U19CS076 DBMS ASSIGNMENT-7

CREATING AND INSERTING TABLES

SELLER TABLE

```
Create table seller(seller_id varchar(2) PRIMARY KEY, seller_name varchar(20), Rating float(2,1));

INSERT INTO seller values('1S', 'Abhay', 3.3);

INSERT INTO seller values('2S', 'Priya', 1);

INSERT INTO seller values('3S', 'Kishan', 4.8);

INSERT INTO seller values('4S', 'Vicky', 4.3);

INSERT INTO seller values('5S', 'Sneha', 3.6);

INSERT INTO seller values('6S', 'Pushpa', 2.8);
```

CATEGORY TABLE

```
Create table category(category_id varchar(2) PRIMARY KEY, category varchar(20));

INSERT INTO category values('1C', 'Books');

INSERT INTO category values('2C', 'Footwear');

INSERT INTO category values('3C', 'Home Decor');

INSERT INTO category values('4C', 'Accessories');
```

PRODUCT TABLE

```
Create table product(product_id varchar(3) PRIMARY KEY, product varchar(50), amount integer,
quantity_rem integer, category_id varchar(2) NOT NULL, seller_id varchar(2) NOT NULL, Rating float(2,1)
DEFAULT 0.0, FOREIGN KEY(seller_id) REFERENCES seller(seller_id), FOREIGN KEY(category_id)
REFERENCES  category(category_id));

INSERT INTO  product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '1P' ,
'The Programming language of ORACLE' , 350 , 4 , '1C' , '1S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '2P' ,
'Nike White shoes' , 7000 , 2 , '2C' , '3S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '3P' ,
'White Lamp' , 800 , 3 , '3C' , '5S' );
```

```
INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '4P' , 'Antique Silver Earrings' , 400 , 7 , '4C' , '2S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '5P' , 'Antique Silver Bracelet' , 700 , 5 , '4C' , '6S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '6P' , 'Catwalk leather flats' , 1599 , 3 , '2C' , '4S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '7P' , 'Introduction to Java' , 650 , 8 , '1C' , '5S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '8P' , 'Portico King size bedsheet' , 1999 , 1 , '3C' , '1S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '9P' , 'Book rack' , 999 , 7 , '3C' , '4S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '10P' , 'Artificial Intelligence 3rd Edition' , 570 , 9 , '1C' , '2S' );

INSERT INTO product(product_id, product, amount, quantity_rem, category_id, seller_id)values( '11P' , 'Introduction to python' , 630 , 10 , '1C' , '5S' );
```

CUSTOMER TABLE

```
INSERT INTO customer values('1' , 'Ashwin' , 'pass1');

INSERT INTO customer values('2' , 'Shivangi' , 'pass2');

INSERT INTO customer values('3' , 'Sharat' , 'pass3');

INSERT INTO customer values('4' , 'Mani' , 'pass4');

INSERT INTO customer values('5' , 'Sunitha' , 'pass5');

INSERT INTO customer values('6' , 'Deepa' , 'pass6');

INSERT INTO customer values('7' , 'Pavithra' , 'pass7');

INSERT INTO customer values('8' , 'Aishwarya' , 'pass8');

INSERT INTO customer values('9' , 'Akshath' , 'pass9');

INSERT INTO customer values('10' , 'Ananya' , 'pass10');
```

ORDERS TABLE

```
Create table orders(order_id varchar(3) PRIMARY KEY, customer_id varchar(3), amount integer,
orderdate DATE, ordertime TIME, FOREIGN KEY (customer_id) REFERENCES customer(customer_id));

INSERT INTO orders values('1O' , '1' ,  1500 , '2021-02-02' , '03:10:19');

INSERT INTO orders values('2O' , '1' , 2500 , '2021-02-03' , '14:33:56');

INSERT INTO orders values('3O' , '2' , 4000 , '2020-01-21' , '17:22:09');

INSERT INTO orders values('4O' , '2' , 5500 , '2020-12-26' , '16:56:56');

INSERT INTO orders values('5O' , '3' , 5400 , '2020-12-03' , '22:22:22');

INSERT INTO orders values('6O' , '3' , 5400 , '2021-10-13' , '23:00:59');

INSERT INTO orders values('7O' , '4' , 1500 , '2021-01-11' , '08:41:02');

INSERT INTO orders values('8O' , '4' , 4650 , '2020-05-22' , '01:12:12');

INSERT INTO orders values('9O' , '6' , 5999 , '2020-08-29' , '12:08:08');

INSERT INTO orders values('10O' , '7' , 1000 , '2021-01-01' , '01:33:01');
```

