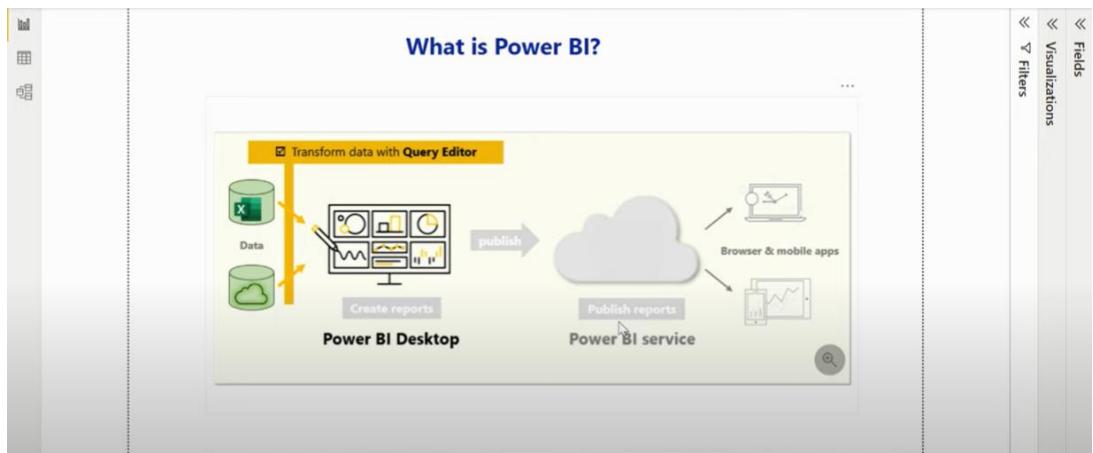
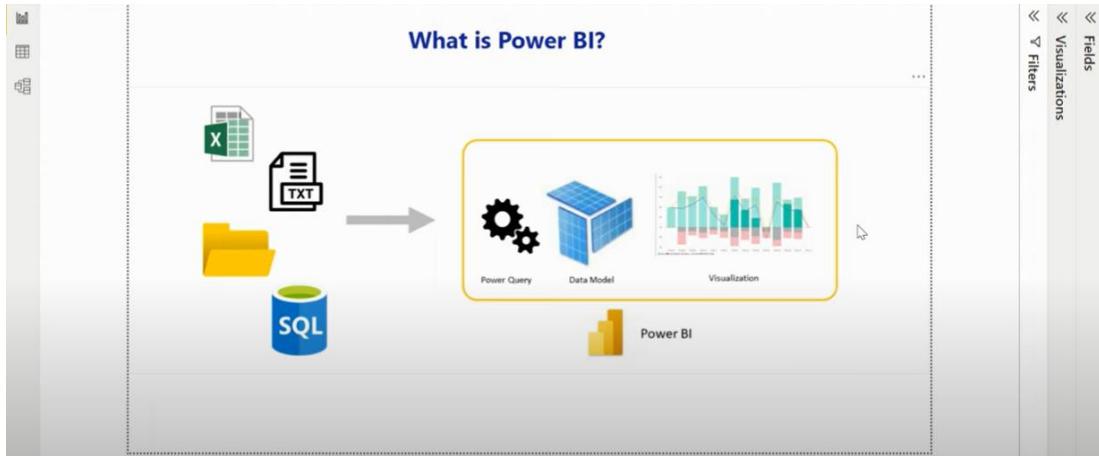
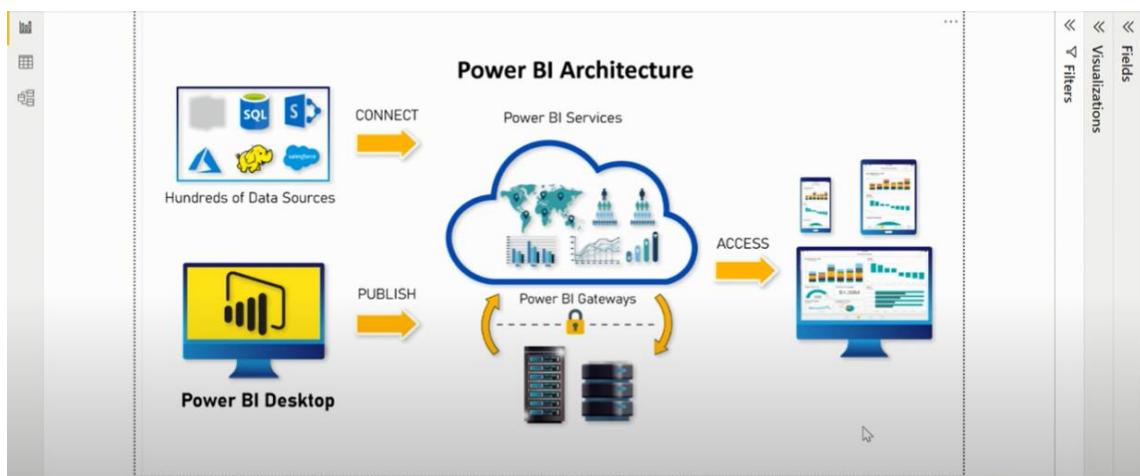


<https://docs.microsoft.com/en-us/dax/>

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, etc. 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:

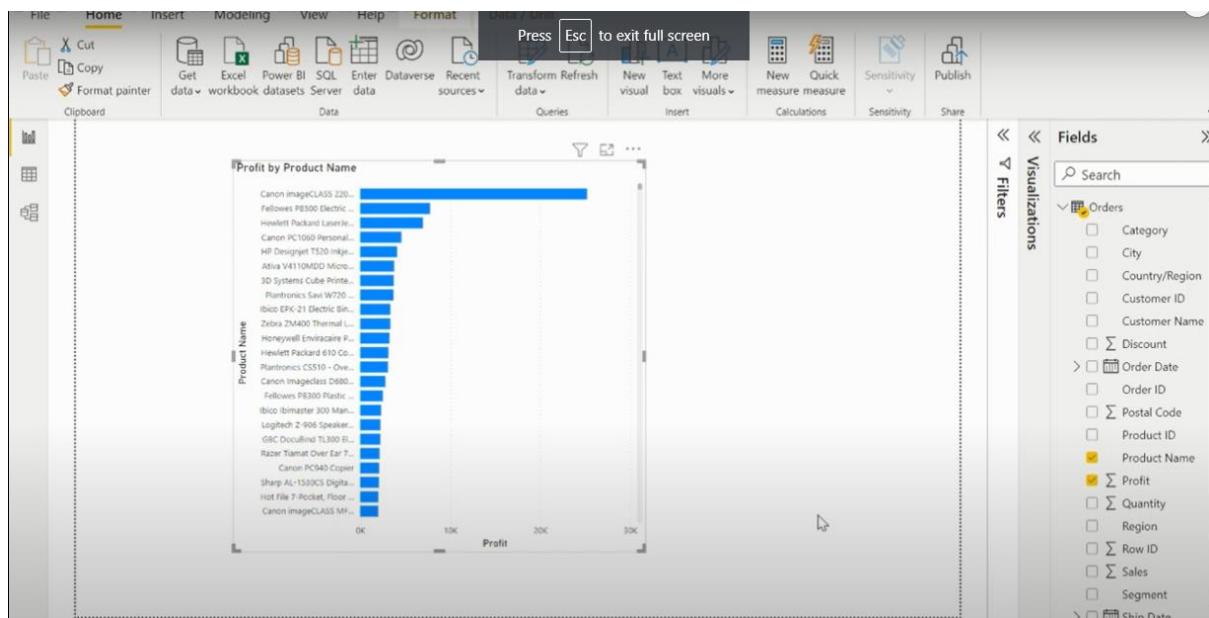
- 1) Power Query - Change Datatype, Renaming the column, deleting the column, new calculated column (M Language)
- 2) Power Pivot - To model the data, define relationship between multiple tables (DAX Language)
- 3) Power View - To build the visualization, where you can see the data visually

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



The screenshot shows the "Get Data" dialog box in Power BI desktop. On the left, the "Navigator" pane lists available files: "Sample - Superstore.xlsx" (selected), "Orders", "Orders2", "People" (selected), "People2" (checked), "Returns", and "Returns2". On the right, a preview of the "People" table is shown, with the heading "People" and the note "Preview downloaded on 23 March 2022". The table has two columns: "Column1" and "Column2". The data rows are:

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

At the bottom of the dialog are buttons for "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: 'Region' and 'Name'. The table contains five rows of data. On the left, the 'Queries [3]' pane lists 'Orders', 'Returns2', and 'People2'. On the right, the 'Properties' pane shows the table name 'People2' and the 'Applied Steps' pane, which includes a step labeled 'Changed Type'.

Region	Name
West	Regional Manager
East	Sadie Pawthorne
Central	Chuck Magee
South	Roxanne Rodriguez
	Fred Suzuki

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

The screenshot shows the Microsoft Power BI desktop application. The ribbon at the top includes tabs for File, Home, Insert, Modeling, View, Help, Format, Data / Drill, and various icons for data sources, queries, insertions, calculations, sensitivity, and publishing. The clipboard pane on the left contains a list of recently copied items, with one item circled in red. The filters pane shows filters applied to the current visual, such as 'Product Name is (All)' and 'Profit is (All)'. The visualizations pane displays a grid of visualization icons. The fields pane on the right lists various data fields categorized by type, with some checked or highlighted.

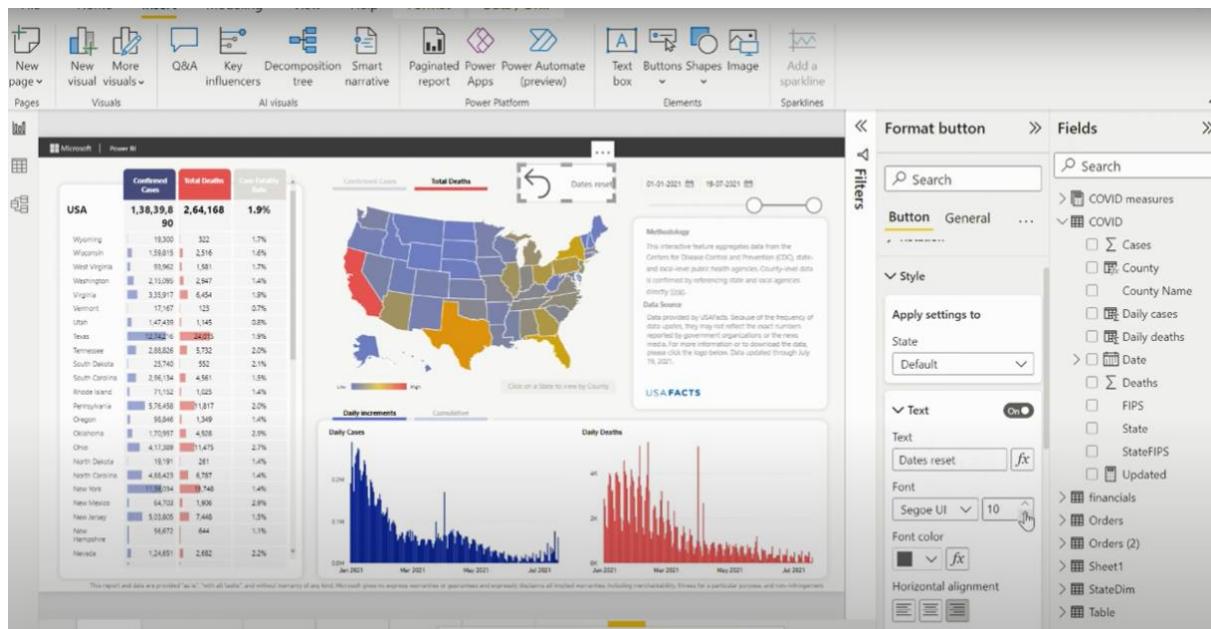
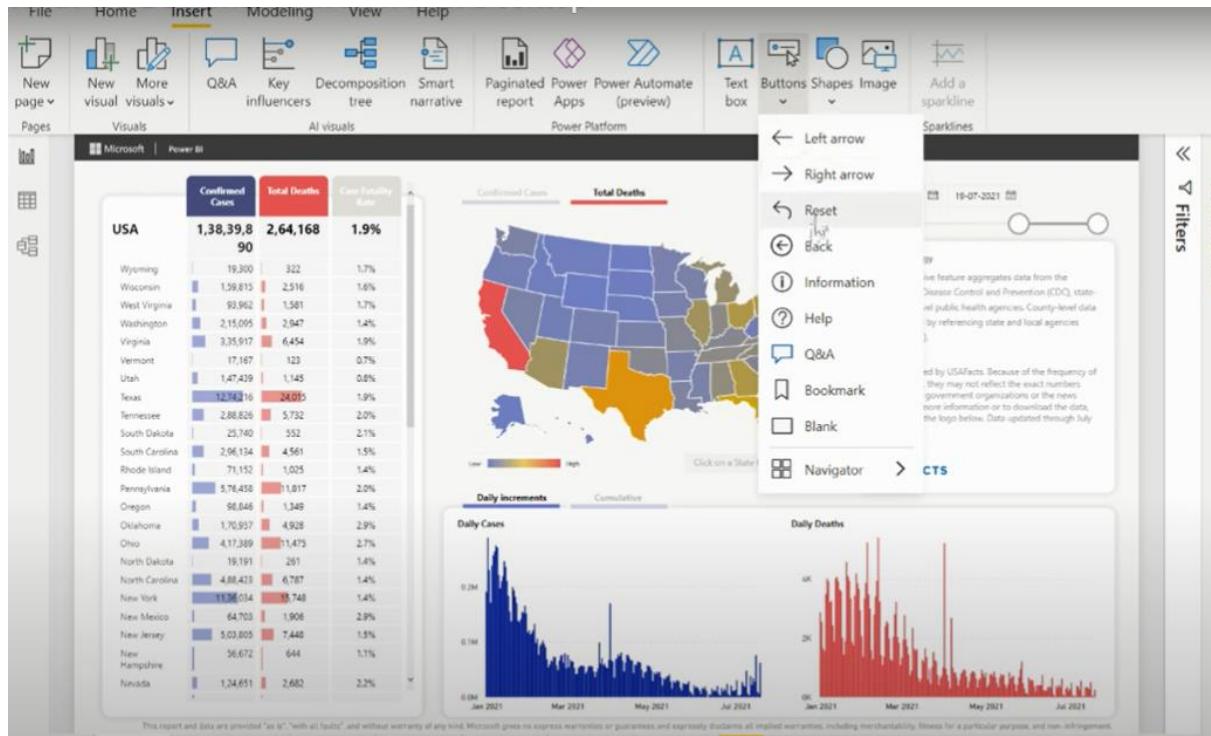
Bookmarks in Power BI?

Bookmark reset the data in the form it is created

View -> Bookmarks -> Add -> Rename

This screenshot shows the Microsoft Power BI desktop application with a different set of panes visible. The ribbon includes File, Home, Insert, Modeling, View, Help, and various icons. The themes pane shows a grid of color swatches. The filters pane is open, showing a list of filters like 'Confirmed Cases' and 'Total Deaths'. The bookmarks pane is open, showing a list of bookmarks such as 'COVID measures', 'COVID', 'Blue States', 'Pink States', 'Daily State', 'Cumulative State', 'Daily County', 'Cumulative County', 'Daily Cnty2', 'County cases', 'County deaths', 'State selection', '2021', and '2020'. The fields pane on the right lists fields like 'Cases', 'County', 'County Name', 'Daily cases', 'Daily deaths', 'Deaths', 'FIPS', 'State', 'StateFIPS', and 'Updated'. A large central area displays a dashboard with a map of the USA showing COVID-19 data by state, and two line charts below it showing 'Daily Cases' and 'Daily Deaths' over time.

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

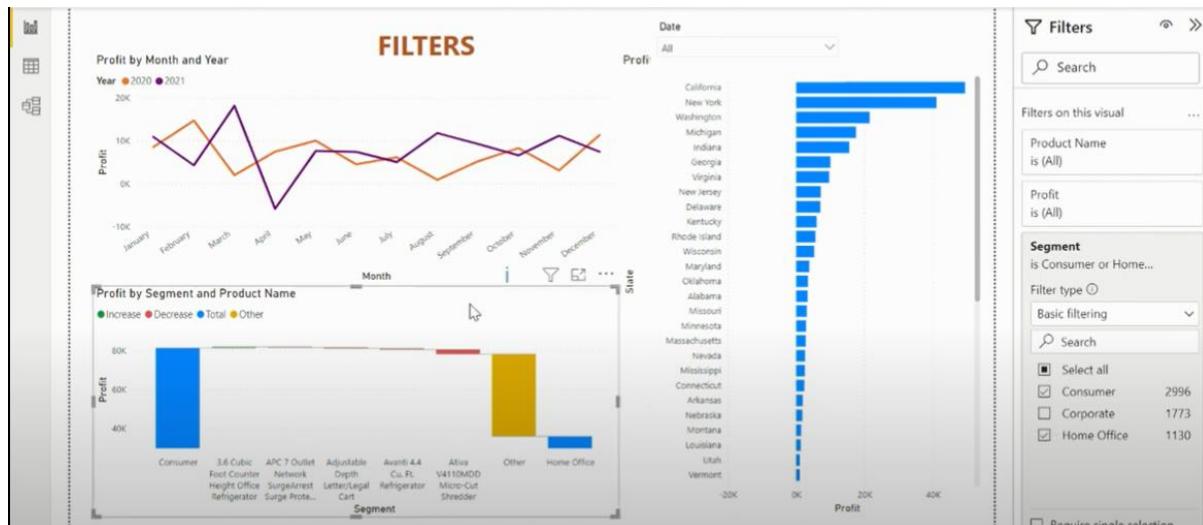


Filters and Slicers :

Both are used to filter the data

Filter :

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



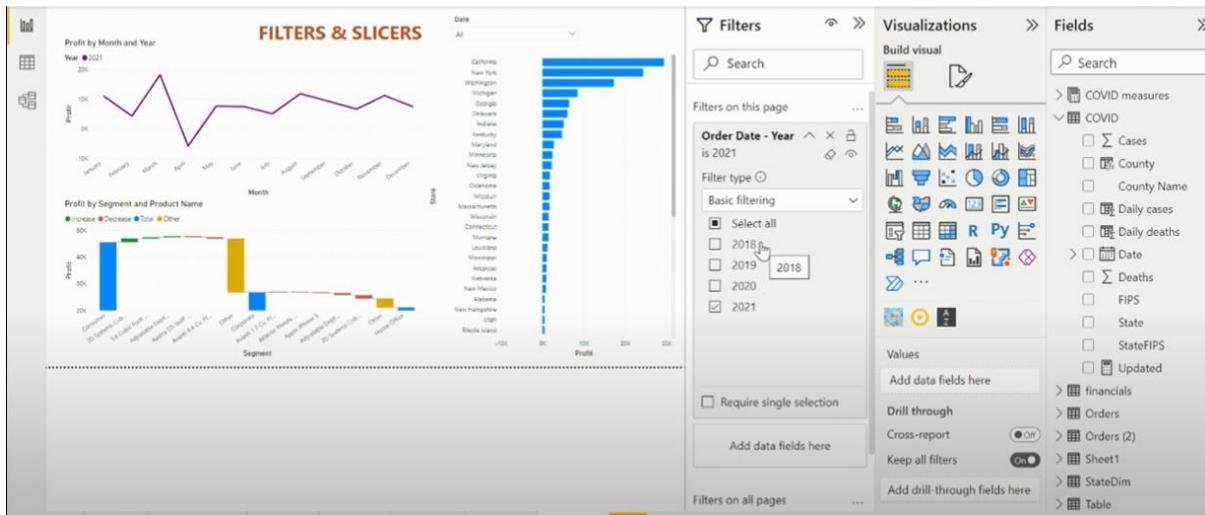
Filter on this page: will filter the current page only

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

Drag & drop the field on required filter type

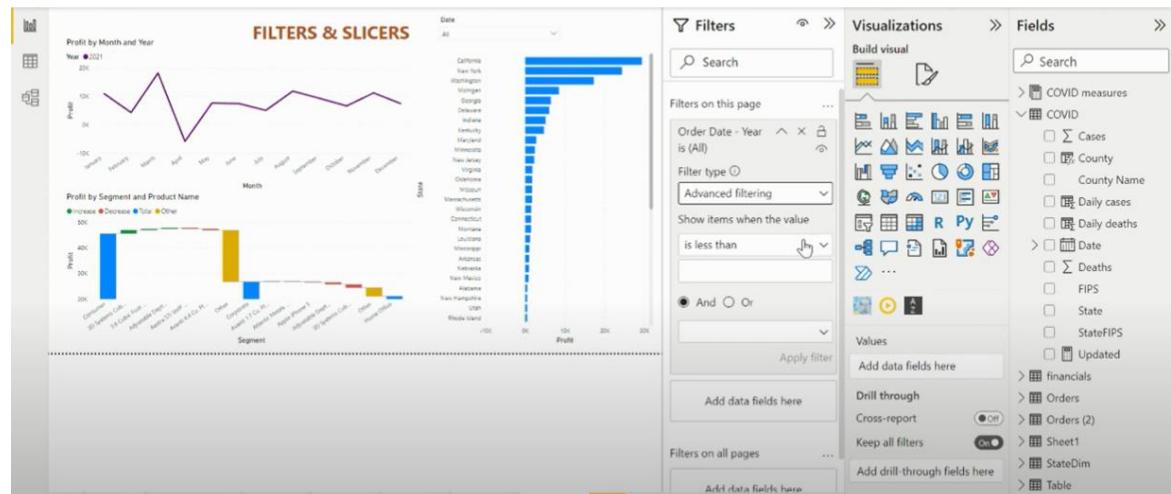


Basic filtering:



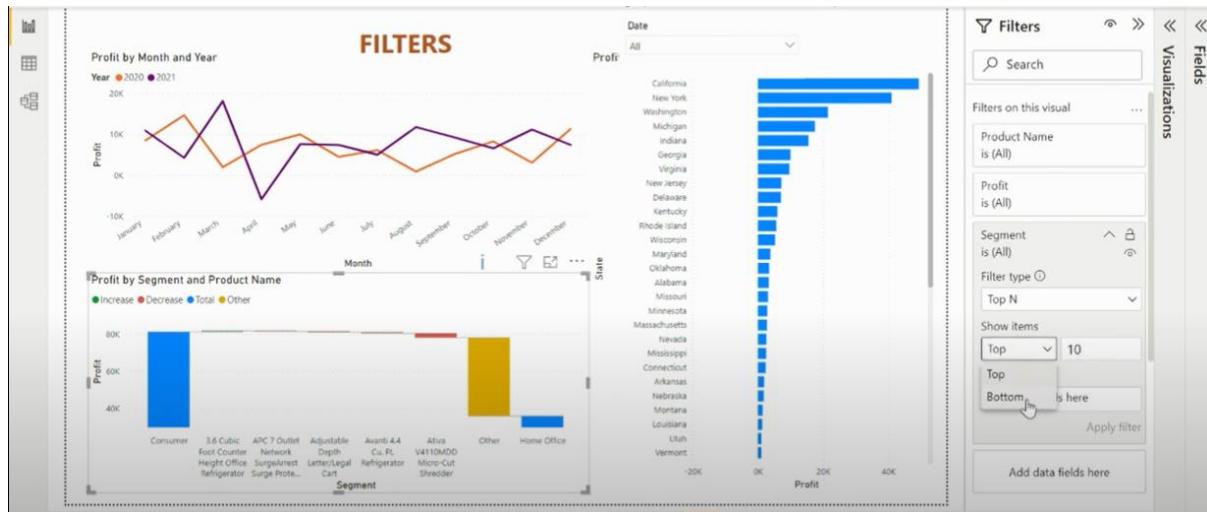
Advanced filtering:

Filtering based on conditions (same as excel)



