EXPERIMENT - 6

DATABASE MANAGEMENT SYSTEMS LAB

Aim

Write the queries for implementing the following functions: MAX(), MIN(), AVG(), COUNT() and other logical pattern matching operations.

Syeda Reeha Quasar
14114802719
4C7

EXPERIMENT – 6

Aim:

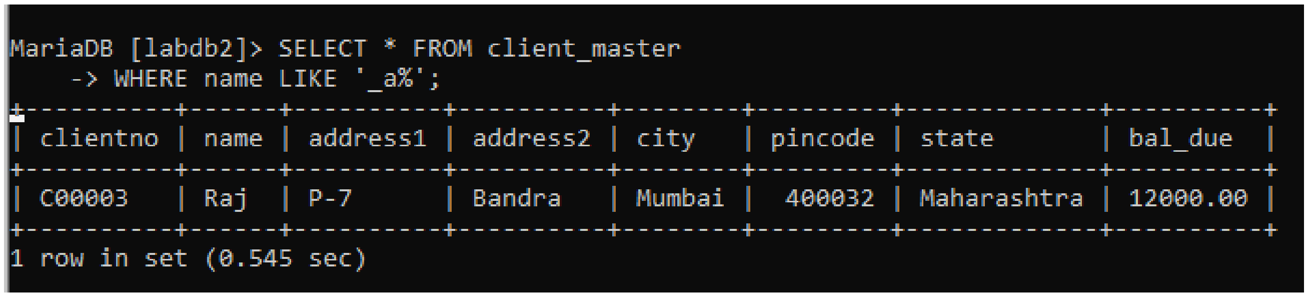
Write the queries for implementing the following functions: MAX(), MIN(), AVG(), COUNT() and other logical pattern matching operations.

Tools Used:

MariaDB

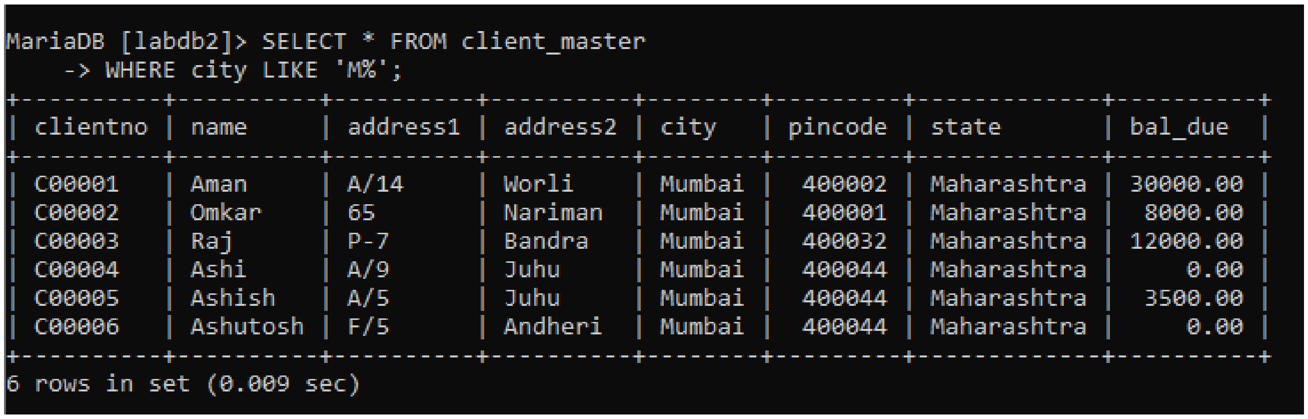
Procedure/ Queries:

1. List the names of all clients having 'a' as the second letter in their names.



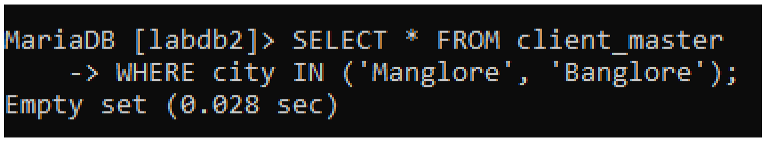
```
MariaDB [labdb2]> SELECT * FROM client_master
    -> WHERE name LIKE '_a%';
+----------+------+----------+----------+--------+---------+-------------+----------+
| clientno | name | address1 | address2 | city   | pincode | state       | bal_due  |
+----------+------+----------+----------+--------+---------+-------------+----------+
| C00003   | Raj  | P-7      | Bandra   | Mumbai |  400032 | Maharashtra | 12000.00 |
+----------+------+----------+----------+--------+---------+-------------+----------+
1 row in set (0.545 sec)
```

2. List the client who stay in the city whose first letter is 'M.



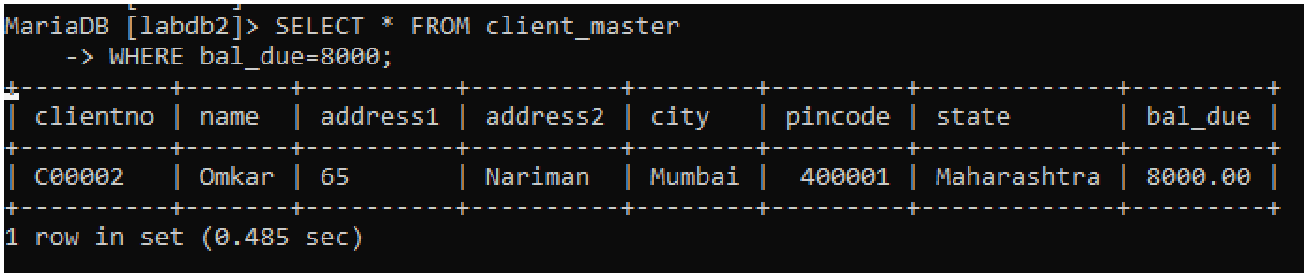
```
MariaDB [labdb2]> SELECT * FROM client_master
    -> WHERE city LIKE 'M%';
+----------+----------+----------+----------+--------+---------+-------------+----------+
| clientno | name     | address1 | address2 | city   | pincode | state       | bal_due  |
+----------+----------+----------+----------+--------+---------+-------------+----------+
| C00001   | Aman     | A/14     | Worli    | Mumbai |  400002 | Maharashtra | 30000.00 |
| C00002   | Omkar    | 65       | Nariman  | Mumbai |  400001 | Maharashtra |  8000.00 |
| C00003   | Raj      | P-7      | Bandra   | Mumbai |  400032 | Maharashtra | 12000.00 |
| C00004   | Ashi     | A/9      | Juhu     | Mumbai |  400044 | Maharashtra |     0.00 |
| C00005   | Ashish   | A/5      | Juhu     | Mumbai |  400044 | Maharashtra |  3500.00 |
| C00006   | Ashutosh | F/5      | Andheri  | Mumbai |  400044 | Maharashtra |     0.00 |
+----------+----------+----------+----------+--------+---------+-------------+----------+
6 rows in set (0.009 sec)
```

