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.

SOLUTION:

```
CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(100),
    city VARCHAR(100),
    country VARCHAR(100)
);

INSERT INTO customers (customer_id, customer_name, city, country) VALUES
(1, 'John Doe', 'New York', 'USA'),
(2, 'Jane Smith', 'Los Angeles', 'USA'),
(3, 'Alice Brown', 'New York', 'USA'),
(4, 'Bob Johnson', 'Chicago', 'USA');

CREATE TABLE orders (
    order_id INT PRIMARY KEY,
```

```
    customer_id INT,
    order_date DATE,
    total_amount DECIMAL(10, 2)
);

INSERT INTO orders (order_id, customer_id, order_date,
total_amount) VALUES
(101, 1, '2024-05-01', 100.00),
(102, 3, '2024-05-03', 150.00),
(103, 2, '2024-05-05', 200.00);

SELECT customer_id
FROM orders
GROUP BY customer_id
HAVING AVG(total_amount) > (
SELECT AVG(total_amount)
FROM orders);

SELECT c.customer_id, c.customer_name, c.city
FROM customers c
```

```
WHERE c.customer_id IN (
    SELECT customer_id
    FROM orders
    GROUP BY customer_id
    HAVING AVG(total_amount) > (
        SELECT AVG(total_amount)
        FROM orders
    )
)
UNION
-- Select all customers
SELECT customer_id, customer_name, city
FROM customers;
```

OUTPUT:

```sql
SELECT customer_id
FROM orders
GROUP BY customer_id
HAVING AVG(total_amount) > (
SELECT AVG(total_amount)
FROM orders);
```

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

customer_id
2

```sql
)
UNION
-- Select all customers
SELECT customer_id, customer_name, city
FROM customers;
```

Result Grid | Filter Rows: | Export: | Wra

customer_id	customer_name	city
2	Jane Smith	Los Angeles
1	John Doe	New York
3	Alice Brown	New York
4	Bob Johnson	Chicago