

Homework 4

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

In **Power Query**, **Merge** and **Append** are both used to combine data from multiple tables, but they work in **different ways**:

◆ Merge

- **Purpose:** Combines columns (side-by-side).
- **How it works:** Similar to doing a SQL **JOIN**.
- You select one or more **key columns** that exist in both tables.
- Rows are matched based on those keys.
- The result adds **new columns** from the second table into the first.
- Example:

You have a *Sales* table with CustomerID and a *Customers* table with CustomerID, Name, City.

- Merge on CustomerID → brings Name and City into the *Sales* table.
-

◆ Append

- **Purpose:** Combines rows (stacking one under another).
- **How it works:** Similar to doing a SQL **UNION**.
- Tables must have the same or similar columns.
- The result is a **longer table** with all rows from both (or more) tables.
- Example:
You have *Sales2023* and *Sales2024* tables with the same columns.
 - Append them → creates one big table with all rows from both years.

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

In **Power Query**, you can split a "Full Name" column into "First Name" and "Last Name" using the **Split Column** feature. Here's how:

◆ Method 1: Using the Ribbon

1. Select the **Full Name** column.
 2. Go to the **Home** or **Transform** tab → **Split Column** → **By Delimiter**.
 3. Choose **Space ()** as the delimiter.
 4. In the dialog, pick **At the left-most delimiter** (if the name has only first and last).
 5. You'll now get two columns: "Full Name.1" and "Full Name.2".
 6. Rename them to "First Name" and "Last Name".
-

◆ Method 2: M Code

If your column is named "Full Name", you can add this step in Advanced Editor:

```
= Table.SplitColumn(  
    Source,  
    "Full Name",  
    Splitter.SplitTextByDelimiter(" ", QuoteStyle.Csv),  
    {"First Name", "Last Name"}  
)
```

3. What is "Pivot Columns" used for?

In **Power Query**, **Pivot Columns** is used to **reshape your data** — it turns **unique values in a column into new column headers**.

◆ **What it does**

- Takes a column with repeated values (categories).
- Turns each **unique value** in that column into a **separate column**.
- Fills those new columns with values from another column you choose.

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

In **Power Query**, every transformation you apply is recorded as a **step** in the **Applied Steps** pane (on the right side of the editor).

◆ **To undo a step:**

1. Look at the **Applied Steps** pane.
2. Find the step you want to undo.
3. Either:
 - Click the  (X) next to the step name → removes it.
 - Or right-click the step and choose **Delete**.

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

◊ **Duplicate**

- Creates a **new, independent copy** of the query.
- The new query has **all the same steps**, but it's **not linked** to the original.
- If you later modify the original query, the duplicate does **not** update.
- Good for: when you want to start fresh from the same source but make **different transformations** independently.

Analogy: Like copy–pasting data into a new file → the two files don't affect each other.

◊ Reference

- Creates a **new query that points back to the original**.
- The reference query starts from the **final result** of the original query.
- If you change the original query, the reference query **updates automatically**.
- Good for: when you want to create multiple outputs/views (e.g., summary tables) that share the **same base transformations**.

Analogy: Like creating a *view* in SQL → changes to the base table flow into the view.

6. Merge Orders.csv and Customers.xlsx on CustID (Inner Join)

- Go to **Home** → **Merge Queries**.
- Select **Orders** as the first table, **Customers** as the second.
- Choose the column **CustID** in both.
- Select **Join kind = Inner Join**.
- Expand the merged table to include the fields you need (OrderID, Product, Quantity, etc.).

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

- Select the **Product** column.
- Go to **Transform** → **Pivot Column**.
- For the values, choose **Quantity**.
- In the aggregation, choose **Sum**.
↳ Each product becomes a column, with total quantities filled in.

8. Append two tables (Orders_Jan.csv + Orders_Feb.csv)

- Go to **Home** → **Append Queries**.
- Select **Orders_Jan** and **Orders_Feb**.
- Since both tables have identical columns, Power Query will stack them together.
↳ You now have one combined table with rows from both months.

9. Use Fill Down to replace nulls in the Email column

- Select the **Email** column.

- Go to **Transform → Fill → Down**.
↳ Any blank cells are filled with the value from the row above.

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

You can extract the **domain part** (everything after @) from an Email column in **Power Query**.

◊ Method 1: Using the Ribbon

1. Select the **Email** column.
2. Go to **Add Column → Extract → Text After Delimiter**.
3. Enter @ as the delimiter.
4. You'll get a new column with just the domain (e.g., example.com).

