
1. What is a primary key in a table?

A primary key is a column (or set of columns) in a table that uniquely identifies each row.

-

Example: In `Products`, `ProductID` is the primary key because each product has a unique ID.

-

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

-

One-to-many (1:*) – Most common (e.g., Customers → Sales)

-
-

Many-to-many (:) – Less common, requires careful handling.

-

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

-

Go to **Model View** → Drag the key from one table to the matching key in another table.

-
-

Or: **Manage Relationships** → **New** → Select columns and cardinality.

-

4. What is a "star schema"?

A star schema is a database design where a central **fact table** (with numeric, measurable data) links to surrounding **dimension tables** (descriptive data).

-

Example: `Sales` (fact) connects to `Products`, `Customers` (dimensions).

-

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

`Sales` is the fact table — it contains transactional details (Quantity, OrderDate, etc.).

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

-

`Customers.CustomerID (1) → Sales.CustomerID (*)`.

-

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

Because it references the unique `ProductID` in the `Products` table.

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

-

In Power Query or Data View: Convert both `ProductID` columns to the same type (e.g., **Whole Number**).

-

9. Explain why a star schema improves performance.

-

Reduces data duplication

-

-

Improves query speed

-

-

Simplifies DAX calculations

-
-

Works well with columnar storage in Power BI.

-

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

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

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

-

Remove unnecessary relationships

-
-

Create intermediate lookup tables

-
-

Use **inactive relationships** and `USERELATIONSHIP()` in DAX.

-

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

-

Duplicate the Date table twice → Name them `OrderDateTable` and `ShipDateTable`

-
-

Create separate relationships for each date column.

-

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

-

Create a **bridge table** with unique combinations of CustomerID and ProductID

-
-

Relate Customers → Bridge → Products.

-

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

-

When analysis requires filter context to flow both ways (e.g., in a many-to-many scenario)

-
-

But avoid for large models to prevent performance issues.

-

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

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

This ensures only sales linked to valid customers are counted.