3. List all clients who stay in 'Manglore’ or ‘Banglore'



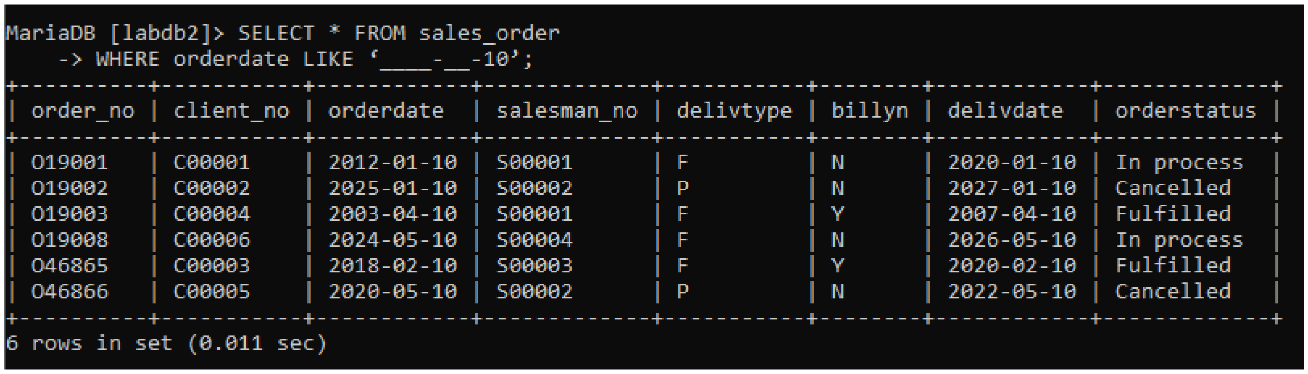
```
MariaDB [labdb2]> SELECT * FROM client_master
    -> WHERE city IN ('Manglore', 'Banglore');
Empty set (0.028 sec)
```

4. List all the clients whose bal due=8,000.



```
MariaDB [labdb2]> SELECT * FROM client_master
    -> WHERE bal_due=8000;
+----------+-------+----------+----------+--------+---------+-------------+---------+
| clientno | name  | address1 | address2 | city   | pincode | state       | bal_due |
+----------+-------+----------+----------+--------+---------+-------------+---------+
| C00002   | Omkar | 65       | Nariman  | Mumbai |  400001 | Maharashtra | 8000.00 |
+----------+-------+----------+----------+--------+---------+-------------+---------+
1 row in set (0.485 sec)
```

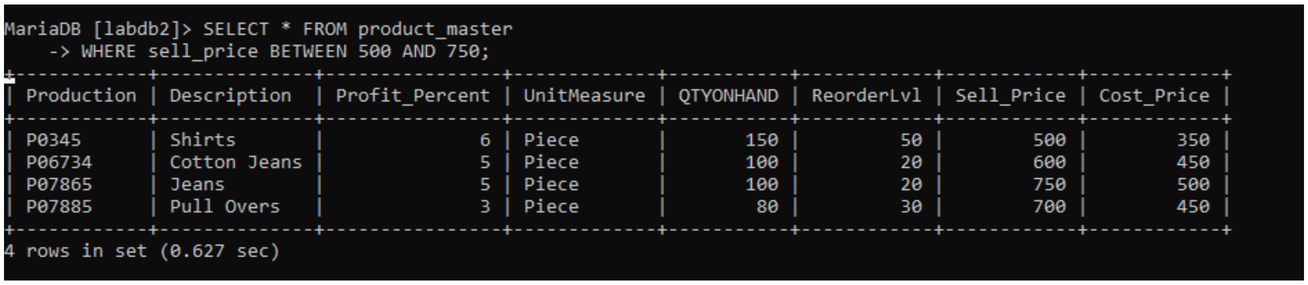
5. List all the information frome sales order for orders placed in the month of June October.



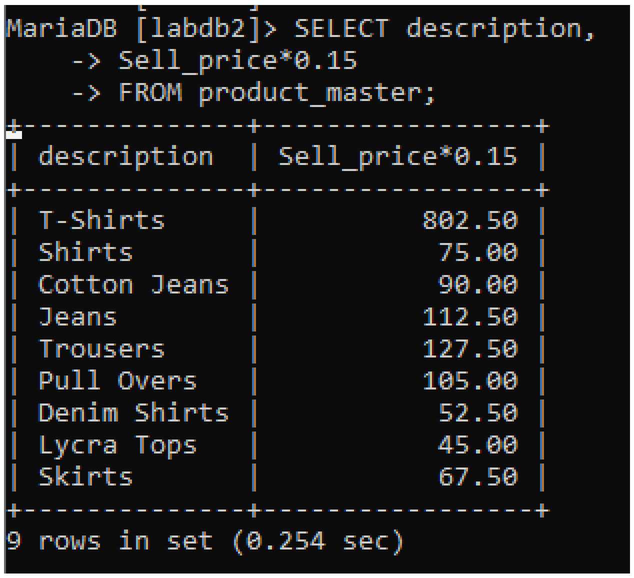
```
MariaDB [labdb2]> SELECT * FROM sales_order
    -> WHERE orderdate LIKE ‘____-__-10’;
+----------+-----------+------------+-------------+-----------+--------+------------+-------------+
| order_no | client_no | orderdate  | salesman_no | delivtype | billyn | delivdate  | orderstatus |
+----------+-----------+------------+-------------+-----------+--------+------------+-------------+
| O19001   | C00001    | 2012-01-10 | S00001      | F         | N      | 2020-01-10 | In process  |
| O19002   | C00002    | 2025-01-10 | S00002      | P         | N      | 2027-01-10 | Cancelled   |
| O19003   | C00004    | 2003-04-10 | S00001      | F         | Y      | 2007-04-10 | Fulfilled   |
| O19008   | C00006    | 2024-05-10 | S00004      | F         | N      | 2026-05-10 | In process  |
| O46865   | C00003    | 2018-02-10 | S00003      | F         | Y      | 2020-02-10 | Fulfilled   |
| O46866   | C00005    | 2020-05-10 | S00002      | P         | N      | 2022-05-10 | Cancelled   |
+----------+-----------+------------+-------------+-----------+--------+------------+-------------+
6 rows in set (0.011 sec)
```

