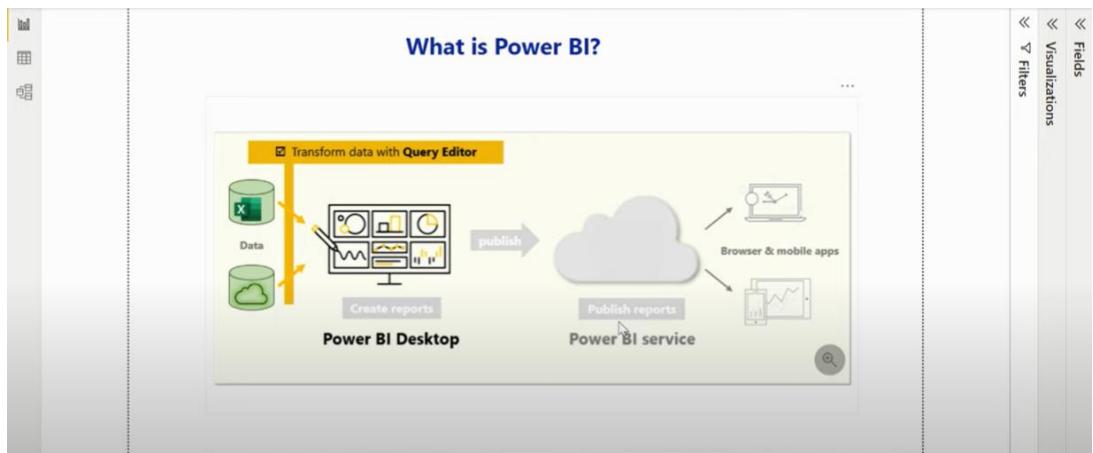
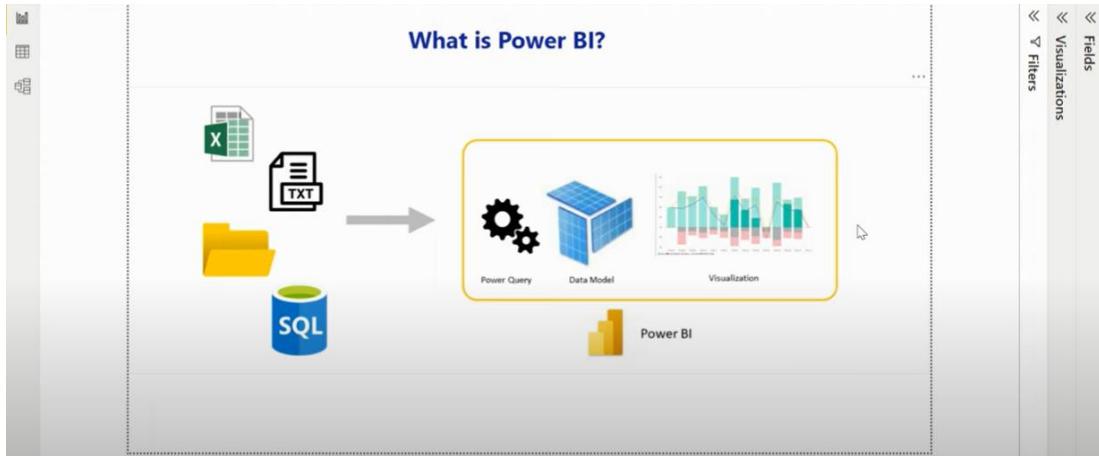
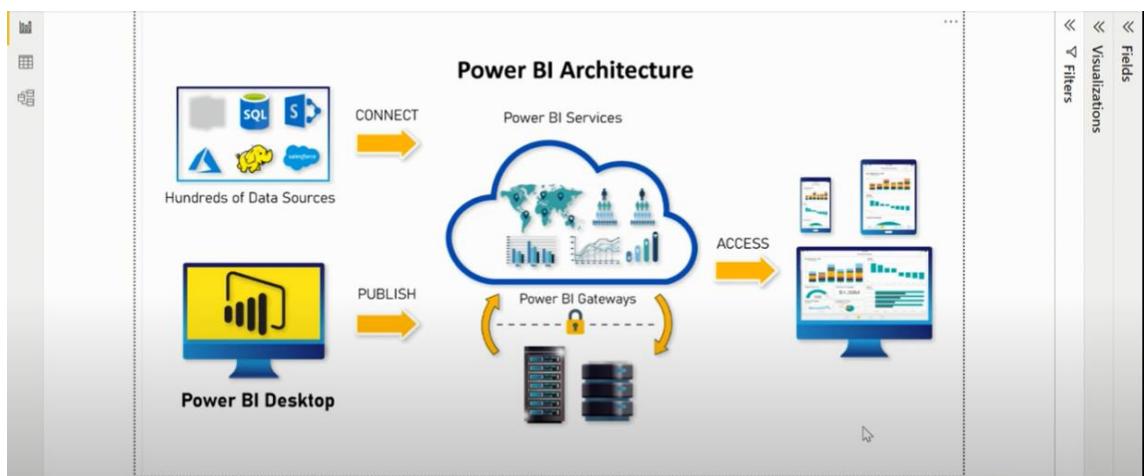


<https://docs.microsoft.com/en-us/dax/>

## What is Power BI?



## Power BI Architecture



## Choose the right data connectivity mode in Power BI

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

Direct Query option creates a connection to the data source

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

## Different parts of Power BI:

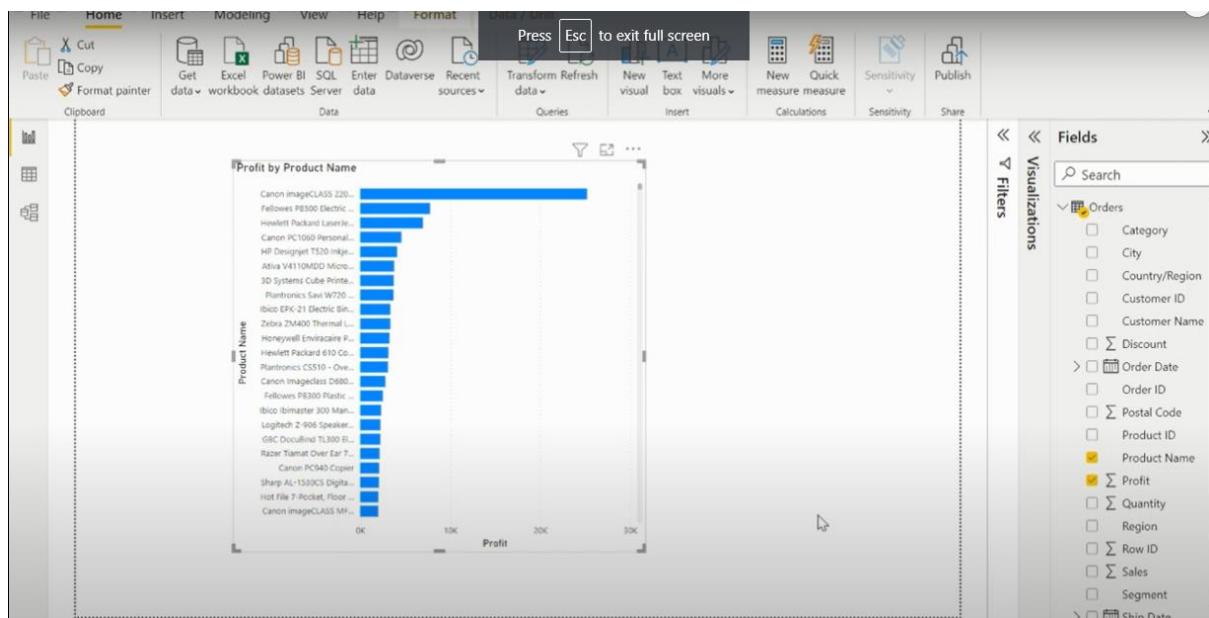
- 1) Power Query - Change Datatype, Renaming the column, deleting the column, new calculated column (M Language)
- 2) Power Pivot - To model the data, define relationship between multiple tables (DAX Language)
- 3) Power View - To build the visualization, where you can see the data visually

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. A preview window displays a table with two columns: 'Region' and 'Name'. The 'Region' column has five entries: 'South', 'Central', 'East', 'West', and 'Regional Manager'. The 'Name' column has five entries: 'Fred Suzuki', 'Roxanne Rodriguez', 'Chuck Magee', 'Sadie Pawthorne', and 'Regional Manager'. Above the preview, a formula bar shows the code: `= Table.TransformColumnTypes(People1, {"column1", type text}, {"column2", type text})`. The ribbon menu at the top includes options like 'Close & Apply', 'New Source', 'Recent Sources', 'Data', 'Data source settings', 'Parameters', 'Refresh', 'Advanced Editor', 'Properties', 'Manage Columns', 'Keep Rows', 'Remove Rows', 'Split Column', 'Group By', 'Sort', 'Data Type: Text', 'Use First Row as Headers', 'Transform', 'Merge Queries', 'Append Queries', 'Combine Files', 'Text Analytics', 'Vision', 'Azure Machine Learning', 'Combine', and 'AI Insights'. On the right side, there are 'Query Settings', 'Properties' (with 'Name' set to 'People2'), and 'Applied Steps' (listing 'Source' and 'Changed Type').

## 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 interface. The ribbon at the top has tabs for File, Home, Insert, Modeling, View, Help, Format, Data / Drill, and Publish. The 'Clipboard' section in the Home tab is circled in red, showing a list of recently copied items such as 'Word Document', 'Excel Workbook', 'Power BI Report', etc.

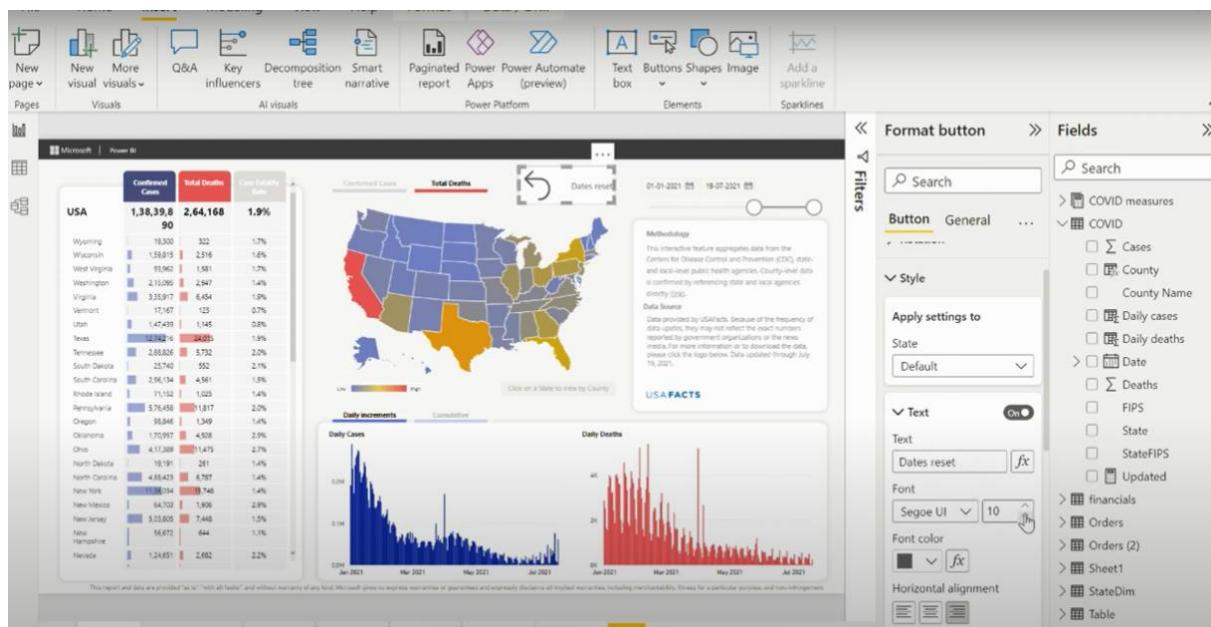
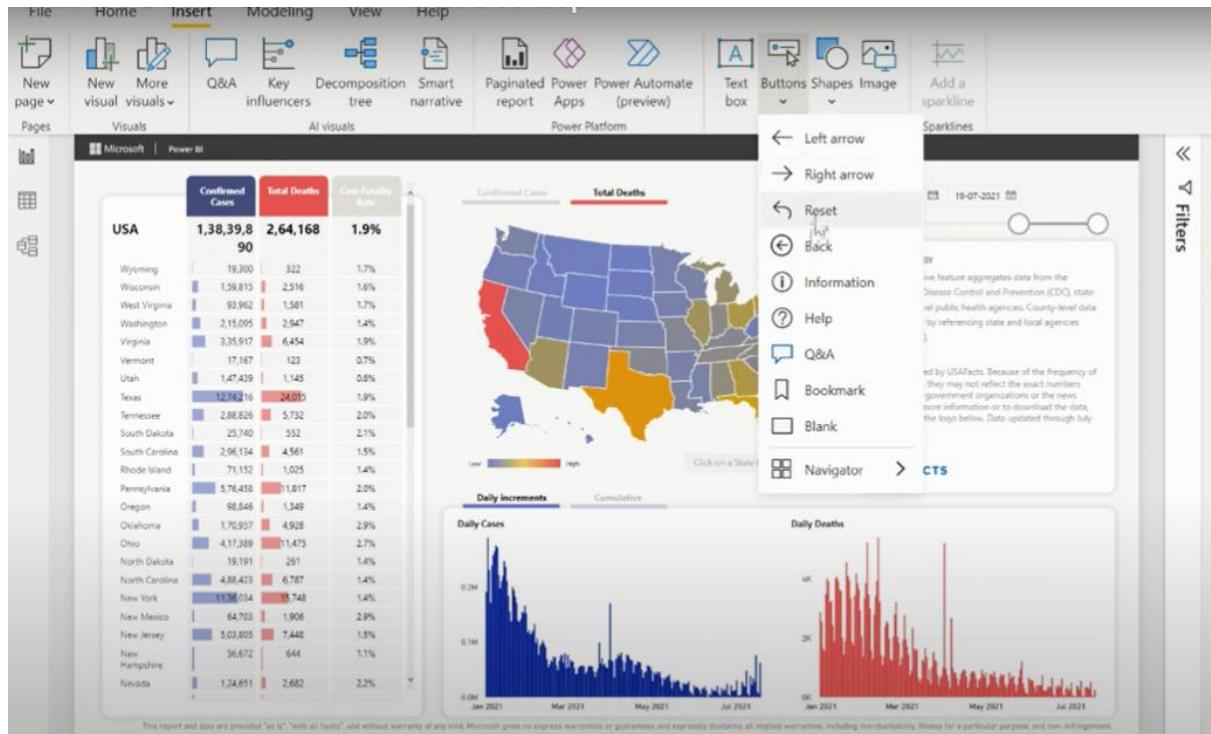
## Bookmarks in Power BI?

Bookmark reset the data in the form it is created

View -> Bookmarks -> Add -> Rename

The screenshot shows the Microsoft Power BI desktop application interface. The ribbon at the top has tabs for File, Home, Insert, Modeling, View, Help, and various icons for themes, filters, and sync. The 'Bookmarks' section in the ribbon is circled in red. The 'Visualizations' pane on the right shows a list of saved bookmarks, including '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'.

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

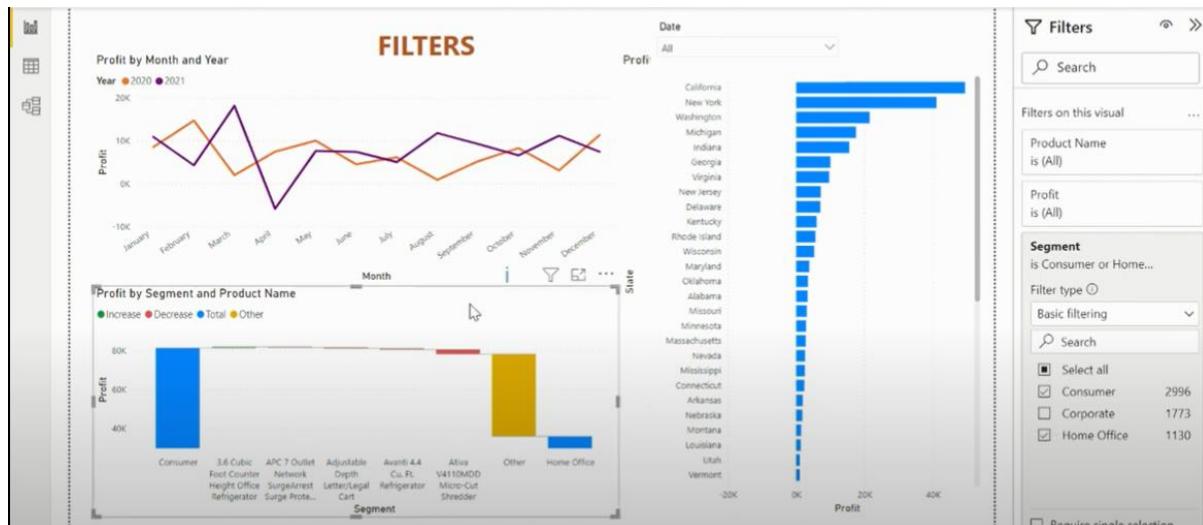


## Filters and Slicers :

Both are used to filter the data

### Filter :

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



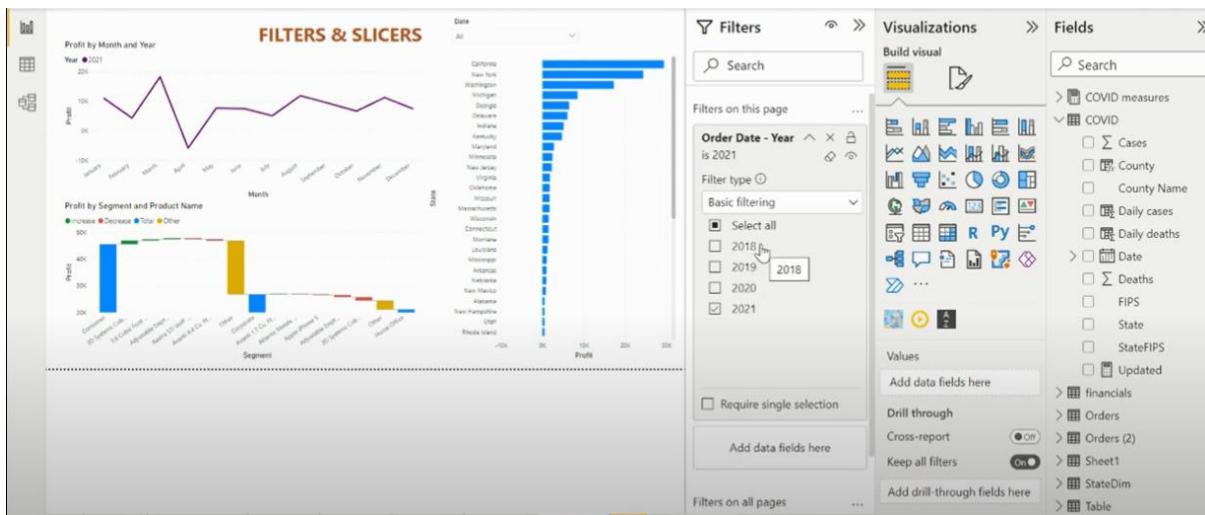
Filter on this page: will filter the current page only

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

Drag & drop the field on required filter type

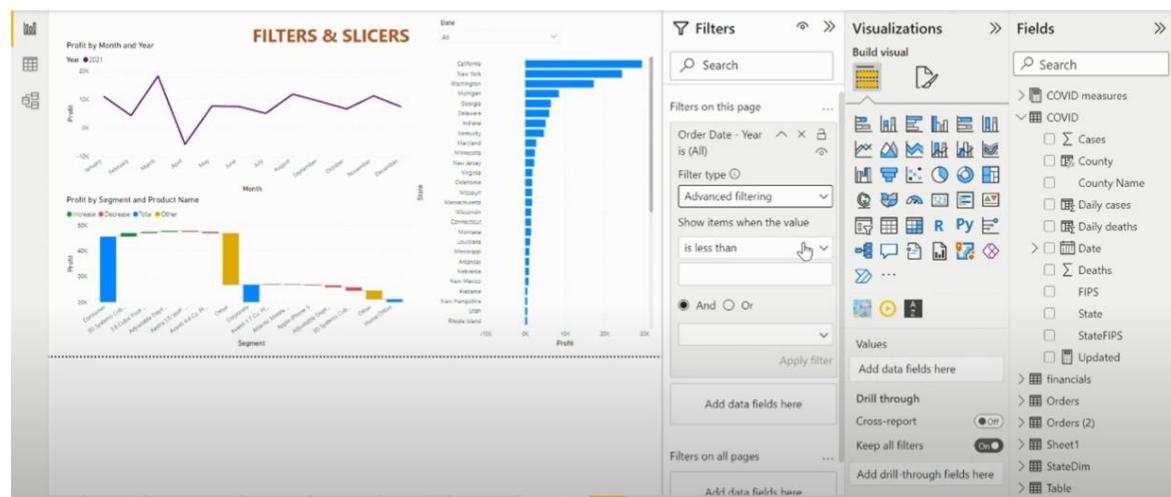


## Basic filtering:



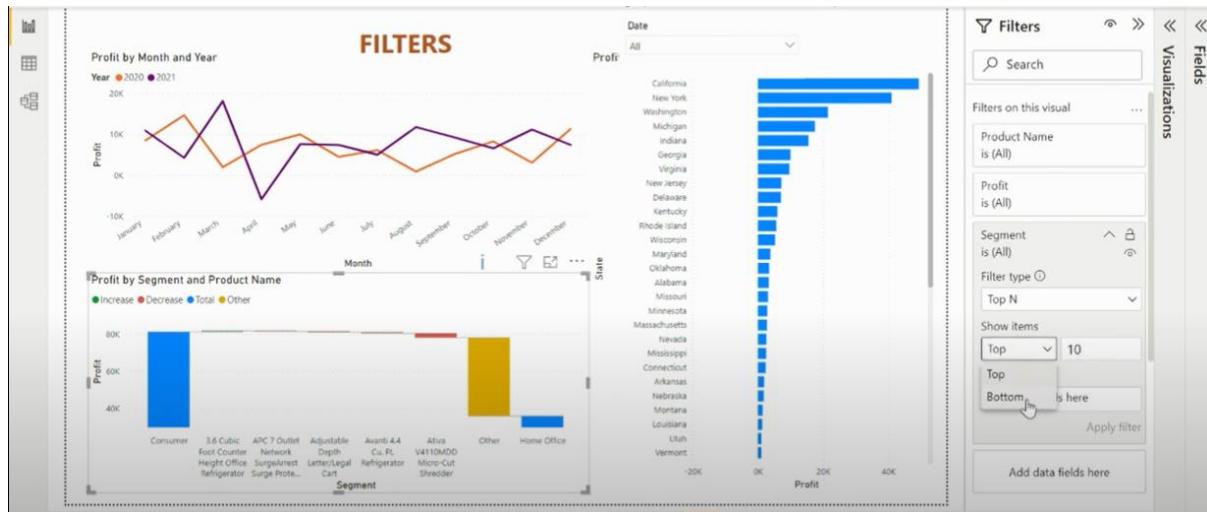
## Advanced filtering:

Filtering based on conditions (same as excel)



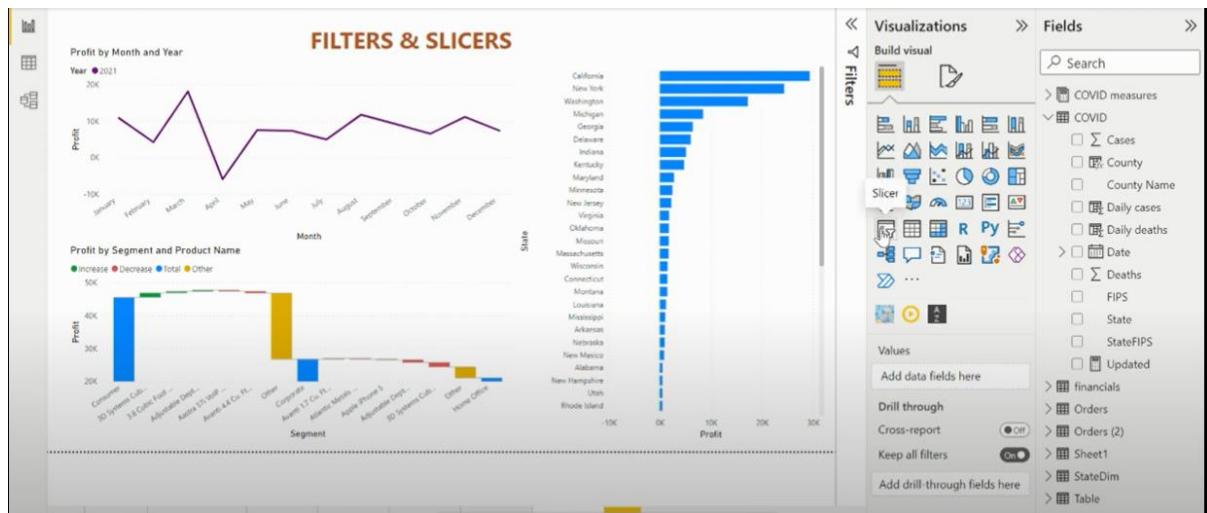
## TOP N filtering:

