

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

Operation	Purpose
Merge	Combines columns from two tables based on a key (like SQL JOIN)
Append	Combines rows from two or more tables (like UNION ALL)

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

- Select `Full Name` → **Transform** → **Split Column** → **By Delimiter**
 - Use **Space** as delimiter → choose **Split into two columns**
-

3. What is "Pivot Columns" used for?

It transforms **row values into column headers**. Useful for summarizing data (e.g., Products as column names with total sales per product).

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

In the **Applied Steps** pane, right-click the step → choose **Delete**

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

Feature	Description
Duplicate	Creates a copy of the query and its data
Reference	Creates a linked query based on the result of the original (lighter and dynamic)

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

- Import both files
 - Go to `Orders` query → **Home** → **Merge Queries**
 - Select `CustID` in both → Join type: **Inner**
-

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

- Select `Product` column → **Transform** → **Pivot Column**
 - Values column: `Quantity` → Aggregation: **Sum**
-

8. Append Orders_Jan.csv + Orders_Feb.csv:

- Import both queries
 - Go to **Home** → **Append Queries** → **Append as New**
 - Select both tables → Click OK
-

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

- Select `Email` column → **Transform** → **Fill** → **Down**
-

10. Extract the domain from the Email column:

- Select `Email` → **Transform** → **Extract** → **Text After Delimiter**
 - Delimiter: `@` → Result: `example.com`
-

11. M-code to merge queries dynamically using a JoinType parameter:

```
let
    JoinType = "InnerJoin", // Can be set as parameter
    Merged = Table.NestedJoin(Orders, {"CustID"}, Customers, {"CustID"},
    "NewTable", JoinKind.FromText(JoinType))
in
    Merged
```

Note: Use a function like `JoinKind.Inner`, `JoinKind.LeftOuter`, etc., depending on parameter value.

12. Unpivot "Jan_Sales", "Feb_Sales", etc. into "Month" and "Sales" format:

- Select all month columns → **Transform** → **Unpivot Columns**
 - Rename: `Attribute` → `Month`, `Value` → `Sales`
-

13. Handle errors in a custom column using try...otherwise:

```
= Table.AddColumn(Source, "SafeDivide", each try [Amount] / [Quantity]
otherwise null)
```

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

```
(phone as text) as text =>
Text.Select(phone, {"0".."9"})
```

Then apply this custom function to the Phone column.

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

- **Remove unused columns early** (step 1–2)
- **Filter rows** before merge/join operations
- **Disable query load** for intermediate queries
- **Consolidate multiple transformations** into fewer steps if possible
- Use **Table.Buffer()** before heavy joins to prevent re-evaluation