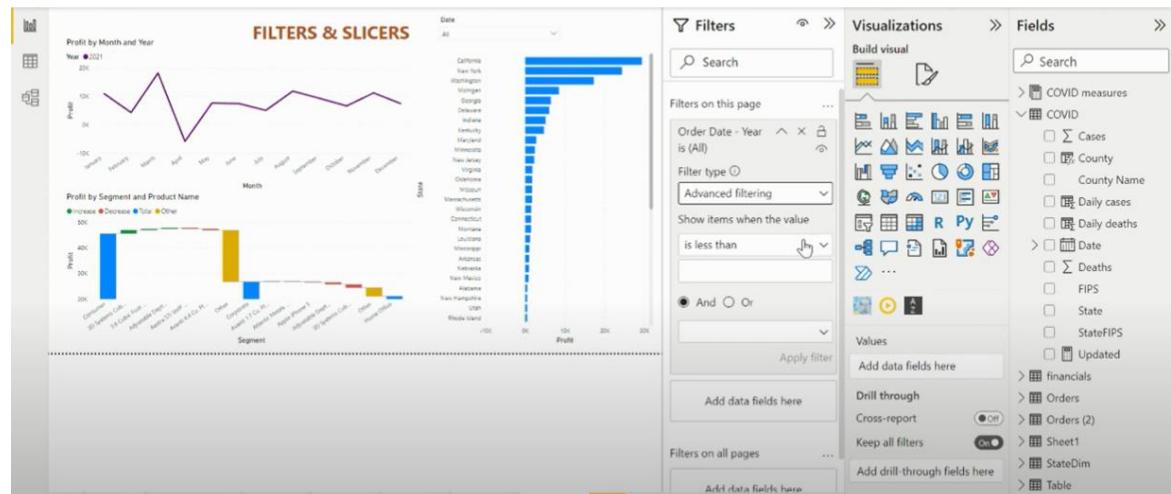


Basic filtering:



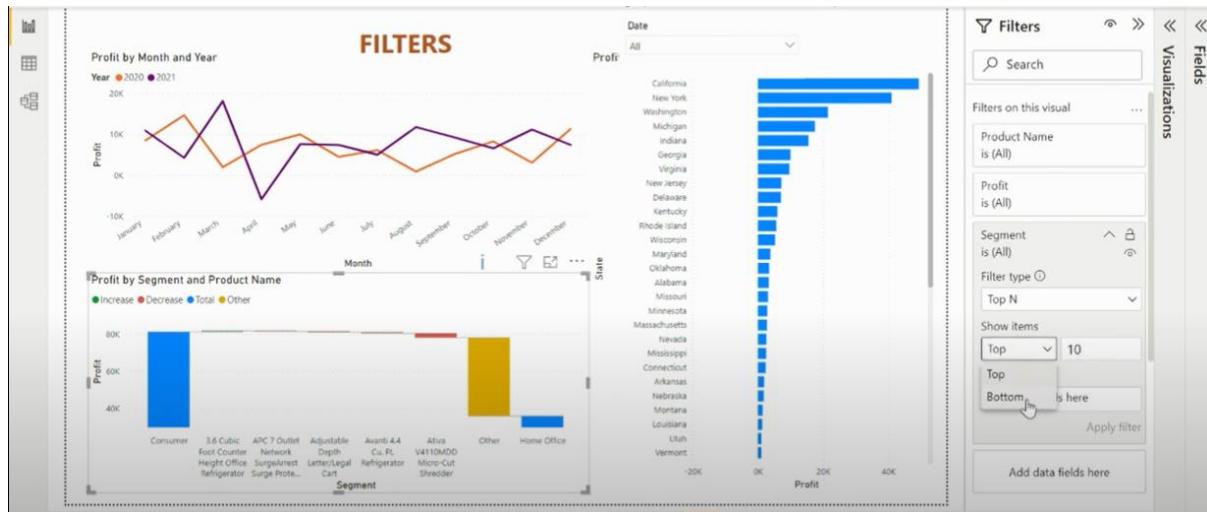
Advanced filtering:

Filtering based on conditions (same as excel)



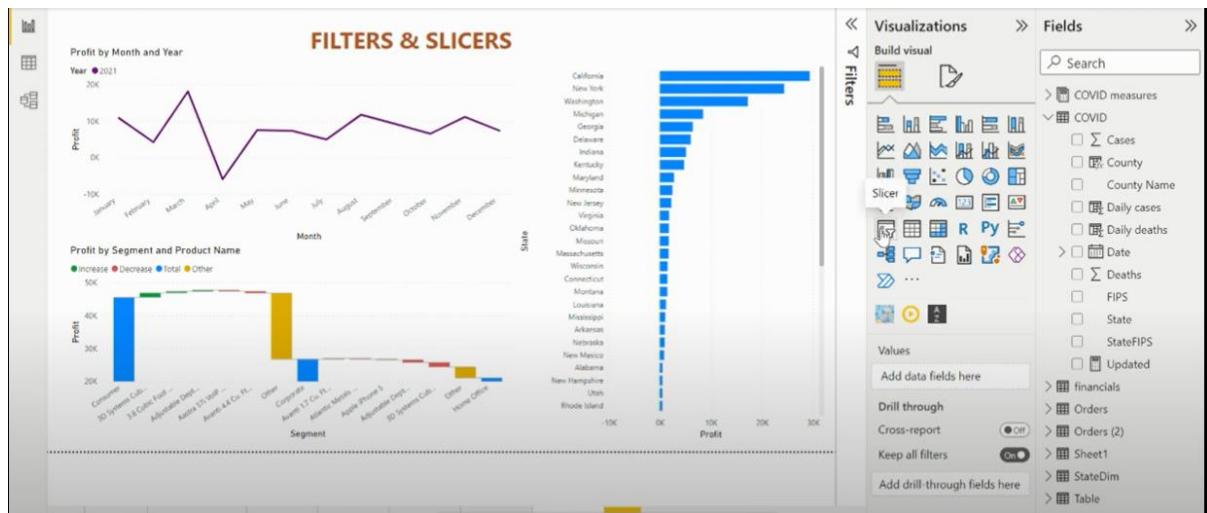
TOP N filtering:

To filter top/bottom contributors



Slicer:

Slicer is a Visual



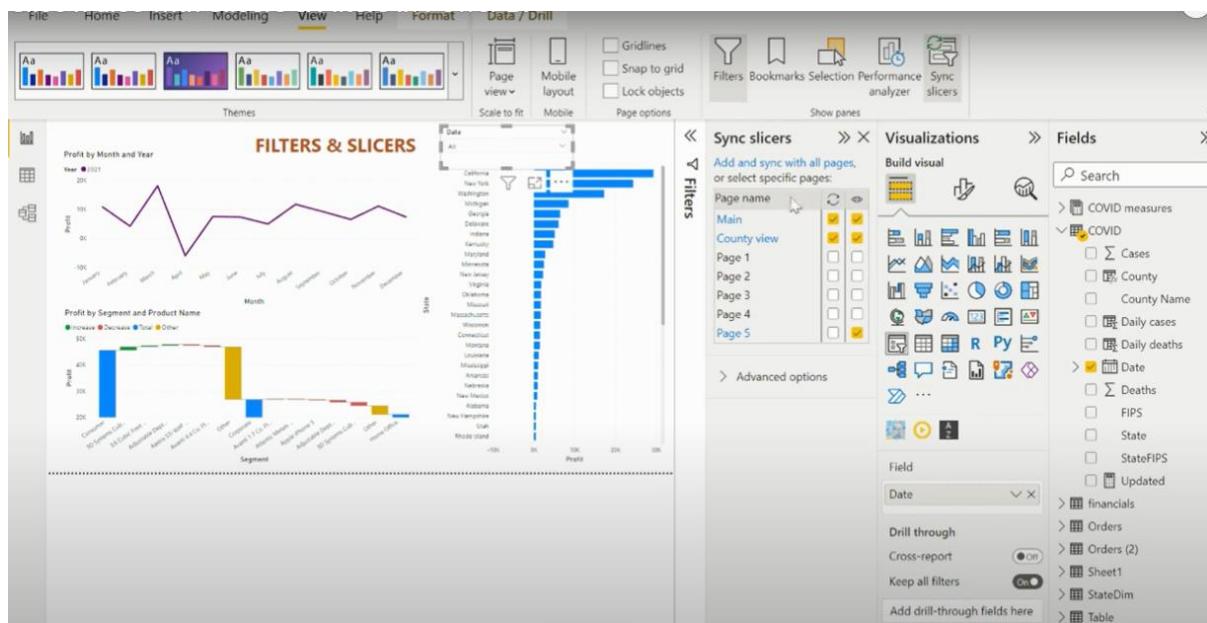
Click on slicer -> place it on report view -> add field -> right top corner (select the type)

Slicer can be applied on all the pages / selected pages

View -> sync slicers

Sync option : data will be filtered on all the selected pages

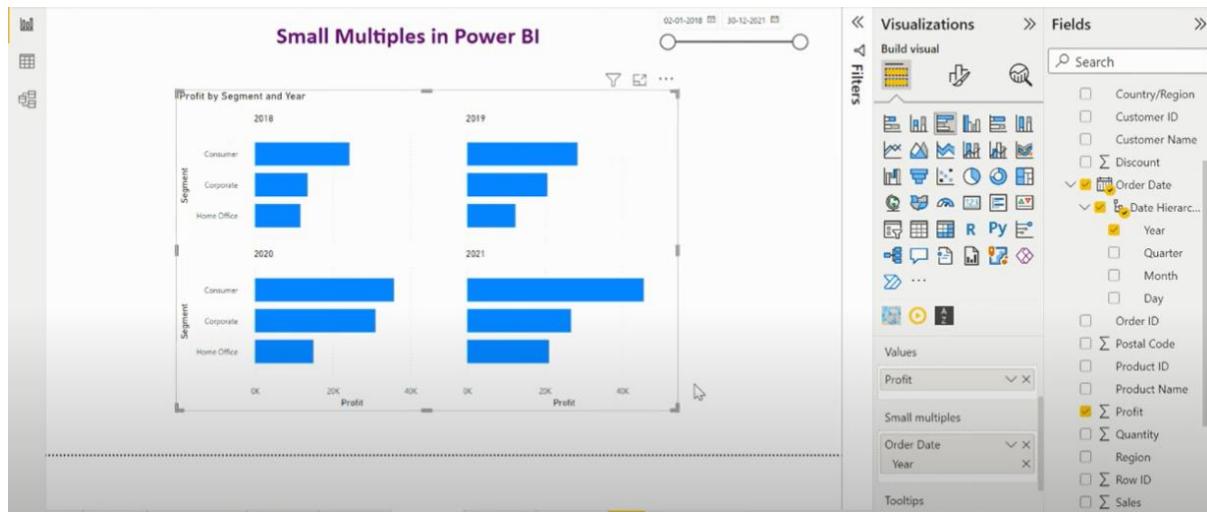
Display option : slicer will be displayed on selected pages



Slicer vs filter: Slicer can be applied on selected pages while filtering can be done on either current or all pages

Small Multiples in Power BI:

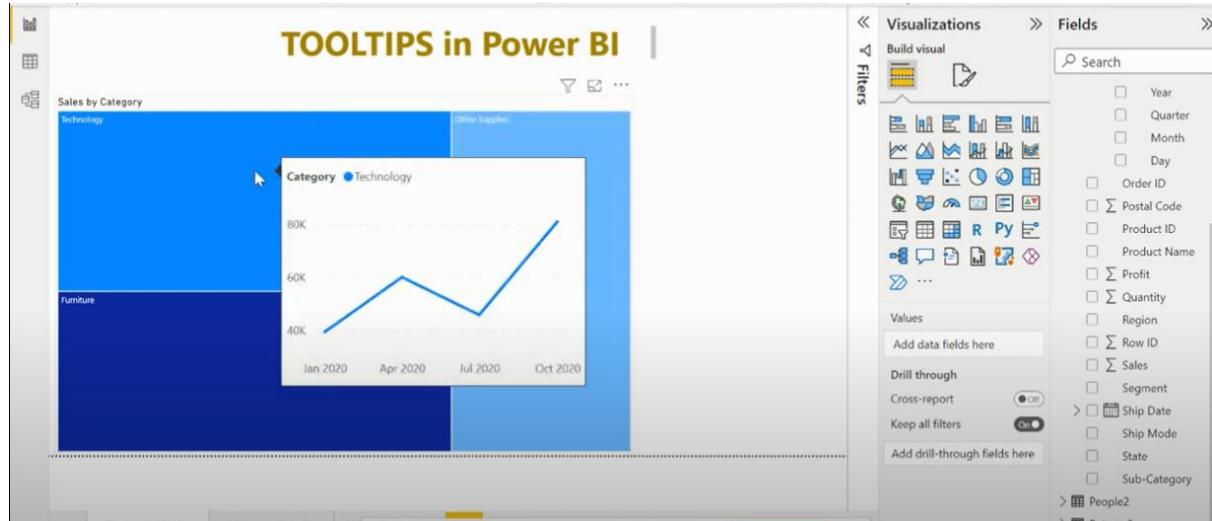
Drag & drop field in small multiples



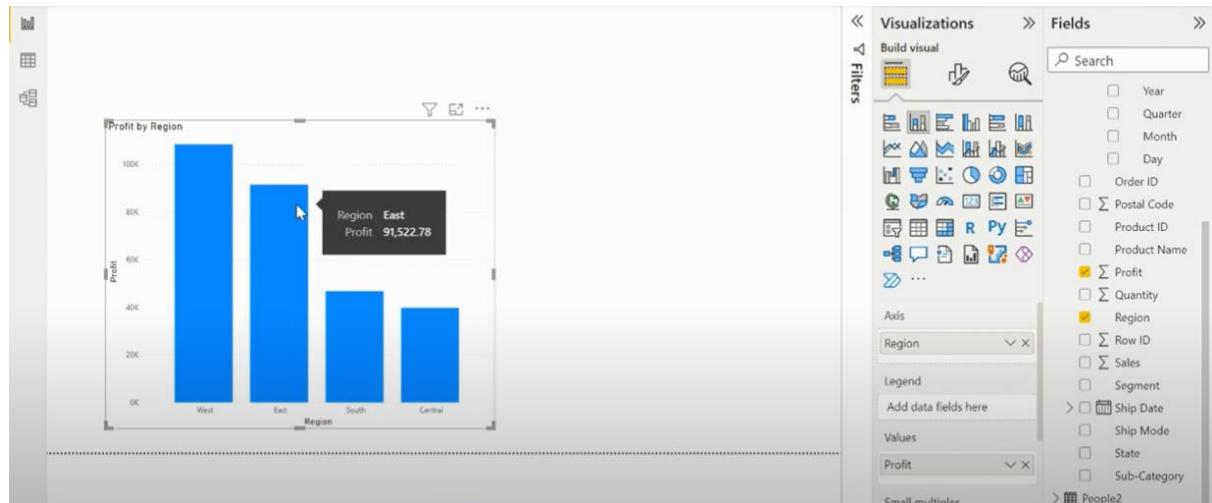
Customized Tooltips in Power BI:

Helps to display the detailed information inside the visual

Customized Tooltip



Basic Tooltip



How to create the customized tooltip?

Create primary visualization -> Add new page -> format page -> page information -> allow use as tooltip on -> name it -> canvas settings -> type (tooltip) -> vertical alignment (middle) -> canvas background -> transparency (0%) -> go to primary visualization page -> format visual -> general -> tooltips -> page (select created tooltip name)

Change of visualization in tooltip will change the same in primary visualization as well.



Visualizations

Format page

Fields

Search

Favorites

Region tooltip

Allow use as tooltip

Allow Q&A

Reset to default

Canvas settings

Canvas background

Wallpaper

Filter pane

Filter cards

Region tooltip

Type: Tooltip

Height: 240 px

Width: 200 px

Vertical alignment: Middle

Reset to default

Visualizations

Format page

Fields

Search

Favorites

Region tooltip

Type: Tooltip

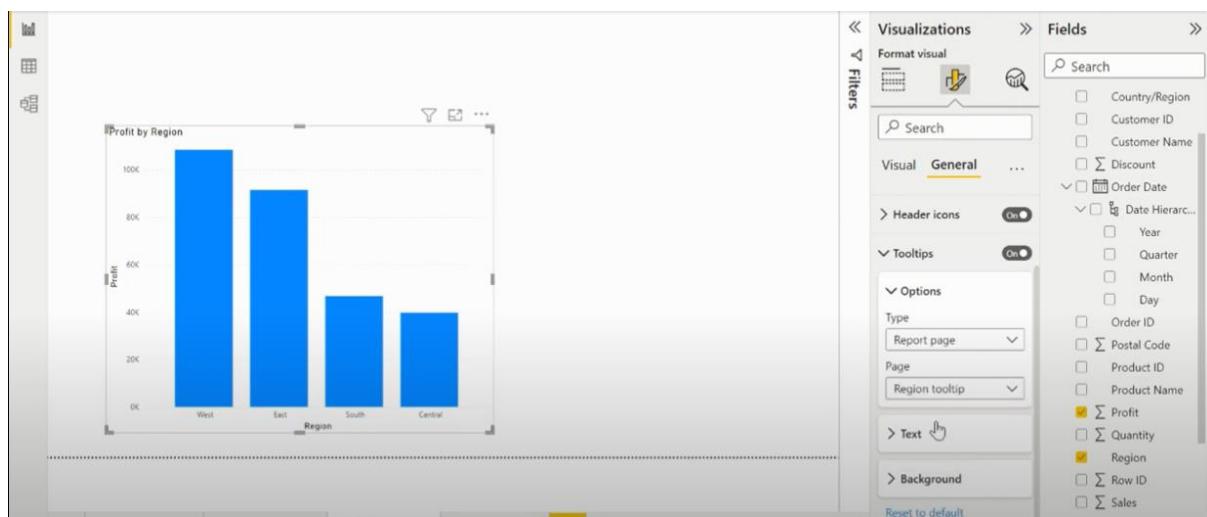
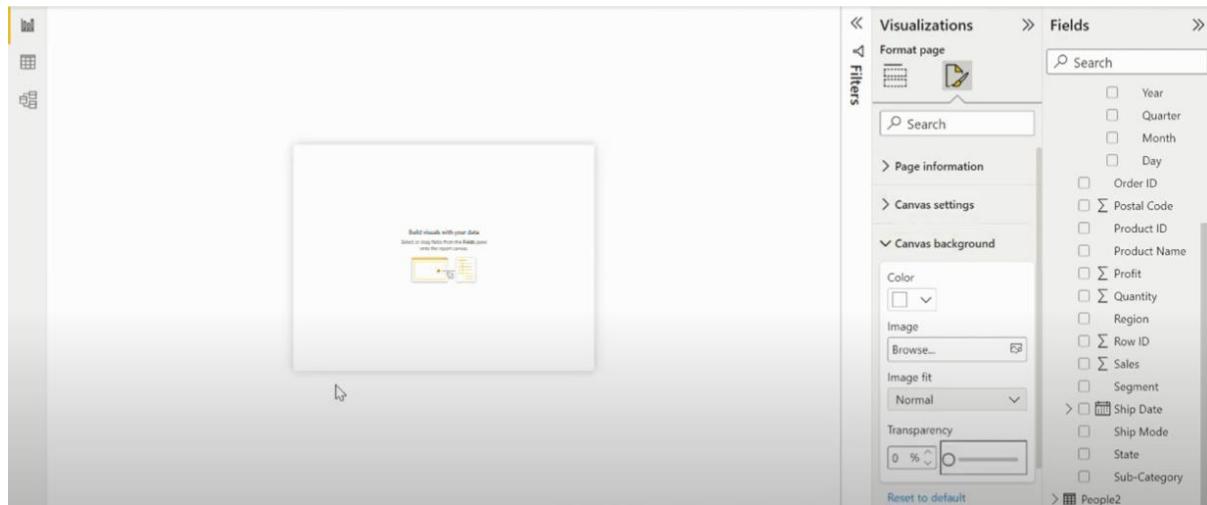
Height: 240 px

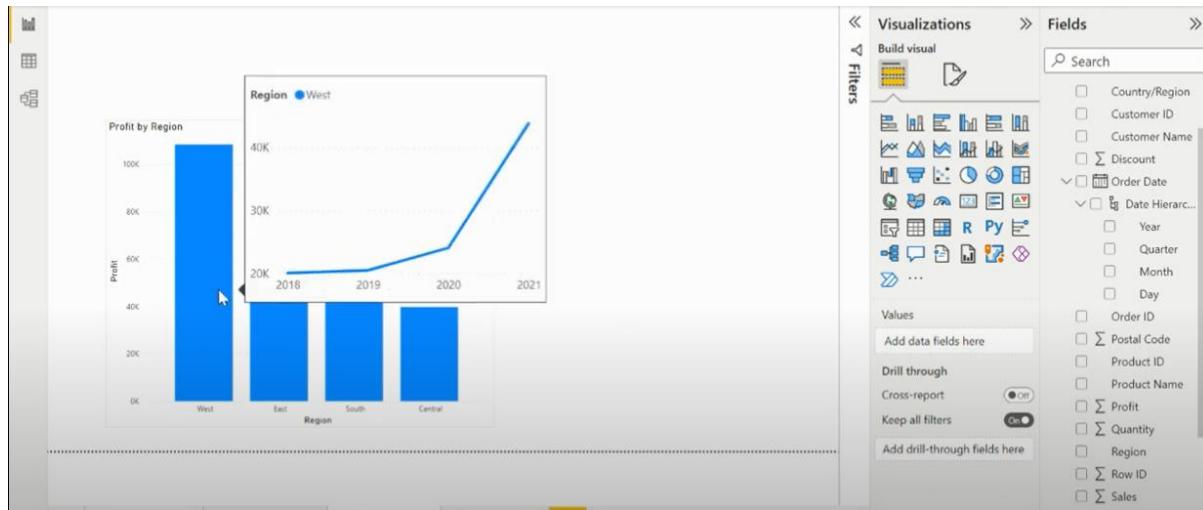
Width: 200 px

Vertical alignment: Middle

Reset to default

Canvas background





Conditional Formatting

Conditional Formatting – Matrix

Select the matrix visualization -> visualization tab -> select dropdown for any of the values ->

Conditional formatting -> select the type of conditional formatting (background colour, font colour, data bars, icons, web url)

Background Colour – cells will be distinguished from each other based on background colour

Font Colour – cells will be distinguished from each other based on font colour

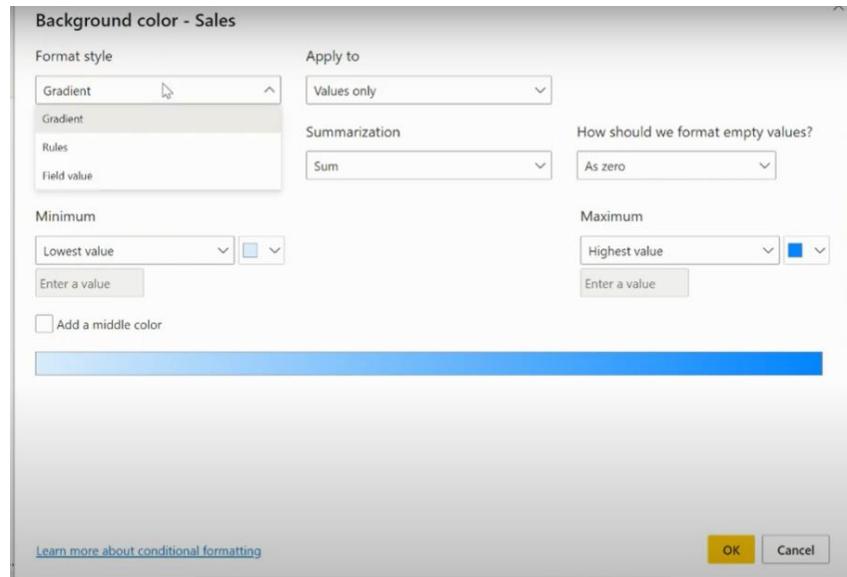
Data Bars –

Icons – cells will be distinguished from each other based on the selected icons from the options,

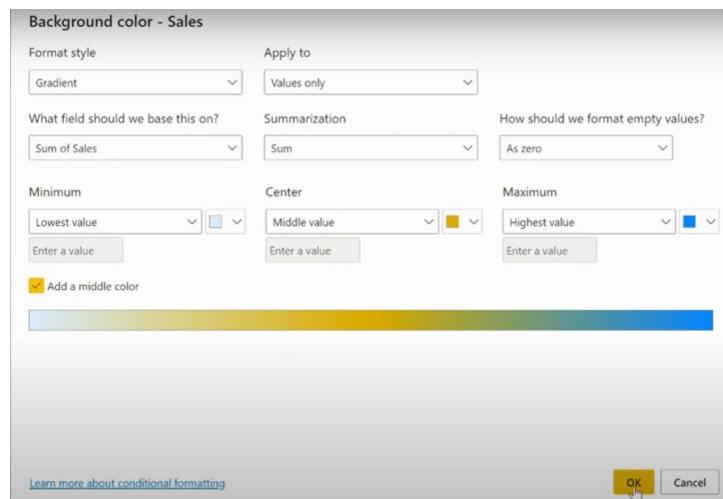
Icons will be displayed alongside the cell values

Format Style – Gradient

Cells will be distinguished by shades of two chosen colours

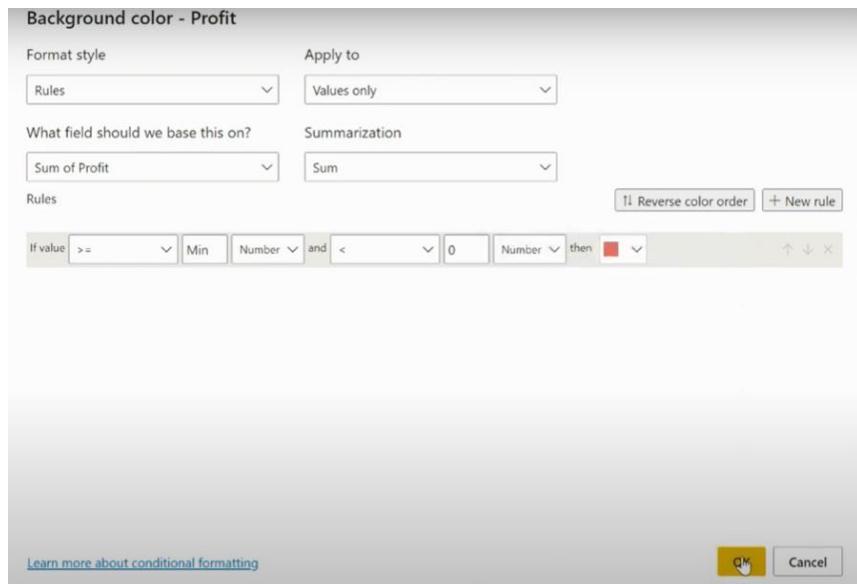
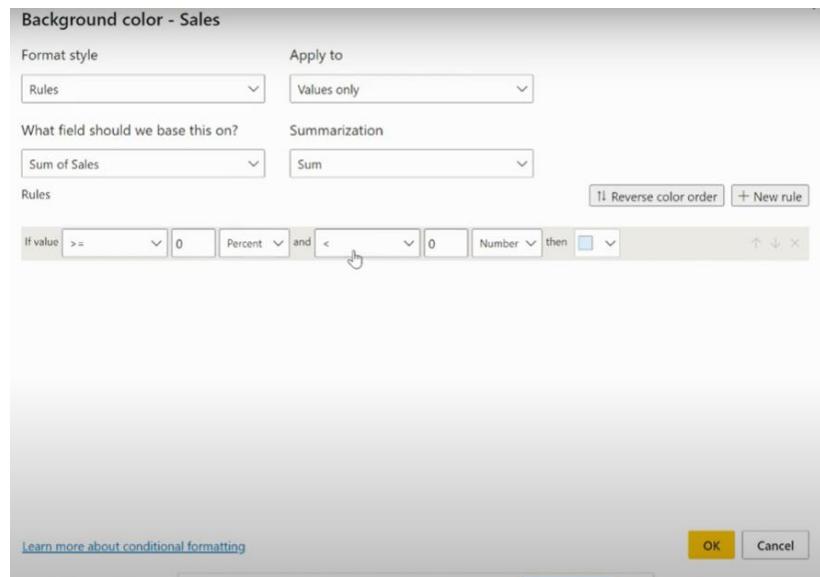


Cells will be distinguished by shades of three chosen colours by selecting the option to add middle colour



The screenshot shows a Power BI report interface. On the left, there is a matrix visual titled 'Conditional Formatting - Matrix'. The columns are labeled 'Year', 'Sales', 'Profit', and 'Average of Discount'. The rows represent years from 2018 to 2021, and each year has four quarters (Qtr 1 to Qtr 4). The cells in the matrix are colored according to a three-color gradient based on their values. The right side of the screen shows the 'Visualizations' pane with various chart and table options, and the 'Values' section where 'Sales', 'Profit', and 'Average of Discount' are listed.

Format Style – Rules



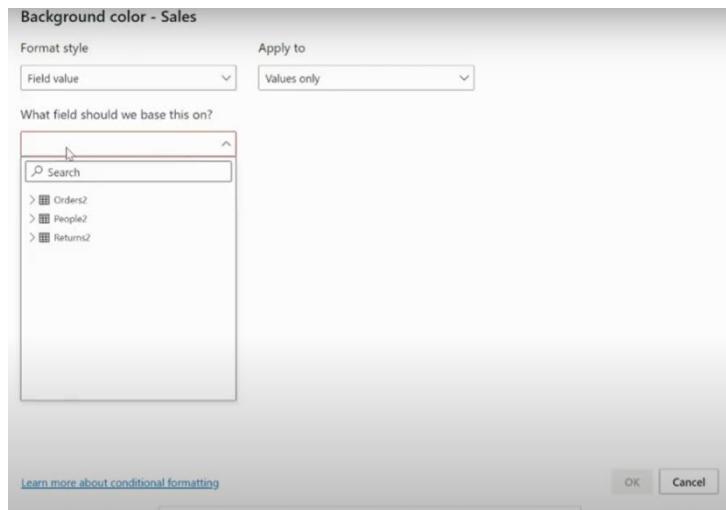
Conditional Formatting - Matrix

The screenshot shows a Power BI matrix visual with conditional formatting applied to the Sales and Profit columns. The matrix displays quarterly sales and profits from 2018 to 2020, broken down by month.

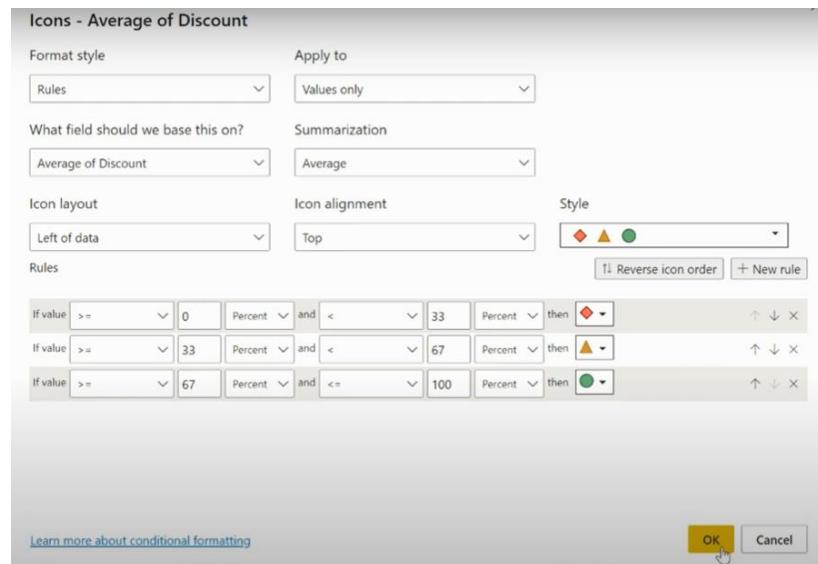
Year	Quarter	Sales	Profit	Average of Disc%
2020	Qtr 4	1,33,764.37	15,499.21	
	Qtr 1	1,23,144.86	23,506.20	
2019	Qtr 4	2,36,098.75	38,139.86	
	Qtr 3	1,43,787.36	15,823.60	
	Qtr 2	1,36,082.30	16,390.34	
	Qtr 1	93,237.18	11,441.37	
2018	Qtr 4	1,79,627.73	21,723.95	
	Qtr 3	1,43,633.21	12,804.72	
	Qtr 2	86,538.76	11,204.07	
	Qtr 1	74,447.80	3,811.23	
2017	Qtr 4	1,82,297.01	23,309.12	
	Qtr 3	1,30,259.58	16,853.62	
	Qtr 2	89,124.19	12,190.92	
	Qtr 1	March	38,726.25	9,732.10
	January	18,174.08	-3,281.01	
	February	11,951.41	2,813.35	
				2,813.85

The matrix visual includes a context menu with options like 'Format style', 'Format painter', 'Copy', 'Paste', 'Delete', 'Format', 'Format as table', 'Format as matrix', and 'Format as card'.

Format Style – Fields



Conditional formatting based on icons



**Additional
Formatting - Matrix**

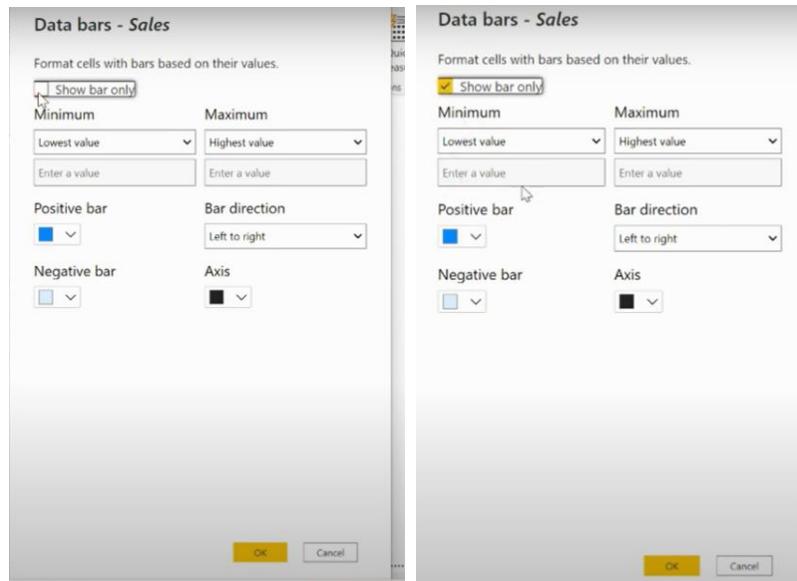
Year	Sales	Profit	Average of Discount
Qtr 2	1,33,764.37	15,499.21	0.17
Qtr 1	1,23,144.86	23,506.20	0.15
2020			
Qtr 4	2,36,098.75	38,139.86	0.15
Qtr 3	1,43,787.36	15,823.60	0.16
Qtr 2	1,36,082.30	16,390.34	0.16
Qtr 1	93,237.18	11,441.37	0.15
2018			
Qtr 4	1,79,627.73	21,723.95	0.17
Qtr 3	1,43,633.21	12,804.72	0.16
Qtr 2	86,538.76	11,204.07	0.15
Qtr 1	74,447.80	3,811.23	0.16
2019			
Qtr 4	1,82,297.01	23,309.12	0.15
Qtr 3	1,30,259.58	16,853.62	0.15
Qtr 2	89,124.19	12,190.92	0.17
Qtr 1	March	38,726.25	9,732.10
	January	18,174.08	3,281.01
	February	11,951.41	2,813.85

The screenshot shows a Power BI report with a matrix visual. The matrix has 'Year' as the column header and months as the row headers. The first three rows show data for 2020, the next three for 2018, and the last three for 2019. Each cell in the matrix contains three numerical values: Sales, Profit, and Average of Discount. The 'Average of Discount' column is formatted with conditional icons: red diamonds for values below 33%, yellow triangles for values between 33% and 67%, and green circles for values above 67%. The report also includes a 'Visualizations' pane on the right showing other available charts and a 'Filters' pane.

Conditional Formatting – Table

In table-based data, conditional formatting can be applied on categorical data as well whereas in matrix-based data conditional formatting can only be applied on measures.

Conditional formatting based on data bars



Conditional Formatting - Table

Order Date	Profit	Sales	Quantity	Discount
22 December 2021	957.12	1,447.04	100	4.40
24 December 2021	1,560.34	1,113.5	54	1.70
29 December 2021	644.43	915.53	41	0.40
25 December 2021	359.20	698.93	87	5.60
23 December 2021	284.10	926.78	63	2.60
28 December 2021	253.12	1,657.35	64	4.12
26 December 2021	61.12	814.59	12	0.90
30 December 2021	101.54	713.79	23	1.00
27 December 2021	-31.97	177.64	6	0.60
Total	4,196.90	24,579.68	450	21.32

The screenshot shows a Power BI visualization titled 'Conditional Formatting - Table'. The table contains data for various dates, with columns for Order Date, Profit, Sales, Quantity, and Discount. The Profit column values range from -31.97 to 1,560.34. The Sales column values range from 177.64 to 1,657.35. The table is styled with data bars where the length of each bar corresponds to the value in the Profit column. The Profit column is highlighted with a yellow background, and the Sales column is highlighted with a light blue background. The rest of the table has a white background.

Conditional Formatting - Table

Order Date	Profit	Sales	Quantity	Discount
22 December 2021	957.12	1,447.04	100	4.40
24 December 2021	1,560.34	1,113.5	54	1.70
29 December 2021	644.43	915.53	41	0.40
25 December 2021	359.20	698.93	87	5.60
23 December 2021	284.10	926.78	63	2.60
28 December 2021	253.12	1,657.35	64	4.12
26 December 2021	61.12	814.59	12	0.90
30 December 2021	101.54	713.79	23	1.00
27 December 2021	-31.97	177.64	6	0.60
Total	4,196.90	24,579.68	450	21.32

This screenshot shows the same 'Conditional Formatting - Table' visualization as the previous one, but with a different configuration. The Profit column is now highlighted with a blue background, and the Sales column is highlighted with a light blue background. The rest of the table has a white background. The Profit column values range from -31.97 to 1,560.34. The Sales column values range from 177.64 to 1,657.35.

