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

- **Merge**: Combines **columns** from two tables based on a matching key (like SQL JOIN).
- **Append**: Stacks **rows** of two or more tables together (like UNION in SQL).

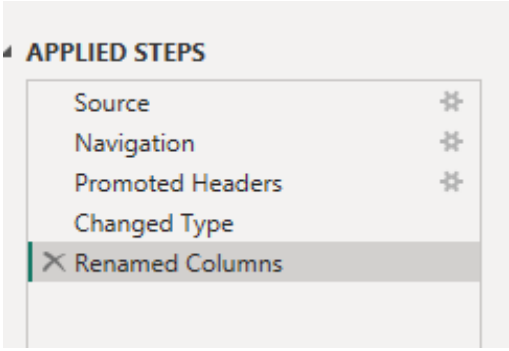
2. How do you split a "Full Name" column into "First Name" and "Last Name"?

- Select column → **Split Column → By Delimiter (space)** → into two columns.

3. What is "Pivot Columns" used for?
It turns **unique values in a column** into new column headers and aggregates corresponding values (like cross-tab).

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

- In **Applied Steps**, click the "X" next to the step, or right-click → **Delete**.

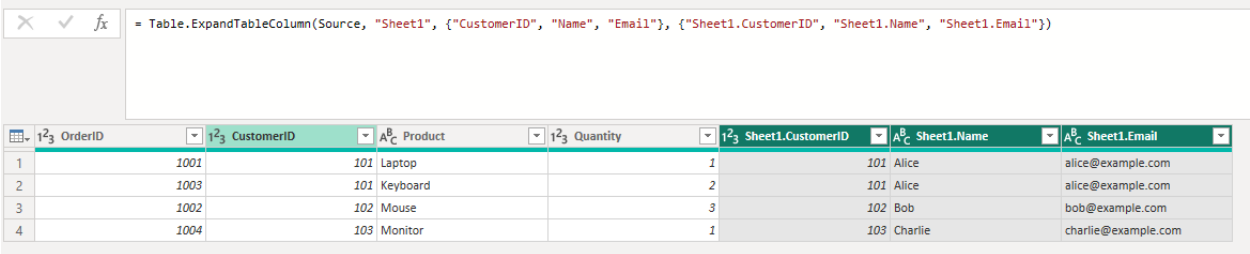


5. What is the purpose of "Reference" vs. "Duplicate" in queries?

- **Duplicate**: Creates a copy of the query including all steps (independent copy).
- **Reference**: Creates a new query that depends on the original query's output (linked)

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

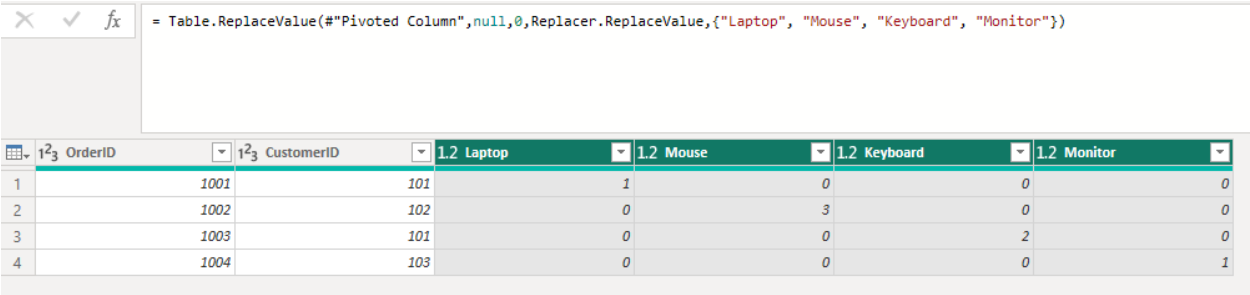
- **Home → Merge Queries** → Select *Orders.csv* and *Customers.xlsx* → Match on CustID → Join Kind = **Inner**.



