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.

- Example: `CustomerID` in a Customers table.
- It ensures no duplicates or nulls.

◇ 2. Two types of table relationships in Power BI

1. **One-to-Many (1:*):** Most common (e.g., one Customer has many Sales).
2. **Many-to-Many (*:*):** When both sides have duplicates (e.g., multiple customers linked to multiple products).

(*There's also One-to-One (1:1), but rarely used.*)

◇ 3. How to create a relationship between two tables

1. In Power BI → go to **Model view**.
2. Drag `CustomerID` from `Sales` → to `CustomerID` in `Customers`.
3. Or use **Manage Relationships → New → select tables + columns**.

◇ 4. What is a Star Schema?

A **star schema** is a model with:

- **Fact table** (transactions: Sales, Orders).
- **Dimension tables** (lookup tables: Customers, Products, Dates).

Looks like a star ★ because dimensions radiate around the central fact table.

◇ 5. Which is typically the Fact Table in a sales dataset?

- **Sales (transactions)** table → contains Quantity, Amount, ProductID, CustomerID, DateID.

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

- `Customers[CustomerID]` = **Primary Key** (unique).
- `Sales[CustomerID]` = **Foreign Key** (repeats).
  Relationship: **One-to-Many (1:*).**

◇ 7. Why is ProductID in Sales.csv a Foreign Key?

Because it **refers to another table (Products)**, not defined in Sales itself.

- Sales only stores the reference (`ProductID`), while full product details live in the Products table.

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

- Example: `Products[ProductID]` is **Text**, while `Sales[ProductID]` is **Whole Number**.
- Fix → In **Power Query**, set both columns to the same type (usually **Whole Number**).

◇ 9. Why a Star Schema improves performance

- Reduces redundancy → smaller data model.
- Simplifies relationships → fewer joins.
- Improves DAX speed (optimized for VertiPaq engine).
- Easier to understand (clean model design).

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

You need a relationship `Sales[ProductID] → Products[ProductID]`.

DAX calculated column in Sales:

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

◇ 11. Optimize a model with circular relationships

- Circular = loops between tables (A → B → C → A).
- Fix:
  - Break the loop by removing redundant relationships.
  - Create a **bridge table** if needed.
  - Use **single-direction filters**.

◇ 12. Create a Role-Playing Dimension for OrderDate and ShipDate

- Load **Date** table once.
- Duplicate it as `OrderDate` and `ShipDate`.
- Create two relationships:
  - `Sales[OrderDate] → Date[Date].`
  - `Sales[ShipDate] → Date[Date].`
- This way, you can slice Sales by OrderDate or ShipDate.

◇ 13. Handle a Many-to-Many relationship between Customers and Products

- Create a **bridge table** with unique combinations (CustomerID, ProductID).
- Use it to connect Customers ↔ Bridge ↔ Products.

◇ 14. Use Bidirectional Filtering sparingly

- Only when you need filters to flow **both ways**.
- Example: Customers ↔ Sales ↔ Regions (to filter Customers by Regions and vice versa).
- Risk: can cause **performance issues** or wrong results with many-to-many.

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

Example: Detect Sales without matching Customers.

```
InvalidSales =
CALCULATE(
    COUNTROWS(Sales),
    ISBLANK(RELATED(Customers[CustomerID]))
)
```

- If result > 0 → means some Sales rows reference missing Customers.