Date:- 30/01/2024 **Business Intelligence** **Semester :- VI**
Class:- T.Y.Bsc.I.T. **Roll No:- IT21060**

Practical 2

Aim: Perform the Extraction Transformation and Loading (ETL) process to construct the database in the SQL server / Power BI.

- **Step 1: Data Extraction:**

  - The data extraction is first step of ETL.
  - There are 2 Types of Data Extraction :

    1. **Full Extraction:** All the data from source systems or operational systems gets extracted to staging area. (Initial Load)
    2. **Partial Extraction**: Sometimes we get notification from the source system to update specific date. It is called as Delta load.

    Source System Performance: The Extraction strategies should not affect source system performance.

- **Step 2: Data Transformation:**

  The data transformation is second step. After extracting the data there is big need to do the transformation as per the target system. I would like to give you some bullet points of Data Transformation.

  - Data Extracted from source system is in to Raw format. We need to transform it before loading in to target server.
  - Data has to be cleaned, mapped and transformed
  - There are following important steps of Data Transformation:

    1. **Selection:** Select data to load in target
    2. **Matching:** Match the data with target system
    3. **Data Transforming:** We need to change data as per target table structures

Real life examples of Data Transformation:

- Standardizing data: Data is fetched from multiple sources so it needs to be standardized as per the target system.
- Character set conversion: Need to transform the character sets as per the target systems. (Firstname and last name example)
- Calculated and derived values: In source system there is first val and second val and in target we need the calculation of first val and second val.

- Data Conversion in different formats : If in source system date in in DDMMYY format and in target the date is in DDMONYYYY format then this transformation needs to be done at transformation phase.

Step 3: Data Loading:

- Data loading phase loads the prepared data from staging tables to main tables.

➢ ETL Process in Power BI

1. Remove other columns to only display columns of interest

In this step you remove all columns except **ProductID, ProductName, UnitsInStock, and QuantityPerUnit**

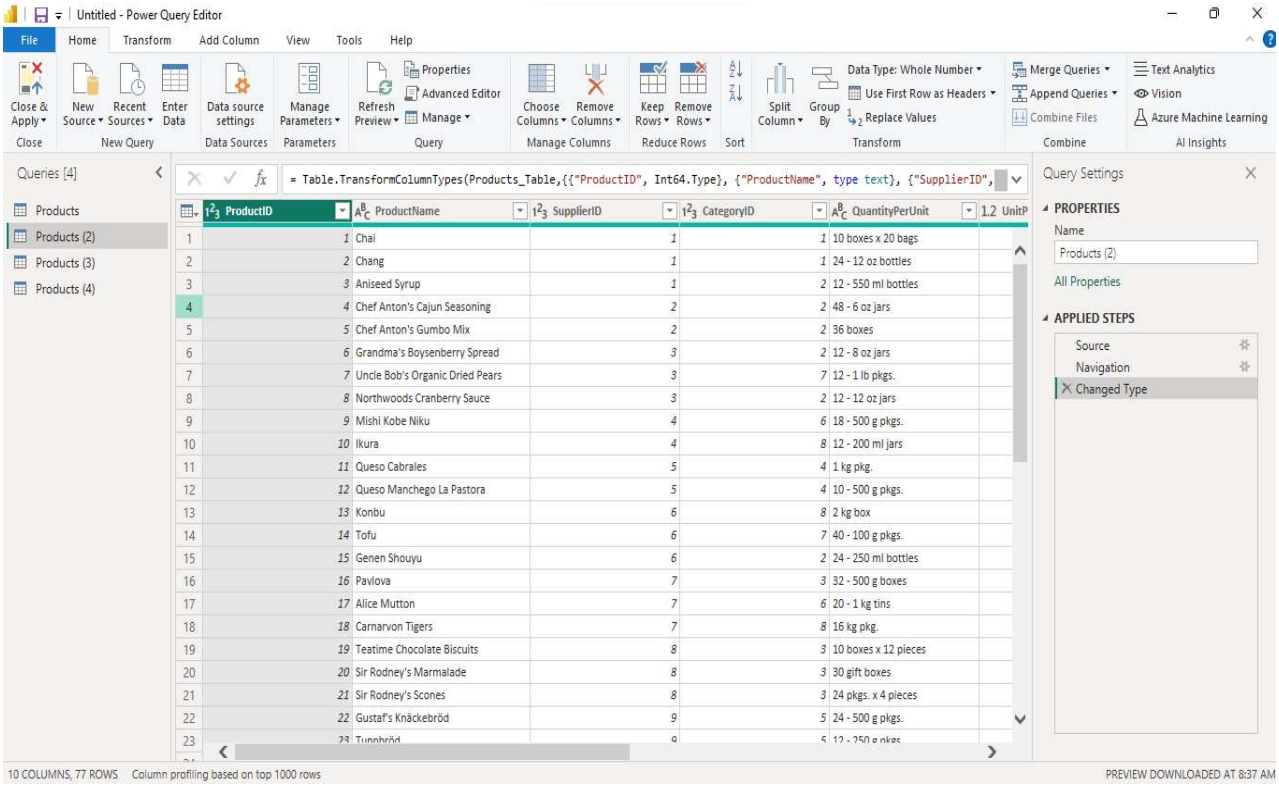
Power BI Desktop includes Query Editor, which is where you shape and transform your data connections. Query Editor opens automatically when you select Edit from Navigator. You can also open the Query Editor by selecting Edit Queries from the Home ribbon in Power BI Desktop.

