

### 1. What is a primary key in a table?

A primary key is a **unique identifier** for each row in a table (e.g., `CustomerID` in Customer table, `ProductID` in Products table).

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

- **One-to-Many (1:\*)**
- **Many-to-Many (:)**

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

Go to **Model View** → **Manage Relationships** → **New** → **Select matching columns** (e.g., `CustomerID` in Sales and Customer).

### 4. What is a "star schema"?

A star schema is a data model where a **central fact table** (transactions) connects to multiple **dimension tables** (lookups like Customers, Products, Dates).

- **Sales (fact):** `OrderID`, `CustomerID`, `ProductID`, `DateID`, `Quantity`, `SalesAmount`
- **Customers (dimension):** `CustomerID`, `Name`, `Region`
- **Products (dimension):** `ProductID`, `ProductName`, `Category`
- **Dates (dimension):** `DateID`, `Year`, `Quarter`, `Month`
- **Regions (dimension):** `RegionID`, `RegionName`

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

The **Sales** table (contains transaction details: `OrderID`, `Quantity`, `ProductID`, etc.).

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

- Relationship: `Customer[CustomerID]` (1) → `Sales[CustomerID]` (\*).

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

Because it **references the primary key** in the Products table and ensures each sale links to a valid product.

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

- In Power Query, set both columns (`ProductID` in Sales, `ProductID` in Products) to the **same data type** (e.g., Whole Number).

### 9. Explain why a star schema improves performance.

- Reduces model complexity
- Speeds up DAX calculations
- Prevents circular relationships
- Easier for users to understand

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

Using DAX:

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

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

- Remove unnecessary relationships
- Create a **bridge table** if needed
- Redesign into a star schema instead of snowflake or loops

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

- Create a **Date table**.
- Link Date[Date] → Sales[OrderDate].
- Duplicate the Date table as "ShipDate" and link Date[Date] → Sales[ShipDate].

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

- Create a **bridge table** (e.g., CustomerProduct) listing valid combinations.
- Use relationships via the bridge instead of direct many-to-many.

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

- When both sides of a relationship need to filter each other (e.g., **security filtering** or **financial models**).
- Not recommended for large models (can cause performance issues).

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

```
Valid Sales =  
CALCULATE (  
    SUMX ( Sales, Sales[Quantity] * RELATED ( Products[Price] ) ),  
    FILTER ( Sales, NOT ISBLANK ( RELATED ( Customers[CustomerID] ) ) )  
)
```

This ensures Sales rows only count if they still have a matching Customer.