

Summary: Basic Functionality of Power BI and Data Transformation

SESSION OVERVIEW:

By the end of this session, students will be able to:

- Architecture of Power BI
- Main building blocks across Power BI Desktop and Service
- Get a sense of data sources in Power BI
- Get up to speed with transforming data using Power Query

KEY TOPICS AND EXAMPLES:

1. Power BI Architecture

1. **Power BI Desktop:** This is an on-premises tool that harnesses the power of Microsoft Query Engine and combines it with data modelling and visualizations for creating reports.
2. **Data Sources:** One may require data to come in from multiple sources and in different file formats. The sources of data that Power BI gets data from are called data sources in Power BI. Examples include Excel, SQL database, Mixpanel, Salesforce, Databricks etc.
3. **Data Storage:** Till this stage our **ETL** process is complete as we have **Extracted**, **Transformed** and **Loaded** data. If you import data in Power BI Desktop, data is stored locally within Power BI in ‘pbix’ (Power BI file). Power BI Service stores data in Azure Blob Storage.
4. **Data Visualisation:** In this step, we visualize the data through charts, graphs, maps and other visualizations.
5. **Power BI Service:** As covered in the previous session, this is a cloud-based tool that can be used to share reports and dashboards within an organization seamlessly. This is an online service.
6. **Power BI Mobile:** The report viewers can log into Power BI service and view it on any device. Data gets refreshed automatically, so you just need to publish once and your viewers will always see the updated data.

The image below explains the data architecture clearly.

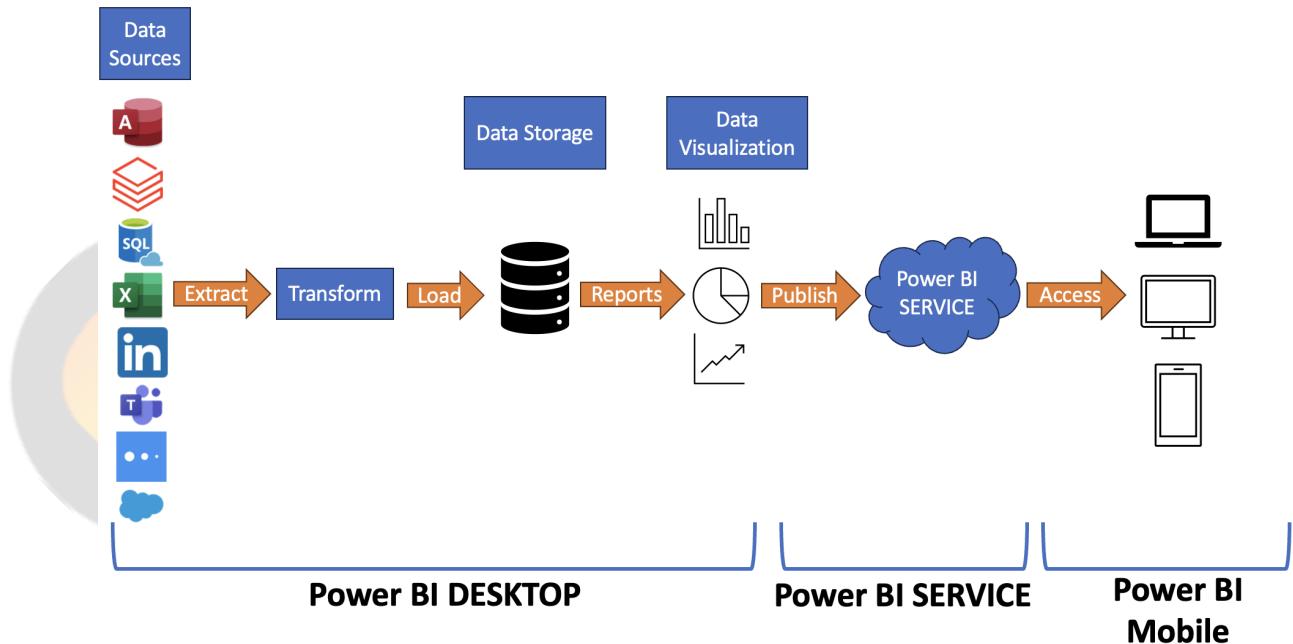


Figure: Data Architecture for Power BI

2. Building Blocks of Power BI

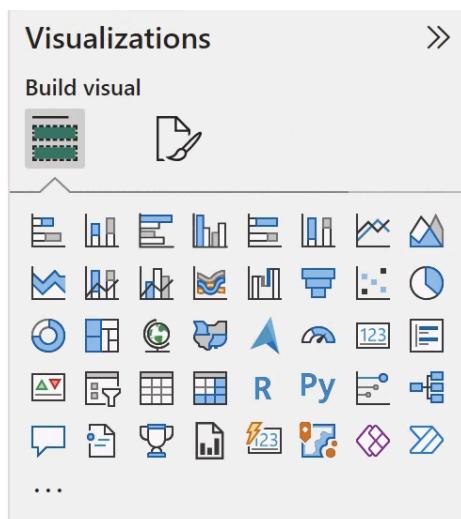
Before we start using Power BI, we must understand the 5 main building blocks.

A. Datasets

A dataset is a collection of data. We can combine data from multiple sources to get a unique collection of data called a dataset. This can be achieved through filtering those sources and using **connectors**. Connectors help you connect to data sources, and bring it into the dataset.

B. Visualisations

Visual representation of data can be done exceptionally well in Power BI. This is possible through many options that Power BI provides such as charts, graphs, slicers, KPIs and maps.



C. Reports

A report is a collection of many visualizations. It can span across one page or multiple pages.

The visualizations in a report are related to each other and have a common goal.

An example of a report created on a Netflix dataset is given below:



Figure: Reports in Power BI

D. Dashboards

A dashboard is a single-screen display of information that displays the most important aspects of the report. A dashboard can be used to tell a story. It can display visualizations from multiple reports in a single place.

Note: A dashboard is an element of Power BI Service and not Power BI Desktop.