6. List the order information for the client number 'C00001' and ' C00002

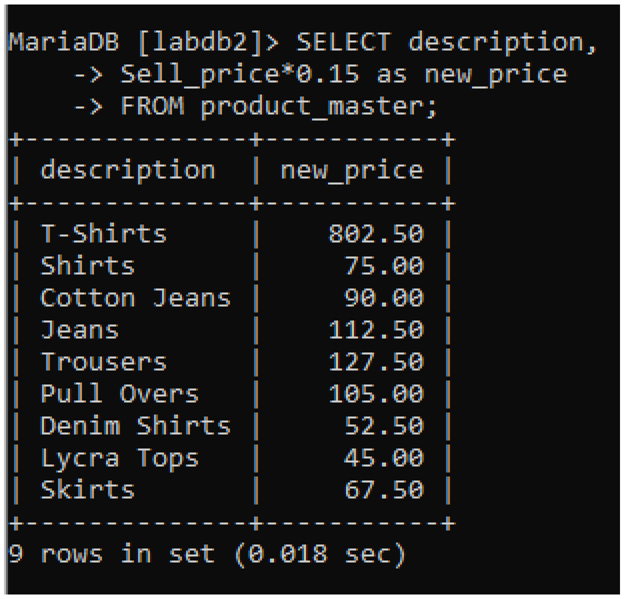
7. List the products who selling price is greater than 500 and less than or equal to 750.


```
MariaDB [labdb2]> SELECT * FROM product_master
    -> WHERE sell_price BETWEEN 500 AND 750;
+------------+--------------+----------------+-------------+-----------+------------+------------+------------+
| Production | Description  | Profit_Percent | UnitMeasure | QTYONHAND | ReorderLvl | Sell_Price | Cost_Price |
+------------+--------------+----------------+-------------+-----------+------------+------------+------------+
| P0345      | Shirts       |              6 | Piece       |       150 |         50 |        500 |        350 |
| P06734     | Cotton Jeans |              5 | Piece       |       100 |         20 |        600 |        450 |
| P07865     | Jeans        |              5 | Piece       |       100 |         20 |        750 |        500 |
| P07885     | Pull Overs   |              3 | Piece       |        80 |         30 |        700 |        450 |
+------------+--------------+----------------+-------------+-----------+------------+------------+------------+
4 rows in set (0.627 sec)
```

8. List products who's selling price is more than 500. Calculate a new Selling Price as original selling price multiplied by 0.15.

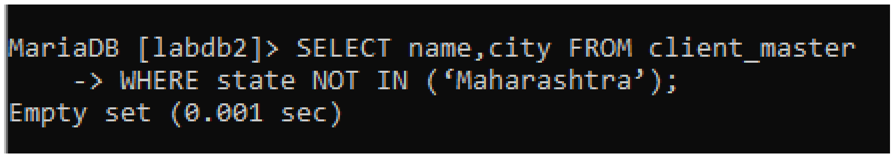

```
MariaDB [labdb2]> SELECT description,
    -> Sell_price*0.15
    -> FROM product_master;
+--------------+-----------------+
| description  | Sell_price*0.15 |
+--------------+-----------------+
| T-Shirts     |          802.50 |
| Shirts       |           75.00 |
| Cotton Jeans |           90.00 |
| Jeans        |          112.50 |
| Trousers     |          127.50 |
| Pull Overs   |          105.00 |
| Denim Shirts |           52.50 |
| Lycra Tops   |           45.00 |
| Skirts       |           67.50 |
+--------------+-----------------+
9 rows in set (0.254 sec)
```

9. Rename the new column in the output of above query as new_price.



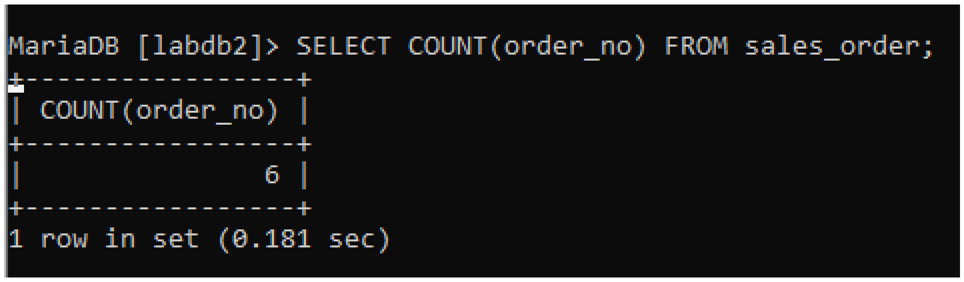
```
MariaDB [labdb2]> SELECT description,
    -> Sell_price*0.15 as new_price
    -> FROM product_master;
+--------------+-----------+
| description  | new_price |
+--------------+-----------+
| T-Shirts     |    802.50 |
| Shirts       |     75.00 |
| Cotton Jeans |     90.00 |
| Jeans        |    112.50 |
| Trousers     |    127.50 |
| Pull Overs   |    105.00 |
| Denim Shirts |     52.50 |
| Lycra Tops   |     45.00 |
| Skirts       |     67.50 |
+--------------+-----------+
9 rows in set (0.018 sec)
```

10. List the name city of clients who are not in the state of Maharashtra'.



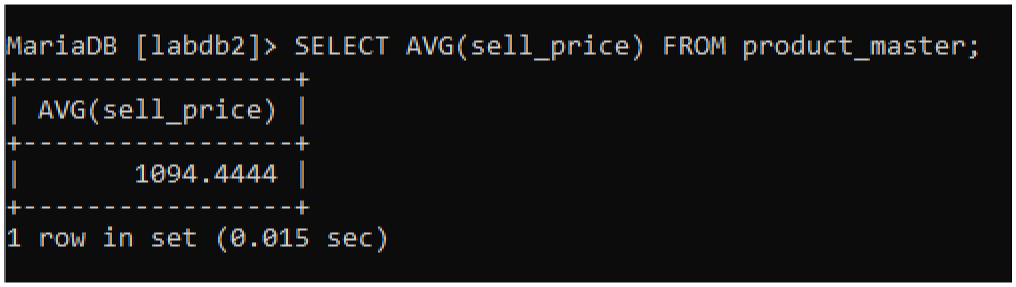
```
MariaDB [labdb2]> SELECT name,city FROM client_master
    -> WHERE state NOT IN (‘Maharashtra’);
Empty set (0.001 sec)
```

11. Count the total number of orders.



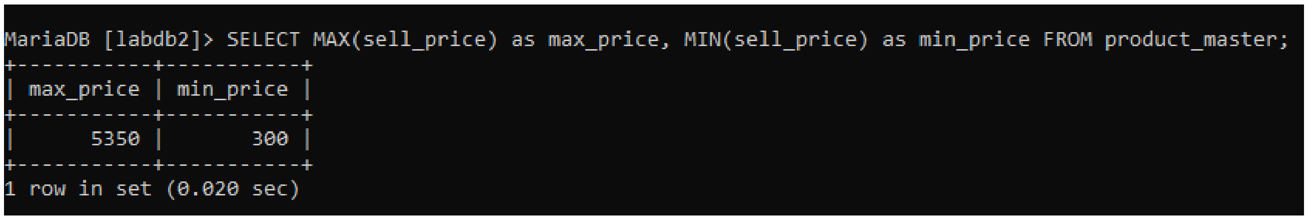
```
MariaDB [labdb2]> SELECT COUNT(order_no) FROM sales_order;
+-----------------+
| COUNT(order_no) |
+-----------------+
|               6 |
+-----------------+
1 row in set (0.181 sec)
```

12. Calculate the average price of all the products.



```
MariaDB [labdb2]> SELECT AVG(sell_price) FROM product_master;
+-----------------+
| AVG(sell_price) |
+-----------------+
|       1094.4444 |
+-----------------+
1 row in set (0.015 sec)
```

13. Determine the maximum and minimum product prices. Rename the output as max price and min_price respectively.



```
MariaDB [labdb2]> SELECT MAX(sell_price) as max_price, MIN(sell_price) as min_price FROM product_master;
+-----------+-----------+
| max_price | min_price |
+-----------+-----------+
|      5350 |       300 |
+-----------+-----------+
1 row in set (0.020 sec)
```