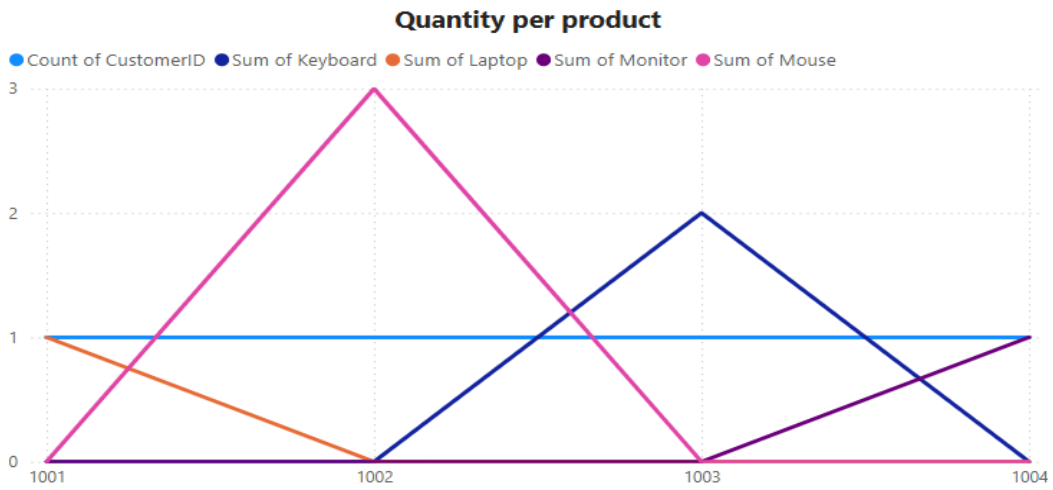
Product	Count of OrderID	Sum of Quantity	Count of Sheet1.CustomerID
Keyboard	**1**	**2**	**1**
101	**1**	**2**	**1**
alice@example.com	**1**	**2**	**1**
Alice	1	2	1
Laptop	**1**	**1**	**1**
101	**1**	**1**	**1**
alice@example.com	**1**	**1**	**1**
Alice	1	1	1
Monitor	**1**	**1**	**1**
103	**1**	**1**	**1**
charlie@example.com	**1**	**1**	**1**
Charlie	1	1	1
Mouse	**1**	**3**	**1**
102	**1**	**3**	**1**
bob@example.com	**1**	**3**	**1**
Bob	1	3	1
Total	**4**	**7**	**4**

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

- Select **Product** → Home → **Pivot Column** → Values column = Quantity → Aggregation = Sum.