ORDER_PRODUCT TABLE

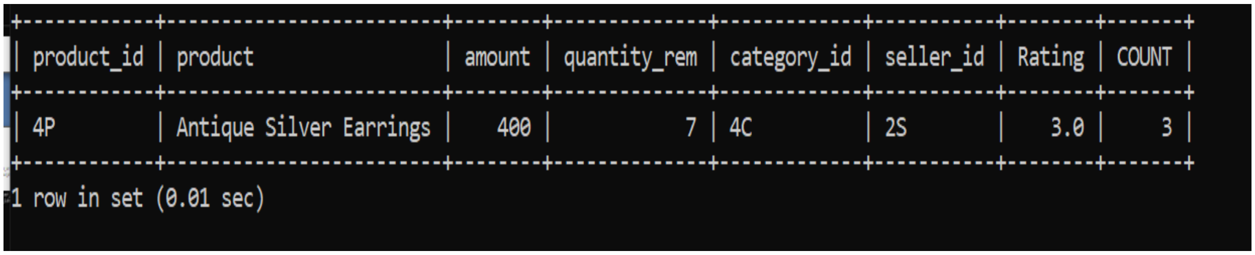
```
create table order_product(order_id varchar(3), product_id varchar(3), quantity integer, seller_id
varchar(3), original_amt integer, discount integer, prod_rating float(2,1), FOREIGN KEY (order_id)
REFERENCES orders(order_id), FOREIGN KEY (product_id) REFERENCES product(product_id), FOREIGN
KEY (seller_id) REFERENCES seller(seller_id));

INSERT INTO order_product values('1O' , '1P' , 1 , '1S' , 3500 , 0 , 4);

INSERT INTO order_product values('2O' , '1P' , 2 , '1S' , 2500 , 20 , 5);

INSERT INTO order_product values('3O' , '3P' , 3 , '5S' , 8000 , 0 , 4);

INSERT INTO order_product values('4O' , '4P' , 4 , '2S' , 2200 , 30 , 3);

INSERT INTO order_product values('5O' , '4P' , 5 , '2S' , 3400 , 0 , 2);

INSERT INTO order_product values('6O' , '4P' , 6 , '2S' , 3400 , 0 , 4);

INSERT INTO order_product values('7O' , '6P' , 1 , '4S' , 3000 , 0 , 1);

INSERT INTO order_product values('8O' , '7P' , 1 , '5S' , 6500 , 20 , 3);

INSERT INTO order_product values('9O' , '8P' , 3 , '1S' , 1999 , 0 , 5);
```

INSERT INTO order_product values('10O' , '9P' , 10 , '4S' ,3999 , 0 , 1);

QUESTIONS

1. Display the highest sold product details.

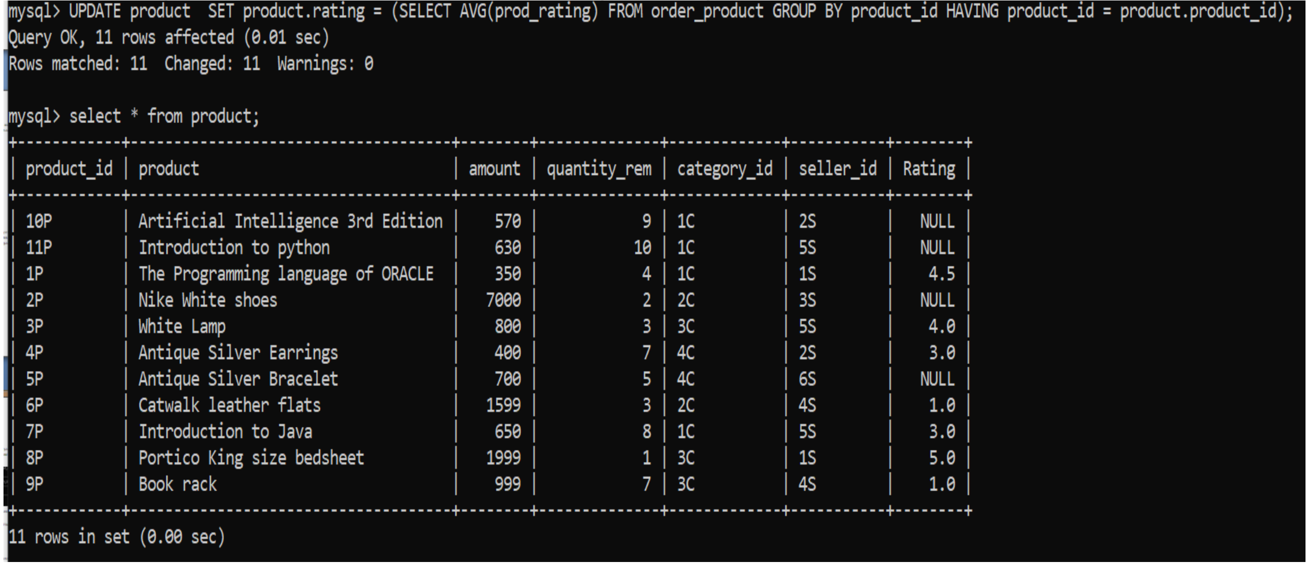
SELECT p.*, COUNT(o.product_id) as COUNT FROM product p, order_product o WHERE p.product_id = o.product_id GROUP BY o.product_id HAVING COUNT(o.product_id) = (SELECT MAX(custom) FROM (SELECT product_id, count(product_id) custom FROM order_product group by product_id)a);



```
+------------+-------------------------+--------+--------------+-------------+-----------+--------+-------+
| product_id | product                 | amount | quantity_rem | category_id | seller_id | Rating | COUNT |
+------------+-------------------------+--------+--------------+-------------+-----------+--------+-------+
| 4P         | Antique Silver Earrings |    400 |            7 | 4C          | 2S        |    3.0 |     3 |
+------------+-------------------------+--------+--------------+-------------+-----------+--------+-------+
1 row in set (0.01 sec)
```