◊ Method 2: M Code

Here's how the transformation looks in **M**:

- ```
= Table.AddColumn(Source, "Domain", each Text.AfterDelimiter([Email], "@"))
• Source = your previous step.
• "Domain" = name of new column.
• Text.AfterDelimiter([Email], "@") → extracts everything after @.
```

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

```
let
 // Example tables
 Customers = Excel.CurrentWorkbook(){[Name="Customers"]}[Content],
 Orders = Excel.CurrentWorkbook(){[Name="Orders"]}[Content],

 // Parameter: JoinType (must be created in Power Query UI as a Text parameter)
 // e.g. "Inner", "LeftOuter", "RightOuter", "FullOuter"

 // Map parameter value to actual JoinKind
 JoinKindMapping =
 if JoinType = "Inner" then JoinKind.Inner
 else
```

**12. Unpivot a table with columns like "Jan\_Sales," "Feb\_Sales" into a "Month" and "Sales" format.**

```
let
 Source = Csv.Document(
 File.Contents("C:\Users\user\Downloads\SalesData.csv"),
 [Delimiter=",", Columns=4, Encoding=1251,
 QuoteStyle=QuoteStyle.None]
),
 #"Promoted Headers" = Table.PromoteHeaders(Source,
 [PromoteAllScalars=true]),
 #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",
 {"Product", type text}, {"Jan_Sales", Int64.Type}, {"Feb_Sales",
 Int64.Type}, {"Mar_Sales", Int64.Type}),

 // Unpivot Sales columns
 #"Unpivoted Columns" = Table.Unpivot(
 #"Changed Type",
 {"Jan_Sales", "Feb_Sales", "Mar_Sales"},
 "Month",
 "Sales"
),

 // Clean up Month column (remove "_Sales")
 #"Extracted Month" = Table.TransformColumns(
 #"Unpivoted Columns",
 {"Month", each Text.BeforeDelimiter("_", "_"), type text}
)
in
#"Extracted Month"
```

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

```
let
 Source = Csv.Document(File.Contents("C:\Users\user\Downloads\Orders_Jan.csv"),
 [Delimiter=",", Columns=4, Encoding=1251, QuoteStyle=QuoteStyle.None]),
 #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
 #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",
 {"OrderID", Int64.Type}, {"Product", type text}, {"Quantity", Int64.Type}, {"Price",
 Int64.Type})),
```

```

// Append with Orders_Feb
#"Appended Query" = Table.Combine({#"Changed Type", Orders_Feb}),

// Custom column with error handling
#"Added Custom" = Table.AddColumn(
 #"Appended Query",
 "UnitPrice",
 each try [Price] / [Quantity] otherwise null,
 type number
)
in
#"Added Custom"

```

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

```

let
 fnCleanPhone = (phone as nullable text) as nullable text =>
 let
 SafePhone = if phone = null then null else phone,
 RemoveDashes = Text.Replace(SafePhone, "-", ""),
 RemoveSpaces = Text.Replace(RemoveDashes, " ", ""),
 RemoveBrackets1 = Text.Replace(RemoveSpaces, "(", ""),
 RemoveBrackets2 = Text.Replace(RemoveBrackets1, ")", ""),
 RemovePlus = Text.Replace(RemoveBrackets2, "+", ""),
 Cleaned = RemovePlus
 in
 Cleaned
in
fnCleanPhone

```

```
#"Cleaned Phone" = Table.TransformColumns(
 #"Replaced Value",
 {"Phone", each fnCleanPhone(_), type text}
)
```

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

---

### ⌚ Common Bottlenecks in Power Query (10+ steps)

#### 1. Loading all columns first

- Many queries load every column and then remove unneeded ones later.
- **! Fix:** Use Table.SelectColumns or *Remove Other Columns* as early as possible.

#### 2. Filtering too late

- If you filter rows only at the end, Power Query processes the full dataset first.
- **! Fix:** Apply filters (e.g., Year = 2023) early in the query.

#### 3. Multiple Changed Type steps

- Power Query auto-generates “Changed Type” for each step.
- **! Fix:** Consolidate type changes into **one single step**.

#### 4. Merging/Appending with unnecessary data

- Joins are slower if extra columns remain.
- **! Fix:** Keep only necessary columns before merging/appending.

#### 5. Heavy Group By / Pivot / Unpivot

- These transformations are slow on large datasets.
- **! Fix:** Reduce data first (filter/remove columns) before grouping/pivoting.

---

### ↳ General Optimization Rules

- **✗ Remove redundant steps (especially duplicate Changed Type).**

- Combine transformations into fewer steps (e.g., one Table.TransformColumns).
  - Remove unneeded columns as early as possible.
  - Apply row filters early.
  - Use **Reference** instead of **Duplicate** queries (References reuse the source, Duplicates reload it).
- 

## ❖ Example: Before vs After

### Before (10+ steps)

```
#"Changed Type" = Table.TransformColumnTypes(Source, {"OrderDate", type date}),
#"Removed Columns" = Table.RemoveColumns(#"Changed Type", {"Extra1", "Extra2"}),
#"Filtered Rows" = Table.SelectRows(#"Removed Columns", each [Quantity] > 1),
#"Changed Type1" = Table.TransformColumnTypes(#"Filtered Rows", {"Quantity", Int64.Type}),
#"Changed Type2" = Table.TransformColumnTypes(#"Changed Type1", {"Price", type number})
```

### After (optimized, 3 steps)

```
#"Removed Columns" = Table.RemoveColumns(Source, {"Extra1", "Extra2"}),
#"Filtered Rows" = Table.SelectRows(#"Removed Columns", each [Quantity] > 1),
#"Changed Types" = Table.TransformColumnTypes(#"Filtered Rows",
 {"OrderDate", type date}, {"Quantity", Int64.Type}, {"Price", type number})
```