Lab Experiment 6: Querying Data from CSV-Query Editor, Connecting the data from the Excel Source, Clean, Transform the data.

Introduction

Power BI is a powerful data visualization and business intelligence tool developed by Microsoft. It enables users to connect to a wide variety of data sources, perform data transformations, and visualize data through interactive reports and dashboards. This experiment is intended to demonstrate the essential steps needed to import, clean, transform, and query data in Power BI, using **customer data** and **sales data** stored in CSV/Excel files.

Key Concepts in the Experiment

1. Power BI Desktop and Power Query Editor

- **Power BI Desktop**: The main tool in the Power BI suite that provides functionalities for data connection, transformation, visualization, and sharing reports.
- **Power Query Editor**: A data connection and transformation tool within Power BI. It is used to **clean, shape, and combine data** before loading it into Power BI for analysis.

2. Data Cleaning and Transformation

- **Data Cleaning**: Ensuring data is accurate, consistent, and free from errors. This includes tasks like removing null values, correcting data types, and handling duplicates.
- **Data Transformation**: Modifying data to make it more suitable for analysis. Common transformations include filtering, merging, splitting columns, changing data formats, and summarizing data.

3. Data Sources: CSV and Excel

- **CSV File:** A simple, structured format for data storage, commonly used for exporting data. In this lab, the Customer Data file will be in CSV format.
- **Excel File:** A widely-used format for data organization, often storing multiple sheets or tables. Here, the Sales Data will be in Excel format.

Objectives

1. **Import and Query** data from a CSV file using Power Query Editor.
2. **Connect** to data from an Excel source and load it into Power Query Editor.
3. **Clean and Transform** the data.
4. **Analyze** the data through visualizations in Power BI.

Querying Data from CSV-Query Editor

Importing Data from a CSV File (Customer Data)

1. In Power BI, go to **Home > Get Data > Text/CSV**.
2. Locate the **Customer Data CSV file** and click **Open**.
3. A preview of the data will appear in the **CSV import dialog**.
4. Click Load Data or **Transform Data** to open the data in **Power Query Editor** for querying and transformation.

Querying and Transforming CSV Data in Power Query Editor

In Power Query Editor, perform the following steps to clean and transform the Customer Data:

1. **Rename Columns**:

   - Rename each column to be more descriptive if needed. Ensure columns are labeled correctly, like CustomerID, CustomerName, Place, and Email.

Option 1: Right-click on the column header and select **Rename**. Then, type the new name and press **Enter**.

Option 2: Double-click on the column header to edit it, type the new name, and press **Enter**.

2. **Check for Null Values**:

   - Use the **Remove Rows** option to remove rows with null values if necessary.

3. **Set Data Types**:
   - Ensure each column has the correct data type:
   - CustomerID: Whole Number
   - CustomerName: Text
   - Place: Text
   - Email: Text
4. **Apply Filters**:
   - If required, filter data by specific criteria, such as customers from **Mysore**.
5. **Split Columns**

- If you have a combined field like a full name or an address, you can split it into multiple columns. For instance, if the CustomerName column has first and last names together (e.g., "Anil Kumar"), you can split it into FirstName and LastName.
- **How**: Right-click the CustomerName column > **Split Column** > choose **By Delimiter** and select a space as the delimiter.

6. Add Custom Columns

- Create a new column with custom logic. For example, add a column to classify customers based on their location.
- **How**: Go to **Add Column** > **Custom Column**. You can add logic like if [Place] = "Mysore" then "Local" else "Non-Local" to create a column that labels customers based on location.

7. Extract Domain from Email

- Extract the domain part of the email (e.g., gmail.com from anil@gmail.com). This is useful if you want to analyze customers based on their email domains.
- **How**: Select the Email column > **Add Column** > **Extract** > **Text After Delimiter**, and set "@" as the delimiter.

8. Remove Duplicate Records

- Ensure there are no duplicate entries in your dataset. For example, if two rows have the same CustomerID, you might want to keep only one.
- **How**: Select the CustomerID column > **Home** > **Remove Duplicates**.

9. Group By

- Group data based on a specific column, such as Place, to summarize data. For example, count the number of customers in each city.
- **How**: Select the Place column > **Home** > **Group By**. Set it to group by Place and add an aggregation like **Count of Rows**.

10. Create Calculated Columns

- You can add calculated columns based on the existing ones. For example, you could create a column to count the characters in CustomerName for analysis purposes.
- **How**: Go to **Add Column** > **Custom Column** and add a formula like Text.Length([CustomerName]).