2. Update product rating column in product table as per the entries in order_product table (calculate average).

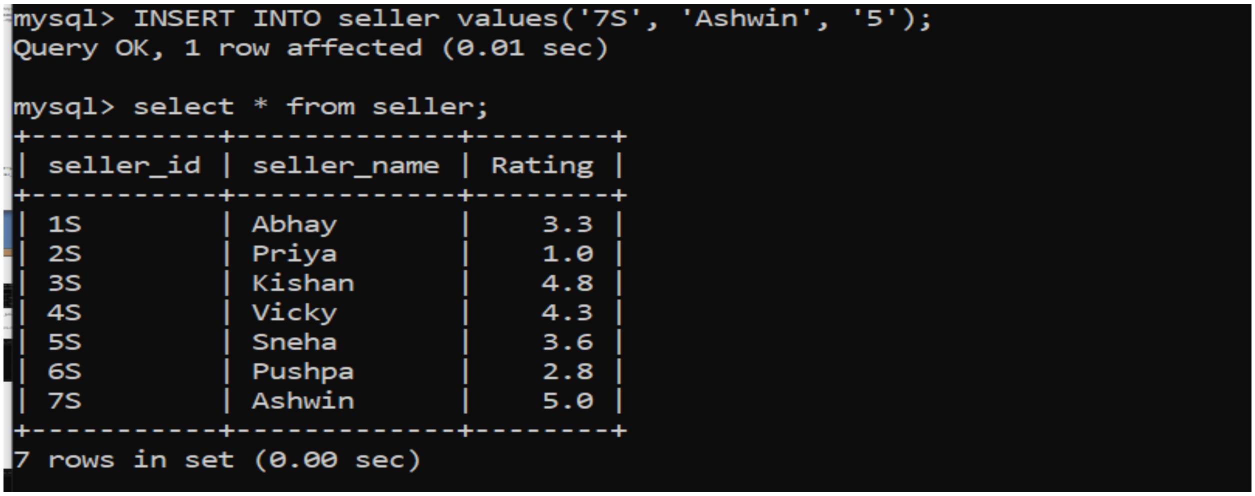
UPDATE product SET product.rating = (SELECT AVG(prod_rating) FROM order_product GROUP BY product_id HAVING product_id = product.product_id);



```
mysql> UPDATE product  SET product.rating = (SELECT AVG(prod_rating) FROM order_product GROUP BY product_id HAVING product_id = product.product_id);
Query OK, 11 rows affected (0.01 sec)
Rows matched: 11  Changed: 11  Warnings: 0

mysql> select * from product;
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
| product_id | product                           | amount | quantity_rem | category_id | seller_id | Rating |
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
| 10P        | Artificial Intelligence 3rd Edition |    570 |            9 | 1C          | 2S        |   NULL |
| 11P        | Introduction to python            |    630 |           10 | 1C          | 5S        |   NULL |
| 1P         | The Programming language of ORACLE |    350 |            4 | 1C          | 1S        |    4.5 |
| 2P         | Nike White shoes                  |   7000 |            2 | 2C          | 3S        |   NULL |
| 3P         | White Lamp                        |    800 |            3 | 3C          | 5S        |    4.0 |
| 4P         | Antique Silver Earrings           |    400 |            7 | 4C          | 2S        |    3.0 |
| 5P         | Antique Silver Bracelet           |    700 |            5 | 4C          | 6S        |   NULL |
| 6P         | Catwalk leather flats             |   1599 |            3 | 2C          | 4S        |    1.0 |
| 7P         | Introduction to Java              |    650 |            8 | 1C          | 5S        |    3.0 |
| 8P         | Portico King size bedsheet        |   1999 |            1 | 3C          | 1S        |    5.0 |
| 9P         | Book rack                         |    999 |            7 | 3C          | 4S        |    1.0 |
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
11 rows in set (0.00 sec)
```