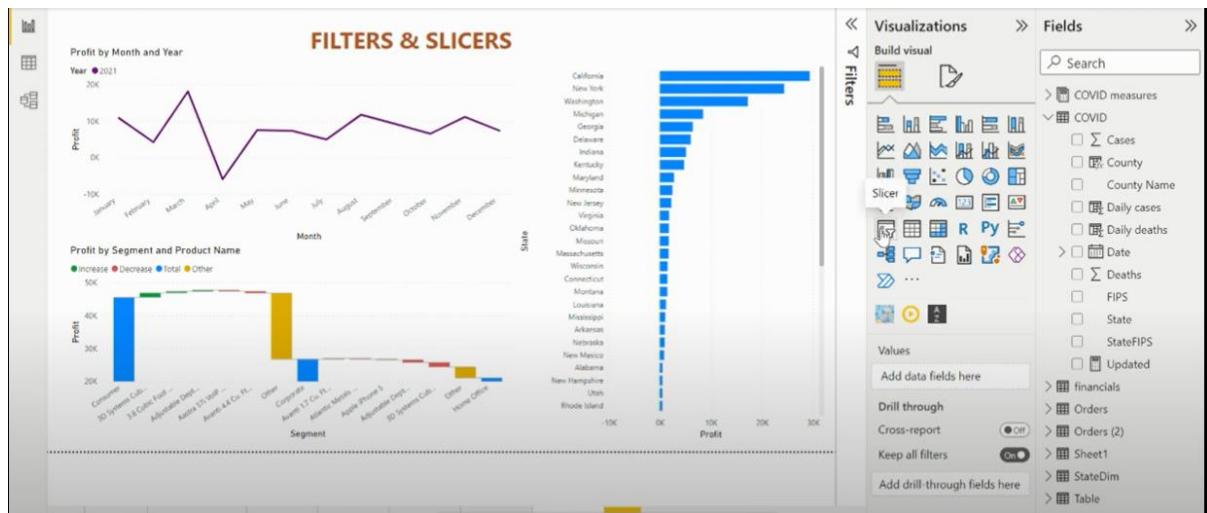
TOP N filtering:

To filter top/bottom contributors



Slicer:

Slicer is a Visual



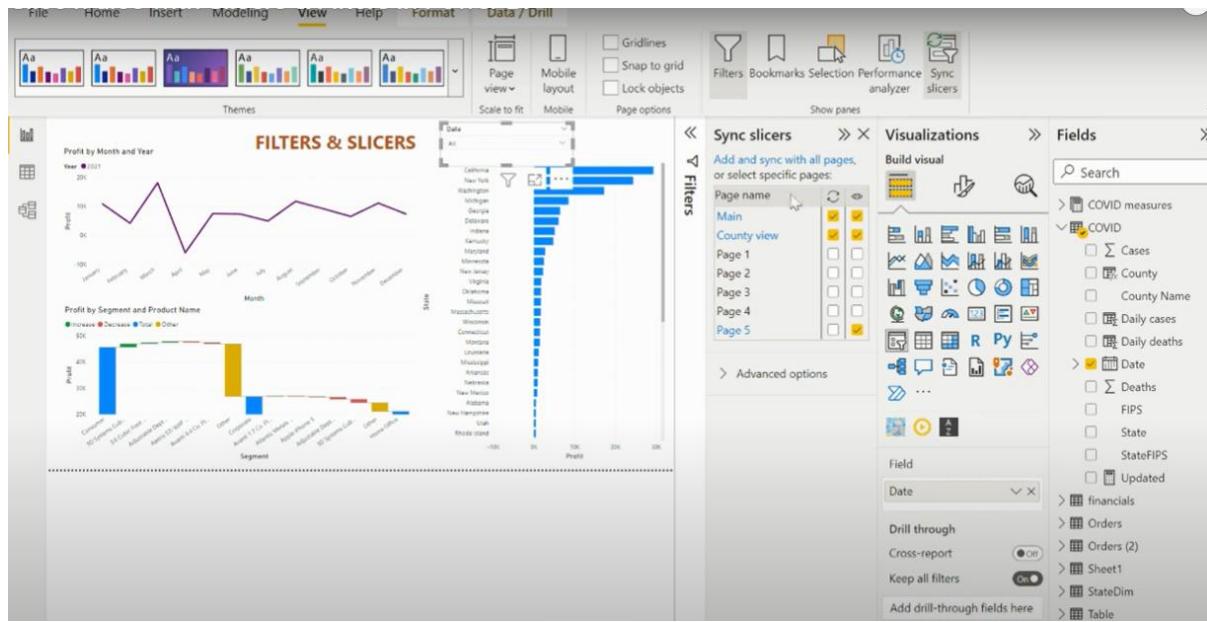
Click on slicer -> place it on report view -> add field -> right top corner (select the type)

Slicer can be applied on all the pages / selected pages

View -> sync slicers

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

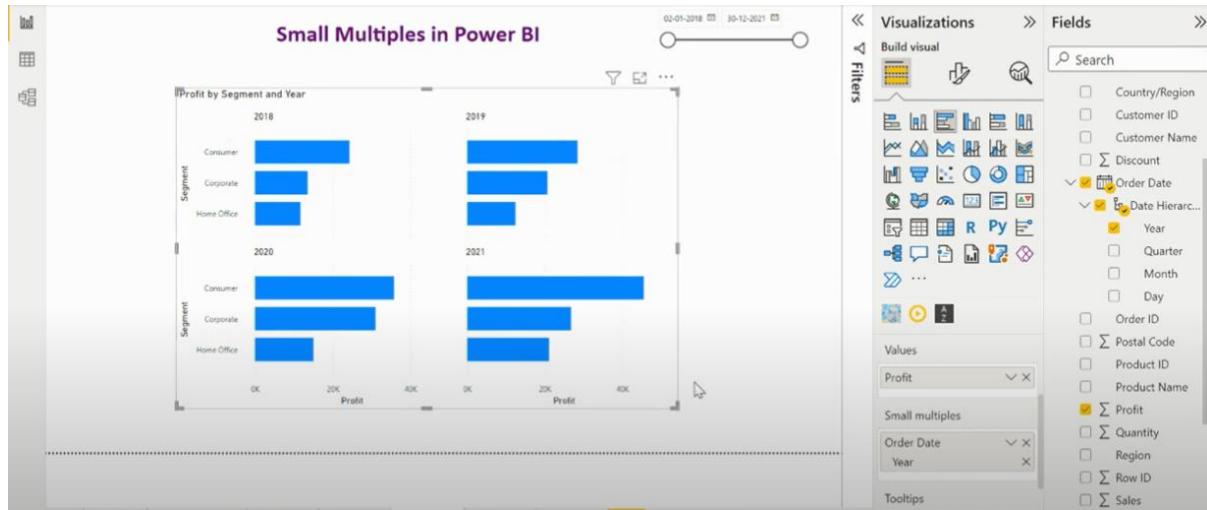
Display option : slicer will be displayed on selected pages



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:

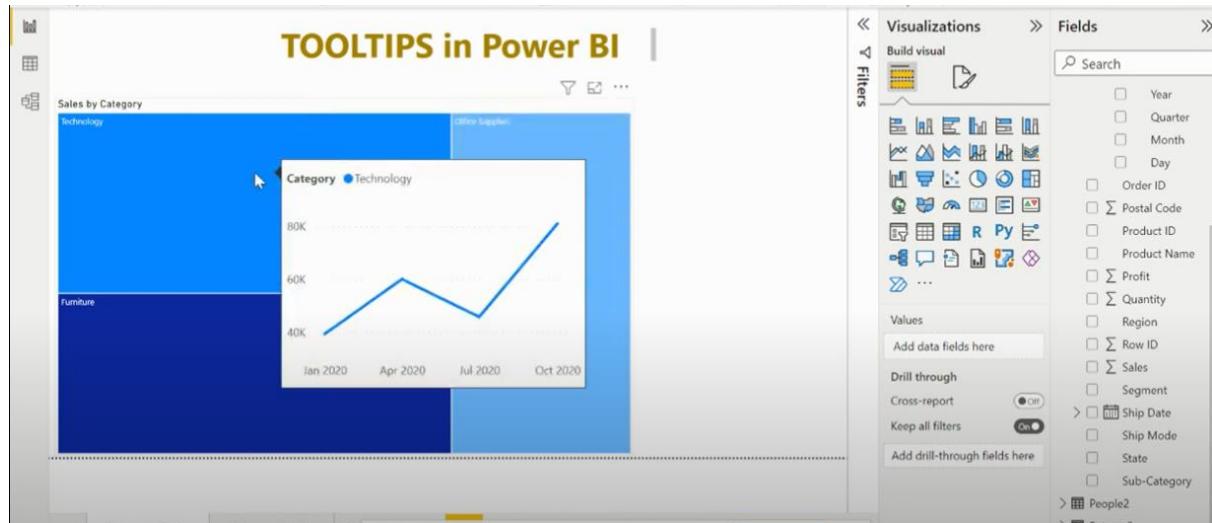
Drag & drop field in small multiples



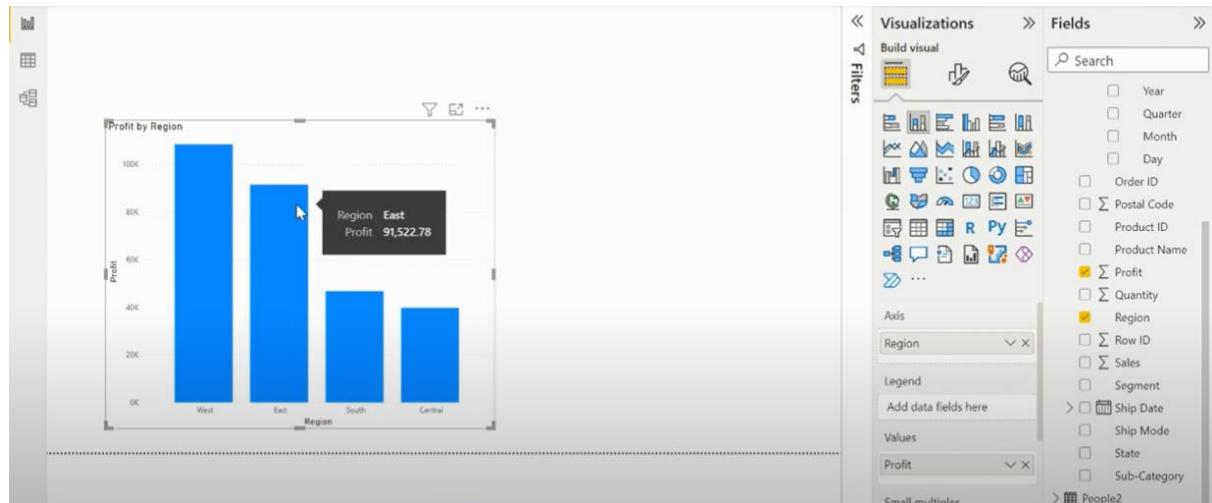
Customized Tooltips in Power BI:

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

Page information

Name: Region tooltip

Allow use as tooltip:

Allow Q&A:

Reset to default

> Canvas settings

> Canvas background

> Wallpaper

> Filter pane

> Filter cards

Visualizations

Format page

Fields

Search

- Year
- Quarter
- Month
- Day
- Order ID
- Postal Code
- Product ID
- Product Name
- Profit**
- Quantity
- Region
- Row ID
- Sales
- Segment
- Ship Date
- Ship Mode
- State
- Sub-Category
- People2

Visualizations

Format page

Fields

Search

Favorites

Page information

> Canvas settings

Type: Tooltip

Height: 240 px

Width: 200 px

Vertical alignment: Middle

Reset to default

> Canvas background

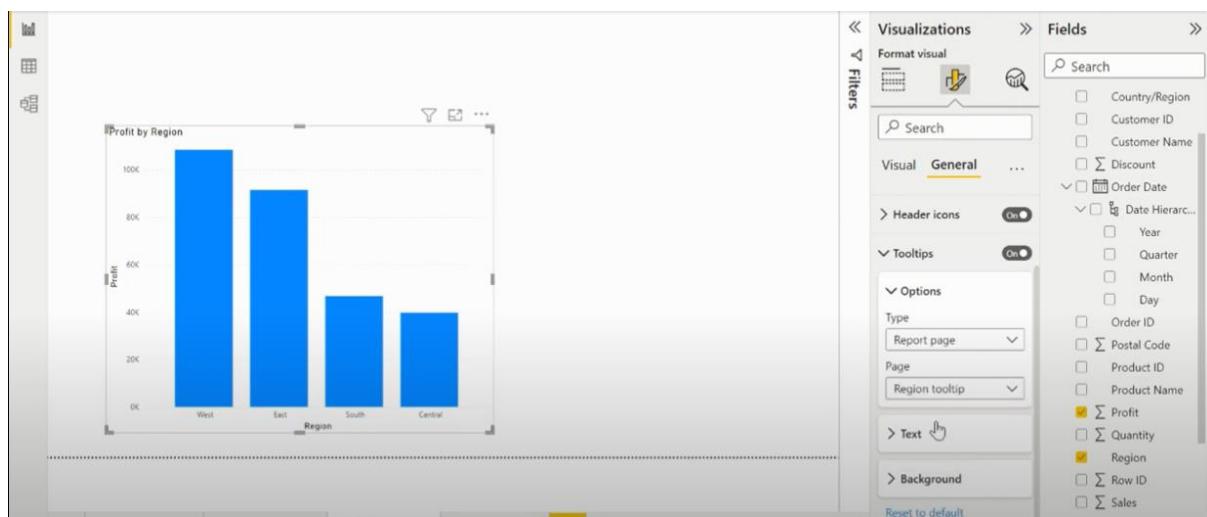
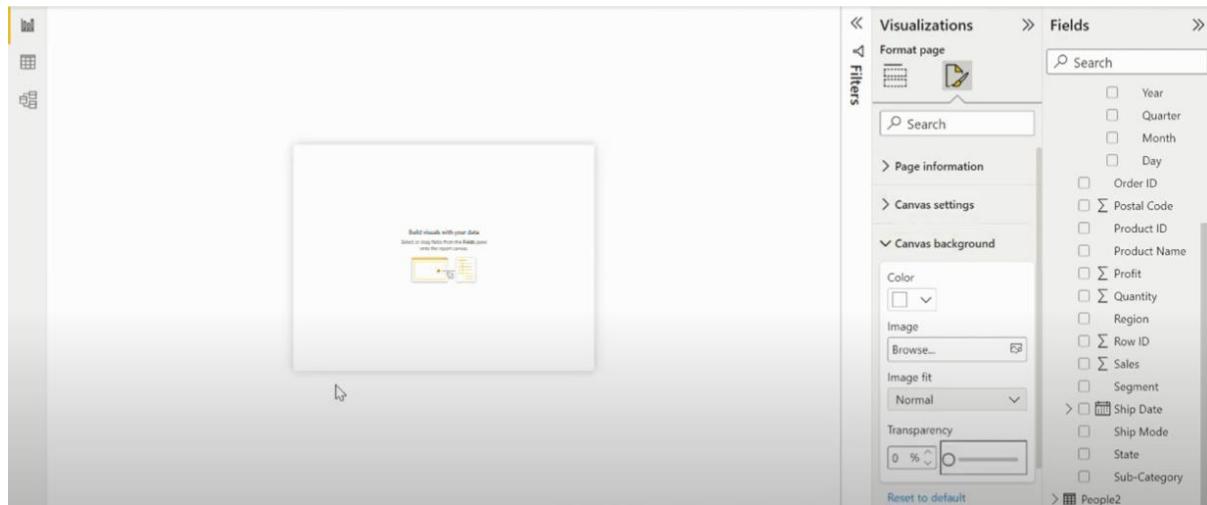
Visualizations

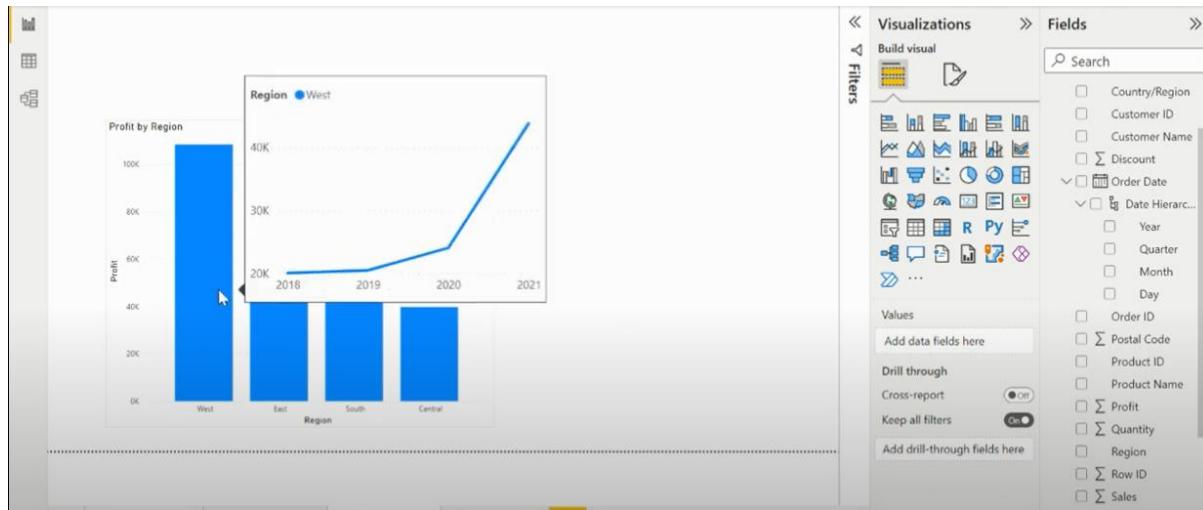
Format page

Fields

Search

- Year
- Quarter
- Month
- Day
- Order ID
- Postal Code
- Product ID
- Product Name
- Profit**
- Quantity
- Region
- Row ID
- Sales
- Segment
- Ship Date
- Ship Mode
- State
- Sub-Category
- People2





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

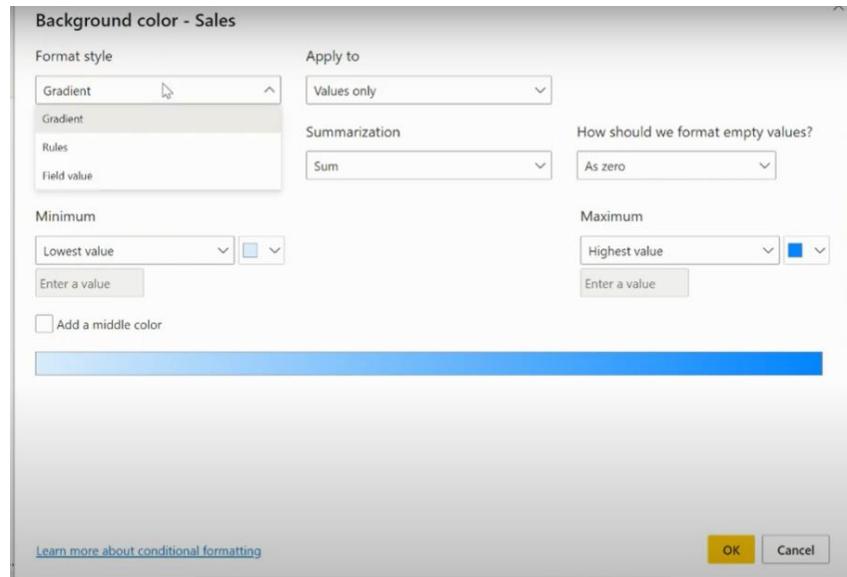
Data Bars –

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

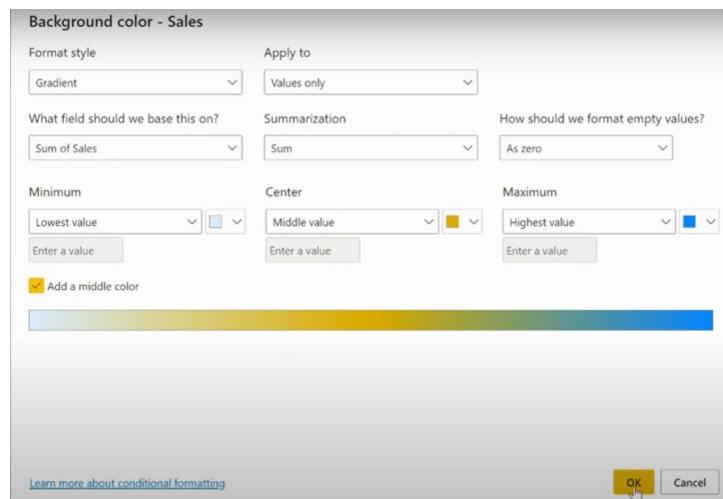
Icons will be displayed alongside the cell values

Format Style – Gradient

Cells will be distinguished by shades of two chosen colours

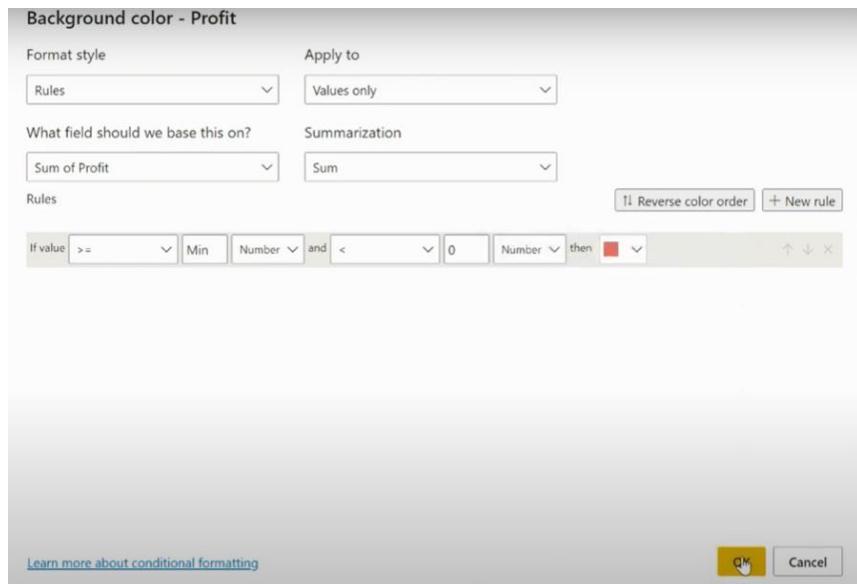
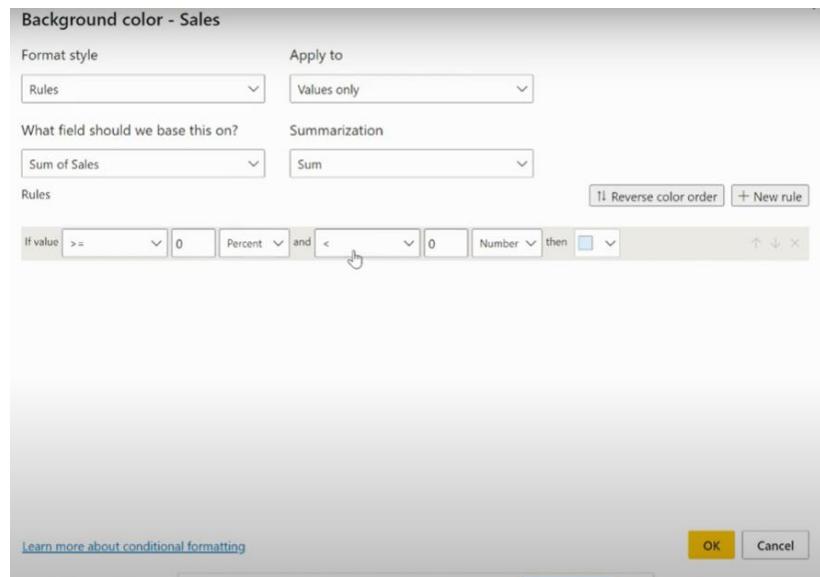


Cells will be distinguished by shades of three chosen colours by selecting the option to add middle colour



The screenshot shows a Power BI report interface. On the left, there is a matrix visual with columns 'Year', 'Sales', 'Profit', and 'Average of Discount'. The data is grouped by year (2021, 2020, 2018, 2019) and further by quarter (Qtr 1, Qtr 2, Qtr 3, Qtr 4). The cells are colored according to a three-color gradient based on their value. The report title is 'Conditional Formatting - Matrix'. On the right, the 'Visualizations' pane is open, showing various chart and table options. The 'Values' section includes fields 'Sales', 'Profit', and 'Average of Discount'. The 'Drill through' section includes 'Cross-report' and 'Keep all Filters' buttons.