Different Types of Conditional Formatting

Profit – Based on Background Colour

Sales– Based on Data bars

Quantity – Based on icons

Avg. Discount – Based on font colour

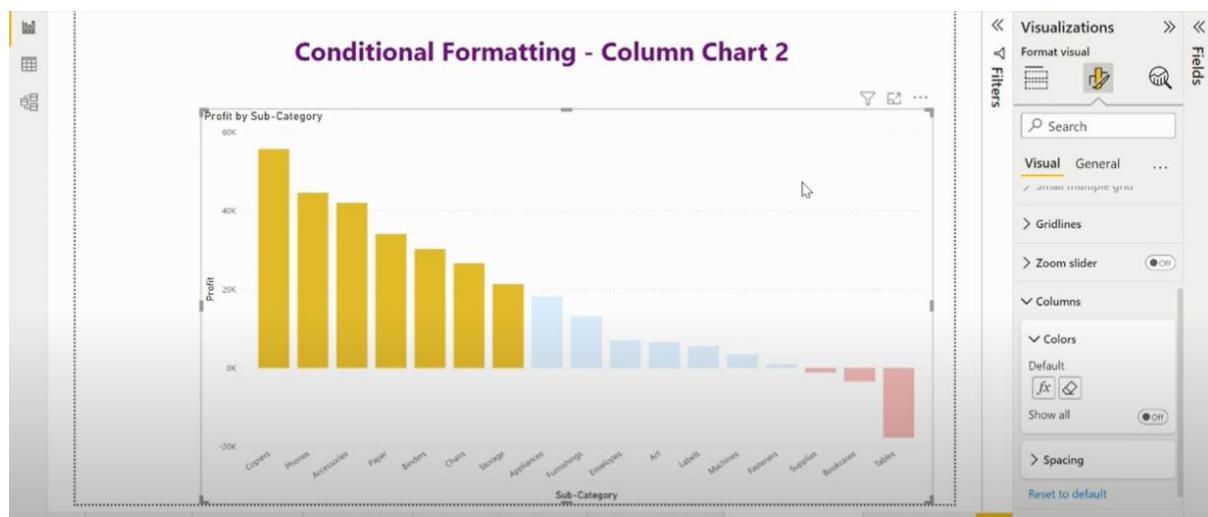
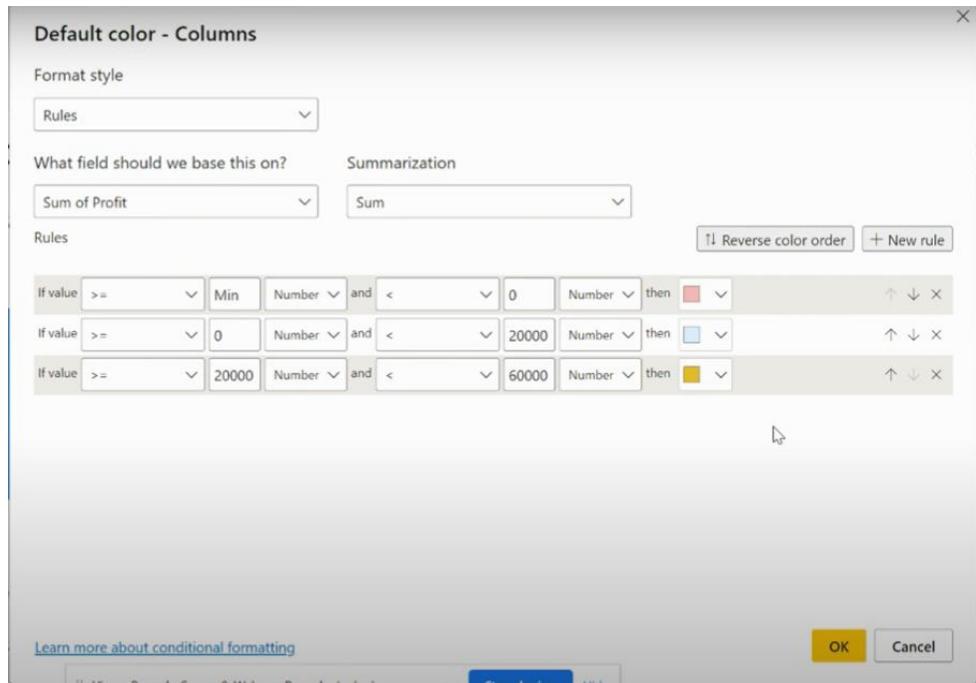
The screenshot shows a Power BI interface with a table visualization titled "Conditional Formatting - Table". The table has columns: Order Date, Profit, Sales, Quantity, and Average of Discount. The rows show data from December 2021, with the last row being "Total". The background color of the table cells changes based on the value in the "Sales" column, creating a heatmap effect. The "Visualizations" pane on the right shows other available visualizations like charts and maps. The "Fields" pane lists the columns used in the table.

Order Date	Profit	Sales	Quantity	Average of Discount
22 December 2021	957.12	7,442.00	100	0.16
24 December 2021	568.24	6,233.15	54	0.11
29 December 2021	644.43	9,155.53	41	0.03
25 December 2021	359.20	6,989.93	87	0.24
23 December 2021	284.10	1,926.78	63	0.17
28 December 2021	253.12	1,657.35	64	0.22
26 December 2021	61.12	814.59	12	0.23
30 December 2021	101.54	713.79	23	0.14
27 December 2021	-31.97	177.64	6	0.30
Total	4,196.90	24,579.68	450	0.17

Conditional Formatting – Column Chart

Select visualization -> format visual -> columns -> colours -> conditional formatting (fx) -> rest same

The screenshot shows a Power BI interface with a column chart titled "Conditional Formatting - Column Chart 1". The chart displays "Sales by Sub-Category" on the Y-axis and "Sub-Category" on the X-axis. The bars are blue, and their heights represent sales volume. A tooltip for one bar indicates "Conditional formatting: Make this property change under different conditions that you define." The "Format visual" icon in the "Visualizations" pane is circled in red, highlighting the specific tool used for this feature. The "Fields" pane on the right shows the available fields for the chart.



Conditional Formatting – based on Calculated Column

Go to data section -> create calculated column using DAX expression (switch is dax expression)

Select visualization -> format visual -> bars -> colours -> conditional formatting (fx) ->

format style (field value) -> first category (select calculated column)

Screenshot of the Power BI Column Tools ribbon showing the formula editor for a calculated column named "CF Color". The formula is:

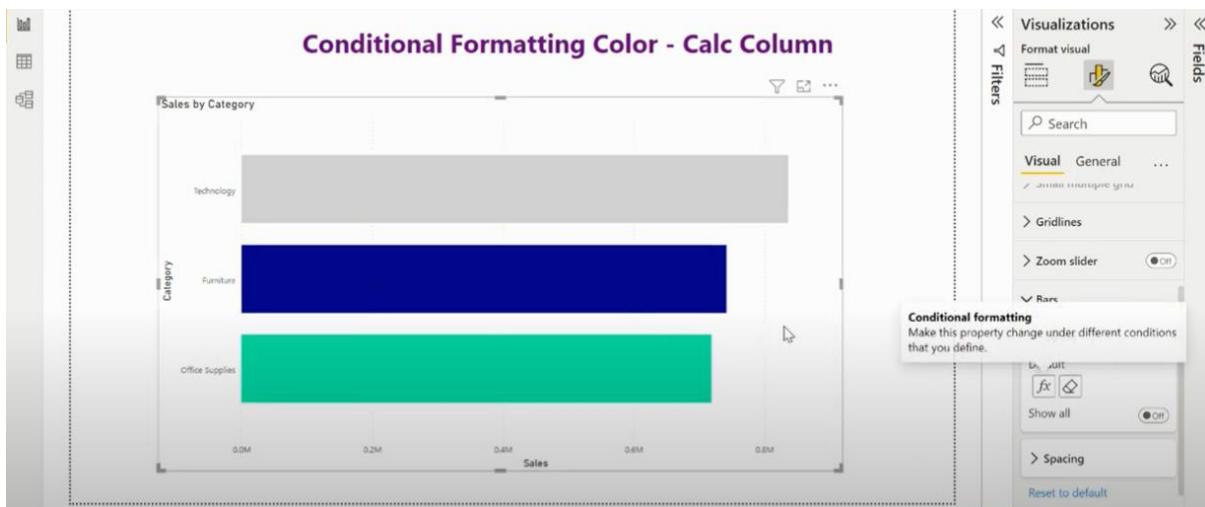
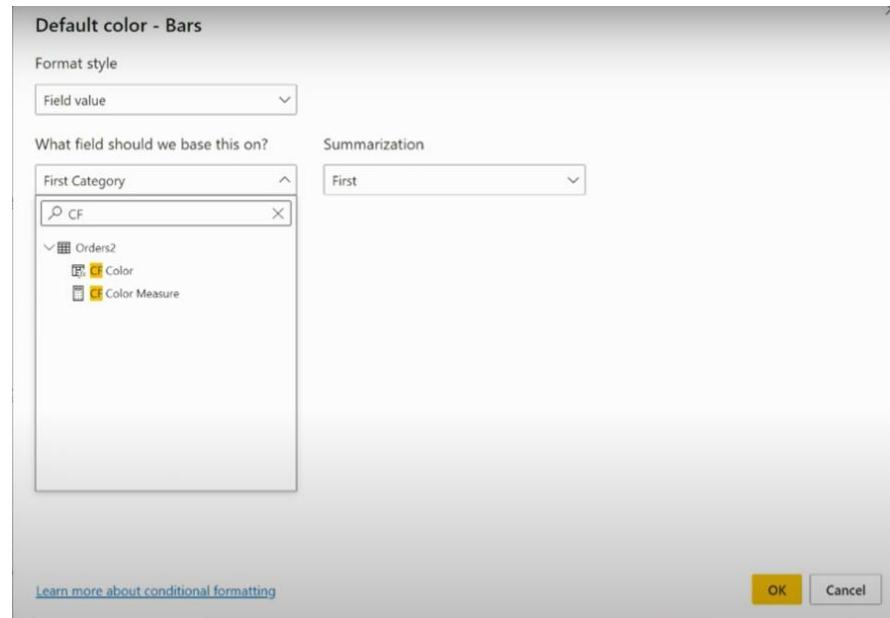
```

1 CF Color =
2 SWITCH(
    Orders[Category],
    "furniture", "darkblue",
    "office supplies", "#00EDA4",
    "lightgrey"
)

```

The formula is circled in red.

Below the ribbon is a table with columns: Product ID, Region, Product Line, Category, Sub-Category, Product Name, Sales, Quantity, Discount, Profit, and CF Color. The table contains various office supply items from different categories and regions.



Conditional Formatting – based on Measure

Go to data section -> create calculated measure using DAX expression (switch is dax expression) (maxx aggregation function is used on top of switch bcz Measures are aggregated values so we need to aggregation function)

Select visualization -> format visual -> bars -> colours -> conditional formatting (fx) ->

format style (field value) -> first category (select calculated column)

The screenshot shows the Power BI Data View interface. A red oval highlights the formula bar at the top, which contains the following DAX code:

```
CF Color Measure =  
MAXX(Orders2,  
SWITCH(  
    Orders2[Category],  
    "Furniture", "darkblue",  
    "Office Supplies", "#00EDA4",  
    "lightgrey")
```

Below the formula bar is a table with columns: Sales Code, Region, Product ID, Category, Sub-Category, Product Name, Sales, Quantity, Discount, Profit, and CF Color. The table contains 18 rows of data. To the right of the table is a Fields pane with various columns and their corresponding data types.

Implicit Measure v/s Explicit Measure

Implicit Measure

Whenever we load any dataset into power Bi, it identifies the numerical columns automatically and it allows us to do summarization on that column by default and it is indicated by the summarization icon(Σ) next to it. It means power bi has identified them as measure. We can change the measure as per our need. This is called as implicit measure.

Implicit measure vs. Explicit measure

The screenshot shows the Power BI interface with the Fields pane open. The pane lists various fields and measures from the 'Orders2' table. Several measures are highlighted with red circles: $\sum \text{Discount}$, $\sum \text{Postal Code}$, $\sum \text{Profit}$, $\sum \text{Quantity}$, $\sum \text{Sales}$, and $\sum \text{Sales2}$.

Implicit measure vs. Explicit measure

The screenshot shows the Power BI interface with the Column Tools ribbon selected. The 'Summation' dropdown is set to 'Sum'. The Fields pane on the right shows the same list of fields and measures as the previous screenshot.

Explicit Measure

Explicit measure can be created by writing the formula. We specify the aggregation to be used explicitly.

Implicit measure are not reusable whereas we can reuse the explicit measure inside other calculations.

Implicit measure vs. Explicit measure

The screenshot illustrates the creation of a new measure in Power BI. In the 'Fields' pane (right), the 'Visuals' section is highlighted with a red circle, and the 'New measure' option is selected. The 'Measure Tools' ribbon shows a measure named 'Total Avg Sales' with the formula `AVERAGE(Orders2[Sales])`. The 'Visualizations' pane (right) displays the newly created measure.

Variables in Power BI

Use of variable makes it easier to read

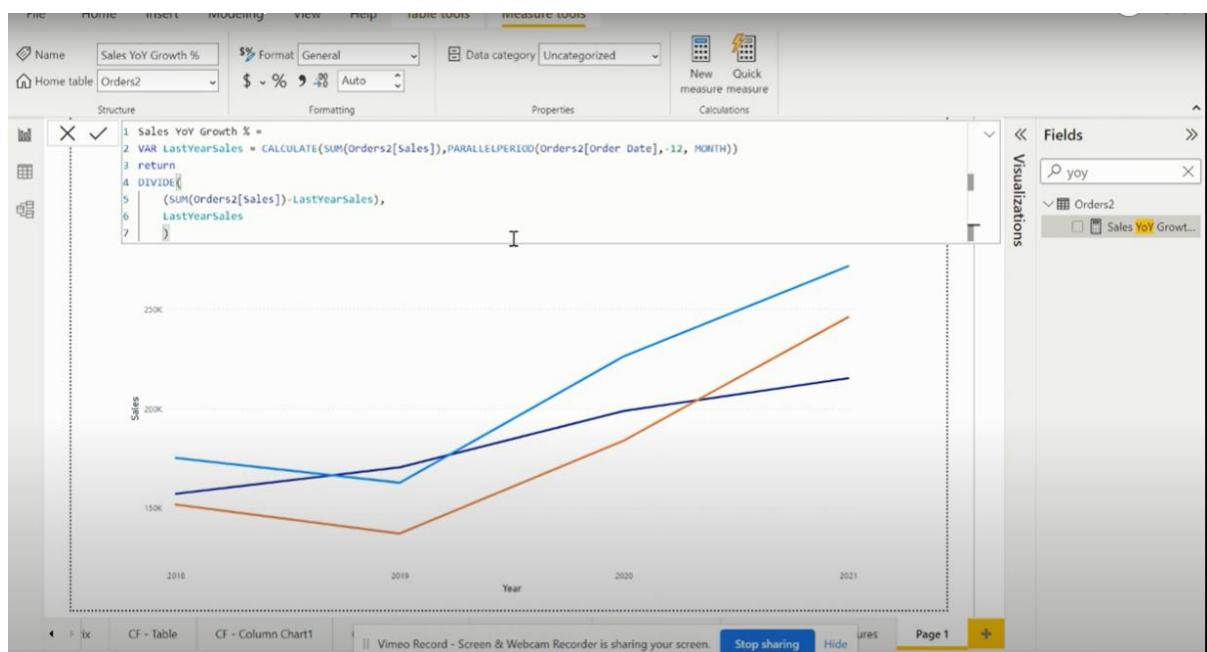
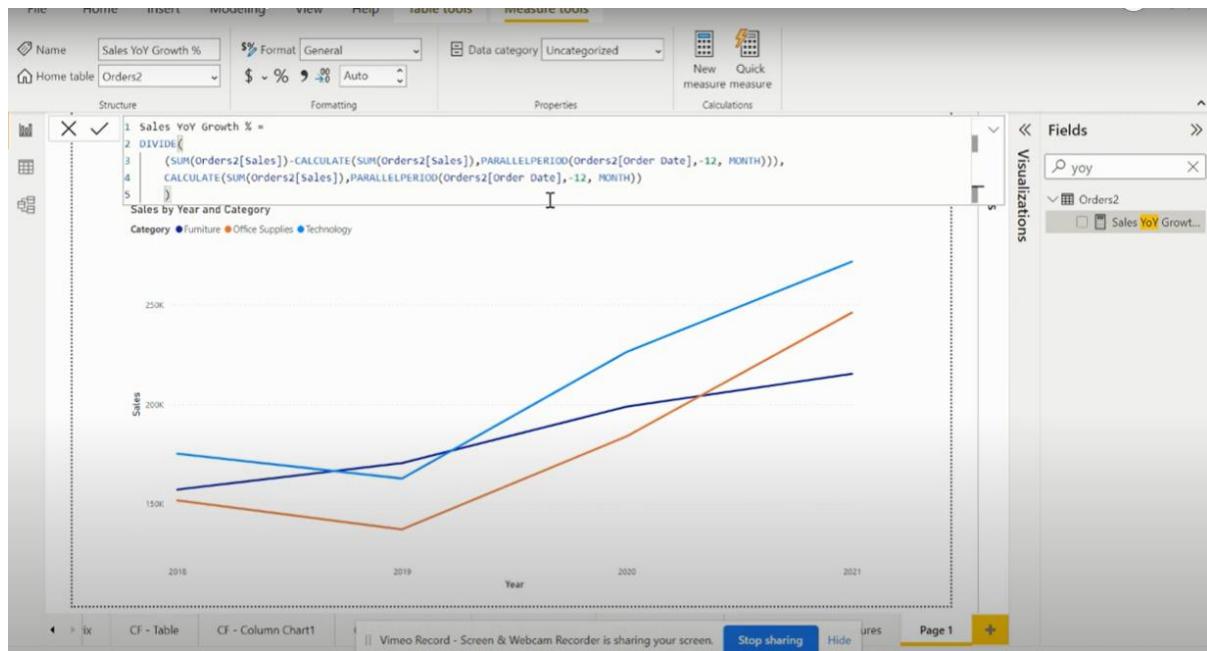
Makes it simpler to debug in case of error

Reduces the complexity of calculation

Improvised the overall performance

“VAR” function is used to define the variable

“Parallelperiod” function is used to calculate sales for previous 12 months



DAX (Data Analysis Expression) in Power BI?

It is used in power bi analysis services and power pivot

It is collection of functions, operators and constants that can be used in formulas/expression in order to calculate or return one or more values.

RankX function in Power BI

Can be created using a new measure

The screenshot shows the Power BI desktop interface. In the top ribbon, the 'Measure tools' tab is selected. A new measure named 'Measure' is being created, with the formula `RANKX(Orders2, [Total Sales],,DESC)`. The formula bar also shows `state ranks = RANKX(Orders2, [Total Sales],,desc)`. The 'Fields' pane on the right lists various fields from the 'Orders2' table, including State, Sales, and other dimensions like Category and Product ID. The 'Values' section under the 'Fields' pane shows 'State' and 'Sales' selected.

Adding newly created **RankX** dax to values. But this is not the result we wanted. Here ranks are calculated at each row (data is filtered at each row)

This screenshot shows the same Power BI desktop environment after the 'State ranks' measure has been added to the 'Sales' column. The formula bar now displays `$% State ranks = RANKX(Orders2, [Total Sales],,DESC)`. The data grid shows the original sales data alongside the newly calculated 'State ranks' column, which contains the value '1' for every row. The 'Fields' pane on the right shows the 'State ranks' measure selected in the 'Values' section.

We will use ALL function inside RankX function to solve this issue.

The screenshot shows the Power BI desktop interface. On the left, there's a table visualization with columns: State, Sales, and State ranks. The table contains data for various US states. The 'State ranks' column is highlighted. In the top ribbon, under 'Measure tools', there's a 'New measure' button. The formula bar at the top shows the formula: `State ranks = RANKX(ALL(Orders2[State]),[total Sales])`. The 'Fields' pane on the right lists several fields, including 'State ranks' which is currently selected.

State	Sales	State ranks
North Dakota	919.91	49
West Virginia	1,209.82	48
Maine	1,270.53	47
South Dakota	1,315.56	46
Wyoming	1,603.14	45
District of Columbia	2,665.02	44
Kansas	2,914.31	43
Idaho	4,382.49	42
Iowa	4,579.76	41
New Mexico	4,783.52	40
Montana	5,589.35	39
New Hampshire	7,292.52	38
Nebraska	7,464.93	37
South Carolina	8,481.71	36
Vermont	8,829.37	35
Louisiana	9,217.03	34
Mississippi	10,771.34	33
Utah	11,220.06	32
Arkansas	11,678.13	31
Connecticut	13,384.36	30
Nevada	16,729.10	29
Total	22,97,200.86	1

Adding another column inside the visualization. Again filtering is done at row level.

This screenshot shows the same Power BI desktop environment. The table now includes an additional column 'City'. The 'State ranks' column remains highlighted. The formula bar still displays the same RANKX formula. The 'Fields' pane on the right shows the expanded data model, including the newly added 'City' field.

State	City	Sales	State ranks
Alabama	Auburn	1,765.83	1
Alabama	Decatur	3,374.82	1
Alabama	Hoover	525.85	1
Alabama	Mobile	5,462.99	1
Alabama	Montgomery	3,722.73	1
Alabama	Tuscaloosa	175.70	1
Arizona	Avondale	946.81	1
Arizona	Bullhead City	22.29	1
Arizona	Chandler	1,076.75	1
Arizona	Gilbert	4,172.38	1
Arizona	Glendale	2,917.87	1
Arizona	Mesa	4,037.74	1
Arizona	Peoria	1,341.35	1
Arizona	Phoenix	11,000.26	1
Arizona	Scottsdale	1,466.31	1
Arizona	Sierra Vista	76.07	1
Arizona	Tempe	1,070.30	1
Arizona	Tucson	6,313.02	1
Arizona	Yuma	840.87	1
Arkansas	Conway	301.96	1
Arkansas	Fayetteville	3,742.81	1
Total		22,97,200.86	1

Inserting 2 ALL functions inside RankX is not an option.

The screenshot shows the Power BI Model view. In the 'Measure tools' ribbon, a measure named 'State ranks' is defined with the formula: `RANKX(ALL(Orders2[State]),ALL(Orders2[City]),[Total Sales])`. The visual area displays an error message: 'Can't display the visual. See details'. The Fields pane on the right lists various dimensions and measures, including 'State', 'City', 'Sales', and 'Order Date'.

2 ALL functions needs to be defined inside crossjoin function.

The screenshot shows the Power BI Model view. In the 'Measure tools' ribbon, a measure named 'State ranks' is defined with the formula: `RANKX(crossjoin(ALL(Orders2[State]),ALL(Orders2[City])),[Total Sales])`. The visual area displays an error message: 'Can't display the visual. See details'. The Fields pane on the right lists various dimensions and measures, including 'State', 'City', 'Sales', and 'Order Date'.

The screenshot shows the Power BI Data view. A table is displayed with columns: State, City, Sales, and State ranks. The data includes rows for New York, California, Washington, etc., with the 'State ranks' column showing the rank of each state's total sales. The ribbon at the top shows the 'HOME' tab selected. The Fields pane on the right lists various dimensions and measures, including 'State', 'City', 'Sales', and 'Order Date'.

State	City	Sales	State ranks
New York	New York City	2,56,368.16	1
California	Los Angeles	1,75,851.34	2
Washington	Seattle	1,19,540.74	3
California	San Francisco	1,12,669.09	4
Pennsylvania	Philadelphia	1,09,077.01	5
Texas	Houston	64,504.76	6
Illinois	Chicago	48,539.54	7
California	San Diego	47,521.03	8
Michigan	Detroit	42,446.94	9
Florida	Jacksonville	39,133.33	10
Texas	San Antonio	21,843.53	11
Delaware	Newark	20,448.05	12
Texas	Dallas	20,131.93	13
Indiana	Lafayette	19,630.45	14
Georgia	Atlanta	17,197.84	15
Minnesota	Minneapolis	16,870.54	16
Virginia	Springfield	16,628.53	17
Rhode Island	Providence	15,980.65	18
Ohio	Columbus	15,900.79	19
Kentucky	Henderson	15,661.01	20
Michigan	Jackson	15,420.04	21
Total		22,97,200.86	1

TOPN function in Power BI

Used to get top and bottom N values from the data.

Can be created using a new table (**data section -> new table**)

Desc – will give top n values

Asc – will give bottom n values

The screenshot shows the Power BI Data Editor interface. At the top, the ribbon has 'File', 'Home', 'Help', and 'Table Tools'. The 'Table Tools' tab is selected, showing a toolbar with icons for Name, Structure, Column, Row, and Cell. The 'Name' field is set to 'Table'. The 'Structure' pane shows a single row with the formula: `1 Top5States = TOPN(5, Orders2, [Total Sales], DESC)`. The 'Fields' pane on the right lists various columns from the 'Orders2' table, including Category, CF Color, CF Color Measure, City, Country/Region, Customer ID, Customer Name, Discount, Order Date, Order ID, Postal Code, Product ID, Product Name, Profit, Quantity, Rankx for State, Region, and Row ID. Below the ribbon, there's another toolbar with 'Mark as date table', 'Manage relationships', 'New measure', 'Quick measure', 'New column', 'New table', and 'Calculations'. The main area displays a table with columns: Row ID, Order ID, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name, Segment, Country/Region, City, State, and Postal Code. The table contains five rows of data.

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	State	Postal Code
4191	CA-2021-166709	17-11-2021 00:00:00	22-11-2021 00:00:00	Standard Class	HL-15040	Hunter Lopez	Consumer	United States	Newark	Delaware	21
8154	CA-2021-140151	23-03-2021 00:00:00	25-03-2021 00:00:00	First Class	RH-19360	Raymond Buch	Consumer	United States	Seattle	Washington	91
2624	CA-2021-127180	22-10-2021 00:00:00	24-10-2021 00:00:00	First Class	TA-21385	Tom Ashbrook	Home Office	United States	New York City	New York	24
2698	CA-2018-145317	18-03-2018 00:00:00	23-03-2018 00:00:00	Standard Class	SM-20320	Sean Miller	Home Office	United States	Jacksonville	Florida	32
6827	CA-2020-118689	02-10-2020 00:00:00	09-10-2020 00:00:00	Standard Class	TC-20980	Tamara Chand	Corporate	United States	Lafayette	Indiana	41

ALL function in Power BI

ALL function is used to ignore any filter being applied on the data

The screenshot shows a Power BI interface with the title "ALL Function in DAX". On the left, there is a "Ship Mode" slicer with options: First Class, Same Day, Second Class, Standard Class. Below it is a "Category" slicer with options: Furniture, Office Supplies, Technology. The main area displays a table titled "Region Sales" with the following data:

Region	Sales
Central	5,01,239.89
East	6,78,781.24
South	3,91,721.91
West	7,25,457.82
Total	22,97,200.86

The Fields pane on the right lists various fields under the "Orders2" table, including Category, CF Color, CF Color Meas..., City, Country/Region, Customer ID, Customer Name, Discount, Order Date, and Postal Code.

After applying filter (from category slicer) sales values are getting updated. But I want to see all sales value for each region alongside filtered value.

The screenshot shows a Power BI interface with the title "ALL Function in DAX". The "Category" slicer on the left has "Furniture" selected. The main area displays a table titled "Region Sales" with the following data:

Region	Sales
Central	1,63,797.16
East	2,08,291.20
South	1,17,298.68
West	2,52,612.74
Total	7,41,999.80

The Fields pane on the right lists various fields under the "Orders2" table, including Category, CF Color, CF Color Meas..., City, Country/Region, Customer ID, Customer Name, Discount, Order Date, and Postal Code.

Let's create new dax function with ALL function (on region column).

The screenshot shows the Power BI Model view. In the top ribbon, the 'Measure Tools' tab is selected. A new measure named 'Sales (ALL)' has been created, defined by the formula: `CALCULATE([Total Sales], ALL(Orders2[Region]))`. The visual below displays 'Region Sales' data:

Region	Sales
Central	5,01,239.89
East	6,78,781.24
South	3,91,721.91
West	7,25,457.82
Total	22,97,200.86

Two filters are applied: 'Ship Mode' (First Class, Same Day, Second Class, Standard Class) and 'Category' (Furniture, Office Supplies, Technology). The Fields pane on the right lists various dimensions and measures, including 'Region' and 'Sales (ALL)'.

But it is giving total sales value for each region.

The screenshot shows the Power BI Data view. The ribbon is set to 'Data / Drill'. A table visual is displayed with the title 'ALL Function in DAX'. The table contains the following data:

Region	Sales	Sales (ALL)
Central	5,01,239.89	22,97,200.86
East	6,78,781.24	22,97,200.86
South	3,91,721.91	22,97,200.86
West	7,25,457.82	22,97,200.86
Total	22,97,200.86	22,97,200.86

Two filters are applied: 'Ship Mode' (First Class, Same Day, Second Class, Standard Class) and 'Category' (Furniture, Office Supplies, Technology). The Fields pane on the right lists various dimensions and measures, including 'Region' and 'Sales (ALL)'.

Lets use **ALL** function (on category column).

The screenshot shows the Power BI Data Editor interface. In the top ribbon, the 'Measure Tools' tab is selected. A new measure named 'Sales (ALL)' is being created with the formula: `Sales (ALL) = CALCULATE([Total Sales],ALL(Orders2[Category]))`. The formula bar has a small error icon. The left sidebar shows two filters applied: 'Ship Mode' (First Class, Same Day, Second Class, Standard Class) and 'Category' (Furniture, Office Supplies, Technology). The main area displays a table with columns 'Region', 'Sales', and 'Sales (ALL)'. The data is as follows:

Region	Sales	Sales (ALL)
Central	5,01,239.89	22,97,200.86
East	6,78,781.24	22,97,200.86
South	3,91,721.91	22,97,200.86
West	7,25,457.82	22,97,200.86
Total	22,97,200.86	22,97,200.86

The right sidebar lists various fields from the data model, with 'Sales (ALL)' highlighted.

This is the way...

The screenshot shows the Power BI Data Editor interface. The formula for the measure 'Sales (ALL)' has been changed to: `Sales (ALL) = CALCULATE([Total Sales],ALL(Orders2[Category]))`. The rest of the interface is identical to the previous screenshot, including the filters, table data, and highlighted 'Sales (ALL)' in the field list.

Relationships in Power BI

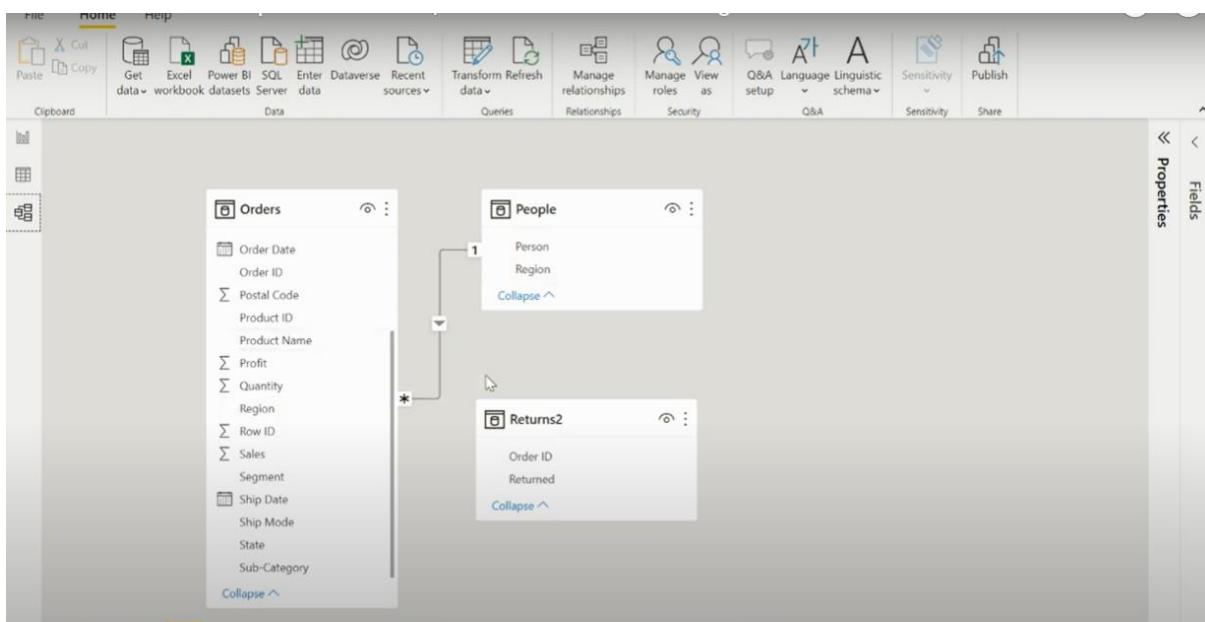
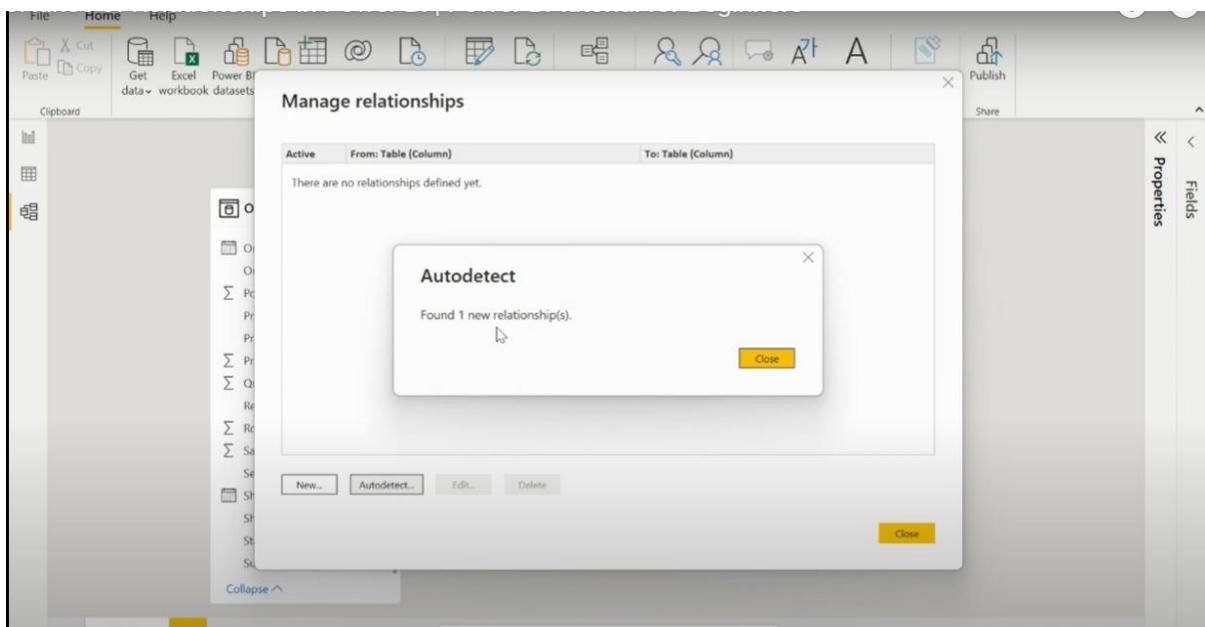
The screenshot shows the Power BI desktop application. On the left, there's a navigation pane with icons for Home, Get data, Data, Transform data, Manage relationships, Q&A, Language setup, Sensitivity, and Share. The main area displays a report titled "Importance of Relationships in Power BI". The report contains the following sections:

- 1.Why is it relevant to create relationships?**
 - It helps to accurately calculate results and display the correct information in our PBI reports.
 - We can work with data in both tables as if they were a single table
- 2.Ways to create relationships in Power BI?**
 - .Manual
 - .Auto detect
- 3.How to edit a relationship?**
- 4.Configure additional options**
- 4.1Cardinality**
 - .Many to one (*:1)
 - .One to one (1:1)
 - .One to many (1:*)
 - .Many to many (*:*)

How to create relationship?

The screenshot shows the Power BI desktop application with the "Manage relationships" dialog box open. The dialog box allows users to "Add, edit, or remove relationships between tables." Three tables are listed: "Orders", "People", and "Returns2".

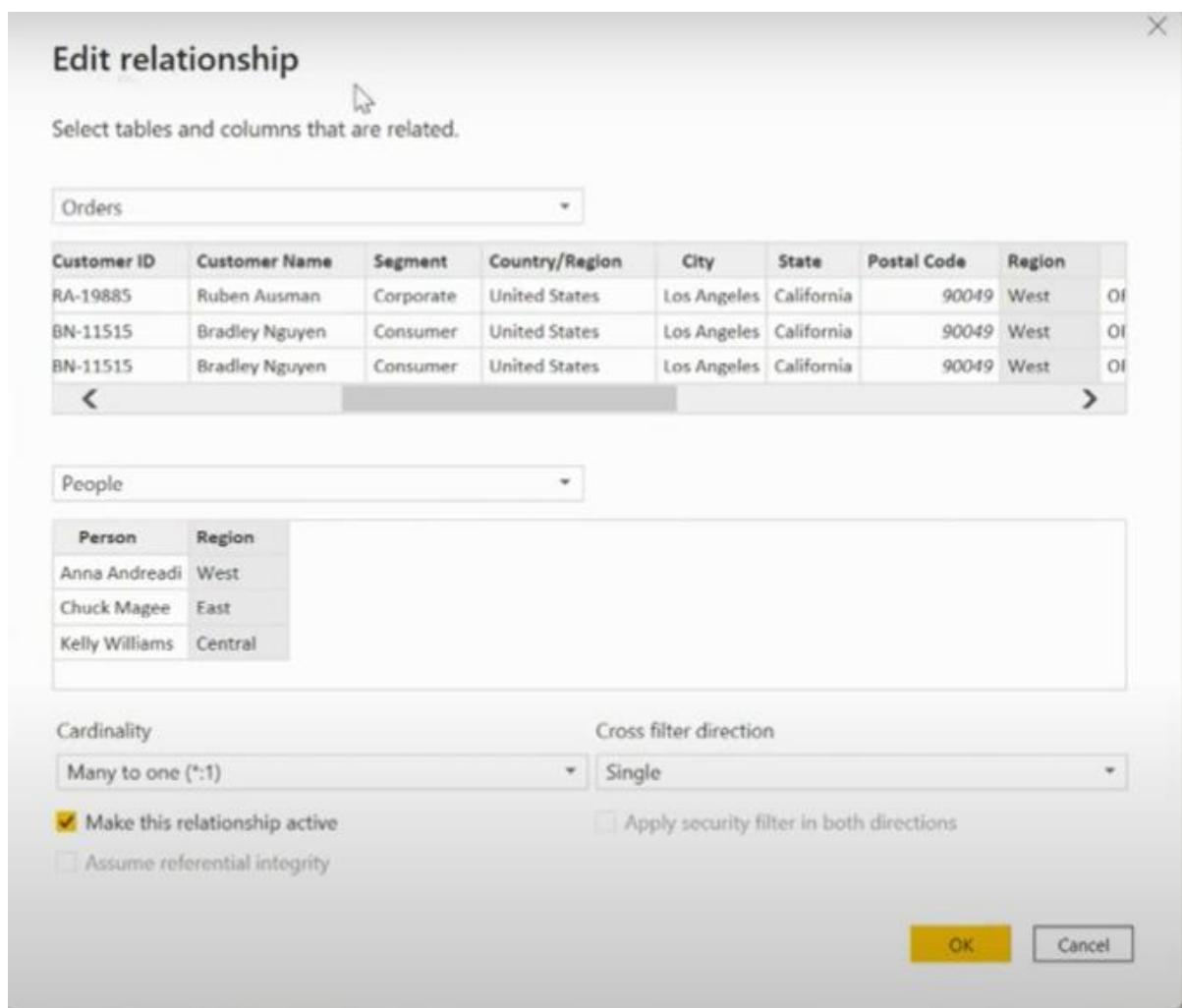
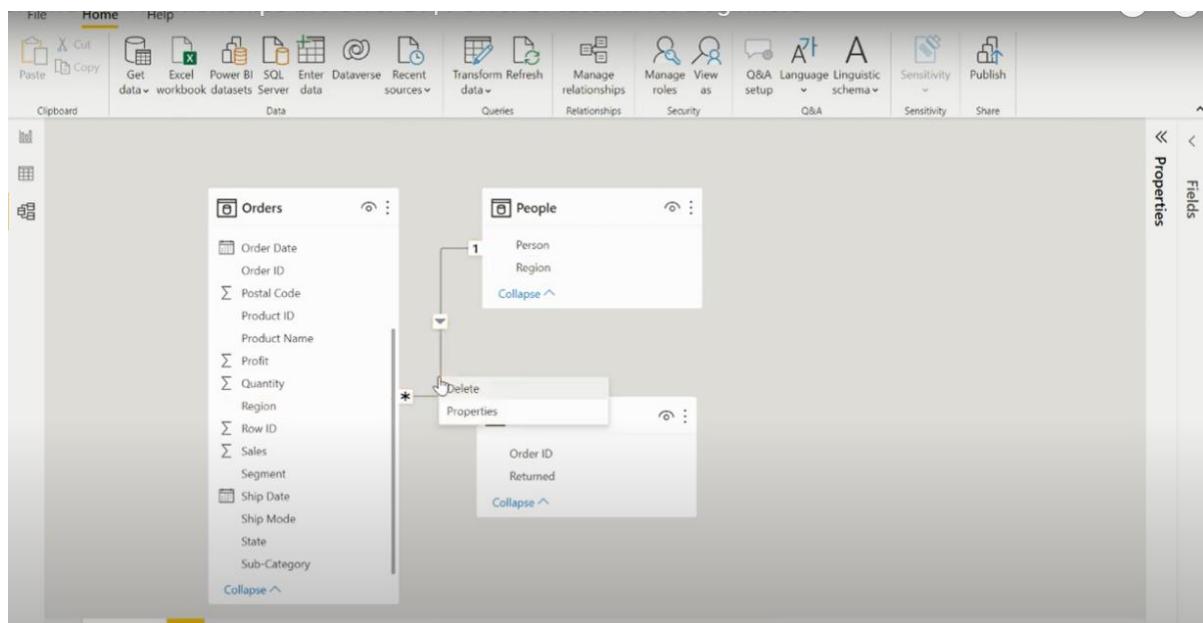
- Orders** table fields:
 - Order Date
 - Order ID
 - Postal Code
 - Product ID
 - Product Name
 - Profit
 - Quantity
 - Region
 - Row ID
 - Sales
 - Segment
 - Ship Date
 - Ship Mode
 - State
 - Sub-Category
- People** table fields:
 - Person
 - Region
- Returns2** table fields:
 - Order ID
 - Returned



To define relationship manually, just drag and drop the field from one table to another table and that will create the relationship between two tables.

