

Assignment 3: 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.

Part 1: Utilize a Subquery to Find Customers Who Have Placed Orders Above the Average Order Value

First, we need to calculate the average order value. Then, we use this average order value in a subquery to find customers who have placed orders above this value.

Assume we have the following tables:

1. customers table:

- customer_id
- customer_name
- email

2. orders table:

- order_id
- order_date
- customer_id
- total_amount

SQL QUERY

```
SELECT c.customer_id, c.customer_name, c.email
FROM customers c
WHERE c.customer_id IN (
    SELECT o.customer_id
    FROM orders o
    WHERE o.total_amount > (SELECT AVG(total_amount) FROM orders));
```

In this query:

- The subquery (SELECT AVG(total_amount) FROM orders) calculates the average order value.
- The inner query SELECT o.customer_id FROM orders o WHERE o.total_amount > ... finds the customer IDs for orders with values above the average.
- The outer query retrieves the customer details for these customer IDs.

Part 2: Write a UNION Query to Combine Two SELECT Statements with the Same Number of Columns

- Let's assume we have two different SELECT statements that retrieve customer information based on different criteria. We want to combine their results using the UNION operator.

Full Example:

- First SELECT statement retrieves customers from the customers table who are from a specific city.
- Second SELECT statement retrieves customers from the customers table who have placed orders after a certain date.

QUERY:

-- First SELECT statement: Customers from a specific city

```
SELECT customer_id, customer_name, email  
FROM customers  
WHERE city = 'New York'
```

UNION

-- Second SELECT statement: Customers who have placed orders after a specific date

```
SELECT c.customer_id, c.customer_name, c.email  
FROM customers c  
JOIN orders o ON c.customer_id = o.customer_id  
WHERE o.order_date > '2023-01-01';
```