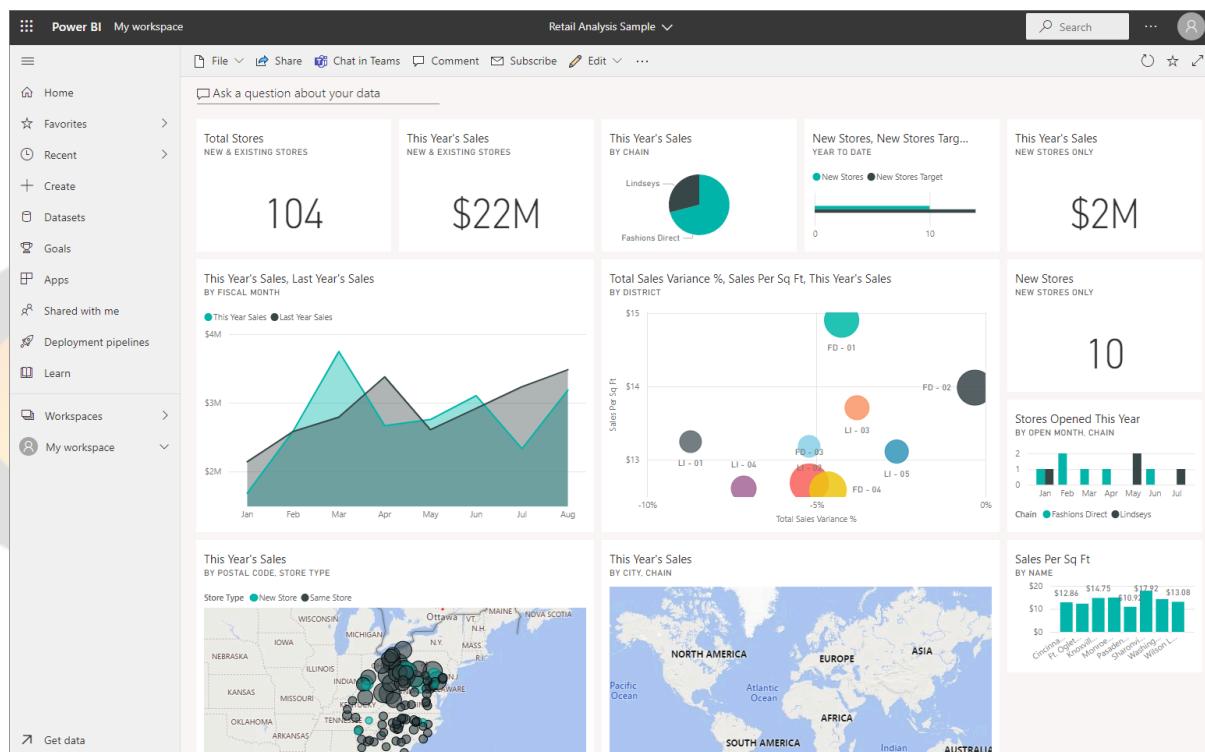


Figure: Dashboards in Power BI

E. Tiles

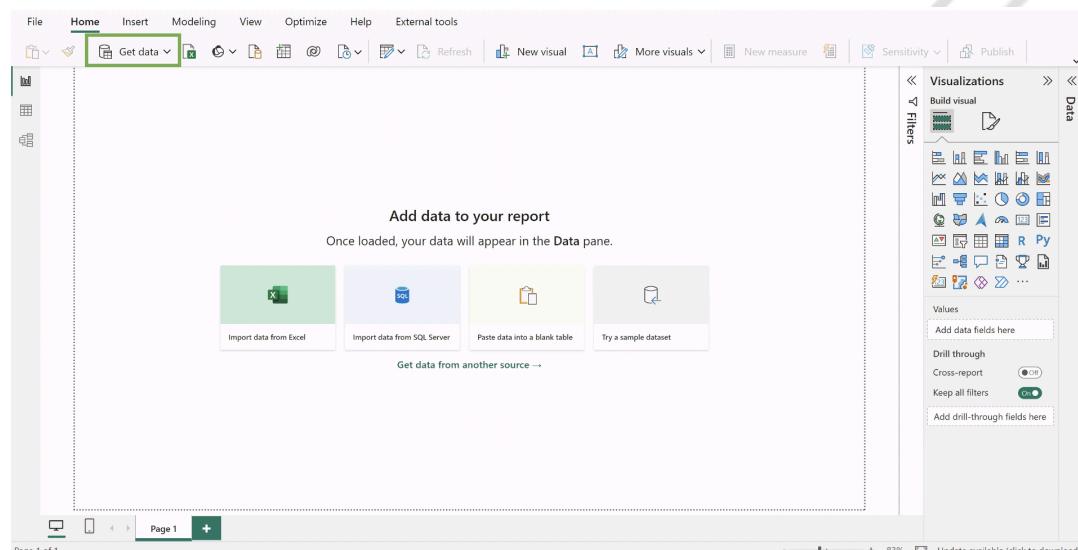
The visualizations on a dashboard are called tiles. They are pinned to the dashboard from reports. The tiles can be customized and arranged to ensure that the dashboard tells a meaningful story.

The individual elements displayed in the previous image are tiles.

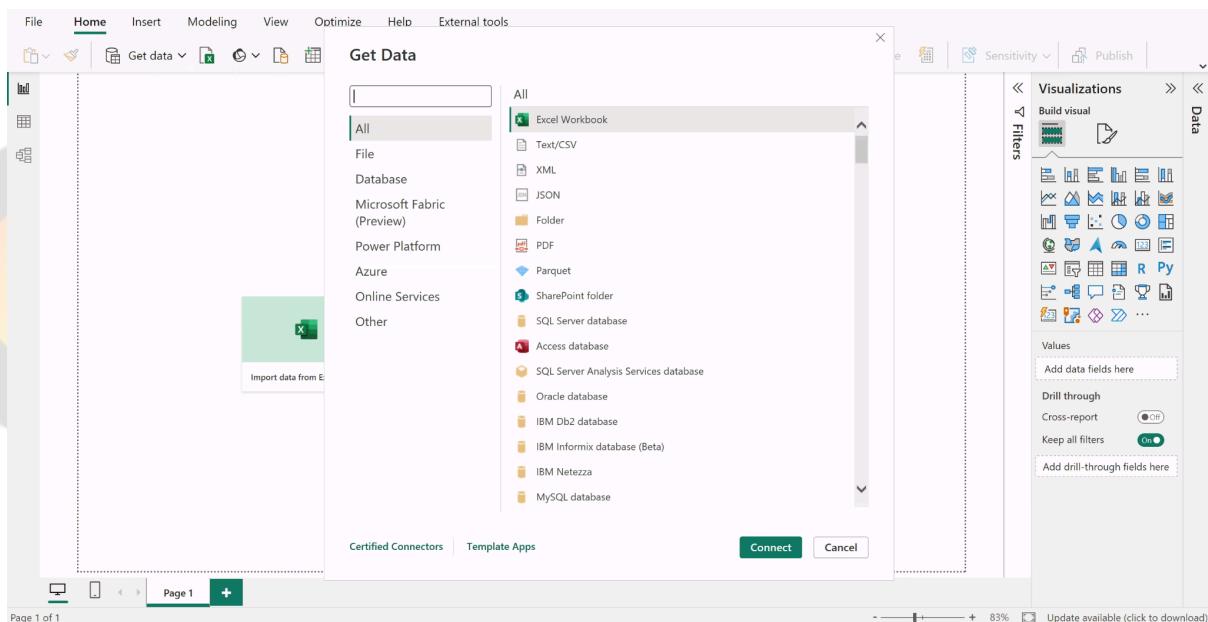
Note: Since a tile is a part of a dashboard, it is an element of Power BI Service and not Power BI Desktop.

3. Power BI Data Sources

- I. **Get Data:** Click on the Get Data button on the top-left of the Power BI Desktop window.



- II. Select Data Source:** Select the data source and click on Connect. The image below shows some of the available data sources:



The categories of data sources from which you can choose are:

Category	Description	Examples
All	You can choose from among all available data sources	Excel, Power BI dataset, Azure SQL Database
File	Shows the flat file types supported in Power BI desktop	Excel, CSV, PDF
Database	Lists all database connections that you can connect to by passing username, password.	SQL Server Database, Oracle Database
Microsoft Fabric	Includes Microsoft / Power BI data sources	Power BI dataset, Datamarts
Power Platform		Dataverse, Dataflows
Azure	Includes Azure data sources	Azure SQL Database, PostgreSQL
Online Services	Contains online data sources	Sharepoint online, Google Analytics, Adobe Analytics
Other		Web, Hadoop File, R script, Python script

In addition to the above data sources, Power BI is continually adding new data sources.

Ways to Load and Store Data

Import vs DirectQuery

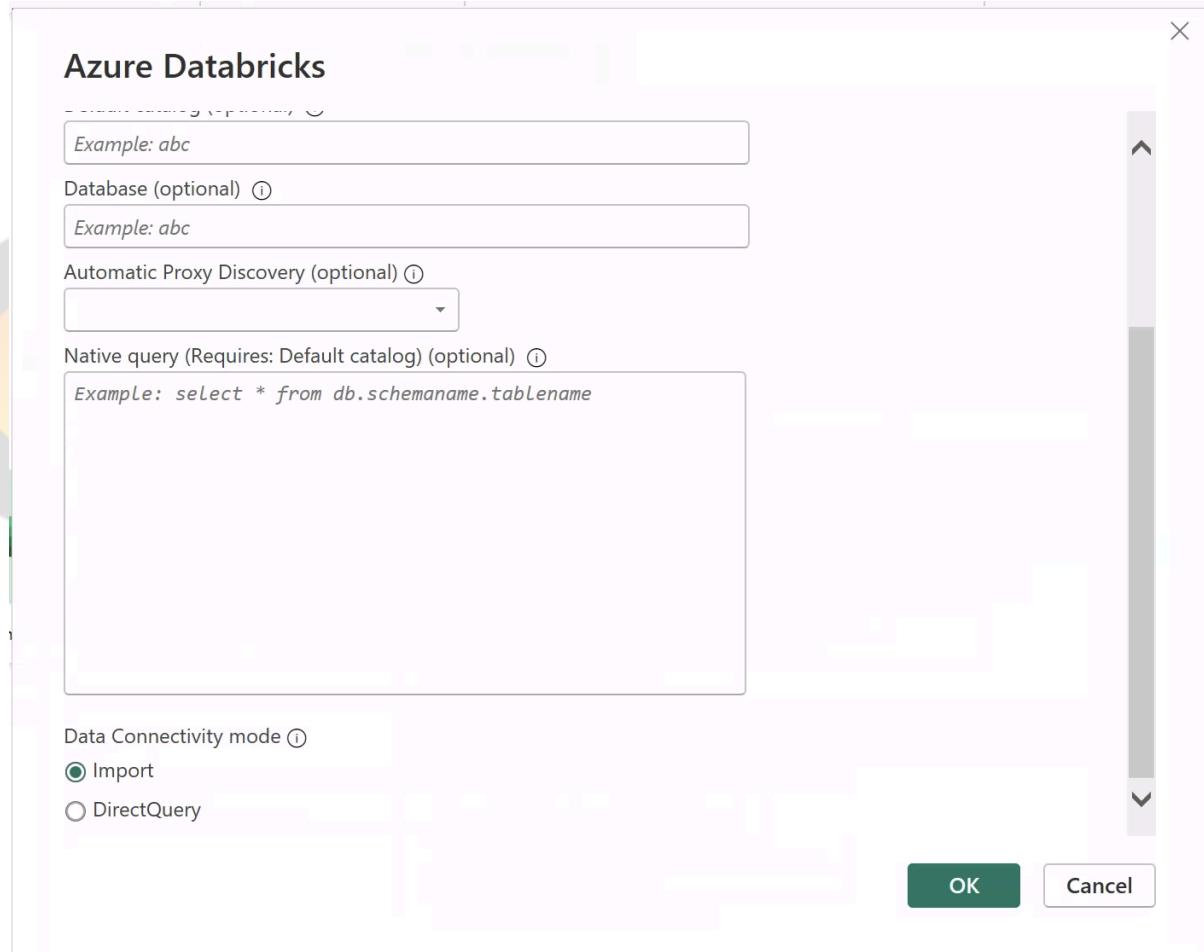
While connecting to a database, Power BI provides you with options to choose from:

- **Import:**
 - Data is stored in Power BI Azure database, with a 1 GB limit.
 - It allows you to transform and manipulate data.
 - We can schedule refreshes for up to 8 times a day on the normal license and up to 48 times a day on the Premium Pro license.
- **DirectQuery:**
 - Data is queried directly from the original database and is not stored in Power BI. All tables should come from a single dataset.
 - It limits the option to transform/manipulate data.
 - It is a real time query and there is no need to schedule a refresh.
- **Live Connection:**
 - This is similar to DirectQuery as it doesn't store any data in Power BI.
 - It is different from DirectQuery in that it is a direct connection to the analysis services model. Instead of being a connection to a dataset, it is a connection to a model.
 - The supported data sources with live connection are limited. Example: SQL Server Analysis Services (SSAS) Tabular, Power BI Service.

When to use which kind of connection:

Import Mode: This is generally preferred for static or slowly changing datasets where high performance and complex data modeling are required within Power BI. It's also suitable when working with data that needs to be transformed or enriched before analysis.

DirectQuery Mode: This is beneficial when dealing with very large datasets that cannot be easily imported due to size limitations or when real-time data updates are critical for the analysis. However, it's important to consider the potential impact on query performance and the limitations on data transformations.



III. Connect to data source

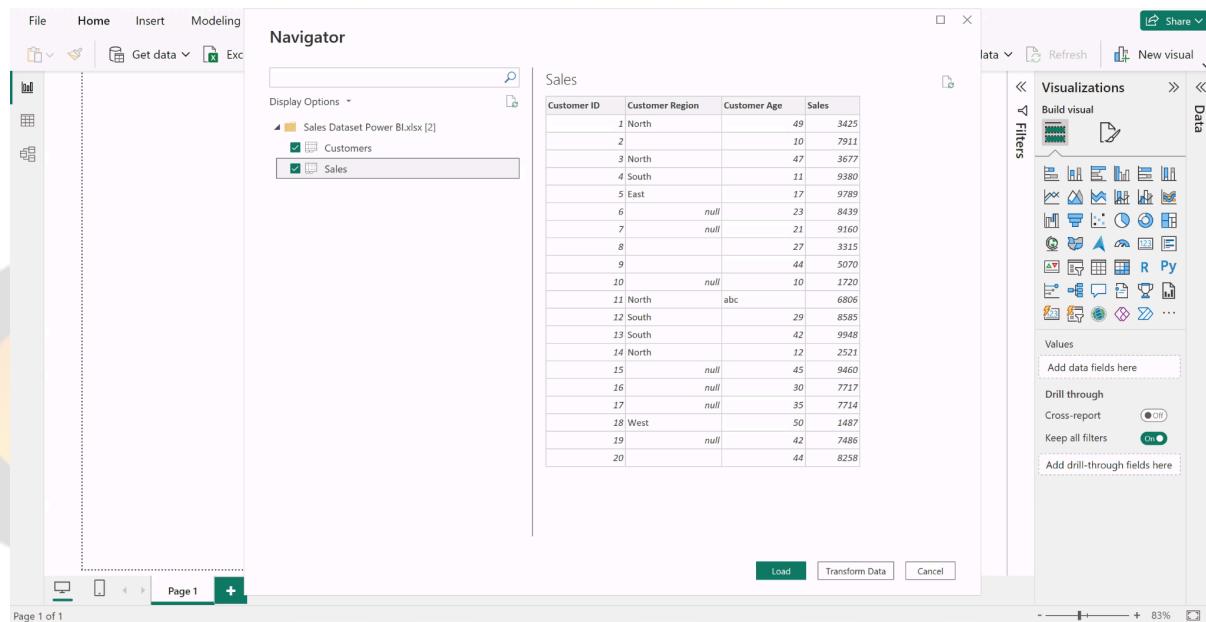
Select the data source (example: Excel file) from your local disk/online.

IV. Load/Transform

In the Navigator pane, select the sheets/fields from the left side and click on Load/Transform Data as per requirement.

Example 1: Loading an Excel file (.xlsx file) - Sales dataset

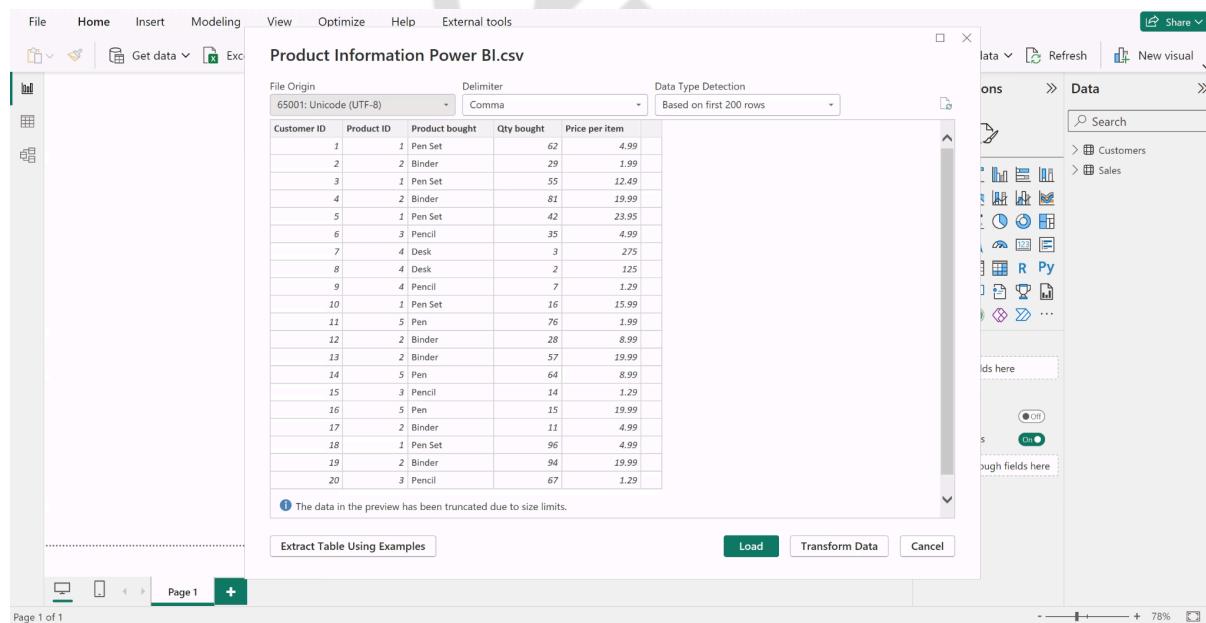
Download [this](#) dataset and get data in Power BI using the Excel workbook option in the Get Data dialog box.



The screenshot shows the Power BI desktop interface. In the top ribbon, the 'Modeling' tab is selected. The left sidebar has a 'Navigator' icon. The main area displays a preview of a 'Sales' dataset. The preview table has columns: Customer ID, Customer Region, Customer Age, and Sales. The data shows various sales records across different regions and ages. On the right side, there's a 'Visualizations' pane with various chart and report options, and a 'Data' pane showing filters and drill-through settings.

Example 2: Load data from a CSV file

Download [this](#) dataset and load it in Power BI using the Text/CSV file option in the Get Data dialog box.

