

# **MICROSOFT POWER-BI**

**SERIES 1**

## **OVERVIEW**

**- MAYURI .D.**

## INTRODUCTION

- Microsoft Power BI is a collection of apps, software services and connectors that come together to turn unrelated data into visually impressive and interactive insights.
- Power BI can work with simple data sources like Microsoft Excel and complicated ones like cloud-based or on-premises hybrid Data warehouses. Power BI has the capabilities to easily connect to your data sources, visualise and share and publish your findings with anyone and everyone.
- Power BI is simple and fast enough to connect to an Excel workbook or a local database.
- It can also be robust and enterprise-grade, ready for extensive modeling and real time analytics.
- This means it can be used in a variety of environments from a personal report and visualisation tool to the analytics and decision engine behind group projects, divisions, or entire corporations.

## PARTS OF POWER-BI

- Power BI constitutes of a Microsoft Windows desktop application called Power BI Desktop, an online **SaaS** (Software as a Service) called Power BI Service and a mobile Power BI apps that can be accessed from Windows phones and tablets, and also available on Apple iOS and Google Android devices.
- These **three elements— Desktop, the Service, and Mobile apps** - are the backbone of the Power BI system and lets users create, share and consume the actionable insights in the most effective way.

# PARTS OF POWER-BI

## 1. POWER-BI DESKTOP

It is a Windows application that provides a rich environment for data modeling, transformation, and visualization.

Key features of Power BI Desktop include the ability to import data from various sources, create data models, design visuals, write DAX calculations, and customize report layouts.

Power BI Desktop is primarily used by data analysts and report developers to build and design reports before publishing them to the Power BI Service.

# PARTS OF POWER-BI

## 2. POWER-BI SERVICE

Power BI Service, also called as Power BI Cloud or Power BI Online.

It is a cloud-based platform where reports and dashboards created in Power BI Desktop can be published, shared, and accessed by others.

With Power BI Service, users can view and interact with reports and dashboards through a web browser or the Power BI mobile app.

It offers collaboration features, including the ability to share reports, create dashboards, set up data alerts, and collaborate with colleagues in real time.

Power BI Service also provides data refresh capabilities to ensure that reports stay up to date with the latest data.

## PARTS OF POWER-BI

### **3. POWER-BI MOBILE**

Power BI Mobile is a mobile application available for iOS and Android devices. It allows users to access and interact with Power BI reports and dashboard.

Users can view, filter, and drill down into data, making it convenient for decision-makers who need access to business insights while away from their computers.

The app also supports offline access, so users can access reports even without an internet connection.

## BUILDING BLOCKS OF POWER-BI

The basic building blocks in Power BI are:

- **Visualizations**

A visualization is a representation of data in a visual format. It could be a line chart, a bar graph, a color coded map or any visual way to present the data.

- **Datasets**

A dataset is a collection of data that Power BI uses to create its visualizations.

- **Reports**

A Report is a collection of visualizations that appear together on one or more pages.

- **Dashboards**

A Power BI dashboard is a collection of visuals from a single page that you can share with others.

# **BENEFITS OF POWER-BI**

## **1. User-Friendly Interface**

Power BI features a user-friendly, drag-and-drop interface, making it easy to both technical and non-technical users.

Users can create reports and dashboards without any extensive training.

## **2. Data Integration and Transformation**

Power BI can connect to a wide range of data sources, including databases, cloud services, web sources, and more. It offers robust data transformation capabilities to clean, reshape, and model data for analysis.

## **3. Advanced Data Modeling**

Users can create sophisticated data models using DAX, enabling complex calculations, aggregations, and relationships. The data model supports the creation of hierarchies, measures, and calculated columns.



## **BENEFITS OF POWER-BI**

### **4. Interactive Visualizations**

Power BI provides a huge library of customizable visualizations, including bar charts, line charts, maps, etc.

Visualizations can be filtered, cross-highlighted, and drilled down to explore data interactively.

### **5. Natural Language Queries**

Users can ask questions using natural language queries, and Power BI's AI capabilities generate visualizations and insights based on the questions.

### **6. Real-Time Data**

Power BI supports real-time data, allowing organizations to monitor and react to data changes in real-time.

## **BENEFITS OF POWER-BI**

### **7. Data Sharing and Collaboration:**

Reports and dashboards created in Power BI can be shared securely with others. Collaboration features include commenting, annotation, and the ability to create and share dashboards with stakeholders.

### **8. Security and Compliance:**

Power BI offers robust security and access control features, including integration with Azure Active Directory for identity management. It complies with various industry standards and regulations, making it suitable for organizations with strict data governance requirements.

### **9. Scalability:**

Power BI is highly scalable, capable of handling large datasets and serving the needs of small to large organizations.

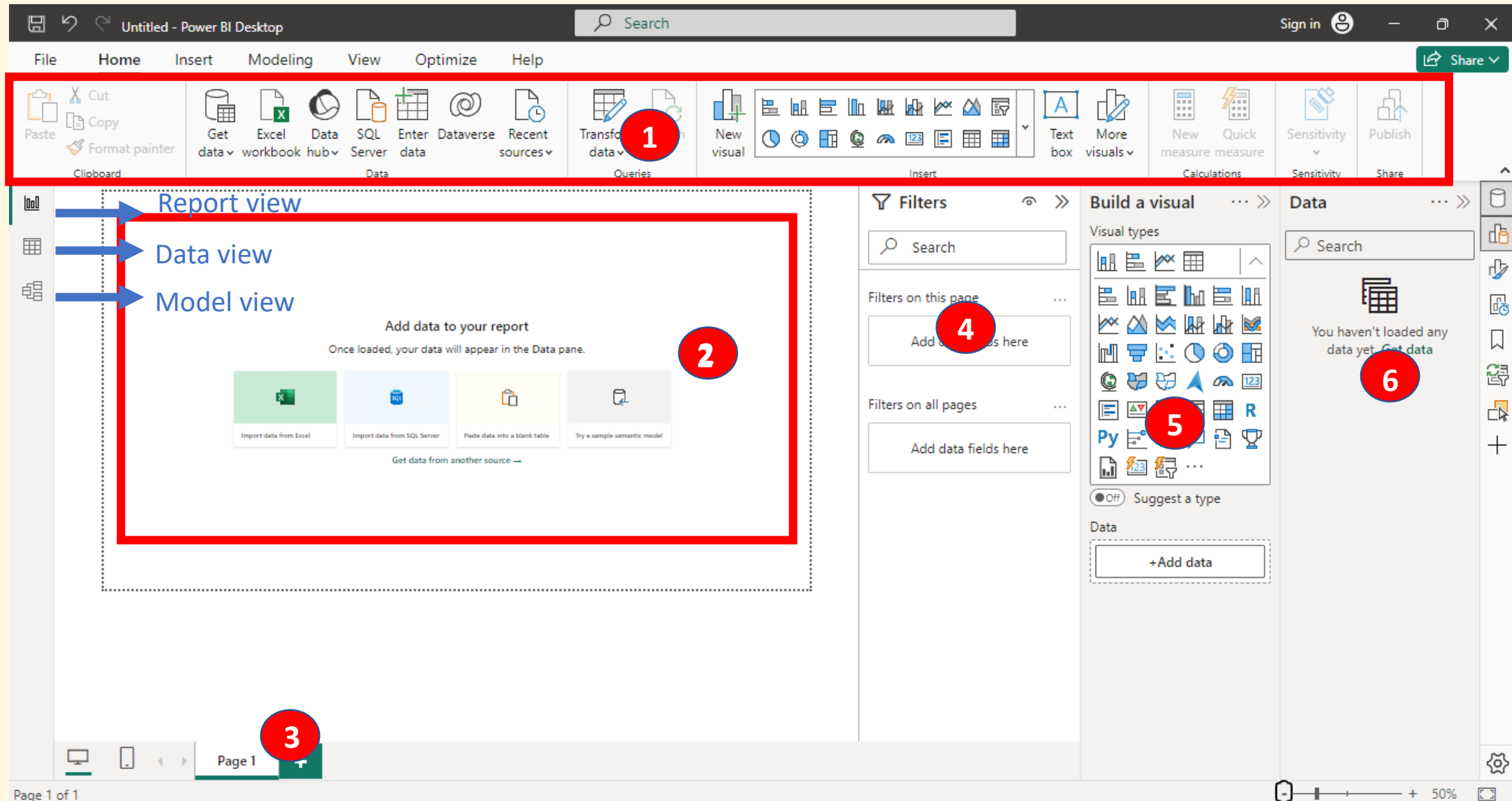
## BENEFITS OF POWER-BI

### **10. Cost-Efficiency**

Power BI offers both free and paid licensing options, making it accessible to organizations of all sizes.

Power BI Pro and Premium licenses provide additional features for advanced use cases.

# POWER-BI DESKTOP VIEW



## POWER-BI DESKTOP VIEW

- 1. Ribbon** - Displays common tasks that are associated with reports and visualizations.
- 2. Report view, or canvas** - Where visualizations are created and arranged. You can switch between Report, Data, and Model views by selecting the icons in the left column.
- 3. Pages tab** - Located along the bottom of the page, this area is where you would select or add a report page.
- 4. Filters Pane** – Where you drag fields in filters as per requirement
- 5. Visualizations pane** - Where you can change visualizations, customize colors or axes, apply filters, drag fields, and more.
- 6. Data pane** - Where dataset tables with respected columns, measures and calculated columns are displayed.

# **MICROSOFT POWER-BI**

**SERIES 2**

## **IMPORTING DATASETS**

**- MAYURI .D.**

A thick yellow bar runs diagonally from the top-left corner towards the bottom-left corner of the slide.

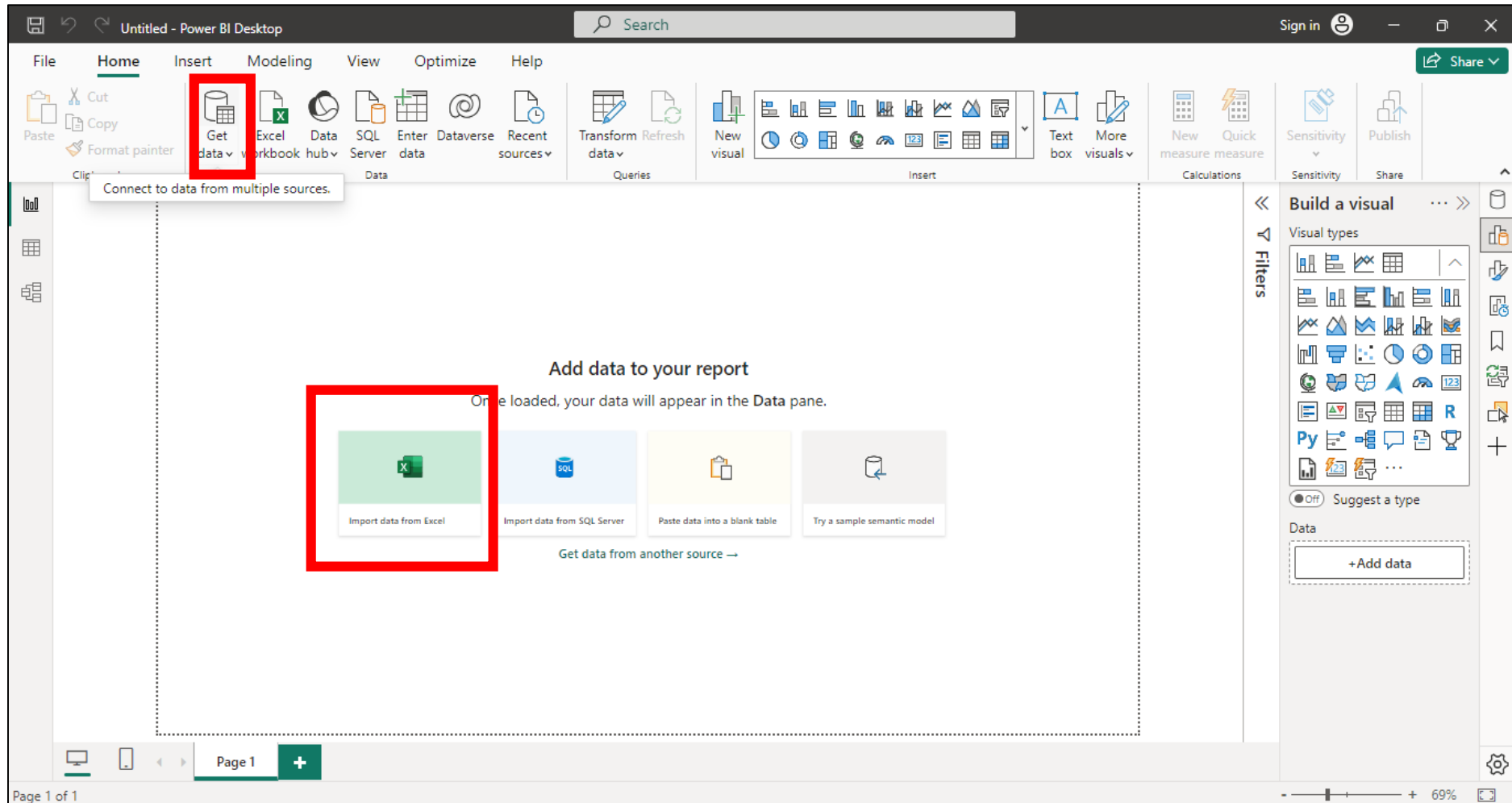
Power BI can connect to a whole range of data sources from Excel sheets and local databases to several Cloud services.

Currently, over 60 different cloud services have specific connectors to help you connect with generic sources through XML, CSV, text, and ODBC.

A yellow semi-circle is located in the bottom-right corner of the slide.

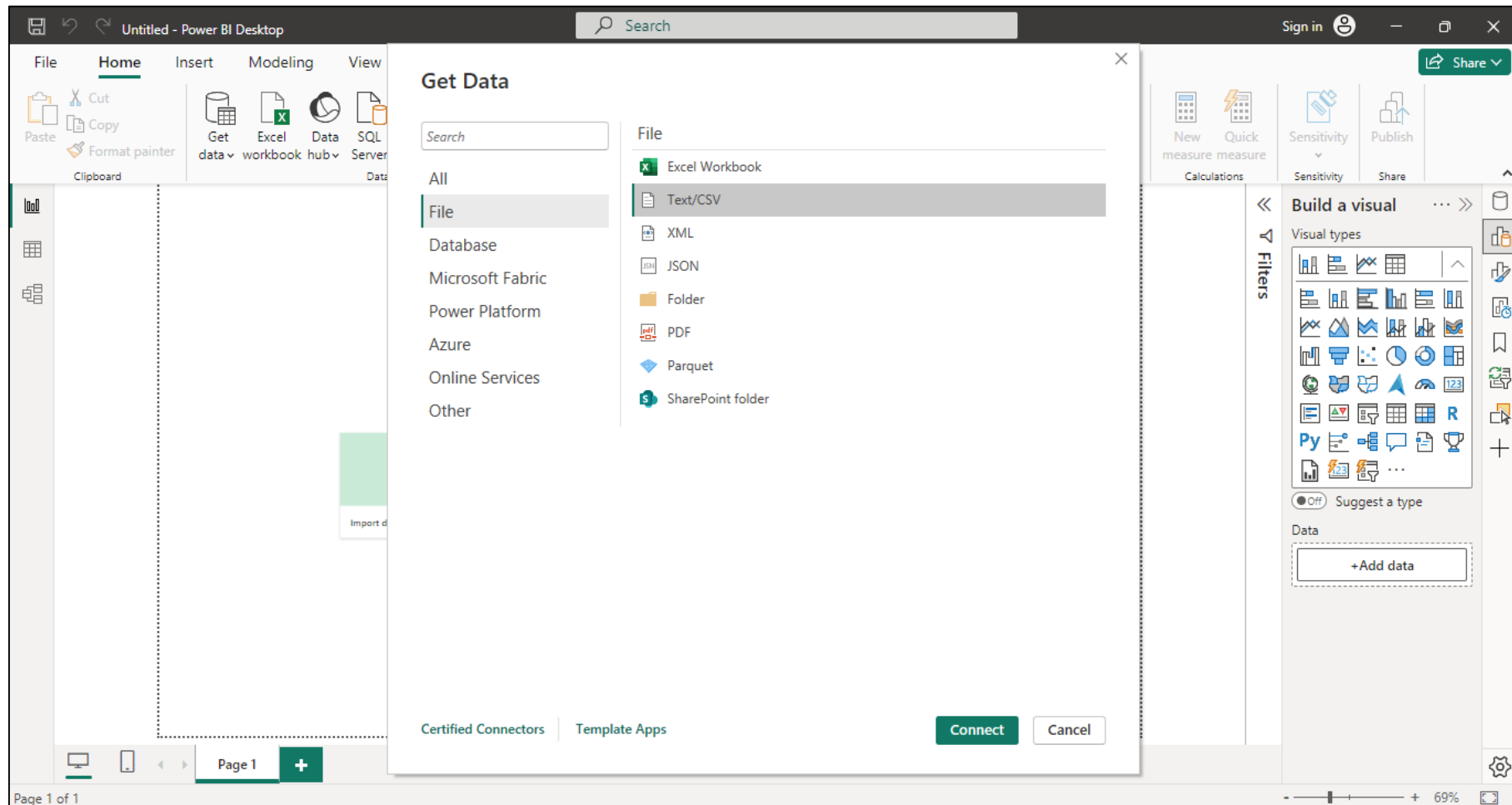
# IMPORTING DATA IN POWER-BI – CSV TO POWER-BI

As your first step is to import the dataset/file into Power BI, click on the “Get Data” icon on the ribbon of Home tab or on canvas “Import data from excel”.

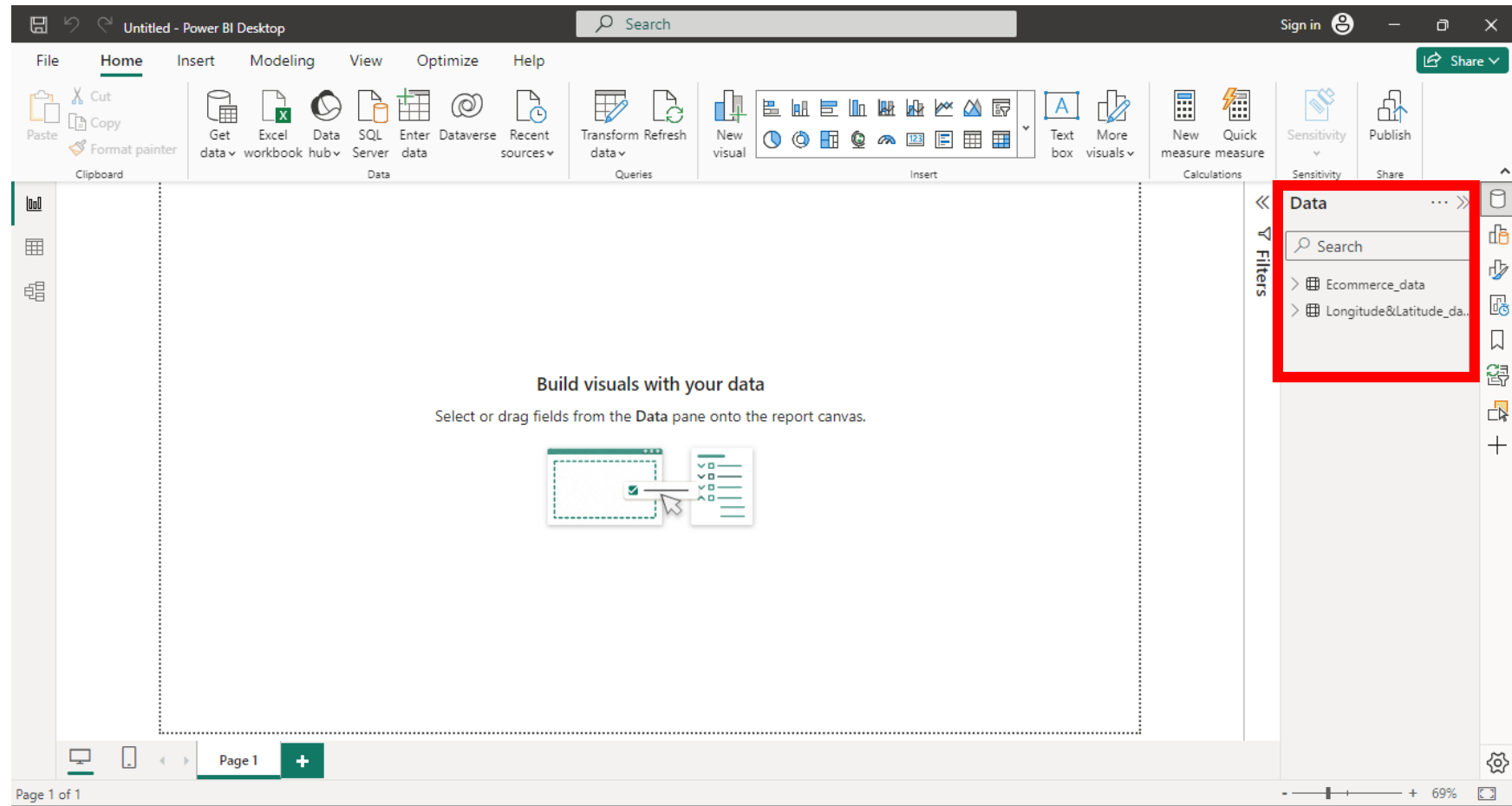




Once you select it, go ahead and select the CSV option under the file subheading. Then browse the file and select the necessary CSV file. Press on Connect to have a quick preview of the file. Once we click on Load, Power BI will successfully import the file.



When you click on Load, file gets loaded in Power-BI



When you click on Transform Data, file gets loaded in Power Query Editor for data modelling purpose.

The screenshot displays the Microsoft Power Query Editor interface. The 'Transform' tab is selected in the ribbon. The main area shows a data table with columns: state, latitude, longitude, and name. The 'state' column is highlighted. The right sidebar shows 'Query Settings' with 'Properties' and 'Applied Steps' sections. The 'Applied Steps' section lists 'Source', 'Promoted Headers', and 'Changed Type'.

Queries [2]

- Ecommerce\_data
- Longitude&Latitude\_data

Table.TransformColumnTypes(#"Promoted Headers",{{"state", type text}, {"latitude", type

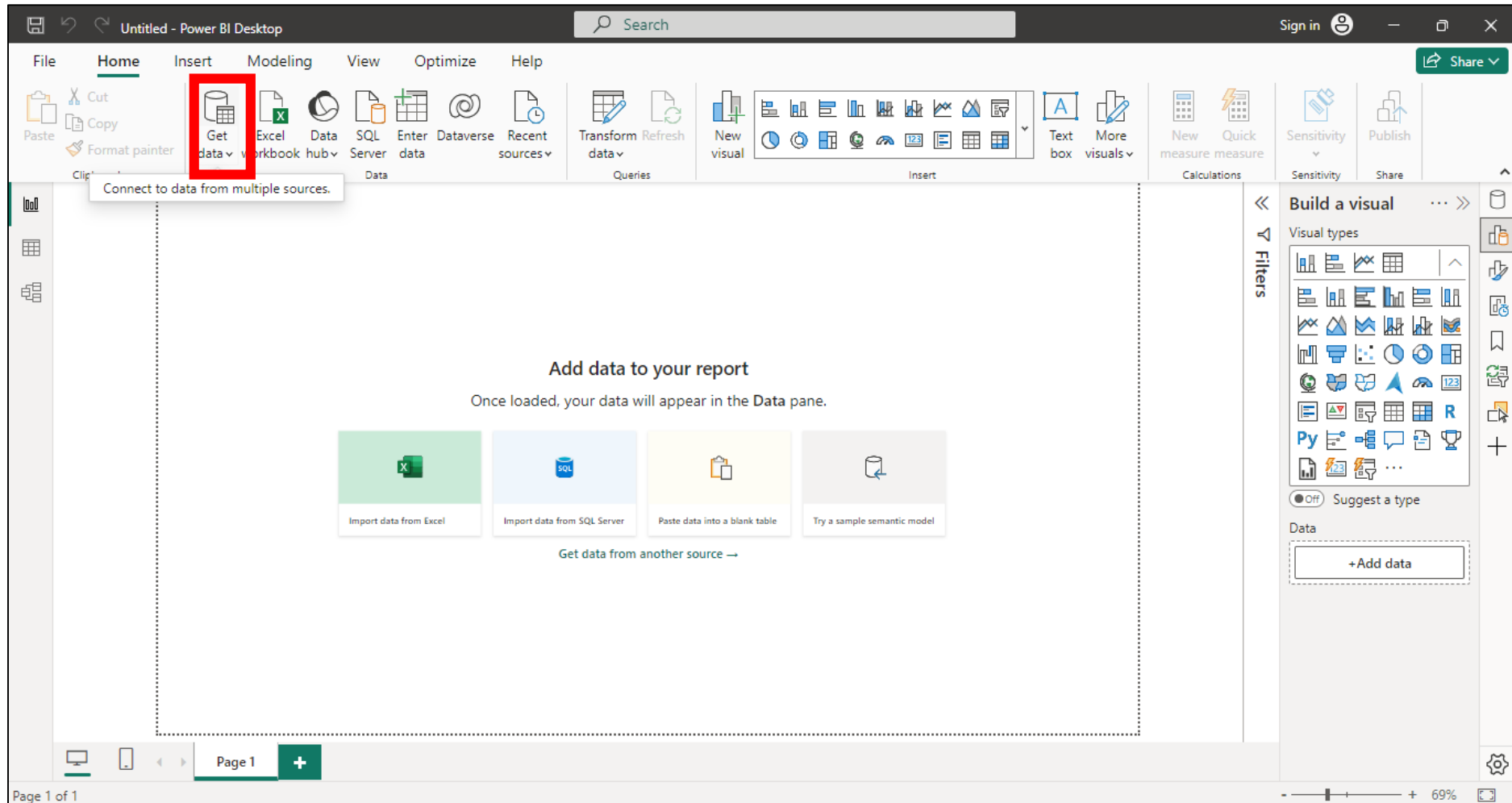
	state	latitude	longitude	name
1	AK	63.588753	-154.493062	Alaska
2	AL	32.318231	-86.902298	Alabama
3	AR	35.20105	-91.831833	Arkansas
4	AZ	34.048928	-111.093731	Arizona
5	CA	36.778261	-119.417932	California
6	CO	39.550051	-105.782067	Colorado
7	CT	41.603221	-73.087749	Connecticut
8	DC	38.905985	-77.033418	District of Columbia
9	DE	38.910832	-75.52767	Delaware
10	FL	27.664827	-81.515754	Florida
11	GA	32.157435	-82.907123	Georgia
12	HI	19.898682	-155.665857	Hawaii
13	IA	41.878003	-93.097702	Iowa
14	ID	44.068202	-114.742041	Idaho
15	IL	40.633125	-89.398528	Illinois
16	IN	40.551217	-85.602364	Indiana
17	KS	39.011902	-98.484246	Kansas

4 COLUMNS, 52 ROWS Column profiling based on top 1000 rows

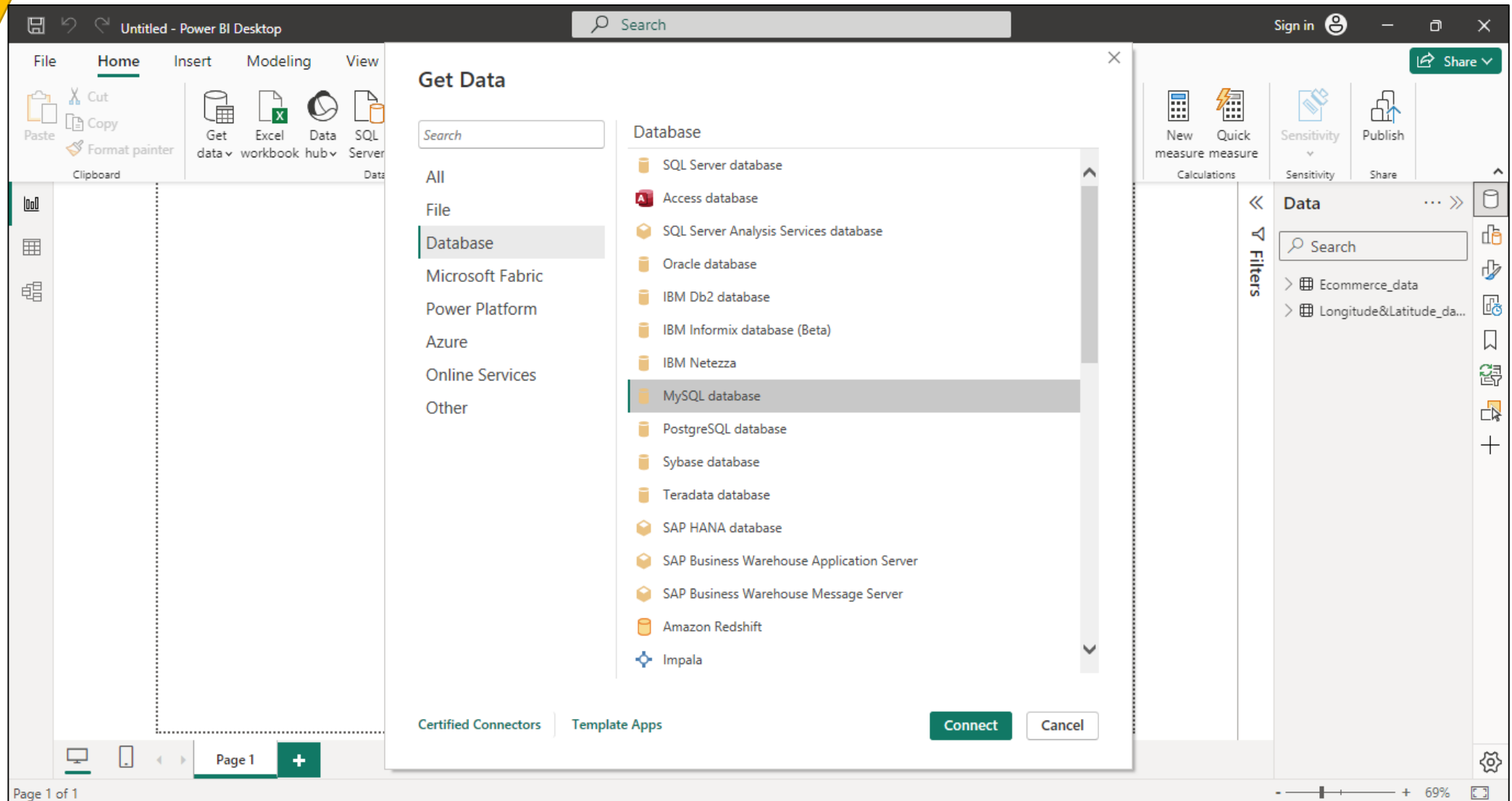
PREVIEW DOWNLOADED AT 08:22 PM

# IMPORTING DATA IN POWER-BI – MYSQL TO POWER-BI

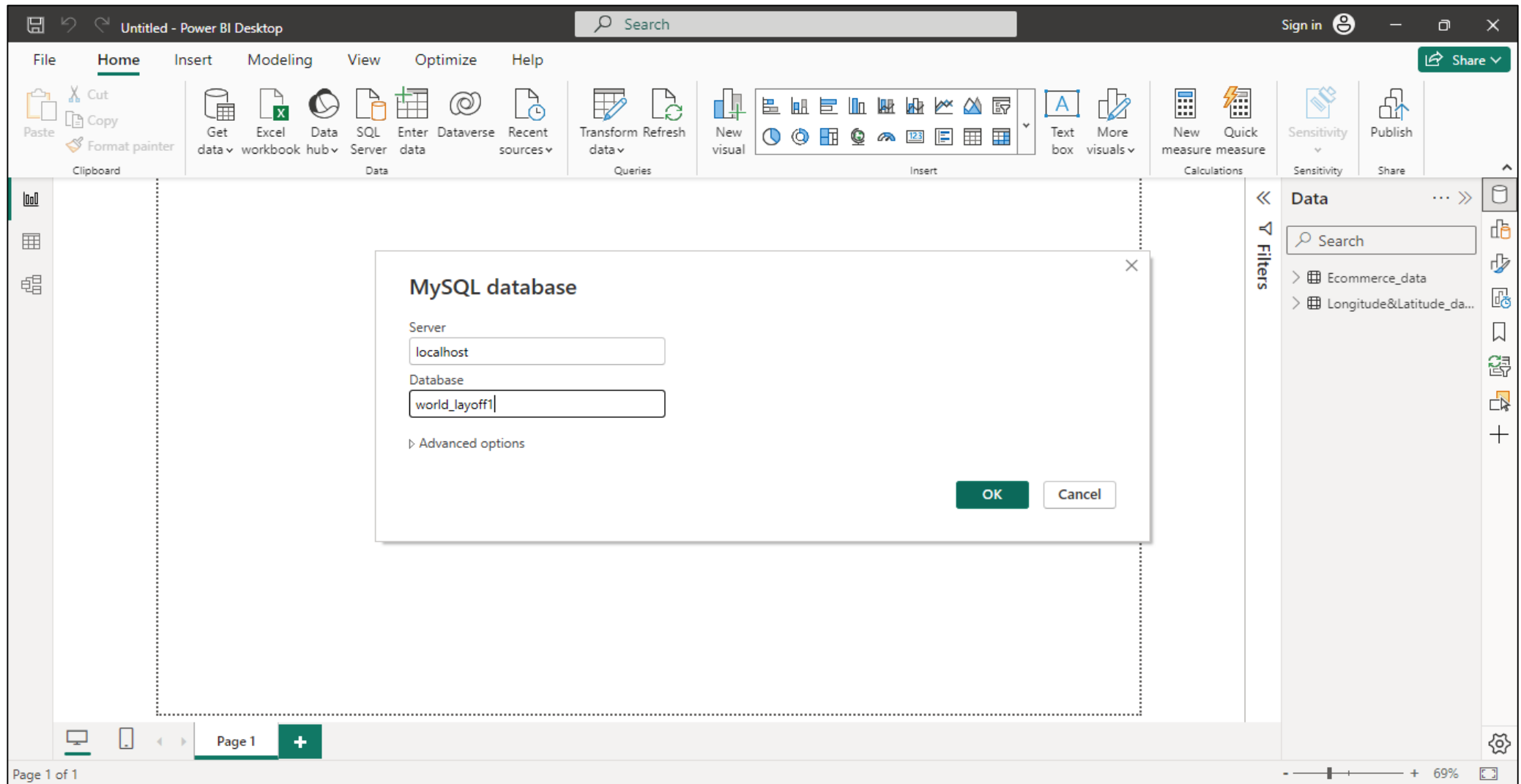
As your first step is to import the dataset/file into Power BI, click on the “Get Data” icon on the ribbon of Home tab.



Once you select it, go ahead and select the MySQL Database option under the Database subheading. Press on Connect to connect with MySQL .



Provide database credential. As localhost and the name of database/schema and click OK



Select the necessary data from list, check its preview and click Load to load the dataset in Power-BI

The screenshot shows the Power BI Desktop interface with the 'Navigator' window open. The 'Navigator' window displays a list of datasets under the folder 'localhost: world\_layoff1 [3]'. The dataset 'world\_layoff1.layoffs\_staging2' is selected, indicated by a red box and a checkmark. The preview of this dataset is shown on the right, displaying a table with columns: company, location, industry, total\_laid\_off, percentage\_laid\_off, and date. The table contains 26 rows of data. At the bottom of the Navigator window, there are buttons for 'Load', 'Transform Data', and 'Cancel'. The 'Load' button is highlighted in green.

