SQL-Case Study: Online Retail Store

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

```sql
/*1*/

create table customers_table(customer_id int primary key,
first_name varchar (100),
last_name varchar (100),
email varchar (100),
phone varchar (100),
registration_date date );
select * from customers_table;
```

```sql
create table orders_table(order_id int primary key,
order_date date,
customer_id int,
foreign key(customer_id) references customers_table(customer_id),
total_amt decimal(5,2));
select * from orders_table;
```

```sql
create table product_table(product_id int primary key,
product_name varchar (100),
category varchar (100),
price decimal (5,2),
stock_qty int);
select * from  product_table;
```

```sql
create table order_details_table(Order_detail_ID INT Primary Key,
order_id INT,
foreign key(order_id) references orders_table(order_id),
product_id int,
foreign key(product_id) references product_table(product_id),
quantity int,
price decimal(5,2));
```

2. Insert Data

```sql
insert into customers_table values
(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');
```

```sql
insert into product_table values
(101,'Laptop', 'Electronics', 999.99, 50),
(102,'Smartphone', 'Electronics', 499.99, 100),
(103,'Tablet', 'Electronics', 299.99, 75),
(104,'Headphones', 'Accessories', 49.99, 200),
(105,'Charger', 'Accessories', 19.99, 300),
(106,'Keyboard', 'Accessories', 29.99, 150),
(107,'Mouse', 'Accessories', 19.99, 250),
(108,'Monitor', 'Electronics', 199.99, 30),
(109,'Printer', 'Electronics', 149.99, 20),
(110,'USB Cable', 'Accessories', 9.99, 400);
```

```sql
insert into orders_table values
(201,'2023-06-01',1,104.98),
(202,'2023-06-05',2,549.98),
(203,'2023-06-10',3,999.99),
(204,'2023-06-15',4,69.98),
(205,'2023-06-20',5,519.98),
(206,'2023-06-25',6,229.98),
(207,'2023-07-02',7,119.97),
(208,'2023-07-12',8,49.98),
(209,'2023-07-18',9,349.98),
(210,'2023-07-22',10,39.98);
```

```sql
insert into  order_details_table values
(301, 201, 101, 1, 999.99),
(302, 204, 103, 2, 49.99),
(303, 202, 101, 1, 499.99),
(304, 205, 101, 10, 49.99),
(305, 201, 105, 5, 999.99),
(306, 204, 101, 7, 49.99),
(307, 205, 110, 4, 19.99),
(308, 202, 109, 12, 499.99),
(309, 205, 105, 1, 19.99),
(310, 203, 103, 19, 199.99),
(311, 205, 110, 12, 29.99),
(312, 201, 106, 25, 999.99),
(313, 203, 107, 6, 199.99),
(314, 201, NULL, 8, 49.99),
(315, 207, 108, 20, 19.99),
(316, 208, 102, 35, 29.99),
(317, 204, 110, 40, 149.99),
(318, 209, 101, 15, 49.99),
(319, 210, 104, 30, 9.99);
```

3. Retrieve all customers who registered in 2023.

```
101     /*3*/
102
103 •   select c.first_name
104     from  customers_table c
105     where c.registration_date>'2021-12-31';
106
```

Result Grid | Filter Rows: | Export:

first_name
John
Jane
Alice
Bob
Charlie
David
Emma
Fiona
George
Hannah

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

```
111     /*4*/
112 •   select p.product_name
113     from product_table p
114     where category='Electronics';
115
116
117
```

Result Grid | Filter Rows: | Export:

product_name
Laptop
Smartphone
Tablet
Monitor
Printer

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

```
116     /*5*/
117 •   select o.customer_id,c.first_name,count(ord.order_id) as total_orders
118     from orders_table o
119     left join customers_table c on c.customer_id=o.customer_id
120     left join order_details_table ord on ord.order_id=o.order_id
121     group by o.customer_id;
122
123     /*6*/
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	total_orders
1	John	4
2	Jane	2
3	Alice	2
4	Bob	3
5	Charlie	4
6	David	0
7	Emma	1
8	Fiona	1
9	George	1
10	Hannah	1

6. Calculate the total sales amount for each product

```
/*6*/
select p.product_id,p.product_name,sum(od.quantity*od.price) as total_sales_amt
from product_table p
join order_details_table od on p.product_id=od.product_id
group by p.product_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	product_name	total_sales_amt
101	Laptop	3099.66
102	Smartphone	1049.65
103	Tablet	3899.79
104	Headphones	299.70
105	Charger	5019.94
106	Keyboard	24999.75
107	Mouse	1199.94
108	Monitor	399.80
109	Printer	5999.88
110	USB Cable	6439.44

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

```
/*7*/

select c.customer_id,c.first_name,c.last_name,o.order_id,o.total_amt
from orders_table o
left join customers_table c on c.customer_id=o.customer_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	order_id	total_amt
1	John	Doe	201	104.98
2	Jane	Smith	202	549.98
3	Alice	Johnson	203	999.99
4	Bob	Brown	204	69.98
5	Charlie	Davis	205	519.98
6	David	Wilson	206	229.98
7	Emma	Thomas	207	119.97
8	Fiona	Garcia	208	49.98
9	George	Martinez	209	349.98
10	Hannah	Rodriguez	210	39.98

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

```
/*8*/

select p.product_id,p.product_name,count(ord.quantity) as total_quantity
from order_details_table ord
left join product_table p on p.product_id=ord.product_id
group by p.product_id;