To filter top/bottom contributors



## Slicer:

Slicer is a Visual



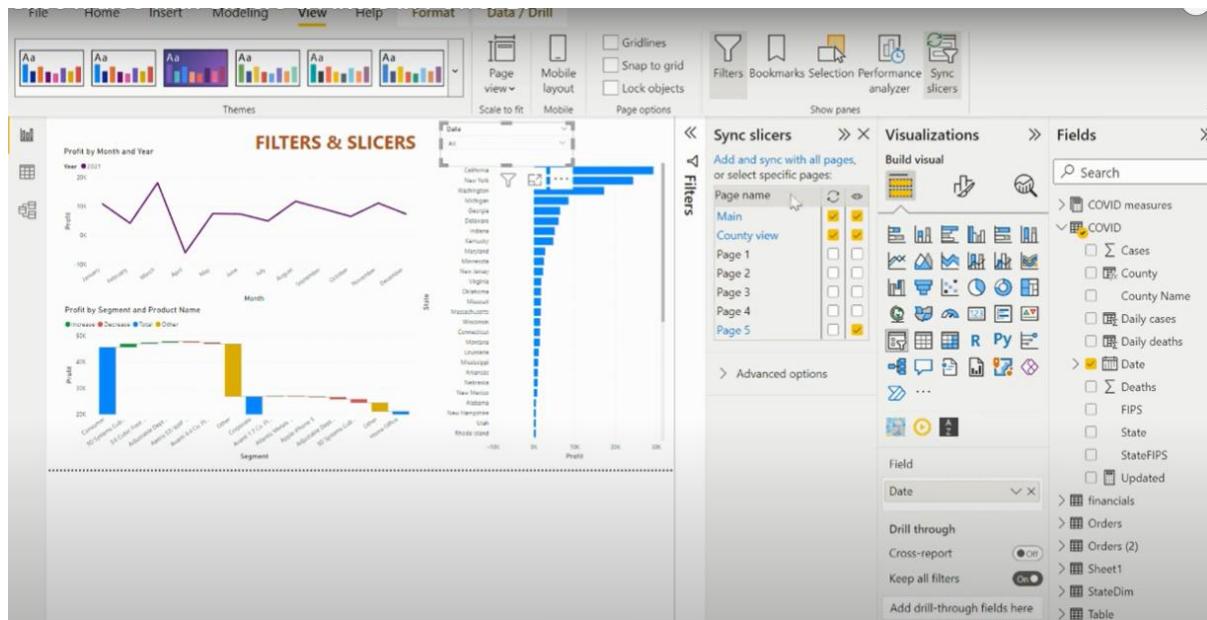
Click on slicer -> place it on report view -> add field -> right top corner (select the type)

Slicer can be applied on all the pages / selected pages

View -> sync slicers

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

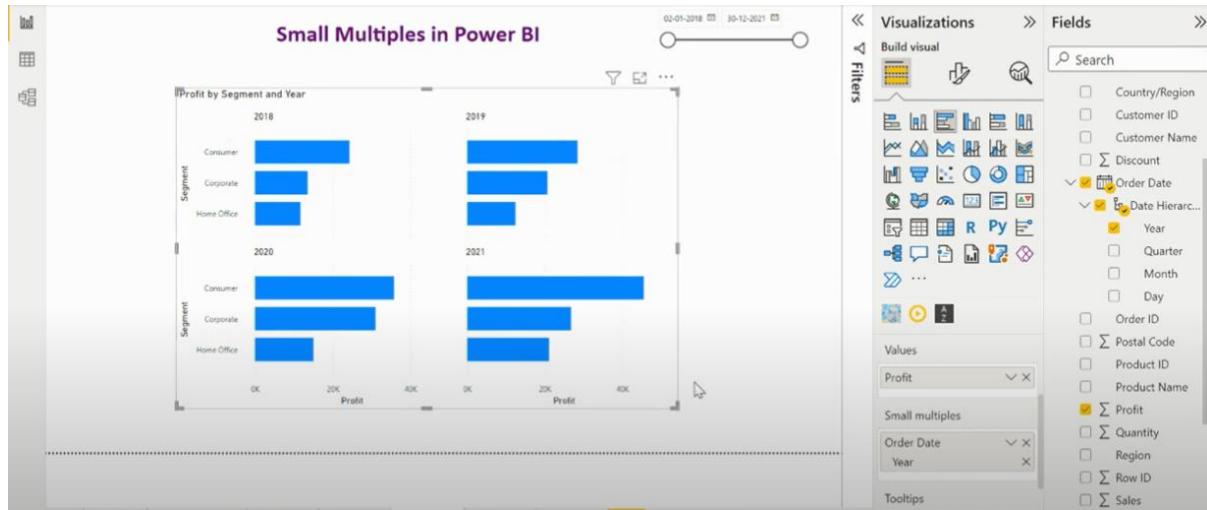
Display option : slicer will be displayed on selected pages



**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:

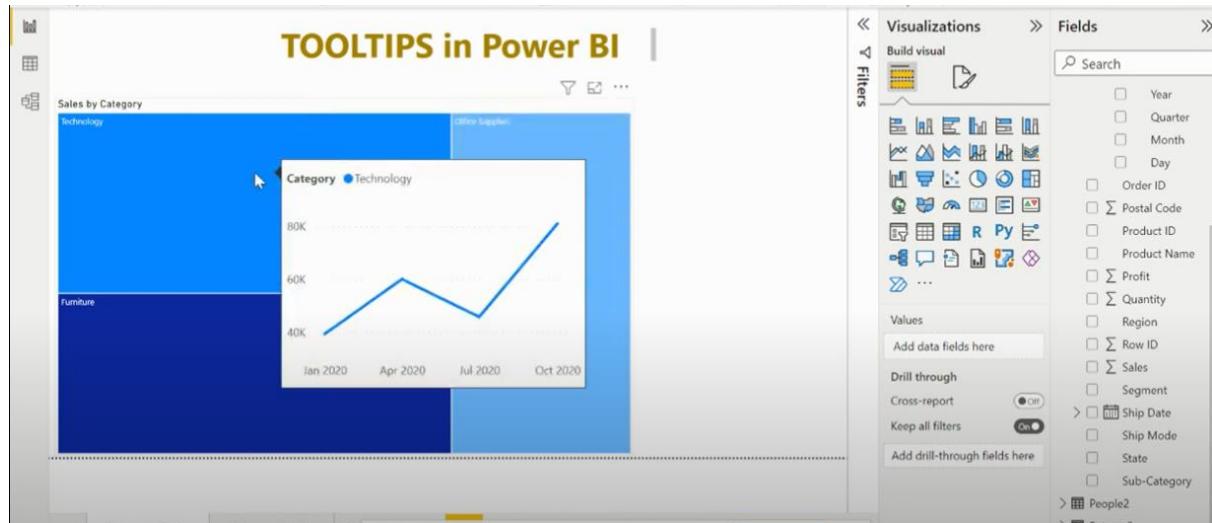
Drag & drop field in small multiples



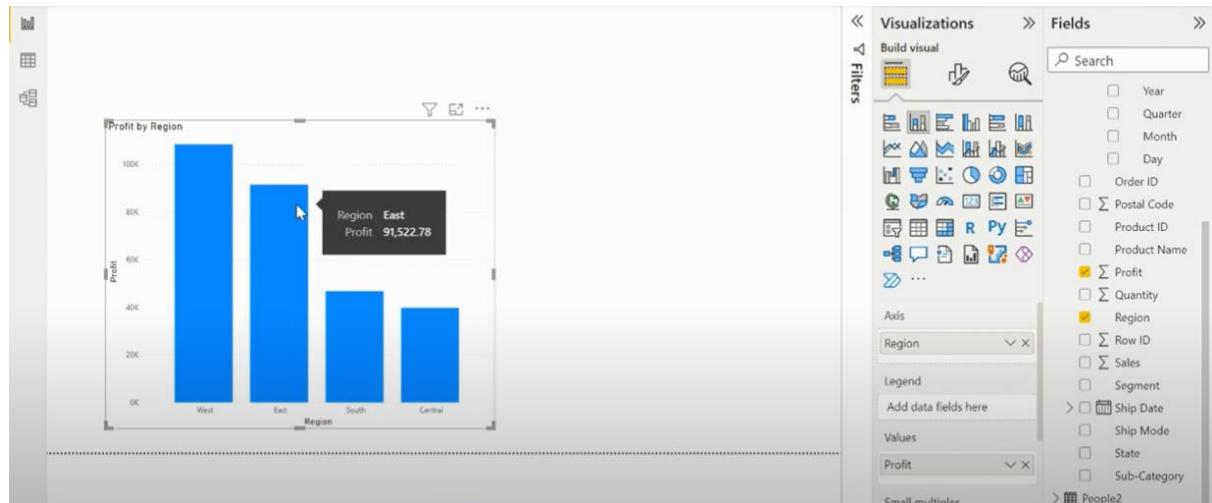
## Customized Tooltips in Power BI:

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

- 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

**Favorites**

**Page information**

Name: Region tooltip

Allow use as tooltip: On

Allow Q&A: Off

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

**Favorites**

**Canvas settings**

Type: Tooltip

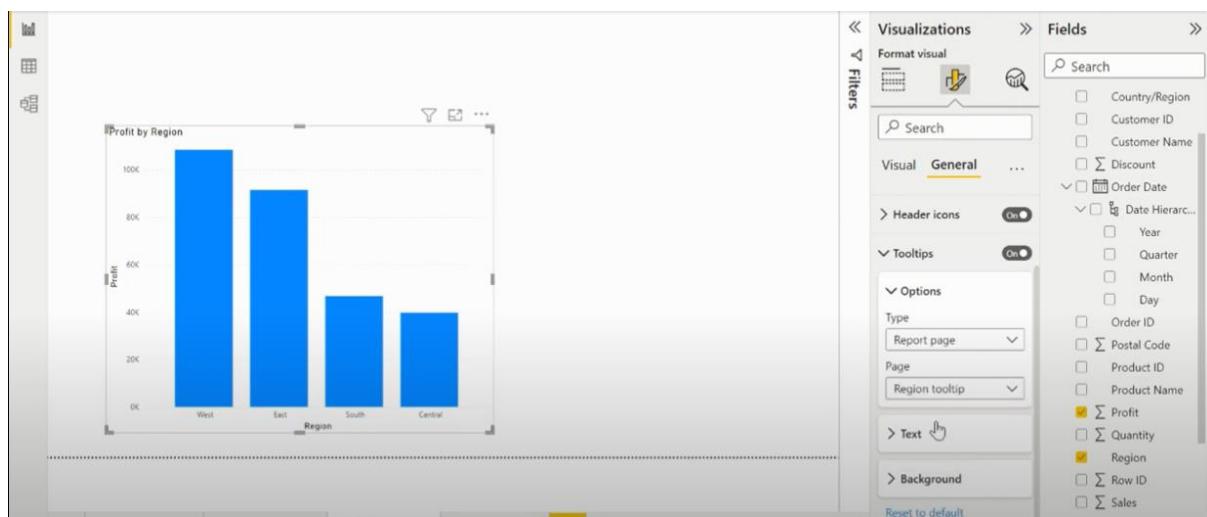
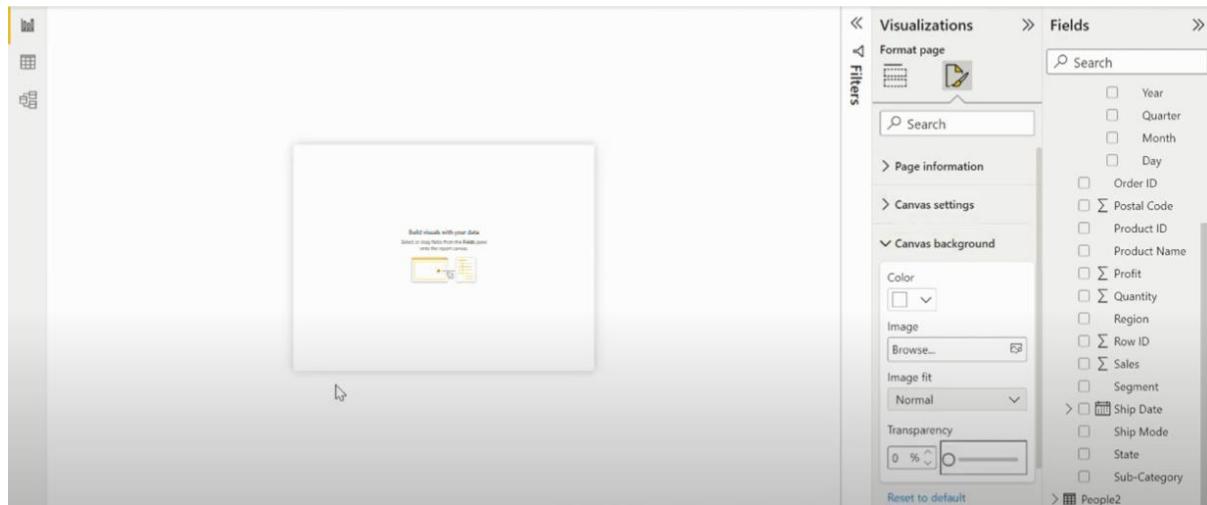
Height: 240 px

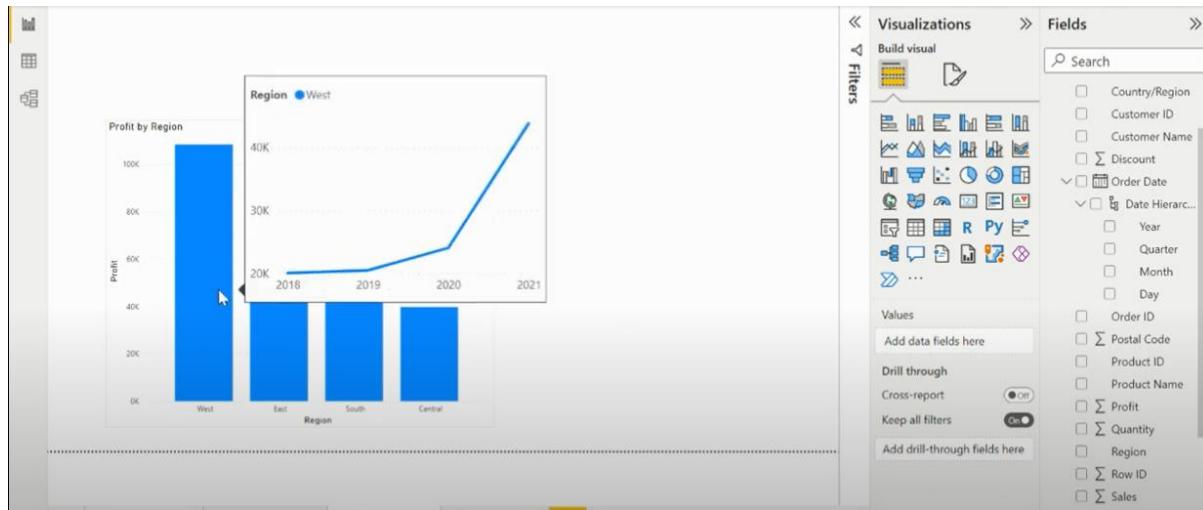
Width: 200 px

Vertical alignment: Middle

Reset to default

> Canvas background





## Conditional Formatting

### Conditional Formatting – Matrix

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

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

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

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

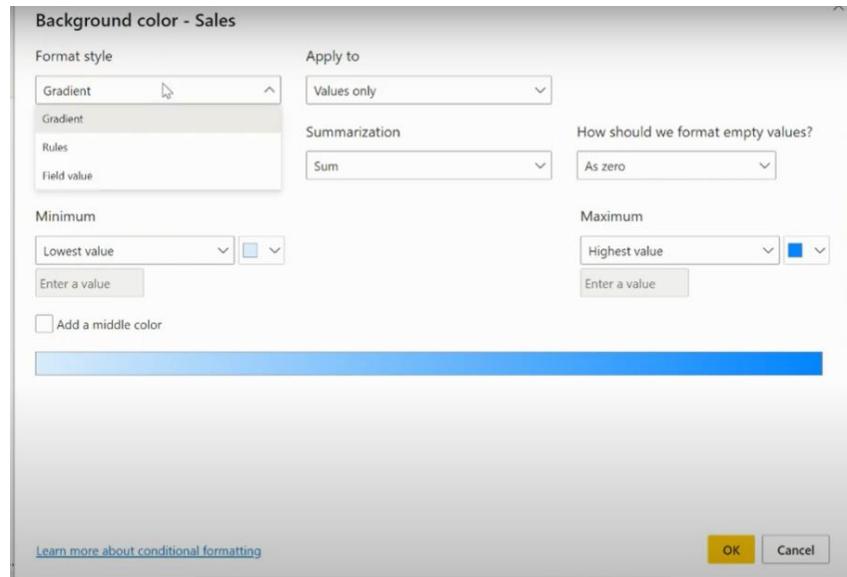
Data Bars –

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

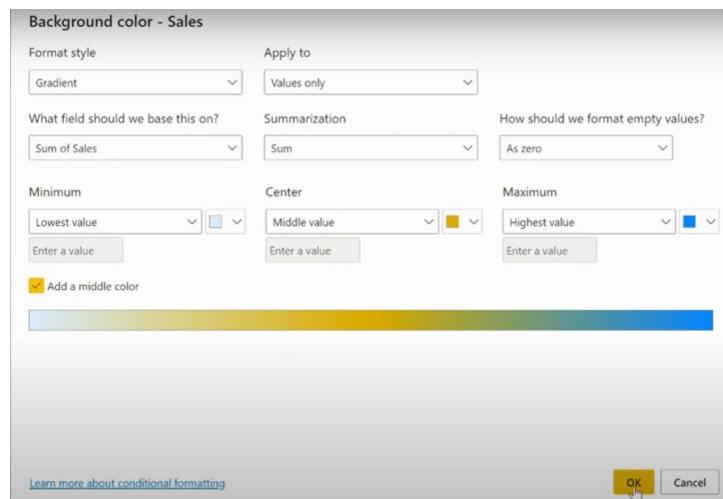
Icons will be displayed alongside the cell values

## Format Style – Gradient

Cells will be distinguished by shades of two chosen colours

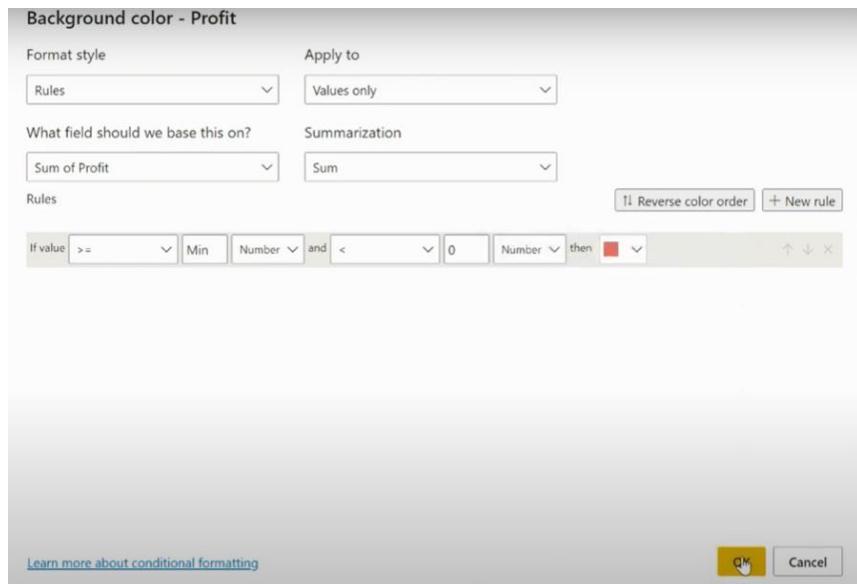
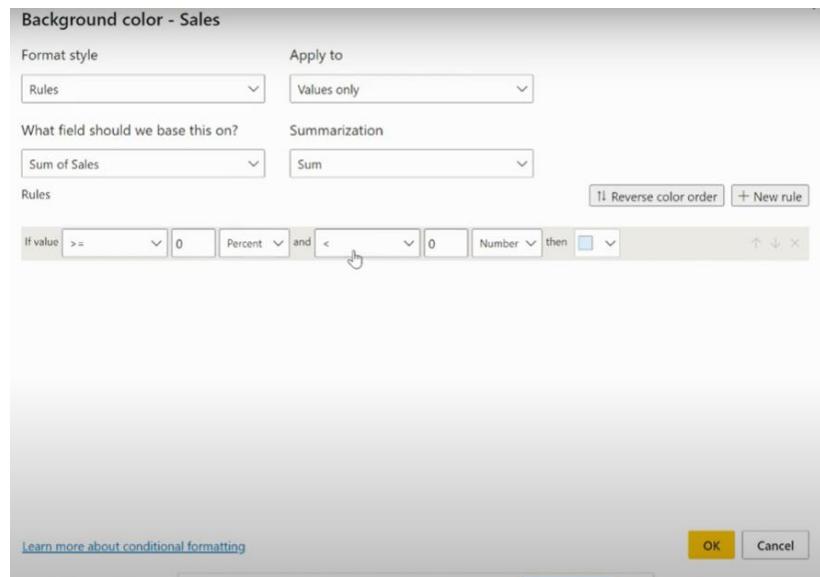


Cells will be distinguished by shades of three chosen colours by selecting the option to add middle colour



A screenshot of a Power BI report interface. On the left, there is a navigation pane with icons for Home, Reports, and Data. The main area shows a matrix visual titled 'Conditional Formatting - Matrix'. The matrix has columns for 'Year', 'Sales', 'Profit', and 'Average of Discount'. The data is grouped by year (2021, 2020, 2018, 2019) and further broken down by quarter (Qtr 1, Qtr 2, Qtr 3, Qtr 4). The cells in the matrix are colored according to a three-color gradient based on their value. The report also includes a 'Visualizations' pane on the right containing various chart and table icons, and a 'Values' pane below it listing 'Sales', 'Profit', and 'Average of Discount'.

## Format Style – Rules



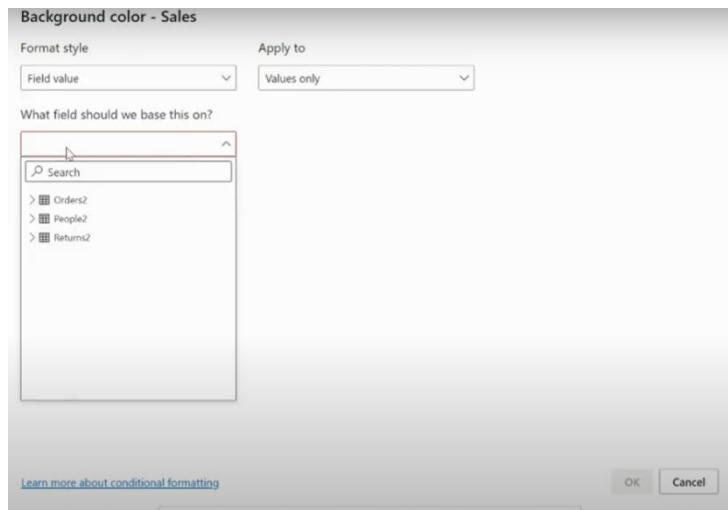
**Conditional Formatting - Matrix**

Visualizations: Build visual, Filters, Fields

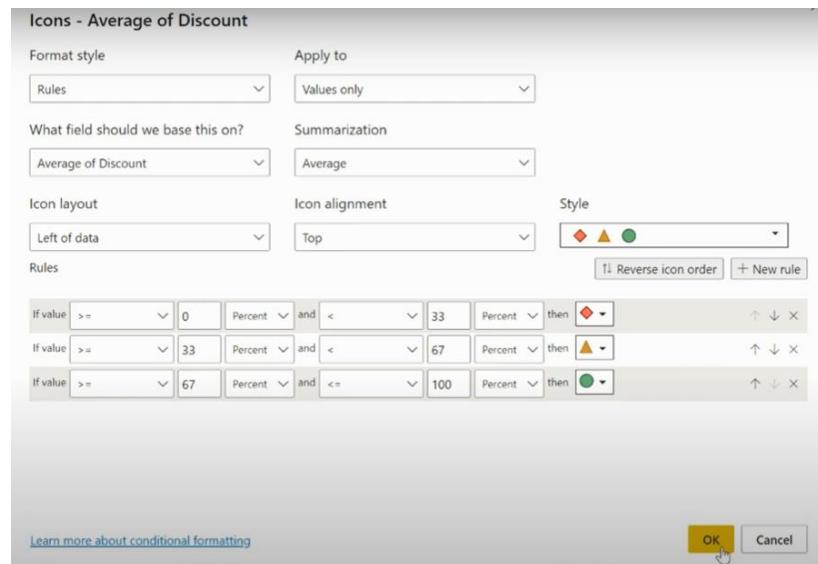
Filters: Add data fields here, Values (Sales, Profit, Average of Discount), Drill through (Cross-report Off, Keep all filters On)

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

The screenshot shows two 'Data bars - Sales' dialog boxes side-by-side, both with the 'Show bar only' checkbox checked. The left dialog has 'Positive bar' set to blue and 'Negative bar' set to light blue. The right dialog has 'Positive bar' set to blue and 'Negative bar' set to light blue. Below the dialogs are two 'Conditional Formatting - Table' visualizations. The top one shows a table with data rows colored by profit (yellow for positive, grey for negative). The bottom one shows the same table with data rows colored by sales (blue for positive, grey for negative). Both visualizations have a 'Fields' pane on the right containing 'Order Date', 'Profit', 'Sales', 'Quantity', and 'Discount'.

