1. What is a primary key in a table?

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

- Example: `CustomerID` in a `Customers` table — no duplicates, no nulls.

---

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

1. **One-to-Many** (`1:*`) – Most common
2. **Many-to-Many** (`*:*`) – Supported with caution

---

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

- Go to **Model view**
- Drag and drop a column (e.g., `CustomerID`) from one table to its match in another
- Or use: **Manage Relationships > New**

---

4. What is a "star schema"?

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

- A central **Fact table** (e.g., Sales) stores transactional data
- Surrounding **Dimension tables** (e.g., Products, Customers, Dates) provide context
- Relationships are **one-to-many** from dimensions to fact

---

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

- The **Sales** table is the fact table.
    - It contains measurable data like `Quantity`, `Revenue`, `ProductID`, etc.

---

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

- `CustomerID` is the **primary key** in `Customers.csv`
- `CustomerID` is a **foreign key** in `Sales.csv`
- Create a **one-to-many** relationship from `Customers` (1) → `Sales` (*)

---

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

Because it points to `ProductID` in the **Products** table.

- It connects each sale to product details (name, price)
- It's not unique in `Sales`, but **must match** an entry in `Products`

---

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

- Open **Power Query**
- Ensure both columns are the same type (e.g., Text or Whole Number):

  ```
  Table.TransformColumnTypes(Source, {{"ProductID", type text}})
  ```

- Reload and recreate the relationship

---

9. Explain why a star schema improves performance

- **Simplifies joins** (1:* relationships)
- Reduces memory usage
- Encourages reusable dimensions
- Optimizes DAX calculations and indexing

---

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

Use **RELATED** in DAX:

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

- `RELATED` pulls the price from the Products table into the Sales context.

---

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

Circular relationships are not allowed. Fix by:

- **Removing unnecessary relationships**
- **Using DAX functions** (e.g., `LOOKUPVALUE`, `TREATAS`) instead of a physical join
- **Creating bridge tables** to isolate connections

---

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

Steps:

1. Duplicate your `Date` table: rename as `Date_Order`, `Date_Ship`
2. Relate `Date_Order[Date]` to `Sales[OrderDate]`, and `Date_Ship[Date]` to `Sales[ShipDate]`
3. Use USERELATIONSHIP in DAX to switch context:

```
Total by Ship Date = CALCULATE([TotalSales], USERELATIONSHIP(Sales[ShipDate],
Date_Ship[Date]))
```

---

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

Example: customers purchase **multiple products**, and each product is purchased by **multiple customers**.
Fix:

- Create a **bridge (fact) table** like `Sales` containing `CustomerID`, `ProductID`
- Remove direct many-to-many
- Create *1: relationships** from `Customers → Sales`, and `Products → Sales`

---

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

Use when:

- You need **cross-filtering** in **both directions** (e.g., slicers affecting both sides)
- Common in **many-to-many** or **composite models**
  But it can **increase model complexity and calculation errors**, so use only if necessary.

---

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

This checks for orphaned `CustomerID`s in Sales:

```
OrphanedSales =
CALCULATE(COUNTROWS(Sales),
    NOT Sales[CustomerID] IN VALUES(Customers[CustomerID])
)
```

Or return blank if there's no matching customer:

```
SafeTotalSales =
IF(
    CONTAINS(Customers, Customers[CustomerID], Sales[CustomerID]),
    Sales[Quantity] * RELATED(Products[Price]),
    BLANK()
)
```