Format Style – Rules



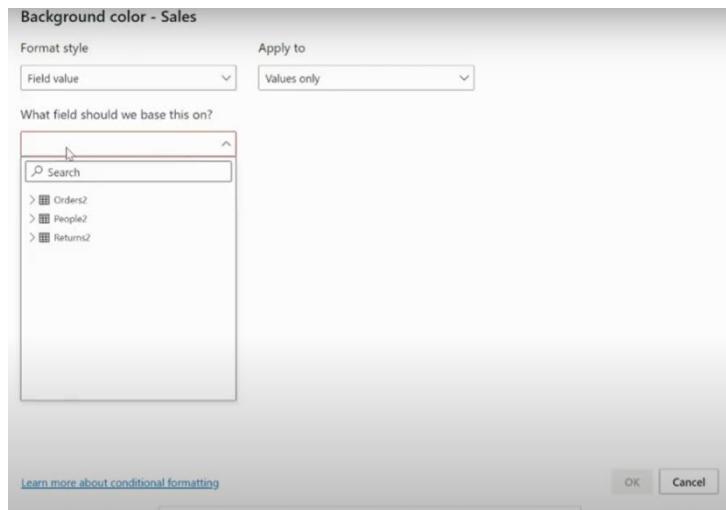
Conditional Formatting - Matrix

Visualizations: Build visual, Filters, Fields

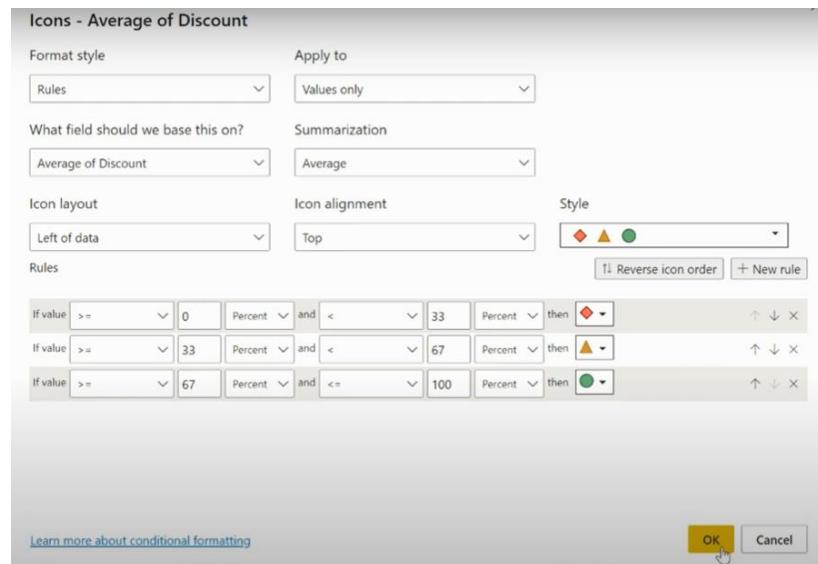
Filters: Add data fields here, Values (Sales, Profit, Average of Discount), Drill through (Cross-report Off, Keep all filters On)

Year	Sales	Profit	Average of Disc%
Qtr 2	1,33,764.37	15,499.21	
Qtr 1	1,23,144.86	23,506.20	
2020			
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	
2019			
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

Format Style – Fields



Conditional formatting based on icons



**Additional
Formatting - Matrix**

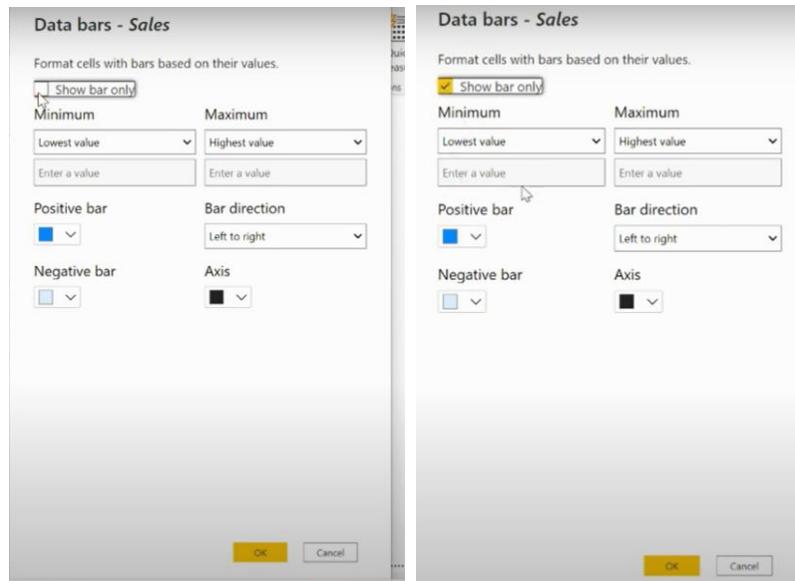
Year	Sales	Profit	Average of Discount
Qtr 2	1,33,764.37	15,499.21	0.17
Qtr 1	1,23,144.86	23,506.20	0.15
2020			
Qtr 4	2,36,098.75	38,139.86	0.15
Qtr 3	1,43,787.36	15,823.60	0.16
Qtr 2	1,36,082.30	16,390.34	0.16
Qtr 1	93,237.18	11,441.37	0.15
2018			
Qtr 4	1,79,627.73	21,723.95	0.17
Qtr 3	1,43,633.21	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

A screenshot of a Power BI report titled 'Conditional Formatting - Table'. It displays a table with columns: Order Date, Profit, Sales, Quantity, and Discount. The Sales column uses a blue-to-white gradient conditional format. The table includes a 'Total' row at the bottom. The Power BI interface shows the 'Visualizations' pane on the right, which has 'Sales' selected under the 'Values' section. Other fields listed in the Values pane include Order Date, Profit, Sales, Quantity, and Discount.

Conditional Formatting - Table

A screenshot of a Power BI report titled 'Conditional Formatting - Table'. It displays a table with columns: Order Date, Profit, Sales, Quantity, and Discount. The Sales column uses a blue-to-white gradient conditional format. The table includes a 'Total' row at the bottom. The Power BI interface shows the 'Visualizations' pane on the right, which has 'Profit' selected under the 'Values' section. Other fields listed in the Values pane include Order Date, Profit, Sales, Quantity, and Discount. A red arrow points to the 'Profit' entry in the Values pane.

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

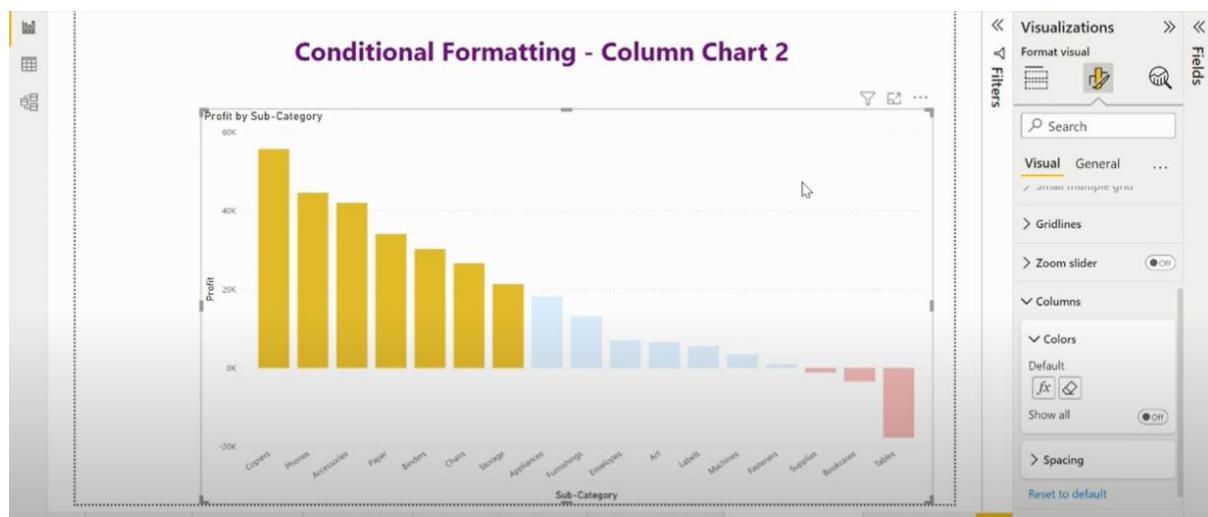
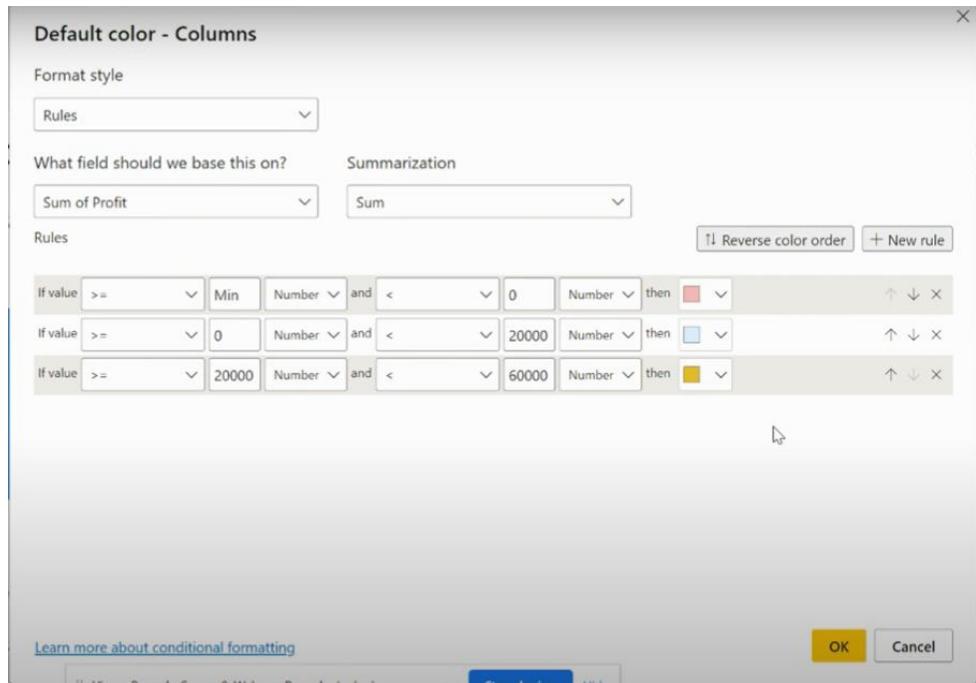
The screenshot shows a Power BI interface with a table titled "Conditional Formatting - Table". The table has columns: Order Date, Profit, Sales, Quantity, and Average of Discount. The rows show data from December 2021, with the last row being the Total. The background color of the table cells changes based on the value in the "Sales" column, creating a heatmap effect. The "Visualizations" pane on the right shows various chart types, and 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	6,231.15	54	0.11
29 December 2021	644.43	9,155.33	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 the sales volume for each sub-category. A tooltip for one of the bars provides information about conditional formatting, stating: "Conditional formatting Make this property change under different conditions that you define." The "Visualizations" pane on the right shows various chart types, and the "Fields" pane lists the columns used in the chart. A red circle highlights the "Format visual" icon in the "Visualizations" pane.



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:

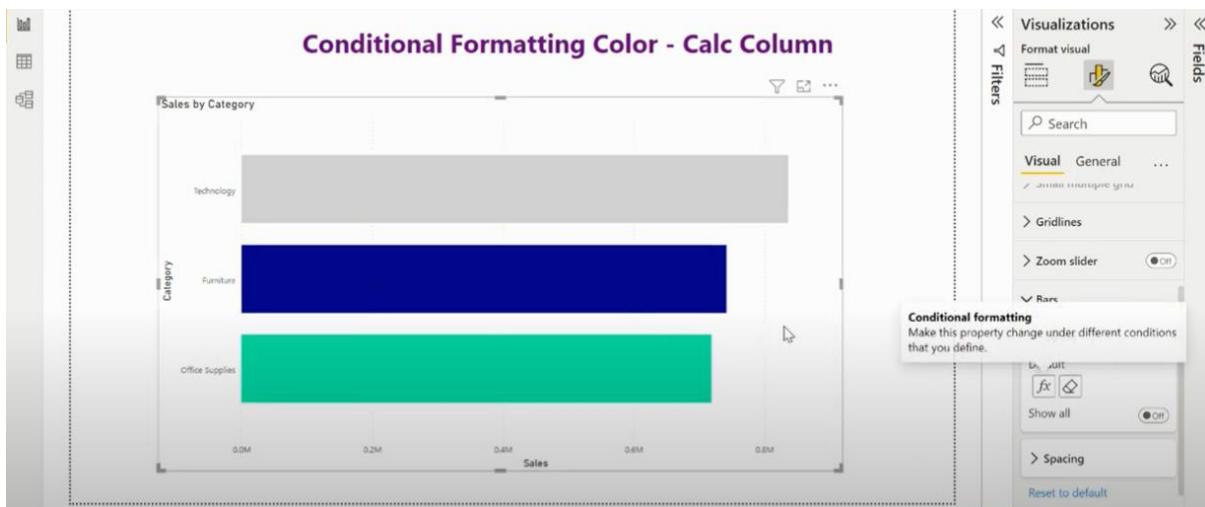
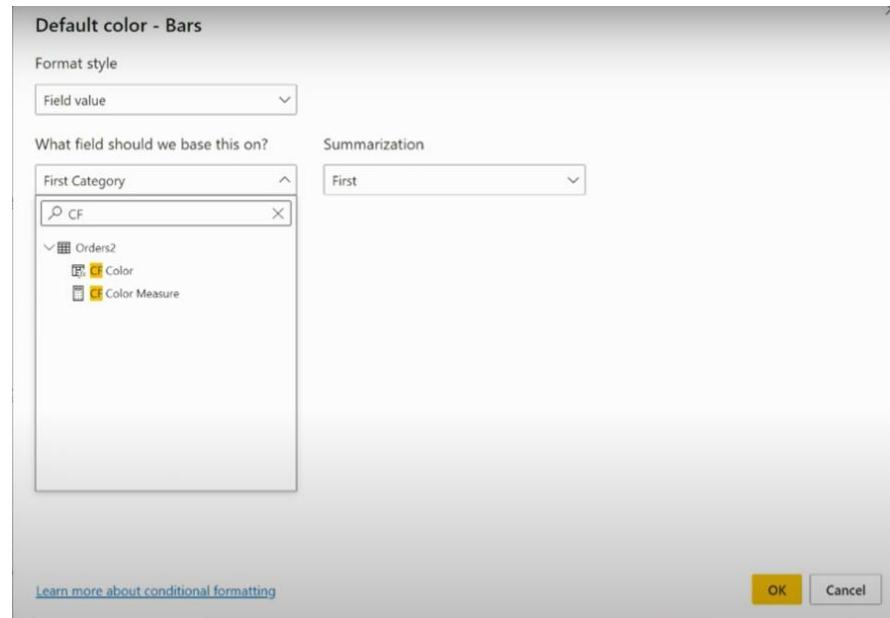
```

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 data shows various office supplies and furniture items across different regions and categories.



Conditional Formatting – based on Measure

Go to data section -> create calculated column 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. On the left, there's a code editor window with the following DAX formula:

```
1 CF Color Measure =  
2 MAXX(Orders2,  
3 SWITCH(  
4     Orders2[Category],  
5     "Furniture", "darkblue",  
6     "Office Supplies", "#00EDA4",  
7     "lightgrey")  
8 )
```

To the right of the code editor is a table with columns: Sales, Quantity, Discount, Profit, and CF Color. The CF Color column contains color codes corresponding to the categories in the Category column. Below the table is a Fields pane listing various columns and their 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 $\sum \text{Region}$.

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

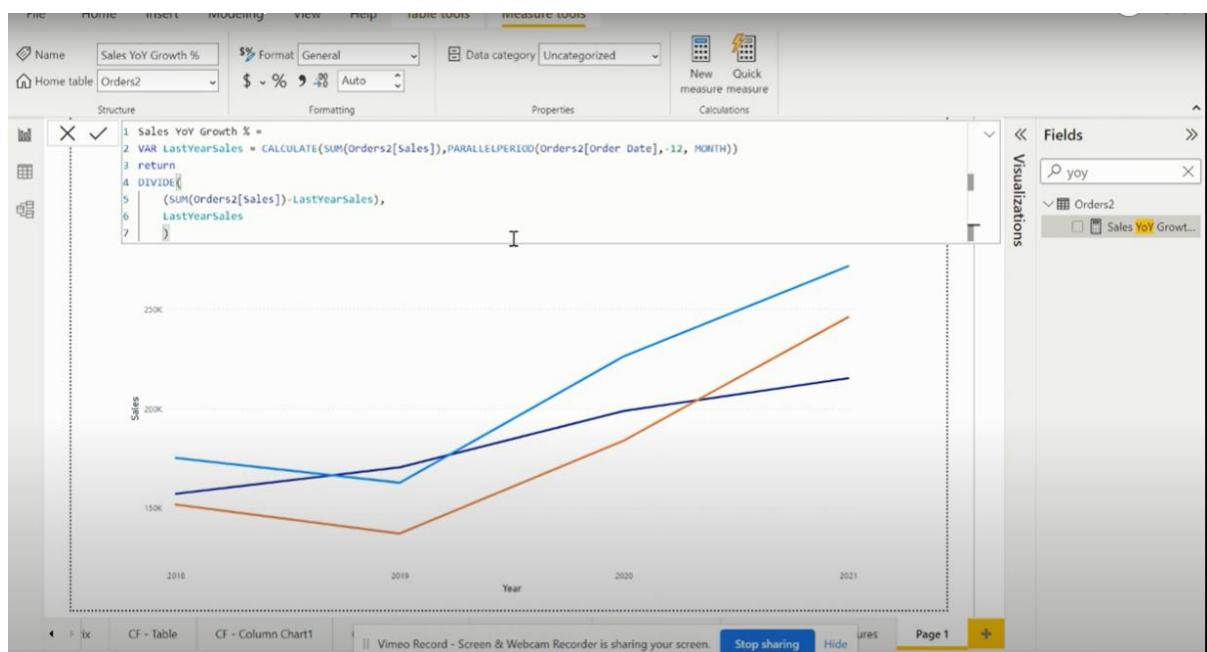
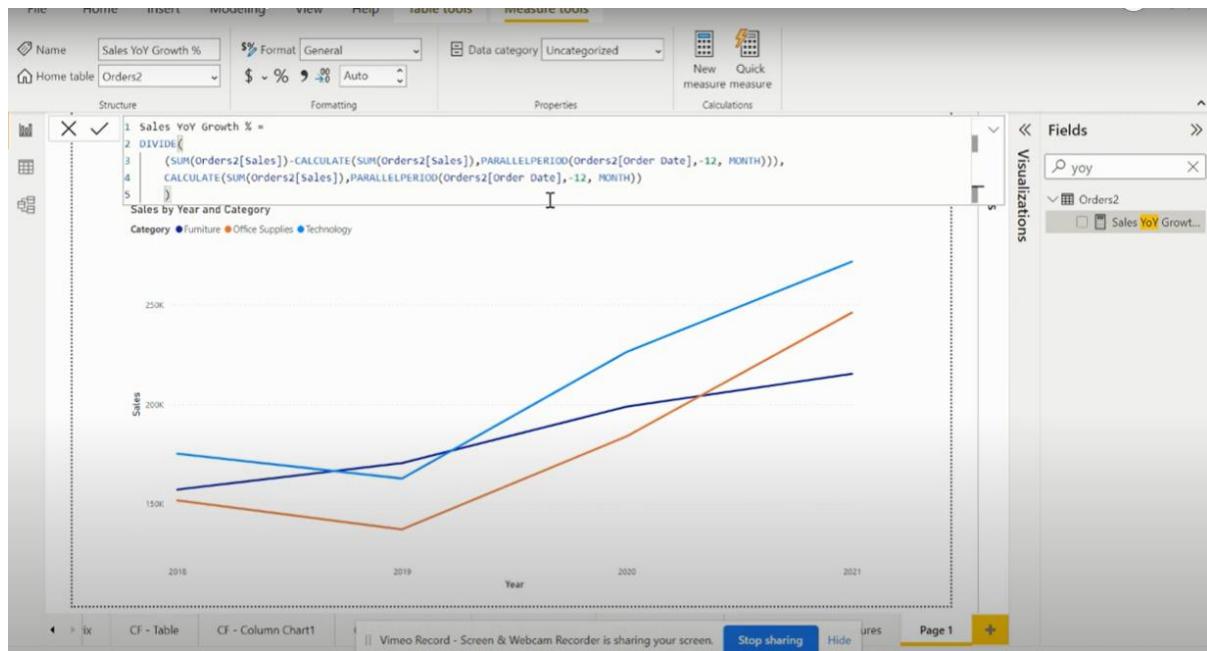
Makes it simpler to debug in case of error

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: `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 as the previous one, but with more columns added to the table visualization. The columns are State, City, Sales, and State ranks. The table now includes data for cities within states like Alabama, Arizona, and Arkansas. The 'State ranks' column remains highlighted. The formula bar still displays the same measure definition: `State ranks = RANKX(ALL(Orders2[State]),[total Sales])`.

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 being defined with the formula: `RANKX(ALL(Orders2[State]),ALL(Orders2[City]),[Total Sales])`. The formula bar has a red error icon. A tooltip says 'Can't display the visual. See details'. The Fields pane on the right lists various dimensions and measures like State, City, Sales, and Order Date.

2 ALL functions needs to be defined inside crossjoin function.

The screenshot shows the Power BI Model view. The same measure 'State ranks' is now defined using the 'crossjoin' function: `RANKX(crossjoin(ALL(Orders2[State]),ALL(Orders2[City])),[Total Sales])`. The formula bar has a red error icon. A tooltip says 'Can't display the visual. See details'. The Fields pane on the right lists various dimensions and measures.

The screenshot shows the Power BI Data view. The 'RANKX Function in DAX' query is displayed, showing a table 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 'Data' tab is selected. The Fields pane on the right lists various dimensions and measures.

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, the 'Mark as date table', 'Manage relationships', 'New measure', 'Quick measure', 'New column', and 'New table' buttons are visible. The main area displays a preview of the 'Top5States' table with 5 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	RB-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 shows the "Orders2" table with various columns like Category, CF Color, City, etc.

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 shows the "Orders2" table with various columns like Category, CF Color, City, etc.

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)' is being created, defined as `CALCULATE([Total Sales], ALL(Orders2[Region]))`. The visual below displays '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

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 top ribbon has the 'Data' tab selected. The same measure 'Sales (ALL)' is present, and the visual shows 'Region Sales' with 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

The same filters ('Ship Mode' and 'Category') are applied. 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

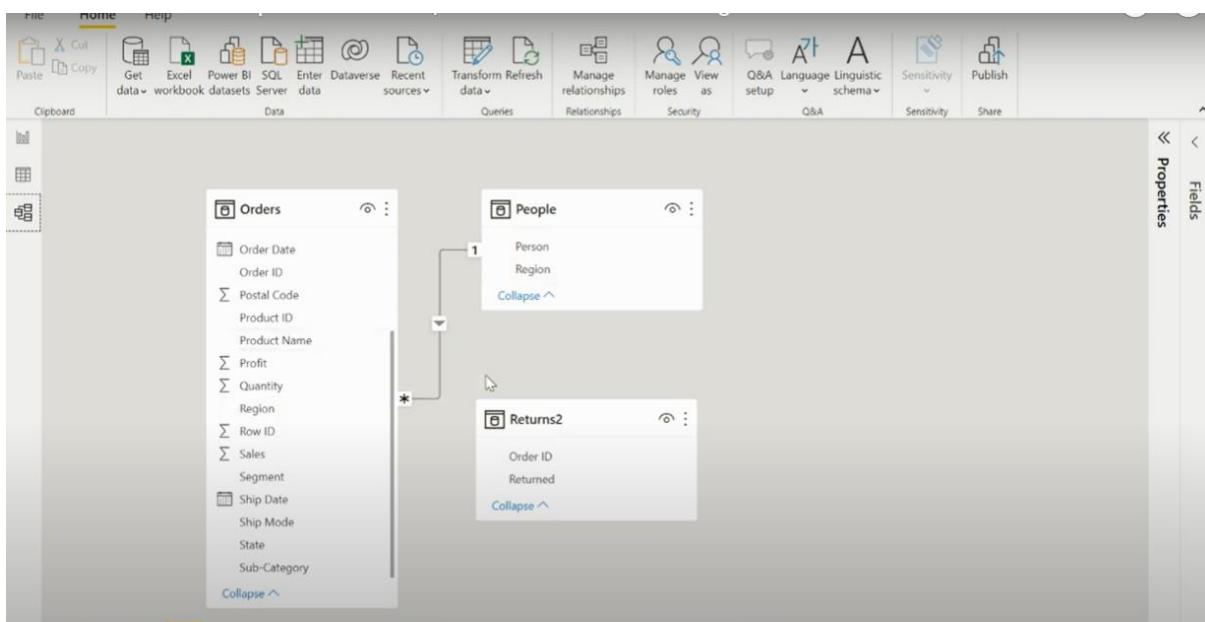
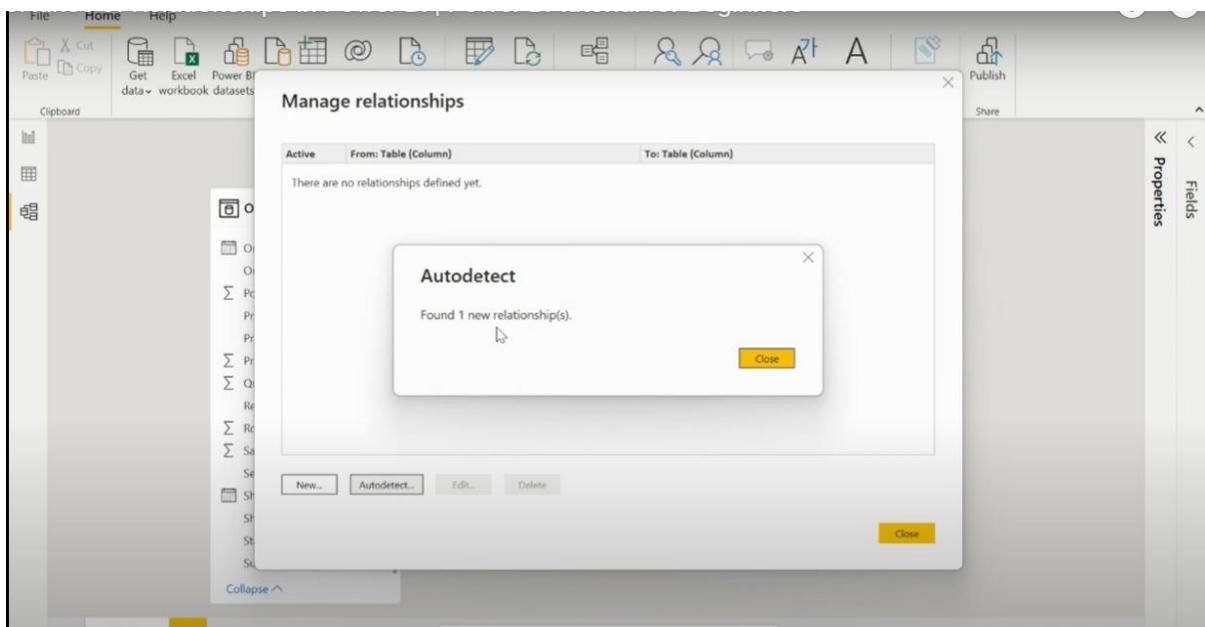
The screenshot shows the Power BI desktop application. On the left, there's a navigation pane with icons for Home, Get data, Data, Transform data, Manage relationships, Q&A, Language setup, Sensitivity, and Share. The main area displays a report titled "Importance of Relationships in Power BI". The report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

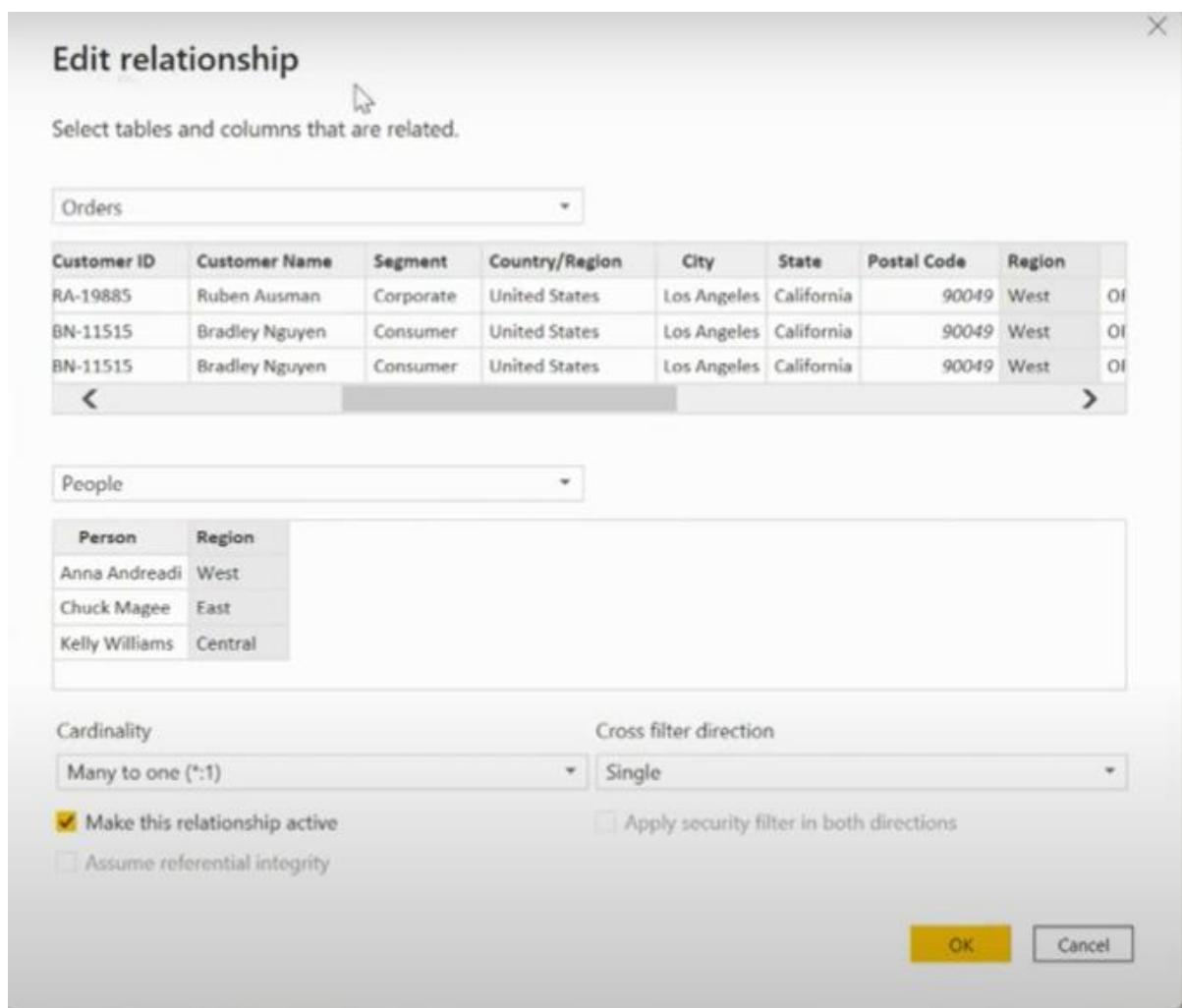
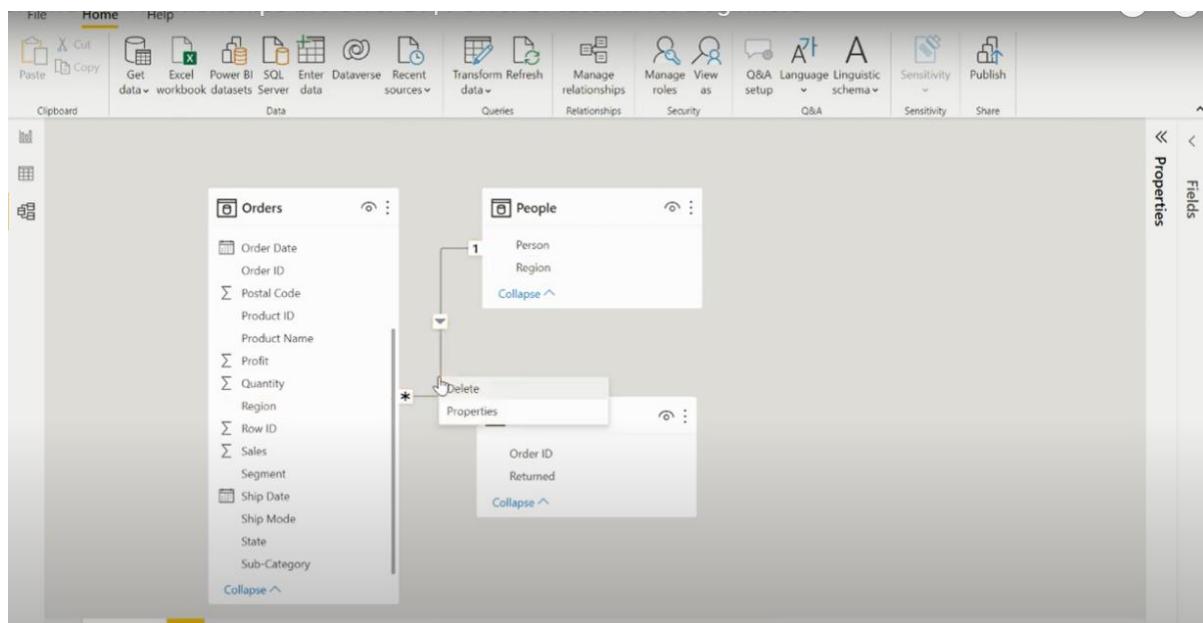
- Orders** table fields:
 - Order Date
 - Order ID
 - Postal Code
 - Product ID
 - Product Name
 - Profit
 - Quantity
 - Region
 - Row ID
 - Sales
 - Segment
 - Ship Date
 - Ship Mode
 - State
 - Sub-Category
- People** table fields:
 - Person
 - Region
- Returns2** table fields:
 - Order ID
 - Returned



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

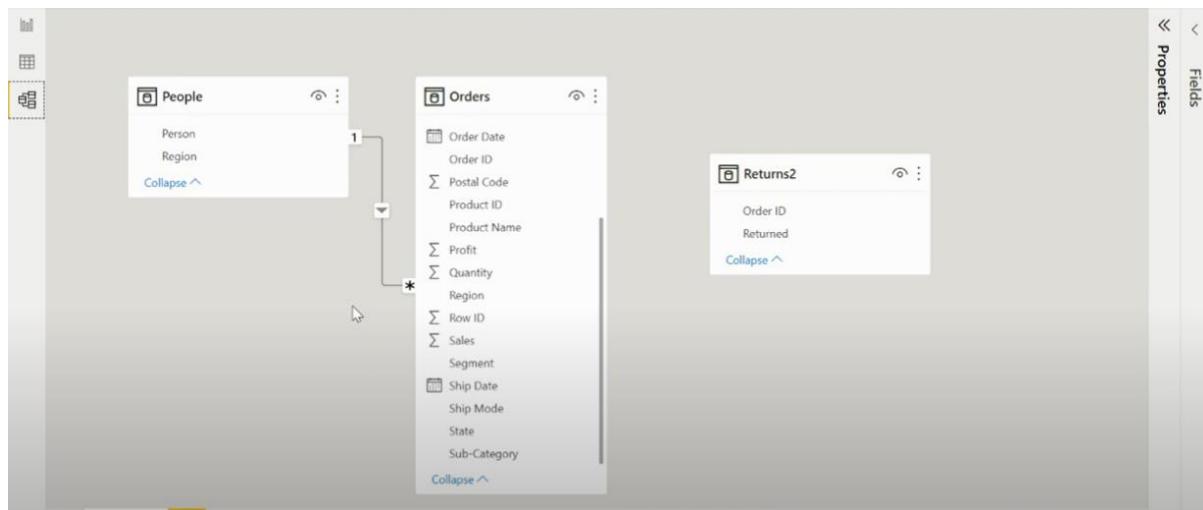
Cardinality and One-to-many relationship in Power BI

· 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. The Sales table is shown below the Doctor and Patient tables.

Person	Sales
Anna Andreadi	22,97,200.86
Cassandra Brant	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 the following text:
· 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.
Below this is a diagram showing a 'Doctor' table and a 'Patient' table connected by a one-to-many relationship. The 'Doctor' table has fields 'DoctorID' (int, PK) and 'DoctorName' (varchar(40)). The 'Patient' table has fields 'PatientID' (int, PK), 'PatientName' (varchar(40)), and 'Doctor_DoctorID' (int, FK).
At the bottom, there is a visual representation of a table titled 'Person Sales' with the following 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.

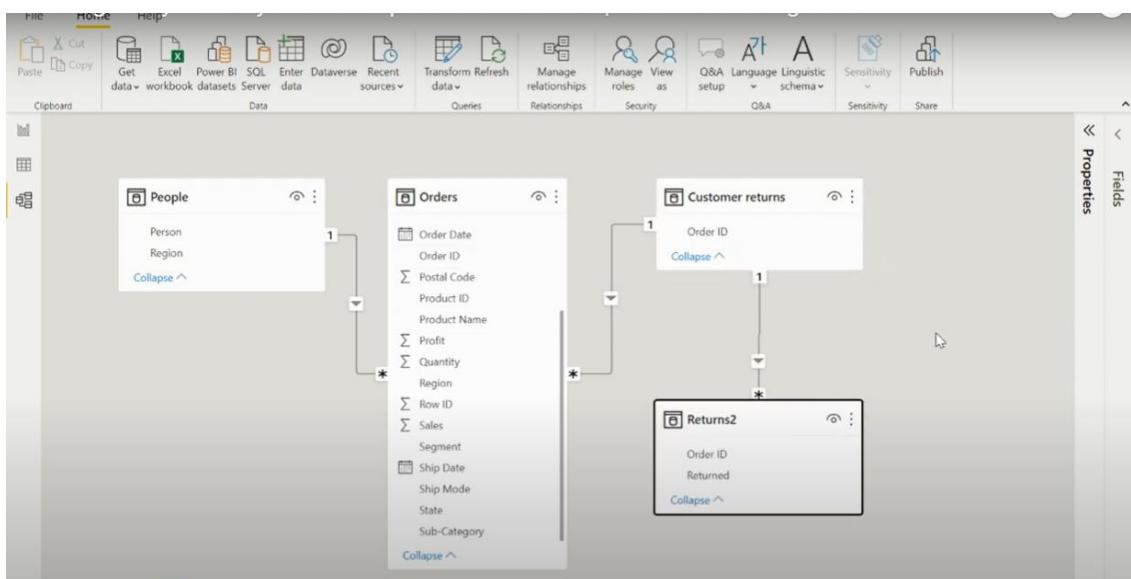
Order ID

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'.

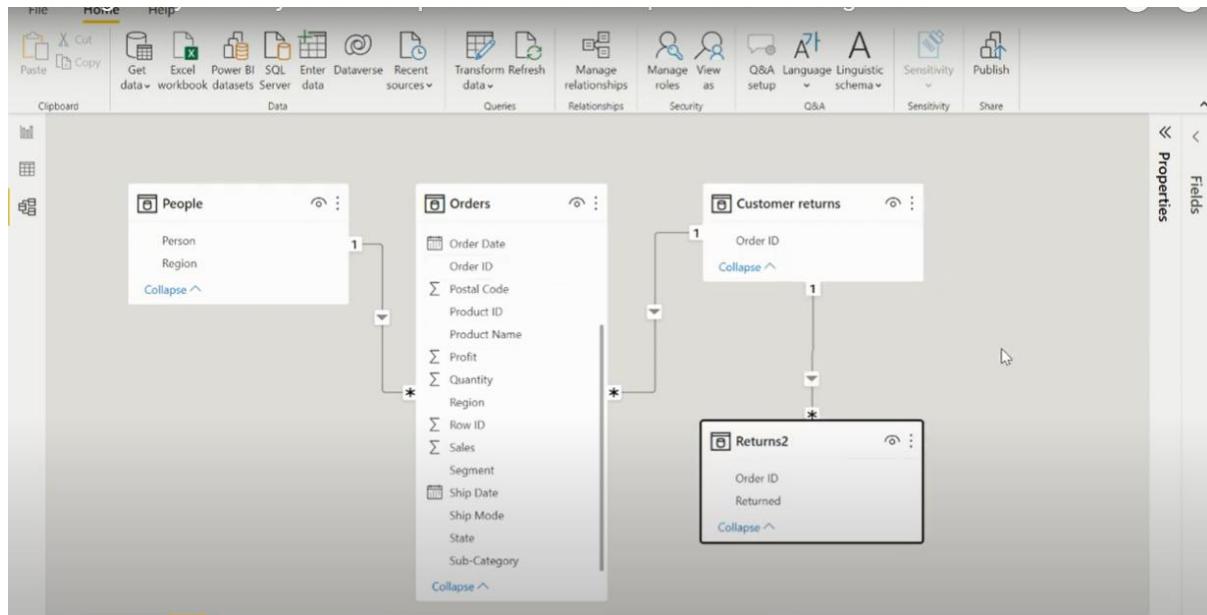
Order ID
CA-2020-152156
CA-2020-138688
US-2019-108966
CA-2018-158182
CA-2021-114412
CA-2020-161389
US-2019-118983
CA-2018-105893
CA-2018-167164
CA-2018-143336
CA-2020-137330
US-2021-156909
CA-2019-106320
CA-2020-121755
US-2019-150630
CA-2021-107727
CA-2020-117590
CA-2019-117415
CA-2021-120999
CA-2020-101343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



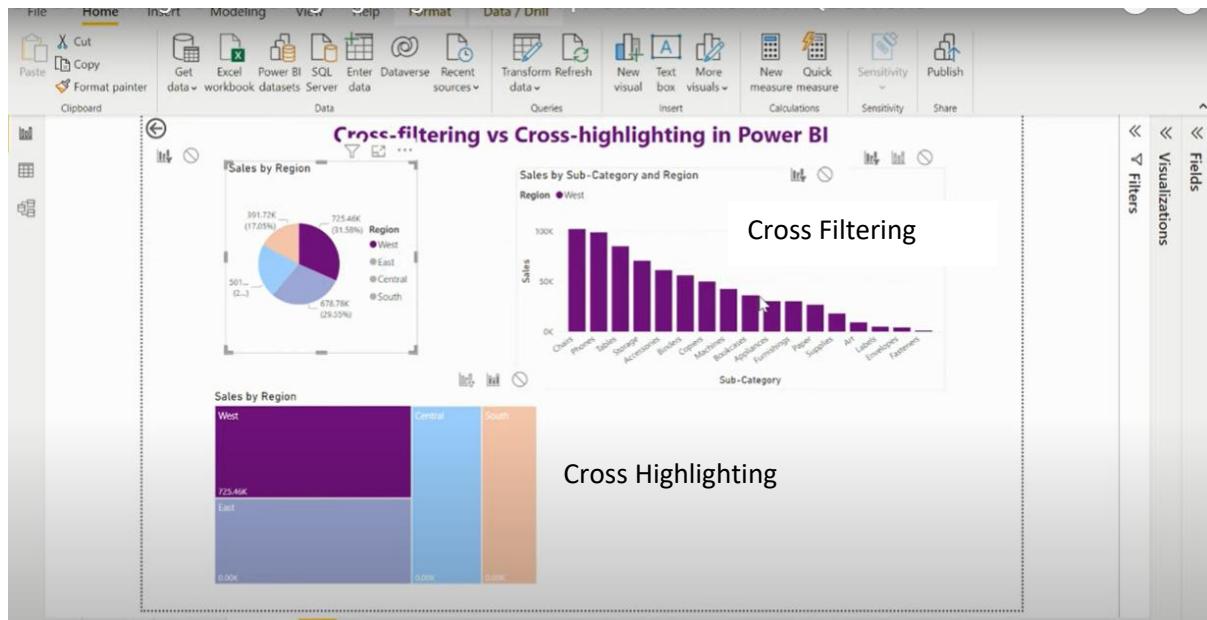
The screenshot shows the Power BI Query Editor interface. A red circle highlights the 'Properties' section on the right, which displays the name 'Customer returns'. Below it, the 'Applied Steps' section lists the steps taken: 'Source', 'Removed Other Columns', and 'Removed Duplicates'. The main area 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.

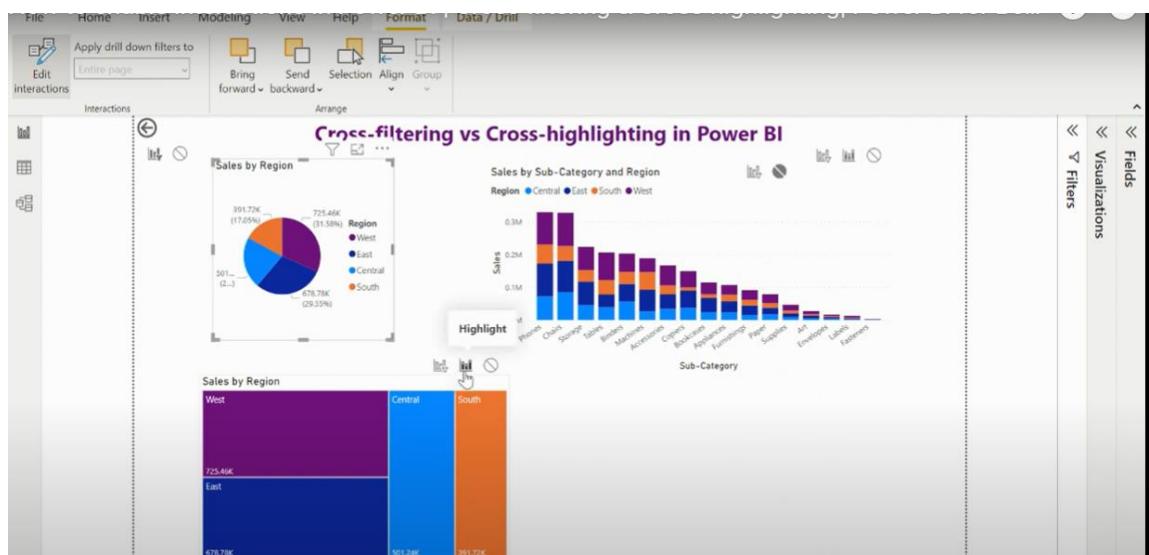
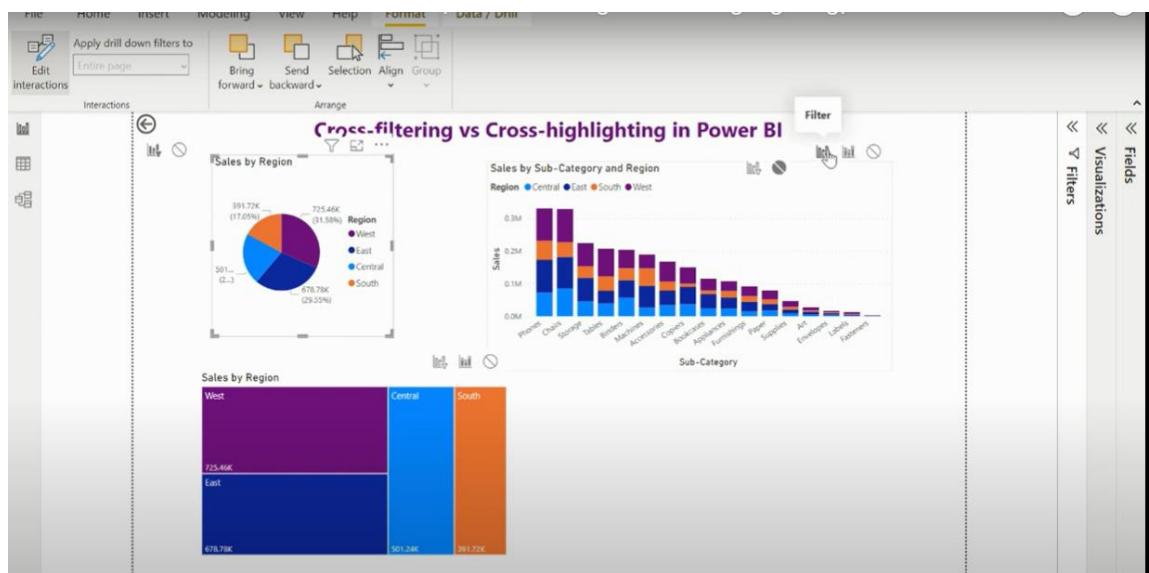
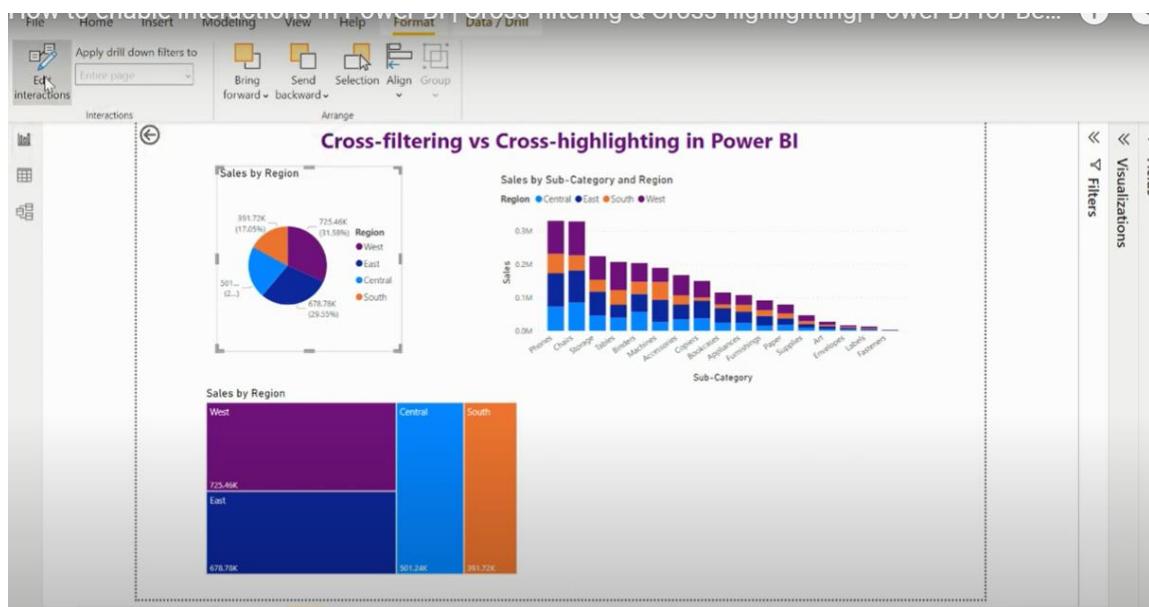


Cross filtering v/s cross highlighting



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 includes a note: '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.' On the right side, the 'Fields' pane is open, showing various tables like 'Customer returns', 'Orders', and 'Measure'. The 'Orders' table is expanded, showing fields such as Category, City, Country/Region, Order ID, Product ID, Product Name, Profit, Quantity, Region, and several summation fields like $\sum \text{Discount}$, $\sum \text{Measure}$, and $\sum \text{Profit}$.

Add Card visualization -> select newly created measure

The screenshot shows the Power BI Model view again. The 'Fields' pane on the right has the 'Measure' measure selected. In the 'Visualizations' pane, the 'Card' icon is highlighted with a red circle. The 'Card' visualization is added to the report canvas, displaying the value '470.53K' under the heading 'Total YTD Sales'. The 'Year' filter on the right shows options for 2018, 2019, 2020, and 2021, with 2019 selected.

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 each card is a small description: "Total YTD Sales" and "Total Year to date Sales". To the right of the cards is a "Year" slicer with options for 2018, 2019, 2020, and 2021. The "2020" option is selected. On the far right 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 also listed here.

This screenshot is identical to the one above, showing the same report title, cards, and slicer. The only difference is the value displayed in the first card: "Total YTD Sales" is now "484.25K". The rest of the interface, including the Fields pane, remains the same.

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 dropdown menu for 'Interval' is open, showing options like 'Year', 'Quarter', 'Month', etc. Below the formula bar, the message 'Too few arguments were passed to the DATEDIFF function. The minimum argument is DAY.' is displayed. The Fields pane on the right lists various fields including 'days_in_ytd'.

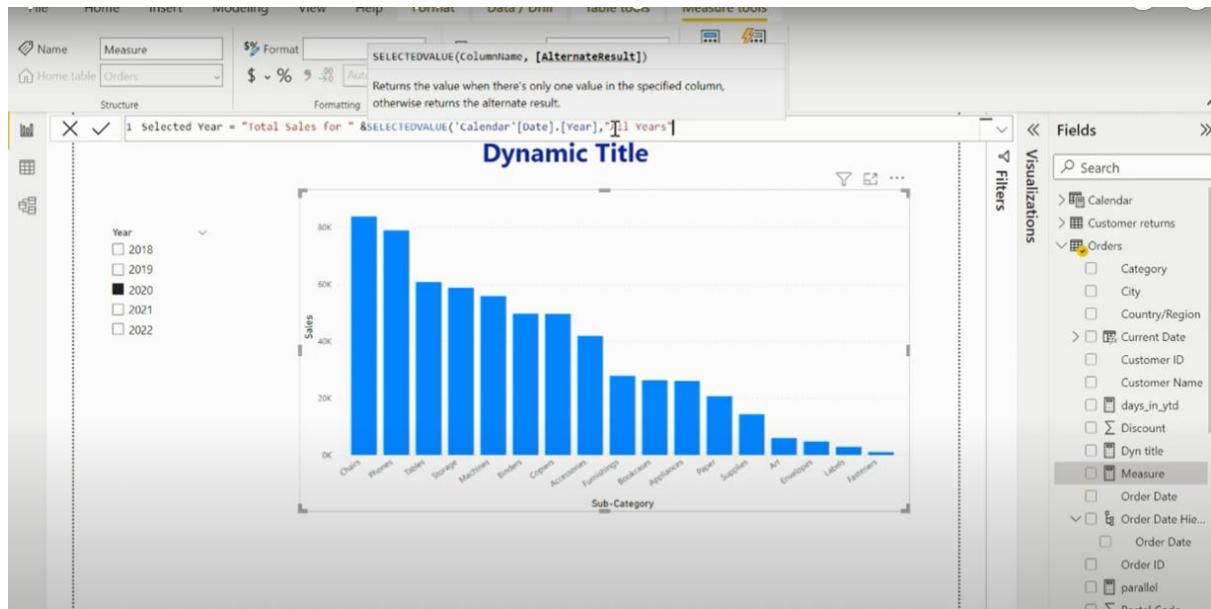
Add Card visualization -> select newly created measure

The screenshot shows the Power BI Home view. A Card visualization displays the value '210' for the measure 'days_in_ytd'. The visualization also includes the formula: `datediff(date1, date2, day)` with `date1: 01-01-2022` and `date2: 30-07-2022 =today()`. The Fields pane on the right shows the selected field 'days_in_ytd'.

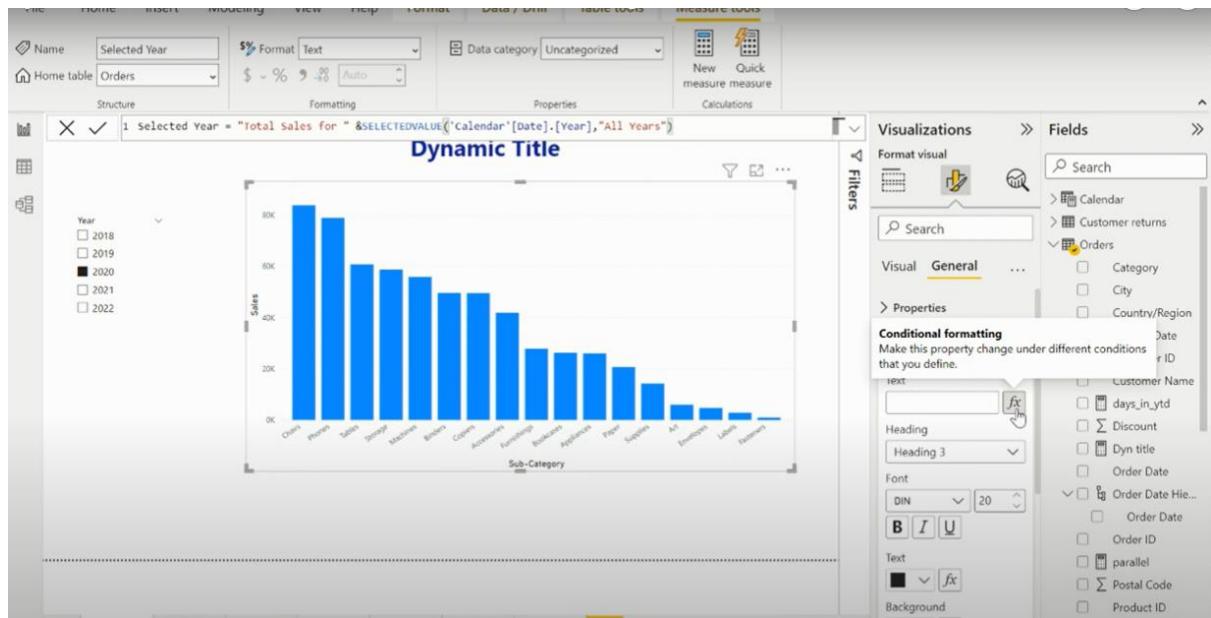
Dynamic Titles in Power BI?

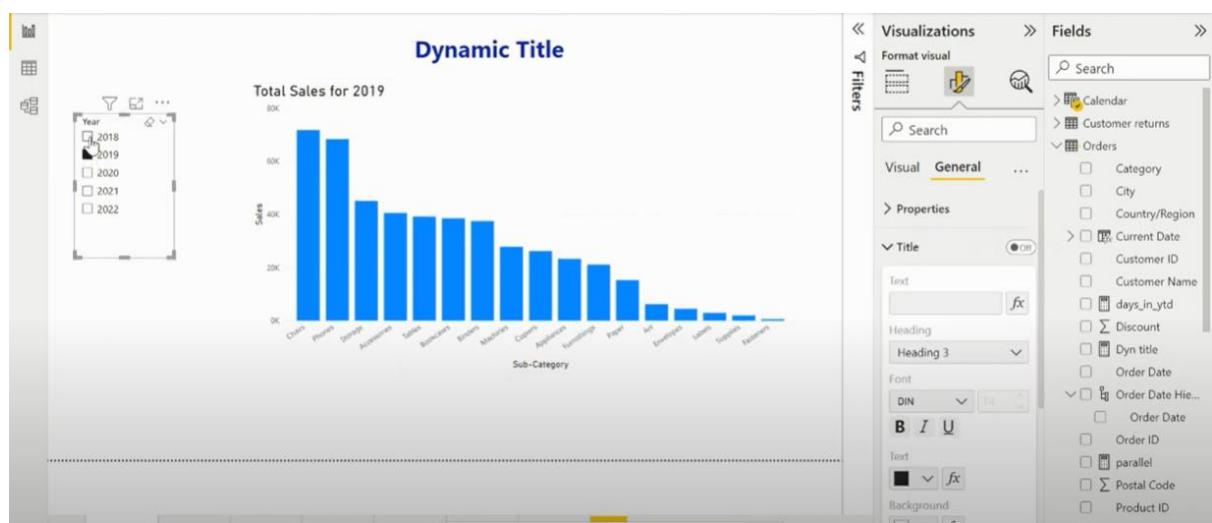
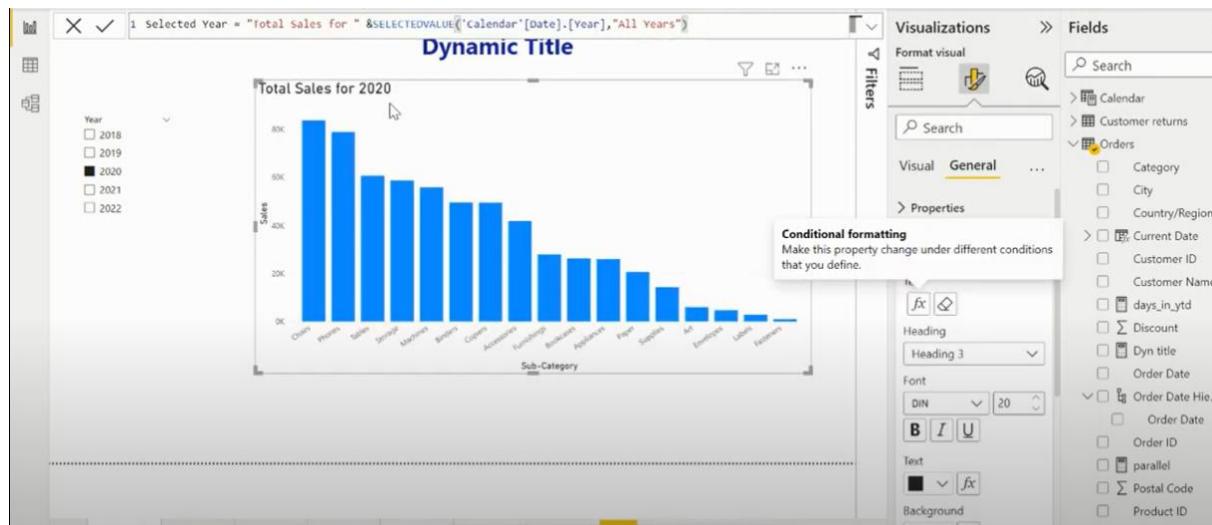
We want to update the visualization table, according to the slicer year selection.

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:

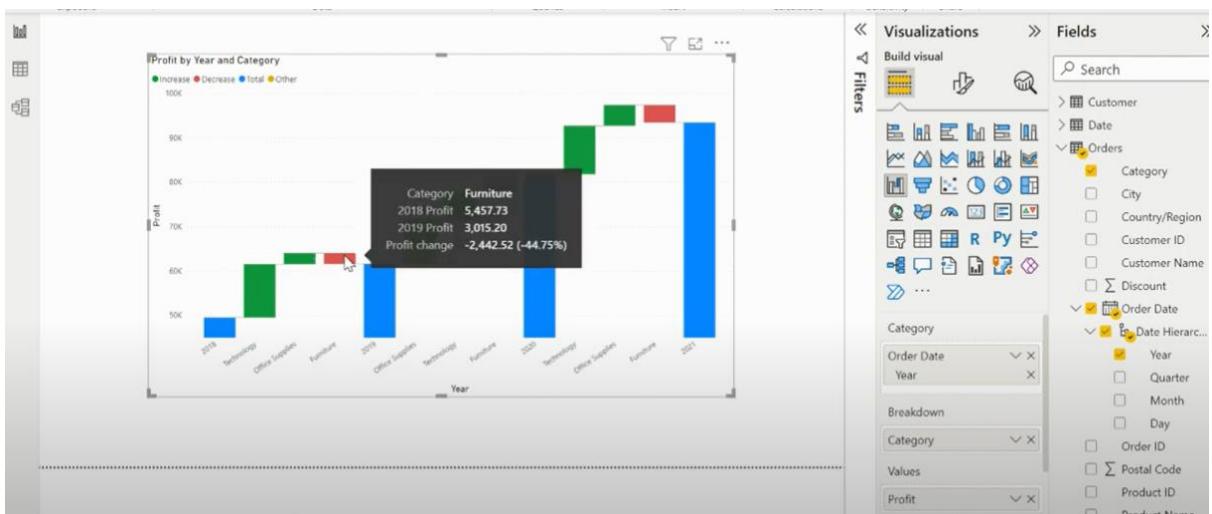
- Top Left:** A large numerical value **97.26M** labeled **Total Product Cost**.
- Top Right:** A large numerical value **110.34M** labeled **TotalSaleAmount**.
- Table:** A table showing sales details:

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.

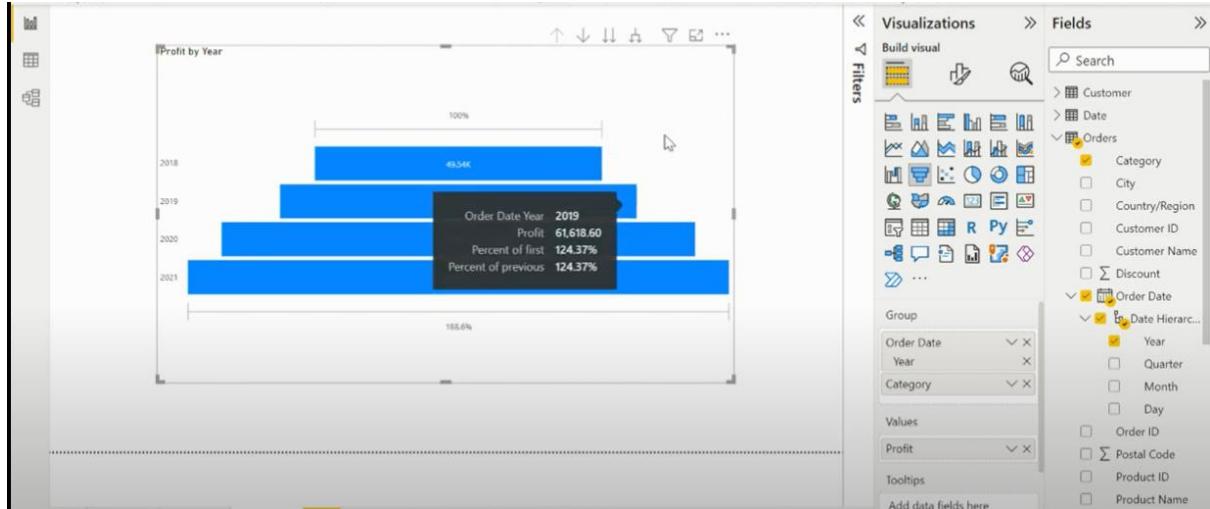
Waterfall Chart

Helps to visualize the change in value over a time period



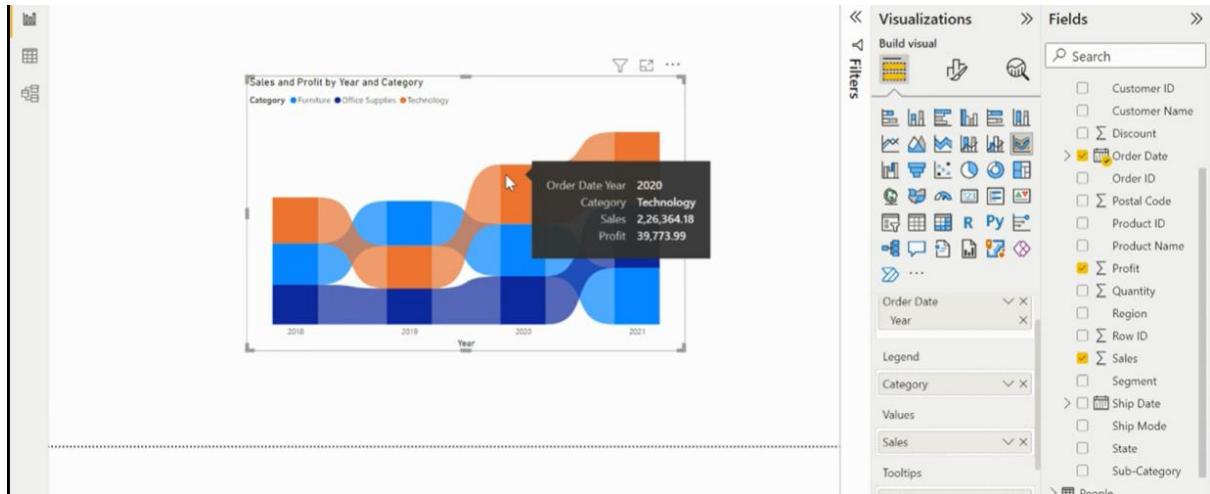
Funnel Chart

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 is being created with the name 'Measure'. 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 is shown with columns 'Year', 'Month', 'Sales', and 'parallel'. The 'parallel' column contains the formula 'PARALLELPERIOD(Orders, -1, MONTH)'. The table data is as follows:

Year	Month	Sales	parallel
2018	January	43,971.37	96,999.04
2019	February	20,301.13	43,971.37
2020	March	58,872.35	20,301.13
2021	April	36,521.54	58,872.35
2022	May	44,261.11	36,521.54
	June	52,981.73	44,261.11
	July	45,264.42	52,981.73
	August	63,120.89	45,264.42
	September	87,866.65	63,120.89
	October	77,776.92	87,866.65
	November	1,18,447.83	77,776.92
	December	83,829.32	1,18,447.83
	Total	7,33,215.26	7,46,384.98

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

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

The screenshot shows the Power BI 'Measure tools' ribbon. A new measure is being created with the name 'Measure'. The formula bar shows '\$ % > CALCULATE(SUM(Orders[sales]),PARALLELPERIOD('calendar'[Date],-1,month))'. A tooltip for 'PARALLELPERIOD' is displayed, showing its definition as 'Returns a parallel period of dates by the given set of dates and a specified interval.' Below the formula, a table is shown with columns 'Year', 'Month', 'Sales', and 'parallel'. The 'parallel' column contains the formula 'PARALLELPERIOD(Dates, NumberofIntervals, Interval)'. The table data is as follows:

Year	Month	Sales	parallel
2018	January	43,971.37	96,999.04
2019	February	20,301.13	43,971.37
2020	March	58,872.35	20,301.13
2021	April	36,521.54	58,872.35
2022	May	44,261.11	36,521.54
	June	52,981.73	44,261.11
	July	45,264.42	52,981.73
	August	63,120.89	45,264.42
	September	87,866.65	63,120.89
	October	77,776.92	87,866.65
	November	1,18,447.83	77,776.92
	December	83,829.32	1,18,447.83
	Total	7,33,215.26	7,46,384.98

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 it returns a set of dates in the current selection from the previous year. To the right, the 'Fields' pane is open, showing various columns like Month, Sales, and Parallel Period LM.

Month	Sales	Parallel Period LM
January	18,542.49	74,919.52
February	22,978.82	18,542.49
March	51,715.88	22,978.82
April	38,750.04	51,715.88
May	56,987.73	38,750.04
June	40,344.53	56,987.73
July	39,261.96	40,344.53
August	31,115.37	39,261.96
September	73,410.02	31,115.37
October	59,687.75	73,410.02
November	79,411.97	59,687.75
December	96,999.04	79,411.97
Total	6,09,205.60	5,87,126.08

