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.

Queries :-
create database Service2;
use Service2;

-- Create the customers table
CREATE TABLE customers (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(100),
    email VARCHAR(100),
    city VARCHAR(50),
    region VARCHAR(50)
);

-- Insert 10 rows of data
INSERT INTO customers (customer_id, customer_name, email, city, region) VALUES
(1, 'John Doe', 'john.doe@example.com', 'New York','Northeast'),
(2, 'Jane Smith', 'jane.smith@example.com', 'Los Angeles','west'),
(3, 'Robert Brown', 'robert.brown@example.com', 'Chicago','Midwest'),
(4, 'Emily Davis', 'emily.davis@example.com', 'Houston','South'),
(5, 'Michael Wilson', 'michael.wilson@example.com', 'Phoenix','West'),
(6, 'Sarah Johnson', 'sarah.johnson@example.com', 'Philadelphia','Northeast'),
(7, 'David Lee', 'david.lee@example.com', 'San Antonio','South'),
(8, 'Laura Martin', 'laura.martin@example.com', 'San Diego','West'),
(9, 'James White', 'james.white@example.com', 'Dallas','South'),
(10, 'Linda Harris', 'linda.harris@example.com', 'San Jose','West');

-- Create the orders table
CREATE TABLE orders (

```sql
    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 10 rows of data into the orders table
INSERT INTO orders (order_id, customer_id, order_date, order_amount)
VALUES
(1, 1, '2024-01-15', 150.00),
(2, 2, '2024-02-20', 200.00),
(3, 1, '2024-03-25', 300.00),
(4, 3, '2024-04-05', 120.00),
(5, 5, '2024-05-10', 450.00),
(6, 6, '2024-06-15', 250.00),
(7, 4, '2024-07-20', 350.00),
(8, 8, '2024-08-25', 400.00),
(9, 1, '2024-09-30', 500.00),
(10, 7, '2024-10-05', 220.00);

-- Average Order values

SELECT
    c.customer_id,
    c.customer_name,
    c.email,
    c.city,
    c.region,
    o.order_id,
    o.order_date,
    o.order_amount
FROM customers c INNER JOIN orders o ON c.customer_id = o.customer_id
WHERE o.order_amount > (SELECT AVG(order_amount) FROM orders);
```

	customer_id	customer_name	email	city	region	order_id	order_date	order_amount
▶	1	John Doe	john.doe@example.com	New York	Northeast	3	2024-03-25	300.00
	5	Michael Wilson	michael.wilson@example.com	Phoenix	West	5	2024-05-10	450.00
	4	Emily Davis	emily.davis@example.com	Houston	South	7	2024-07-20	350.00
	8	Laura Martin	laura.martin@example.com	San Diego	West	8	2024-08-25	400.00
	1	John Doe	john.doe@example.com	New York	Northeast	9	2024-09-30	500.00

-- UNION

SELECT
    customer_id,
    customer_name,
    email,
    city,
    region
FROM customers WHERE region = 'West' UNION
SELECT
    customer_id,
    customer_name,
    email,
    city,
    region
FROM customers WHERE region = 'Northeast';

	customer_id	customer_name	email	city	region
▶	2	Jane Smith	jane.smith@example.com	Los Angeles	west
	5	Michael Wilson	michael.wilson@example.com	Phoenix	West
	8	Laura Martin	laura.martin@example.com	San Diego	West
	10	Linda Harris	linda.harris@example.com	San Jose	West
	1	John Doe	john.doe@example.com	New York	Northeast
	6	Sarah Johnson	sarah.johnson@example.com	Philadelphia	Northeast

-- UNION ALL
SELECT
    customer_id,
    customer_name,
    email,

```sql
    city,
    region
FROM customers WHERE region = 'West'
UNION ALL
SELECT
    customer_id,
    customer_name,
    email,
    city,
    region
FROM customers WHERE region = 'Northeast';
```

	customer_id	customer_name	email	city	region
▶	2	Jane Smith	jane.smith@example.com	Los Angeles	west
	5	Michael Wilson	michael.wilson@example.com	Phoenix	West
	8	Laura Martin	laura.martin@example.com	San Diego	West
	10	Linda Harris	linda.harris@example.com	San Jose	West
	1	John Doe	john.doe@example.com	New York	Northeast
	6	Sarah Johnson	sarah.johnson@example.com	Philadelphia	Northeast

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