

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

A **primary key** is a column (or combination of columns) that uniquely identifies each row in a table. No duplicates, no nulls. Example: CustomerID in a Customers table.

---

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

- **One-to-Many (1:\*)** → most common, e.g., Customers → Sales.
  - **Many-to-Many (:)** → used when both sides can have multiple matches. *(There's also One-to-One, but in practice, 1: and : are the big two.)\**
- 

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

- Go to **Model View**.
  - Drag a field (e.g., CustomerID in Sales to CustomerID in Customers).
  - Or use **Manage Relationships** → **New** → **choose the columns**.
- 

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

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

- **Fact tables** in the center (e.g., Sales).
  - **Dimension tables** around it (e.g., Customers, Products, Date).  
It looks like a star and keeps relationships simple.
- 

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

The **Sales table** → it contains transactions (Quantity, Price, Date, ProductID, CustomerID).

---

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

- Customers[CustomerID] (primary key) → **one side**.
  - Sales[CustomerID] (foreign key) → **many side**.
- 

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

Because ProductID points to the **Products table** for details (Name, Price, Category). It references another table's primary key.

---

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

#### Steps to Fix Data Type Mismatch

##### 1. Check current types

- In **Model View**, hover over the columns.
- Example: Sales[ProductID] = *Text*, Products[ProductID] = *Whole Number*.

##### 2. Convert columns in Power Query

- Go to **Transform Data** → Power Query.
- Select the ProductID column in Sales.

- On the ribbon, choose **Data Type** → **Whole Number** (or Text, whichever matches Products).
- Do the same for Products[ProductID].

### 3. Apply changes

- Click **Close & Apply**.
- Now you can create the relationship without errors.



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

- Simpler relationships (no loops).
- Smaller fact table joins with slim dimension tables.
- Query folding works better, reducing refresh time.
- Avoids ambiguity and circular dependencies.



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

Since Price is in Products, use **DAX calculated column** in Sales:

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



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

- Remove unnecessary relationships.
- Use **bridge tables** or **dimension tables** instead of loops.
- Switch some relationships to **single-direction** filtering.



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

- Duplicate the Date table → Date (Order) and Date (Ship).
- Link each separately:
  - Sales[OrderDate] → Date (Order)[Date]
  - Sales[ShipDate] → Date (Ship)[Date].



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

- Create a **bridge table** (e.g., CustomerProducts) with unique CustomerID + ProductID pairs.
- Connect Customers → bridge → Products.
- Avoid direct many-to-many unless necessary.



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

- When you **need filters to flow both ways** (e.g., security roles, slicers affecting both fact and dimension tables).
- Avoid on large models (can cause performance issues or ambiguity).



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

For example, in a measure that checks invalid Sales rows:

InvalidSales =

```
COUNTROWS (
    FILTER (
        Sales,
        ISBLANK ( RELATED ( Customers[CustomerID] ) )
    )
)
```

This returns the number of Sales records without a matching Customer.