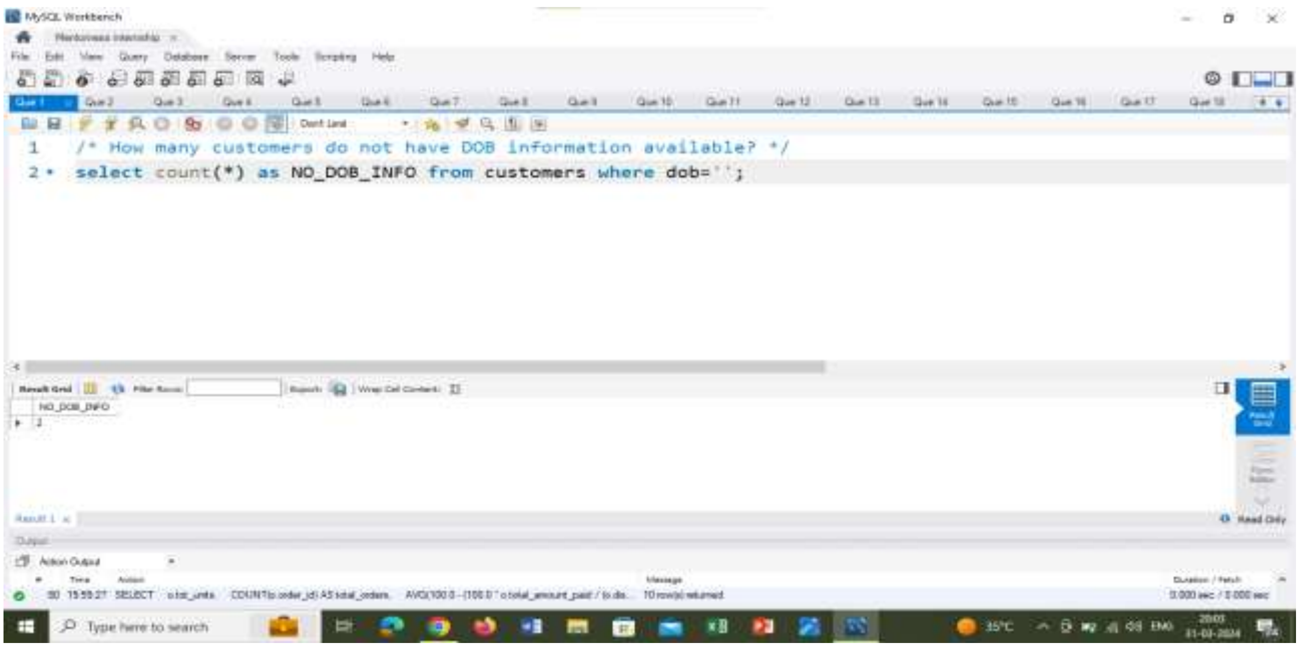
MENTORNESS TASK 2

SALES ANALYSIS PROJECT USING MYSQL

By – V.Akshatha

QUE 1

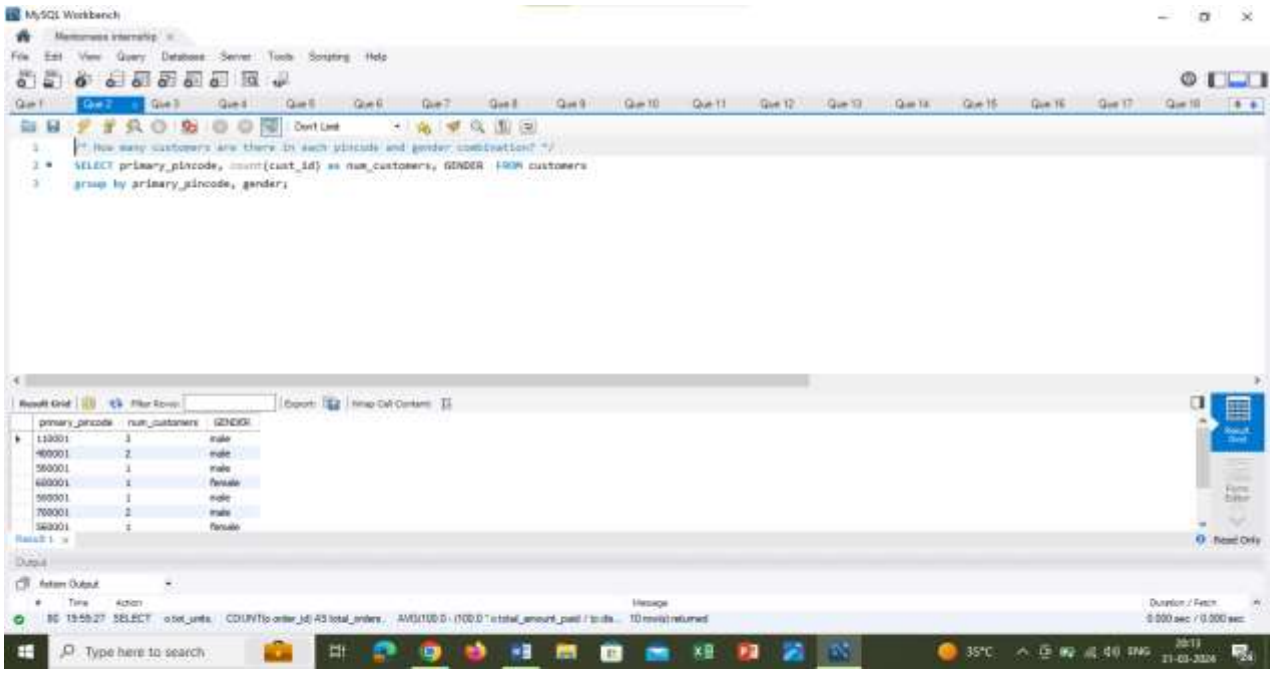


```sql
/* How many customers do not have DOB information available? */
select count(*) as NO_DOB_INFO from customers where dob='';
```

```
NO_DOB_INFO
2
```

```
80  15:55:27  SELECT  o.tot_units  COUNT(o.order_id) AS total_orders  AVG(100.0 - (100.0 * o.total_amount_paid / to.dis...  10 row(s) returned  0.000 sec / 0.000 sec
```

QUE 2

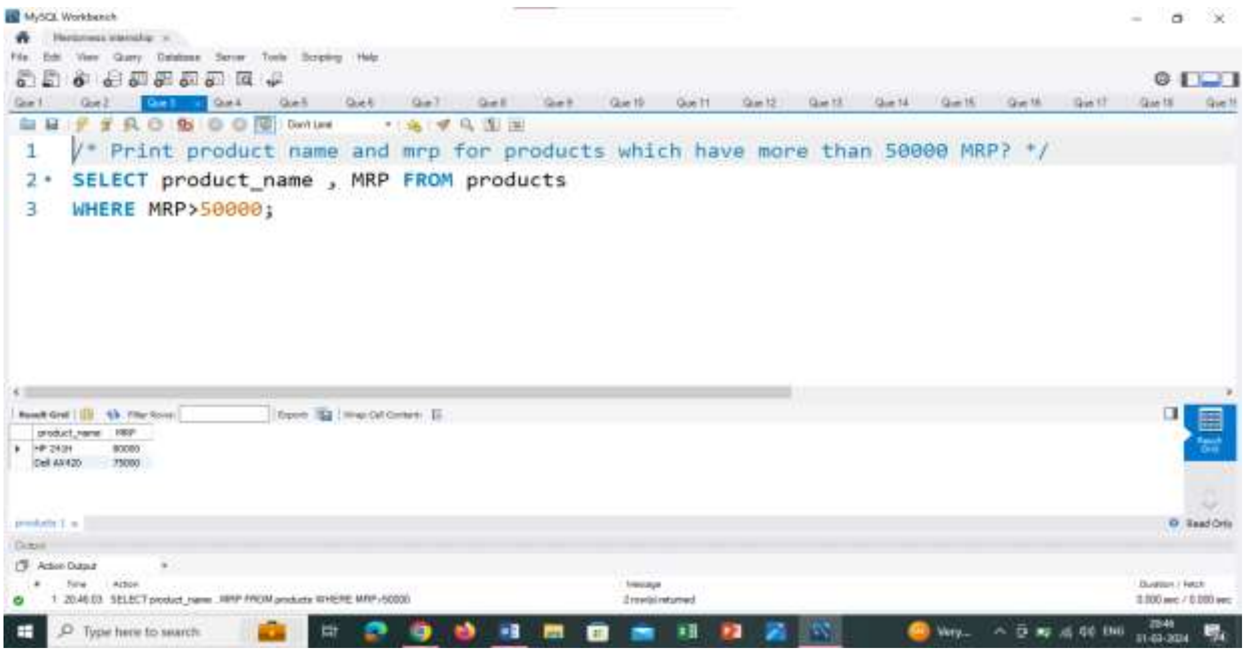


```sql
/* How many customers are there in each pincode and gender combination? */
SELECT primary_pincode, count(cust_id) as num_customers, GENDER  FROM customers
group by primary_pincode, gender;
```

```
primary_pincode  num_customers  GENDER
110001           3              male
400001           2              male
560001           1              male
600001           1              female
500001           1              male
700001           2              male
560001           1              female
```

```
80  15:55:27  SELECT  o.tot_units  COUNT(o.order_id) AS total_orders  AVG(100.0 - (100.0 * o.total_amount_paid / to.dis...  10 row(s) returned  0.000 sec / 0.000 sec
```

