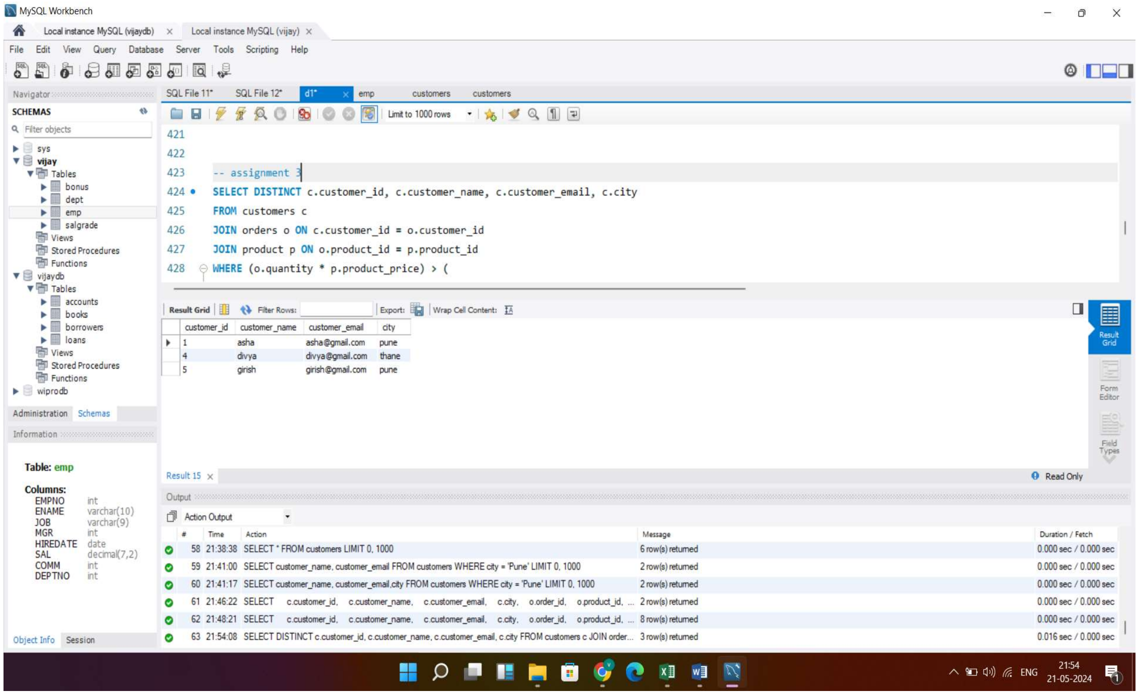
Assignment 3: Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.

ANS:

subquery to find customers who have placed orders above the average order value

```
SELECT DISTINCT c.customer_id, c.customer_name, c.customer_email, c.city

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

JOIN product p ON o.product_id = p.product_id

WHERE (o.quantity * p.product_price) > (

    SELECT AVG(o2.quantity * p2.product_price)

    FROM orders o2

    JOIN product p2 ON o2.product_id = p2.product_id

);
```



```
-- assignment 3
SELECT DISTINCT c.customer_id, c.customer_name, c.customer_email, c.city
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN product p ON o.product_id = p.product_id
WHERE (o.quantity * p.product_price) > (
```

customer_id	customer_name	customer_email	city
1	asha	asha@gmail.com	pune
4	divya	divya@gmail.com	thane
5	girish	girish@gmail.com	pune