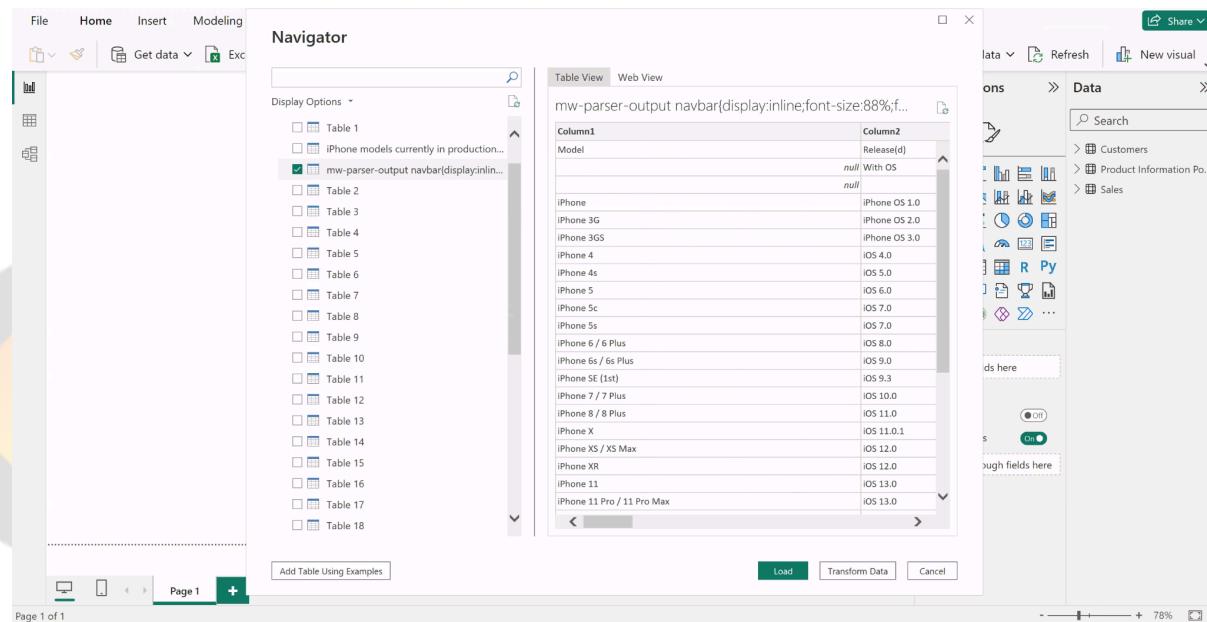


The screenshot shows the 'Get data' dialog for a CSV file named 'Product Information Power BI.csv'. The dialog includes fields for 'File Origin' (set to '65001: Unicode (UTF-8)'), 'Delimiter' (set to 'Comma'), and 'Data Type Detection' (set to 'Based on first 200 rows'). The preview table shows a list of products with columns: Customer ID, Product ID, Product bought, Qty bought, and Price per item. The data includes various items like Pen Sets, Binders, and Desks. On the right, the 'Data' pane shows the loaded data source and its tables.

Example 3: Get data from the Web

Get Data → go to Other data sources option in Get Data dialog box and enter this url:

<https://en.wikipedia.org/wiki/IPhone>

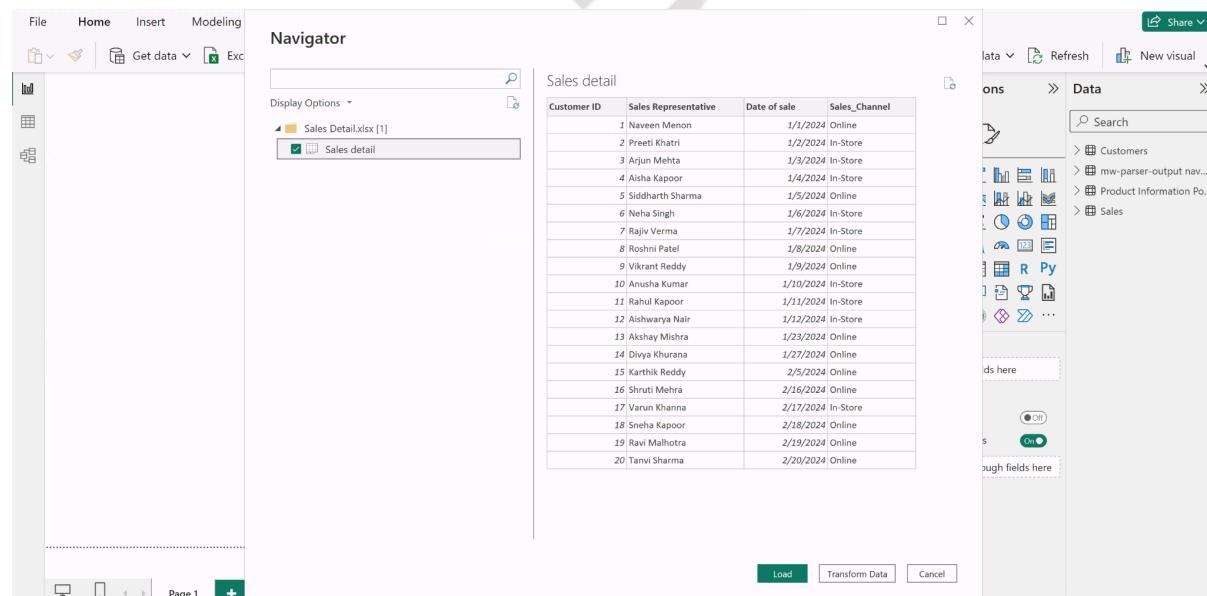


The screenshot shows the Power BI Data view. In the center, there is a table titled "mw-parser-output navbar" with two columns: "Column1" and "Column2". The data includes rows for various iPhone models like iPhone, iPhone 3G, iPhone 3GS, iPhone 4, iPhone 4S, iPhone 5, iPhone 5S, iPhone 6 / 6 Plus, iPhone 6s / 6s Plus, iPhone SE (1st), iPhone 7 / 7 Plus, iPhone 8 / 8 Plus, iPhone X, iPhone XS / XS Max, iPhone XR, iPhone 11, iPhone 11 Pro / 11 Pro Max, and iPhone 12. The "Data" pane on the right shows the schema with columns "Model", "Release(d)", and "iOS".

Select the table with the name: “**Availability and support lifespan of all iPhone models**” to get the list of iPhones along with their ‘release date’, ‘date discontinued’, ‘support ended date’, and lifespan.

Example 4: Get data from [Sales Detail](#) Excel file

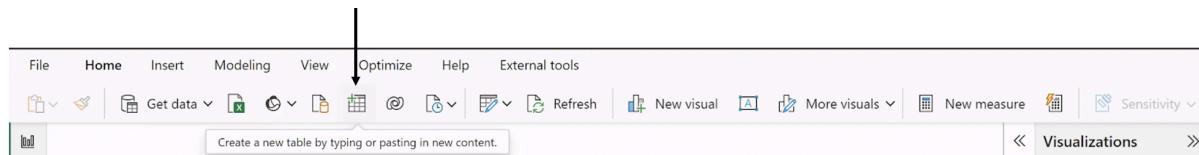
Download [this](#) dataset and get data in Power BI using the Excel workbook option in the Get Data menu.