QUE 3

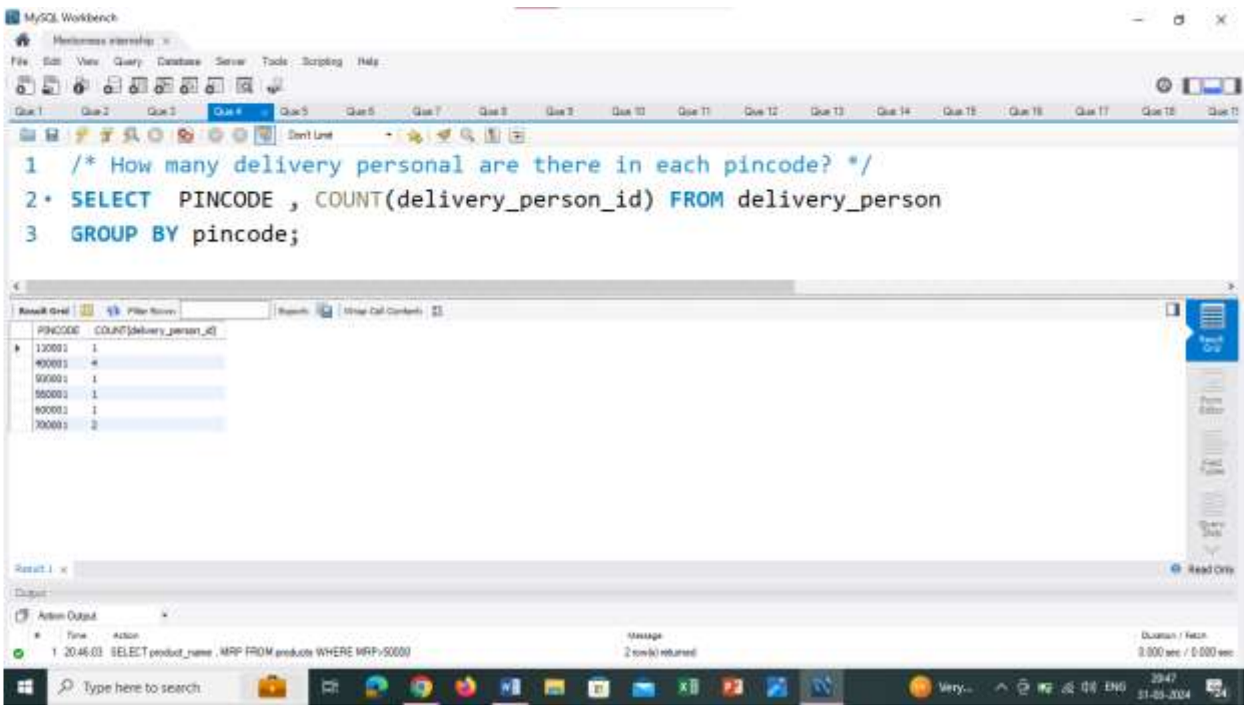


```
/* Print product name and mrp for products which have more than 50000 MRP? */
SELECT product_name , MRP FROM products
WHERE MRP>50000;
```

product_name	MRP
HP 2431H	80000
Dell AX420	75000

```
#  Time      Action                                                      Message             Duration / Fetch
1  20:46:03  SELECT product_name , MRP FROM products WHERE MRP>50000     2 row(s) returned   0.000 sec / 0.000 sec
```

QUE 4

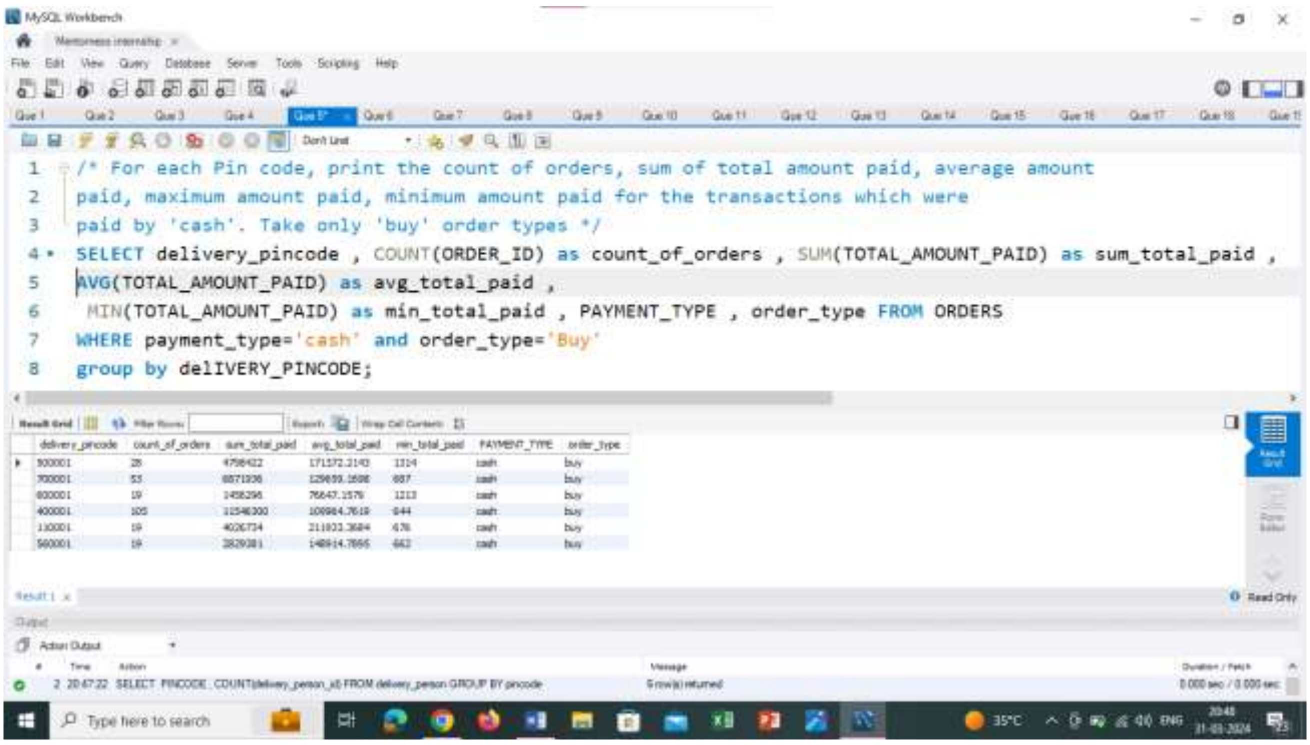


```
/* How many delivery personal are there in each pincode? */
SELECT  PINCODE , COUNT(delivery_person_id) FROM delivery_person
GROUP BY pincode;
```

PINCODE	COUNT(delivery_person_id)
110001	1
400001	4
500001	1
560001	1
600001	1
700001	2

```
#  Time      Action                                                      Message             Duration / Fetch
1  20:46:03  SELECT product_name , MRP FROM products WHERE MRP>50000     2 row(s) returned   0.000 sec / 0.000 sec
```

QUE 5

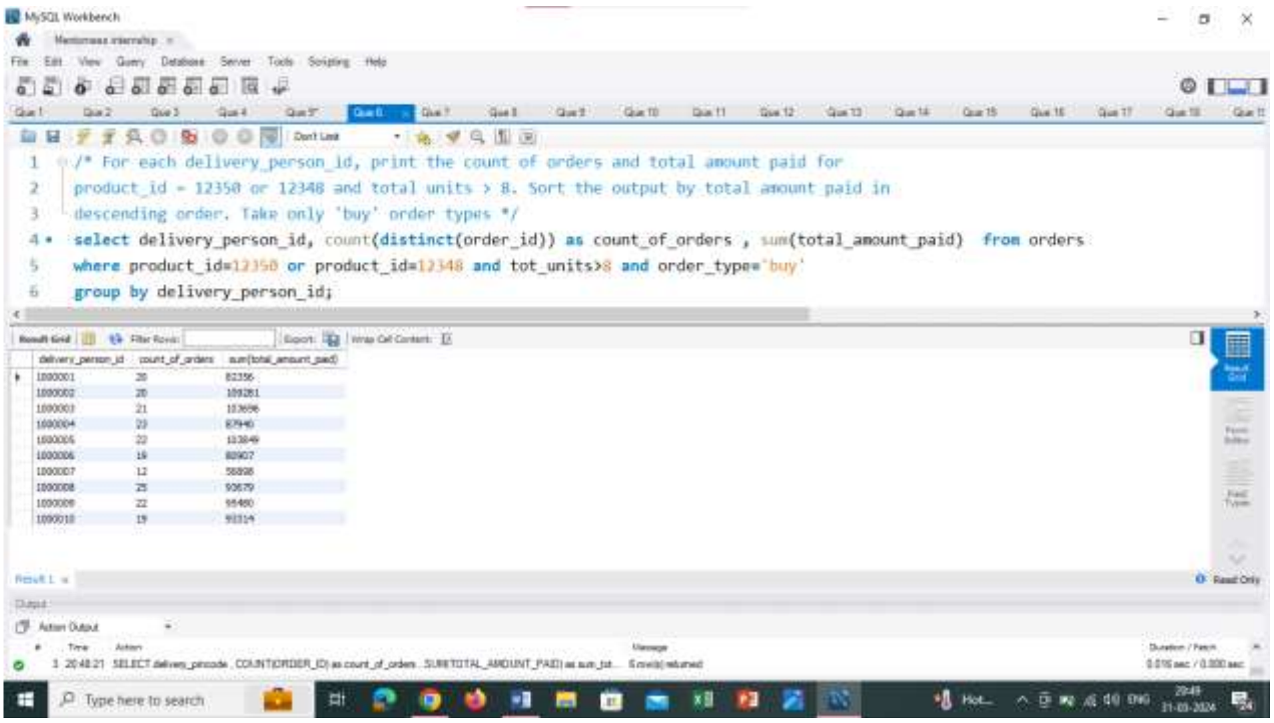


```
/* For each Pin code, print the count of orders, sum of total amount paid, average amount
paid, maximum amount paid, minimum amount paid for the transactions which were
paid by 'cash'. Take only 'buy' order types */
SELECT delivery_pincode , COUNT(ORDER_ID) as count_of_orders , SUM(TOTAL_AMOUNT_PAID) as sum_total_paid ,
AVG(TOTAL_AMOUNT_PAID) as avg_total_paid ,
 MIN(TOTAL_AMOUNT_PAID) as min_total_paid , PAYMENT_TYPE , order_type FROM ORDERS
WHERE payment_type='cash' and order_type='Buy'
group by delIVERY_PINCODE;
```