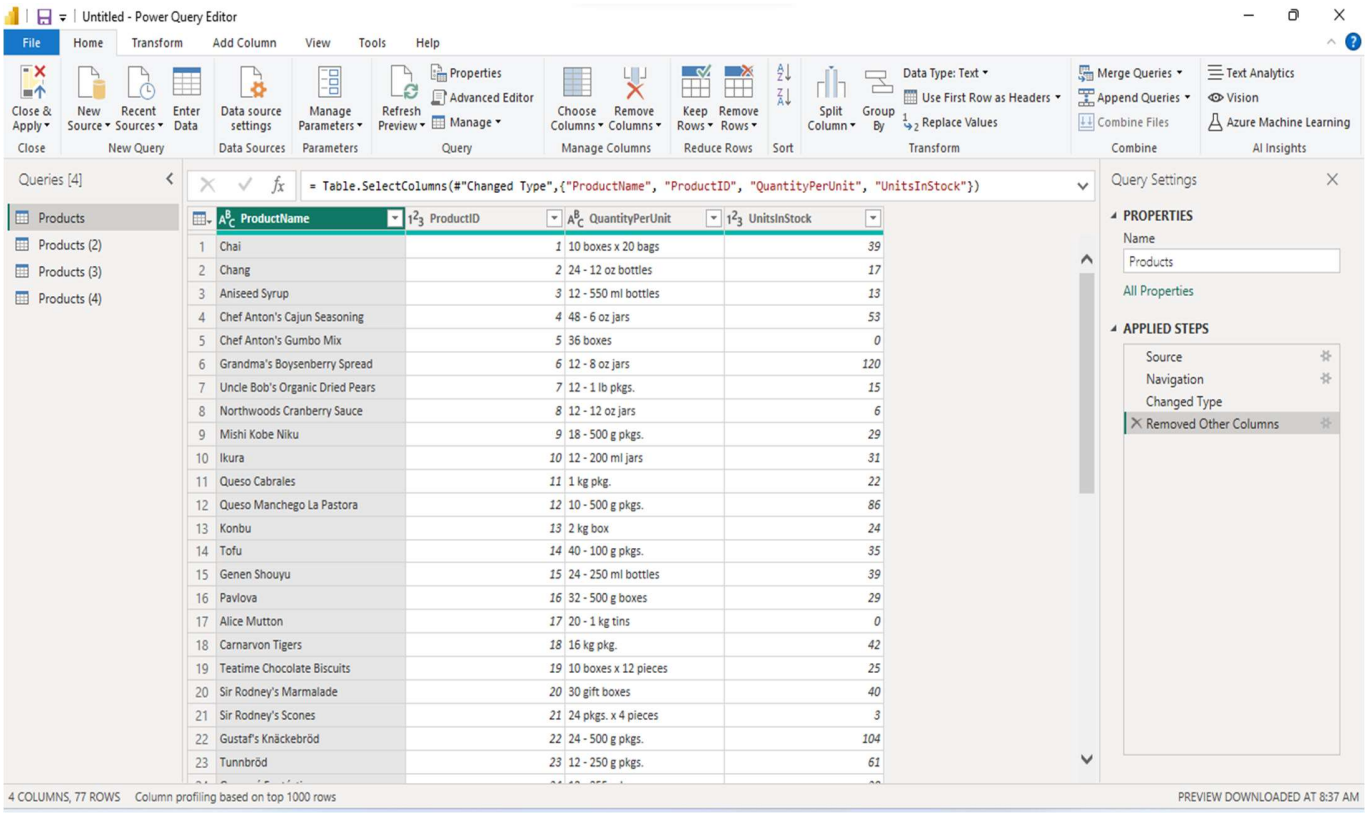
The following steps are performed in Query Editor.

1. In **Query Editor,** select the **ProductID, ProductName, QuantityPerUnit**, and **UnitsInStock** columns (use **Ctrl+Click** to select more than one column, or **Shift+Click** to select columns that are beside each other).
2. Select **Remove Columns > Remove** Other Columns from the ribbon, or right-click on a column header and click Remove Other Columns.

