

Case Study Mysql Online Retail Store

1. Write SQL statements to create the tables as described in the database schema.

2. Insert Data

3. Retrieve all customers who registered in 2023.

```
1 • CREATE DATABASE OnlineRetailStore;
2 • USE OnlineRetailStore;
3
4 • CREATE TABLE Customers (
5     CustomerID INT PRIMARY KEY,
6     FirstName VARCHAR(255),
7     LastName VARCHAR(255),
8     Email VARCHAR(255),
9     Phone VARCHAR(255),
10    RegistrationDate DATE
11 );
12 • INSERT INTO Customers VALUES
13    (1, 'John', 'Doe', 'john.doe@example.com', '123-456-7890', '2023-01-15'),
14    (2, 'Jane', 'Smith', 'jane.smith@example.com', '234-567-8901', '2023-02-20'),
15    (3, 'Alice', 'Johnson', 'alice.johnson@example.com', '345-678-9012', '2023-03-10'),
16    (4, 'Bob', 'Brown', 'bob.brown@example.com', '456-789-0123', '2023-04-05'),
17    (5, 'Charlie', 'Davis', 'charlie.davis@example.com', '567-890-1234', '2023-05-12'),
18    (6, 'David', 'Wilson', 'david.wilson@example.com', '678-901-2345', '2023-06-15'),
19    (7, 'Emma', 'Thomas', 'emma.thomas@example.com', '789-012-3456', '2023-07-01'),
20    (8, 'Fiona', 'Garcia', 'fiona.garcia@example.com', '890-123-4567', '2023-07-10'),
21    (9, 'George', 'Martinez', 'george.martinez@example.com', '901-234-5678', '2023-07-20'),
22    (10, 'Hannah', 'Rodriguez', 'hannah.rodriguez@example.com', '012-345-6789', '2023-07-25');
23
```

```

32 • INSERT INTO Products VALUES
33     (1, 'Laptop', 'Electronics', 999.99, 50),
34     (2, 'Smartphone', 'Electronics', 499.99, 100),
35     (3, 'Tablet', 'Electronics', 299.99, 75),
36     (4, 'Headphones', 'Accessories', 49.99, 200),
37     (5, 'Charger', 'Accessories', 19.99, 300),
38     (6, 'Keyboard', 'Accessories', 29.99, 150),
39     (7, 'Mouse', 'Accessories', 19.99, 250),
40     (8, 'Monitor', 'Electronics', 199.99, 30),
41     (9, 'Printer', 'Electronics', 149.99, 20),
42     (10, 'USB Cable', 'Accessories', 9.99, 400);

44 • CREATE TABLE Orders (
45     OrderID INT PRIMARY KEY,
46     OrderDate DATE,
47     CustomerID INT,
48     TotalAmount DECIMAL(10, 2),
49     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
50 );
51

52 • INSERT INTO Orders VALUES
53     (1, '2023-06-01', 1, 1049.98),
54     (2, '2023-06-05', 2, 549.98),
55     (3, '2023-06-10', 3, 999.99),
56     (4, '2023-06-15', 4, 69.98),
57     (5, '2023-06-20', 5, 519.98),
58     (6, '2023-06-25', 6, 229.98),
59     (7, '2023-07-02', 7, 1199.97),
60     (8, '2023-07-12', 8, 49.98),
61     (9, '2023-07-18', 9, 349.98),
62     (10, '2023-07-22', 10, 39.98);
63

