To develop a database for an online store (e-store), we'll create tables for key entities such as products, customers, orders, and payments. I'll guide you through the design of the database schema, followed by example SQL queries for managing customer orders.

**Step 1: Database Schema Design**

We will create the following tables:

1. **Customers** – to store customer details.
2. **Products** – to store product details.
3. **Orders** – to store order information.
4. **Order\_Items** – to store which products were ordered.
5. **Payments** – to handle payment details.

**1. Customers Table**

This table stores customer information.

sql

Copy code

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

PhoneNumber VARCHAR(15),

Address VARCHAR(255)

);

**2. Products Table**

This table holds product details.

sql

Copy code

CREATE TABLE Products (

ProductID INT PRIMARY KEY AUTO\_INCREMENT,

ProductName VARCHAR(100),

ProductDescription TEXT,

Price DECIMAL(10, 2),

Stock INT

);

**3. Orders Table**

This table stores the basic order information.

sql

Copy code

CREATE TABLE Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

CustomerID INT,

OrderDate DATETIME,

TotalAmount DECIMAL(10, 2),

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

**4. Order\_Items Table**

This table stores the details of each product within an order.

sql

Copy code

CREATE TABLE Order\_Items (

OrderItemID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

ProductID INT,

Quantity INT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);

**5. Payments Table**

This table handles the payment information.

sql

Copy code

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY AUTO\_INCREMENT,

OrderID INT,

PaymentDate DATETIME,

PaymentMethod VARCHAR(50),

PaymentAmount DECIMAL(10, 2),

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)

);

**Step 2: Example SQL Queries to Handle Customer Orders**

**1. Insert a New Customer**

sql

Copy code

INSERT INTO Customers (FirstName, LastName, Email, PhoneNumber, Address)

VALUES ('John', 'Doe', 'john.doe@example.com', '1234567890', '123 Main St');

**2. Insert a New Product**

sql

Copy code

INSERT INTO Products (ProductName, ProductDescription, Price, Stock)

VALUES ('Laptop', 'High-performance laptop', 1200.00, 50);

**3. Place an Order**

You’ll first need to create an order, then add items to that order.

1. Insert a new order for a customer:

sql

Copy code

INSERT INTO Orders (CustomerID, OrderDate, TotalAmount)

VALUES (1, NOW(), 1200.00);

1. Add products to the order:

sql

Copy code

INSERT INTO Order\_Items (OrderID, ProductID, Quantity)

VALUES (1, 1, 1); -- Order 1, Product 1, Quantity 1

1. Update the stock after placing the order:

sql

Copy code

UPDATE Products

SET Stock = Stock - 1

WHERE ProductID = 1;

**4. Record a Payment**

sql

Copy code

INSERT INTO Payments (OrderID, PaymentDate, PaymentMethod, PaymentAmount)

VALUES (1, NOW(), 'Credit Card', 1200.00);

**5. Query Customer Orders**

To retrieve details of a customer’s orders:

sql

Copy code

SELECT o.OrderID, o.OrderDate, p.ProductName, oi.Quantity, o.TotalAmount

FROM Orders o

JOIN Order\_Items oi ON o.OrderID = oi.OrderID

JOIN Products p ON oi.ProductID = p.ProductID

WHERE o.CustomerID = 1;