# SQL-ASSESSMENT

**Write SQL query to solve the problem given below**

There given a table named as product

The products are the computer components like keyboard, motherboard, monitor , speaker, etc..

The product table contains attributes like product id, product name, price and product code

Example of such table is given below :

| PRO_ID | PRO_NAME | PRO_PRICE | PRO_COM |
|---|---|---|---|
| 101 | Mother Board | 3200.00 | 15 |
| 102 | Key Board | 450.00 | 16 |
| 103 | ZIP drive | 250.00 | 14 |
| 104 | Speaker | 550.00 | 16 |
| 105 | Monitor | 5000.00 | 11 |
| 106 | DVD drive | 900.00 | 12 |
| 107 | CD drive | 800.00 | 12 |
| 108 | Printer | 2600.00 | 13 |
| 109 | Refill cartridge | 350.00 | 13 |
| 110 | Mouse | 250.00 | 12 |

Make sure to user right sql syntax to solve the query given below :

- Write sql query to find the items whose prices are higher than or equal 250rs. Order the result by product price in descending, then product name in ascending. Return pro_name and pro_price
- Write a sql query to find the cheapest item. Return pro_name and pro_price.
- Write the sql query to calculate the average price of the items for each company. Return average price and company code.
- Write the sql query to find the average total for all the product mention in the table

1.

Query | History

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags.

```
1   CREATE DATABASE shop;
2   USE shop;
3   CREATE TABLE product
4   (
5     product_id INT ,
6     product_name VARCHAR(30),
7     price INT ,
8     product_code INT ,
9     PRIMARY KEY(product_id)
10
11  );
12  INSERT INTO product
13  (product_id, product_name,price, product_code)
14  VALUES
15  (101,"mother board", 3200.00, 15),
16  (102,"key board", 450.00, 16),
17  (103,"zip driver", 250.00, 14),
18  (104,"speaker", 550.00, 16),
19  (105,"monitor", 5000.00, 11),
20  (106,"dvd driver", 900.00, 12),
21  (107,"cd drive", 800.00, 12),
22  (108,"printer", 2600.00, 13),
23  (109,"refill cartridge", 350.00, 13),
24  (110,"mouse", 250.00, 12);
25
26  SELECT*FROM product;
27
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

| product_id | product_name | price | product_code |
|---|---|---|---|
| 101 | mother board | 3200 | 15 |
| 102 | key board | 450 | 16 |
| 103 | zip driver | 250 | 14 |
| 104 | speaker | 550 | 16 |
| 105 | monitor | 5000 | 11 |
| 106 | dvd driver | 900 | 12 |
| 107 | cd drive | 800 | 12 |
| 108 | printer | 2600 | 13 |
| 109 | refill cartridge | 350 | 13 |
| 110 | mouse | 250 | 12 |

2.

Query | History

Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matchir

```
22    (108,"printer", 2600.00, 13),
23    (109,"refill cartridge", 350.00, 13),
24    (110,"mouse", 250.00, 12);
25
26    SELECT*FROM product;
27
28    SELECT product_name, price FROM product
29    WHERE price>=250.00
30    ORDER BY
31    product_name ASC, price DESC;
32
33
34
35    SELECT product_name, price FROM product
36    WHERE price=(SELECT MIN(price) FROM product);
37
38
39    SELECT product_code ,AVG(price) AS avg_price
40    FROM product
41    GROUP BY product_code;
42
43    SELECT AVG(price) AS avg_total
44    FROM product;
45
46
47
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

| product_name | price |
|---|---|
| cd drive | 800 |
| dvd driver | 900 |
| key board | 450 |
| monitor | 5000 |
| mother board | 3200 |
| mouse | 250 |
| printer | 2600 |
| refill cartridge | 350 |
| speaker | 550 |
| zip driver | 250 |

3.

```
33
34
35      SELECT product_name, price FROM product
36      WHERE price=(SELECT MIN(price) FROM product);
37
38
39      SELECT product_code ,AVG(price) AS avg_price
40      FROM product
41      GROUP BY product_code;
42
43      SELECT AVG(price) AS avg_total
44      FROM product;
45
46
47
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

| product_name | price |
|---|---|
| zip driver | 250 |
| mouse | 250 |

4.

```
39  SELECT product_code ,AVG(price) AS avg_price
40  FROM product
41  GROUP BY product_code;
42
43  SELECT AVG(price) AS avg_total
44  FROM product;
45
46
47
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

| product_code | avg_price |
|---|---|
| 11 | 5000.0000 |
| 12 | 650.0000 |
| 13 | 1475.0000 |
| 14 | 250.0000 |
| 15 | 3200.0000 |
| 16 | 500.0000 |

5.

```
42
43  SELECT AVG(price) AS avg_total
44  FROM product;
45
46
47
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only)

| avg_total |
|---|
| 1435.0000 |