**Navigator**

Display Options

- localhost: world\_layoff1 [3]
  - ☐ world\_layoff1.layoffs
  - ☐ world\_layoff1.layoffs\_staging
  - ☒ world\_layoff1.layoffs\_staging2

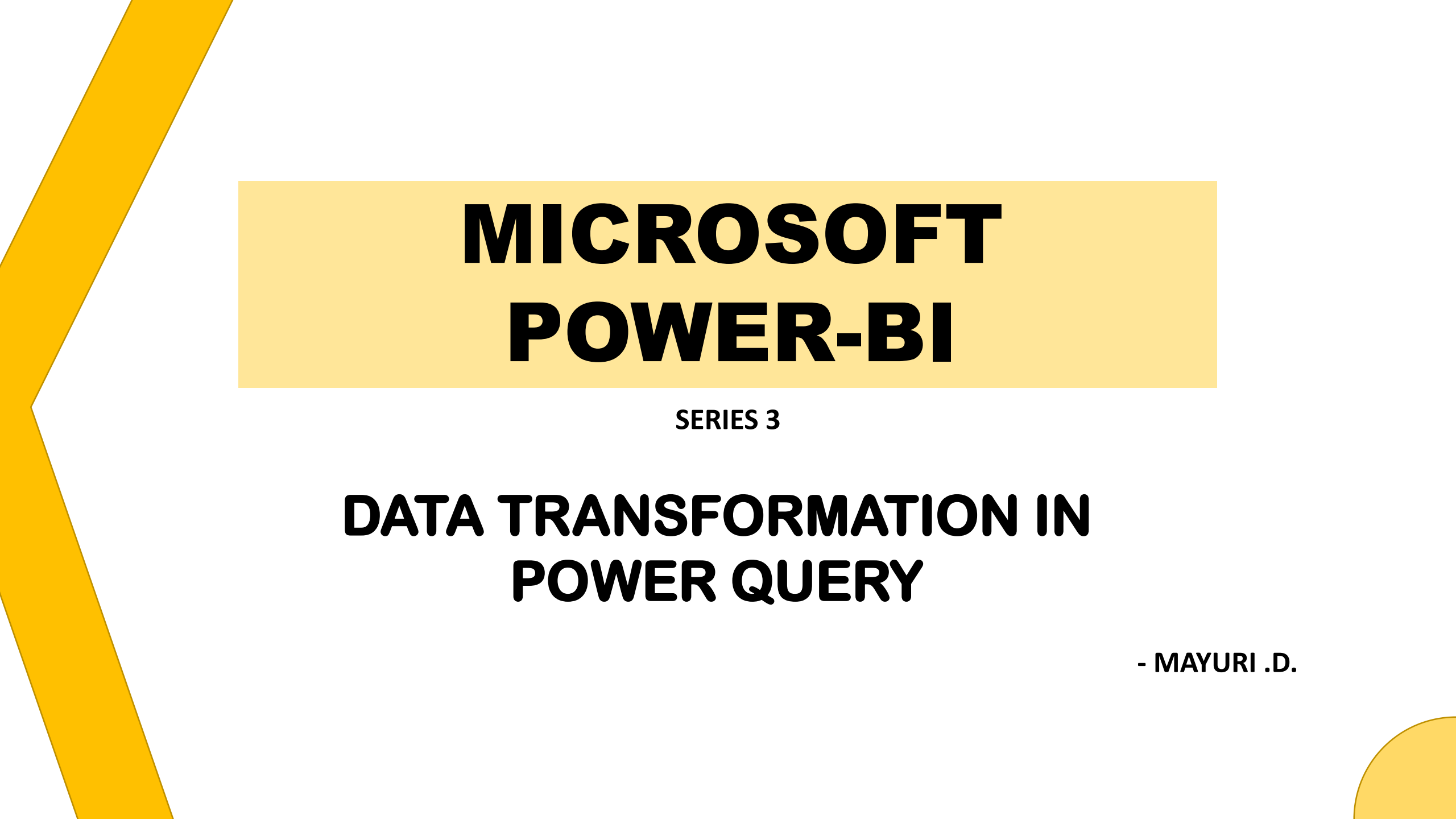
**world\_layoff1.layoffs\_staging2**

company	location	industry	total_laid_off	percentage_laid_off	date
Included Health	SF Bay Area	Healthcare	null	0.06	25-
&Open	Dublin	Marketing	9	0.09	17-
#Paid	Toronto	Marketing	19	0.17	27-
100 Thieves	Los Angeles	Consumer	12	null	13-
10X Genomics	SF Bay Area	Healthcare	100	0.08	04-
1stdibs	New York City	Retail	70	0.17	02-
2TM	Sao Paulo	Crypto	90	0.12	01-
2TM	Sao Paulo	Crypto	100	0.15	01-
2U	Washington D.C.	Education	null	0.2	28-
54gene	Washington D.C.	Healthcare	95	0.3	29-
5B Solar	Sydney	Energy	null	0.25	03-
6sense	SF Bay Area	Sales	150	0.1	12-
80 Acres Farms	Cincinnati	Food	null	0.1	18-
8x8	SF Bay Area	Support	155	0.07	18-
8x8	SF Bay Area	Support	200	0.09	04-
98point6	Seattle	Healthcare	null	0.1	21-
99	Sao Paulo	Transportation	75	0.02	20-
Abra	SF Bay Area	Crypto	12	0.05	30-
Absci	Vancouver	Healthcare	40	null	09-
Acast	Stockholm	Media	70	0.15	15-
Acko	Mumbai	Finance	45	0.09	01-
Acorns	Portland	Finance	50	null	26-
Actifio	Boston	Data	54	null	16-

Page 1 of 1

Select Related Tables

Load Transform Data Cancel



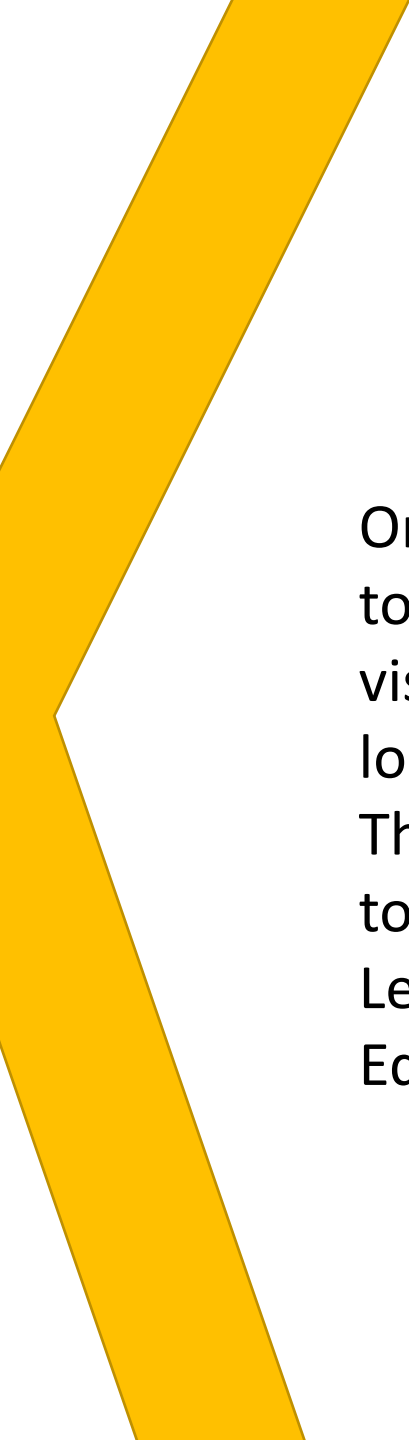
# **MICROSOFT POWER-BI**

**SERIES 3**

## **DATA TRANSFORMATION IN POWER QUERY**

**- MAYURI .D.**





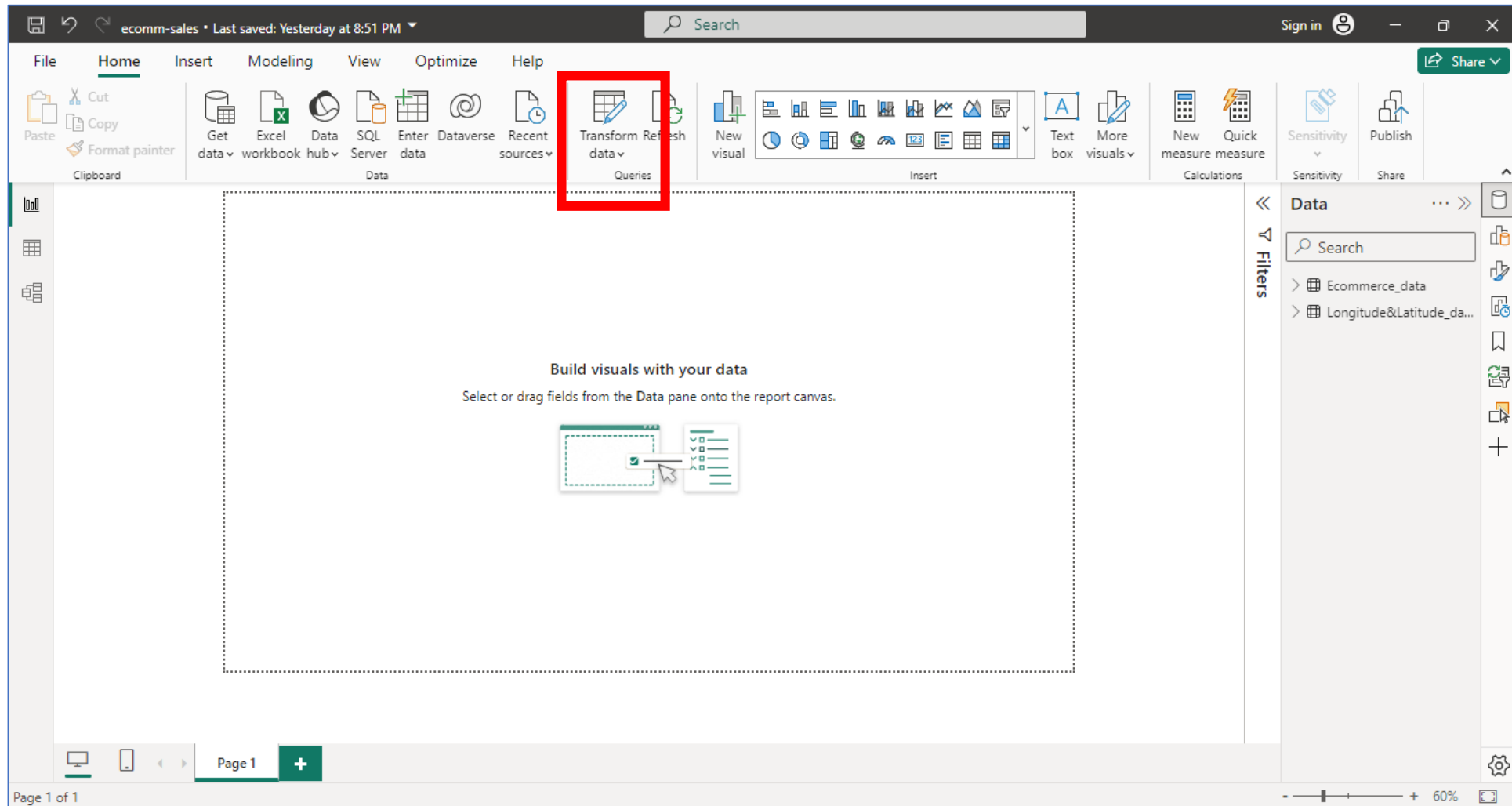
Once you've connected to your data source in Power BI, the next step is to transform the data into a format that can be easily analyzed and visualized. This process is often known as ETL (extract, transform, load).

The Power Query Editor provides a user-friendly interface that allows users to perform a series of data transformations through a set of tabs.

Lets go through some basic data transformation steps in Power Query Editor---



To open Power Query Editor, click on “Transform data” under Home tab.



# 1. FORMAT TABLE HEADER

Sometimes, power query include headers in data as a first row so to set that first row as header click on “Use First Row as Headers” under Transform tab.

The screenshot shows the Microsoft Power Query Editor interface. The 'Transform' tab is active, and the 'Use First Row as Headers' option is highlighted with a red box. The data table below has the following columns and data:

	customer_id	customer_first_name	customer_last_name	category_name	product_name
1	C_ID_45866	Mary	Fuller	Office Supplies	Xerox 1913
2	C_ID_44932	Alan	Edelman	Office Supplies	#6 3/4 Gummed Flap White Envelope
3	C_ID_70880	Mary	Gauman	Office Supplies	Belkin 8 Outlet Surge Protector

## 2. CHANGING DATA TYPE OF COLUMN

Check the data type of column shown beside column name. If found any wrong data type select particular column, click on “Detect Data Type” under Transform tab to automatically set data type OR click on data type beside column name and select respected type.

The screenshot displays the Microsoft Excel interface, specifically the 'Transform' tab. The 'Detect Data Type' button is highlighted with a red box. Below the ribbon, the 'Queries [2]' pane shows 'Ecommerce\_data' and 'Longitude&Latitude\_data'. The main data table has columns: 'customer\_id', 'customer\_first\_name', 'customer\_last\_name', 'category\_name', and 'product\_name'. The 'customer\_id' column is selected, and a dropdown menu is open, showing various data types like 'Decimal Number', 'Fixed decimal number', 'Whole Number', 'Percentage', 'Date/Time', 'Date', 'Time', 'Date/Time/Timezone', 'Duration', 'Text', 'True/False', 'Binary', and 'Using Locale...'. The 'Text' option is highlighted with a red box. The formula bar shows the formula: `= Table.TransformColumnTypes("#Promoted Headers",{{"customer_id", type text}}, {"customer_first_name",`

	customer_id	customer_first_name	customer_last_name	category_name	product_name
1	C_ID_45866			Office Supplies	Xerox 1913
2	C_ID_44932			Office Supplies	#6 3/4 Gummed Flap
3	C_ID_70880			Office Supplies	Belkin 8 Outlet Surge
4	C_ID_33157			Office Supplies	GBC VeloBinder Manu
5	C_ID_58303			Furniture	Eldon Pizzaz Desk Acc
6	C_ID_34556			Office Supplies	Hoover Shoulder Vac
7	C_ID_56882			Office Supplies	Rogers Handheld Barr
8	C_ID_51638			Office Supplies	Dixon Prang Watercol
9	C_ID_60578			Office Supplies	Ibico Hi-Tech Manual

### 3. RENAMING COLUMN NAME

Columns from raw data can be difficult to read or meaningless. Renaming the columns in your query to a meaningful name will make it easier for you to understand your data. Either double-click on column name and rename it OR simply select the column and click on “Rename” under Transform tab.

The screenshot displays the Power Query Editor interface for a query named 'ecommerce-sales'. The 'Transform' tab is active, and the 'Rename' button in the ribbon is highlighted with a red rectangle. Below the ribbon, the formula bar shows the following formula:

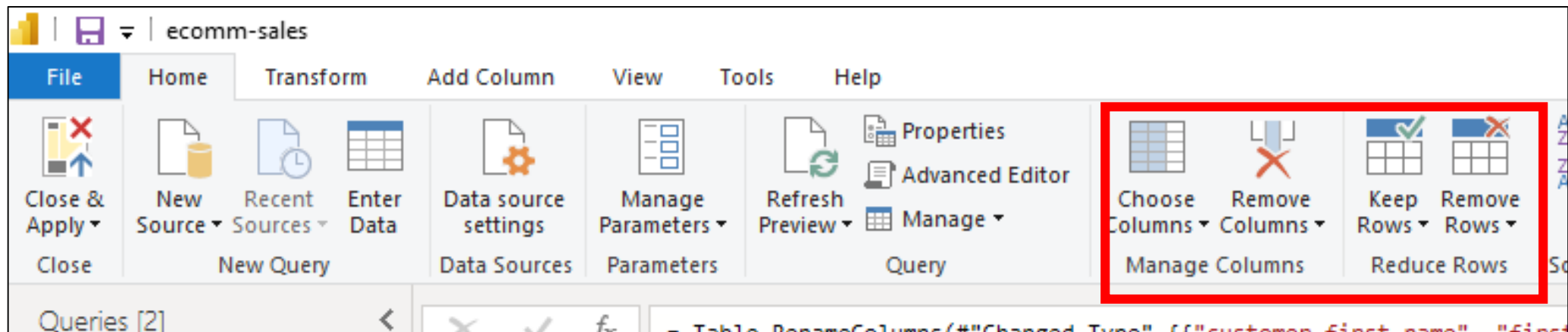
```
= Table.RenameColumns(#"Changed Type", {"customer_first_name", "first_name"})
```

The data table below the formula bar has the following columns: customer\_id, first\_name, customer\_last\_name, category\_name, and product\_name. The 'customer\_last\_name' column is highlighted in blue, and a red box is drawn around it. The data rows show various values for each column, with status indicators (Valid, Error, Empty) and percentages (100%, 0%) for each column.

## 4. REMOVE COLUMNS/ROWS

Use this transformation to clean up your dataset by removing unnecessary or redundant columns, which can improve query and report performance.

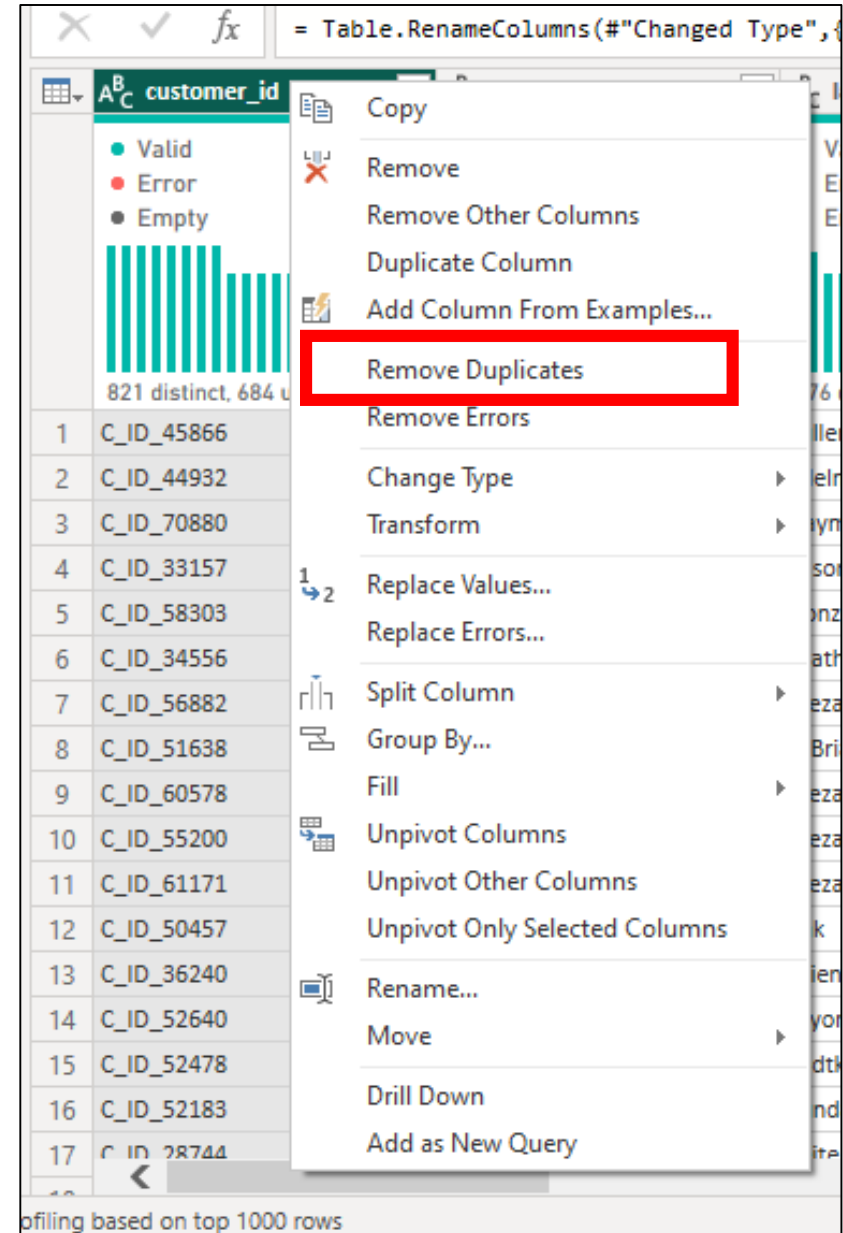
Simply select the column OR row and click on “Remove column” or “Remove row” under Home tab.



## 5. REMOVE DUPLICATES

It allows you to eliminate duplicate rows from your dataset based on the values in one or more columns. It helps you maintain data integrity by keeping only unique Records.

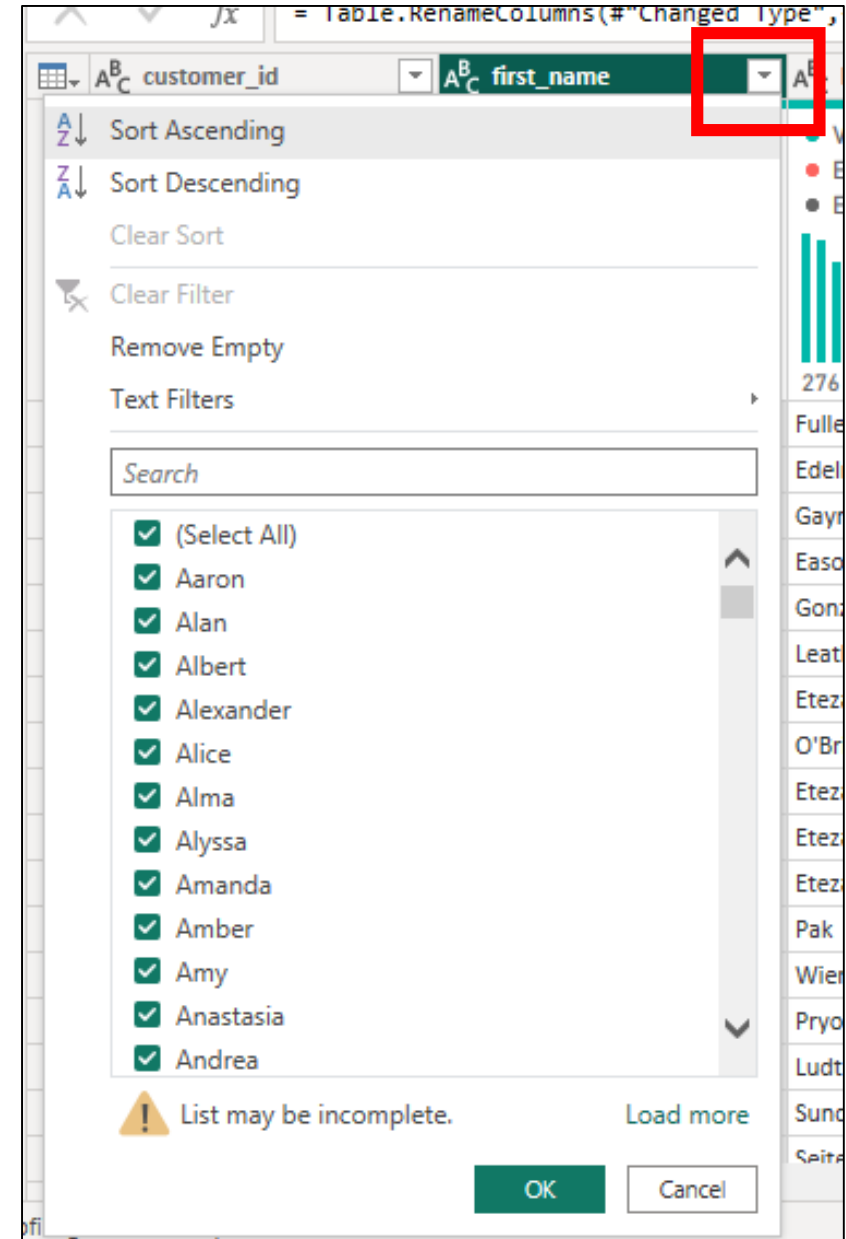
Simply right-click on the column and click on “Remove duplicates”.



## 6. FILTER ROWS

It enables you to include or exclude rows from your dataset based on specific criteria or conditions.

Click on the filter arrow besides column name and check the distinct values- if there are any misspell words, formats, missing values, etc





## 7. MERGE COLUMN

It lets you combine the values from multiple columns into a single new column. You can choose the delimiter or separator to use between merged values.

Select 2 columns that you need to concatenate and click on “Merge columns” under Transform tab.

The screenshot shows the 'Merge Columns' dialog box in the Power Query editor. The dialog is titled 'Merge Columns' and has a 'Text Column' section. The 'Delimiter' is set to 'Space' and the 'Separator' is set to 'None'. The 'Columns to merge' list contains 'first\_name' and 'last\_name'. The 'Merge Columns' button is highlighted with a red box.

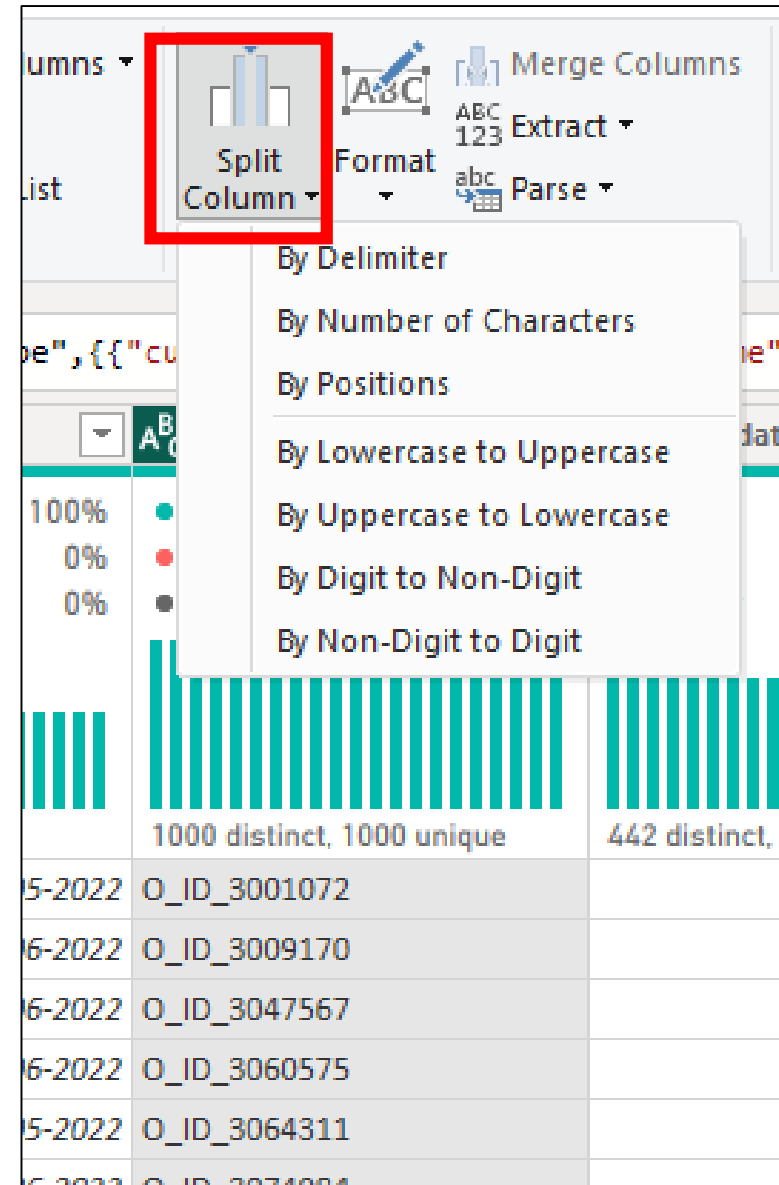
Below the dialog, the resulting data table is shown. The table has three columns: 'first\_name', 'last\_name', and 'title'. The 'first\_name' column has 232 distinct values and 93 unique values. The 'last\_name' column has 276 distinct values and 89 unique values. The 'title' column has 3 distinct values and 3 unique values. The table contains 8 rows of data.

first_name	last_name	title
Mary	Fuller	Offic
Alan	Edelman	Offic
Mary	Gayman	Offic
Raymond	Eason	Offic
Mary	Gonzalez	Furn
Joe	Leatherbury	Offic
Mary	Etezadi	Offic

## 8. SPLIT COLUMN

It allows you to divide a single column into multiple columns based on a specified delimiter or pattern.

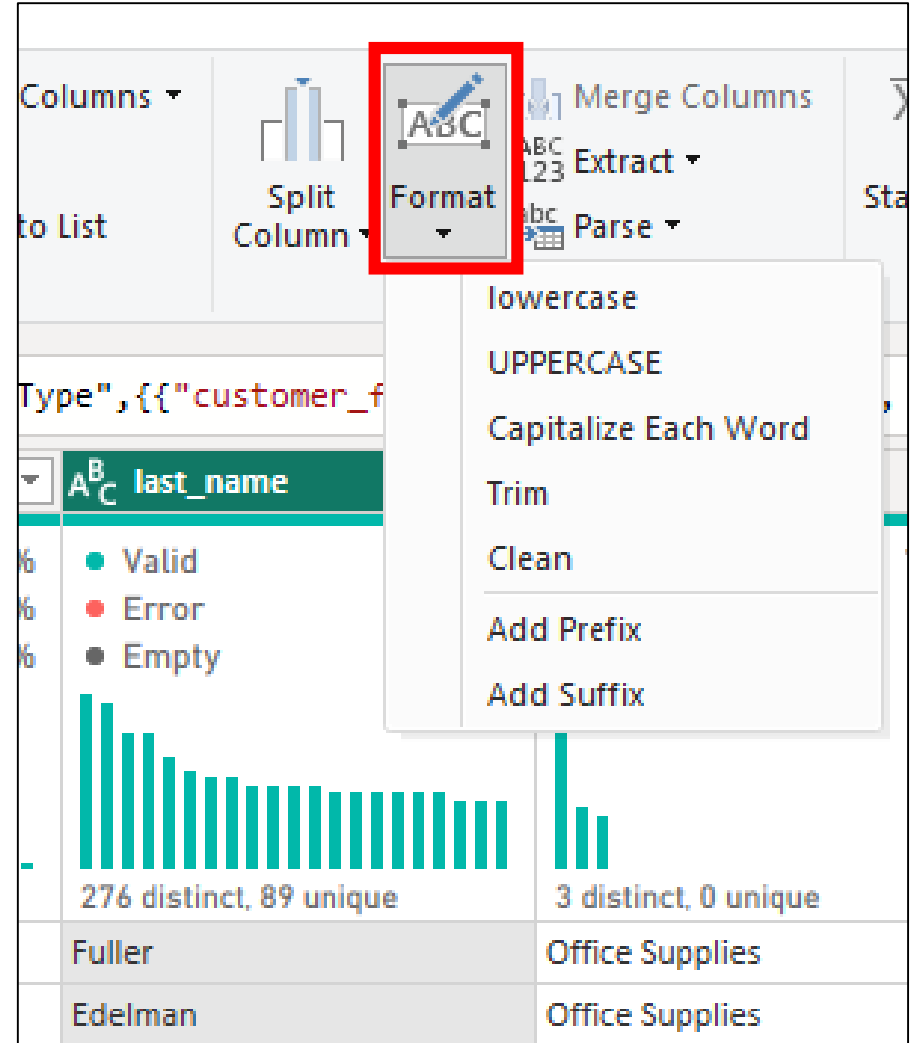
Select the column and click on “split column” and select the respective option as per selected column.



## 9. FORMAT COLUMN

It enables you to standardized the data like trim, lowercase, uppercase, etc

Select the column and click on “format” and select respected option.



The screenshot shows a data tool interface. At the top, there's a 'Columns' dropdown and a 'Split Column' button. A red box highlights the 'Format' button, which has a dropdown menu open. The menu options are: lowercase, UPPERCASE, Capitalize Each Word, Trim, Clean, Add Prefix, and Add Suffix. Below the menu, a table is visible with a column named 'last\_name'. The table has a header row and two data rows. The first data row is 'Fuller' and the second is 'Edelman'. To the right of the table, there's a summary section with two bar charts. The first chart shows '276 distinct, 89 unique' and the second chart shows '3 distinct, 0 unique'. Below the charts, there's a table with two rows: 'Fuller' and 'Edelman', both with the value 'Office Supplies'.

Type	last_name
Valid	Fuller
Error	Edelman
Empty	

276 distinct, 89 unique

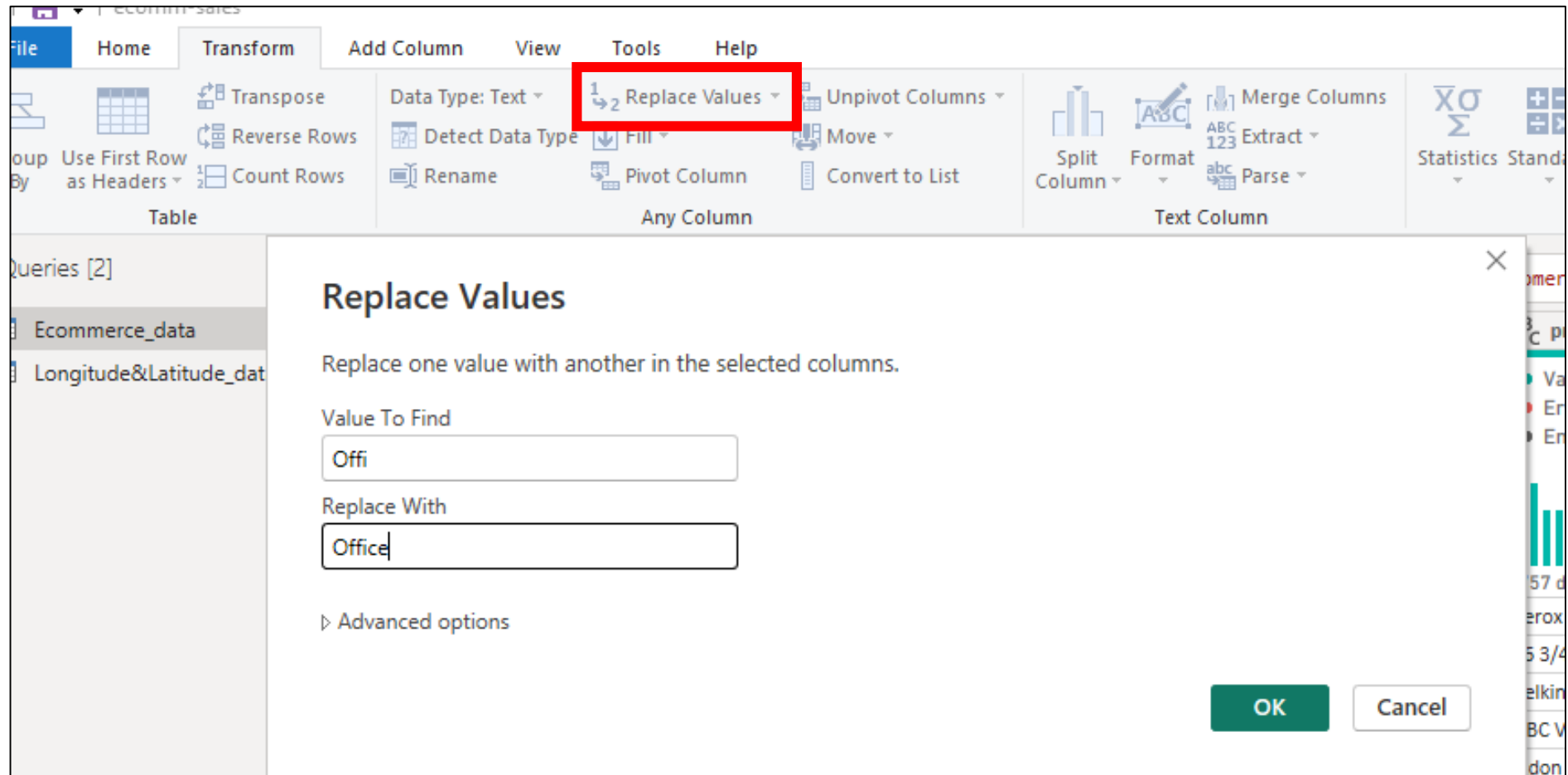
3 distinct, 0 unique

Fuller	Office Supplies
Edelman	Office Supplies

## 10. REPLACE VALUES

It enables you to format misspell words.

Click on “Replace values” under Transform tab and specify the required words.

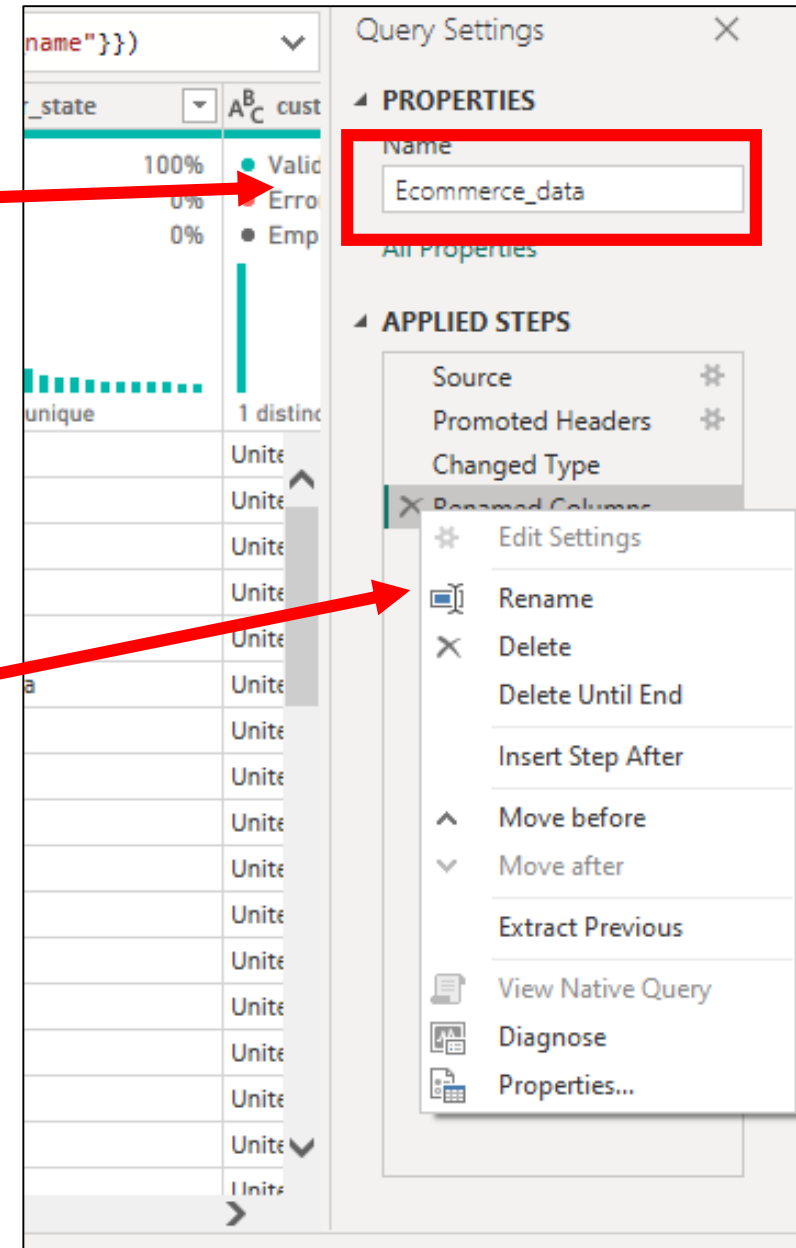


## 11. EDIT TABLE NAME AND RENAME APPLIED STEPS

You can edit your table name here.

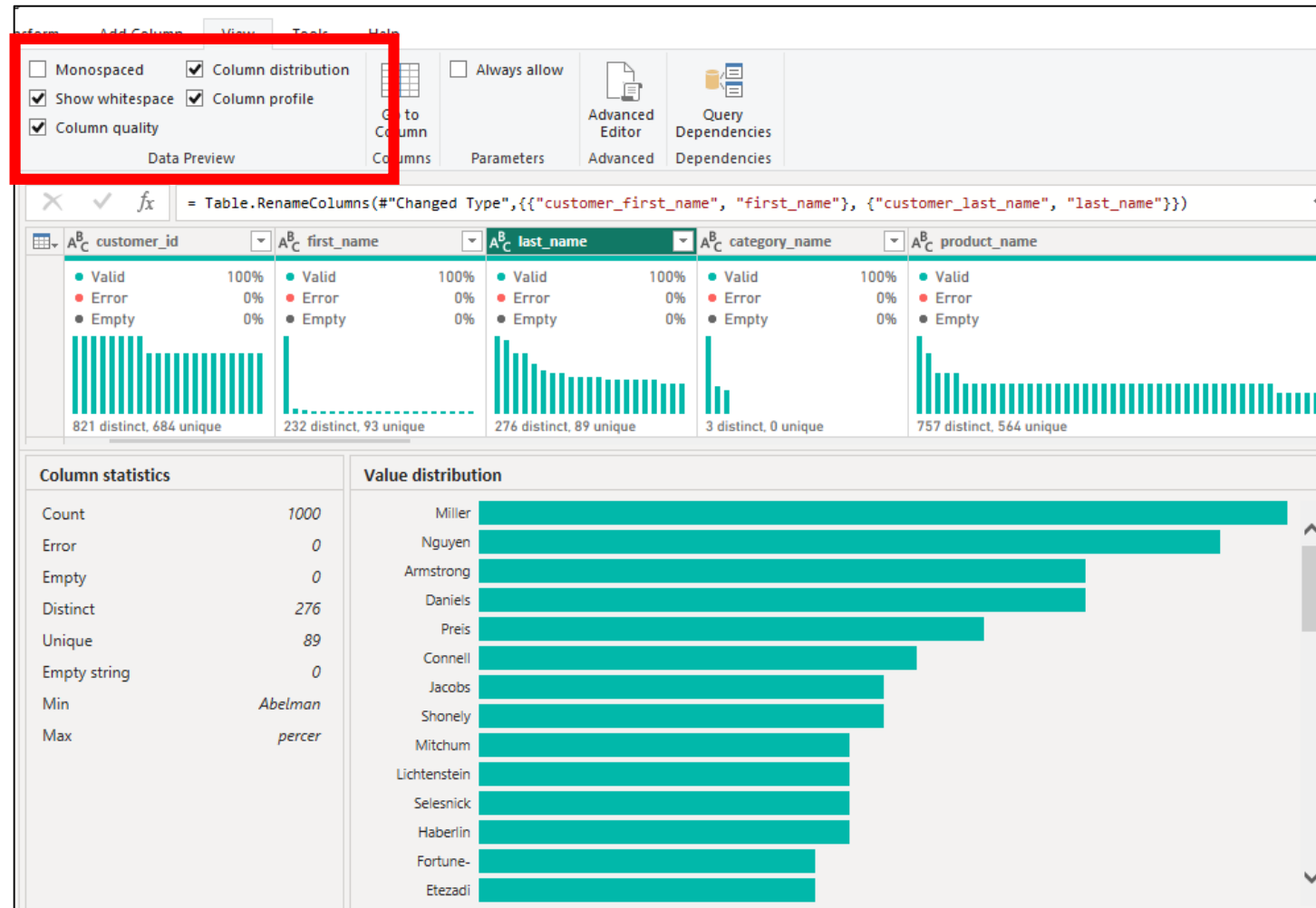
Rename the steps that you performed while transforming the data so that when you revisit the project after days you know what steps were performed.

