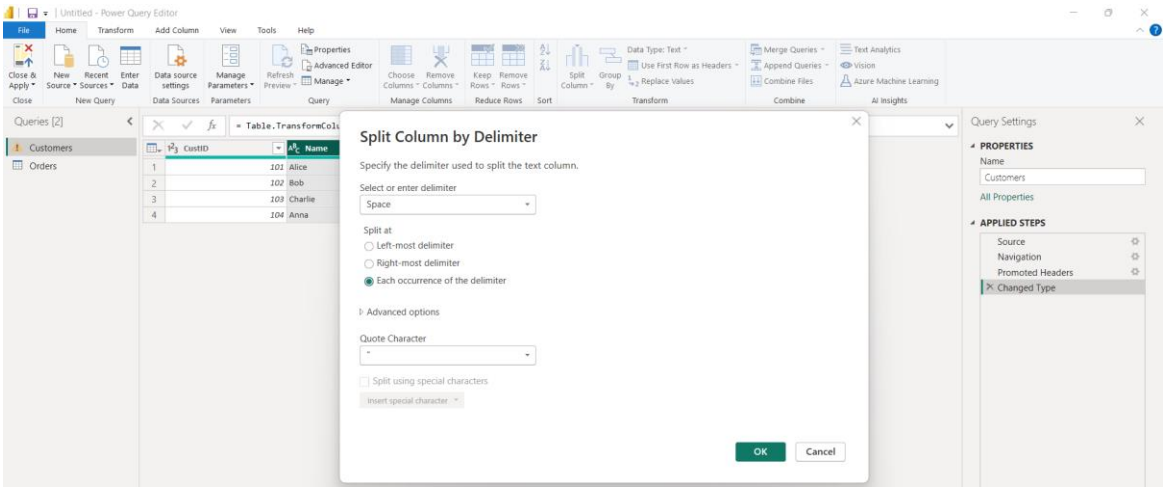
1. What is the difference between "Merge" and "Append" in Power Query?

Merge is used to join two tables based on matching column values (like SQL JOIN).

Append is used to stack tables vertically (like adding more rows).

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

Column should be chosen. And then Split column --- By delimeter ----- Delimeter- Space --- Ok. Then we will have 2 columns out of one. Columns should be named to First Name and Second Name

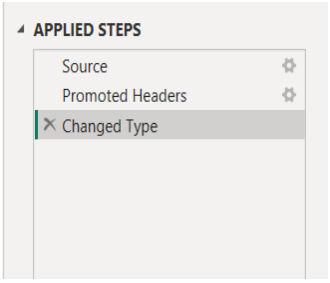


3. What is "Pivot Columns" used for?

In Power Query, Pivot Columns is used to transform row values into column headers and rearrange the data accordingly. It converts unique values from one column into multiple new columns, and summarizes the corresponding data (e.g., sum, count, average).

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

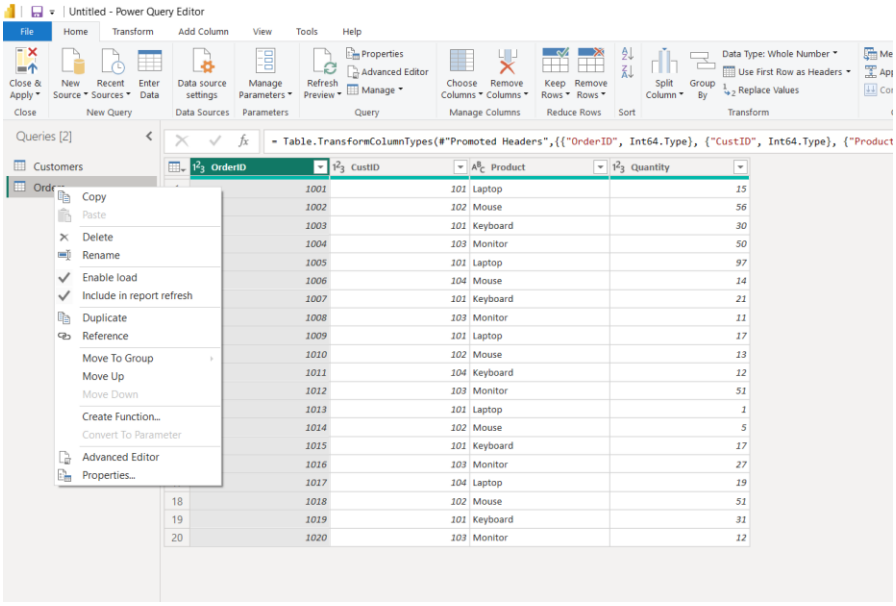
At the right side, in the applied steps tab, there are steps that were done to the table. It is possible to delete the steps and the table will come to the initial stage when the operation that was deleted, was not done