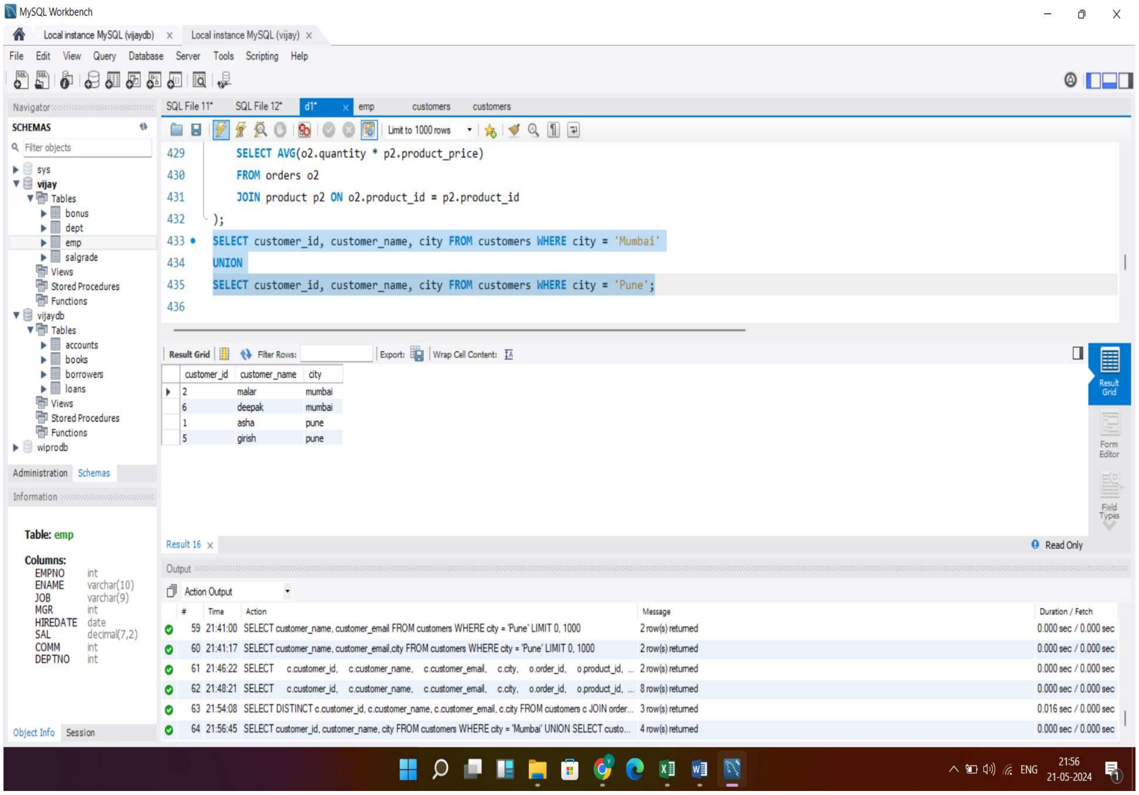
#	Time	Action	Message	Duration / Fetch
58	21:38:38	SELECT * FROM customers LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
59	21:41:00	SELECT customer_name, customer_email FROM customers WHERE city = 'Pune' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
60	21:41:17	SELECT customer_name, customer_email,city FROM customers WHERE city = 'Pune' LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
61	21:46:22	SELECT c.customer_id, c.customer_name, c.customer_email, c.city, o.order_id, o.product_id, ...	2 row(s) returned	0.000 sec / 0.000 sec
62	21:48:21	SELECT c.customer_id, c.customer_name, c.customer_email, c.city, o.order_id, o.product_id, ...	8 row(s) returned	0.000 sec / 0.000 sec
63	21:54:08	SELECT DISTINCT c.customer_id, c.customer_name, c.customer_email, c.city FROM customers c JOIN order...	3 row(s) returned	0.016 sec / 0.000 sec

UNION query to combine two SELECT statements with the same number of columns.

SELECT customer_id, customer_name, city FROM customers WHERE city = 'Mumbai'

UNION

SELECT customer_id, customer_name, city FROM customers WHERE city = 'Pune';



```sql
    SELECT AVG(o2.quantity * p2.product_price)
    FROM orders o2
    JOIN product p2 ON o2.product_id = p2.product_id
);
SELECT customer_id, customer_name, city FROM customers WHERE city = 'Mumbai'
UNION
SELECT customer_id, customer_name, city FROM customers WHERE city = 'Pune';
```

customer_id	customer_name	city
2	malar	mumbai
6	deepak	mumbai
1	asha	pune
5	girish	pune

Table: emp

Columns:

EMPNO	int
ENAME	varchar(10)
JOB	varchar(9)
MGR	int
HIREDATE	date
SAL	decimal(7,2)
COMM	int
DEPTNO	int

```
#   Time      Action                                                                                    Message           Duration / Fetch
59  21:41:00  SELECT customer_name, customer_email FROM customers WHERE city = 'Pune' LIMIT 0, 1000     2 row(s) returned 0.000 sec / 0.000 sec
60  21:41:17  SELECT customer_name, customer_email,city FROM customers WHERE city = 'Pune' LIMIT 0, 1000 2 row(s) returned 0.000 sec / 0.000 sec
61  21:46:22  SELECT    c.customer_id,    c.customer_name,    c.customer_email,    c.city,    o.order_id,    o.product_id, ...  2 row(s) returned 0.000 sec / 0.000 sec
62  21:48:21  SELECT    c.customer_id,    c.customer_name,    c.customer_email,    c.city,    o.order_id,    o.product_id, ...  8 row(s) returned 0.000 sec / 0.000 sec
63  21:54:08  SELECT DISTINCT c.customer_id, c.customer_name, c.customer_email, c.city FROM customers c JOIN order...  3 row(s) returned 0.016 sec / 0.000 sec
64  21:56:45  SELECT customer_id, customer_name, city FROM customers WHERE city = 'Mumbai' UNION SELECT custo...  4 row(s) returned 0.000 sec / 0.000 sec
```