<table>
<tr><td colspan="4">Name of Student : Pushkar Sane</td></tr>
<tr><td colspan="2">Roll No : 45</td><td colspan="2">LAB Assignment Number : 9</td></tr>
<tr><td colspan="4">Title of LAB Assignment : To learn handling of tables and Queries in Power BI</td></tr>
<tr><td colspan="2">DOP : 30/09/2024</td><td colspan="2">DOS: 15/10/2024</td></tr>
<tr><td>CO MAPPED :

CO6</td><td>PO MAPPED:

PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PSO1 , PSO2</td><td>FACULTY

SIGNATURE:</td><td>MARKS:</td></tr>
</table>

AIM: TO LEARN HANDLING OF TABLES AND QUERIES IN POWER BI.

THEORY:

Power BI is a powerful business intelligence tool that enables users to extract, transform, and analyze data from various sources. Learning how to handle tables and queries is crucial for making data-driven decisions, and it involves a few key concepts and operations.

1. Tables in Power BI

- **Definition**: A table in Power BI is a structured set of data consisting of rows and columns, where each row represents a record (e.g., a sales transaction) and each column represents an attribute or field (e.g., Invoice ID, Customer Type, Total).
- **Sources of Tables**: Tables can come from different data sources such as Excel, databases (e.g., SQL Server), web data, and APIs. In Power BI, these tables are loaded into the **Data Model** for analysis.

Operations on Tables:

- **Loading Tables**: Power BI can import data from different sources. You use the **Get Data** feature to connect to data sources and load tables.
- **Relationships between Tables**: When working with multiple tables, relationships define how data is related. This allows Power BI to perform calculations and aggregations across multiple tables. For example, linking a Customer ID from a Sales table to a Customers table lets you analyze sales based on customer attributes like gender or city.
- **Table Formats and Data Types**: Tables in Power BI consist of columns, each having a specific data type such as Text, Number, or Date. Correct formatting ensures accurate calculations and visualizations.

2. Power Query Editor

- **Purpose**: Power Query Editor is used to clean, transform, and reshape data before loading it into Power BI. It allows users to write queries without knowing SQL or other programming languages.
- **Key Functions**:
    - **Transformations**: These include renaming columns, filtering data, changing data types, and removing duplicate rows.
    - **Combining Data**: Power Query enables merging or appending data from different tables or queries.
    - **Data Cleaning**: Users can remove unwanted columns, filter unnecessary rows, split columns, or change text cases to clean data.

- **No Impact on Source Data**: Power Query performs transformations on data within Power BI without altering the original source files.

Common Query Operations:

- **Renaming Columns**: Helps in giving meaningful names to columns.
- **Filtering Data**: Allows you to remove rows that do not meet specific criteria, such as filtering out incomplete records or unwanted categories.
- **Changing Data Types**: Correct data types (e.g., text, number, date) must be applied to columns for proper analysis.
- **Grouping Data**: Data can be grouped to calculate summary statistics (e.g., sum of sales by branch).

3. Merging and Appending Queries

- **Merging Queries**: This operation is similar to a SQL JOIN. It combines data from two tables based on a common column (such as Invoice ID or Customer ID). This is useful when you want to combine related information stored in different tables. For example, you might merge a Sales table with a Products table to analyze product-level sales.
- **Appending Queries**: This operation stacks tables vertically, increasing the number of rows. It's used when you have data split across multiple tables with the same structure. For example, if sales data is stored in separate tables for different branches, you can append them to create a single dataset for analysis.

4. Working with Columns

- **Adding Custom Columns**: Power BI allows users to add calculated columns using expressions in DAX (Data Analysis Expressions). For instance, you can create a new column for the total sales amount by multiplying Quantity by Unit Price.
- **Removing and Reordering Columns**: Unnecessary columns can be removed to streamline the data model and improve performance. Reordering helps in organizing columns logically.

5. Advanced Query Features

- **Conditional Columns**: These allow users to create new columns based on conditions. For example, you might create a High Value column that labels sales transactions as high if the total value exceeds a certain threshold.

- **Applied Steps**: Power Query keeps track of each transformation step in the form of "applied steps." Users can edit, reorder, or remove steps to change how data is processed.

6. Performance Considerations

- **Efficient Queries**: Using filters and transformations smartly reduces the amount of data being loaded, which can improve the performance of reports and dashboards. For example, instead of loading entire datasets, apply filters to load only relevant data (e.g., filtering for data from a specific year or branch).
- **Query Folding**: Power BI attempts to push as many transformations as possible back to the data source (especially in relational databases), ensuring that operations are performed at the source level, which enhances performance.

7. Data Loading and Refresh

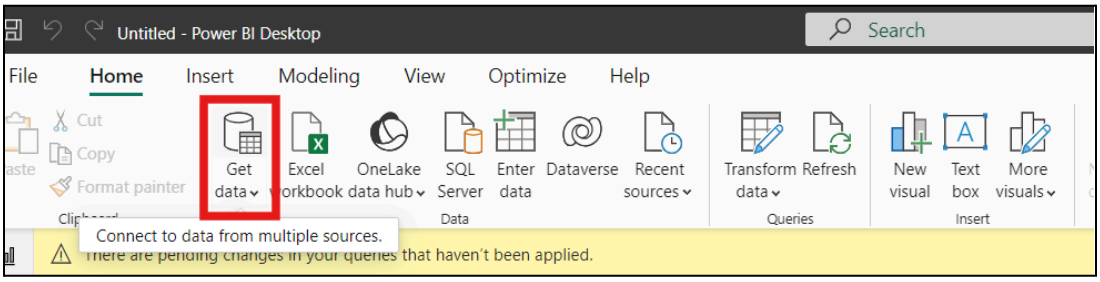
- After transforming data in Power Query, it is loaded into the Power BI **Data Model**. Data can be refreshed to reflect updates in the source data without redoing the transformations.
- You can set scheduled refreshes in Power BI Service to automatically update the data at defined intervals.

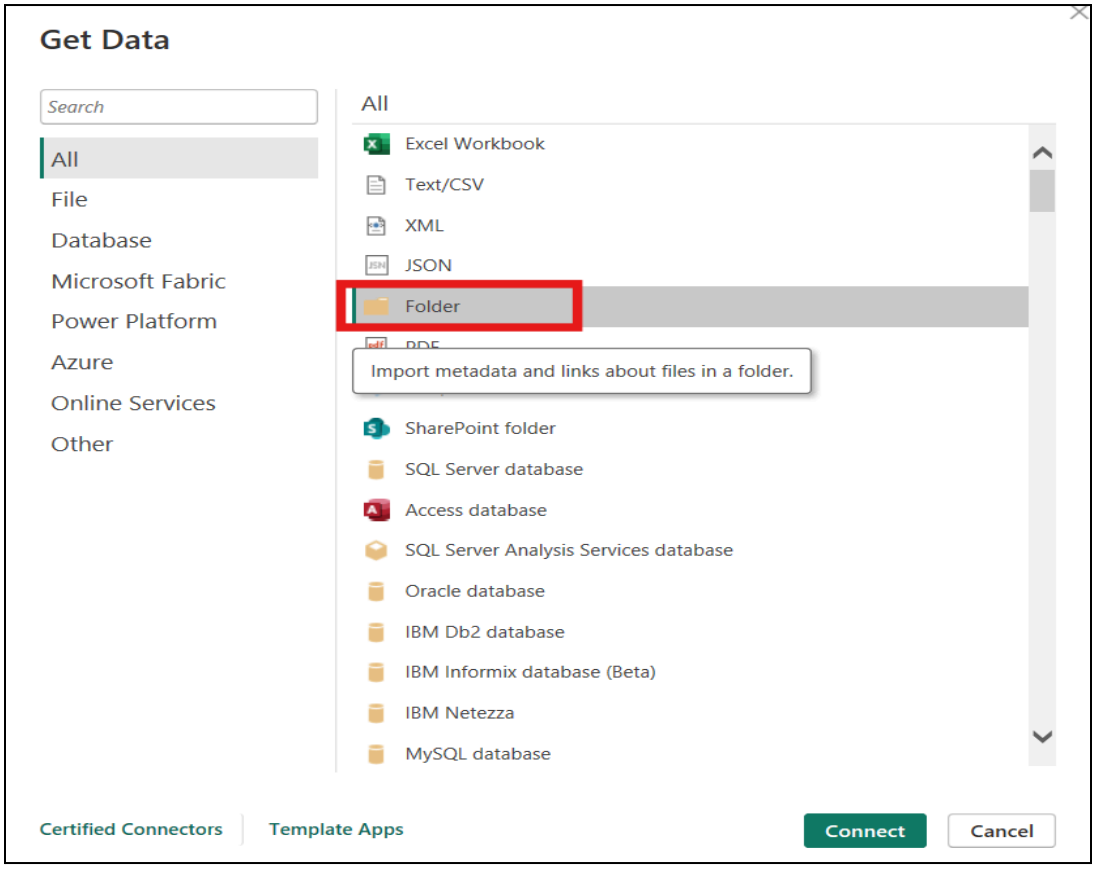
Dataset Link:

