

## Lesson 3

**Topic:** Data Transformation with Power Query (Part 1)

**Prerequisites:** Download Customer\_Orders.txt, Orders.txt

1. What is the purpose of the "Applied Steps" pane in Power Query?
  - The "Applied Steps" pane in Power Query serves to record, display, and allow for editing of the sequence of data transformations applied to a query. It provides a transparent, step-by-step log of all actions taken to shape and clean data, enabling users to review, modify, delete, or even rearrange these steps to understand and control the data processing flow.
2. How do you remove duplicate rows in Power Query?
  - To remove duplicate rows in Power BI, open the Power Query Editor by selecting Transform data on the **Home** tab. In the editor, select the desired column(s), then go to the Home tab and click **Remove Rows > Remove Duplicates** to delete rows with identical values in the selected column(s).
3. What does the "Filter" icon do in Power Query?
  - It filters rows of data based on the values within that specific column. Clicking filter icon (usually represented as a small arrow or inverse triangle in a column header) opens a menu with various filtering options, such as selecting specific values from a list, using text or numeric filters, or clearing existing filters, enabling you to refine your dataset to show only the relevant records.
4. How would you rename a column from "CustID" to "CustomerID"?
  - There are several ways to rename column name:
    - Double-Clicking the Column Header:
      - Open the Power Query Editor.
      - Locate the column "CustID" and double-click on the header.
      - Type the new name ("CustomerID") for the column and press Enter.
    - Using the Right-Click Context Menu:
      - Open the Power Query Editor.
      - Right-click on the header of the column "CustID" and select "Rename" from the context menu.
      - Type the new name, "CustomerID" and press Enter.
5. What happens if you click "Close & Apply" in Power Query?
  - It applies all changes and closes the editor.

6. Remove all rows where Quantity is less than 2.

Apply a Filter

- Click the drop-down arrow on the column "Quantity".
- Select "Number filter"->"Less Than.." and the value is 2
- This removes those rows from the dataset.

7. Split the OrderDate column into separate "Year," "Month," and "Day" columns.

- Select the "OrderDate" column, and then the Add Column tab's Date dropdown menu choose Year, Month, and Day to create the new columns.

| 1 <sup>2</sup> <sub>3</sub> Year | 1 <sup>2</sup> <sub>3</sub> Month | 1 <sup>2</sup> <sub>3</sub> Day |
|----------------------------------|-----------------------------------|---------------------------------|
| 2023                             | 1                                 | 10                              |
| 2023                             | 1                                 | 15                              |
| 2023                             | 1                                 | 20                              |
| 2023                             | 1                                 | 25                              |

8. Replace all "Mouse" entries in the Product column with "Computer Mouse."

- **Transform** tab->**Replace Values** on dialogue box enter the value "Mouse" to "Value To Find" and "Computer Mouse" to "Replace With" field.

| A <sup>B</sup> <sub>C</sub> Product | 1 |
|-------------------------------------|---|
| 3 Laptop                            |   |
| 3 Computer Mouse                    |   |
| 3 Keyboard                          |   |
| 3 Monitor                           |   |

9. Sort the table by OrderDate (newest first).

- On "Home" tab and click "Transform Data" to open the Power Query Editor and select column "OrderDate".
- Click on the drop-down arrow next to the date column header and choose "Sort Descending."

10. How would you handle null values in the Price column?

- Select the Price column.
- Go to the Replace Values button on the Transform tab in the ribbon.
- In the "Value To Find" field, type null.

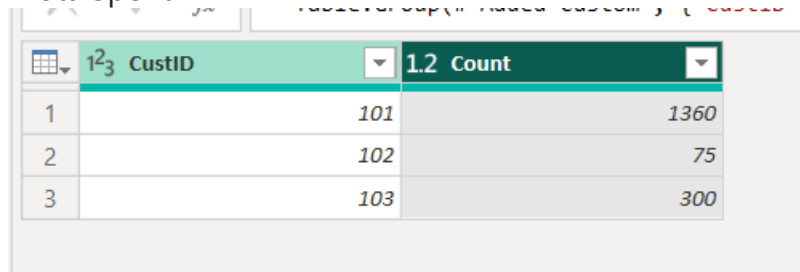
- In the "Replace With" field, enter 0 or another logical default value, like the column's average.
- Click OK.

11. Write custom M-code to add a column calculating TotalSpent = Quantity \* Price.

- = Table.AddColumn("#Removed Columns", "TotalSpent", each [Quantity]\*[Price])

12. Group the table by CustID to show total spending per customer.

- Go to **Transform** tab, select CustID column and click on **Group by**.
- Next on Operation field select SUM and on column field select "TotalSpent"



The screenshot shows a Power Query table with two columns: 'CustID' and 'Count'. The 'CustID' column has three rows with values 101, 102, and 103. The 'Count' column has corresponding values 1360, 75, and 300. The table is grouped by CustID.

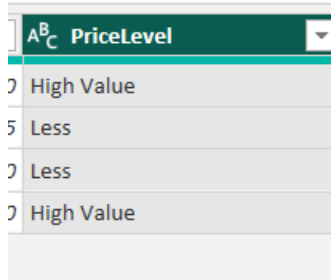
| CustID | Count |
|--------|-------|
| 101    | 1360  |
| 102    | 75    |
| 103    | 300   |

13. Fix inconsistent date formats (e.g., 01/10/2023 vs. 2023-01-10) in OrderDate.

- Right-click on the header of the column and select Change Type, then choose Using Locale.

14. Create a conditional column: Label orders as "High Value" if Price > 100.

- On **Add Column** tab click on **Conditional Column**.
- On New Column field enter new column name "Pricelevel", inside if **Column Name** select "Price" column, on Operator select "is greater than", Value field will be "100" and Output field will be "High Value".
- On else part we can enter another level as "Less", but it is not in task.



The screenshot shows a Power Query table with a new column named 'PriceLevel'. The column has four rows with values 'High Value', 'Less', 'Less', and 'High Value'.

| PriceLevel |
|------------|
| High Value |
| Less       |
| Less       |
| High Value |

15. Optimize the query to reduce refresh time (e.g., remove unused columns early).

- Reduce Dataset Size: Remove unnecessary columns and rows, filter out irrelevant historical data, and consider using aggregations or summarized tables for large datasets.

- Optimize DAX: Simplify complex DAX calculations or precompute them if feasible. Use measures instead of calculated columns where possible, as measures are calculated on demand, while calculated columns store results directly in the model.
- Star Schema: Structure your data model using a star schema (fact tables and dimension tables) for efficient querying.
- Cardinality: Avoid excessive relationships or high-cardinality columns, which can impact performance.
- Incremental Refresh: For large datasets, configure incremental refresh to only load new or updated data, rather than refreshing the entire dataset each time.

Orders

| CustID | Name    | OrderDate  | Product  | Quantity | Price |
|--------|---------|------------|----------|----------|-------|
| 101    | Alice   | 2023-01-10 | Laptop   | 1        | 1200  |
| 102    | Bob     | 2023-01-15 | Mouse    | 3        | 25    |
| 101    | Alice   | 2023-01-20 | Keyboard | 2        | 80    |
| 103    | Charlie | 2023-01-25 | Monitor  | 1        | 300   |

Customer\_orders

| CustID | Name    | Email  |
|--------|---------|--|
| 101    | Alice   | <a href="mailto:alice@example.com">alice@example.com</a>     |
| 102    | Bob     | <a href="mailto:bob@example.com">bob@example.com</a>         |
| 103    | Charlie | <a href="mailto:charlie@example.com">charlie@example.com</a> |