3. Add a new seller with all details.

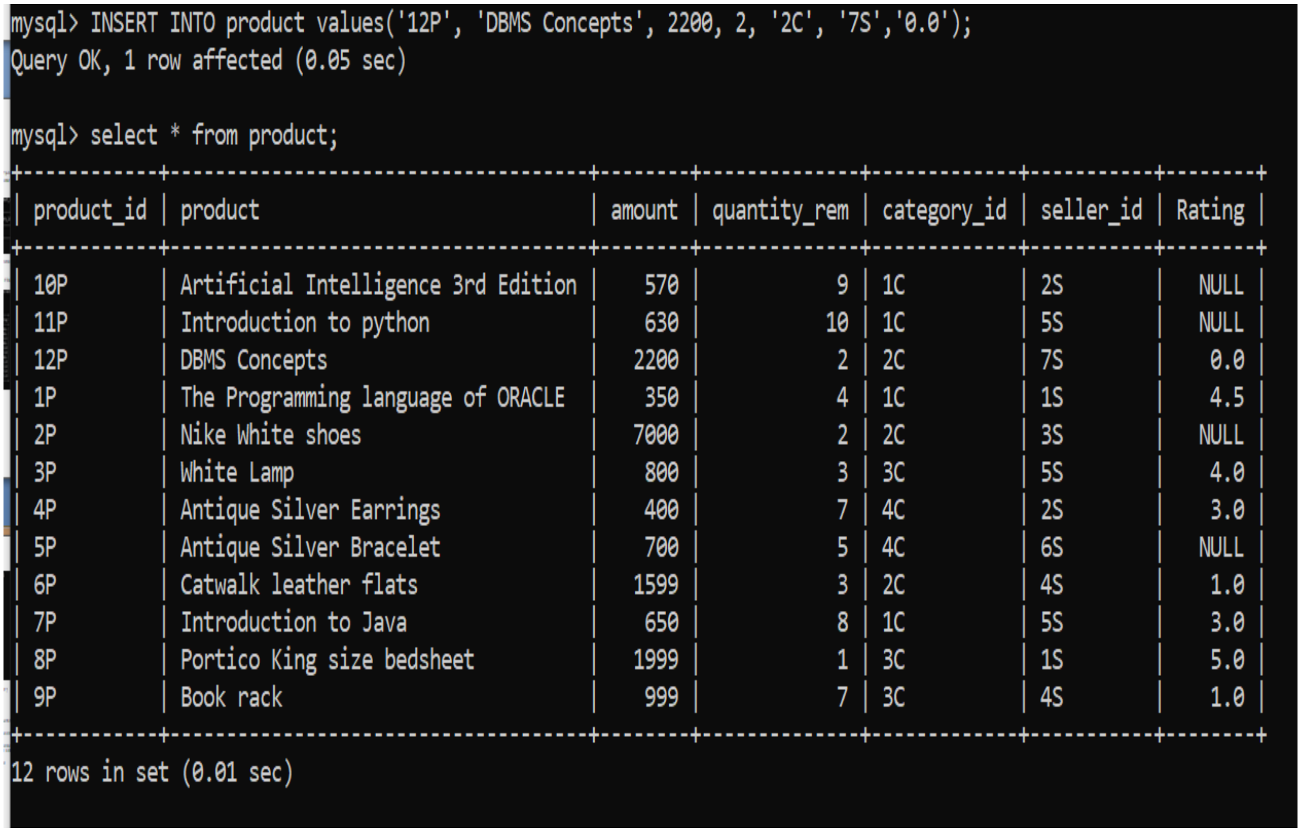
INSERT INTO seller values('7S', 'Ashwin', '5');



```
mysql> INSERT INTO seller values('7S', 'Ashwin', '5');
Query OK, 1 row affected (0.01 sec)

mysql> select * from seller;
+-----------+-------------+--------+
| seller_id | seller_name | Rating |
+-----------+-------------+--------+
| 1S        | Abhay       |    3.3 |
| 2S        | Priya       |    1.0 |
| 3S        | Kishan      |    4.8 |
| 4S        | Vicky       |    4.3 |
| 5S        | Sneha       |    3.6 |
| 6S        | Pushpa      |    2.8 |
| 7S        | Ashwin      |    5.0 |
+-----------+-------------+--------+
7 rows in set (0.00 sec)
```

4. Add a new product with all details.

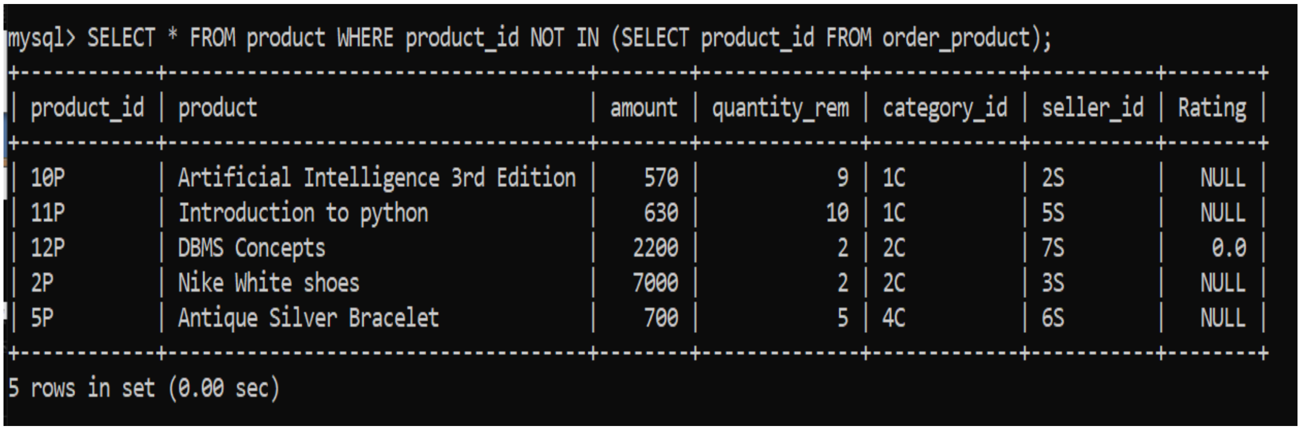
INSERT INTO product values('12P', 'DBMS Concepts', 2200, 2, '2C', '7S', '0.0');



```
mysql> INSERT INTO product values('12P', 'DBMS Concepts', 2200, 2, '2C', '7S','0.0');
Query OK, 1 row affected (0.05 sec)

mysql> select * from product;
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
| product_id | product                           | amount | quantity_rem | category_id | seller_id | Rating |
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
| 10P        | Artificial Intelligence 3rd Edition |  570 |            9 | 1C          | 2S        |   NULL |
| 11P        | Introduction to python            |    630 |           10 | 1C          | 5S        |   NULL |
| 12P        | DBMS Concepts                     |   2200 |            2 | 2C          | 7S        |    0.0 |
| 1P         | The Programming language of ORACLE |   350 |            4 | 1C          | 1S        |    4.5 |
| 2P         | Nike White shoes                  |   7000 |            2 | 2C          | 3S        |   NULL |
| 3P         | White Lamp                        |    800 |            3 | 3C          | 5S        |    4.0 |
| 4P         | Antique Silver Earrings           |    400 |            7 | 4C          | 2S        |    3.0 |
| 5P         | Antique Silver Bracelet           |    700 |            5 | 4C          | 6S        |   NULL |
| 6P         | Catwalk leather flats             |   1599 |            3 | 2C          | 4S        |    1.0 |
| 7P         | Introduction to Java              |    650 |            8 | 1C          | 5S        |    3.0 |
| 8P         | Portico King size bedsheet        |   1999 |            1 | 3C          | 1S        |    5.0 |
| 9P         | Book rack                         |    999 |            7 | 3C          | 4S        |    1.0 |
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
12 rows in set (0.01 sec)
```