Date:- 30/01/2024 **Business Intelligence** **Semester :- VI**

Class:- T.Y.Bsc.I.T. **Roll No:- IT21060**

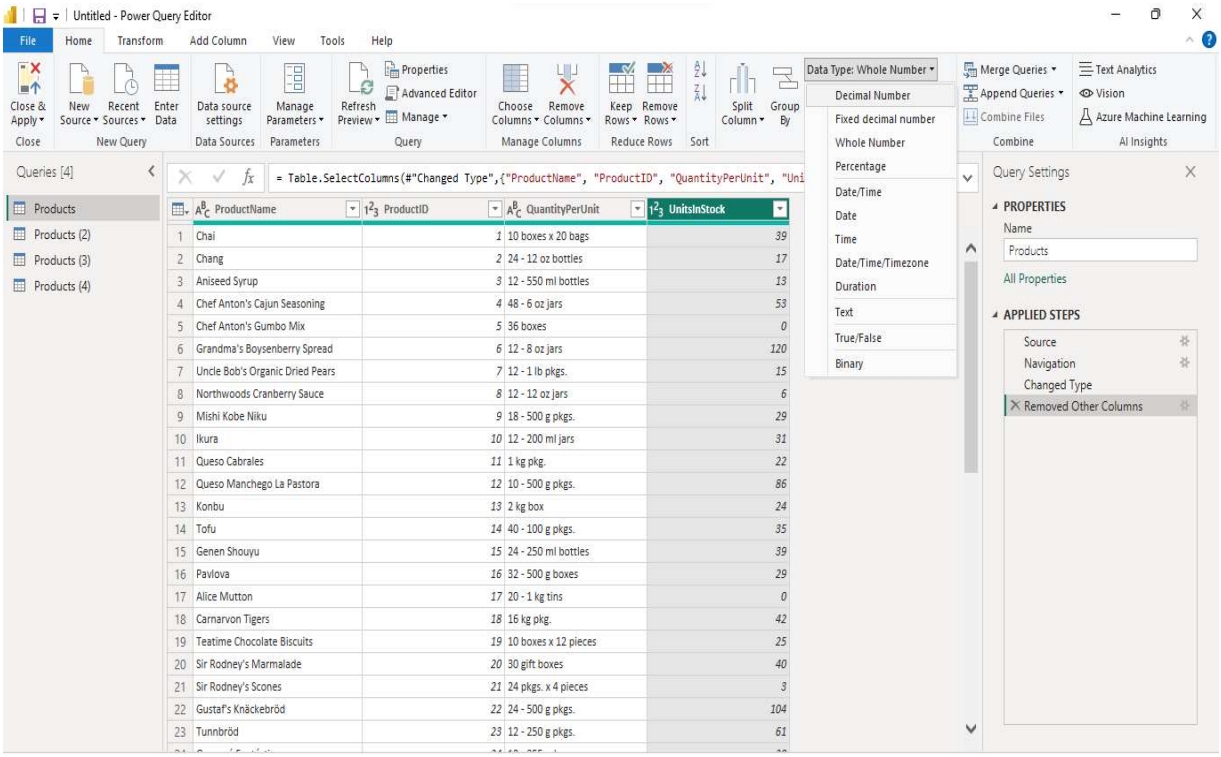




2. Change the data type of the UnitsInStock column

When Query Editor connects to data, it reviews each field and to determine the best data type. For the Excel workbook, products in stock will always be a whole number, so in this step you confirm the **UnitsInStock** column's datatype is Whole Number.

1. Select the **UnitsInStock** column.
2. Select the **Data Type drop-down button in the Home ribbon.**
3. If not already a Whole Number, select **Whole Number** for data type from the drop down (the Data Type: button also displays the data type for the current selection).



3. Expand the Order_Details table

The Orders table contains a reference to a Details table, which contains the individual products that were included in each Order. When you connect to data sources with multiples tables (such as a relational database) you can use these references to build up your query.

In this step, you expand the **Order_Details** table that is related to the Orders table, to combine the **ProductID, UnitPrice**, and **Quantity** columns from **Order_Details** into the **Orders table.** This is a representation of the data in these tables:

Date:- 30/01/2024 **Business Intelligence** **Semester :- VI**
Class:- T.Y.Bsc.I.T. **Roll No:- IT21060**

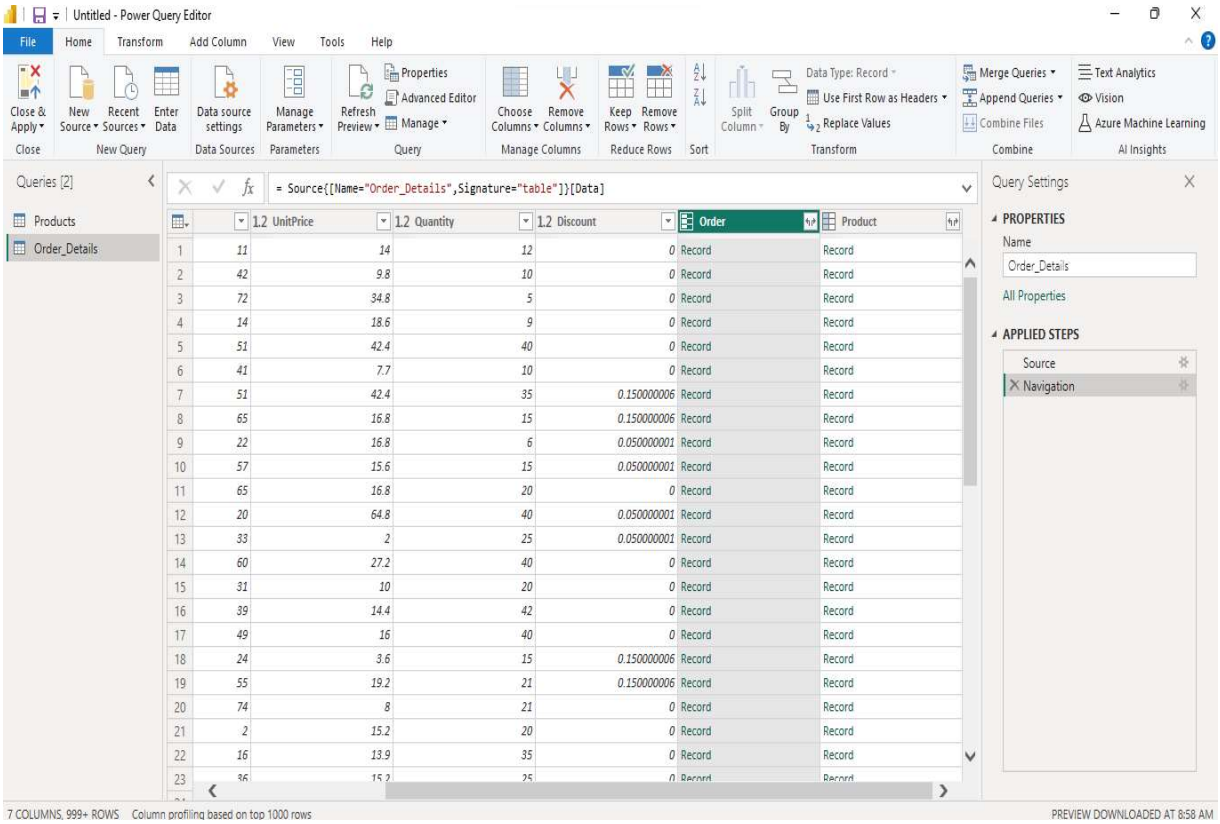
The Expand operation combines columns from a related table into a subject table. When the query runs, rows from the related table (Order_Details) are combined into rows from the subject table **(Orders).**

After you expand the Order_Details table, three new columns and additional rows are added to the Orders table, one for each row in the nested or related table.