11. Format Column Values

- Standardize text or values by changing the case of text columns (like making all place names lowercase or uppercase).
- **How**: Select the Place column > **Transform** > **Format** > **Uppercase** or **Lowercase**.

12. Fill Missing Data

- If there are null or blank values in your dataset, you can fill these with a default value. For example, fill missing entries in the Place column with "Unknown".
- **How**: Select the column with missing values > **Transform** > **Replace Values** or **Fill Down** (if you want to carry forward the previous value).

13. Apply Data Type Transformations

- Enforce the correct data type for each column to improve data accuracy. For example, make sure CustomerID is set as a whole number and Email as text.
- **How**: Click the icon next to the column name and choose the appropriate data type.

14. Close & Apply:

- Once completed, click **Close & Apply** in the Home tab to load the transformed Customer Data into Power BI.

Connecting the data from the Excel Source, Clean, Transform the data.

Step 1:Import Data from Excel File

1. **Get Data:** Again, click on "Get Data" > "Excel."
2. **Select File:** Browse to `sales_data.xlsx` and select it.
3. **Navigator Window:** In the Navigator window, select the sheet containing your data (e.g., "Sheet1").
4. **Load Data:** Click "Load" to bring the Excel data into Power BI.

Step 2: Open Query Editor

1. **Home Tab:** Click on "Transform Data" to open the Query Editor.

Step 3: Clean and Transform Excel Data

1. **Remove Unnecessary Columns**

   - Select any column you do not need (e.g., `Price`) and click on "Remove Columns."

2. **Change Data Types**

   - Click on the column header (e.g., `OrderDate`) and select "Data Type" to change it to `Date`.

3. **Create New Column**

   - Click on "Add Column" > "Custom Column."

   Enter a name (e.g., `TotalSales`) and use the formula:

   ```
   [Quantity] * [Price]
   ```

   Click "OK."

4. **Merge Queries**
   - Go to "Home" > "Merge Queries."
   - Select `sales_data` as the first table and `customer_data` as the second table.
   - Choose the matching column (e.g., `CustomerID` from `customer_data` and create a relation based on order data).
   - Click "OK."

5. Filter Orders by Date Range

- **Purpose**: Show only orders from the range "01-01-2023" to "15-01-2023.
- **Steps**:
    1. In the Power Query Editor, click the filter icon on the OrderDate column.
    2. Select "Date Filters" > "Between."
    3. Set the date range to "01-01-2023" to "15-01-2023."
    4. Click "OK" to apply the filter.

Filtering by date allows you to focus on specific time periods, which is useful for generating yearly reports.

6. Group Data by Product

Purpose: Summarize data to see the total quantity for each product.

Steps:

- In the Power Query Editor, select the Product column.
- Go to "Transform" > "Group By."
- In the "Group By" dialog, set:
- "Group By" to Product.
- Click "OK."

Grouping by product gives insights into which products have the highest sales in terms of quantity.

7. Add Year Column from OrderDate

Purpose: Extract the year from each OrderDate to allow for easier year-based analysis.

Steps:

- In the Power Query Editor, select the OrderDate column.
- Go to "Add Column" > "Date" > "Year."
- This allows you to group or filter data by year, making it easier to analyze trends over time.

8. Summarize the Quantity Sold for Each Product

Select the Product Column:

- In Power Query Editor, click on the `Product` column to select it.

- **Apply Group By Transformation**:
    - Go to the **Transform** tab and select **Group By**.
- **Set Group By Options**:
    - In the Group By dialog box:
        - **Group By**: Set this to `Product`.
        - **New Column Name**: Enter `TotalQuantity`.
        - **Operation**: Choose **Sum**.
        - **Column**: Select `Quantity`.
    - Click "OK" to apply the transformation.

You should now see a table with each product and the total quantity sold for each (e.g., Product A, Product B, Product C).

9. Create Conditional Column: High/Low Quantity

- **Purpose**: Categorize orders as "High Quantity" or "Low Quantity" based on Quantity.
- **Steps**:
    1. Go to "Add Column" > "Conditional Column."
    2. Name the column "QuantityCategory."
    3. Set conditions:
        - If Quantity is greater than or equal to 8, then "High Quantity."
        - Otherwise, "Low Quantity."
    4. Click "OK."

This categorization provides an easy way to identify bulk orders vs. smaller ones.