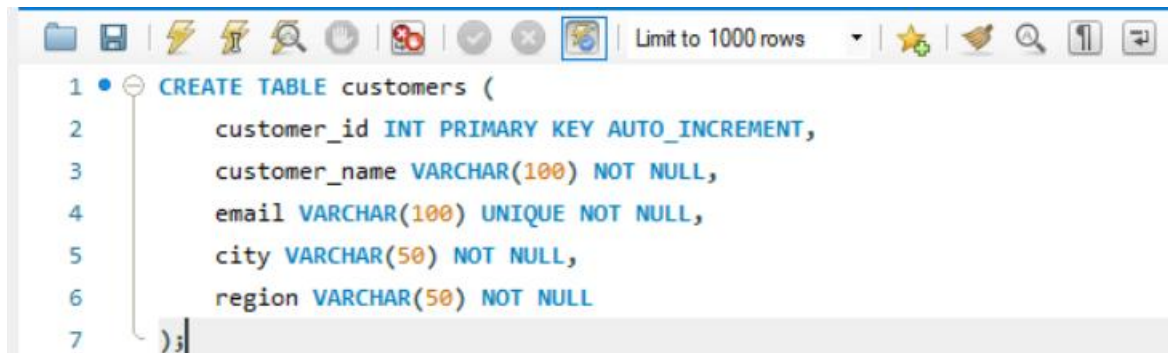


Assignment 3: Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.

Step 1: Create the customers Table.



```
1 CREATE TABLE customers (  
2     customer_id INT PRIMARY KEY AUTO_INCREMENT,  
3     customer_name VARCHAR(100) NOT NULL,  
4     email VARCHAR(100) UNIQUE NOT NULL,  
5     city VARCHAR(50) NOT NULL,  
6     region VARCHAR(50) NOT NULL  
7 );
```

Step 2: Insert Dummy Data into customers.



```
8 INSERT INTO customers (customer_name, email, city, region) VALUES  
9 ('John Doe', 'john.doe@example.com', 'New York', 'East'),  
10 ('Jane Smith', 'jane.smith@example.com', 'Los Angeles', 'West'),  
11 ('Mike Johnson', 'mike.johnson@example.com', 'Chicago', 'Midwest'),  
12 ('Emily Davis', 'emily.davis@example.com', 'New York', 'East'),  
13 ('David Wilson', 'david.wilson@example.com', 'San Francisco', 'West'),  
14 ('Sarah Brown', 'sarah.brown@example.com', 'Houston', 'South'),  
15 ('James White', 'james.white@example.com', 'Los Angeles', 'West'),  
16 ('Olivia Martin', 'olivia.martin@example.com', 'Chicago', 'Midwest'),  
17 ('Daniel Thompson', 'daniel.thompson@example.com', 'New York', 'East'),  
18 ('Sophia Garcia', 'sophia.garcia@example.com', 'Miami', 'South');
```

Step 3: Create the orders Table.

```
19 • CREATE TABLE orders (  
20     order_id INT PRIMARY KEY AUTO_INCREMENT,  
21     customer_id INT,  
22     order_date DATE NOT NULL,  
23     total_amount DECIMAL(10,2) NOT NULL,  
24     FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE CASCADE  
25 );
```

Step 4: Insert Dummy Data into orders.

```
26 • INSERT INTO orders (customer_id, order_date, total_amount) VALUES  
27     (1, '2024-02-01', 150.75),  
28     (2, '2024-02-05', 200.50),  
29     (3, '2024-02-10', 99.99),  
30     (1, '2024-02-15', 300.00),  
31     (4, '2024-02-18', 50.25),  
32     (5, '2024-02-20', 500.00),  
33     (7, '2024-02-22', 75.00);
```

