1. What is the purpose of the "Applied Steps" pane in Power Query?
It shows the sequence of transformations applied to the data (like a history of steps). Each step can be edited, reordered, or removed.

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

- Select the relevant columns → Right-click → **Remove Duplicates**.

3. What does the "Filter" icon do in Power Query?
It allows you to filter rows by values, conditions, or ranges (similar to Excel filters).

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

- Right-click the column header → **Rename** → type `CustomerID`

```
= Table.RenameColumns(#"Changed Type",{{"CustID", "CustomerID"}})
```

	CustomerID	Name	OrderDate	Product	Quantity	Price
1	101	Alice	10.01.2023	Laptop	1	1200
2	102	Bob	15.01.2023	Mouse	3	25
3	101	Alice	20.01.2023	Keyboard	2	80
4	103	Charlie	25.01.2023	Monitor	1	300

5. What happens if you click "Close & Apply" in Power Query?
The transformed data is **loaded into Power BI Desktop's data model** and Power Query closes.

6. Remove all rows where Quantity is less than 2.
Filter the Quantity column → **Number Filters → Greater Than or Equal To 2**.

```
= Table.SelectRows(#"Renamed Columns", each [Quantity] >= 2)
```

	CustomerID	Name	OrderDate	Product	Quantity	Price
1	102	Bob	15.01.2023	Mouse	3	25
2	101	Alice	20.01.2023	Keyboard	2	80

7. Split the OrderDate column into separate "Year," "Month," and "Day" columns.

- Select OrderDate → **Split Column → By Date → Year/Month/Day**.

```
Table.SplitColumn(Table.TransformColumnTypes(#"Filtered Rows", {{"OrderDate", type text}}, "ru-RU"), "OrderDate", Splitter.SplitTextByDelimiter(".", QuoteStyle.Csv), {"OrderDate.1", "OrderDate.2",
    "OrderDate.3"})
```

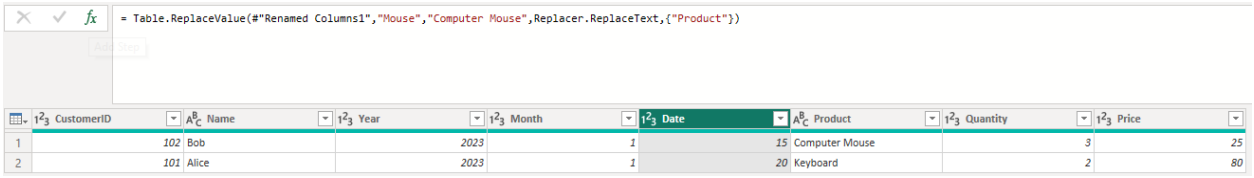
	CustomerID	Name	OrderDate.1	OrderDate.2	OrderDate.3	Product	Quantity	Price
1	102	Bob	15	01	2023	Mouse	3	25
2	101	Alice	20	01	2023	Keyboard	2	80

```
= Table.RenameColumns(#"Reordered Columns",{{"OrderDate.3", "Year"}, {"OrderDate.2", "Month"}, {"OrderDate.1", "Date"}})
```

	CustomerID	Name	Year	Month	Date	Product	Quantity	Price
1	102	Bob	2023	1	15	Mouse	3	25
2	101	Alice	2023	1	20	Keyboard	2	80

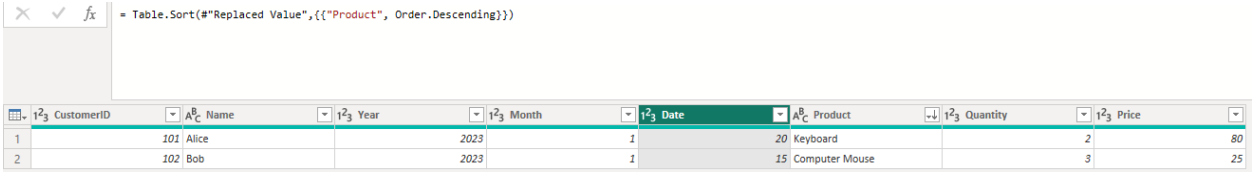
8. Replace all "Mouse" entries in the Product column with "Computer Mouse."

- Right-click Product column → **Replace Values** → find `"Mouse"`, replace with `"Computer Mouse"`.



9. Sort the table by OrderDate (newest first).

- Click the dropdown on OrderDate → **Sort Descending**.

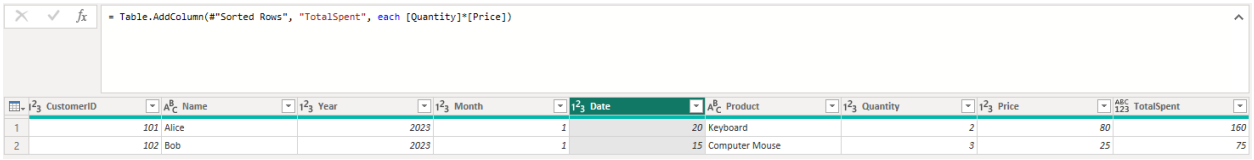


10. How would you handle null values in the Price column?
Options:

- Replace nulls with a default (e.g., 0).
- Remove rows containing nulls.
- Fill down/up if appropriate.