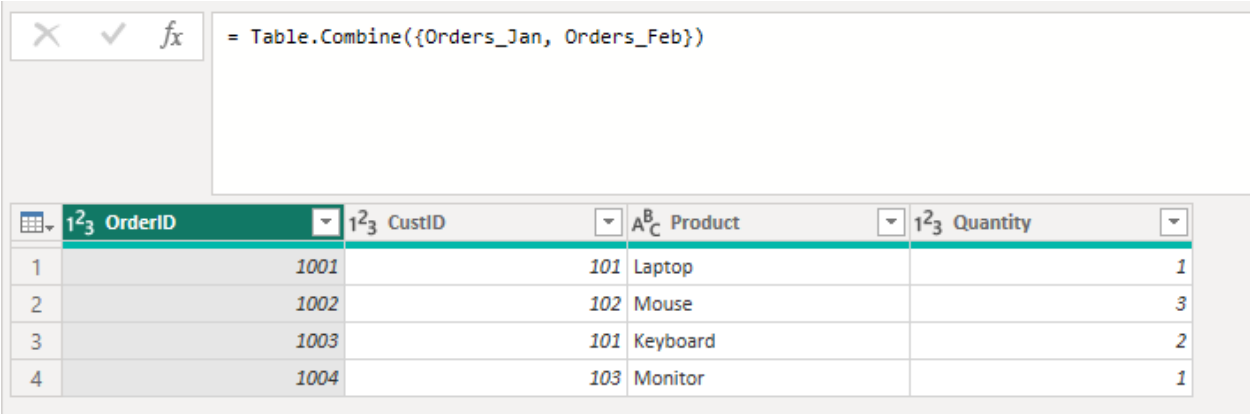


	OrderID	CustomerID	Laptop	Mouse	Keyboard	Monitor
1	1001	101	1	0	0	0
2	1002	102	0	3	0	0
3	1003	101	0	0	2	0
4	1004	103	0	0	0	1



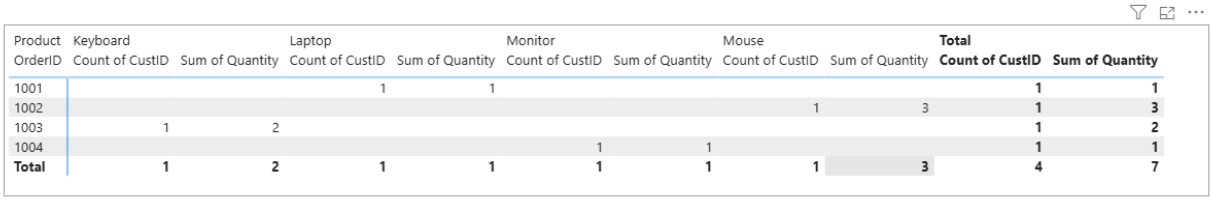
8. Append two tables with identical columns (e.g., Orders_Jan.csv + Orders_Feb.csv).

- Home → **Append Queries** → Select both tables → OK.



```
= Table.Combine({Orders_Jan, Orders_Feb})
```

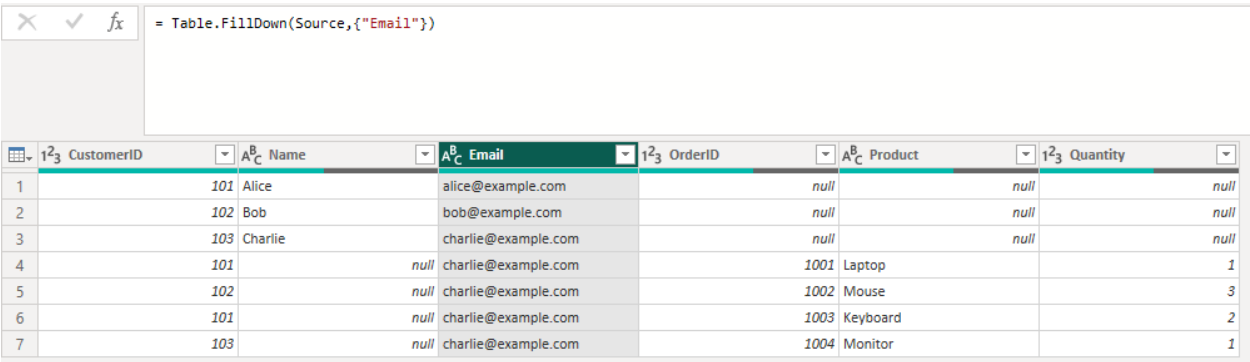
	OrderID	CustID	Product	Quantity
1	1001	101	Laptop	1
2	1002	102	Mouse	3
3	1003	101	Keyboard	2
4	1004	103	Monitor	1



Product	Keyboard		Laptop		Monitor		Mouse		Total	
OrderID	Count of CustID	Sum of Quantity	Count of CustID	Sum of Quantity	Count of CustID	Sum of Quantity	Count of CustID	Sum of Quantity	Count of CustID	Sum of Quantity
1001			1	1					1	1
1002							1	3	1	3
1003	1	2							1	2
1004					1	1			1	1
Total	**1**	**2**	**1**	**1**	**1**	**1**	**1**	**3**	**4**	**7**

