1. What is a primary key in a table?

A **primary key** is a unique identifier for each row in a table. It must:

- Contain **unique** values;
- **Not be null**;
- Be used to create **relationships** with other tables.

Example: CustomerID in the Customers table uniquely identifies each customer.

---

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

Power BI supports two main types:

1. **One-to-Many (1:*):** A single value in one table relates to multiple rows in another.
2. **Many-to-Many (:):** Both sides contain multiple matching values.

---

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

1. Go to the **Model view**;
2. Drag a field (e.g., CustomerID) from one table to another;
3. Confirm the **cardinality** (1:* or :);
4. Set **cross-filter direction** if needed;
5. Click **OK**.

---

4. What is a "star schema"?

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

- A central **fact table** (e.g., Sales) contains numeric data;
- Surrounding **dimension tables** (e.g., Customers, Products, Dates) provide context;
- All dimensions connect **directly** to the fact table.

Benefits:

- Simpler structure;
- Faster performance;
- Better for aggregations and filtering.

---

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

In a sales dataset, the fact table is typically **Sales.csv** because:

- It stores **transactions**;
- References other tables using foreign keys;
- Contains measurable values (e.g., quantity, total amount).

---

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

- Go to the **Model view**;
- Drag CustomerID from Customers to Sales;
- Ensure the direction is **one (Customers) → many (Sales)**.

---

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

Because ProductID in Sales:

- Refers to ProductID in the Products table;
- Isn't unique (a product appears in many sales);
- Creates a relationship between **facts** and **product attributes**.

---

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

1. Open **Power Query Editor**;
2. Check the data types of ProductID in both Sales and Products;
3. Make sure they match (e.g., both **whole number** or both **text**);
4. Apply changes (**Close & Apply**);
5. Recreate the relationship.

---

9. Explain why a star schema improves performance.

A star schema:

- Reduces complex joins;
- Minimizes relationship paths;
- Allows Power BI to process filters and measures more efficiently;

- Is easier to understand and maintain.

---

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

1. Ensure Sales and Products are related via ProductID;
2. In Sales, create a new calculated column:

DAX

CopyEdit

TotalSales =

Sales[Quantity] * RELATED(Products[Price])

---

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

Circular (cyclical) relationships cause errors and poor performance.

Fix strategies:

- **Remove one** of the relationships;
- Change relationship to **single-direction filtering**;
- Replace calculated columns with **measures**;
- Use **bridge tables** to break the cycle.

---

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

1. Load a Date table;
2. Duplicate it: OrderDateTable and ShipDateTable;
3. Relate:
    - Sales[OrderDate] to OrderDateTable[Date];
    - Sales[ShipDate] to ShipDateTable[Date];
4. Use the appropriate date table in your visuals.

---

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

Create a **bridge table**, such as CustomerProduct, with:

- CustomerID;

- ProductID.

Then:

- Link Customers → CustomerProduct;
- Link Products → CustomerProduct.

This allows filtering and analysis without ambiguity.

---

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

Use **bidirectional filtering** when:

- You have a many-to-many relationship;
- You need filters to work **both ways**;
- You're using a **bridge table** in the middle.

⚠️ Avoid it if:

- Performance is critical;
- It causes ambiguous relationships.

---

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

To detect **orphaned records** in Sales:

DAX

CopyEdit

```
OrphanedSales =
CALCULATE(
  COUNTROWS(Sales),
  ISBLANK(RELATED(Customers[CustomerID]))
)
```

Or to filter valid records:

DAX

CopyEdit

```
ValidSales =
FILTER(Sales, NOT ISBLANK(RELATED(Customers[CustomerID])))
```