14. Count the number of products having price less than or equal to 500

15. List the products whose qtyonhand is less than 3 order level.

SELECT * FROM client_master
WHERE name LIKE '_a%';

SELECT * FROM client_master
WHERE city LIKE 'M%';

SELECT * FROM client_master
WHERE city IN ('Manglore', 'Banglore');

SELECT * FROM client_master
WHERE bal_due=8000;

SELECT * FROM sales_order
WHERE orderdate LIKE '____-__-10';

```
SELECT * FROM sales_order_details
WHERE clientno IN ('C00001', 'C00002');
```

```
SELECT * FROM client_master
WHERE city IN ('Manglore', 'Banglore');
```

```
SELECT * FROM product_master
WHERE sell_price BETWEEN 500 AND 750;
```

```
SELECT * FROM product_master
WHERE sell_price BETWEEN 500 AND 750;
```

```
SELECT description,

Sell_price*0.15

FROM product_master;
```

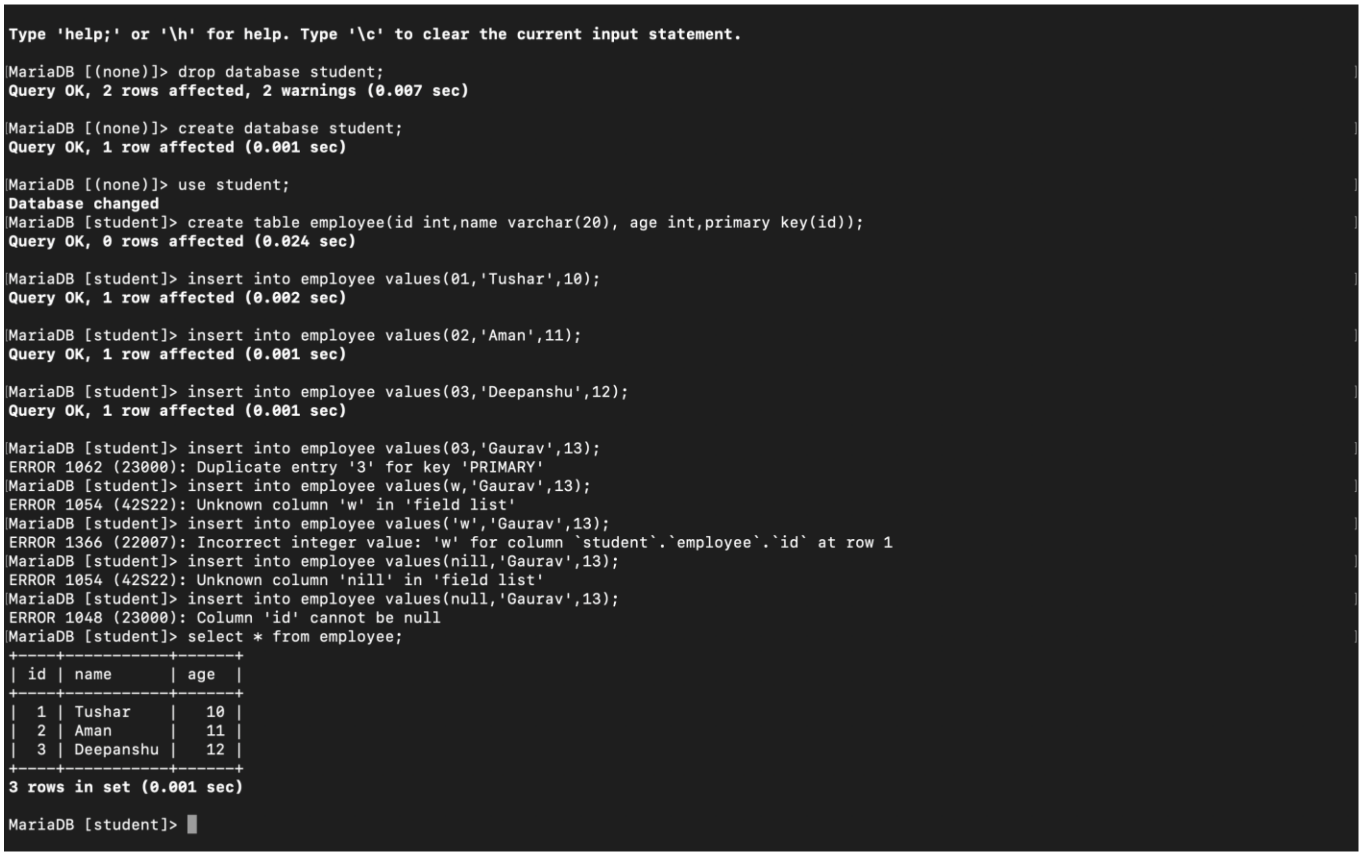
```
SELECT description,

Sell_price*0.15 as new_price

FROM product_master;
```

```
SELECT name,city FROM client_master
WHERE state NOT IN (‘Maharashtra’);
```

```
SELECT COUNT(order_no) FROM sales_order;
```

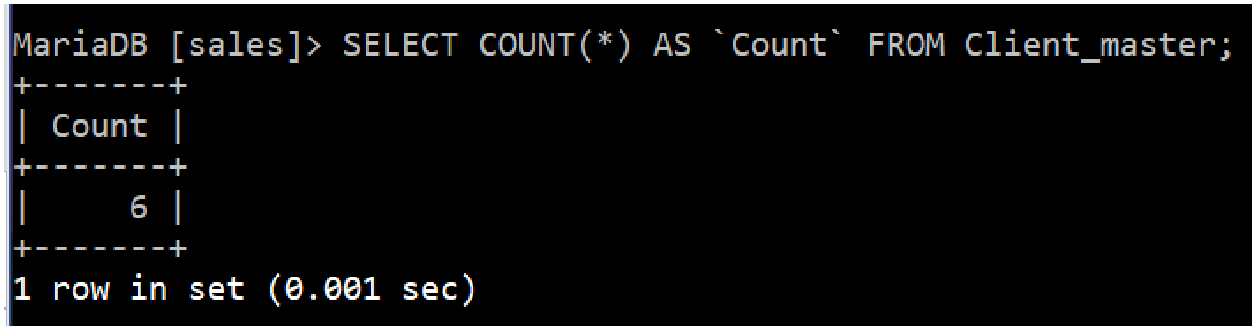
```
SELECT AVG(sell_price) FROM product_master;
```

```
SELECT MAX(sell_price) as max_price, MIN(sell_price) as min_price
FROM product_master;
```