delivery_pincode	count_of_orders	sum_total_paid	avg_total_paid	min_total_paid	PAYMENT_TYPE	order_type
500001	28	4798422	171372.2143	1314	cash	buy
700001	53	6871936	129659.1698	687	cash	buy
600001	19	1456296	76647.1579	1213	cash	buy
400001	105	11546300	109964.7619	644	cash	buy
110001	19	4026734	211933.3684	676	cash	buy
560001	19	2829381	148914.7895	662	cash	buy

```
2  20:47:22  SELECT PINCODE , COUNT(delivery_person_id) FROM delivery_person GROUP BY pincode    5 row(s) returned    0.000 sec / 0.000 sec
```

QUE 6

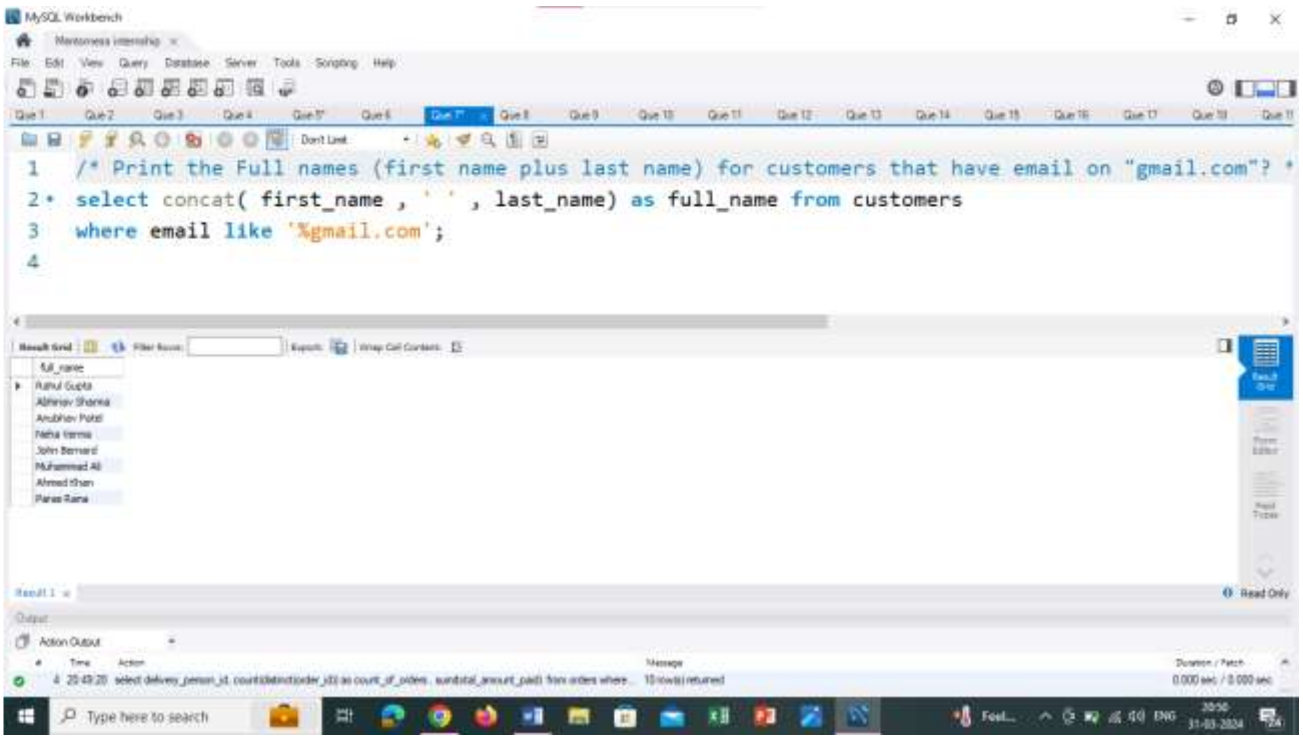


```
/* For each delivery_person_id, print the count of orders and total amount paid for
product_id = 12350 or 12348 and total units > 8. Sort the output by total amount paid in
descending order. Take only 'buy' order types */
select delivery_person_id, count(distinct(order_id)) as count_of_orders , sum(total_amount_paid)  from orders
where product_id=12350 or product_id=12348 and tot_units>8 and order_type='buy'
group by delivery_person_id;
```

delivery_person_id	count_of_orders	sum(total_amount_paid)
1000001	20	82356
1000002	20	109281
1000003	21	113696
1000004	23	87940
1000005	22	113849
1000006	16	80907
1000007	12	56898
1000008	25	93679
1000009	22	95480
1000010	19	93314

```
3  20:48:21  SELECT delivery_pincode , COUNT(ORDER_ID) as count_of_orders , SUM(TOTAL_AMOUNT_PAID) as sum_tot...    6 row(s) returned    0.016 sec / 0.000 sec
```

QUE 7

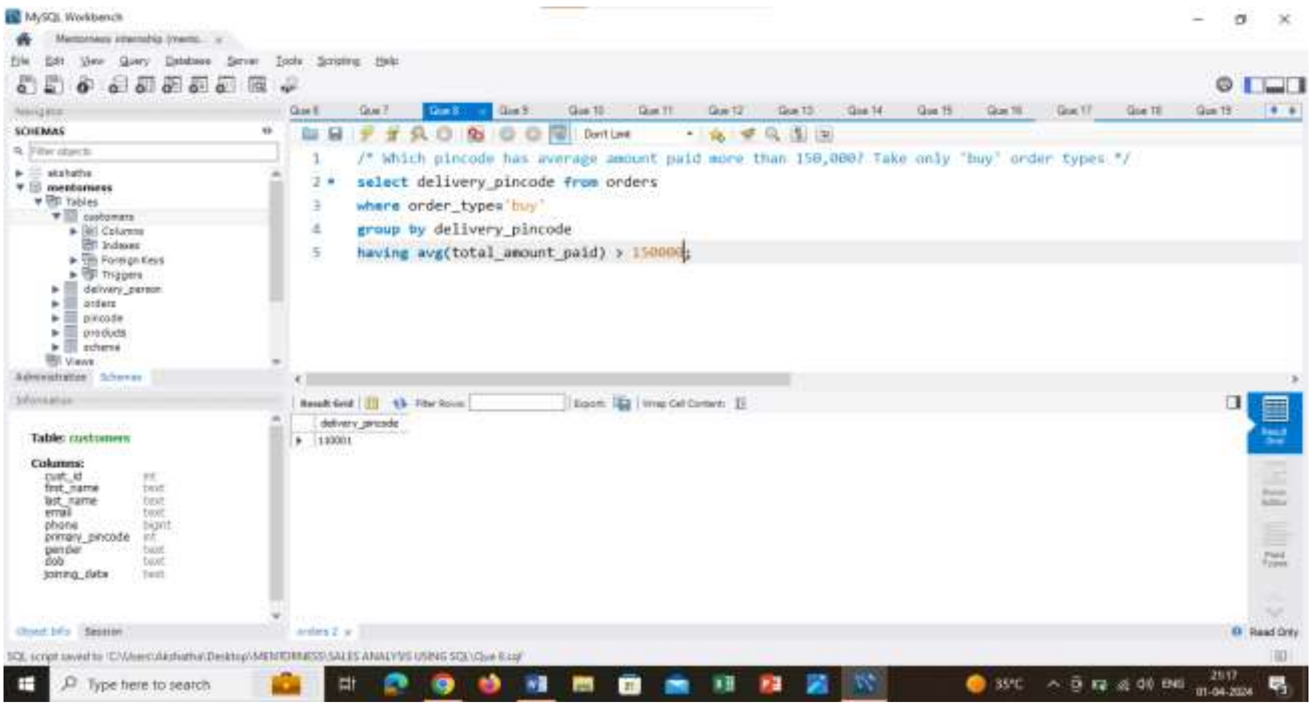


```
/* Print the Full names (first name plus last name) for customers that have email on "gmail.com"? *
select concat( first_name , ' ' , last_name) as full_name from customers
where email like '%gmail.com';
```

```
full_name
Rahul Gupta
Abhinav Sharma
Anubhav Patel
Neha Verma
John Bernard
Muhammad Ali
Ahmed Khan
Paras Raina
```

```
#  Time      Action                                                                                                          Message
4  20:49:20  select delivery_person_id, count(distinct(order_id)) as count_of_orders, sum(total_amount_paid) from orders where ...  10 row(s) returned
```

