---

**1. What is a primary key in a table?**
A **primary key** is a unique identifier for each row in a table. No two rows can have the same primary key value, and it cannot be null.

---

**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 (like `CustomerID`) from one table to the matching field in another table, or use **Manage Relationships** to define it manually.

---

**4. What is a "star schema"?**
A **star schema** is a data model design where a central **fact table** connects to multiple **dimension tables**, forming a star-like structure. Each dimension is only joined to the fact table.

---

**5. Which table is typically the fact table in a sales dataset?**
The **Sales** table, as it contains transactional data (quantities, totals, etc.).

---

**6. Link Sales.csv to Customers.csv using CustomerID (one-to-many).**
Ensure `CustomerID` is a **primary key** in `Customers.csv` and a **foreign key** in `Sales.csv`, then create a **one-to-many** relationship in Power BI.

---

**7. Why is ProductID in Sales.csv a foreign key?**
Because it **references** the primary key `ProductID` in the **Products** table to identify which product was sold.

---

**8. Fix a relationship error where ProductID has mismatched data types.**
Go to **Power Query Editor**, select both columns, and convert them to the **same data type** (e.g., both to *Whole Number* or *Text*).

**9. Explain why a star schema improves performance.**
It simplifies queries and enhances **read performance** because relationships are straightforward and **indexes** work more efficiently.

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

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

**11. Optimize a model with circular relationships—how would you resolve it?**
Break the cycle by:

- Removing or disabling one relationship
- Using **DAX measures** or **TREATAS()** to connect indirectly
- Avoiding bi-directional filters unless truly needed

**12. Create a role-playing dimension for OrderDate and ShipDate.**
Duplicate the Date table (e.g., `Date_Order`, `Date_Ship`), then relate each to `Sales[OrderDate]` and `Sales[ShipDate]` separately. Use **USERELATIONSHIP** in measures to switch.

**13. Handle a many-to-many relationship between Customers and Products.**
Create a **bridge table** (e.g., `CustomerProduct`) listing combinations, and use it to connect both original tables via **one-to-many** relationships.

**14. Use bidirectional filtering sparingly—when is it appropriate?**
Only when **both tables need to filter each other**, such as in **complex reporting scenarios** (e.g., financials) where filters must flow in both directions.

**15. Write DAX to enforce referential integrity if a CustomerID is deleted.**
Create a calculated column or measure to detect orphan records:

```
CustomerExists = IF(COUNTROWS(RELATEDTABLE(Customers)) = 0, "Missing", "Valid")
```