CLEAR, CONCRETE EXAMPLES USING **PRODUCTS, CATEGORIES, AND SALES**, and you specifically want to see **multiple (composite) primary keys** plus foreign keys. Let's walk through it cleanly like you'd see in a real database

---

1. Categories table (simple primary key)

Each category is unique.

```
CREATE TABLE Categories (
    CategoryID INT PRIMARY KEY,
    CategoryName VARCHAR(100)
);
```

Primary Key

- `CategoryID`

---

2. Products table (foreign key + primary key)

Each product belongs to one category.

```
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    CategoryID INT,
    Price DECIMAL(10,2),
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);
```

Primary Key

- `ProductID`

Foreign Key

- `CategoryID → Categories(CategoryID)`

---

3. Sales table (COMPOSITE primary key)

Here's where multiple primary keys come in.

A sale is uniquely identified by:

- `SaleID`
- `ProductID`

Together, they form a **composite primary key**.

```
CREATE TABLE Sales (
    SaleID INT,
    ProductID INT,
    SaleDate DATE,
    Quantity INT,
    TotalAmount DECIMAL(10,2),
    PRIMARY KEY (SaleID, ProductID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

Composite Primary Key

- `(SaleID, ProductID)`

Foreign Key

- `ProductID → Products(ProductID)`

Why composite?

- One sale (`SaleID`) can contain **multiple products**
- The same product can appear in **many sales**
- The combination guarantees uniqueness

---

4. Example data (to make it click)

Categories

CategoryID	CategoryName
1	Electronics
2	Clothing

Products

ProductID	ProductName	CategoryID	Price
101	Laptop	1	1000
102	T-Shirt	2	25

Sales

SaleID	ProductID	SaleDate	Quantity	TotalAmount
1	101	2025-02-01	1	1000
1	102	2025-02-01	2	50
2	101	2025-02-02	1	1000

Notice:

- `SaleID = 1` appears **twice**
- `ProductID = 101` appears **twice**
- But `(SaleID, ProductID)` is **never duplicated**

---

5. Key takeaway (exam-friendly)

- **Primary Key** → uniquely identifies a row
- **Foreign Key** → links tables together
- **Composite Primary Key** → multiple columns together ensure uniqueness (very common in sales/order tables)