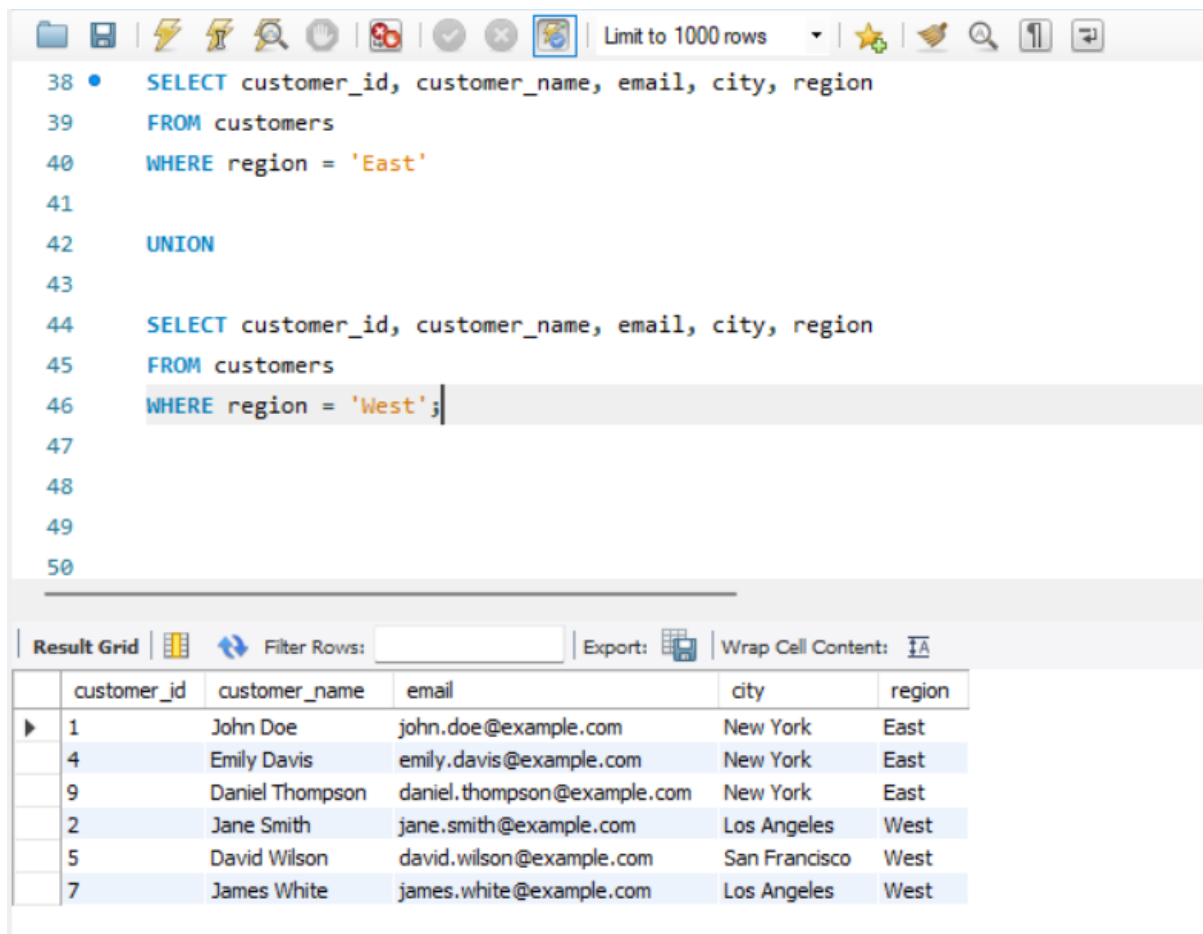
Step 5: Query Customers Who Placed Orders Above the Average Order Value.

```
34 • SELECT DISTINCT c.customer_id, c.customer_name, c.email, o.total_amount  
35 FROM customers c  
36 JOIN orders o ON c.customer_id = o.customer_id  
37 WHERE o.total_amount > (SELECT AVG(total_amount) FROM orders);  
38  
39
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

	customer_id	customer_name	email	total_amount
▶	2	Jane Smith	jane.smith@example.com	200.50
	1	John Doe	john.doe@example.com	300.00
	5	David Wilson	david.wilson@example.com	500.00

Step 6: Combine Two SELECT Statements Using UNION.



The screenshot shows a database query editor with a toolbar at the top. The query text is as follows:

```
38 • SELECT customer_id, customer_name, email, city, region
39 FROM customers
40 WHERE region = 'East'
41
42 UNION
43
44 SELECT customer_id, customer_name, email, city, region
45 FROM customers
46 WHERE region = 'West';
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are shown in a table with 6 columns: customer_id, customer_name, email, city, and region. The table contains 7 rows of data, with the first three rows representing customers from the 'East' region and the last four rows representing customers from the 'West' region.

	customer_id	customer_name	email	city	region
▶	1	John Doe	john.doe@example.com	New York	East
	4	Emily Davis	emily.davis@example.com	New York	East
	9	Daniel Thompson	daniel.thompson@example.com	New York	East
	2	Jane Smith	jane.smith@example.com	Los Angeles	West
	5	David Wilson	david.wilson@example.com	San Francisco	West
	7	James White	james.white@example.com	Los Angeles	West