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

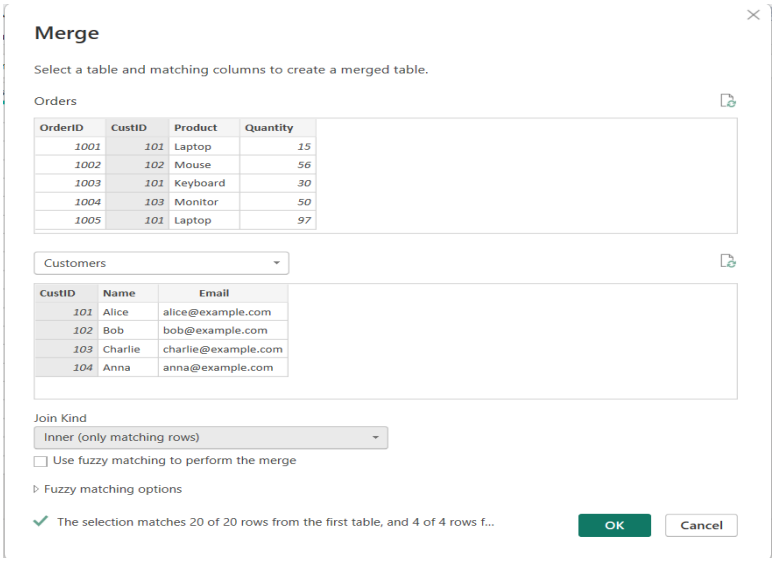
Duplicate creates a full copy of the existing query, including all applied steps. The new query is independent — if you change the original query, the duplicate will not update automatically.

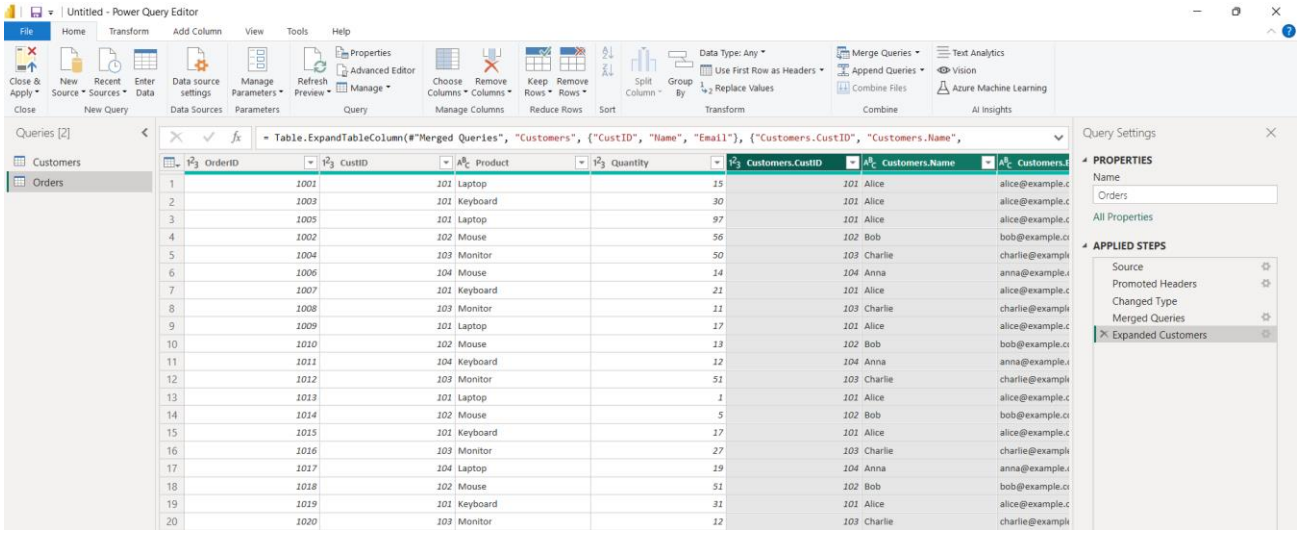
Reference creates a new query that points to the output of the original query. If the original query changes, the referenced query automatically reflects those changes.



