What is the difference between "Merge" and "Append" in Power Query?
In **Power Query**, **Merge** and **Append** both combine tables, but they work in different ways:

Merge 🔗 (Like SQL JOIN)

- **Purpose:** Combines **columns** from two tables based on a matching key column (or columns).
- **How it works:**
  - You choose one or more key columns in each table.
  - Power Query matches rows where the keys are equal.
  - It returns one table with columns from both sources.
- **Example:**
  - Table 1: Customer info
  - Table 2: Orders
  - Merge on **CustID** → get one table showing customers with their orders.

Append ➕ (Like SQL UNION)

- **Purpose:** Stacks tables **on top of each other**, combining rows.
- **How it works:**
  - Tables should have the same column structure (same names and data types).
  - All rows from the second table are added after the rows from the first table.
- **Example:**
  - Table 1: January sales
  - Table 2: February sales
  - Append → one table with sales from January and February together.

💡 **Quick memory tip:**

- **Merge** → "Match & Add columns"
- **Append** → "Stack & Add rows"

If you want, I can prepare a **visual diagram** showing Merge vs. Append with a simple example. That makes it much easier to remember.

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

In **Power Query**, to split a **Full Name** column into **First Name** and **Last Name**:

Steps:

1. Select the **Full Name** column.
2. Go to **Home** (or **Transform**) → **Split Column** → **By Delimiter**.
3. In the dialog:
   - **Delimiter:** Select **Space** (or type a space character).
   - **Split At:** Choose **Left-most delimiter** (so only the first space is used).
4. Click **OK**.

This will create two columns — rename them **First Name** and **Last Name**.

What is "Pivot Columns" used for?

In **Power Query**, **Pivot Columns** is used to **turn unique values from one column into new column headers** and arrange the related data accordingly.

Purpose:

- Converts **row values** into **columns**.
- Helps summarize and reshape data for reporting.

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

In **Power Query**, **Reference** and **Duplicate** both create a new query from an existing one, but they behave very differently:

---

1. Duplicate

- **What it does:** Creates a **completely independent copy** of the query, including all steps.
- **Effect:** If you change the original query later, the duplicate will **not** be updated automatically.
- **When to use:** If you want to reuse the same transformation steps but then modify them independently.

---

2. Reference

- **What it does:** Creates a **linked query** that starts from the **final result** of the original query.
- **Effect:** If the original query changes, the reference query will **update automatically**.
- **When to use:** If you want to build on top of an existing query without copying all its steps.

Analogy:

- **Duplicate** = photocopy a sheet of paper — later edits to the original won't appear in the copy.
- **Reference** = using a live link to a spreadsheet cell — if the original changes, your link shows the new value.

If you want, I can make a **side-by-side diagram** showing how Reference and Duplicate flow works in Power Query.

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

```
let
    // Parametr (user bu yerda tanlaydi: "Inner", "LeftOuter", "RightOuter", "FullOuter", "LeftAnti", "RightAnti")
    JoinType = if Text.Upper(Parameters[JoinType]) = "INNER" then JoinKind.Inner
            else if Text.Upper(Parameters[JoinType]) = "LEFTOUTER" then JoinKind.LeftOuter
            else if Text.Upper(Parameters[JoinType]) = "RIGHTOUTER" then JoinKind.RightOuter
            else if Text.Upper(Parameters[JoinType]) = "FULLOUTER" then JoinKind.FullOuter
            else if Text.Upper(Parameters[JoinType]) = "LEFTANTI" then JoinKind.LeftAnti
            else if Text.Upper(Parameters[JoinType]) = "RIGHTANTI" then JoinKind.RightAnti
            else JoinKind.Inner, // default Inner
```

```
    // Ikkita query misol uchun
    Source1 = Orders_Jan,
    Source2 = Orders_Feb,

    // Dinamik merge
    Merged = Table.NestedJoin(Source1, {"custid"}, Source2, {"custid"}, "JoinedTable",
JoinType),

    // Expand qilingan variant
    Expanded = Table.ExpandTableColumn(Merged, "JoinedTable", {"cust"})
 in
    Expanded
```

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

- **Unpivot using the UI (no M-code needed)**

1. Load the table into Power Query.
2. Select the columns you **don't want to unpivot** (e.g. CustID, CustName).
3. Right-click → **Unpivot Other Columns**.
4. Rename the new columns:
    - Attribute → Month
    - Value → Sales
5. If you want just the month name (e.g. "Jan" from "Jan_Sales"):
    - Go to **Transform → Extract → Text Before Delimiter _**.

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

```
 let
    Source = Table.FromRecords({
       [A=10, B=2],
       [A=15, B=0],
       [A=20, B=5]
    }),
    AddedColumn = Table.AddColumn(Source, "SafeDivide", each try [A] / [B] otherwise
null)
 in
    AddedColumn
```

1. ***Create a function in Power Query to clean phone numbers (e.g., remove dashes).***
2. - **Step 1. Create a blank query → turn it into a function**
3. Go to **Home → Advanced Editor** and paste this code:

```
// Function to clean phone numbers
(phone as text) as text =>
let
    RemoveDashes   = Text.Replace(phone, "-", ""),
    RemoveSpaces   = Text.Replace(RemoveDashes, " ", ""),
    RemoveBrackets = Text.Remove(RemoveSpaces, {"(", ")", "+", "."}),
     Result         = RemoveBrackets
in
```

12. Result

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

Optimization is about **removing unnecessary steps, folding as much as possible, and reducing transformations**.
Here's how you can approach it systematically:

- **1. Identify Bottlenecks**
  - **Applied Steps Pane** → Look for:
    - Multiple type changes (Changed Type steps repeated).
    - Repeated column removals.
    - Sorting & filtering done multiple times.
    - Joins/merges done too early.
  - **Query Diagnostics** (Power BI / Excel → *Tools > Diagnose Step*) shows which steps consume the most time.