1. What is a primary key in a table?

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

- Example: In `Customers.csv`, `CustomerID` is unique → serves as the primary key.
  ☑ No duplicates, no nulls.

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

Power BI supports:

1. **One-to-many (1:*):** Most common — one customer can have many sales.
2. **Many-to-many (*:*):** When both sides have repeating values (e.g., customers and products through purchases).
   (Also one-to-one exists but is rare.)

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

- Go to **Model View**
- Drag the key field (e.g., `CustomerID`) from one table to the other
- Or use: **Home → Manage Relationships → New → Select columns**

🌟 4. What is a "star schema"?

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

- A central **fact table** (e.g., Sales) contains **measures and foreign keys**.
- It connects to multiple **dimension tables** (e.g., Customer, Product, Date).

Why "star"? Because it visually resembles a star — one fact table in the center, surrounded by dimensions.

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

☑ **Sales.csv**

- It contains **transactional data** (Quantity, Price, IDs, etc.)
- Other tables (Customer, Product, Date) describe attributes — they are **dimension tables**.

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

Steps:

- Model view → Drag `CustomerID` in `Customers` → `CustomerID` in `Sales`
- Cardinality: **One-to-many (1:*).**
- Cross filter: **Single direction (from Customers → Sales)**

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

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

- It does not uniquely identify records in `Sales`.
- It links each sale to its **related product** in the Products table.

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

If `ProductID` in one table is **text** and in another **number**, Power BI cannot link them.

Fix:

- Go to **Data View**
- Select both tables → Change `ProductID` column's **Data Type** to match (e.g., *Whole Number* in both).
  Then re-establish the relationship.

🚀 9. Why a star schema improves performance

☑ Reasons:

- Reduces redundant joins (compared to snowflake or flat models).
- Optimizes **query folding** and DAX performance.
- Simplifies relationships (one-to-many only).
- Improves **VertiPaq compression** in Power BI (fewer unique keys).

In short: **simpler model = faster performance**.

💰 10. Add a new column TotalSales = Quantity × Price

You can do this in Power BI using **DAX** or **Power Query**.

DAX (in Data View):

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

👉 Uses `RELATED()` to pull price from the Products table via relationship.

🔁 11. Optimize a model with circular relationships

Circular relationships (loops) confuse Power BI's filter propagation.

Solutions:

1. **Remove one relationship** if redundant.
2. **Use inactive relationship** (set to inactive, activate via `USERELATIONSHIP` in DAX).
3. **Split a dimension** into two role-playing versions (e.g., `Date_Ordered`, `Date_Shipped`).
4. **Flatten snowflake dimensions** (combine related dimension tables).

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

Duplicate the `Date` table:

- `Date_Ordered` → connected to `Sales[OrderDate]`
- `Date_Shipped` → connected to `Sales[ShipDate]`

Power BI can't use one Date table for both roles simultaneously.

Alternative (in DAX):
Use `USERELATIONSHIP()` when needed:

```
ShipDate Sales =
CALCULATE(
    SUM(Sales[TotalSales]),
    USERELATIONSHIP(Sales[ShipDate], Dates[Date])
)
```

🔁 13. Handle many-to-many relationship (Customers ↔ Products)

If customers can purchase multiple products and products are purchased by multiple customers → many-to-many.

Fix:

- Create a **bridge table** (e.g., Sales) that contains unique combinations of `CustomerID` and `ProductID`.
- Then connect both sides to it using **one-to-many** relationships.

☑ Avoid direct many-to-many links — they can create ambiguity.

🔁 14. Use bidirectional filtering sparingly — when appropriate

Default: Filters should flow **from dimension → fact** (single direction).
Use **bidirectional** only when:

- You have **shared dimension tables** (e.g., a common Date or Customer table across multiple facts).
- You need a **dynamic measure filter context** from both sides.

⚠️ Overusing it can cause **ambiguous relationships** and **slow performance**.

🧠 15. DAX to enforce referential integrity (CustomerID deleted)

If a CustomerID in Sales no longer exists in Customers → catch that case:

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

Or to **flag orphan records:**

```
IsOrphan =
IF(
    ISBLANK(RELATED(Customers[CustomerID])),
    "Missing Customer",
    "Valid"
)
```