1. In the Query View, scroll to the Order_Details column.

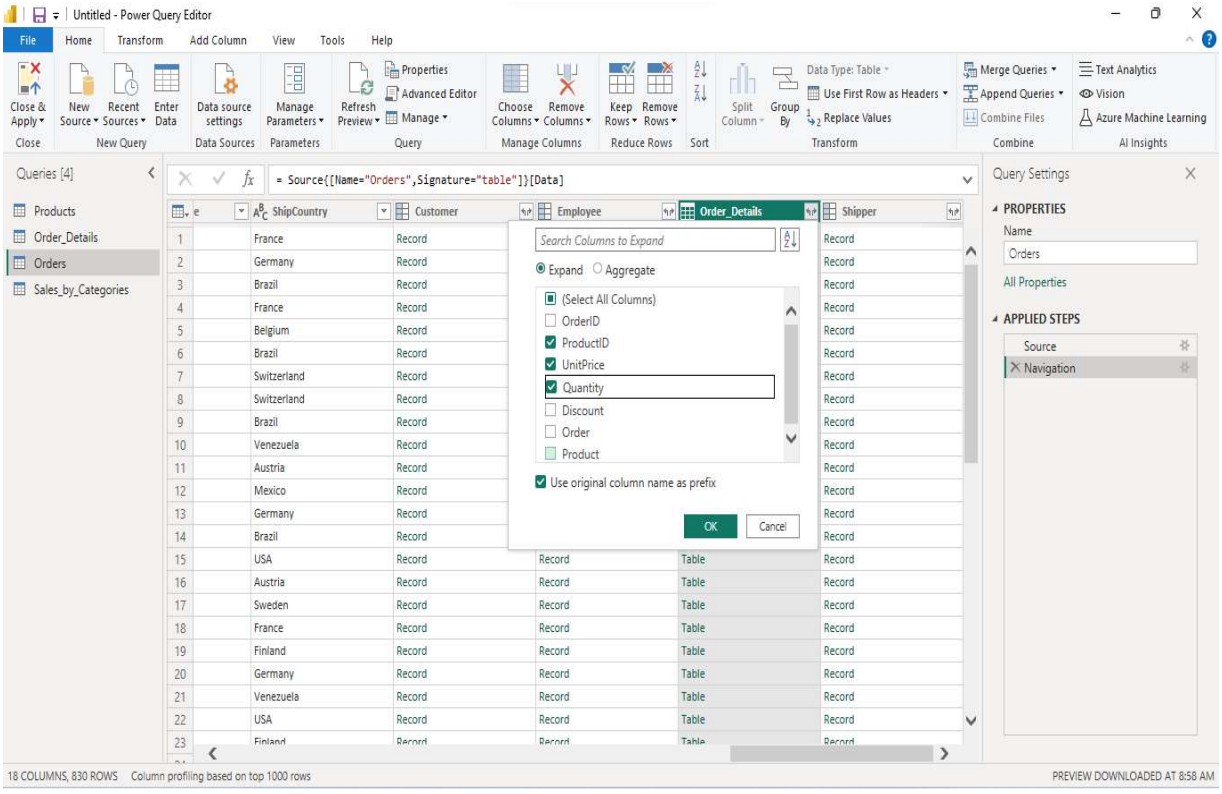
2. In the Order_Details column, select the expand icon ( ).

