

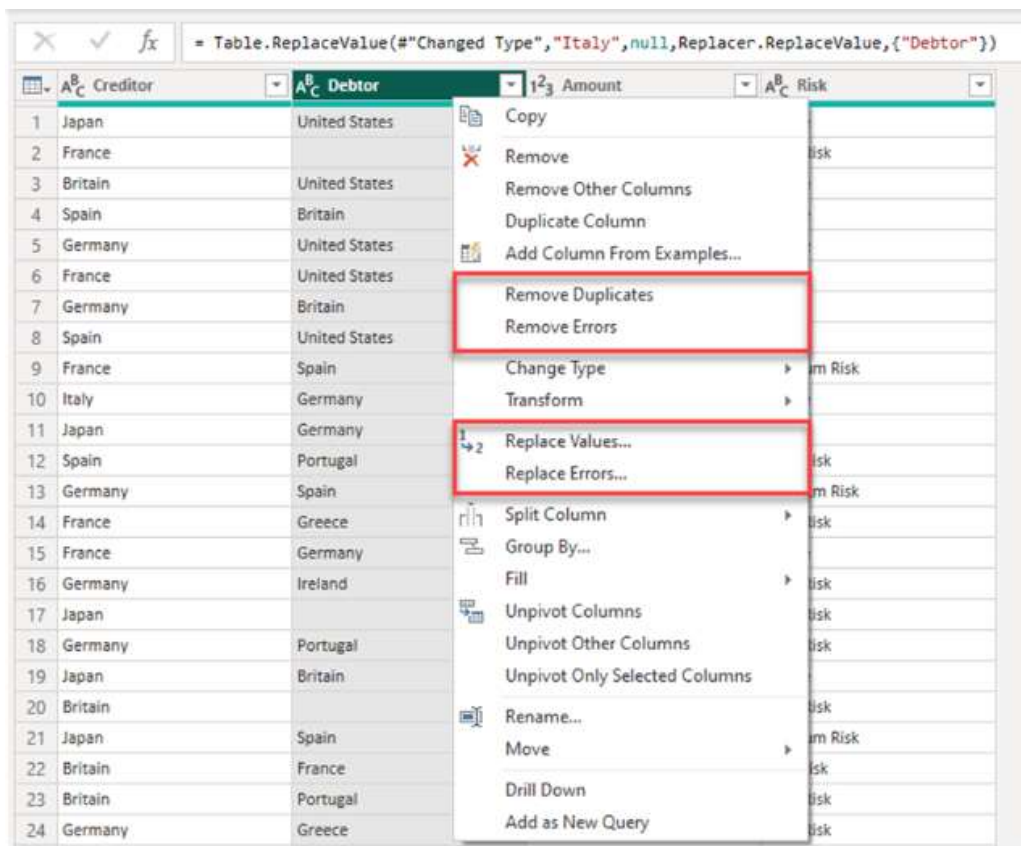
# **Data Warehouse**

## **Experiment No.: 7**

**Perform the Transformation and Loading  
process to construct the database in the  
Sqlserver / Power BI.**

# Experiment No. 7

1. **Aim:** Perform the Transformation and Loading process to construct the database in the Sqlserver / Power BI.
2. **Objectives:** To perform the data transformation process to generate meaningful reports.
3. **Course Outcomes:** Initiate the transformation process by connecting the data source and transform the data, which encompasses a wide range of options, including data cleaning, column renaming and data type conversion.
4. **Hardware / Software Required:** Power BI tool to extract and load data into the staging area.
5. **Theory: Data Cleaning:** Data cleaning stands as an essential step in working with data, as it often arrives with inconsistencies, missing values, and errors. Some fundamental functionalities you should master in Power BI for data cleaning include removing duplicates, filling in missing values, and correcting data types.

