✅ 1. Difference between "Merge" and "Append"

- **Merge**
  - Combines **columns side by side** from two tables based on matching keys (similar to SQL JOIN).
  - Example: Adding customer details to each order.
- **Append**
  - Combines **rows** from two or more tables into one (stacks them vertically).
  - Example: Combining January and February sales tables into one dataset.

✅ 2. Split "Full Name" into "First Name" and "Last Name"

Steps:

1. Select **Full Name** column.
2. Go to **Home > Split Column > By Delimiter**.
3. Choose **Space** as delimiter.
4. Result: Two columns – First Name and Last Name.

✅ 3. Purpose of "Pivot Columns"

Pivot Columns transforms **rows into columns**.

Example:

Product	Month	Sales
A	Jan	10
A	Feb	20

Becomes:

Product	Jan	Feb
A	10	20

Useful for **creating summary tables** with months or categories as column headers.

✅ 4. Undo a step in Power Query

In the **Applied Steps** pane (right side), click the **X** next to the step you want to delete.

✅ 5. Reference vs. Duplicate

Feature	Reference	Duplicate
Creates	A **linked copy** dependent on the original query's output	A **separate copy** of the entire query and its steps
Impact	Changes in the original affect the reference	Independent – changes do not affect each other
Use case	When you want to build on an existing query's result without duplicating all steps	When you want a separate query with same initial logic

---

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

Steps:

1. Load both tables into Power Query.
2. Select **Orders** query.
3. Click **Merge Queries**.
4. Choose **Customers** table.
5. Select **CustID** in both tables.
6. Choose **Inner Join**.
7. Expand merged table to include customer details.

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

Steps:

1. Select **Product** column.
2. Go to **Transform > Pivot Column**.
3. For **Values Column**, select **Quantity**.
4. Use **Sum** as aggregation.

✅ 8. Append two tables with identical columns

Steps:

1. Go to **Home > Append Queries**.
2. Select both tables (Orders_Jan and Orders_Feb).
3. Click **OK** to stack them together.

✅ 9. Fill Down to replace nulls in Email column

Steps:

1. Select **Email** column.
2. Go to **Transform > Fill > Down**.

This fills null cells with the value above them.

✅ 10. Extract domain from Email

Steps:

1. Add a **Custom Column** with formula:

```
m
КопироватьРедактировать
Text.AfterDelimiter([Email], "@")
```

This returns everything after the **"@"**, e.g., example.com.

✅ 11. M-code to merge queries dynamically based on a parameter (JoinType)

Assuming you created a **parameter named JoinType** with value like "Inner", "LeftOuter", etc.

```
m
КопироватьРедактировать
let
    Orders = ... , // your Orders table
    Customers = ... , // your Customers table
    Merge = Table.NestedJoin(
        Orders,
        {"CustID"},
        Customers,
        {"CustID"},
        "Customers",
        Record.Field(JoinKind, JoinType)
    )
in
    Merge
```

✔ **Note:** JoinKind.Inner, JoinKind.LeftOuter, etc. are standard keywords in Power Query.

✅ 12. Unpivot Jan_Sales, Feb_Sales to Month, Sales format

Steps:

1. Select all other columns except Jan_Sales, Feb_Sales.
2. Go to **Transform > Unpivot Columns**.

This transforms:

Product	Jan_Sales	Feb_Sales
A	10	20

Into:

Product	Attribute	Value
A	Jan_Sales	10
A	Feb_Sales	20

You can then rename **Attribute** to **Month** and **Value** to **Sales**.

✅ 13. Handle division by zero with try...otherwise

Example custom column:

```
m
КопироватьРедактировать
try [Amount] / [Quantity] otherwise null
```

If Quantity is zero or causes error, it returns **null** instead.

✅ 14. Function to clean phone numbers (remove dashes)

Steps:

1. Go to **Home > Advanced Editor**.
2. Write this function:

```
m
КопироватьРедактировать
(phone as text) as text =>
    Text.Remove(phone, {"-", " "})
```

Name it **CleanPhoneNumber**.

Usage:
Add a custom column:

```
m
КопироватьРедактировать
= CleanPhoneNumber([Phone])
```

✅ 15. Optimize a query with 10+ steps

✔ **Best practices:**

- **Remove unnecessary steps** (e.g. intermediate renames not needed).
- **Combine transformations** into fewer steps (e.g. multiple Replace operations in one).
- **Disable load** for intermediate queries not needed in final output.
- **Use Table.Buffer** when referencing a static table multiple times to avoid re-evaluation.
- Avoid **expensive column-by-column operations** inside loops; use native transformations where possible.