https://github.com/Ayushi0214/Datasets/blob/main/classic_models_dataset.zip

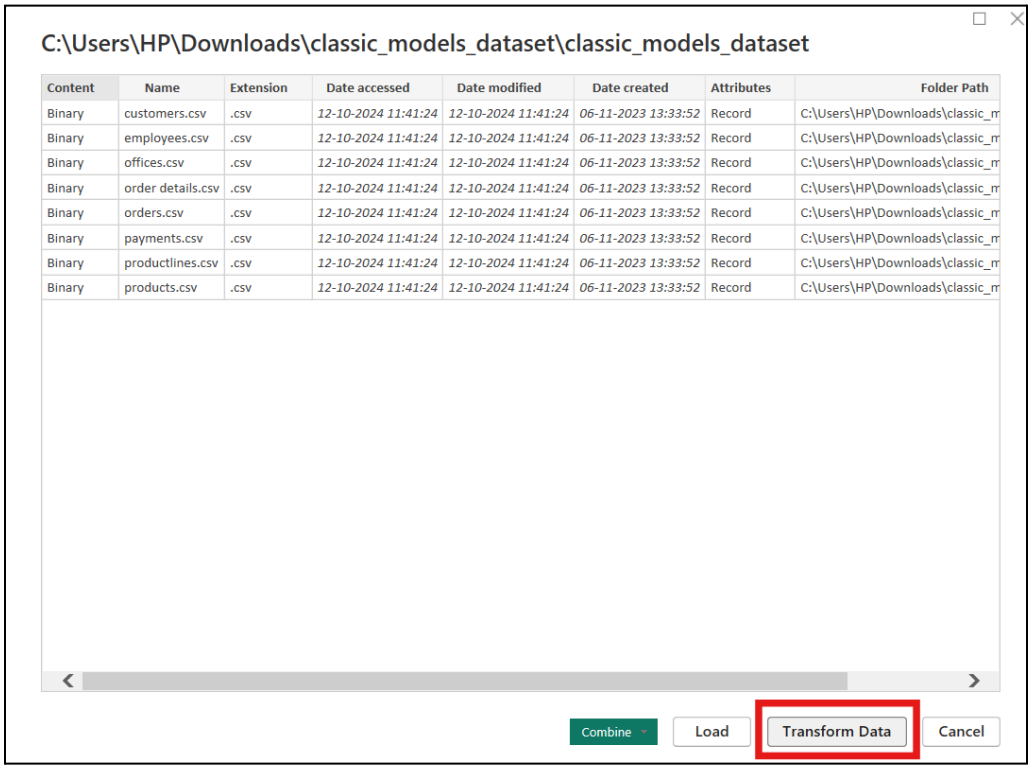
CODE/ STEPS:

Load Dataset:

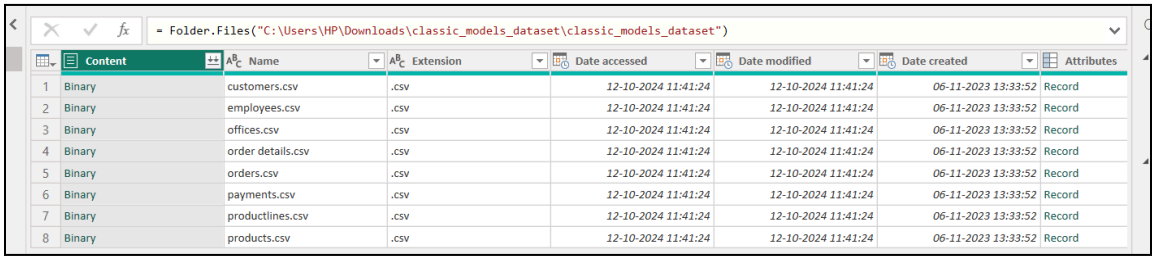




Browse your folder path:

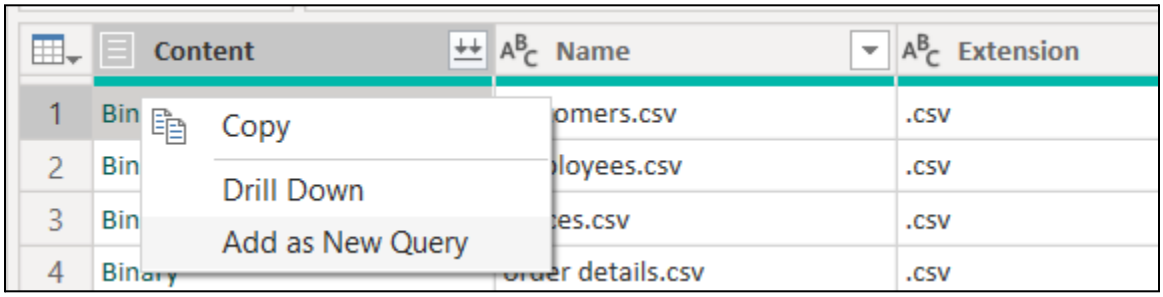


Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	customers.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m
Binary	employees.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m
Binary	offices.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m
Binary	order details.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m
Binary	orders.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m
Binary	payments.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m
Binary	productlines.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m
Binary	products.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record	C:\Users\HP\Downloads\classic_m

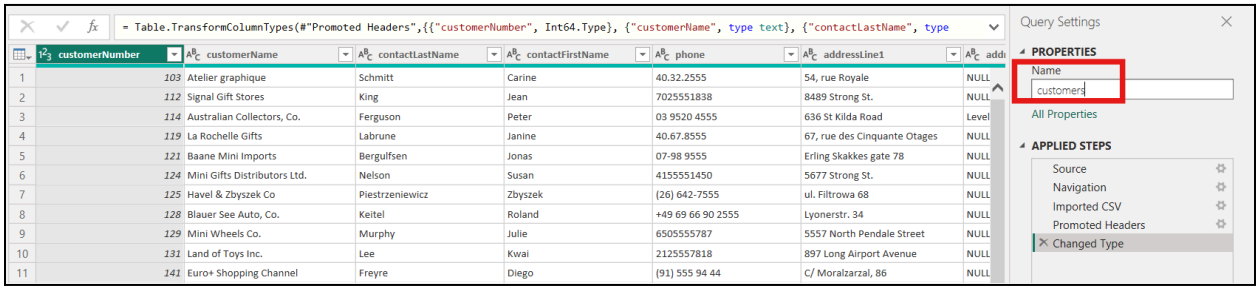


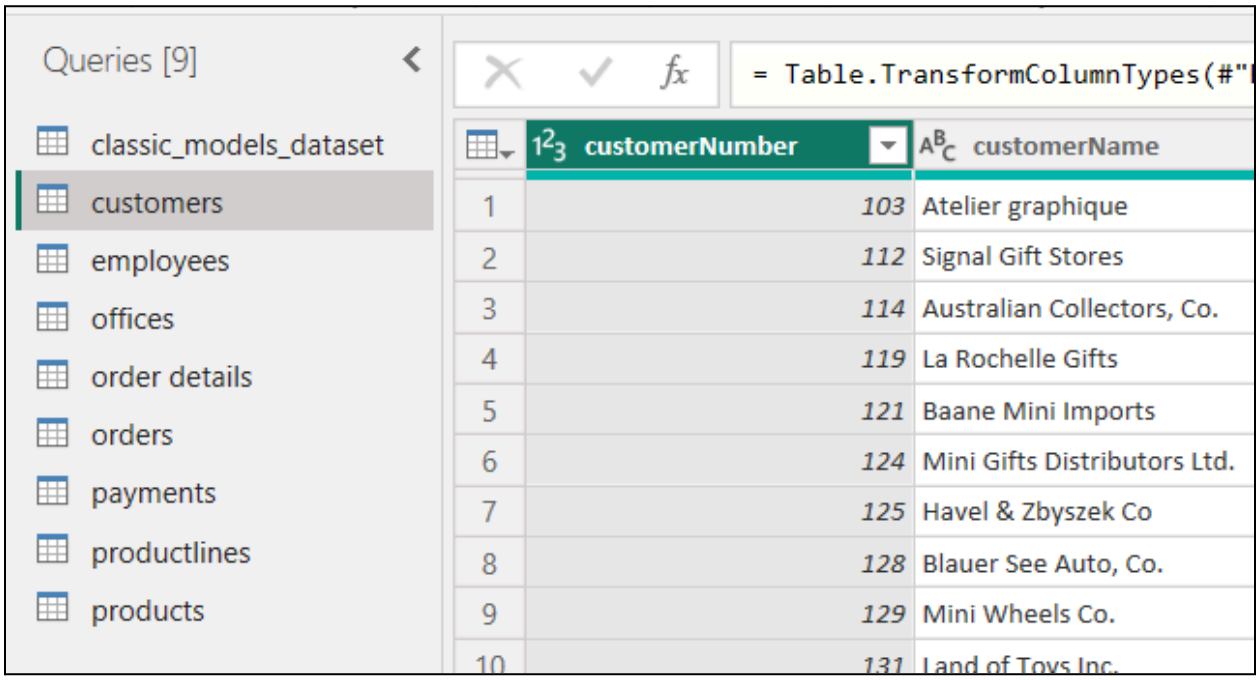
	Content	Name	Extension	Date accessed	Date modified	Date created	Attributes
1	Binary	customers.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record
2	Binary	employees.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record
3	Binary	offices.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record
4	Binary	order details.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record
5	Binary	orders.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record
6	Binary	payments.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record
7	Binary	productlines.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record
8	Binary	products.csv	.csv	12-10-2024 11:41:24	12-10-2024 11:41:24	06-11-2023 13:33:52	Record

Now add each table as a new query by right clicking on binary



Put the table name here and do the same for meaning datasets:



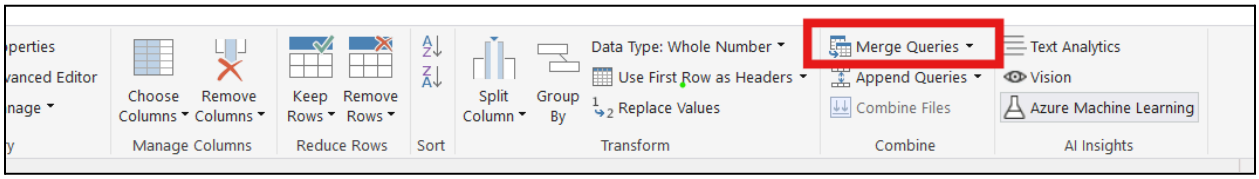


1. Merge Queries and Append Queries

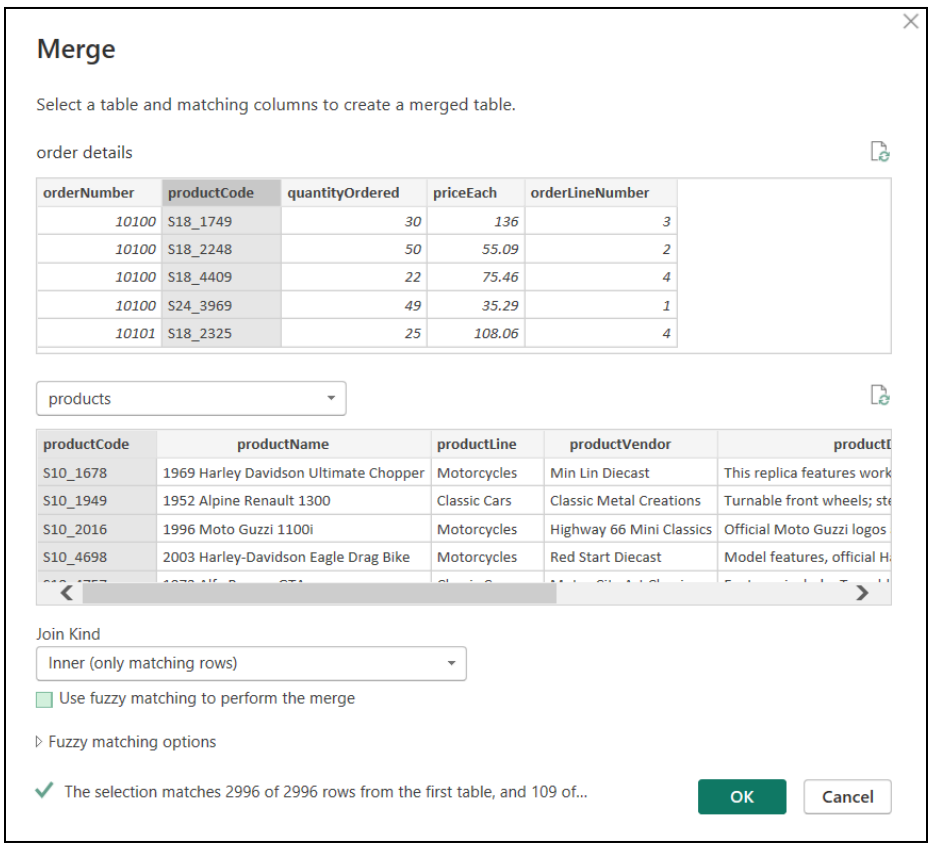
Merge Queries: Merging queries is equivalent to performing SQL joins. It combines columns from two tables based on a shared key or column. This is useful when you need information from both tables in a single table for analysis, like merging customer details with transaction data based on Customer Type.

- Merging Products and Order Details table based on common Product Code

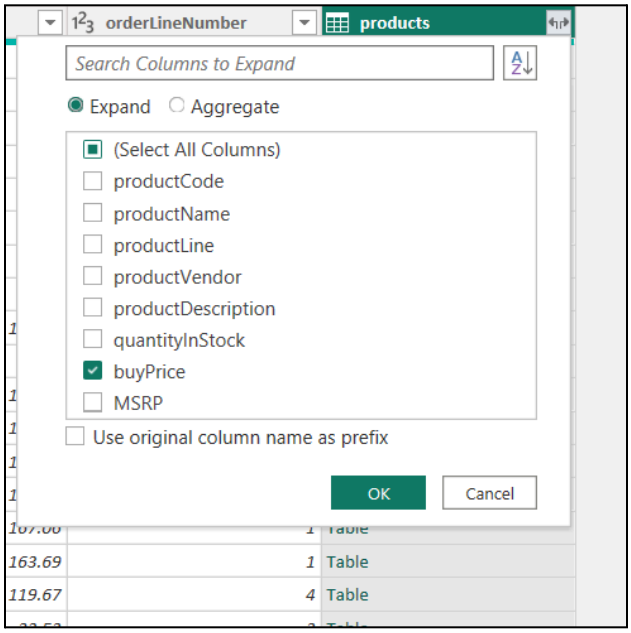
- In the Home Tab, click on "Merge Queries"



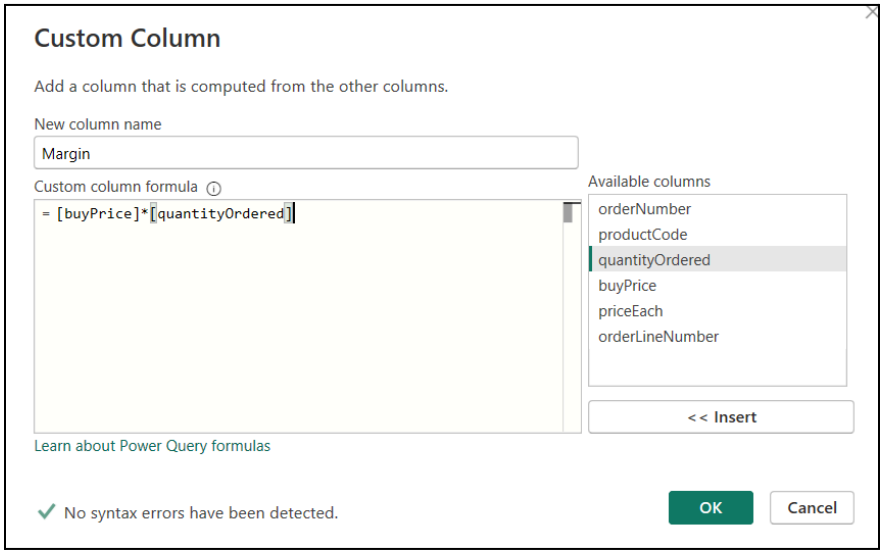
- In the dialog box select Product table and matching columns. Select the type of join(Here: Inner Join)



➢ Extracting only “Buy Price” from Products Table

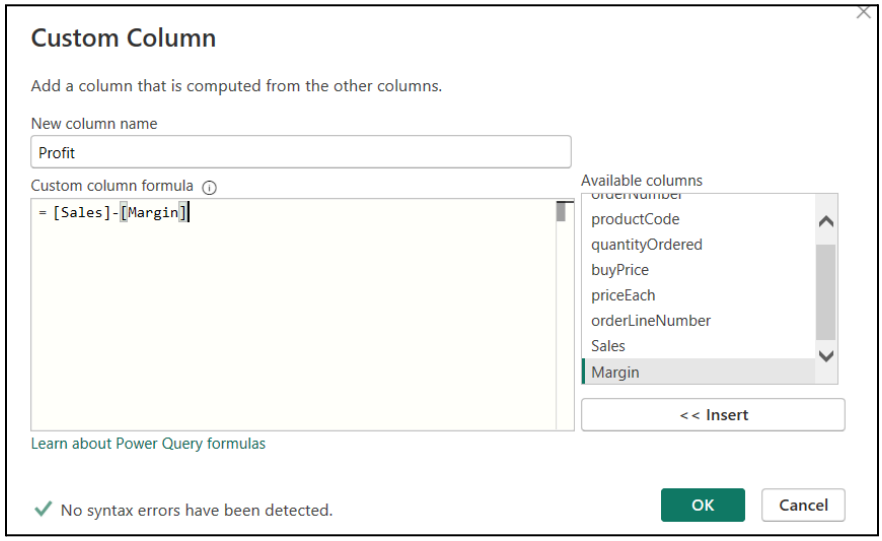


➢ Calculating margin from Buy Price and quantity by creating custom column



ered	buyPrice	priceEach	orderLineNumber	Sales	Margin
30	86.7	136	3	4080	2601
42	86.7	153	7	6426	3641.4
30	48.81	81.35	2	2440.5	1464.3
50	33.3	55.09	2	2754.5	1665
32	33.3	51.46	6	1646.72	1065.6
26	98.58	214.3	11	5571.8	2563.08
22	43.26	75.46	4	1660.12	951.72
28	43.26	81.91	8	2293.48	1211.28
39	68.99	105.86	5	4128.54	2690.61
49	21.75	35.29	1	1729.21	1065.75
27	91.02	172.36	4	4653.72	2457.54
25	58.48	108.06	4	2701.5	1462
33	58.48	115.69	4	3817.77	1929.84
50	85.68	127.84	2	6392	4284
26	72.56	167.06	1	4343.56	1886.56
31	72.56	163.69	1	5074.39	2249.36

➢ Calculating Profit



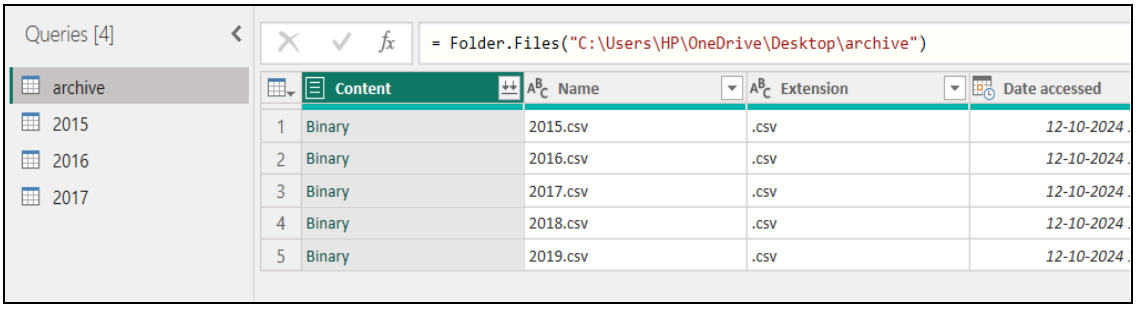
ber	Sales	Margin	Profit
3	4080	2601	1479
7	6426	3641.4	2784.6
2	2440.5	1464.3	976.2
2	2754.5	1665	1089.5
6	1646.72	1065.6	581.12
11	5571.8	2563.08	3008.72
4	1660.12	951.72	708.4
8	2293.48	1211.28	1082.2
5	4128.54	2690.61	1437.93
1	1729.21	1065.75	663.46
4	4653.72	2457.54	2196.18
4	2701.5	1462	1239.5
4	3817.77	1929.84	1887.93
2	6392	4284	2108

Append Queries: This operation is similar to a SQL UNION. It stacks rows from two or more tables on top of each other. Append is used when the structure of both tables is the same, but the data might represent different periods or entities (e.g., sales data from different branches or time periods).