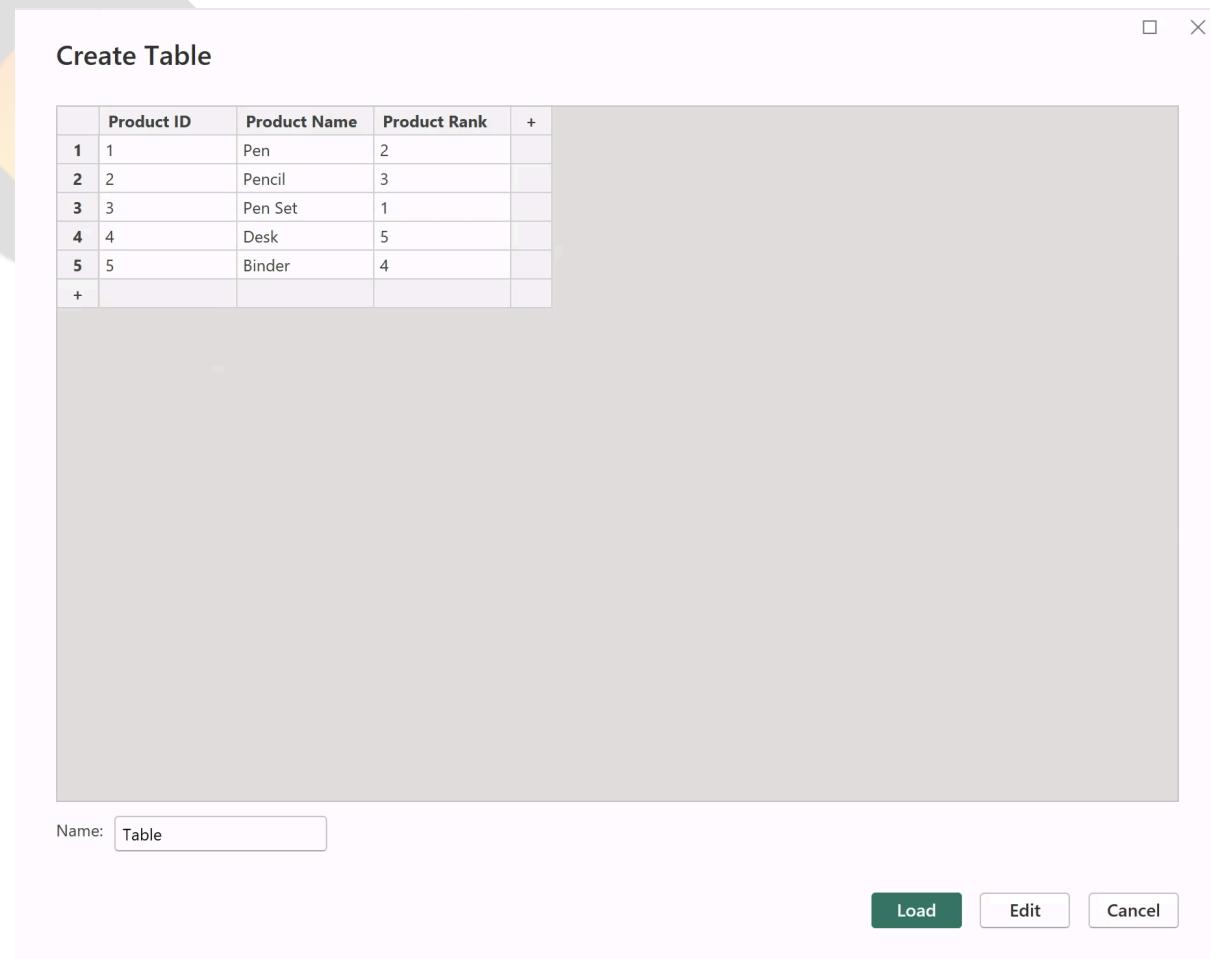
The screenshot shows the Power BI Data view. In the center, there is a table titled "Sales detail" with four columns: "Customer ID", "Sales Representative", "Date of sale", and "Sales_Channel". The data includes 20 rows of sales transactions. The "Data" pane on the right shows the schema with columns "Customer ID", "Sales Representative", "Date of sale", and "Sales_Channel".

Example 5: Create your own table in Power BI

Not everything needs to be imported in Power BI. We can also create our own table. In the ‘Home’ tab on the ribbon, we can find the option to ‘Enter Data’.



It provides an option to add our own data, as we have done in the image below:



The screenshot shows the 'Create Table' dialog box. It contains a table with the following data:

	Product ID	Product Name	Product Rank	+
1	1	Pen	2	
2	2	Pencil	3	
3	3	Pen Set	1	
4	4	Desk	5	
5	5	Binder	4	
	+			

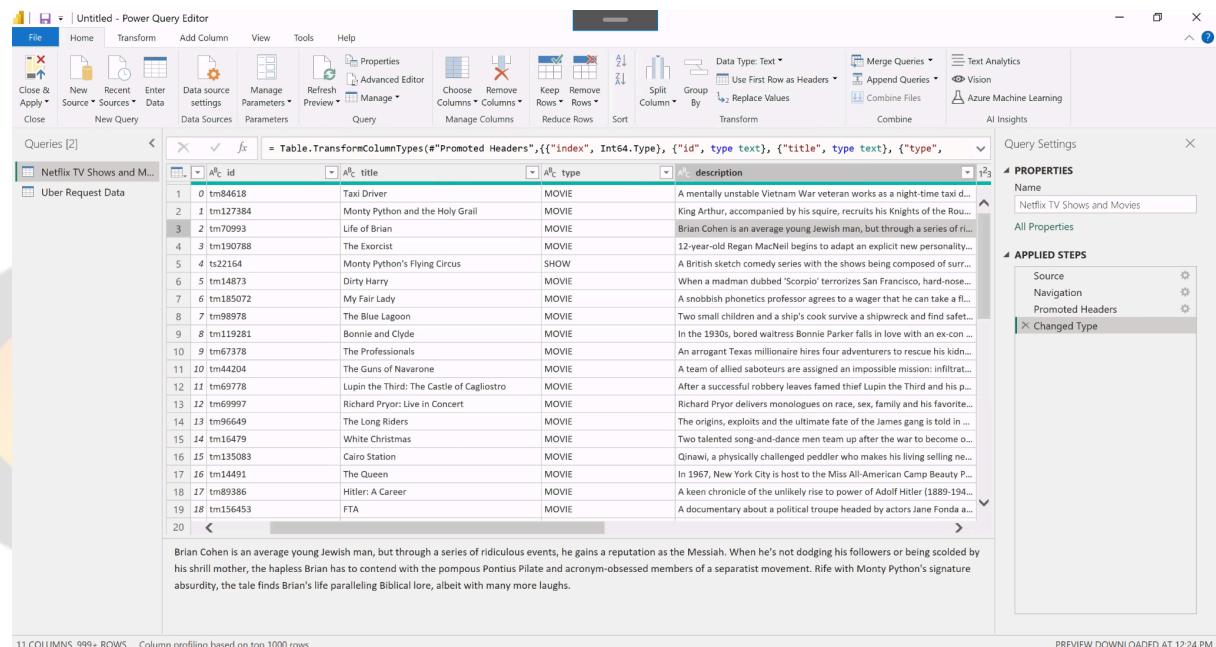
Below the table, there is a 'Name:' field containing 'Table'. At the bottom right are three buttons: 'Load' (green), 'Edit' (white), and 'Cancel' (white).

After we're done entering our own data, we need to select the 'Load' option to load the data. We can select the 'Edit' option to open the new table in the Power Query Editor for transformation.

4. Using Power Query Editor in Power BI for Data Transformation

For data transformation, Power Query Editor is used. It is available within Power BI Desktop. This is where we transform the data and make it more consumable and clean for our usage. When you Get Data in Power BI Desktop, it is first loaded in Power Query Editor **in the RAM**. Once all the transformations are done, data is loaded on Power BI Desktop and RAM is freed.

This is how the Power Query Editor looks like this:



The screenshot shows the Microsoft Power Query Editor interface. The ribbon at the top has tabs for File, Home, Transform, Add Column, View, Tools, and Help. The Home tab is selected. The main area displays a table with four columns: id, title, type, and description. The 'Promoted Headers' step is highlighted in the 'Applied Steps' pane on the right. The status bar at the bottom indicates '11 COLUMNS, 999+ ROWS' and 'PREVIEW DOWNLOADED AT 12:24 PM'.

On the top of the screen, the ribbon contains **7 tabs**:

1. **File:** It has multiple options for applying the changes and closing Power Query Editor (Close & Apply, Close, Apply) and for saving the file (Save, Save As).
2. **Home:** The Home tab provides essential tools for data transformation, such as connecting to data sources, editing queries, adding new queries, and managing query options.
3. **Transform:** The Transform tab offers a wide range of data transformation functionalities, including splitting columns, renaming columns, changing data types, and applying conditional transformations.
4. **Add Column:** The Add Column tab allows users to add new columns to the dataset using various methods, such as creating custom columns with calculated values, formatting columns, and adding conditional columns.
5. **View:** The View tab provides options for customizing the view of the data in the Power Query Editor. For example, users can show whitespace, column quality and column profile using the 'View' tab.
6. **Tools:** The tools tab has the Diagnose option which helps identify and troubleshoot issues with queries.
7. **Help:** The Help tab provides access to documentation, resources, and community support for Power Query Editor.

Note: Difference between 'Transform' and 'Add Column' tab

Analogy: Imagine a sculptor working with a piece of clay.

- The Transform tab functions like the tools used to shape the clay (cutting, molding, smoothing).
- The Add Column tab is like adding details or embellishments to the sculpture (carving patterns, adding features).

These tabs work in conjunction. You typically use the **Transform** tab for initial data cleaning and shaping, then leverage the **Add Column** tab to create new columns with custom calculations or derived data points based on the transformed data.

While the Transform tab modifies and restructures the existing data, the Add Column tab expands the dataset by introducing new columns derived from existing data.

For us to master Power BI, it is important to understand Power Query Editor in detail. Hence, we need to study in detail the use of each tab and the various functionalities in each.

1. Home

The Home tab contains common tasks like adding data sources, splitting columns, removing columns and Group by functionality.