/*9*/
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	product_name	total_quantity
101	Laptop	5
103	Tablet	2
105	Charger	2
110	USB Cable	3
109	Printer	1
106	Keyboard	1
107	Mouse	1
NULL	NULL	1
108	Monitor	1
102	Smartphone	1
104	Headphones	1

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

```
/*9*/

select ord.Order_detail_ID ,ord.quantity,ord.price
from order_details_table ord
join orders_table o on o.order_id=ord.order_id
where o.customer_id=1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Order_detail_ID	quantity	price
301	1	999.99
305	5	999.99
312	25	999.99
314	8	49.99

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

```
/*10*/

select c.customer_id,o.order_id,o.total_amt
from orders_table o
join customers_table c on c.customer_id=o.customer_id
where o.total_amt>500.00;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cor

customer_id	order_id	total_amt
2	202	549.98
3	203	999.99
5	205	519.98

11. List the products that have never been ordered.

```
/*11*/

select p.product_id,p.product_name
from product_table p
left join order_details_table ord on p.product_id=ord.product_id
where ord.product_id is null;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	product_name

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

```
/*12*/
SELECT o.order_date, p.product_name, SUM(ord.quantity) AS total_quantity
FROM orders_table o
JOIN order_details_table ord ON ord.order_id = o.order_id
JOIN product_table p ON p.product_id = ord.product_id
WHERE o.customer_id = 2
GROUP BY o.order_date, p.product_name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_date	product_name	total_quantity
2023-06-05	Laptop	1
2023-06-05	Printer	12

13. Calculate the average order value for each customer.

```
160     /*13*/
161
162 •   select o.customer_id,avg(ord.quantity*ord.price) as average_value
163     from order_details_table ord
164     left join   orders_table o on o.order_id=ord.order_id
165     group by o.customer_id
166     order by o.customer_id asc ;
167
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	average_value
1	7849.902500
2	3249.935000
3	2499.875000
4	2149.836667
5	239.932500
7	399.800000
8	1049.650000
9	749.850000
10	299.700000

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

```
168     /*14*/
169
170 •   select p.product_id,p.product_name,sum(ord.quantity) as max_quantity
171     from order_details_table ord
172     join product_table p on p.product_id=ord.product_id
173     group by p.product_id
174     order by max_quantity desc
175     limit 1;
176
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_id	product_name	max_quantity
110	USB Cable	56

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

```
177     /*15*/
178
179 •   select o.customer_id,count(o.order_id) as order_id_count
180     from order_details_table ord
181     join orders_table o on o.order_id=ord.order_id
182     group by o.customer_id
183     having order_id_count>1;
184
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	order_id_count
1	4
2	2
3	2
4	3
5	4

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

```
185    /*16*/
186
187    select p.product_id,p.product_name, sum( o.quantity * o.price) as total_amount
188    from order_details_table o
189    left join product_table p on p.product_id=o.product_id
190    group by p.product_id,p.product_name;
191
192
193
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	product_name	total_amount
101	Laptop	3099.66
103	Tablet	3899.79
105	Charger	5019.94
110	USB Cable	6439.44
109	Printer	5999.88
106	Keyboard	24999.75
107	Mouse	1199.94
NULL	NULL	399.92
108	Monitor	399.80
102	Smartphone	1049.65
104	Headphones	299.70

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

```
192    /*17*/
193
194    select o.customer_id,o.order_id,sum( ord.quantity * ord.price) as total_amount
195    from order_details_table ord
196    left join orders_table o on o.order_id = ord.order_id
197    group by o.customer_id,o.order_id
198    order by o.customer_id;
199
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	order_id	total_amount
1	201	31399.61
2	202	6499.87
3	203	4999.75
4	204	6449.51
5	205	959.73
7	207	399.80
8	208	1049.65
9	209	749.85
10	210	299.70

18. Find the average price of products in each category.

```
200    /*18*/
201
202    SELECT product_id, AVG(total_amount) AS average_total_amount
203    FROM (
204        SELECT o.product_id, SUM(o.quantity * o.price) AS total_amount
205        FROM order_details_table o
206        GROUP BY o.product_id
207    ) AS subquery
208    GROUP BY product_id;
209
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	average_total_amount
NULL	399.920000
101	3099.660000
102	1049.650000
103	3899.790000
104	299.700000
105	5019.940000
106	24999.750000
107	1199.940000
108	399.800000
109	5999.880000
110	6439.440000

19. Find all customers who have not placed any orders.

```
/*19*/

select o.customer_id,o.order_id,count(ord.order_id) as total_orders
from order_details_table ord
left join orders_table o on o.order_id=ord.order_id
group by o.order_id,o.customer_id
having total_orders=0;

/*20*/
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	order_id	total_orders

20. List the top 3 products with the highest total sales amount

```
/*20*/

select p.product_name,SUM(ord.quantity * ord.price) AS total_amount
from order_details_table ord
left join product_table p on p.product_id=ord.product_id
group by p.product_id
order by total_amount desc
limit 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product_name	total_amount
Keyboard	24999.75
USB Cable	6439.44
Printer	5999.88

21. Find customers who have placed orders for more than 3 different products.

```
/*21*/

select o.customer_id,o.order_id,count(ord.order_id) as total_orders
from order_details_table ord
left join orders_table o on o.order_id=ord.order_id
group by o.order_id
having total_orders > 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	order_id	total_orders
1	201	4
5	205	4