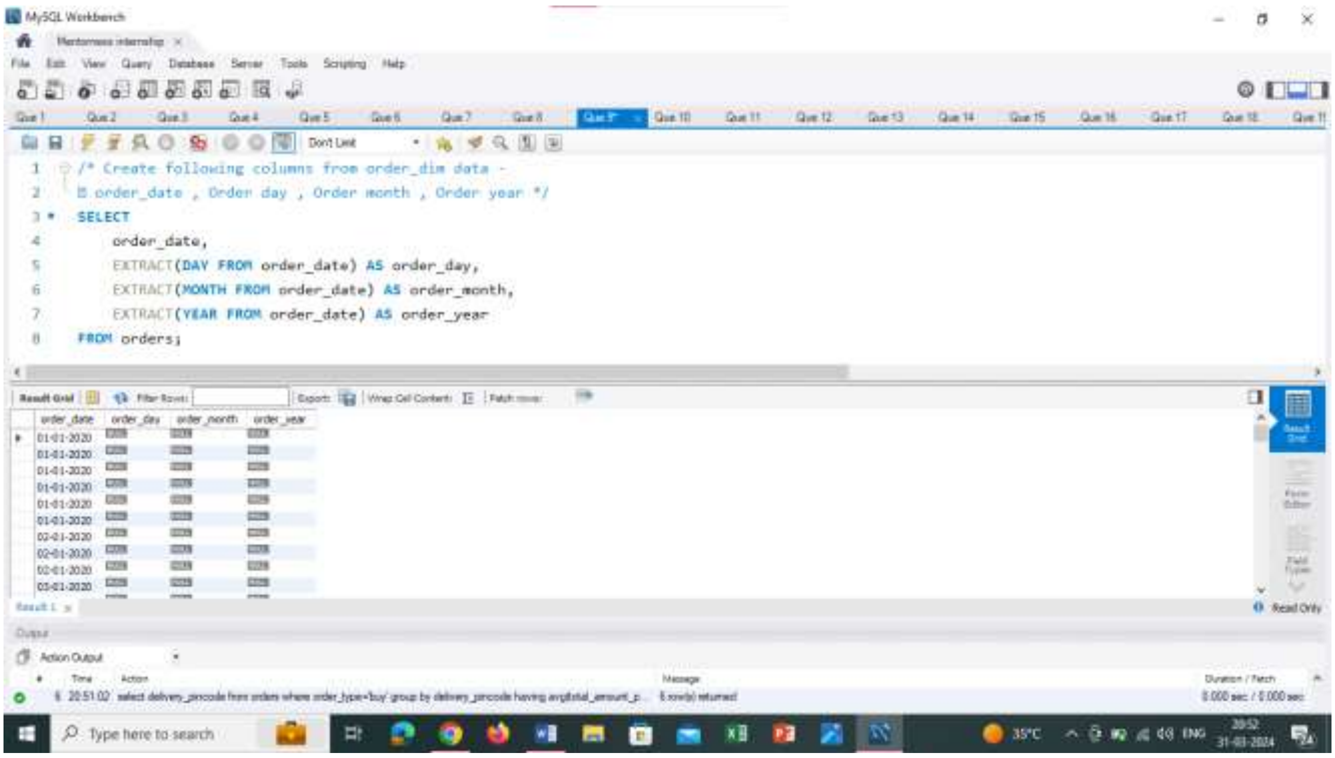
QUE 8



```
/* Which pincode has average amount paid more than 150,000? Take only 'buy' order types */
select delivery_pincode from orders
where order_type='buy'
group by delivery_pincode
having avg(total_amount_paid) > 150000;
```

```
delivery_pincode
110001
```

QUE 9

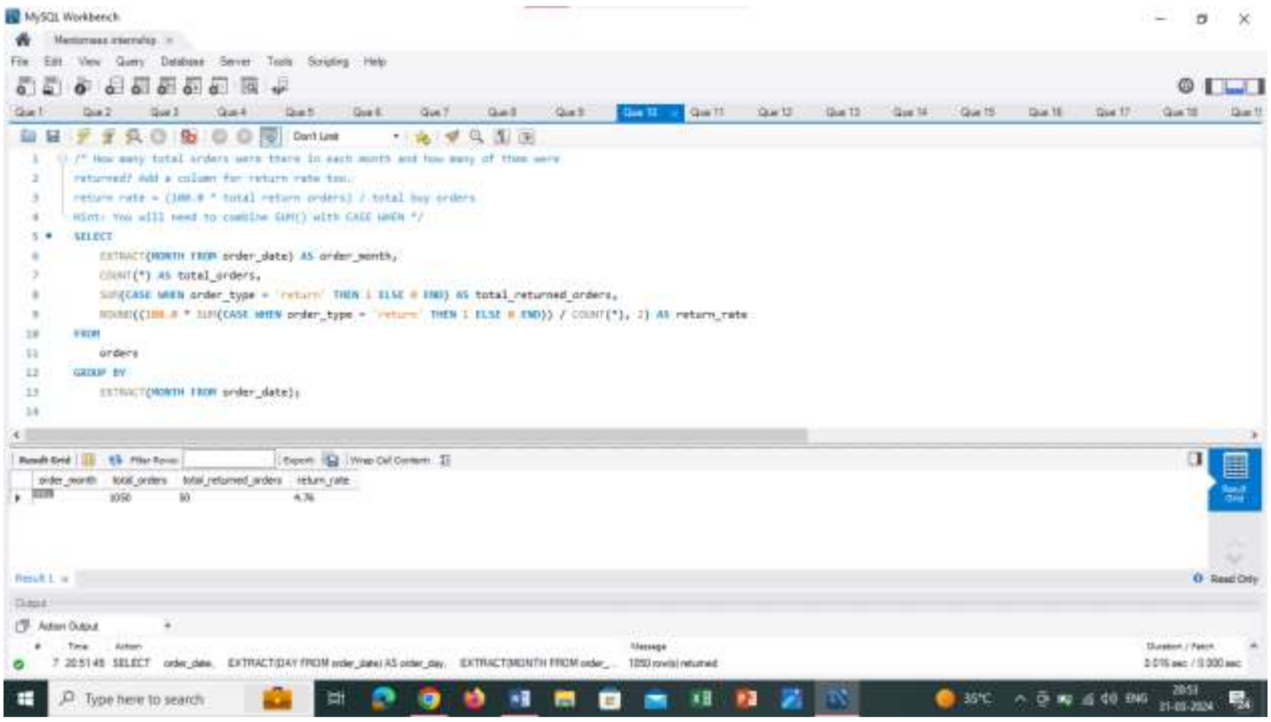


```sql
/* Create following columns from order_dim data -
order_date , Order day , Order month , Order year */
SELECT
    order_date,
    EXTRACT(DAY FROM order_date) AS order_day,
    EXTRACT(MONTH FROM order_date) AS order_month,
    EXTRACT(YEAR FROM order_date) AS order_year
FROM orders;
```

```
order_date   order_day  order_month  order_year
01-01-2020   NULL       NULL         NULL
01-01-2020   NULL       NULL         NULL
01-01-2020   NULL       NULL         NULL
01-01-2020   NULL       NULL         NULL
01-01-2020   NULL       NULL         NULL
01-01-2020   NULL       NULL         NULL
02-01-2020   NULL       NULL         NULL
02-01-2020   NULL       NULL         NULL
02-01-2020   NULL       NULL         NULL
03-01-2020   NULL       NULL         NULL
```

```
#  Time      Action                                                                                              Message              Duration / Fetch
6  20:51:02  select delivery_pincode from orders where order_type='buy' group by delivery_pincode having avg(total_amount_p...  6 row(s) returned  0.000 sec / 0.000 sec
```