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

After selecting orders.csv and entering it, merge queries should be selected in Home Tab. As the second table, customers table should be chosen. As the common columns, customerID column should be selected and inner join option should be selected. Then Ok. After, columns of customers will be added to the columns of orders.

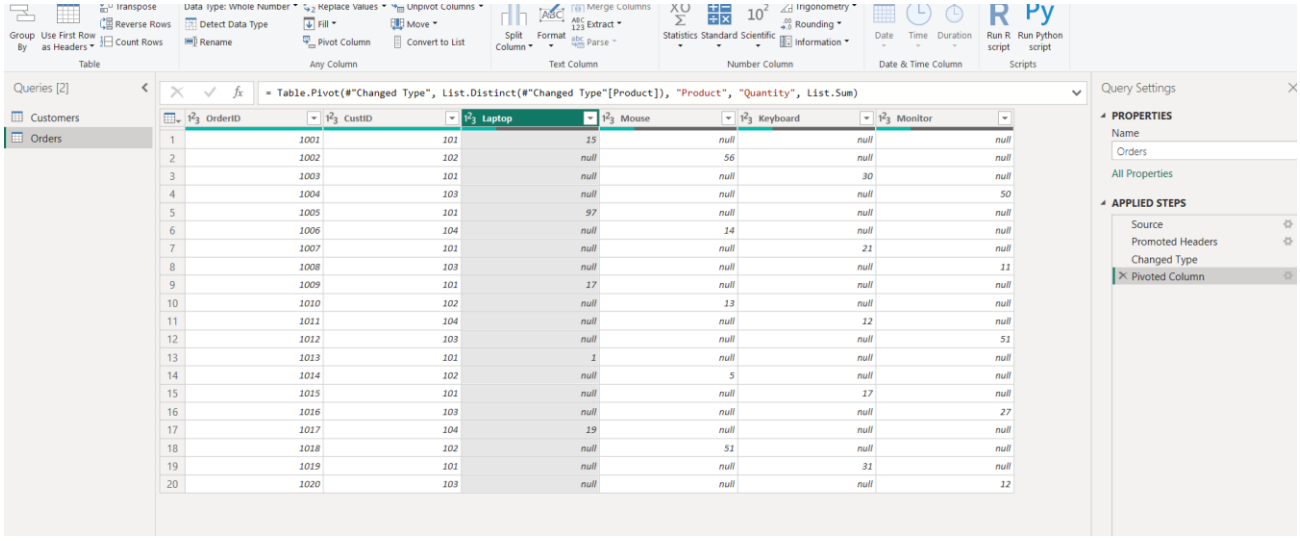




7. Pivot the Product column to show total Quantity per product.

To pivot the Product column in Power Query so you see total Quantity per product, follow these steps:

1. Select the Product column.
2. Go to the Transform tab.
3. Click Pivot Column.
4. In the dialog:
    - Values Column → Select Quantity.
    - Aggregate Function → Select Sum (so quantities add up for each product).
5. Click OK.