How to edit relationship?

Select relationship line -> it will turn yellow upon selecting -> right click -> properties



Cardinality

Cardinality and One-to-many relationship in Power BI

· Cardinality is a mathematical term. It translates into the number of elements in a set.
· Cardinality refers to the relationships between the data in two database tables. Cardinality defines how many instances of one entity are related to instances of another entity.

The diagram illustrates a one-to-many relationship between the Doctor and Patient tables. The Doctor table has columns DoctorID (PK) and DoctorName. The Patient table has columns PatientID (PK), PatientName, and Doctor_DoctorID (FK). A line connects DoctorID to Doctor_DoctorID, indicating that many patients can be associated with one doctor.

One to many: Doctor can have many patients on a given day

One to one: A patient can have one doctor

Person (from peoples table) and sales (from orders table) – same

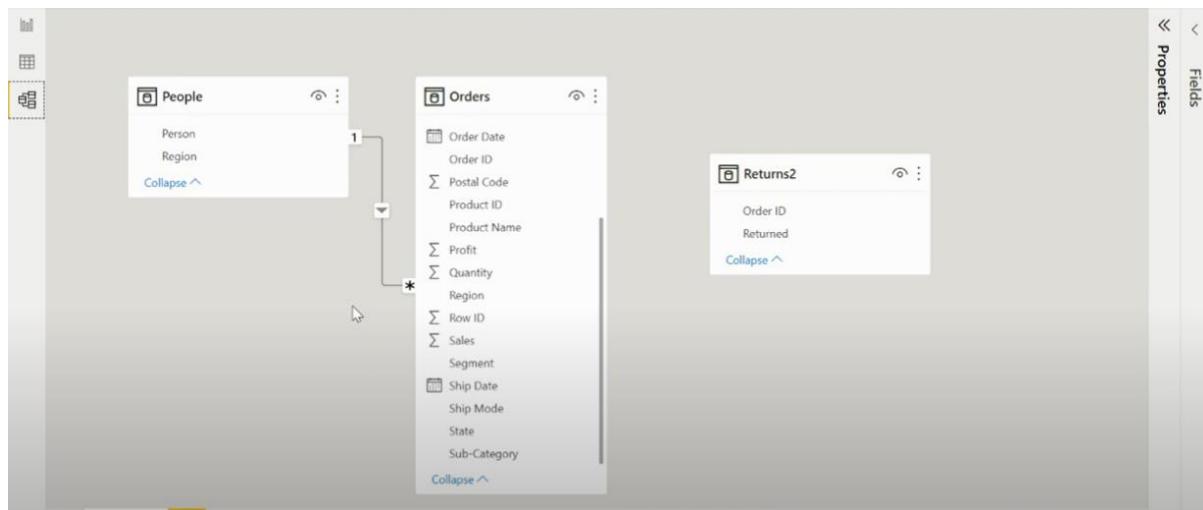
Cardinality and One-to-many relationship in Power BI

· Cardinality is a mathematical term. It translates into the number of elements in a set.
· Cardinality refers to the relationships between the data in two database tables. Cardinality defines how many instances of one entity are related to instances of another entity.

The diagram illustrates a one-to-many relationship between the Doctor and Patient tables, and a fact table Sales. The Doctor table has columns DoctorID (PK) and DoctorName. The Patient table has columns PatientID (PK), PatientName, and Doctor_DoctorID (FK). The Sales table has columns Person and Sales, listing individual sales records. A line connects DoctorID to Doctor_DoctorID, indicating that many patients can be associated with one doctor.

Person	Sales
Anna Andreadi	22,97,200.86
Cassandra Bran	22,97,200.86
Chuck Magee	22,97,200.86
Kelly Williams	22,97,200.86
Total	22,97,200.86

Let's define the relationship between this two tables



After defining the relationship, the sales values are updated

The screenshot shows a Power BI report. At the top, a section titled 'Cardinality and One-to-many relationship in Power BI' contains two bullet points: '.Cardinality is a mathematical term. It translates into the number of elements in a set.' and '.Cardinality refers to the relationships between the data in two database tables. Cardinality defines how many instances of one entity are related to instances of another entity.' Below this is a diagram showing a 'Doctor' table (DoctorID int PK, DoctorName varchar(40)) connected via a one-to-many relationship to a 'Patient' table (PatientID int PK, PatientName varchar(40), Doctor_DoctorID int FK). A table below shows 'Person' and 'Sales' data:

Person	Sales
Anna Andreadi	7,25,457.82
Cassandra Brandow	3,91,721.91
Chuck Magee	6,78,781.24
Kelly Williams	5,01,239.89
Total	22,97,430.86

Let's define relationship between orders table and returns table. The relationship is having many to many cardinality. So, we cannot create the relationship between this two tables. This is because these two tables are having duplicate order ids in them.

How to overcome this?

Model view -> home -> Transform data -> transform data -> it will take to power query editor -> append queries -> append queries as new -> select two tables option -> first table (orders) -> second table (returns) -> right click on order id column -> remove other columns -> right click on order id column -> remove duplicates -> rename the table -> close & apply -> close & apply

Append

Concatenate rows from two tables into a single table.

Two tables Three or more tables

First table: Orders

Second table: Returns2

OK Cancel

Row ID	Order ID	Order Date	Ship Date	Ship Mode
1	1	2020-01-01	2020-01-02	Standard Class
2	2	2020-01-01	2020-01-02	Standard Class
3	3	2020-01-01	2020-01-02	Standard Class
4	4	2020-01-01	2020-01-02	Standard Class
5	5	2020-01-01	2020-01-02	Standard Class
6	6	2020-01-01	2020-01-02	Standard Class
7	7	2020-01-01	2020-01-02	Standard Class
8	8	2020-01-01	2020-01-02	Standard Class
9	9	2020-01-01	2020-01-02	Standard Class
10	10	2020-01-01	2020-01-02	Standard Class
11	11	2020-01-01	2020-01-02	Standard Class
12	12	2020-01-01	2020-01-02	Standard Class
13	13	2020-01-01	2020-01-02	Standard Class
14	14	2020-01-01	2020-01-02	Standard Class
15	15	2020-01-01	2020-01-02	Standard Class
16	16	2020-01-01	2020-01-02	Standard Class
17	17	2020-01-01	2020-01-02	Standard Class
18	18	2020-01-01	2020-01-02	Standard Class
19	19	2020-01-01	2020-01-02	Standard Class
20	20	2020-01-01	2020-01-02	Standard Class
21	21	2020-01-01	2020-01-02	Standard Class
22	22	2020-01-01	2020-01-02	Standard Class
23	23	2020-01-01	2020-01-02	Standard Class

Append1

Combine rows from two tables into a single table.

Orders, Returns2

OK Cancel

Row ID	Order ID	Order Date	Ship Date	Ship Mode
1	1	2020-01-01	2020-01-02	Second Class
2	2	2020-01-01	2020-01-02	Second Class
3	3	2020-01-01	2020-01-02	Second Class
4	4	2020-01-01	2020-01-02	Second Class
5	5	2020-01-01	2020-01-02	Second Class
6	6	2020-01-01	2020-01-02	Second Class
7	7	2020-01-01	2020-01-02	Second Class
8	8	2020-01-01	2020-01-02	Second Class
9	9	2020-01-01	2020-01-02	Second Class
10	10	2020-01-01	2020-01-02	Second Class
11	11	2020-01-01	2020-01-02	Second Class
12	12	2020-01-01	2020-01-02	Second Class
13	13	2020-01-01	2020-01-02	Second Class
14	14	2020-01-01	2020-01-02	Second Class
15	15	2020-01-01	2020-01-02	Second Class
16	16	2020-01-01	2020-01-02	Second Class
17	17	2020-01-01	2020-01-02	Second Class
18	18	2020-01-01	2020-01-02	Second Class
19	19	2020-01-01	2020-01-02	Second Class
20	20	2020-01-01	2020-01-02	Second Class
21	21	2020-01-01	2020-01-02	Second Class
22	22	2020-01-01	2020-01-02	Second Class
23	23	2020-01-01	2020-01-02	Second Class

Append1

Select columns from the source table.

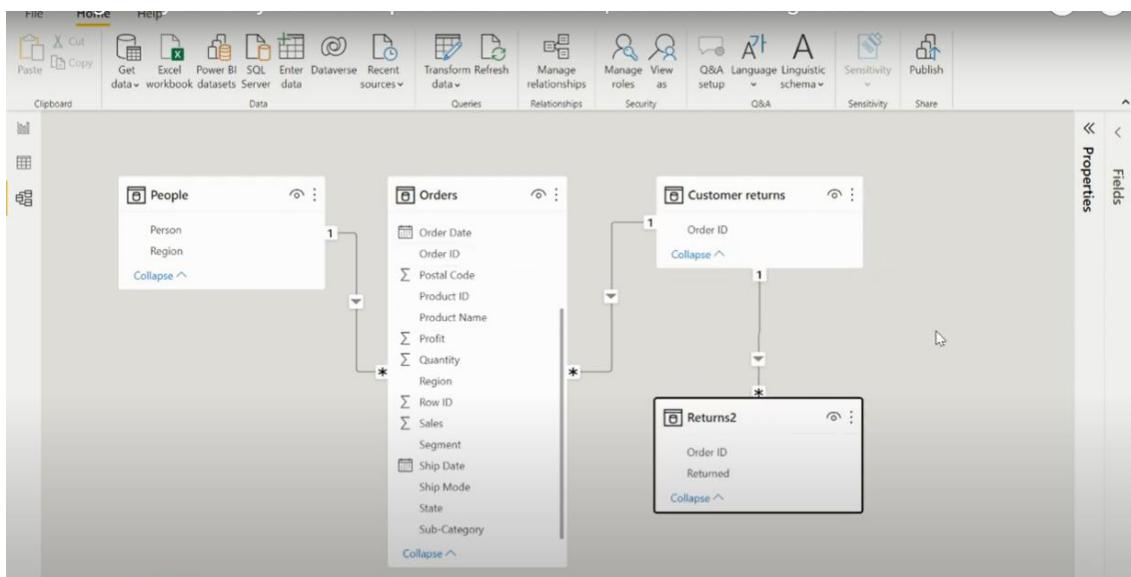
Order ID

OK Cancel

Order ID
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

The screenshot shows the Power BI Query Editor interface. A red circle highlights the 'Properties' section on the right, which displays the name 'Customer returns'. Below it, the 'Applied Steps' section lists the steps taken: 'Source', 'Removed Other Columns', and 'Removed Duplicates'. The main area shows a table with 25 rows of data under the heading 'Order ID'.

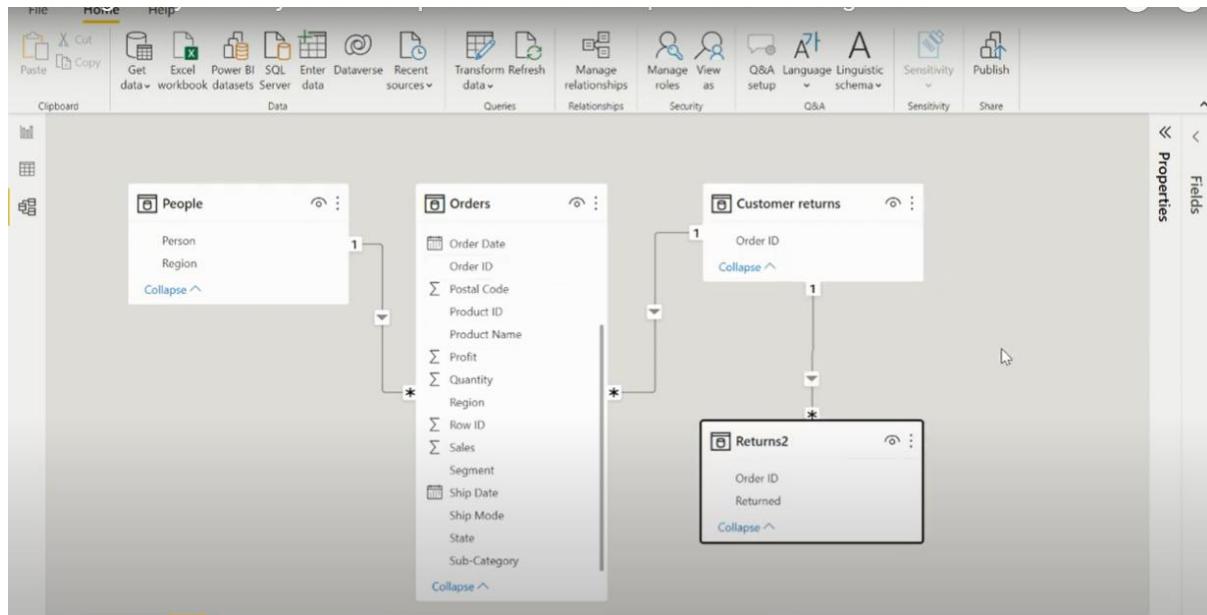
Order ID
CA-2020-152156
CA-2020-138688
US-2019-108966
CA-2018-158182
CA-2021-114412
CA-2020-161389
US-2019-118983
CA-2018-105893
CA-2018-167164
CA-2018-143336
CA-2020-137330
US-2021-156909
CA-2019-106320
CA-2020-121755
US-2019-150630
CA-2021-107727
CA-2020-117590
CA-2019-117415
CA-2021-120999
CA-2020-101343
CA-2021-139619
CA-2020-118255
CA-2018-146703
CA-2020-169194
CA-2019-115742



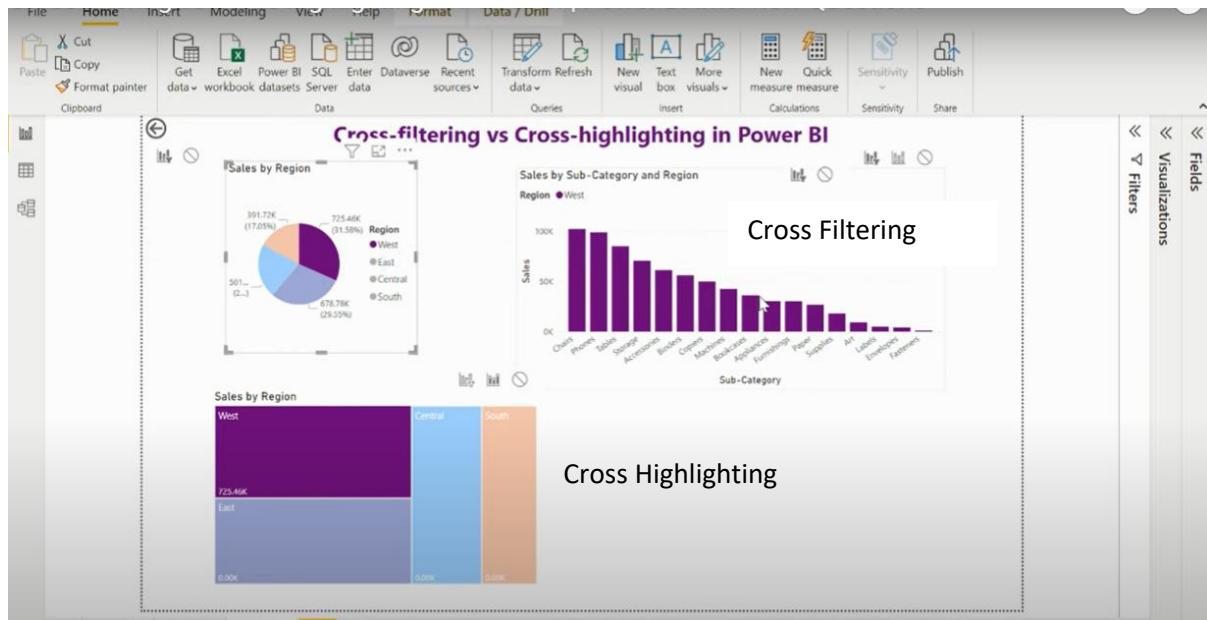
The screenshot shows the Power BI Query Editor interface. A red circle highlights the 'Properties' section on the right, which displays the name 'Customer returns'. Below it, the 'Applied Steps' section lists the steps taken: 'Source', 'Removed Other Columns', and 'Removed Duplicates'. The main area shows a table with 25 rows of data under the heading 'Order ID'.

Order ID
CA-2020-152156
CA-2020-138688
US-2019-108966
CA-2018-158182
CA-2021-114412
CA-2020-161389
US-2019-118983
CA-2018-105893
CA-2018-167164
CA-2018-143336
CA-2020-137330
US-2021-156909
CA-2019-106320
CA-2020-121755
US-2019-150630
CA-2021-107727
CA-2020-117590
CA-2019-117415
CA-2021-120999
CA-2020-101343
CA-2021-139619
CA-2020-118255
CA-2018-146703
CA-2020-169194
CA-2019-115742

Now the relationship is created between newly created customer returns table & orders table as one to many and also customer returns table & returns table as one to many.

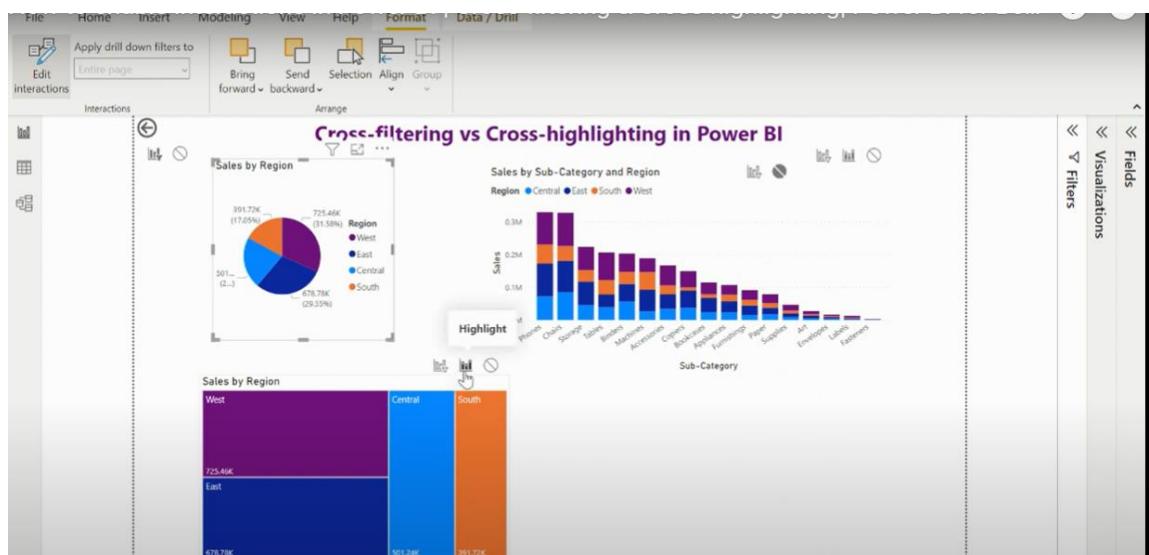
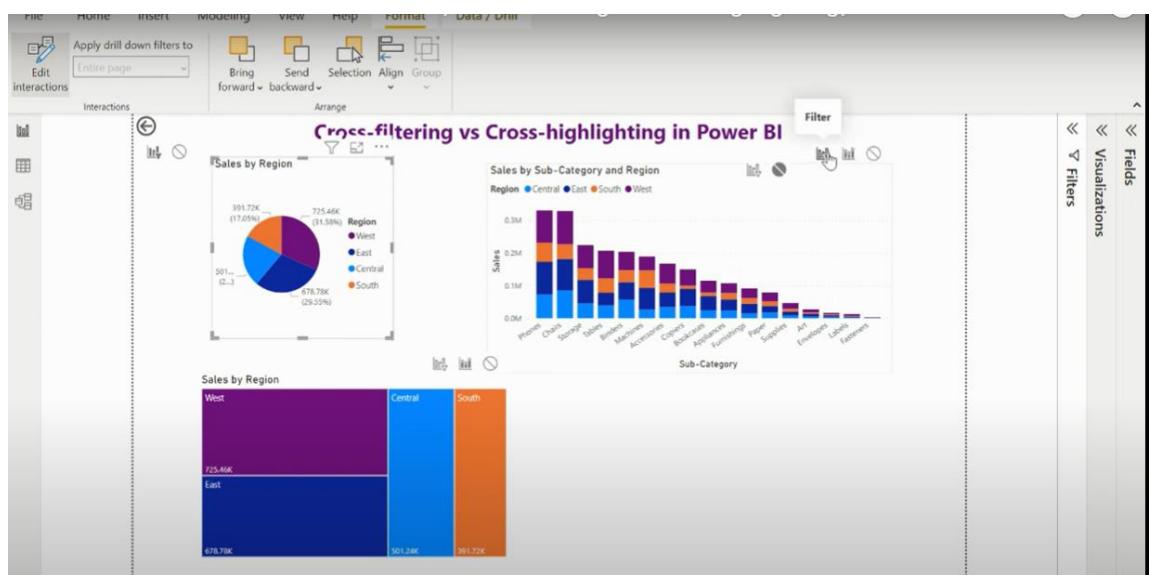
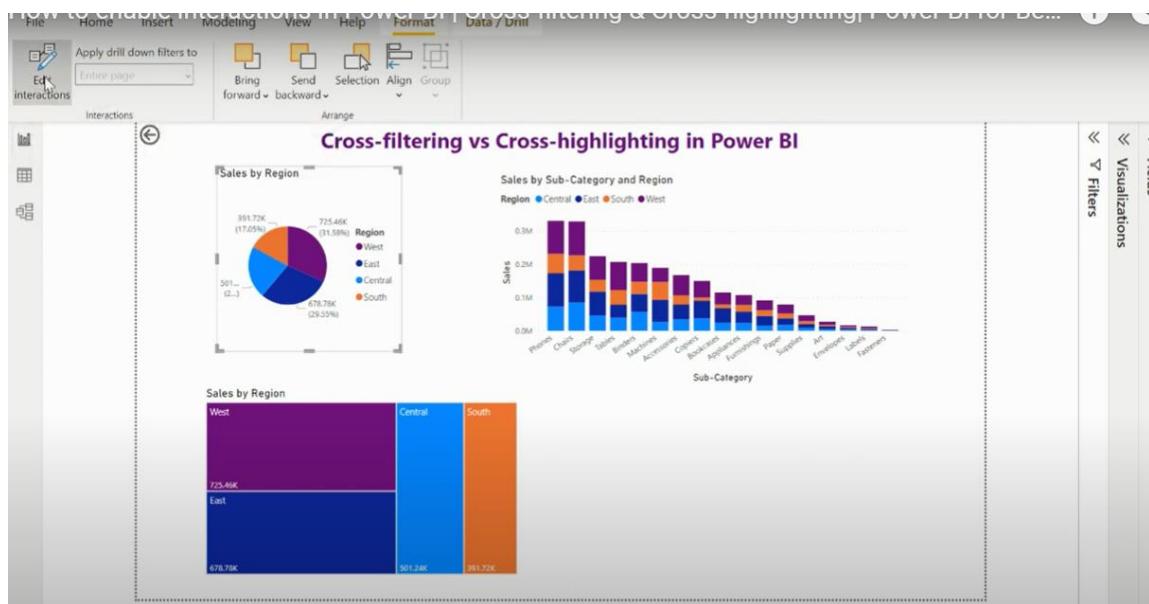


Cross filtering v/s cross highlighting



How to enable interactions (Cross filtering & cross highlighting) in Power BI ?

Select the visualization -> go to format -> edit interactions -> select filter/highlight option



How to calculate YTD, QTD and MTD in Power BI?

YTD – Start of current year to today's date

QTD – Start of current quarter to today's date

MTD – Start of current month to today's date

Create a new measure

The screenshot shows the Power BI Model view. In the top ribbon, 'Measure tools' is selected. A new measure is being created with the name 'Measure'. The formula is set to `TOTALYTD(SUM(Orders[Sales]),Orders[Order Date])`. The formula bar also displays the description: 'Evaluates the specified expression over the interval which begins on the first day of the year and ends with the last date in the specified date column after applying specified filters.' A dropdown menu is open, showing various date-related fields: .[Date], .[Day], .[MonthNo], .[Month], .[QuarterNo], .[Quarter], and .[Year]. On the right side, there is a 'Fields' pane containing a tree view of available fields from the 'Orders' table, such as Category, City, Country/Region, Customer ID, Customer Name, Discount, Measure, Order ID, Order Date, Postal Code, Product ID, Product Name, Profit, Quantity, Region, and Sales.

Add Card visualization -> select newly created measure

The screenshot shows the Power BI Model view. The 'Visualizations' pane on the right has the 'Card' icon highlighted with a red circle. The main area displays a card visualization with the value '470.53K' and the text 'Total YTD Sales'. The top ribbon shows 'Measure tools' is still selected. The formula for the measure is visible in the formula bar: `TOTALYTD(SUM(Orders[Sales]),Orders[Order Date])`. The 'Fields' pane on the right lists various fields from the 'Orders' table, including Order ID, Postal Code, Product ID, Product Name, Profit, Quantity, Region, Sales, Segment, Ship Date, Ship Mode, State, Sub-Category, Total QTD Sales, Total Year to date Sales, Total YTD days, and Total YTD Sales.

If we want to see values for the previous years we can select the slicer accordingly.

The screenshot shows a Power BI report titled "YTD | QTD | MTD in Power BI". It displays two cards: "Total YTD Sales" with a value of "609.21K" and "Total Year to date Sales" with a value of "609.21K". Below the cards is a "Year" slicer with options for 2018, 2019, 2020, and 2021. The "2020" option is selected. To the right of the report is the Power BI Fields pane, which lists various fields such as Order ID, Postal Code, Product ID, Product Name, Profit, Quantity, Region, Row ID, Sales, Segment, Ship Date, Ship Mode, State, Sub-Category, Total QTD Sales, Total Year to da..., Total YTD days, and Total YTD Sales. The "Year" field is highlighted in the Fields pane.

The screenshot shows a second instance of the Power BI report "YTD | QTD | MTD in Power BI". It displays two cards: "Total YTD Sales" with a value of "484.25K" and "Total Year to date Sales" with a value of "484.25K". Below the cards is a "Year" slicer with options for 2018, 2019, 2020, and 2021. The "2018" option is selected. To the right of the report is the Power BI Fields pane, which lists various fields such as Order ID, Postal Code, Product ID, Product Name, Profit, Quantity, Region, Row ID, Sales, Segment, Ship Date, Ship Mode, State, Sub-Category, Total QTD Sales, Total Year to da..., Total YTD days, and Total YTD Sales. The "Year" field is highlighted in the Fields pane.

Similarly, QTD and MTD measures can be created.

Calculate days in YTD (Calculating no of days elapsed between start of year and current date)

Create a new measure

The screenshot shows the Power BI Model view. A new measure is being created with the name 'days_in_ytd'. The formula bar contains the DATEDIFF function: `DATEDIFF(Date1, Date2, Interval)`. The 'Date1' argument is set to `DATE(year(TODAY()),1,1)`, and the 'Date2' argument is set to `TODAY()`. The 'Interval' argument is set to `DAY`. A tooltip indicates that 'Too few arguments were passed to the DATEDIFF function. The minimum argument is 2.' The Fields pane on the right lists various fields, and the Visualizations pane shows a card visualization with the value '733.22K'.

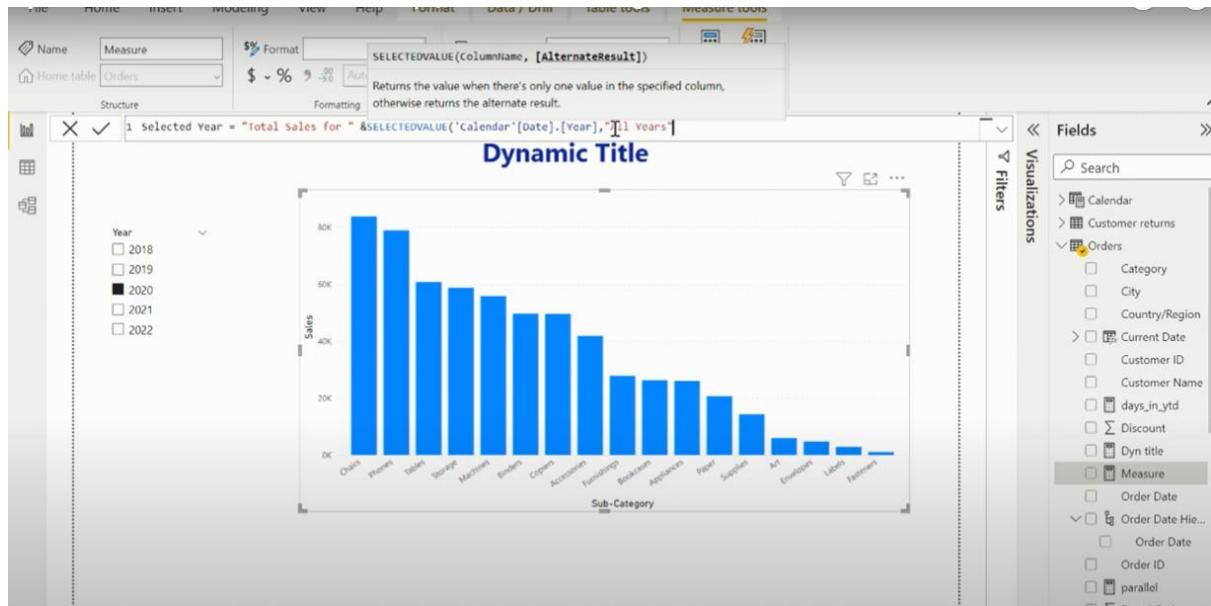
Add Card visualization -> select newly created measure

The screenshot shows the Power BI Home view. A card visualization displays the value '210' for the measure 'days_in_ytd'. The Fields pane on the right shows the newly created measure 'days_in_ytd' selected. The visualizations pane shows the available card visualization type.

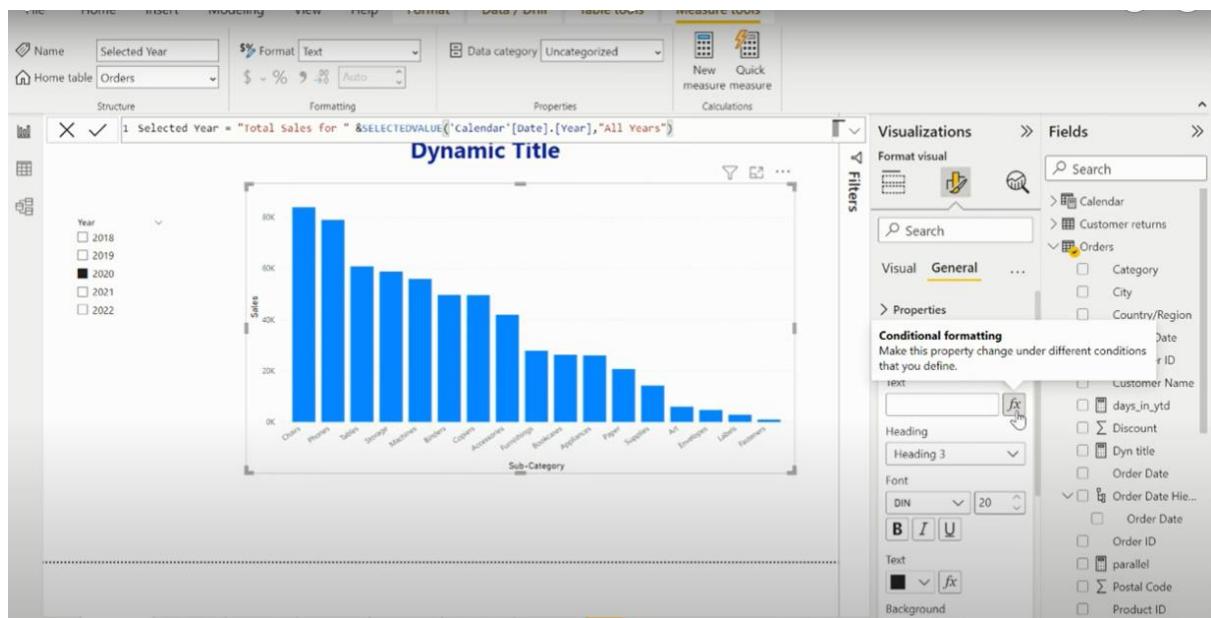
Dynamic Titles in Power BI?

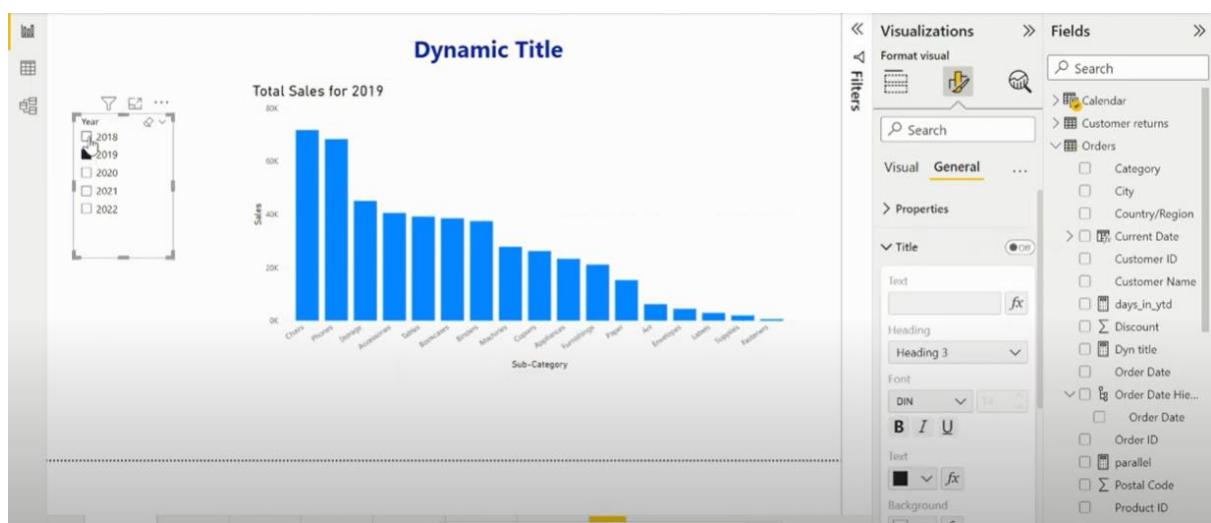
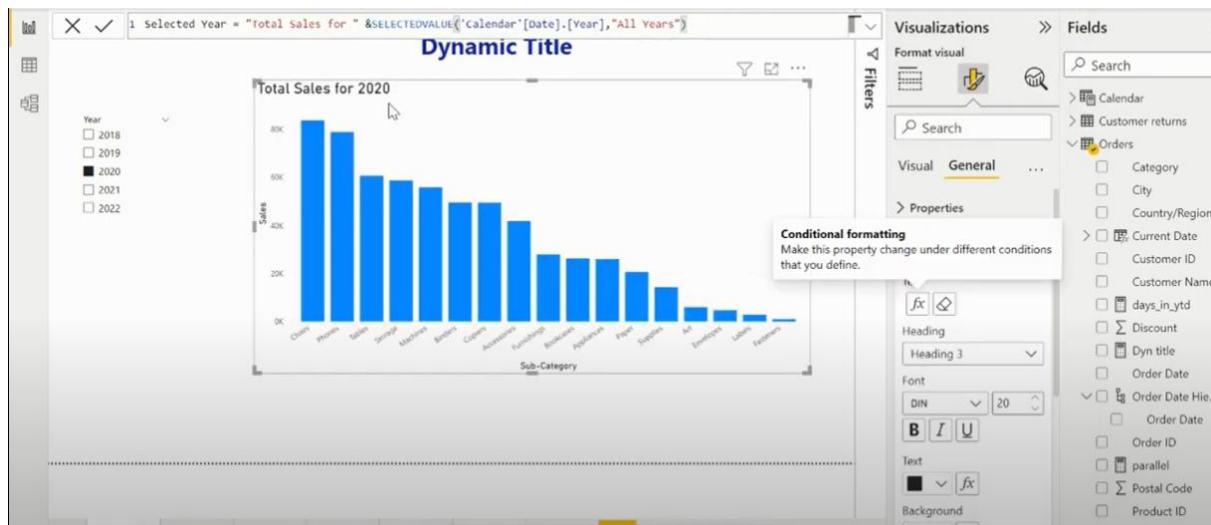
We want to update the visualization table, according to the slicer year selection.