5. Display the details of the products which have never sold.

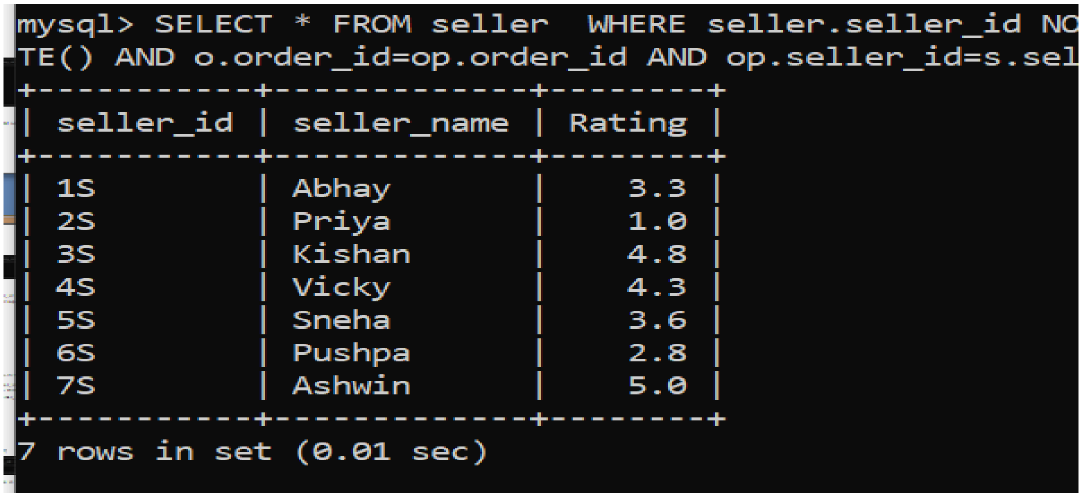
SELECT * FROM product WHERE product_id NOT IN (SELECT product_id FROM order_product);



```
mysql> SELECT * FROM product WHERE product_id NOT IN (SELECT product_id FROM order_product);
+------------+------------------------------------+--------+--------------+-------------+-----------+--------+
| product_id | product                            | amount | quantity_rem | category_id | seller_id | Rating |
+------------+------------------------------------+--------+--------------+-------------+-----------+--------+
| 10P        | Artificial Intelligence 3rd Edition |    570 |            9 | 1C          | 2S        |   NULL |
| 11P        | Introduction to python             |    630 |           10 | 1C          | 5S        |   NULL |
| 12P        | DBMS Concepts                      |   2200 |            2 | 2C          | 7S        |    0.0 |
| 2P         | Nike White shoes                   |   7000 |            2 | 2C          | 3S        |   NULL |
| 5P         | Antique Silver Bracelet            |    700 |            5 | 4C          | 6S        |   NULL |
+------------+------------------------------------+--------+--------------+-------------+-----------+--------+
5 rows in set (0.00 sec)
```