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

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



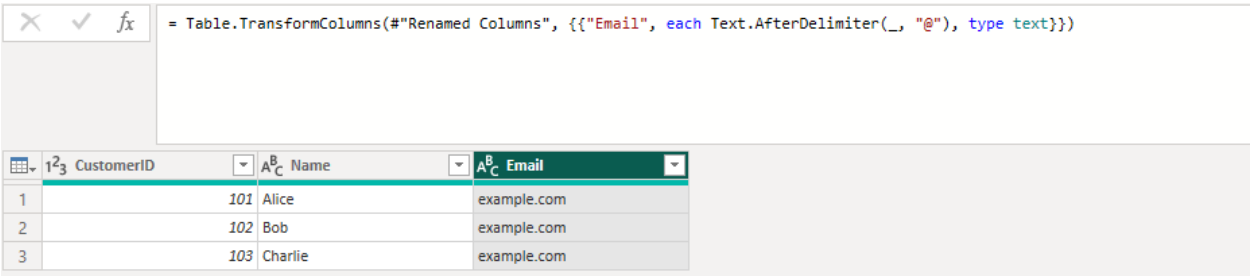
```
= Table.FillDown(Source,{"Email"})
```

	CustomerID	Name	Email	OrderID	Product	Quantity
1	101	Alice	alice@example.com	null	null	null
2	102	Bob	bob@example.com	null	null	null
3	103	Charlie	charlie@example.com	null	null	null
4	101	null	charlie@example.com	1001	Laptop	1
5	102	null	charlie@example.com	1002	Mouse	3
6	101	null	charlie@example.com	1003	Keyboard	2
7	103	null	charlie@example.com	1004	Monitor	1

Name			Total	
CustomerID	Count of OrderID	Sum of Quantity	**Count of OrderID**	**Sum of Quantity**
101	**2**	**3**	**2**	**3**
charlie@example.com	2	3	**2**	**3**
102	**1**	**3**	**1**	**3**
charlie@example.com	1	3	**1**	**3**
103	**1**	**1**	**1**	**1**
charlie@example.com	1	1	**1**	**1**
Total	**4**	**7**	**4**	**7**

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

- Transform → **Extract** → **Text After Delimiter "@"**.



```
= Table.TransformColumns(#"Renamed Columns", {{"Email", each Text.AfterDelimiter(_, "@"), type text}})
```

	CustomerID	Name	Email
1	101	Alice	example.com
2	102	Bob	example.com
3	103	Charlie	example.com

CustomerID	Email	Name
101	example.com	Alice
102	example.com	Bob
103	example.com	Charlie

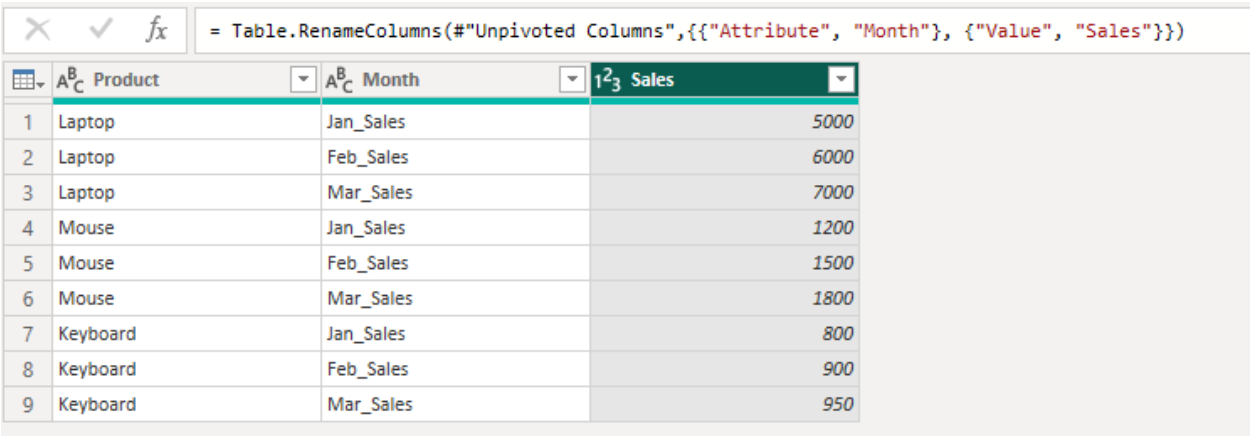
11. Write M-code to merge queries dynamically based on a parameter (e.g., JoinType = "Inner").

```
= Table.NestedJoin(Orders, {"CustID"}, Customers, {"CustID"}, "CustomerData", JoinType.Inner)
```