Create a new measure



Go To Report Section ->format visual -> general -> title -> text -> conditional formatting -> format style (field value) ->sub-category (newly created measure)





Sum v/s SumX in Power BI

Sum : same as sum function in excel

SumX : same as sumproduct function in excel

SumX Syntax : (Table Name, value1 * value2)

The screenshot shows a Power BI dashboard with the following components:

- Top Left:** A large numerical value **97.26M** labeled **Total Product Cost**.
- Top Right:** A large numerical value **110.34M** labeled **TotalSaleAmount**.
- Table:** A table showing sales details:

Date	Prod	Qty	Unit Price
05-01-2022	A	1	1.5
06-01-2022	C	2	2
05-01-2022	A	4	2.25
04-02-2022	B	2	3.5
- Bottom Left:** A table showing totals:

Total Products	sum(qty)	9
Total Sale Amount	sumx(qty*unitprice)	21.5
- Right Panel:** The **Fields** pane is open, showing the data model. The **TotalSaleAmount** field is selected.

Waterfall Chart

Helps to visualize the change in value over a time period

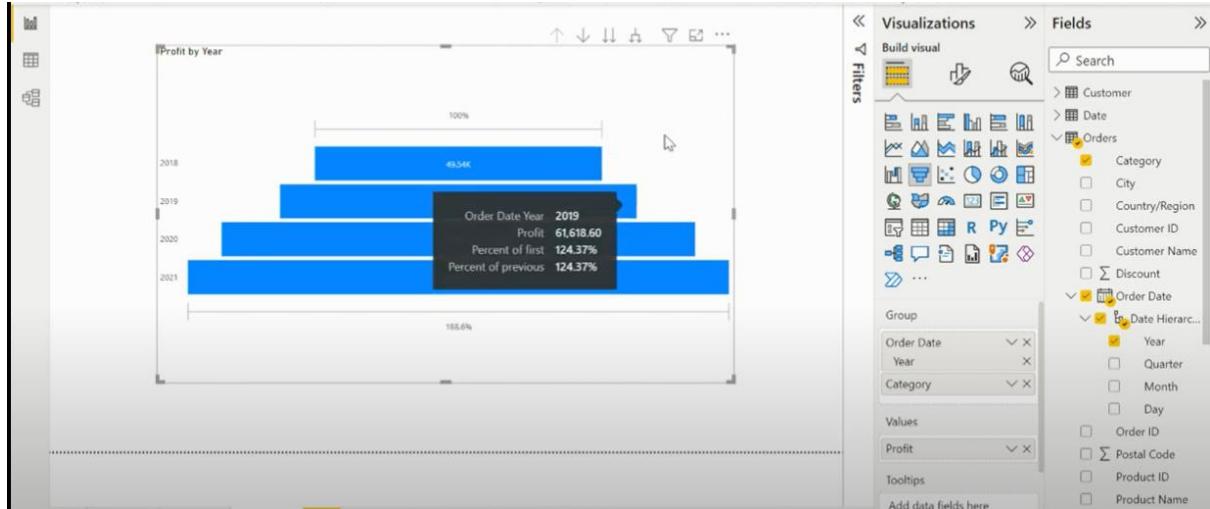
The screenshot shows a Waterfall chart titled "Profit by Year and Category". The chart displays profit values for different categories across two years (2018 and 2019). A tooltip is shown for the "Furniture" category, which shows the following data:

Category	Furniture	2018 Profit	5,457.73	2019 Profit	3,015.20	Profit change	-2,442.52 (-44.75%)
----------	-----------	-------------	----------	-------------	----------	---------------	---------------------

The chart uses green bars for increases and red bars for decreases. The **Fields** pane on the right shows the data model, with the **Order Date** and **Category** fields selected.

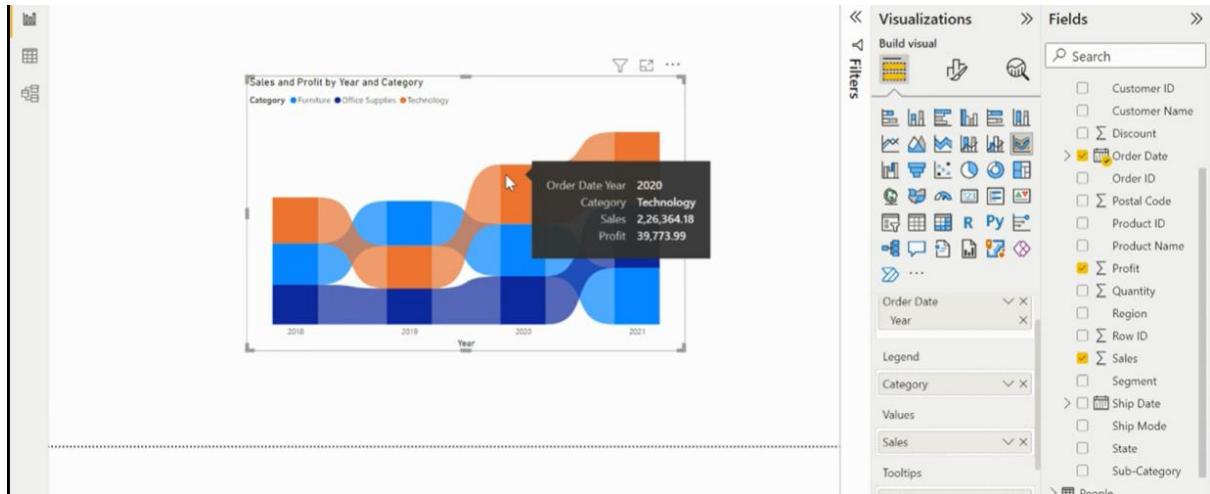
Funnel Chart

Helps to visualize the value over a time period



Ribbon Chart

Helps to visualize the ranks of the categories based on a value



Parallelperiod DAX function in Power BI:

Create a new measure

The screenshot shows the Power BI 'Measure tools' ribbon. A new measure named 'Measure' is being created. The formula bar shows '\$ % > CALCULATE(SUM(Orders[sales]),parallel)'. A tooltip for 'parallel' is displayed, showing its definition as 'Evaluates an expression in a context modified by filters.' Below the formula, a table displays monthly sales data from January to December 2021, with a column labeled 'parallel' showing identical values to the 'Sales' column. The Fields pane on the right lists various dimensions and measures, including 'Orders' and 'Measure'.

,-1, month will compare current month sales with last month sales.

For e.g. it will compare aug'22 sales with jul'22 sales

This screenshot shows the same Power BI environment as the previous one, but the formula has been updated to use the 'PARALLELPERIOD' function: '\$ % > CALCULATE(SUM(orders[sales]),PARALLELPERIOD('calendar'[Date],-1,month))'. The tooltip for 'MONTH' is visible. The table below shows the same monthly sales data as before. The Fields pane remains the same.

Sameperiod DAX function in Power BI:

Create a new measure

The screenshot shows the Power BI desktop interface with the 'Measure tools' tab selected. A new measure named 'Measure' is being created, based on the 'Orders' table. The formula is set to `CALCULATE(SUM(Orders[Sales]), SAMEPERIODLASTYEAR(Orders[Date]))`. A tooltip for the `SAMEPERIODLASTYEAR` function is displayed, explaining its purpose: "Returns a set of dates in the current selection from the previous year." The Fields pane on the right lists various columns from the 'Orders' table, including Date, Month, Sales, and Parallel Period LM.

sameperiod will compare current month sales with same month last year sales.

For e.g. it will compare aug'22 sales with aug'21 sales

This screenshot shows the continuation of measure creation. The formula has been updated to `CALCULATE(SUM(Orders[Sales]), SAMEPERIODLASTYEAR('Calendar'[Date]))`. The tooltip for `SAMEPERIODLASTYEAR` is still visible, stating "Returns a set of dates in the current selection from the previous year." The Fields pane on the right shows the 'Calendar' table, specifically the [Date] column, which includes detailed date hierarchy levels like [Year], [Quarter], [Month], and [Day].

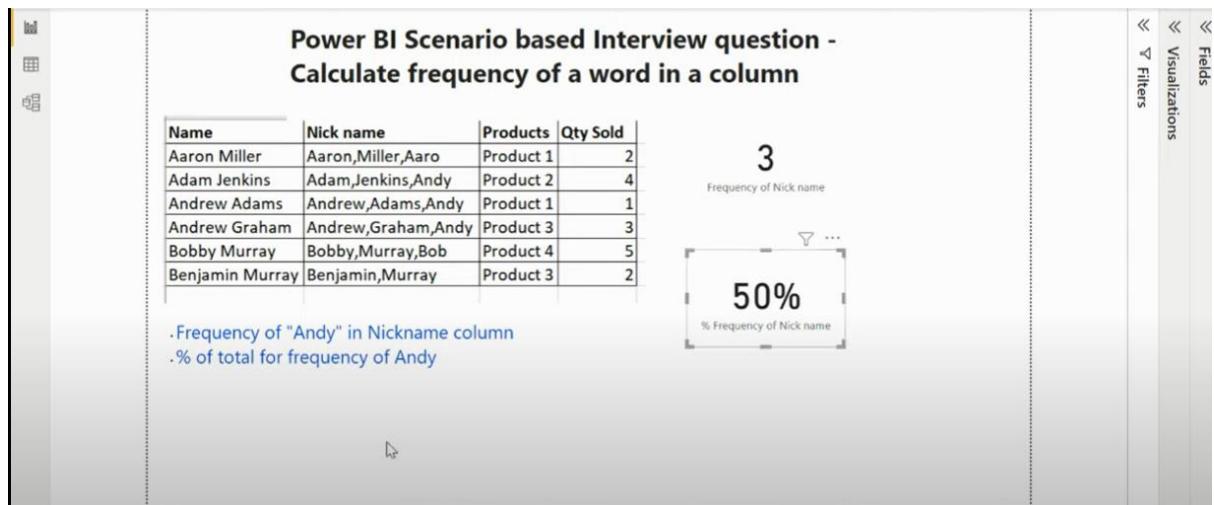
Parallelperiod (-12,month) will also give same result as same period but sameperiod is more reliable function in case of data for some months are missing

PARALLELPERIOD vs SAMEPERIODLASTYEAR in Power BI									
Year	Month	Sales	Parallel Period LM	Month	Sales	Parallel Period LY	Month	Sales	Same period LY
2018	January	18,542.49	74,919.52	January	18,542.49	4,70,532.51	December	96,999.04	74,919.52
2019	February	22,978.82	18,542.49	February	22,978.82	4,70,532.51	November	79,411.97	75,972.56
2020	March	51,715.88	22,978.82	March	51,715.88	4,70,532.51	September	73,410.02	64,595.92
2021	April	38,750.04	51,715.88	April	38,750.04	4,70,532.51	October	59,687.75	31,404.92
2022	May	56,987.73	38,750.04	May	56,987.73	4,70,532.51	May	56,987.73	30,131.69
Month	June	40,344.53	56,987.73	June	40,344.53	4,70,532.51	March	51,715.88	38,726.25
January	July	39,261.96	40,344.53	July	39,261.96	4,70,532.51	June	40,344.53	24,797.29
February	August	31,115.37	39,261.96	August	31,115.37	4,70,532.51	July	39,261.96	28,765.33
March	September	31,115.37	31,115.37	September	73,410.02	4,70,532.51	April	38,750.04	34,195.21
April	October	73,410.02	31,115.37	October	59,687.75	4,70,532.51	August	31,115.37	36,898.33
May	November	59,687.75	73,410.02	November	79,411.97	4,70,532.51	February	22,978.82	11,951.41
June	December	79,411.97	59,687.75	December	96,999.04	4,70,532.51	January	18,542.49	18,174.08
July	Total	96,999.04	79,411.97	Total	6,09,205.60	4,70,532.51	Total	6,09,205.60	4,70,532.51
August									
September									
October									
November									
December									

Parallelperiod Parallelperiod Sameperiod

-1,month -1,year

Calculate frequency of a word in a column in Power BI



Create a measure

File Home Insert Modeling View Help Table tools Measure tools

Name Measure Data category Uncategorized

CALCULATE(Expression, [Filter], ...)

Evaluates an expression in a context modified by filters.

New Quick measure measure Calculations

Calculate frequency o

Fields

Visualizations

Filters

Measure

1 Count of Andy = CALCULATE(COUNT(Sheet1[Nick name]),

2

Calculate frequency o

Name	Nick name	Products
Aaron Miller	Aaron,Miller,Aaro	Product 1
Adam Jenkins	Adam,Jenkins,Andy	Product 2
Andrew Adams	Andrew,Adams,Andy	Product 1
Andrew Graham	Andrew,Graham,Andy	Product 3
Bobby Murray	Bobby,Murray,Bob	Product 4
Benjamin Murray	Benjamin,Murray	Product 3
		5
		2

50%

% Frequency of Nick name

.Frequency of "Andy" in Nickname column
.% of total for frequency of Andy

File Home Insert Modeling View Help Table tools Measure tools

Name Measure Data category Uncategorized

SEARCH(FindText, WithinText, [StartPosition], [NotFoundValue])

Returns the starting position of one text string within another text string.
SEARCH is not case-sensitive.

Calculate frequency of a word in a column

Fields

Visualizations

Filters

Measure

1 Count of Andy = CALCULATE(COUNT(Sheet1[Nick name]),SEARCH("Andy",Sheet1[Nick name],1,0))

2

Calculate frequency of a word in a column

Name	Nick name	Products	Qty Sold
Aaron Miller	Aaron,Miller,Aaro	Product 1	2
Adam Jenkins	Adam,Jenkins,Andy	Product 2	4
Andrew Adams	Andrew,Adams,Andy	Product 1	1
Andrew Graham	Andrew,Graham,Andy	Product 3	3
Bobby Murray	Bobby,Murray,Bob	Product 4	5
Benjamin Murray	Benjamin,Murray	Product 3	2

3

Frequency of Nick name

50%

% Frequency of Nick name

.Frequency of "Andy" in Nickname column
.% of total for frequency of Andy

File Home Insert Modeling View Help Table tools Measure tools

Name Measure Data category Uncategorized

DIVIDE(numerator, denominator, [AlternateResult])

Safe Divide function with ability to handle divide by zero case.

Calculate frequency of a word in a column

Fields

Visualizations

Filters

Measure

1 % of Andy = DIVIDE((Count of Andy),COUNTROWS(Sheet1))

2

Calculate frequency of a word in a column

Name	Nick name	Products	Qty Sold
Aaron Miller	Aaron,Miller,Aaro	Product 1	2
Adam Jenkins	Adam,Jenkins,Andy	Product 2	4
Andrew Adams	Andrew,Adams,Andy	Product 1	1
Andrew Graham	Andrew,Graham,Andy	Product 3	3
Bobby Murray	Bobby,Murray,Bob	Product 4	5
Benjamin Murray	Benjamin,Murray	Product 3	2

3

Frequency of Nick name

50%

% Frequency of Nick name

.Frequency of "Andy" in Nickname column
.% of total for frequency of Andy

3

Count of Andy

POWER BI SCENARIO-BASED INTERVIEW QUESTIONS

Calculate frequency of a word in a column

Table:

Name	Nick name	Products	Qty Sold
Aaron Miller	Aaron,Miller,Aaro	Product 1	2
Adam Jenkins	Adam,Jenkins,Andy	Product 2	4
Andrew Adams	Andrew,Adams,Andy	Product 1	1
Andrew Graham	Andrew,Graham,Andy	Product 3	3
Bobby Murray	Bobby,Murray,Bob	Product 4	5
Benjamin Murray	Benjamin,Murray	Product 3	2

Measure: % of Andy = DIVIDE([Count of Andy],COUNTRWS(Sheet1))

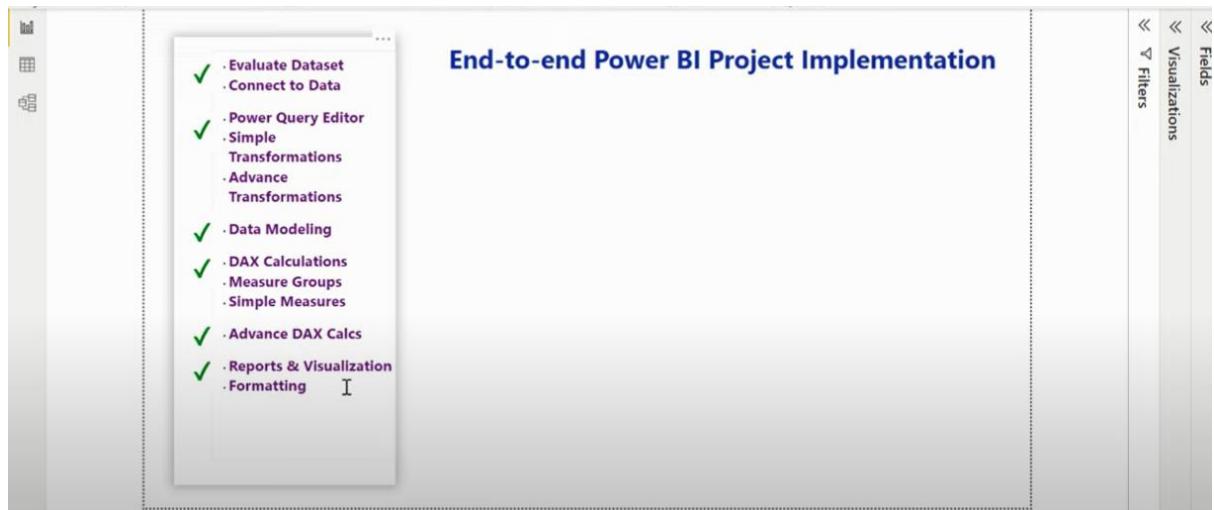
Visualizations:

- Count of Andy: 3
- % Frequency of Nick name: 50%
- Frequency of Nick name: 3

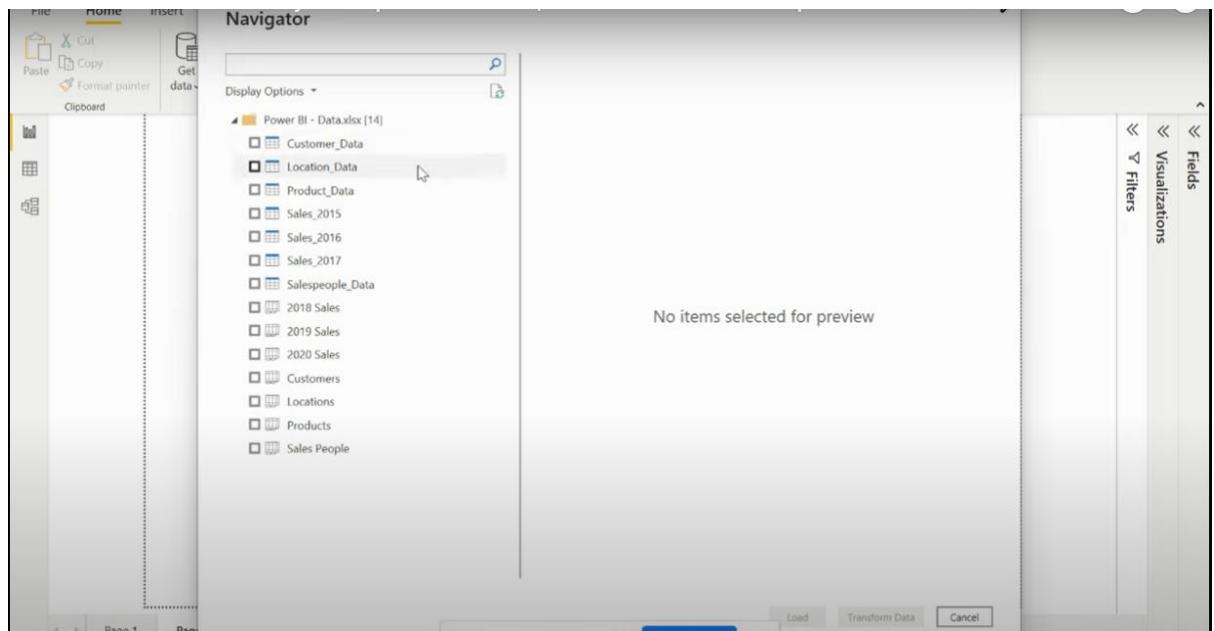
Fields:

- Values: % Frequency of ... (selected), % of Andy, Count of Andy, Frequency of Nic..., Name, Nick name, Products, Sum Qty Sold
- Drill through: Off
- Cross-report: On
- Keep all filters: On
- Add drill-through fields here

End to End Power BI Project Implementation



Get Data -> Excel Workbook -> go to location -> select all the tables to load -> transform data



Salespeople_Data

Salesperson ID	Salesperson Name
EMP1000	Fred Robertson
EMP1001	Kevin Butler
EMP1002	Andrew Bowman
EMP1003	Christopher Tucker
EMP1004	Kenneth Bradley
EMP1005	Ryan Welch
EMP1006	Sean Miller
EMP1007	Jeremy Mendoza
EMP1008	Carl Elliott
EMP1009	Kenneth Fields
EMP1010	Joshua Cook
EMP1011	Larry Marshall
EMP1012	John Reyes
EMP1013	Charles Harper
EMP1014	Ronald Reed
EMP1015	Justin Lynch
EMP1016	Henry Nelson
EMP1017	Brian Hansen
EMP1018	Jimmy Young
EMP1019	Roger Robertson
EMP1020	Ryan Butler
EMP1021	Clarence Fox
EMP1022	Walter Cook
EMP1023	Brian Davis

Rename table names if required

Properties

Name: Sales_2017

Applied Steps

Changed Type

Rename columns if required

The screenshot shows the Power BI Data Editor interface. A table named 'Products' is displayed with columns: Product ID, Product Name, Product Cost, Original Sale Price, and Discount. The 'Product Cost' column is currently selected. The ribbon bar at the top has several tabs: Close & Apply, New Source, Recent Sources, Enter Data, Data Sources, Manage Parameters, Refresh Preview, Advanced Editor, Choose Columns, Remove Columns, Manage Columns, Keep Rows, Remove Rows, Sort, Split Column, Group By, Data Type: Whole Number, Use First Row as Headers, Merge Queries, Text Analytics, Append Queries, Vision, Combine Files, Azure Machine Learning, Combine, AI Insights, and Query Settings. The 'Properties' pane on the right shows the table is named 'Products'. The 'Applied Steps' pane shows a step named 'Changed Type'.

To remove unrequired columns

Select the unrequired columns with **ctrl + select** -> right click -> remove columns

The screenshot shows the Power BI Data Editor interface. A table is displayed with columns: Households, Median Income, Land Area, and Water Area. The 'Water Area' column is selected, and a context menu is open, showing options such as Copy, Remove Columns, Remove Other Columns, Add Column From Examples, Remove Duplicates, Remove Errors, Replace Values, Change Type, Transform, Merge Columns, Sum, Product, Group By, Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, and Move. The ribbon bar at the top has various transformation tools. The 'Properties' pane on the right shows the table is named 'Locations'. The 'Applied Steps' pane shows a step named 'Changed Type'.

To merge tables with similar columns

Home -> append queries -> three or more tables -> select table -> click add -> click ok (once all tables are added)

Once table appended , right click on unrequired tables and deselect Enable Load

To Group the columns

Select the tables -> right click -> move to group -> new group -> rename group -> click ok -> two groups are created (1. Created by us 2. other queries)

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [7]' pane lists several tables: Customers, Locations, Products, Sales_2018, Sales_2019, Sales_2020, and Salespeople. The Salespeople table is currently selected. In the main area, a table preview for the Salespeople query is displayed, showing columns 'Salesperson ID' and 'Salesperson Name'. A context menu is open over the first few rows of the table, with the 'Move To Group' option highlighted. Below it, there is a 'New Group...' option.

This screenshot shows the 'New Group' dialog box from the previous step. It has 'Name' set to 'Data model' and an empty 'Description' field. The 'OK' button is highlighted with a yellow box.

The screenshot shows the 'Queries [7]' pane again. The 'Data model' group now contains the Salespeople query, indicating it has been moved. The 'Other Queries' group contains the Sales_2019 and Sales_2020 queries.

To Duplicate Columns

Select column -> right click -> duplicate column -> it will be duplicated at the end -> drag and drop to wherever you want

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [7]' pane lists several queries including 'Customers', 'Locations', 'Products', and various 'Sales' queries. The main area displays a table with columns: 'State Code', 'State', 'Latitude', and 'Longitude'. A context menu is open over the 'State' column, with 'Duplicate Column' highlighted. Other options in the menu include 'Copy', 'Remove', 'Remove Other Columns', 'Add Column From Examples...', 'Remove Duplicates', 'Remove Errors', 'Change Type', 'Transform', 'Replace Values...', 'Replace Errors...', 'Split Column', 'Group By...', 'Fill', 'Unpivot Columns', 'Unpivot Other Columns', 'Unpivot Only Selected Columns', 'Rename...', 'Move', 'Drill Down', and 'Add as New Query'. To the right, the 'Query Settings' pane shows the query name 'Locations' and the applied step 'Changed Type'.

This screenshot shows the same Power BI Data Editor interface after the 'State' column has been duplicated. The table now includes two 'State' columns: 'State' and 'State - Copy'. The 'State - Copy' column contains identical data to the original 'State' column. The context menu from the previous screenshot is no longer visible. The 'Query Settings' pane remains the same, showing 'Locations' and 'Changed Type'.

This screenshot shows the Power BI Data Editor after the 'State - Copy' column has been moved to the second position in the table. The columns are now ordered: 'State', 'State - Copy', 'Latitude', 'Longitude', and 'Population'. The 'Query Settings' pane still displays 'Locations' and 'Changed Type'.

To format the column (lowercase/uppercase/proper/trim/clean/add prefix/add suffix)

Transform -> format -> uppercase

The screenshot shows the Power BI Editor interface with the 'Format' context menu open over a 'State' column. The menu includes options for text transformation: lowercase, uppercase, Capitalize Each Word, Trim, Clean, Add Prefix, and Add Suffix. The 'uppercase' option is selected. The 'APPLIED STEPS' pane on the right lists the step 'Reordered Columns'.

To split the column (by delimiter/no of characters/positions/lowercase to uppercase/uppercase to lowercase/digit to non-digit/ non-digit to digit)

Transform -> split column -> by no of characters -> specify no of characters -> select appropriate option -> click ok

The screenshot shows the Power BI Editor interface with the 'Split Column' context menu open over a 'State' column. The menu includes options for splitting columns: By Delimiter, By Number of Characters, By Position, By Lower, By Upper, By Digit to Non-Digit, and By Non-Digit to Digit. The 'By Upper' option is selected. The 'APPLIED STEPS' pane on the right lists the step 'Uppercased Text'.

Queries [7]

- Customers
- Locations
- Products
- Sales_2018
- Sales_2019
- Sales_2020
- Salespeople

Split Column by Number of Characters

Specify the number of characters used to split the text column.

Number of characters: 4

Split:

- Once, as far left as possible
- Once, as far right as possible
- Repeatedly

Advanced options

OK Cancel

	A ^U C State Code	A ^U C State	A ^U C State - Copy	A ^U C Type	1.2 Latitude
1	CA	California	CALIFORNIA	City	
2	CA	California	CALIFORNIA	City	
3	CA	California	CALIFORNIA	City	
4	CA	California	CALIFORNIA	City	
5	CA	California	CALIFORNIA	City	
6	CA	California	CALIFORNIA	City	
7					
8					
9	Y				
10					
11					
12					
13					
14					
15					
16					
17	CA	California	CALIFORNIA	City	
18	CA	California	CALIFORNIA	City	
19	CA	California	CALIFORNIA	City	
20	inty	California	CALIFORNIA	City	
21	CA	California	CALIFORNIA	City	
22	CA	California	CALIFORNIA	City	
23	CA	California	CALIFORNIA	City	
24	CA	California	CALIFORNIA	City	

Queries [7]

- Customers
- Locations
- Products
- Sales_2018
- Sales_2019
- Sales_2020
- Salespeople

Table.TransformColumnTypes(#"Split Column by Position", [{"State - Copy.1": type text}, {"state - Copy.2": type text}])

	A ^U C State Code	A ^U C State	A ^U C State - Copy.1	A ^U C State - Copy.2	A ^U C Type
1	CA	California	CALI	FORNIA	City
2	CA	California	CALI	FORNIA	City
3	CA	California	CALI	FORNIA	City
4	CA	California	CALI	FORNIA	City
5	CA	California	CALI	FORNIA	City
6	CA	California	CALI	FORNIA	City
7	CA	California	CALI	FORNIA	City
8	CA	California	CALI	FORNIA	City
9	Y	California	CALI	FORNIA	City
10	CA	California	CALI	FORNIA	City
11	CA	California	CALI	FORNIA	City
12	CA	California	CALI	FORNIA	City
13	CA	California	CALI	FORNIA	City
14	CA	California	CALI	FORNIA	City
15	CA	California	CALI	FORNIA	City
16	CA	California	CALI	FORNIA	City
17	CA	California	CALI	FORNIA	City
18	CA	California	CALI	FORNIA	City
19					

How to bring in data from web in table

We want the California population data by city in the table

Home -> New source -> web -> paste the url -> click ok -> In the navigator we can see options such as table/html code/displayed text -> select table -> click ok -> new table is added to the dataset

Rename the new table -> right click -> move to required group

End to End Power BI Project Implementation | From Novice to an Expert in Power BI ⚡

The screenshot shows the Power BI Editor interface. On the left, the 'Queries [7]' pane lists various data sources and queries, with 'Locations' selected. In the center, a query editor displays a table with columns: 'A#C State', 'A#C State - Copy.1', 'A#C State - Copy.2', 'A#C Type', and '1.2 Lat'. The 'A#C State' column contains values like California, CALI, FORNIA, and CDP. The 'A#C State - Copy.1' and 'A#C State - Copy.2' columns show intermediate state representations. The 'A#C Type' column contains City or CDP. The '1.2 Lat' column is empty. The 'APPLIED STEPS' pane on the right shows a step named 'Changed Type1' under the 'Changed Type' section.

This screenshot is similar to the one above, showing the Power BI Editor with the 'Locations' query selected. A 'From Web' dialog box is open over the query editor, prompting for a URL. The URL entered is 'https://www.california-demographics.com/cities_by_population'. The 'APPLIED STEPS' pane shows the 'Changed Type1' step.

This screenshot shows the Power BI Editor with the 'Locations' query selected. A 'Table View' window is open, displaying a table titled 'Table 1' with columns 'Rank', 'City', and 'Population'. The table lists various California cities with their population counts. The 'APPLIED STEPS' pane shows the 'Changed Type1' step.

Rank	City	Population
8	Oakland	422575
9	Bakersfield	379879
10	Anaheim	353085
11	Santa Ana	332610
12	Riverside	327569
13	Stockton	311103
14	Irvine	272694
15	Chula Vista	268779
16	Fremont	234829
17	San Bernardino	216784
18	Modesto	214485
19	Fontana	213704
20	Santa Clarita	212519
21	Moreno Valley	208751
22	Oxnard	207722
23	Huntington Beach	199778
24	Glendale	199357
25	Santa Rosa	178391
26	Ontario	178194
27	Rancho Cucamonga	178060
28	Oceanside	175694
29	El Cajon	172320

Queries [8]

Data model [5]

- Customers
- Locations
- Products
- Sales_2018
- Salespeople

Other Queries [3]

- Sales_2019
- Sales_2020
- Table 1**

Table 1

Rank | City | Population

Rank	City	Population
1	Los Angeles	3973278
2	San Diego	1414545
3	San Jose	1029409
4	San Francisco	874784
5	Fresno	526147
6	Sacramento	503482
7	Long Beach	462081
8	Oakland	422575
9	Bakersfield	379879
10	Anaheim	353085
11	Santa Ana	332610
12	Riverside	327569
13	Stockton	311103
14	Irvine	272694
15	Chula Vista	268779
16	Fremont	234829
17	San Bernardino	216784
18	Modesto	214485
19	Fontana	212704
20	Santa Clarita	212519
21	Moreno Valley	208751
22	Oxnard	207722
23	Huntington Beach	199778
24	Glendale	199357
25	Santa Rosa	178391

Query Settings

PROPERTIES

- Name: Table 1
- All Properties

APPLIED STEPS

- Source
- Extracted Table From Html
- Promoted Headers
- Changed Type**

Queries [8]

Data model [5]

- Customers
- Locations
- Products
- Sales_2018
- Salespeople

Other Queries [3]

- Sales_2019
- Sales_2020
- CA Population**

CA Population

Rank | City | Population

Rank	City	Population
1	Los Angeles	3973278
2	San Diego	1414545
3	San Jose	1029409
4	San Francisco	874784
5	Fresno	526147
6	Sacramento	503482
7	Long Beach	462081
8	Oakland	422575
9	Bakersfield	379879
10	Anaheim	353085
11	Santa Ana	332610
12	Riverside	327569
13	Stockton	311103
14	Irvine	272694
15	Chula Vista	268779
16	Fremont	234829
17	San Bernardino	216784
18	Modesto	214485
19	Fontana	212704
20	Santa Clarita	212519
21	Moreno Valley	208751
22	Oxnard	207722
23	Huntington Beach	199778
24	Glendale	199357
25	Santa Rosa	178391

Query Settings

PROPERTIES

- Name: CA Population
- All Properties

APPLIED STEPS

- Source
- Extracted Table From Html
- Promoted Headers
- Changed Type**

Contextual menu open on CA Population:

- Copy
- Paste
- Delete
- Rename
- Enable load**
- Include in report refresh**
- Duplicate
- Reference
- Move To Group
- Move Up
- Move Down
- Create Function...
- Convert To Parameter
- Advanced Editor
- Properties...

To combine different tables

Select Primary table -> Home -> merge queries -> select other table ->select common columns -> specify join condition -> column is merged into primary table (but it is in the form of a table) -> click expand -> select required columns -> click ok -> rename the column -> select the unrequired table -> right click -> unselect enable load -> right click -> move to group -> other queries

End to End Power BI Project Implementation | From Novice to an Expert in Power BI ⚡

The screenshot shows the Power BI desktop interface with the following details:

- Queries [8]**: A list of queries including "Data model [6]" (Customers, Locations, Products, Sales_2018, Salespeople, CA Population) and "Other Queries [2]".
- Transform ribbon**: Includes Close & Apply, New Source, Refresh, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Properties, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Split Column, Group By, Sort, Data Type Text, Use First Row as Headers, Replace Values, and Transform.
- Table view**: Shows a table with columns: Location ID, Name, County, State Code, and State. The table has 21 rows of California city data.
- Properties pane**: Shows the "Name" field set to "Locations".
- Applied Steps pane**: Displays the steps taken: Source, Navigation, Changed Type (highlighted), Removed Columns, Duplicated Column, Reordered Columns, Uppercased Text, Split Column by Position, and a note about "Changed Type1".