6. Display the details of the seller who has not sold any product today.

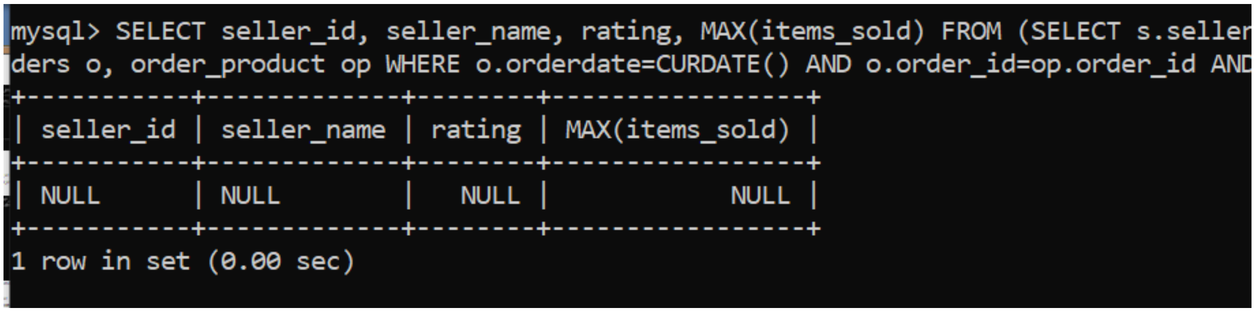
SELECT * FROM seller WHERE seller.seller_id NOT IN (SELECT DISTINCT s.seller_id FROM seller s, orders o, order_product op WHERE o.orderdate=CURDATE() AND o.order_id=op.order_id AND op.seller_id=s.seller_id);



```
mysql> SELECT * FROM seller  WHERE seller.seller_id NO
TE() AND o.order_id=op.order_id AND op.seller_id=s.sel
+-----------+-------------+--------+
| seller_id | seller_name | Rating |
+-----------+-------------+--------+
| 1S        | Abhay       |    3.3 |
| 2S        | Priya       |    1.0 |
| 3S        | Kishan      |    4.8 |
| 4S        | Vicky       |    4.3 |
| 5S        | Sneha       |    3.6 |
| 6S        | Pushpa      |    2.8 |
| 7S        | Ashwin      |    5.0 |
+-----------+-------------+--------+
7 rows in set (0.01 sec)
```

7. Display the details of the seller who has sold the highest amount of products today.

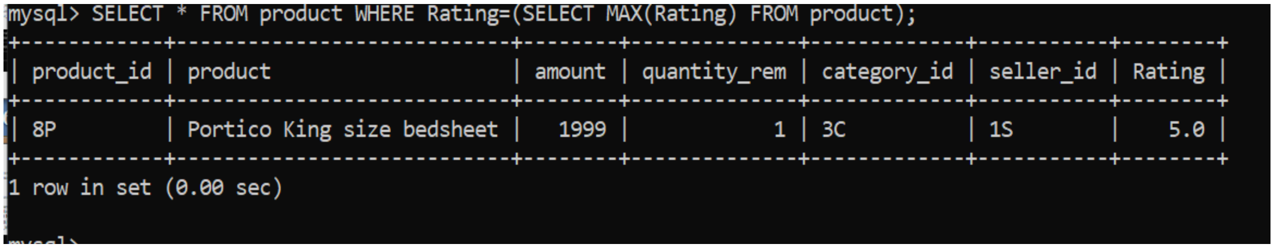
SELECT seller_id, seller_name, rating, MAX(items_sold) FROM (SELECT s.seller_id, s.seller_name, s.rating, COUNT(*) as items_sold FROM seller s, orders o, order_product op WHERE o.orderdate=CURDATE() AND o.order_id=op.order_id AND op.seller_id=s.seller_id GROUP BY op.seller_id)A;



```
mysql> SELECT seller_id, seller_name, rating, MAX(items_sold) FROM (SELECT s.seller
ders o, order_product op WHERE o.orderdate=CURDATE() AND o.order_id=op.order_id AND
+-----------+-------------+--------+-----------------+
| seller_id | seller_name | rating | MAX(items_sold) |
+-----------+-------------+--------+-----------------+
| NULL      | NULL        |   NULL |            NULL |
+-----------+-------------+--------+-----------------+
1 row in set (0.00 sec)
```

8. Display the product details with the highest rating.

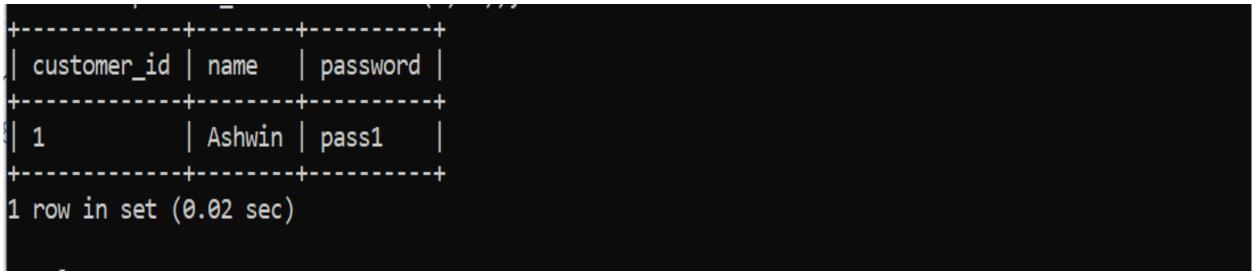
SELECT * FROM product WHERE Rating=(SELECT MAX(Rating) FROM product);



```
mysql> SELECT * FROM product WHERE Rating=(SELECT MAX(Rating) FROM product);
+------------+--------------------------+--------+--------------+-------------+-----------+--------+
| product_id | product                  | amount | quantity_rem | category_id | seller_id | Rating |
+------------+--------------------------+--------+--------------+-------------+-----------+--------+
| 8P         | Portico King size bedsheet |   1999 |            1 | 3C          | 1S        |    5.0 |
+------------+--------------------------+--------+--------------+-------------+-----------+--------+
1 row in set (0.00 sec)

mysql>
```