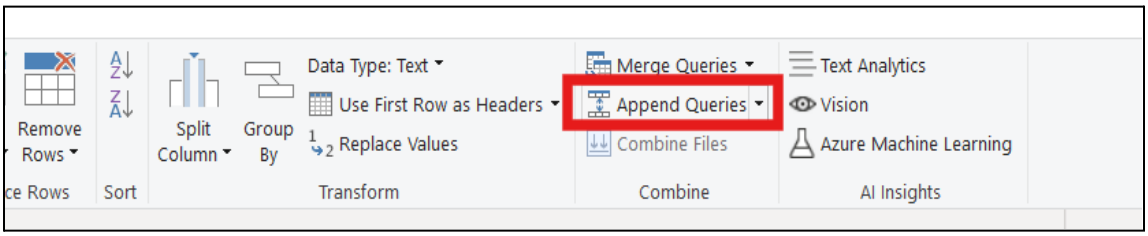
➢ Load new Dataset “World Happiness Report” to perform Append .

C:\Users\HP\OneDrive\Desktop\archive

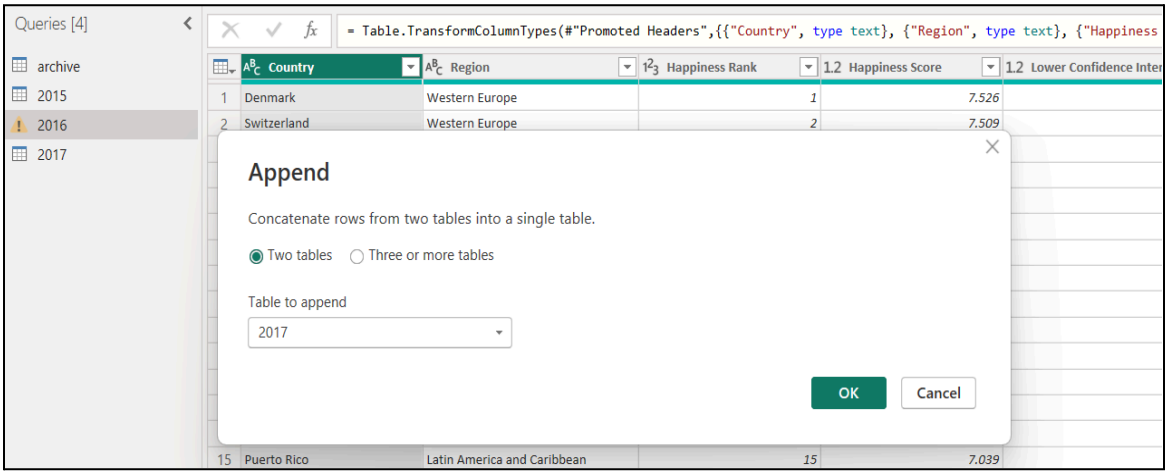
Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	2015.csv	.csv	12-10-2024 13:02:21	12-10-2024 13:02:21	27-11-2019 04:41:52	Record	C:\Users\HP\OneDrive\Desktop\archive\
Binary	2016.csv	.csv	12-10-2024 13:02:21	12-10-2024 13:02:21	27-11-2019 04:41:52	Record	C:\Users\HP\OneDrive\Desktop\archive\
Binary	2017.csv	.csv	12-10-2024 13:02:21	12-10-2024 13:02:21	27-11-2019 04:41:52	Record	C:\Users\HP\OneDrive\Desktop\archive\
Binary	2018.csv	.csv	12-10-2024 13:02:21	12-10-2024 13:02:21	27-11-2019 04:41:52	Record	C:\Users\HP\OneDrive\Desktop\archive\
Binary	2019.csv	.csv	12-10-2024 13:02:21	12-10-2024 13:02:21	27-11-2019 04:41:52	Record	C:\Users\HP\OneDrive\Desktop\archive\

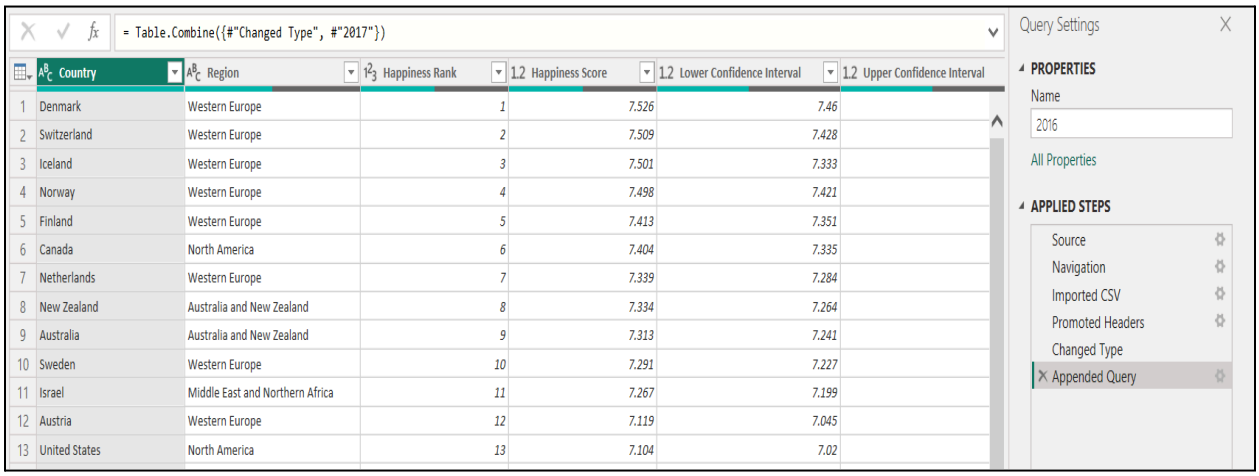


➢ In Home Tab , click on Append Queries



➢ Select Table to append and Click OK.



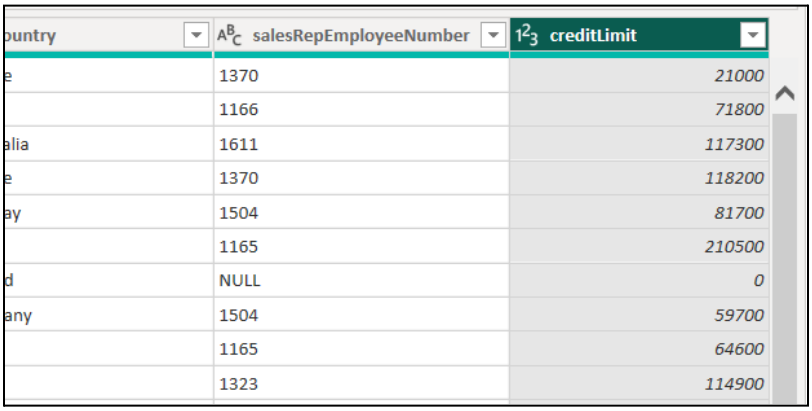


2. Column Formats

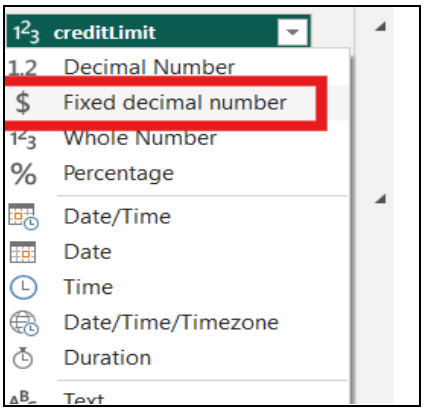
Column formatting refers to defining how data is displayed and processed. Different types of data require different formats to ensure proper analysis:

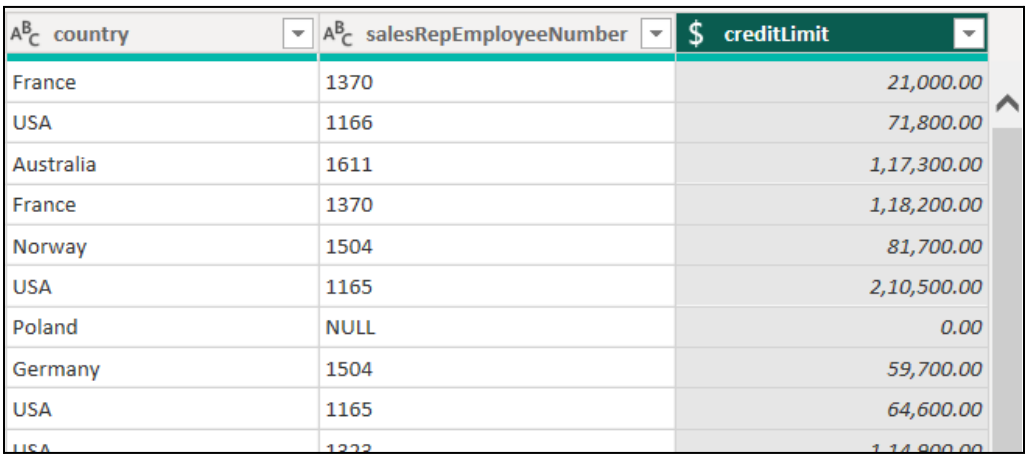
- **Text Format**: Used for categorical data like Invoice ID, Customer Type, or Branch. Text data is treated as labels and is not suitable for calculations.
- **Numeric Format**: Used for values that require mathematical operations. Examples include Quantity, Unit Price, Total, Gross Income, etc.
- **Date Format**: For time-related data like Date and Time, which allows the creation of time-based analyses like trends and comparisons.
- Proper formatting ensures correct data aggregation and calculations in reports and visualizations.