Right-click on particular step and select rename.



## 12. DATA PREVIEW

While working on large datasets, it is necessary to get an basic idea of the data in a seconds, for that turn ON following options under View tab – Column distribution, Column profile, Column quality.



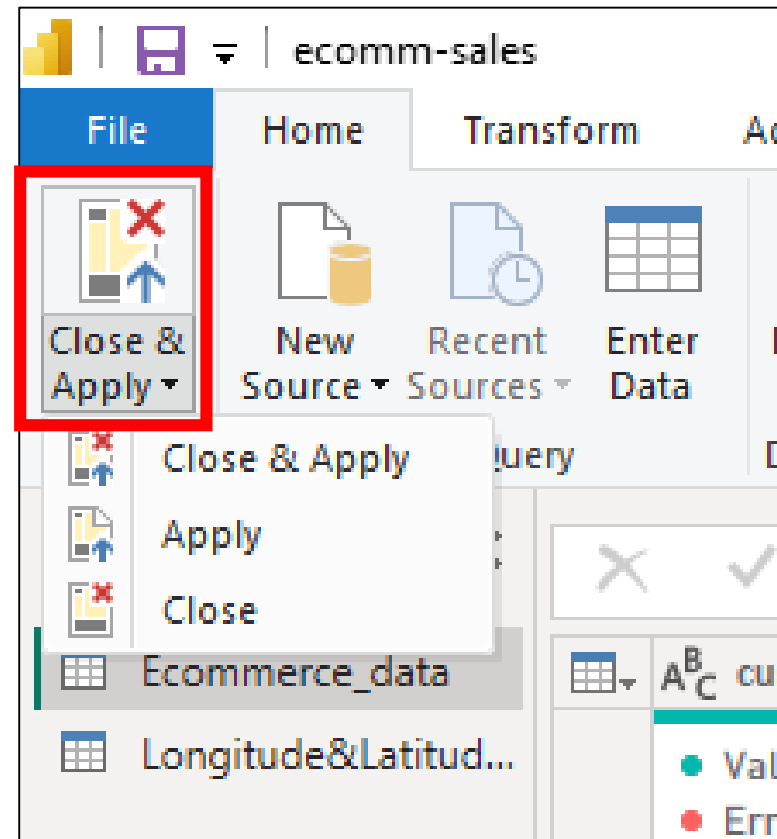
# LOADING DATA TO POWER-BI DESKTOP FROM POWER QUERY EDITOR

After transforming your data in Power Query, you can load it into Power BI Desktop to create your reports and visualizations.

Under home tab Click on “close & Apply” to save changes and close power query.

Selecting “Apply ” will just save the changes.

Selecting “close” will close the power query without saving the changes.





# **MICROSOFT POWER-BI**

**SERIES 4**

## **DATA MODELLING**

**- MAYURI .D.**



# INTRO

Data Modelling in Power-BI is fundamental process that involves creating a structures, interconnected data framework to enable efficient data analysis and reporting.

Key aspects in data modelling within Power-BI are—

1. Designing the data model
2. Establishing relationships

## DESIGNING THE DATA MODEL

The core of data modelling in Power-BI involves designing a schema that defines how different data tables relate to each other.

This is usually done using a star schema or snowflake schema.

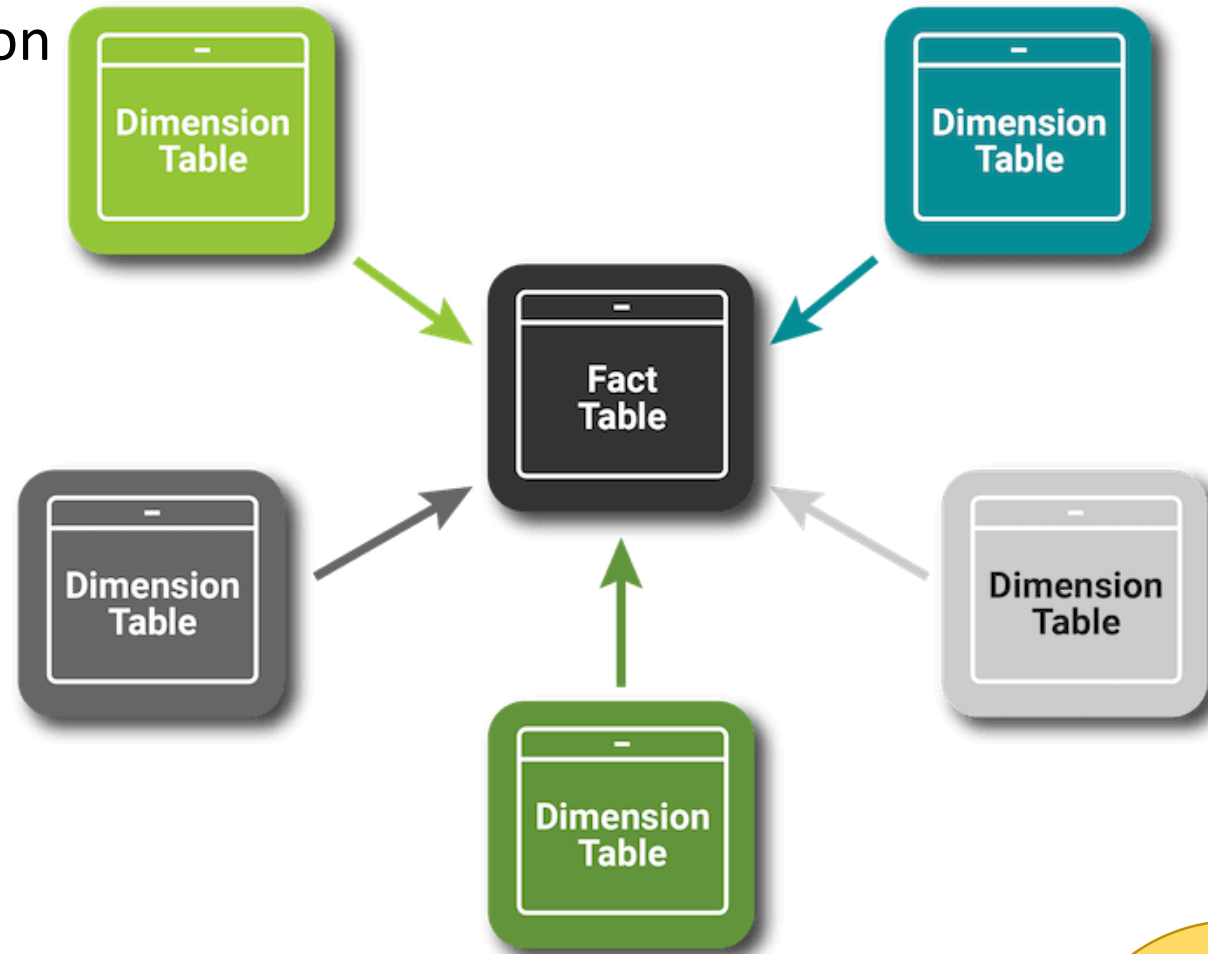
There are two types of tables in any data model—

### 1. Slicer tables – Dimensions

- They keep descriptive information
- They are used for slicing and dicing
  - Slicers or filters
  - Axis/legends of chart
- Descriptive doesn't means only text data type, also includes other data types
- Small table in no. of rows but big in no. of columns

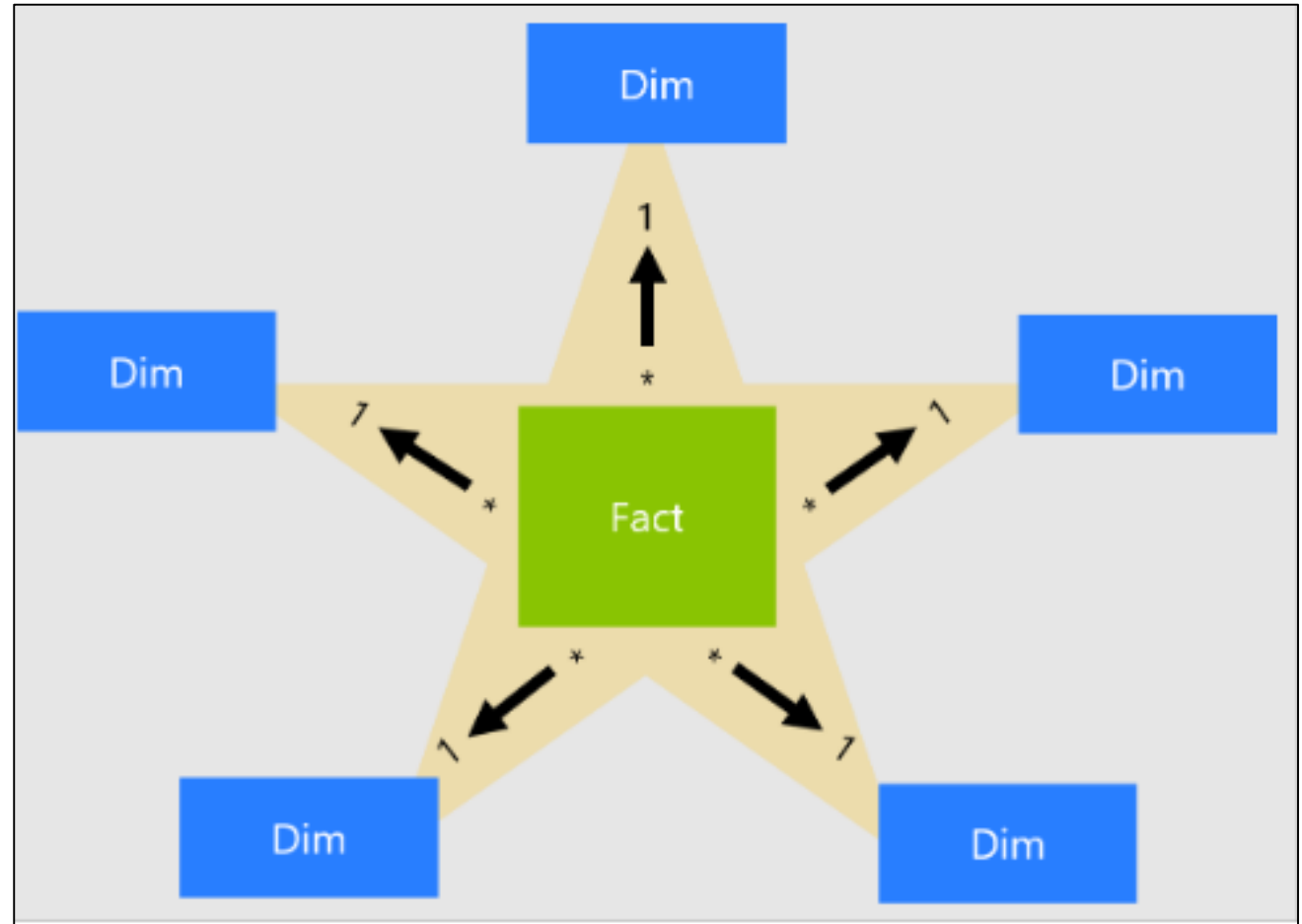
## 2. Value tables – Facts

- Explains happening of an action
- Keeps values (numeric, aggregable)
- Used in values in charts and visuals
- Small table in no. of columns and huge in no. of rows



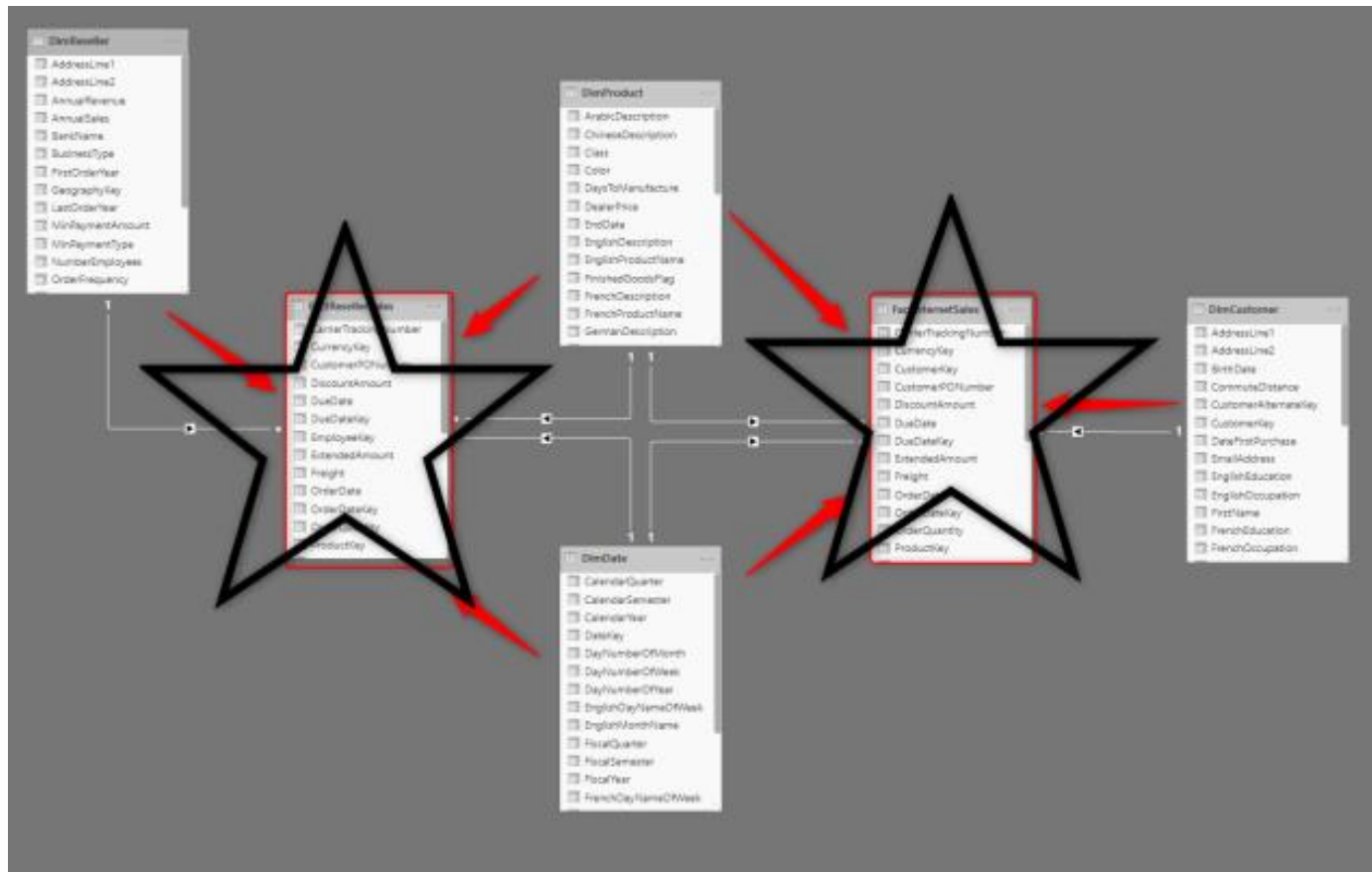
## Relationship between Fact and Dimensions tables---

- One to Many
- From dimension to fact
- Single direction
- Star schema



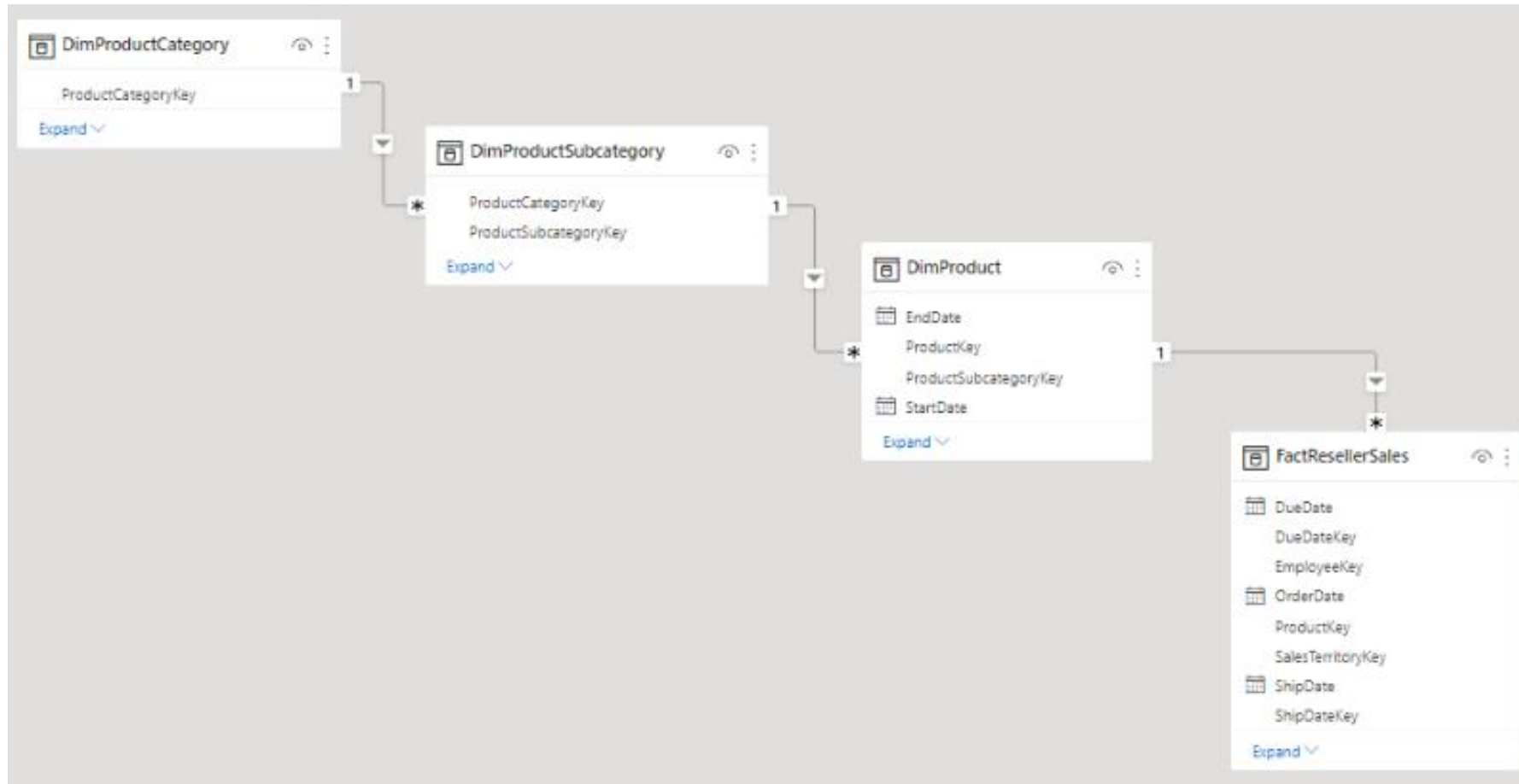
# STAR SCHEMA

- Fact table in the middle and other dim tables surrounding it.
- It can also have more than 1 fact tables surrounded by dim tables.
- It cannot have relationship between dim to dim table.



# SNOWFLAKE SCHEMA

- It is a variation of star schema
- Dim tables are normalized into multiple related tables.



## WHAT IS A GOOD DATA MODEL?

- It has star schema
- It has shared dimensions tables
- It includes fact table
- Single direction relationship
- One to many relationship

## ESTABLISHING RELATIONSHIPS

In Power-BI, relationships between tables are created to enable data to be combined across them.

This is done using—

- **Primary and foreign keys**— ensuring unique identifiers in the fact tables match the corresponding keys in dimension tables
- **Cardinality** – defining the type of relationship (one to one, one to many or many to many)
- **Cross-filter direction** – specifying the direction in which filters should flow (single or both direction)



## CARDINALITY

Each model relationship is defined by a cardinality type. There are four cardinality type options, representing the data characteristics of the "from" and "to" related columns.

The "one" side means the column contains unique values; the "many" side means the column can contain duplicate values.

There are 4 types of cardinality--

**One-to-many (and many-to-one) cardinality**-- When configuring a one-to-many or many-to-one relationship, you'll choose the one that matches the order in which you related the columns.

**One-to-one cardinality** -- A one-to-one relationship means both columns contain unique values

**Many-to-many cardinality** -- A many-to-many relationship means both columns can contain duplicate values

## CROSS FILTER DIRECTION

Each model relationship is defined with a cross filter direction. Your setting determines the direction(s) that filters will propagate. The possible cross filter options are dependent on the cardinality type.

Single cross filter direction means "single direction", and Both means "both directions".

A relationship that filters in both directions is commonly described as bi-directional

Cardinality type	Cross filter options
One-to-many (or Many-to-one)	Single Both
One-to-one	Both
Many-to-many	Single (Table1 to Table2) Single (Table2 to Table1) Both

## ACTIVE AND INACTIVE RELATIONSHIPS

By default, relationship is always active.

If the tables contains more than one relationship, then first relationship is active rest all remains inactive.

Active relationship is a solid line and inactive relationship is a dotted line.

Inactive relationship doesn't play any role, it just remains there. So to make it working there are 2 methods– Define it using USERELATIONSHIP() in DAX or Role playing dimension(RPD)



## **METHOD 1--** Define using USERELATIONSHIP() in DAX

You can use the USERELATIONSHIP DAX function to activate a specific relationship for relevant model calculations.

### **Syntax--**

USERELATIONSHIP(<columnName1>,<columnName2>)

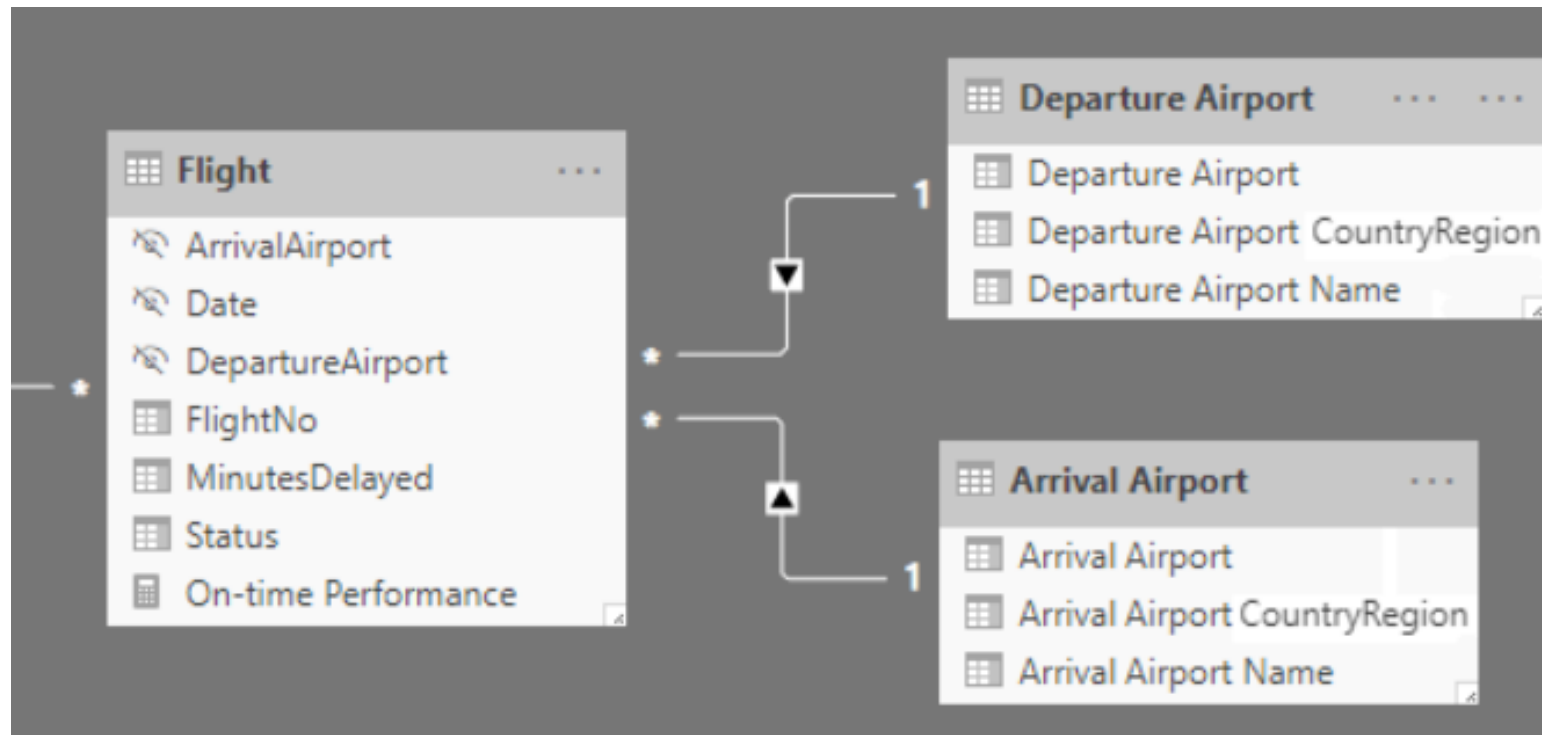
In USERELATIONSHIP, the status of a relationship is not important; that is, whether the relationship is active or not does not affect the usage of the function.

Even if the relationship is inactive, it will be used and overrides any other active relationships that might be present in the model but not mentioned in the function arguments.

## METHOD 2– Role Playing Dimensions

Duplicate table is created, and then create a active relationship between them.

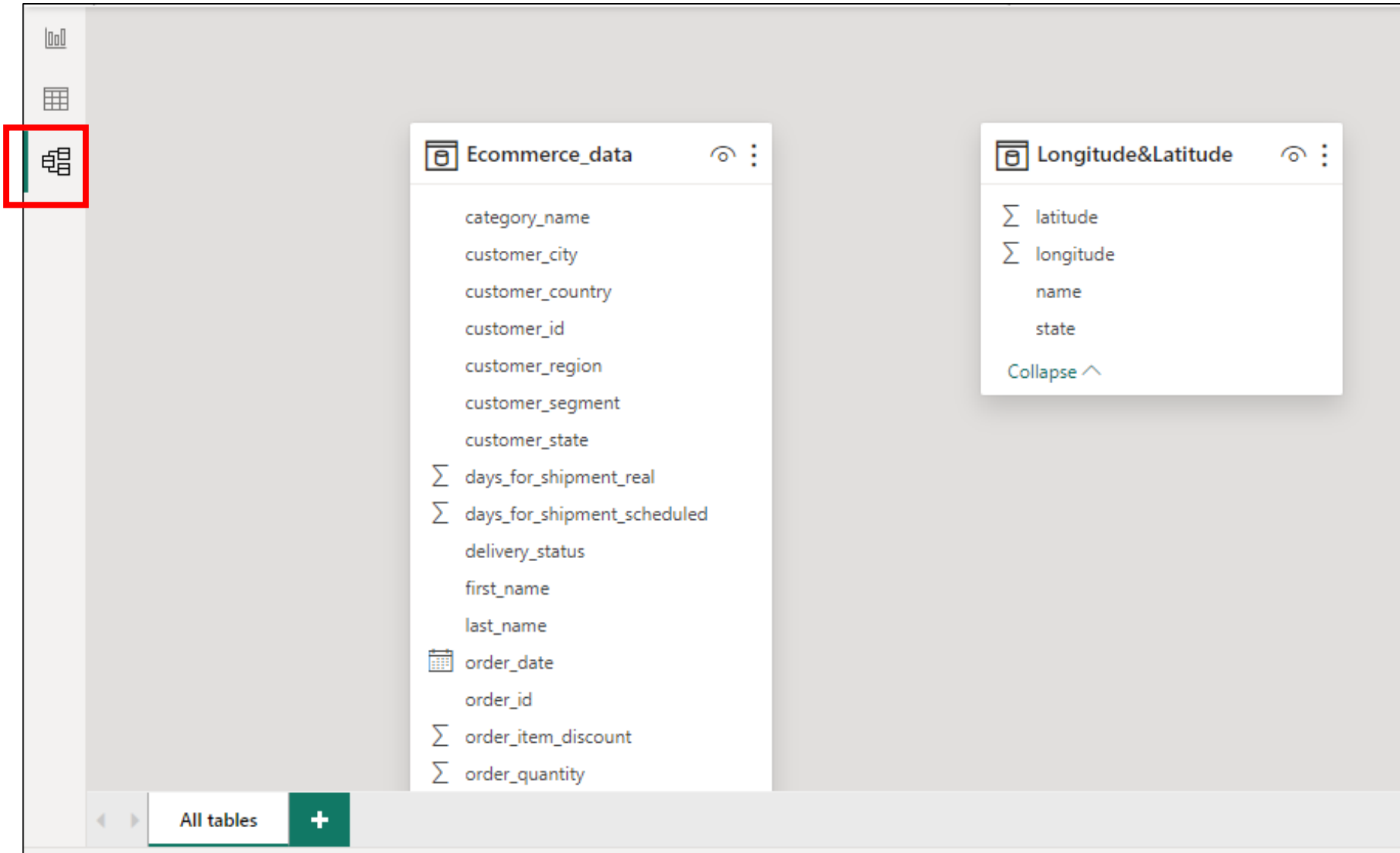
Here, Airport table is duplicated and named them accordingly.





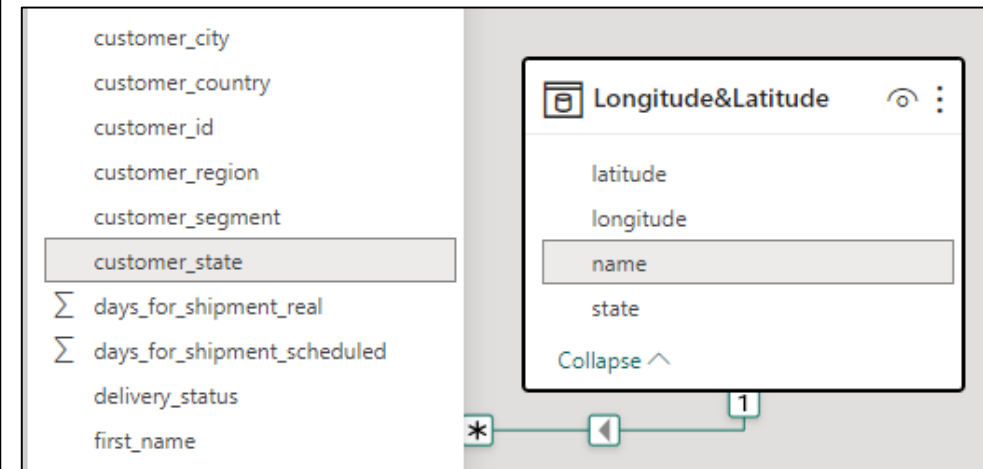
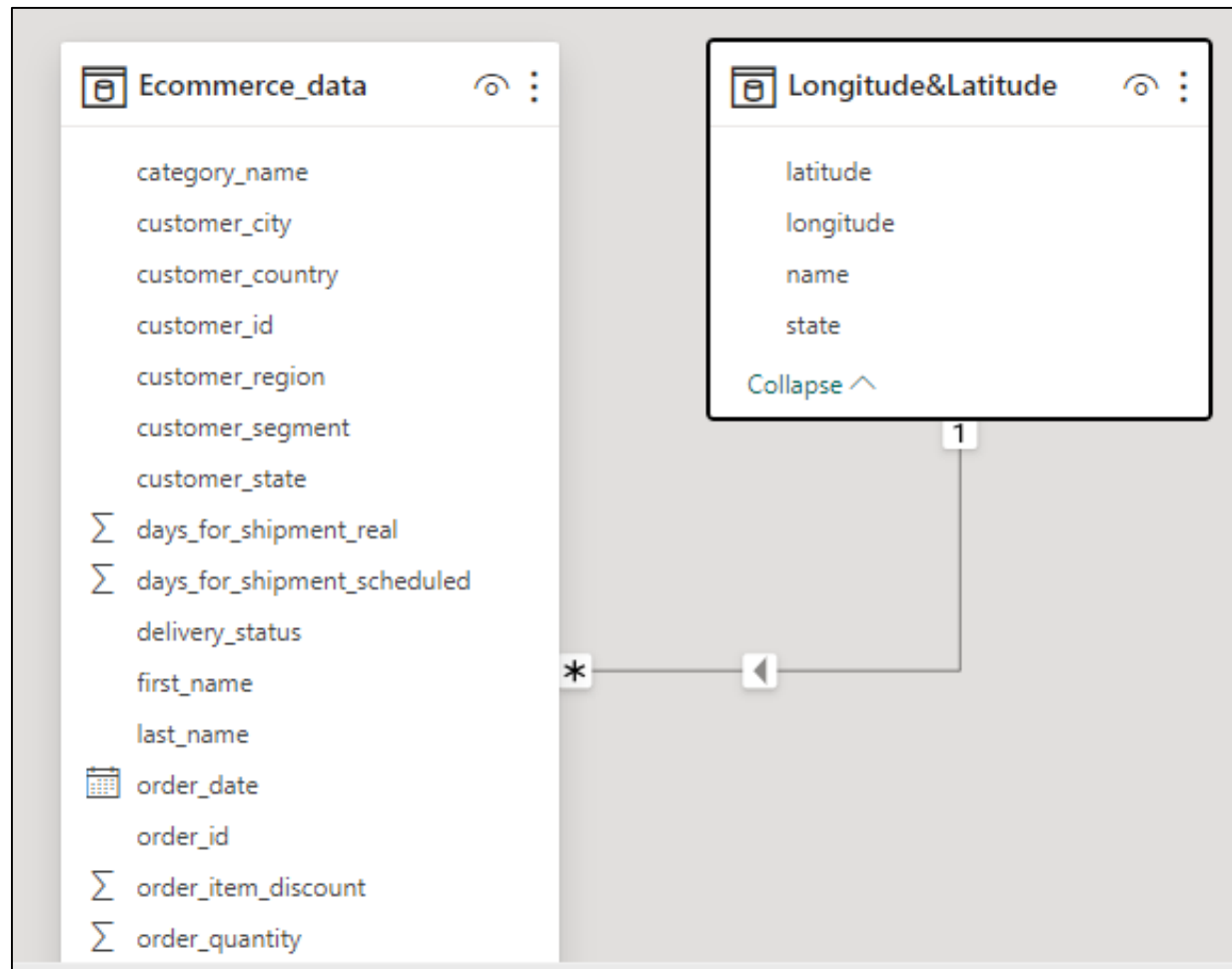
## **Practical Implementation on Ecommerce sales dataset**

Open Model View and check if Power-BI has automatically detected relationship, if not you have to create them.



## Method 1---

Check which columns are common in both table. Drag column from one table on the other table's respective column, it creates automatic relationship.



Hover on relationship line to see which columns are connected.



## Method 2---

Step1--

Click on “ Manage Relationship” then “New”

The screenshot displays the Power BI Desktop ribbon and the 'Manage relationships' pane. The 'Manage relationships' button on the ribbon is highlighted with a red box. The 'Manage relationships' pane is open, showing a table with columns 'Active', 'From: Table (Column)', and 'To: Table (Column)'. The table is empty, with the text 'There are no relationships defined yet.' displayed. The 'New...' button at the bottom of the pane is also highlighted with a red box. Other buttons visible in the pane include 'Autodetect...', 'Edit...', 'Delete', and 'Close'.

**Manage relationships**

Active	From: Table (Column)	To: Table (Column)
There are no relationships defined yet.		

**New...** Autodetect... Edit... Delete

Close

Step 2--

Select table

Select column

Select column

### Create relationship

Select tables and columns that are related.

