Customers table

```
CREATE DATABASE RETAIL_STORE;

USE RETAIL_STORE;

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);

INSERT INTO CUSTOMERS (CUSTOMER_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE, ADDRESS, JOIN_DATE) VALUES
(1, 'Ravi', 'Kumar', 'ravi.kumar@example.com', '9876543210', '123 MG Road, Delhi', '2022-01-15'),
(2, 'Priya', 'Sharma', 'priya.sharma@example.com', '9876543211', '456 Park Street, Mumbai', '2022-02-20'),
(3, 'Amit', 'Verma', 'amit.verma@example.com', '9876543212', '789 Ring Road, Bangalore', '2022-03-05'),
(4, 'Sneha', 'Joshi', 'sneha.joshi@example.com', '9876543213', '321 Green Lane, Pune', '2022-03-25'),
(5, 'Rahul', 'Mehta', 'rahul.mehta@example.com', '9876543214', '654 North Ave, Hyderabad', '2022-04-10'),
(6, 'Anjali', 'Singh', 'anjali.singh@example.com', '9876543215', '987 South Blvd, Chennai', '2022-04-18'),
(7, 'Vikram', 'Patel', 'vikram.patel@example.com', '9876543216', '147 East Road, Jaipur', '2022-05-01'),
(8, 'Pooja', 'Reddy', 'pooja.reddy@example.com', '9876543217', '258 Market Lane, Kochi', '2022-05-15'),
(9, 'Saurabh', 'Kapoor', 'saurabh.kapoor@example.com', '9876543218', '369 High Street, Noida', '2022-06-05'),
(10, 'Neha', 'Chopra', 'neha.chopra@example.com', '9876543219', '753 College Rd, Ahmedabad', '2022-06-20'),
(11, 'Deepak', 'Gupta', 'deepak.gupta@example.com', '9876543220', '852 Lake View, Surat', '2022-07-01'),
(12, 'Kavita', 'Desai', 'kavita.desai@example.com', '9876543221', '963 Mall Road, Bhopal', '2022-07-12'),
(13, 'Manish', 'Tiwari', 'manish.tiwari@example.com', '9876543222', '174 Station Rd, Lucknow', '2022-08-01'),
(14, 'Shalini', 'Mishra', 'shalini.mishra@example.com', '9876543223', '385 Business Bay, Indore', '2022-08-15'),
(15, 'Nikhil', 'Agarwal', 'nikhil.agarwal@example.com', '9876543224', '296 River St, Patna', '2022-09-01');
```

Products tables

```
CREATE TABLE PRODUCTS (
PRODUCT_ID INT PRIMARY  KEY,
PRODUCT_NAME VARCHAR(100),
CATEGORY VARCHAR(50),
PRICE DECIMAL(10,2),
STOCK_QUANTITY INT);
```

```
INSERT INTO PRODUCTS (PRODUCT_ID, PRODUCT_NAME, CATEGORY, PRICE, STOCK_QUANTITY) VALUES
(1, 'Samsung Galaxy M14', 'Electronics', 13999.00, 50),
(2, 'Apple iPhone 14', 'Electronics', 70999.00, 30),
(3, 'Sony WH-1000XM4 Headphones', 'Electronics', 19999.00, 25),
(4, 'Dell Inspiron 15 Laptop', 'Computers', 54999.00, 15),
(5, 'HP DeskJet Printer', 'Computers', 5999.00, 20),
(6, 'Nike Running Shoes', 'Footwear', 3999.00, 60),
(7, 'Adidas Sneakers', 'Footwear', 4499.00, 45),
(8, 'Levi’s Jeans', 'Clothing', 1999.00, 100),
(9, 'Ray-Ban Sunglasses', 'Accessories', 6999.00, 40),
(10, 'Fastrack Watch', 'Accessories', 2999.00, 35),
```

Orders table

```
CREATE TABLE ORDERS (
    ORDER_ID INT PRIMARY KEY,
    CUSTOMER_ID INT,
    TOTAL_AMOUNT DECIMAL(10,2),
    ORDER_STATUS ENUM('PENDING', 'SHIPPED'),
    FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMERS(CUSTOMER_ID)
);

INSERT INTO ORDERS (ORDER_ID, CUSTOMER_ID, TOTAL_AMOUNT, ORDER_STATUS) VALUES
(1, 1, 13999.00, 'SHIPPED'),
(2, 2, 70999.00, 'PENDING'),

(3, 3, 19999.00, 'SHIPPED'),

(4, 4, 54999.00, 'SHIPPED'),

(5, 5, 5999.00, 'PENDING'),

(6, 6, 3999.00, 'SHIPPED'),

(7, 7, 4499.00, 'PENDING'),

(8, 8, 1999.00, 'SHIPPED'),

(9, 9, 6999.00, 'PENDING'),

(10, 10, 2999.00, 'SHIPPED'),

(11, 11, 3499.00, 'PENDING'),

(12, 12, 2499.00, 'SHIPPED'),

(13, 13, 11999.00, 'SHIPPED'),

(14, 14, 15999.00, 'PENDING'),

(15, 15, 1299.00, 'SHIPPED');
```

