

# Lesson 5

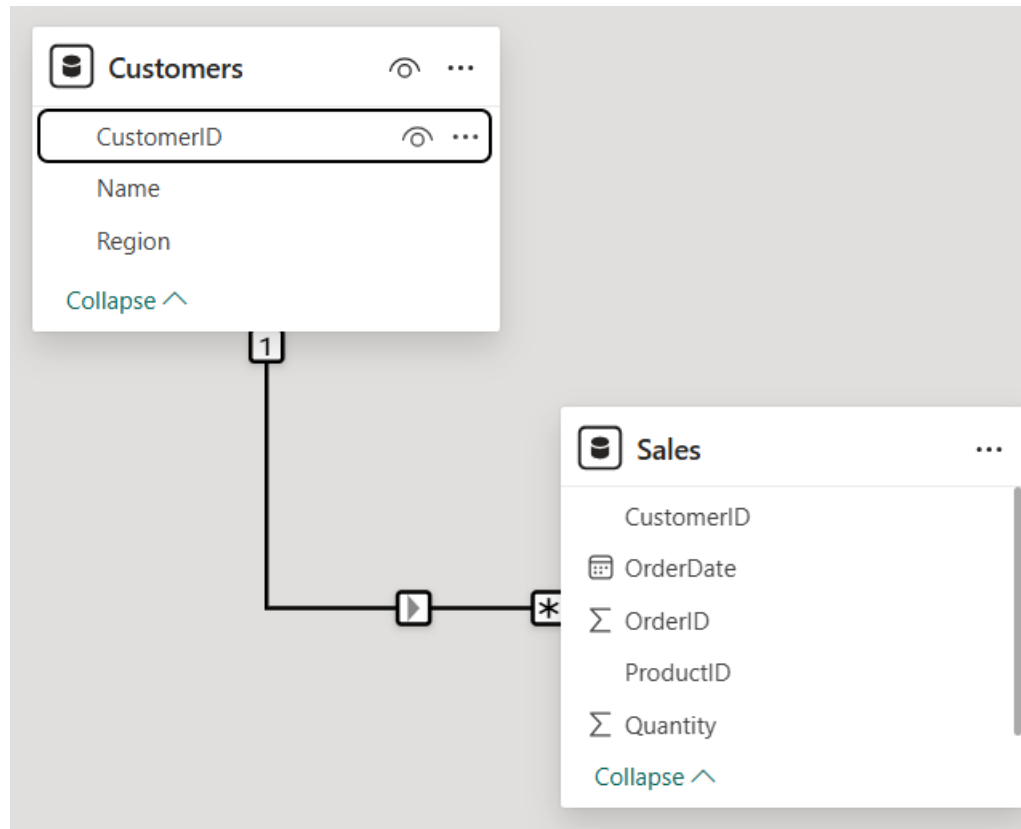
**Topic:** Data Modeling Basics

**Prerequisites:** Download customer.csv, product.csv, sales.csv

1. What is a primary key in a table?
  - A primary key is a column (or set of columns) that provides a unique identifier for every record, ensuring no two rows are the same
2. Name the two types of table relationships in Power BI.
  - one-to-many;
  - many-to-one relationships.
3. How do you create a relationship between two tables in Power BI?

Create relationship manually:

- On the Modeling tab, select Manage relationships > New. In the Create relationship dialog box, in the first table drop-down list, select a table. Select the column you want to use in the relationship. In the second table drop-down list, select the other table you want in the relationship.
4. What is a "star schema"?
    - A star schema is a data modeling method used in data warehouses that organizes data into a central fact table containing measurable business events and surrounding dimension tables that provide context
  5. Which table is typically the fact table in a sales dataset?
    - Fact table is "sales" table.
  6. Link Sales.csv to Customers.csv using CustomerID (one-to-many).
    - Usually, it creates automatically, but if not we can do it manually. On the Modeling tab select the column CustomerID from Customers table and drag it on CustomerID column on Sales table. On the "New relationship" window on "Cardinality" field select "one-to-many (1:\*)" from drop down list.