There are multiple groups in the Home tab. The most useful ones are explained below:

- Close**: This has a functionality to close power query editor and apply changes.
- New Query**: This has options to add a new data source in the Power Query editor. This will let us query/edit both data sources from the same window.
- Data Sources**
- Parameters**
- Query**: This has the advanced editor option and data refresh options.
- Manage columns**: Here we can choose/remove columns.
- Reduce rows**: It has options to keep/remove rows.
- Sort**
- Transform**: It has options to split columns (by delimiter, by positions, by number of characters etc.), group by and replace.
- Combine**: Queries can be merged/appended using the options present in this group

2. Transform

It has options for manipulating columns of the dataset like transpose, split columns and find statistics in the data.

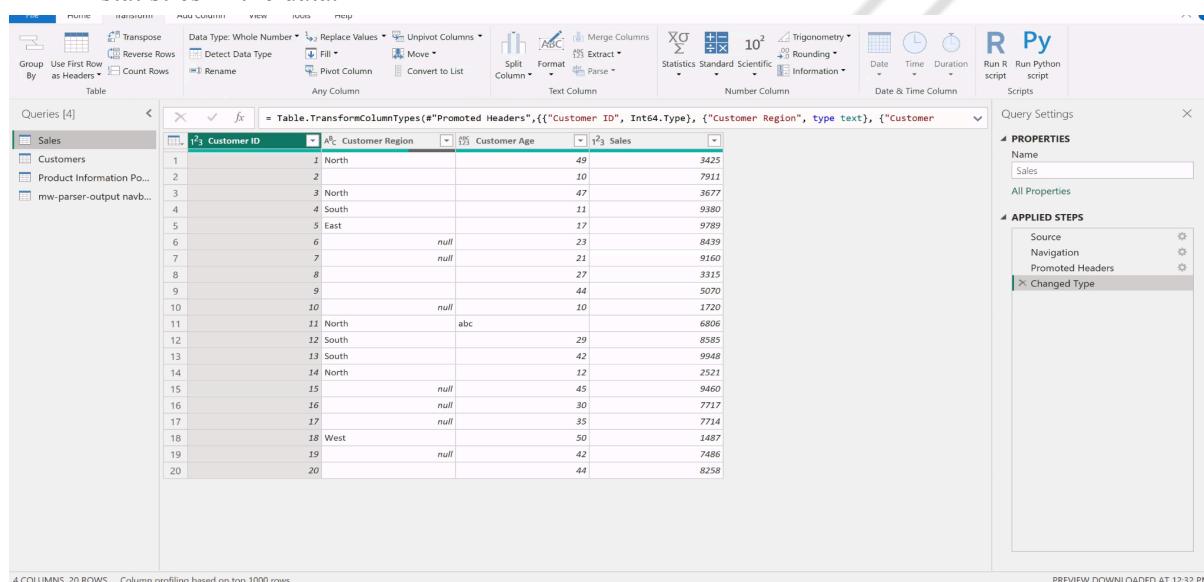
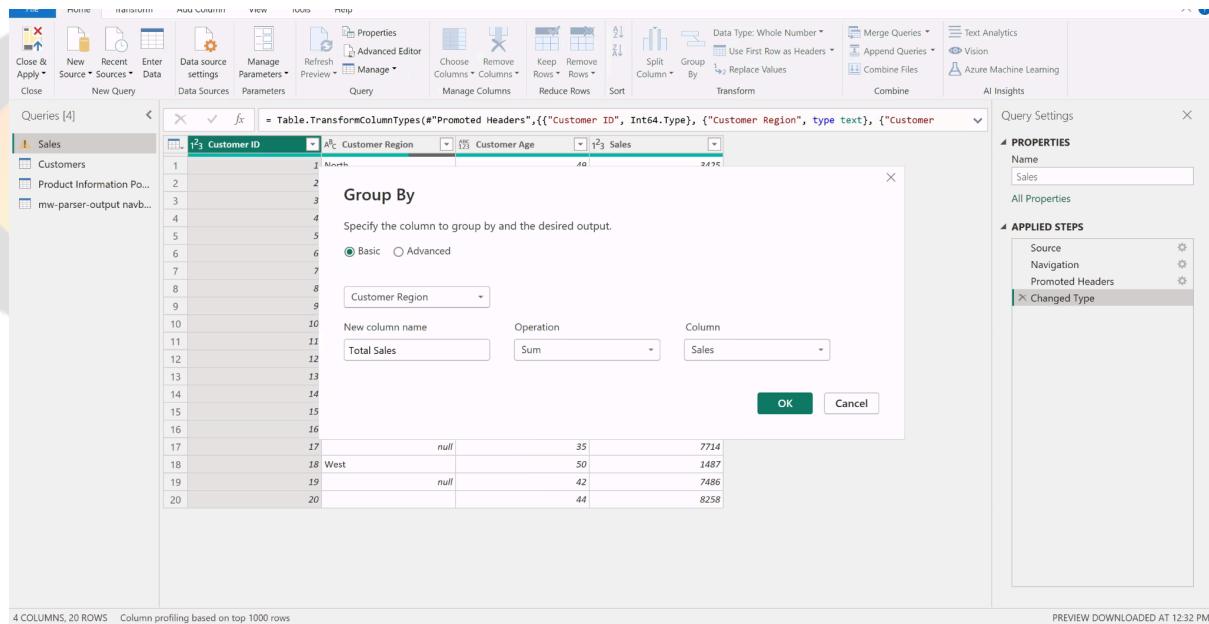


Figure: The image above shows some of the options in the Transform tab

Some of the useful features in the **Table** group of the Transform tab are:

- Group By:** It is used to aggregate/summarise data based on a column in the data.

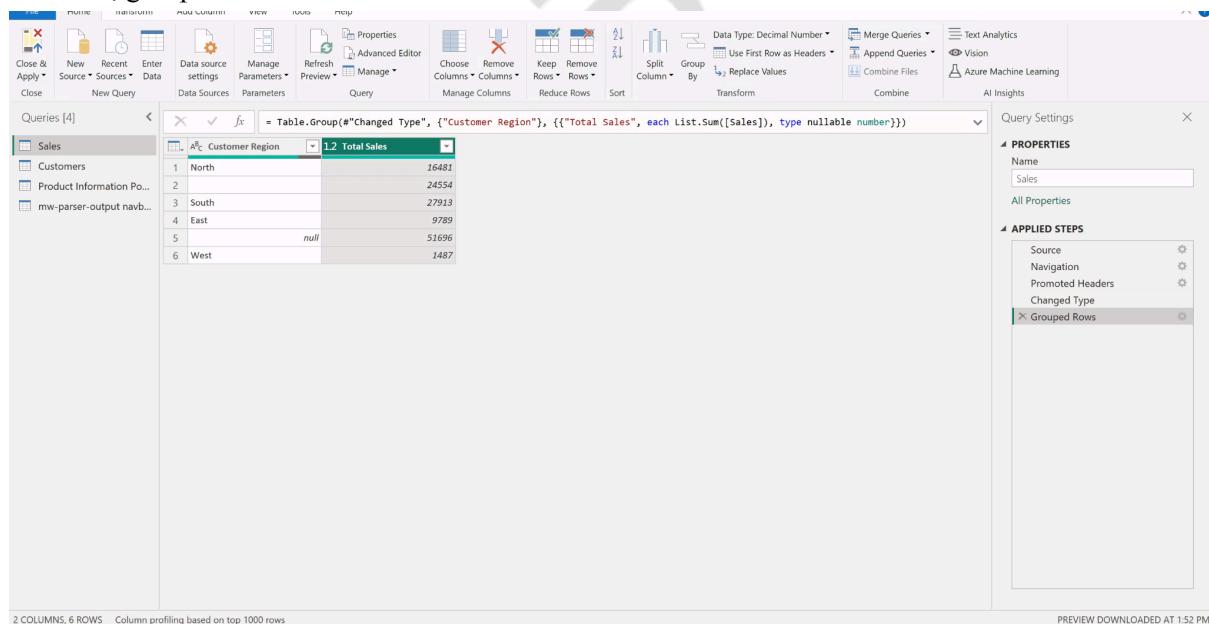
Example: In the Sales [Dataset](#) we are using, if we want to find total sales for each region, we can use ‘Group By’ on the Sales column to sum the sales for each region.



The screenshot shows the Power BI Data Editor interface. A 'Group By' dialog box is open over a table. The table has columns: Customer ID, Customer Region, Customer Age, and Sales. The 'Group By' dialog specifies 'Customer Region' as the grouping column, 'Total Sales' as the new column name, and 'Sum' as the operation. The main table preview shows data grouped by region.