➢ Changing CreditLimit format from whole number to fixed decimal Number in Customers Table



ountry	salesRepEmployeeNumber	creditLimit
e	1370	21000
	1166	71800
alia	1611	117300
e	1370	118200
ay	1504	81700
	1165	210500
d	NULL	0
any	1504	59700
	1165	64600
	1323	114900

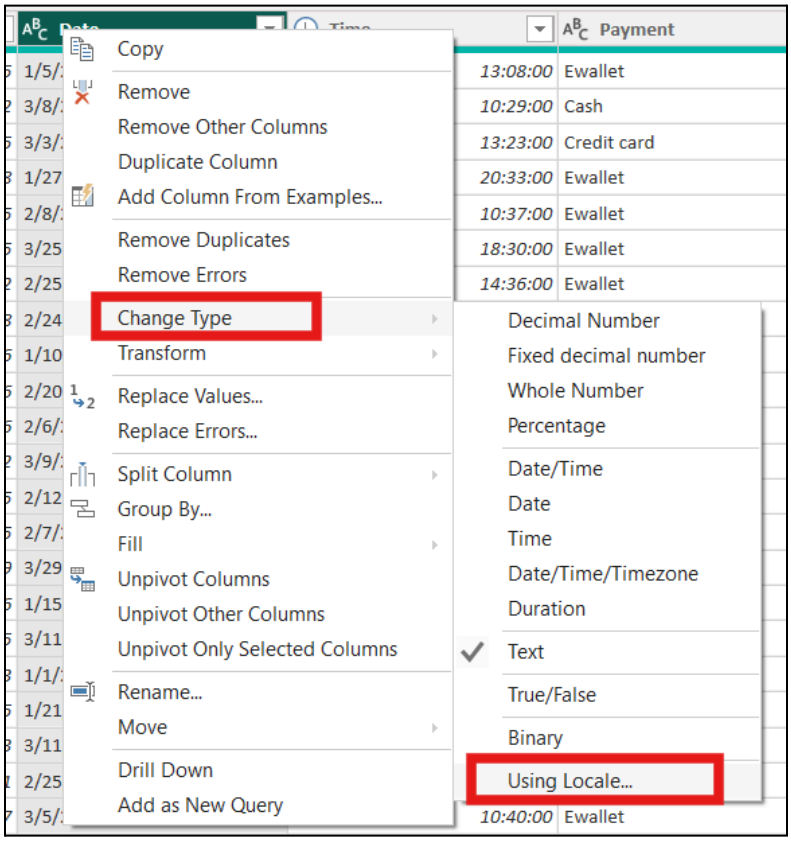


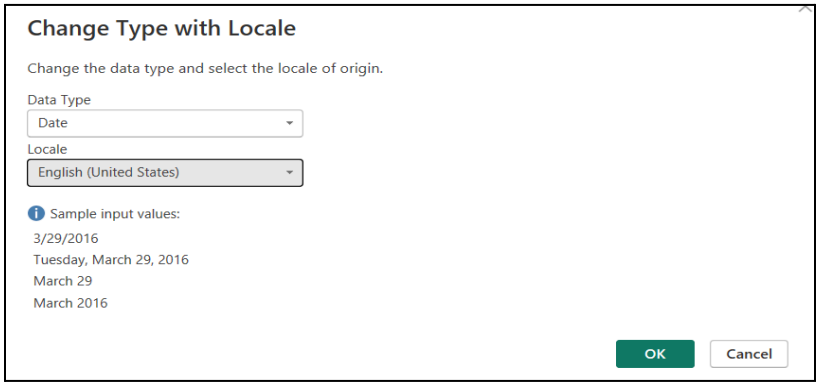


country	salesRepEmployeeNumber	creditLimit
France	1370	21,000.00
USA	1166	71,800.00
Australia	1611	1,17,300.00
France	1370	1,18,200.00
Norway	1504	81,700.00
USA	1165	2,10,500.00
Poland	NULL	0.00
Germany	1504	59,700.00
USA	1165	64,600.00
USA	1323	1,14,900.00

➢ Changing US Date type to Indian Date Format

	1.2 Tax 5%	1.2 Total	Date
7	26.1415	548.9715	1/5/2019
5	3.82	80.22	3/8/2019
7	16.2155	340.5255	3/3/2019
8	23.288	489.048	1/27/2019
7	30.2085	634.3785	2/8/2019
7	29.8865	627.6165	3/25/2019
6	20.652	433.692	2/25/2019
10	36.78	772.38	2/24/2019





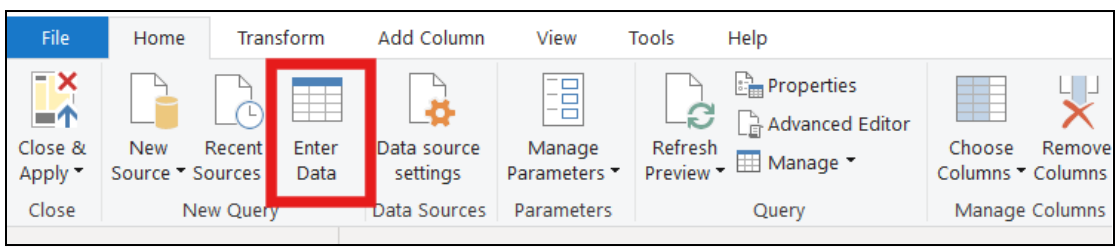
➢ Result:

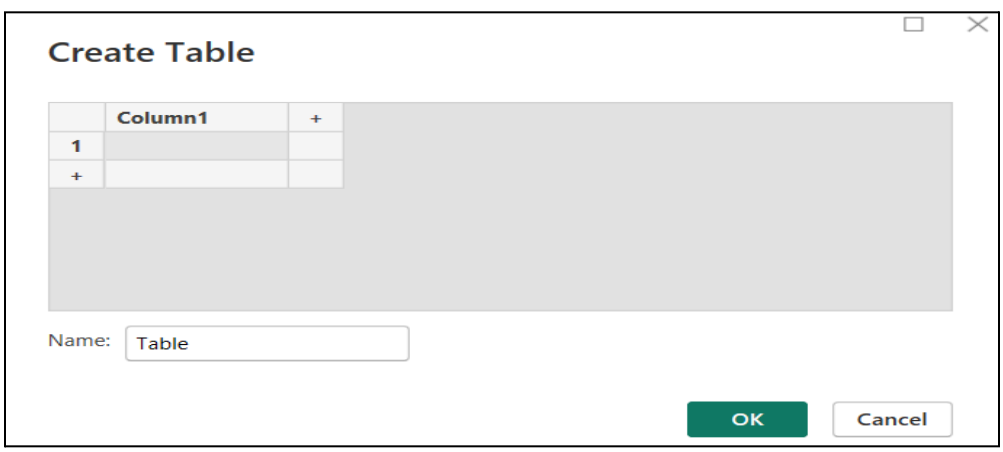
Tax 5%	Total	Date
26.1415	548.9715	05-01-2019
3.82	80.22	08-03-2019
16.2155	340.5255	03-03-2019
23.288	489.048	27-01-2019
30.2085	634.3785	08-02-2019
29.8865	627.6165	25-03-2019
20.652	433.692	25-02-2019
36.78	772.38	24-02-2019
3.626	76.146	10-01-2019
8.226	172.746	20-02-2019
2.896	60.816	06-02-2019
5.102	107.142	09-03-2019

3. Creating a Table

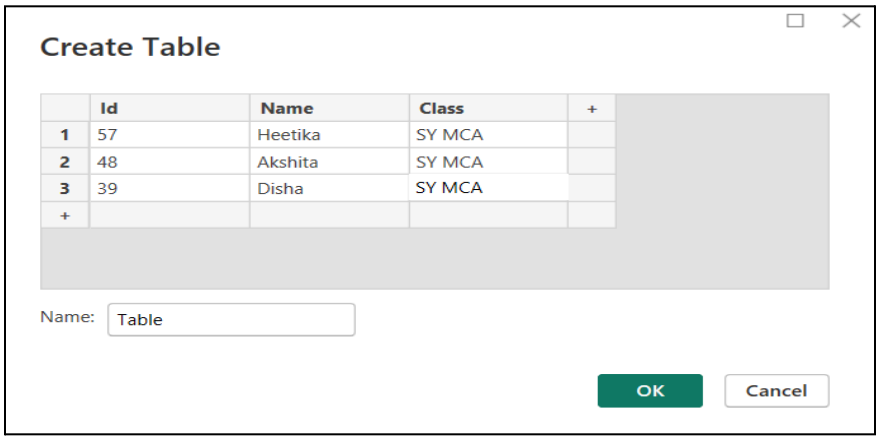
Creating a table in Power BI can be done by importing data or manually entering data. Manually created tables are often used for reference purposes, like a lookup table containing Product Line or Branch details. This helps in organizing and categorizing data, making it easier to perform analyses like product-line-wise or branch-wise sales performance.