Ecommerce\_data

customer_segment	customer_city	customer_state	customer_country	customer_region	delivery_status
Consumer	Los Angeles	California	United States	West	Advance shipping
Consumer	Los Angeles	California	United States	West	Shipping on time
Consumer	Los Angeles	California	United States	West	Late delivery

Longitude&Latitude

state	latitude	longitude	name
AK	63.588753	-154.493062	Alaska
AL	32.318231	-86.902298	Alabama
AR	35.20105	-91.831833	Arkansas

Cardinality

Many to one (\*:1)

Cross filter direction

Single

☒ Make this relationship active

☐ Assume referential integrity

☐ Apply security filter in both directions

OK

Cancel

Cardinality types



AR	55.20105	-91.831855	Arkansas
----	----------	------------	----------

Cardinality

Many to one (\*:1)

Many to one (\*:1)

One to one (1:1)

One to many (1:\*)

Many to many (\*:\*)

Cross filter direction

Single

☐ Apply second filter

Relationship direction



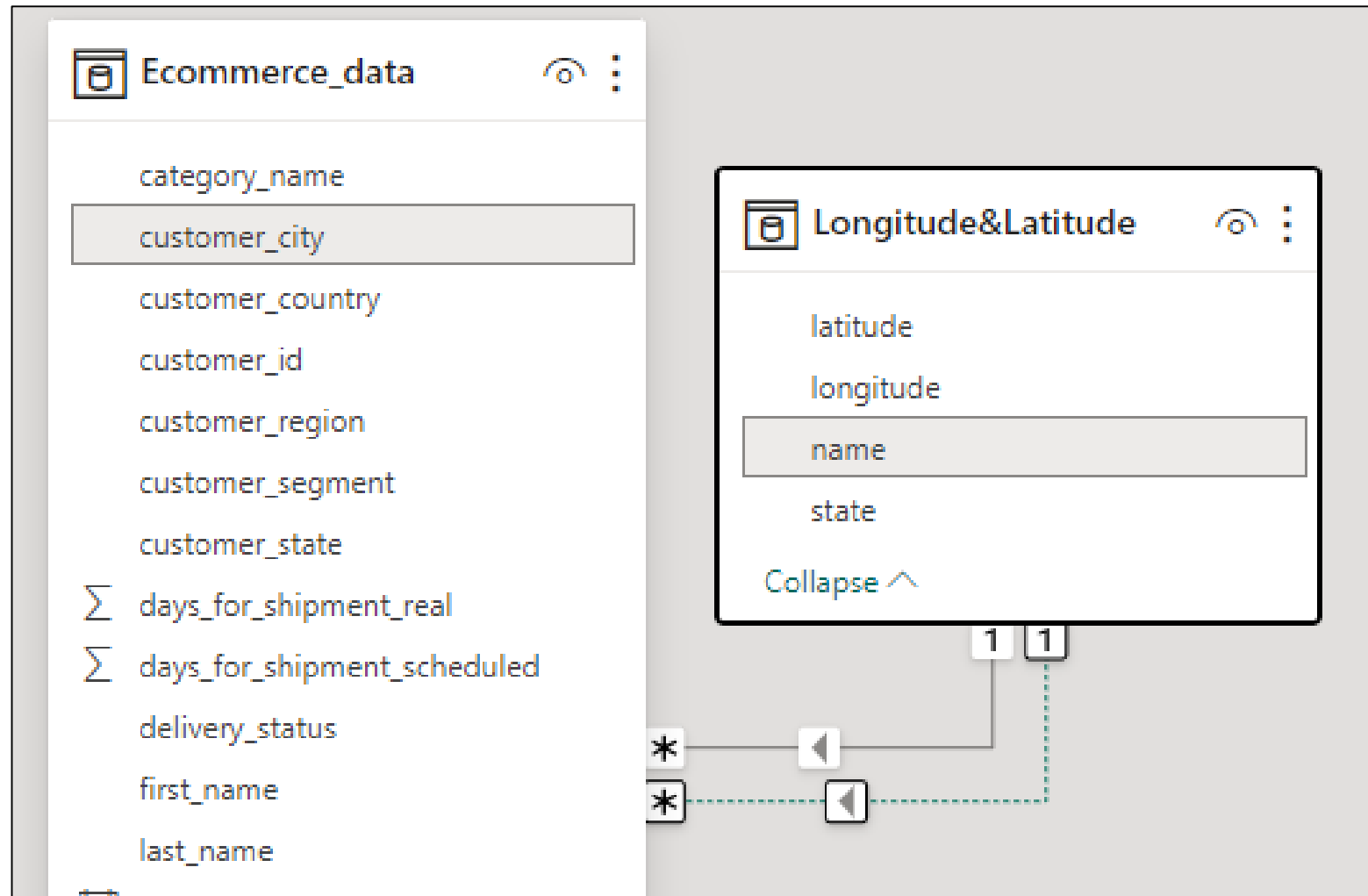
Cross filter direction

Single

Single

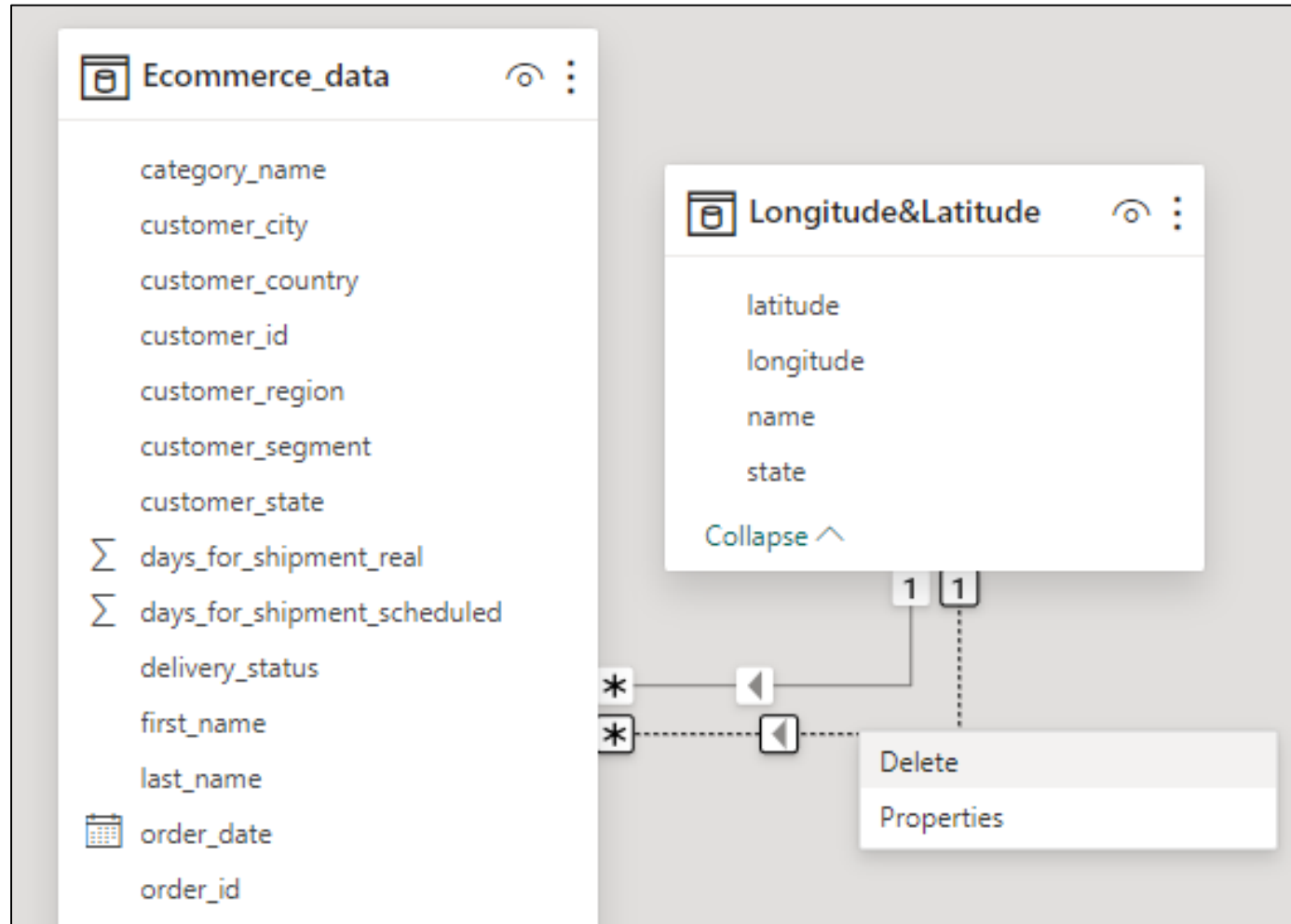
Both

(JUST FOR DEMO)– Inactive relationship  
It is identified by dotted line.



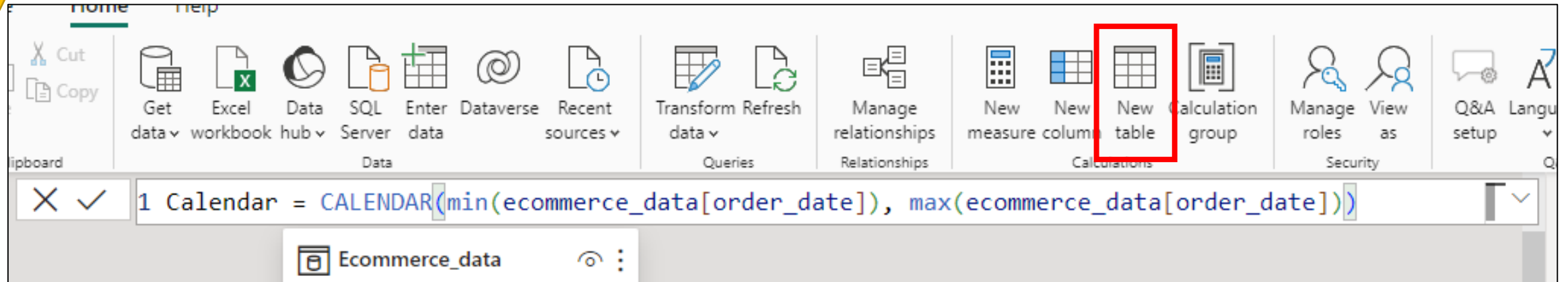
If any relationship is wrong, you can delete it or edit it.  
Right click on the line—

- To delete click delete
- To edit click properties



You need to create a Calendar table.

A separate date table enhances time intelligence capabilities, ensures consistency across fact tables, and enables more flexible and efficient date-based filtering and calculations.

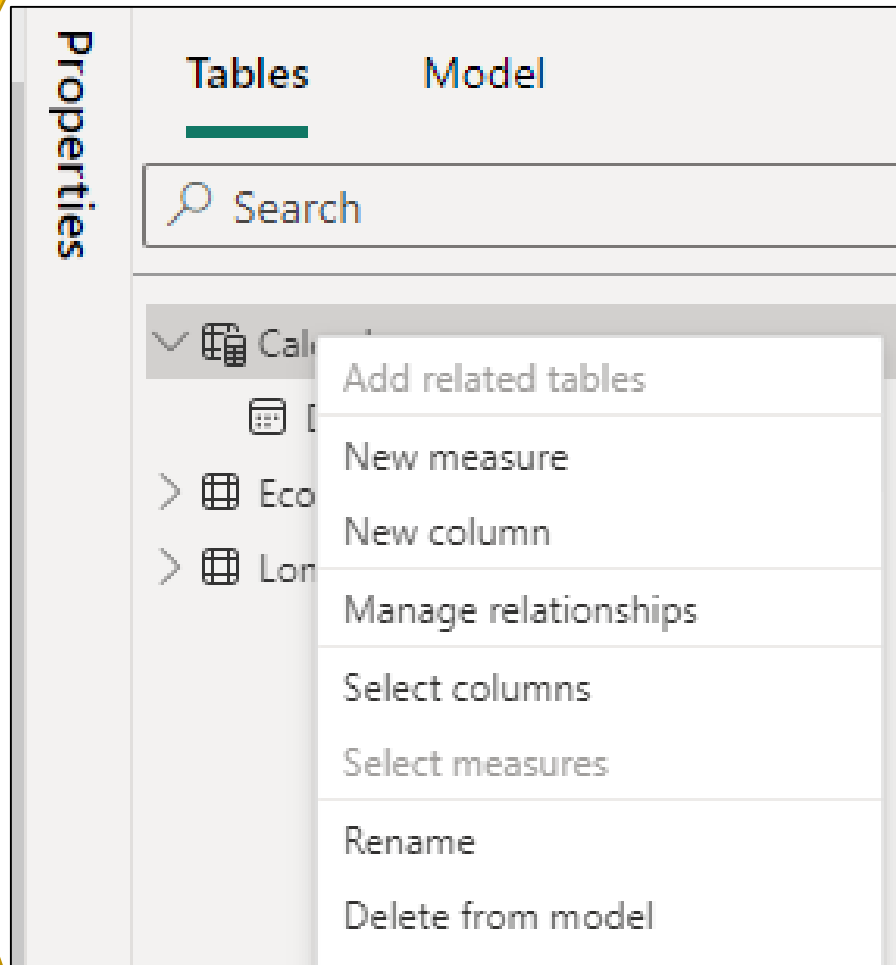


Click on “new table”

Write DAX query for new table

Press Enter

You need to extract Year, Month, Month\_Number from Date column.



Right-Click on table name select “new column”  
write following queries for each column



```
1 Year = year('Calendar'[Date])
```

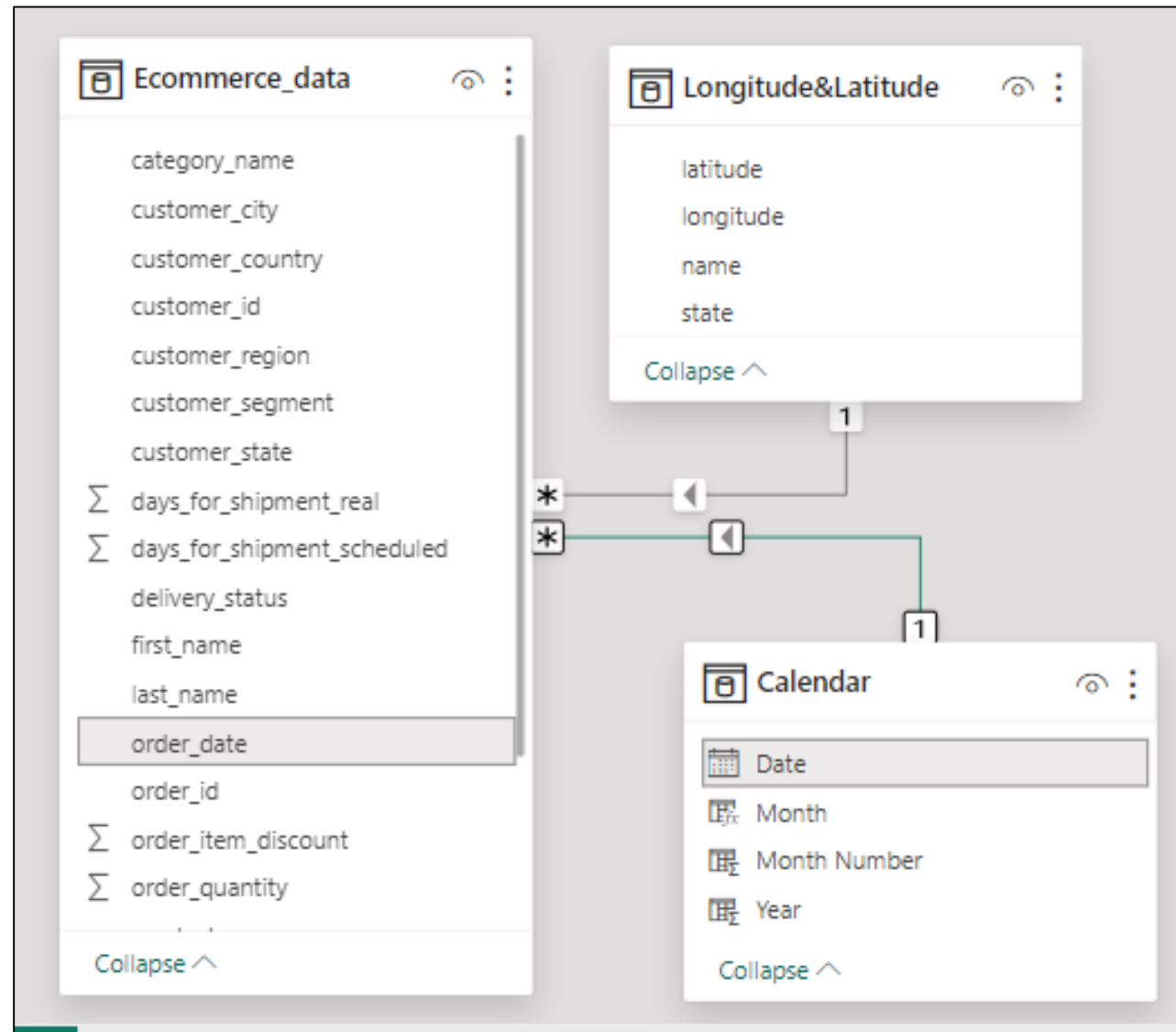


```
1 Month = format('Calendar'[Date], "mmm")
```



```
1 Month Number = MONTH('Calendar'[Date])
```

## Create relationship between date columns





# **MICROSOFT POWER-BI**

**SERIES 5**

## **DAX (DATA ANALYSIS EXPRESSION)**

**- MAYURI .D.**

# INTRO

Data Analysis Expressions (DAX) is a formula expression language used in Analysis Services, Power BI, and Power Pivot in Excel.

DAX formulas include functions, operators, and values to perform advanced calculations and queries on data in related tables and columns in tabular data models.

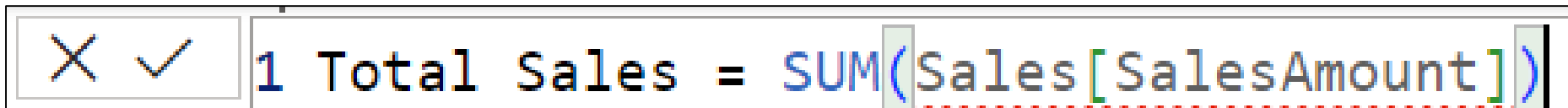
DAX formulas are used in measures, calculated columns, calculated tables, and row-level security.

- **Calculated columns** are computed during data load and are stored in the data model, often used for intermediate calculations.
- **Measures** are dynamic calculations evaluated at query time and provide flexibility and efficiency for reporting.
- **Calculated table** is a computed object, based on a formula expression, derived from all or part of other tables in the same model. Instead of querying and loading values into your new table's columns from a data source, a DAX formula defines the table's values.
- With **row-level security**, a DAX formula must evaluate to a Boolean TRUE/FALSE condition, defining which rows can be returned by the results of a query by members of a particular role.

## SYNTAX

Syntax includes the various elements that make up a formula, or more simply, how the formula is written.

For example, here's a simple DAX formula for a measure:

A screenshot of a software interface showing a formula bar. On the left, there is a small box containing a red 'X' and a green checkmark. To the right of this box, the text '1 Total Sales = SUM(Sales[SalesAmount])' is displayed. The text is color-coded: '1' is blue, 'Total Sales' is black, '=' is black, 'SUM' is blue, 'Sales' is black, '[SalesAmount]' is black, and the closing parenthesis ')' is blue. A red dashed line is visible below the text.

The measure name, **Total Sales**.

The equals sign operator (**=**), which indicates the beginning of the formula. When calculated, it will return a result.

The DAX function **SUM**, which adds up all of the numbers in the Sales[SalesAmount] column.

Parenthesis (**()**), which surround an expression that contains one or more arguments. An argument passes a value to a function.

The referenced table, **Sales**.

The referenced column, **[SalesAmount]**, in the Sales table. With this argument, the SUM function knows on which column to aggregate a SUM.

# FUNCTIONS

**Functions** are predefined formulas that perform calculations by using specific values, called arguments, in a particular order or structure.

Arguments can be other functions, another formula, expression, column references, numbers, text, logical values such as TRUE or FALSE, or constants.

DAX includes the following categories of functions:-

Aggregation functions, Date and Time functions, Filter functions, Financial functions, Information functions, Logical functions, Math and trig functions, Other functions, Parent and child functions, Relationship functions, Statistical functions, Table manipulation functions, Text functions, Time intelligence functions.

Some of the common DAX functions—

- **AVERAGE**-- Returns the average (arithmetic mean) of all the numbers in a column.  
syntax-- AVERAGE(<column>)
- **AVERAGEX**-- Calculates the average (arithmetic mean) of a set of expressions evaluated over a table  
syntax-- AVERAGEX(<table>,<expression>)
- **DISTINCTCOUNT**- Counts the number of distinct values in a column  
syntax-- DISTINCTCOUNT(<column>)

- **SUM**– Adds all the numbers in a column  
syntax-- SUM(<column>)
- **SUMX**- Returns the sum of an expression evaluated for each row in a table  
syntax-- SUMX(<table>,<expression>)
- **CALENDAR** Returns a table with a single column named "Date" that contains a contiguous set of dates. The range of dates is from the specified start date to the specified end date, inclusive of those two dates.  
syntax-- CALENDAR(<start\_date>, <end\_date>)
- **DATEDIFF**-- Returns the number of interval boundaries between two dates.  
syntax-- DATEDIFF(<Date1>, <Date2>, <Interval>)

- **NETWORKDAYS**-- Returns the number of whole workdays between two dates (inclusive). Parameters specify which and how many days are weekend days. Weekend days and days specified as holidays are not considered as workdays.  
syntax-- NETWORKDAYS(<start\_date>, <end\_date>[, <weekend>[, <holidays>]])
- **ALL**-- Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied. This function is useful for clearing filters and creating calculations on all the rows in a table.  
syntax-- ALL( [<table> | <column>[, <column>[, <column>[,...]]]] )
- **CALCULATE**-- Evaluates an expression in a modified filter context  
syntax-- CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]])



- **SWITCH**– Evaluates an expression against a list of values and returns one of multiple possible result expressions. This function can be used to avoid having multiple nested IF statements.  
syntax-- SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])
- **DIVIDE**- Performs division and returns alternate result or BLANK() on division by 0.  
syntax-- DIVIDE(<numerator>, <denominator> [, <alternateresult>])
- **GENERATESERIES**- Returns a single column table containing the values of an arithmetic series, that is, a sequence of values in which each differs from the preceding by a constant quantity. The name of the column returned is Value.  
syntax-- GENERATESERIES(<startValue>, <endValue>[, <incrementValue>])

- **CONCATENATE**-- Joins two text strings into one text string.  
syntax-- CONCATENATE(<text1>, <text2>)
- **DATESBETWEEN**-- Returns a table that contains a column of dates that begins with a specified start date and continues until a specified end date.  
syntax-- DATESBETWEEN(<Column>, <StartDate>, <EndDate>)
- **TOTALMTD**-- Evaluates the value of the expression for the month to date, in the current context.  
syntax-- TOTALMTD(<expression>, <dates>[, <filter>])
- **TOTALQTD**-- Evaluates the value of the expression for the dates in the quarter to date, in the current context.  
syntax-- TOTALQTD(<expression>, <dates>[, <filter>])

- **TOTALYTD--** Evaluates the year-to-date value of the expression in the current context  
syntax-- TOTALYTD(<expression>,<dates>[,<filter>],[<year\_end\_date>])
- **DATESYTD-** Returns a table that contains a column of the dates for the year to date, in the current context.  
syntax-- DATESYTD(<dates> [,<year\_end\_date>])
- **DATESMTD-** Returns a table that contains a column of the dates for the month to date, in the current context.  
syntax-- DATESMTD(<dates>)
- **DATESQTD-** Returns a table that contains a column of the dates for the quarter to date, in the current context.  
syntax-- DATESQTD(<dates>)

# DAX OPERATORS

The DAX language uses operators to create expressions that compare values, perform arithmetic calculations, or work with strings.

There are four different types of calculation operators: -

- Arithmetic operator,

- Comparison operator,

- Text concatenation operator, and

- Logical operator.

## ARITHMETIC OPERATORS

To perform basic mathematical operations such as addition, subtraction, or multiplication; combine numbers; and produce numeric results, use the following arithmetic operators

Arithmetic operator	Meaning	Example
+ (plus sign)	Addition	3+3
– (minus sign)	Subtraction or sign	3–1–1
* (asterisk)	Multiplication	3*3
/ (forward slash)	Division	3/3
^ (caret)	Exponentiation	16^4

## COMPARISON OPERATORS

You can compare two values with the following operators. When two values are compared by using these operators, the result is a logical value, either TRUE or FALSE

Comparison operator	Meaning	Example
=	Equal to	[Region] = "USA"
==	Strict equal to	[Region] == "USA"
>	Greater than	[Sales Date] > "Jan 2009"
<	Less than	[Sales Date] < "Jan 1 2009"
>=	Greater than or equal to	[Amount] >= 20000
<=	Less than or equal to	[Amount] <= 100
<>	Not equal to	[Region] <> "USA"

## TEXT CONCATENATION OPERATOR

Use the ampersand (&) to join, or concatenate, two or more text strings to produce a single piece of text.

Text operator	Meaning	Example
& (ampersand)	Connects, or concatenates, two values to produce one continuous text value	[Region] & ", " & [City]

## LOGICAL OPERATORS

Use logical operators (&&) and (||) to combine expressions to produce a single result.

Text operator	Meaning	Examples
&& (double ampersand)	Creates an AND condition between two expressions that each have a Boolean result. If both expressions return TRUE, the combination of the expressions also returns TRUE; otherwise the combination returns FALSE.	<code>(([Region] = "France") &amp;&amp; ([BikeBuyer] = "yes"))</code>
(double pipe symbol)	Creates an OR condition between two logical expressions. If either expression returns TRUE, the result is TRUE; only when both expressions are FALSE is the result FALSE.	<code>((([Region] = "France")    ([BikeBuyer] = "yes"))</code>
IN	Creates a logical OR condition between each row being compared to a table. Note: the table constructor syntax uses curly braces.	<code>'Product'[Color] IN { "Red", "Blue", "Black" }</code>



## ERROR HANDLING IN DAX

The **BLANK** function represents missing or undefined values in DAX calculations.

**ISBLANK** checks if a value is blank and returns TRUE if it is, otherwise FALSE.

**IFERROR** provides a way to handle errors by returning a specified value if an error occurs in a calculation

## FORMATING DAX FORMULAS

- Proper formatting improves the readability and maintainability of DAX code, making it easier to understand and debug.
- Use indentation and line breaks to separate different parts of the formula clearly.
- Add comments to explain complex logic and document your thought process.
- Consistent formatting practices help in collaborative environments and future proof your reports.



# **PRACTICAL IMPLEMENTATION OF DAX ON ECOMM SALES ANALYSIS PROJECT**

## AIM OF THE PROJECT—

The primary goal is to provide actionable insights into the e-commerce landscape, pinpointing areas for enhancement and growth through comprehensive analysis.

On having a look at aim of project and available dataset structure, you can decide the primary columns or measures needed during visualization process.

Here, you can create basic measures like –

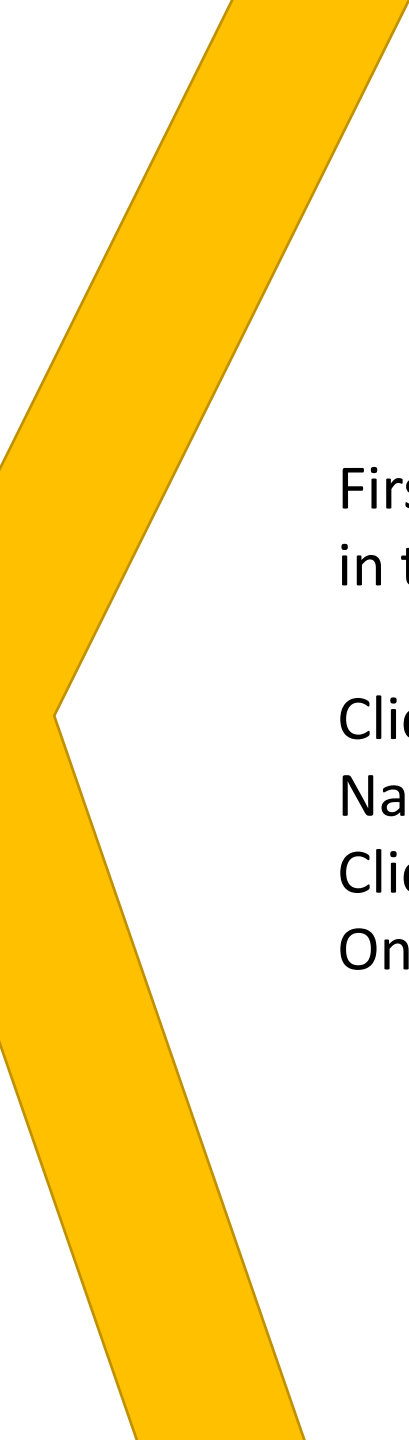
( YTD= YearToDate, YOY= YearOverYear, PYTD= PreviousYTD )

YTD sales, YTD Qty, YTD Profit, YTD Profit margin

YOY Sales, YOY Qty, YOY Profit, YOY Profit margin

To get YOY, you will need--

PYTD Sales, PYTD Qty, PYTD Profit, PYTD Profit margin




First of all, create a separate table for measures, so that it won't get confusing in the dataset tables. You can create them in existing tables itself.

Click on “Enter Data” under Home tab.

Name the table

Click Load

On right side in DATA panel, the new table is displayed.



Home Insert Modeling View Optimize Help

Cut Copy Format painter

Get data Excel workbook Data hub SQL Server Enter data Dataverse Recent sources Transform data Refresh data New visual

### Create Table

	Column1	+
1		
+		

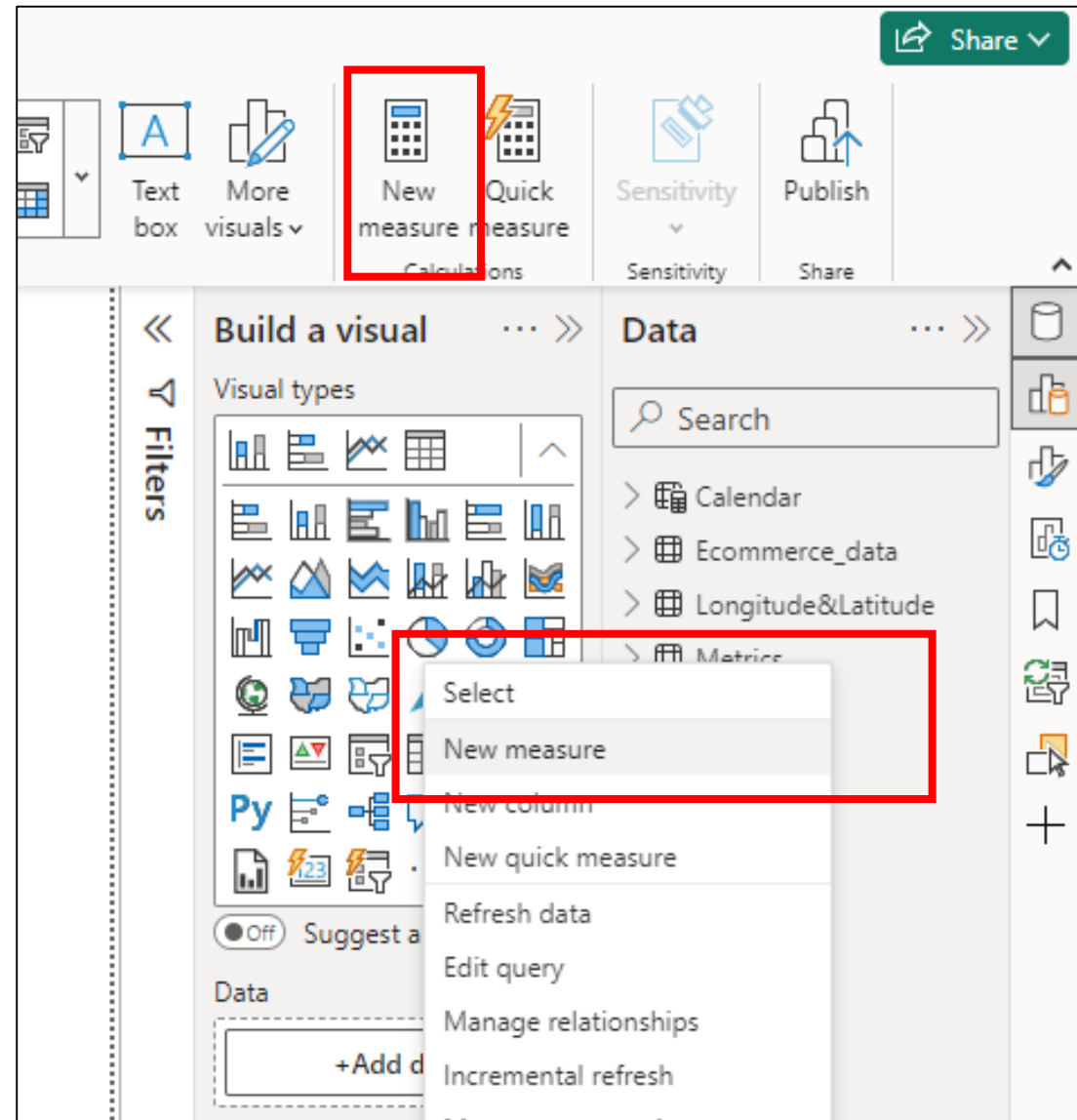
Name: Metrics

Load Edit Cancel

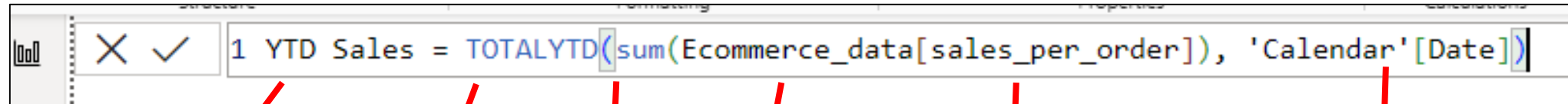
To create 1<sup>st</sup> measure  
right click on new table  
→ create measure

OR

Click on “New  
Measure” under Home  
tab



Write following query and press Enter



```
1 YTD Sales = TOTALYTD(sum(Ecommerce_data[sales_per_order]), 'Calendar'[Date])
```

Measure name

Aggregate function to  
perform on columnn

Date for YTD

Specify function

Referenced Table name

Referenced Column name

( First, it calculates the sum of the sales per order from the `ecmm\_data` table.  
Then, it takes this sum and computes the year-to-date total using the dates provided in the  
`Calendar[Date]` column.)



Following same process write other queries.

```
1 YTD Qty = TOTALYTD(sum(Ecommerce_data[order_quantity]), 'Calendar'[Date])
```

```
1 YTD Profit = TOTALYTD(sum(Ecommerce_data[profit_per_order]), 'Calendar'[Date])
```

```
1 Profit Margin = SUM(Ecommerce_data[profit_per_order])/SUM(Ecommerce_data[sales_per_order])
```

```
1 YTD Profit Margin = TOTALYTD([Profit Margin], 'Calendar'[Date])
```

```
1 PYTD Sales = CALCULATE(SUM(Ecommerce_data[sales_per_order]), DATESYTD(SAMEPERIODLASTYEAR('Calendar'[Date])))
```

```
1 PYTD Qty = CALCULATE(SUM(Ecommerce_data[order_quantity]), DATESYTD(SAMEPERIODLASTYEAR('Calendar'[Date])))
```

```
1 PYTD Profit = CALCULATE(SUM(Ecommerce_data[profit_per_order]), DATESYTD(SAMEPERIODLASTYEAR('Calendar'[Date])))
```

```
1 PYTD Profit Margin = CALCULATE([Profit Margin], DATESYTD(SAMEPERIODLASTYEAR('Calendar'[Date])))
```

1 YOY Sales = ([YTD Sales] - [PYTD Sales]) / [PYTD Sales]

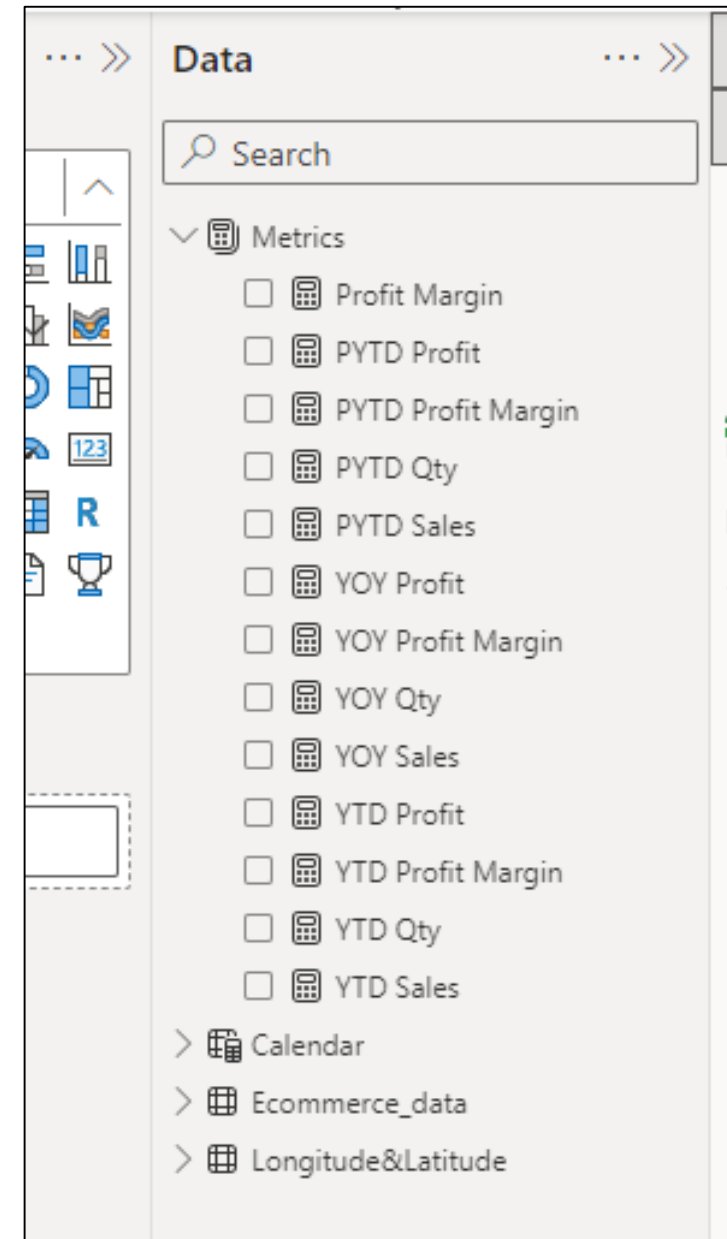
1 YOY Qty = ([YTD Qty] - [PYTD Qty]) / [PYTD Qty]


1 YOY Profit = ([YTD Profit] - [PYTD Profit]) / [PYTD Profit]

1 YOY Profit Margin = ([YTD Profit Margin] - [PYTD Profit Margin]) / [PYTD Profit Margin]

List of primary measures created.

Other measures can be created as per requirement during visualization process.