Merge

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

Locations

Location ID	Name	County	State Code	State	State - Copy.1	State - Copy.2	Type
A100	Anaheim	Orange County	CA	California	CALI	FORNIA	City
A101	Antioch	Contra Costa County	CA	California	CALI	FORNIA	City
A102	Bakersfield	Kern County	CA	California	CALI	FORNIA	City
A103	Berkeley	Alameda County	CA	California	CALI	FORNIA	City
A104	Burbank	Los Angeles County	CA	California	CALI	FORNIA	City
A105	Carlsbad	San Diego County	CA	California	CALI	FORNIA	City
A106	Chula Vista	San Diego County	CA	California	CALI	FORNIA	City
A107	Clovis	Fresno County	CA	California	CALI	FORNIA	City
A108	Concord	Contra Costa County	CA	California	CALI	FORNIA	City
A109	Corona	Riverside County	CA	California	CALI	FORNIA	City
A110	Costa Mesa	Orange County	CA	California	CALI	FORNIA	City
A111	Daly City	San Mateo County	CA	California	CALI	FORNIA	City
A112	Downey	Los Angeles County	CA	California	CALI	FORNIA	City
A113	East Los Angeles	Los Angeles County	CA	California	CALI	FORNIA	City
A114	El Cajon	San Diego County	CA	California	CALI	FORNIA	City
A115	Elk Grove	Sacramento County	CA	California	CALI	FORNIA	City
A116	El Monte	Los Angeles County	CA	California	CALI	FORNIA	City
A117	Eskcondido	San Diego County	CA	California	CALI	FORNIA	City
A118	Fairfield	Solano County	CA	California	CALI	FORNIA	City
A119	Fontana	San Bernardino County	CA	California	CALI	FORNIA	City
A120	Fremont	Alameda County	CA	California	CALI	FORNIA	City

CA Population

Rank	City	Population
1	Los Angeles	3973278
2	San Diego	1414545
3	San Jose	1029409
4	San Francisco	874784
5	Fresno	526347

Join Kind: Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

Fuzzy matching options

OK Cancel

Merge

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

Locations

Location ID	Name	County	State Code	State	State - Copy.1	State - Copy.2	Type
A100	Anaheim	Orange County	CA	California	CALI	FORNIA	City
A101	Antioch	Contra Costa County	CA	California	CALI	FORNIA	City
A102	Bakersfield	Kern County	CA	California	CALI	FORNIA	City
A103	Berkeley	Alameda County	CA	California	CALI	FORNIA	City
A104	Burbank	Los Angeles County	CA	California	CALI	FORNIA	City
A105	Carlsbad	San Diego County	CA	California	CALI	FORNIA	City
A106	Chula Vista	San Diego County	CA	California	CALI	FORNIA	City
A107	Clovis	Fresno County	CA	California	CALI	FORNIA	City
A108	Concord	Contra Costa County	CA	California	CALI	FORNIA	City
A109	Corona	Riverside County	CA	California	CALI	FORNIA	City
A110	Costa Mesa	Orange County	CA	California	CALI	FORNIA	City
A111	Daly City	San Mateo County	CA	California	CALI	FORNIA	City
A112	Downey	Los Angeles County	CA	California	CALI	FORNIA	City
A113	East Los Angeles	Los Angeles County	CA	California	CALI	FORNIA	City
A114	El Cajon	San Diego County	CA	California	CALI	FORNIA	City
A115	Elk Grove	Sacramento County	CA	California	CALI	FORNIA	City
A116	El Monte	Los Angeles County	CA	California	CALI	FORNIA	City
A117	Eskcondido	San Diego County	CA	California	CALI	FORNIA	City
A118	Fairfield	Solano County	CA	California	CALI	FORNIA	City
A119	Fontana	San Bernardino County	CA	California	CALI	FORNIA	City
A120	Fremont	Alameda County	CA	California	CALI	FORNIA	City

CA Population

Rank	City	Population
1	Los Angeles	3973278
2	San Diego	1414545
3	San Jose	1029409
4	San Francisco	874784
5	Fresno	526347

Join Kind: Left Outer (all from first, matching from second)

Use fuzzy matching to perform the merge

Fuzzy matching options

Estimating matches based on data previews

OK Cancel

Queries [8]

DATA MODEL

- Data model [6]
 - Customers
 - Locations
 - Products
 - Sales_2018
 - Salespeople
 - CA Population
- Other Queries [2]

Table

	Latitude	Longitude	Population	Time Zone	CA Population
1	33.83529	-117.9145	350742	America/Los Angeles	Table
2	38.00492	-121.80579	110542	America/Los Angeles	Table
3	35.37329	-119.01871	373640	America/Los Angeles	Table
4	37.87159	-122.27275	120972	America/Los Angeles	Table
5	34.18084	-118.30897	105313	America/Los Angeles	Table
6	33.15809	-117.35059	113453	America/Los Angeles	Table
7	32.64	-117.08417	265757	America/Los Angeles	Table
8	36.82523	-119.70292	104180	America/Los Angeles	Table
9	37.97798	-122.03107	128867	America/Los Angeles	Table
10	33.87529	-117.56644	164226	America/Los Angeles	Table
11	33.64113	-117.91867	113204	America/Los Angeles	Table
12	37.70583	-122.46194	106562	America/Los Angeles	Table
13	33.94001	-118.13257	114219	America/Los Angeles	Table
14	34.0239	-118.17202	127610	America/Los Angeles	Table
15	32.79477	-116.96253	103679	America/Los Angeles	Table
16	38.4088	-121.37162	166913	America/Los Angeles	Table
17	34.06862	-118.02757	116732	America/Los Angeles	Table
18	33.11921	-117.08642	151451	America/Los Angeles	Table
19	38.24936	-122.03997	112970	America/Los Angeles	Table
20	34.09223	-117.43505	207460	America/Los Angeles	Table
21	37.54827	-121.98857	232206	America/Los Angeles	Table
22	36.74773	-119.77237	520052	America/Los Angeles	Table
23	33.87029	-117.92534	140847	America/Los Angeles	Table
24	33.77391	-117.94145	175393	America/Los Angeles	Table

Queries [8]

DATA MODEL

- Data model [6]
 - Customers
 - Locations
 - Products
 - Sales_2018
 - Salespeople
 - CA Population
- Other Queries [2]

Table

Expand Aggregate

(Select All Columns)

Rank

City

Population

Use original column name as prefix

OK Cancel

Queries [8]

DATA MODEL

- Data model [6]
 - Customers
 - Locations
 - Products
 - Sales_2018
 - Salespeople
 - CA Population
- Other Queries [2]

Table

Search Columns to Expand

Expand Aggregate

(Select All Columns)

Rank

City

Population

Use original column name as prefix

OK Cancel

Queries [8]

Data model [6]

- Customers
- Locations
- Products
- Sales_2018
- Salespeople
- CA Population
- Other Queries [2]

= Table.ExpandTableColumn(#"Merged Queries", "CA Population", {"Population"}, {"CA Population.Population"})

de	Longitude	Population	Time Zone	CA Population.Population
1	33.83529	-117.9145	350742 America/Los Angeles	353085
2	34.05223	-118.24368	3971883 America/Los Angeles	3973278
3	38.00492	-121.80579	110542 America/Los Angeles	111468
4	32.71533	-117.15726	1394928 America/Los Angeles	1414545
5	35.37329	-119.01871	373640 America/Los Angeles	379879
6	37.33939	-121.89496	1026908 America/Los Angeles	1029409
7	37.87159	-122.27275	120972 America/Los Angeles	123065
8	37.775	-122.41944	864816 America/Los Angeles	874784
9	36.74773	-119.77237	520052 America/Los Angeles	526147
10	33.15809	-117.35059	113453 America/Los Angeles	114411
11	38.58157	-121.4944	490712 America/Los Angeles	503482
12	32.64	-117.08417	265757 America/Los Angeles	268779
13	33.76696	-118.18923	474140 America/Los Angeles	462081
14	36.82523	-119.70292	104180 America/Los Angeles	112663
15	37.80437	-122.2708	419267 America/Los Angeles	422575
16	37.97798	-122.03107	128667 America/Los Angeles	129227
17	33.87529	-117.56644	164226 America/Los Angeles	168112
18	33.64113	-117.91867	113204 America/Los Angeles	112958
19	33.74557	-117.86783	335400 America/Los Angeles	332610
20	37.70583	-122.46194	106562 America/Los Angeles	107197
21	33.95333	-117.39611	322424 America/Los Angeles	327569
22	33.94001	-118.13257	114219 America/Los Angeles	111263
23	37.9577	-121.29078	305658 America/Los Angeles	311103
24	34.0239	-118.17202	127610 America/Los Angeles	120504
25				

Query Settings

Properties

- Name: Locations
- All Properties

Applied Steps

- Source
- Navigation
- Changed Type
- Removed Columns
- Duplicated Column
- Reordered Columns
- Uppercased Text
- Split Column by Position
- Changed Type1
- Merged Queries
- Expanded CA Population

Queries [8]

Data model [6]

- Customers
- Locations
- Products
- Sales_2018
- Salespeople
- CA Population
- Other Queries [2]

= Table.RenameColumns(#"Expanded CA Population", {"CA Population.Population", "CA Population"})

Latitude	Longitude	Population	Time Zone	CA Population
1	33.83529	-117.9145	350742 America/Los Angeles	353085
2	34.05223	-118.24368	3971883 America/Los Angeles	3973278
3	38.00492	-121.80579	110542 America/Los Angeles	111468
4	32.71533	-117.15726	1394928 America/Los Angeles	1414545
5	35.37329	-119.01871	373640 America/Los Angeles	379879
6	37.33939	-121.89496	1026908 America/Los Angeles	1029409
7	37.87159	-122.27275	120972 America/Los Angeles	123065
8	37.775	-122.41944	864816 America/Los Angeles	874784
9	36.74773	-119.77237	520052 America/Los Angeles	526147
10	33.15809	-117.35059	113453 America/Los Angeles	114411
11	38.58157	-121.4944	490712 America/Los Angeles	503482
12	32.64	-117.08417	265757 America/Los Angeles	268779
13	33.76696	-118.18923	474140 America/Los Angeles	462081
14	36.82523	-119.70292	104180 America/Los Angeles	112663
15	37.80437	-122.2708	419267 America/Los Angeles	422575
16	37.97798	-122.03107	128667 America/Los Angeles	129227
17	33.87529	-117.56644	164226 America/Los Angeles	168112
18	33.64113	-117.91867	113204 America/Los Angeles	112958
19	33.74557	-117.86783	335400 America/Los Angeles	332610
20	37.70583	-122.46194	106562 America/Los Angeles	107197
21	33.95333	-117.39611	322424 America/Los Angeles	327569
22	33.94001	-118.13257	114219 America/Los Angeles	111263
23	37.9577	-121.29078	305658 America/Los Angeles	311103
24	34.0239	-118.17202	127610 America/Los Angeles	120504
25				

Query Settings

Properties

- Name: Locations
- All Properties

Applied Steps

- Source
- Navigation
- Changed Type
- Removed Columns
- Duplicated Column
- Reordered Columns
- Uppercased Text
- Split Column by Position
- Changed Type1
- Merged Queries
- Expanded CA Population
- Renamed Columns

Queries [8]

Data model [6]

- Customers
- Locations
- Products
- Sales_2018
- Salespeople
- CA Population
- Other Queries [2]

= Table.TransformColumnTypes(#"Promoted Headers", {"Rank", Int64.Type}, {"City", type text}, {"Population", type number})

Rank	City	Population
1	Los Angeles	3973278
2	San Diego	1414545
3	San Jose	1029409
4	San Francisco	874784
5	Fresno	526147
6	Sacramento	503482
7	Long Beach	462081
8	Oakland	422575
9	Bakersfield	379879
10	Anaheim	353085
11	Santa Ana	332610
12	Riverside	327569
13	Stockton	311103
14	Irvine	272694
15	Chula Vista	268779
16	Redding	234829
17	San Jose	216784
18	San Leandro	214485
19	Fontana	212704
20	Santa Clarita	212519
21	Moreno Valley	208751
22	Oxnard	207722
23	Huntington Beach	199778
24	Glendale	199357
25	Santa Rosa	198301

Query Settings

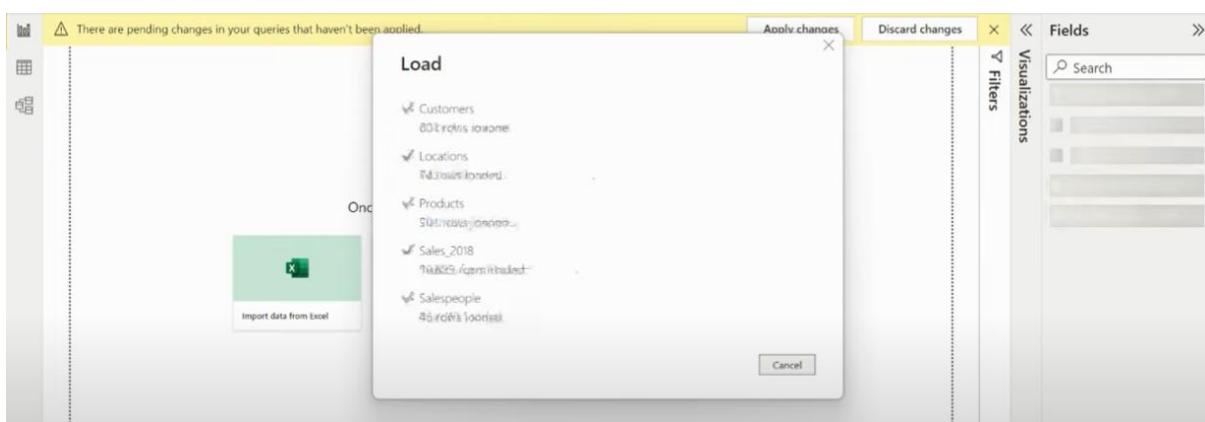
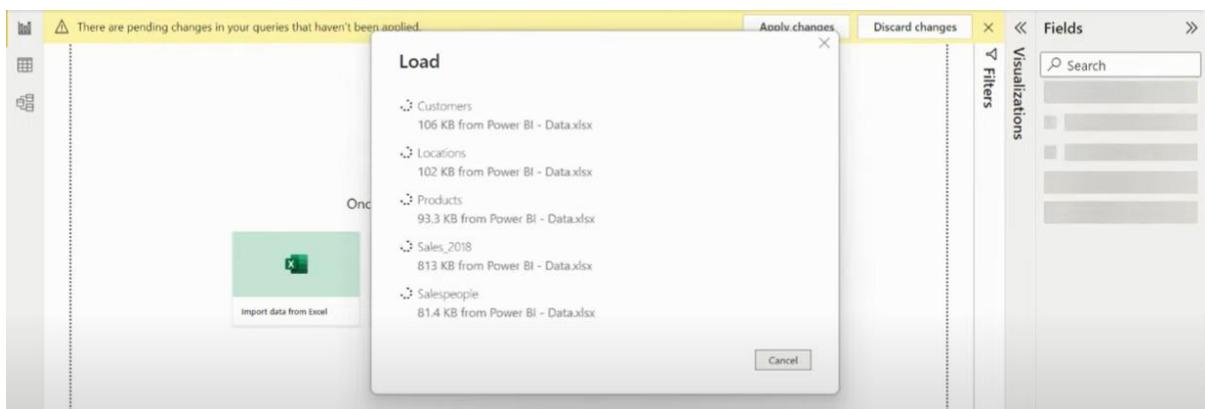
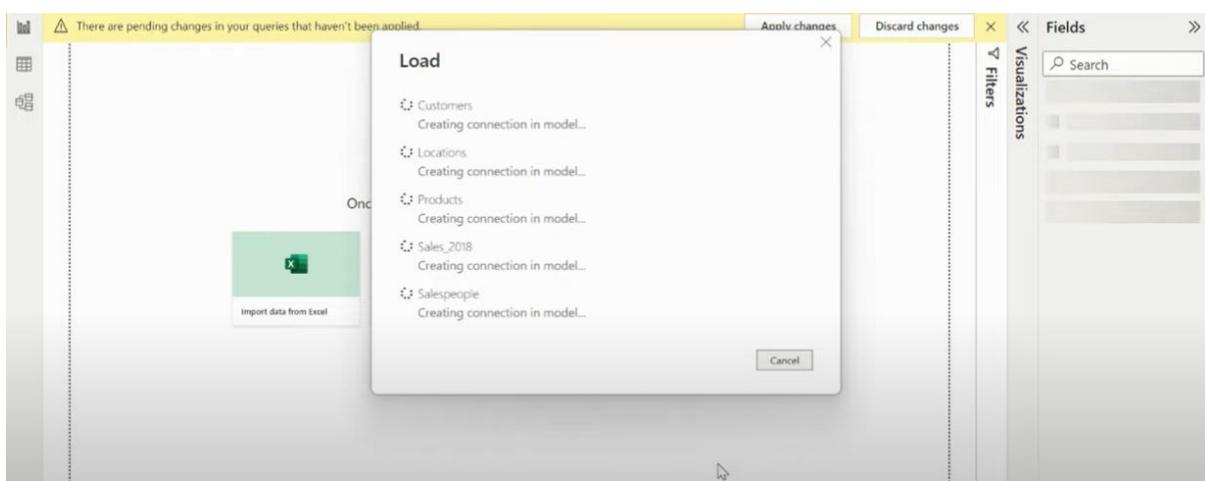
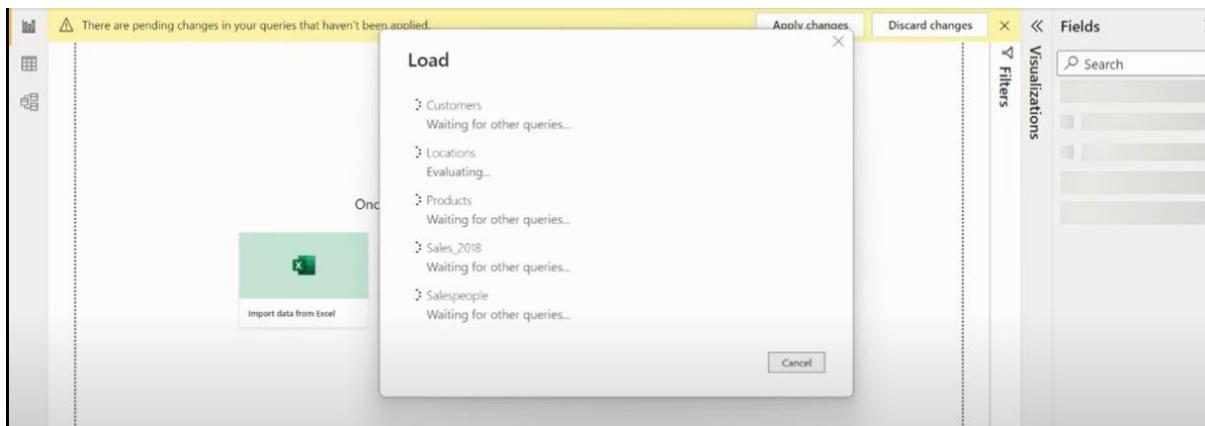
Properties

- Name: CA Population
- All Properties

Applied Steps

- Source
- Extracted Table From Html
- Promoted Headers
- Changed Type

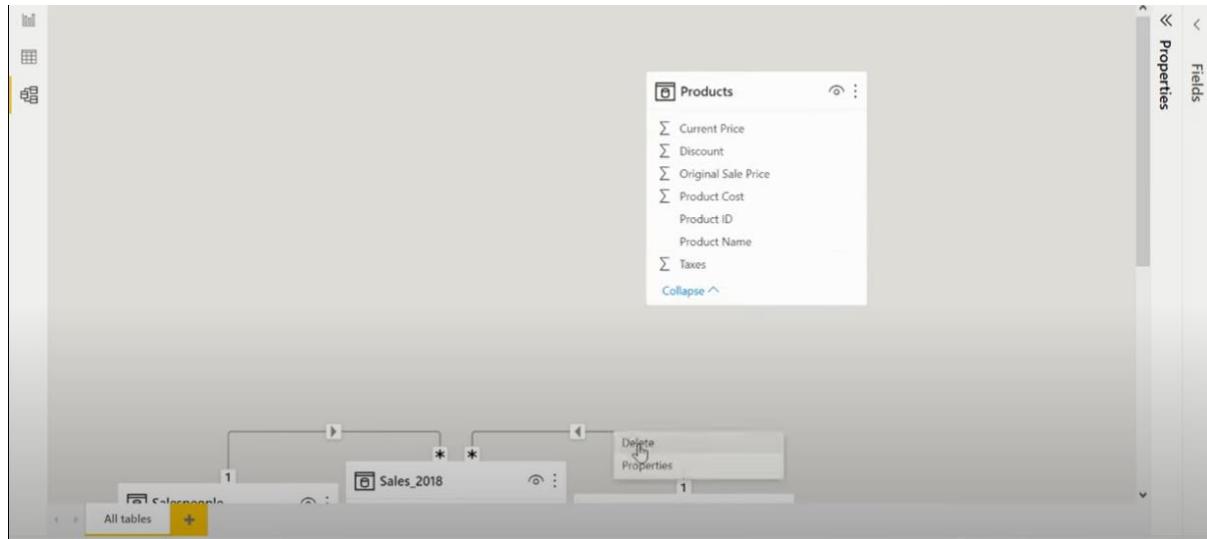
Once transformations are done, click on Close & apply then data will be loaded into Power BI



Data Modelling

Power BI will create the relations between the tables automatically.

Select the relation line -> right click -> delete



Use separate time-based table to use in time intelligence functions

Go to Data section -> table tools -> new table -> rename as dates -> use **calenderauto()** function to extract dates as new table -< new date table will be created

The screenshot shows the Power BI Data Model view. A new table named 'Customers' is being created. The 'Structure' tab is selected, showing a table with columns 'Customer ID' and 'Customer Name'. The data in the table is as follows:

Customer ID	Customer Name
C1000	Jesse Evans
C1001	Victor Ramos
C1002	Mark Montgomery
C1003	Dennis Morris
C1004	Gregory Simmons
C1005	Jeremy Vasquez
C1006	Anthony Simpson
C1007	Ernest Rivera
C1008	Victor Martinez
C1009	Bobby Burton
C1010	Bruce Porter
C1011	Nicholas Simmons
C1012	Bruce Butler
C1013	Raymond Alexander
C1014	Jason Duncan
C1015	Philip Peters
C1016	James Castillo
C1017	Benjamin Kim
C1018	Shawn Long
C1019	Steve Diaz

The ribbon at the top shows 'Table Tools' and 'Calculated Columns'. On the right, a 'Fields' pane lists other tables: Customers, Locations, Products, Sales_2018, and Salespeople. A search bar is also present in the Fields pane.

The screenshot shows the Power BI Data Model view. On the left, there is a table named "Date" with 17 rows, each containing a date from "01-01-2018 00:00:00" to "17-01-2018 00:00:00". On the right, there is a "Fields" pane with a search bar and a list of fact tables: Customers, Dates (which is selected), Locations, Products, Sales_2018, and Salespeople.

Place the lookup tables on the top and fact table below them

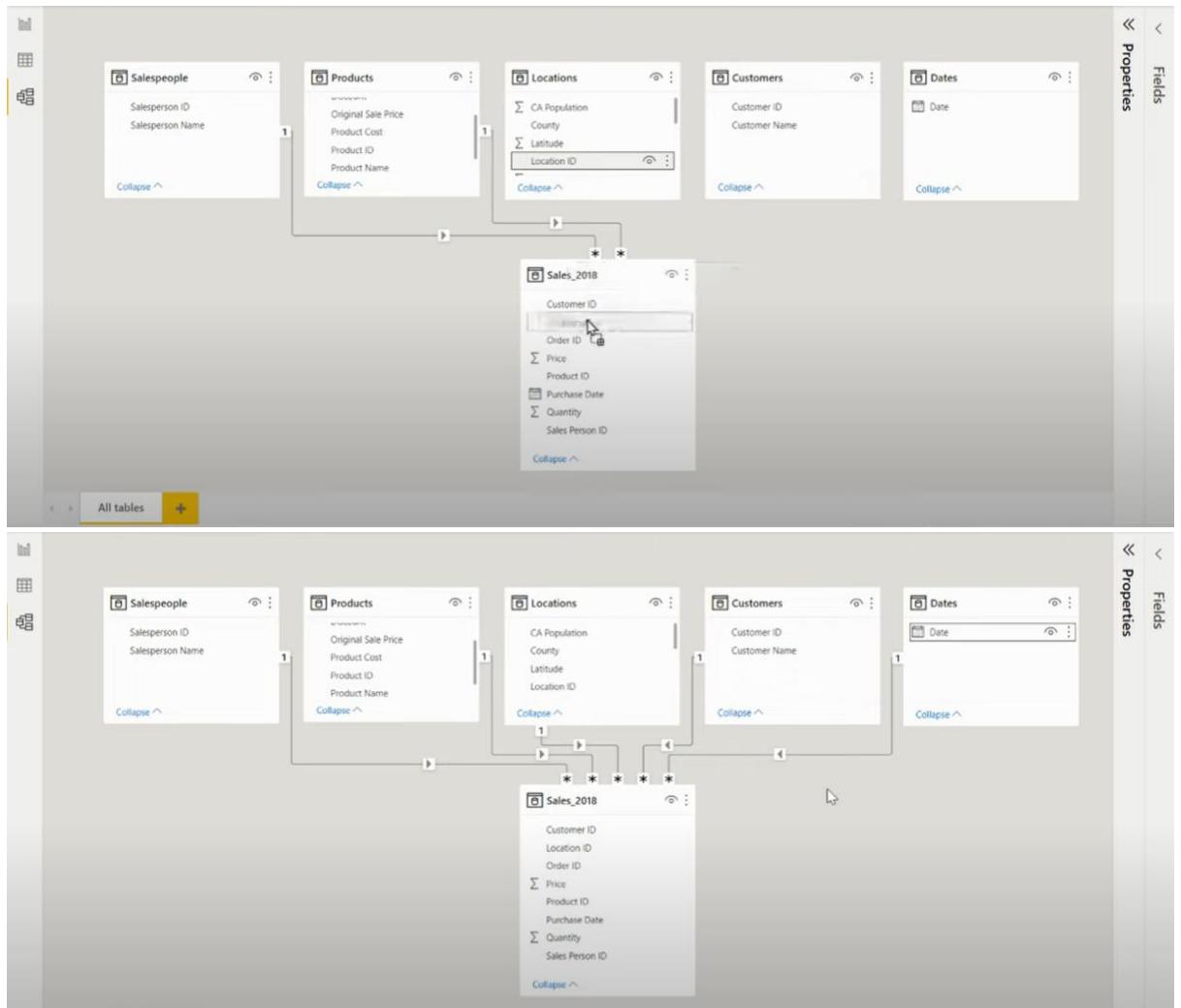
Lookup table (data table) – holds the primary keys

Fact table – holds the transactions of the lookup tables

The screenshot shows the Power BI Data Model view with five tables arranged horizontally. From left to right: Salespeople, Products, Locations, Customers, and Dates. The Dates table is highlighted with a yellow border. Below these tables is the Sales_2018 fact table. A vertical "Properties" pane is visible on the right side of the interface.

Create relationships between 2 tables

1. Drag column name from lookup table and drop it on fact table or
2. Go to manage relationships and create it



Creating DAX Expressions

New measure v/s new column

New measure – only logic will be computed in visualization

New column – will be created in the data (shall be created only if it is important)

The screenshot shows the Power BI Desktop interface. On the right, the 'Fields' pane is open, displaying a list of fields categorized under 'Sales_2018'. The 'Quantity Sold' measure is selected. The report canvas in the center shows a small icon representing a chart or visualization.

- Fields**
- Visualizations**
- Filters**
- Search**
- Sales_2018** (selected)
 - Customer ID
 - Customer Name
 - Measure
 - Dates
 - Locations
 - Products
 - Sales_2018
 - Customer ID
 - Location ID
 - Order ID
 - \sum Price
 - Product ID
 - Purchase Date
 - \sum Quantity
 - Sales Person ID
 - Salespeople

To move new created measures into a group

Go to Home ->enter data -> name the table (key measures) -> click load -> new group is created

The screenshot shows the Power BI Desktop interface with a table visual on the left and the Fields pane on the right. A red box highlights the 'Key Measures' group in the Fields pane.

Name	Key Measures
Aaron Carr	24 24
Aaron Cruz	26 26
Aaron Day	16 16
Aaron Johnson	16 16
Aaron Miller	20 20
Aaron Mills	20 20
Aaron Moreno	26 26
Aaron Tucker	26 26
Adam Alexander	21 21
Adam Bailey	32 32
Adam Duncan	30 30
Adam Hernandez	21 21
Adam Hunter	17 17
Adam Jenkins	38 38
Adam McCoy	22 22
Adam McDonald	18 18
Adam Myers	24 24
Adam Riley	26 26
Adam Thompson	29 29
Adam Wheeler	18 18
Adam White	21 21
Alan Gomez	19 19
Alan Green	31 31
Alan Miller	38 38
Alan Parker	36 36
Alan Perry	38 38
Alan Scott	43 43
Total	21078 21078

Storage mode: Import
Data refreshed: 8/14/2022, 9:10:59 PM

- Fields**
- Visualizations**
- Filters**
- Search**
- Key Measures** (highlighted)
- Dates**
- Locations**
- Products**
- Sales_2018**
- Salespeople**

Select new created measure -> measure tools -> home table -> select the group

The screenshot shows the Power BI Measure Tools dialog. The 'Name' field contains 'Quantity Sold'. The 'Home table' dropdown is set to 'Customers'. The 'Format' dropdown is set to 'Whole number'. The 'Data category' dropdown is set to 'Uncategorized'. The 'Calculations' section shows the formula: `Quantity Sold = SUM(Sales_2018[Quantity])`. The 'Fields' pane on the right lists various entities: Customers, Key Measures, Locations, Products, Sales_2018, and Salespeople. The 'Quantity Sold' field is selected under the 'Key Measures' group.

This screenshot is similar to the previous one, but the 'Quantity Sold' measure is now selected in the 'Fields' pane. A checkmark is visible next to the measure name in the pane. The rest of the interface remains the same, showing the measure definition and the list of entities.

More DAX expressions

The screenshot shows the Power BI Data Editor interface. In the top ribbon, the 'Measure tools' tab is selected. A new measure named 'Total Sales' has been created, based on the 'Sales_2018' table. The formula is `SUMX(Sales_2018,Sales_2018[Quantity]*Sales_2018[Price])`. The Fields pane on the right lists the 'Key Measures' group, which includes 'Quantity Sold' and 'Total Sales'. The main area displays a table of sales data.

Salesperson	Quantity	Price	Total Sales
Aaron Carr	24	24	
Aaron Cruz	26	26	
Aaron Day	16	16	
Aaron Johnson	16	16	
Aaron Miller	20	20	
Aaron Mills	20	20	
Aaron Moreno	26	26	
Aaron Tucke	26	26	
Adam Alexander	21	21	
Adam Bailey	32	32	
Adam Duncan	30	30	
Adam Hernandez	21	21	
Adam Hunter	17	17	
Adam Jenkins	38	38	
Adam McCoy	22	22	
Adam McDonald	16	18	
Adam Myers	24	24	
Adam Riley	26	26	
Adam Thompson	29	29	
Adam Wheeler	18	18	
Adam White	21	21	
Alan Gomez	19	19	
Alan Green	31	31	
Alan Miller	38	38	
Alan Parker	36	36	
Alan Perry	38	38	
Alan Scott	43	43	
Alan Smart	36	36	
Total	21078	21078	25661209

To change the currency

Select measure -> measure tools -> formatting group -> select the currency type

In the same way, measure can be converted into percentage and decimal places can be specified

The screenshot shows the Power BI Data Editor interface with the 'Format' tab selected in the top ribbon. The 'Total Sales' measure has its format set to 'Whole number'. The Fields pane on the right shows the 'Key Measures' group, which includes 'Quantity Sold' and 'Total Sales'. The main area displays the same sales data table.

Salesperson	Quantity	Price	Total Sales
Aaron Carr	24	24	564
Aaron Cruz	26	26	676
Aaron Day	16	16	256
Aaron Johnson	16	16	256
Aaron Miller	20	20	400
Aaron Mills	20	20	400
Aaron Moreno	26	26	676
Aaron Tucke	26	26	676
Adam Alexander	21	21	441
Adam Bailey	32	32	1024
Adam Duncan	30	30	900
Adam Hernandez	21	21	441
Adam Hunter	17	17	289
Adam Jenkins	38	38	1444
Adam McCoy	22	22	484
Adam McDonald	18	18	324
Adam Myers	24	24	576
Adam Riley	26	26	676
Adam Thompson	29	29	841
Adam Wheeler	18	18	324
Adam White	21	21	441
Alan Gomez	19	19	361
Alan Green	31	31	961
Alan Miller	38	38	1444
Alan Parker	36	36	1296
Alan Perry	38	38	1444
Alan Scott	43	43	1849
Alan Smart	36	36	1296
Total	21078	21078	25661209

Screenshot of Power BI Data Editor showing a table named "Sales_2018". The table has columns: Name, Sales ID, Order ID, and Price. A measure named "Total Transactions" is defined as COUNTROWS(Sales_2018). The Fields pane on the right shows various measures like Total Sales, Total Transactions, and Quantity Sold.

Name	Sales ID	Order ID	Price
Aaron Carr	24	24	\$26.846
Aaron Cruz	26	26	\$32.099
Aaron Day	16	16	\$22.123
Aaron Johnson	16	16	\$18.038
Aaron Miller	20	20	\$28.716
Aaron Mills	20	20	\$17.891
Aaron Moreno	26	26	\$27.996
Aaron Tucker	26	26	\$38.158
Adam Alexander	21	21	\$32.187
Adam Bailey	32	32	\$48.930
Adam Duncan	30	30	\$34.143
Adam Hernandez	21	21	\$18.793
Adam Hunter	17	17	\$21.762
Adam Jenkins	38	38	\$50.294
Adam Mccoy	22	22	\$31.402
Adam McDonald	18	18	\$27.727
Adam Myers	24	24	\$30.997
Adam Riley	26	26	\$45.066
Adam Thompson	29	29	\$39.779
Adam Wheeler	18	18	\$10.411
Adam White	21	21	\$18.967
Alan Gomez	19	19	\$26.129
Alan Green	31	31	\$36.911
Alan Miller	38	38	\$51.804
Alan Parker	36	36	\$39.137
Alan Perry	38	38	\$44.752
Alan Scott	43	43	\$52.058
Alan Smith	36	36	\$34.043
Total	21078	21078	\$2,56.61.209

To create DAX expression, two columns located in different tables

Quantity coming from sales table and product cost coming from products table

Screenshot of Power BI Data Editor showing a table named "Sales_2018". A measure named "Total Costs" is defined as SUMX(Sales_2018, Sales_2018[Quantity] * [Price]). The Fields pane on the right shows various measures like Total Sales, Total Transactions, and Total Costs.

	Quantity	Price	Total Cost
Aaron Carr	24	\$26.846	\$604.224
Aaron Cruz	16	\$32.099	\$513.584
Aaron Day	10	\$22.123	\$221.230
Aaron Johnson	9	\$18.038	\$162.342
Aaron Miller	11	\$28.716	\$315.876
Aaron Mills	13	\$17.891	\$232.583
Aaron Moreno	15	\$27.996	\$419.940
Aaron Tucker	16	\$38.158	\$606.528
Adam Alexander	11	\$32.187	\$354.057
Adam Bailey	19	\$48.930	\$937.670
Adam Duncan	17	\$34.143	\$574.431
Adam Hernandez	13	\$18.793	\$247.302
Adam Hunter	9	\$21.762	\$195.858
Adam Jenkins	19	\$50.294	\$955.586
Adam Mccoy	12	\$31.402	\$376.824
Adam McDonald	11	\$27.727	\$304.997
Adam Myers	11	\$30.997	\$340.967
Adam Riley	15	\$45.066	\$675.990
Adam Thompson	14	\$39.779	\$556.886
Adam Wheeler	9	\$18.411	\$165.699
Adam White	11	\$18.967	\$208.637
Alan Gomez	9	\$26.129	\$235.161
Alan Green	13	\$36.911	\$483.443
Alan Miller	23	\$51.804	\$1,187.448
Alan Parker	18	\$39.137	\$704.466
Alan Perry	18	\$44.752	\$805.544
Alan Scott	22	\$52.058	\$1,145.276
Alan Smith	17	\$34.043	\$578.731
Total	1089	21078	\$2,56.61.209

1 Total Costs = SUMX(Sales_2018,Sales_2018[Quantity] * RELATED(Products[Product Cost]))

Too few arguments were passed to the RELATED function. The minimum argument count for the function is 1.

Customer Name	Total Sales	Quantity Sold	Total Profits	Total Costs	Total Transactions
Aaron Carr	\$20,946	11	-\$2,094	\$2,099	26
Aaron Cruz	\$32,099	16	-\$3,209	\$2,099	26
Aaron Day	\$22,123	10	-\$2,212	\$2,099	16
Aaron Johnson	\$18,038	9	-\$1,804	\$2,099	16
Aaron Miller	\$28,716	11	-\$2,872	\$2,099	20
Aaron Mills	\$17,891	13	-\$1,789	\$2,099	20
Aaron Moreno	\$27,996	15	-\$2,799	\$2,099	26
Aaron Tucker	\$38,158	16	-\$3,816	\$2,099	26
Adam Alexander	\$32,187	11	-\$3,219	\$2,099	21
Adam Bailey	\$48,930	19	-\$4,893	\$2,099	32
Adam Duncan	\$34,143	17	-\$3,414	\$2,099	30
Adam Hernandez	\$18,793	13	-\$1,879	\$2,099	21
Adam Hunter	\$21,762	9	-\$2,176	\$2,099	17
Adam Jenkins	\$50,294	19	-\$5,029	\$2,099	38
Adam McCoy	\$31,402	12	-\$3,140	\$2,099	22
Adam McDonald	\$27,727	11	-\$2,773	\$2,099	18
Adam Myers	\$30,997	11	-\$3,099	\$2,099	24
Adam Riley	\$45,066	15	-\$4,507	\$2,099	26
Adam Thompson	\$39,779	14	-\$3,978	\$2,099	29
Adam Wheeler	\$18,411	9	-\$1,841	\$2,099	18
Adam White	\$18,967	11	-\$1,897	\$2,099	21
Alan Gomez	\$26,129	9	-\$2,613	\$2,099	19
Alan Green	\$36,911	13	-\$3,691	\$2,099	31
Alan Miller	\$51,804	23	-\$5,180	\$2,099	38
Alan Parker	\$39,137	18	-\$3,914	\$2,099	36
Alan Perry	\$44,752	18	-\$4,475	\$2,099	38
Alan Scott	\$52,058	22	-\$5,206	\$2,099	43
Alan Sims	\$34,741	17	-\$3,474	\$2,099	36
Total	\$2,56,61,209	10889	-\$2,56,61,209	\$2,099	21078

Using 2 measures to create 3rd measure

1 Total Profits = [Total Sales] - [Total Costs]

Customer Name	Total Sales	Total Costs	Total Profits
Aaron Carr	\$26,846	\$17,146	\$9,700
Aaron Cruz	\$32,099	\$20,715	\$11,384
Aaron Day	\$22,123	\$16,046	\$6,077
Aaron Johnson	\$18,038	\$11,812	\$6,226
Aaron Miller	\$28,716	\$19,256	\$9,460
Aaron Mills	\$17,891	\$12,519	\$5,372
Aaron Moreno	\$27,996	\$15,594	\$12,402
Aaron Tucker	\$38,158	\$26,291	\$11,867
Adam Alexander	\$32,187	\$23,439	\$8,748
Adam Bailey	\$48,930	\$33,488	\$15,452
Adam Duncan	\$34,143	\$24,473	\$9,669
Adam Hernandez	\$18,793	\$12,056	\$6,737
Adam Hunter	\$21,762	\$15,122	\$6,640
Adam Jenkins	\$50,294	\$26,954	\$13,340
Adam McCoy	\$31,402	\$21,636	\$9,766
Adam McDonald	\$27,727	\$19,470	\$8,257
Adam Myers	\$30,997	\$23,027	\$7,970
Adam Riley	\$45,066	\$34,057	\$10,999
Adam Thompson	\$39,779	\$24,656	\$15,123
Adam Wheeler	\$18,411	\$10,419	\$8,392
Adam White	\$18,967	\$12,625	\$6,342
Alan Gomez	\$26,129	\$16,208	\$9,921
Alan Green	\$36,911	\$23,836	\$13,075
Alan Miller	\$51,804	\$36,701	\$15,103
Alan Parker	\$39,137	\$25,094	\$14,043
Alan Perry	\$44,752	\$32,093	\$12,659
Alan Scott	\$52,058	\$35,767	\$16,291
Alan Sims	\$34,741	\$14,003	\$20,738
Total	\$2,56,61,209	\$1,73,17,316	\$1,83,493

1 Profit Margin = DIVIDE([Total Profits],[Total Sales])

Safe Divide function with ability to handle divide by zero case.

Customer Name	Total Sales	Total Profits	Profit Margin
Alan Carr	\$26,846	\$9,700	36%
Alan Thomas	\$32,099	\$11,384	35%
Albert Cunningham	\$22,123	\$6,077	27%
Albert Jacobs	\$18,038	\$6,226	34%
Albert Kennedy	\$28,716	\$9,460	33%
Albert King	\$17,891	\$5,372	30%
Albert Robinson	\$27,996	\$12,402	44%
Albert Young	\$38,158	\$11,867	31%
Andrew Burns	\$34,143	\$9,669	28%
Andrew Burns	\$18,411	\$6,737	37%
Andrew Butler	\$50,294	\$13,340	26%
Andrew Fernandez	\$31,402	\$7,970	25%
Andrew Graham	\$27,727	\$8,257	32%
Andrew Hansen	\$26,129	\$9,921	37%
Andrew James	\$36,911	\$13,075	36%
Andrew Martin	\$51,804	\$15,103	29%
Andrew Peters	\$39,137	\$14,043	36%
Andrew Reynolds	\$44,752	\$12,659	32%
Andrew Robinson	\$52,058	\$16,291	31%
Anthony Banks	\$32,187	\$6,077	19%
Anthony Berry	\$10,760	\$3,969	37%
Anthony Chapman	\$10,804	\$3,501	33%
Anthony Davis	\$10,848	\$3,197	30%
Anthony Little	\$30,997	\$13,099	42%
Anthony Parker	\$18,891	\$4,087	22%
Anthony Simpson	\$41,170	\$18,038	44%
Anthony Torres	\$10,576	\$3,017	29%
Anthony Turner	\$16,477	\$4,706	28%
Antonio Cooper	\$3,611	\$1,129	31%
Total	\$83,493	\$2,56,61,209	\$1,73,17,316

To remove data hierarchy

click on values dropdown -> unclick date hierarchy -> and click hierarchy

The screenshot shows the Power BI interface with a table visual on the left and a Fields pane on the right. In the Fields pane, under the 'Values' section, the 'Date' dropdown is open. A red circle highlights the 'Date' dropdown menu, which includes options like 'Remove field', 'Move', 'New quick measure', 'Show items with no data', and 'Drill through'. The 'Drill through' option is selected.