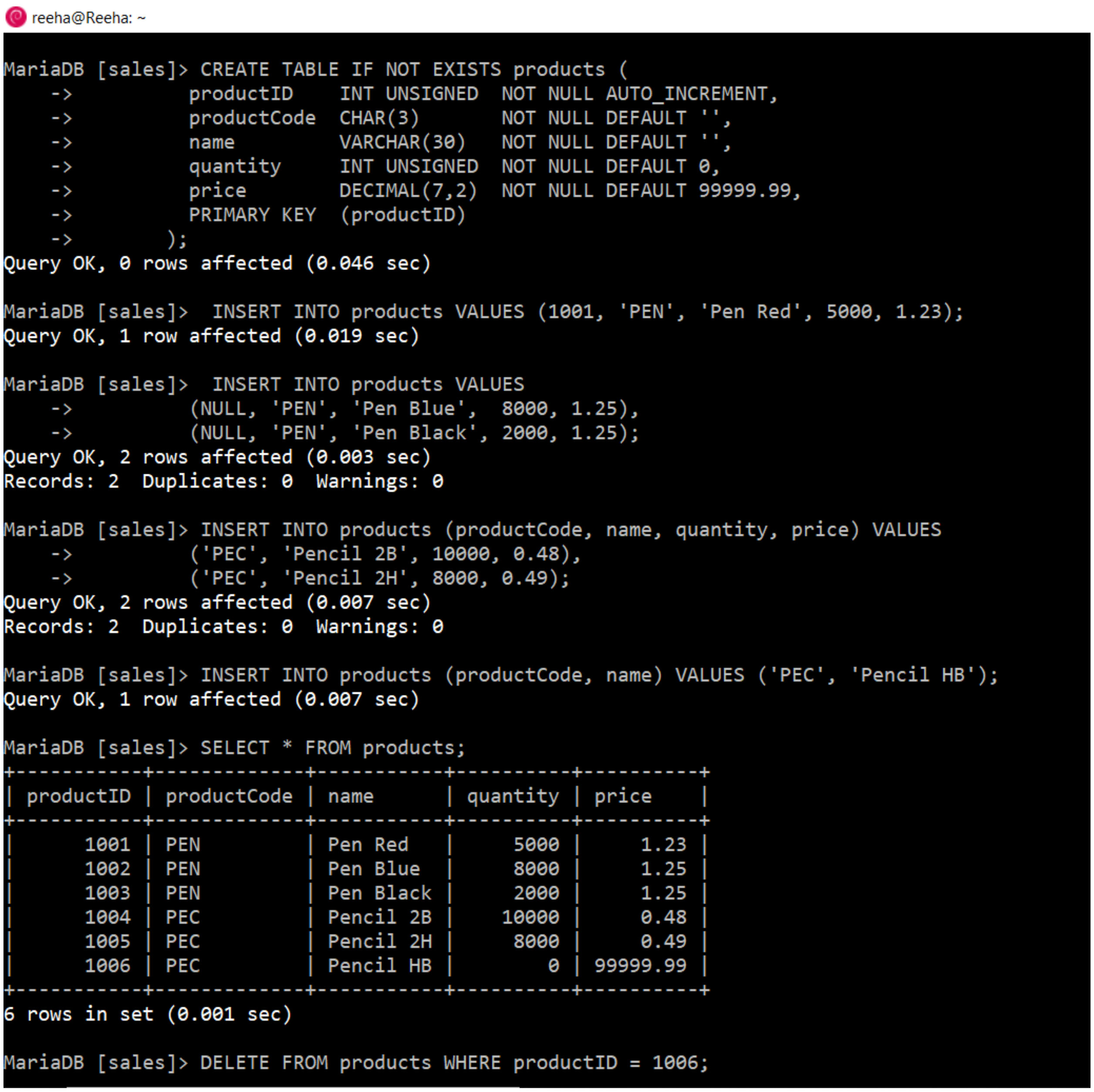


```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> drop database student;
Query OK, 2 rows affected, 2 warnings (0.007 sec)

MariaDB [(none)]> create database student;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use student;
Database changed
MariaDB [student]> create table employee(id int,name varchar(20), age int,primary key(id));
Query OK, 0 rows affected (0.024 sec)

MariaDB [student]> insert into employee values(01,'Tushar',10);
Query OK, 1 row affected (0.002 sec)

MariaDB [student]> insert into employee values(02,'Aman',11);
Query OK, 1 row affected (0.001 sec)

MariaDB [student]> insert into employee values(03,'Deepanshu',12);
Query OK, 1 row affected (0.001 sec)

MariaDB [student]> insert into employee values(03,'Gaurav',13);
ERROR 1062 (23000): Duplicate entry '3' for key 'PRIMARY'
MariaDB [student]> insert into employee values(w,'Gaurav',13);
ERROR 1054 (42S22): Unknown column 'w' in 'field list'
MariaDB [student]> insert into employee values('w','Gaurav',13);
ERROR 1366 (22007): Incorrect integer value: 'w' for column `student`.`employee`.`id` at row 1
MariaDB [student]> insert into employee values(nill,'Gaurav',13);
ERROR 1054 (42S22): Unknown column 'nill' in 'field list'
MariaDB [student]> insert into employee values(null,'Gaurav',13);
ERROR 1048 (23000): Column 'id' cannot be null
MariaDB [student]> select * from employee;
+----+-----------+------+
| id | name      | age  |
+----+-----------+------+
|  1 | Tushar    |   10 |
|  2 | Aman      |   11 |
|  3 | Deepanshu |   12 |
+----+-----------+------+
3 rows in set (0.001 sec)

MariaDB [student]> 
```

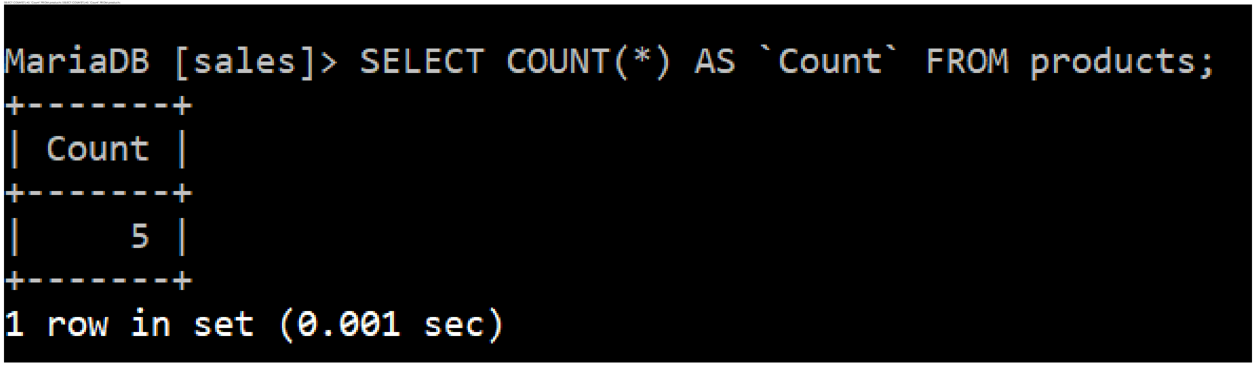
1. SELECT COUNT(*) AS `Count` FROM Client_master;