# **MICROSOFT POWER-BI**

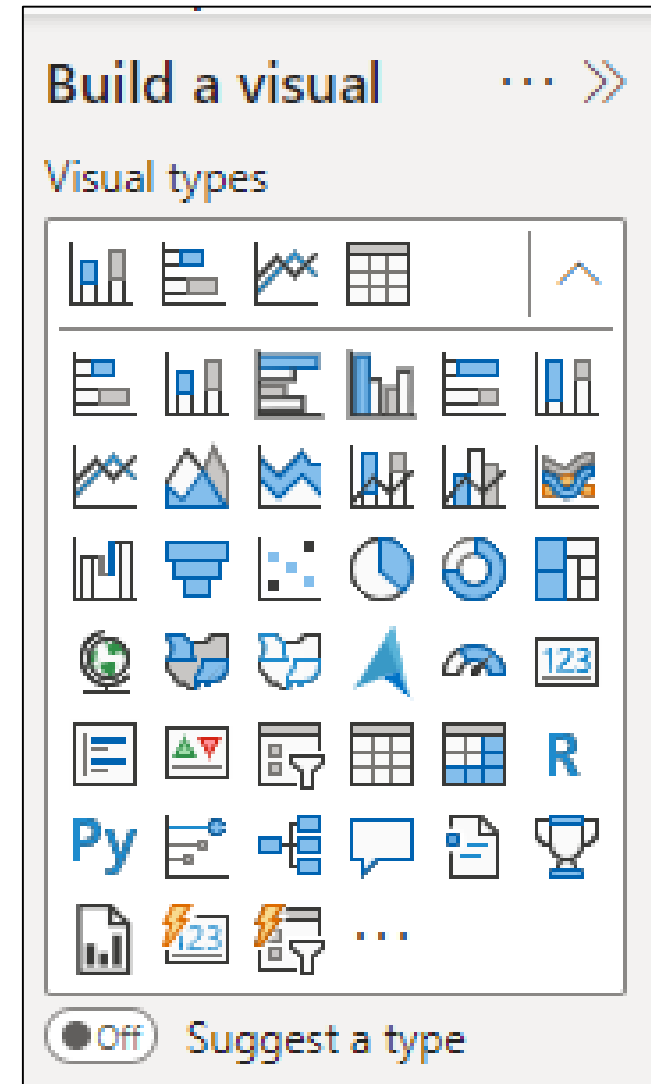
**SERIES 6**

## **VISUALIZATION**

**- MAYURI .D.**

# INTRO

- Visualizations, also called visuals for short, display insights that are discovered in the data.
- Data visualization is the process of converting raw information (text, numbers, or symbols) into a graphical format.
- A Power BI report might have a single page with one visual or it might have pages full of visuals. In the Power BI service, visuals can be pinned from reports to dashboards.
- There are many different visual types available directly from the Power BI Visualizations pane.



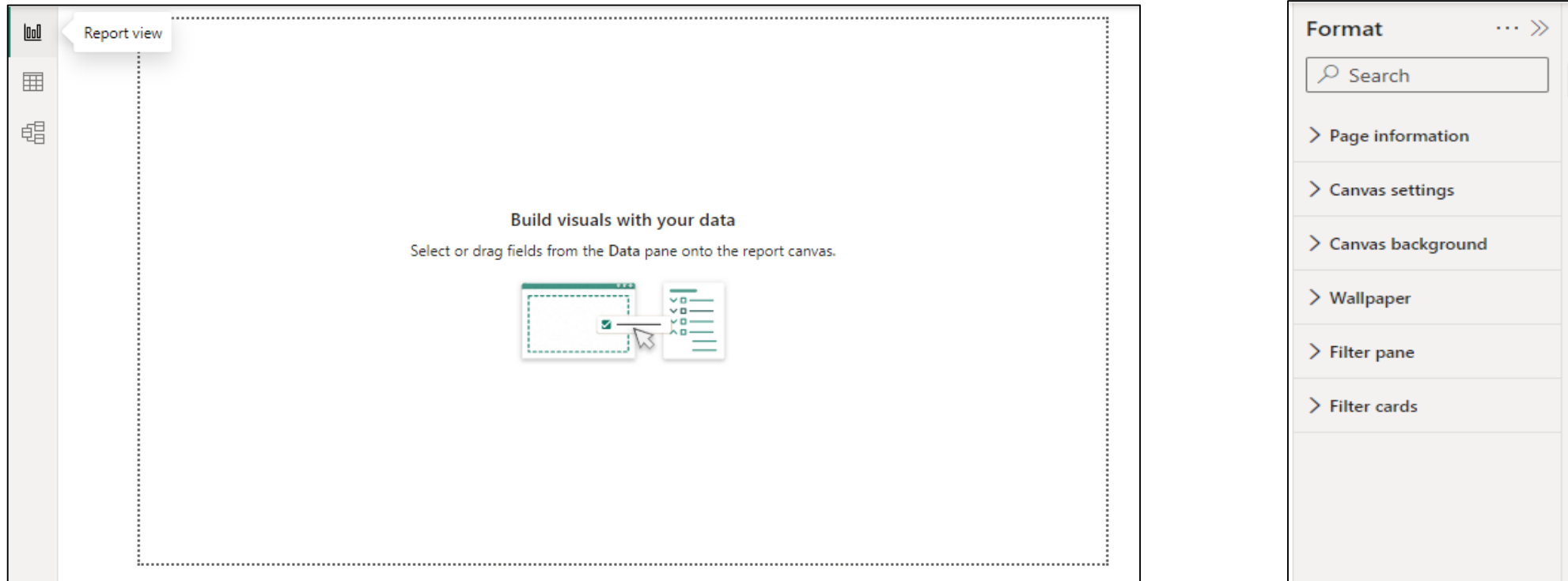
# DASHBOARD CANVAS

Open Report view from left sidebar.

Dotted square is the place called canvas to create the dashboards or reports.

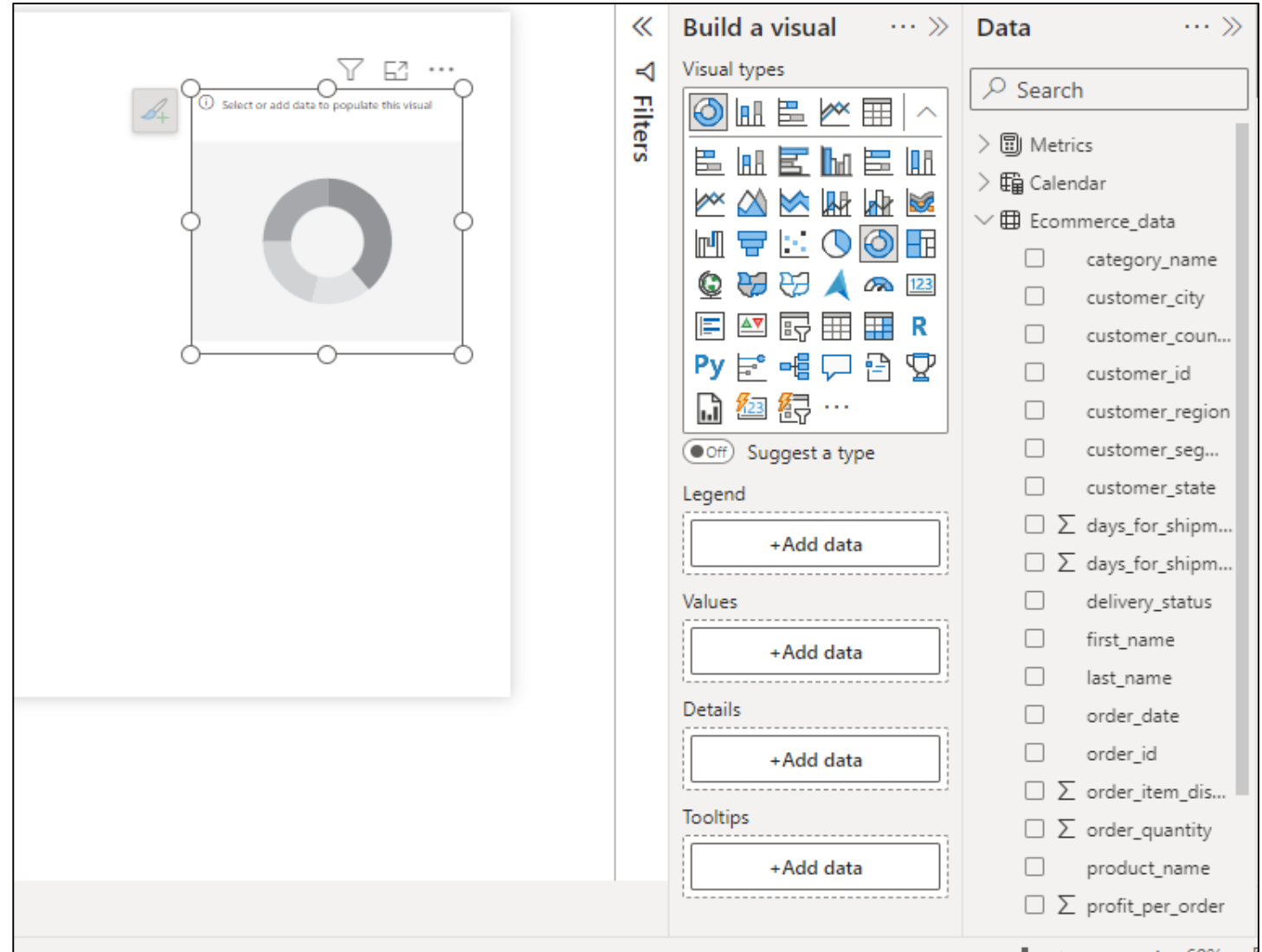
Format the canvas under FORMAT pane.

- Page information– name the page and select its type
- Canvas settings– size of report or select custom and define height width
- Canvas background– space inside the dotted square
- Wallpaper – space outside dotted square



# CREATE A VISUAL

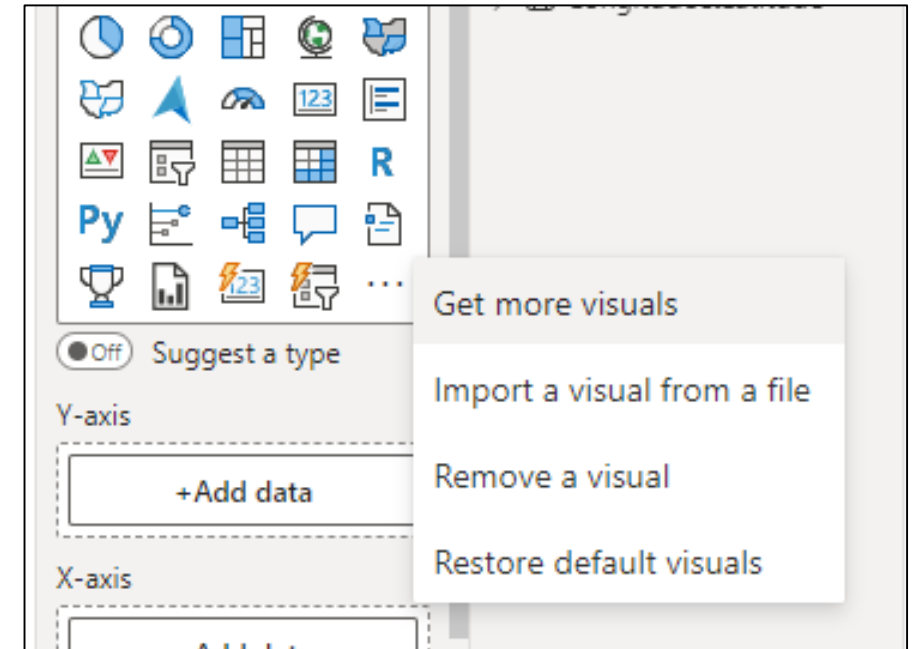
- Click on respected visual as per the required result.
- Add data fields gets activated
- Select the required columns from table, they automatically gets placed in the fields.  
(Sometimes need to replaced them manually by just drag and drop)





## ADD A VISUAL IN POWER-BI DESKTOP

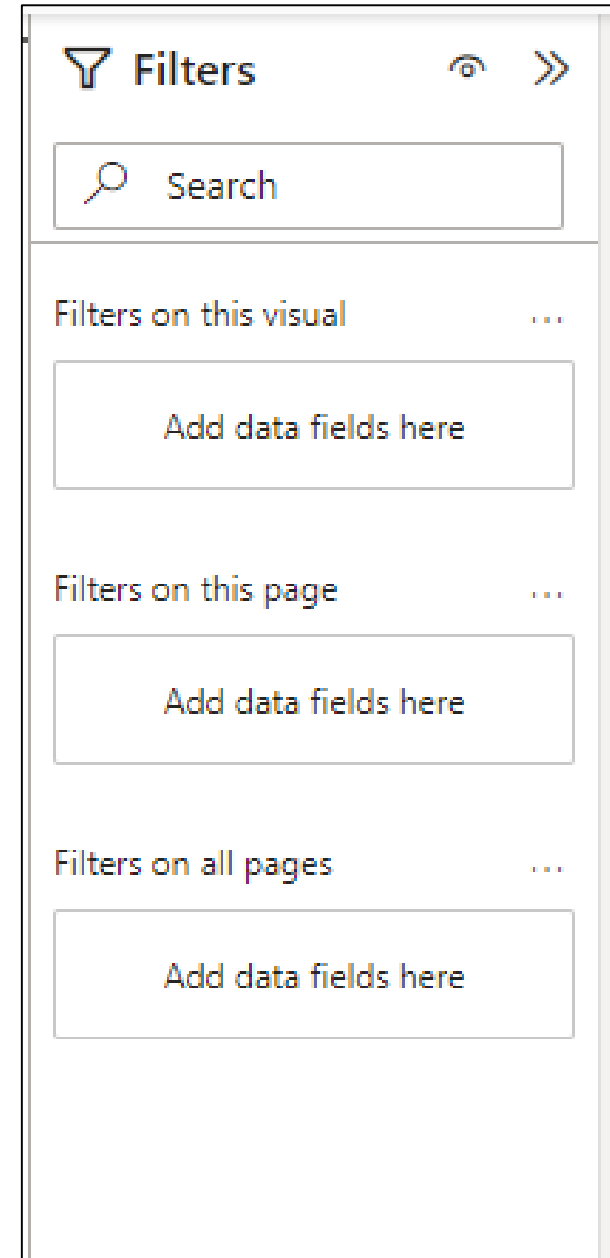
- Click on 3dots in visuals section under Visualization pane.
- Get more visuals: upload more visuals from Microsoft app-source. Once uploaded, right-click on resp. visual and pin it to visualization pane.
- Import a visual from a file : upload a visual from the file/folder from your device.
- Remove a visual : you can remove visuals.
- Restore default visuals: you can restore removed visuals.



# POWER BI DESKTOP FILTERS

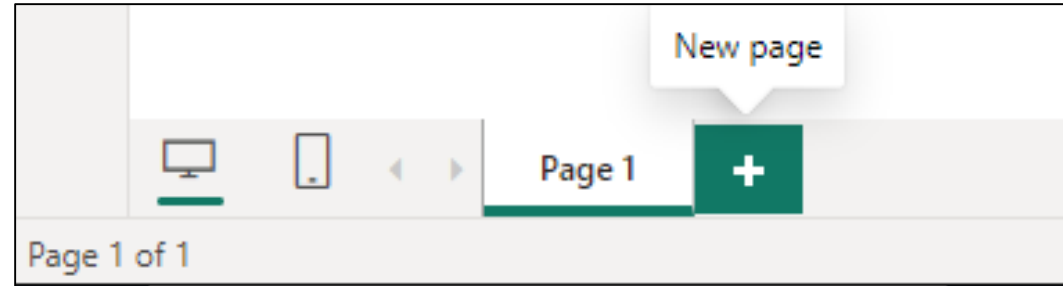
- Visual Level Filter: This gets applied to only to the active visual.
- Page Level Filter: This gets applied to all the visuals in the existing page.
- Report Level Filter: This gets applied to all the visuals in all the existing pages in the report.

(select any visual to activate visual level filter)



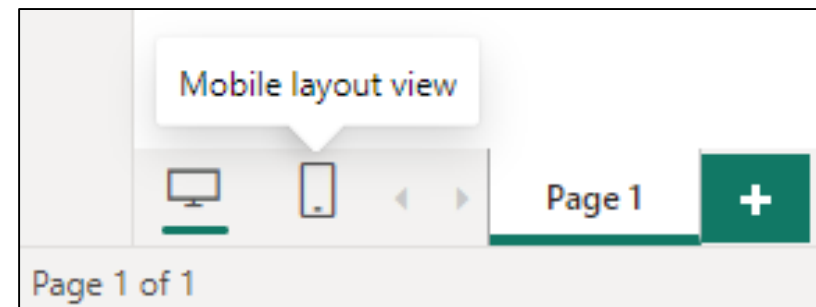
## ADD NEW PAGES TO REPORT

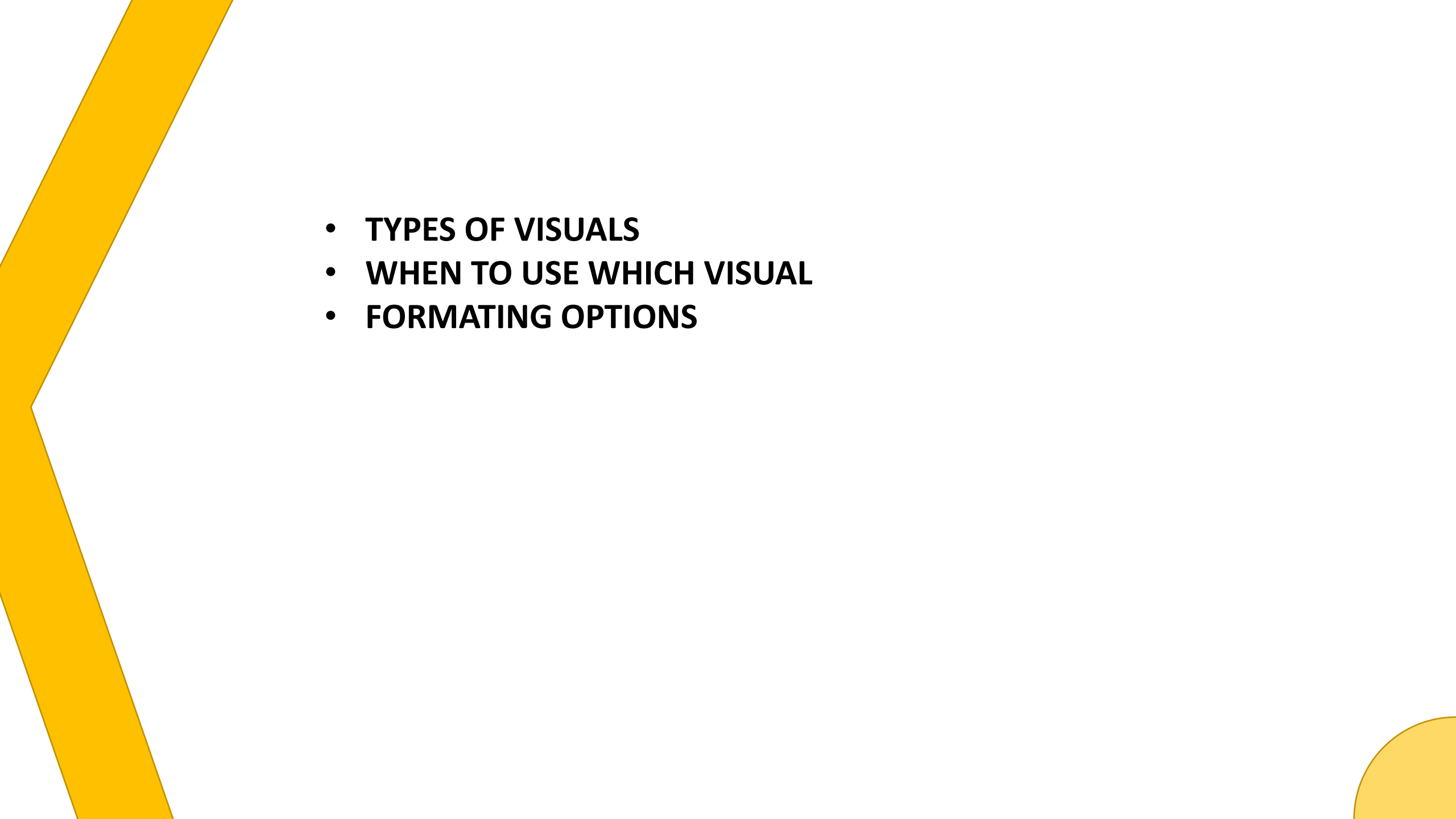
Click on + icon to add pages  
Double click on name to rename them.



## DESIGN MOBILE VIEW

By default desktop view is on.  
Click on mobile icon to view and edit mobile view of dashboard

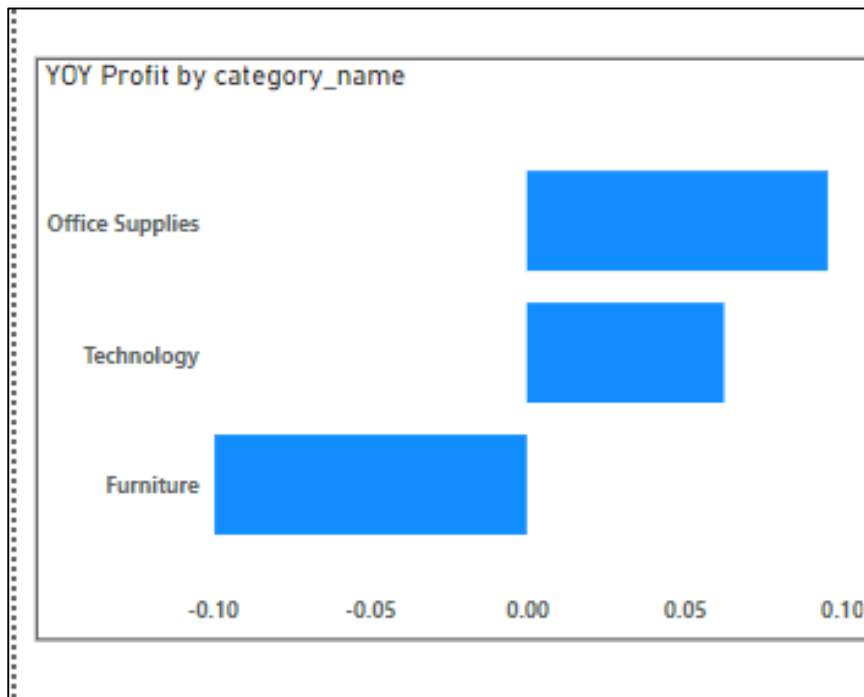


- 
- **TYPES OF VISUALS**
  - **WHEN TO USE WHICH VISUAL**
  - **FORMATING OPTIONS**

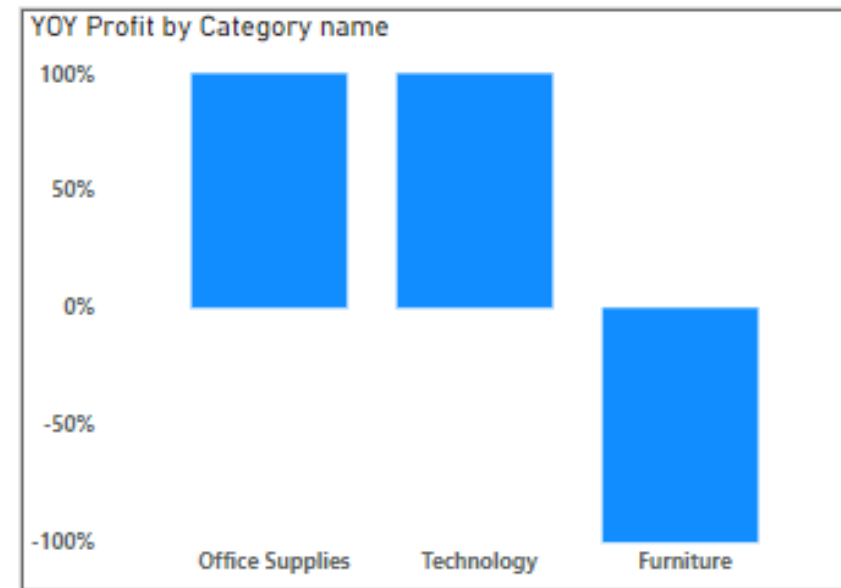
## COLUMN AND BAR CHARTS

A bar/column chart represents categorical data with rectangular bars, their heights being proportional to the values displayed.

BAR



COLUMN



Use a bar chart for the following reasons:

- compare two or more values in the same category
- compare parts of a whole
- don't have too many groups (less than 10 works best)
- understand how multiple similar data sets relate to each other

Don't use a bar chart for the following reasons:

- you're visualizing only has one value associated with it
- to visualize continuous data

key design best practices:

- Use consistent colors and labeling throughout so that you can identify relationships more easily
- Simplify the length of the y-axis labels and don't forget to start from 0 so you can keep your data in order

# COMMON FORMATTING OPTIONS

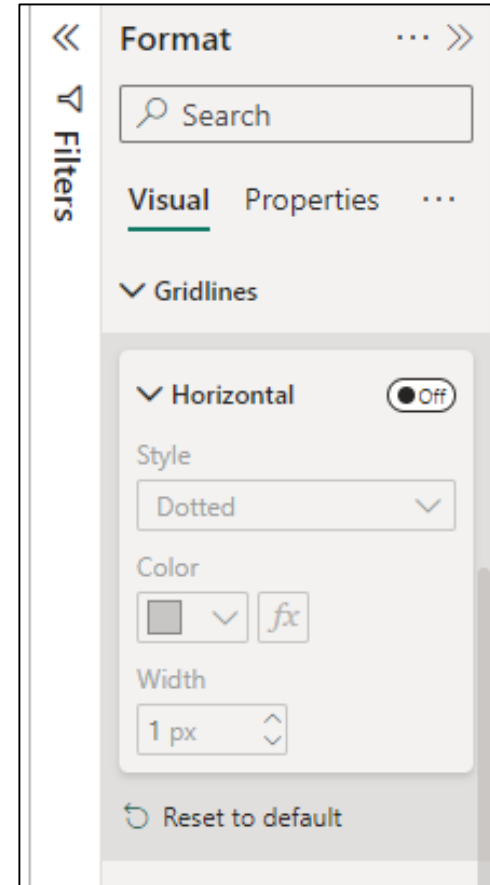
1. First remove background gridlines

Go to Format pane

→ visual

→ Gridlines

→ turn off horizontal/vertical



## 2. Apply border and background

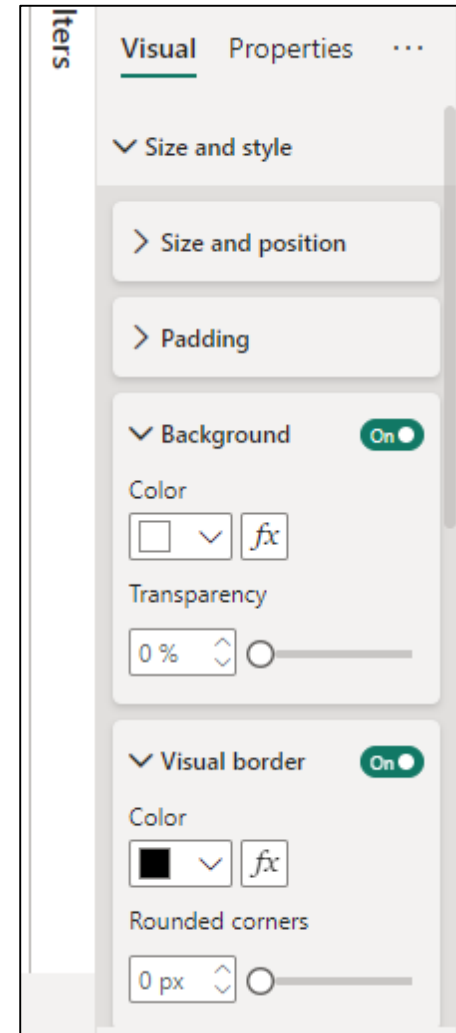
Go to Format pane

→ visual

→ size and style

→ turn ON

→ Select your color

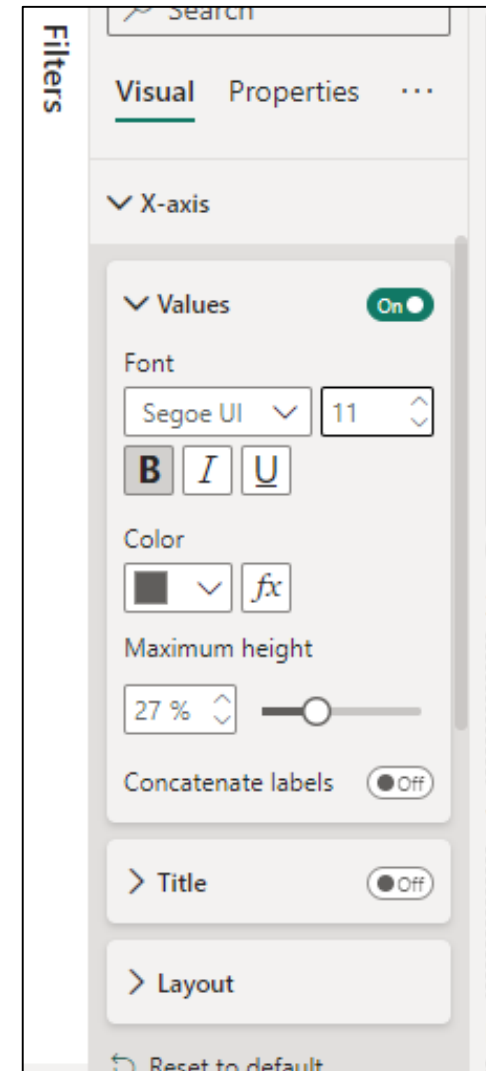




### 3. Edit X-axis and Y-axis

Go to Format pane

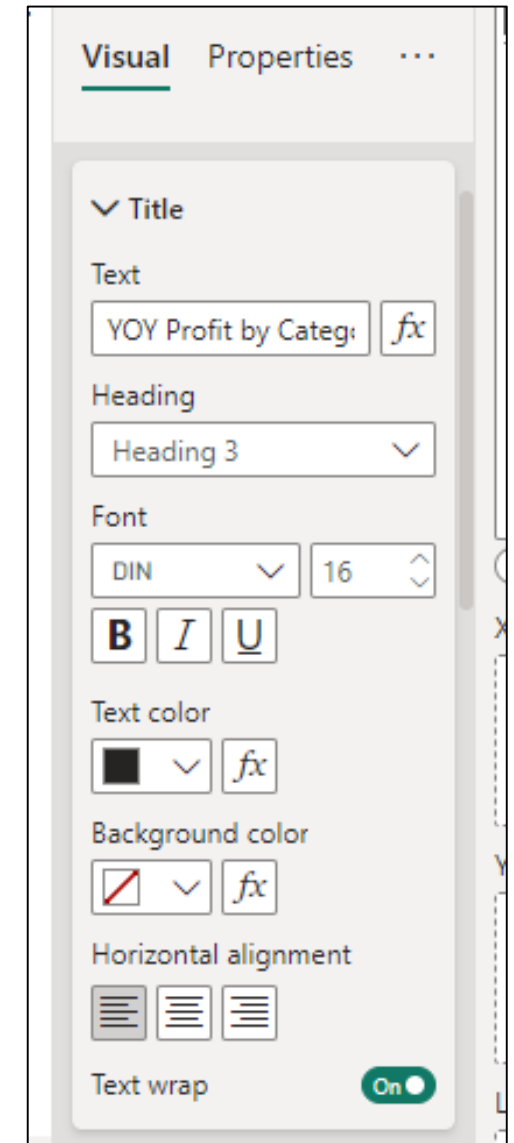
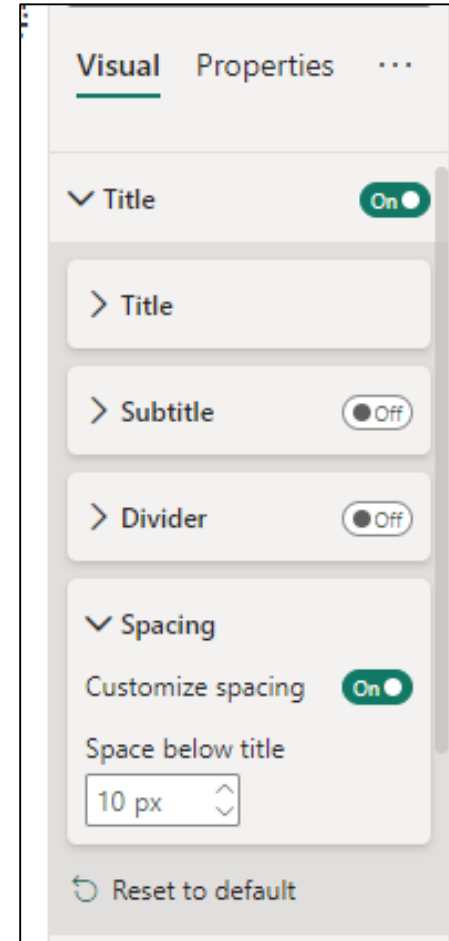
- visual
- X-axis/Y-axis
- turn ON/OFF values and title (as per requirement)
- Select font, size, bold, color, etc



#### 4. Edit Title. Add subtitle, divider

Go to Format pane

- visual
- Title
- turn ON/OFF title, subtitle, divider, spacing (as per requirement)
- Edit your title
- Select font, size, bold, color, etc



## 5. Data Labels (add values on the bars)

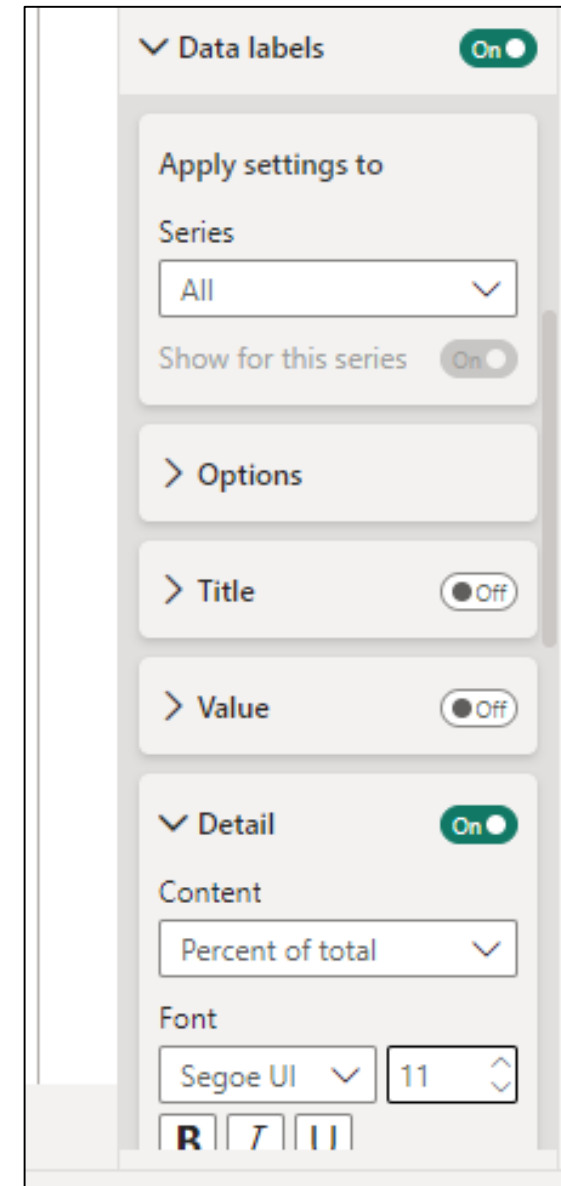
Go to Format pane

→ visual

→ Data Labels

→ turn ON/OFF Edit your title

→ Select font, size, bold, etc



6. Color each bars with different colors

Go to Format pane

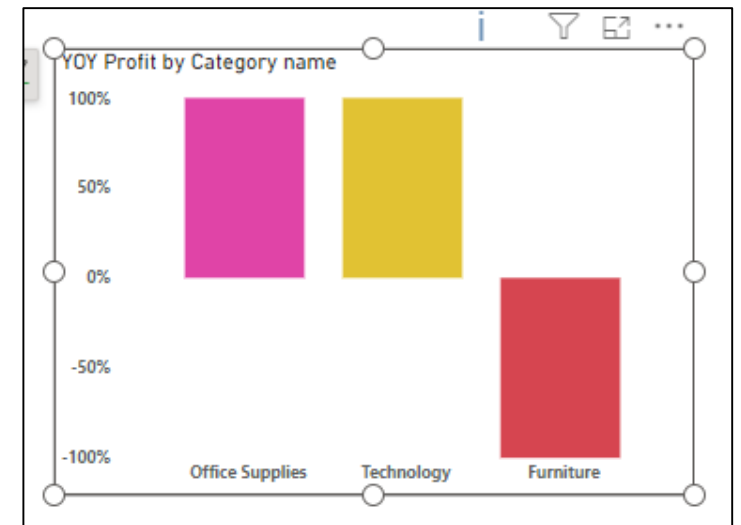
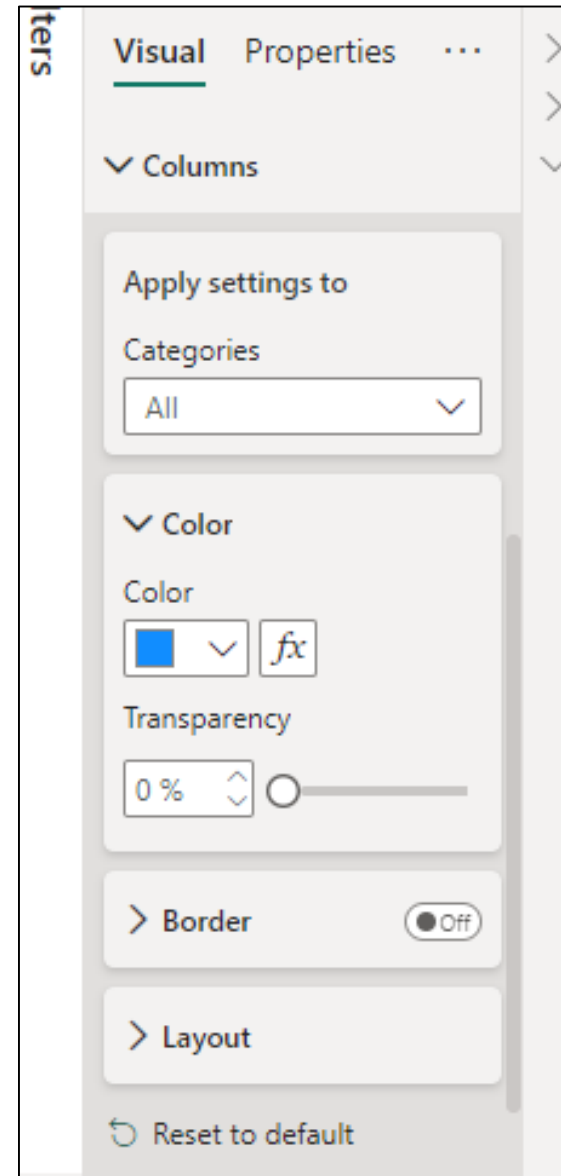
→ visual

→ Columns/ bars

→ Select bar category

→ Select color with 0 transparency

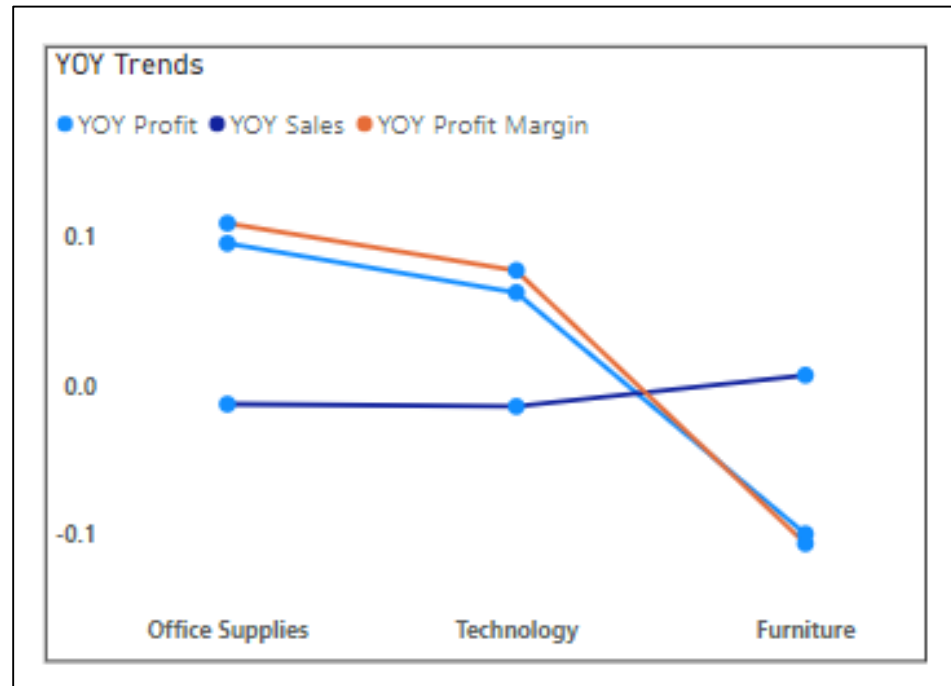
(select bars one by one to define colors, select ALL to apply one color to all the bars.)