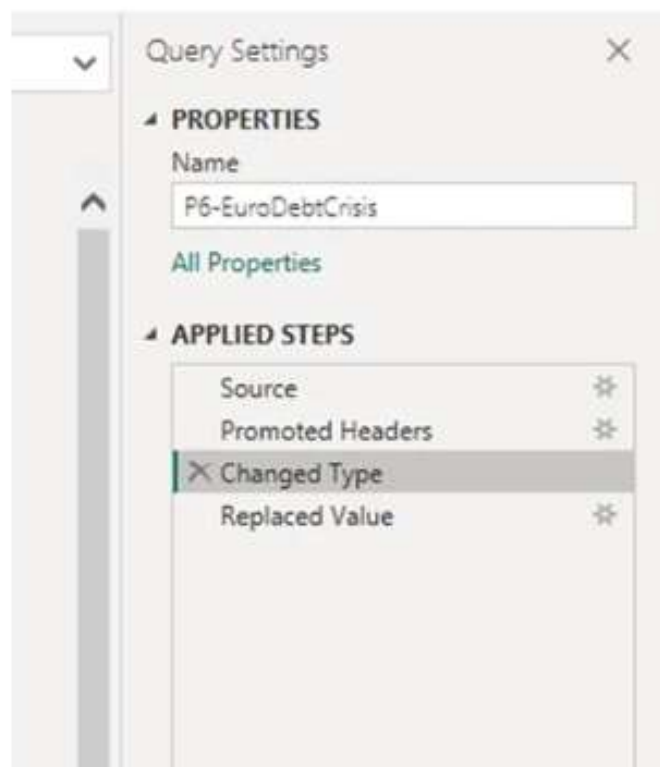


### Column Renaming:

Column renaming can be a cumbersome task, but it can significantly enhance the clarity of your reports. In Power Query Editor, rename columns to make them more comprehensible. Simply right-click on a column header and select "Rename..." (Refer to the image above for guidance).

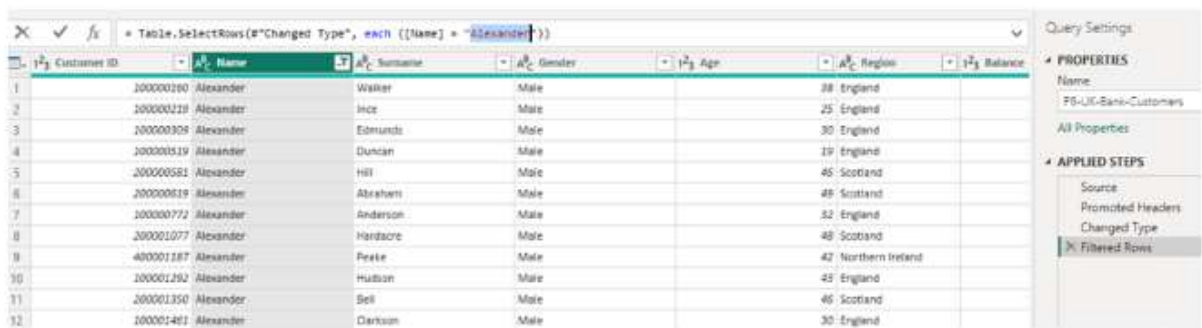
### Data Type Conversion:

Power BI generally does a commendable job of recognizing data types during the initial data source load. However, in cases where this information is incorrect, it is advisable to navigate to the "Change Type" option in your Change Pane, located on the rightmost side of the screen. Ensure you are at this step and update any incorrect data types. This is especially crucial if you have numeric fields that should be text-based, such as when you need to retain leading zeros. Use the "Change Type" feature in Power Query Editor to convert columns to the appropriate data types, like changing a text column to a date.



## Filtering Data:

Remove unnecessary rows or filter data based on specific criteria. Apply filters to select the relevant data for your analysis. This is particularly valuable when dealing with large datasets, where you may only be interested in specific data subsets. Additionally, when working with extensive datasets, you may not immediately see the desired filter, especially when filtering for specific values. In such cases, it can be advantageous to select a value first and then subsequently refine it to your specific criteria within the formula.

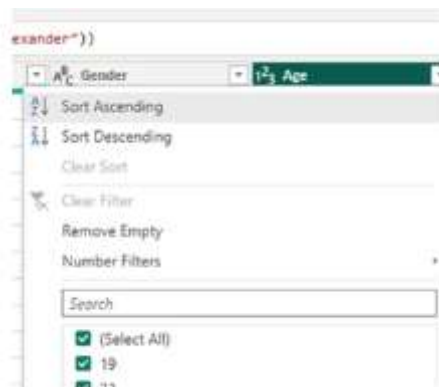


The screenshot shows a data table with columns: Customer ID, Name, Surname, Gender, Age, Region, and Balance. A filter is applied to the Name column, showing only rows where the name is 'Alexander'. The formula bar at the top shows: `Table.SelectRows(#"Changed Type", each ([Name] = "Alexander"))`. The right sidebar shows the 'Query Settings' pane with 'Name' set to 'PS-UC-Bank-Customers' and 'Applied Steps' including 'Source', 'Promoted Headers', 'Changed Type', and 'Filtered Rows'.