	Customer ID	Customer Region	Customer Age	Sales
1		North	35	7714
2		West	50	1487
3		South	42	7486
4		East	44	8258
5		null		

The final, grouped columns would look like this:

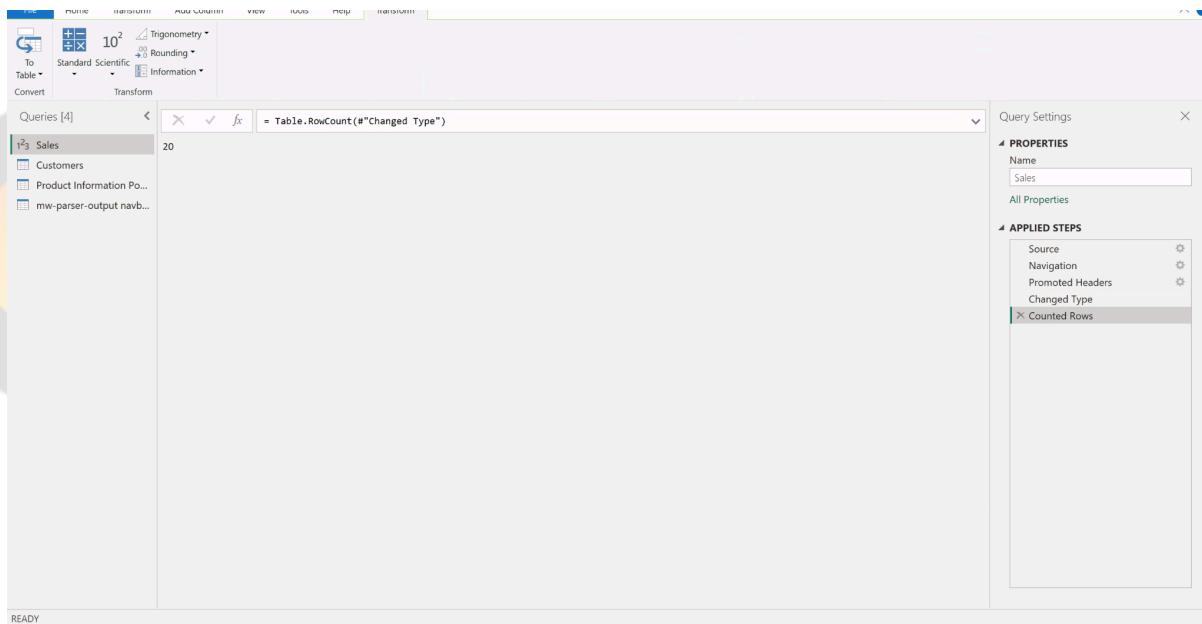


The screenshot shows the Power BI Data Editor interface after applying the 'Group By' operation. The table now has two columns: 'Customer Region' and 'Total Sales'. The 'Applied Steps' pane shows the 'Grouped Rows' step. The main table preview shows data grouped by region.

	Customer Region	Total Sales
1	North	16481
2	West	24554
3	South	27913
4	East	9789
5	null	51696
6		1487

- Transpose:** Just like it does in Excel, transpose helps us swap rows with columns and columns with rows.
- Use First Row as Headers:** This changes the first row to column name/header.
- Count Rows:** Counts the number of records in the dataset.

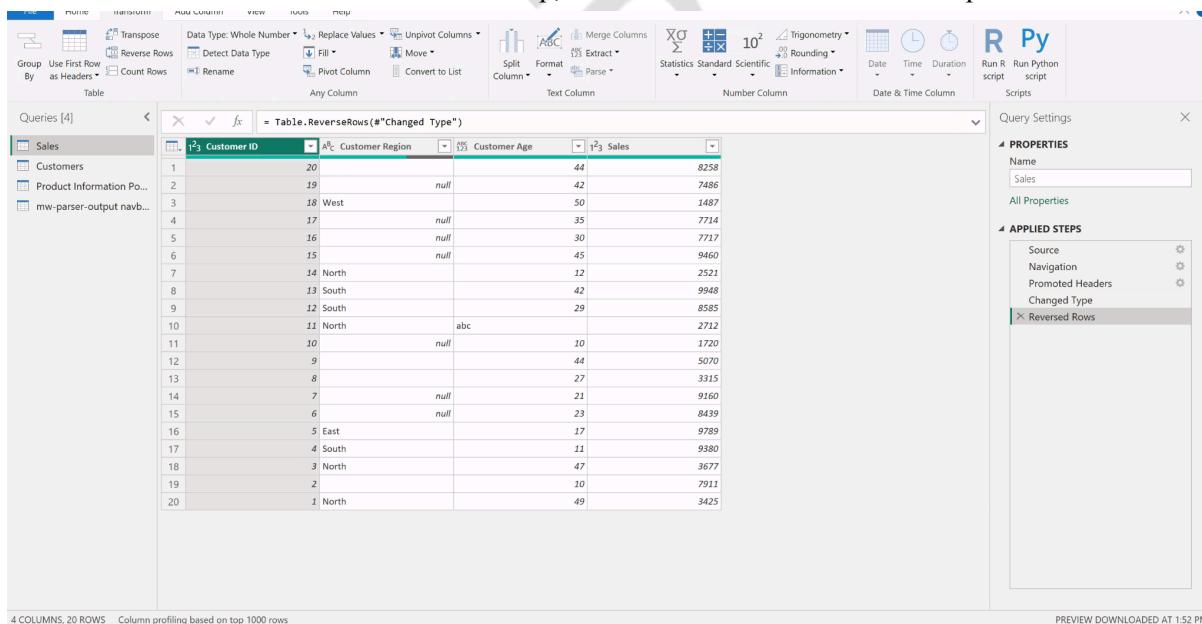
Example: In our [dataset](#), to count the number of rows we can select the Count Rows option in the Transform tab to count the total number of rows. By default, Power Query only shows the first 1000 rows.



The screenshot shows the Power Query Editor interface with the 'Transform' tab selected. The formula bar displays the expression `= Table.RowCount(#"Changed Type")`. The 'APPLIED STEPS' pane on the right lists the step 'Counted Rows'.

V. Reverse Rows: It reverses the order in which the rows are displayed.

Example: In our [dataset](#), if we want to reverse the row order and see the rows that are at the end of the dataset on top, we can select the Reverse Rows option.



The screenshot shows the Power Query Editor interface with the 'Transform' tab selected. The formula bar displays the expression `= Table.ReverseRows(#"Changed Type")`. The 'APPLIED STEPS' pane on the right lists the step 'Reversed Rows'.

Customer ID	Customer Region	Customer Age	Sales
1	20	44	8258
2	null	42	7486
3	West	50	1487
4	null	35	7714
5	null	30	7717
6	null	45	9460
7	North	12	2521
8	South	42	9948
9	South	29	8585
10	North	abc	2712
11	null	10	1720
12		44	5070
13		27	3315
14	null	21	9160
15	null	23	8439
16	East	17	9789
17	South	11	9580
18	North	47	3677
19	2	10	7911
20	North	49	3425

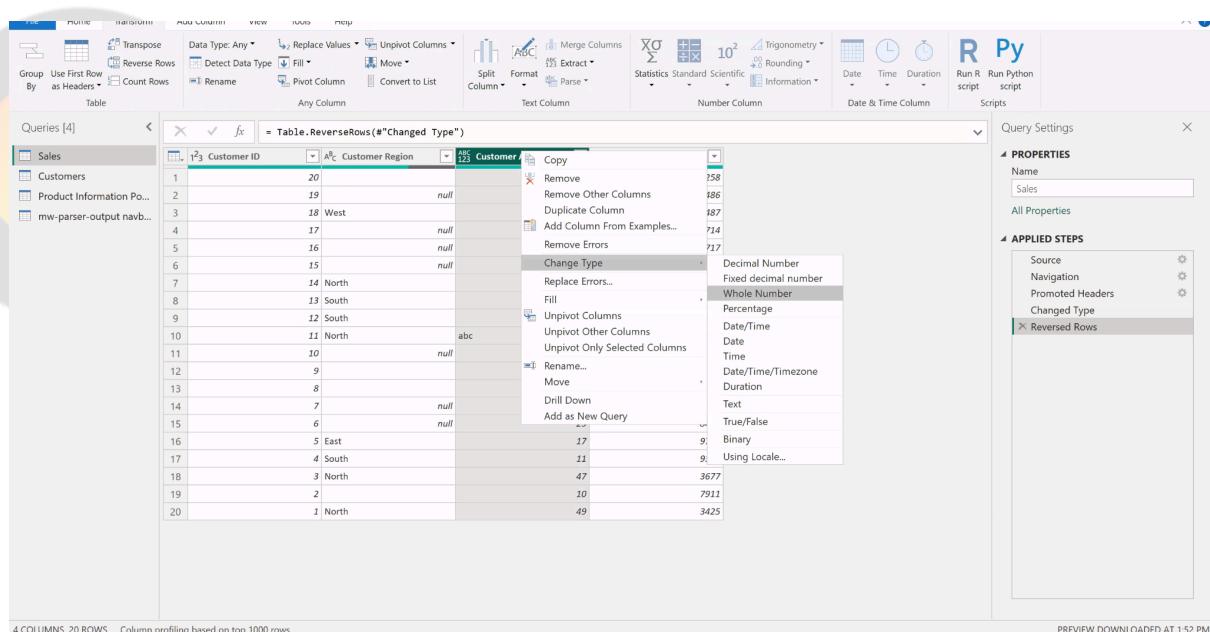
Note: Here the last index is at the top, this confirms that our dataset row order has been reversed.