**Data bars - Sales**

Format cells with bars based on their values.

Show bar only

Minimum Maximum

Lowest value Highest value

Enter a value Enter a value

Positive bar Bar direction

Left to right

Negative bar Axis

OK Cancel

**Data bars - Sales**

Format cells with bars based on their values.

Show bar only

Minimum Maximum

Lowest value Highest value

Enter a value Enter a value

Positive bar Bar direction

Left to right

Negative bar Axis

OK Cancel

**Conditional Formatting - Table**

Order Date	Profit	Sales	Quantity	Discount
22 December 2021	957.12	-4,473.04	100	4.40
24 December 2021	1,560.34	-1,113.5	54	1.70
29 December 2021	644.43	-915.53	41	0.40
25 December 2021	359.20	-698.93	87	5.60
23 December 2021	284.10	-926.78	63	2.60
28 December 2021	253.12	-1,657.35	64	4.12
26 December 2021	61.12	-814.59	12	0.90
30 December 2021	101.54	-713.79	23	1.00
27 December 2021	-31.97	-177.64	6	0.60
Total	4,196.90	24,579.68	450	21.32

**Conditional Formatting - Table**

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

Visualizations: Build visual, Filters, Fields

Values: Order Date, Profit, Sales, Quantity, Discount

Drill through, Cross-report

Visualizations: Build visual, Filters, Fields

Values: Order Date, Profit, Sales, Quantity, Discount

Drill through, Cross-report, Keep all filters

Vimeo Record - Screen & Webcam Recorder is sharing your screen. Stop sharing Hide CF-Matrix CF-Color Measure

## Different Types of Conditional Formatting

Profit – Based on Background Colour

Sales – Based on Data bars

Quantity – Based on icons

Avg. Discount – Based on font colour

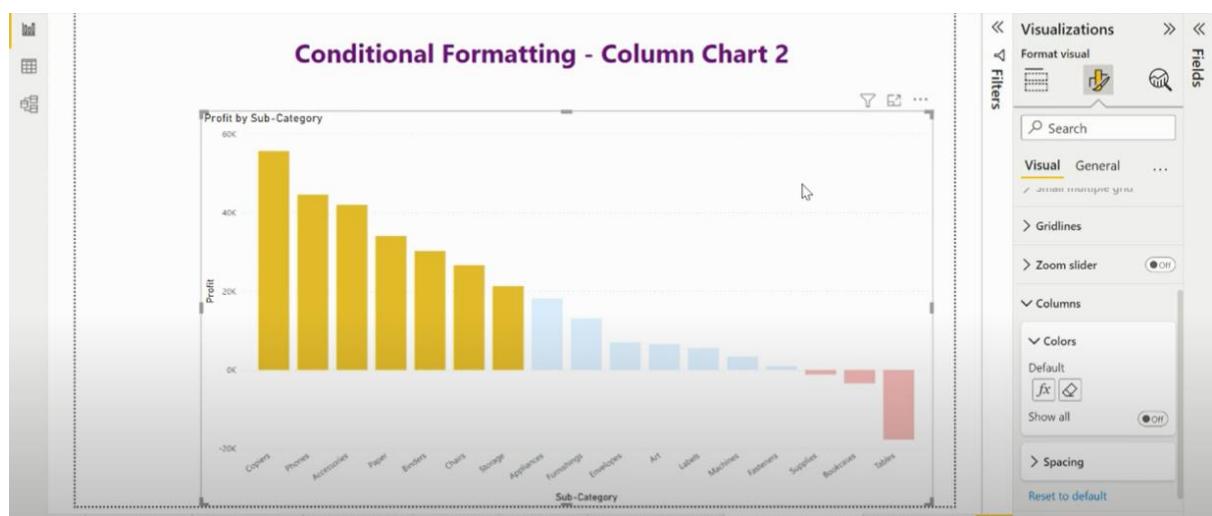
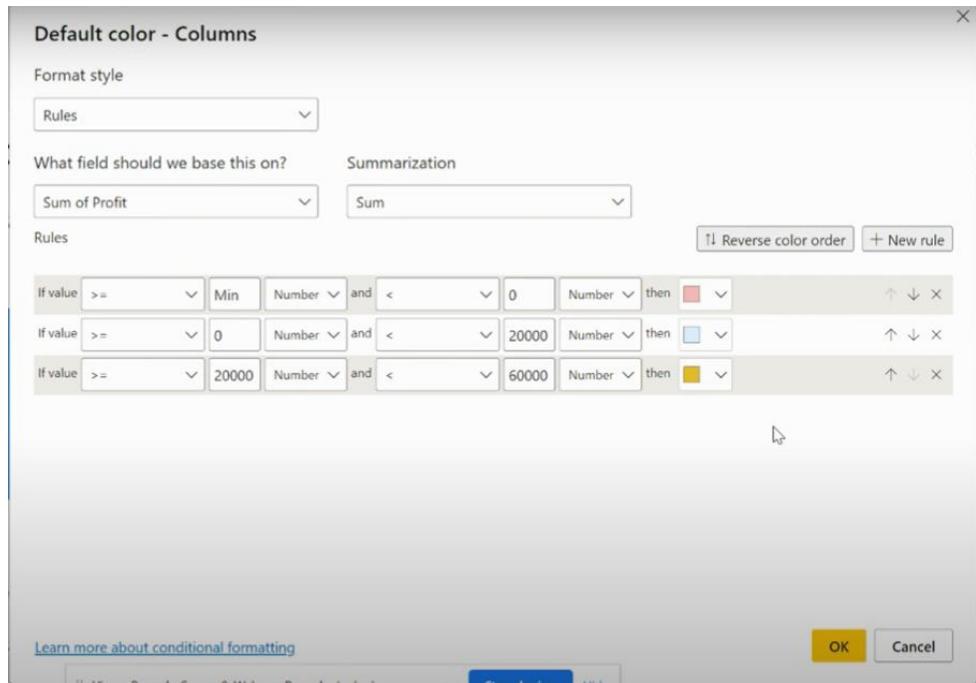
The screenshot shows a Power BI interface with a table visualization titled "Conditional Formatting - Table". The table has columns: Order Date, Profit, Sales, Quantity, and Average of Discount. The rows show data from December 2021, with the last row being the Total. The background color of the table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

Order Date	Profit	Sales	Quantity	Average of Discount
22 December 2021	957.12	7,442.00	100	0.16
24 December 2021	568.24	6,233.15	54	0.11
29 December 2021	644.43	9,155.33	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 Power BI ribbon is visible at the top, and the Fields pane on the right shows the selected "CF-Color Column" field.