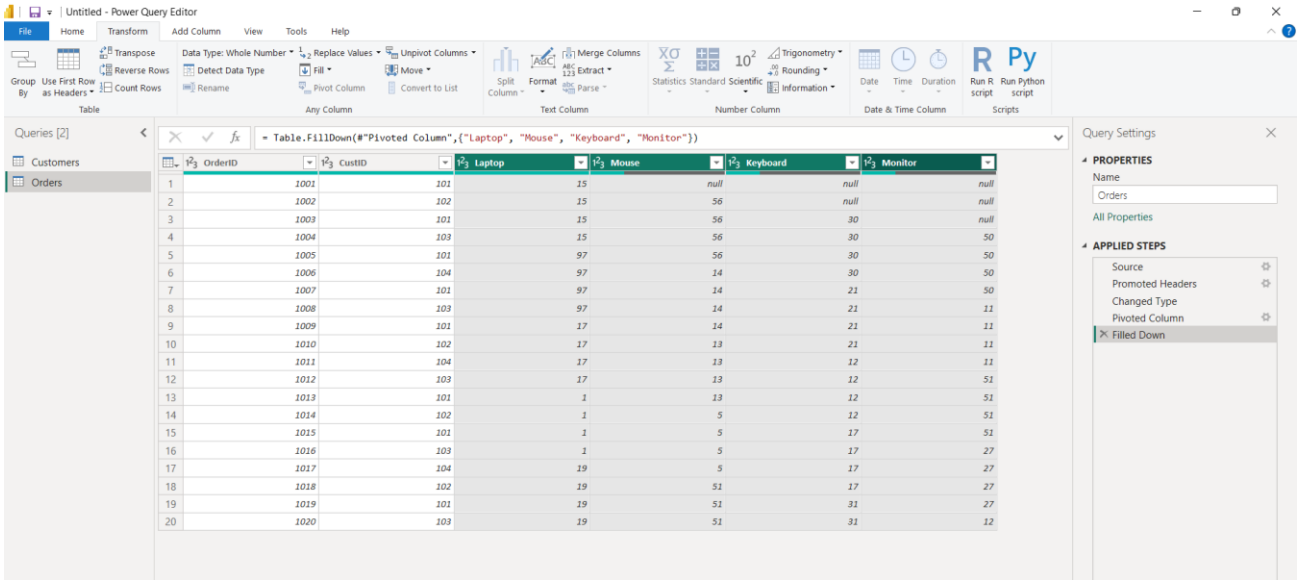


8. Append two tables with identical columns (e.g., Orders_Jan.csv + Orders_Feb.csv).

After importing Orders_Jan and Orders_Feb, In Power Query, go to **Home → Append Queries**. Choose **Two tables**. Pick Orders_Jan and Orders_Feb. Click **OK**. You should now see all rows from both files in one table. Click **Close & Load** to apply.

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

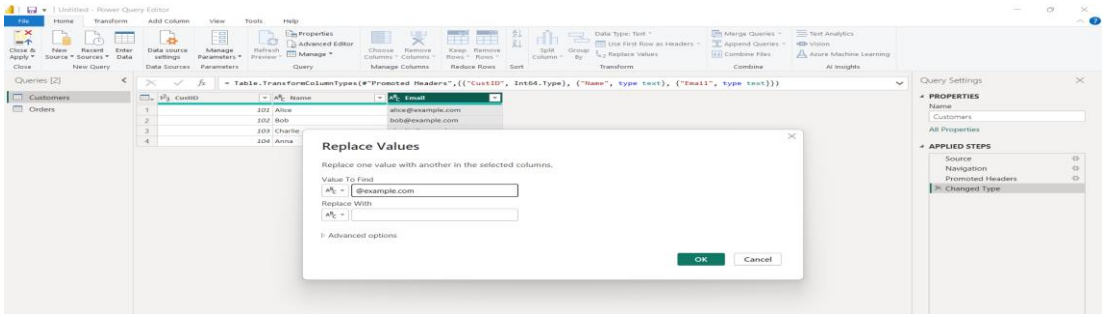
The email column should be selected. In Home tab, fill should be selected and Down option should be selected. In the result, null value will be replaced with the previous value.