Some important features in the **Any Column Group** in the **Transform** tab are given below. We can also right click any column in the Power Query Editor to see the following options to transform columns:

- Duplicate Column:** It is used to duplicate a column.

- b. **Change type:** It changes the type of data in the column. The data type of numerical columns needs to be the whole number or Decimal number.

Example: The Customer Age column needs to be changed to a Whole number data type. This can be done as shown below: right click on the Customer Age column → Change Data Type → Whole number

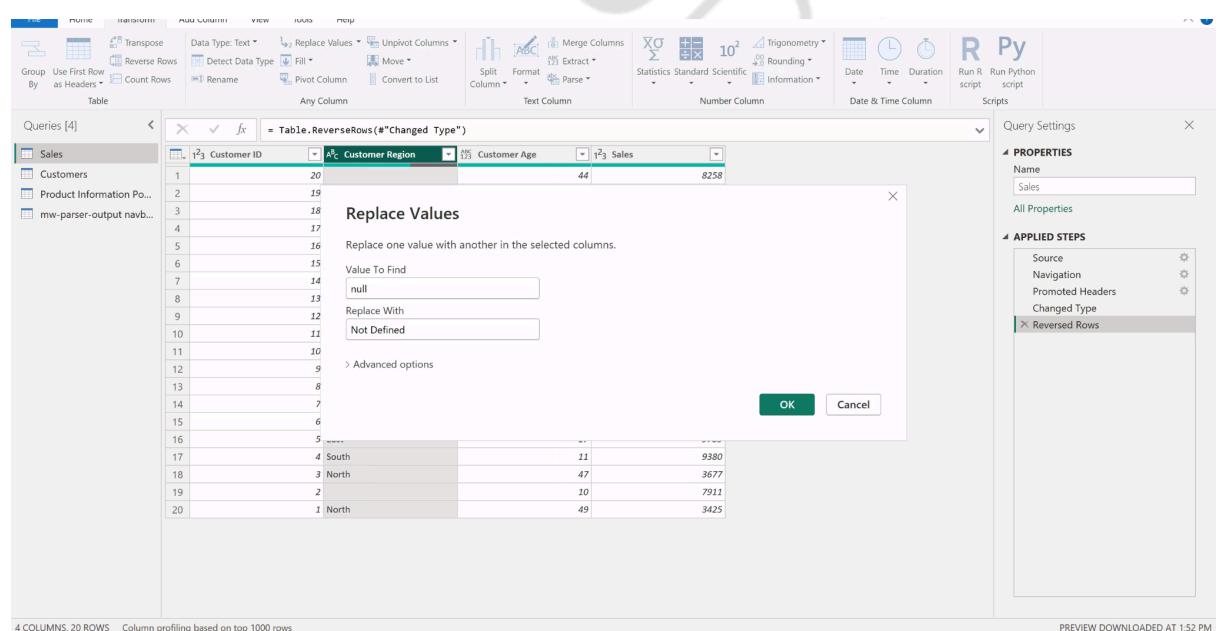


The screenshot shows the Power BI desktop interface with the 'Customer' table selected. The 'Customer Age' column is currently set to 'Any Column'. A context menu is open over this column, with 'Change Type' selected. Under 'Change Type', 'Whole Number' is highlighted. The 'APPLIED STEPS' pane on the right shows that the 'Changed Type' step has been applied. The preview pane at the bottom right indicates the preview was downloaded at 1:52 PM.

- c. **Rename:** It is used to rename the current column.

- d. **Replace Values:** It is used to replace column by a specified new column

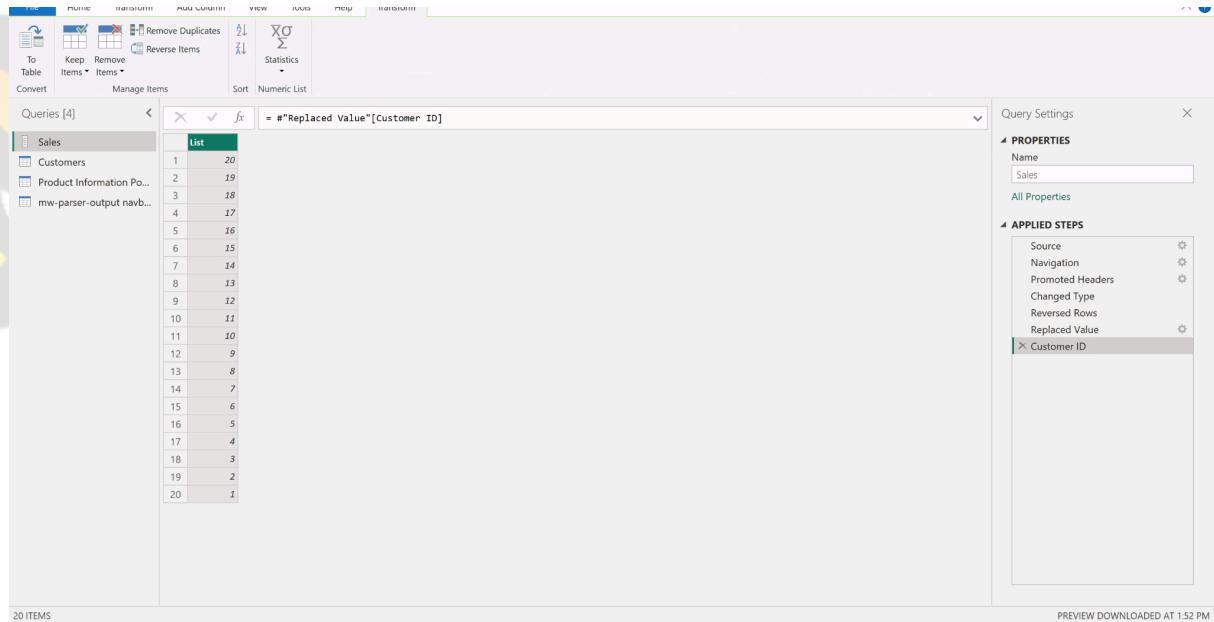
Example: In our [dataset](#) we can replace the null values in Customer Region with a fixed dummy value like 'Not Defined' as shown below.



The screenshot shows the Power BI desktop interface with the 'Customer' table selected. The 'Customer Region' column is currently set to 'Text'. A 'Replace Values' dialog box is open, showing 'Value To Find' as 'null' and 'Replace With' as 'Not Defined'. The 'OK' button is visible. The 'APPLIED STEPS' pane on the right shows that the 'Changed Type' step has been applied. The preview pane at the bottom right indicates the preview was downloaded at 1:52 PM.

- e. **Pivot columns:** This can be used to change the structure of the data. Data can be pivoted from rows into columns.
- f. **Unpivot columns:** It unpivots the data and brings it back to a vertical arrangement.
- g. **Convert to List:** It converts the selected data to a structured list format.

Example: The Customer ID column in the [dataset](#) can be converted to a list as shown below:



The screenshot shows the Power Query Editor interface. The ribbon at the top has the 'Transform' tab selected. In the center, there's a table view showing 20 items with values ranging from 1 to 20. On the left, under 'Queries [4]', the 'Sales' query is selected. On the right, the 'Query Settings' pane shows 'Name' set to 'Sales'. The 'Applied Steps' pane lists steps: 'Source', 'Navigation', 'Promoted Headers', 'Changed Type', 'Reversed Rows', 'Replaced Value', and 'Customer ID' (which is currently selected). The status bar at the bottom right says 'PREVIEW DOWNLOADED AT 1:52 PM'.

- h. **Move:** This helps us move the selected column to a new location in the datasets. We can also move columns by simply dragging the column header to the desired position.

The other groups in the Transform tab include Text Column, Number Column and Date & Time Column. These comprise options to manipulate data in each of these formats.

Note: The rest of the tabs in the Power Query Editor ribbon are covered in the next session.