### 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:

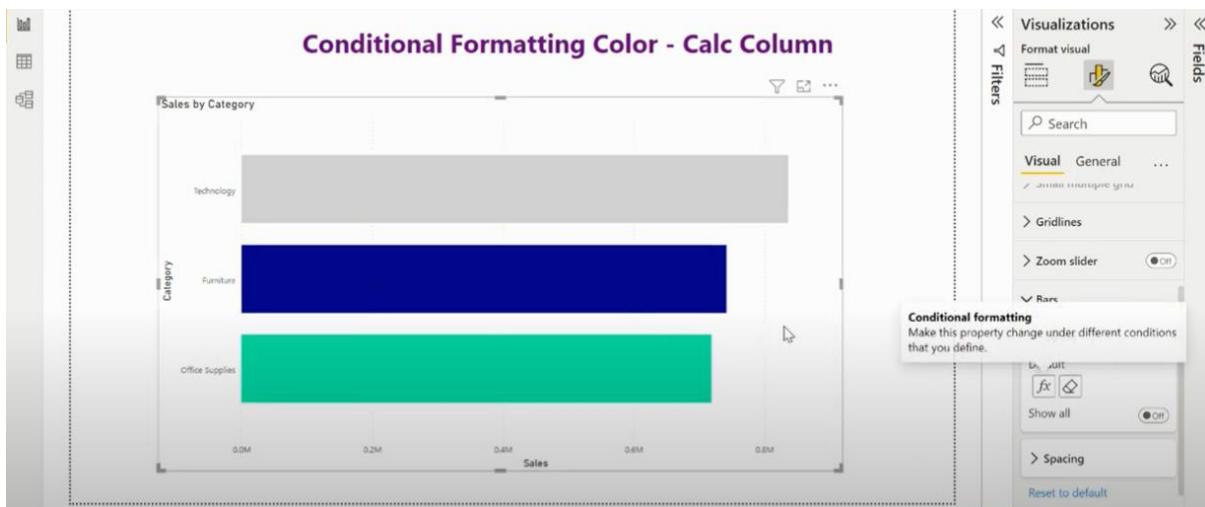
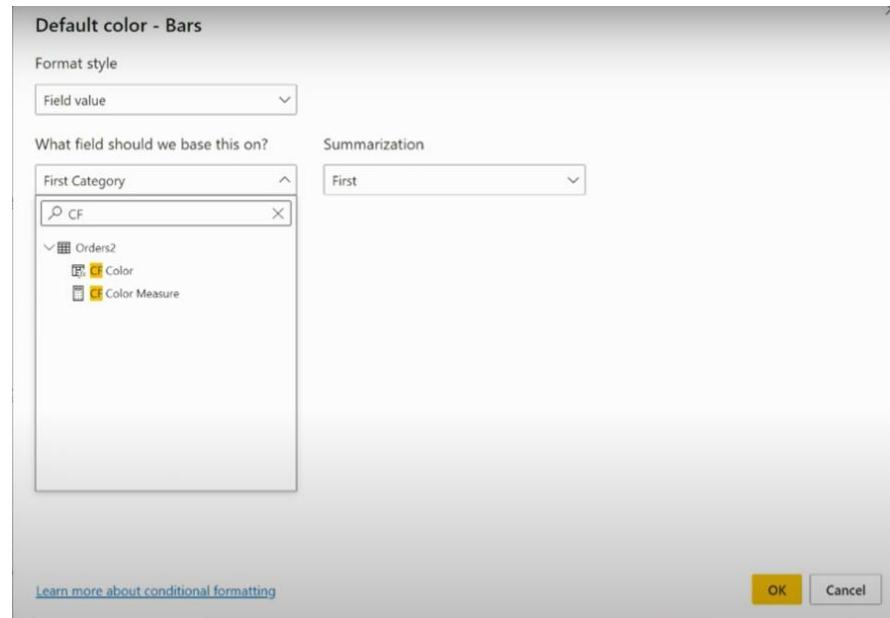
```

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 measure using DAX expression (switch is dax expression) (maxx aggregation function is used on top of switch bcz Measures are aggregated values so we need to aggregation function)

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

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

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

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

Below the formula bar is a table with columns: Sales Code, Region, Product ID, Category, Sub-Category, Product Name, Sales, Quantity, Discount, Profit, and CF Color. The table contains 20 rows of data. To the right of the table is a Fields pane with various columns listed, including City, Country/Region, Customer ID, Customer Name, Discount, Order Date, Order ID, Postal Code, Product ID, Product Name, Profit, Quantity, Region, Row ID, and Sales.

## 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( $\Sigma$ ) 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  $\sum \text{Row ID}$ .

**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

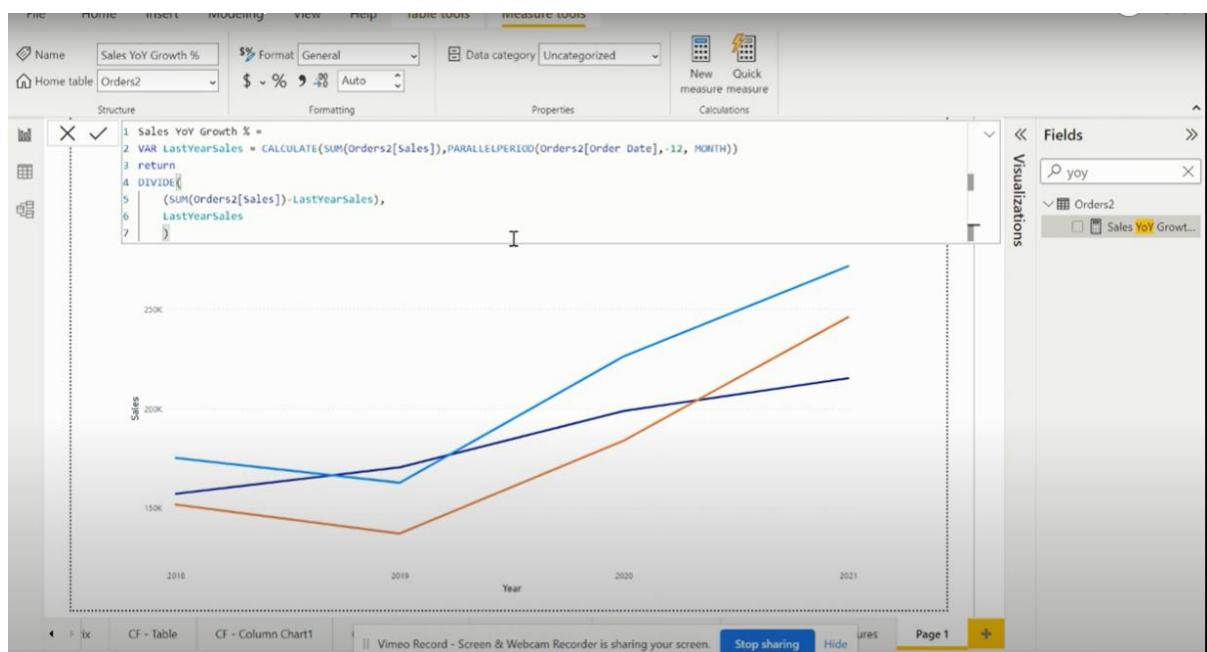
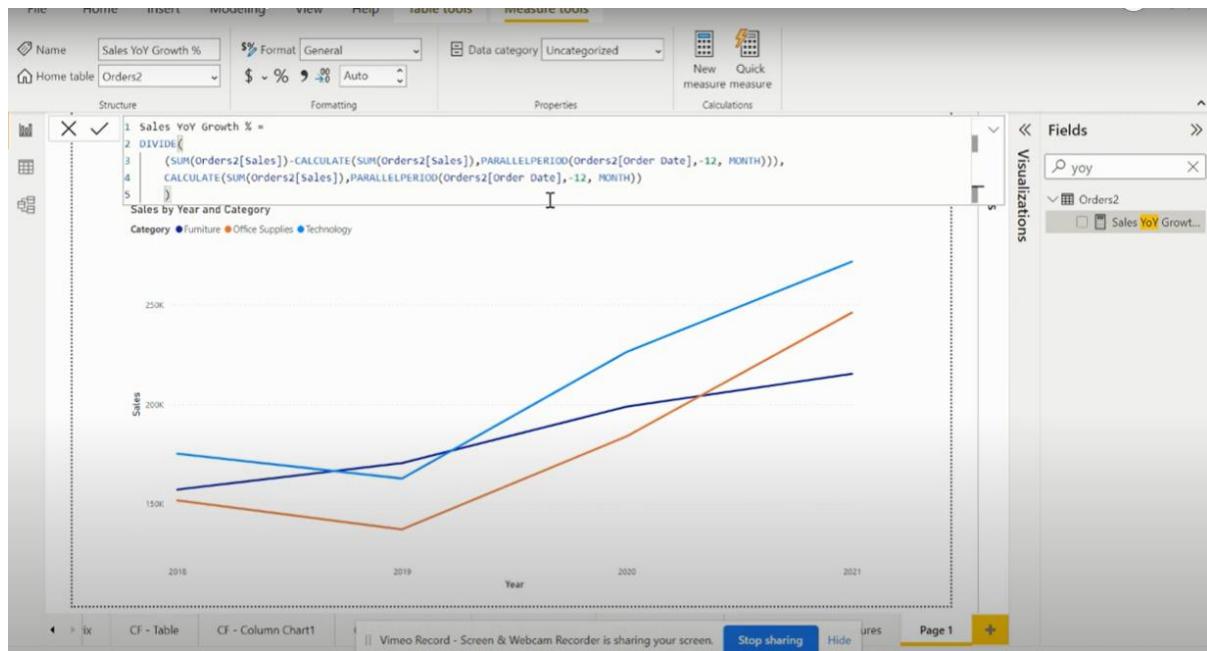
Makes it simpler to debug in case of error

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 the full expression: `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 includes the measure: `State ranks = RANKX(Orders2, [Total Sales],,DESC)`. The table view displays the same data as before, but the 'Sales' column now includes the calculated rank for each state. The 'Fields' pane on the right shows the 'State ranks' field selected under the 'Values' section. Other measures like 'Total Sales' and 'Sales' are also listed.

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

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

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

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

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

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

Inserting 2 ALL functions inside RankX is not an option.

The screenshot shows the Power BI Model view. In the 'Measure tools' ribbon, a measure named 'State ranks' is defined with the formula: `RANKX(ALL(Orders2[State]),ALL(Orders2[City]),[Total Sales])`. The visual area displays an error message: 'Can't display the visual. See details'. The Fields pane on the right lists various dimensions and measures, including 'Orders2', 'Category', 'CF Color', 'CF Color Meas...', 'City', 'Country/Region', 'Customer ID', 'Customer Name', 'Discount', 'Order Date', 'Date Hierarc...', 'Year', 'Quarter', 'Month', 'Day', 'Order ID', 'Postal Code', 'Product ID', and 'Product Name'.

2 ALL functions needs to be defined inside crossjoin function.

This screenshot shows the same Power BI Model view as the previous one, but the formula has been modified to use 'crossjoin' instead of 'ALL': `RANKX(crossjoin(ALL(Orders2[State]),ALL(Orders2[City])),[Total Sales])`. The error message 'Can't display the visual. See details' remains. The Fields pane is identical to the first screenshot.

The screenshot shows the Power BI Data view. A table titled 'RANKX Function in DAX' is displayed with the following data:

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

The visual area shows the data table. The Fields pane on the right is identical to the previous screenshots.

## 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 ribbon bar with icons for Name, Structure, Column, and Row. The 'Name' section is set to 'Table'. The 'Structure' section contains a formula bar: '1 Top5States = TOPN(5, Orders2, [Total Sales], DESC)'. Below the formula bar is a dropdown menu with 'DESC' selected. To the right of the formula bar is a 'Fields' pane with a search bar and a list of available fields: 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. The main workspace shows a table with columns: Row ID, Order ID, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name, Segment, Country/Region, City, State, and Postal Code. The table contains five rows of data. The bottom part of the screenshot shows the same table structure with the same data, indicating the creation of a new table named 'Top5States'.

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
<b>Total</b>	<b>22,97,200.86</b>

The Fields pane on the right shows the "Orders2" table with various columns like Category, CF Color, City, etc. The "Filters" section is expanded, showing the "Category" filter is selected.

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 is selected, showing "Furniture" and "Office Supplies". 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
<b>Total</b>	<b>7,41,999.80</b>

The Fields pane on the right shows the "Orders2" table with various columns like Category, CF Color, City, etc. The "Filters" section is expanded, showing the "Category" filter is selected.

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
<b>Total</b>	<b>22,97,200.86</b>

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 the same Region Sales data as before. Additionally, a new column 'Sales (ALL)' is added to the table, showing the total value of 22,97,200.86 for each region. The filters and fields pane are identical to the previous screenshot.

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
<b>Total</b>	<b>22,97,200.86</b>	<b>22,97,200.86</b>

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 defined 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
<b>Total</b>	<b>22,97,200.86</b>	<b>22,97,200.86</b>

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.

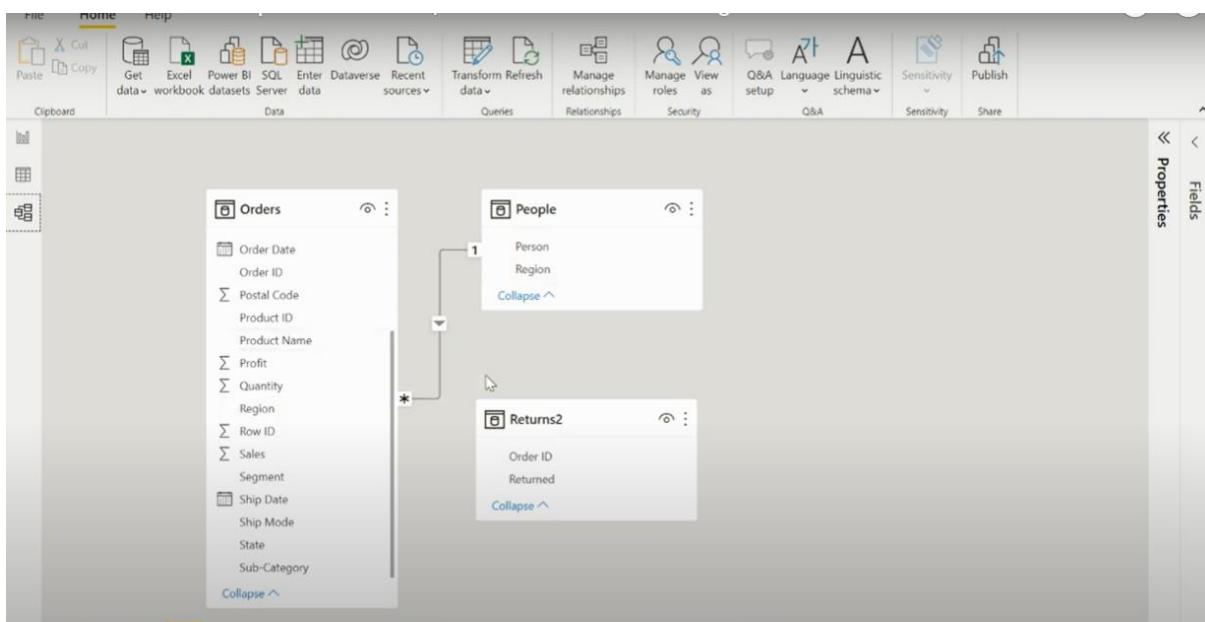
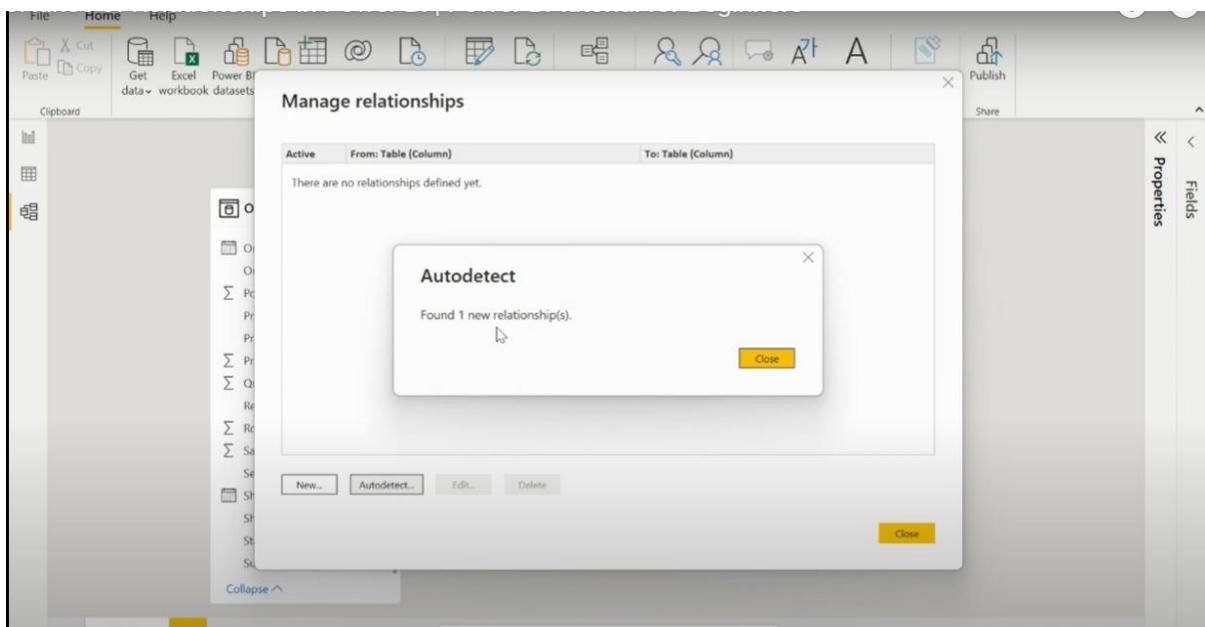
## Relationships in Power BI

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?

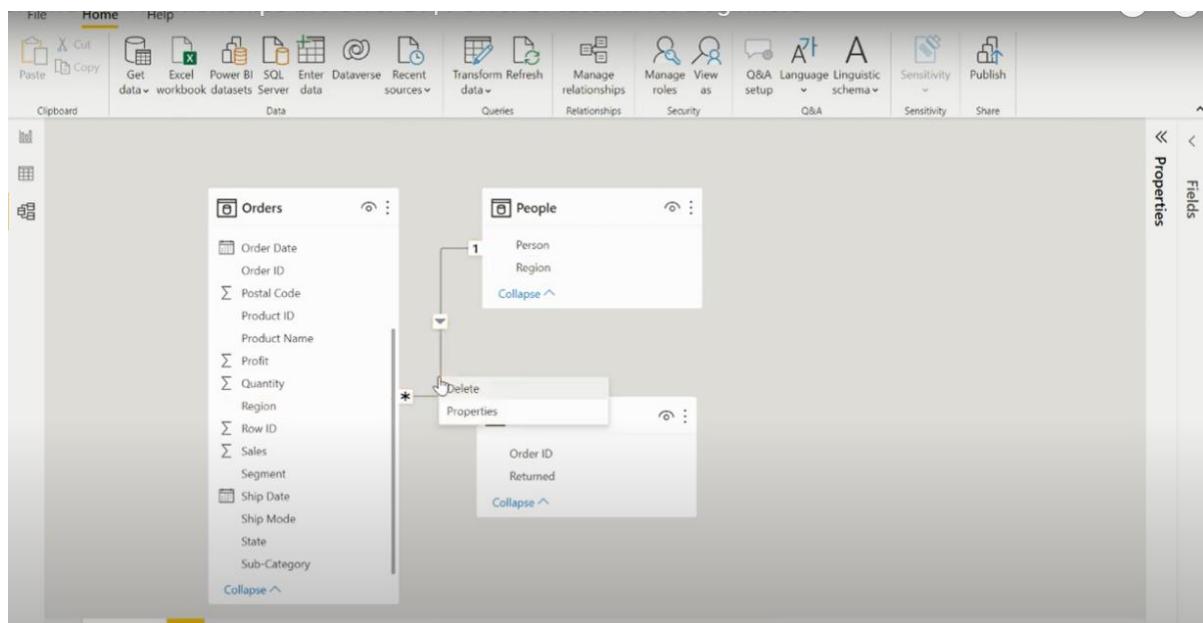
The screenshot shows the Power BI desktop application with the ribbon menu open. The "Home" tab is selected. In the center, three tables are displayed: "Orders", "People", and "Returns2". The "Manage relationships" button in the ribbon is highlighted. The "Manage relationships" ribbon tab is selected, showing sub-options like "Add, edit, or remove relationships between tables.", "Transform Refresh data", "Manage roles", "View as", "Q&A", "Language setup", "Sensitivity", and "Share".



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



### Edit relationship

Select tables and columns that are related.

Orders								
Customer ID	Customer Name	Segment	Country/Region	City	State	Postal Code	Region	Of
RA-19885	Ruben Ausman	Corporate	United States	Los Angeles	California	90049	West	Of
BN-11515	Bradley Nguyen	Consumer	United States	Los Angeles	California	90049	West	Of
BN-11515	Bradley Nguyen	Consumer	United States	Los Angeles	California	90049	West	Of

People	
Person	Region
Anna Andreadi	West
Chuck Magee	East
Kelly Williams	Central

**Cardinality:** Many to one (\*:1)      **Cross filter direction:** Single

Make this relationship active       Apply security filter in both directions

Assume referential integrity

**OK**      **Cancel**

## 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

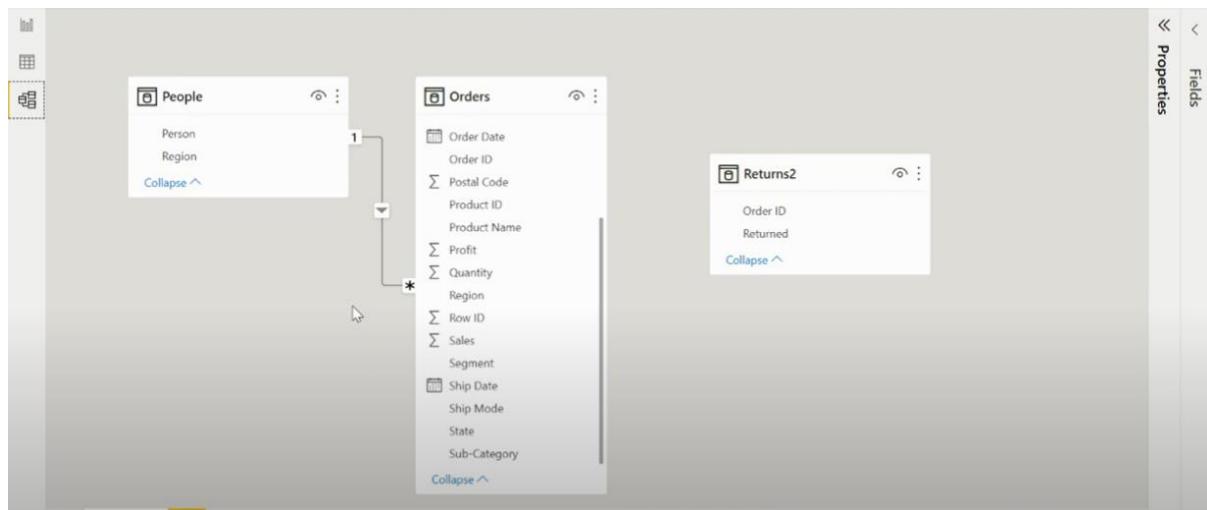
**Cardinality and One-to-many relationship in Power BI**

· 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' (PK) and 'DoctorName'. The 'Patient' table has fields 'PatientID' (PK), 'PatientName', and 'Doctor\_DoctorID' (FK).  
At the bottom, there is a visual representation of a table with two columns: 'Person' and 'Sales'. The 'Person' column lists names: Anna Andreadi, Cassandra Brandow, Chuck Magee, and Kelly Williams. The 'Sales' column lists their respective sales amounts: 7,25,457.82, 3,91,721.91, 6,78,781.24, and 5,01,239.89. A 'Total' row shows a sum of 22,97,430.86.

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

## How to overcome this?

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

**Append**

Concatenate rows from two tables into a single table.

Two tables  Three or more tables

First table: Orders

Second table: Returns2

OK Cancel

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

**Append1**

Combine rows from two tables into a single table.

Orders, Returns2

OK Cancel

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

**Append1**

Select columns from the source table.

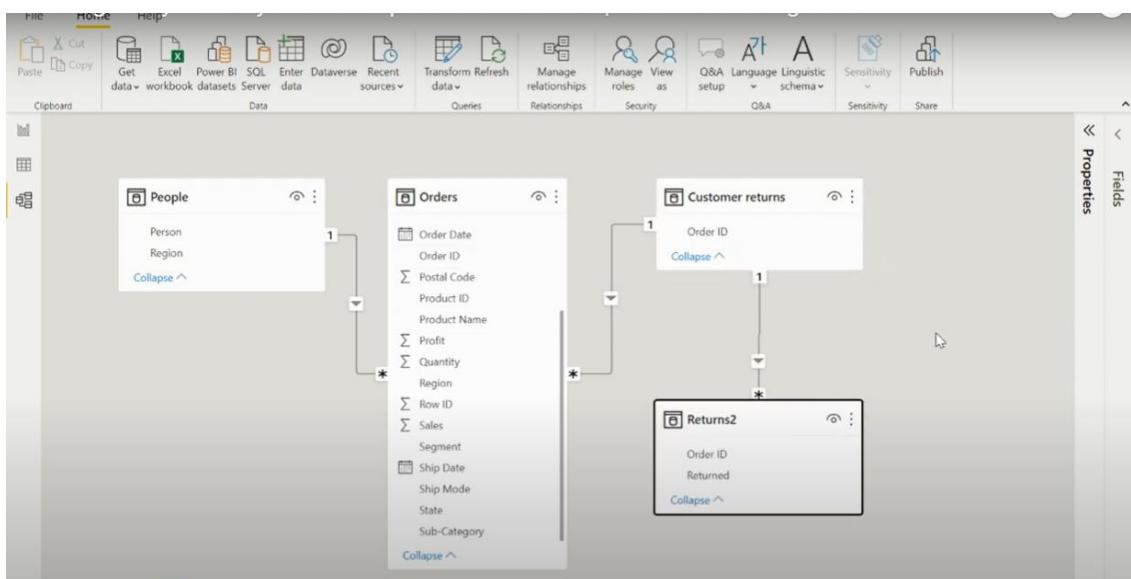
Order ID

OK Cancel

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

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

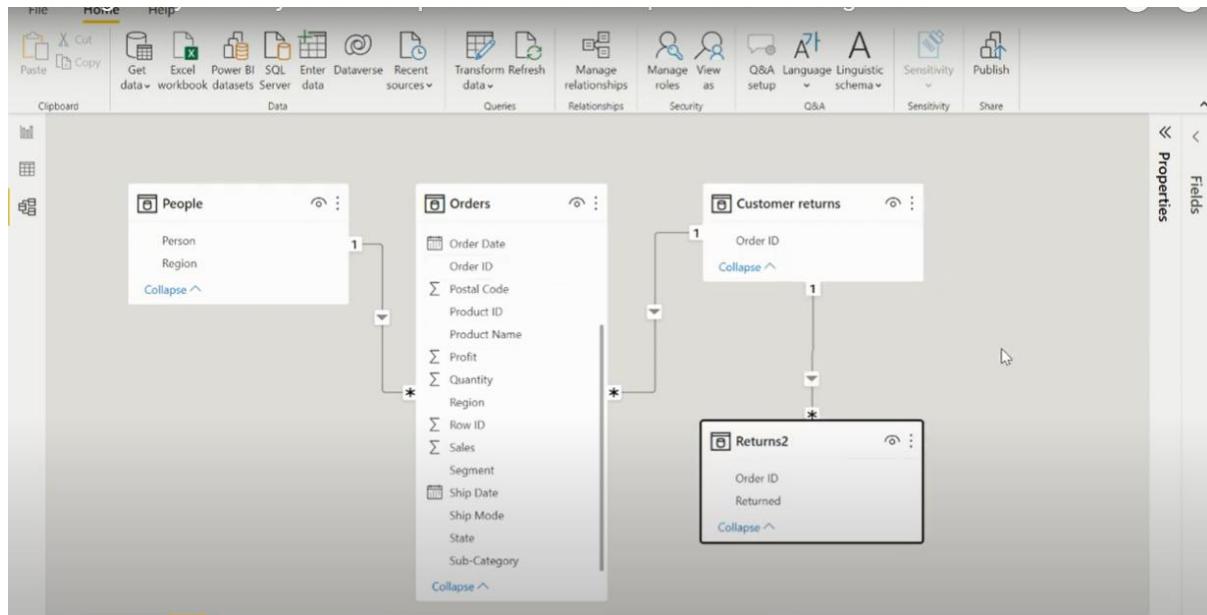
Order ID
CA-2020-152156
CA-2020-138688
US-2019-108966
CA-2018-158182
CA-2021-114412
CA-2020-161389
US-2019-118983
CA-2018-105893
CA-2018-167164
CA-2018-143336
CA-2020-137330
US-2021-156909
CA-2019-106320
CA-2020-121755
US-2019-150630
CA-2021-107727
CA-2020-117590
CA-2019-117415
CA-2021-120999
CA-2020-101343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



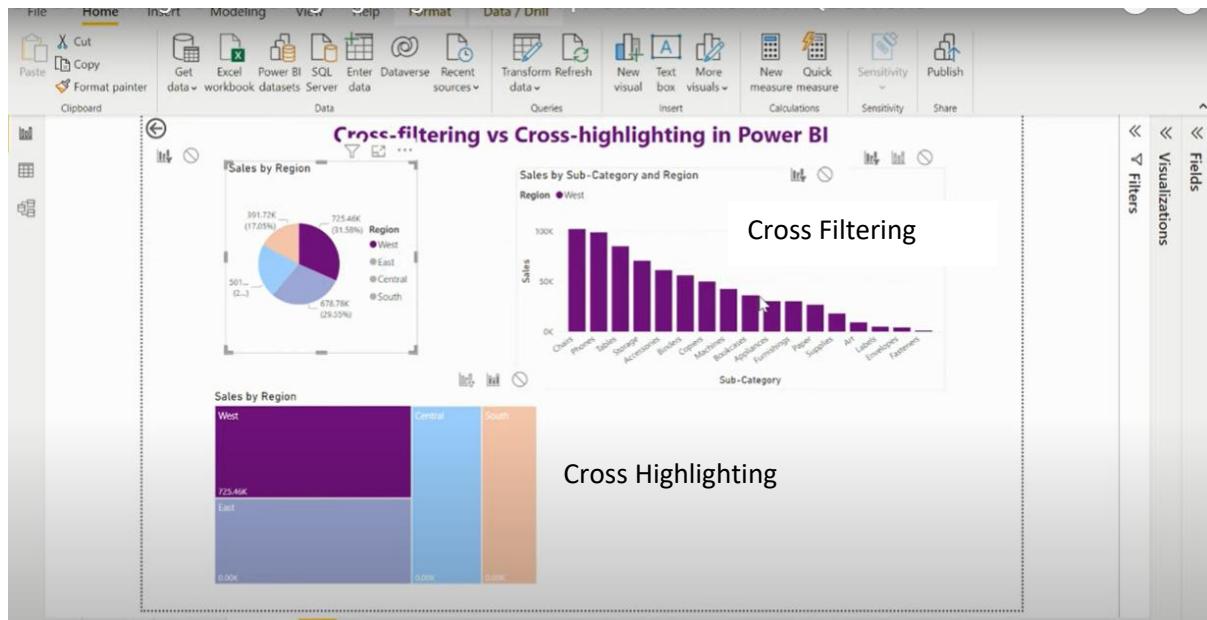
The screenshot shows the Power BI Query Editor interface. A red circle highlights the 'Properties' section on the right, which displays the name 'Customer returns'. Below it, the 'Applied Steps' section lists the steps taken: 'Source', 'Removed Other Columns', and 'Removed Duplicates'. The main area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

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

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

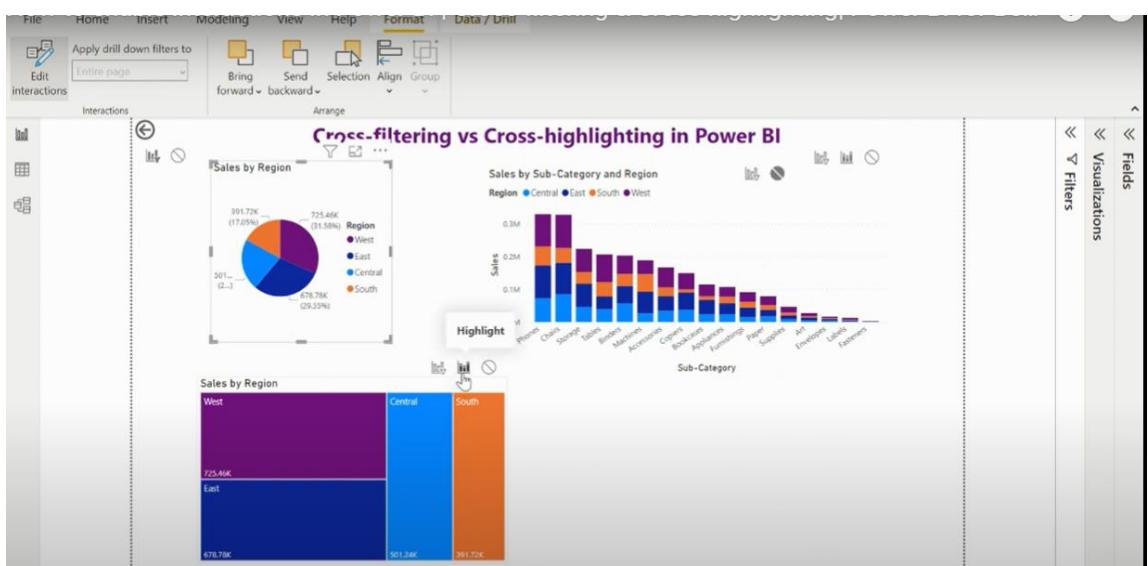
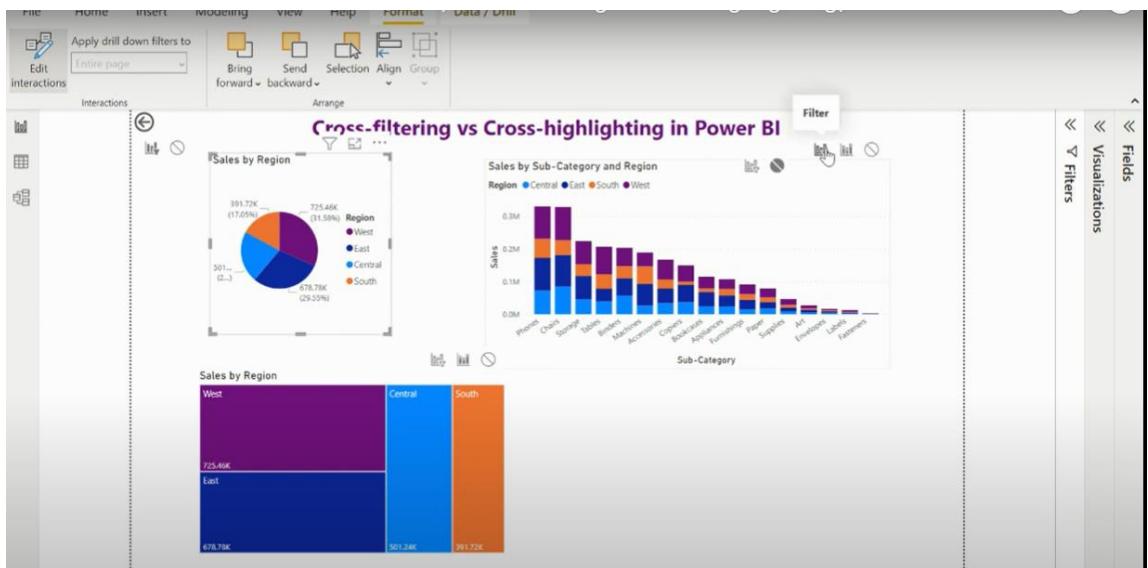
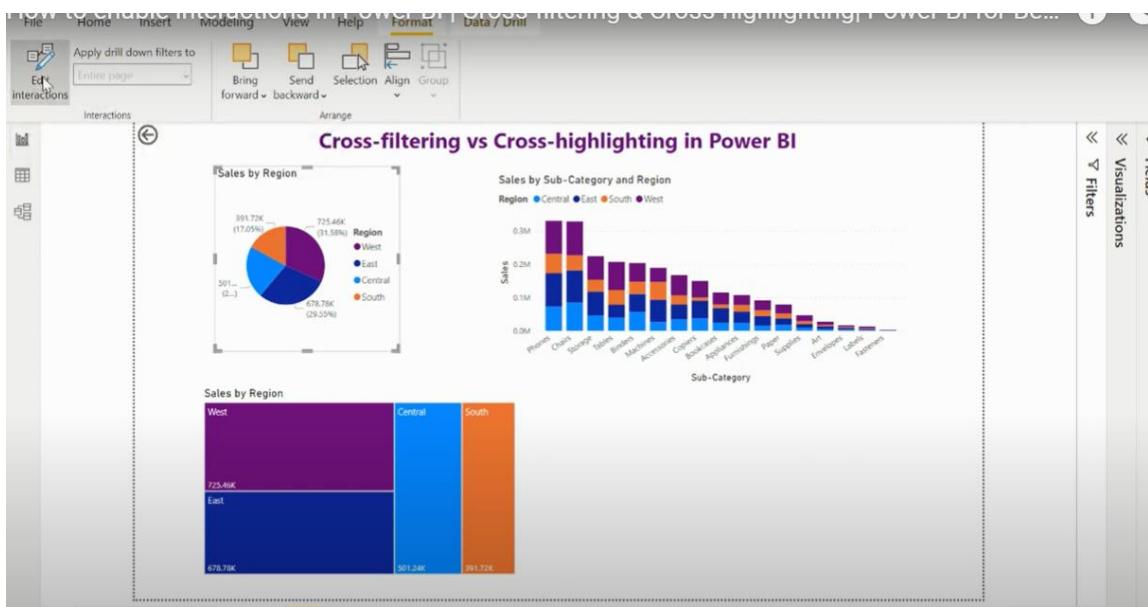


### Cross filtering v/s cross highlighting



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

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



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

YTD – Start of current year to today's date

QTD – Start of current quarter to today's date

MTD – Start of current month to today's date

### Create a new measure

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

Add Card visualization -> select newly created measure

The screenshot shows the Power BI Model view. The 'Visualizations' pane on the right is highlighted with a red circle, indicating the selection of the 'Card' visualization type. The 'Fields' pane on the far right lists various fields from the 'Orders' table, including Order ID, Postal Code, Product ID, Product Name, Profit, Quantity, Region, Sales, Segment, Ship Date, Ship Mode, State, Sub-Category, Total QTD Sales, Total Year to date Sales, Total YTD days, Total YTD Sales, People, and Returns2. The newly created measure, 'Total Year to date Sales', is listed under the 'Values' section of the 'Card' visualization settings.

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

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

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

Similarly, QTD and MTD measures can be created.

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

Create a new measure

The screenshot shows the Power BI Model view. A new measure is being created with the name 'days\_in\_ytd'. The formula bar contains the DATEDIFF function: `DATEDIFF(Date1, Date2, Interval)`. The dropdown menu for 'Interval' is open, showing 'DAY' as the selected option. A tooltip message says: 'Too few arguments were passed to the DATEDIFF function. The minimum argument is 2.' The Fields pane on the right lists various fields, and the Visualizations pane shows a card visualization with the value '733.22K'.

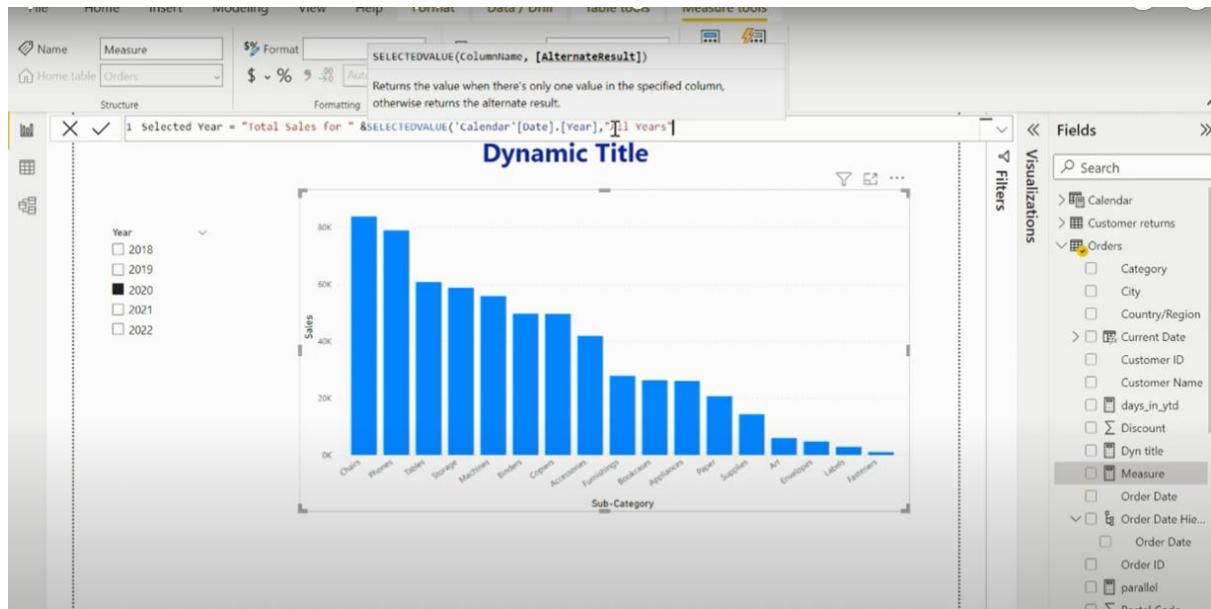
Add Card visualization -> select newly created measure

The screenshot shows the Power BI Home view. A card visualization displays the value '210' for the measure 'days\_in\_ytd'. The Fields pane on the right shows the newly created measure 'days\_in\_ytd' selected. The formula bar at the top of the screen also shows the DATEDIFF function with 'DAY' selected for the interval.

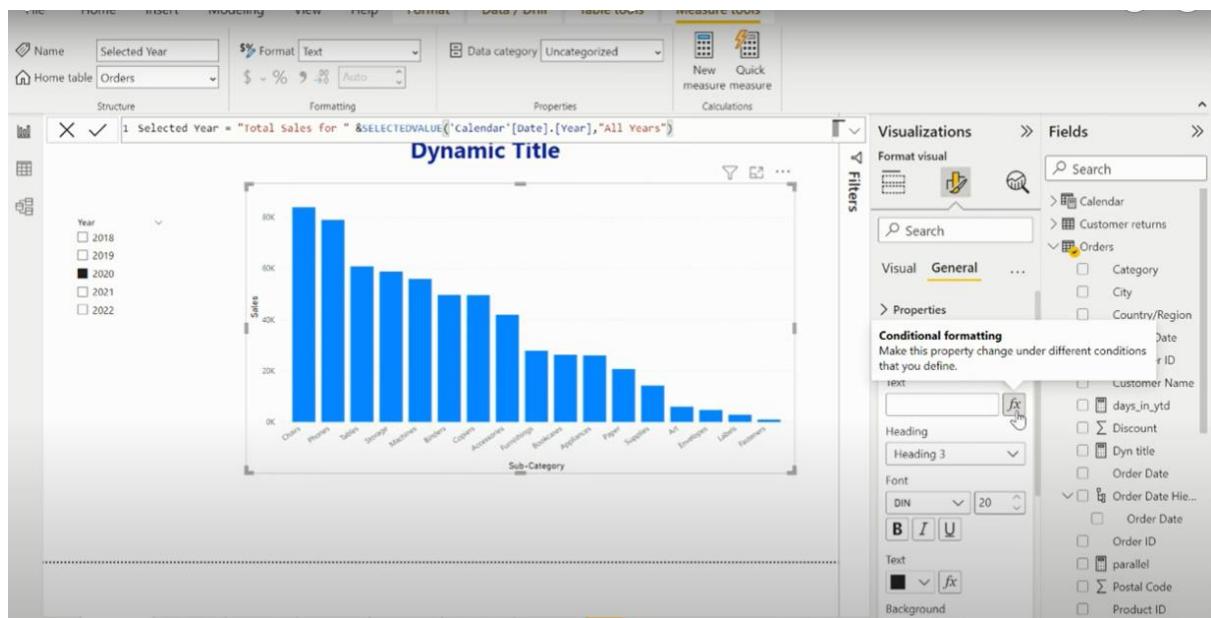
## Dynamic Titles in Power BI?

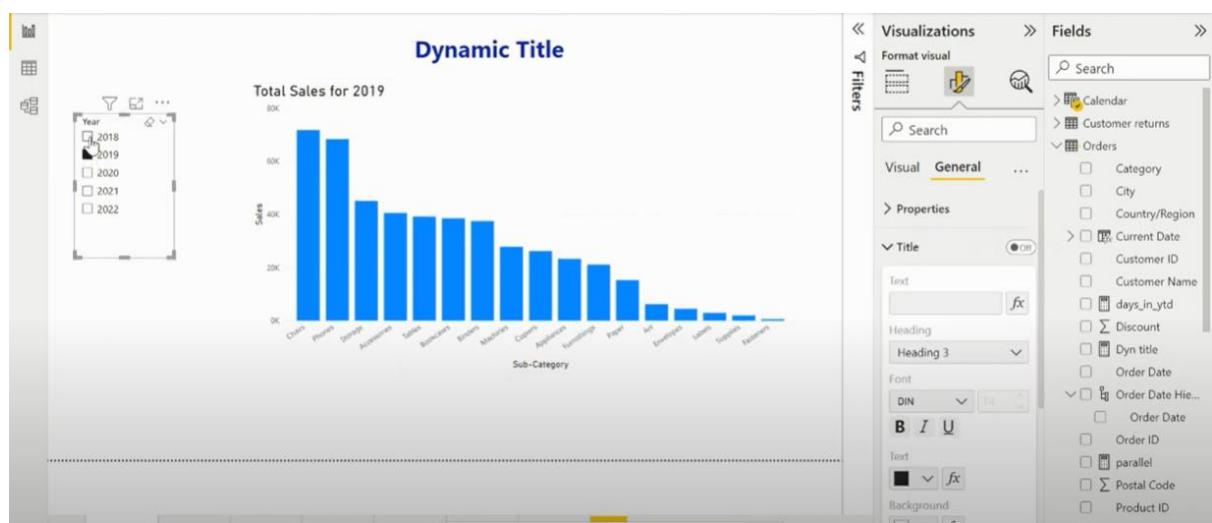
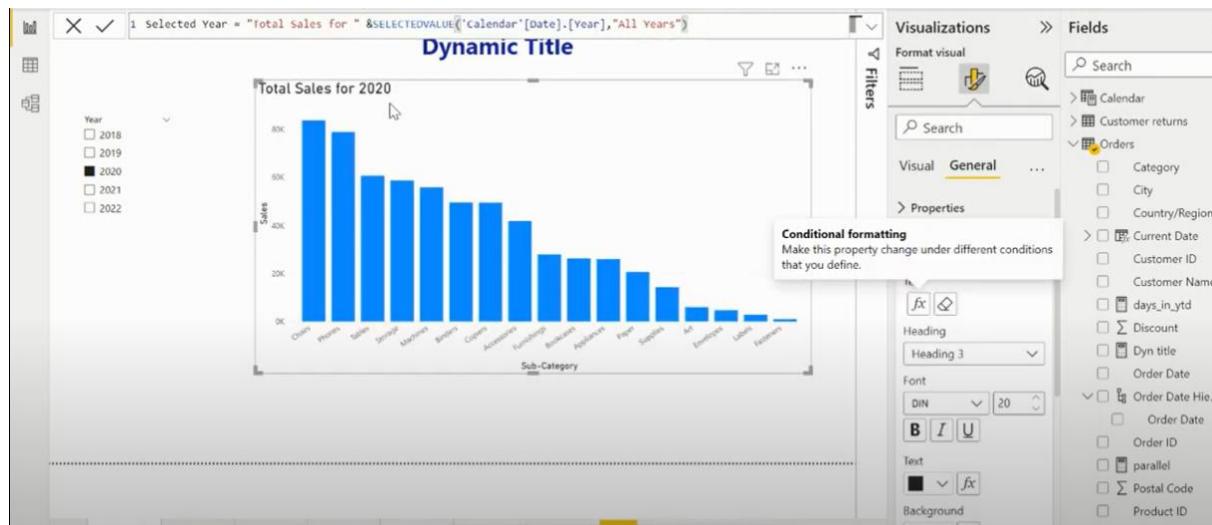
We want to update the visualization table, according to the slicer year selection.

Create a new measure



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





## Sum v/s SumX in Power BI

Sum : same as sum function in excel

SumX : same as sumproduct function in excel

SumX Syntax : (Table Name, value1 \* value2)

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

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

