

Lesson 5

1. What is a primary key in a table?

A **primary key** is a column (or combination of columns) in a table that **uniquely identifies each row**. It must contain unique, non-null values.

2. Name the two types of table relationships in Power BI.

1. **One-to-Many (1:*)**: Most common; one record in Table A matches many in Table B.
 2. **Many-to-Many (:)**: Used when both tables can have multiple matching rows.
-

3. How do you create a relationship between two tables in Power BI?

- Go to **Model view**.
 - Drag a column (e.g., `CustomerID`) from one table to the matching column in the other.
 - Ensure the data types match.
-

4. What is a "star schema"?

A **star schema** is a data model where:

- A **central fact table** (e.g., `Sales`) contains measurable data.
 - It's surrounded by related **dimension tables** (e.g., `Customers`, `Products`, `Date`).
This design improves **performance and clarity**.
-

5. Which table is typically the fact table in a sales dataset?

The **Sales table**—it holds transactional data like:

- `CustomerID`
- `ProductID`
- `OrderDate`
- `Quantity`
- `TotalAmount`

6. Link Sales.csv to Customers.csv using CustomerID (one-to-many).

In **Model view**:

- Drag `CustomerID` from **Customers** (one side) to `CustomerID` in **Sales** (many side).
- This creates a *1: relationship**

7. Why is ProductID in Sales.csv a foreign key?

Because it **refers to** the `ProductID` in the **Products table**—which holds the detailed product info. It creates a relationship but doesn't uniquely identify Sales rows.

8. Fix a relationship error where ProductID has mismatched data types.

Go to **Power Query**:

- Ensure both `ProductID` columns have the **same data type** (e.g., both Text or both Whole Number).
- Use: `Transform` → `Data Type`

9. Explain why a star schema improves performance.

- Reduces **data duplication**.
 - Simplifies **DAX formulas**.
 - Allows **efficient indexing**.
 - Supports **faster queries** than a flat or snowflake schema.
-

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

Use a **DAX calculated column**:

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

This uses the **RELATED** function to fetch price from the **Products** table.

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

- **Avoid circular paths** by:
 - **Reevaluating table design**.
 - Using **inactive relationships** with `USERELATIONSHIP()` in DAX.
 - Creating **bridge tables** if necessary.
 - Power BI does **not allow circular relationships** to be active.
-

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

- Load the **Date table** once.
- Duplicate it in the model as:
 - `OrderDateTable`
 - `ShipDateTable`

- Relate each one to the relevant column in Sales.
 - Use them in visuals as needed.
-

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

- Create a **bridge table** (e.g., CustomerProductBridge) with unique combinations of CustomerID and ProductID.
 - Link it to both Customers and Products using one-to-many relationships.
 - Avoid direct many-to-many unless necessary.
-

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

Only when:

- Filters must **flow both ways** for a visual to work.
 - **Bridge tables** are used (e.g., for many-to-many).
 - Be cautious—it can introduce **performance issues** and **ambiguous paths**.
-

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

Use a measure or calculated column to check integrity:

```
IsCustomerValid = IF(
    ISBLANK(RELATED(Customers[CustomerName])),
    "Missing",
    "Valid"
)
```

This flags sales records that no longer match a customer.