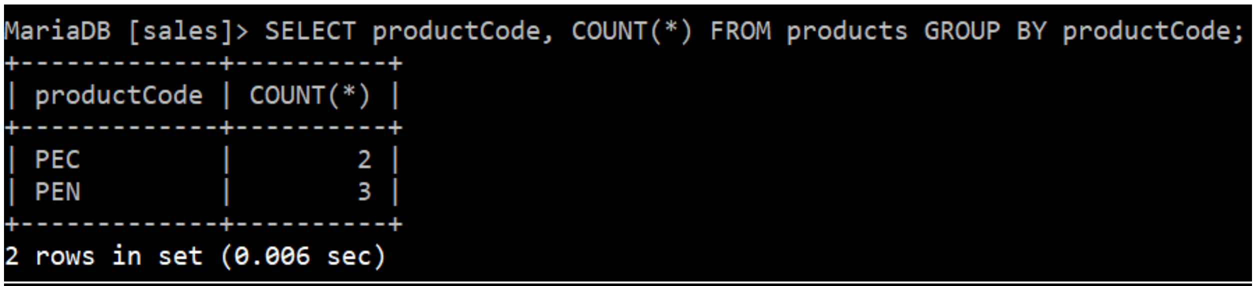
```
MariaDB [sales]> SELECT COUNT(*) AS `Count` FROM Client_master;
+-------+
| Count |
+-------+
|     6 |
+-------+
1 row in set (0.001 sec)
```


reeha@Reeha: ~

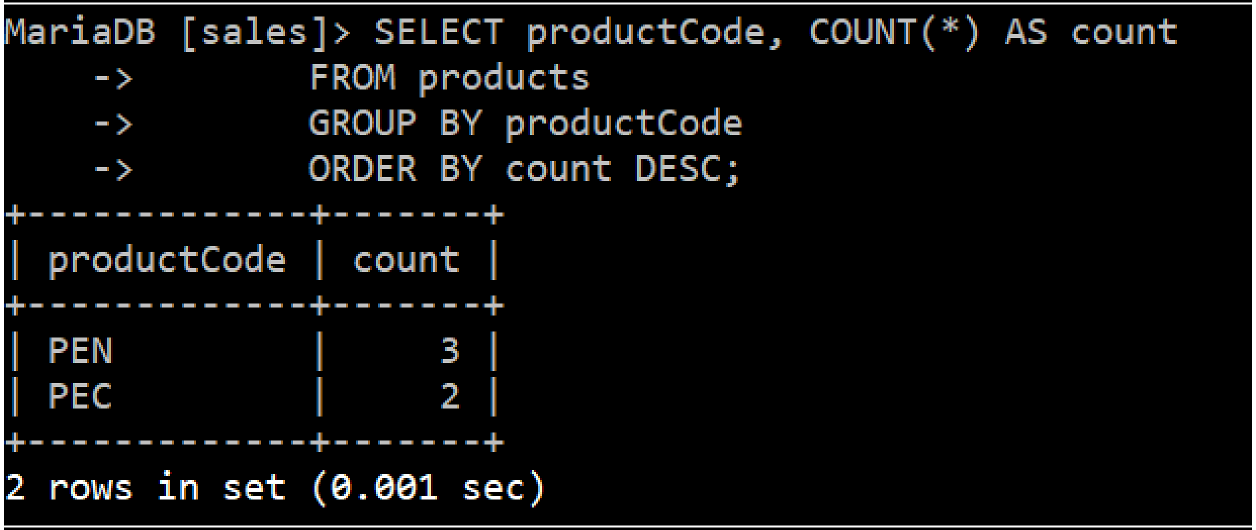
```
MariaDB [sales]> CREATE TABLE IF NOT EXISTS products (
    ->          productID    INT UNSIGNED  NOT NULL AUTO_INCREMENT,
    ->          productCode  CHAR(3)       NOT NULL DEFAULT '',
    ->          name         VARCHAR(30)   NOT NULL DEFAULT '',
    ->          quantity     INT UNSIGNED  NOT NULL DEFAULT 0,
    ->          price        DECIMAL(7,2)  NOT NULL DEFAULT 99999.99,
    ->          PRIMARY KEY  (productID)
    ->        );
Query OK, 0 rows affected (0.046 sec)

MariaDB [sales]>  INSERT INTO products VALUES (1001, 'PEN', 'Pen Red', 5000, 1.23);
Query OK, 1 row affected (0.019 sec)

MariaDB [sales]>  INSERT INTO products VALUES
    ->          (NULL, 'PEN', 'Pen Blue',  8000, 1.25),
    ->          (NULL, 'PEN', 'Pen Black', 2000, 1.25);
Query OK, 2 rows affected (0.003 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [sales]> INSERT INTO products (productCode, name, quantity, price) VALUES
    ->          ('PEC', 'Pencil 2B', 10000, 0.48),
    ->          ('PEC', 'Pencil 2H', 8000, 0.49);
Query OK, 2 rows affected (0.007 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [sales]> INSERT INTO products (productCode, name) VALUES ('PEC', 'Pencil HB');
Query OK, 1 row affected (0.007 sec)

MariaDB [sales]> SELECT * FROM products;
+-----------+-------------+-----------+----------+----------+
| productID | productCode | name      | quantity | price    |
+-----------+-------------+-----------+----------+----------+
|      1001 | PEN         | Pen Red   |     5000 |     1.23 |
|      1002 | PEN         | Pen Blue  |     8000 |     1.25 |
|      1003 | PEN         | Pen Black |     2000 |     1.25 |
|      1004 | PEC         | Pencil 2B |    10000 |     0.48 |
|      1005 | PEC         | Pencil 2H |     8000 |     0.49 |
|      1006 | PEC         | Pencil HB |        0 | 99999.99 |
+-----------+-------------+-----------+----------+----------+
6 rows in set (0.001 sec)

MariaDB [sales]> DELETE FROM products WHERE productID = 1006;
```



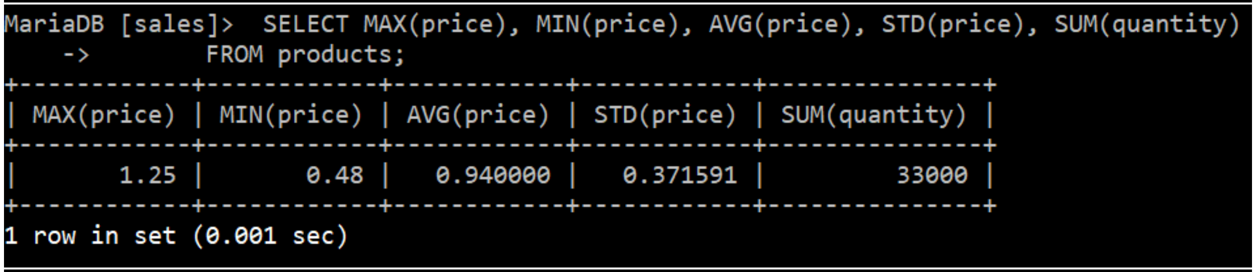
```
MariaDB [sales]> SELECT COUNT(*) AS `Count` FROM products;
+-------+
| Count |
+-------+
|     5 |
+-------+
1 row in set (0.001 sec)
```



