Lesson 4

Topic: Data Transformation with Power Query (Part 2)
Prerequisites: Download Customers.xlsx, Orders.csv

1. **What is the difference between "Merge" and "Append" in Power Query?**

   In Power Query, "Merge" and "Append" are two different operations used to combine data, but they work in very different ways.

   Merge:

- The "Merge" operation is used to combine columns from two or more tables.
- It is similar to a JOIN in SQL.
- You need to have at least one common column (like CustomerID) in both tables.
- Merge is typically used when you want to bring additional related information into a table.
  For example, if you have an Orders table and a Customers table, you can merge them using the CustomerID so that each order also includes the customer's name and details.

  Append:

- The "Append" operation is used to combine rows from two or more tables.
- It is similar to a UNION in SQL.
- The tables you append should have the same column structure (i.e., same column names and data types).
- Append is commonly used when you have multiple tables with the same type of data and you want to stack them together.
  For example, if you have separate order files for January and February, you can append them to create a single table containing all orders.

2. **How do you split a "Full Name" column into "First Name" and "Last Name"?**

   To split a "Full Name" column into separate "First Name" and "Last Name" columns in Power Query, follow these steps:

   1. Load your data into Power Query by selecting the table that contains the "Full Name" column.
   2. Select the "Full Name" column by clicking on its header.
   3. Go to the Home tab or the Transform tab on the ribbon.

4. Click on "Split Column", then choose "By Delimiter".
5. In the delimiter options, select Space as the delimiter (assuming the full name is in the format "First Last").
6. Choose to split at the left-most delimiter, which will separate the first word (First Name) from the rest (Last Name).
7. Power Query will create two new columns: typically named "Full Name.1" and "Full Name.2", which you can rename to "First Name" and "Last Name".

3. What is "Pivot Columns" used for?

Pivot Columns is used to transform rows into columns. It helps reshape your data to make it more analysis-friendly, especially when you want to:

- Turn values from one column into column headers.
- Aggregate the data (like sum, count, average) based on those new headers.

How to Use in Power BI:

1. Go to Power Query Editor.
2. Select the column to pivot (e.g., Year).
3. Click Transform → Pivot Column.
4. In the popup, choose the column to use for values (e.g., Sales).
5. Choose an aggregation function (e.g., Sum).

4. How do you undo a step in Power Query?

In **Power BI's Power Query Editor**, every change you make is recorded as a **step** in the **"Applied Steps"** pane (on the right side). To undo a step, you can simply **delete or modify** it.

Ways to Undo a Step:

1. Using the "Applied Steps" pane:

- Look at the right-side panel in Power Query under "Applied Steps".
- Hover over the step you want to remove.
- Click the X icon to delete that step.

This is the easiest and most common way to undo a transformation.

2. Right-click and delete:

- Right-click on any step in the "Applied Steps" pane.
- Select "Delete" or "Delete Until End".

"Delete Until End" will remove the selected step and all steps after it.

3. Undo Keyboard Shortcut (limited use):

- Press Ctrl + Z immediately after a step, but this only works before any other actions are taken.
- It's safer to use the Applied Steps pane.

5. **What is the purpose of "Reference" vs. "Duplicate" in queries?**

Duplicate:

You have a query called Sales. If you duplicate it:

- A new query Sales (2) is created.
- It contains all the same steps, but is completely separate.
- Changes made to Sales later will not affect Sales (2).

Use this when you want:

- To explore a different version of the same data
- To try something new without affecting the original

Reference:

If you create a reference from Sales:

- A new query (e.g., Sales_Ref) is created.
- It starts from the final output of the Sales query.
- Any changes in the original Sales query will affect Sales_Ref.

Use this when you:

- Want to reuse the cleaned/transformed data from another query

- Want to apply additional steps without repeating the entire process

6. Merge Orders.csv and Customers.xlsx on CustID (inner join).

To merge Orders.csv and Customers.xlsx on CustID using an inner join in Power BI (Power Query), follow these steps:

Step-by-step in Power BI (Power Query):

1. Load the data sources:

- Go to Home → Get Data → Text/CSV and load Orders.csv.
- Go to Home → Get Data → Excel and load Customers.xlsx.
- After loading both, click Transform Data to open Power Query Editor.

2. Merge Queries:

- In Power Query Editor, select the Orders query.
- Go to the ribbon → Home → Merge Queries → Merge Queries as New (creates a new merged query).
- In the Merge dialog:
    - For the first table, select Orders.
    - For the second table, select Customers.
    - Select the CustID column in both tables by clicking the column header.
    - At the bottom, choose Join Kind → Inner (only matching rows).
- Click OK.

3. Expand merged table:

- You will see a new column (usually called Customers).
- Click the expand icon (two arrows) next to the Customers column.
- Select the columns you want to include from Customers (like CustomerName, Email, etc.).
- Uncheck Use original column name as prefix if you want clean column names.
- Click OK.

4. Close and Apply:

- Click Close & Apply to load the merged data back into Power BI.

7. **Pivot the Product column to show total Quantity per product.**

   Steps to Pivot the Product column to show total Quantity per product:

   1. Load your data into Power BI → Power Query Editor.

   2. Select the Product column.

   3. On the ribbon, go to:

   Transform → Pivot Column

   4. In the Pivot Column dialog:

- For Values Column, select Quantity.
- For Aggregate Value Function, select Sum.

