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

**Done**

**What is a primary key in a table?**

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

**One-to-Many (1:\*):**

* This is the **most common** type of relationship.
* One record in the first table (like a Customer) can relate to **many records** in the second table (like Orders).
* Example:
* **Customers** (1) → **Orders** (many)

**Many-to-Many (*:*):**

* Used when both tables can have **multiple matching records** in each.
* Often appears when you don’t have a unique key on either side.
* Example:
* **Students** ↔ **Courses** (a student can take many courses, and a course can have many students).

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

**Open your Power BI file** → Go to the **Model view** (the icon that looks like a table with links).

You’ll see all your tables displayed as boxes (with their fields inside).

**Click and drag** the column from one table (for example, CustomerID in **Customers**)

 → **drop it** on the matching column in the other table (for example, CustomerID in **Orders**).

Power BI will automatically open the **"Create Relationship"** dialog box.

Check these settings:

* **Column:** Correct fields are selected (e.g., CustomerID ↔ CustomerID)
* **Cardinality:** Choose *One-to-Many (1:*)\* or *Many-to-Many (*:*)*
* **Cross filter direction:** Usually set to *Single*

Click **OK**

**What is a "star schema"?**

A **star schema** connects **one central fact table** (which holds numeric data, like sales or revenue)

 to several **dimension tables** (which hold descriptive details, like customer, product, or date).

**Which table is typically the fact table in a sales dataset?**

In a sales dataset, the **Sales (or Orders)** table is the **fact table** because it stores the core transactional data — quantities, prices, and totals.

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

done

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

In the **Products** table, each product has a **unique ProductID** — this is the **primary key**.

In the **Sales** table, the same **ProductID** appears many times — one for each sale that includes that product — so it becomes a **foreign key**.

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

**Fix the data types in Power Query**

Step 1: Open Power Query

* Go to **Home → Transform Data**

Step 2: Select each table

* Select the **Sales** table
* Find the **ProductID** column

Step 3: Change its data type

* Click the small **data type icon** (ABC / 123) next to the column name
* Choose **Text** (if ProductID contains letters like “P01”)
* or **Whole Number** (if it’s numeric, like 101)

Do the **same** in the **Products** table — make sure both use **the same type**.

Step 4: Apply the change

* Click **Close & Apply** (top left)

Step 5: Recreate the relationship

* Go to **Model view** → drag ProductID from **Products** → onto ProductID in **Sales**
* It should now connect successfully

**Explain why a star schema improves performance.**

A **star schema** improves performance because it:

* Reduces redundant data
* Minimizes joins between tables
* Compresses data better in VertiPaq
* Simplifies relationships for faster query execution

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

done

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

A **circular relationship** happens when tables form a **loop** — meaning Power BI can’t determine a clear, single path between tables.

Customers → Orders → Products → Customers

**Break the loop**

The simplest and most important step:

 Go to **Model view**, find the circular path, and **delete one of the relationships** that’s creating the loop.

**Create a Bridge (Helper) Table**

If you need to connect tables that share common columns (like IDs or categories),

 create a **bridge table** instead of linking them directly.

**Example:**

Instead of linking Sales ↔ Returns ↔ Products directly,

 create a small **ProductKey** bridge table:

**2. Use “Single” cross-filter direction**

If relationships are set to **Both**, they can cause loops.

 Change them to **Single**:

* Go to **Model view → select the relationship line → Properties pane**
* Change **Cross filter direction** to **Single**

 This keeps filters flowing in **one direction**, preventing loops.

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

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

**Use bidirectional filtering sparingly—when is it appropriate?**

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