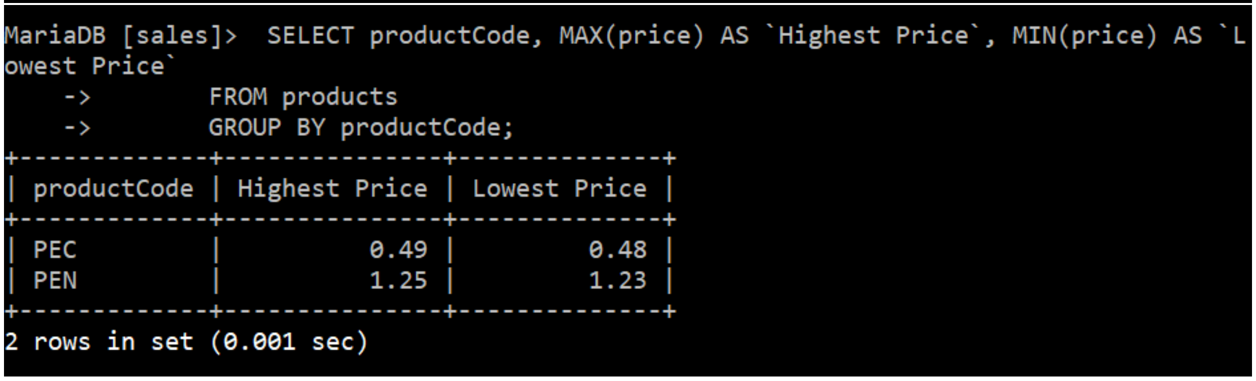
```
MariaDB [sales]> SELECT productCode, COUNT(*) FROM products GROUP BY productCode;
+-------------+----------+
| productCode | COUNT(*) |
+-------------+----------+
| PEC         |        2 |
| PEN         |        3 |
+-------------+----------+
2 rows in set (0.006 sec)
```



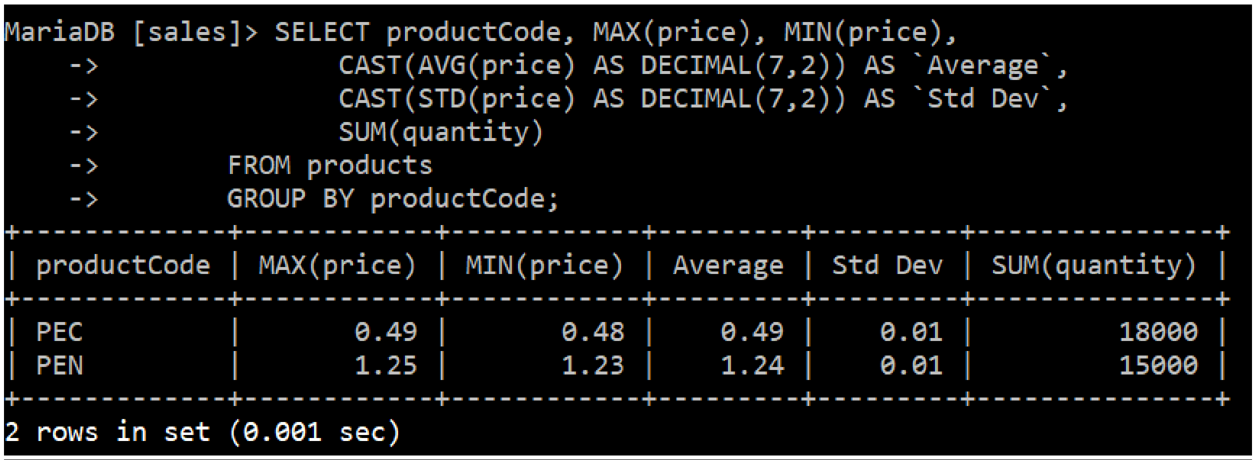
```
MariaDB [sales]> SELECT productCode, COUNT(*) AS count
    ->         FROM products
    ->         GROUP BY productCode
    ->         ORDER BY count DESC;
+-------------+-------+
| productCode | count |
+-------------+-------+
| PEN         |     3 |
| PEC         |     2 |
+-------------+-------+
2 rows in set (0.001 sec)
```



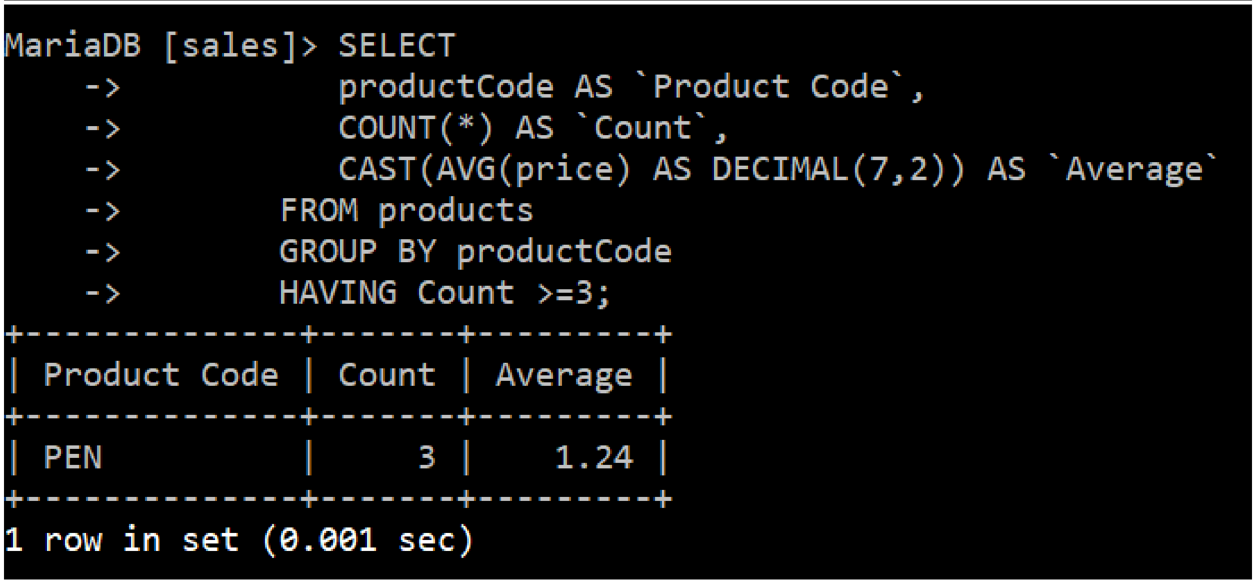
```
MariaDB [sales]>  SELECT MAX(price), MIN(price), AVG(price), STD(price), SUM(quantity)
    ->         FROM products;
+------------+------------+------------+------------+---------------+
| MAX(price) | MIN(price) | AVG(price) | STD(price) | SUM(quantity) |
+------------+------------+------------+------------+---------------+
|       1.25 |       0.48 |   0.940000 |   0.371591 |         33000 |
+------------+------------+------------+------------+---------------+
1 row in set (0.001 sec)
```



