1. What is a primary key in a table?

A **primary key** is a column (or set of columns) with **unique and non-null values** that uniquely identify each row in a 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**, drag the key from one table (e.g., CustomerID in Sales) to the matching column in another (e.g., CustomerID in Customer).

---

4. What is a "star schema"?

A **star schema** is a data model where a central **fact table** links to multiple **dimension tables** using one-to-many relationships.

---

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

The **Sales** table is the **fact table**, containing transactional data (e.g., Quantity, OrderDate).

---

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

In **Model view**:

- Drag CustomerID from **Sales** to **Customer**
- Ensure:
  - Sales → many side
  - Customer → one side

---

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

Because it **refers to a unique ProductID** in the Products table and is used to **link** the two tables.

---

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

Open **Power Query** > convert both columns (ProductID in Sales and Products) to the same data type (e.g., Whole Number).

---

9. Explain why a star schema improves performance.

It simplifies relationships and **reduces joins**, making **query processing faster and more efficient**, especially for DAX calculations.

---

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

Create a **calculated column** in Power BI using DAX:

dax

КопироватьРедактировать

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

---

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

- Break the loop by **removing one relationship**,
- Or change **cross-filter direction** from bidirectional to **single**,
- Consider using **DAX functions** instead of physical relationships.

---

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

1. Duplicate the **Date** table twice (e.g., OrderDateTable, ShipDateTable)
2. Relate OrderDateTable[Date] to Sales[OrderDate] and ShipDateTable[Date] to Sales[ShipDate]

---

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

Use a **bridge table** containing unique CustomerID-ProductID combinations, then link it to both Customer and Product tables.

---

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

When you **need filtering to flow in both directions**, such as in **many-to-many relationships** or **when slicers affect multiple tables**.

---

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

You can create a measure to detect invalid CustomerIDs:

dax

КопироватьРедактировать

```
InvalidCustomerCheck =
CALCULATE(
    COUNTROWS(Sales),
    ISBLANK(RELATED(Customer[Name]))
)
```

This will show count of Sales rows where CustomerID doesn't match any row in Customer.