

2) What is the purpose of the "Applied Steps" pane in Power Query?

The "**Applied Steps**" pane in **Power Query** shows a list of all the **transformations** (changes) you've made to your data — **step by step** — in the exact order they were applied.

It's like a **history of your data cleaning process**.

Every time you do something — like **rename a column**, **filter rows**, or **change data type** — Power Query automatically records it as a new *step*.

2) How do you remove duplicate rows in Power Query?

1. Open Power Query

- In Power BI Desktop, go to **Home** → **Transform Data**.
- This opens the **Power Query Editor**.

2. Select the Table or Columns

- Click on the table you want to clean.
- Then select the **columns** where you want to check for duplicates.
 - If you select **multiple columns**, Power Query will remove duplicates **only if all selected column values are identical**.

3. Remove Duplicates

- Go to the **Home** tab.
- Click **Remove Rows** → **Remove Duplicates**.

Power Query will automatically:

- Keep the **first occurrence** of each duplicate.
- Delete all others.

3) What does the "Filter" icon do in Power Query?

The **Filter** icon in **Power Query** is used to **select which rows of data to keep or remove** based on certain conditions — just like filtering in Excel.

When you click the small **filter icon** next to a column header, Power Query shows a list of unique values in that column. You can:

- **Check** the values you want to keep, or
- **Uncheck** the values you want to remove.

4) How would you rename a column from "CustID" to "CustomerID"?

Using the Interface

1. In **Power Query Editor**, find the column named **CustID**.
2. **Right-click** the column header.

3. Choose **Rename**.
4. Type **CustomerID**, then press **Enter**.

That's it — the column is now renamed.

Using M Code

If you look in the **formula bar**, Power Query automatically adds this step:

```
#"Renamed Columns" = Table.RenameColumns(#"Previous Step", {{"CustID", "CustomerID"}})
```

5) What happens if you click "Close & Apply" in Power Query?

1. **Close** — Power Query Editor window closes, and you return to the main **Power BI Desktop** window.
2. **Apply** — All the transformations (Applied Steps) you made in Power Query — like filtering, renaming, or removing duplicates — are **applied** to your dataset in Power BI.

Power BI then:

- Loads the **cleaned and transformed data** into the **data model**.
- Updates all visuals, reports, and relationships using the new data.

6) Remove all rows where Quantity is less than 2.

Method 1 — Using the Interface

1. Open **Power Query Editor** (Home → Transform Data).
2. Click the **filter icon** next to the **Quantity** column.
3. Choose **Number Filters** → **Greater Than or Equal To...**
4. In the box, type **2**, then click **OK**.

Power Query will now only keep rows where **Quantity** ≥ 2 , removing all smaller values.

Method 2 — Using M Code

Power Query automatically generates this line in the formula bar:

```
#"Filtered Rows" = Table.SelectRows(#"Previous Step", each [Quantity] >= 2)
```

This means:

- Take the previous step's table
- Keep only rows where Quantity is **greater than or equal to 2**

10) How would you handle null values in the Price column?

Replace Null Values with a Default

If you want to keep those rows but give them a default value (for example, 0):

- Select the **Price** column.
- Go to **Transform** → **Replace Values**.
- In “Value to Find,” type **null**.
- In “Replace With,” type **0**.

This replaces all null prices with 0.

```
#"Replaced Nulls" = Table.ReplaceValue("#Previous Step", null, 0, Replacer.ReplaceValue, {"Price"})
```

Fill Missing Values (Up or Down)

If you want to fill nulls using nearby values (useful for time series):

- Select the **Price** column.
- Go to **Transform** → **Fill** → **Down** (or **Up**).

This copies the nearest non-null value above or below into the null cells.

11) Write custom M-code to add a column calculating $\text{TotalSpent} = \text{Quantity} * \text{Price}$.

15) Optimize the query to reduce refresh time

- **Select only needed columns immediately** (column pruning).
- **Filter rows as early as possible** (date filter, recent years, etc.).
- **Push filters and column selection to the source** (use native SQL or make sure query folding remains active).
- **Turn off loading for staging/intermediate queries** (Enable Load → off).
- **Disable automatic type detection** for unstructured sources and set data types after filtering/pruning.
- **Use incremental refresh** for very large tables (requires Premium / Premium Per User).
- **Use Query Diagnostics / View Native Query** to find and verify slow steps and folding.
- **Avoid Table.Buffer unless you know it helps** — it can increase memory usage.
- **Use the Folder connector properly** (combine binaries at source) if importing many files.