	OrderID	CustID	Laptop	Mouse	Keyboard	Monitor
1	1001	101	15	null	null	null
2	1002	102	15	56	null	null
3	1003	101	15	56	30	null
4	1004	103	15	56	30	50
5	1005	101	97	56	30	50
6	1006	104	97	14	30	50
7	1007	101	97	14	21	50
8	1008	103	97	14	21	11
9	1009	101	17	14	21	11
10	1010	102	17	13	21	11
11	1011	104	17	13	12	11
12	1012	103	17	13	12	51
13	1013	101	1	13	12	51
14	1014	102	1	5	12	51
15	1015	101	1	5	17	51
16	1016	103	1	5	17	27
17	1017	104	19	5	17	27
18	1018	102	19	51	17	27
19	1019	101	19	51	31	27
20	1020	103	19	51	31	12

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

Email column should be selected, in the Transform tab, replace values should be selected. The new window will be opened. In value to find entry, @example.com should be entered and replace with should be empty. In this way, @example.com will be extracted



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

```
let
    // Parameters
    JoinType = "Inner", // Change this parameter to control join type
    Query1 = YourFirstQuery,
    Query2 = YourSecondQuery,

    // Merge dynamically based on parameter
    MergedData = Table.NestedJoin(
        Query1,
        {"ID"},        // Key column(s) from Query1
        Query2,
        {"ID"},        // Key column(s) from Query2
        "NewTable",
        JoinType
    ),

    // Expand merged table
    ExpandedData = Table.ExpandTableColumn(MergedData, "NewTable",
{"ColumnFromQuery2"})
in
    ExpandedData
```

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

Load your table into Power Query (Home → Transform Data).

Select the columns you want to unpivot — for example, Jan_Sales and Feb_Sales.

Go to the Transform tab → click Unpivot Columns.

The result will have:

A new column (usually named Attribute) — rename it to Month.

A column with values (usually named Value) — rename it to Sales.

Close & Load to return the transformed table to Excel or Power BI.

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

Go to Add Column → Custom Column.

Enter:
try [Sales] / [Orders] otherwise 0
try attempts the calculation.

If it fails (e.g., division by zero or null), otherwise returns the fallback value (here 0).

Click OK → you'll now have a safe calculation column.

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

Open Power Query (Home → Transform Data).
Create a blank query:
Home → New Source → Blank Query.
- **Convert to a function**:

Go to **View → Advanced Editor**.
Replace everything with:

```
let
    // Remove dashes
    cleaned = Text.Replace(phone, "-", ""),
    // Remove spaces
    cleaned2 = Text.Replace(cleaned, " ", ""),
    // Remove parentheses
    cleaned3 = Text.Remove(cleaned2, {"(", ")", "+"})
in
    cleaned3
```

Name it **fnCleanPhone**.

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

1. Identify Bottlenecks

- **Check the Applied Steps pane** (right side in Power Query).
- Steps with a **gear icon** often re-load or recompute the entire dataset.
- **Merge**, **Append**, and **Group By** are usually heavy steps — see if they can be postponed or simplified.
- Look for steps where:
    - You filter after expanding a large table (instead, filter earlier).
    - You sort unnecessarily.
    - You add columns using inefficient operations.

---

2. Optimize Step Order

- **Filter Early** — Remove unnecessary rows before merges or joins.
- **Select Columns Early** — Remove unused columns as soon as possible.
- **Avoid sorting** unless absolutely necessary.
- **Reduce data size** before expensive transformations.

---

3. Combine Steps

- Merge multiple "Added Custom Column" steps into **one formula**.
- Combine multiple filters into **one filter step**.
- Avoid extra rename or reorder steps if they can be combined into a single `Table.TransformColumnNames` or drag-drop action.

---

4. Use Native Query Folding

- When connected to SQL or other databases, keep transformations *foldable* (shown by the **"View Native Query"** option).
- Operations like filters and column selection fold well; complex M formulas may break folding.

---

5. Replace Complex M Functions

- For heavy text manipulation, use **Text.Replace** or **Text.Remove** instead of multiple nested `Text.Middle`/`Text.Range`.
- For joins, prefer **Table.Join** with minimal columns rather than `Table.NestedJoin` followed by expansion.

---

6. Reduce Repeated Calculations

- If you calculate the same thing in multiple columns, calculate it once and reference it.
- For example, instead of calculating `TotalPrice` in two steps, calculate it once and reuse the column.

---

7. Disable Data Preview Auto-Load

- File → Options → Data Load → Turn off “Enable data preview to download in the background” — speeds up editing.