9. Display the customer details who has repeated the same product purchase in the last three months.

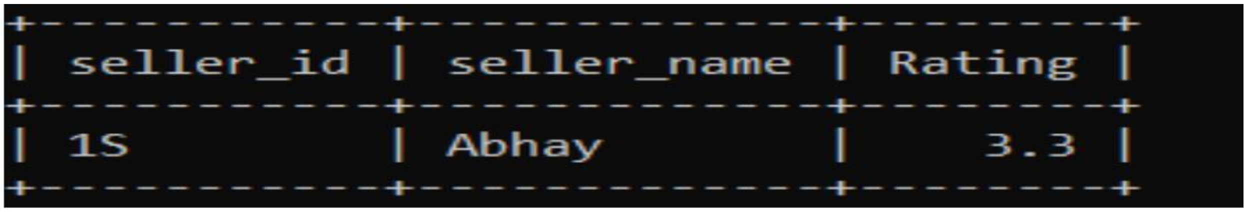
SELECT * FROM customer WHERE customer_id=(SELECT customer_id FROM orders WHERE order_id=(SELECT order_id FROM order_product WHERE order_id IN (SELECT order_id FROM orders WHERE customer_id IN (SELECT customer_id FROM orders WHERE orderdate>=DATE_SUB(CURDATE(),INTERVAL 3 MONTH) GROUP BY customer_id HAVING count(*)>1)) GROUP BY product_id HAVING count(*)>1));



```
+------------+--------+----------+
| customer_id | name   | password |
+------------+--------+----------+
| 1           | Ashwin | pass1    |
+------------+--------+----------+
1 row in set (0.02 sec)
```

10. Display the seller details who is second highest in selling products in the last three months.

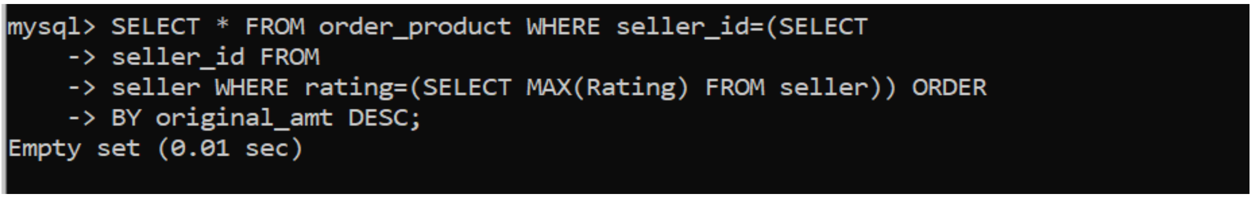
SELECT s.* FROM(SELECT order_id, seller_id, SUM(original_amt) sold FROM order_product GROUP BY seller_id HAVING order_id IN (SELECT order_id FROM orders WHERE MONTH(orderdate) >= MONTH(CURRENT_DATE - 3))) op, seller s WHERE s.seller_id = op.seller_id AND op.sold = (SELECT MAX(sold) FROM (SELECT order_id, seller_id, SUM(original_amt) sold FROM order_product GROUP BY seller_id HAVING order_id IN (SELECT order_id FROM orders WHERE MONTH(orderdate) >= MONTH(CURRENT_DATE - 3))) a WHERE sold < (SELECT MAX(sold) FROM (SELECT order_id, seller_id, SUM(original_amt) sold FROM order_product GROUP BY seller_id HAVING order_id IN (SELECT order_id FROM orders WHERE MONTH(orderdate) >= MONTH(CURRENT_DATE - 3))) a));



```
+-----------+-------------+--------+
| seller_id | seller_name | Rating |
+-----------+-------------+--------+
| 1S        | Abhay       |    3.3 |
+-----------+-------------+--------+
```