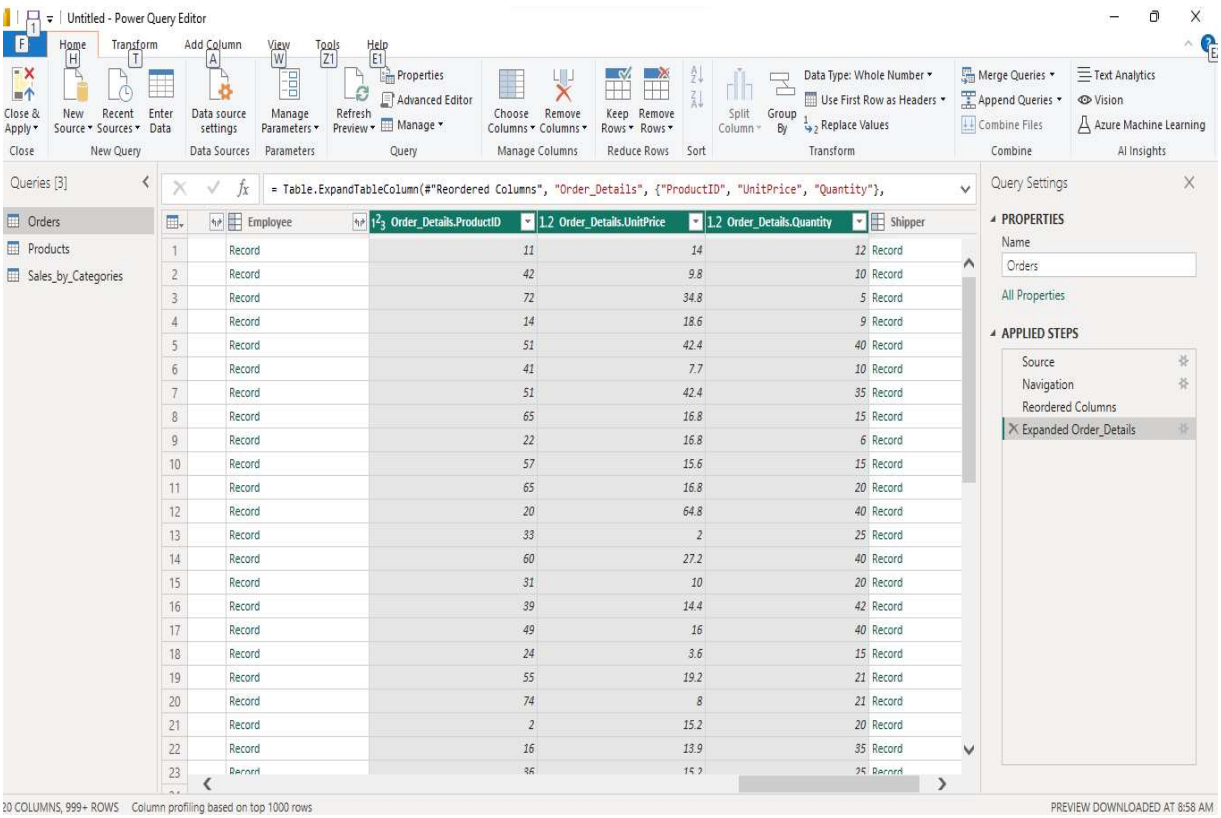
3. In the Expand drop-down:
    a. Select (Select All Columns) to clear all columns.
    b. Select ProductID, UnitPrice, and Quantity.
    c. click



Date:- 30/01/2024 **Business Intelligence** **Semester :- VI**

Class:- T.Y.Bsc.I.T. **Roll No:- IT21060**



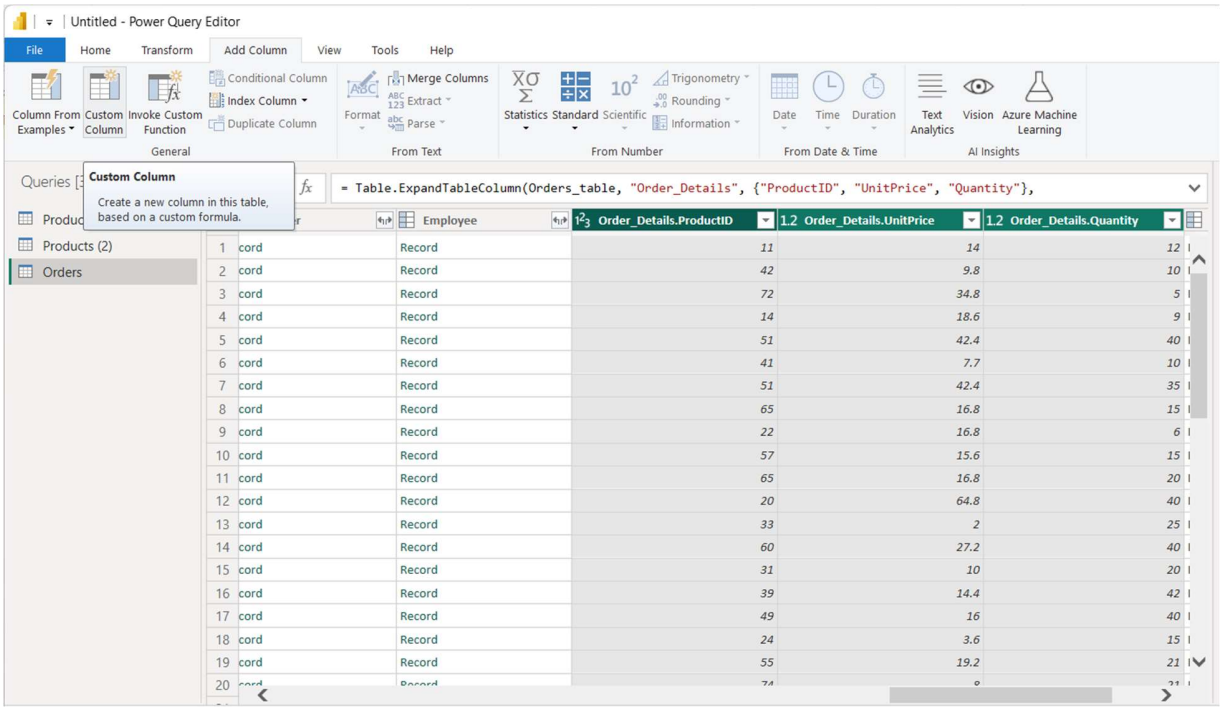


4. Calculate the line total for each Order_Details row

Power BI Desktop lets you to create calculations based on the columns you are importing, so you can enrich the data that you connect to. In this step, you create a Custom Column to calculate the line total for each Order_Details row.