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 open, providing the same description. The table below shows the same monthly sales data as before. The tooltip also displays the detailed structure of the 'Date' column from the 'Calendar' table, including Date, Day, Month, MonthNo, QuarterNo, Quarter, and Year.

Month	Sales	Parallel Period LM
January	18,542.49	47,053.25
February	22,978.82	47,053.25
March	51,715.88	47,053.25
April	38,750.04	47,053.25
May	56,987.73	47,053.25
June	40,344.53	47,053.25
July	39,261.96	47,053.25
August	31,115.37	47,053.25
September	73,410.02	47,053.25
October	59,687.75	47,053.25
November	79,411.97	47,053.25
December	96,999.04	47,053.25
Total	6,09,205.60	5,87,126.08

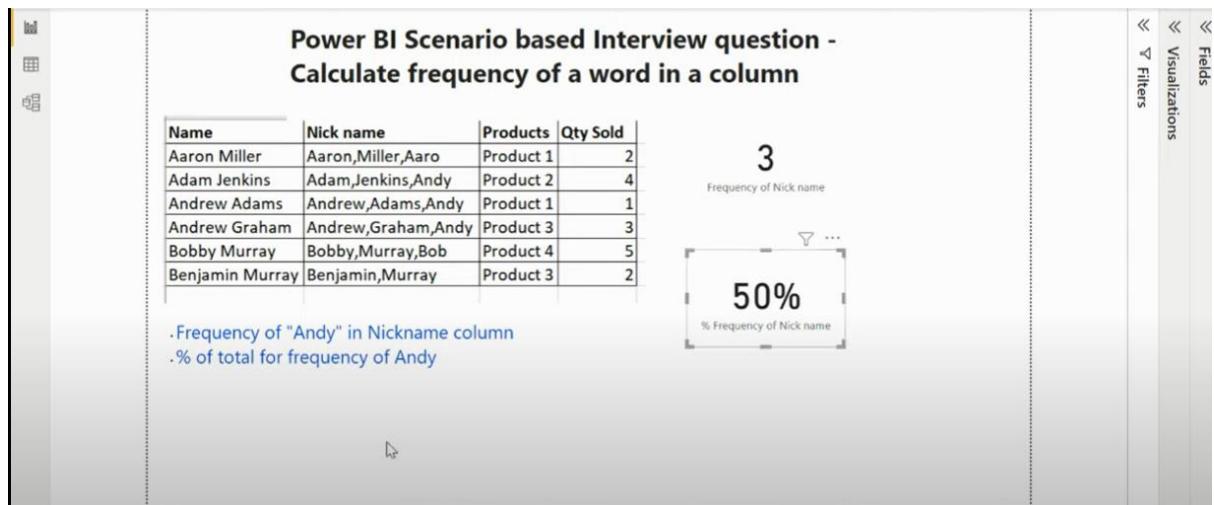
Parallelperiod (-12,month) will also give same result as same period but sameperiod is more reliable function in case of data for some months are missing

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))

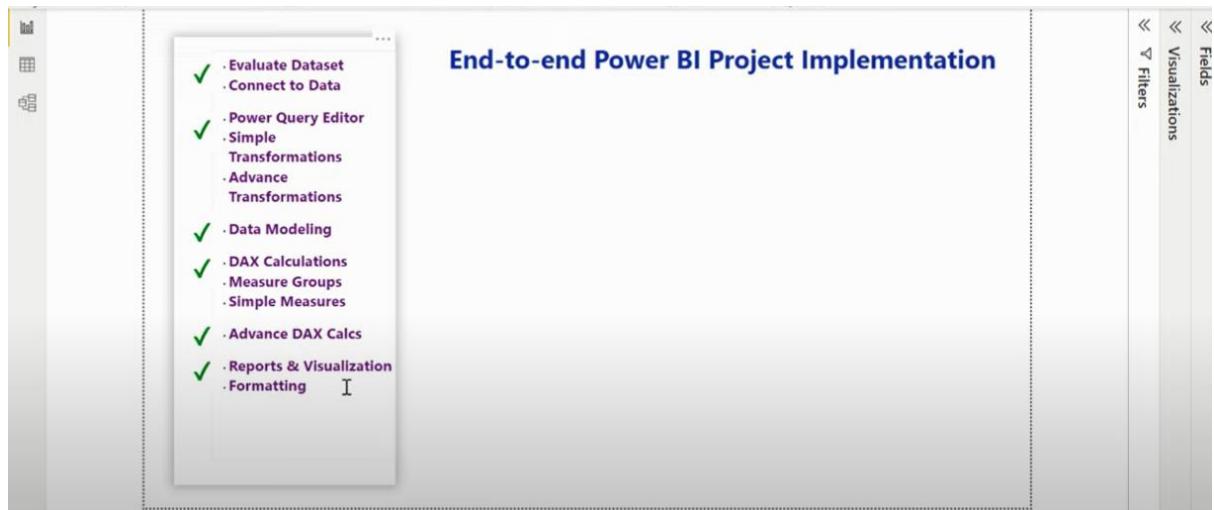
Visualizations:

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

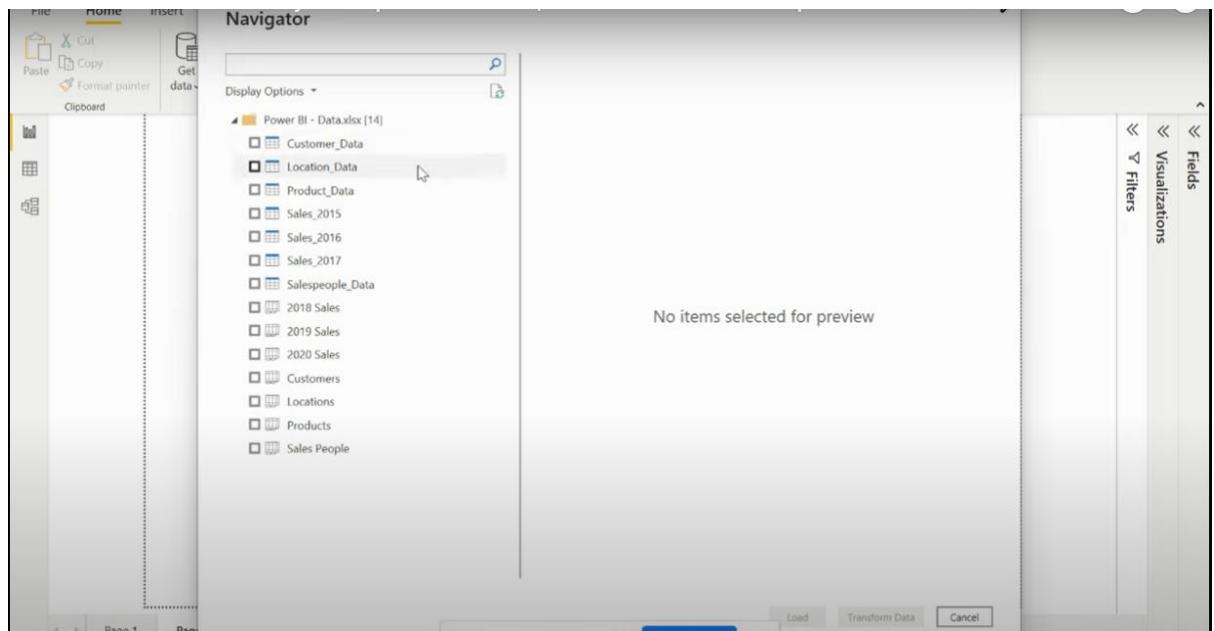
Fields:

- 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 is open 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 sections for Close & Apply, New Source, Data Sources, Data Source settings, Manage Parameters, Refresh Preview, Advanced Editor, Properties, 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 'APPLIED STEPS' pane on the right lists a single step: '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 open with columns: Households, Median Income, Land Area, and Water Area. The 'Households' column is currently selected. A context menu is open over the 'Water Area' column, listing options such as Copy, Remove Columns, Remove Other Columns, Add Column From Examples..., Remove Duplicates, Remove Errors, Replace Values..., Fill, Change Type, Transform, Merge Columns, Sum, Product, Group By..., Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, Move, and a dropdown for 'America/Los Angeles'. The ribbon bar at the top and the 'APPLIED STEPS' pane on the right are visible, similar to the previous screenshot.

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 contains fields for 'Name' (set to 'Data model') and 'Description'. The 'OK' button is highlighted, indicating the next action.

The screenshot shows the 'Queries [7]' pane again. The 'Data model' group now includes the Salespeople query, and the 'Other Queries' group includes Sales_2019 and Sales_2020. This indicates that the Salespeople query has been successfully moved into its own group.

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'.

In this final screenshot, the 'State - Copy' column has been moved to the second position in the table using a drag-and-drop operation. The table now has columns: '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 Data Editor interface. A context menu is open over a column named 'State'. The 'Format' option is selected, revealing a submenu with various text transformation tools: lowercase, uppercase, Capitalize Each Word, Trim, Clean, Add Prefix, and Add Suffix. The 'uppercase' option is currently highlighted. In the bottom right corner, the 'APPLIED STEPS' pane lists the steps taken so far, including 'Source', 'Navigation', 'Changed Type', 'Removed Columns', 'Duplicated Column', and '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 Data Editor interface. A context menu is open over a column named 'State'. The 'Split Column' option is selected, revealing a submenu with 'By Delimiter', 'By Number of Characters', 'By Position', and 'By Lowercase'. The 'By Delimiter' option is currently highlighted. Within this submenu, 'Split Column by Delimiter' is selected, with a tooltip explaining it splits values in the selected column based on the specified delimiter. The 'APPLIED STEPS' pane lists the steps taken so far, including 'Source', 'Navigation', 'Changed Type', 'Removed Columns', 'Duplicated Column', 'Reordered Columns', and '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 ₁ C State Code	A ₁ C State	A ₁ C State - Copy	A ₁ 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 ₁ C State Code	A ₁ C State	A ₁ C State - Copy.1	A ₁ C State - Copy.2	A ₁ 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	CDP
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 desktop interface. In the center is a query editor displaying a table with columns: 'State', 'State - Copy.1', 'State - Copy.2', and 'Type'. The 'Type' column contains values like 'City', 'CDP', and 'City'. On the left, the 'Queries' pane shows a tree structure with 'Data model [5]', 'Locations', 'Products', 'Sales_2018', 'Salespeople', and 'Other Queries [2]'. A 'Web' icon is selected. On the right, the 'Properties' and 'Applied Steps' panes are visible, showing the step 'Changed Type1'.

This screenshot is similar to the one above, but the 'From Web' dialog box is open in the foreground. It shows a URL field with 'https://www.california-demographics.com/cities_by_population'. The 'Basic' radio button is selected. The rest of the interface is identical to the first screenshot.

This screenshot shows the 'Navigator' feature in Power BI. The 'Locations' item is selected in the 'Queries' pane. In the center, a 'Table View' window displays a table titled 'Table 1' with columns 'Rank', 'City', and 'Population'. The data includes rows for Oakland, Bakersfield, Anaheim, Santa Ana, Riverside, Stockton, Irvine, Chula Vista, Fremont, San Bernardino, Modesto, Fontana, Santa Clarita, Moreno Valley, Oxnard, Huntington Beach, Glendale, Santa Rosa, Ontario, Rancho Cucamonga, Oceanside, and El Cajon. The 'Properties' and 'Applied Steps' panes are also present on the right.

Queries [8]

Data model [5]

- Customers
- Locations
- Products
- Sales_2018
- Salespeople

Other Queries [3]

- Sales_2019
- Sales_2020
- Table 1**

Table 1

Rank

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

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:

- Top Bar:** Includes Close & Apply, New Source, Recent Sources, Enter Data, 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, Merge Queries, Append Queries, Text Analytics, Vision, Combine Files, Azure Machine Learning, Combine, and AI Insights.
- Left Sidebar:** Shows the "Data model [6]" section with "Locations" selected. Other items include Customers, Products, Sales_2018, Salespeople, CA Population, and Other Queries [2].
- Query Editor:** Displays a table with columns: Location ID, Name, County, State Code, and State. The table has 21 rows of California city data.
- Properties Panel:** Shows the "Name" field set to "Locations".
- Applied Steps Panel:** Lists the steps applied to the query, including "Changed Type" (highlighted).

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 [6]

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

= Table.NestedJoin(#"Changed Type1", {"Name"}, #"CA Population", {"City"}, "CA Population", CA Population

	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

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

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

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

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

Queries [8]

Data model [6]

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

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

Applied Steps

Queries [8]

Data model [6]

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

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

Applied Steps

Queries [8]

Data model [6]

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

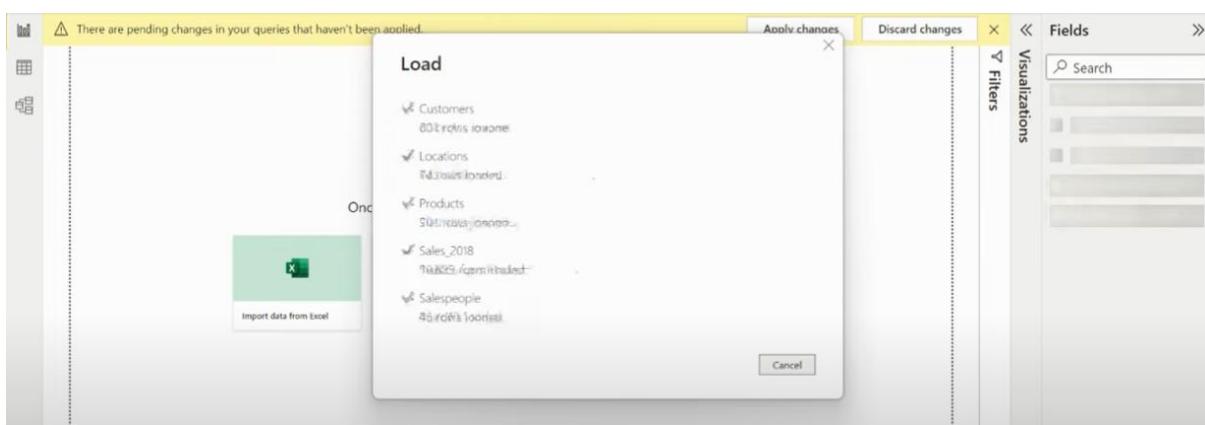
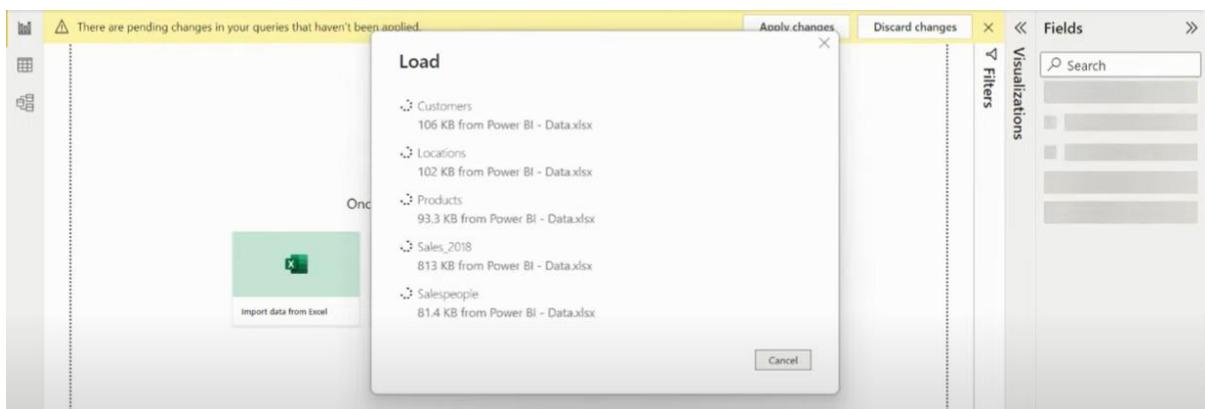
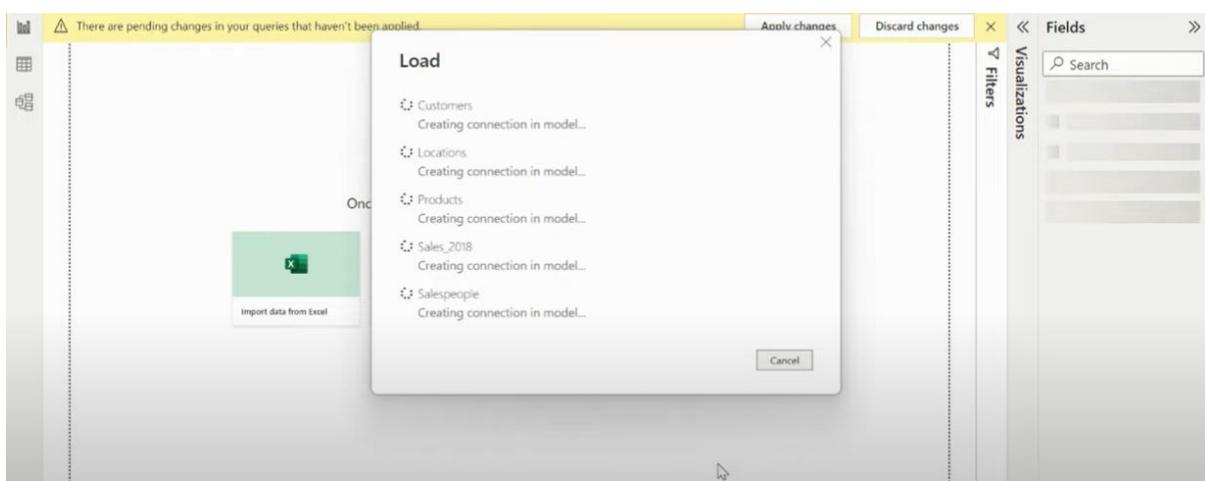
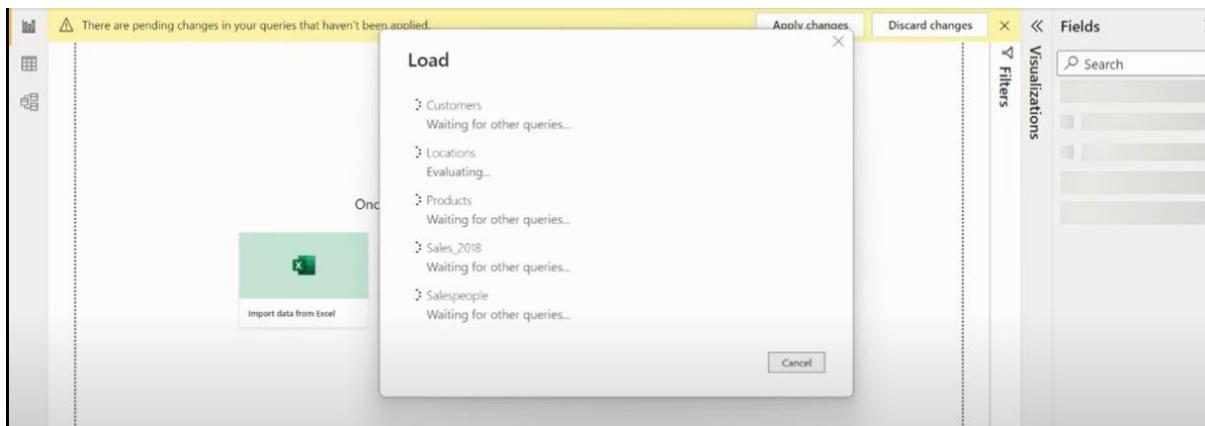
Rank	City	Population
1	Los Angeles	3973278
2	San Diego	1414545
3	San Jose	1029409
4	San Francisco	874784
5	Fresno	526147
6	Sacramento	503482
7	Long Beach	462081
8	Oakland	422575
9	Bakersfield	379879
10	Anaheim	353085
11	Santa Ana	332610
12	Riverside	327569
13	Stockton	311103
14	Irvine	272694
15	Chula Vista	268779
16	Redding	234829
17	Palo Alto	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	198201

Query Settings

Properties

Applied Steps

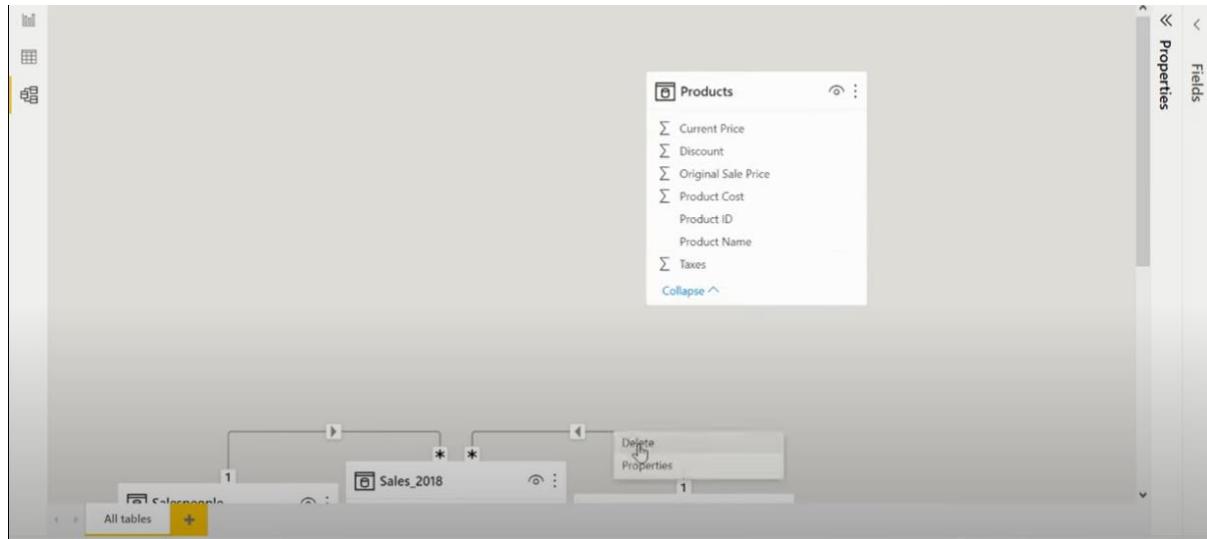
Once transformations are done, click on Close & apply then data will be loaded into Power BI



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 tables: Customers, Dates (which is selected), Locations, Products, Sales_2018, and Salespeople.

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

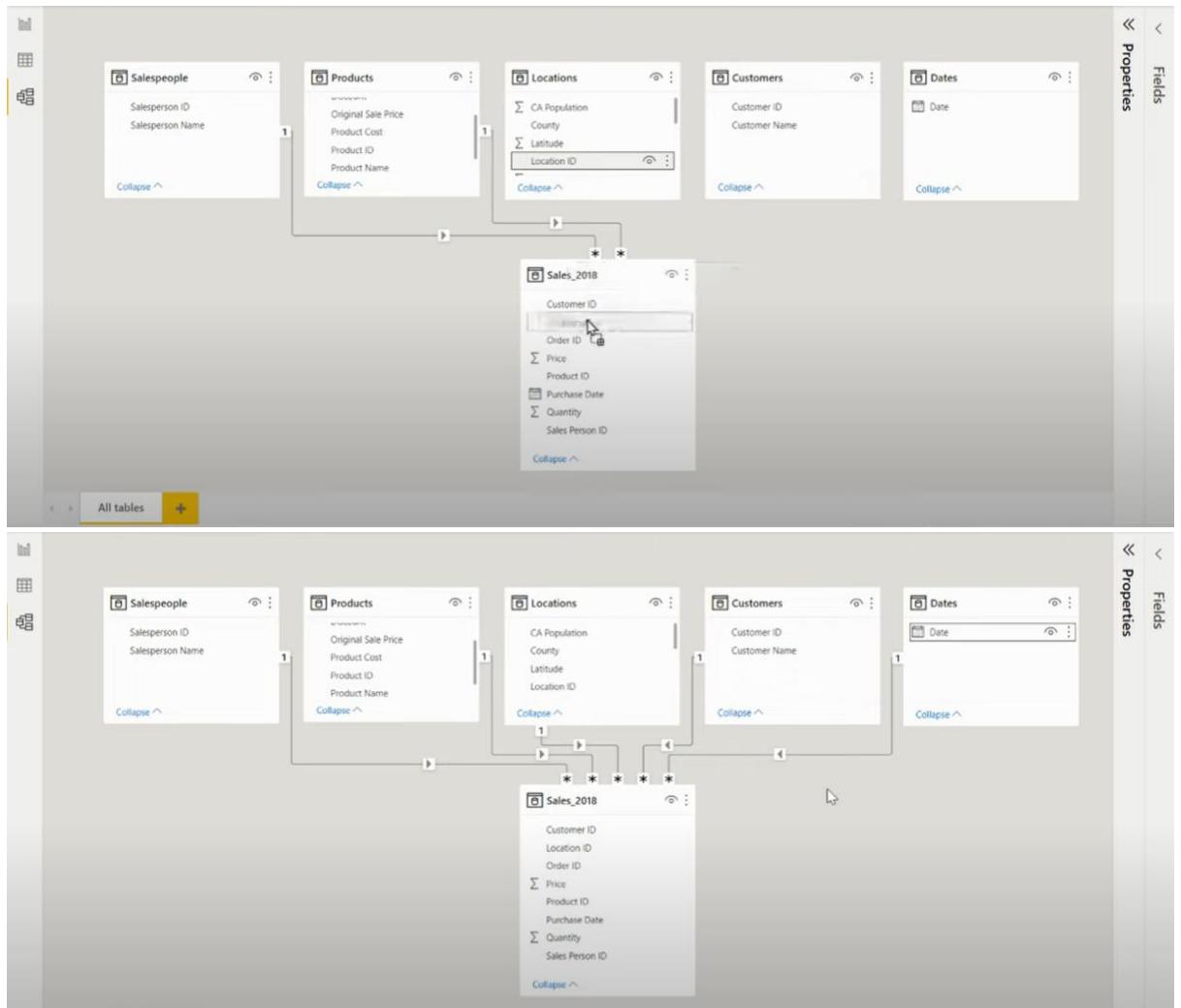
Lookup table (data table) – holds the primary keys

Fact table – holds the transactions of the lookup tables

The screenshot shows the Power BI Data Model view with five tables: 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 'Formatting' section shows 'Whole number' selected. The 'Data category' is 'Uncategorized'. The 'Calculations' section displays the formula: `old = SUM(Sales_2018[Quantity])`. The 'Key Measures' section lists various measures grouped by category: Customers, Dates, Locations, Products, Sales_2018, and Salespeople. The 'Fields' pane on the right shows the 'Quantity Sold' measure under the 'Key Measures' group.

This screenshot is similar to the first one, but the 'Quantity Sold' measure is now selected in the 'Key Measures' group. A checkmark is visible next to it in the list. The rest of the interface remains the same, including the formula, categories, and Fields pane.

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 expression `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 for 20 salespeople, with the total sales value being 21078.

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	21078

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

This screenshot shows the Power BI Data Editor with the 'Format' tab selected in the ribbon. The 'Measure tools' tab is still active, and the 'Total Sales' measure is defined with the same expression as before. The Fields pane on the right shows the 'Key Measures' group, including 'Quantity Sold' and 'Total Sales'. The main area displays the same sales data table as the first screenshot.

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 tables like Customers, Dates, Locations, Products, and Sales_2018, with "Total Transactions" selected.

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 tables like Customers, Dates, Locations, Products, and Sales_2018, with "Measure" selected under the "Customers" node.

	Quantity	Price	Total Cost
Aaron Carr	26	24	\$26.846
Aaron Cruz	32	26	\$32.099
Aaron Day	16	16	\$22.123
Aaron Johnson	16	9	\$18.038
Aaron Miller	20	11	\$28.716
Aaron Mills	20	13	\$17.891
Aaron Moreno	27	15	\$27.996
Aaron Tucker	38	16	\$38.158
Adam Alexander	32	21	\$32.187
Adam Bailey	48	32	\$48.930
Adam Duncan	34	30	\$34.143
Adam Hernandez	18	21	\$18.793
Adam Hunter	21	9	\$21.762
Adam Jenkins	50	38	\$50.294
Adam Mccoy	31	22	\$31.402
Adam McDonald	27	11	\$18.967
Adam Myers	30	24	\$30.997
Adam Riley	45	26	\$45.066
Adam Thompson	39	29	\$39.779
Adam Wheeler	18	9	\$10.411
Adam White	18	21	\$18.967
Alan Gomez	26	9	\$26.129
Alan Green	36	13	\$36.911
Alan Miller	51	31	\$51.804
Alan Parker	39	36	\$39.137
Alan Perry	44	38	\$44.752
Alan Scott	52	43	\$52.058
Alan Smith	34	56	\$34.043
Total	21078	1089	\$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	Total Profits	Total Costs	Total Transactions
Aaron Carr	\$20,946	11	24	
Aaron Cruz	\$32,099	16	26	
Aaron Day	\$22,123	10	16	
Aaron Johnson	\$18,038	9	16	
Aaron Miller	\$28,716	11	20	
Aaron Mills	\$17,891	13	20	
Aaron Moreno	\$27,996	15	26	
Aaron Tucker	\$38,158	16	26	
Adam Alexander	\$32,187	11	21	
Adam Bailey	\$48,930	19	32	
Adam Duncan	\$34,143	17	30	
Adam Hernandez	\$18,793	13	21	
Adam Hunter	\$21,762	9	17	
Adam Jenkins	\$50,294	19	38	
Adam McCoy	\$31,402	12	22	
Adam McDonald	\$27,727	11	18	
Adam Myers	\$30,997	11	24	