## LINE CHART

Line charts are used to show resulting data relative to a continuous variable - most commonly time or money.

The proper use of color in this visualization is necessary because different colored lines can make it even easier for users to analyze information.






Use a line chart for the following reasons:

- understand trends, patterns, and fluctuations in your data
- compare different yet related data sets with multiple series
- make projections beyond your data

Don't use a line chart for the following reason:

- demonstrate an in-depth view of your data

key design best practices:

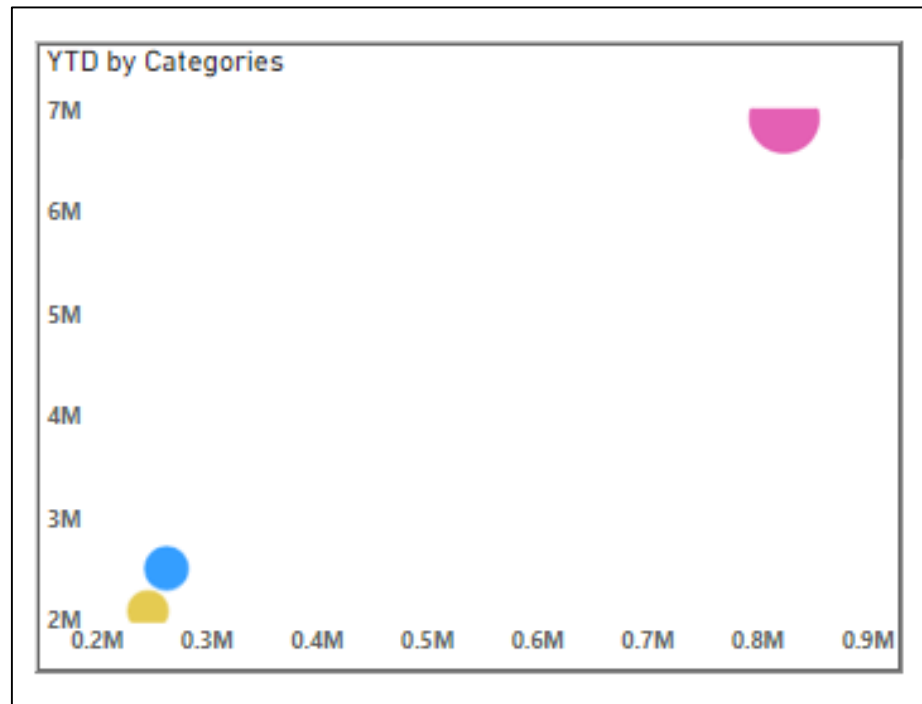
- Along with using a different color for each category you're comparing, make sure you also use solid lines to keep the line chart clear and concise
  - To avoid confusion, try not to compare more than 4 categories in one line chart
- 

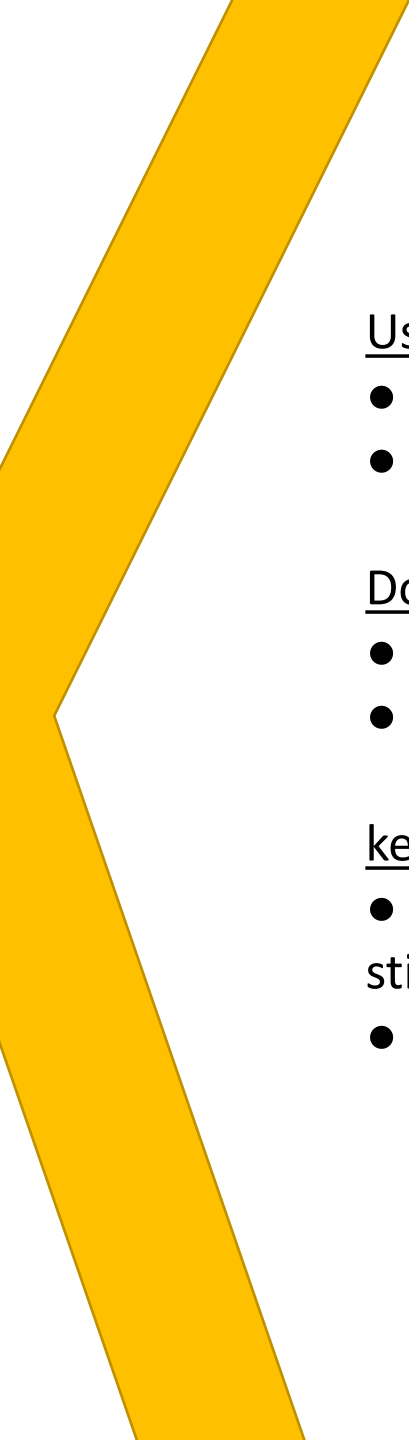
## SCATTER CHART

Scatterplots are the right data visualizations to use when there are many different data points, and you want to highlight similarities in the data set.

If the data forms a band extending from lower left to upper right, there most likely a positive correlation between the two variables.

If the band runs from upper left to lower right, a negative correlation is probable. If it is hard to see a pattern, there is probably no correlation.






Use a scatterplot for the following reasons:

- show the relationship between two variables
- compact data visualization

Don't use a scatterplot for the following reasons:

- rapidly scan information
- clear and precise data points

key design best practices:

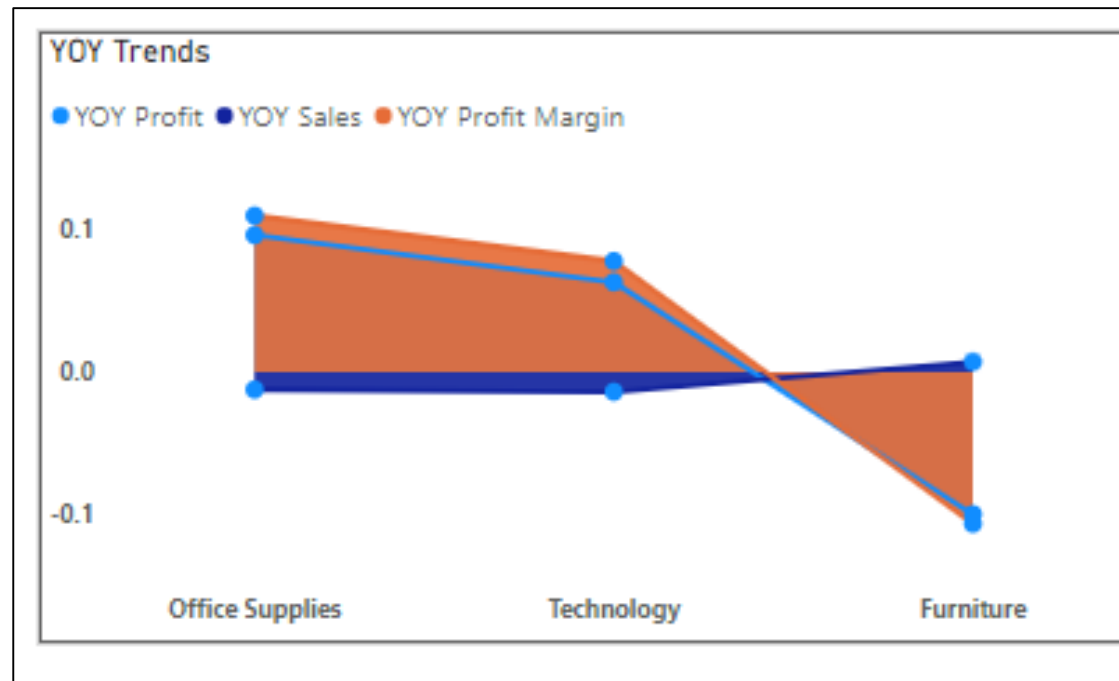
- Although trend lines are a great way to analyze the data on a scatterplot, ensure you stick to 1 or 2 trend lines to avoid confusion
  - Don't forget to start at 0 for the y-axis
- 



## AREA CHART

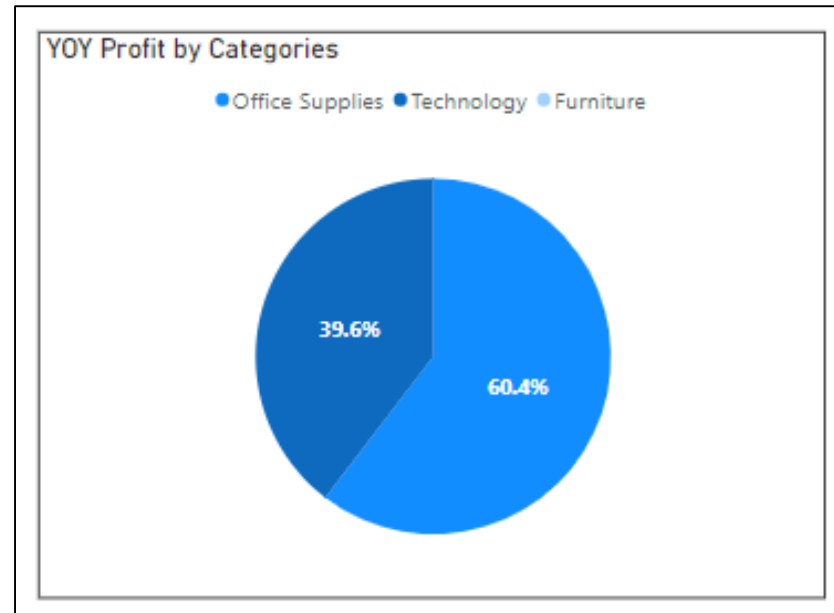
An area chart is very similar to a line graph but may do a better job of highlighting the relative differences between items.

Use an area chart when you want to see how different items stack up or contribute to the whole.



## PIE CHART

Pie charts are an interesting graph visualization. At a high-level, they're easy to read and understand because the parts-of-a-whole relationship is made very obvious



Use a pie chart for the following reasons:

- compare relative values
- compare parts of a whole
- rapidly scan metrics

Don't use a pie chart for the following reason:

- precisely compare data

key design best practices:

- Make sure that the pie slices add up to 100%. To make this easier, add the numerical values and percentages to your pie chart
- Order the pieces of your pie according to size
- Use a pie chart if you have only up to 5 categories to compare. If you have too many categories, you won't be able to differentiate between the slices

## TABLE

If you're someone who wants a little bit of everything in front of you to make thorough decisions, then tables are the visualization to go with. Tables are great because you can display both data points and graphics, such as bullet charts, icons, and sparklines. This visualization type also organizes your data into columns and rows, which is great for reporting

YOY Trends				
category_name	YOY Profit	YOY Sales	YOY Profit Margin	YOY Qty
Office Supplies	0.10	-0.01	0.11	-0.07
Technology	0.06	-0.01	0.08	-0.08
Furniture	-0.10	0.01	-0.11	-0.09

Use a table for the following reasons:

- You want to display two-dimensional data sets that can be organized categorically
- You can drill down to break up large data sets with a natural drill-down path

Don't use a table for the following reason:

- You want to display large amounts of data

key design best practices:

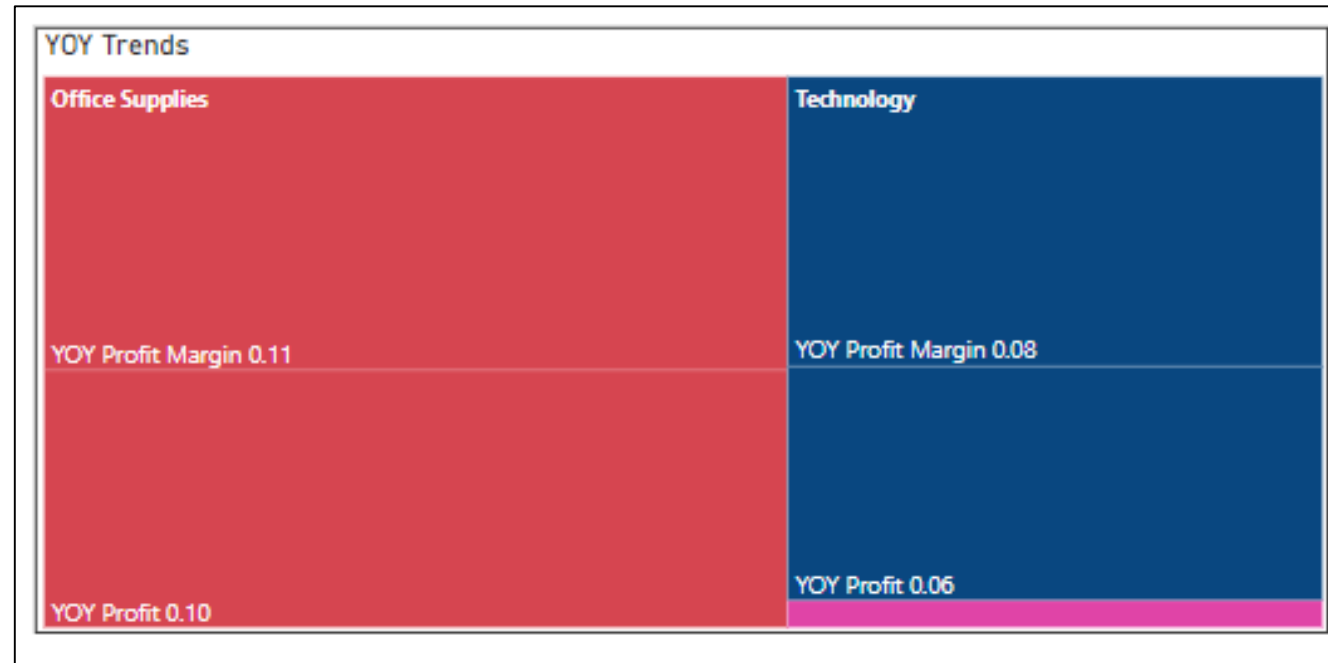
- Be mindful of the order of the data. Make sure that labels, categories, and numbers come first then move on to the graphics
- Try not to have more than 10 different rows on your table to avoid clutter

## TREE MAP

A treemap is a visual tool that can be used to break down the relationships between multiple variables in your data.

They can be used strictly as a presentation vehicle to show how your products roll up into different categories, for example.

A treemap can be broken down into 2-3 different layers to show the hierarchical relationship between items.





Use a treemap for the following reasons:

- display hierarchical data
- proportion of each category as a whole.

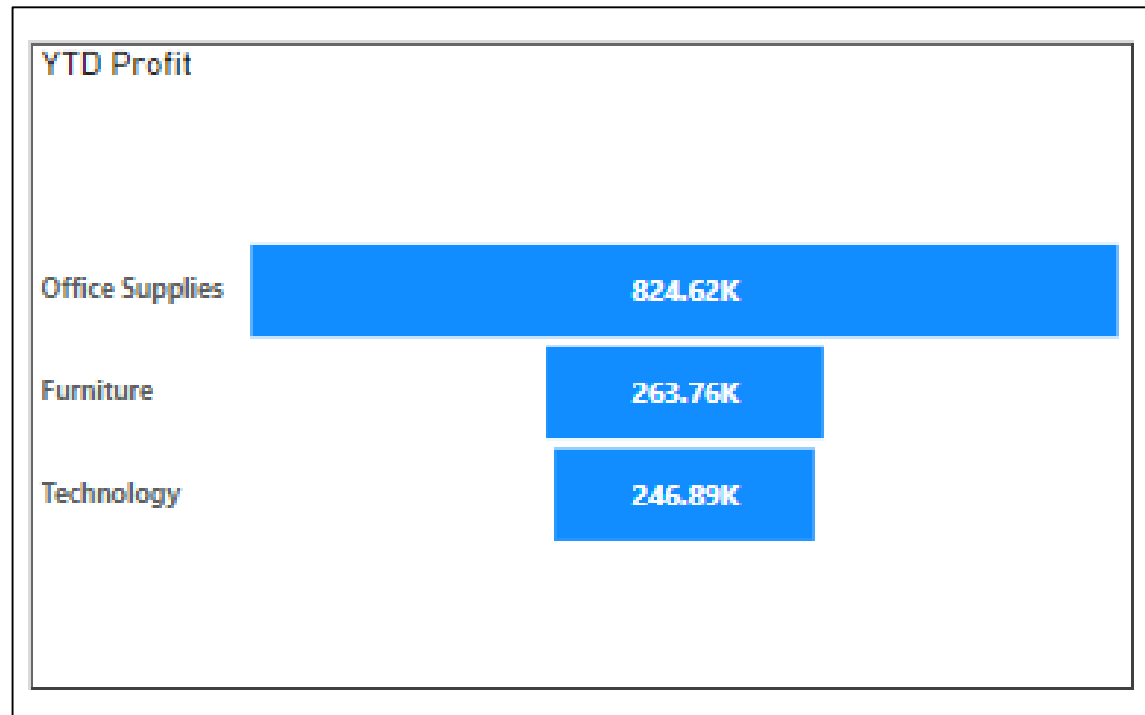
Don't use a treemap for the following reason:

- there are too many categories or the data is not hierarchical
- 

# FUNNEL

A funnel chart is your data visualization of choice if you want to display a series of steps and the completion rate for each step.

This can be used to track the sales process, a marketing funnel or the conversion rate across a series of pages or steps.








Use a funnel chart for the following reason:

- To display a series of steps and each step's completion rate

Don't use a funnel chart for the following reason:

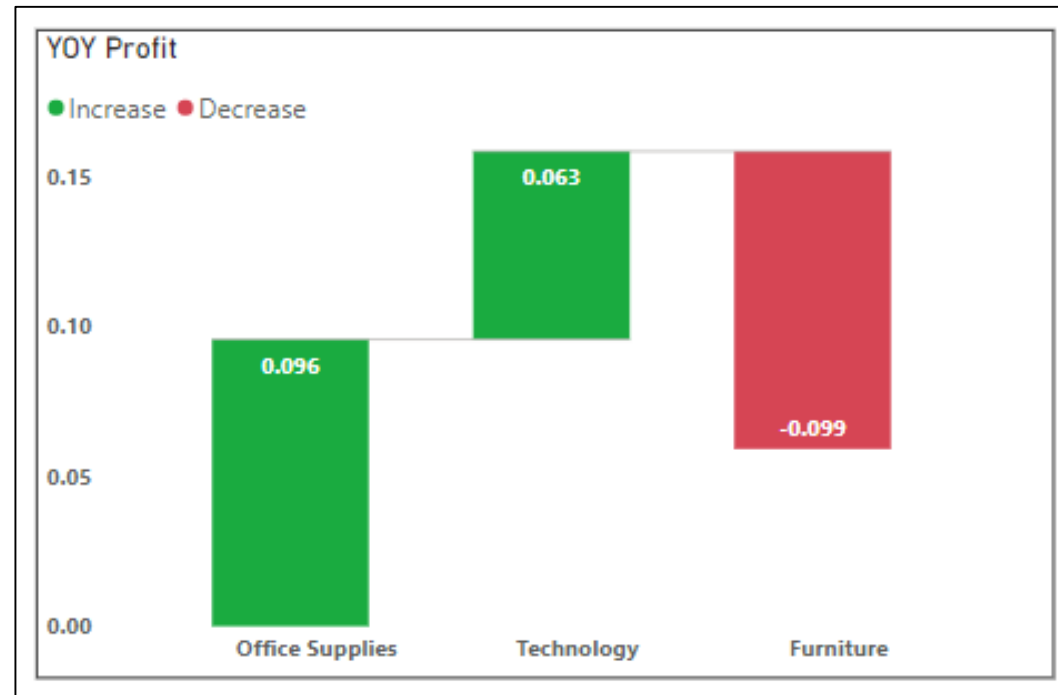
- To visualize individual, unconnected metrics

key design best practices:

- Scale the size of each section to accurately reflect the size of its data set
  - Use contrasting colors or one color in gradating hues, from darkest to lightest as the size of the funnel decreases
- 

## WATERFALL CHART

A waterfall chart is an information visualization that should be used to show how an initial value is affected by intermediate values and resulted in a final value. The values can be either negative or positive.






Use a waterfall chart for the following reason:

- To reveal the composition or makeup of a number

Don't use a waterfall chart for the following reason:

- You want to focus on more than one number or metric

key design best practices:

- Use contrasting colors to highlight differences in data sets
  - Choose warm colors to indicate increases and cool colors to indicate decreases
- 

## REPORT INTERACTIONS

By default, all the visualizations are connected to each other and filtering items in one visual will impact others too.

Through “Edit interactions” we can prevent certain visualizations to get filtered.

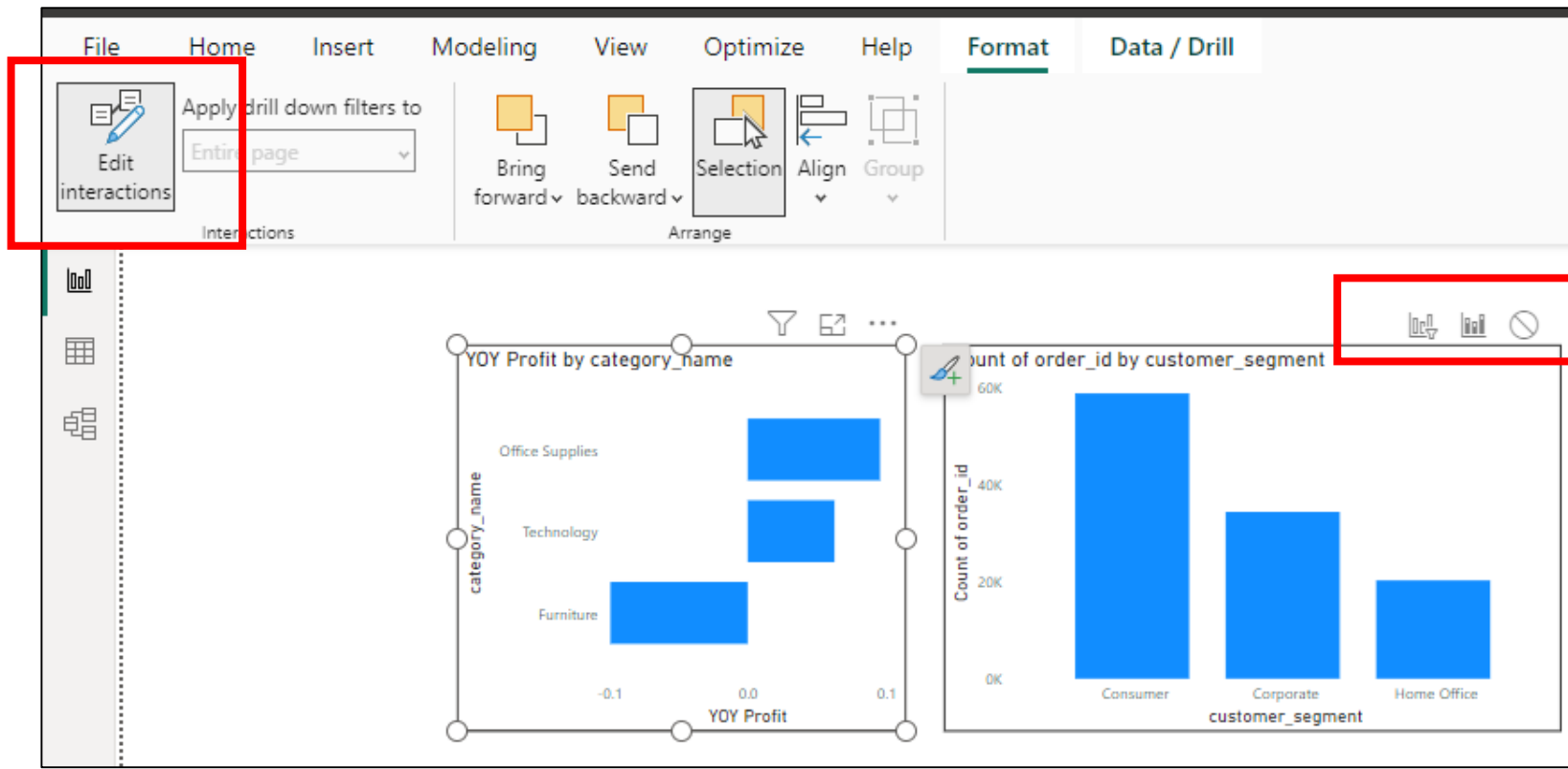
Here, when click on “office supplier” on bar chart, other chart changes color based on selected category. If you don’t want other charts to change color, use “Edit interactions”.



Remove filter: Select chart → format → edit interaction → none → edit interaction

Apply filter(no color change): Select chart → format → edit interaction → filter → edit interaction

Original (color change): Select chart → format → edit interaction → highlight → edit interaction



3 Options

- Filter
- Highlight
- None



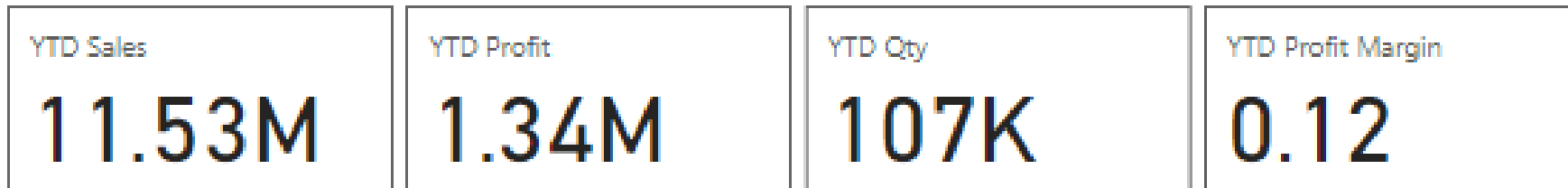
# **PRACTICAL IMPLEMENTATION OF ECOMM SALES ANALYSIS PROJECT**

To get the overall idea of a Ecomm sales, certain important metrics needs to be displayed.

This is achieved with the help of visual called “card” or “card(new)”.

Here, card(new) visual is used, displaying below metrics-

- YTD sales
- YTD profit
- YTD Qty
- YTD profit margin

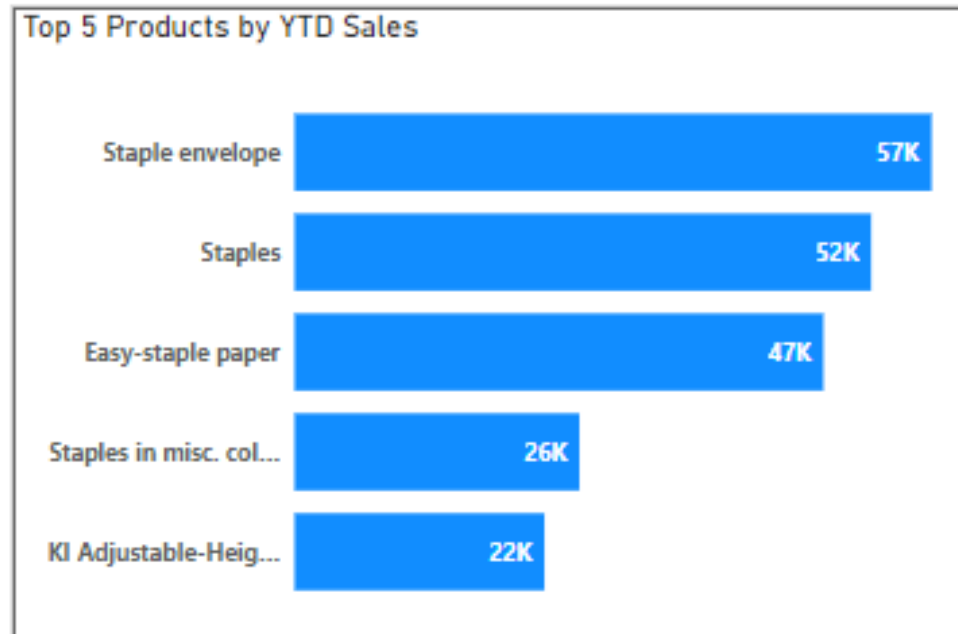


Format the font , size , etc under format pane. (this visual is still under process so it provides limited formatting options)

(You can even use “card” visual, you will need to edit each card separately or simply edit one card and do “format painter” to other cards so that all cards gets edited in one go)

To display Top 5 products by YTD sales

Use bar chart and insert fields – product name and YTD sales  
In filter pane select type top N , items as top 5, by value YTD sales click apply filter



**Filters**

Search

Filters on this visual

product\_name is (All)

Filter type ①

Top N

Show items

Top 5

By value

YTD Sales

Apply filter

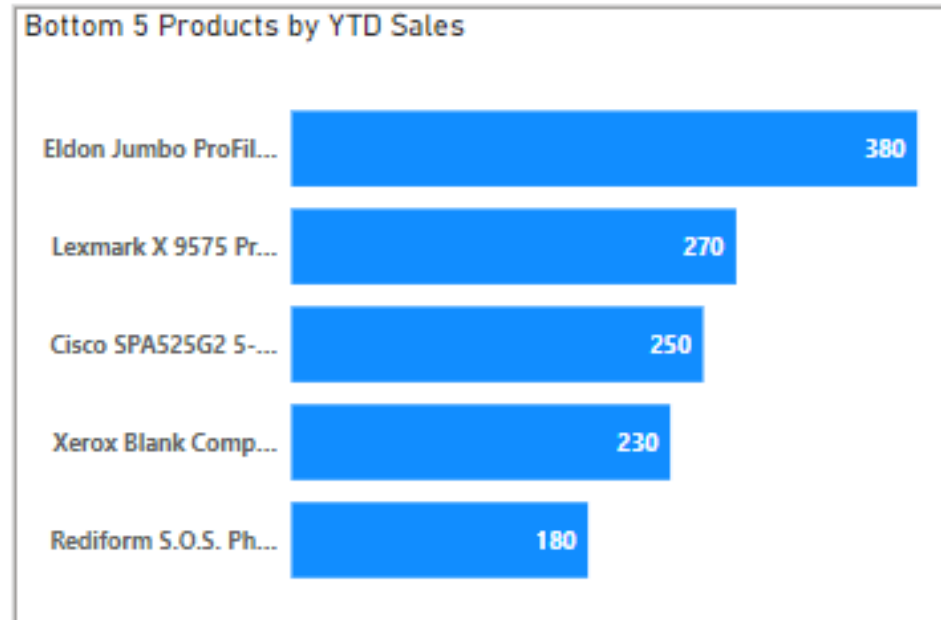
YTD Sales is (All)

Add data fields here



To display Bottom 5 products by YTD sales

Copy and paste Top 5 products bar chart.  
In filter change items from top to bottom.  
Edit chart title.



**Filters**

Search

Filters on this visual ...

**product\_name**  
bottom 5 by YTD Sales

Filter type ⓘ  
Top N

Show items  
Bottom 5

By value  
YTD Sales

Apply filter

YTD Sales  
is (All)

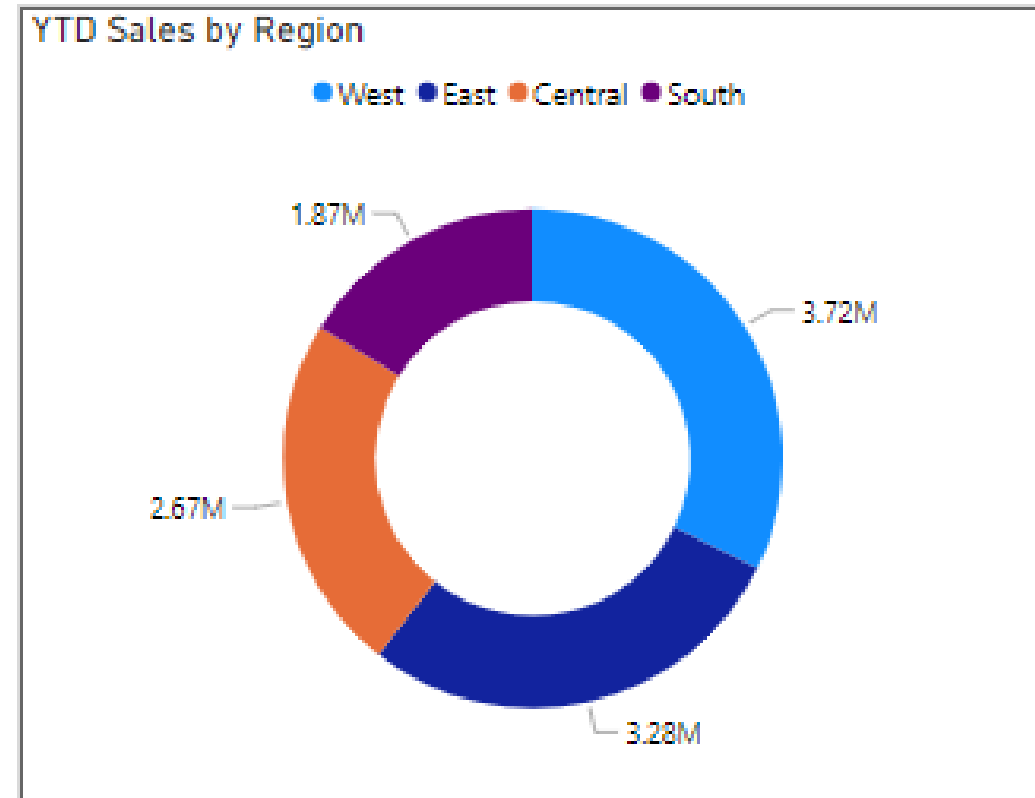
Add data fields here

To display YTD sales by Regions

Select Donut chart

Take fields customer region and YTD sales

Edit the chart in Format pane

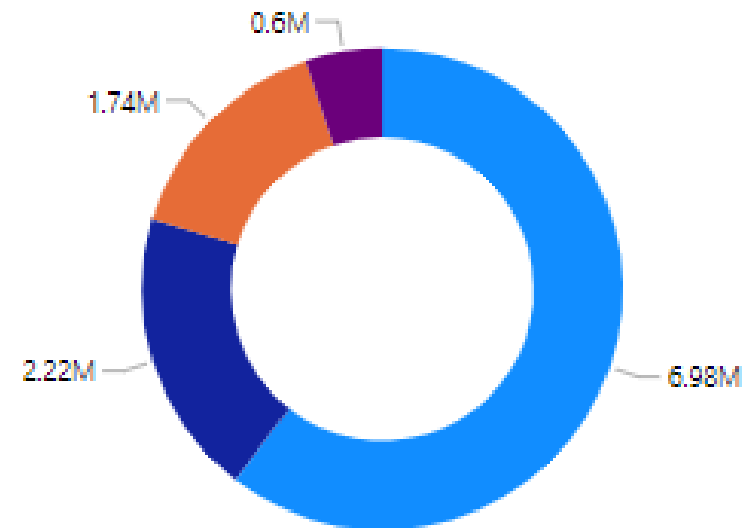


To display YTD sales by Shipping Type

Copy paste previous donut chart.  
Remove regions field and select shipping  
type.  
Edit title

YTD Sales by Shipping Type

Standard Class Second Class First Class Same Day



## Display tabular information

- Select table visual, select fields YTD sales, PYTD sales, YOY sales
- Edit the decimal pt. and unit in “format pane” → specific column
- To display \$ / %, click on measure name in “Data pane”, “measure tools” tab opens select the format

▼ Specific column

Apply settings to

Series

YTD Sales \$ ▼

Apply to header ☐ Off


Apply to subtotals ☐ Off

Apply to total ☐ Off

Apply to values ☒ On


▼ Values

Text color


 ▼

▼ Values




Text color

 ▼

Background color

 ▼

Alignment

Display units

Millions ▼

Value decimal places

2 ▲ ▼

Reset to default

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

Name YOY Sales

Home table Metrics ▼

Structure

\$% Format Percentage ▼

\$ ▼ % , .00 2 ▲ ▼

Formatting

Data category Uncategorized ▼

Properties

New measure Quick measure

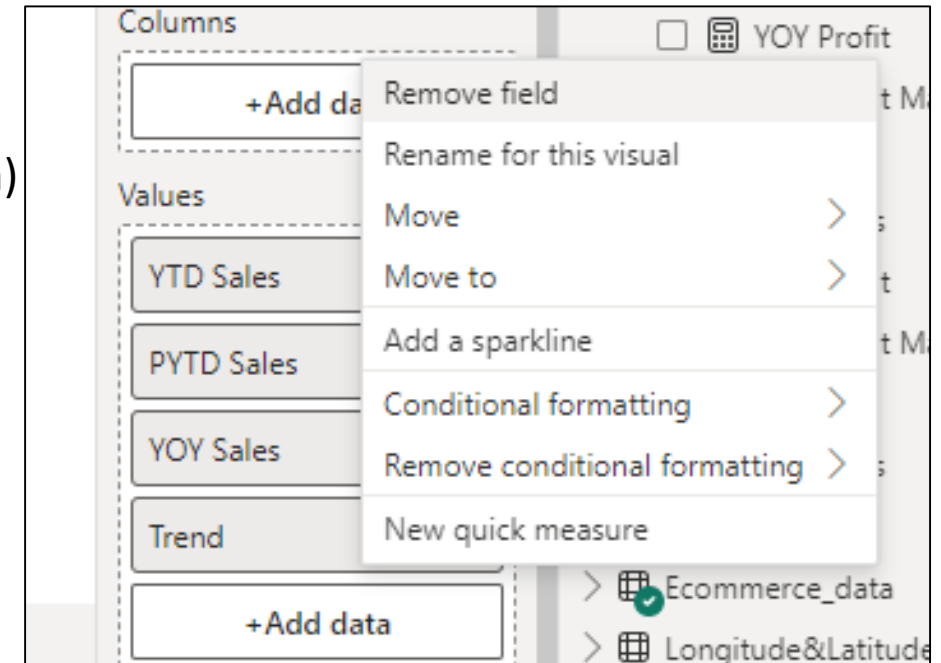
Calculations

- Create 2 new measures—

```
Trend = var positive_icon = UNICHAR(9650)
      var negative_icon = UNICHAR(9660)
      var result = IF([YOY Sales]>0, positive_icon, negative_icon)
      return result
```

```
Sales Colour = IF([YOY Sales]>0, "Green", "Red")
```

- Drag trend measure in fields
- Right click on trend → conditional formatting → font color



## Font color - Trend



Format style

Apply to

Rules

Values only

What field should we base this on?