	Customer ID	Name	Surname	Gender	Age	Region	Balance
1	100000280	Alexander	Walker	Male		38 England	
2	100000219	Alexander	Ince	Male		25 England	
3	100000308	Alexander	Edmunds	Male		30 England	
4	100000519	Alexander	Duncan	Male		19 England	
5	100000581	Alexander	Hill	Male		46 Scotland	
6	100000619	Alexander	Abraham	Male		48 Scotland	
7	100000772	Alexander	Anderson	Male		52 England	
8	100001077	Alexander	Hardacre	Male		48 Scotland	
9	400001187	Alexander	Peake	Male		42 Northern Ireland	
10	100001262	Alexander	Hudson	Male		45 England	
11	100001350	Alexander	Sell	Male		46 Scotland	
12	100001461	Alexander	Clarkson	Male		30 England	

## Sorting Data:

Arrange data in ascending or descending order to enhance readability and analysis. You can sort columns by clicking the dropdown menu for the specific column and choosing "Sort Ascending" or "Sort Descending."



## Adding Custom Columns:

Create new columns based on existing data using Power Query's "Add Column" feature. This is especially useful for calculated columns or for merging data from multiple columns. This feature will quickly become one of your most frequently used tools for data transformation. It's essential to consider when and where to use this; sometimes, creating measures instead of columns may be more advantageous. Below is an example that demonstrates how to segment your data into age buckets.

The screenshot displays the Power Query Editor interface. At the top, a table with columns: Age, Region, Job Classification, Date Joined, Balance, and Age Bucket is visible. The 'Age Bucket' column contains values like 'Age 20-29', 'Age 30-39', etc. A 'Custom Column' dialog box is open in the foreground. It has a 'New column name' field containing 'Age Bucket'. The 'Custom column formula' field contains the following M code:

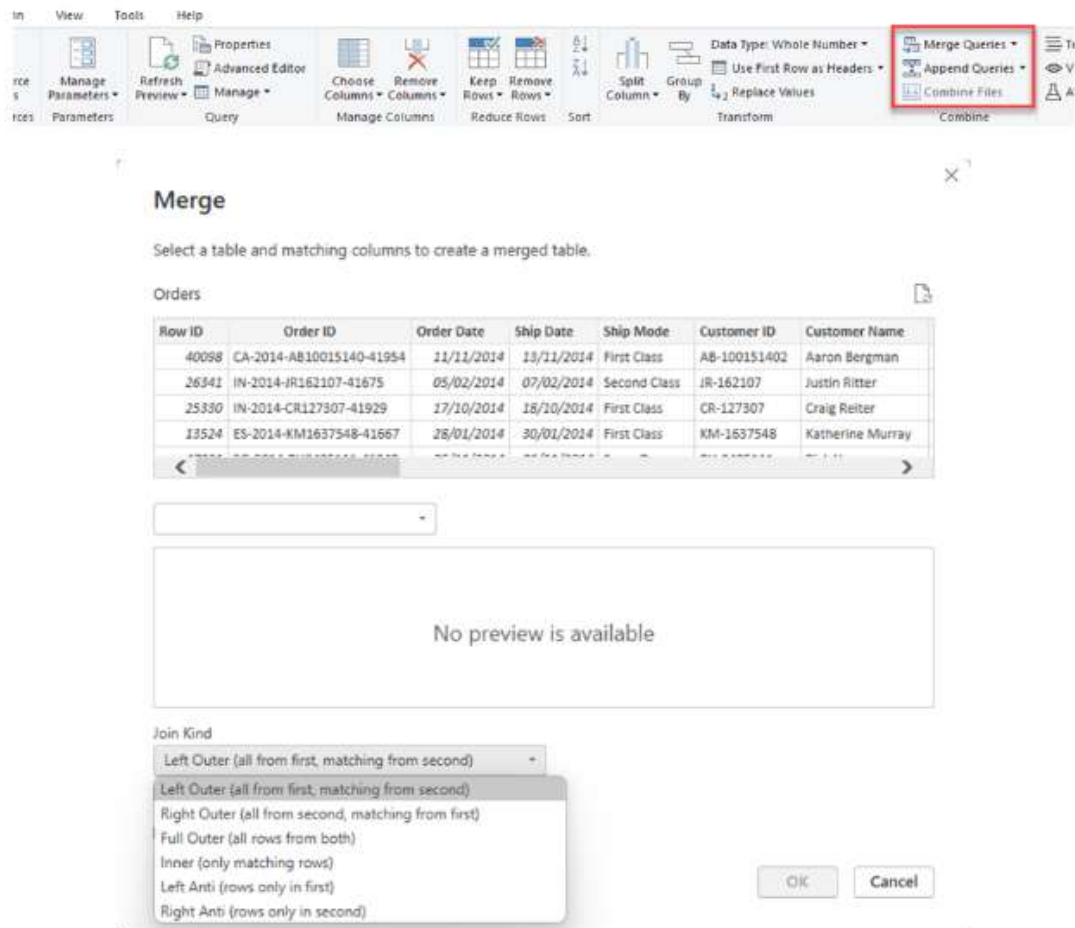
```
= "Age - " &  
    Number.ToText(Number.RoundDown([Age]/10)*10) &  
    "-" &  
    Number.ToText(Number.RoundDown([Age]/10)*10+9)
```

