**Assignment No 1**

**Question:** Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.

**Solution:**

CREATE TABLE customers (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR2(100),

email VARCHAR2(100)

);

Insert into customers (customer\_id, customer\_name, email) values

(1, ‘Amar’, amar[@example.com](mailto:john@example.com)'),

Insert into customers (customer\_id, customer\_name, email) values

(2, 'Jane Smith', '[jane@example.com](mailto:jane@example.com)'),

Insert into customers (customer\_id, customer\_name, email) values

(3, 'Johnson', 'john[@example.com](mailto:bob@example.com)'),

Insert into customers (customer\_id, customer\_name, email) values

(4, 'Brown', 'brown@example.com'),

Insert into customers (customer\_id, customer\_name, email) values

(5, 'Charlie', '[charlie@example.com](mailto:charlie@example.com)');

Select \* from customer ;

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE,

order\_amount DECIMAL(10, 2),

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

);

Insert into orders (order\_id, customer\_id, order\_date, order\_amount) values

(101, 1, '2023-01-15', 150.00),

Insert into orders (order\_id, customer\_id, order\_date, order\_amount) values

(102, 2, '2023-02-20', 200.50),

Insert into orders (order\_id, customer\_id, order\_date, order\_amount) values

(103, 1, '2023-03-10', 75.25),

Insert into orders (order\_id, customer\_id, order\_date, order\_amount) values

(104, 3, '2023-04-05', 300.00),

Insert into orders (order\_id, customer\_id, order\_date, order\_amount) values

(105, 2, '2023-05-12', 180.75),

Insert into orders (order\_id, customer\_id, order\_date, order\_amount) values

(106, 4, '2023-06-18', 250.00),

Insert into orders (order\_id, customer\_id, order\_date, order\_amount) values

(107, 5, '2023-07-22', 120.00);

Select \* from ordes;

Select DISTINCT c.customer\_id, c.customer\_name, c.email

From customers c JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_amount > (

SELECT AVG(order\_amount)

From orders

)

Order by c.customer\_id;

—--------------------------------------------------------------

-- Customers with high-value orders (above 200)

SELECT c.customer\_id, c.customer\_name, 'High Value' AS customer\_type, o.order\_amount

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_amount > 200

UNION

Customers with recent orders (in the last 3 months)

SELECT c.customer\_id, c.customer\_name, 'Recent Order' AS customer\_type, o.order\_amount

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

WHERE o.order\_date >= DATE\_SUB(CURDATE(), INTERVAL 3 MONTH)

ORDER BY customer\_id, customer\_type;