Calculate the line total for each Order_Details row:

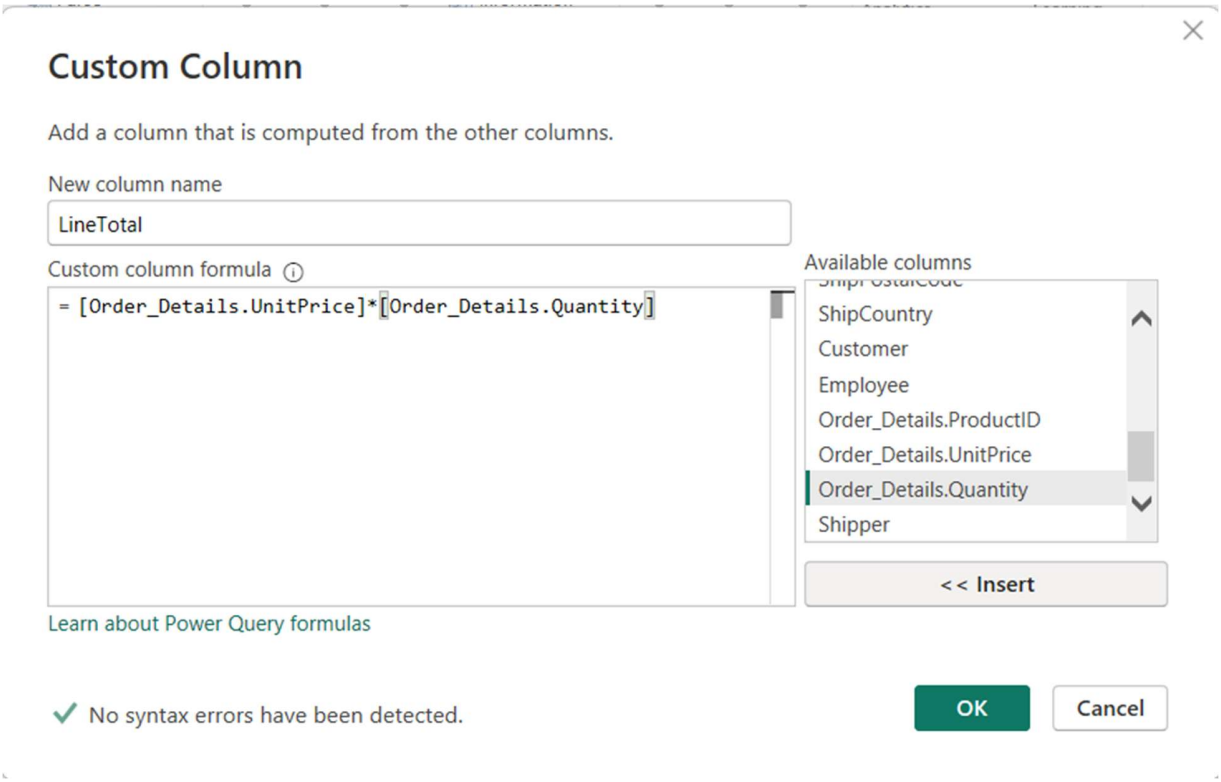
1. In the Add Column ribbon tab, click Add Custom Column.



2. In the Add Custom Column dialog box, in the Custom Column Formula textbox, enter **[Order_Details.UnitPrice] * [Order_Details.Quantity].**

3. In the New column name textbox, enter LineTotal.

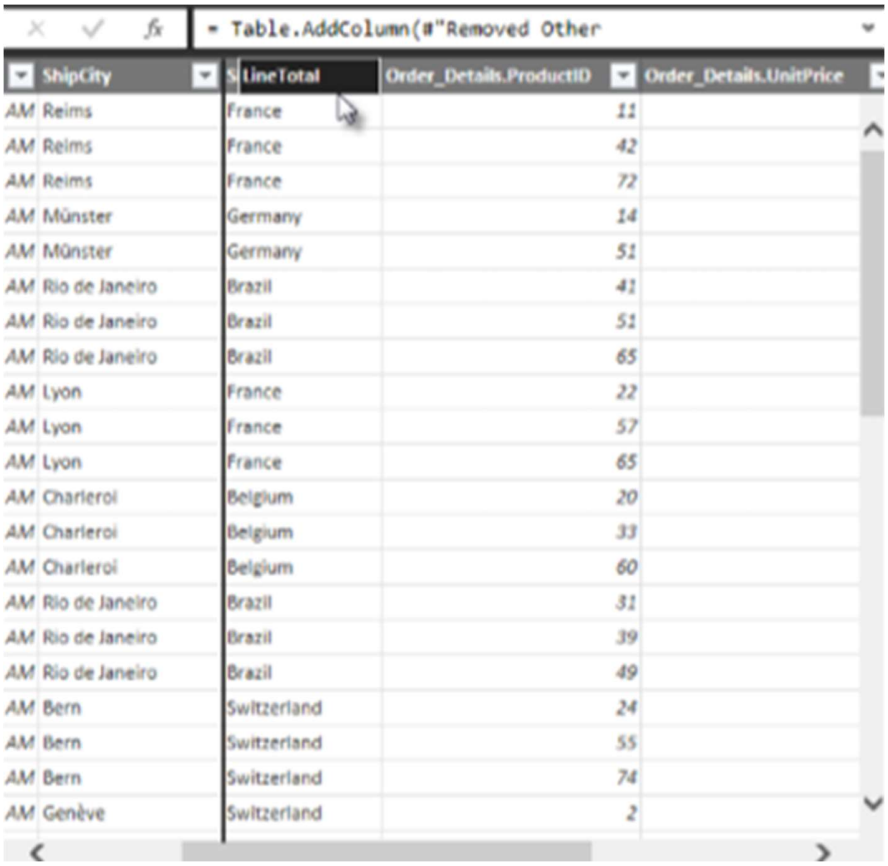
4. Click OK.