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

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

## Waterfall Chart

Helps to visualize the change in value over a time period

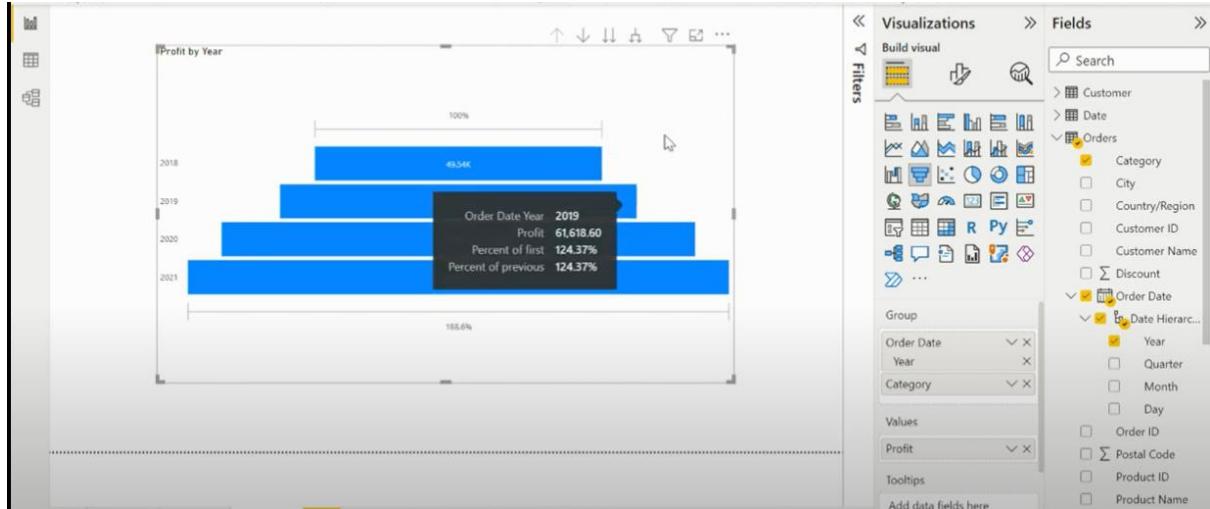
The screenshot shows a Waterfall chart titled "Profit by Year and Category". The chart displays profit values for different categories across two years (2018 and 2019). A tooltip is shown for the "Furniture" category, indicating the following data:

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

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

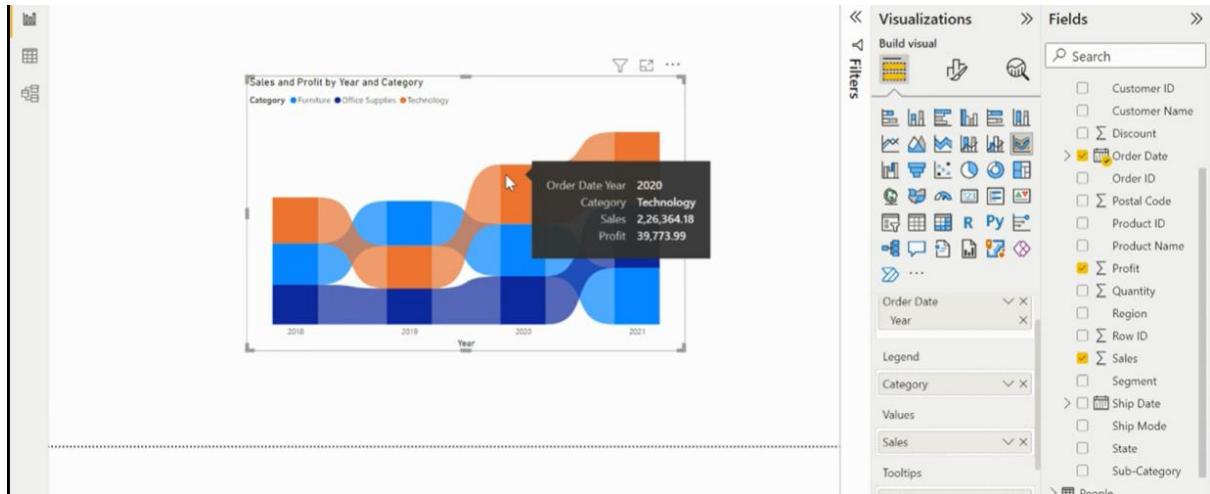
## Funnel Chart

Helps to visualize the value over a time period



## Ribbon Chart

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



## Parallelperiod DAX function in Power BI:

Create a new measure

The screenshot shows the Power BI 'Measure tools' ribbon. A new measure is being created with the name 'Measure'. The formula bar shows '\$ % > CALCULATE(SUM(Orders[sales]),parallel)'. A tooltip for 'parallel' is displayed, explaining it evaluates an expression in a context modified by filters. The Fields pane on the right lists various dimensions and measures, including 'Orders' and 'Measure'.

Month	Sales	parallel
January	43,971.37	96,999.04
February	20,301.13	43,971.37
March	58,872.35	20,301.13
April	36,521.54	58,872.35
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, explaining it returns a parallel period of dates by the given set of dates and a specified interval. The Fields pane on the right lists various dimensions and measures, including 'Orders' and 'Measure'.

Month	Sales	parallel
January	43,971.37	96,999.04
February	20,301.13	43,971.37
March	58,872.35	20,301.13
April	36,521.54	58,872.35
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 its purpose: "Returns a set of dates in the current selection from the previous year." The Fields pane on the right lists various columns from the 'Orders' table, including Date, Month, Sales, and Parallel Period LM.

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

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

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

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

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

Parallelperiod      Parallelperiod      Sameperiod

-1,month            -1,year

### Calculate frequency of a word in a column in Power BI

Power BI Scenario based Interview question - Calculate frequency of a word in a column									
Name	Nick name	Products	Qty Sold	Frequency of Nick name					
Aaron Miller	Aaron,Miller,Aaro	Product 1	2	3	Frequency of Nick name				
Adam Jenkins	Adam,Jenkins,Andy	Product 2	4						
Andrew Adams	Andrew,Adams,Andy	Product 1	1	50%	% Frequency of Nick name				
Andrew Graham	Andrew,Graham,Andy	Product 3	3						
Bobby Murray	Bobby,Murray,Bob	Product 4	5						
Benjamin Murray	Benjamin,Murray	Product 3	2						

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

## Create a measure

File Home Insert Modeling View Help Table tools Measure tools

Name Measure Data category Uncategorized

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

Evaluates an expression in a context modified by filters.

New Quick measure measure Calculations

**Calculate frequency o**

**Fields**

Visualizations

Filters

**Measure**

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

2

**Calculate frequency o**

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

50%

% Frequency of Nick name

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

File Home Insert Modeling View Help Table tools Measure tools

Name Measure Data category Uncategorized

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

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

**Calculate frequency of a word in a column**

**Fields**

Visualizations

Filters

**Measure**

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

2

**Calculate frequency of a word in a column**

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

3

Frequency of Nick name

50%

% Frequency of Nick name

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

File Home Insert Modeling View Help Table tools Measure tools

Name Measure Data category Uncategorized

DIVIDE(numerator, denominator, [AlternateResult])

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

**Calculate frequency of a word in a column**

**Fields**

Visualizations

Filters

**Measure**

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

2

**Calculate frequency of a word in a column**

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

3

Frequency of Nick name

50%

% Frequency of Nick name

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

3

Count of Andy

**POWER BI SCENARIO-BASED INTERVIEW QUESTIONS**

### Calculate frequency of a word in a column

**Table:**

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

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

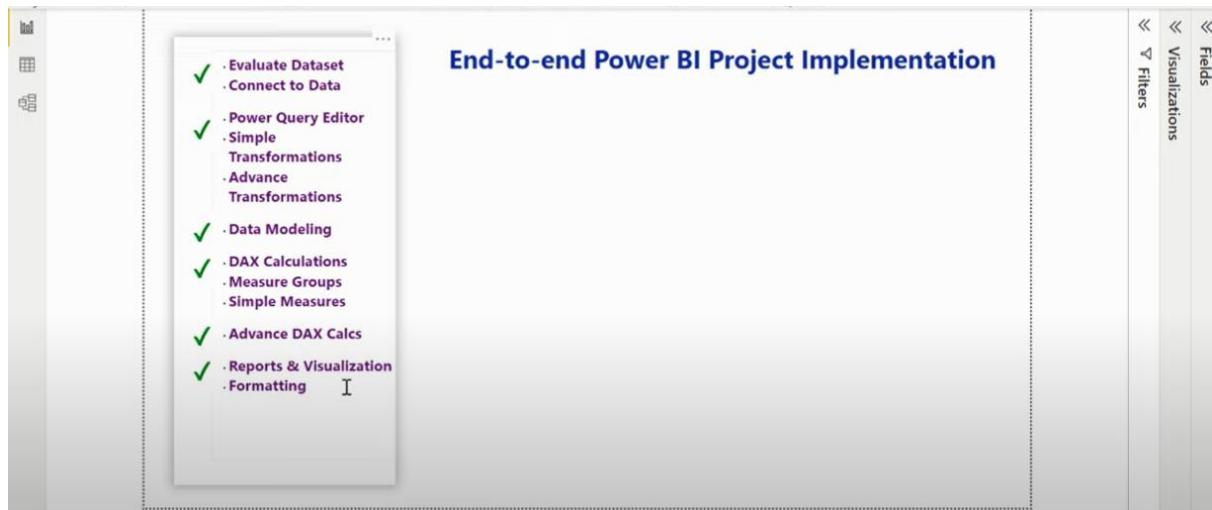
**Visualizations:**

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

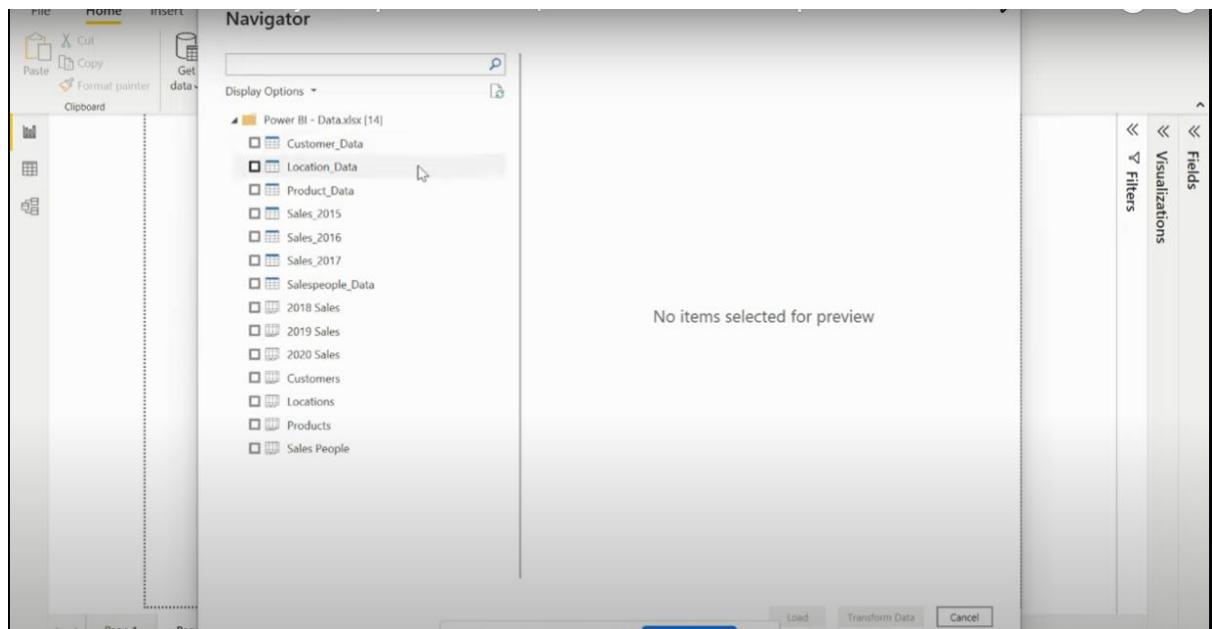
**Fields:**

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

## End to End Power BI Project Implementation



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



Salespeople\_Data

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

Rename table names if required

Properties

Name: Sales\_2017

Applied Steps

Changed Type

## Rename columns if required

The screenshot shows the Power BI Data Editor interface. A table named 'Table.TransformColumnTypes(Product\_Data\_Table, {"Product ID", type text}, {"Product Name", type text}, {"Product Cost", type whole number}, {"Original Sale Price", type whole number}, {"Discount", type whole number})' is displayed. The 'Product Cost' column has been converted to a whole number type. The 'Properties' pane on the right indicates the name is 'Products' and the applied step is '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 named 'Table.TransformColumnTypes(Location\_Data\_Table, {"Location ID", type text}, {"Name", type text}, {"Households", type whole number}, {"Median Income", type whole number}, {"Land Area", type whole number}, {"Water Area", type whole number})' is displayed. The 'Water Area' column is selected, and a context menu is open with options like Copy, Remove Columns, and Replace Values. The 'Properties' pane on the right indicates the name is 'Locations' and the applied step is 'Changed Type'.

## To merge tables with similar columns

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

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

## To Group the columns

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

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [7]' pane lists several tables: Customers, Locations, Products, Sales\_2018, Sales\_2019, Sales\_2020, and Salespeople. The Salespeople table is currently selected. In the main area, a table preview for the Salespeople query is displayed, showing columns 'Salesperson ID' and 'Salesperson Name'. A context menu is open over the first few rows of the table, with the 'Move To Group' option highlighted. Below it, there is a 'New Group...' option.

This screenshot shows the 'New Group' dialog box from the previous step. It contains fields for 'Name' (set to 'Data model') and 'Description'. The 'OK' button is highlighted, indicating the next step in creating a new group.

The final 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

Queries [7]

Customers

Locations

Products

Sales\_2018

Sales\_2019

Sales\_2020

Salespeople

State

1 CA California 33.83529

2 CA California 38.00492

3 CA California 35.37329

4 CA California 37.87159

5 CA California 34.18084

6 CA California 33.15809

7 CA California 32.64

8 CA California 36.82523

9 CA California 37.97798

10 CA California 33.87529

11 CA California 33.64113

12 CA California 37.70583

13 CA California 33.94001

14 CA California 34.0239

15 CA California 32.79477

16 CA California 38.4088

17 CA California 34.06862

18 CA California 33.11921

19 CA California 38.24936

20 CA California 34.09223

21 CA California 37.54827

22 CA California 36.74773

23 CA California 33.87029

24 CA California 33.77391

25

1.2 Latitude

1.2 Longitude

Query Settings

PROPERTIES

Name: Locations

APPLIED STEPS

Source: Navigation: Changed Type: Removed Columns

Queries [7]

Customers

Locations

Products

Sales\_2018

Sales\_2019

Sales\_2020

Salespeople

Latitude

1 33.83529

2 38.00492

3 35.37329

4 37.87159

5 34.18084

6 33.15809

7 32.64

8 36.82523

9 37.97798

10 33.87529

11 33.64113

12 37.70583

13 33.94001

14 34.0239

15 32.79477

16 38.4088

17 34.06862

18 33.11921

19 38.24936

20 34.09223

21 37.54827

22 36.74773

23 33.87029

24 33.77391

1.2 Longitude

1.2 Population

Time Zone

State

State - Copy

Query Settings

PROPERTIES

Name: Locations

APPLIED STEPS

Source: Navigation: Changed Type: Removed Columns Duplicated Column

Queries [7]

Customers

Locations

Products

Sales\_2018

Sales\_2019

Sales\_2020

Salespeople

State

1 California City 33.83529

2 California City 38.00492

3 California City 35.37329

4 California City 37.87159

5 California City 34.18084

6 California City 33.15809

7 California City 32.64

8 California City 36.82523

9 California City 37.97798

10 California City 33.87529

11 California City 33.64113

12 California City 37.70583

13 California City 33.94001

14 California CDP 34.0239

15 California City 32.79477

16 California City 38.4088

17 California City 34.06862

18 California City 33.11921

19 California City 38.24936

20 California City 34.09223

21 California City 37.54827

22 California City 36.74773

23 California City 33.87029

24 California City 33.77391

1.2 Latitude

1.2 Longitude

1.2 Population

Query Settings

PROPERTIES

Name: Locations

APPLIED STEPS

Source: Navigation: Changed Type: Removed Columns Duplicated Column

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 dropdown menu with various text transformation tools. The 'uppercase' option is highlighted. A preview pane on the right shows the original data ('CA', 'California') being converted to uppercase ('CALIFORNIA', 'California'). The 'APPLIED STEPS' panel on the right side of the editor shows the step 'Reordered Columns'.

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

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

The screenshot shows the Power BI 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' and 'By Position'. 'By Delimiter' is chosen, and a sub-menu shows 'Split Column by Delimiter' with a placeholder 'Type text')).' A preview pane on the right shows the original data ('CA', 'California') being converted to uppercase ('CALIFORNIA', 'California'). The 'APPLIED STEPS' panel on the right side of the editor shows the step 'Uppercased Text'.

Queries [7]

- Customers
- Locations
- Products
- Sales\_2018
- Sales\_2019
- Sales\_2020
- Salespeople

Split Column by Number of Characters

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

Number of characters: 4

Split:

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

Advanced options

OK Cancel

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

## How to bring in data from web in table

We want the California population data by city in the table

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

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

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

The screenshot shows the Power BI 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 center. It shows a preview of data from 'https://www.california-demographics.com/cities\_by\_population'. The dialog includes tabs for 'Basic' and 'Advanced', a URL input field, and 'OK' and 'Cancel' buttons.

This screenshot shows the 'Navigator' feature in Power BI. The 'Queries' pane on the left lists 'Data model [5]', 'Locations', 'Products', 'Sales\_2018', 'Salespeople', and 'Other Queries [2]'. The 'Locations' item is selected. In the center, a 'Table View' window displays a table titled 'Table 1' with columns 'Rank', 'City', and 'Population'. The table lists various California cities with their populations. The 'Properties' and 'Applied Steps' panes on the right show the step 'Changed Type1'.

Queries [8]

Data model [5]

- Customers
- Locations
- Products
- Sales\_2018
- Salespeople

Other Queries [3]

- Sales\_2019
- Sales\_2020
- Table 1**

Table 1

Rank | City | Population

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