11. Display products in the descending order of product amount sold by the seller who is having the highest rating.

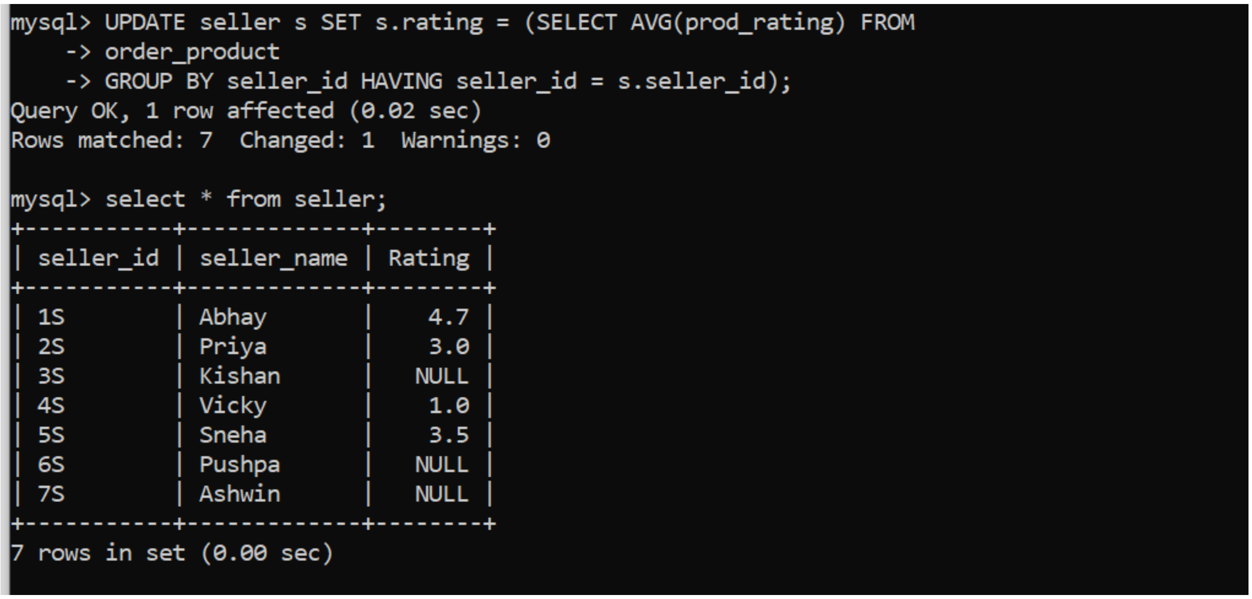
SELECT * FROM order_product WHERE seller_id=(SELECT seller_id FROM seller WHERE rating=(SELECT MAX(Rating) FROM seller)) ORDER BY original_amt DESC;



```
mysql> SELECT * FROM order_product WHERE seller_id=(SELECT
    -> seller_id FROM
    -> seller WHERE rating=(SELECT MAX(Rating) FROM seller)) ORDER
    -> BY original_amt DESC;
Empty set (0.01 sec)
```

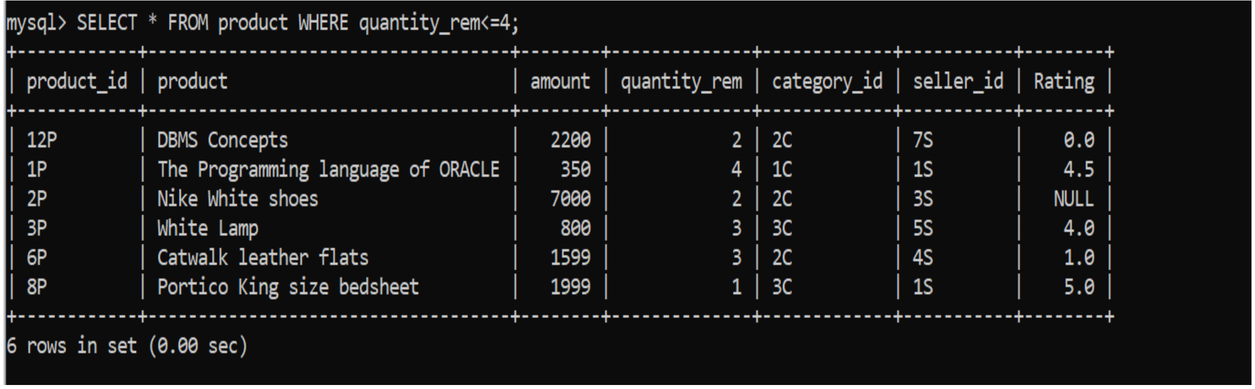
12. Update the seller ratings as per the new entries in Order_Products table.

UPDATE seller s SET s.rating = (SELECT AVG(prod_rating) FROM order_product GROUP BY seller_id HAVING seller_id = s.seller_id);



```
mysql> UPDATE seller s SET s.rating = (SELECT AVG(prod_rating) FROM
    -> order_product
    -> GROUP BY seller_id HAVING seller_id = s.seller_id);
Query OK, 1 row affected (0.02 sec)
Rows matched: 7  Changed: 1  Warnings: 0

mysql> select * from seller;
+-----------+-------------+--------+
| seller_id | seller_name | Rating |
+-----------+-------------+--------+
| 1S        | Abhay       |    4.7 |
| 2S        | Priya       |    3.0 |
| 3S        | Kishan      |   NULL |
| 4S        | Vicky       |    1.0 |
| 5S        | Sneha       |    3.5 |
| 6S        | Pushpa      |   NULL |
| 7S        | Ashwin      |   NULL |
+-----------+-------------+--------+
7 rows in set (0.00 sec)
```

13. Display the list of products having quantity remaining <= 4.
SELECT * FROM product WHERE quantity_rem<=4;



```
mysql> SELECT * FROM product WHERE quantity_rem<=4;
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
| product_id | product                           | amount | quantity_rem | category_id | seller_id | Rating |
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
| 12P        | DBMS Concepts                     |   2200 |            2 | 2C          | 7S        |    0.0 |
| 1P         | The Programming language of ORACLE |   350 |            4 | 1C          | 1S        |    4.5 |
| 2P         | Nike White shoes                  |   7000 |            2 | 2C          | 3S        |   NULL |
| 3P         | White Lamp                        |    800 |            3 | 3C          | 5S        |    4.0 |
| 6P         | Catwalk leather flats             |   1599 |            3 | 2C          | 4S        |    1.0 |
| 8P         | Portico King size bedsheet        |   1999 |            1 | 3C          | 1S        |    5.0 |
+------------+-----------------------------------+--------+--------------+-------------+-----------+--------+
6 rows in set (0.00 sec)
```