5. Rename and reorder columns in the query

In this step you finish making the model easy to work with when creating reports, by renaming the final columns and changing their order.

1. In Query Editor, drag the LineTotal column to the left, after Shipper



	ShipCity	LineTotal	Order_Details.ProductID	Order_Details.UnitPrice
AM	Reims	France	11	
AM	Reims	France	42	
AM	Reims	France	72	
AM	Münster	Germany	14	
AM	Münster	Germany	51	
AM	Rio de Janeiro	Brazil	41	
AM	Rio de Janeiro	Brazil	51	
AM	Rio de Janeiro	Brazil	65	
AM	Lyon	France	22	
AM	Lyon	France	57	
AM	Lyon	France	65	
AM	Charleroi	Belgium	20	
AM	Charleroi	Belgium	33	
AM	Charleroi	Belgium	60	
AM	Rio de Janeiro	Brazil	31	
AM	Rio de Janeiro	Brazil	39	
AM	Rio de Janeiro	Brazil	49	
AM	Bern	Switzerland	24	
AM	Bern	Switzerland	55	
AM	Bern	Switzerland	74	
AM	Genève	Switzerland	2	

2.
3. Remove the Order_Details. prefix from the Order_Details.ProductID, Order_Details.UnitPrice and Order_Details.Quantity columns, by double-clicking on each column header, and then deleting that text from the column name.

6. Combine the Products and Total Sales queries

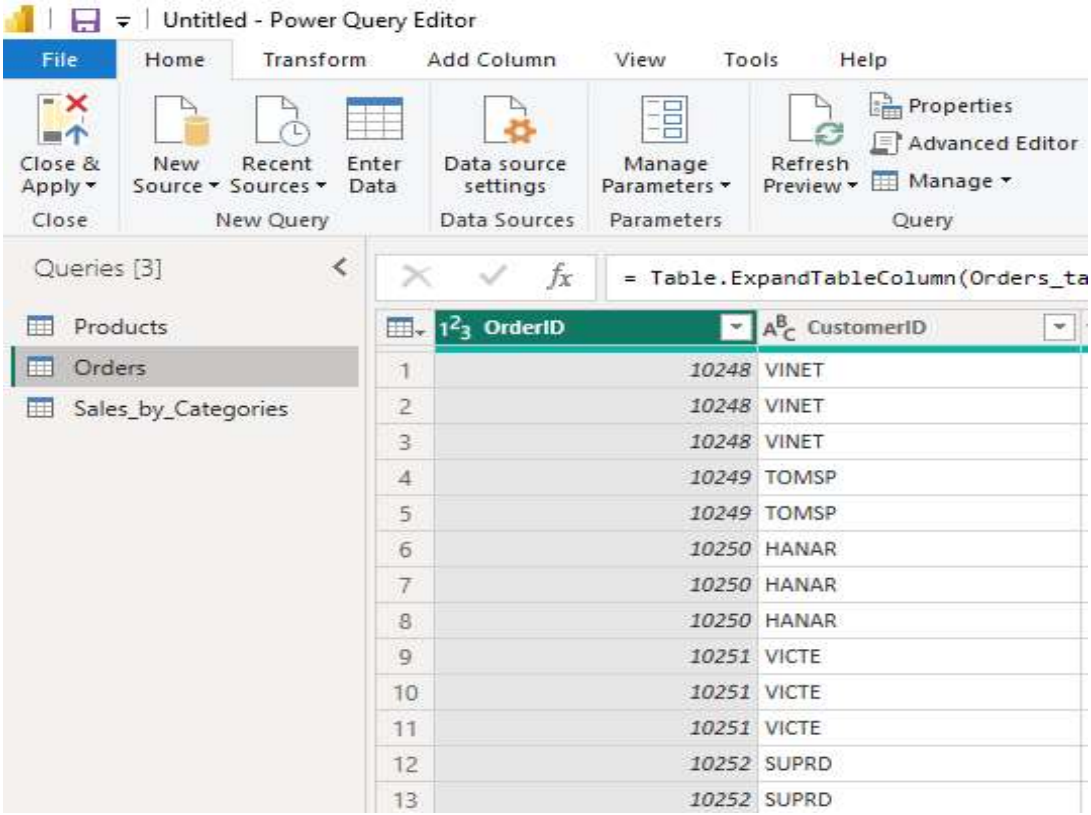
Power BI Desktop does not require you to combine queries to report on them. Instead, you can create Relationships between datasets. These relationships can be created on any column that is common to your datasets we have Orders and Products data that share a common 'ProductID' field, so we need to ensure there's a relationship between them in the model we're using with Power BI Desktop. Simply specify in Power BI Desktop that the columns from each table are related (i.e. columns that have the same values). Power BI Desktop works out the direction and cardinality of the relationship for you. In some cases, it will even detect the relationships automatically.