7. Why is ProductID in Sales.csv a foreign key?
  - The ProductID column in a sales table serves as a foreign key to link each sale to a specific product in the Products table, enforcing data integrity by ensuring that every sale is associated with a valid, existing product. This creates a logical relationship, preventing sales of non-existent products and allowing the database to maintain consistency and enable efficient data retrieval through joins. Fix a relationship error where ProductID has mismatched data types.
8. Fix a relationship error where ProductID has mismatched data types.
  - Change Data Types:

If the data types are different (e.g., one is "Text" and the other is "Whole Number"), select the ProductID column in one of the tables.

Click on the data type icon next to the column name (e.g., "ABC 123").

Choose the desired data type that matches the other ProductID column (e.g., "Whole Number" or "Text").

Confirm the change when prompted.

- **Error Handling:**  
If there are non-matching values (e.g., text in a column designated as numeric), Power Query will flag these as errors. We may need to address these errors by replacing or removing them, depending on the data cleaning strategy.

9. Explain why a star schema improves performance.

- A star schema improves performance by simplifying the data model, which reduces the number of complex joins needed for queries.

10. Add a new column TotalSales in Sales (Quantity \* Price from Products).

In Power BI Desktop, right-click on the table Sales in the Fields pane.

Select New column from the context menu. A formula bar will appear. On formula add this: "TotalSales = Sales[Quantity]\*RELATED(Products[Price])"

"

Press Enter or click the checkmark icon next to the formula bar to create the column.

✕

✓

1

TotalSales = Sales[Quantity]\*RELATED(Products[Price])

OrderID	CustomerID	ProductID	Quantity	OrderDate	TotalSales
1001	101	1	2	Thursday, January 5, 2023	2400
1002	102	2	5	Tuesday, January 10, 2023	125
1003	101	3	1	Sunday, January 15, 2023	80

11. Optimize a model with circular relationships—how would you resolve it?

- A circular relationship happens when relationships between tables create a loop so Power BI can't decide the correct filter path.

There are several approaches you can mention:

### **Break the loop**

- Remove one of the relationships causing the cycle.
- Decide which path is actually needed for reporting.

### **Use a bridge (lookup) table**

- Instead of connecting multiple tables directly, create a **bridge (dimension) table** that holds unique keys, and link other tables through it.

### Change cross-filter directions

- Circular relationships often appear when relationships are set to **bi-directional filtering**.
- Change one or more relationships to **single direction** to break the cycle.

### Use DAX instead of direct relationships

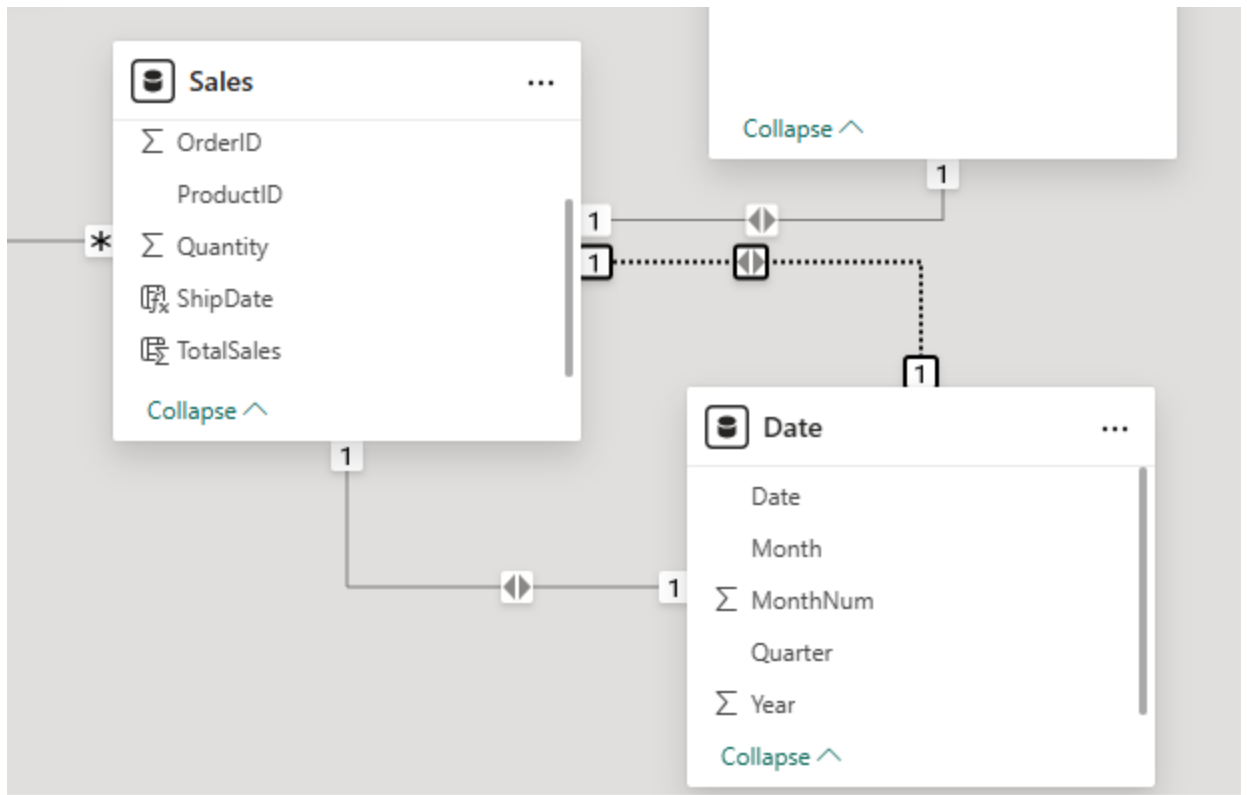
- Sometimes, instead of forcing a relationship, you can use DAX functions like `TREATAS()`, `LOOKUPVALUE()`, or `USERELATIONSHIP()` to bring values together only when needed.