Query Settings

PROPERTIES

- Name: Table 1
- All Properties

APPLIED STEPS

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

Queries [8]

Data model [5]

- Customers
- Locations
- Products
- Sales\_2018
- Salespeople

Other Queries [3]

- Sales\_2019
- Sales\_2020
- CA Population**

CA Population

Rank | City | Population

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

Query Settings

PROPERTIES

- Name: CA Population
- All Properties

APPLIED STEPS

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

Contextual menu open on CA Population:

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

## To combine different tables

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

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

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

- 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, listing items like 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 contains 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 "Source", "Navigation", "Changed Type", "Removed Columns", "Duplicated Column", "Reordered Columns", "Uppercased Text", "Split Column by Position", and "Changed Type1".

**Merge**

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

**Locations**

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

**CA Population**

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

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

Use fuzzy matching to perform the merge

Fuzzy matching options

OK Cancel

**Merge**

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

**Locations**

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

**CA Population**

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

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

Use fuzzy matching to perform the merge

Fuzzy matching options

Estimating matches based on data previews

OK Cancel

Queries [8]

DATA MODEL

- Data model [6]
  - Customers
  - Locations
  - Products
  - Sales\_2018
  - Salespeople
  - CA Population
- Other Queries [2]

Table

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

Queries [8]

DATA MODEL

- Data model [6]
  - Customers
  - Locations
  - Products
  - Sales\_2018
  - Salespeople
  - CA Population
- Other Queries [2]

Table

Expand Aggregate

(Select All Columns)

Rank

City

Population

Use original column name as prefix

OK Cancel

Queries [8]

DATA MODEL

- Data model [6]
  - Customers
  - Locations
  - Products
  - Sales\_2018
  - Salespeople
  - CA Population
- Other Queries [2]

Table

Search Columns to Expand

Expand Aggregate

(Select All Columns)

Rank

City

Population

Use original column name as prefix

OK Cancel

Queries [8]

Data model [6]

- Customers
- Locations
- Products
- Sales\_2018
- Salespeople
- CA Population
- Other Queries [2]

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

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

Queries [8]

Data model [6]

- Customers
- Locations
- Products
- Sales\_2018
- Salespeople
- CA Population
- Other Queries [2]

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

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

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	198301

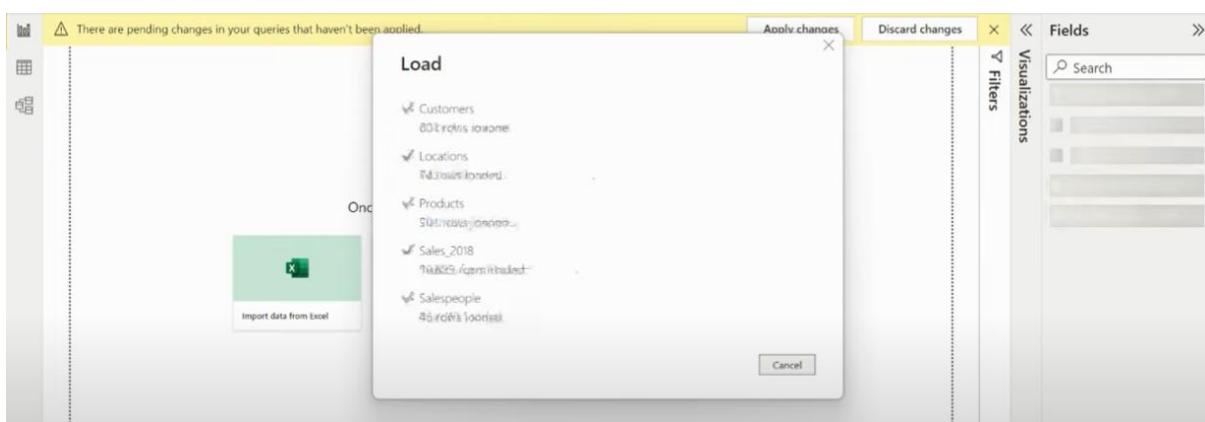
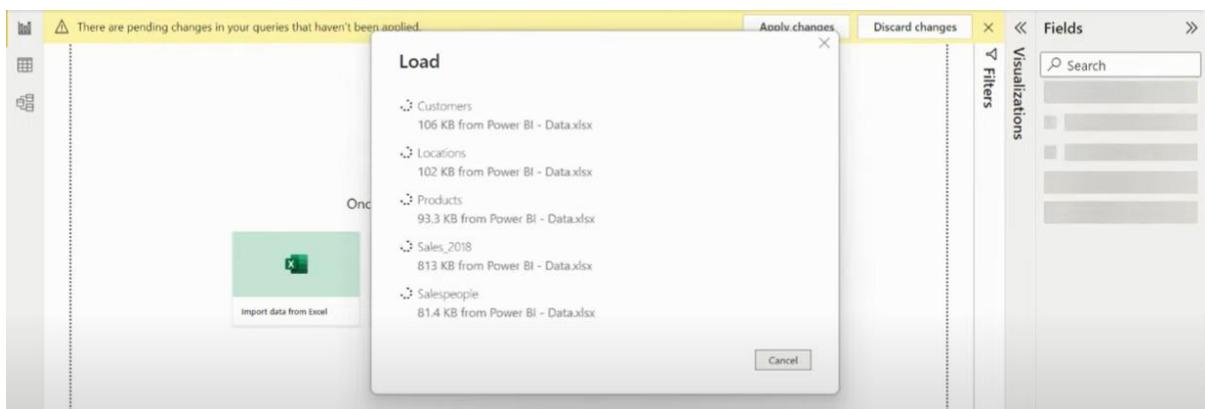
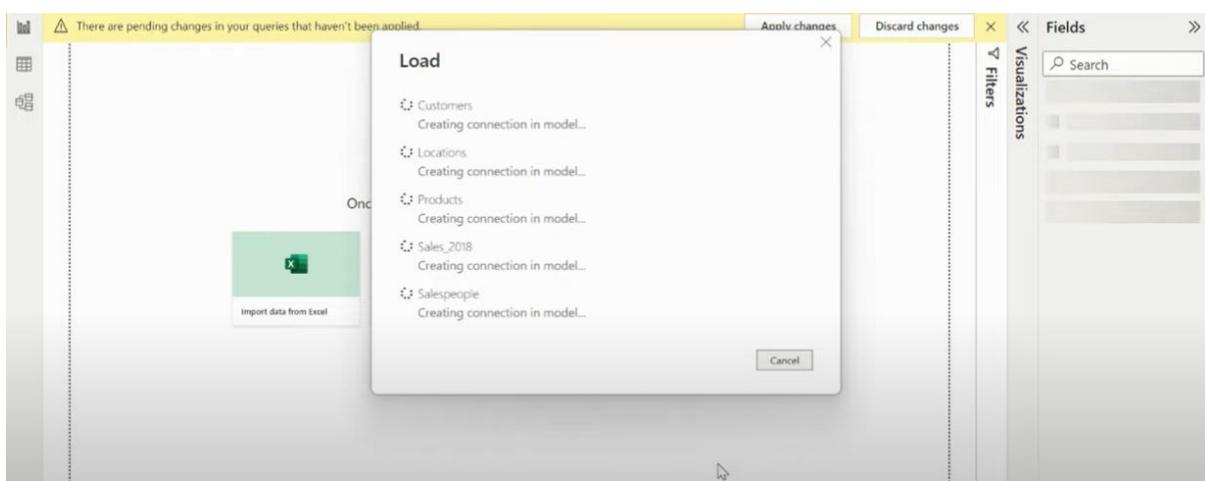
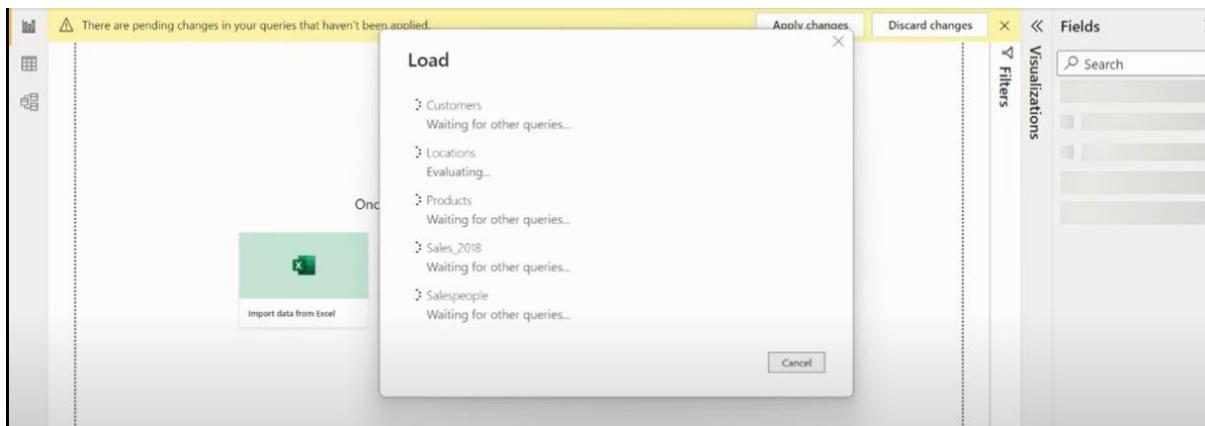
Query Settings

Properties

Applied Steps

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

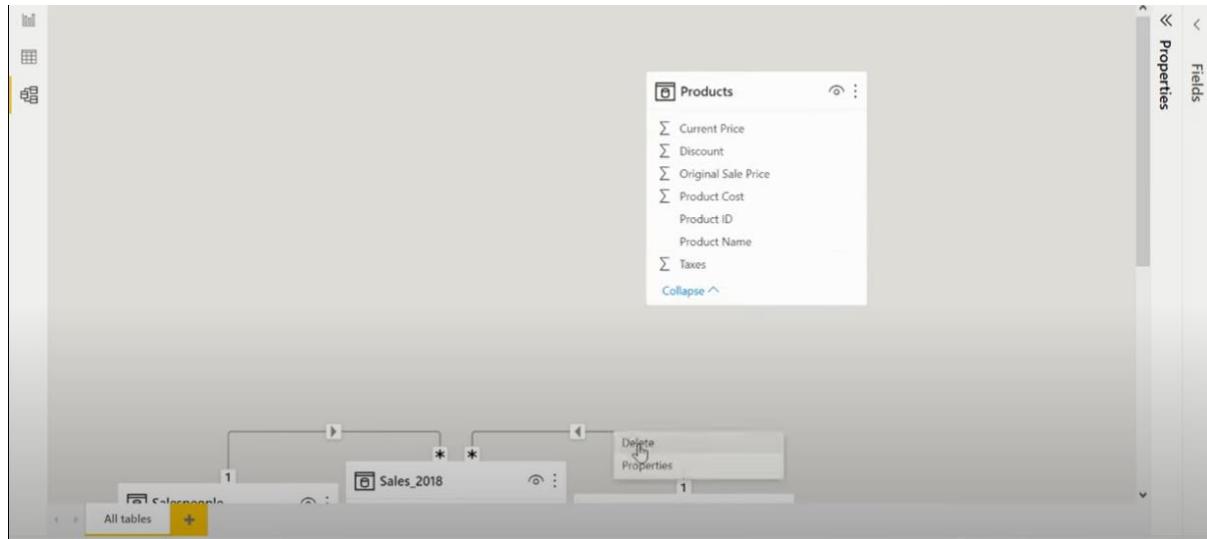
Once transformations are done, click on Close & apply then data will be loaded into Power BI



## Data Modelling

Power BI will create the relations between the tables automatically.

Select the relation line -> right click -> delete



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

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

The screenshot shows the Power BI Data Model view. A new table named 'Customers' has been created. A context menu is open over the table, with the 'Calendars' option selected. The table itself contains 19 rows of customer data, each with a unique Customer ID and name. On the right side, a 'Fields' pane lists other tables: Customers, Locations, Products, Sales\_2018, and Salespeople. The 'Customers' table is currently selected in the 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

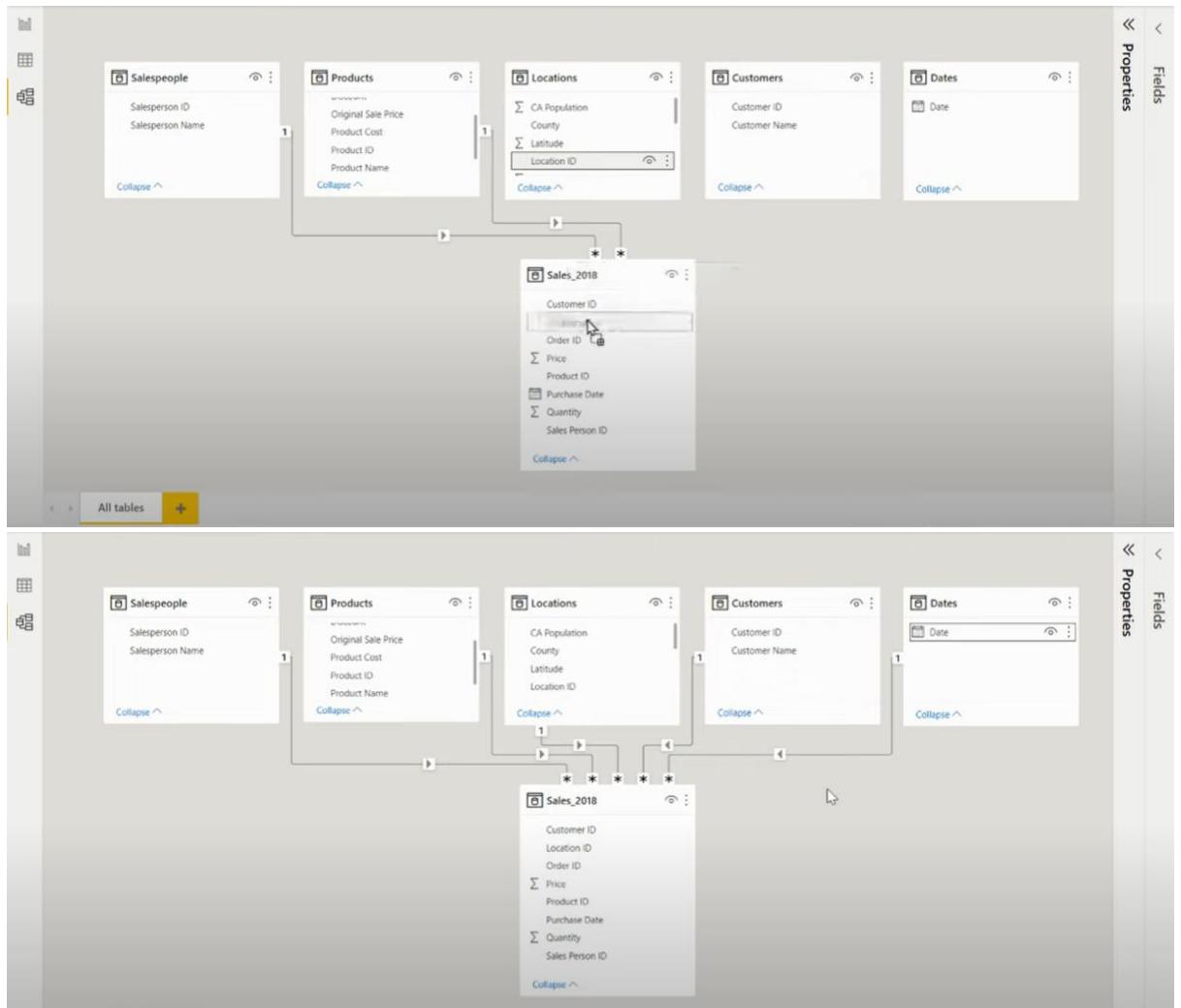
Lookup table (data table) – holds the primary keys

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 tree view of available fields. Under the 'Measures' node, there is a single item: 'Quantity Sold = SUM(Sales\_2018[Quantity])'. The report canvas in the center shows a small preview of a visual with a grid and some text. The status bar at the bottom indicates 'Data refreshed 8/14/2022, 9:10:59 PM'.

### 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 after performing the steps described in the text. The report canvas now displays a table with two columns: 'Name' and 'Key Measures'. The table contains data for various names and their corresponding key measures. The 'Fields' pane on the right shows a group named 'Key Measures' containing the 'Quantity Sold' measure. A red box highlights this 'Key Measures' group in the Fields pane. The status bar at the bottom indicates 'Data refreshed 8/14/2022, 9:10:59 PM'.

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
Alan Smith	26 26
<b>Total</b>	<b>21078 21078</b>

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

The screenshot shows the Power BI Measure Tools dialog. The 'Name' field contains 'Quantity Sold'. The 'Home table' dropdown is set to 'Customers'. The 'Format' dropdown is set to 'Whole number'. The 'Data category' dropdown is set to 'Uncategorized'. The 'Calculations' section shows the formula: `Quantity Sold = SUM(Sales_2018[Quantity])`. The 'Fields' pane on the right lists various entities: Customers, Key Measures, Locations, Products, Sales\_2018, and Salespeople. The 'Quantity Sold' field is selected under the 'Key Measures' group.

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

## More DAX expressions

The screenshot shows the Power BI Data Editor interface. In the top ribbon, the 'Measure tools' tab is selected. A new measure named 'Total Sales' has been created, based on the 'Sales\_2018' table. The formula is `SUMX(Sales_2018,Sales_2018[Quantity]*Sales_2018[Price])`. The Fields pane on the right lists the 'Key Measures' group, which includes 'Quantity Sold' and 'Total Sales'. The main area displays a table of sales data.

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

To change the currency

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

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

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

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

Screenshot of Power BI Data Editor showing a table named "Sales\_2018". The table has columns: Name, Sales ID, Order ID, and Price. A measure named "Total Transactions" is defined as COUNTROWS(Sales\_2018). The Fields pane on the right shows various measures like Total Sales, Total Transactions, and Quantity Sold.

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

To create DAX expression, two columns located in different tables

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

Screenshot of Power BI Data Editor showing a table named "Sales\_2018". A measure named "Total Costs" is defined as SUMX(Sales\_2018, Sales\_2018[Quantity] \* [Price]). The Fields pane on the right shows various measures like Total Sales, Total Transactions, and Total Costs.

	Quantity	Price	Total Cost
Aaron Carr	11	24	\$26.846
Aaron Cruz	16	26	\$32.099
Aaron Day	10	16	\$22.123
Aaron Johnson	9	16	\$18.038
Aaron Miller	11	20	\$28.716
Aaron Mills	13	20	\$17.891
Aaron Moreno	15	26	\$27.996
Aaron Tucker	16	26	\$38.158
Adam Alexander	11	21	\$27.727
Adam Bailey	19	32	\$48.930
Adam Duncan	17	30	\$34.143
Adam Hernandez	13	21	\$18.793
Adam Hunter	9	17	\$21.762
Adam Jenkins	19	38	\$50.294
Adam Mccoy	12	22	\$31.402
Adam McDonald	11	18	\$27.727
Adam Myers	11	24	\$30.997
Adam Riley	15	26	\$45.066
Adam Thompson	14	29	\$39.779
Adam Wheeler	9	16	\$18.411
Adam White	11	21	\$18.967
Alan Gomez	9	19	\$26.129
Alan Green	13	31	\$36.911
Alan Miller	23	38	\$51.804
Alan Parker	18	36	\$39.137
Alan Perry	18	38	\$44.752
Alan Scott	22	43	\$52.058
Alan Smith	17	36	\$34.043
Total	21078	10809	\$2,56.61.209

1 Total Costs = SUMX(Sales\_2018,Sales\_2018[Quantity] \* RELATED(Products[Product Cost]))

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

Customer Name	Total Sales	Quantity Sold	Total Profits	Total Costs	Total Transactions
Aaron Carr	\$20,946	11	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		
<b>Total</b>	<b>\$2,56,61,209</b>	<b>10889</b>	<b>21078</b>		

Using 2 measures to create 3<sup>rd</sup> measure

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

Customer Name	Total Sales	Quantity Sold	Total Profits	Total Costs	Total Transactions
Aaron Carr	\$26,846	\$17,146	\$9,700		
Aaron Cruz	\$32,099	\$20,715	\$11,384		
Aaron Day	\$22,123	\$16,046	\$6,077		
Aaron Johnson	\$18,038	\$11,812	\$6,226		
Aaron Miller	\$28,716	\$19,256	\$9,460		
Aaron Mills	\$17,891	\$13,519	\$4,372		
Aaron Moreno	\$27,996	\$15,594	\$12,402		
Aaron Tucker	\$38,158	\$26,291	\$11,867		
Adam Alexander	\$32,187	\$23,439	\$8,758		
Adam Bailey	\$48,930	\$33,488	\$15,442		
Adam Duncan	\$34,143	\$24,473	\$9,669		
Adam Hernandez	\$18,793	\$12,056	\$6,737		
Adam Hunter	\$21,762	\$15,122	\$6,640		
Adam Jenkins	\$50,294	\$28,954	\$11,340		
Adam McCoy	\$31,402	\$21,636	\$9,766		
Adam McDonald	\$27,727	\$19,470	\$8,257		
Adam Myers	\$30,997	\$23,027	\$7,970		
Adam Riley	\$45,066	\$34,057	\$10,999		
Adam Thompson	\$39,779	\$24,656	\$15,123		
Adam Wheeler	\$18,411	\$10,419	\$8,392		
Adam White	\$18,967	\$12,625	\$6,342		
Alan Gomez	\$26,129	\$16,208	\$9,921		
Alan Green	\$36,911	\$23,836	\$13,075		
Alan Miller	\$51,804	\$36,701	\$15,103		
Alan Parker	\$39,137	\$25,094	\$14,043		
Alan Perry	\$44,752	\$32,093	\$12,660		
Alan Scott	\$52,058	\$35,767	\$16,291		
Alan Sims	\$34,741	\$14,003	\$10,738		
<b>Total</b>	<b>\$2,56,61,209</b>	<b>\$1,73,17,316</b>	<b>10889</b>	<b>21078</b>	

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

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

Customer Name	Total Sales	Quantity Sold	Total Profits	Total Costs	Total Transactions
Alan Sims	\$9,051	\$24,043	\$14,992		
Alan Thomas	\$17,145	\$52,423	\$35,278		
Albert Cunningham	\$14,110	\$44,945	\$30,835		
Albert Jacobs	\$10,981	\$35,470	\$24,489		
Albert Kennedy	\$10,227	\$33,240	\$23,913		
Albert King	\$12,235	\$46,671	\$31,436		
Albert Robinson	\$10,753	\$35,545	\$24,792		
Albert Young	\$13,556	\$33,193	\$10,827		
Andrew Burns	\$7,980	\$25,429	\$17,459		
Andrew Burns	\$18,431	\$33,915	\$23,384		
Andrew Butler	\$8,811	\$31,918	\$23,107		
Andrew Fernandez	\$7,791	\$16,527	\$10,736		
Andrew Graham	\$9,347	\$34,635	\$25,388		
Andrew Hansen	\$9,822	\$28,823	\$19,999		
Andrew James	\$7,643	\$23,815	\$16,172		
Andrew Martin	\$13,114	\$40,179	\$26,665		
Andrew Peters	\$15,385	\$41,127	\$25,762		
Andrew Reynolds	\$15,007	\$41,646	\$26,639		
Andrew Robinson	\$7,076	\$20,602	\$13,526		
Anthony Banks	\$8,721	\$30,869	\$22,148		
Anthony Berry	\$10,760	\$33,969	\$25,209		
Anthony Chapman	\$10,804	\$30,305	\$19,501		
Anthony Chapman	\$10,804	\$30,305	\$19,501		
Anthony Little	\$9,381	\$31,696	\$22,333		
Anthony Parker	\$18,991	\$40,087	\$29,499		
Anthony Simpson	\$4,170	\$18,031	\$13,861		
Anthony Torres	\$10,576	\$31,017	\$20,441		
Anthony Turner	\$16,477	\$47,706	\$31,229		
Antonio Cooper	\$3,611	\$11,129	\$7,518		
<b>Total</b>	<b>\$83,43,893</b>	<b>\$2,56,61,209</b>	<b>\$1,73,17,316</b>	<b>10889</b>	<b>21078</b>

## To remove data hierarchy

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

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

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

## To change datatype of visual

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

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

1 Dates = CALENDAR(AUTO())

	17.145	\$52,423	\$35,278	17	35	32.71%
Alan Thomas	\$17,145	\$52,423	\$35,278	17	35	32.71%
Albert Cunningham	\$14,110	\$44,945	\$30,835	17	38	31.99%
Albert Jacobs	\$10,961	\$35,470	\$24,469	16	38	30.69%
Albert Kennedy	\$10,257	\$34,640	\$23,913	17	25	30.57%
Albert Rice	\$12,235	\$46,071	\$31,436	14	24	32.44%
Albert Robinson	\$10,733	\$35,545	\$24,792	13	30	30.35%
Albert Young	\$3,556	\$14,383	\$10,827	11	15	24.72%
Andrew Adams	\$7,661	\$25,420	\$17,759	12	20	30.49%
