

SQL and Databases :

Project Report

Customer Metrics

List all customers

```
SELECT * FROM customers;
```

List all orders

```
SELECT * FROM orders;
```

Find customers from New York

```
SELECT customer_name FROM customers WHERE city = 'New York';
```

Count customers per city

```
SELECT city, COUNT(*) FROM customers GROUP BY city;
```

Total revenue generated

```
SELECT SUM(amount) AS total_revenue FROM orders;
```

Average order value

```
SELECT AVG(amount) AS avg_order_value FROM orders;
```

Total spending per customer

```
SELECT c.customer_name, SUM(o.amount) AS total_spent
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id GROUP BY c.customer_name;
```

Customers with no orders

```
SELECT c.customer_name
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
WHERE o.order_id IS NULL;
```

Orders made in October

```
SELECT * FROM orders WHERE MONTH(order_date) = 10;
```

Highest order amount

```
SELECT * FROM orders ORDER BY amount DESC LIMIT 1;
```

Customers who spent more than 200 total

```
SELECT c.customer_name, SUM(o.amount) AS total_spent
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_name
HAVING SUM(o.amount) > 200;
```

Number of orders per customer

```
SELECT customer_id, COUNT(*) AS num_orders  
FROM orders GROUP BY customer_id;
```

Orders over \$150

```
SELECT * FROM orders WHERE amount > 150;
```

First 5 customers by signup date

```
SELECT * FROM customers ORDER BY signup_date LIMIT 5;
```

Last order of each customer (using MAX)

```
SELECT customer_id, MAX(order_date) AS last_order  
FROM orders GROUP BY customer_id;
```

Join customer + order details

```
SELECT c.customer_name, o.order_id, o.amount  
FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id;
```

Total revenue per city

```
SELECT c.city, SUM(o.amount) AS revenue  
FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id
```

Customers who joined after July

```
SELECT * FROM customers WHERE signup_date > '2023-07-01';
```

Monthly order revenue

```
SELECT MONTH(order_date) AS month, SUM(amount)
FROM orders GROUP BY MONTH(order_date);
```

Find duplicate city names

```
SELECT city, COUNT(*)  
FROM customers  
GROUP BY city  
HAVING COUNT(*) > 1;
```

Top 5 highest spending customers

```
SELECT c.customer_name, SUM(o.amount) AS spending
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_name
ORDER BY spending DESC
LIMIT 5;
```

Orders sorted by amount (descending)

```
SELECT * FROM orders ORDER BY amount DESC;
```

Customers and their total number of orders (0 included)

```
SELECT c.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_name;
```

Window function: rank customers by spending

```
SELECT customer_id, SUM(amount) AS total_spent,  
      RANK() OVER (ORDER BY SUM(amount) DESC) AS spend_rank  
FROM orders GROUP BY customer_id;
```

Running total of order amounts (window function)

```
SELECT order_id, order_date, amount,  
      SUM(amount) OVER (ORDER BY order_date) AS running_total FROM orders;
```

THANK YOU