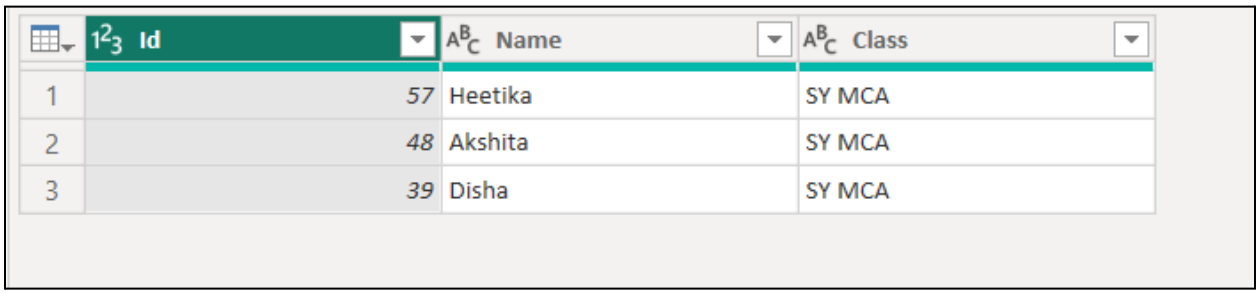
➢ In the Home Tab, click on “Enter Data”





➢ Add columns and rows





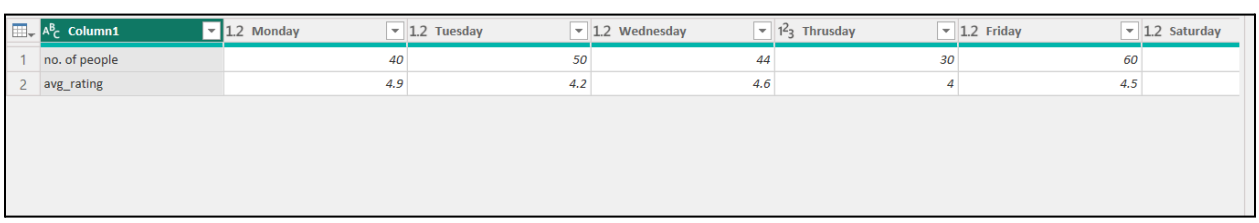
	Id	Name	Class
1	57	Heetika	SY MCA
2	48	Akshita	SY MCA
3	39	Disha	SY MCA

4. Pivoting and Unpivoting Data

Pivoting: Pivoting transforms data from rows into columns, allowing you to summarize and aggregate the data in a different structure. For example, pivoting sales data by Branch might transform rows of Branch A, B, and C into separate columns, making it easier to compare sales performance across branches.

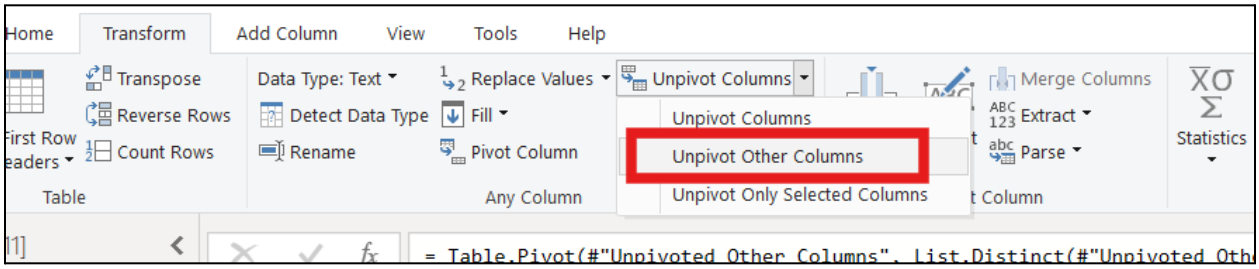
Unpivoting: Unpivoting is the opposite of pivoting. It converts columns into rows, which can help when analyzing multiple variables that were originally separate. For example, unpivoting Sales by year (where each year is a separate column) would make all sales records appear as rows, making time-based analysis easier.

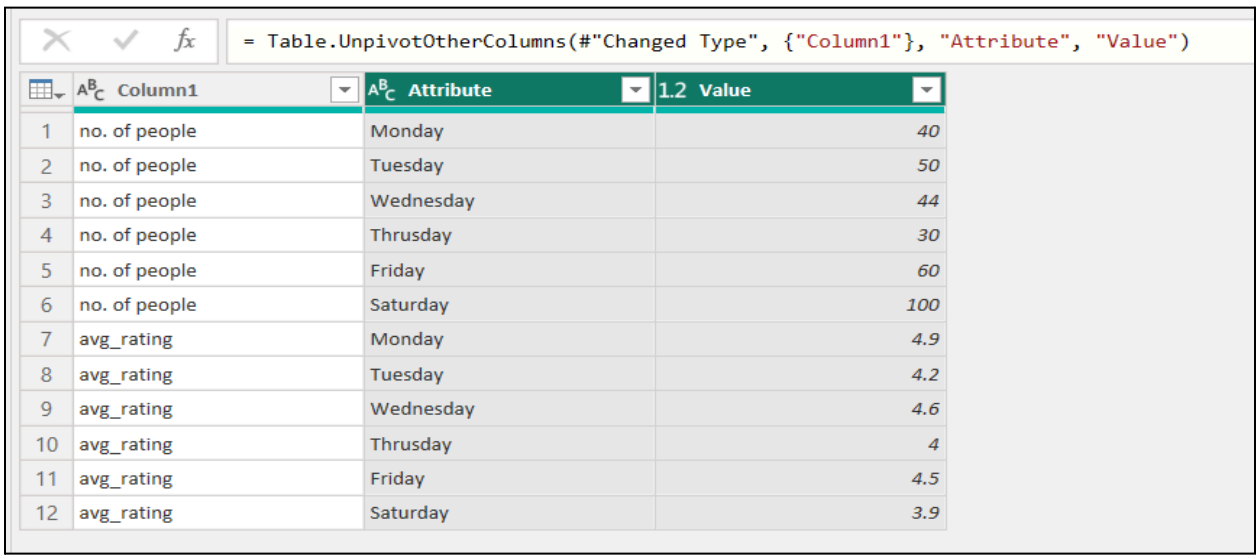
➢ Suppose we have data like this and we are asked to convert this table to vertical table:



	Column1	Monday	Tuesday	Wednesday	Thrusday	Friday	Saturday
1	no. of people	40	50	44	30	60	
2	avg_rating	4.9	4.2	4.6	4	4.5	

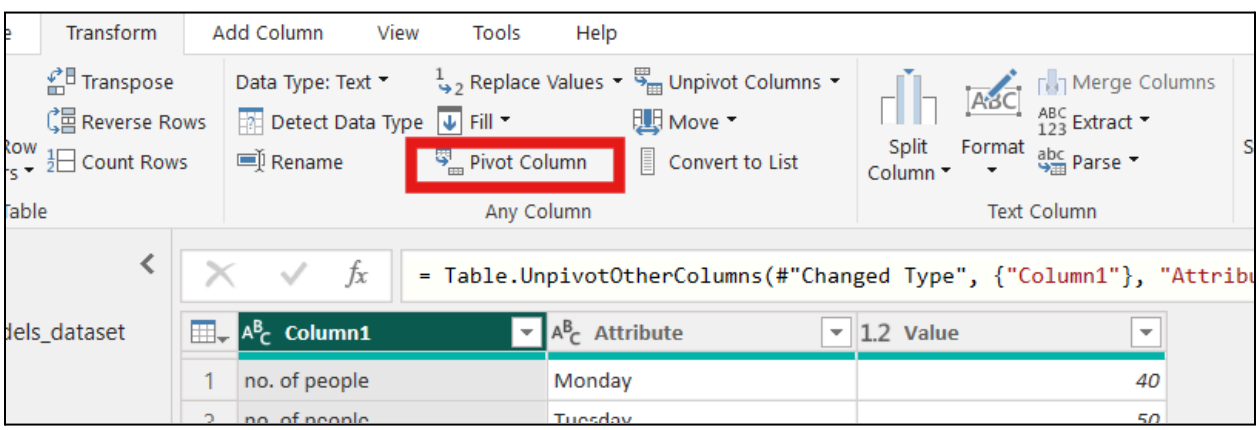
➢ Go to Transform > Unpivot other columns



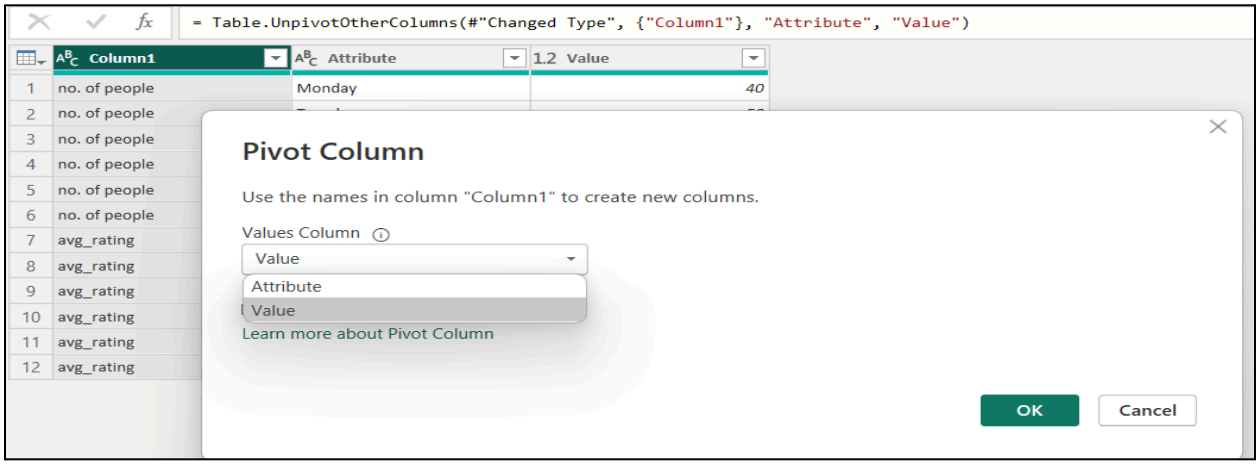


	Column1	Attribute	Value
1	no. of people	Monday	40
2	no. of people	Tuesday	50
3	no. of people	Wednesday	44
4	no. of people	Thrusday	30
5	no. of people	Friday	60
6	no. of people	Saturday	100
7	avg_rating	Monday	4.9
8	avg_rating	Tuesday	4.2
9	avg_rating	Wednesday	4.6
10	avg_rating	Thrusday	4
11	avg_rating	Friday	4.5
12	avg_rating	Saturday	3.9