8. **Append two tables with identical columns (e.g., Orders_Jan.csv + Orders_Feb.csv).**

   Step-by-Step Guide in Power BI:

- Step 1: Load both CSV files
  1. Open Power BI Desktop.
  2. Go to Home → Get Data → Text/CSV.
  3. Load Orders_Jan.csv, then load Orders_Feb.csv.
  4. Click Transform Data to open Power Query Editor.
- Step 2: Append Queries
  1. In Power Query Editor, go to the Home tab.
  2. Click Append Queries → Append Queries as New (to create a new combined table).
  3. In the Append dialog:
     - Select Orders_Jan as Table 1.
     - Select Orders_Feb as Table 2.
     - Click OK.

9. **Use "Fill Down" to replace nulls in the Email column with the previous value.**

1. Open **Power BI Desktop**.
2. Load your data and open **Power Query Editor** (Home → Transform Data).
3. Select **Students** table.
4. Select the **Email** column.
5. Go to the **Transform** tab.
6. Click on **Fill → Down**.

10. Extract the domain (e.g., "example.com") from the Email column.

Method 1: Using Power Query UI (no code)

1. Load your table into Power BI → open Power Query Editor.
2. Select the Email column.
3. Go to Add Column → Extract → Text After Delimiter.
4. Enter @ as the delimiter.
5. Click OK.

Method 2: Using M Code (Custom Column)

If you prefer M code or want to customize:

1. Go to Add Column → Custom Column.
2. Use this formula:

m

CopyEdit

Text.AfterDelimiter([Email], "@")

11. Write M-code to merge queries dynamically based on a parameter (e.g., JoinType = "Inner").

1. In Power BI → Home → Manage Parameters → New Parameter.
2. Name it: JoinType
3. Data Type: Text
4. Suggested Values: List
   - Inner
   - Left Outer

- Right Outer
- Full Outer

5. Set default value to "Inner"
6. Click OK

```
let
  // Sample join type parameter
  JoinType = Text.From(#"JoinType"),

  // Define actual join kinds mapping
  JoinKindMap = [
    "Inner" = JoinKind.Inner,
    "Left Outer" = JoinKind.LeftOuter,
    "Right Outer" = JoinKind.RightOuter,
    "Full Outer" = JoinKind.FullOuter
  ],

  // Get the join kind from the map
  JoinMode = Record.Field(JoinKindMap, JoinType),

  // Perform dynamic merge
  MergedTables = Table.NestedJoin(Customers, {"CustID"}, Orders, {"CustID"}, "OrderDetails", JoinMode),

  // Expand merged table
  Expanded = Table.ExpandTableColumn(MergedTables, "OrderDetails", {"OrderID", "Amount"})
in
  Expanded
```

12. **Unpivot a table with columns like "Jan_Sales," "Feb_Sales" into a "Month" and "Sales" format.**

    1. Load your table into Power Query (e.g., from Excel or CSV).
    2. Select the columns you *don't* want to unpivot — e.g., select Product.
    3. Go to the Transform tab → Unpivot Columns (or right-click → Unpivot Other Columns).

       This will transform all the sales columns into two columns:

- Attribute → becomes Month
- Value → becomes Sales

    4. Rename:
        - Attribute → Month
        - Value → Sales

13. **Handle errors in a custom column (e.g., division by zero) using try...otherwise.**

Syntax:

m

try [expression] otherwise [fallback_value]

```
let
  Source = Table.FromRecords({
    [Product="Apple", Sales=100, Units=10],
    [Product="Banana", Sales=80, Units=0],
    [Product="Orange", Sales=60, Units=5]
  }),
  AddColumn = Table.AddColumn(Source, "PricePerUnit", each try [Sales] / [Units] otherwise null)
in
  AddColumn
```

14. **Create a function in Power Query to clean phone numbers (e.g., remove dashes).**

Home → Advanced Editor → paste this function code:

m

```
let
  CleanPhone = (phone as text) as text =>
    let
      noDashes = Text.Replace(phone, "-", ""),
      noSpaces = Text.Replace(noDashes, " ", ""),
      noParens = Text.Replace(Text.Replace(noSpaces, "(", ""), ")", "")
    in
      noParens
in
  CleanPhone
```

This function removes:

- - dashes
- (and) parentheses
- Spaces

Example Usage in a Query

Assume you have a column called PhoneNumber.

1. Add a new Custom Column:

```
m
= CleanPhone([PhoneNumber])
```

15. Optimize a query with 10+ steps—identify bottlenecks and simplify.

1. Remove Unnecessary Columns Early

- Keep only the columns you need.
- This reduces memory usage and speeds up processing.

2. Filter Rows Before Heavy Operations

- Apply filters before joins, grouping, or sorting.

3. Combine Steps

- Merge multiple steps into one when possible.

Example: Combine column additions and transformations together.

4. Minimize "Changed Type" Steps

- Apply data types once at the end instead of repeatedly.

5. Use Table.Buffer() If Needed

- Use when the same table is used multiple times to avoid recalculating.

Be careful—it uses memory.

6. Avoid Unnecessary Columns After Joins

- After merging or expanding, immediately remove unwanted columns.

7. Use Query Dependencies View

- Check View → Query Dependencies to find long or complex chains.

8. Profile Columns

- Turn on Column Profiling to detect nulls or bad data early.