The screenshot shows the same Power BI interface after the hierarchy was removed. The 'Date' dropdown in the Fields pane is now closed, indicating that the hierarchy has been successfully removed from the values dropdown.

To change datatype of visual

Select visual -> column tools -> change datatype from structure group -> date format from formatting group

The screenshot shows the Power BI ribbon with the 'Column tools' tab selected. In the 'Formatting' group, the 'Data type' dropdown is set to 'Date/time'. Below it, the 'Format' dropdown shows '14-03-2001 13:30...'. The 'Properties' group shows 'Summarization: Don't summarize' and 'Data category: Uncategorized'. To the right, the 'Visualizations' and 'Fields' panes are visible, showing the same data structure as the first screenshot.

1 Dates = CALENDAR(AUTO())

	17.145	\$52,423	\$35,278	17	35	32.71%
Alan Thomas	\$17,145	\$52,423	\$35,278	17	35	32.71%
Albert Cunningham	\$14,110	\$44,945	\$30,835	17	38	31.99%
Albert Jacobs	\$10,961	\$35,470	\$24,469	16	38	30.69%
Albert Kennedy	\$10,257	\$34,640	\$23,913	17	25	30.57%
Albert Rice	\$12,235	\$46,071	\$31,436	14	24	32.44%
Albert Robinson	\$10,793	\$35,545	\$24,792	13	30	30.35%
Albert Young	\$3,556	\$14,383	\$10,827	11	15	24.72%
Andrew Adams	\$7,661	\$25,420	\$17,759	12	20	30.49%
Andrew Burns	\$18,421	\$33,815	\$35,384	20	28	34.23%
Andrew Butler	\$8,811	\$31,918	\$23,107	12	24	27.61%
Andrew Fernandez	\$5,791	\$16,527	\$10,736	7	15	35.04%
Andrew Graham	\$9,347	\$34,635	\$25,288	18	32	26.99%
Andrew Hansen	\$9,823	\$28,823	\$19,000	13	23	34.08%
Andrew James	\$7,643	\$23,815	\$16,172	12	24	32.99%
Andrew Martin	\$13,314	\$40,179	\$26,865	18	42	33.14%
Andrew Peters	\$15,365	\$41,127	\$25,762	18	33	37.36%
Andrew Reynolds	\$15,007	\$41,846	\$26,859	12	30	35.60%
Andrew Simpson	\$10,961	\$35,470	\$24,469	10	19	34.23%
Anthony Banks	\$8,721	\$20,869	\$22,149	14	22	20.23%
Anthony Berry	\$10,760	\$35,960	\$25,200	10	21	29.22%
Anthony Chapman	\$10,004	\$30,305	\$19,501	14	24	35.65%
Anthony Fisher	\$7,844	\$22,721	\$14,877	9	26	34.57%
Anthony Little	\$9,361	\$31,696	\$22,335	9	20	29.53%
Anthony Parker	\$18,591	\$40,087	\$29,496	13	32	38.66%
Anthony Simpson	\$14,170	\$37,706	\$18,031	10	15	23.13%
Anthony Torres	\$10,576	\$31,017	\$20,441	14	23	34.10%
Anthony Turner	\$16,477	\$47,706	\$31,229	15	34	34.43%
Antonio Cooper	\$3,611	\$11,129	\$7,518	10	13	32.45%
Antonio Diaz	\$13,711	\$33,451	\$19,740	13	31	40.99%
Total	\$83,43,893	\$2,56,61,209	\$1,73,17,316	10889	21078	32.52%

01 January 2018 \$60,777
02 January 2018 \$39,297
03 January 2018 \$7,861
04 January 2018 \$28,193
05 January 2018 \$25,968
06 January 2018 \$27,311
07 January 2018 \$37,426
08 January 2018 \$36,614
09 January 2018 \$34,114
10 January 2018 \$34,114
Total **\$2,56,61,209**

Visualizations Fields

Filters

Build visual

Key Measures

- Profit Margin
- Quantity Sold
- Total Costs
- Total Profits
- Total Sales**
- Total Transactions

Dates

Locations

Products

Sales_2018

Salespeople

Time Intelligence

Keep all filters

Time Intelligence Calculations

Customer Name	Total Profits	Total Sales	Total Costs	Total Transactions	Quantity Sold	Profit Margin
Alan Sims	\$9,051	\$24,043	\$14,992	17	26	37.95%
Alan Thomas	\$17,145	\$52,423	\$35,278	17	35	32.71%
Albert Cunningham	\$14,110	\$44,945	\$30,835	17	38	31.99%
Albert Jacobs	\$10,961	\$35,470	\$24,469	16	38	30.69%
Albert Kennedy	\$10,257	\$33,640	\$23,013	11	25	30.77%
Albert Rice	\$12,235	\$46,071	\$31,436	14	24	32.44%
Albert Robinson	\$10,793	\$35,545	\$24,792	13	30	30.35%
Albert Young	\$3,556	\$14,383	\$10,827	11	15	24.72%
Andrew Adams	\$7,661	\$25,420	\$17,759	12	20	30.14%
Andrew Burns	\$18,421	\$33,815	\$35,384	20	28	34.23%
Andrew Butler	\$8,811	\$31,918	\$23,107	12	24	27.61%
Andrew Fernandez	\$5,791	\$16,527	\$10,736	7	15	35.04%
Andrew Graham	\$9,347	\$34,635	\$25,288	18	32	26.99%
Andrew Hansen	\$9,823	\$28,823	\$19,000	13	23	34.08%
Andrew James	\$7,643	\$23,815	\$16,172	12	24	32.99%
Andrew Martin	\$13,314	\$40,179	\$26,865	18	42	33.14%
Andrew Peters	\$15,365	\$41,127	\$25,762	18	33	37.36%
Andrew Reynolds	\$15,007	\$41,846	\$26,859	12	30	35.60%
Andrew Simpson	\$10,961	\$35,470	\$24,469	10	19	34.23%
Anthony Banks	\$8,721	\$20,869	\$22,149	14	22	20.23%
Anthony Berry	\$10,760	\$35,960	\$25,200	10	21	29.22%
Anthony Chapman	\$10,004	\$30,305	\$19,501	14	24	35.65%
Anthony Fisher	\$7,844	\$22,721	\$14,877	9	26	34.57%
Anthony Little	\$9,361	\$31,696	\$22,335	9	20	29.53%
Anthony Parker	\$18,591	\$40,087	\$29,496	13	32	38.66%
Anthony Simpson	\$14,170	\$37,706	\$18,031	10	15	23.13%
Anthony Torres	\$10,576	\$31,017	\$20,441	14	23	34.10%
Anthony Turner	\$16,477	\$47,706	\$31,229	15	34	34.43%
Antonio Cooper	\$3,611	\$11,129	\$7,518	10	13	32.45%
Antonio Diaz	\$13,711	\$33,451	\$19,740	13	31	40.99%
Total	\$83,43,893	\$2,56,61,209	\$1,73,17,316	10889	21078	32.52%

Visualizations Fields

Filters

Build visual

Key Measures

- Customers
- Dates
- Locations
- Products
- Sales_2018
- Salespeople
- Time Intelligence

Add data fields here

Drill through

Cross-report

Keep all filters

Add drill-through fields here

File Home Insert Modeling View Help Format Data / Drill Table tools

Name Measure Format Data category Uncategorized

Home table Time Intelligence CALCULATE(Expression, [Filter], ...)

Evaluates an expression in a context modified by filters.

Measure Tools

New Quick measure measure Calculations

Structure

1 LY Sales = CALCULATE([Total Sales],SAMEPERIODLASTYEAR(Dates[Date]))

	1 LY Sales = CALCULATE([Total Sales],SAMEPERIODLASTYEAR(Dates[Date]))
Aaron Cruz	\$6,077
Aaron Day	\$22,123
Aaron Johnson	\$18,098
Aaron Miller	\$19,460
Aaron Mirek	\$12,154
Aaron Moreno	\$27,998
Aaron Tucker	\$28,158
Adam Alexander	\$18,748
Adam Bailey	\$15,442
Adam Duncan	\$24,143
Adam Hernandez	\$18,793
Adam Hunter	\$12,640
Adam Jenkins	\$20,340
Adam McCoy	\$9,766
Adam McDonald	\$8,257
Adam Myers	\$7,970
Adam Riley	\$11,009
Adam Thompson	\$15,123
Adam Verner	\$12,154
Adam White	\$6,342
Alan Gomez	\$9,021
Alan Green	\$11,073
Alan Miller	\$15,103
Alan Parker	\$14,043
Alan Perry	\$12,659
Alan Scott	\$16,591
Alan Sims	\$9,051
Alan Thomas	\$17,145
Alfred Cunningham	\$14,110
Total	\$83,43,893

Date Total Sales

01-01-2018 \$60,777
02-01-2018 \$39,297
03-01-2018 \$7,861
04-01-2018 \$28,193
05-01-2018 \$25,968
06-01-2018 \$27,311
07-01-2018 \$37,426
08-01-2018 \$36,614
09-01-2018 \$34,114
10-01-2018 \$34,114
Total **\$2,56,61,209**

Visualizations Fields

Filters

Build visual

Key Measures

- Profit Margin
- Quantity Sold
- Total Costs
- Total Profits
- Total Sales**
- Total Transactions

Dates

Locations

Products

Sales_2018

Salespeople

Time Intelligence

Keep all filters

The screenshot shows a Power BI desktop interface. On the left is a table visualization titled "Customer Name" with columns: Customer Name, Total Profits, Total Sales, Total Costs, Total Transactions, Quantity Sold, and Profit Margin. The data includes rows for various customers like Aaron Cruz, Aaron Day, etc., with their respective sales figures. On the right is the "Fields" pane, which contains a search bar and a list of fields categorized under "Key Measures" (Profit Margin, Quantity Sold, Total Costs, Total Profits, Total Sales, Total Transactions), "Customers", "Dates" (with a "Date Hierarchy" section), "Products", "Salespeople", and "Time Intelligence". A "Filters" section at the bottom allows for drill-through, cross-reporting, and keeping all filters.

To Calculate Cumulative Sales

This screenshot shows the same Power BI desktop environment. A new DAX measure named "Cumulative Sales" has been created in the "Measure" tab of the ribbon. The formula for this measure is: `FILTER(ALLSELECTED(Dates), Dates[Date] <= MAX(Dates[Date]))`. The table visualization now includes a column for "Cumulative Sales" which shows the total sales up to the current date for each row. The "Fields" pane remains the same, displaying the available fields and categories.

Cumulative Sales:

It will calculate cumulative sales based on Total sales DAX function

Filter function will filtered out the data

Allselected function will ignore all the applied filters

Filter condition: current date \leq max date

1 Cumulative Sales = CALCULATE([Total Sales],
FILTER(ALLSELECTED(Dates),
Dates[Date] <= MAX(Dates[Date])))

Date	Total Sales	LY Sales
26-12-2019	\$24,187	\$44,831
27-12-2019	\$45,914	\$25,966
28-12-2019	\$28,064	\$27,636
29-12-2019	\$35,905	\$29,084
30-12-2019	\$53,183	\$46,534
31-12-2019	\$15,641	\$20,200
01-01-2020	\$16,995	\$18,501
02-01-2020	\$9,921	\$26,102
03-01-2020	\$4,974	\$33,949
04-01-2020	\$21,347	\$30,639
05-01-2020	\$20,399	\$20,695
06-01-2020	\$23,023	\$28,171
07-01-2020	\$16,889	\$35,972
08-01-2020	\$14,172	\$24,416
09-01-2020	\$9,912	\$37,324
Total	\$97,01,923	\$1,01,61,697

Visualizations Fields

Build visual

Search:

Key Measures

- Profit Margin
- Quantity Sold
- Total Costs
- Total Profits
- Total Sales**
- Total Transactions

Customers

Dates

Locations

Products

Sales_2018

Salespeople

Time Intelligence

Column1

LY Sales

Add drill through fields here

Report Building

Change the label contents

Select visual -> format visual -> visual -> options -> label contents -> select option

Visualizations Fields

Format visual

Search: county

Visual General ...

Legend

Slices

Detail labels

Options

Position: Outside

Overflow text

Label contents: Data value, percent of

Values

Visualizations Fields

Format visual

Search: county

Visual General ...

Legend

Slices

Category

Data value

Percent of total

Category, data value

Category, percent of total

Data value, percent of total

All detail labels

Data value, percent of ▾

Values

Home -> text box (for report heading)

The screenshot shows a donut chart titled "Total Sales by County" with various county names and their sales values. The chart is displayed in the main workspace. To the right, the "Visualizations" pane is open, showing the "Build visual" section with a donut icon selected. The "Fields" pane is also visible, with "county" selected in the search bar and "County" listed under the "Locations" category.

Select visual -> format visual -> visual -> data labels -> turn on

The screenshot displays three visualizations: a donut chart, a bar chart titled "Total Sales by Customer Name", and a line chart titled "Total Sales by Date". The bar chart is currently selected. The "Format visual" pane is open on the right, showing the "Visual" tab selected. Under "Data labels", the switch is turned on, and the "Position" dropdown is set to "Auto". Other options like "Show data labels" and "Zoom slider" are also visible.

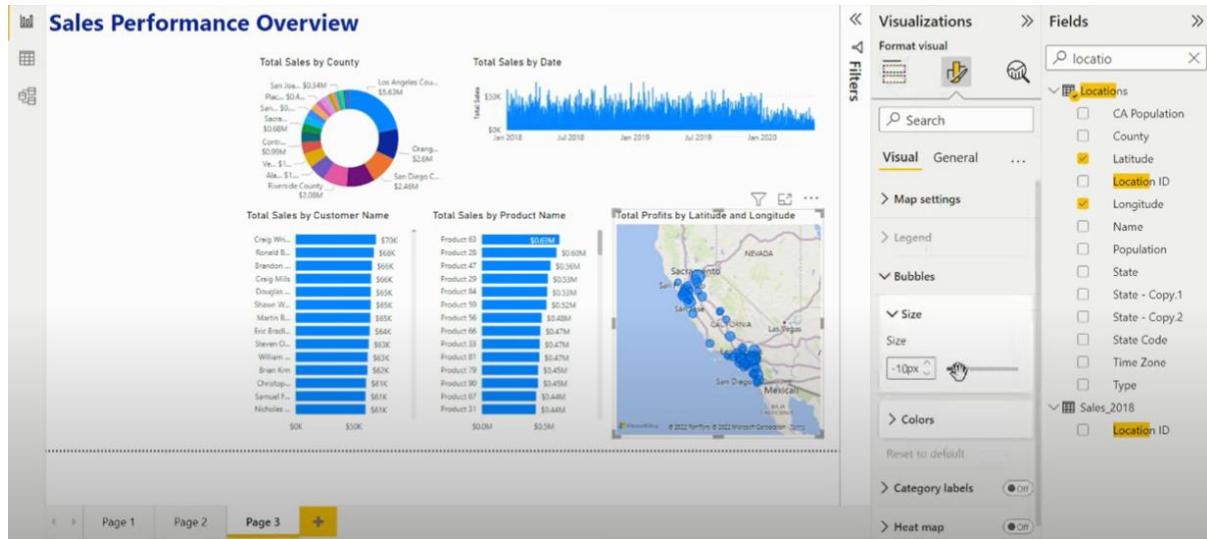
To show map as visual

The screenshot shows the same three visualizations as before. The "Visualizations" pane is now open on the right, showing the "Build visual" section with a map icon selected. The "Fields" pane is also visible, with "produ" selected in the search bar and "Product ID" listed under the "Products" category.

Use latitude and longitude for location

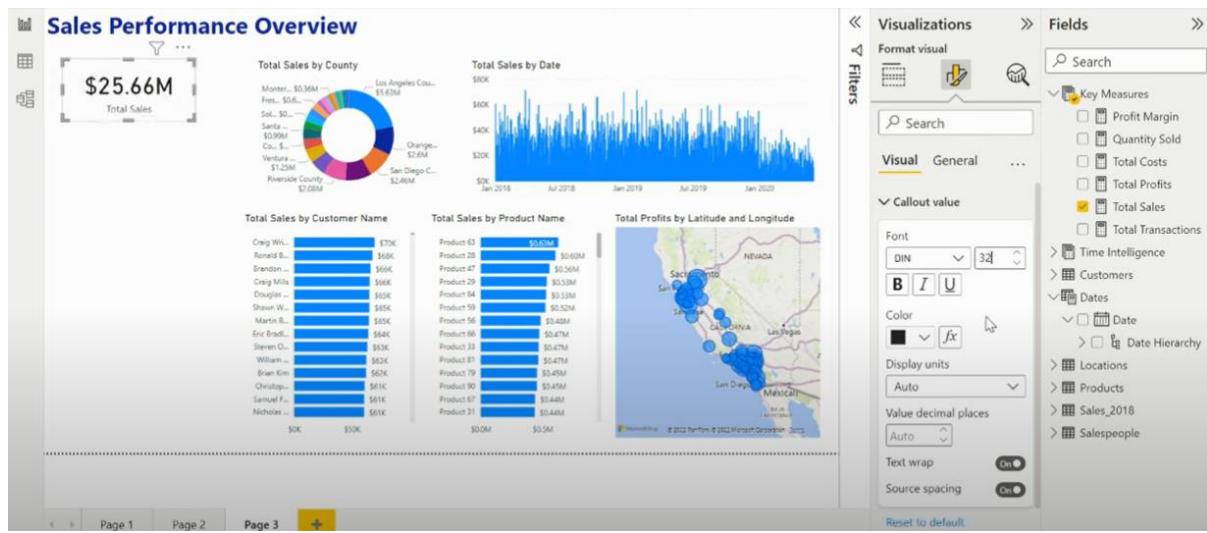
To increase the bubble size

Select visual -> format visual -> visual -> bubbles -> change size



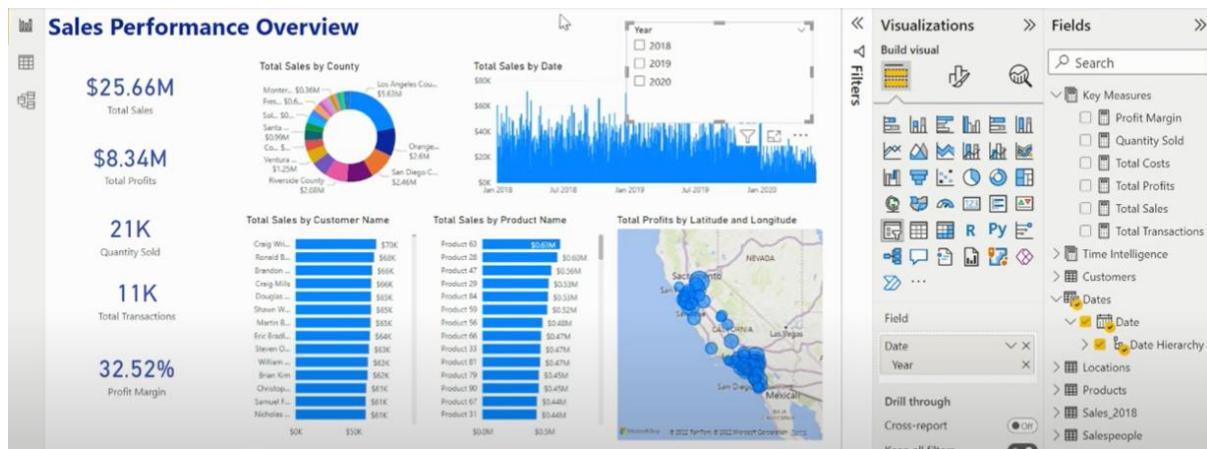
To format the cards

Select visual -> format visual -> visual -> callout value-> change size/colour/font

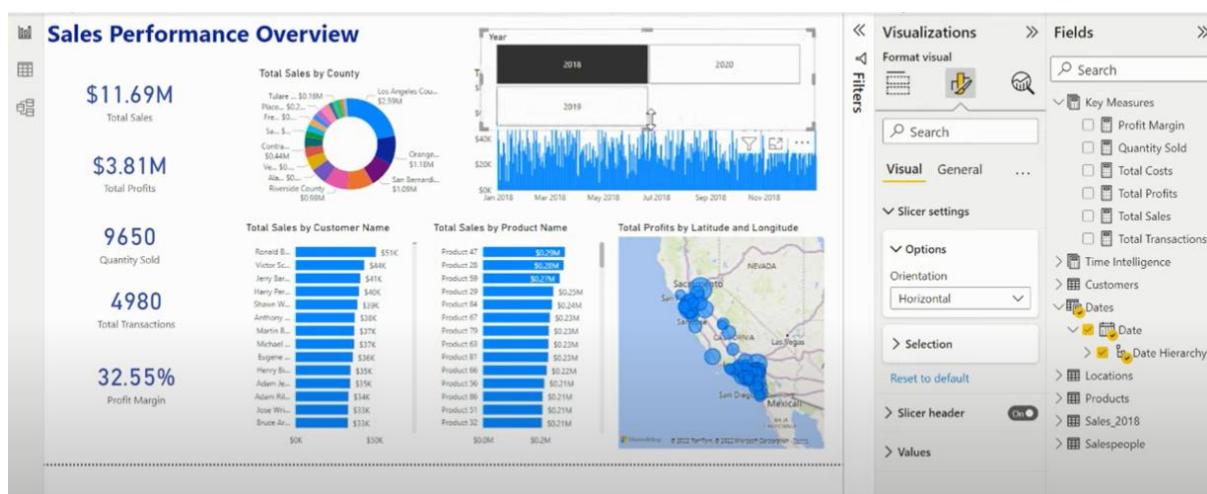
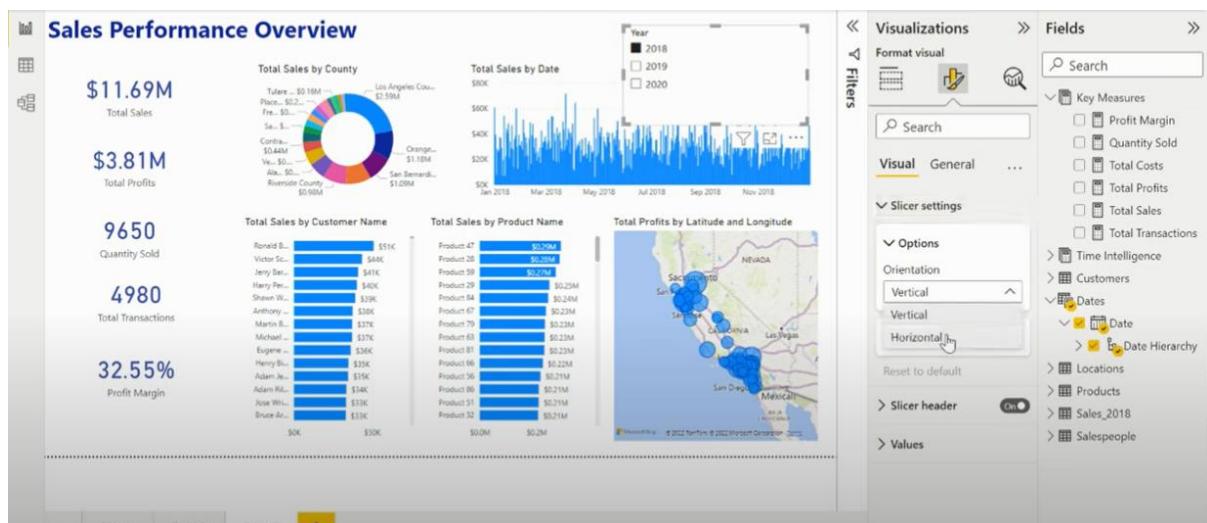


Insert slicer

Select date column -> select slicer -> date hierarchy -> keep year only

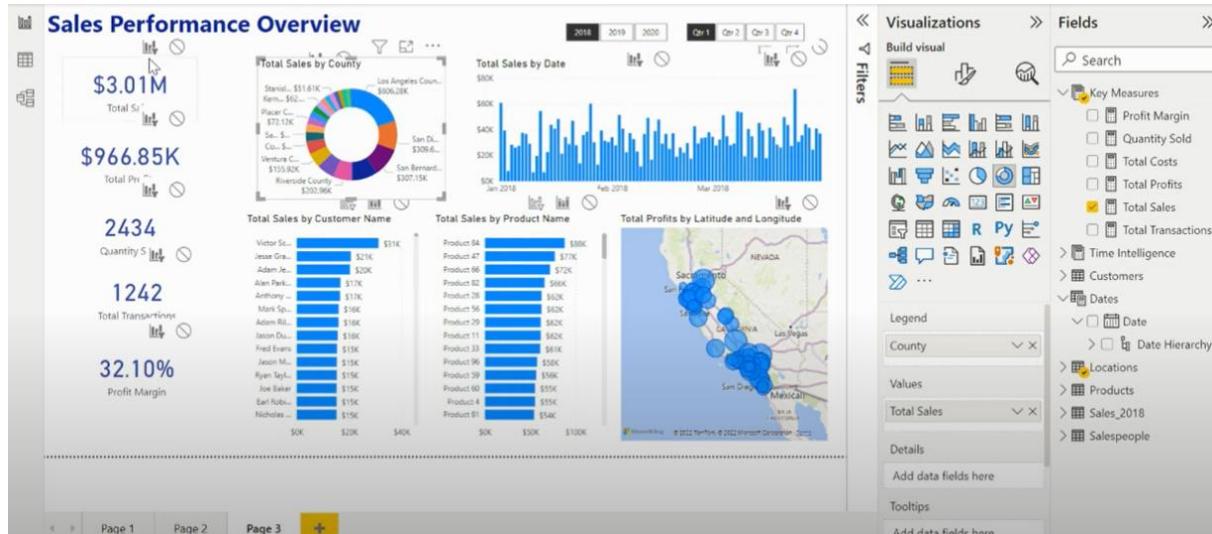


Select visual -> format visual -> visual -> slicer settings->options -> orientation -> horizontal -> adjust the slicer -> turn off slicer header



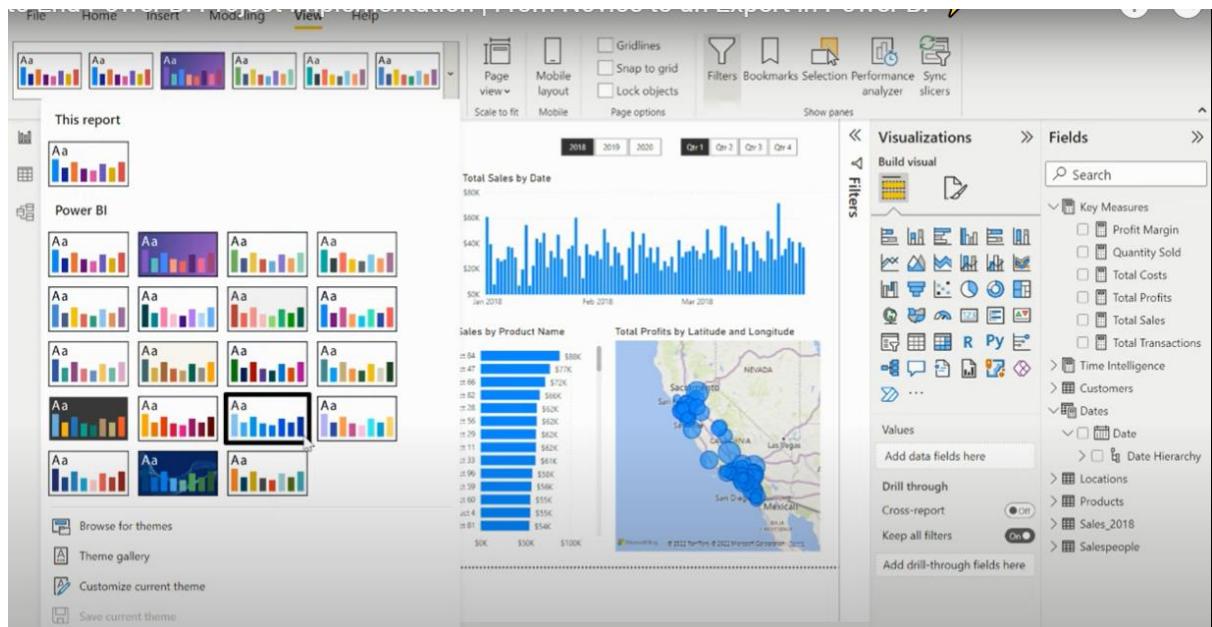
To format/highlight data as per selection in primary visual

Select primary visual -> format -> edit interactions -> select format/highlight for every other visual



To change theme

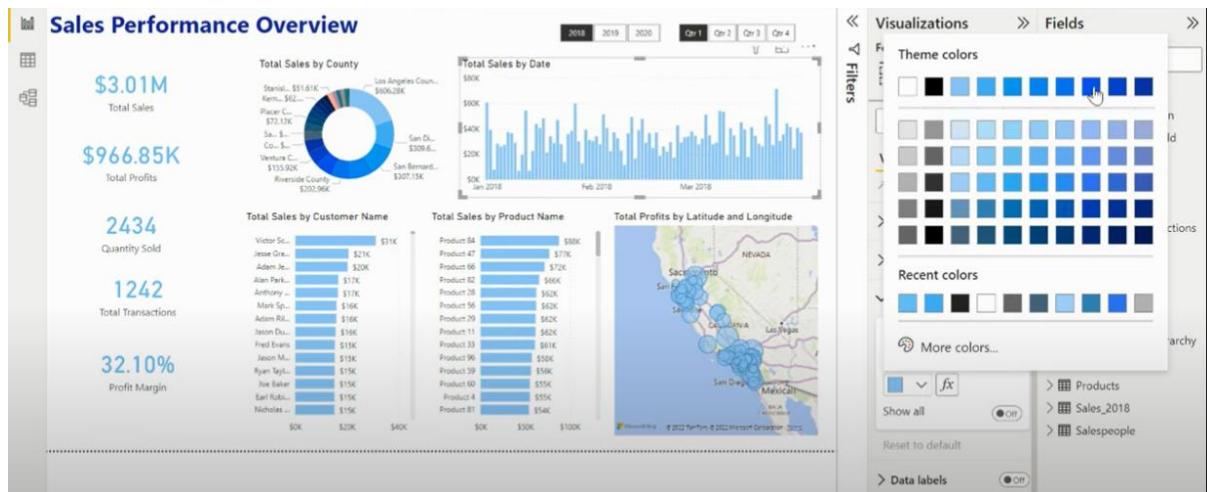
View -> themes group -> select theme



To change colour of visuals

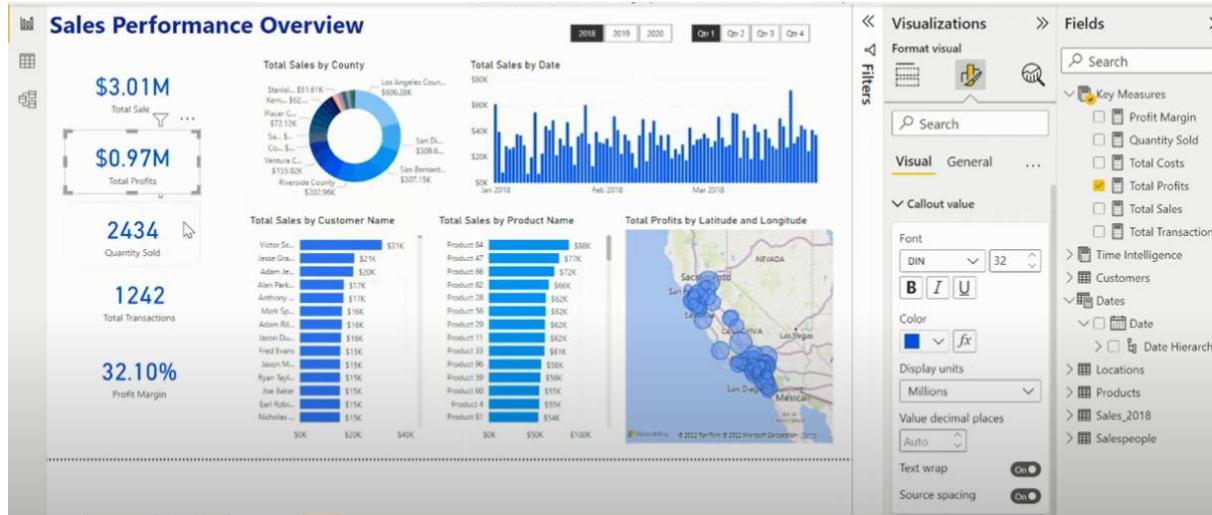
Select visual -> format visual -> visual -> columns -> color-> change colour

Can multi select visuals to give them same colour in one go (for cards)



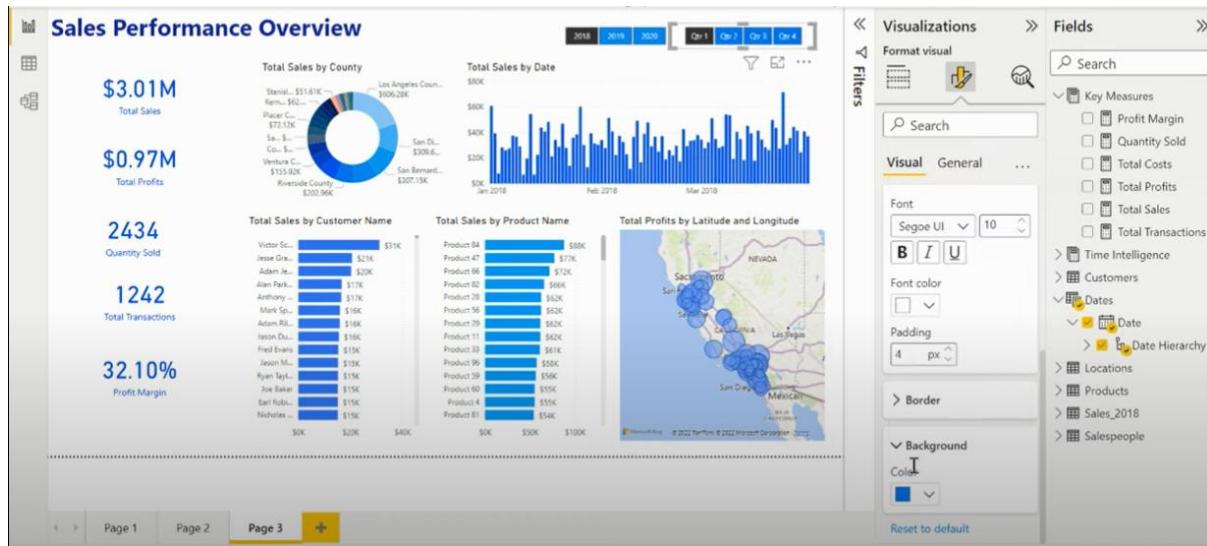
To change units

Select visual -> format visual -> visual -> callout value -> display units -> select unit



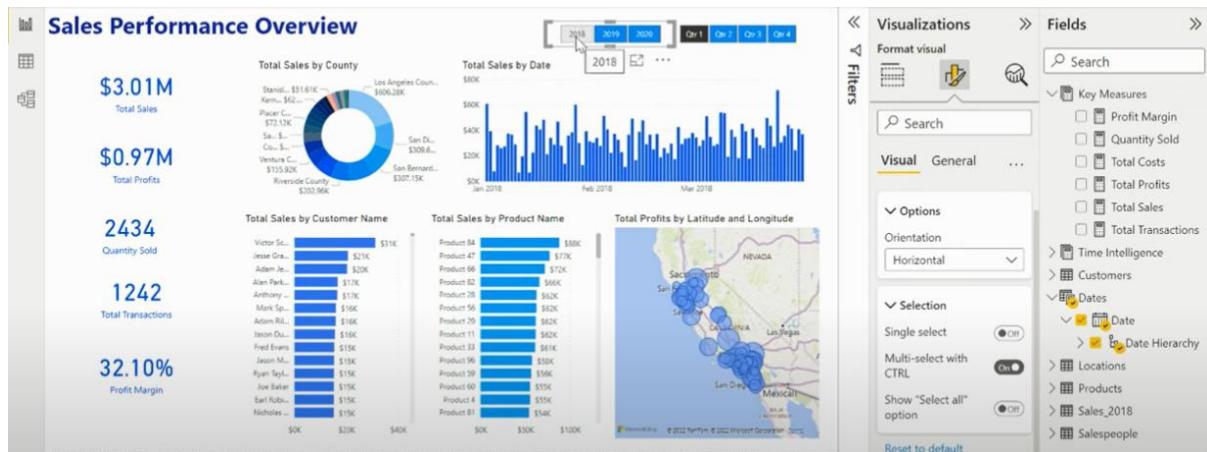
To change slicer colour

Select slicer -> format visual -> visual -> values -> background -> color -> change colour



Single / Multi Select Slicer

Select slicer -> format visual -> visual -> slicer settings->options -> selection -> choose selection option



How to find business working day or non-working day?

Load bank holiday list

Screenshot of Microsoft Power BI Data Get Data dialog showing the "Sheet1" table from "AP Bank Holidays.xlsx".

Date	Day	Occasion
01-01-2001	Saturday	New Year Day
01-01-2014	Friday	Bhogi
01-01-2015	Saturday	Makara Sankranti
01-01-2016	Wednesday	Republic Day
01-03-2001	Tuesday	Maha Sivarathri
01-03-2018	Friday	Holi
01-04-2002	Saturday	Ugadi
01-04-2010	Sunday	Sri Rama Navami
01-04-2014	Thursday	Dr.B.R.Ambedkar's Birthday/Mahaveer Jayanthi
01-04-2015	Friday	Good Friday
01-04-2017	Sunday	Easter
01-05-2001	Sunday	May Day
01-05-2003	Tuesday	Ramzan (Eid-ul-Fitr)
01-05-2016	Monday	Budha Purnima
01-07-2010	Sunday	Bakrid (Eid-ul-Azha)
01-08-2009	Tuesday	Moharrum
01-08-2015	Monday	Independence Day
01-08-2019	Friday	Sri Krishna Astami
01-08-1931	Wednesday	Vinayaka Chavithi
01-10-2002	Sunday	Mahatma Gandhi Jayanthi
01-10-2003	Monday	Durgashtami
01-10-2005	Wednesday	Vijayadasami
01-10-2009	Sunday	Eid Miladun Nabi
01-10-2024	Monday	Deepavali

Splitting the column by delimiter

Screenshot of Microsoft Power BI Advanced Editor showing the "Split Column by Delimiter" dialog.

