# Task-1

## Objective:

Create a well-structured relational schema with tables and relationships for an E-commerce system.

## Tools:

You can use any one of the following tools:

- MySQL Workbench
- pgAdmin (for PostgreSQL)
- SQLiteStudio

**Choose Domain**: E-commerce System

### Identify Entities and Relationships:

| Entity | Attributes |
|---|---|
| Customers | customer_id, name, email, phone, address |
| Products | product_id, name, price, stock_qty, category_id |
| Categories | category_id, category_name |
| Orders | order_id, customer_id, order_date, total_amount |
| OrderItems | order_item_id, order_id, product_id, quantity, price |
| Payments | payment_id, order_id, payment_method, payment_date, amount |

**Relationships**:

- A Customer can place many Orders.
- An Order can have multiple OrderItems.
- Each OrderItem references a Product.
- Each Product belongs to one Category.
- Each Order has one Payment.

---

## 3. SQL Script to Create Schema

```
sql
CopyEdit
-- Create Database
CREATE DATABASE IF NOT EXISTS ecommerce_db;
USE ecommerce_db;

-- Customers Table
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(15),
    address TEXT
);

-- Categories Table
CREATE TABLE Categories (
    category_id INT PRIMARY KEY AUTO_INCREMENT,
    category_name VARCHAR(50)
);

-- Products Table
CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    price DECIMAL(10, 2),
    stock_qty INT,
    category_id INT,
    FOREIGN KEY (category_id) REFERENCES Categories(category_id)
);

-- Orders Table
CREATE TABLE Orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_date DATE,
    total_amount DECIMAL(10,2),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);

-- OrderItems Table
CREATE TABLE OrderItems (
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    product_id INT,
    quantity INT,
    price DECIMAL(10,2),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

-- Payments Table
CREATE TABLE Payments (
    payment_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    payment_method VARCHAR(50),
```

```
    payment_date DATE,
    amount DECIMAL(10,2),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);
```

---

# ☑ Outcome

- A well-structured **SQL schema** with **referential integrity** using **foreign keys**.
- An **ER diagram** showing entities, attributes, and relationships.

---

- An **ER diagram** showing entities, attributes, and relationships.