

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?
 - Merge: Adds columns based on matching values in common key columns.
 - Append: Adds rows when tables have the same structure.
2. How do you split a "Full Name" column into "First Name" and "Last Name"?
 - Navigate to the "Transform" tab in the Power Query Editor ribbon.
 - In the "Text Column" group, click on "Split Column" and then select "By Delimiter."
 - The "Split Column by Delimiter" dialog box will appear.
 - In the "Select or enter delimiter" dropdown, choose "Space" as the delimiter, assuming the first and last names are separated by a space. If your names are separated by a comma and space (e.g., "Doe, John"), you would choose "Custom" and enter ", " (comma and a space).
 - After the split, you will have new columns (e.g., "Full Name.1", "Full Name.2"). Double-click on the header of each new column and rename them to "First Name" and "Last Name" respectively.
 -
3. What is "Pivot Columns" used for?
 - "Pivot Columns" transforms data by taking the distinct values from a selected column and turning them into new column headers. The values from another specified column are then organized under these new headers, effectively rotating your table's orientation from vertical rows to horizontal columns for that specific data
4. How do you undo a step in Power Query?
 - To undo a step in Power Query, navigate to the Applied Steps pane within the Power Query Editor, locate the desired step in the list and click the 'X' icon (red cross mark) next to it. This will remove that specific step and revert the data to its state before that step was applied.

You can remove multiple steps by clicking the 'X' icon next to each one in reverse chronological order. This effectively "undos" the changes in the order they were applied.

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

- A Duplicate query creates an independent, separate copy of the original, allowing for isolated transformations without affecting the source. A Reference query, conversely, creates a dynamic, linked query that points to the original data source, automatically reflecting any updates made to the source or other referencing queries. You should use Duplicate for sandboxing and independent testing, and Reference for efficiency, data reuse, and maintaining a single source of truth across different views.

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

	1 ² ₃ OrderID	1 ² ₃ CustID	A ⁸ _C Product	1 ² ₃ Quantity	1 ² ₃ Customers.CustID	A ⁸ _C Customers.Name	A ⁸ _C Customers.Email
1		1001	101 Laptop		1	101 Alice	alice@example.com
2		1003	101 Keyboard		2	101 Alice	alice@example.com
3		1002	102 Mouse		3	102 Bob	bob@example.com
4		1004	103 Monitor		1	103 Charlie	charlie@example.com

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

- Transform>Pivot Column

	1 ² ₃ Laptop	1 ² ₃ Keyboard	1 ² ₃ Mouse	1 ² ₃ Monitor
	1	null	null	null
	null	null	3	null
	null	2	null	null
	null	null	null	1

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

Go to on Home Tab GetData>More..>Folder and select the path where these two tables are. In my case the path where Orders_Jan.csv and Orders_Feb.csv files are on "D:\Learning\DataAnalytics\PowerBI\Class4\CSV_FILES ". Click "OK", then select "Combine and Transform Data" or "Combine and Load" options from "Combine" button and OK.

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

- Select the 'Email' Column. In the Power Query Editor, locate and select the 'Email' column in your dataset that contains the null values you wish to fill. Apply Fill Down: Go to the "Transform" tab in the Power Query Editor ribbon.
- In the "Any Column" group, click on the "Fill" dropdown. Select "Down"

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

- In Power Editor Add Column tab > Extract > Text After Delimiter. In Delimiter field type "@" and OK.

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

let

 // Sample queries

 Source1 = Orders,

 Source2 = Customers,

 // Mapping between parameter text and actual JoinKind

 JoinKinds = [

 Inner = JoinKind.Inner,

 LeftOuter = JoinKind.LeftOuter,

 RightOuter = JoinKind.RightOuter,

 FullOuter = JoinKind.FullOuter,

 LeftAnti = JoinKind.LeftAnti,

 RightAnti = JoinKind.RightAnti

],

```

// Pick the JoinKind dynamically based on the parameter

SelectedJoin = Record.Field(JoinKinds, JoinType),

// Perform the merge

Merged = Table.NestedJoin(Source1, {"CustID"},

    Source2, {"CustID"},

    "NewTable",

    SelectedJoin),

#"Expanded NewTable" = Table.ExpandTableColumn(Merged, "NewTable",
{"CustID", "Name", "Email"}, {"NewTable.CustID", "NewTable.Name",
"NewTable.Email"}),

Expanded = Table.ExpandTableColumn(#"Expanded NewTable", "NewTable",
{"Name","email"}),

#"Changed Type" = Table.TransformColumnTypes(Expanded,{{"email", type
text}})

in

#"Changed Type"

```

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

Load your table into Power Query.

Select the columns you want to keep as identifiers (e.g., ProductID).

Right-click → Unpivot Other Columns.

Rename the new columns:

"Attribute" → Month (will contain Jan_Sales, Feb_Sales, ...).

"Value" → Sales.

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

- = try [Sales] / [Quantity] otherwise null

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

- Create a new blank query

In Power BI / Power Query: Home → New Source → Blank Query

Rename it: fnCleanPhone: M function:

let

fnCleanPhone = (phone as text) as text =>

let

// Remove common formatting characters

Step1 = Text.Remove(phone, {"-", "(", ")", " ", "."}),

// Optional: keep only digits

Cleaned = Text.Select(Step1, {"0".."9"})

in

Cleaned

in

fnCleanPhone

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

Common Bottlenecks in Power Query

1. Row-by-row operations

- Custom column calculations that aren't foldable (e.g., `Text.Contains`, `List.Generate`, `Table.Buffer`) can kill performance.

2. Unnecessary intermediate steps

- Each step is a new query execution → avoid steps that just rename, reorder, or expand multiple times.

3. Unpivot/Pivot on large tables

- Can be very expensive if done before filtering.

4. Merging/Joining before filtering

- Joins on millions of rows are slow; always filter down first.

5. Use of `Table.Buffer()` incorrectly

- Can prevent query folding and increase memory usage.

6. Using Excel/CSV as a source without specifying range

- Brings in unused rows/columns.

Strategies to Simplify & Optimize

1. Push filters upstream (query folding)

- Apply filters, column selection, and data reduction *as early as possible*.
- Check if steps are folded (View → Query Diagnostics or check SQL in data source).

2. Combine simple transformations

- Instead of multiple steps:
- `#"Renamed Columns" = Table.RenameColumns(...),`
- `#"Changed Type" = Table.TransformColumnTypes(...)`

→ combine into one step if possible.

3. Avoid unnecessary sorts

- Sorting breaks folding unless required. Remove if not needed.

4. Minimize the use of `Table.Distinct` and `Remove Duplicates`

- Very expensive. Filter upstream or deduplicate in the source if possible.

5. Defer column renames until the end

- Helps folding. Rename once at the end, not multiple times.
- 6. **Diagnostics**
 - Use **Tools** → **Diagnose Step** to see which step takes the most time.

In general, **Simplify** → push filters early, combine steps, reduce columns early, check folding, use query diagnostics.