### 12. Create a role-playing dimension for OrderDate and ShipDate.

- A role-playing dimension means the same dimension table is reused multiple times in different roles.
- I created column ShipDate on Sales table. Created **Date table**, and referenced it multiple times:
  - `Date (Order Date)`
  - `Date (Ship Date)`

### • relationships

- `Sales[OrderDate] → Date (Order Date) [Date]`
- `Sales[ShipDate] → Date (Ship Date) [Date]`



13. Handle a many-to-many relationship between Customers and Products.

- To handle many-to-many (M:N) relationships in Power BI, we typically create a bridge table that serves as an intermediary between two related tables, establishing two separate one-to-many (1:N) relationships. This approach avoids a direct M:N relationship and simplifies the model by using the bridge table as a shared dimension. Alternatively, We can use Power BI's native M:N support and bidirectional relationships, but this should be done with caution due to potential performance and ambiguity issues.

14. Use bidirectional filtering sparingly—when is it appropriate?

- Bidirectional filters in Power BI are appropriate for scenarios like many-to-many relationships, complex reporting needs, dimension-to-dimension analysis, and to sync slicers across tables, although they should be used cautiously due to potential negative impacts on performance and clarity. They are generally not needed for standard star schema models and should be avoided in favor of single-direction filters unless there's a clear, specific requirement.
- Use bidirectional relationships with caution, as they can lead to unpredictable results and reduced performance if not implemented correctly.

- For complex filtering needs, consider using DAX functions like CROSSFILTER() to establish temporary, case-by-case bidirectional relationships, which can be more efficient than an always-on bidirectional filter.
- 

15. Write DAX to enforce referential integrity if a CustomerID is deleted.

We can enforce referential integrity in DAX by:

- Using LOOKUPVALUE (fetch attributes):

Sales with CustomerName =

ADDCOLUMNS (

Sales,

"CustomerName",

LOOKUPVALUE ( Customers[CustomerName], Customers[CustomerID],  
Sales[CustomerID] )

- Or TREATAS (simulate a relationship in calculations):

Sales Amount by Customer =

CALCULATE (

SUM ( Sales[Amount] ),

TREATAS ( VALUES ( Customers[CustomerID] ), Sales[CustomerID] )

)

- And by writing validation measures (to find rows with no matching dimension):

Invalid Sales =

COUNTROWS (

FILTER (

Sales,

ISBLANK (

LOOKUPVALUE ( Customers[CustomerName], Customers[CustomerID],  
Sales[CustomerID] )

)

)

)