Andrew Burns	\$18,421	\$33,815	\$35,384	20	28	34.23%
Andrew Butler	\$8,811	\$31,918	\$23,107	12	24	27.61%
Andrew Fernandez	\$5,791	\$16,527	\$10,736	7	15	35.04%
Andrew Graham	\$9,347	\$34,635	\$25,288	18	32	26.99%
Andrew Hansen	\$9,823	\$28,823	\$19,000	13	23	34.08%
Andrew James	\$7,643	\$23,815	\$16,172	12	24	32.99%
Andrew Martin	\$13,314	\$40,179	\$26,865	18	42	33.14%
Andrew Peters	\$15,365	\$41,127	\$25,762	18	33	37.36%
Andrew Reynolds	\$15,007	\$41,846	\$26,859	12	30	35.60%
Andrew Simpson	\$10,961	\$35,470	\$24,469	10	19	34.23%
Anthony Banks	\$8,721	\$20,869	\$22,149	14	22	20.23%
Anthony Berry	\$10,760	\$35,960	\$25,200	10	21	29.22%
Anthony Chapman	\$10,004	\$30,305	\$19,501	14	24	35.65%
Anthony Fisher	\$7,844	\$22,721	\$14,877	9	26	34.57%
Anthony Little	\$9,361	\$31,696	\$22,335	9	20	29.53%
Anthony Parker	\$18,591	\$40,087	\$29,496	13	32	38.66%
Anthony Simpson	\$14,170	\$37,706	\$18,031	10	15	23.13%
Anthony Torres	\$10,576	\$31,017	\$20,441	14	23	34.10%
Anthony Turner	\$16,477	\$47,706	\$31,229	15	34	34.43%
Antonio Cooper	\$3,611	\$11,129	\$7,518	10	13	32.45%
Antonio Diaz	\$13,711	\$33,451	\$19,740	13	31	40.99%
<b>Total</b>	<b>\$83,43,893</b>	<b>\$2,56,61,209</b>	<b>\$1,73,17,316</b>	<b>10889</b>	<b>21078</b>	<b>32.52%</b>

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

Visualizations Fields

Filters

Build visual

Key Measures

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

Dates

Locations

Products

Sales\_2018

Salespeople

Time Intelligence

Keep all filters

## Time Intelligence Calculations

Customer Name	Total Profits	Total Sales	Total Costs	Total Transactions	Quantity Sold	Profit Margin
Alan Sims	\$9,051	\$24,043	\$14,992	17	26	37.95%
Alan Thomas	\$17,145	\$52,423	\$35,278	17	35	32.71%
Albert Cunningham	\$14,110	\$44,945	\$30,835	17	38	31.99%
Albert Jacobs	\$10,961	\$35,470	\$24,469	16	38	30.69%
Albert Kennedy	\$10,257	\$33,640	\$23,013	11	25	30.77%
Albert Rice	\$12,235	\$46,071	\$31,436	14	24	32.44%
Albert Robinson	\$10,733	\$35,545	\$24,792	13	30	30.35%
Albert Young	\$3,556	\$14,383	\$10,827	11	15	24.72%
Andrew Adams	\$7,661	\$25,420	\$17,759	12	20	30.14%
Andrew Burns	\$18,421	\$33,815	\$35,384	20	28	34.23%
Andrew Butler	\$8,811	\$31,918	\$23,107	12	24	27.61%
Andrew Fernandez	\$5,791	\$16,527	\$10,736	7	15	35.04%
Andrew Graham	\$9,347	\$34,635	\$25,288	18	32	26.99%
Andrew Hansen	\$9,823	\$28,823	\$19,000	13	23	34.08%
Andrew James	\$7,643	\$23,815	\$16,172	12	24	32.99%
Andrew Martin	\$13,314	\$40,179	\$26,865	18	42	33.14%
Andrew Peters	\$15,365	\$41,127	\$25,762	18	33	37.36%
Andrew Reynolds	\$15,007	\$41,846	\$26,859	12	30	35.60%
Andrew Simpson	\$10,961	\$35,470	\$24,469	10	19	34.23%
Anthony Banks	\$8,721	\$20,869	\$22,149	14	22	20.23%
Anthony Berry	\$10,760	\$35,960	\$25,200	10	21	29.22%
Anthony Chapman	\$10,004	\$30,305	\$19,501	14	24	35.65%
Anthony Fisher	\$7,844	\$22,721	\$14,877	9	26	34.57%
Anthony Little	\$9,361	\$31,696	\$22,335	9	20	29.53%
Anthony Parker	\$18,591	\$40,087	\$29,496	13	32	38.66%
Anthony Simpson	\$14,170	\$37,706	\$18,031	10	15	23.13%
Anthony Torres	\$10,576	\$31,017	\$20,441	14	23	34.10%
Anthony Turner	\$16,477	\$47,706	\$31,229	15	34	34.43%
Antonio Cooper	\$3,611	\$11,129	\$7,518	10	13	32.45%
Antonio Diaz	\$13,711	\$33,451	\$19,740	13	31	40.99%
<b>Total</b>	<b>\$83,43,893</b>	<b>\$2,56,61,209</b>	<b>\$1,73,17,316</b>	<b>10889</b>	<b>21078</b>	<b>32.52%</b>

Visualizations Fields

Filters

Build visual

Key Measures

- Customers
- Dates
- Locations
- Products
- Sales\_2018
- Salespeople
- Time Intelligence

Add data fields here

Drill through

Cross-report

Keep all filters

Add drill-through fields here

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

Name Measure Format Data category Uncategorized

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

Evaluates an expression in a context modified by filters.

Measure Tools

New Quick measure measure Calculations

Structure

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

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

Date Total Sales

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

Visualizations Fields

Filters

Build visual

Key Measures

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

Dates

Locations

Products

Sales\_2018

Salespeople

Time Intelligence

Keep all filters

The screenshot shows the Power BI desktop interface. On the left is a table visualization titled "Customer Name" with columns: Customer Name, Total Profits, Total Sales, Total Costs, Total Transactions, Quantity Sold, and Profit Margin. The data includes rows for various customers like Aaron Cruz, Aaron Day, etc., with their respective sales figures. On the right is the "Fields" pane, which contains a search bar and a tree view of available fields categorized under "Key Measures", "Customers", "Dates", "Locations", "Products", "Salespeople", and "Time Intelligence". A "Filters" section at the top of the pane is also visible.

## To Calculate Cumulative Sales

This screenshot shows the Power BI desktop interface with a table visualization. In the "Measure" tab of the ribbon, a new measure named "Cumulative Sales" is being defined using the DAX formula: `FILTER(ALLSELECTED(Dates), Dates[Date] <= MAX(Dates[Date]))`. The table visualization shows cumulative sales data from December 2019 to January 2020. The "Fields" pane on the right is identical to the one in the first screenshot, showing the same field categories and filters.

## Cumulative Sales:

It will calculate cumulative sales based on Total sales DAX function

**Filter** function will filtered out the data

**Allselected** function will ignore all the applied filters

Filter condition: current date  $\leq$  max date

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

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

Filters

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

Locations

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

Locations

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 selected visual type as a donut chart. The "Fields" pane shows the selected field "county".

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 "Format visual" pane is open for the bar chart, specifically the "Data labels" section, which has a toggle switch turned on.

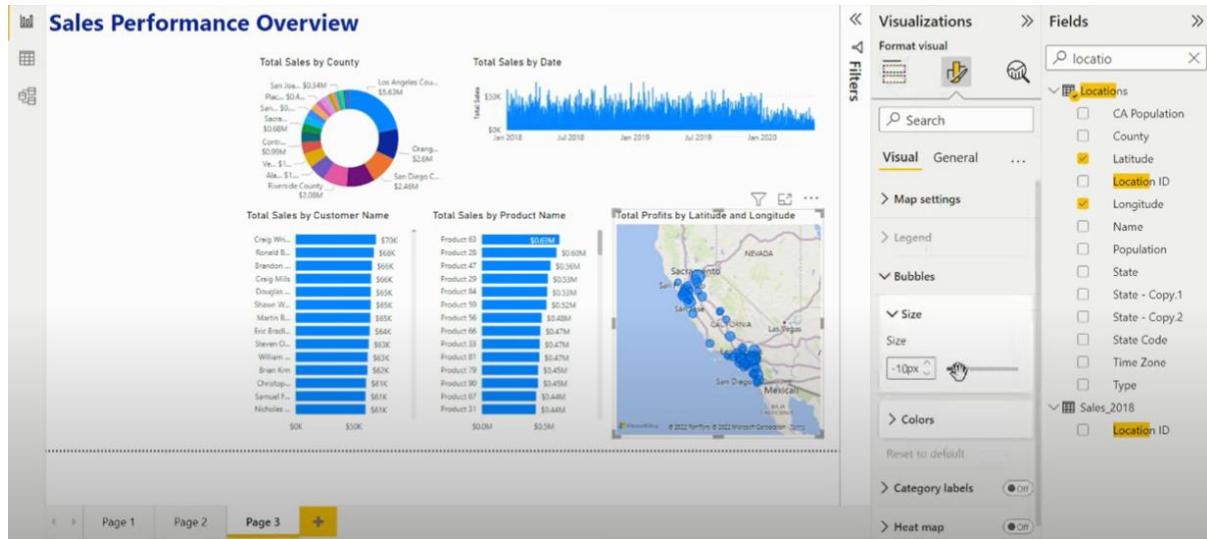
To show map as visual

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 "Visualizations" pane is open, showing the selected visual type as a map. The "Fields" pane shows the selected field "product".

Use latitude and longitude for location

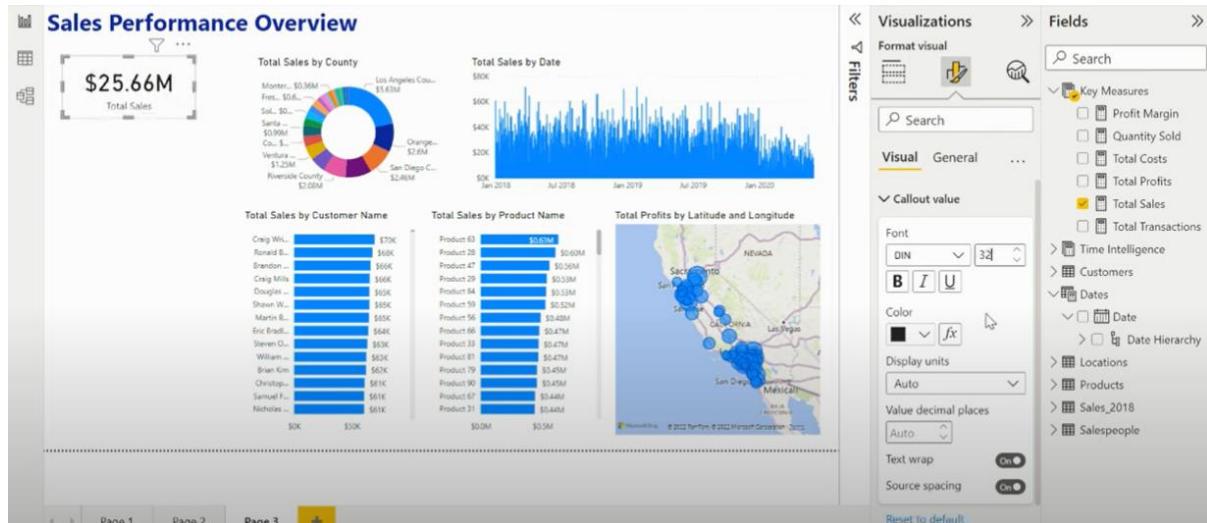
To increase the bubble size

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



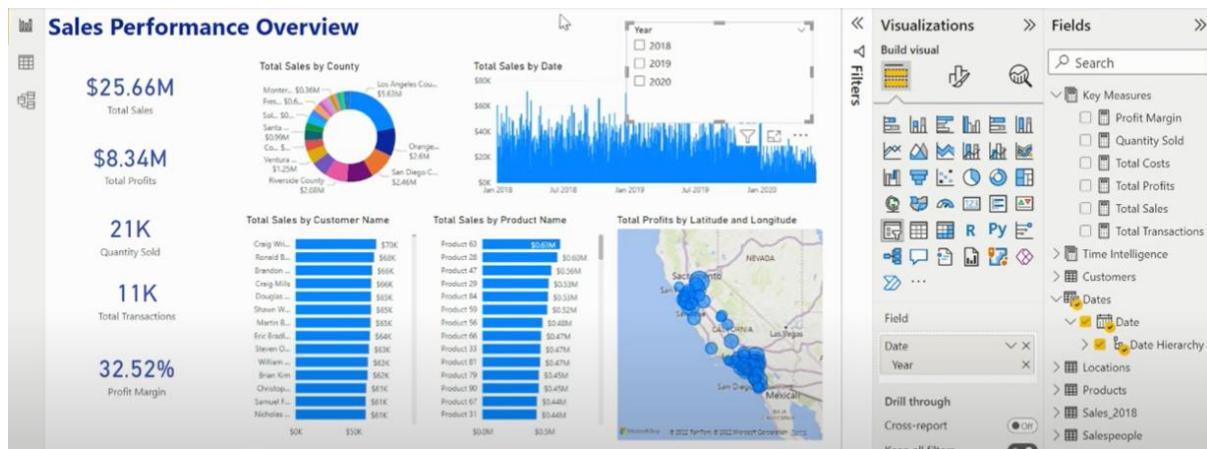
To format the cards

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

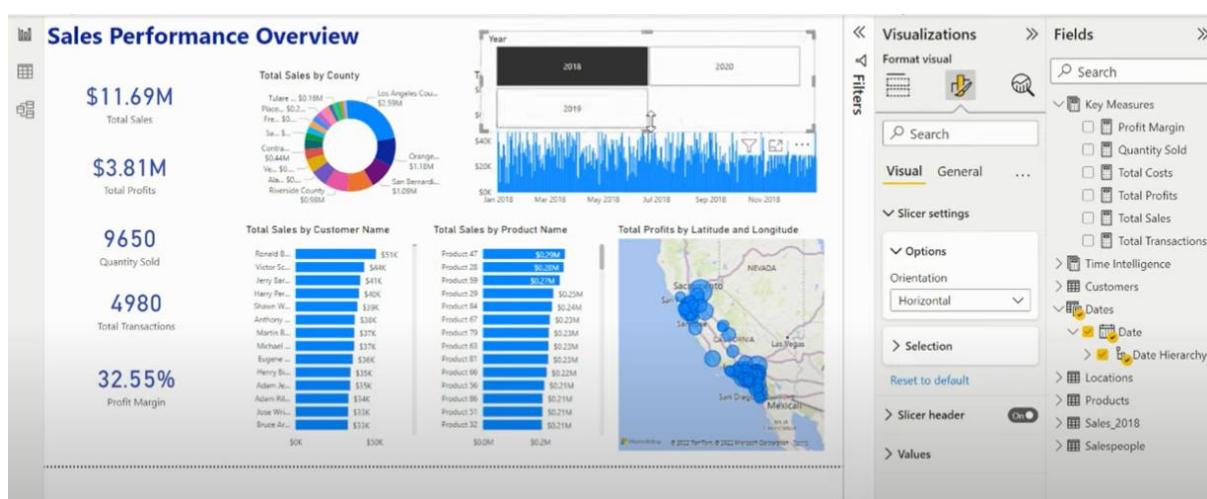
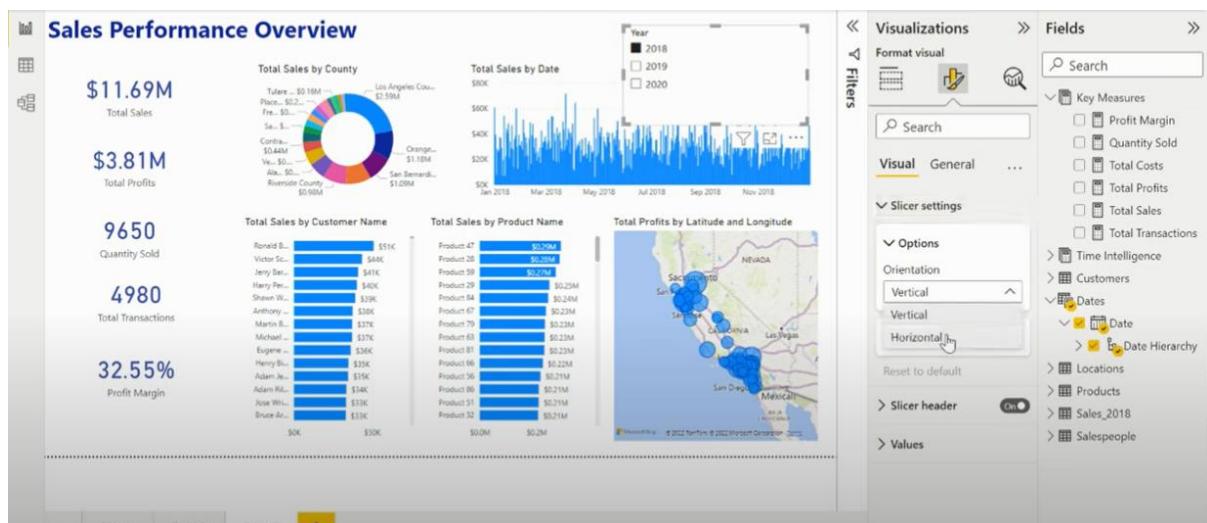


Insert slicer

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

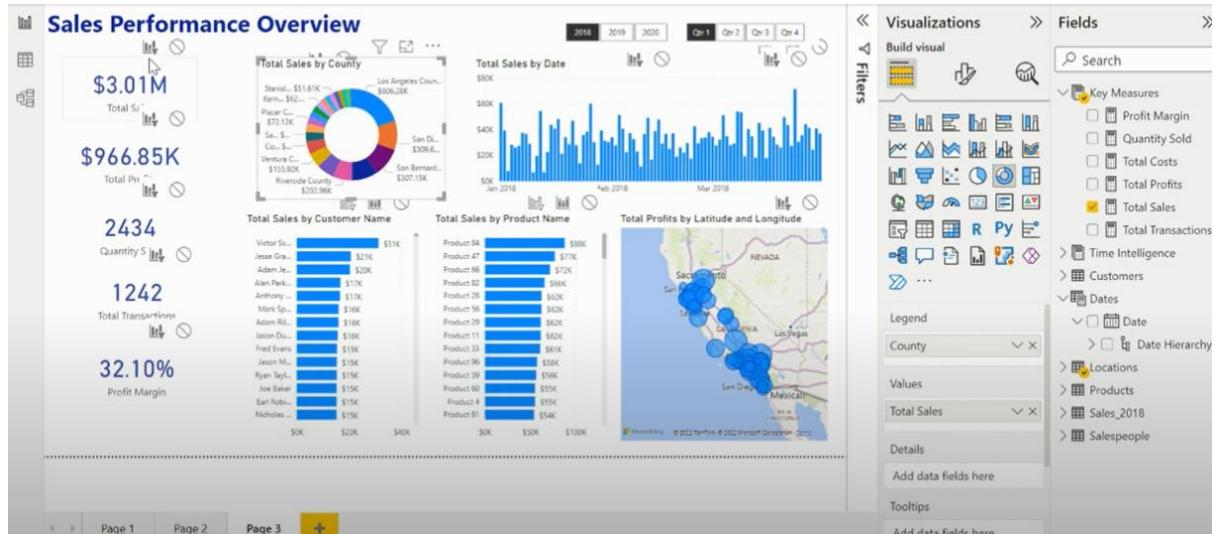


Select visual -> format visual -> visual -> slicer settings->options -> orientation -> horizontal -> adjust the slicer -> turn off slicer header



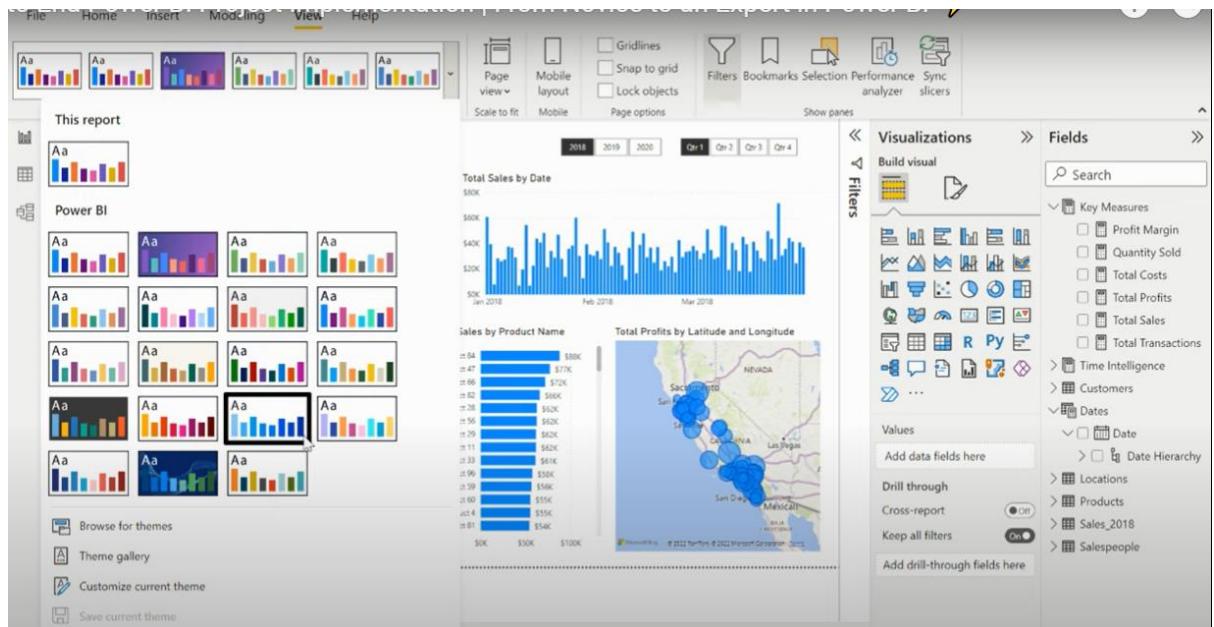
## To format/highlight data as per selection in primary visual

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



## To change theme

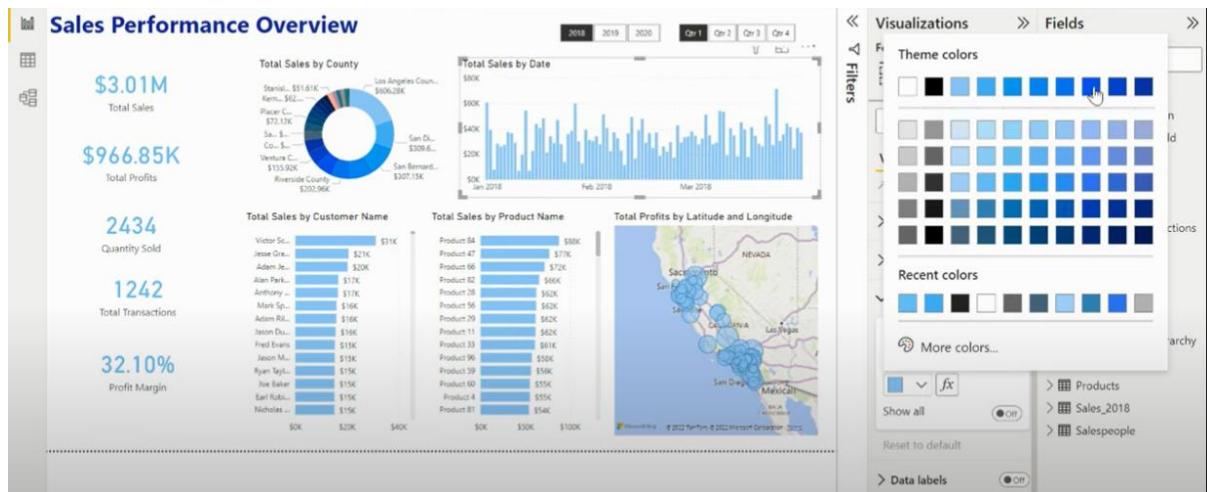
View -> themes group -> select theme



## To change colour of visuals

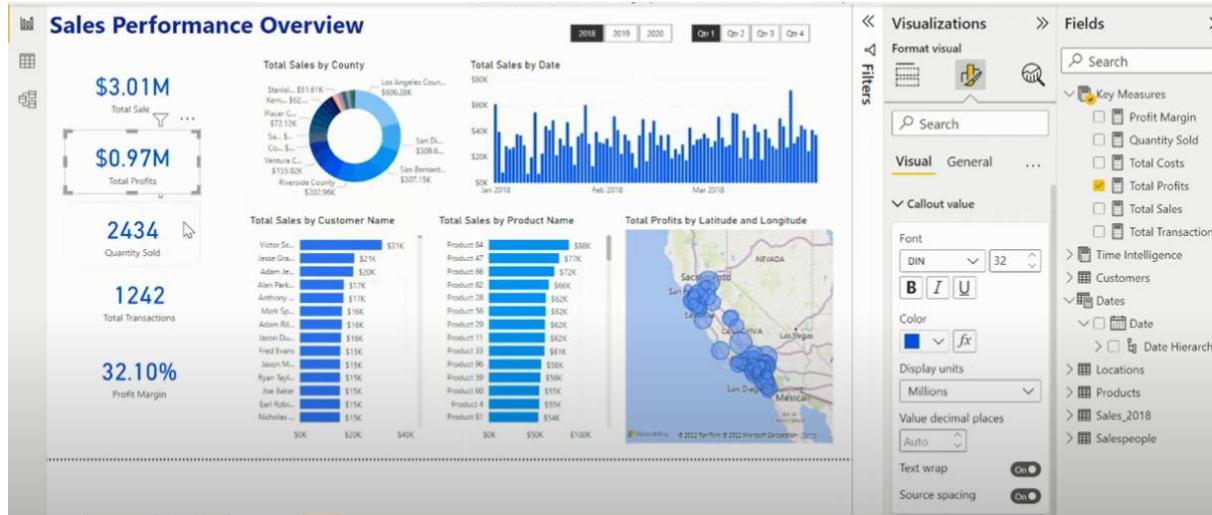
Select visual -> format visual -> visual -> columns -> color-> change colour

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



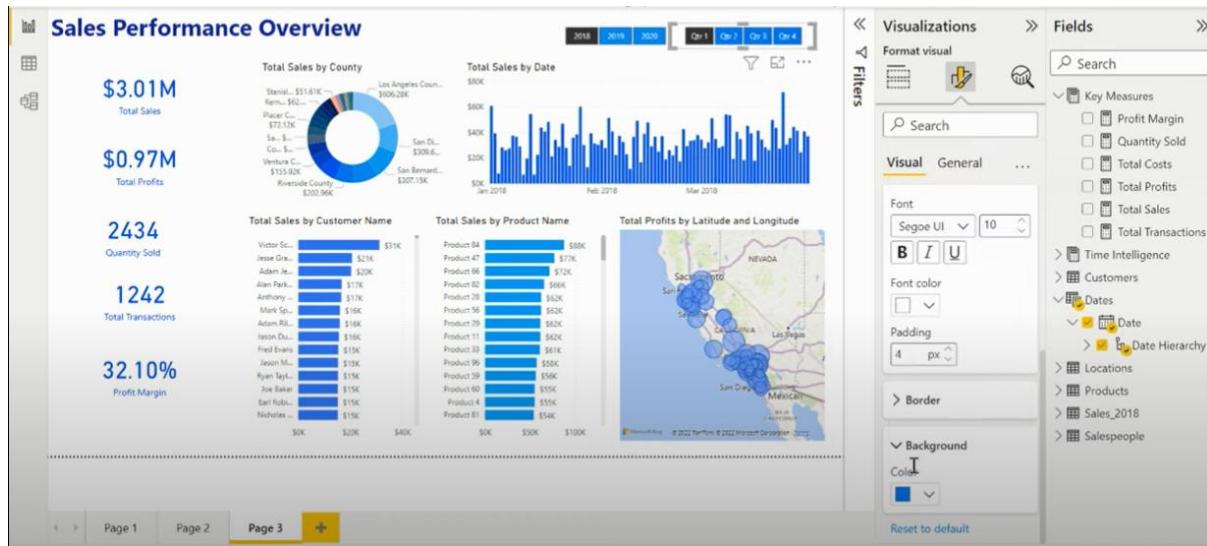
## To change units

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



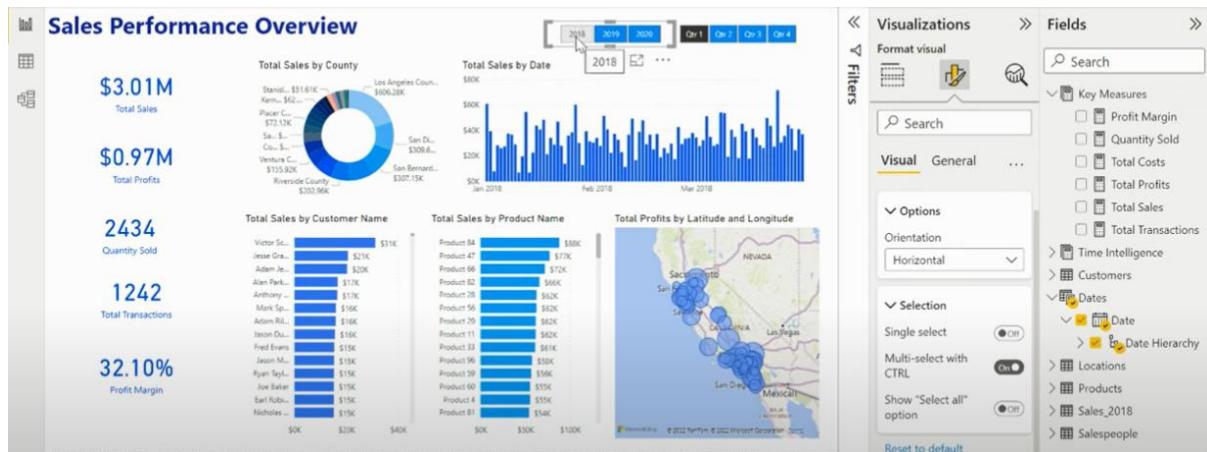
## To change slicer colour

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 the Power BI Advanced Editor showing the "Bank holidays" query being transformed.

The "Split Column by Delimiter" dialog is open, set to split at "Each occurrence of the delimiter".

The resulting table shows the split columns:

Date	Day	Occasion
01-10-2002	sunday	Mahatma Gandhi Jayanthi
01-10-2003	Monday	Durgashtami
01-10-2005	Wednesday	Vijayadasami
01-10-2009	Sunday	Eid Miladun Nabi

Screenshot of the Power BI Advanced Editor showing the "Bank holidays" query after transformation.

The table now has three columns: Date.1, Date.2, and Date.3, which correspond to the split columns from the previous step.

The "Query Settings" pane shows the "Changed Type" step applied to the "Date" column.

## Splitting the incorrect date column

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

The screenshot shows the Power BI desktop interface with the "Split Column by Number of Characters" dialog box open. The "Number of characters" input field contains "1". The "Split" section includes options: "Once, as far left as possible" (radio button selected), "Once, as far right as possible", and "Repeatedly". Below the dialog is a preview of the table showing the split results for rows 8 and 19. The "APPLIED STEPS" pane on the right shows the step "Split Column by Delimiter" with "Changed Type1" applied.

The screenshot shows the Power BI desktop interface with the table now having five columns: Date.2, Date.3.1, Day, weekday, and Ocassion. The "APPLIED STEPS" pane on the right shows the steps "Split Column by Delimiter", "Changed Type1", "Removed Columns", "Split Column by Position", and "Changed Type2". The "Renamed Columns" step is currently selected.

## 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. A table named 'Table.ReorderColumns(a"Added Custom","year", "month", "day", "weekday", "Ocassion")' is displayed. The 'year' column is currently set to 'Decimal Number' but is being changed to 'Whole Number'. The 'Properties' pane on the right shows the column name is 'Bank holidays'. The 'Applied Steps' pane lists several steps taken to transform the data, including 'Reordered Columns'.

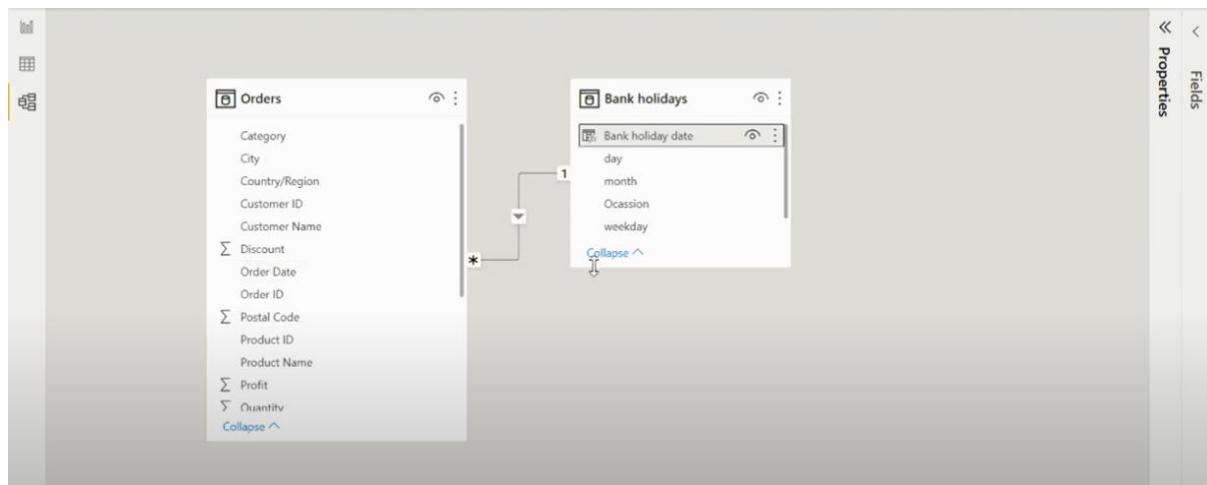
Select day column -> go to column tools -> properties section -> change summarization to don't summarize (same for month and year as well)

The screenshot shows the Power BI Desktop interface with the 'Column Tools' ribbon tab selected. In the 'Formatting' group, the 'Format' dropdown is set to 'Whole number'. 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 its columns: day, month, Ocation, weekday, and year.

## 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 a new column 'date' added under 'Visualizations'.

## Specify relationship between tables



In weekday function, 1<sup>st</sup> day of the week start from monday

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

This screenshot shows the 'Column tools' ribbon tab selected in Power BI. In the formula bar, a calculated column named 'is bank holiday' is being defined with the formula: `IF(ISBLANK(RELATED('Bank holidays'[Bank holiday date])),FALSE(),TRUE())`. The formula bar also includes a note: 'Checks whether a condition is met, and returns one value if TRUE, and another value if FALSE.' To the right, the 'Fields' pane is open, showing the 'Bank holidays' and 'Orders' tables and their respective columns.

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
06 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 documentation for the `NETWORKDAYS` function. On the left, there's a sidebar with a search bar and a list of related functions like `NOW`, `QUARTER`, etc. The main content area has a heading "Syntax" with a code snippet: `NETWORKDAYS(<start_date>, <end_date>[, <weekend>, <holidays>])`. Below it is a "Parameters" table:

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 documentation page with expanded details for the `weekend` parameter. It lists weekend numbers from 1 to 17, each corresponding to a specific day 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

This screenshot shows the "Return Value" section of the documentation, stating that the function returns "An integer number of whole workdays."

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

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

On the right, the "Fields" pane shows a visualization tree with nodes for `2018 Sales`, `Customers`, `Locations`, and `Orders`. Under `Orders`, there are several columns listed: `business_days`, `Category`, `City`, `Country/Region`, and `Cust_name`.

The screenshot shows the Power BI Data Editor interface. On the left, there is a code editor window containing 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')
```