➢ Now click on column and click on “Pivot Column” in Transform Tab



➢ Select value in drop down



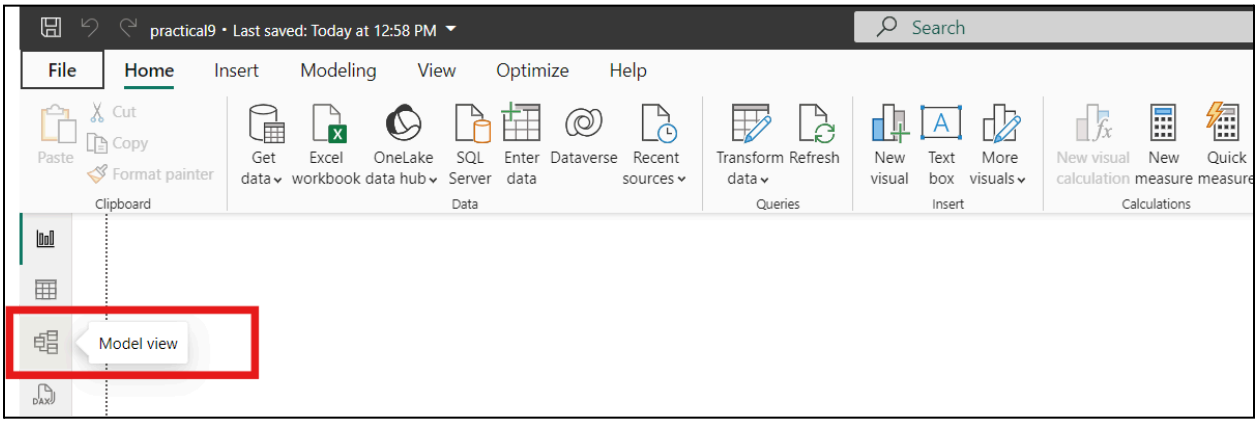
	Attribute	no. of people	avg_rating
1	Friday	60	4.5
2	Monday	40	4.9
3	Saturday	100	3.9
4	Thrusday	30	4
5	Tuesday	50	4.2
6	Wednesday	44	4.6

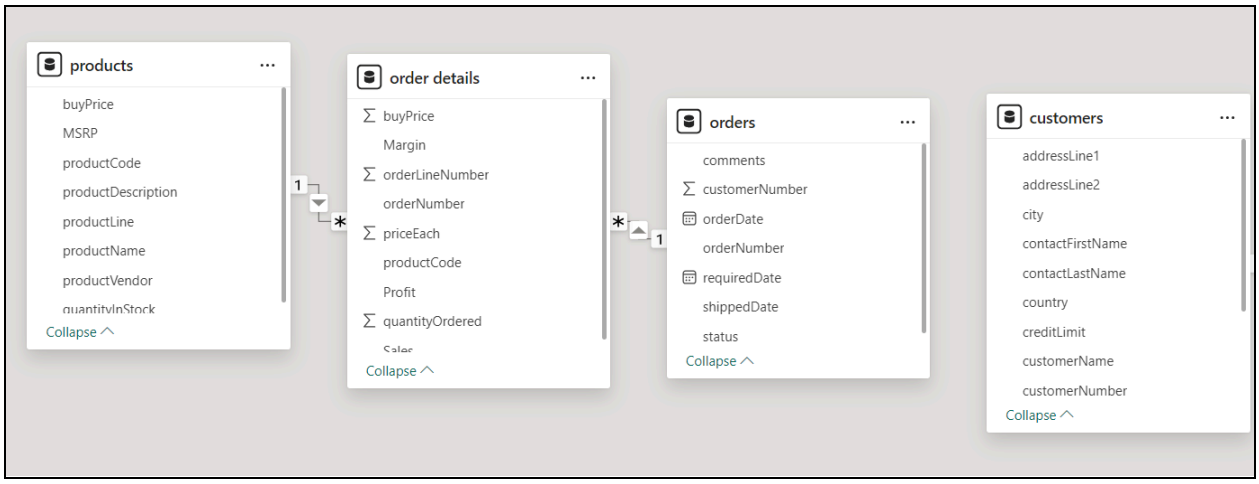
5. Data Model and Importance of Data Modeling

Data Model: A data model in Power BI defines how different tables relate to each other. This is essential for combining data from multiple tables into meaningful reports. The data model is created by establishing relationships between tables based on shared columns (called keys).

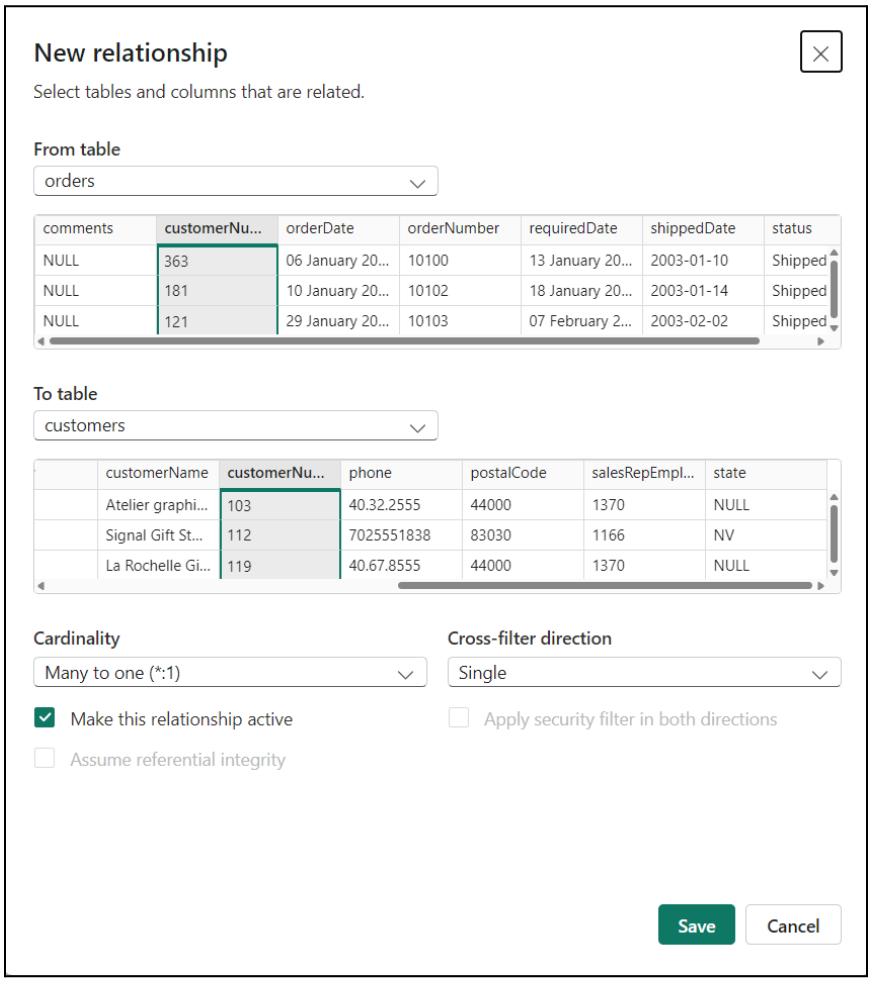
Importance of Data Modeling: A well-structured data model ensures accurate results, improves query performance, and simplifies data analysis. It enables slicing and dicing data in multiple ways (e.g., filtering sales by product category or customer type).

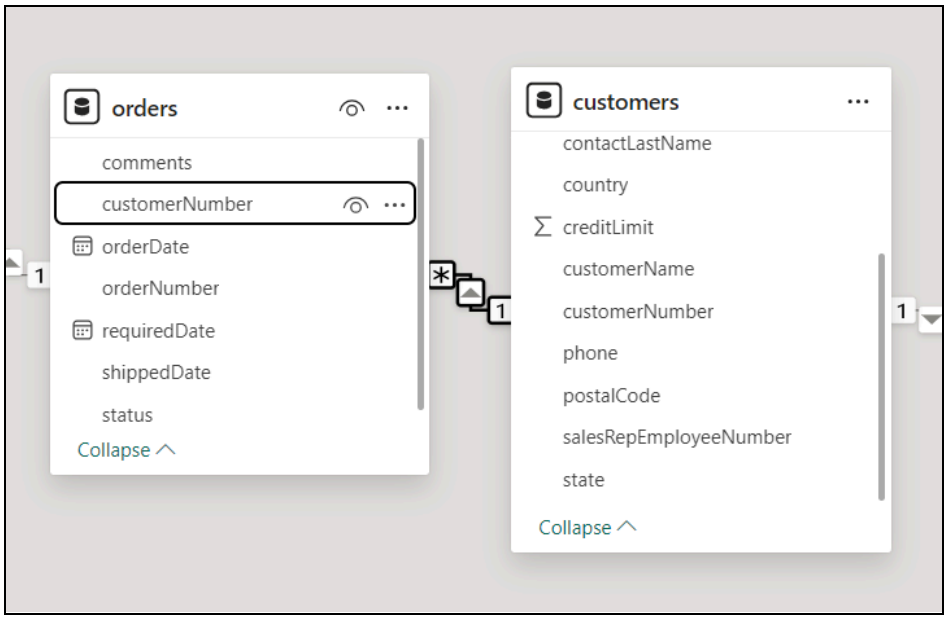
➢ Go to Model View





➢ Since the Orders table and Customers table have a common column "CustomerNumber". We can drag this column from Orders and put it in Customers table and it will create a relationship