The dialog shows the following settings:

- Select or enter delimiter: --Custom--
- Split at: Each occurrence of the delimiter (radio button selected)
- Quote Character: "
- Advanced options: Split using special characters (checkbox unchecked)

The preview pane shows the original data and the transformed data after splitting the "Date" column by the delimiter. The transformed data has three columns: Date.1, Date.2, and Date.3, which correspond to the split parts of the original Date column.

Screenshot of Microsoft Power BI Advanced Editor showing the transformed data and the "Query Settings" pane.

The transformed data table has columns: Date.1, Date.2, Date.3, Day, and Occasion.

The "Query Settings" pane shows the following properties:

- PROPERTIES**: Name: Bank holidays
- APPLIED STEPS**: Source, Navigation, Promoted Headers, Changed Type, Split Column by Delimiter

Splitting the incorrect date column

The screenshot shows the Power BI desktop interface with a table named "Bank holidays". The table has four columns: Date.2, Date.3, Day, and Ocassion. A context menu is open over the Date.3 column, specifically under the "Split Column" section. The "By Number of Characters" option is selected. The "APPLIED STEPS" pane on the right shows the step "Split Column by Delimiter" with "Changed Type1" applied.

The screenshot shows the Power BI desktop interface with the "Split Column by Number of Characters" dialog box open. The "Number of characters" input field contains "1". The "Split" section includes options: "Once, as far left as possible" (radio button selected), "Once, as far right as possible", and "Repeatedly". Below the dialog is a preview of the table showing the split results for rows 8 and 19.

The screenshot shows the Power BI desktop interface with a table named "Bank holidays". The table has five columns: Day, weekday, Date.3.2, day, and Ocassion. A context menu is open over the Day column, specifically under the "Rename Columns" section. The "Renamed Columns" option is selected. The "APPLIED STEPS" pane on the right shows the step "Renamed Columns" with "Changed Type2" applied.

Delete unwanted columns

Adding new column

Add new column -> custom column -> give column name -> write custom formula -> click ok -> new column created at end -> rearrange it

Custom Column

Add a column that is computed from the other columns.

New column name: Custom

Custom column formula: =

Available columns: month, day, weekday, Ocassion

No syntax errors have been detected.

OK Cancel

Custom Column

Add a column that is computed from the other columns.

New column name: year

Custom column formula: = 2018

Available columns: month, day, weekday, Ocassion

No syntax errors have been detected.

OK Cancel

	day	weekday	Ocassion	year
1	1	Saturday	New Year Day	2018
2	1	Friday	Bhogi	2018
3	1	Saturday	Makara Sankranti	2018
4	1	Wednesday	Republic Day	2018
5	3	Tuesday	Maha Sivarathri	2018
6	3	Friday	Holi	2018
7	4	Saturday	Ugadi	2018
8	4	Sunday	Sri Rama Navami	2018
9	4	Thursday	Dr.B.R.Ambedkar's Birthday/Mahaveer Jayanthi	2018
10	4	Friday	Good Friday	2018
11	4	Sunday	Easter	2018
12	5	1	May Day	2018
13	5	Tuesday	Ramzan (Eid-ul-Fitr)	2018
14	5	Monday	Budha Purnima	2018
15	7	Sunday	Bakrid (Eid-ul-Azha)	2018
16	8	Tuesday	Moharrum	2018
17	8	Monday	Independence Day	2018

Query Settings

Properties

Name: Bank holidays

All Properties

Applied Steps

Source, Navigation, Promoted Headers, Changed Type, Split Column by Delimiter, Changed Type1, Removed Columns, Split Column by Position, Changed Type2, Renamed Columns, Removed Columns1, **Added Custom**

Change data type of year column to whole number

The screenshot shows the Power BI Query Editor interface. On the left, there's a tree view with 'Orders' and 'Bank holidays' selected. The main area displays a table with columns: year, month, day, weekday, and Occasion. The 'year' column is highlighted in the 'Selected columns' dropdown, which is currently set to 'Whole Number'. The 'Applied Steps' pane on the right lists various steps taken during the query creation, including 'Changed Type'.

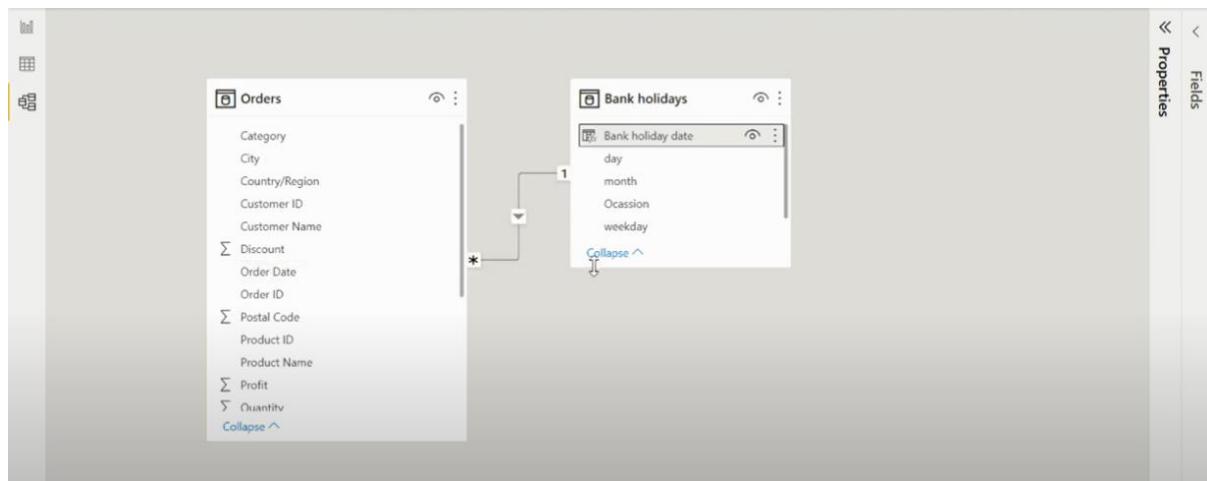
Select day column -> go to column tools -> properties section -> change summarization to don't summarize (same for month and year as well)

The screenshot shows the Power BI Desktop interface with the 'Column Tools' ribbon tab selected. In the 'Structure' group, the 'day' column is selected. In the 'Properties' group, the 'Summarization' dropdown is set to 'Don't summarize'. The 'Fields' pane on the right shows the 'Bank holidays' table with the 'day' field selected.

Convert day, month and year to date

The screenshot shows the Power BI Desktop interface with the M code editor open. The formula is: `Bank holiday date = DATE('Bank holidays'[year], 'Bank holidays'[month], 'Bank holidays'[day])`. The 'Fields' pane on the right shows the 'Bank holidays' table with the 'day' field selected.

Specify relationship between tables



In weekday function, 1st day of the week start from monday

This screenshot shows the 'Column tools' ribbon tab selected in Power BI. In the formula bar, a calculated column named 'is weekend' is being defined with the formula: `IF(WEEKDAY(Orders[Order Date],2)>5,TRUE(),FALSE())`. The formula bar also includes a note: 'Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE.' To the right, the 'Fields' pane is open, showing the 'Orders' table structure and various columns like Category, City, and Order Date.

This screenshot shows the continuation of the calculated column creation. The formula for 'is bank holiday' is: `IF(ISBLANK(RELATED('Bank holidays'[Bank holiday date])),FALSE(),TRUE())`. The formula bar includes a note: 'Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE.' The 'Fields' pane on the right shows the 'Bank holidays' table structure and the newly created 'is weekend' column under the 'Orders' table.

1 is bank holiday = IF(ISBLANK(RELATED('Bank holidays'[Bank holiday date])),FALSE(),TRUE())

Order Date	is weekend	is bank holiday
03 January 2018	False	False
04 January 2018	False	False
05 January 2018	False	False
06 January 2018	True	False
07 January 2018	True	False
09 January 2018	False	False
10 January 2018	False	False
11 January 2018	False	False
13 January 2018	True	False
14 January 2018	True	True
15 January 2018	False	True
16 January 2018	False	False
18 January 2018	False	False
19 January 2018	False	False
20 January 2018	False	False
21 January 2018	True	False
23 January 2018	False	False
26 January 2018	False	True
27 January 2018	True	False
28 January 2018	True	False
30 January 2018	False	False
31 January 2018	False	False
01 February 2018	False	False
02 February 2018	False	False
03 February 2018	True	False
04 February 2018	True	False

IF(LogicalTest, ResultIfTrue, [ResultIfFalse])
Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE.

1 is working day = IF(or([orders[is bank holiday],Orders[is weekend]],"not working day", "working day")

Order Date	is weekend	is bank holiday	is working day
03 January 2018	False	False	False
04 January 2018	False	False	False
05 January 2018	False	False	False
06 January 2018	True	False	False
07 January 2018	True	False	False
09 January 2018	False	False	False
10 January 2018	False	False	False
11 January 2018	False	False	False
13 January 2018	True	False	False
14 January 2018	True	True	True
15 January 2018	False	True	True
16 January 2018	False	False	False
18 January 2018	False	False	False
19 January 2018	False	False	False
20 January 2018	True	False	False
21 January 2018	True	False	False
23 January 2018	False	False	False
26 January 2018	False	False	False
27 January 2018	True	False	False
28 January 2018	True	False	False
30 January 2018	False	False	False
31 January 2018	False	False	False
01 February 2018	False	False	False
02 February 2018	False	False	False
03 February 2018	True	False	False
04 February 2018	True	False	False

Order Date	is weekend	is bank holiday	is working day
03 January 2018	False	False	working day
04 January 2018	False	False	working day
05 January 2018	False	False	working day
06 January 2018	True	False	not working day
07 January 2018	True	False	not working day
09 January 2018	False	False	working day
10 January 2018	False	False	working day
11 January 2018	False	False	working day
13 January 2018	True	False	not working day
14 January 2018	True	True	not working day
15 January 2018	False	True	not working day
16 January 2018	False	False	working day
18 January 2018	False	False	working day
19 January 2018	False	False	working day
20 January 2018	True	False	not working day
21 January 2018	True	False	not working day
23 January 2018	False	False	working day
26 January 2018	False	False	not working day
27 January 2018	True	False	not working day
28 January 2018	True	False	not working day
30 January 2018	False	False	working day
31 January 2018	False	False	working day
01 February 2018	False	False	working day
02 February 2018	False	False	working day
03 February 2018	True	False	not working day
04 February 2018	True	False	not working day
05 February 2018	False	False	working day
07 February 2018	False	False	working day
08 February 2018	False	False	working day

Networkdays in Power BI

New and easy way to calculate Business working days

The screenshot shows the Microsoft DAX documentation page for the `NETWORKDAYS` function. The left sidebar lists various DAX functions under the `MISC` category. The main content area describes the `NETWORKDAYS` function, which returns the number of whole workdays between two dates (inclusive). It specifies parameters for start_date, end_date, weekend, and holidays. A syntax example is provided: `NETWORKDAYS(<start_date>, <end_date>[, <weekend>, <holidays>])`. A "Parameters" table defines the terms and their meanings.

Term	Definition
start_date	A date that represents the start date. The dates for which the difference is to be computed. The start_date can be earlier than, the same as, or later than the end_date.
end_date	A date that represents the end date. The dates for which the difference is to be computed. The start_date can be earlier than, the same as, or later than the end_date.
weekend	Indicates the days of the week that are weekend days and are not included in

This screenshot shows the same DAX documentation page for the `NETWORKDAYS` function, but with the "weekend" parameter expanded. The expanded definition includes a table of weekend numbers (1 through 17) corresponding to specific days of the week: 1 for Saturday/Sunday, 2 for Sunday/Monday, 3 for Monday/Tuesday, 4 for Tuesday/Wednesday, 5 for Wednesday/Thursday, 6 for Thursday/Friday, 7 for Friday/Saturday, 11 for Sunday only, 12 for Monday only, 13 for Tuesday only, 14 for Wednesday only, 15 for Thursday only, 16 for Friday only, and 17 for Saturday only.

This screenshot shows the "Return Value" section of the DAX documentation for the `NETWORKDAYS` function. It states that the function returns an integer number of whole workdays.

This screenshot shows the Power BI Data Editor interface. On the left, a DAX query editor contains the following code:

```
1 netdays = NETWORKDAYS (
2     SELECTEDVALUE(Orders[Order Date]),
3     SELECTEDVALUE(Orders[Ship Date])
4 )
```

On the right, the "Fields" pane is open, showing a hierarchy of fields: `2018 Sales`, `Customers`, `Locations`, `Orders`, `business_days`, `Category`, `City`, `Country/Region`, and `Cust_name`.

```

1 business_days = var order_date = SELECTEDVALUE(Orders[Order Date])
2             var ship_date = SELECTEDVALUE(Orders[Ship Date])
3             return
4             NETWORKDAYS(order_date, ship_date,1,'Public Holidays')

```

The screenshot shows the Power BI Data Editor interface. On the left is the DAX formula view, which contains the provided DAX code. On the right is the Fields pane, which lists various fields from the Orders table, including 'business_days'.

The screenshot shows the Power BI Report View. It displays a table visual titled "NETWORK DAYS IN POWER BI DAX". The table has three columns: Order Date, Ship Date, and business_days. A filter for Order Date is applied, showing the value "05 January 2018". The result in the table is 6 business days between the two dates.

Calendar and CalendarAuto in Power BI

```

1 Calendar = CALENDAR(#date(2022,09,01))

```

The screenshot shows the Power BI Data Editor interface. On the left is the DAX formula view, which contains the provided DAX code to create a calendar table. On the right is the Fields pane, which lists fields from the Orders table and also includes fields from the newly created Calendar table, such as 'Date' and 'Year'.

The screenshot shows a data visualization interface with a sidebar on the right containing a search bar and a list of fields. The fields listed are: Calendar, Date, Category, City, Country/Region, Customer ID, Customer Name, Discount, Order Date, Order ID, Postal Code, Product ID, Product Name, Profit, Profit ratio, Profit ratio measure, Profit ratio new, Quantity, Region, Row ID, and Sales.

Fields

Search

Calendar

Date

Category

City

Country/Region

Customer ID

Customer Name

Discount

Order Date

Order ID

Postal Code

Product ID

Product Name

Profit

Profit ratio

Profit ratio measure

Profit ratio new

Quantity

Region

Row ID

The screenshot shows a data visualization interface. On the left, there is a table with three columns and four rows. The first row contains the header "1 Calendar = CALENDAR(" followed by two formulas: "TODAY()-365," and "TODAY()+365)". The second row has three empty cells. The third row has three cells containing the numbers "2", "3", and "4". The fourth row has three empty cells. A dropdown menu above the table is set to "Date". To the right of the table is a sidebar titled "Fields" with a search bar and a list of fields categorized under "Calendar".

1	Calendar = CALENDAR(TODAY()-365, TODAY()+365)	2
3		4
Date		
03-09-2021 00:00:00		
04-09-2021 00:00:00		
05-09-2021 00:00:00		
06-09-2021 00:00:00		
07-09-2021 00:00:00		
08-09-2021 00:00:00		
09-09-2021 00:00:00		
10-09-2021 00:00:00		
11-09-2021 00:00:00		
12-09-2021 00:00:00		
13-09-2021 00:00:00		
14-09-2021 00:00:00		
15-09-2021 00:00:00		
16-09-2021 00:00:00		
17-09-2021 00:00:00		
18-09-2021 00:00:00		
19-09-2021 00:00:00		
20-09-2021 00:00:00		

Fields

- Search
- Calendar
- Orders
- Category
- City
- Country/Region
- Customer ID
- Customer Name
- Discount
- Order Date
- Order ID
- Postal Code
- Product ID
- Product Name
- Profit
- Profit ratio
- Profit ratio measure
- Profit ratio new
- Quantity
- Region

The screenshot shows the Power BI Data View interface. On the left, there's a code editor pane with the following DAX code:

```
1 Calendar = CALENDAR(  
2                         MIN(Orders[Order Date]),  
3                         MAX(Orders[Order Date]))  
4 )
```

Below the code is a preview table for the 'Date' column, displaying dates from 03-01-2018 to 18-01-2018.

On the right, there's a 'Fields' pane with a search bar and a list of fields categorized by table:

- Calendar**:
 - Date
- Orders**:
 - Category
 - City
 - Country/Region
 - Customer ID
 - Customer Name
 - Discount
 - Name 'Orders[Order Date]' (highlighted)
 - Order ID
 - Postal Code
 - Product ID
 - Product Name
 - Profit
 - Profit ratio
 - Profit ratio measure
 - Quantity
 - Region

1 Calendar = CALENDARAUTO([FiscalYearEndMonth])

This is a table with one column
values calculated from the
automatically.

Date
03-01-2018 00:00:00
04-01-2018 00:00:00
05-01-2018 00:00:00
06-01-2018 00:00:00
07-01-2018 00:00:00
08-01-2018 00:00:00
09-01-2018 00:00:00
10-01-2018 00:00:00
11-01-2018 00:00:00
12-01-2018 00:00:00
13-01-2018 00:00:00
14-01-2018 00:00:00
15-01-2018 00:00:00
16-01-2018 00:00:00

1 Calendar = CALENDARAUTO()

Date
01-01-2018 00:00:00
02-01-2018 00:00:00
03-01-2018 00:00:00
04-01-2018 00:00:00
05-01-2018 00:00:00
06-01-2018 00:00:00
07-01-2018 00:00:00
08-01-2018 00:00:00
09-01-2018 00:00:00
10-01-2018 00:00:00
11-01-2018 00:00:00
12-01-2018 00:00:00
13-01-2018 00:00:00
14-01-2018 00:00:00
15-01-2018 00:00:00
16-01-2018 00:00:00
17-01-2018 00:00:00
18-01-2018 00:00:00
19-01-2018 00:00:00
20-01-2018 00:00:00
21-01-2018 00:00:00
22-01-2018 00:00:00

Advanced Calender Creation (Including Financial Year)

```

1 Calendar =
2 ADDCOLUMNS (
3     CALENDARAUTO ( 3 ),
4     "MIndex", MONTH ( EDATE ( [Date], -3 ) ),
5     "CalMonth", FORMAT ( [Date], "mmm" ),
6     "CalQtr", "Q"
7     & CEILING ( MONTH ( [Date] ), 3 ) / 3,
8     "CalYear", YEAR ( [Date] ),
9     "FinQtr", "Q"
10    & CEILING ( MONTH ( EDATE ( [Date], -3 ) ), 3 ) / 3,
11    "FY",
12    VAR CY =
13    | RIGHT ( YEAR ( [Date] ), 2 )
14    VAR NY =
15    | RIGHT ( YEAR ( [Date] ) + 1, 2 )
16    VAR PY =
17    | RIGHT ( YEAR ( [Date] ) - 1, 2 )
18    VAR FinYear =
19    | IF ( MONTH ( [Date] ) > 3, CY & "-" & NY, PY & "-" & CY )
20    RETURN
21    | FinYear,
22    "CalWeekNo", WEEKNUM ( [Date], 2 ),
23    "Weekend/Working", IF ( WEEKDAY ( [Date], 2 ) > 5, "Weekend", "Working" ),
24    "Day", FORMAT ( [Date], "ddd" ),
25    "CustomDate", FORMAT ( [Date], "d/mm" )

```

Date	MIndex	CalMonth	CalQtr	CalYear	FinQtr	FY	CalWeekNo	Weekend/Working	Day	CustomDate	MonthYr
01-07-2020 00:00:00	4	Jul	Q3	2020	Q2	20-21	27	Working	Wed	1/07	Jul 2020
02-07-2020 00:00:00	4	Jul	Q3	2020	Q2	20-21	27	Working	Thu	2/07	Jul 2020
03-07-2020 00:00:00	4	Jul	Q3	2020	Q2	20-21	27	Working	Fri	3/07	Jul 2020

Table: Calendar (1,461 rows)

Lookupvalue in Power BI

If created using a measure, it gives 1 result at a time corresponding to given search value and can be visualized using a card

The screenshot shows the Power BI Data Editor interface. In the top ribbon, 'Measure tools' is selected. A measure named 'Measure 2' is being edited. The formula is: `LOOKUPVALUE(Result_ColumnName, Search_ColumnName1, Search_Value1, ... [Alternate_Result])`. The 'Home table' dropdown is set to 'Orders'. The 'Structure' pane shows the formula retrieves a value from the 'Orders' table. On the right, the 'Fields' pane lists various tables and their columns, including 'Orders' (business_days, Category, City, Country/Region, Customer ID n...), '2018 Sales', 'Customers', and 'Locations'.

This screenshot shows the same Power BI Data Editor interface as above, but with a different formula: `1 Cust_name = LOOKUPVALUE(Orders[Customer Name],Orders[Customer ID new],19885)`. A yellow warning bar at the top indicates: "Argument '3' in LOOKUPVALUE function is required." The 'Fields' pane on the right shows the 'Orders' table with its columns, and the 'Cust_name' column is highlighted.

This screenshot shows a Power BI report. At the top, the title is 'LOOKUPVALUE() IN POWER BI DAX'. Below it is a card visualization containing the text 'Ruben Ausman' and 'Cust_name'. The 'Fields' pane on the right shows the 'Orders' table with its columns, and the 'Cust_name' column is highlighted. The 'Visualizations' pane shows various chart and card icons.

If created using a column works same as vlookup in excel

The screenshot shows the Power BI Advanced Editor interface. In the top ribbon, 'Column tools' is selected. A formula bar at the top contains the DAX code: `1 Location_name = LOOKUPVALUE(Location[Name],Locations[Location ID],'2018 Sales'[location ID])`. Below the formula bar, the 'Structure' pane shows the query definition. The main workspace displays a card visual for 'John Lee' with the text 'Cust_name'. On the right side, the 'Fields' pane lists fields from the '2018 Sales' table, including Customer ID, location, Location ID, Order ID, Price, Product ID, Purchase Date, Quantity, Sales Person ID, Customers, Locations, Orders, People2, Products, and Public Holidays. The 'Visualizations' pane shows various chart and report icons.

This screenshot shows the Power BI Advanced Editor with a table visual titled 'LOOKUPVALUE() IN POWER BI DAX'. The table displays data from the 'Locations' table, with columns 'Location ID' and 'Location_name'. The 'Customer ID' and 'location' columns are highlighted with yellow boxes. The 'Fields' pane on the right lists the same fields as the previous screenshot, with the 'Customer ID' and 'location' fields highlighted by a red oval. The 'Visualizations' pane is also visible.

What is M (Mashup) Language in Power BI?

The steps taken during the data transformation stage are written in M language, the code is generated automatically.

M language is used to the transformation which are not possible with the default options given (for e.g. deleting custom rows)

Home -> Query Group -> Advanced Editor

Code is generated for all applied steps on right hand side

There are two main keywords **let** and **in**

Code is written inside **let** and ends with last step inside **in**

Each step from 2nd step onwards contains reference of previous step

```

let
    Source = Excel.Workbook(File.Contents("C:/Users/salla/OneDrive/Documents/PBI Training/Sample - Superstore.xls"), null, true),
    Orders1 = Source[[Name="Orders2"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(Orders1,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}, {"Column5", type text}})
in
    #"Changed Type"
  
```

How to delete custom rows in Power BI?

Home -> Query Group -> Advanced Editor

Advanced Editor

Orders

```

let
    Source = Excel.Workbook(File.Contents("C:\Users\saila\OneDrive\Documents\PBI Training\Sample - Superstore.xls"), null, true),
    Orders1 = Source[[Name="Orders2"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(Orders1,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}}),
    #"Removed Top Rows" = Table.Skip(#"Changed Type",4),
    #"Promoted Headers" = Table.PromoteHeaders(#"Removed Top Rows", [PromoteAllScalars=true]),
    #"Changed Type1" = Table.TransformColumnTypes(#"Promoted Headers",{{"Customer Name", type text}, {"Segment", type text}, {"Country/Region", type text}, {"Order Date", type date}}),
    #"Remove Custom Rows" = Table.RemoveRows(#"Changed Type1",5,7)
in
    #"Changed Type1"

```

No syntax errors have been detected.

Done

Advanced Editor

Orders

```

let
    Source = Excel.Workbook(File.Contents("C:\Users\saila\OneDrive\Documents\PBI Training\Sample - Superstore.xls"), null, true),
    Orders1 = Source[[Name="Orders2"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(Orders1,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}}),
    #"Removed Top Rows" = Table.Skip(#"Changed Type",4),
    #"Promoted Headers" = Table.PromoteHeaders(#"Removed Top Rows", [PromoteAllScalars=true]),
    #"Changed Type1" = Table.TransformColumnTypes(#"Promoted Headers",{{"Customer Name", type text}, {"Segment", type text}, {"Country/Region", type text}, {"Order Date", type date}}),
    #"Remove Custom Rows" = Table.RemoveRows(#"Changed Type1",5,7)
in
    #"Changed Type1"

```

Table.RemoveRows(table as table, offset as number, count as nullable number)
 nullable number
 Removes the specified number of rows.

A Comma cannot precede a RightParen. Show error

Done Cancel

Orders

```

let
    Source = Excel.Workbook(File.Contents("C:\Users\saila\OneDrive\Documents\PBI Training\Sample - Superstore.xls"), null, true),
    Orders1 = Source[[Name="Orders2"]][Data],
    #"Changed Type" = Table.TransformColumnTypes(Orders1,{{"Column1", type text}, {"Column2", type text}, {"Column3", type text}, {"Column4", type text}}),
    #"Removed Top Rows" = Table.Skip(#"Changed Type",4),
    #"Promoted Headers" = Table.PromoteHeaders(#"Removed Top Rows", [PromoteAllScalars=true]),
    #"Changed Type1" = Table.TransformColumnTypes(#"Promoted Headers",{{"Customer Name", type text}, {"Segment", type text}, {"Country/Region", type text}, {"Order Date", type date}}),
    #"Remove Custom Rows" = Table.RemoveRows(#"Changed Type1",5,7)
in
    #"Remove Custom Rows"

```

No syntax errors have been detected.

Done Cancel

Rolling Average

DatesinPeriod: Returns the dates between the specified start dates, no of intervals & type of interval
 Minx: Returns the smallest value by evaluating expression on each row of table

The screenshot shows the Power BI Data Editor interface. On the left is the DAX editor pane, and on the right is the Fields pane.

DAX Editor:

```

1 Sales Avg =
2 VAR NumofMonths = 12
3 VAR LastCurrentDate = MAX('Date'[Date])
4 VAR Period = DATESINPERIOD('Date'[Date],LastCurrentDate,-NumofMonths,MONTH)
5 VAR Result =
6     CALCULATE(
7         AVERAGEX(
8             VALUES('Date'[Calendar Year Month Number]),
9             Sales[Sales Amount]
10        ),
11        Period
12    )
13 VAR FirstDateinPeriod = MINX(Period, 'Date'[Date])
14 VAR LastDatewithSales = MAX(Sales[Order Date])
15 Return
16 if (FirstDateinPeriod <= LastDatewithSales, Result)
Total | 3,05,91,343.98

```

Fields Pane:

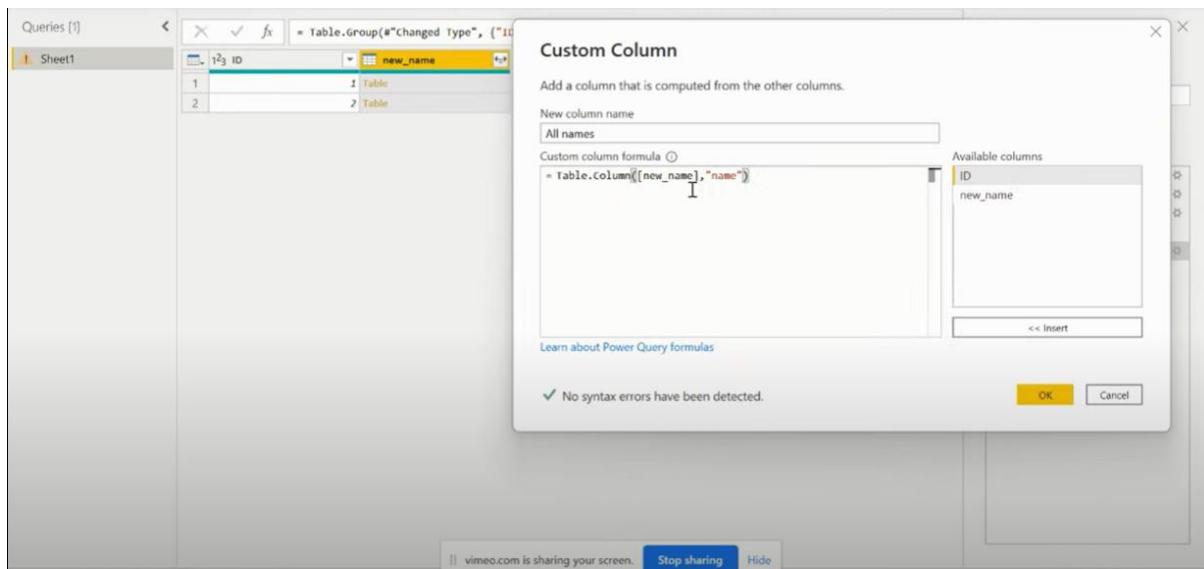
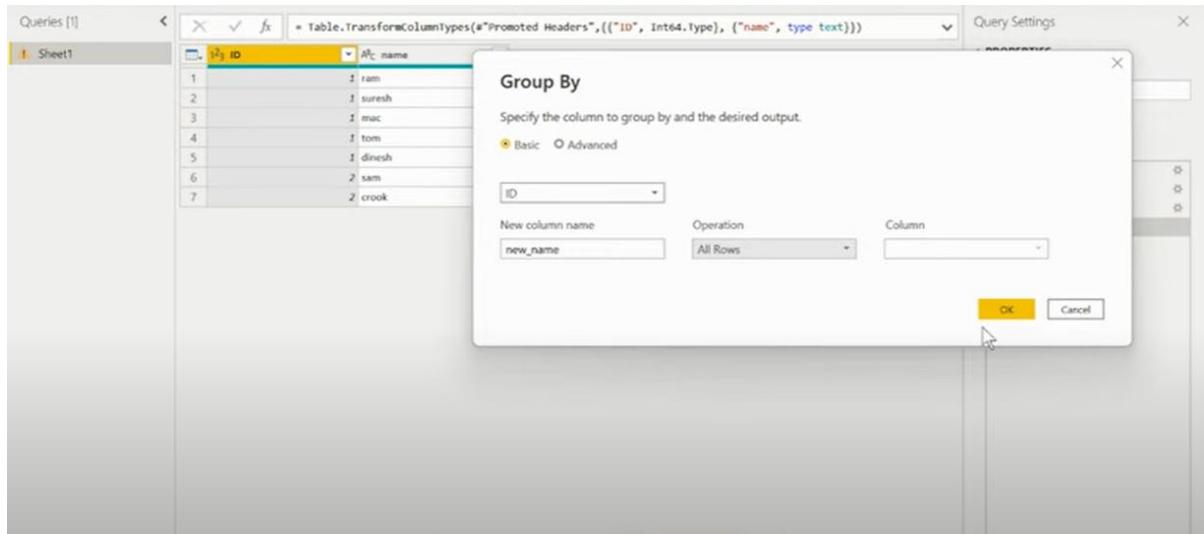
- Sales:**
 - Rolling 12 mont...
 - Sales Amount
 - Sales Avg** (selected)
 - Sales R12M day ...
- Date:**
 - Calendar Year
 - Calendar Year M...
 - Date
 - Day of Week
 - Month

Power BI Report Visual:

Calendar Year	Sales Amount	Sales Avg
CY 2007	1,13,09,946.12	9,42,495.51
January	7,94,248.24	7,94,248.24
February	8,91,135.91	8,42,692.08
March	9,61,289.24	8,82,224.46
April	11,28,104.82	9,43,694.55
May	9,36,192.74	9,42,194.19
June	9,82,304.46	9,48,879.23
July	9,22,542.98	9,45,116.91
August	9,52,834.59	9,46,081.62
September	10,09,868.98	9,53,169.11
October	9,14,273.54	9,49,279.55
November	8,25,601.87	9,38,036.12
December	9,91,548.75	9,42,495.51
CY 2008	99,27,582.99	8,27,298.58
January	6,56,766.69	9,31,038.71
February	6,00,080.00	9,06,784.06
March	5,50,528.52	8,73,204.82
Total	3,05,91,343.98	

How to convert rows to comma separated values in Power BI

Go to Home -> Transform Data -> Select Table -> Home -> Groupby -> Specify New Column Name -> Operation -> All rows -> Add Column -> Custom Column -> Specify New Column Name -> Use Table.Column -> New Column name as Table (Newly created column during groupby) -> original column name as column -> Expand values -> Extract Values -> Use comma as delimiter -> Click Ok



The screenshot shows the Power Query Editor interface. On the left, there's a 'Queries [1]' pane with 'Sheet1'. In the center, a table is displayed with columns: 'ID', 'new_name', and 'All names'. The 'All names' column contains two entries: 'Table' and 'List'. A context menu is open over the 'All names' column, with the 'Extract Values...' option highlighted. On the right, there's a 'Query Settings' pane with sections for 'PROPERTIES' (Name: Sheet1) and 'APPLIED STEPS' (which includes 'Source', 'Navigation', 'Promoted Headers', 'Changed Type', 'Grouped Rows', and 'Added Custom').

The screenshot shows the 'Extract values from list' dialog box. It asks to 'Select a delimiter to use for concatenating list values' with a dropdown set to 'Comma'. There are 'OK' and 'Cancel' buttons at the bottom.

Show/Hide Visual based on Slicer Selection in Power BI

Create a measure -> select visual -> add measure to the filter on this visual -> value is 1 -> apply filter

`Isfilter`: Checks if any filter is applied on selected column (output is Boolean T/F)

`Int`: converts Boolean output into 1/0

Note: If no category is selected in slicer, then visual will be blank

Show/Hide Visual based on slicer selection

The screenshot shows a Power BI interface. On the left, there is a slicer titled "Category" with options: Furniture, Office Supplies, and Technology. To the right of the slicer is a bar chart titled "Sales by Sub-Category". The chart has "Sub-Category" on the Y-axis and "Sales" on the X-axis. The bars represent sales for various sub-categories like Phones, Chairs, Storage, etc. The X-axis ranges from 0.0M to 0.3M. The Power BI ribbon is visible at the top.

1 Measure 2 = int(ISFILTERED(Orders[Category]))

This screenshot shows a Power BI visualization window. The title bar says "1 Measure 2 = int(ISFILTERED(Orders[Category]))". Below it is a bar chart titled "Sales by Sub-Category". The chart displays sales for different sub-categories. The Power BI ribbon is visible at the top.

Filters

This screenshot shows the "Filters" pane in Power BI. It lists two filters: "Measure 2" (set to "is (All)") and "Sales" (set to "is (All)"). The pane also includes sections for "Show items when the value" and "Add data fields here". The Power BI ribbon is visible at the top.

Web Scrapping Example 1 (Extracting tables from multiple pages of html link)

Want to extract Lat long data for us cities from below web link. This link has total of 1139 records. The records are spanned over 12 pages with each page containing 100 records (except last).

<https://www.latlong.net/category/cities-236-15.html>

The screenshot displays two consecutive pages from the website [LatLong.net](https://www.latlong.net/category/cities-236-15.html).

Page 1: Shows a list of cities in the United States. The table below lists the first few entries:

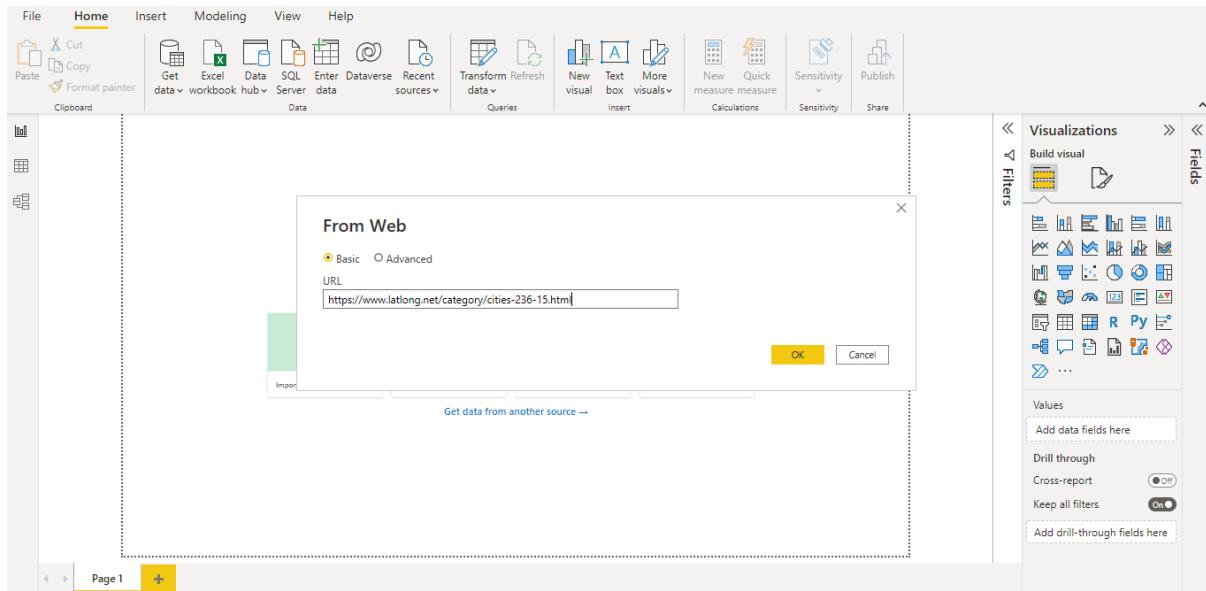
Place Name	Latitude	Longitude
Peabody, MA, USA	42.536457	-70.985786
Northampton, MA, USA	42.328674	-72.664658

Page 2: Shows a detailed view of a single city record for Valparaiso, IN, USA. The coordinates listed are 41.483845 and -87.063965.

Both pages include standard web navigation elements like back, forward, and search, along with Google AdSense banners.