On the right, there is a Fields pane with a search bar and a tree view of fields. The 'Orders' node is expanded, showing various fields including 'business\_days'. The 'business\_days' field is selected.

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 filter is applied to the Order Date column, showing "05 January 2018". The table data is:

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

On the right, there is a Fields pane with a search bar and a tree view of fields. The 'Orders' node is expanded, showing various fields including 'business\_days'. The 'business\_days' field is selected.

## Calendar and CalendarAuto in Power BI

The screenshot shows the Power BI Data Model ribbon with the 'Fields' tab selected. On the left, there is a 'Column' dropdown menu. The main area displays a list of fields from the 'Orders' table, including Category, City, Country/Region, Customer Name, Discount, Order ID, Postal Code, Product ID, and Product Name. A search bar at the top right allows filtering of the field list.

The screenshot shows the Power BI Data View interface. On the left, a table named 'Calendar' is displayed with columns 'Date' and 'Day'. The 'Date' column lists dates from September 1, 2022, to September 23, 2022. The 'Day' column contains the corresponding day numbers (1 through 23). On the right, a 'Fields' pane is open, showing a list of available fields categorized under 'Orders'. The listed fields include 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, and Sales.

Date	Day
01-09-2022 00:00:00	1
02-09-2022 00:00:00	2
03-09-2022 00:00:00	3
04-09-2022 00:00:00	4
05-09-2022 00:00:00	5
06-09-2022 00:00:00	6
07-09-2022 00:00:00	7
08-09-2022 00:00:00	8
09-09-2022 00:00:00	9
10-09-2022 00:00:00	10
11-09-2022 00:00:00	11
12-09-2022 00:00:00	12
13-09-2022 00:00:00	13
14-09-2022 00:00:00	14
15-09-2022 00:00:00	15
16-09-2022 00:00:00	16
17-09-2022 00:00:00	17
18-09-2022 00:00:00	18
19-09-2022 00:00:00	19
20-09-2022 00:00:00	20
21-09-2022 00:00:00	21
22-09-2022 00:00:00	22
23-09-2022 00:00:00	23

1 Calendar = CALENDAR(		Fields
2	TODAY()-365,	
3	TODAY()+365)	
Date		
03-09-2021 00:00:00		Category
04-09-2021 00:00:00		City
05-09-2021 00:00:00		Country/Region
06-09-2021 00:00:00		Customer ID
07-09-2021 00:00:00		Customer Name
08-09-2021 00:00:00		Σ Discount
09-09-2021 00:00:00		> Order Date
10-09-2021 00:00:00		Order ID
11-09-2021 00:00:00		Σ Postal Code
12-09-2021 00:00:00		Product ID
13-09-2021 00:00:00		Product Name
14-09-2021 00:00:00		Σ Profit
15-09-2021 00:00:00		Profit ratio
16-09-2021 00:00:00		Profit ratio new
17-09-2021 00:00:00		Σ Quantity
18-09-2021 00:00:00		Region
19-09-2021 00:00:00		
20-09-2021 00:00:00		

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

```
X ✓ 1 Calendar = CALENDAR(  
    MIN(Orders[Order Date]),  
    MAX(Orders[Order Date])  
)
```

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

On the right, there's a 'Fields' pane with a search bar and a tree view of available fields:

- Category
- City
- Country/Region
- Customer ID
- Customer Name
- Discount

A context menu is open over the 'Order Date' field, showing options like 'Name', 'Order ID', 'Postal Code', etc.

The screenshot shows the Power BI Data View interface. A calculated column named 'Calendar' is defined as follows:

```
1 Calendar = CALENDARAUTO(  
    DARAUTO([FiscalYearEndMonth])  
)
```

A tooltip for the formula is displayed, stating:

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

The 'Calendar' column contains the following data:

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

The Fields pane on the right lists various columns and measures, including 'Calendar' under the 'Orders' category.

Date	Fields
01-01-2018 00:00:00	Search
02-01-2018 00:00:00	> Calendar
03-01-2018 00:00:00	Orders
04-01-2018 00:00:00	Category
05-01-2018 00:00:00	City
06-01-2018 00:00:00	Country/Region
07-01-2018 00:00:00	Customer ID
08-01-2018 00:00:00	Customer Name
09-01-2018 00:00:00	Σ Discount
10-01-2018 00:00:00	> Order Date
11-01-2018 00:00:00	Order ID
12-01-2018 00:00:00	Postal Code
13-01-2018 00:00:00	Product ID
14-01-2018 00:00:00	Product Name
15-01-2018 00:00:00	Σ Profit
16-01-2018 00:00:00	Profit ratio
17-01-2018 00:00:00	Profit ratio measure
18-01-2018 00:00:00	Profit ratio new
19-01-2018 00:00:00	Σ Quantity
20-01-2018 00:00:00	Region
21-01-2018 00:00:00	
22-01-2018 00:00:00	

## 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 area containing the same formula. To the right is a 'Fields' pane with a search bar and a tree view of available fields, including 'Orders' and its sub-fields like 'Customer Name'. The status bar at the bottom indicates '1 row(s) found'.

This screenshot shows the same Power BI Data Editor interface as the previous one, 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 says 'Argument '3' in LOOKUPVALUE function is required.' The 'Fields' pane on the right shows the 'Cust\_name' field under the 'Orders' node is highlighted.

This screenshot shows a Power BI report page titled 'LOOKUPVALUE() IN POWER BI DAX'. It features a single card visualization with the text 'Ruben Ausman' and 'Cust\_name' below it. The 'Fields' pane on the right shows the 'Cust\_name' field is selected. The status bar at the bottom indicates '1 row(s) found'.

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 has two columns: 'Location ID' and 'Location\_name'. The data is as follows:

Location ID	Location_name
A100	Anaheim
A101	Antioch
A102	Bakersfield
A103	Berkeley
A104	Burbank
A105	Carlsbad
A106	Chula Vista
A107	Clovis
A108	Concord
A109	Corona

The 'Fields' pane on the right shows the same list of fields as the previous screenshot, with the 'Customer ID' field highlighted by a red circle. The 'Visualizations' pane is also visible.

## What is M (Mashup) Language in Power BI?

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

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

Home -> Query Group -> Advanced Editor

The screenshot shows the Power BI Advanced Editor interface. A query named "Orders" is selected, containing 24 rows of data. The columns are labeled "Column1" through "Column5". The "APPLIED STEPS" pane on the right shows a single step named "#Changed Type".

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 2<sup>nd</sup> 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"
  
```

The screenshot shows the Power BI Advanced Editor interface with the "Orders" query selected. The M code for the query is displayed in the main editor area:

```

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"
  
```

A note at the bottom of the editor states "No syntax errors have been detected." The "APPLIED STEPS" pane on the right shows a single step named "#Changed Type".

## How to delete custom rows in Power BI?

Home -> Query Group -> Advanced Editor

Advanced Editor

### Orders

```

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

```

No syntax errors have been detected.

Done

Advanced Editor

### Orders

```

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

```

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

A Comma cannot precede a RightParen. Show error

Done Cancel

### Orders

```

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

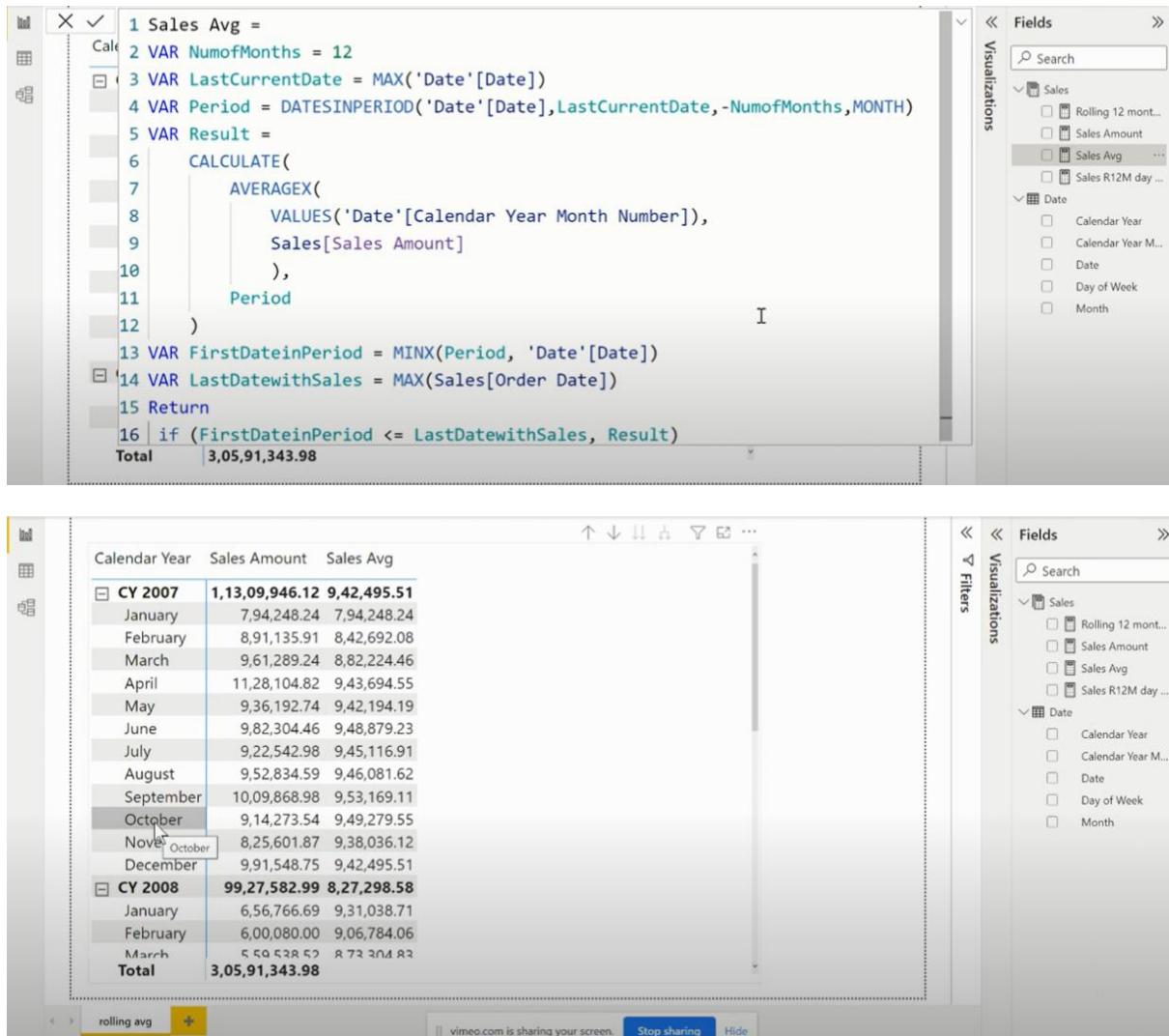
```

No syntax errors have been detected.

Done Cancel

## Rolling Average

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



The screenshot shows the Power BI Data Editor interface. On the left is the DAX editor pane, which contains the following DAX code:

```

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

```

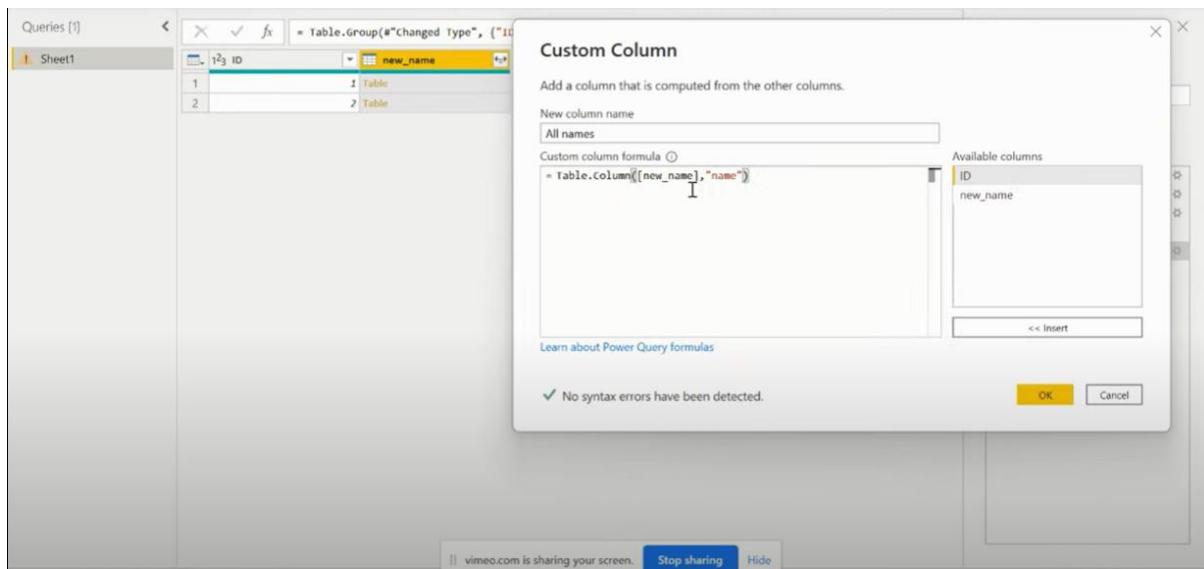
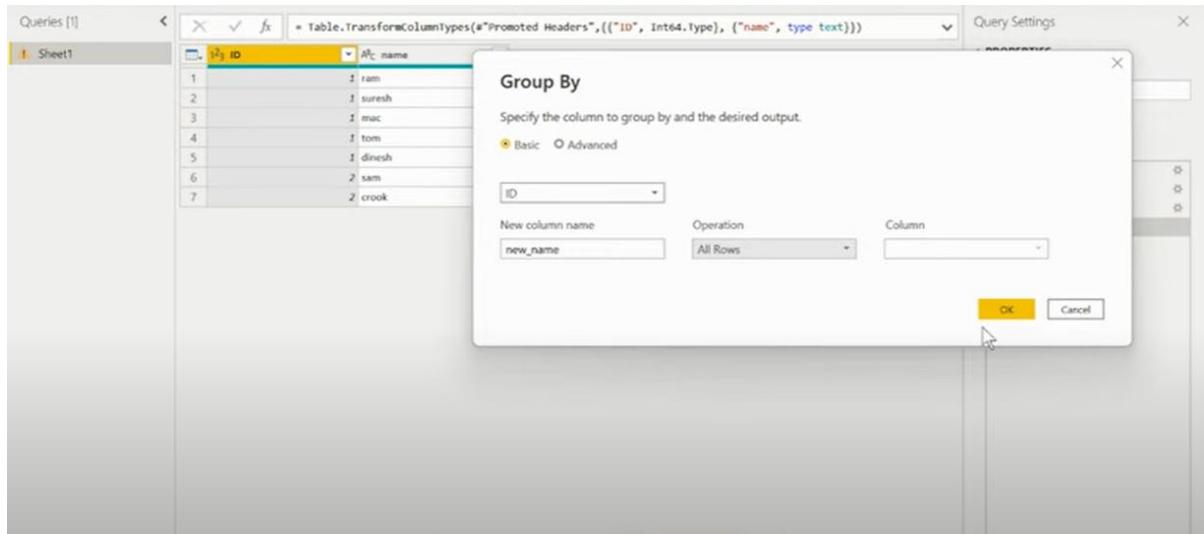
To the right of the DAX editor is the Fields pane, which lists the available fields for the Sales and Date tables. The Sales table includes measures like Rolling 12 mont..., Sales Amount, Sales Avg, and Sales R12M day ...; the Date table includes dimensions like Calendar Year, Month, and Day of Week.

Below the DAX editor is a table visualization showing monthly sales data for two years. The columns are Calendar Year, Sales Amount, and Sales Avg. The data is grouped by month, with totals for each year and a final total for the year-to-date.

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

## How to convert rows to comma separated values in Power BI

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



The screenshot shows the Power Query Editor interface. On the left, there's a 'Queries [1]' pane with 'Sheet1'. In the main area, a table has a context menu open over the 'All names' column header. The menu options are 'Expand to New Rows' and 'Extract Values...'. The 'Extract Values...' option is highlighted with a mouse cursor. To the right, there's a 'Query Settings' pane with sections for 'PROPERTIES' (Name: Sheet1) and 'APPLIED STEPS' (listing 'Source', 'Navigation', 'Promoted Headers', 'Changed Type', 'Grouped Rows', and 'Added Custom').

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

## Show/Hide Visual based on Slicer Selection in Power BI

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

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

`Int`: converts Boolean output into 1/0

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

**Show/Hide Visual based on slicer selection**

The screenshot shows a Power BI interface with a slicer on the left labeled "Category" containing "Furniture", "Office Supplies", and "Technology". To the right is a bar chart titled "Sales by Sub-Category" with categories like Phones, Chairs, Storage, Tables, Binders, Machines, Accessories, Copiers, Bookcases, Appliances, Furnishings, Paper, Supplies, Art, Envelopes, Labels, and Fasteners. The Y-axis is "Sub-Category" and the X-axis is "Sales". On the far right is a "Fields" pane listing various dimensions and measures.

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

This screenshot shows a calculated measure "Measure 2 = int(ISFILTERED(Orders[Category]))" in the query editor. To the right is a bar chart titled "Sales by Sub-Category" showing sales for various sub-categories. The Y-axis is "Sub-Category" and the X-axis is "Sales". The "Fields" pane on the right includes "Measure 2" in the list of selected fields.

This screenshot shows a bar chart titled "Sales by Sub-Category" with the same data as previous screens. The "Filters" pane on the right shows an applied filter for "Measure 2" set to "is (All)". Other filters listed are "Sales is (All)" and "Sub-Category is (All)". The "Fields" pane on the right includes "Measure 2" in the list of selected fields.