Create database Retail

1.CREATE TABLE Customers (

customer\_id Integer PRIMARY KEY,

first\_name Text(100),

last\_name Text(100),

email Text(100),

phone Text(20),

address TEXT,

join\_date DATE

);

2. CREATE TABLE Products (

product\_id Integer PRIMARY KEY,

product\_name Text(100),

category Text(50),

price DECIMAL(10,2),

stock\_quantity Integer

);

3. CREATE TABLE Orders (

order\_id Integer PRIMARY KEY,

customer\_id Integer,

order\_date DATE,

total\_amount DECIMAL(10,2),

order\_status Text(20),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

4. CREATE TABLE OrderDetails (

order\_detail\_id Integer PRIMARY KEY,

order\_id Integer,

product\_id Integer,

quantity Integer,

unit\_price DECIMAL(10,2),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

5. CREATE TABLE Payments (

payment\_id Integer PRIMARY KEY,

order\_id Integer,

payment\_date DATE,

payment\_amount DECIMAL(10,2),

payment\_method Text(20),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id));

1.INSERT INTO Customers (customer\_id, first\_name, last\_name, email, phone, address, join\_date) VALUES

(1, 'Amit', 'Sharma', '[amit.sharma@gmail.com](mailto:amit.sharma@gmail.com)', '9876543210', 'Delhi', '2024-06-15'),

(2, 'Priya', 'Mehta', '[priya.mehta@gmail.com](mailto:priya.mehta@gmail.com)', '9876543211', 'Kolkata', '2024-07-20'),

(3, 'Rahul', 'Verma', '[rahul.verma@gmail.com](mailto:rahul.verma@gmail.com)', '9876543212', 'Mumbai', '2023-08-05'),

(4, 'Sneha', 'Patel', '[sneha.patel@gmail.com](mailto:sneha.patel@gmail.com)', '9876543213', 'Ahmedabad', '2023-08-22'),

(5, 'Vikram', 'Reddy', '[vikram.reddy@gmail.com](mailto:vikram.reddy@gmail.com)', '9876543214', 'Hyderabad', '2023-09-10'),

(6, 'Anjali', 'Kumar', '[anjali.kumar@gmail.com](mailto:anjali.kumar@gmail.com)', '9876543215', 'Pune', '2025-01-01'),

(7, 'Rohan', 'Joshi', '[rohan.joshi@gmail.com](mailto:rohan.joshi@gmail.com)', '9876543216', 'Bangalore', '2025-02-25'),

(8, 'Kavita', 'Nair', '[kavita.nair@gmail.com](mailto:kavita.nair@gmail.com)', '9876543217', 'Kochi', '2025-03-03'),

(9, 'Suresh', 'Mishra', '[suresh.mishra@gmail.com](mailto:suresh.mishra@gmail.com)', '9876543218', 'Patna', '2023-11-18'),

(10, 'Neha', 'Singh', '[neha.singh@gmail.com](mailto:neha.singh@gmail.com)', '9876543219', 'Lucknow', '2025-04-01');

2. INSERT INTO Products (product\_id, product\_name, category, price, stock\_quantity) VALUES

(1, 'Basmati Rice 5kg', 'Grocery', 499.00, 120),

(2, 'Masala Dabba', 'Kitchenware', 799.00, 50),

(3, 'Chikankari Kurti', 'Clothing', 1199.00, 80),

(4, 'Ayurvedic Hair Oil', 'Personal Care', 299.00, 150),

(5, 'Copper Water Bottle', 'Wellness', 599.00, 60),

(6, 'Sandalwood Soap Pack', 'Personal Care', 199.00, 200),

(7, 'Kolhapuri Chappal', 'Footwear', 899.00, 40),

(8, 'Mysore Silk Saree', 'Clothing', 2499.00, 25),

(9, 'Terracotta Planter Set', 'Home Decor', 699.00, 35),

(10, 'Himalayan Pink Salt 1kg', 'Grocery', 149.00, 100);