The 'Available columns' list on the right includes Customer ID, Name, Surname, Gender, Age, Region, and Job Classification. The '<< Insert' button is at the bottom of the list. Below the formula field, it says 'Learn about Power Query formulas' and 'No syntax errors have been detected.' The 'OK' and 'Cancel' buttons are at the bottom right of the dialog. Below the dialog, the 'Add Column' ribbon tab is selected, showing options like 'Conditional Column', 'Index Column', 'Duplicate Column', 'Merge Columns', 'Extract', and 'Parse'.

## Merging Queries:

Combine data from multiple sources or tables through query merging. Power BI offers various join types, such as inner join and left outer join, among others. It's crucial to

assess whether your goal can be achieved through your data model, as this can help reduce the data load on your report. However, it's not always possible to achieve the desired result solely through data modeling. Merging and appending data is a straightforward process. One thing to note is that sometimes you must create a new query for the merge to be executed correctly. Also, if you intend to append queries, ensure that the columns have the same naming conventions.



### Aggregating Data:

Group data by specific columns and perform aggregations like sum, average, or count using the "Group By" function. This is particularly useful for summarizing data. You can access the "Group By" feature either in the Ribbon or by right-clicking on a column. If you need

to group by multiple columns, either highlight all of them or add them to your grouping in the pop-up window.

## Pivot and Unpivot:

Reshape your data using the "Pivot" and "Unpivot" transformations. Pivot allows you to convert columns into rows, while Unpivot accomplishes the reverse. To pivot, select the column you want to use for column headers and press the pivot option in the Ribbon under the Transform Tab. In the pop-up window, select the column you want to use for values. All other columns will remain as they are. To unpivot, select all the columns you want to transform from pivot to row structure. This process will create two columns: "Attribute" with the column header and "Value."

The screenshot displays the Alteryx software interface. The top ribbon shows various transformation tools, with 'Pivot Column' highlighted. Below the ribbon, a data table is visible with columns: Gender, Region, Job Classification, Age 20-29, Age 30-39, Age 40-49, and Age 50-59. A 'Pivot Column' dialog box is open in the foreground. The dialog box has a title bar 'Pivot Column' and a close button. The main text reads: 'Use the names in column "Age Bucket" to create new columns.' Below this, there is a dropdown menu labeled 'Values Column: Sum' with 'Balance' selected. At the bottom of the dialog box, there are 'OK' and 'Cancel' buttons. The background table shows data for 22 rows, with columns for Gender, Region, Job Classification, and four age buckets.

	Gender	Region	Job Classification	Age 20-29	Age 30-39	Age 40-49	Age 50-59
1	Female	England	Blue Collar	212917084	325980910	109988973	10588758
2	Female	England	Other	127104421	320527777	162587901	9139162
3	Female	England	White Collar	570831799	2942630267	589481216	52009572
4	Female	Northern Ireland	Blue Collar	45528872	38872120	22948754	null
5	Female	Northern Ireland	Other	135161738	113178856	29391520	18135267
6	Female	Northern Ireland	White Collar	35286641	42581681	10378219	null
7	Female	Scotland	Blue Collar	8067391	null	24504787	23809480
8	Female	Scotland	Other	13987623	207588002	809135241	425880297
9	Female	Scotland	White Collar				53881267
10	Female	Wales					4091808
11	Female	Wales					3158882
12	Female	Wales					12311436
13	Male	England					9904830
14	Male	England					5013786
15	Male	England					57825532
16	Male	Northern Ireland					null
17	Male	Northern Ireland					null
18	Male	Northern Ireland					null
19	Male	Scotland					788078752
20	Male	Scotland					403700845
21	Male	Scotland					102289480
22	Male	Wales					null

## **6. Results**

Include the steps required to connect Power BI's capabilities to transform and visualize your data effectively. To create powerful and insightful reports and dashboards, you often need to clean and shape your data to make it suitable for analysis.

**7. Conclusions:** By performing the data transformation, we ensure that the data is clean and in uniform format to load and analyze the data.

**8. Viva Questions:** A list of potential questions related to the ETL operations can be expected.

## **9. References:**

1. Kimball Group: Kimball Group's Website offers articles and resources on dimensional modeling and data warehousing.
2. "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
3. Building the Data Warehouse" by William H. Inmon
4. <https://www.cittros.com/insights/basic-data-transformation-techniques-in-power-bi>