QUE 10

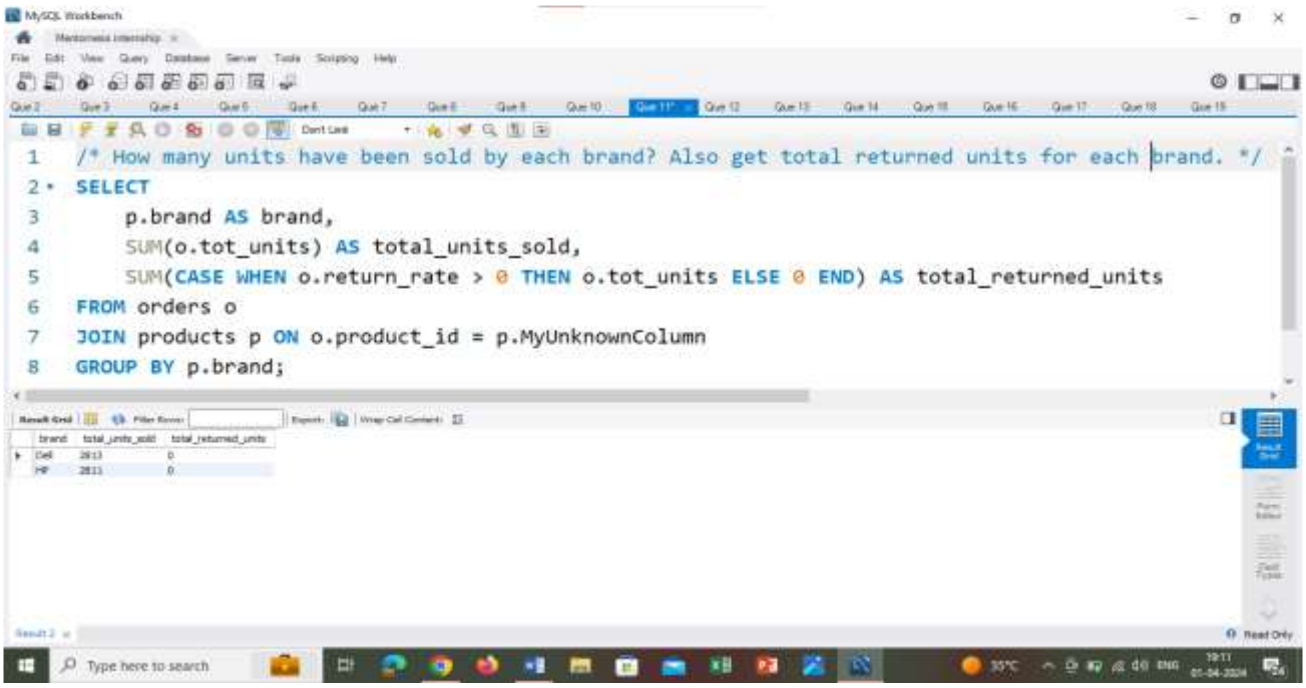


```sql
/* How many total orders were there in each month and how many of them were
returned? Add a column for return rate too.
return rate = (100.0 * total return orders) / total buy orders
Hint: You will need to combine SUM() with CASE WHEN */
SELECT
    EXTRACT(MONTH FROM order_date) AS order_month,
    COUNT(*) AS total_orders,
    SUM(CASE WHEN order_type = 'return' THEN 1 ELSE 0 END) AS total_returned_orders,
    ROUND((100.0 * SUM(CASE WHEN order_type = 'return' THEN 1 ELSE 0 END)) / COUNT(*), 2) AS return_rate
FROM
    orders
GROUP BY
    EXTRACT(MONTH FROM order_date);
```

```
order_month  total_orders  total_returned_orders  return_rate
NULL         1050          50                     4.76
```

```
#  Time      Action                                                                                 Message                 Duration / Fetch
7  20:51:45  SELECT   order_date,   EXTRACT(DAY FROM order_date) AS order_day,   EXTRACT(MONTH FROM order_...  1050 row(s) returned  0.016 sec / 0.000 sec
```

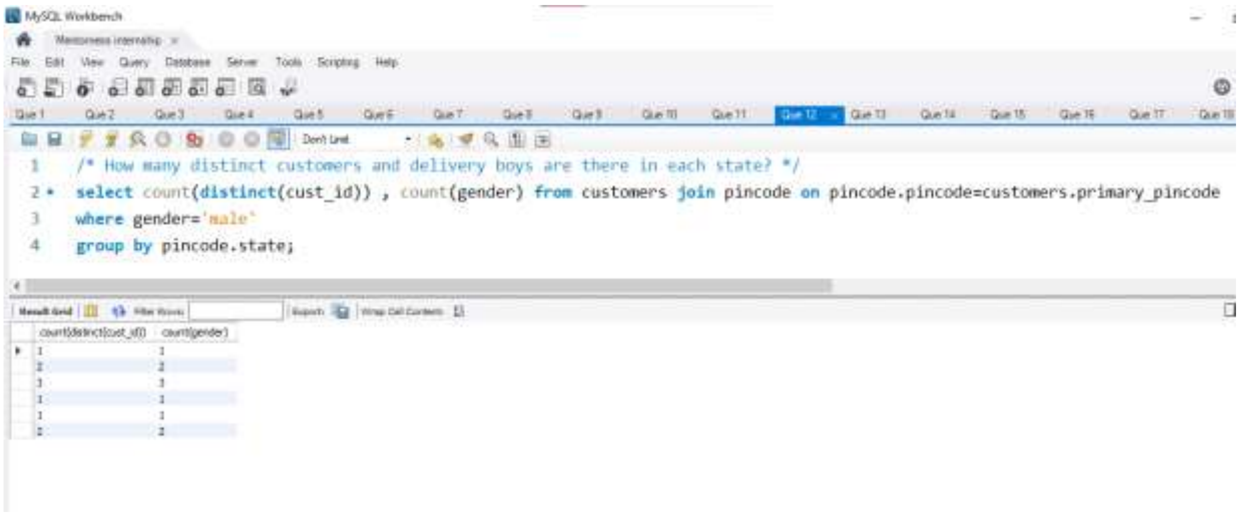
QUE 11



```
/* How many units have been sold by each brand? Also get total returned units for each brand. */
SELECT
    p.brand AS brand,
    SUM(o.tot_units) AS total_units_sold,
    SUM(CASE WHEN o.return_rate > 0 THEN o.tot_units ELSE 0 END) AS total_returned_units
FROM orders o
JOIN products p ON o.product_id = p.MyUnknownColumn
GROUP BY p.brand;
```

brand	total_units_sold	total_returned_units
Dell	2813	0
HP	2811	0

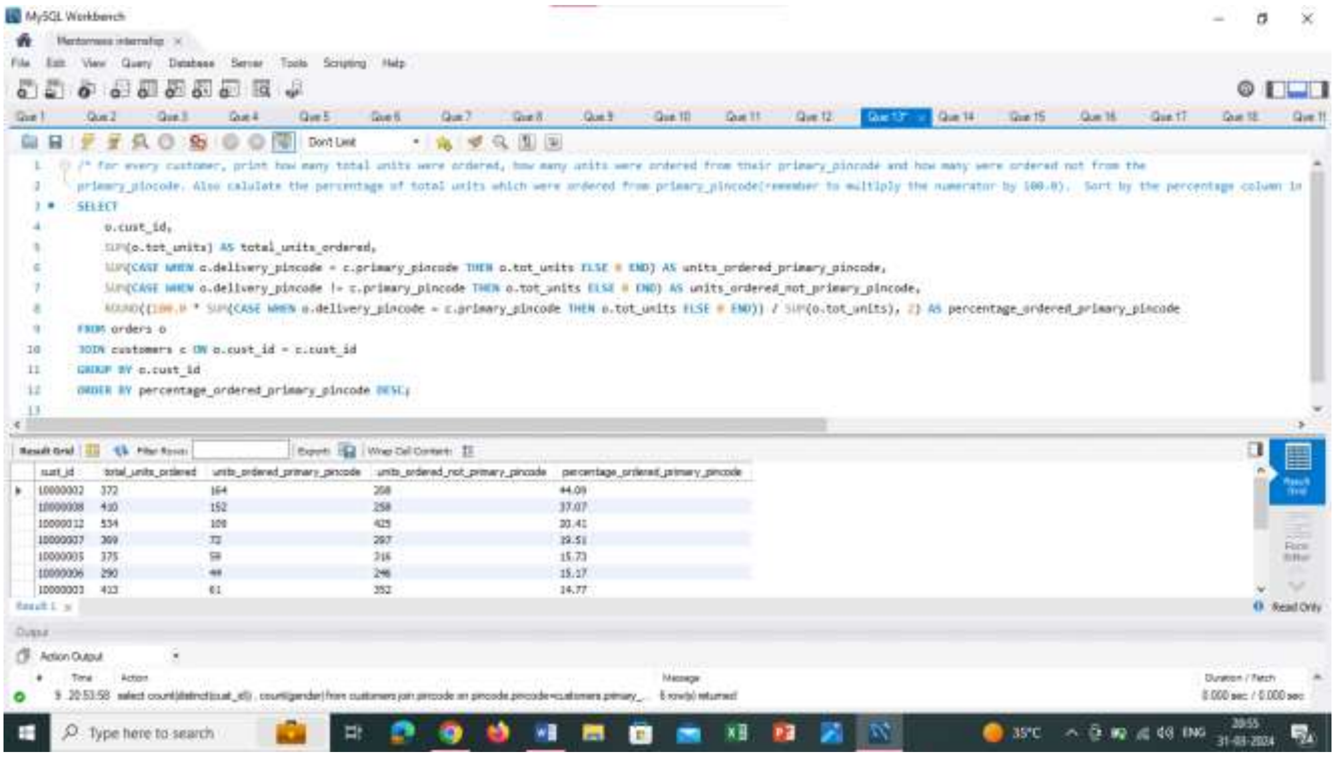
QUE 12



```
/* How many distinct customers and delivery boys are there in each state? */
select count(distinct(cust_id)) , count(gender) from customers join pincode on pincode.pincode=customers.primary_pincode
where gender='male'
group by pincode.state;
```

