

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. It ensures there are no duplicate rows and is used to relate data across tables.

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 → drag a field (e.g., CustomerID) from one table and drop it onto the matching field in another table → set relationship cardinality and cross-filter direction → click OK.

4. What is a "star schema"?

A star schema is a data modeling design where:

One central fact table contains transactional data (e.g., sales),

Surrounded by multiple dimension tables (e.g., customer, product),

Relationships are typically one-to-many from dimensions to fact.

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

Sales.csv is the fact table. It contains measurable business transactions like OrderID, Quantity, CustomerID, and ProductID. It links to dimensions (Customers, Products) via foreign keys.

6. Link Sales.csv to Customers.csv using CustomerID (one-to-many). In Power BI Model View:

Drag CustomerID from Customers to Sales.

This forms a one-to-many (1:*) relationship:

One customer → Many sales records

Ensure:

Cardinality: One-to-many

Cross filter direction: Single

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

Because Sales.csv[ProductID] references Products.csv[ProductID].

This: Links each sale to the product sold. Enables lookup of product details like ProductName, Category, and Price

8. Fix a relationship error where ProductID has mismatched data types. ProductID in both tables is numeric (integer).
9. Explain why a star schema improves performance.

Fact Table: Sales

Dimension Tables: Customers, Products

Benefits:

- Simpler joins (1:* relationships only)
- Faster aggregations in Power BI
- Better columnar compression and indexing
- Easier DAX writing and visualization logic

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

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

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

Avoiding bidirectional filters unless necessary

Removing redundant relationships

Using DAX instead of a relationship,

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

1. Create a **Date table** (either manually or using CALENDAR):

dax

Date = CALENDAR(DATE(2023,1,1), DATE(2023,12,31))

2. Duplicate the table:

- One for OrderDate
- One for ShipDate

3. Link:

- Sales[OrderDate] → Date[Date]
- Sales[ShipDate] → ShipDateTable[Date]

4. Use different slicers to control visuals based on either.

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

Products

In sales data, if:

Customers can buy many products

Products can be bought by many customers

To model this:

We should use the Sales table as a bridge table:

Connect Customers → Sales ← Products

Both relationships are one-to-many

In visuals:

Sales to filter both ends

Avoiding direct many-to-many links

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

Use only when:

When we need dynamic slicers that affect both sides (e.g., sales filtered by both customer and product context)

In many-to-many relationships where filters must flow both ways

In row-level security (RLS) models

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

IsValidCustomer =

```
IF(
    ISBLANK(RELATED(Customers[CustomerID])),
    "Invalid",
    "Valid"
);
```