Sales Colour

Rules

↕ Reverse color order

+ New rule

If value

is

Green

then



↑ ↓ ×

If value

is

Red

then



↑ ↓ ×

category_name	YTD Sales	PYTD Sales	YOY Sales	Trend
---------------	-----------	------------	-----------	-------

Furniture	\$2.52M	\$2.50M	0.73%	▲
-----------	---------	---------	-------	---

Office Supplies	\$6.92M	\$7.00M	-1.22%	▼
-----------------	---------	---------	--------	---

Technology	\$2.10M	\$2.13M	-1.37%	▼
------------	---------	---------	--------	---

Create Map

Select Map visual

Select fields as shown

(map visual works only when network connectivity is ON)



**Build a visual** ... >>

☐ Off Suggest a type

Location

+Add data

Legend

customer\_region X | >

Latitude

latitude X | >

Longitude

longitude X | >

Bubble size

YTD Sales X | >

Tooltips

YOY Sales X | >

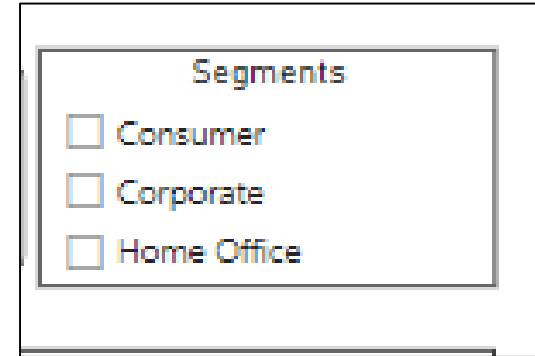
First first\_name X | >

+Add data

## Create Filter

Select Slicer visual

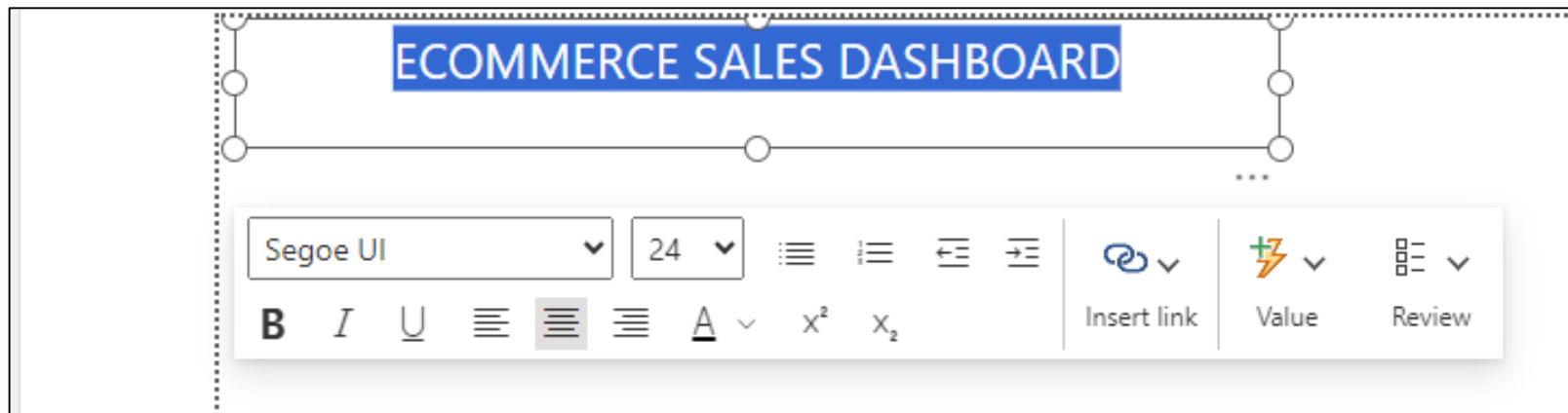
Select fields as “customer segment”



## Add dashboard title

Select Text box under Home tab

Name your dashboard

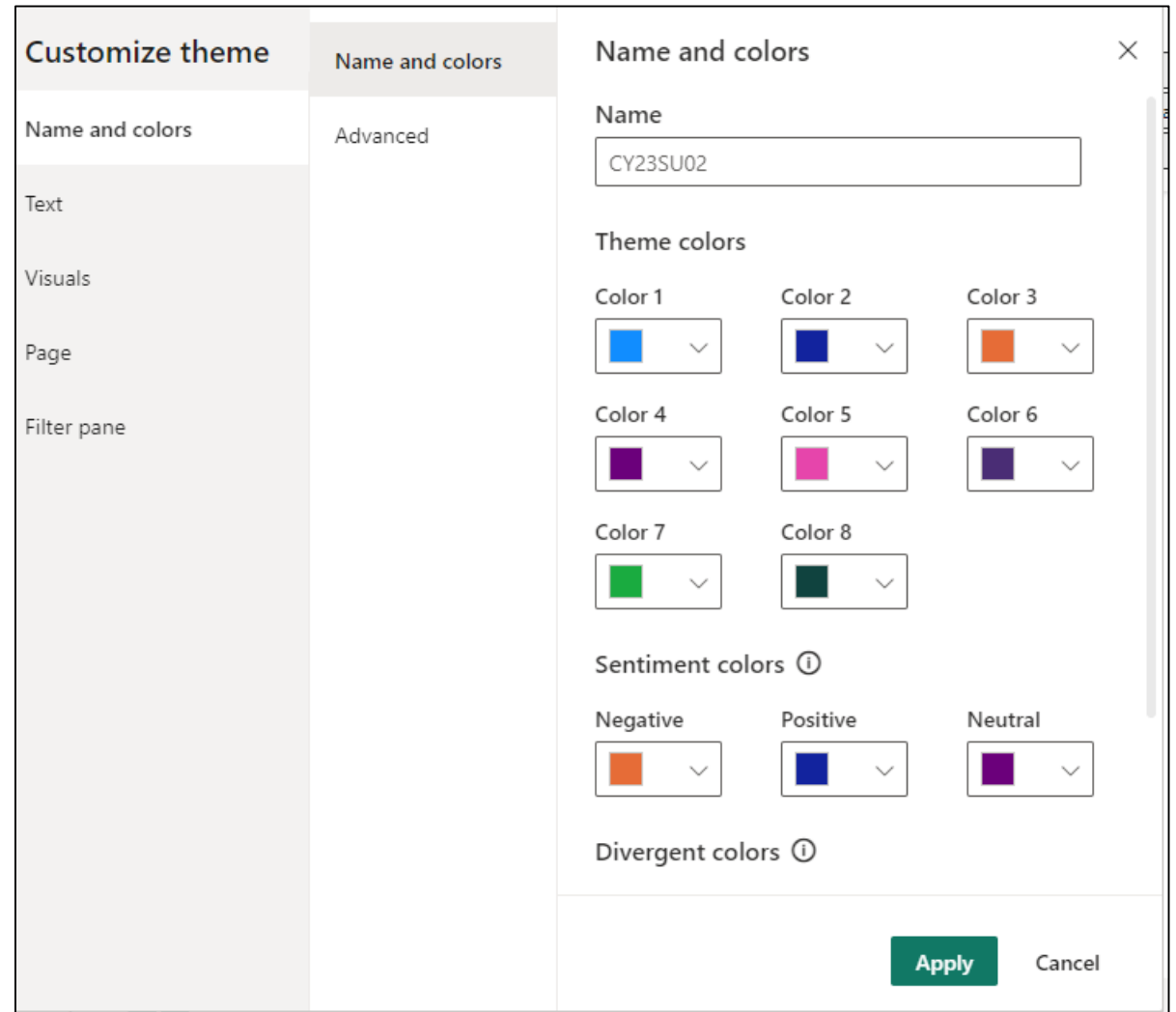
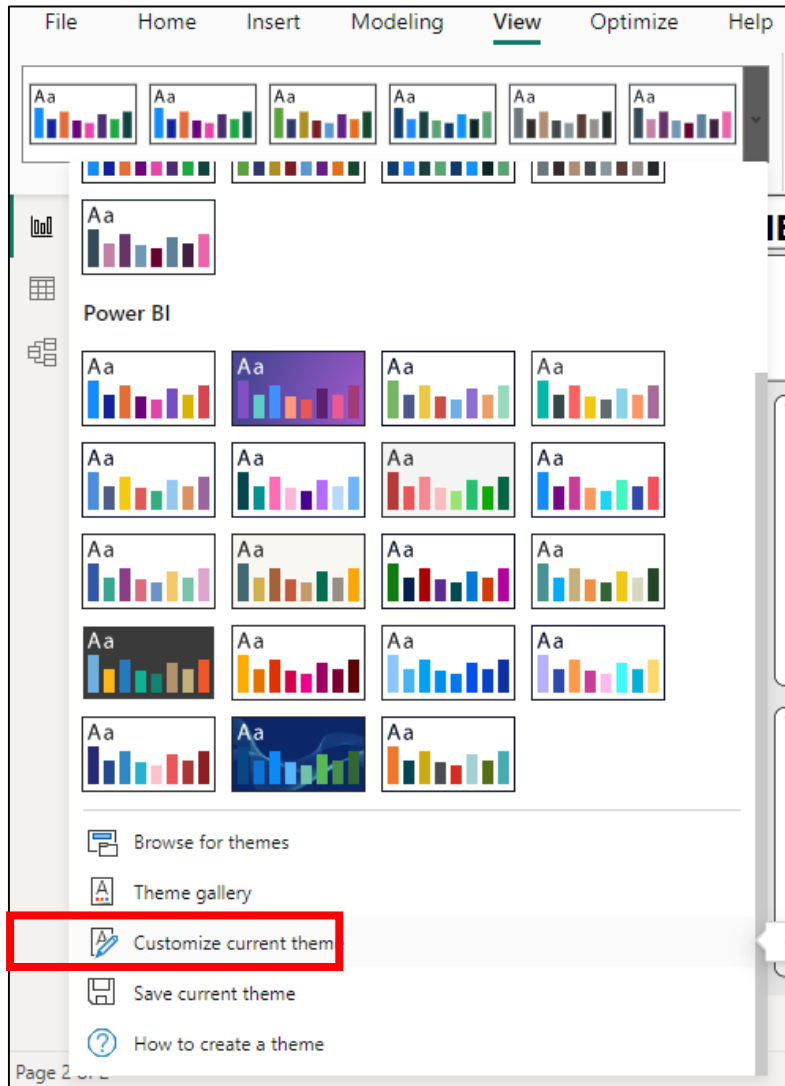




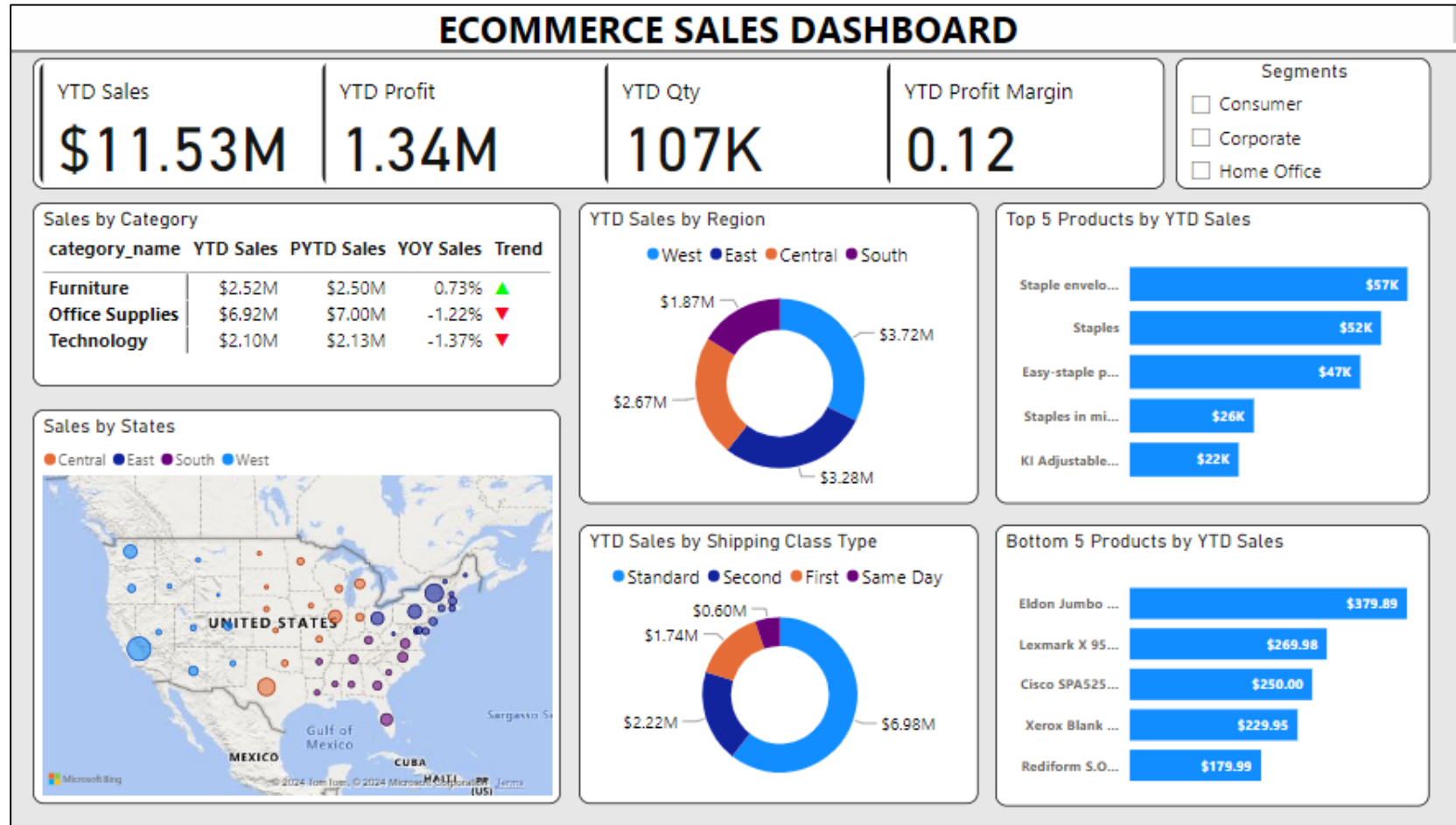
## Dashboard Theme

Existing theme: View → select from available options

Create theme: view → customize current theme → customize as per your choice.



Edit all the visuals and align them properly.  
Below is the final dashboard.



# **MICROSOFT POWER-BI**

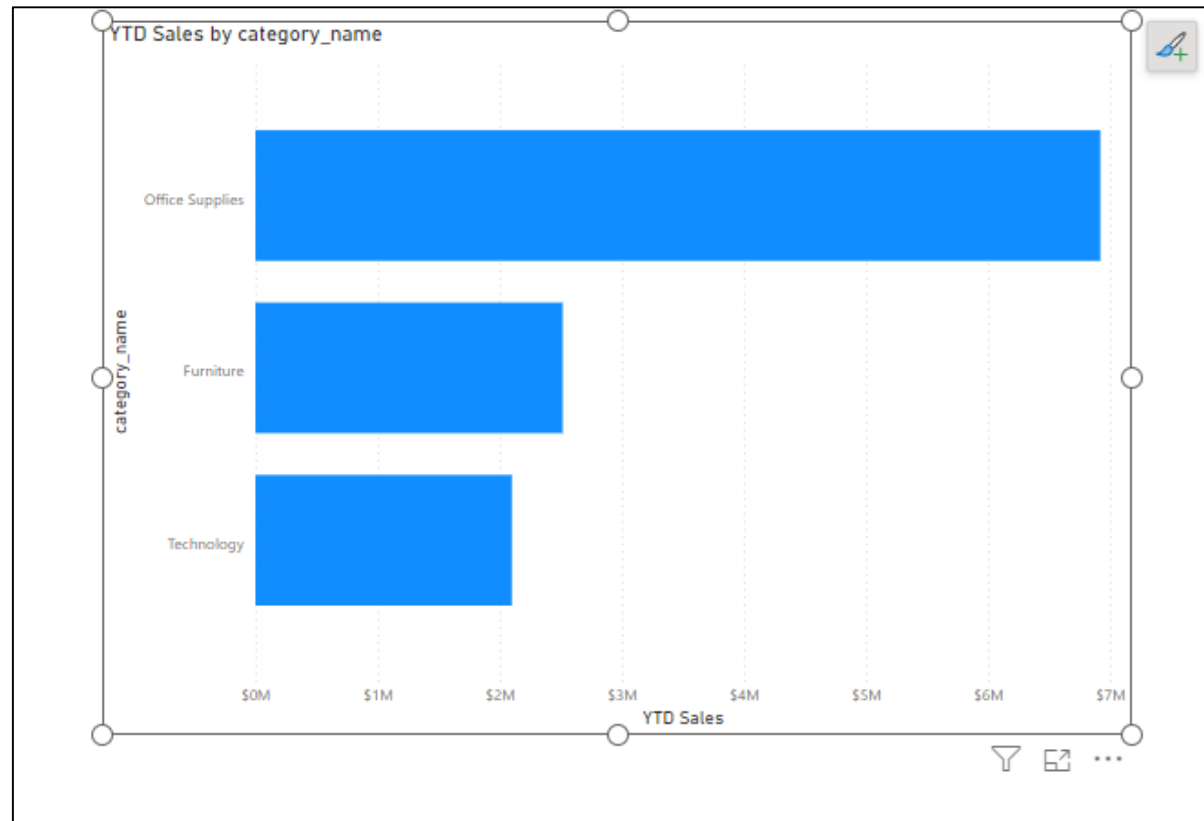
**SERIES 7**

## **VISUALIZATION (PART II)**

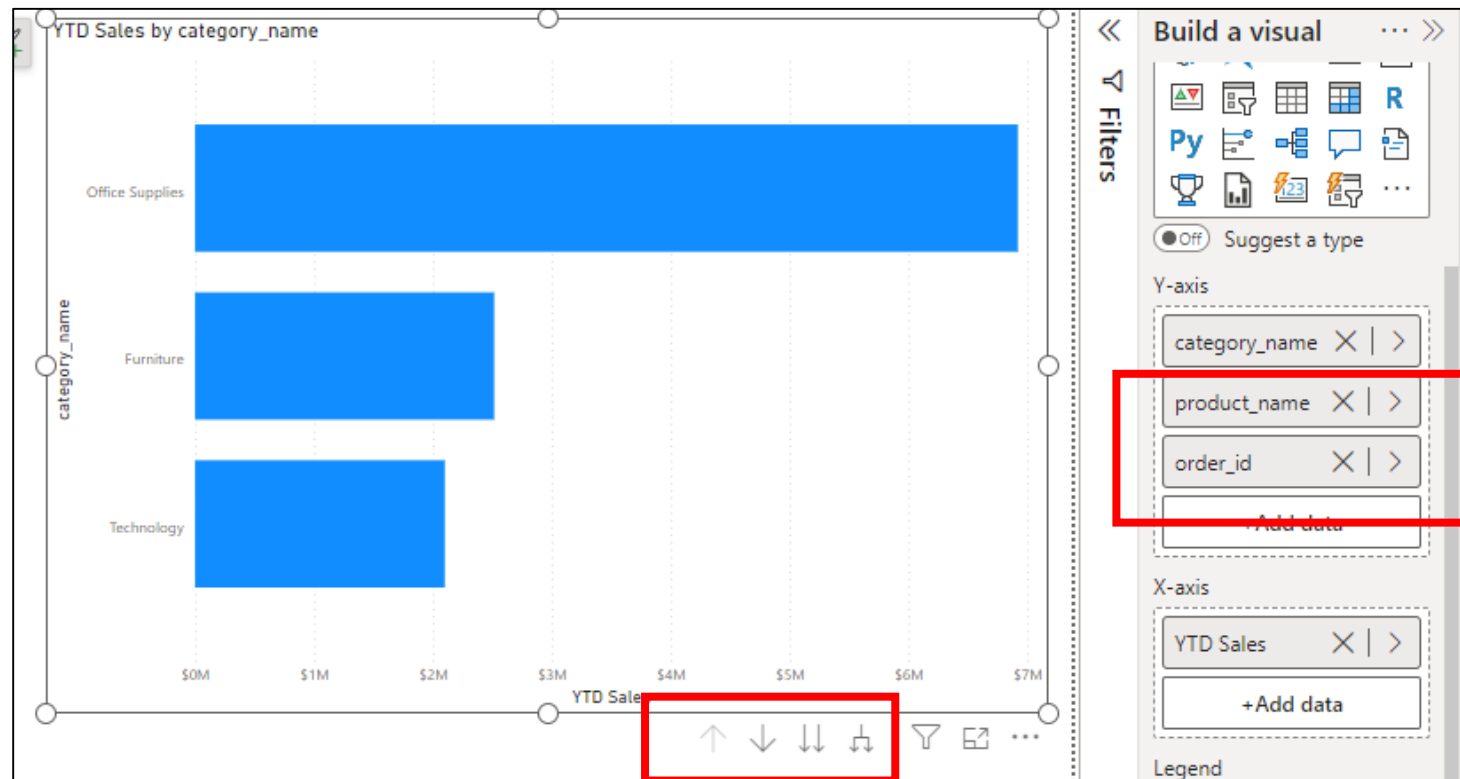
**- MAYURI .D.**

## DRILL DOWN-UP

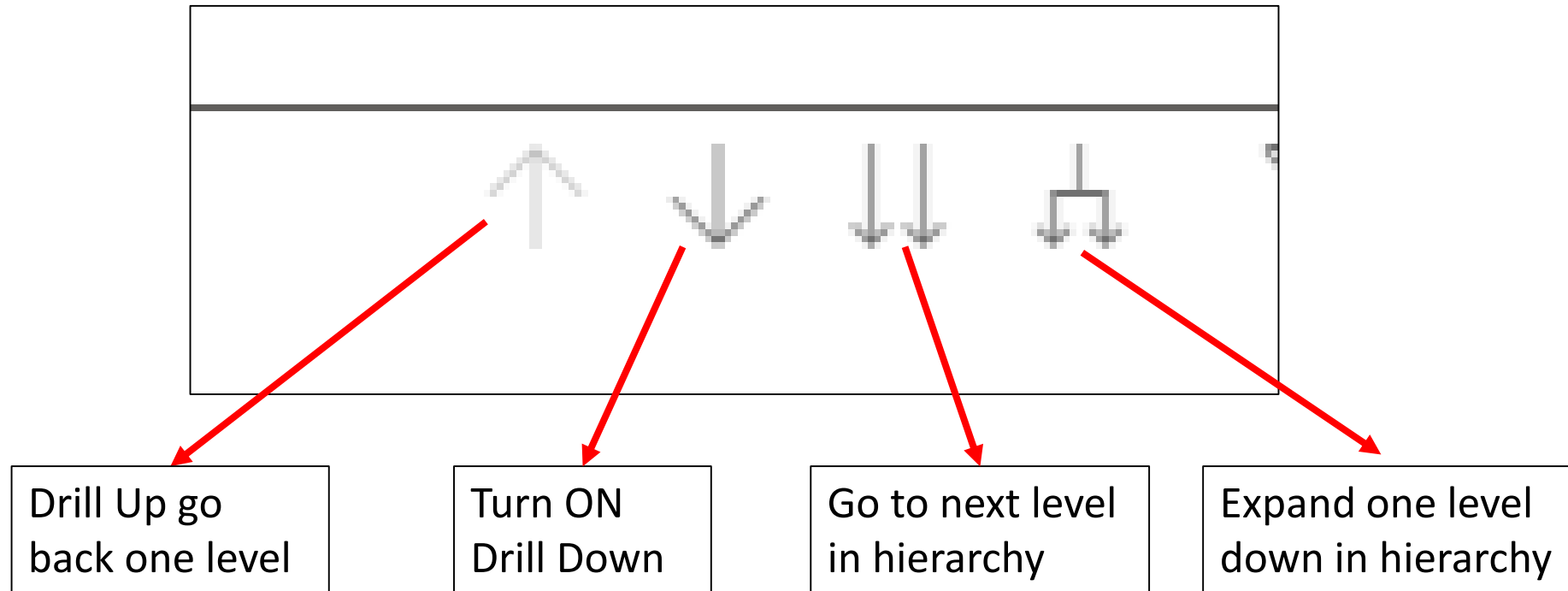
- Drill Down means basically when you click, you go to next level.
- For e.g – bar chart shows results of years, when you click you go to next level i.e. quarter
- Drill Up means going back to the original data. i.e. quarter to years
- We will go through the below chart that has YTD sales vs category name.



- To activate the drill down option, you need to add hierarchy fields like product name, order id.
- You can add 2 fields/ 3 fields, depends on your final result.

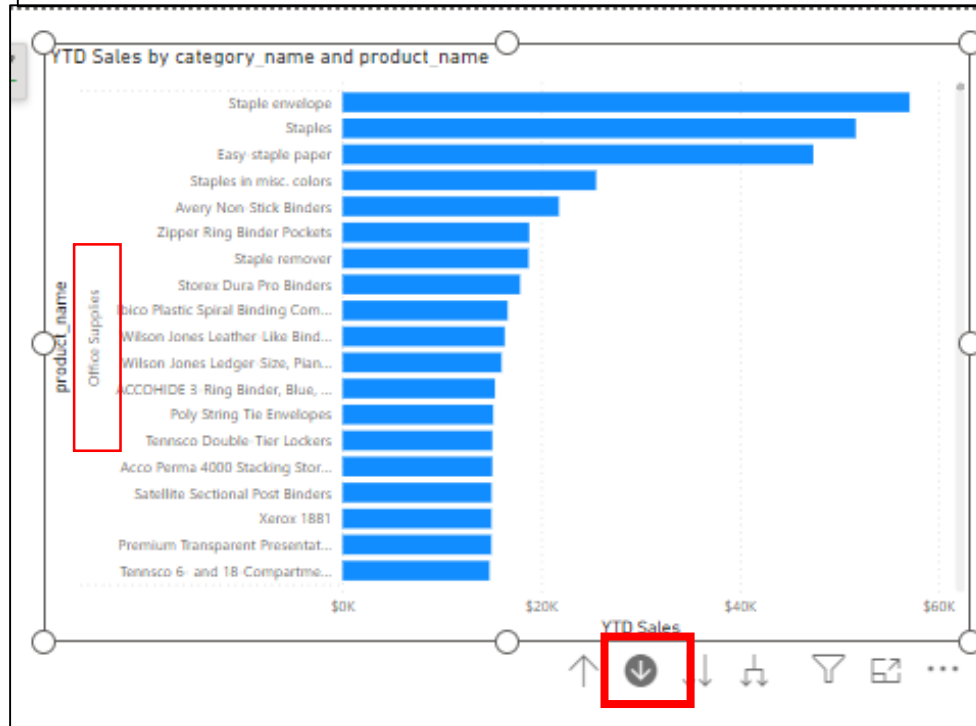


- After adding the hierarchy fields, you get 4 options activated.

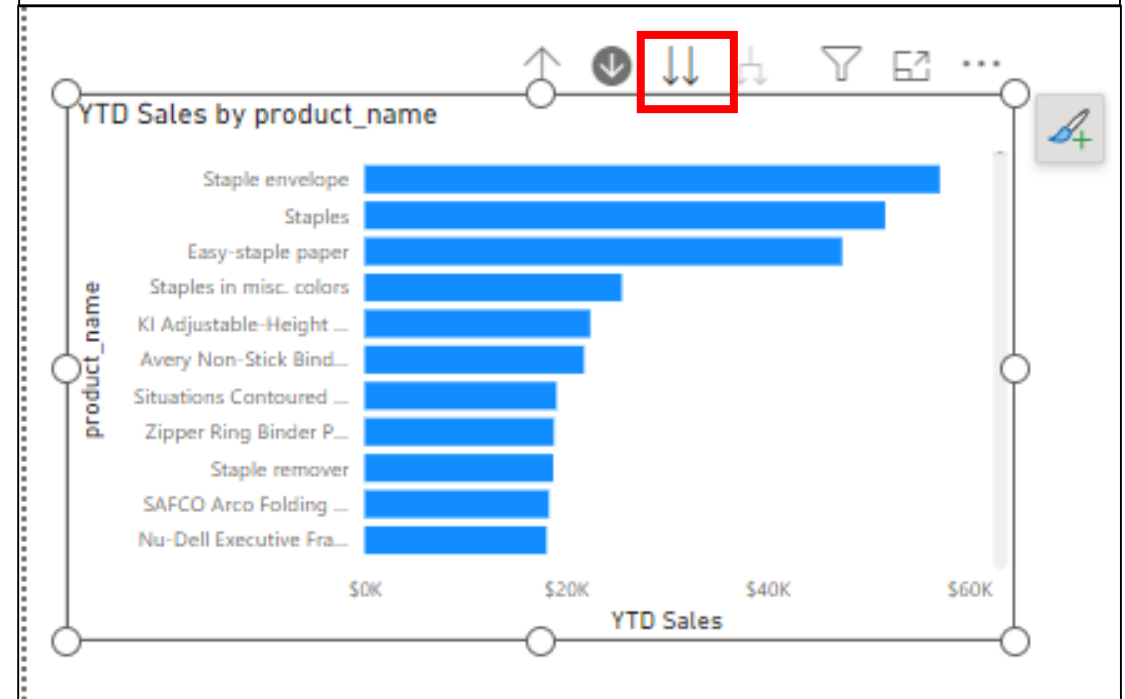


- Before any drill down, first you need to **turn ON** drill down option
- **Understand the difference** in both charts having same output.

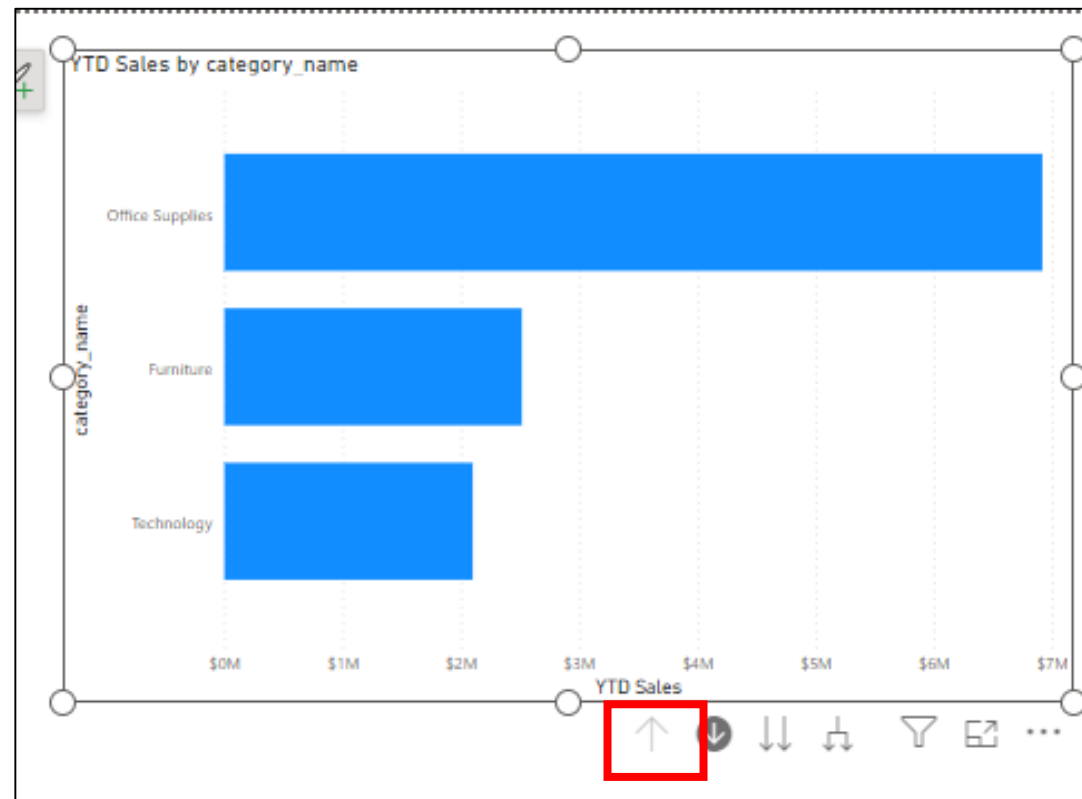
Here, I have click directly on Office Suppliers bar. You can see vertical “office supplier” category is displayed also the change in title of chart



Here, I have click on 3<sup>rd</sup> option i.e. go to next level hierarchy. Here you can not see vertical category but just “product name” also the change in title of chart



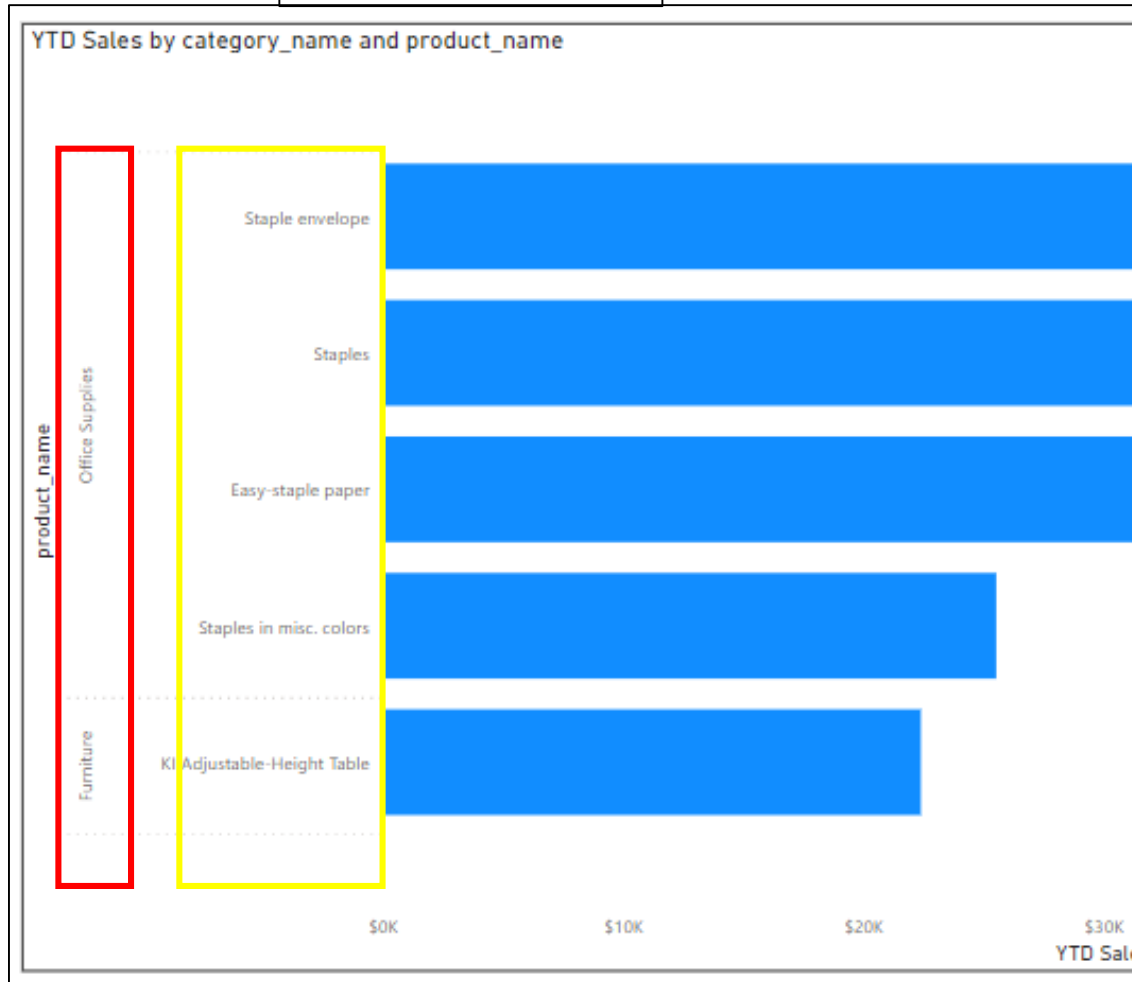
- Drill Up- back to original chart.
- It doesn't go directly to original chart, goes by level.
- This option is one shade lighter in color.