3. INSERT INTO Orders (order\_id, customer\_id, order\_date, total\_amount, order\_status) VALUES

(1, 1, '2024-01-05', 1498.00, 'Shipped'),

(2, 2, '2024-01-10', 299.00, 'Delivered'),

(3, 3, '2024-01-15', 2499.00, 'Pending'),

(4, 4, '2024-01-18', 499.00, 'Shipped'),

(5, 5, '2024-01-20', 1598.00, 'Shipped'),

(6, 6, '2024-01-22', 699.00, 'Pending'),

(7, 7, '2024-01-25', 1098.00, 'Delivered'),

(8, 8, '2024-01-28', 199.00, 'Shipped'),

(9, 9, '2024-02-01', 899.00, 'Cancelled'),

(10, 10, '2024-02-05', 2499.00, 'Delivered');

4. INSERT INTO OrderDetails (order\_detail\_id, order\_id, product\_id, quantity, unit\_price) VALUES

(1, 1, 1, 2, 499.00),

(2, 1, 4, 1, 299.00),

(3, 2, 6, 1, 199.00),

(4, 3, 8, 1, 2499.00),

(5, 4, 1, 1, 499.00),

(6, 5, 3, 1, 1199.00),

(7, 5, 2, 1, 399.00),

(8, 6, 9, 1, 699.00),

(9, 7, 5, 2, 549.00),

(10, 10, 8, 1, 2499.00);

5. INSERT INTO Payments (payment\_id, order\_id, payment\_date, payment\_amount, payment\_method) VALUES

(1, 1, '2024-01-05', 1498.00, 'Credit Card'),

(2, 2, '2024-01-10', 299.00, 'UPI'),

(3, 3, '2024-01-15', 2499.00, 'Net Banking'),

(4, 4, '2024-01-18', 499.00, 'Cash on Delivery'),

(5, 5, '2024-01-20', 1598.00, 'Credit Card'),

(6, 6, '2024-01-22', 699.00, 'UPI'),

(7, 7, '2024-01-25', 1098.00, 'Debit Card'),

(8, 8, '2024-01-28', 199.00, 'UPI'),

(9, 9, '2024-02-01', 899.00, 'Net Banking'),

(10, 10, '2024-02-05', 2499.00, 'Credit Card');

Assignment Queries

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

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,

COUNT(o.order\_id) AS total\_orders

FROM

Customers c

LEFT JOIN

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

GROUP BY

c.customer\_id, c.first\_name, c.last\_name;

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

SELECT

p.product\_id,

p.product\_name,

SUM(od.quantity \* od.unit\_price) AS total\_revenue

FROM

OrderDetails od

JOIN

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

GROUP BY

p.product\_id, p.product\_name

ORDER BY

total\_revenue DESC;

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

SELECT

p.product\_id,

p.product\_name,

od.unit\_price

FROM

OrderDetails od

JOIN

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

ORDER BY

od.unit\_price DESC

LIMIT 1;

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

SELECT

DISTINCT c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,

c.email,

o.order\_date

FROM

Customers c

JOIN

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

WHERE

o.order\_date >= CURRENT\_DATE - INTERVAL 30 DAY;

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

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,

SUM(p.payment\_amount) AS total\_amount\_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

ORDER BY

total\_amount\_paid DESC;

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

SELECT

p.category,

SUM(od.quantity) AS total\_products\_sold

FROM

OrderDetails od

JOIN

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

GROUP BY

p.category

ORDER BY

total\_products\_sold DESC;

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

SELECT

o.order\_id,

CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,

o.order\_date,

o.total\_amount,

o.order\_status

FROM

Orders o

JOIN

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

WHERE

o.order\_status = 'Pending'

ORDER BY

o.order\_date DESC;

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

SELECT

ROUND(SUM(total\_amount) / COUNT(order\_id), 2) AS average\_order\_value

FROM

Orders;

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

SELECT

c.customer\_id,

CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_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;

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

SELECT

p.product\_id,

p.product\_name,

p.category,

p.price

FROM

Products p

LEFT JOIN

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

WHERE

od.product\_id IS NULL;