(Replace `JoinType.Inner` with a parameter variable, e.g. `JoinType`)

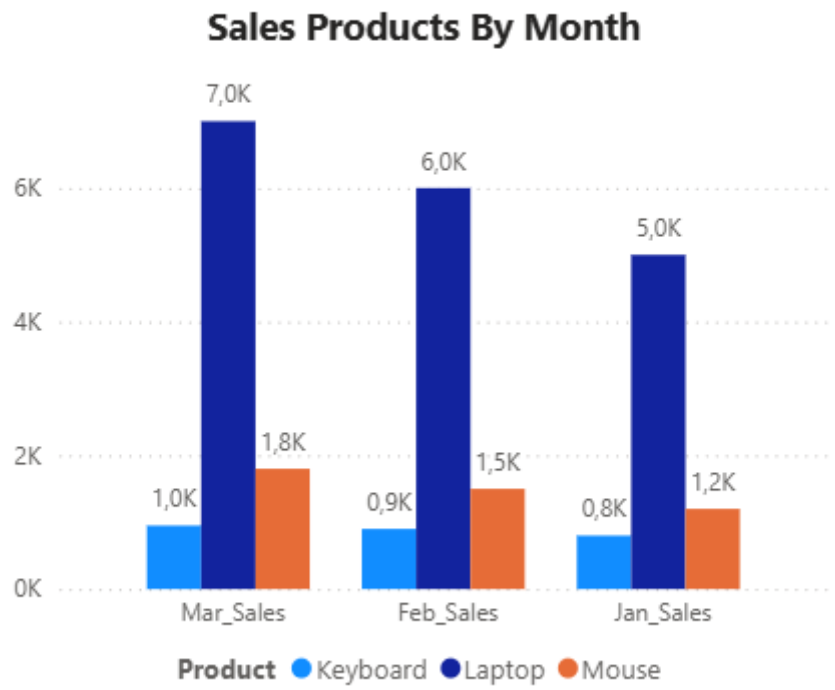
12. Unpivot a table with columns like "Jan_Sales," "Feb_Sales" into a "Month" and "Sales" format.

- Select those columns → **Unpivot Columns** → rename new attributes column to "Month" and values column to "Sales".



```
= Table.RenameColumns(#"Unpivoted Columns",{{"Attribute", "Month"}, {"Value", "Sales"}})
```

	Product	Month	Sales
1	Laptop	Jan_Sales	5000
2	Laptop	Feb_Sales	6000
3	Laptop	Mar_Sales	7000
4	Mouse	Jan_Sales	1200
5	Mouse	Feb_Sales	1500
6	Mouse	Mar_Sales	1800
7	Keyboard	Jan_Sales	800
8	Keyboard	Feb_Sales	900
9	Keyboard	Mar_Sales	950



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

```
= Table.AddColumn(Source, "SafeDivision", each try [Value1] / [Value2] otherwise null)
```

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

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

Usage: `= Table.TransformColumns(Source, {"Phone", each CleanPhone(_), type text})`

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

- Remove unused columns early.
- Filter rows before joins.
- Combine steps (e.g., multiple transformations into one).
- Disable query load for intermediate queries.
- Avoid unnecessary data type changes until the final step.