```

```

64 • CREATE TABLE OrderDetails (
65     OrderDetailID INT PRIMARY KEY,
66     OrderID INT,
67     ProductID INT,
68     Quantity INT,
69     Price DECIMAL(10, 2),
70     FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
71     FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
72 );
73
74 • INSERT INTO OrderDetails VALUES
75 (1, 1, 1, 1, 999.99), (2, 1, 4, 1, 49.99), (3, 2, 2, 1, 499.99), (4, 2, 5, 1, 49.99), (5, 3, 1, 1, 999.99),
76 (6, 4, 4, 1, 49.99), (7, 4, 5, 1, 19.99), (8, 5, 2, 1, 499.99), (9, 5, 5, 1, 19.99), (10, 6, 3, 1, 199.99),
77 (11, 6, 5, 1, 29.99), (12, 7, 1, 1, 999.99), (13, 7, 3, 1, 199.99), (14, 8, 7, 1, 19.99), (15, 8, 8, 1, 29.99),
78 (16, 9, 4, 2, 149.99), (17, 9, 9, 1, 49.99), (18, 10, 10, 4, 9.99);
79
80 • SELECT * FROM Customers;
81 • SELECT * FROM Products;
82 • SELECT * FROM Orders;
83 • SELECT * FROM OrderDetails;
84

```

3. Retrieve all customers who registered in 2023:

```

129 • SELECT FirstName, LastName, SUM(TotalAmount) as TotalSpent
130 FROM Customers
131 JOIN Orders ON Customers.CustomerID = Orders.CustomerID
132 GROUP BY Customers.CustomerID;

```

Result Grid			
		Filter Rows:	
		Export:	
		Wrap Cell Content:	
	FirstName	LastName	TotalSpent
▶	John	Doe	1049.98
	Jane	Smith	549.98
	Alice	Johnson	999.99
	Bob	Brown	69.98
	Charlie	Davis	519.98
	David	Wilson	229.98
	Emma	Thomas	1199.97
	Fiona	Garcia	49.98
	George	Martinez	349.98
	Hannah	Rodriguez	39.98

4. List all products in the 'Electronics' category.

```

133 • SELECT Category, AVG(Price) as AvgPrice
134 FROM Products
135 GROUP BY Category;

```

Result Grid

↺↻

Filter Rows:

Export:

Wrap Cell Cont

	Category	AvgPrice
▶	Electronics	429.990000
	Accessories	25.990000

5. Find the total number of orders placed by each customer.

```

136 • SELECT * FROM Customers
137 WHERE CustomerID NOT IN (SELECT DISTINCT CustomerID FROM Orders);
138

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	CustomerID	FirstName	LastName	Email	Phone	RegistrationDate
*	NULL	NULL	NULL	NULL	NULL	NULL

6. Calculate the total sales amount for each product.

```

138 • SELECT ProductName, SUM(Products.Price * Quantity) as TotalSales
139 FROM OrderDetails
140 JOIN Products ON OrderDetails.ProductID = Products.ProductID
141 GROUP BY ProductName
142 ORDER BY TotalSales DESC LIMIT 3;
143
144

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	ProductName	TotalSales
▶	Laptop	2999.97
	Smartphone	999.98
	Tablet	599.98

7. Retrieve the details of all orders, including the customer name and total amount.

```

143 • SELECT Customers.*
144 FROM Customers
145 JOIN Orders ON Customers.CustomerID = Orders.CustomerID
146 JOIN (SELECT OrderID FROM OrderDetails GROUP BY OrderID HAVING COUNT(DISTINCT ProductID) > 3) as MultipleProducts
147 ON Orders.OrderID = MultipleProducts.OrderID;

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

CustomerID	FirstName	LastName	Email	Phone	RegistrationDate
------------	-----------	----------	-------	-------	------------------

8. List all products that have been ordered along with the quantity ordered for each.

9. Find the order details for orders placed by 'John Doe'.

10. Find customers who have placed an order totaling more than \$500.