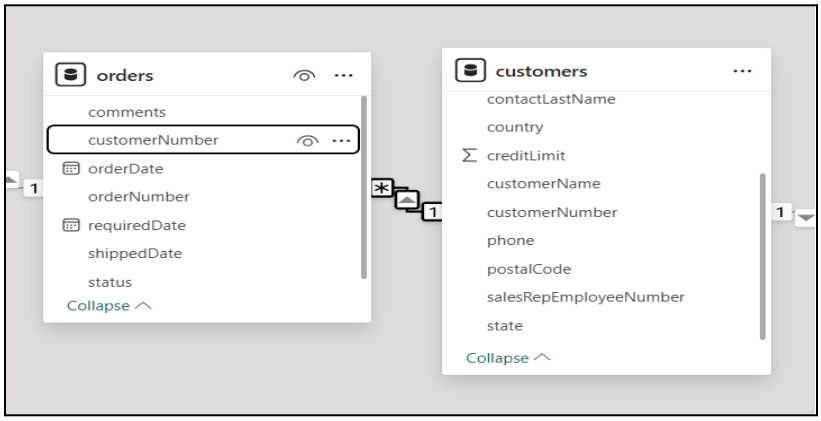


6. Managing Data Relationships

Relationships: Power BI enables you to create and manage relationships between tables. Relationships define how data in different tables connects, allowing Power BI to perform calculations and aggregations across them. Common relationship types are:

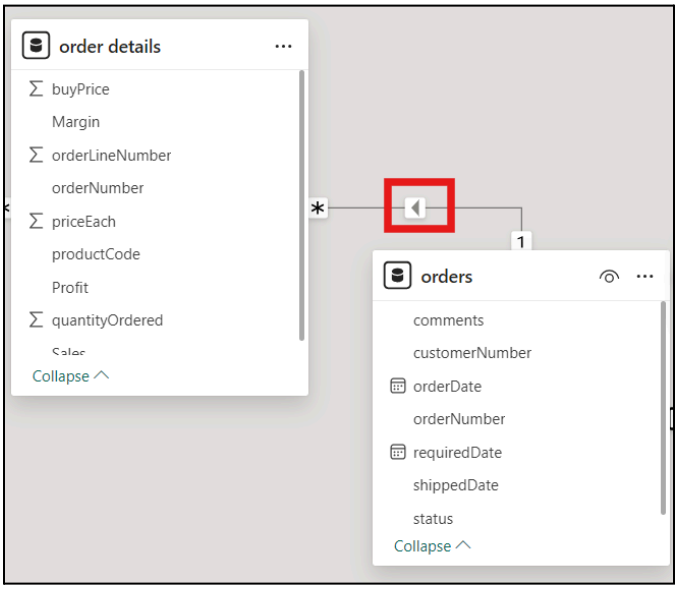
- **One-to-Many**: One record in a table (e.g., a Product ID in a product table) is related to multiple records in another table (e.g., multiple sales transactions involving that product).
- **Many-to-One**: The inverse of one-to-many.
- **Many-to-Many**: Allows for multiple matching records on both sides of the relationship (e.g., multiple branches and multiple cities in a geographical analysis).

➢ This is a One to Many relationship between Customers and Orders table. As one customer can have multiple orders

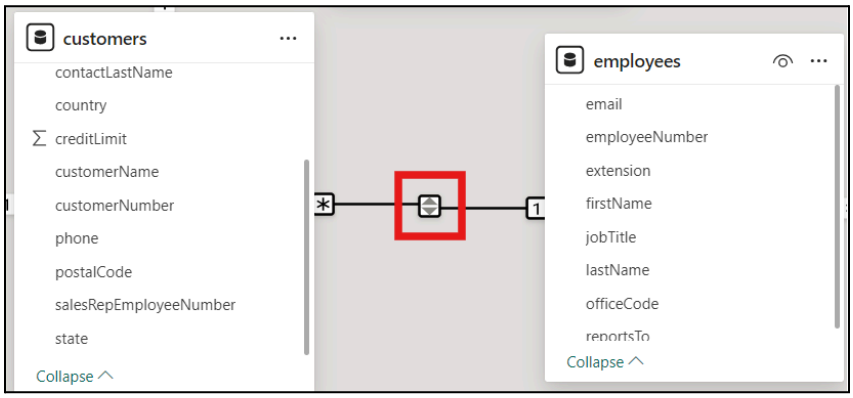


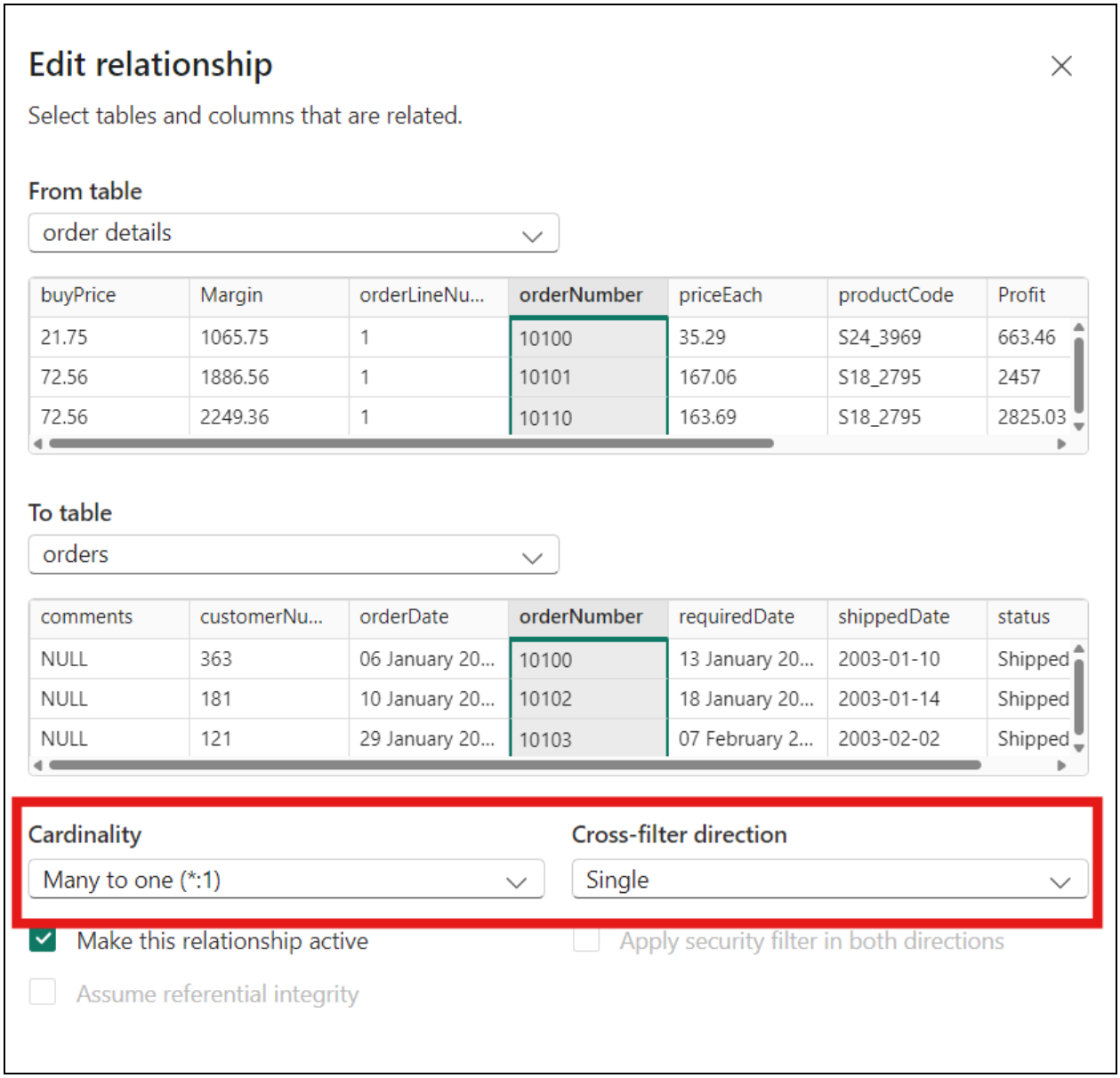
7. Cardinality and Cross-Filter Direction

- **Cardinality**: Refers to the nature of the relationship between two tables. Power BI supports three types of cardinality:
  - **One-to-Many**: One value in a column corresponds to many values in another column. This is the most common relationship type.
  - **Many-to-One**: Reverse of one-to-many, where multiple records in one table are linked to a single record in another.
  - **Many-to-Many**: Both tables can have multiple matching records, commonly used when there is no single unique key.
- **Cross-Filter Direction**: This determines how filters applied to one table affect the related table. There are two types:
  - **Single Direction**: Filters flow in one direction only. This is useful when one table is dependent on another (e.g., filtering products by product category).



  - **Both Directions**: Filters flow in both directions, which means a change in one table can filter data in both related tables. This is useful for complex models where both tables need to influence each other





CONCLUSION:

Handling tables and queries in Power BI involves various tasks like loading tables from data sources, transforming data using Power Query, combining and merging tables, and managing relationships between them. These operations are crucial for building an optimized and accurate data model, which serves as the foundation for generating insights and building reports.