count(distinct(cust_id))	count(gender)
1	1
2	2
1	1
1	1
1	1
2	2

QUE 13

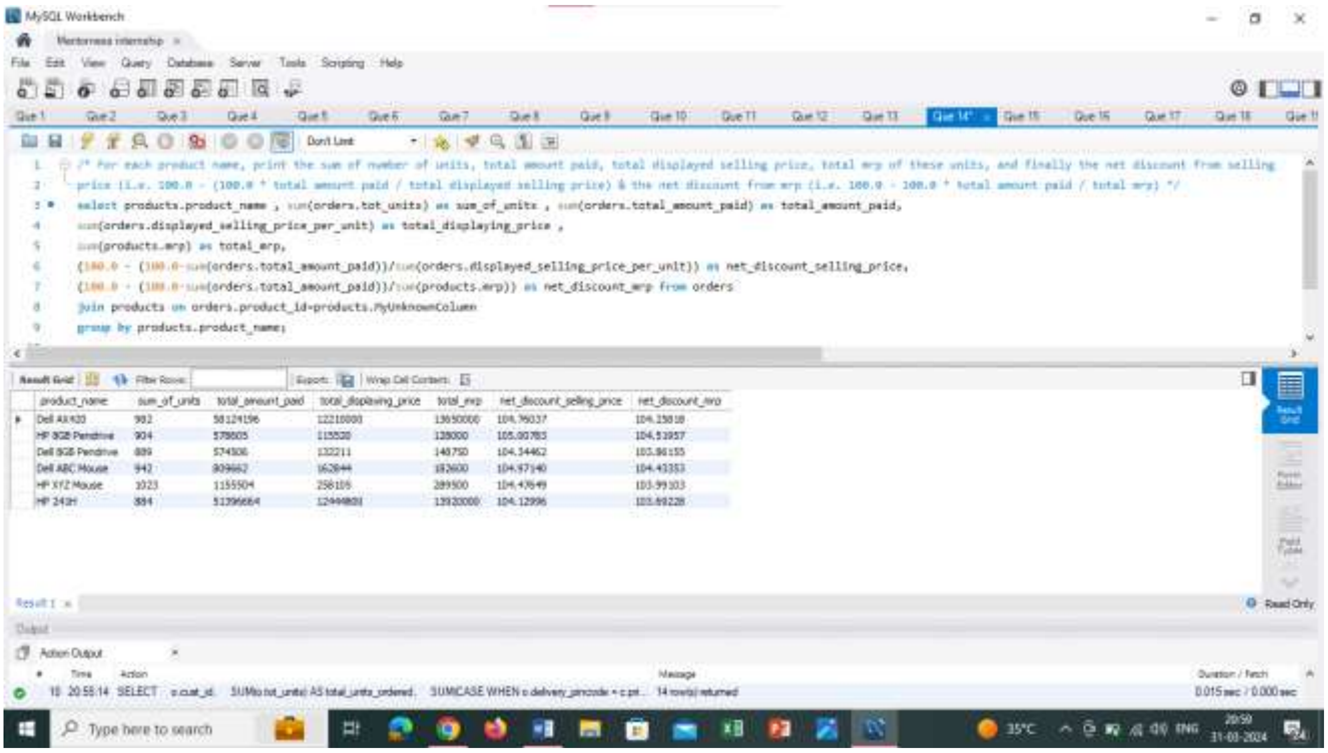


```sql
/* For every customer, print how many total units were ordered, how many units were ordered from their primary_pincode and how many were ordered not from the
primary_pincode. Also calulate the percentage of total units which were ordered from primary_pincode(remember to multiply the numerator by 100.0). Sort by the percentage column in
SELECT
    o.cust_id,
    SUM(o.tot_units) AS total_units_ordered,
    SUM(CASE WHEN o.delivery_pincode = c.primary_pincode THEN o.tot_units ELSE 0 END) AS units_ordered_primary_pincode,
    SUM(CASE WHEN o.delivery_pincode != c.primary_pincode THEN o.tot_units ELSE 0 END) AS units_ordered_not_primary_pincode,
    ROUND((100.0 * SUM(CASE WHEN o.delivery_pincode = c.primary_pincode THEN o.tot_units ELSE 0 END)) / SUM(o.tot_units), 2) AS percentage_ordered_primary_pincode
FROM orders o
JOIN customers c ON o.cust_id = c.cust_id
GROUP BY o.cust_id
ORDER BY percentage_ordered_primary_pincode DESC;
```

cust_id	total_units_ordered	units_ordered_primary_pincode	units_ordered_not_primary_pincode	percentage_ordered_primary_pincode
10000002	372	164	208	44.09
10000008	410	152	258	37.07
10000012	534	109	425	20.41
10000007	369	72	297	19.51
10000005	375	59	316	15.73
10000006	290	44	246	15.17
10000003	413	61	352	14.77

```
9  20:53:58  select count(distinct(cust_id)) , count(gender) from customers join pincode on pincode.pincode=customers.primary_...  6 row(s) returned  0.000 sec / 0.000 sec
```

QUE 14

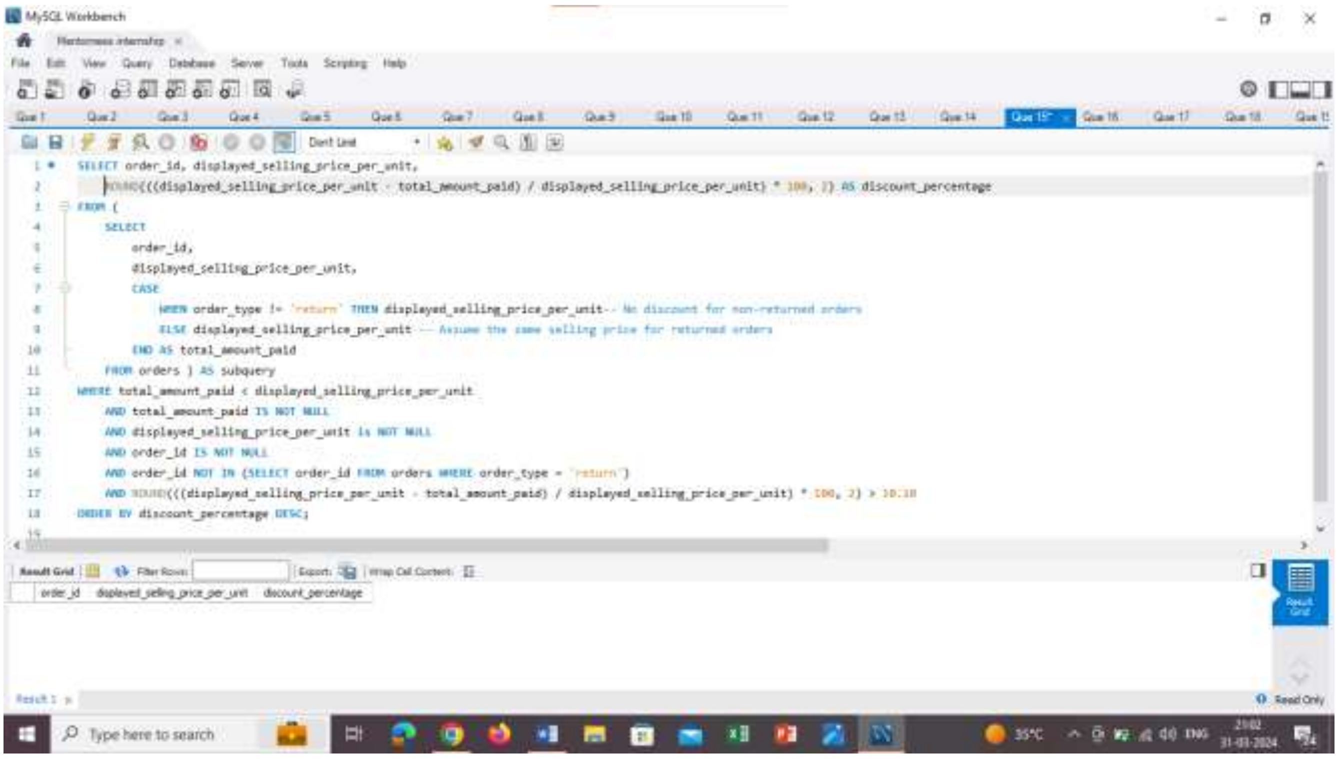


```sql
/* For each product name, print the sum of number of units, total amount paid, total displayed selling price, total mrp of these units, and finally the net discount from selling
price (i.e. 100.0 - (100.0 * total amount paid / total displayed selling price) & the net discount from mrp (i.e. 100.0 - 100.0 * total amount paid / total mrp) */
select products.product_name , sum(orders.tot_units) as sum_of_units , sum(orders.total_amount_paid) as total_amount_paid,
sum(orders.displayed_selling_price_per_unit) as total_displaying_price ,
sum(products.mrp) as total_mrp,
(100.0 - (100.0-sum(orders.total_amount_paid))/sum(orders.displayed_selling_price_per_unit)) as net_discount_selling_price,
(100.0 - (100.0-sum(orders.total_amount_paid))/sum(products.mrp)) as net_discount_mrp from orders
join products on orders.product_id=products.MyUnknownColumn
group by products.product_name;
```

product_name	sum_of_units	total_amount_paid	total_displaying_price	total_mrp	net_discount_selling_price	net_discount_mrp
Dell AX420	982	58124196	12210000	13650000	104.76037	104.25818
HP 8GB Pendrive	904	578605	115520	128000	105.00783	104.51957
Dell 8GB Pendrive	889	574306	122211	148750	104.34462	103.86155
Dell ABC Mouse	942	809662	162844	182600	104.97140	104.43353
HP XYZ Mouse	1023	1155504	258105	289500	104.47649	103.99103
HP 240H	884	51396664	12444800	13920000	104.12996	103.69228

```
10  20:55:14  SELECT  o.cust_id,  SUM(o.tot_units) AS total_units_ordered,  SUM(CASE WHEN o.delivery_pincode = c.pri...  14 row(s) returned  0.015 sec / 0.000 sec
```