Adam Riley	\$45,066	15	26	
Adam Thompson	\$39,779	14	29	
Adam Wheeler	\$18,411	9	18	
Adam White	\$18,967	11	21	
Alan Gomez	\$26,129	9	19	
Alan Green	\$36,911	13	31	
Alan Miller	\$51,804	23	38	
Alan Parker	\$39,137	18	36	
Alan Perry	\$44,752	18	38	
Alan Scott	\$52,058	22	43	
Alan Sims	\$34,741	17	26	
Total	\$2,56,61,209	10889	21078	

Using 2 measures to create 3rd measure

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

Customer Name	Total Sales	Total Profits	Total Costs	Total Transactions
Aaron Carr	\$26,846	\$17,146	11	24
Aaron Cruz	\$32,099	\$20,715	16	26
Aaron Day	\$22,123	\$16,046	10	16
Aaron Johnson	\$18,038	\$11,812	9	16
Aaron Miller	\$28,716	\$19,256	11	20
Aaron Mills	\$17,891	\$12,519	13	20
Aaron Moreno	\$27,996	\$15,594	15	26
Aaron Tucker	\$38,158	\$26,291	16	26
Adam Alexander	\$32,187	\$23,439	11	21
Adam Bailey	\$48,930	\$33,488	19	32
Adam Duncan	\$34,143	\$24,473	17	30
Adam Hernandez	\$18,793	\$12,056	13	21
Adam Hunter	\$21,762	\$15,122	9	17
Adam Jenkins	\$50,294	\$26,954	19	38
Adam McCoy	\$31,402	\$21,636	12	22
Adam McDonald	\$27,727	\$19,470	11	18
Adam Myers	\$30,997	\$23,027	11	24
Adam Riley	\$45,066	\$34,057	15	26
Adam Thompson	\$39,779	\$24,656	14	29
Adam Wheeler	\$18,411	\$10,419	9	18
Adam White	\$18,967	\$12,625	11	21
Alan Gomez	\$26,129	\$16,208	9	19
Alan Green	\$36,911	\$23,836	13	31
Alan Miller	\$51,804	\$36,701	23	38
Alan Parker	\$39,137	\$25,094	18	36
Alan Perry	\$44,752	\$32,093	18	38
Alan Scott	\$52,058	\$35,767	22	43
Alan Sims	\$34,741	\$14,003	17	26
Total	\$2,56,61,209	\$1,73,17,316	10889	21078

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

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

Customer Name	Total Sales	Profit Margin	Total Profits	Total Costs	Total Transactions
Alan Sims	\$9,051	\$24,043	\$14,992	17	26
Alan Thomas	\$17,145	\$52,423	\$35,278	17	35
Albert Cunningham	\$14,110	\$44,945	\$30,835	17	38
Albert Jacobs	\$10,981	\$53,470	\$24,489	16	38
Albert Kennedy	\$10,227	\$33,240	\$23,013	11	25
Albert King	\$13,235	\$46,671	\$21,436	14	34
Albert Robinson	\$10,753	\$35,545	\$24,792	13	30
Albert Young	\$13,556	\$43,913	\$10,827	11	24
Andrew Burns	\$7,980	\$25,429	\$17,539	12	24
Andrew Burns	\$18,431	\$53,915	\$33,384	20	38
Andrew Butler	\$8,811	\$31,918	\$23,107	12	24
Andrew Fernandez	\$5,791	\$16,527	\$10,736	7	15
Andrew Graham	\$9,347	\$34,635	\$25,388	10	32
Andrew Hansen	\$9,822	\$28,823	\$19,000	12	23
Andrew James	\$7,643	\$23,815	\$16,172	12	24
Andrew Martin	\$13,114	\$40,179	\$26,665	18	42
Andrew Peters	\$15,385	\$41,127	\$25,762	18	33
Andrew Reynolds	\$15,307	\$41,646	\$26,639	12	30
Andrew Robinson	\$7,076	\$20,602	\$13,526	10	16
Anthony Banks	\$8,721	\$30,869	\$22,148	14	22
Anthony Berry	\$10,760	\$33,969	\$25,309	10	21
Anthony Chapman	\$10,804	\$30,305	\$19,501	14	24
Anthony Davis	\$9,843	\$32,196	\$19,979	9	26
Anthony Little	\$9,381	\$31,696	\$22,133	9	20
Anthony Parker	\$18,991	\$48,087	\$29,499	13	32
Anthony Simpson	\$4,170	\$18,031	\$13,861	10	15
Anthony Torres	\$10,576	\$31,017	\$20,441	14	23
Anthony Turner	\$16,477	\$47,706	\$31,129	15	34
Antonio Cooper	\$3,611	\$11,129	\$7,518	10	13
Total	\$83,43,893	\$2,56,61,209	\$1,73,17,316	10889	21078

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.

This screenshot shows the same Power BI environment after the data hierarchy has been removed. The table visual now displays the data without the hierarchical structure. The Fields pane remains the same, showing the 'Values' section with the 'Date' dropdown still open.

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 at the top and the Column Tools tab selected. In the Column Tools ribbon, the 'Data type' dropdown is set to 'Date/time'. Below it, the 'Formatting' group shows the current format as '14-03-2001 13:30...'. The main area shows a table with various columns and their current data types. The 'Date' column is highlighted with a red box, indicating it is being modified. The 'Formatting' group also includes 'Summarization' and 'Data category' dropdowns, and 'Properties' and 'Sort' buttons.

The screenshot shows the Power BI desktop interface. On the left is a data grid with columns for Customer Name, Total Profits, Total Sales, Total Costs, Total Transactions, Quantity Sold, and Profit Margin. The data includes rows for various customers like Alan Thomas, Andrew Burns, and Antonio Diaz, with corresponding values for each metric. On the right is the 'Fields' pane, which contains sections for 'Visualizations', 'Filters', 'Key Measures', 'Dates', 'Locations', 'Products', 'Salespeople', and 'Time Intelligence'. The 'Key Measures' section is expanded, showing items like Profit Margin, Total Sales, and Total Profits. The 'Dates' section is also expanded, showing a date hierarchy. The 'Values' section shows 'Date' and 'Total Sales' selected. The 'Drill through' and 'Cross-report' options are turned off.

Time Intelligence Calculations

This screenshot is similar to the one above, showing the same data grid and Fields pane. The data grid contains the same columns and rows of customer information. The Fields pane is also identical, showing the same sections and expanded 'Key Measures' and 'Dates' sections. The 'Values' section shows 'Add data fields here' and 'Add drill-through fields here'.

This screenshot shows the Power BI desktop interface with a focus on the formula bar. The formula bar displays the measure definition: `LY Sales = CALCULATE([Total Sales],SAMEPERIODLASTYEAR(Dates[Date]))`. Below the formula bar is a data grid with the same structure as the previous screenshots. To the right is the 'Fields' pane, which is identical to the ones in the other screenshots, showing the 'Visualizations', 'Filters', and various time intelligence sections. The 'Values' section shows 'Date' and 'Total Sales' selected.

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. To the right of the table is the "Fields" pane, which contains sections for "Key Measures", "Customers", "Dates", "Locations", "Products", "Salespeople", and "Time Intelligence". A "Filters" section is also present at the top of the pane.

To Calculate Cumulative Sales

This screenshot shows the same Power BI desktop environment as above, but with a calculated column added to the table. The calculated column is named "Cumulative Sales" and is defined by the DAX formula: `CUMULATIVE SALES = CALCULATE(SUM(TOTAL_SALES), FILTER(ALLSELECTED(Dates), Dates[Date] <= MAX(Dates[Date])))`. This formula filters the total sales table to include only dates up to the current date, summing them to get the cumulative total. The resulting table shows the cumulative sales for each customer, starting from \$0 and increasing to a total of \$97,01,923.

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	\$10,641	\$20,200
01-01-2020	\$10,995	\$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 shows a search bar for "county" and a list of fields under "Locations" including "County". The "Legend" section shows "County" and the "Values" section shows "Total Sales".

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

The screenshot shows 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, showing the "Visual" tab with "Bars" selected. Under "Data labels", the switch is turned on, and the "Position" dropdown is set to "Auto". The "Fields" pane on the right shows a search for "custo" and lists "Customers" and "Sales_2018" with "Customer ID" highlighted.

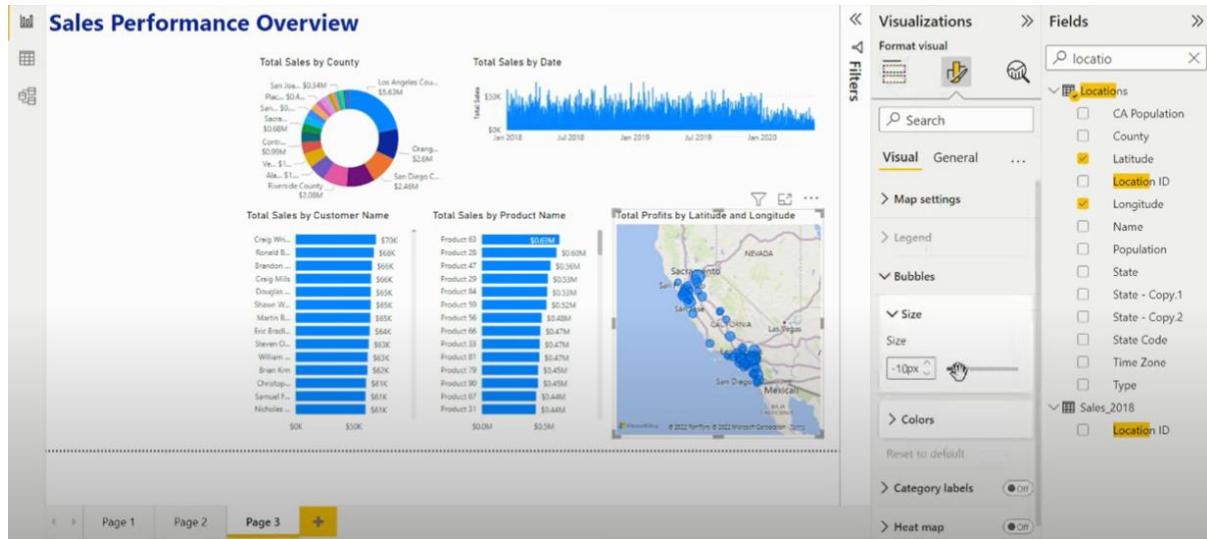
To show map as visual

The screenshot shows the same three visualizations as the previous screenshot. The "Visualizations" pane is open, showing the "Build visual" section with a map icon selected. The "Fields" pane shows a search for "produ" and lists "Products" and "Sales_2018" with "Product ID" highlighted.

Use latitude and longitude for location

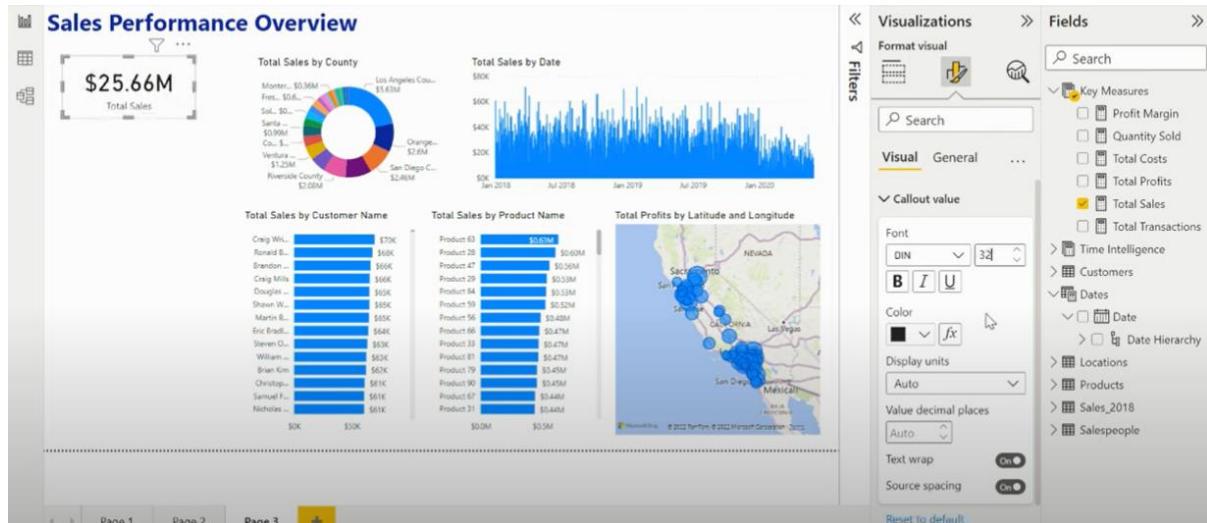
To increase the bubble size

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



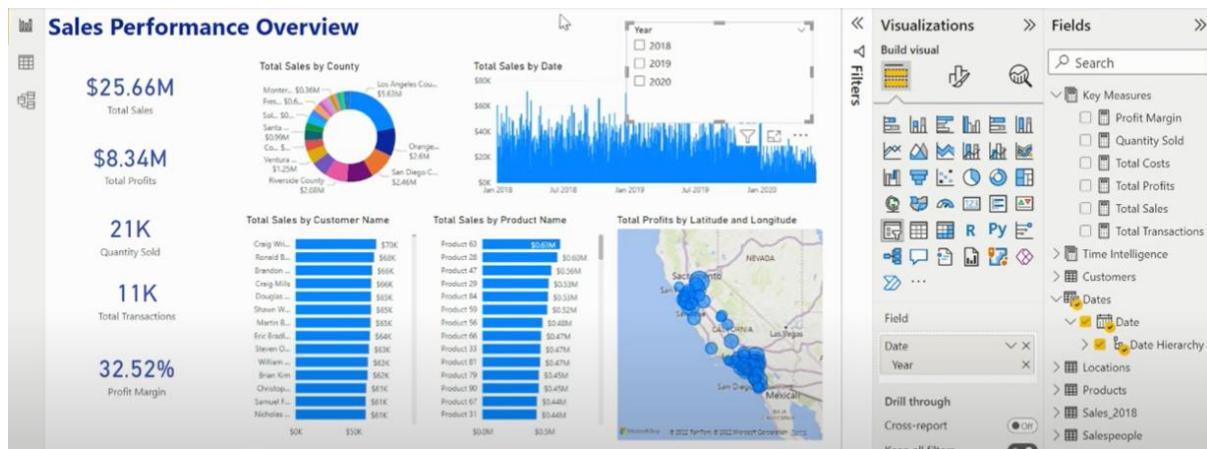
To format the cards

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

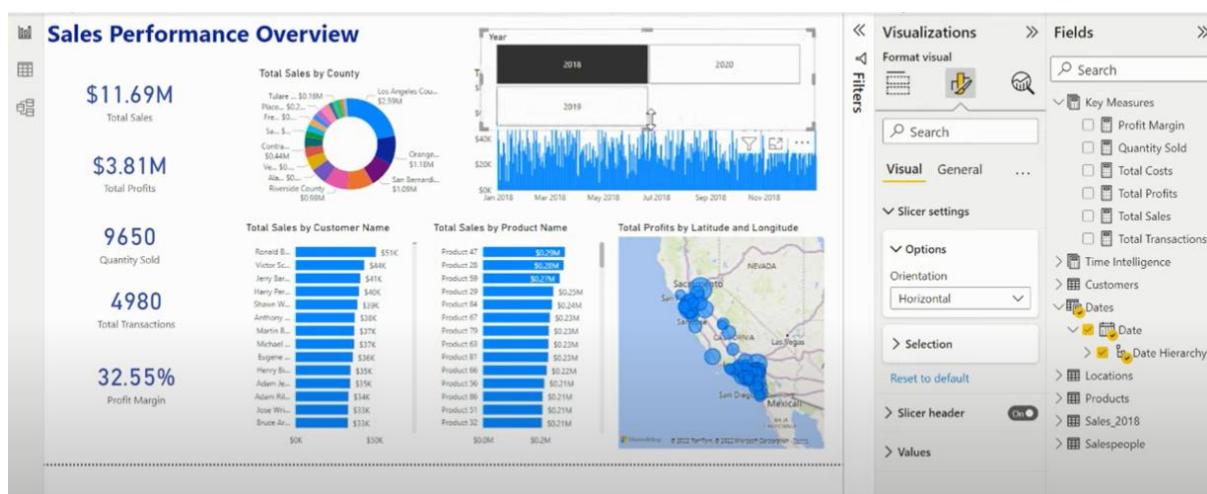
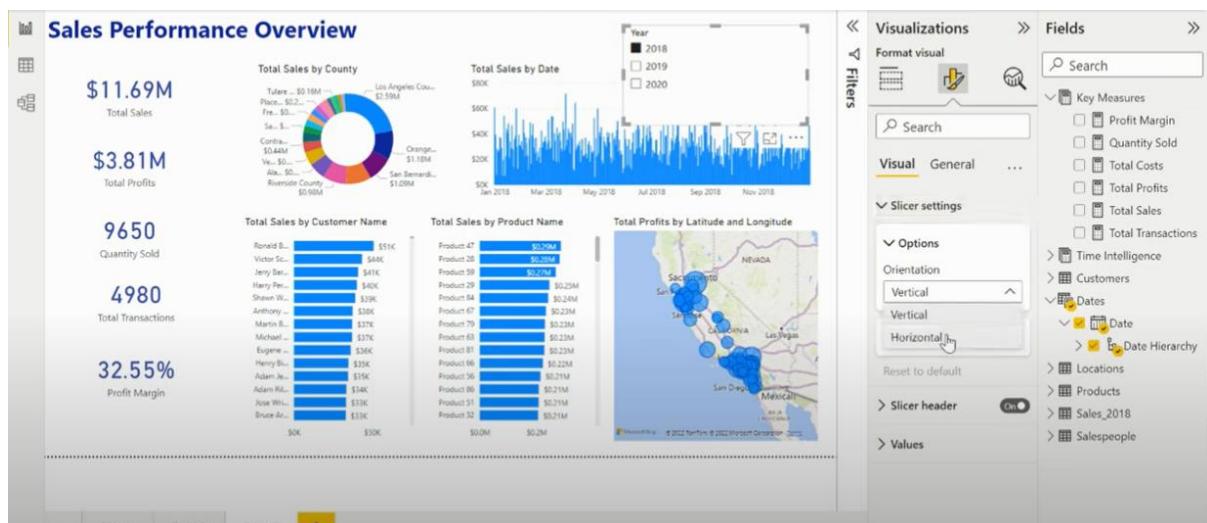


Insert slicer

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

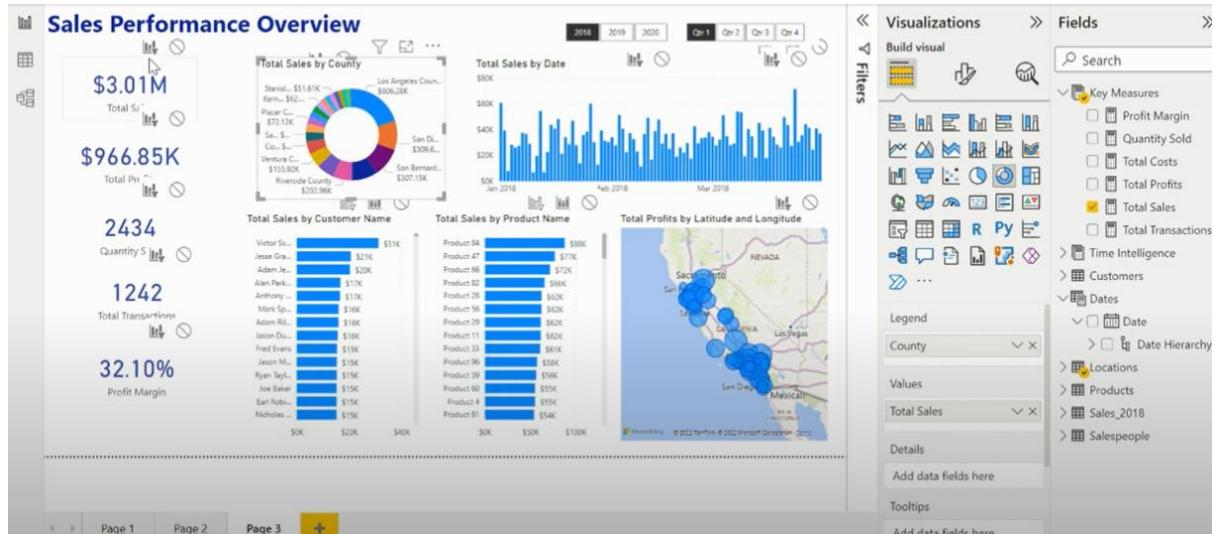


Select visual -> format visual -> visual -> slicer settings->options -> orientation -> horizontal -> adjust the slicer -> turn off slicer header



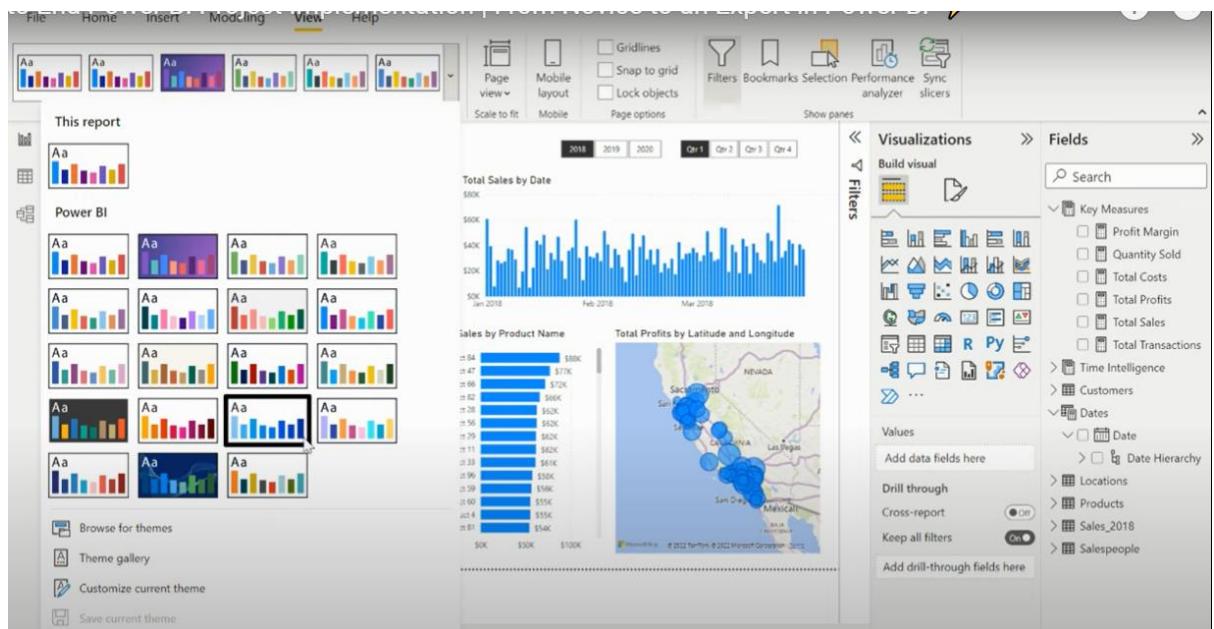
To format/highlight data as per selection in primary visual

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



To change theme

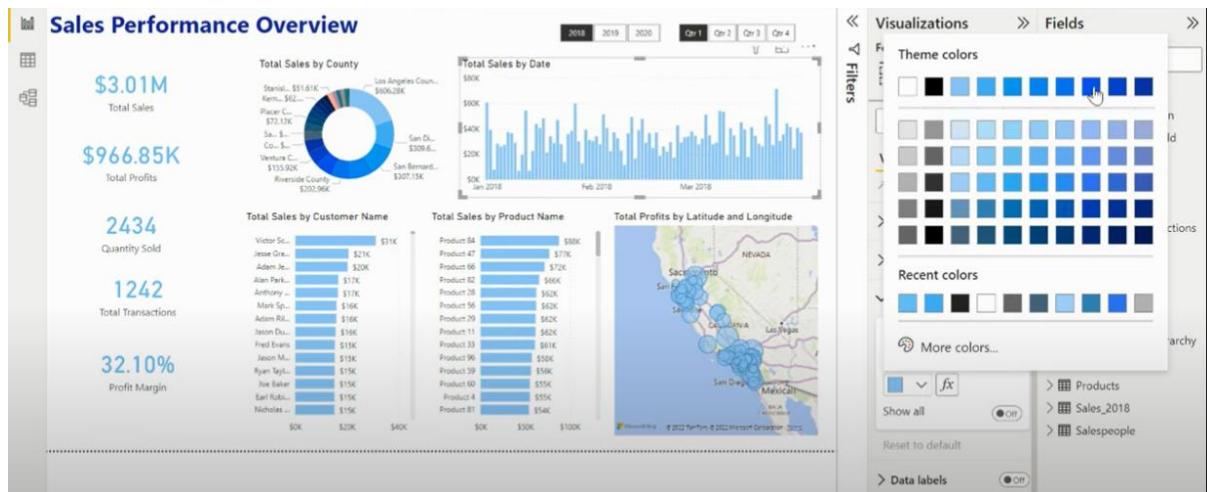
View -> themes group -> select theme



To change colour of visuals

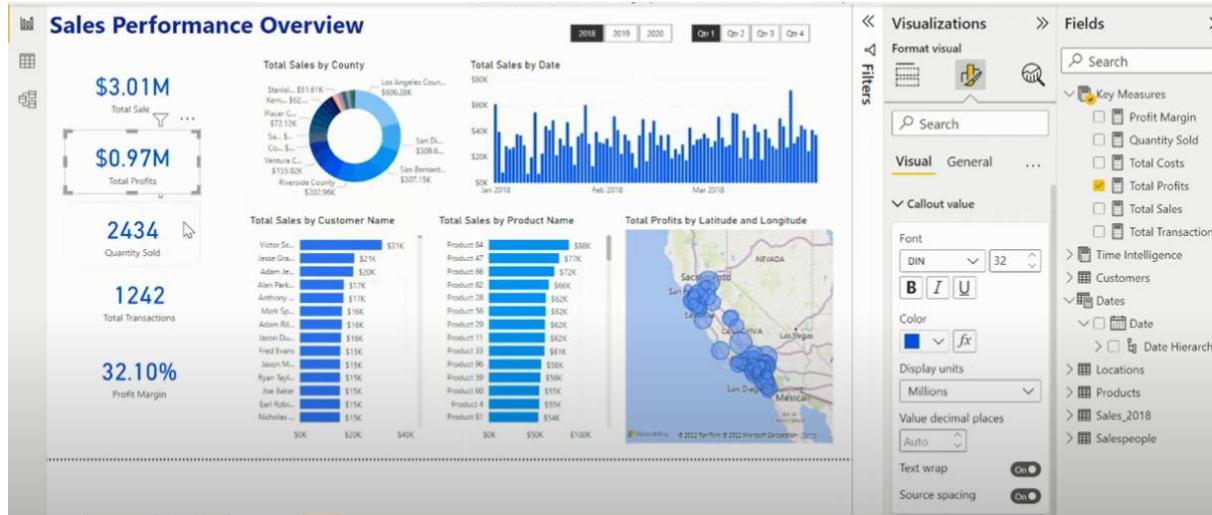
Select visual -> format visual -> visual -> columns -> color-> change colour

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



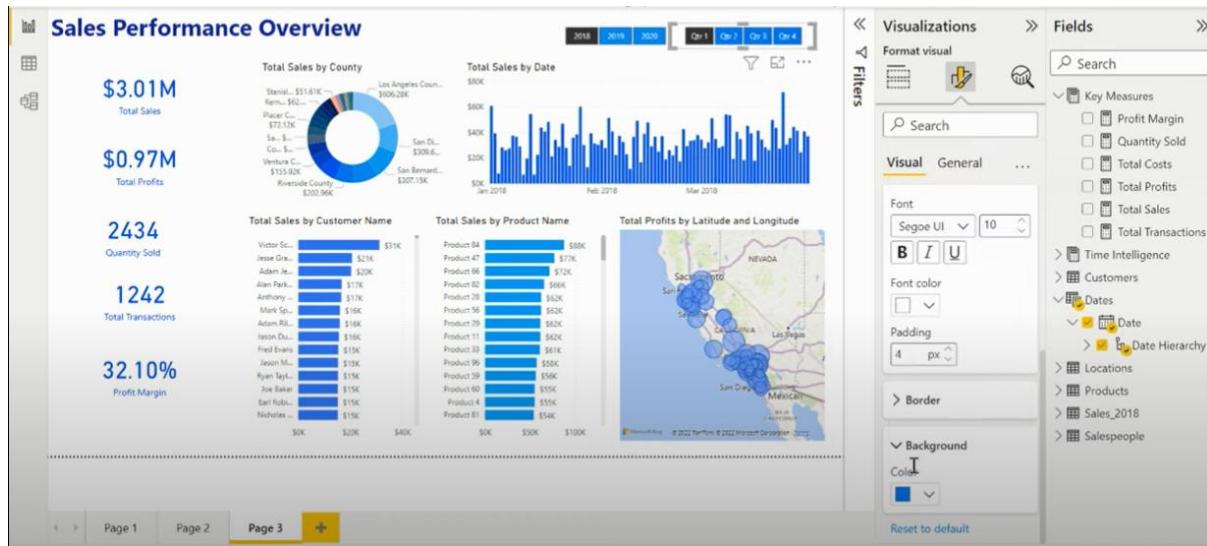
To change units

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



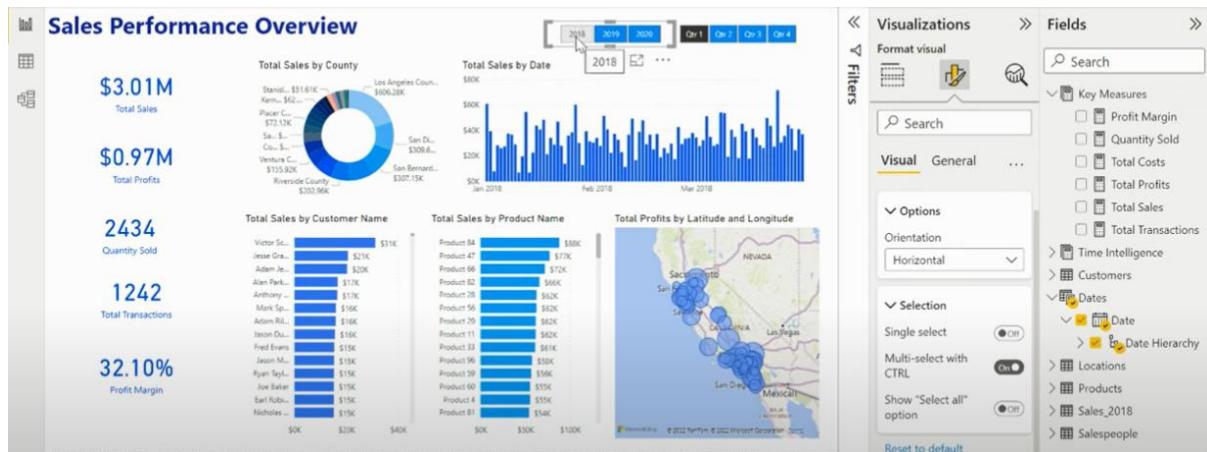
To change slicer colour

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, and a new step named "Changed Type1".

Splitting the incorrect date column

The screenshot shows the Power BI Desktop interface with a table containing four columns: Date.2, Date.3, Day, and Ocassion. The 'Split Column' context menu is open, with 'By Number of Characters' selected. Other options visible include 'By Delimiter', 'By Positions', 'By Lowercase to Uppercase', 'By Uppercase to Lowercase', 'By Digit to Non-Digit', and 'By Non-Digit to Digit'. The 'APPLIED STEPS' pane on the right shows the step 'Split Column by Delimiter'.

The screenshot shows the 'Split Column by Number of Characters' dialog box. It asks for the 'Number of characters' to split the text column, with a default value of 1. Below it, there are three radio button options: 'Once, as far left as possible' (selected), 'Once, as far right as possible', and 'Repeatedly'. At the bottom are 'OK' and 'Cancel' buttons.

The screenshot shows the Power BI Desktop interface with a table containing columns Day, weekday, Date.3.2, and day. The 'Rename Columns' context menu is open, with 'Renamed Columns' selected. Other options include 'Source', 'Navigation', 'Promoted Headers', 'Changed Type', 'Split Column by Delimiter', 'Changed Type1', and 'Changed Type2'. The 'APPLIED STEPS' pane on the right shows the step 'Renamed Columns'.

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 Ocassion. 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 several steps, including 'Changed Type'.

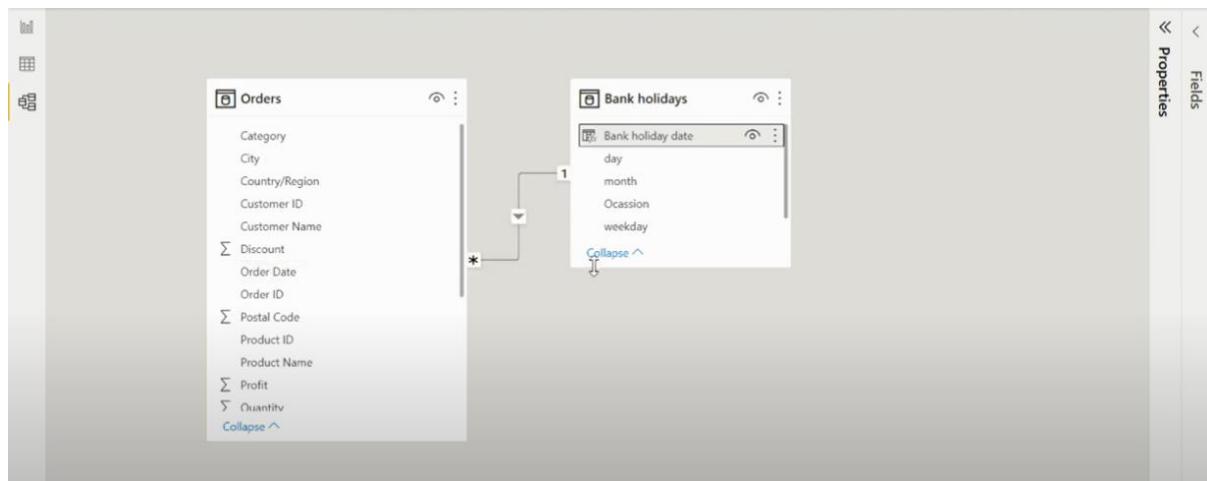
Select day column -> go to column tools -> properties section -> change summarization to don't summarize (same for month and year as well)

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' column 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. A calculated column named 'is weekend' is being created with the formula: `IF(WEEKDAY(Orders[Order Date],2)>5,TRUE(),FALSE())`. The formula uses the WEEKDAY function to check if the day of the week is greater than 5 (Wednesday), returning TRUE if it is and FALSE otherwise. The 'Bank holiday date' table is visible in the data pane below.

This screenshot shows the continuation of the calculated column creation process. The formula for 'is bank holiday' is: `IF(ISBLANK(RELATED('Bank holidays'[Bank holiday date])),FALSE(),TRUE())`. This formula checks if the related value from the 'Bank holidays' table is blank, returning FALSE if it is and TRUE otherwise. The 'Orders' table is visible in the data pane below.

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, and the main content area provides the following details:

Description: Returns the number of whole workdays between two dates (inclusive). Parameters specify which and how many days are weekend days. Weekend days and days specified as holidays are not considered as workdays.

Syntax:

```
DAX
NETWORKDAYS(<start_date>, <end_date>[, <weekend>, <holidays>])
```

Parameters:

Term	Definition
<code>start_date</code>	A date that represents the start date. The dates for which the difference is to be computed. The <code>start_date</code> can be earlier than, the same as, or later than the <code>end_date</code> .
<code>end_date</code>	A date that represents the end date. The dates for which the difference is to be computed. The <code>start_date</code> can be earlier than, the same as, or later than the <code>end_date</code> .
<code>weekend</code>	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 and their corresponding days of the week:

Weekend Number	Days of the Week
1 or omitted	Saturday, Sunday
2	Sunday, Monday
3	Monday, Tuesday
4	Tuesday, Wednesday
5	Wednesday, Thursday
6	Thursday, Friday
7	Friday, Saturday
11	Sunday only
12	Monday only
13	Tuesday only
14	Wednesday only
15	Thursday only
16	Friday only
17	Saturday only

The `holidays` parameter is also defined as a column table of one or more dates that are to be excluded from the working day calendar.

This screenshot shows the final part of the DAX documentation for the `NETWORKDAYS` function, detailing the return value:

Return Value: An integer number of whole workdays.

The screenshot shows the Power BI Data Editor interface. A new query named "netdays" is being created with the following DAX code:

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

The "Fields" pane on the right lists several columns: `business_days`, `Category`, `City`, `Country/Region`, and `Cust_name`. The `Orders` table is currently selected.

The screenshot shows the Power BI Data Editor interface. On the left, there is a code editor window containing the following DAX code:

```
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 code defines a variable `business_days` that calculates the number of business days between the selected `Order Date` and `Ship Date`, excluding public holidays.

On the right side of the interface, there is a "Fields" pane with a search bar and a tree view of fields. The "Orders" node is expanded, showing the following fields:

- business_days
- Category
- City
- Country/Region
- Cust_name
- Customer ID n...
- Customer ID.1
- Customer Name
- Discount
- End date
- Measure
- netdays
- Order Date
- Order ID

The screenshot shows the Power BI Report View interface. 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 single row is shown with the following data:

Order Date	Ship Date	business_days
05 January 2018	12 January 2018	6

On the left, there is a filter pane for "Order Date" set to "05 January 2018". On the right, there is a "Fields" pane with a search bar and a tree view of fields. The "Orders" node is expanded, showing the same list of fields as the Data Editor.

Calendar and CalendarAuto in Power BI

The screenshot shows three Power BI data modeling steps illustrating the use of CALENDAR and CALENDAR functions.

Step 1: A single column named "Calendar" is defined using the formula `CALENDAR(NOW(),2022,09,01)`. The Fields pane shows the "Orders" table with various columns like Category, City, Country/Region, etc.

Date
01-09-2022 00:00:00
02-09-2022 00:00:00
03-09-2022 00:00:00
04-09-2022 00:00:00
05-09-2022 00:00:00
06-09-2022 00:00:00
07-09-2022 00:00:00
08-09-2022 00:00:00
09-09-2022 00:00:00
10-09-2022 00:00:00
11-09-2022 00:00:00
12-09-2022 00:00:00
13-09-2022 00:00:00
14-09-2022 00:00:00
15-09-2022 00:00:00
16-09-2022 00:00:00
17-09-2022 00:00:00
18-09-2022 00:00:00
19-09-2022 00:00:00
20-09-2022 00:00:00
21-09-2022 00:00:00
22-09-2022 00:00:00
23-09-2022 00:00:00
24-09-2022 00:00:00

Step 2: The formula is updated to `CALENDAR(date(2022,09,01),date(2022,09,30))`, expanding the date range to cover the entire month of September 2022.

Date
01-09-2022 00:00:00
02-09-2022 00:00:00
03-09-2022 00:00:00
04-09-2022 00:00:00
05-09-2022 00:00:00
06-09-2022 00:00:00
07-09-2022 00:00:00
08-09-2022 00:00:00
09-09-2022 00:00:00
10-09-2022 00:00:00
11-09-2022 00:00:00
12-09-2022 00:00:00
13-09-2022 00:00:00
14-09-2022 00:00:00
15-09-2022 00:00:00
16-09-2022 00:00:00
17-09-2022 00:00:00
18-09-2022 00:00:00
19-09-2022 00:00:00
20-09-2022 00:00:00
21-09-2022 00:00:00
22-09-2022 00:00:00
23-09-2022 00:00:00
24-09-2022 00:00:00
25-09-2022 00:00:00
26-09-2022 00:00:00
27-09-2022 00:00:00
28-09-2022 00:00:00
29-09-2022 00:00:00
30-09-2022 00:00:00

Step 3: The formula is refined to `CALENDAR(TODAY()-365, TODAY()+365)`, creating a calendar spanning one year from the current date minus 365 days to plus 365 days.

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

`1 Calendar = CALENDAR(`

- `2`
- `3`
- `4)`

Date

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
17-01-2018 00:00:00
18-01-2018 00:00:00

Fields

- > `Calendar`
- > `Orders`
- Category
- City
- Country/Region
- Customer ID
- Customer Name
- Discount
- Name
- `*Orders[Order Date]`
- Order ID
- Postal Code
- Product ID
- Product Name
- Profit
- Profit ratio
- Profit ratio measure
- Profit ratio new
- Quantity
- Region

`1 Calendar = CALENDARAUTO(`

DARAUTO([FiscalYearEndMonth])

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

Date

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

Fields

- > `Calendar`
- > `Orders`
- Category
- City
- Country/Region
- Customer ID
- Customer Name
- Discount
- > `Order Date`
- Order ID
- Postal Code
- Product ID

`1 Calendar = CALENDARAUTO()`

Date

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

Fields

- > `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

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. The top navigation bar has 'Measure tools' selected. A measure named 'Measure 2' is defined with the DAX formula: `LOOKUPVALUE(Result_ColumnName, Search_ColumnName1, Search_Value, ... [Alternate_Result])`. The formula is described as 'Retrieves a value from a table.' Below the formula, there is a code editor window containing the same formula. To the right is a 'Fields' pane listing various tables and their columns, including 'Orders' and its columns like 'Customer Name', 'Customer ID', etc.

This screenshot shows the same Power BI Data Editor interface as above, but with a different formula in the code editor: `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 remains the same, showing the 'Orders' table structure.

The screenshot shows the Power BI Report view. A card visualization displays the text 'Ruben Ausman'. The report title is 'LOOKUPVALUE() IN POWER BI DAX'. The 'Fields' pane on the right shows the 'Cust_name' field selected. The 'Visualizations' pane shows the card visualization icon.

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 the title 'LOOKUPVALUE() IN POWER BI DAX'. It displays a table visual titled 'John Lee' with the column 'Cust_name'. The table contains 10 rows of location data. To the right, the 'Fields' pane shows the same list of fields from the '2018 Sales' table. A red circle highlights the 'Location ID' and 'Location_name' columns in the 'Fields' pane, indicating they are being used in the current query.

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