```

85 • SELECT * FROM Customers WHERE YEAR(RegistrationDate) = 2023;
86

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:



CustomerID	FirstName	LastName	Email	Phone	RegistrationDate
1	John	Doe	john.doe@example.com	123-456-7890	2023-01-15
2	Jane	Smith	jane.smith@example.com	234-567-8901	2023-02-20
3	Alice	Johnson	alice.johnson@example.com	345-678-9012	2023-03-10
4	Bob	Brown	bob.brown@example.com	456-789-0123	2023-04-05
5	Charlie	Davis	charlie.davis@example.com	567-890-1234	2023-05-12
6	David	Wilson	david.wilson@example.com	678-901-2345	2023-06-15
7	Emma	Thomas	emma.thomas@example.com	789-012-3456	2023-07-01
8	Fiona	Garcia	fiona.garcia@example.com	890-123-4567	2023-07-10
9	George	Martinez	george.martinez@example.com	901-234-5678	2023-07-20
10	Hannah	Rodriguez	hannah.rodriguez@example.com	012-345-6789	2023-07-25
NULL	NULL	NULL	NULL	NULL	NULL




```


86 • SELECT * FROM Products WHERE Category = 'Electronics';

```

Result Grid

  Filter Rows:

Edit:   

Export/Import: 

ProductID	ProductName	Category	Price	StockQuantity
1	Laptop	Electronics	999.99	50
2	Smartphone	Electronics	499.99	100
3	Tablet	Electronics	299.99	75
8	Monitor	Electronics	199.99	30
9	Printer	Electronics	149.99	20
NULL	NULL	NULL	NULL	NULL

11. List the products that have never been ordered.

12. Retrieve the order history for a specific customer, including order date, product names, and quantities. (Assume customer ID is 2)

13. Calculate the average order value for each customer.

```
87 • SELECT CustomerID, COUNT(*) as TotalOrders FROM Orders GROUP BY CustomerID;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
CustomerID	TotalOrders			
1	1			
2	1			
3	1			
4	1			
5	1			
6	1			
7	1			
8	1			
9	1			
10	1			

```
88 • SELECT ProductID, SUM(Price * Quantity) as TotalSales FROM OrderDetails GROUP BY ProductID;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
ProductID	TotalSales			
1	2999.97			
2	999.98			
3	399.98			
4	399.96			
5	119.96			
7	19.99			
8	29.99			
9	49.99			
10	39.96			

```
89 • SELECT Orders.OrderID, FirstName, LastName, TotalAmount
90 FROM Orders
91 JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
92
```





Result Grid				Filter Rows:	Export:	Wrap Cell Content:
OrderID	FirstName	LastName	TotalAmount			
1	John	Doe	1049.98			
2	Jane	Smith	549.98			
3	Alice	Johnson	999.99			
4	Bob	Brown	69.98			
5	Charlie	Davis	519.98			
6	David	Wilson	229.98			
7	Emma	Thomas	1199.97			
8	Fiona	Garcia	49.98			
9	George	Martinez	349.98			
10	Hannah	Rodriguez	39.98			

14. Find the most popular product category based on the number of orders.

15. List all customers who have ordered more than one product in a single order.

16. Find the total revenue generated from each product category.

```
96 • SELECT OrderDetails.*
97 FROM OrderDetails
98 JOIN Orders ON OrderDetails.OrderID = Orders.OrderID
99 JOIN Customers ON Orders.CustomerID = Customers.CustomerID
100 WHERE FirstName = 'John' AND LastName = 'Doe';
101
```



Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	OrderDetailID	OrderID	ProductID	Quantity	Price
▶	1	1	1	1	999.99
	2	1	4	1	49.99

```

92 • SELECT ProductName, SUM(Quantity) as TotalQuantity
93 FROM OrderDetails
94 JOIN Products ON OrderDetails.ProductID = Products.ProductID
95 GROUP BY ProductName;



```

Result Grid		
Filter Rows: <input type="text"/>		
Export: 		
Wrap Cell Content: 		
	ProductName	TotalQuantity
▶	Laptop	3
	Headphones	4
	Smartphone	2
	Charger	4
	Tablet	2
	Mouse	1
	Monitor	1
	Printer	1
	USB Cable	4

```

101 • SELECT Customers.*
102 FROM Customers
103 JOIN Orders ON Customers.CustomerID = Orders.CustomerID
104 WHERE TotalAmount > 500;
105