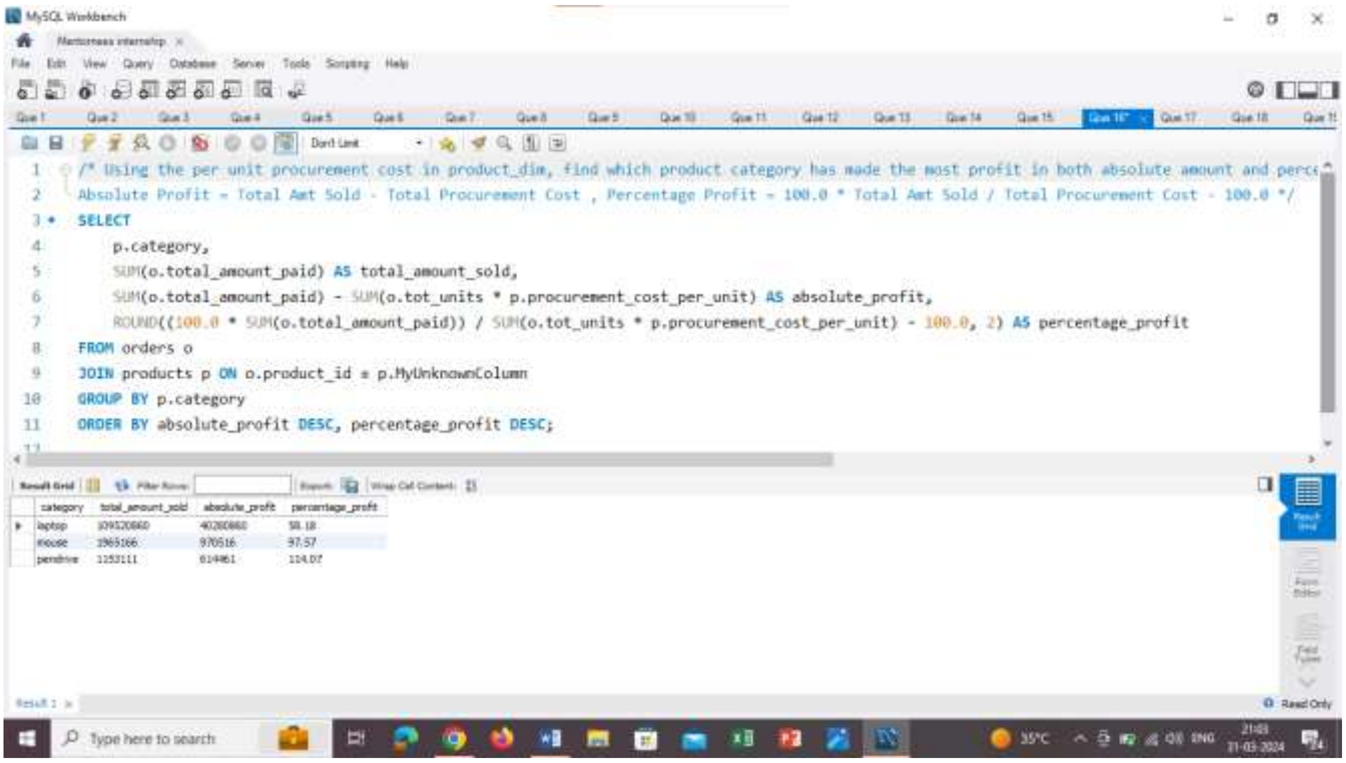
QUE 15



```sql
SELECT order_id, displayed_selling_price_per_unit,
    ROUND(((displayed_selling_price_per_unit - total_amount_paid) / displayed_selling_price_per_unit) * 100, 2) AS discount_percentage
FROM (
    SELECT
        order_id,
        displayed_selling_price_per_unit,
        CASE
            WHEN order_type != 'return' THEN displayed_selling_price_per_unit-- No discount for non-returned orders
            ELSE displayed_selling_price_per_unit -- Assume the same selling price for returned orders
        END AS total_amount_paid
    FROM orders ) AS subquery
WHERE total_amount_paid < displayed_selling_price_per_unit
    AND total_amount_paid IS NOT NULL
    AND displayed_selling_price_per_unit is NOT NULL
    AND order_id IS NOT NULL
    AND order_id NOT IN (SELECT order_id FROM orders WHERE order_type = 'return')
    AND ROUND(((displayed_selling_price_per_unit - total_amount_paid) / displayed_selling_price_per_unit) * 100, 2) > 10.10
ORDER BY discount_percentage DESC;
```

order_id	displayed_selling_price_per_unit	discount_percentage

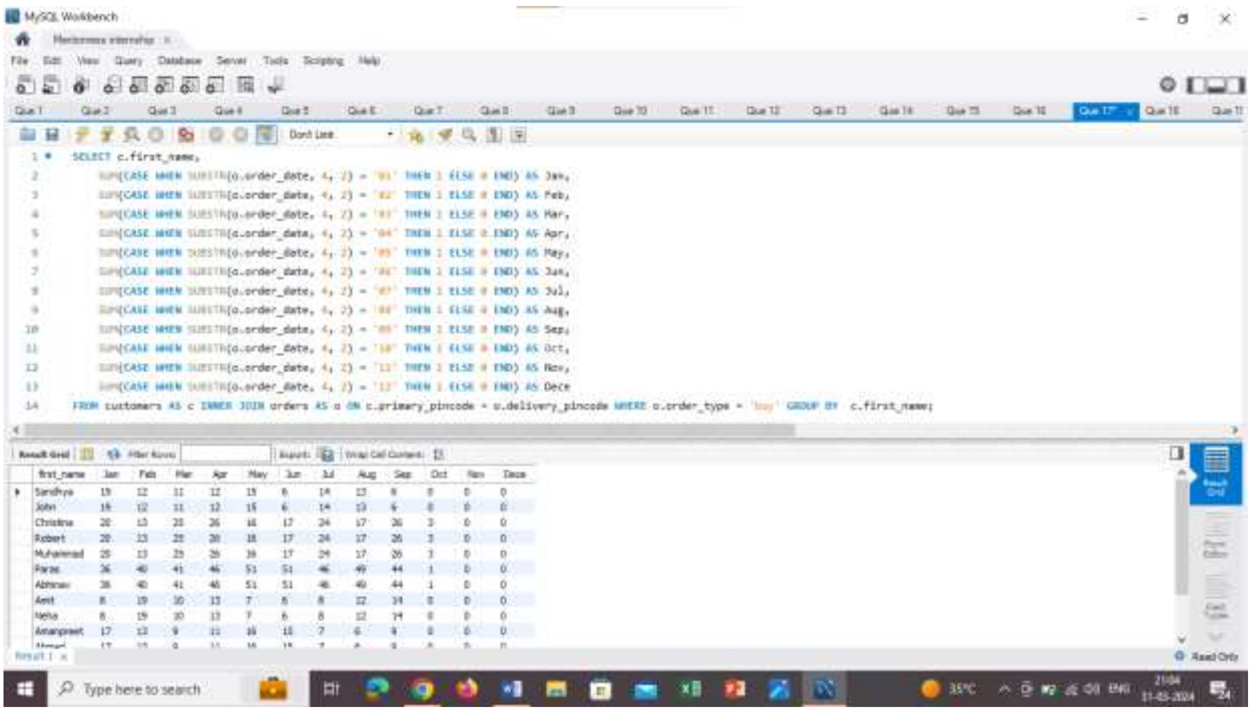
QUE 16



```sql
/* Using the per unit procurement cost in product_dim, find which product category has made the most profit in both absolute amount and perce
Absolute Profit = Total Amt Sold - Total Procurement Cost , Percentage Profit = 100.0 * Total Amt Sold / Total Procurement Cost - 100.0 */
SELECT
    p.category,
    SUM(o.total_amount_paid) AS total_amount_sold,
    SUM(o.total_amount_paid) - SUM(o.tot_units * p.procurement_cost_per_unit) AS absolute_profit,
    ROUND((100.0 * SUM(o.total_amount_paid)) / SUM(o.tot_units * p.procurement_cost_per_unit) - 100.0, 2) AS percentage_profit
FROM orders o
JOIN products p ON o.product_id = p.MyUnknownColumn
GROUP BY p.category
ORDER BY absolute_profit DESC, percentage_profit DESC;
```

category	total_amount_sold	absolute_profit	percentage_profit
laptop	109520860	40200860	58.18
mouse	1965166	970516	97.57
pendrive	1253111	614461	114.07

