

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
```

```
);
```

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- assignment 3  
424 • SELECT DISTINCT c.customer_id, c.customer_name, c.customer_email, c.city  
425 FROM customers c  
426 JOIN orders o ON c.customer_id = o.customer_id  
427 JOIN product p ON o.product_id = p.product_id  
428 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  
);
```

The Results window displays the following data:

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

The bottom panel shows the Output window with the following message:

```
Result 15 x  
Output  
Action Output  
# 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';

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query that combines two SELECT statements using UNION. The first SELECT statement filters for customers in Mumbai, and the second filters for customers in Pune. The results are displayed in a grid below the editor.

```
429 SELECT AVG(o2.quantity * p2.product_price)
430 FROM orders o2
431 JOIN product p2 ON o2.product_id = p2.product_id
432 );
433 SELECT customer_id, customer_name, city FROM customers WHERE city = 'Mumbai'
434 UNION
435 SELECT customer_id, customer_name, city FROM customers WHERE city = 'Pune';
436
```

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

The bottom panel shows the 'Action Output' tab, which displays the execution details of the query, including the time taken and the number of rows returned for each step.

#	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	21:49:22	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	21:49:21	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	21:54:00	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