```

Result Grid						
Filter Rows: <input type="text"/>						
Export: 						
Wrap Cell Content: 						
	CustomerID	FirstName	LastName	Email	Phone	RegistrationDate
▶	1	John	Doe	john.doe@example.com	123-456-7890	2023-01-15
	2	Jane	Smith	jane.smith@example.com	234-567-8901	2023-02-20
	3	Alice	Johnson	alice.johnson@example.com	345-678-9012	2023-03-10
	5	Charlie	Davis	charlie.davis@example.com	567-890-1234	2023-05-12
	7	Emma	Thomas	emma.thomas@example.com	789-012-3456	2023-07-01

17. Retrieve the list of customers along with the total amount they have spent.

18. Find the average price of products in each category.
19. Find all customers who have not placed any orders.
20. List the top 3 products with the highest total sales amount.
21. Find customers who have placed orders for more than 3 different products

```
105 • SELECT * FROM Products
106 WHERE ProductID NOT IN (SELECT DISTINCT ProductID FROM OrderDetails);
107
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit:					
Export/Import:					
Wrap Cell Content:					
	ProductID	ProductName	Category	Price	StockQuantity
▶	6	Keyboard	Accessories	29.99	150
*	NULL	NULL	NULL	NULL	NULL

```
107 • SELECT Orders.OrderDate, ProductName, Quantity
108 FROM Orders
109 JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID
110 JOIN Products ON OrderDetails.ProductID = Products.ProductID
111 WHERE Orders.CustomerID = 2;
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:			
Wrap Cell Content:			
	OrderDate	ProductName	Quantity
▶	2023-06-05	Smartphone	1
	2023-06-05	Charger	1

```

112 • SELECT CustomerID, AVG(TotalAmount) as AvgOrderValue
113 FROM Orders
114 GROUP BY CustomerID;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	CustomerID	AvgOrderValue			
▶	1	1049.980000			
	2	549.980000			
	3	999.990000			
	4	69.980000			
	5	519.980000			
	6	229.980000			
	7	1199.970000			
	8	49.980000			
	9	349.980000			
	10	39.980000			

```

115 • SELECT Category, COUNT(*) as OrderCount
116 FROM OrderDetails
117 JOIN Products ON OrderDetails.ProductID = Products.ProductID
118 GROUP BY Category
119 ORDER BY OrderCount DESC LIMIT 1;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	Category	OrderCount				
▶	Electronics	9				

```

120 • SELECT Customers.*
121 FROM Customers
122 JOIN Orders ON Customers.CustomerID = Orders.CustomerID
123 JOIN (SELECT OrderID FROM OrderDetails GROUP BY OrderID HAVING COUNT(DISTINCT ProductID) > 1) as MultipleProducts
124 ON Orders.OrderID = MultipleProducts.OrderID;
125

```

Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	CustomerID	FirstName	LastName	Email	Phone	RegistrationDate	
▶	1	John	Doe	john.doe@example.com	123-456-7890	2023-01-15	
	2	Jane	Smith	jane.smith@example.com	234-567-8901	2023-02-20	
	4	Bob	Brown	bob.brown@example.com	456-789-0123	2023-04-05	
	5	Charlie	Davis	charlie.davis@example.com	567-890-1234	2023-05-12	
	6	David	Wilson	david.wilson@example.com	678-901-2345	2023-06-15	
	7	Emma	Thomas	emma.thomas@example.com	789-012-3456	2023-07-01	
	8	Fiona	Garcia	fiona.garcia@example.com	890-123-4567	2023-07-10	
	9	George	Martinez	george.martinez@example.com	901-234-5678	2023-07-20	

```

125 • SELECT Category, SUM(Products.Price * Quantity) as TotalRevenue
126 FROM OrderDetails
127 JOIN Products ON OrderDetails.ProductID = Products.ProductID
128 GROUP BY Category;

```

Result Grid		
Filter Rows: <input type="text"/>		
Export: Wrap Cell Content:		
	Category	TotalRevenue
▶	Electronics	4949.91
	Accessories	339.87