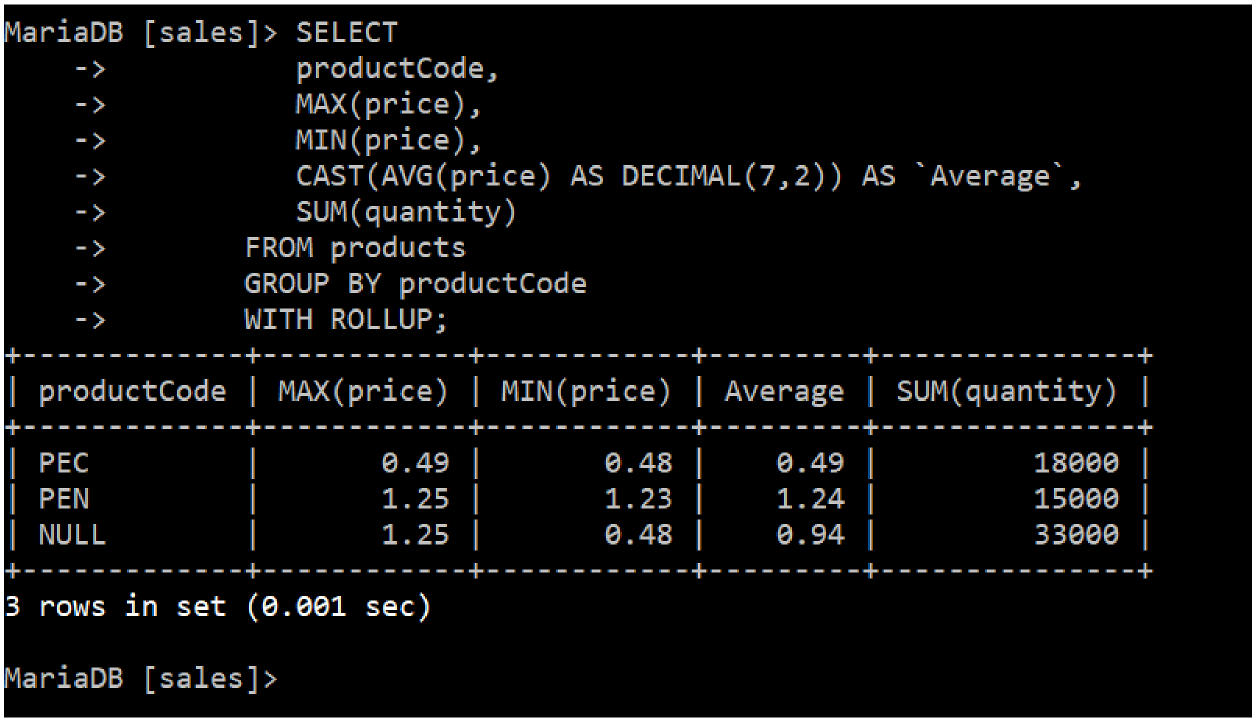
```
MariaDB [sales]>  SELECT productCode, MAX(price) AS `Highest Price`, MIN(price) AS `L
owest Price`
    ->         FROM products
    ->         GROUP BY productCode;
+-------------+---------------+--------------+
| productCode | Highest Price | Lowest Price |
+-------------+---------------+--------------+
| PEC         |          0.49 |         0.48 |
| PEN         |          1.25 |         1.23 |
+-------------+---------------+--------------+
2 rows in set (0.001 sec)
```



```
MariaDB [sales]> SELECT productCode, MAX(price), MIN(price),
    ->               CAST(AVG(price) AS DECIMAL(7,2)) AS `Average`,
    ->               CAST(STD(price) AS DECIMAL(7,2)) AS `Std Dev`,
    ->               SUM(quantity)
    ->        FROM products
    ->        GROUP BY productCode;
+-------------+------------+------------+---------+---------+---------------+
| productCode | MAX(price) | MIN(price) | Average | Std Dev | SUM(quantity) |
+-------------+------------+------------+---------+---------+---------------+
| PEC         |       0.49 |       0.48 |    0.49 |    0.01 |         18000 |
| PEN         |       1.25 |       1.23 |    1.24 |    0.01 |         15000 |
+-------------+------------+------------+---------+---------+---------------+
2 rows in set (0.001 sec)
```



```
MariaDB [sales]> SELECT
    ->              productCode AS `Product Code`,
    ->              COUNT(*) AS `Count`,
    ->              CAST(AVG(price) AS DECIMAL(7,2)) AS `Average`
    ->         FROM products
    ->         GROUP BY productCode
    ->         HAVING Count >=3;
+--------------+-------+---------+
| Product Code | Count | Average |
+--------------+-------+---------+
| PEN          |     3 |    1.24 |
+--------------+-------+---------+
1 row in set (0.001 sec)
```


```
MariaDB [sales]> SELECT
    ->           productCode,
    ->           MAX(price),
    ->           MIN(price),
    ->           CAST(AVG(price) AS DECIMAL(7,2)) AS `Average`,
    ->           SUM(quantity)
    ->        FROM products
    ->        GROUP BY productCode
    ->        WITH ROLLUP;
+-------------+------------+------------+---------+---------------+
| productCode | MAX(price) | MIN(price) | Average | SUM(quantity) |
+-------------+------------+------------+---------+---------------+
| PEC         |       0.49 |       0.48 |    0.49 |         18000 |
| PEN         |       1.25 |       1.23 |    1.24 |         15000 |
| NULL        |       1.25 |       0.48 |    0.94 |         33000 |
+-------------+------------+------------+---------+---------------+
3 rows in set (0.001 sec)

MariaDB [sales]>
```

VIVA VOCE QUESTIONS

Q.1 What are pattern matching operation?

Ans

SQL pattern matching allows you to search for patterns in data if you don't know the exact word or phrase you are seeking. This kind of SQL query uses wild card characters to match a pattern rather than specifying it exactly.

Q.2 What are different variants of like command?

Ans.

LIKE Operator	DESCRIPTION
WHERE CustomerName LIKE ‘a%’	Finds any value that starts with “a”
WHERE CustomerName LIKE ‘%a’	Finds any value that ends with “a”
WHERE CustomerName LIKE ‘%or%’	Finds any value that has “or” in any position
WHERE CustomerName LIKE ‘_r%’	Finds any value that has “r” in the second position.
WHERE CustomerName LIKE ‘a_%_%’	Finds any value that starts with “a” and are at least 3 characters in length.
WHERE ContactName LIKE ‘a%o’	Finds any value that starts with “a” and ends with “o”.

Q.3 What are different Logical operations?

Ans.

OPERATOR	**DESCRIPTION**
ALL	True if all of the subquery values meet the condition
AND	True if all the conditions separated by AND is true.

ANY	True if any of the subquery meets the condition.
BETWEEN	True if operand is within the range of comparisons.
EXISTS	True if the subquery returns one or more records.
IN	True if the operand is equal to one of a list of expressions.
NOT	Displays a record if the condition(s) is NOT TRUE
OR	True if any of the conditions separated by OR is true.

Q.4 What is difference between IN and BETWEEN command?

Ans.

The IN command allows you to specify multiple values in a WHERE clause. it is a shorthand for multiple OR conditions.

Whereas BETWEEN command select values within a given range the values can be numbers, text or dates. The BETWEEN operator is inclusive: begin an end value are included.