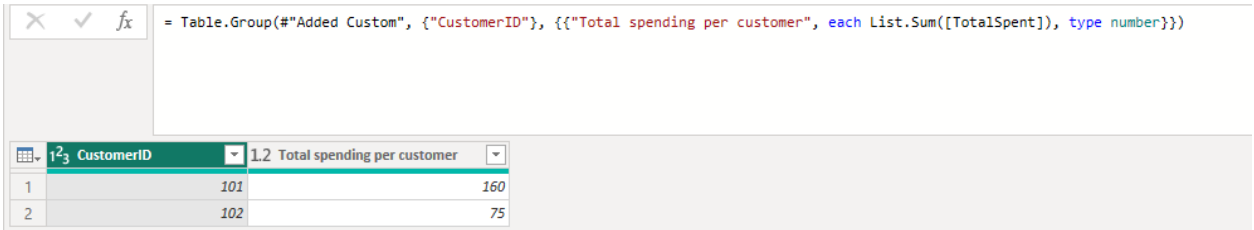
11. Write custom M-code to add a column calculating TotalSpent = Quantity * Price.

```
= Table.AddColumn(#"Sorted Rows", "TotalSpent", each [Quantity]*[Price])
```



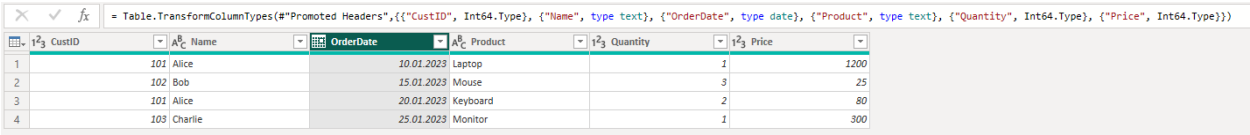
12. Group the table by CustID to show total spending per customer.

- Home → **Group By** → Group by CustID → Operation: Sum of TotalSpent.



13. Fix inconsistent date formats (e.g., 01/10/2023 vs. 2023-01-10) in OrderDate.

- Change data type of OrderDate column to **Date** → Power Query automatically standardizes formats.

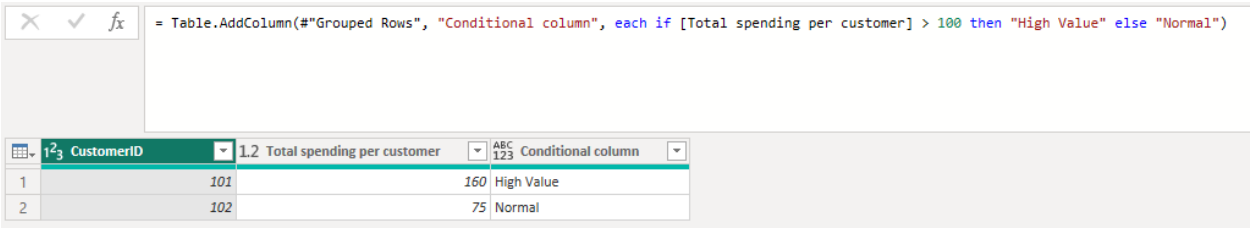


```
= Table.TransformColumnTypes(#"Promoted Headers",{{"CustID", Int64.Type}, {"Name", type text}, {"OrderDate", type date}, {"Product", type text}, {"Quantity", Int64.Type}, {"Price", Int64.Type}})
```

	CustID	Name	OrderDate	Product	Quantity	Price
1	101	Alice	10.01.2023	Laptop	1	1200
2	102	Bob	15.01.2023	Mouse	3	25
3	101	Alice	20.01.2023	Keyboard	2	80
4	103	Charlie	25.01.2023	Monitor	1	300

14. Create a conditional column: Label orders as "High Value" if Price > 100.

- Add Column → **Conditional Column** → If Price > 100 then `"High Value"` else `"Normal"`.



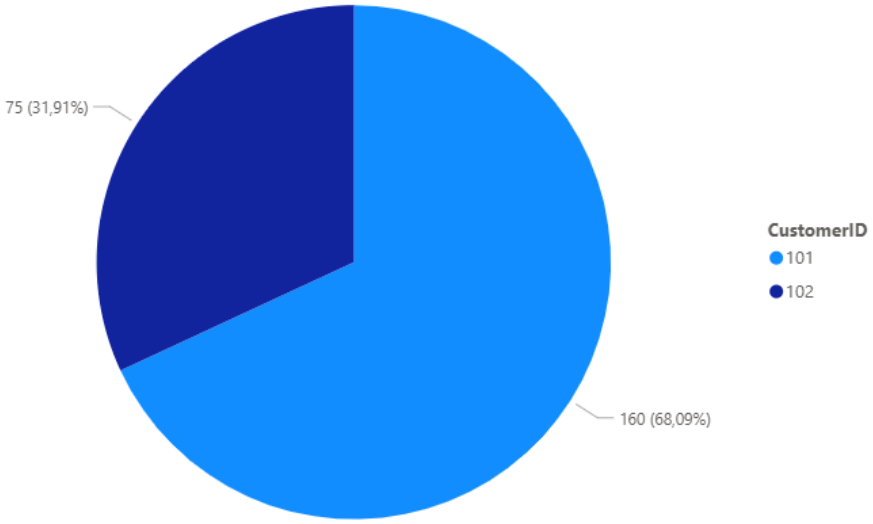
```
= Table.AddColumn(#"Grouped Rows", "Conditional column", each if [Total spending per customer] > 100 then "High Value" else "Normal")
```

	CustomerID	Total spending per customer	Conditional column
1	101	160	High Value
2	102	75	Normal

15. Optimize the query to reduce refresh time (e.g., remove unused columns early).

- **Remove unnecessary columns** as early as possible.
- Apply filters before joins/merges.
- Disable loading intermediate queries.
- Avoid steps that keep unneeded data in memory.

Total income by CustomerID



High Value

101
CustomerID

160,00
Sum of Total spending per customer

Normal

102
CustomerID

75,00
Sum of Total spending per customer