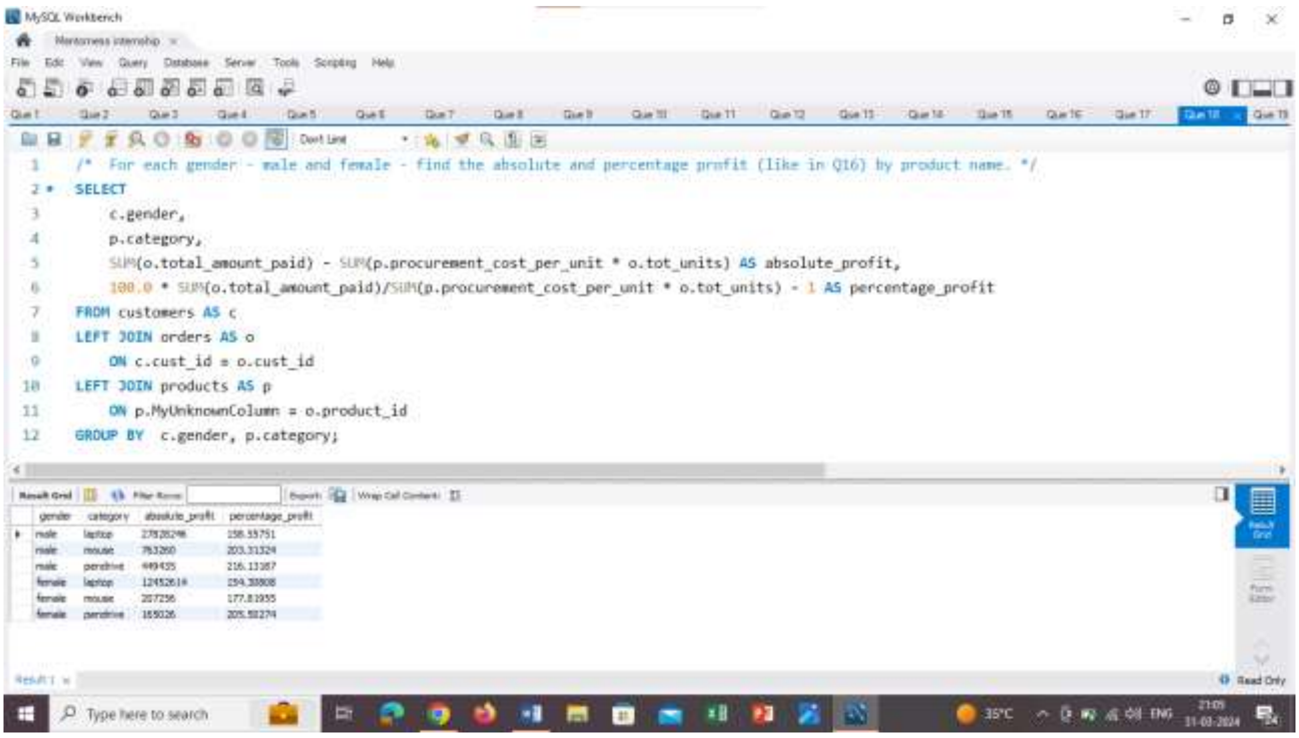
QUE 17



```sql
SELECT c.first_name,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '01' THEN 1 ELSE 0 END) AS Jan,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '02' THEN 1 ELSE 0 END) AS Feb,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '03' THEN 1 ELSE 0 END) AS Mar,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '04' THEN 1 ELSE 0 END) AS Apr,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '05' THEN 1 ELSE 0 END) AS May,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '06' THEN 1 ELSE 0 END) AS Jun,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '07' THEN 1 ELSE 0 END) AS Jul,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '08' THEN 1 ELSE 0 END) AS Aug,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '09' THEN 1 ELSE 0 END) AS Sep,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '10' THEN 1 ELSE 0 END) AS Oct,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '11' THEN 1 ELSE 0 END) AS Nov,
    SUM(CASE WHEN SUBSTR(o.order_date, 4, 2) = '12' THEN 1 ELSE 0 END) AS Dece
FROM customers AS c INNER JOIN orders AS o ON c.primary_pincode = o.delivery_pincode WHERE o.order_type = 'buy' GROUP BY  c.first_name;
```

first_name	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dece
Sandhya	19	12	11	12	15	6	14	13	8	0	0	0
John	18	12	11	12	15	6	14	13	6	0	0	0
Christina	20	13	20	26	18	17	24	17	26	3	0	0
Robert	20	13	25	20	18	17	24	17	26	3	0	0
Muhammad	25	13	25	26	16	17	24	17	26	3	0	0
Paras	36	40	41	46	51	51	46	49	44	1	0	0
Abhinav	36	40	41	46	51	51	46	49	44	1	0	0
Amit	8	19	10	13	7	6	8	12	14	0	0	0
Neha	8	19	10	13	7	6	8	12	14	0	0	0
Amanpreet	17	13	9	11	16	15	7	6	9	0	0	0

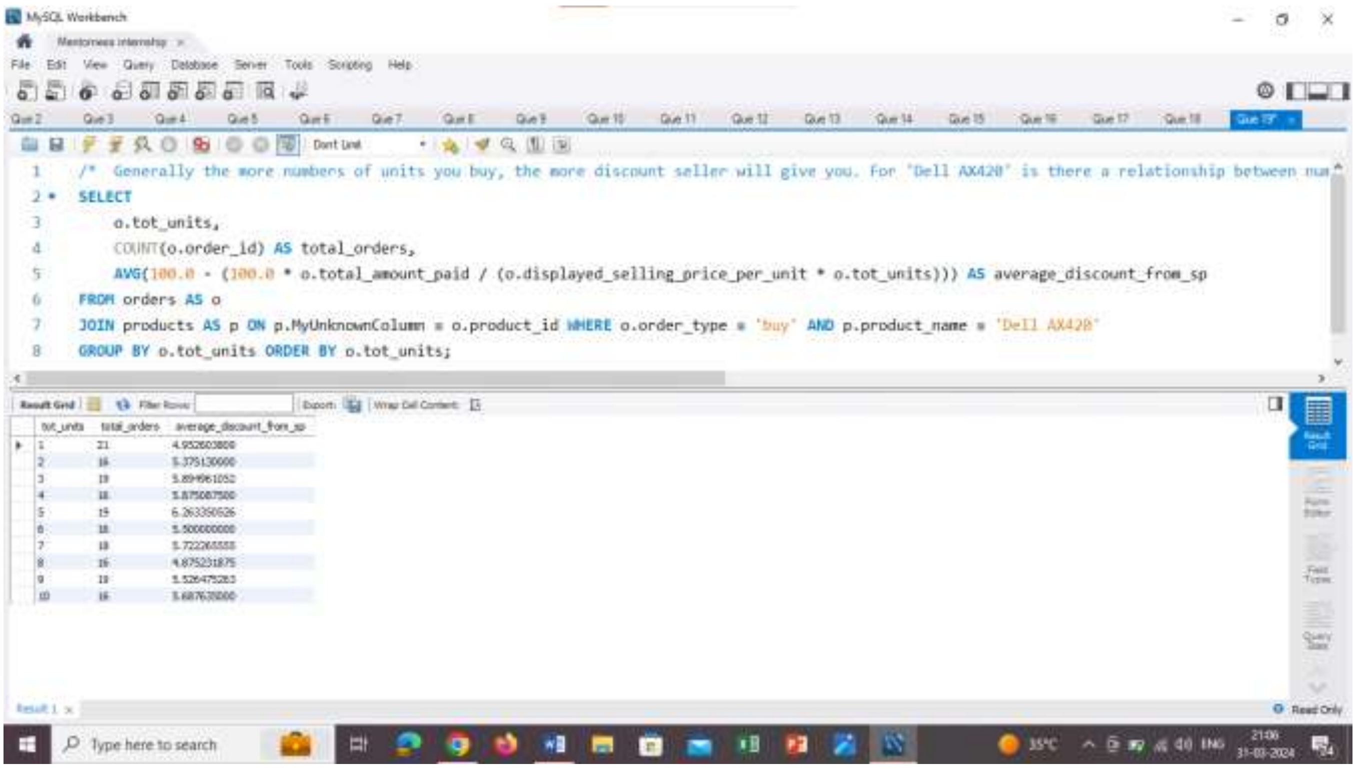
QUE 18



```sql
/* For each gender - male and female - find the absolute and percentage profit (like in Q16) by product name. */
SELECT
    c.gender,
    p.category,
    SUM(o.total_amount_paid) - SUM(p.procurement_cost_per_unit * o.tot_units) AS absolute_profit,
    100.0 * SUM(o.total_amount_paid)/SUM(p.procurement_cost_per_unit * o.tot_units) - 1 AS percentage_profit
FROM customers AS c
LEFT JOIN orders AS o
    ON c.cust_id = o.cust_id
LEFT JOIN products AS p
    ON p.MyUnknownColumn = o.product_id
GROUP BY  c.gender, p.category;
```

```
gender   category   absolute_profit   percentage_profit
male     laptop     27828248          158.55751
male     mouse      763260            203.31324
male     pendrive   449435            216.13187
female   laptop     12452614          234.30808
female   mouse      207256            177.81955
female   pendrive   165026            205.50274
```

QUE 19



```sql
/* Generally the more numbers of units you buy, the more discount seller will give you. For 'Dell AX420' is there a relationship between num
SELECT
    o.tot_units,
    COUNT(o.order_id) AS total_orders,
    AVG(100.0 - (100.0 * o.total_amount_paid / (o.displayed_selling_price_per_unit * o.tot_units))) AS average_discount_from_sp
FROM orders AS o
JOIN products AS p ON p.MyUnknownColumn = o.product_id WHERE o.order_type = 'buy' AND p.product_name = 'Dell AX420'
GROUP BY o.tot_units ORDER BY o.tot_units;
```

```
tot_units   total_orders   average_discount_from_sp
1           21             4.952603800
2           16             5.375130000
3           19             5.894961052
4           18             5.875087500
5           19             6.263350626
6           18             5.500000000
7           18             5.722265555
8           16             4.875231875
9           19             5.526475263
10          16             5.687630000
```