**Task -3 : Retail Store Task**

CREATE DATABASE retail\_store;

USE retail\_store;

**Create Tables :**

**Customers Table-**

CREATE TABLE Customers(customer\_id int PRIMARY KEY,first\_name varchar(100),last\_name varchar(100),email varchar(100),phone varchar(20),address text,join\_date date);

**Products Table-**

CREATE TABLE Products(product\_id int PRIMARY KEY,product\_name varchar(100),category varchar(50),price decimal(10,2),stock\_quantity int);

**Orders Table-**

CREATE TABLE Orders(order\_id int PRIMARY KEY,customer\_id int,order\_date date,total\_amount decimal(10,2),order\_status varchar(20),FOREIGN KEY(customer\_id) REFERENCES customers(customer\_id));

**OrderDetails Table-**

CREATE TABLE OrderDetails(order\_detail\_id int PRIMARY KEY,order\_id int,product\_id int,quantity int,unit\_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,order\_id int,payment\_date date,payment\_amount decimal(10,2),payment\_method varchar(20),FOREIGN KEY(order\_id) REFERENCES orders(order\_id));

**Insert Data in The Tables –**

**Customer Table –**

INSERT INTO Customers VALUES

(1, 'Amit', 'Sharma', 'amit.sharma@example.com', '9876543210', 'Delhi, India', '2023-01-10'), (2, 'Priya', 'Verma', 'priya.verma@example.com', '9876543211', 'Mumbai, India', '2023-02-15'), (3, 'Rahul', 'Gupta', 'rahul.gupta@example.com', '9876543212', 'Bangalore, India', '2023-03-20'), (4, 'Neha', 'Singh', 'neha.singh@example.com', '9876543213', 'Kolkata, India', '2023-04-10'), (5, 'Anil', 'Kumar', 'anil.kumar@example.com', '9876543214', 'Chennai, India', '2023-05-05'), (6, 'Sita', 'Pandey', 'sita.pandey@example.com', '9876543215', 'Hyderabad, India', '2023-06-15'), (7, 'Raj', 'Mehta', 'raj.mehta@example.com', '9876543216', 'Pune, India', '2023-07-25'), (8, 'Kiran', 'Das', 'kiran.das@example.com', '9876543217', 'Ahmedabad, India', '2023-08-05'), (9, 'Arjun', 'Yadav', 'arjun.yadav@example.com', '9876543218', 'Lucknow, India', '2023-09-10'), (10, 'Meera', 'Nair', 'meera.nair@example.com', '9876543219', 'Coimbatore, India', '2023-10-01');

**Products Table –**

INSERT INTO Products VALUES

(1, 'Laptop', 'Electronics', 55000.00, 50), (2, 'Smartphone', 'Electronics', 15000.00, 100), (3, 'Air Conditioner', 'Home Appliances', 30000.00, 20), (4, 'Washing Machine', 'Home Appliances', 25000.00, 15), (5, 'Refrigerator', 'Home Appliances', 20000.00, 25), (6, 'Microwave Oven', 'Kitchen Appliances', 12000.00, 30), (7, 'Television', 'Electronics', 40000.00, 40), (8, 'Mixer Grinder', 'Kitchen Appliances', 5000.00, 50), (9, 'Water Purifier', 'Kitchen Appliances', 10000.00, 35), (10, 'Vacuum Cleaner', 'Home Appliances', 8000.00, 10);

**Orders Table –**

INSERT INTO Orders VALUES

(1, 1, '2023-11-01', 60000.00, 'Shipped'), (2, 2, '2023-11-10', 25000.00, 'Delivered'), (3, 3, '2023-11-15', 20000.00, 'Pending'), (4, 4, '2023-11-20', 40000.00, 'Shipped'), (5, 5, '2023-11-25', 55000.00, 'Delivered'), (6, 6, '2023-11-30', 30000.00, 'Pending'), (7, 7, '2023-12-01', 15000.00, 'Delivered'), (8, 8, '2023-12-05', 20000.00, 'Shipped'), (9, 9, '2023-12-10', 5000.00, 'Delivered'), (10, 10, '2023-12-15', 8000.00, 'Pending');

**OrderDetails Table –**

INSERT INTO OrderDetails VALUES

(1, 1, 1, 1, 55000.00), (2, 1, 2, 1, 15000.00), (3, 2, 4, 1, 25000.00), (4, 3, 5, 1, 20000.00), (5, 4, 7, 1, 40000.00), (6, 5, 1, 1, 55000.00), (7, 6, 3, 1, 30000.00), (8, 7, 2, 1, 15000.00), (9, 8, 9, 1, 20000.00), (10, 9, 8, 1, 5000.00);

**Payments Table –**

INSERT INTO Payments VALUES

(1, 1, '2023-11-02', 60000.00, 'Credit Card'), (2, 2, '2023-11-11', 25000.00, 'UPI'), (3, 3, '2023-11-16', 20000.00, 'Cash'), (4, 4, '2023-11-21', 40000.00, 'Debit Card'), (5, 5, '2023-11-26', 55000.00, 'Net Banking'), (6, 6, '2023-12-01', 30000.00, 'Cash'), (7, 7, '2023-12-02', 15000.00, 'UPI'), (8, 8, '2023-12-06', 20000.00, 'Credit Card'), (9, 9, '2023-12-11', 5000.00, 'Cash'), (10, 10, '2023-12-16', 8000.00, 'Debit Card');

**SQL Queries for the Case Study**

1. **Find the Total Number of Orders for Each Customer**

SELECT customer\_id,COUNT(order\_id) as total\_orders

FROM orders

GROUP BY customer\_id;

1. **Find the Total Sales Amount for Each Product (Revenue per Product)**

SELECT product\_id,sum(quantity\*unit\_price) as total\_sales

from orderdetails

GROUP BY product\_id;

1. **Find the Most Expensive Product Sold**

SELECT product\_id,max(unit\_price) as highest\_price

from orderdetails;

1. **Get the List of Customers Who Have Placed Orders in the Last 30 Days**

SELECT DISTINCT c.customer\_id,c.first\_name,c.last\_name

FROM customers as c

JOIN orders as o

ON c.customer\_id = o.customer\_id

where o.order\_date >=DATE\_SUB(CURDATE(), INTERVAL 30 DAY);

1. **Calculate the Total Amount Paid by Each Customer**

SELECT o.customer\_id,sum(p.payment\_amount) as total\_paid

from payments as p

JOIN orders as o

ON p.order\_id = o.order\_id

GROUP BY o.customer\_id;

1. **Get the Number of Products Sold by Category**

SELECT p.category , sum(od.quantity) AS total\_product\_sold

from products as p

JOIN orderdetails as od

ON p.product\_id = od.product\_id

GROUP BY p.category;

1. **List All Orders That Are Pending (i.e., Orders that haven't been shipped yet)**

SELECT \* from orders

WHERE order\_status = 'Pending';

1. **Find the Average Order Value (Total Order Amount / Number of Orders)**

SELECT AVG(total\_amount) as avg\_order\_value

FROM orders;

1. **List the Top 5 Customers Who Have Spent the Most Money**

SELECT o.customer\_id,sum(p.payment\_amount) as total\_spent

from payments as p

JOIN orders as o

ON p.order\_id = o.order\_id

GROUP BY o.customer\_id

ORDER BY total\_spent DESC

LIMIT 5;

1. **Find the Products That Have Never Been Sold**

SELECT p.product\_id,p.product\_name

FROM products as p

LEFT JOIN orderdetails as od

on p.product\_id = od.product\_id

where od.product\_id IS null;