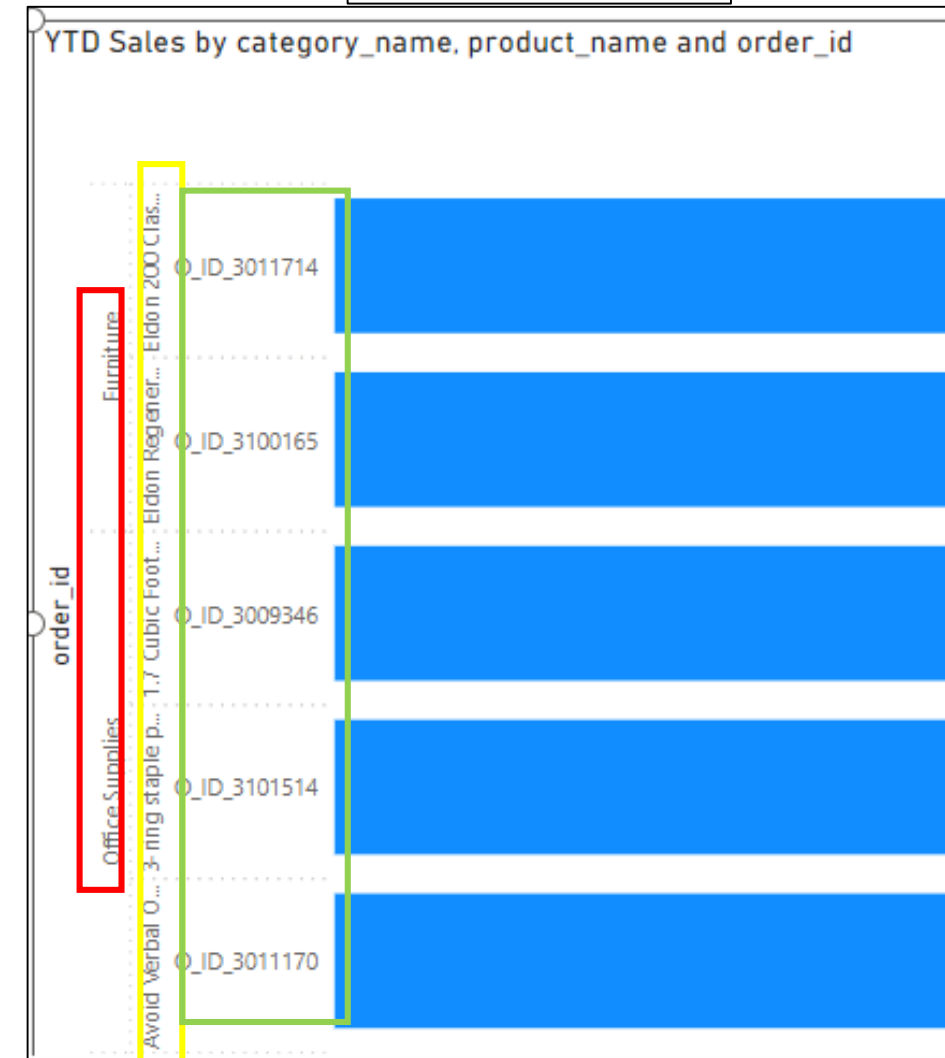


- This is how you will see when you go to 4<sup>th</sup> option expands hierarchy

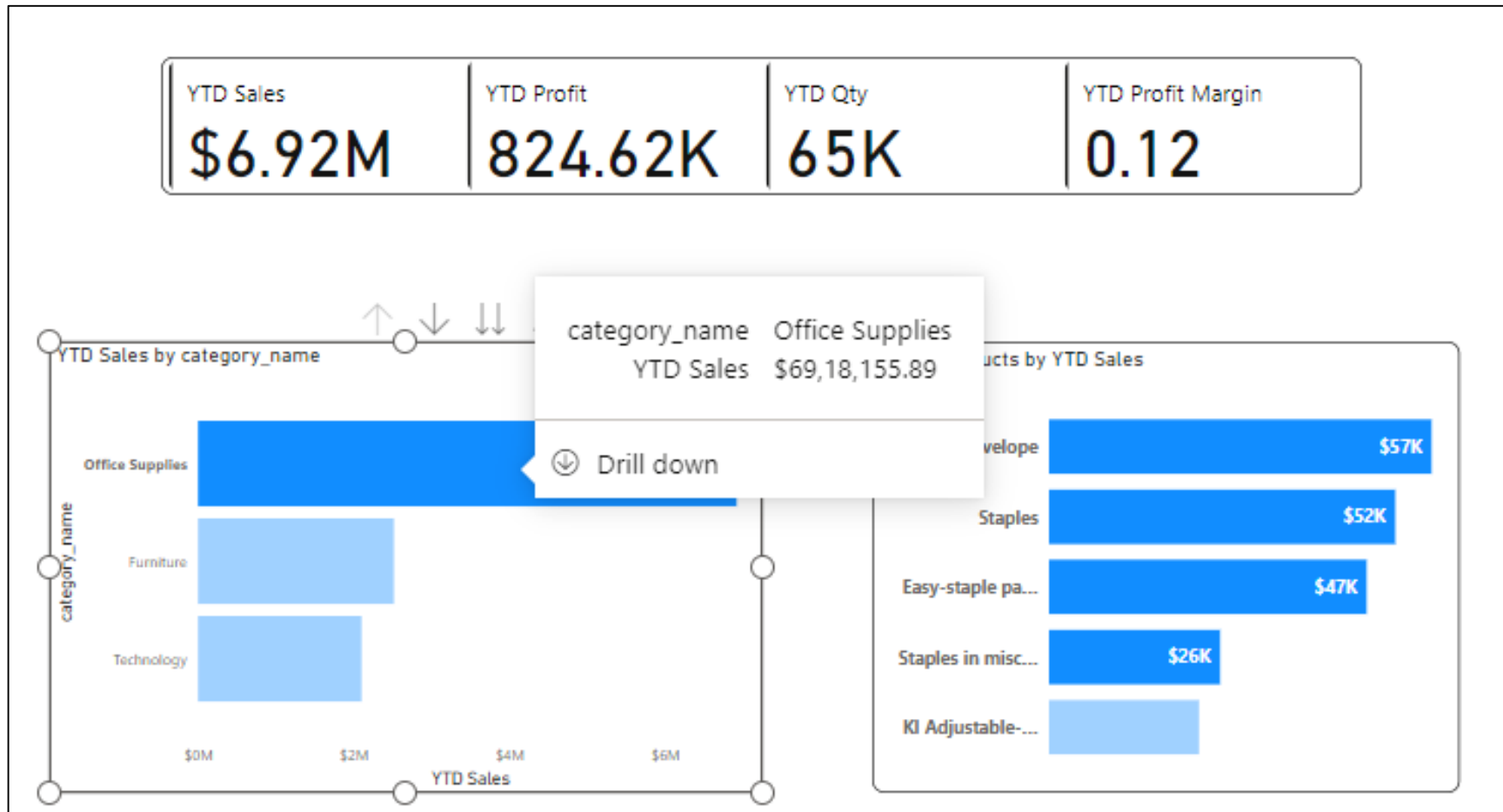
### One level down



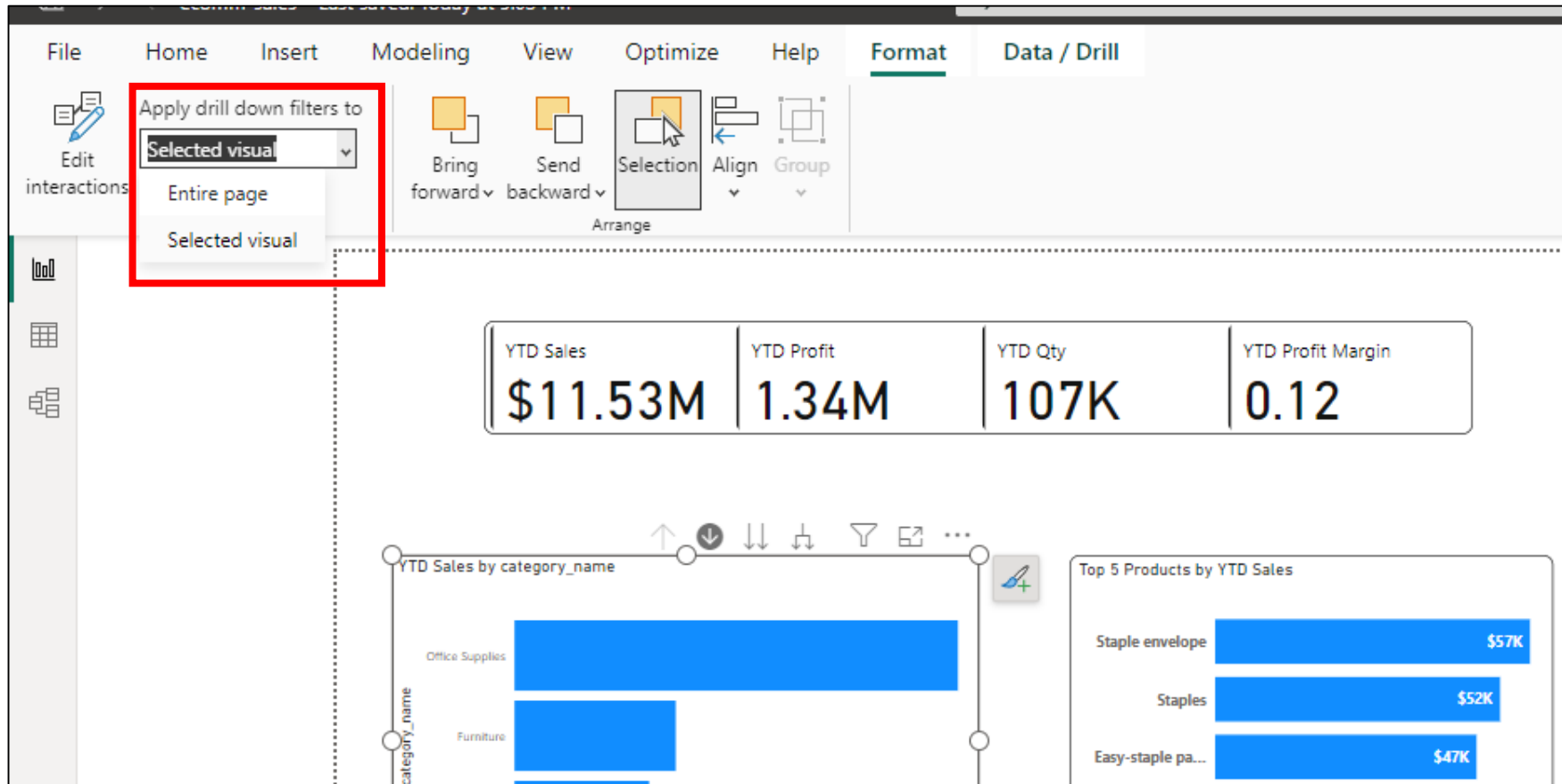
### Two level down



- When you click on any specific category, other charts too gets filter.
- If you don't want other charts to get filtered, change the setting from entire page to only visual.



- Turn ON Drill Down option, go to “interaction” under “format” tab select “selected visual”.
- “selected visual” doesn’t work when Drill Down option is turn OFF, then you can use “Edit Interaction”



The screenshot shows the Microsoft Power BI interface. The 'Format' tab is selected, and the 'Data / Drill' sub-tab is active. A red box highlights the 'Apply drill down filters to' dropdown menu, which is set to 'Selected visual'. Below the menu, the 'Edit interactions' pane is visible. The main area displays a KPI card for YTD Sales (\$11.53M), YTD Profit (1.34M), YTD Qty (107K), and YTD Profit Margin (0.12). Below the KPI card is a horizontal bar chart titled 'YTD Sales by category\_name' showing sales for Office Supplies and Furniture. To the right is a vertical bar chart titled 'Top 5 Products by YTD Sales' showing sales for Staple envelope, Staples, and Easy-staple pa...

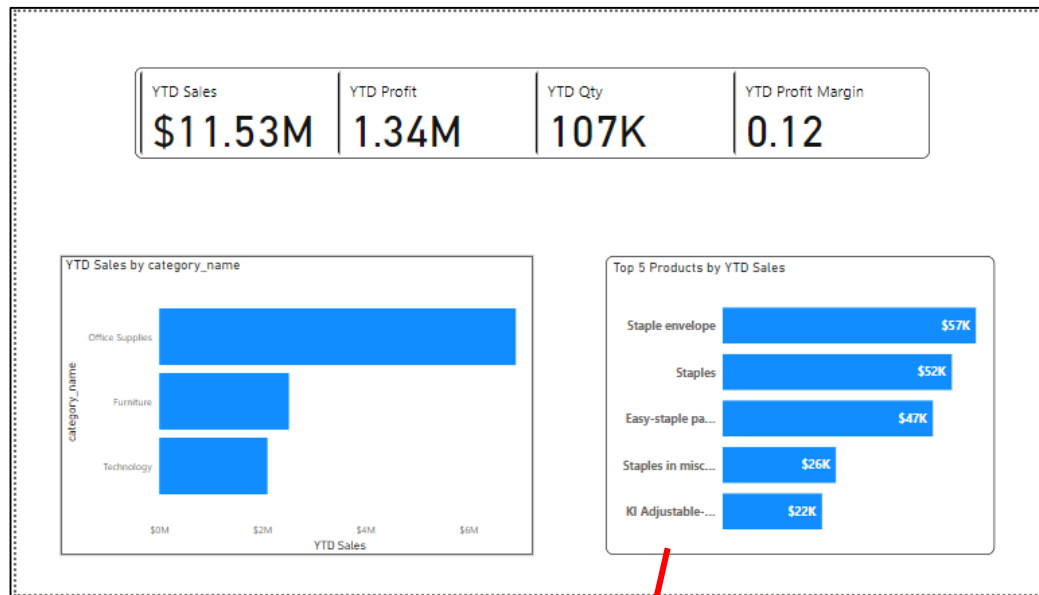
YTD Sales	YTD Profit	YTD Qty	YTD Profit Margin
\$11.53M	1.34M	107K	0.12

Category	YTD Sales
Office Supplies	\$11.53M
Furniture	\$1.34M

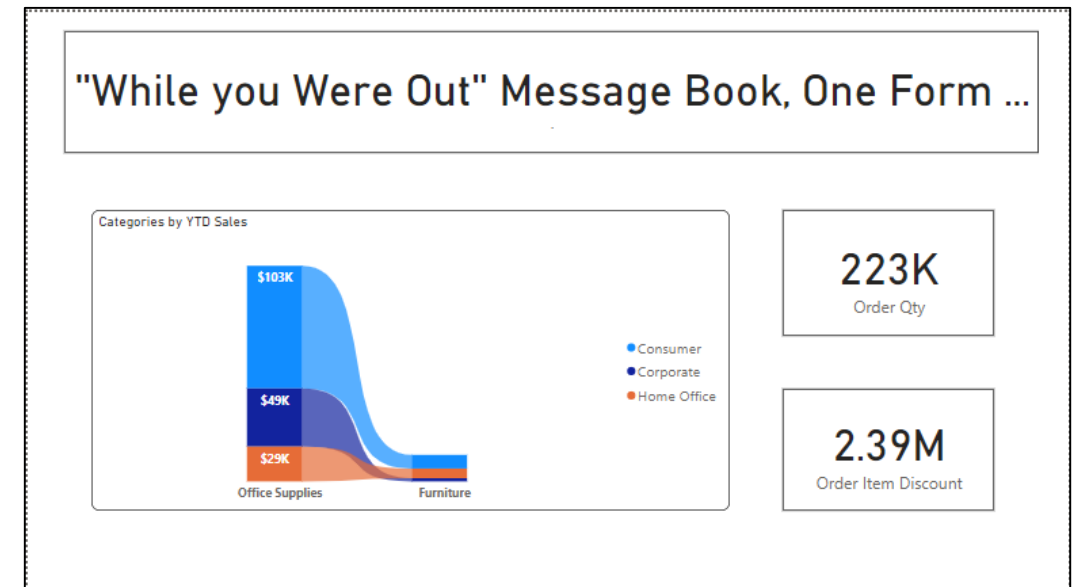
Product	YTD Sales
Staple envelope	\$57K
Staples	\$52K
Easy-staple pa...	\$47K

# DRILL THROUGH (Method 1)

With drillthrough in Power BI reports, you can create a destination target page in your report that focuses on a specific entity such as a supplier, customer, or manufacturer. When your report readers right-click a data point in other source report pages, they drill through to the target page to get details that are filtered to that context.



This is main page.  
We will use Top 5 products Bar chart



This is Drillthrough target page

**Format** ... >>

🔍 Search

▼ Page information

Name  
Drill through target

Page type  
Drillthrough ▼

Keep all filters ☒ On

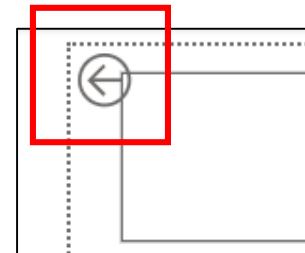
Cross-report ☐ Off

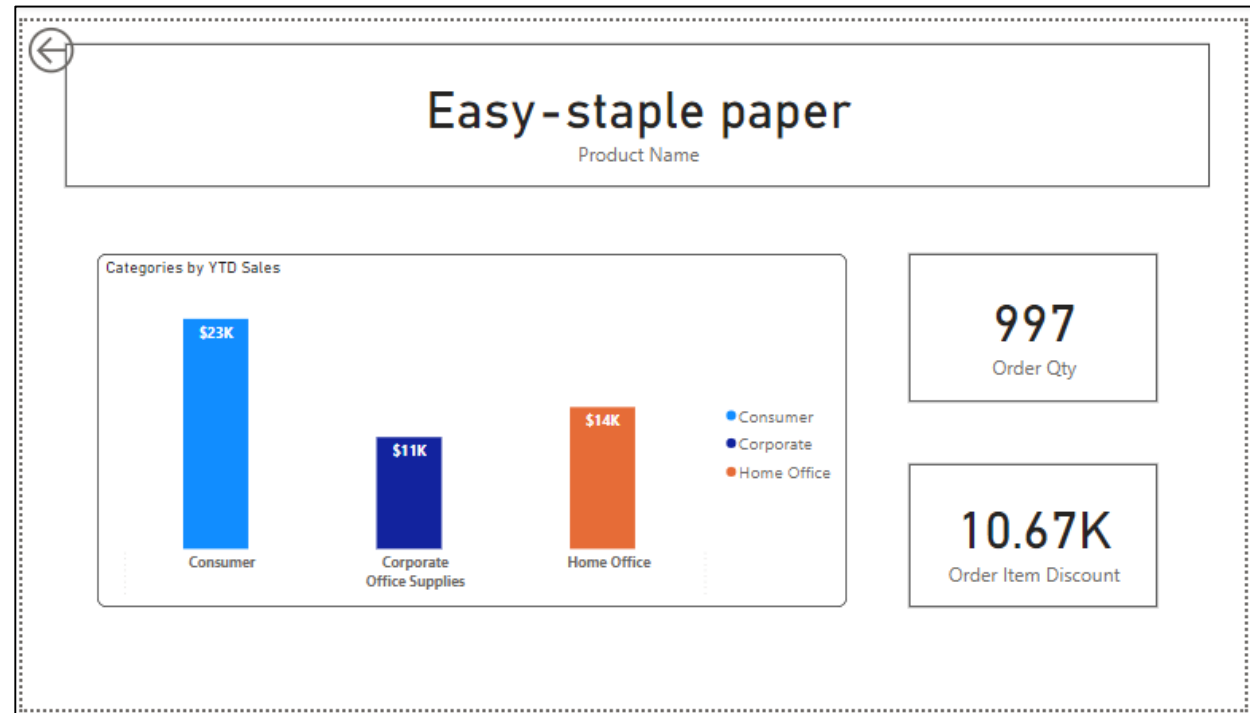
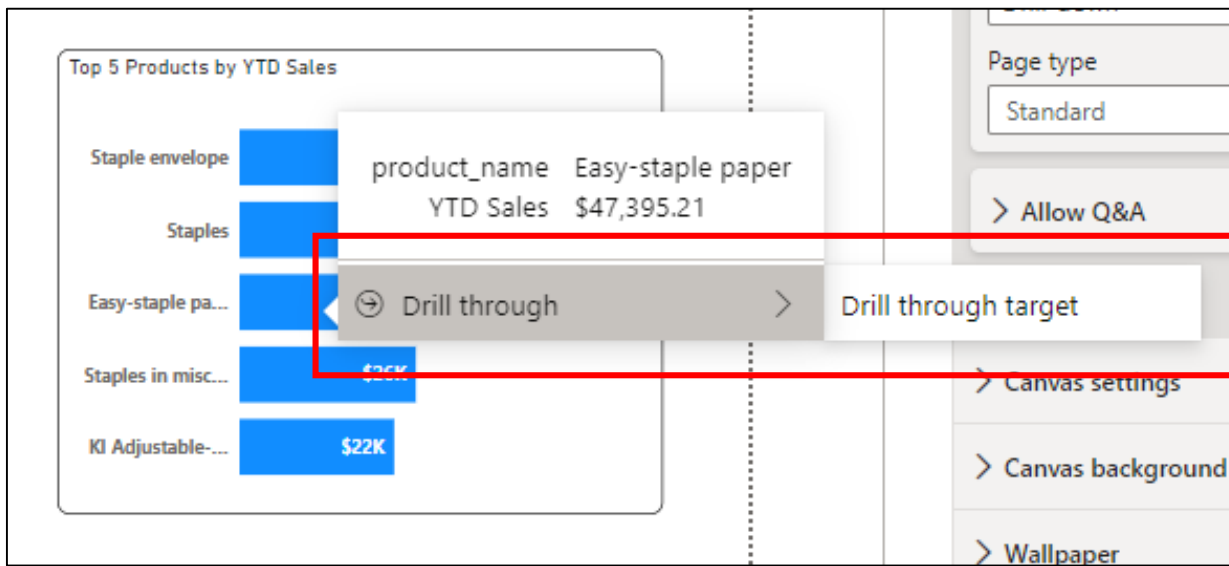
Drill through from  
product\_name ✕  
+Add data

▼ Drill through when  
product\_name is  
Used as category ▼

> Allow Q&A ☐ Off

- Go to target page click on canvas → format pane → page information → page type = drillthrough → keep all filters ON → drill through from = (field through which you want more details like I want products details so I selected product name) → drill through when = used as category
- You will find a back arrow in top corner on target page





## DRILL THROUGH (Method 2)

The screenshot shows a dashboard with a bar chart titled "Top 5 Products by YTD Sales". The chart lists five products with their sales values: Staple envelope (\$57K), Staples (\$52K), Easy-staple pa... (\$47K), Staples in misc... (\$26K), and KI Adjustable-... (\$22K). A red box highlights a "Details" button located at the bottom right of the chart area. To the right of the chart is a "Format" pane with the following settings:

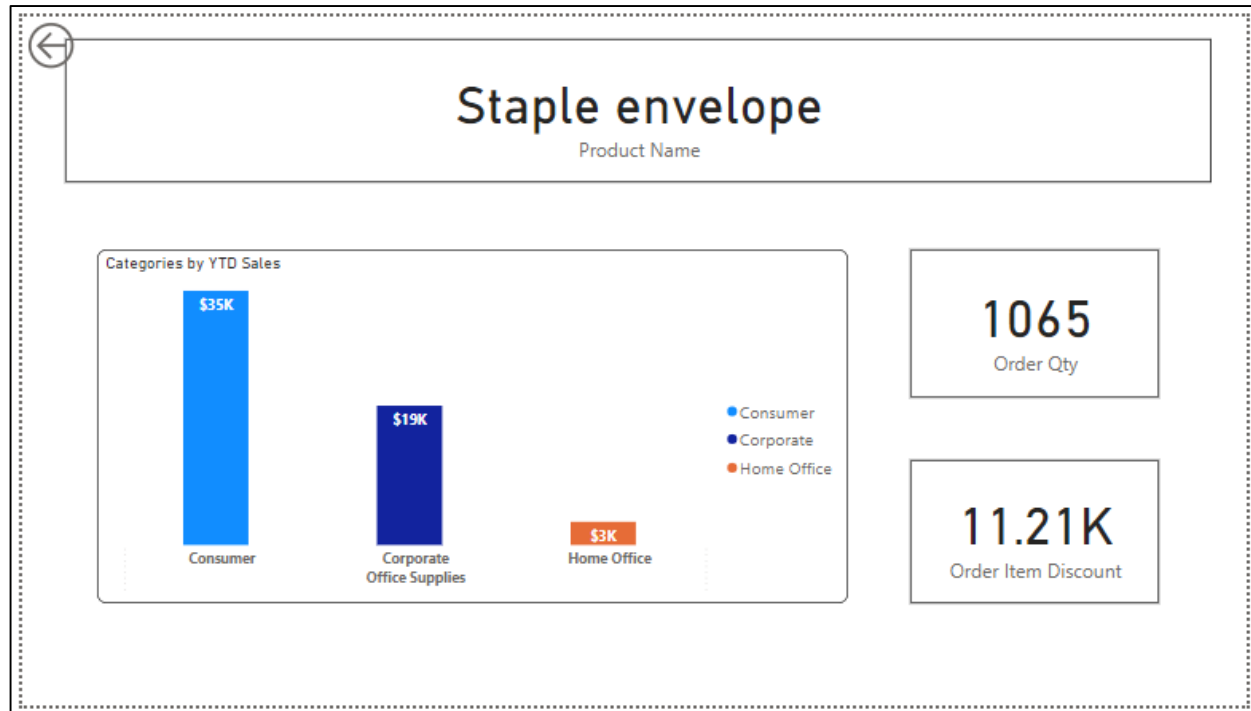
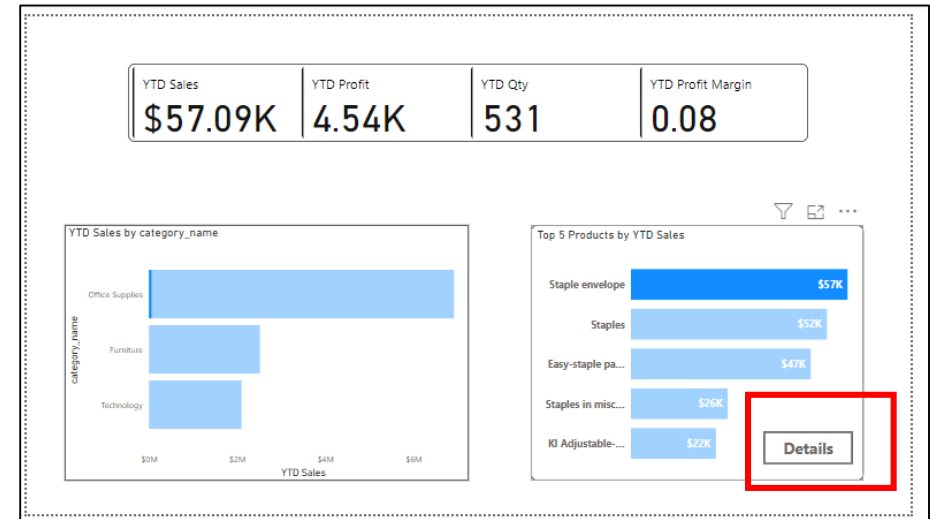
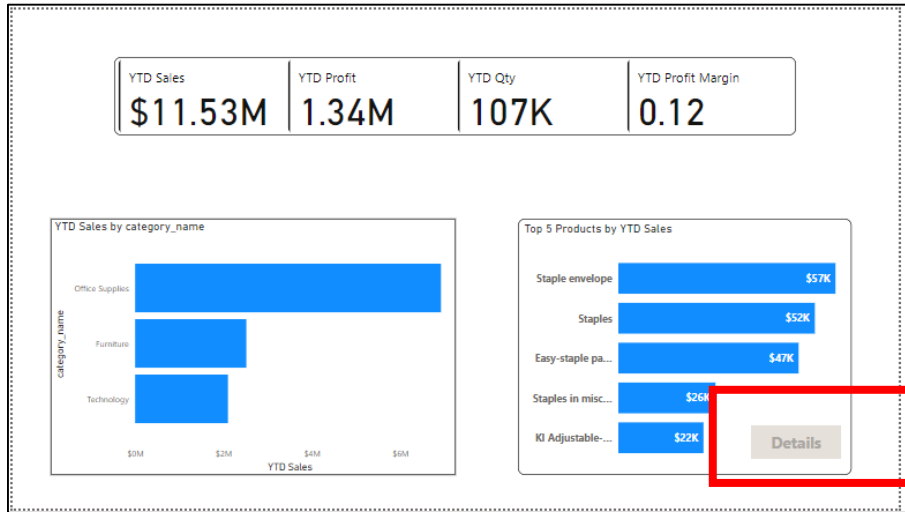
- Format** (Search bar)
- Button** (Properties)
- Action**
  - Type: Drill through
  - Bookmark: None
  - Destination: Drill through target
  - Web URL: (empty)
- Tooltip** (On)
  - Enabled text: (empty)
  - Disabled text: Click for details

Insert → Button → Blank

Format pane → action → type =  
drillthrough → destination = (select  
target page) → tooltip → disabled  
text = click for details

Button style → text → enter your  
text and do other formatting

Click on any product bar, button  
gets active, click on button, you  
are directed to target page.





# BOOKMARK

Bookmark enables users to save a particular view of a report page. This ability is extremely useful in enhancing the user experience of navigating a report or opening opportunities to add more depth to a report.

Create table “bookmark”

Home tab → enter data → enter values as shown → name bookmark → load

Create slicer with visual type field  
Make it single select in slicer setting option

**Create Table**

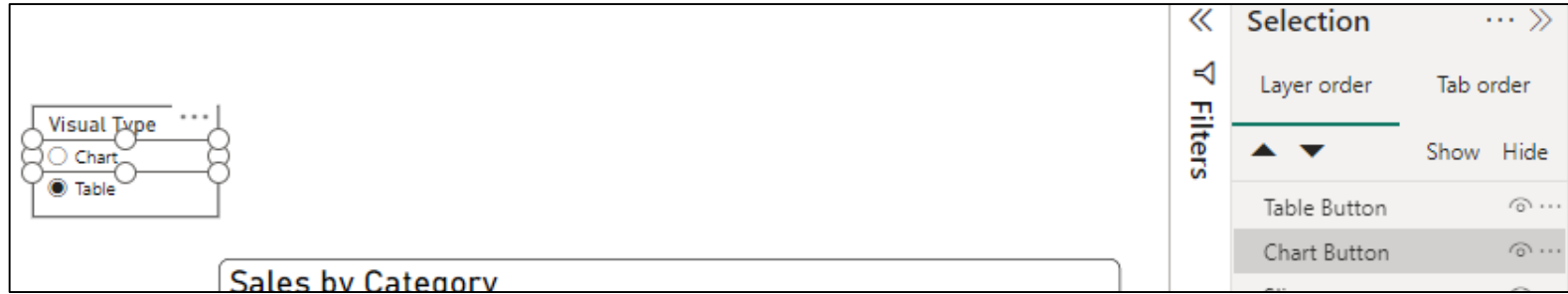
	Visual Type	Sort	+
1	Table	1	
2	Chart	2	
+			

Visual Type ▾

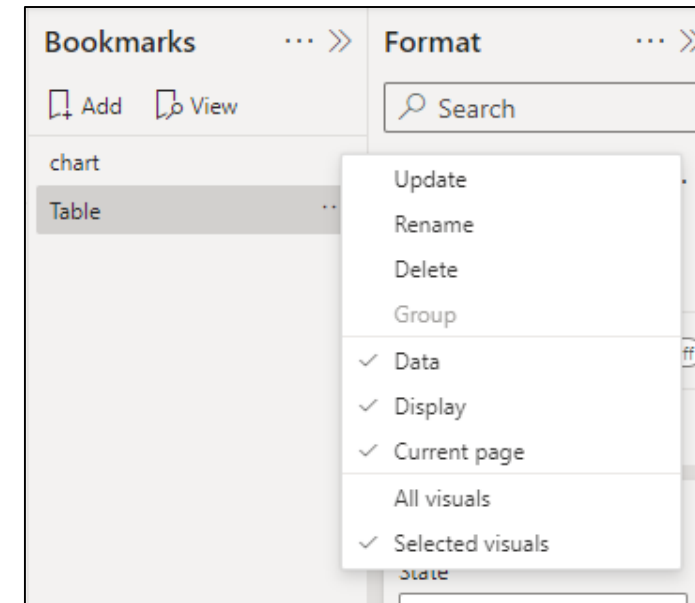
☐ Chart

☐ Table

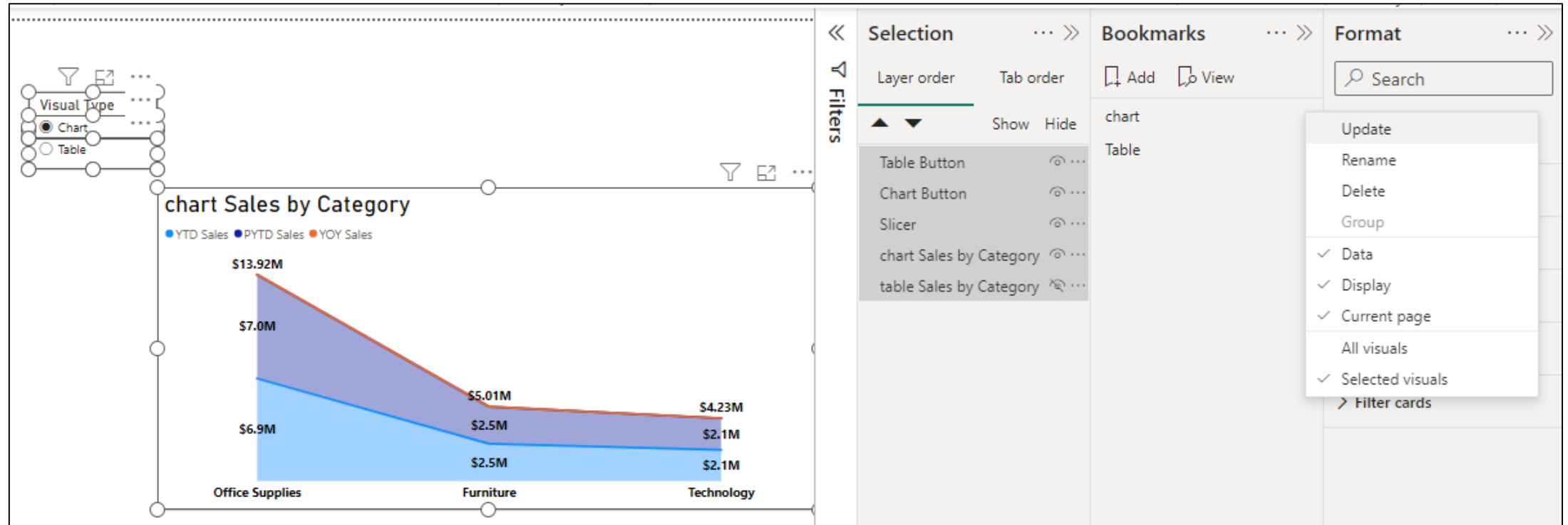
- Turn on selection and bookmark under view tab
- Insert two blank button and overlap them on visuals types and rename them in selection pane



In bookmark pane add two bookmark by clicking on add and rename them chart and table.  
Click on 3 dots and make sure all 4 option are tick



- Select chart on slicer.
- Disable table visual in selection pane
- In bookmark pane, click on 3dots of chart and click update
- Chart bookmark is set



- Select table on slicer.
- Disable chart visual in selection pane
- In bookmark pane, click on 3dots of table and click update
- Table bookmark is set

The screenshot shows the Power BI interface with a table visual titled "table Sales by Category". The table has four columns: "category\_name", "YTD Sales", "PYTD Sales", and "YOY Sales Trend". The data is as follows:

category_name	YTD Sales	PYTD Sales	YOY Sales Trend
Furniture	\$2.52M	\$2.50M	0.73% ▲
Office Supplies	\$6.92M	\$7.00M	-1.22% ▼
Technology	\$2.10M	\$2.13M	-1.37% ▼

The Selection pane on the right shows the following items:

- Table Button
- Chart Button
- Slicer
- chart Sales by Category
- table Sales by Category

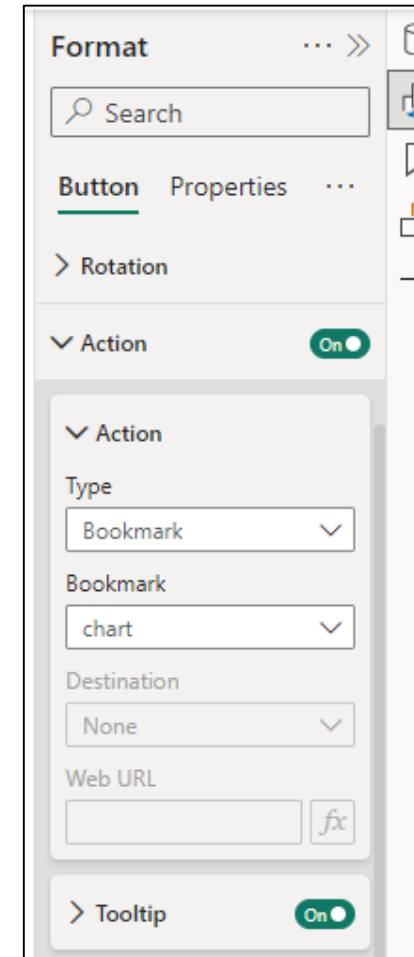
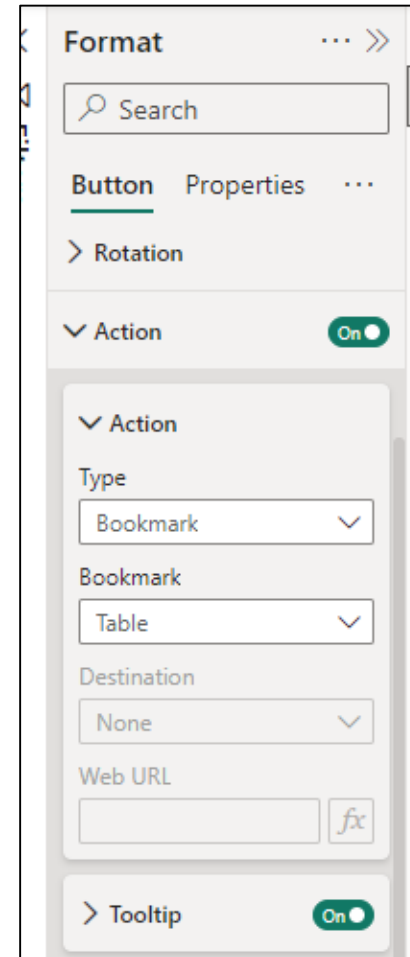
The Bookmarks pane on the right shows the following items:

- chart
- Table

The Format pane on the right shows the following options:

- Page information
- Update
- Rename
- Delete
- Group
- ✓ Data
- ✓ Display
- ✓ Current page
- All visuals
- ✓ Selected visuals

- To navigate buttons with charts
- click on buttons → format tab → turn on action → select type and bookmark name

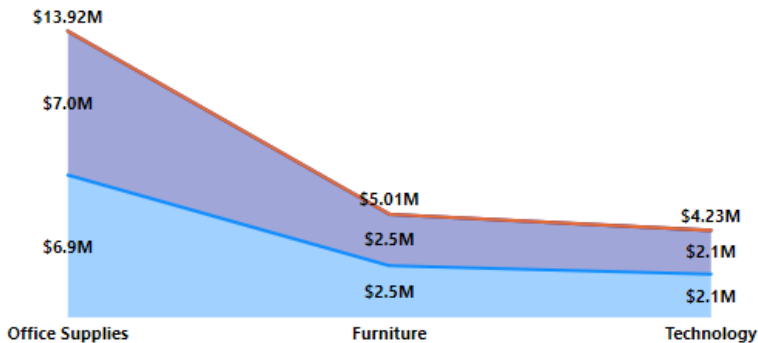


Visual Type ▾

☒ Chart
 ☐ Table

chart Sales by Category

YTD Sales PYTD Sales YOY Sales



Final look

(NOTE– while clicking use CTRL+CLICK)

Visual Type ▾

☐ Chart
 ☒ Table

table Sales by Category				
category_name	YTD Sales	PYTD Sales	YOY Sales	Trend
Furniture	\$2.52M	\$2.50M	0.73%	▲
Office Supplies	\$6.92M	\$7.00M	-1.22%	▼
Technology	\$2.10M	\$2.13M	-1.37%	▼



# **THANK YOU**

**IF YOU FIND ANY ERROR, FEEL FREE TO  
COMMENT BELOW**

**- MAYURI .D.**