Step 1:

Go To Home -> Get Data -> Web -> Paste the URL -> Click OK -> Select Table -> click Transform data



File Home Insert Modeling View Help

Cut Copy Format painter Get data Data SQL Enter Data Transform data New visual Text box More visuals Insert Calculations Sensitivity Share

Navigator

Display Options ▾

- HTML Tables [3]
 - Table 1
 - Table 2
 - Table 3
- Suggested Tables [1]
 - Table 4
- Text [2]
 - HTML Code
 - Displayed Text

Add Table Using Examples

Table View Web View

Table 3

Place Name	Latitude	Longitude
Peabody, MA, USA	42.536457	-70.985786
Northampton, MA, USA	42.328674	-72.664658
Newton, MA, USA	42.341042	-71.217133
Newburyport, MA, USA	42.810356	-70.893875
New Bedford, MA, USA	41.638409	-70.941208
Medford, MA, USA	42.419331	-71.11972
Malden, MA, USA	42.429752	-71.071022
Leominster, MA, USA	42.525482	-71.764183
Lawrence, MA, USA	42.701283	-71.175682
Holyoke, MA, USA	42.203217	-72.625481
Greenfield, MA, USA	42.587334	-72.609416
Framingham, MA, USA	42.280418	-71.423233
Fitchburg, MA, USA	42.586716	-71.814468
Everett, MA, USA	42.408623	-71.056999
Chelsea, MA, USA	42.392925	-71.037109
Amesbury, MA, USA	42.856842	-70.96344
Takoma Park, MD, USA	38.981544	-77.010674

Load Transform Data Cancel

File Home Insert Modeling View Help

Cut Copy Format painter Get data Data SQL Enter Data Transform data New visual Text box More visuals Insert Calculations Sensitivity Share

Queries [1]

Table 3

Place Name Latitude Longitude

Peabody, MA, USA	42.536457	-70.985786
Northampton, MA, USA	42.328674	-72.664658
Newton, MA, USA	42.341042	-71.217133
Newburyport, MA, USA	42.810356	-70.893875
New Bedford, MA, USA	41.638409	-70.941208
Medford, MA, USA	42.419331	-71.11972
Malden, MA, USA	42.429752	-71.071022
Leominster, MA, USA	42.525482	-71.764183
Lawrence, MA, USA	42.701283	-71.175682
Holyoke, MA, USA	42.203217	-72.625481
Greenfield, MA, USA	42.587334	-72.609416
Framingham, MA, USA	42.280418	-71.423233
Fitchburg, MA, USA	42.586716	-71.814468
Everett, MA, USA	42.408623	-71.056999
Chelsea, MA, USA	42.392925	-71.037109
Amesbury, MA, USA	42.856842	-70.96344
Takoma Park, MD, USA	38.981544	-77.010674
Salisbury, MD, USA	38.36335	-75.605919
Rockville, MD, USA	39.086437	-77.161263
Hagerstown, MD, USA	39.644207	-77.73143
Greenbelt, MD, USA	38.998318	-76.896332
Cumberland, MD, USA	39.649109	-78.769714
Cambridge, MD, USA	38.563461	-76.085251
Aberdeen, MD, USA	39.514877	-76.174111

Close & Apply Close New Source Recent Sources Enter Data Data source settings Manage Parameters Refresh Query Advanced Editor Choose Columns Remove Columns Keep Rows Remove Rows Sort Split Column Group By Use First Row as Headers Transform Merge Queries Append Queries Combine Files Text Analytics Vision Azure Machine Learning AI Insights

Query Settings

PROPERTIES

Name: Table 3

All Properties

APPLIED STEPS

Source: Extracted Table From HTML
Promoted Headers
Changed Type

Step 2:

Go To Home -> Query Group -> Advanced Editor -> Define variable **PageNo as text** before let command (text because web link is text string) -> replace Page number in the link with “**&PageNo&**” (this will ensure only variable will change, rest of the link will be same) -> click Done -> it will be saved as fn -> rename the fn (fnLocations)

(In the link /cities-236-15.html is similar to /cities-236-15-1.html ---- for 1st page, similarly for 2nd page ---- /cities-236-15-1.html, we will change this no to extract data from all pages)

The screenshot shows two instances of the Power BI Advanced Editor interface.

Initial Query (Top Window):

```
let
    Source = Web.BrowserContents("https://www.latlong.net/category/cities-236-15.html"),
    #Extracted Table From Html1 = Html.Table(Source, {"Column1", "TABLE:nth-child(4) > * > TR > :nth-child(1)"}, {"Column2", "TABLE:nth-child(4) > * > TR > :nth-child(2)"})
in
    #Changed Type
```

Modified Query (Bottom Window):

```
(PageNo as text) =>
let
    Source = Web.BrowserContents("https://www.latlong.net/category/cities-236-15-&PageNo#.html"),
    #Extracted Table From Html1 = Html.Table(Source, {"Column1", "TABLE:nth-child(4) > * > TR > :nth-child(1)"}, {"Column2", "TABLE:nth-child(4) > * > TR > :nth-child(2)"})
in
    #Changed Type
```

In the modified query, the URL contains the variable `&PageNo#`. Below the queries, a preview table shows four rows of data:

21	Greenbelt, MD, USA	38.998318	-76.896332
22	Cumberland, MD, USA	39.649109	-78.769714
23	Cambridge, MD, USA	38.563461	-76.085251
24	Aberdeen, MD, USA	39.514877	-76.17411

The screenshot shows the Power Query Editor interface. On the left, there's a tree view with 'fnLocations' selected. In the center, there's a 'Enter Parameter' dialog with a 'PageNo' field containing 'abc'. Below it is a code editor with the following text:

```
function (PageNo as text) as any
```

On the right, the 'Query Settings' pane shows the properties for 'fnLocations': Name is 'fnLocations' and the applied step is 'fnLocations'.

Step 3:

Go To Home -> New Source -> Blank Query -> Type = {1..12} -> It will create a list of 12 numbers -> File -> Convert Group -> To Table -> Click Ok -> Table with single column will be created -> change data type of this column to **text**

The screenshot shows the Power Query Editor with a list of numbers from 1 to 12. The 'Applied Steps' pane shows 'Source' as the current step. The 'Properties' pane shows 'Name' is 'Query1'.

Value	Value
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12

The screenshot shows the Power Query Editor with the 'To Table' dialog open. The dialog instructions say 'Create a table from a list of values.' It has fields for 'Select or enter delimiter' (set to 'None') and 'How to handle extra columns' (set to 'Show as errors'). The 'OK' button is highlighted.

Step 4:

Go To Add Column -> Invoke Custom Function -> select function quey name (fnLocation) -> Click Ok
-> Expand -> Select Required Columns -> Click OK -> Delete Column 1

Queries [2]

fx fnLocations

Query1

= Table.AddColumn(#"Changed Type", "fnLocations", each fnLocations([Column1]))

ABC Column1 ABC fnLocations

Expand Aggregate
 (Select All Columns)
 Place Name
 Latitude
 Longitude

Use original column name as prefix
List may be incomplete. Load more

OK Cancel

Query Settings

PROPERTIES
Name: Query1
All Properties

APPLIED STEPS
Source
Converted to Table
Changed Type
Invoked Custom Function

Queries [2]

fx fnLocations

Query1

= Table.ExpandTableColumn(#"Invoked Custom Function", "fnLocations", {"Place Name", "Latitude", "Longitude"})

ABC Column1 ABC Place Name ABC Latitude ABC Longitude

		Place Name	Latitude	Longitude
1	1	Peabody, MA, USA	42.536457	-70.985786
2	1	Northampton, MA, USA	42.328674	-72.664658
3	1	Newton, MA, USA	42.341042	-71.217133
4	1	Newburyport, MA, USA	42.810356	-70.893875
5	1	New Bedford, MA, USA	41.638409	-70.941208
6	1	Medford, MA, USA	42.419331	-71.11972
7	1	Malden, MA, USA	42.429752	-71.071022
8	1	Leominster, MA, USA	42.525482	-71.764183
9	1	Lawrence, MA, USA	42.701283	-71.175682
10	1	Holyoke, MA, USA	42.203217	-72.625481
11	1	Greenfield, MA, USA	42.587334	-72.603416
12	1	Framingham, MA, USA	42.280418	-71.423233
13	1	Fitchburg, MA, USA	42.586716	-71.814468
14	1	Everett, MA, USA	42.408623	-71.056999
15	1	Chelsea, MA, USA	42.392925	-71.037109
16	1	Amesbury, MA, USA	42.856842	-70.96344
17	1	Takoma Park, MD, USA	38.981544	-77.010674
18	1	Salisbury, MD, USA	38.36335	-75.605919
19	1	Rockville, MD, USA	39.086437	-77.161263
20	1	Hagerstown, MD, USA	39.64207	-77.73143
21	1	Greenbelt, MD, USA	38.998318	-76.896332

Query Settings

PROPERTIES
Name: Query1
All Properties

APPLIED STEPS
Source
Converted to Table
Changed Type
Invoked Custom Function
Expanded fnLocations

Queries [2]

fx fnLocations

Query1

= Table.RemoveColumns(#"Expanded fnLocations", {"Column1"})

ABC Place Name ABC Latitude ABC Longitude

	Place Name	Latitude	Longitude
1	Peabody, MA, USA	42.536457	-70.985786
2	Northampton, MA, USA	42.328674	-72.664658
3	Newton, MA, USA	42.341042	-71.217133
4	Newburyport, MA, USA	42.810356	-70.893875
5	New Bedford, MA, USA	41.638409	-70.941208
6	Medford, MA, USA	42.419331	-71.11972
7	Malden, MA, USA	42.429752	-71.071022
8	Leominster, MA, USA	42.525482	-71.764183
9	Lawrence, MA, USA	42.701283	-71.175682
10	Holyoke, MA, USA	42.203217	-72.625481
11	Greenfield, MA, USA	42.587334	-72.603416
12	Framingham, MA, USA	42.280418	-71.423233
13	Fitchburg, MA, USA	42.586716	-71.814468
14	Everett, MA, USA	42.408623	-71.056999
15	Chelsea, MA, USA	42.392925	-71.037109
16	Amesbury, MA, USA	42.856842	-70.96344
17	Takoma Park, MD, USA	38.981544	-77.010674
18	Salisbury, MD, USA	38.36335	-75.605919
19	Rockville, MD, USA	39.086437	-77.161263

Query Settings

PROPERTIES
Name: Query1
All Properties

APPLIED STEPS
Source
Converted to Table
Changed Type
Invoked Custom Function
Expanded fnLocations
Removed Columns

Go To Transform -> Table Group -> Count Rows (To match the records from web link)

Queries [2]

fx fnLocations

Query1

= Table.RowCount(#"Removed Columns")

1139

Query Settings

PROPERTIES
Name: Query1
All Properties

APPLIED STEPS
Source
Converted to Table
Changed Type
Invoked Custom Function
Expanded fnLocations
Removed Columns
Counted Rows

Web Scrapping Example 2 (Extracting Custom data from multiple pages of web link)

Want to extract book name, author, book type etc. from below web link.

<https://www.worldcat.org/search?q=Eysenck&limit=10&offset=1>

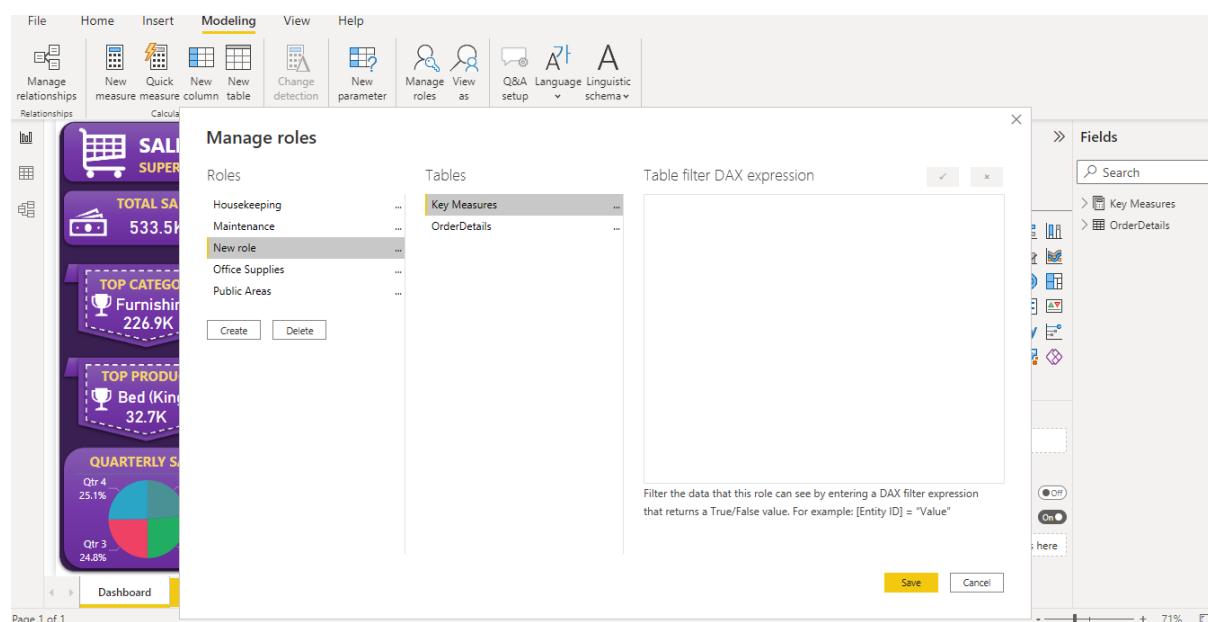
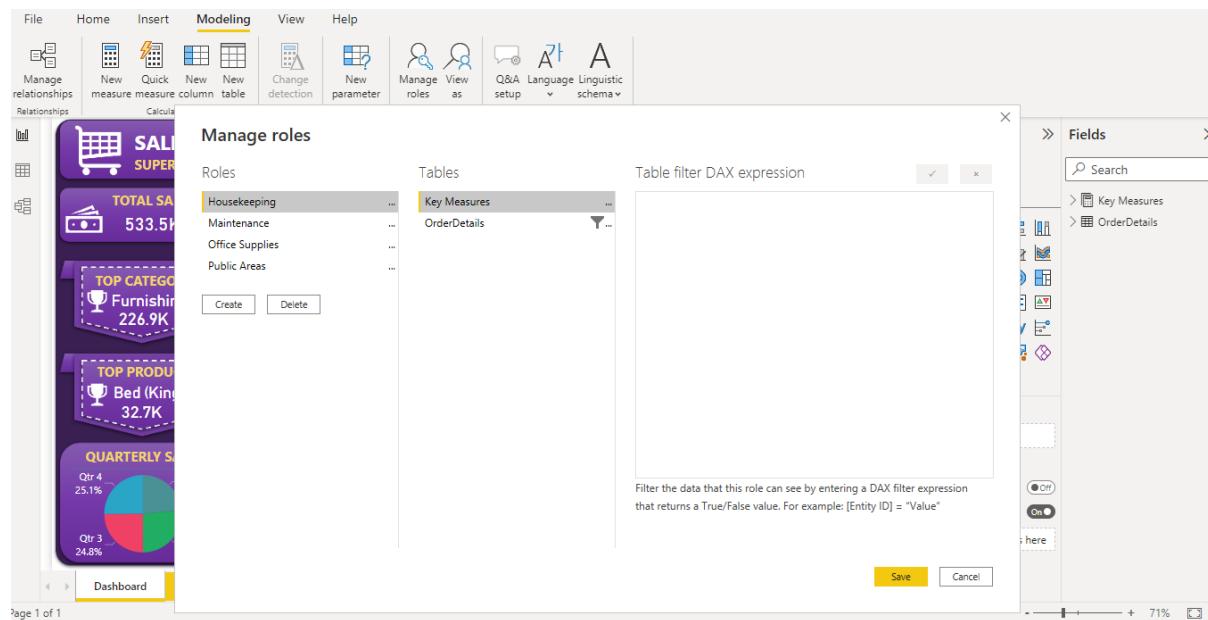
The screenshot shows the WorldCat search results for the query 'Eysenck'. The search bar at the top contains 'Eysenck'. Below the search bar, there are navigation links for Home, Libraries, Topics, Lists, About, and For Librarians. A location dropdown shows 'Mumbai, India'. The main content area displays 1-10 of 13,422 results. On the left, a sidebar provides filtering options for Open Content (Open Access), Format (Book, Article, Book), and Author/Creator (Eysenck H J, Eysenck, H. J.). The first result is a book titled 'Playing with fire : the controversial career of Hans J. Eysenck' by Roderick D. Buchanan. It includes a thumbnail image, the title, author, summary, and publication details: 'Print Book, 2010 English Publisher: Oxford University Press, Oxford, 2010'. Below the main result, there is a link to 'View All Formats & Editions'.

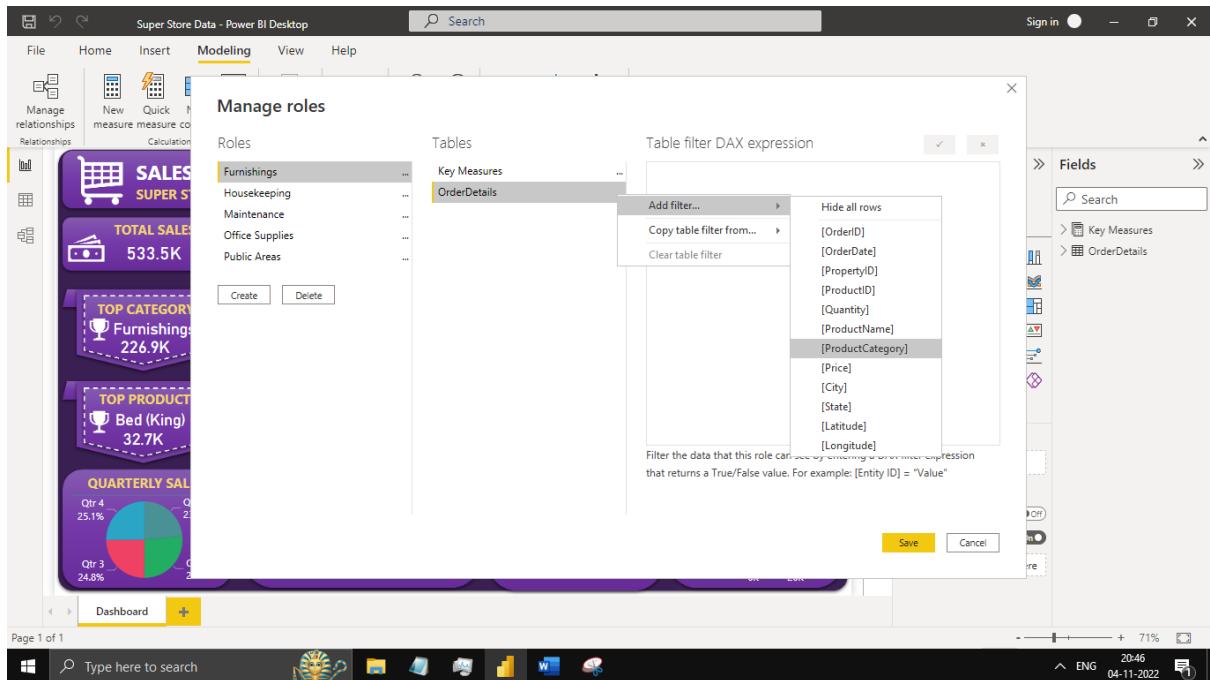
The screenshot shows the Microsoft Power BI interface with the 'From Web' import dialog open. The dialog has 'Basic' selected and the URL 'https://www.worldcat.org/search?q=Eysenck&limit=10&offset=1' entered. The Power BI ribbon is visible at the top, showing tabs for File, Home, Insert, Modeling, View, and Help. The 'Data' tab is selected. The right side of the screen shows the 'Visualizations' pane with various chart and table icons, and the 'Fields' pane which is currently empty. The 'Values' section contains a placeholder 'Add data fields here'. The 'Drill through' and 'Keep all filters' options are also present.

Row Level Security

It is done to show only relevant data to the end user (for e.g. giving access of category wise data to the person to lead that category)

Go To Modelling -> Manage Roles -> Create -> Rename Role (**Category**) -> Select Table -> Add Filter -> Select Field -> Change value as **Category Name** -> Save





Manage roles

Roles

Furnishings
Housekeeping
Maintenance
Office Supplies
Public Areas

Tables

Key Measures
OrderDetails

Table filter DAX expression

```
[ProductCategory] = "Value"
```

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

Save Cancel

Manage roles

Roles

- Furnishings
- Housekeeping
- Maintenance
- Office Supplies
- Public Areas

Tables

- Key Measures
- OrderDetails

Table filter DAX expression

```
[ProductCategory] = "Furnishings"
```

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

Save **Cancel**

Creating multiple roles with single piece of code (for city)

Go To Modelling -> Manage Roles -> Create -> Rename Role (**City**) -> Select Table -> Add Filter -> Select Field -> Change value as **UserPrincipalName()** -> Save

Super Store Data - Power BI Desktop

Modeling

Manage roles

Roles

- City
- Furniture
- Housekeeping
- Maintenance
- Office Supplies
- Public Areas

Tables

- Key Measures
- OrderDetails

Table filter DAX expression

Add filter... Hide all rows
 Copy table filter from...
 Clear table filter

[OrderID]
[OrderDate]
[PropertyID]
[ProductID]
[Quantity]
[ProductName]
[ProductCategory]
[Price]
[City]
[State]
[Latitude]
[Longitude]

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

Fields

Key Measures
OrderDetails

Save **Cancel**

Manage roles

Roles

City	...
Furniture	...
Housekeeping	...
Maintenance	...
Office Supplies	...
Public Areas	...

[Create](#) [Delete](#)

Tables

Key Measures	...
OrderDetails	...

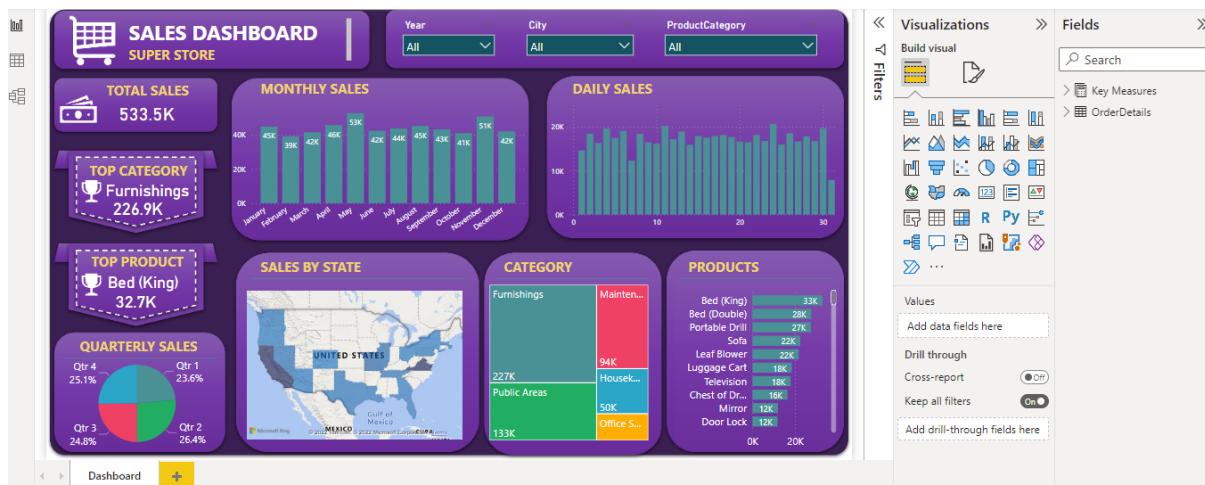
Table filter DAX expression

```
[City] = UserPrincipalName()
```

Filter the data that this role can see by entering a DAX filter expression that returns a True/False value. For example: [Entity ID] = "Value"

[Save](#) [Cancel](#)

Dashboard – Sales for all categories



To View from the point of end user

Go To Modelling -> View as -> Select Role (**Furnishings**) -> Click Ok

Sales Dashboard - Super Store

TOTAL SALES: 533.5K

MONTHLY SALES

TOP CATEGORY: Furnishings 226.9K

TOP PRODUCT: Bed (King) 32.7K

QUARTERLY SALES

SALES BY STATE

View as roles

- None
- Other user
- City
- Furnishings
- Housekeeping
- Maintenance
- Office Supplies
- Public Areas

Fields

Build visual

Filters

Search

Key Measures

OrderDetails

Data is filtered as per Furnishing Category and values are updated accordingly in Power View as well as data view.

SUPER STORE

TOTAL SALES: 226.9K

MONTHLY SALES

DAILY SALES

TOP CATEGORY: Furnishings 226.9K

TOP PRODUCT: Bed (King) 32.7K

SALES BY STATE

CATEGORY

PRODUCTS

View as roles

Fields

Build visual

Filters

Search

Key Measures

OrderDetails

Fields

OrderId	OrderDate	PropertyID	ProductID	Quantity	ProductName	ProductCategory	Price	City	State	Latitude	Longitude
421	03-03-2015	19	48	3	Side Table	Furnishings	\$47.608	-118.243683	-84.38633		
413	03-03-2015	11	49	3	Large Vase	Furnishings	\$28K	-122.335167			
394	28-02-2015	4	36	3	Armoire	Furnishings	\$18K	-122.431297			
387	27-02-2015	20	33	3	Towel Rack	Furnishings	\$16K	-87.623177			
385	27-02-2015	15	45	3	Computer Desk	Furnishings	\$14K	-71.057083			
369	25-02-2015	7	38	3	Bed (Double)	Furnishings	\$12K	-122.431297			
350	22-02-2015	20	54	3	Chest of Drawers	Furnishings	\$11K	-115.172813			
340	21-02-2015	17	36	3	Armoire	Furnishings	\$10K	-81.379234			
336	21-02-2015	6	43	3	Table Lamp	Furnishings	\$8K	-104.991531			
303	17-02-2015	16	35	3	Tissue Box	Furnishings	\$7K	-71.057083			
269	13-02-2015	7	49	3	Large Vase	Furnishings	\$6K	-94.578331			
249	10-02-2015	5	49	3	Large Vase	Furnishings	\$5K	-115.172813			
239	09-02-2015	17	34	3	Shower Curtain	Furnishings	\$4K	-122.335167			
238	09-02-2015	4	36	3	Armoire	Furnishings	\$3K	-104.991531			
221	04-02-2015	1	53	3	Iron	Furnishings	\$2K	-73.935242			
213	03-02-2015	3	33	3	Towel Rack	Furnishings	\$1K	-122.676483			
205	02-02-2015	1	42	3	Reading Chair	Furnishings	\$1K	-70. New York City	New York	\$40.7306	-\$73.935242
178	27-01-2015	2	52	3	Ironing Board	Furnishings	\$1K	18. Cincinnati	Ohio	\$39.1031	-\$44.512016
158	24-01-2015	14	37	3	Bed (King)	Furnishings	\$1K	300. Nashville	Tennessee	\$36.1745	-\$66.76796
155	24-01-2015	1	52	3	Ironing Board	Furnishings	\$1K	18. New York City	New York	\$40.7306	-\$73.935242
153	24-01-2015	17	50	3	Small Vase	Furnishings	\$1K	33. Las Vegas	Nevada	\$36.1146	-\$15.172813

Fields

Build visual

Filters

Search

Key Measures

OrderDetails

City

Longitude

OrderDate

OrderId

Price

ProductCategory

ProductID

ProductName

PropertyID

Quantity

State

Go To Modelling -> View as -> Select Role (City) -> Select Role (City Name: Orlando) -> Click Ok

The screenshot shows the Power BI interface with the 'SALES DASHBOARD' selected. A modal window titled 'View as roles' is open, showing a list of roles and their selection status. The 'City' role is selected, and 'Orlando' is entered in the input field. The dashboard itself includes sections for Total Sales (533.5K), Monthly Sales (with a bar chart showing values from 40K to 53K for Jan to Sep), Top Category (Furnishings, 226.9K), Top Product (Bed (King), 32.7K), Quarterly Sales (a pie chart showing Qtr 1: 23.6%, Qtr 2: 26.4%, Qtr 3: 24.8%, Qtr 4: 25.1%), Sales by State (a map of the US), and a table of products and their quantities.

Data is filtered as per Orlando City and values are updated accordingly in Power View as well as data view

The screenshot shows the same Sales Dashboard after applying the Orlando filter. The monthly sales chart now only displays data for Orlando, with values ranging from 0K to 3.0K. The products table has been updated to show items available in Orlando, such as Bed (King), Sofa, Luggage Cart, and so on.

The screenshot shows the 'OrderDetails' table in Power BI. A filter dialog is open over the table, specifically for the 'City' column. The option 'Orlando' is selected. The table lists 238 rows of order details, including columns for OrderID, OrderDate, PropertyID, ProductID, Quantity, ProductName, ProductCategory, Price, City, State, Latitude, and Longitude.

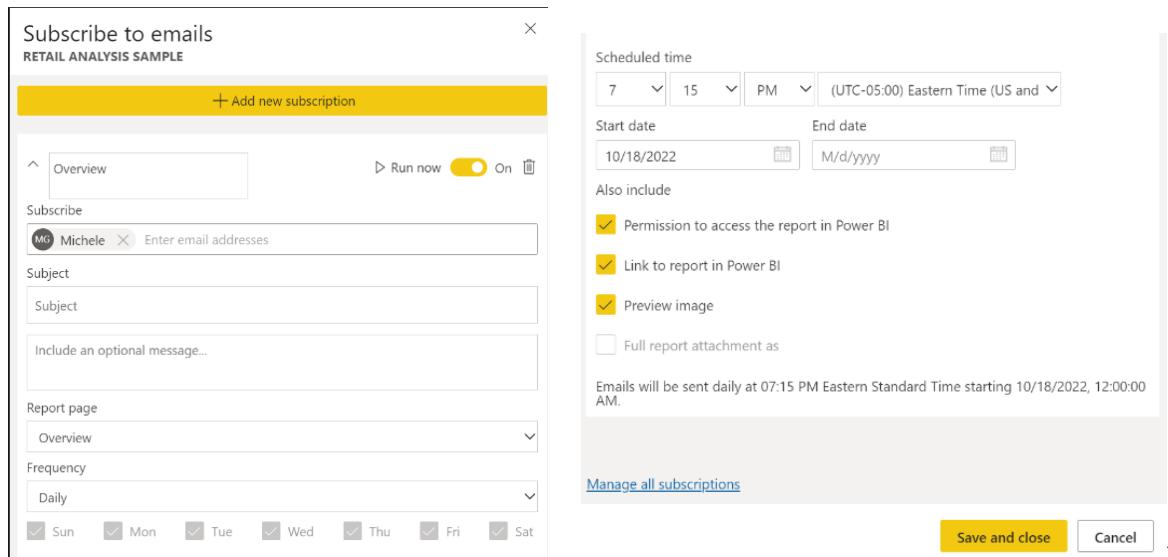
Workspace

Published reports can be saved in the required workspace.

Subscription

To share the report in image format on mail (in case of single page)

And in PDF/PPT format (in case of multiple page), same option should be ticked

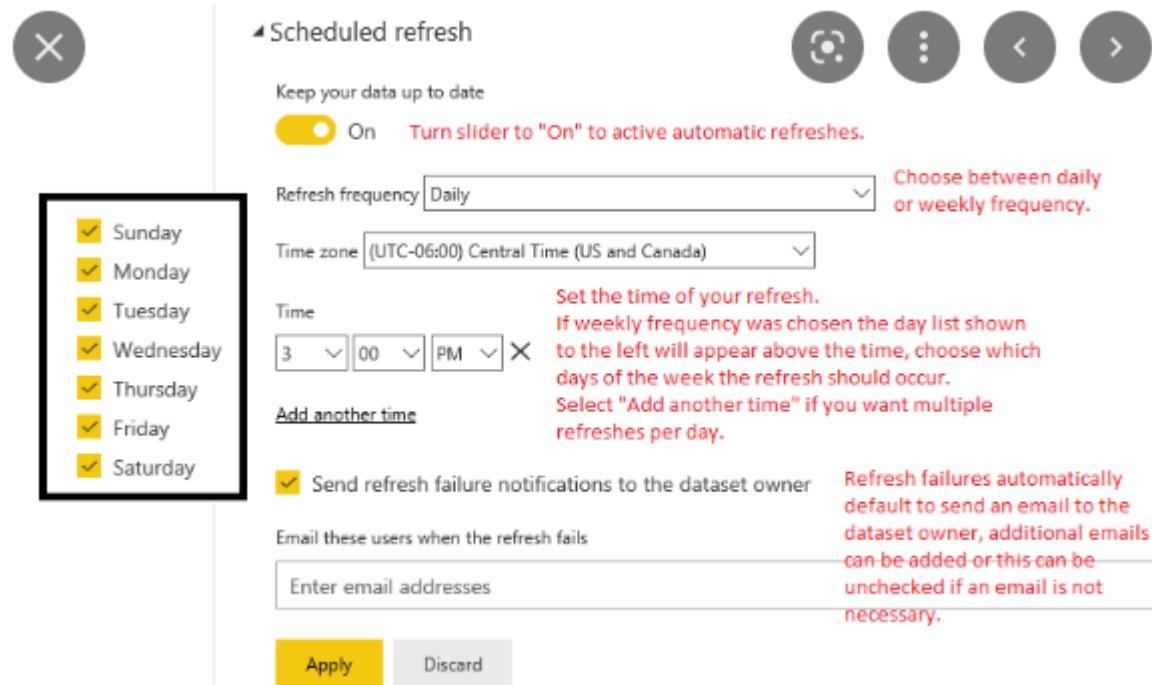


Scheduled Refresh

To refresh the report automatically

8 Scheduled Refresh can be set with pro licence

48 Scheduled Refresh can be set with premium licence



What is Query Folding in Power BI?

Query folding is the ability for a Power Query to generate a single query statement that retrieves and transforms source data.

Mashup engine is able to “translate” M language to a language that the underlying data source will “understand” — in most cases it’s SQL.

By pushing complex calculations and transformations directly to a source, Power Query leverages the capabilities of the robust relational database engines, that are built to cope with large volumes of data in the most efficient way.

How to check if Query folding has been achieved or not?

Right click on Applied Steps -> View Native Query -> Click Ok (You can see the folded query)

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [1]' pane lists 'world country'. The main area displays a table with columns: 'LocalName', 'GovernmentForm', and 'HeadofState'. The 'APPLIED STEPS' pane on the right shows a step named 'Renamed C' which has 'Rename' selected. A context menu is open over this step, with 'View Native Query' highlighted and underlined.

Native Query:

```

select
    [Code] as 'code',
    [Name] as 'Country Name',
    [Continent] as 'Continent',
    [Region] as 'Region',
    [SurfaceArea] as 'SurfaceArea',
    [IndepYear] as 'IndepYear',
    [Population] as 'Population',
    [LifeExpectancy] as 'LifeExpectancy',
    [GNP] as 'GNP',
    [GNPold] as 'GNPold',
    [LocalName] as 'LocalName',
    [GovernmentForm] as 'GovernmentForm',
    [HeadofState] as 'HeadofState',
    [Capital] as 'Capital',
    [code2] as 'code2'
from 'world'.country

```

What if the View Native Query option is not available?

GoTo Tools -> Diagnose Step -> Diagnose -> Click -> It will generate queries (For Counters, Detailed and Aggregation) -> Detailed Query Option -> GoTo Operation Column -> Filter by GetResponseStream -> Click Ok -> GoTo DataSourceQuery Column -> Click on the Request -> Check if the applied steps are reflecting or not

Queries [2]

world country

Customers

= Table.SelectRows("Removed Other Columns", each {[Region] = "SP"})

	CustomerID	CompanyName	Region
1	COMM1	Comércio Mineiro	SP
2	FAMIL1	Família Arquibaldo	SP
3	GOURL	Gourmet Lanchonetes	SP
4	QUEEN	Queen Cozinha	SP
5	TRADH	Tradição Hipermercados	SP
6	WELLU	Wellington Importadora	SP

Query Settings

PROPERTIES

Name: Customers

All Properties

APPLIED STEPS

- Source
- Navigation
- Removed Other Columns
- Filtered Row
 - Edit Settings
 - Rename
 - Delete
 - Delete Until End
 - Insert Step After
 - Move before
 - Move after
 - Extract Previous
 - View Native Query
 - Diagnose
 - Properties...

The screenshot shows the Power BI Diagnostic Step interface. The top navigation bar includes 'Step *', 'Start Diagnostics', 'Stop Diagnostics', 'Diagnostic Options', and 'Diagnostic Step'. Below the navigation is a 'Step' ribbon with tabs: 'Diagnostics' (selected), 'Diagnostic Step', and 'Cancel Diagnostics'. The main area displays a query editor with the following code:

```
= Table.SelectRows(#"Removed Other Columns", each ([Region] = "SP"))
```

The data table below the code contains the following rows:

	CustomerID	CompanyName	Region
1	COMM1	Comércio Mineiro	SP
2	FAMILIA	Família Arquibaldo	SP
3	GOURL	Gourmet Lanchonetes	SP
4	QUEEN	Queen Cozinha	SP
5	TRADH	Tradição Hipermercados	SP
6	WELLU	Wellington Importadora	SP

To the right, the 'Query Settings' pane is open, showing 'Properties' (Name: Customers, All Properties) and 'Applied Steps' (Source, Navigation, Removed Other Columns, Filtered Rows).

Queries [5]

	Category	Data Source Kind
1	Evaluation	
2	Document Evaluator	
3	Evaluator	
4	Other	
5	Evaluator	
6	Evaluator	
7	Evaluator	
8	Evaluator	
9	Data Source	OData
10	Data Source	OData
11	Data Source	OData
12	Data Source	OData
13	Data Source	OData

fx Table.TransformColumnTypes(#"Expanded Column1", {{"Column1", typeText}})

Start Time

Sort Ascending
Sort Descending
Clear Sort
Clear Filter
Remove Empty
Text Filters
Search
GetAsStream
Create
Dispose
Evaluation
Generate preview
GetResponse
GetReadStream
GetResult
GetResult: Compile
GetResult: Evaluate
RemoteCancellationServiceFactory: Proxy: Canc
RemoteServiceEnvironment: CreateServiceStub
RemoteServiceEnvironment: DisposeServiceStub
Request
Service: OnBeginGetResult

Query Settings

PROPERTIES

Name: Customers_Filtered_Rows_Detailed_2022-
All Properties

APPLIED STEPS

Source
Converted to Table
Expanded Column1
Changed Type

The screenshot shows the Power BI Query Editor interface. On the left, the 'Queries [5]' pane lists several queries, including 'Customers_Filter...' which is currently selected. The main area displays the query code:

```
= Table.SelectRows(#"Changed Type", each {[Operation] = "GetResponseStream"})
```

The 'Data Source Query' pane shows the results of the query, including the request URL and response content. The 'Query Settings' pane on the right shows the properties of the selected query, including its name ('Customers_Filtered Rows_Detailed_2022') and applied steps.

Where Data is Stored in Power BI?

Data Source	Metadata	Data
Live Connection (Analysis Services)	Nothing stored except database name encrypted in Azure SQL DB	Nothing Stored
Direct Query (SQL Server, Oracle, etc.)	Encrypted in Azure Blob Storage	Nothing Stored
Pushed or streamed data	Encrypted in Azure Blob Storage	V1 – stored encrypted in Azure Blob storage V2 – stored encrypted in Azure SQL Database
Data loaded into model (data may be refreshable or non refreshable)	Encrypted in Azure Blob Storage	Encrypted in Azure Blob Storage

Power BI Refresh Types?

<https://learn.microsoft.com/en-us/power-bi/connect-data/refresh-data>