In this task, you confirm that a relationship is established in Power BI Desktop between the Products and Total Sales queries

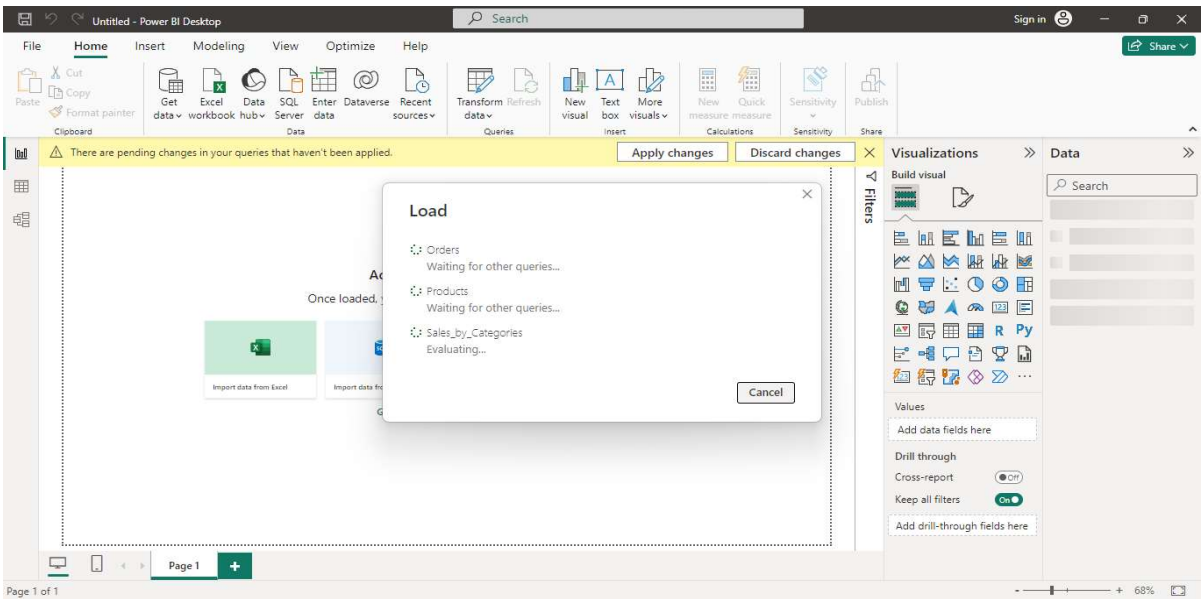
Date:- 30/01/2024 **Business Intelligence** **Semester :- VI**
Class:- T.Y.Bsc.I.T. **Roll No:- IT21060**

Step 1: Confirm the relationship between Products and Total Sales

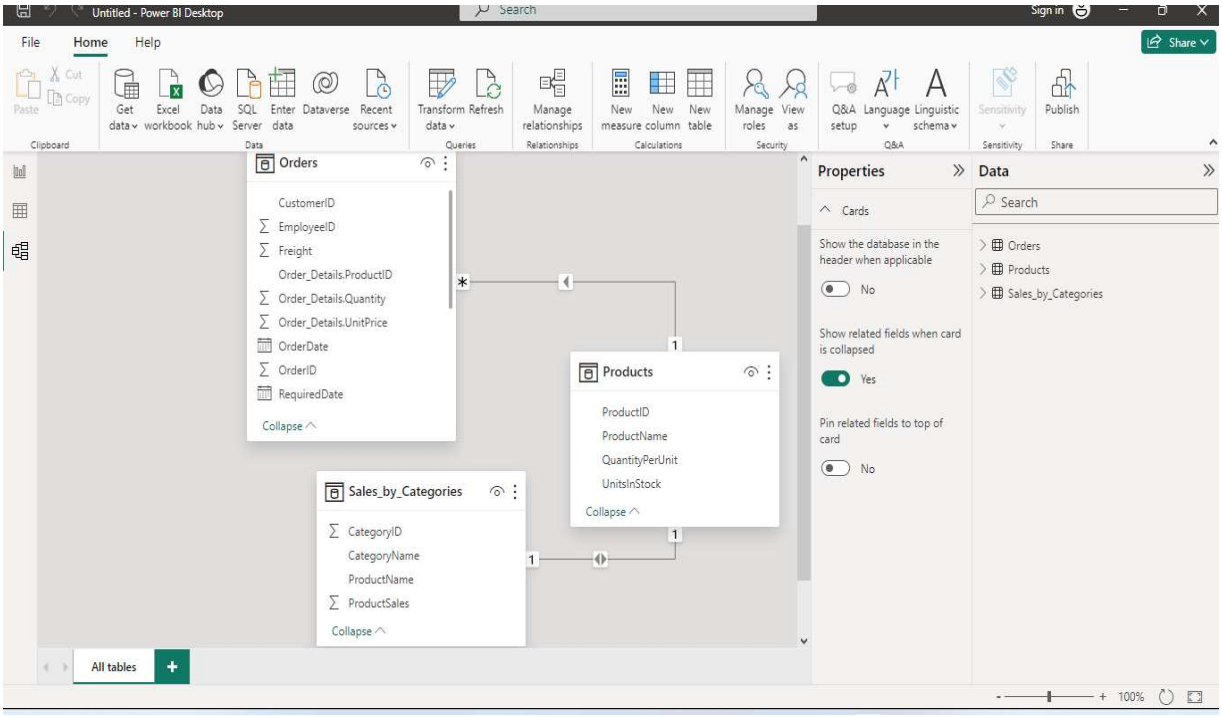
1. First, we need to load the model that we created in Query Editor into Power BI Desktop. From the Home ribbon of Query Editor, select Close & Load.



2. Power BI Desktop loads the data from the three queries.
3. Once the data is loaded, select the Manage Relationships button Home ribbon.
4. Select the New… button



5. When we attempt to create the relationship, we see that one already exists! As shown in the Create Relationship dialog (by the shaded columns), the ProductsID fields in each query already have an established relationship.
6. Select Cancel, and then select Relationship view in Power BI Desktop.
7. We see the following, which visualizes the relationship between the queries.



8. When you double-click the arrow on the line that connects the to queries, an Edit Relationship dialog appears.

9. No need to make any changes, so we'll just select Cancel to close the Edit Relationship dialog.