Order details table

```
CREATE TABLE ORDER_DETAILS (

   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)
);
INSERT INTO ORDER_DETAILS VALUES
(1, 1, 101, 2, 13999.00),
(2, 2, 102, 1, 70999.00),
(3, 3, 103, 3, 19999.00),
(4, 4, 104, 1, 54999.00),
(5, 5, 105, 4, 5999.00),
(6, 6, 106, 2, 3999.00),
(7, 7, 107, 1, 4499.00),
(8, 8, 108, 5, 1999.00),
(9, 9, 109, 2, 6999.00),
(10, 10, 110, 1, 2999.00);
```

Payments tables

```
CREATE TABLE PAYMENTS (
    PAYMENT_ID INT PRIMARY KEY,
    ORDER_ID INT,
    PAYMENT_DATE DATE,
    PAYMENT_AMOUNT DECIMAL(10,2),
    PAYMENT_METHOD ENUM('CREDIT_CARD','PAYPAL','DEBIT_CARD','CASH','CHEQUE','UPI'),
    FOREIGN KEY (ORDER_ID) REFERENCES ORDERS(ORDER_ID)
);
```

```
INSERT INTO PAYMENTS (PAYMENT_ID, ORDER_ID, PAYMENT_DATE, PAYMENT_AMOUNT,
PAYMENT_METHOD)

VALUES

(1, 1, '2024-01-05', 120.50, 'CREDIT_CARD'),

(2, 2, '2024-01-10', 99.99, 'PAYPAL'),

(3, 3, '2024-01-12', 45.75, 'DEBIT_CARD'),

(4, 4, '2024-01-15', 250.00, 'CASH'),

(5, 5, '2024-01-18', 300.20, 'UPI'),

(6, 6, '2024-01-20', 150.00, 'CREDIT_CARD'),

(7, 7, '2024-01-21', 210.75, 'CHEQUE'),

(8, 8, '2024-01-22', 180.50, 'PAYPAL'),

(9, 9, '2024-01-23', 99.90, 'CASH'),

(10, 10, '2024-01-24', 275.30, 'UPI');
```

Sql Queries for the case study

Find the Total Number of Orders for Each Customer

```
    SELECT CUSTOMER_ID, COUNT(*) AS TOTAL_ORDERS FROM ORDERS

   GROUP BY CUSTOMER_ID;
```

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

```
      SELECT PRODUCT_ID, SUM(QUANTITY * UNIT_PRICE) AS TOTAL_REVENUE FROM
      ORDER_DETAILS
       GROUP BY PRODUCT_ID;
```

3. Find the Most Expensive Product Sold

```
SELECT

  P.PRODUCT_ID, P.PRODUCT_NAME, MAX(OD.UNIT_PRICE) AS MAX_PRICE_SOLD
```

```
FROM
   ORDER_DETAILS OD
JOIN
   PRODUCTS P ON OD.PRODUCT_ID = P.PRODUCT_ID
GROUP BY
   P.PRODUCT_ID, P.PRODUCT_NAME
ORDER BY MAX_PRICE_SOLD DESC LIMIT  1;
```

4.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, O.ORDER_ID, P.PAYMENT_DATE
FROM
   CUSTOMERS C
JOIN
   ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID
JOIN
   PAYMENTS P ON O.ORDER_ID = P.ORDER_ID
WHERE
   P.PAYMENT_DATE >= DATE('2025-05-23') - INTERVAL 30 DAY;
```

5.Calculate the Total Amount Paid by Each Customer

```
  SELECT C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME,
   SUM(P.PAYMENT_AMOUNT) AS TOTAL_PAID
FROM CUSTOMERS C
JOIN  ORDERS O ON C.CUSTOMER_ID = O.CUSTOMER_ID
JOIN PAYMENTS P ON O.ORDER_ID = P.ORDER_ID
GROUP BY C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME;
```

6. Get the Number of Products Sold by Category

```
 SELECT  P.CATEGORY, SUM(OD.QUANTITY) AS TOTAL_SOLD
FROM PRODUCTS P
JOIN ORDER_DETAILS OD
   ON P.PRODUCT_ID = OD.PRODUCT_ID
GROUP BY P.CATEGORY;
```

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

```
SELECT * FROM ORDERS
WHERE ORDER_STATUS = 'PENDING';
```

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

```
SELECT AVG(TOTAL_AMOUNT) AS AVERAGE_ORDER_VALUE
FROM ORDERS;
```

List the Top 5 Customers Who Have Spent the Most Money

```
SELECT C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME,
   SUM(P.PAYMENT_AMOUNT) AS TOTAL_SPENT
FROM CUSTOMERS C
JOIN ORDERS O
   ON C.CUSTOMER_ID = O.CUSTOMER_ID
```

```
JOIN PAYMENTS P
   ON O.ORDER_ID = P.ORDER_ID
GROUP BY
   C.CUSTOMER_ID, C.FIRST_NAME, C.LAST_NAME
ORDER BY TOTAL_SPENT DESC
LIMIT 5;
```

Find the Products That Have Never Been Sold

```
SELECT P.PRODUCT_ID, P.PRODUCT_NAME
FROM PRODUCTS P
LEFT JOIN
   ORDER_DETAILS OD
   ON P.PRODUCT_ID = OD.PRODUCT_ID
WHERE OD.PRODUCT_ID IS NULL;
```