Functions and Group by - Exercises, Practice, Solution

1. From the following table, write a SQL query to calculate total purchase amount of all orders. Return total purchase amount. Go to the editor

Sample table: orders

```
SELECT sum(purch_amt)

from orders;
```

Sample Output:

```
sum
17541.18
```

Click me to see the solution with pictorial presentation

2. From the following table, write a SQL query to calculate the average purchase amount of all orders. Return average purchase amount. Go to the editor

Sample table: orders

```
SELECT AVG(purch_amt)

from orders;
```

Sample Output:

```
avg
1461.7650000000000000
```

Click me to see the solution with pictorial presentation

3. From the following table, write a SQL query that counts the number of unique salespeople. Return number of salespeople. Go to the editor

Sample table: orders

```
SELECT count(DISTINCT salesman_id)

from orders;
```

Sample Output:

```
count
6
```

Click me to see the solution with pictorial presentation

4. From the following table, write a SQL query to count the number of customers. Return number of customers. Go to the editor

Sample table: customer

```
SELECT COUNT(*)

FROM customer;
```

Sample Output:

```
count
8
```

Click me to see the solution with pictorial presentation

5. From the following table, write a SQL query to determine the number of customers who received at least one grade for their activity. Go to the editor

Sample table: customer

```
SELECT COUNT (ALL grade) FROM customer;
```

Sample Output:

```
count
7
```

Click me to see the solution with pictorial presentation

6. From the following table, write a SQL query to find the maximum purchase amount. Go to the editor

Sample table: orders

```
SELECT  MAX(purch_amt)

FROM orders;
```

Sample Output:

```
max
5760.00
```

Click me to see the solution with pictorial presentation

7. From the following table, write a SQL query to find the minimum purchase amount. Go to the editor

Sample table: orders

```
SELECT  MIN(purch_amt)

FROM orders;
```

Sample Output:

```
min
65.26
```

Click me to see the solution with pictorial presentation

8. From the following table, write a SQL query to find the highest grade of the customers in each city. Return city, maximum grade. Go to the editor

Sample table: customer

```
SELECT city,MAX(grade)

FROM customer
```

```
GROUP BY city;
```

Sample Output:

```
city       max
London          300
Paris      300
New York   200
California 200
Berlin          100
Moscow          200
```

Click me to see the solution with pictorial presentation

9. From the following table, write a SQL query to find the highest purchase amount ordered by each customer. Return customer ID, maximum purchase amount. Go to the editor

Sample table: orders

```
SELECT customer_id,MAX(purch_amt)

FROM orders

GROUP BY customer_id;
```

Sample Output:

```
customer_id     max
3007       2400.60
3008       250.45
3002       5760.00
3001       270.65
3009       2480.40
3004       1983.43
3003       75.29
3005       948.50
```

Click me to see the solution with pictorial presentation

10. From the following table, write a SQL query to find the highest purchase amount ordered by each customer on a particular date. Return, order date and highest purchase amount.   ;Go to the editor

Sample table: orders

```
SELECT customer_id, ord_date, MAX(purch_amt)

FROM orders

GROUP BY customer_id,ord_date;
```

Sample Output:

```
customer_id      ord_date   max
3002        2012-10-05 65.26
3003        2012-08-17 75.29
3005        2012-10-05 150.50
3007        2012-07-27 2400.60
3009        2012-08-17 110.50
3001        2012-09-10 270.65
3002        2012-09-10 5760.00
3005        2012-09-10 948.50
3009        2012-10-10 2480.40
3008        2012-06-27 250.45
3004        2012-10-10 1983.43
3002        2012-04-25 3045.60
```

Click me to see the solution with pictorial presentation

11. From the following table, write a SQL query to determine the highest purchase amount made by each salesperson on '2012-08-17'. Return salesperson ID, purchase amount   Go to the editor

Sample table: orders

```
SELECT salesman_id,MAX(purch_amt)

FROM orders

where ord_date='2012-08-17'

GROUP BY salesman_id;
```

Sample Output:

```
salesman_id     max
5003       110.50
5007       75.29
```

Click me to see the solution with pictorial presentation

12. From the following table, write a SQL query to find the highest order (purchase) amount by each customer on a particular order date. Filter the result by highest order (purchase) amount above 2000.00. Return customer id, order date and maximum purchase amount. Go to the editor

Sample table: orders

```
SELECT customer_id,ord_date,MAX(purch_amt)

FROM orders

where purch_amt>='2000'

GROUP BY customer_id,ord_date;
```

```
SELECT customer_id,ord_date,MAX(purch_amt) FROM orders GROUP BY
customer_id,ord_date HAVING MAX(purch_amt)>2000.00;
```

Sample Output:

```
customer_id     ord_date   max
3007       2012-07-27 2400.60
3002       2012-09-10 5760.00
3009       2012-10-10 2480.40
3002       2012-04-25 3045.60
```

Click me to see the solution with pictorial presentation

13. From the following table, write a SQL query to find the maximum order (purchase) amount in the range 2000 - 6000 (Begin and end values are included.)

by combination of each customer and order date. Return customer id, order date and maximum purchase amount. Go to the editor

Sample table: orders

```
SELECT customer_id,ord_date,MAX(purch_amt)

FROM orders

GROUP BY customer_id,ord_date

HAVING MAX(purch_amt) BETWEEN 2000 and 6000;
```

Sample Output:

```
customer_id      ord_date   max
3007        2012-07-27 2400.60
3002        2012-09-10 5760.00
3009        2012-10-10 2480.40
3002        2012-04-25 3045.60
```

Click me to see the solution with pictorial presentation

14. From the following table, write a SQL query to find the maximum order (purchase) amount based on the combination of each customer and order date. Filter the rows for maximum order (purchase) amount is either 2000, 3000, 5760, 6000. Return customer id, order date and maximum purchase amount. Go to the editor

Sample table: orders

```
SELECT customer_id,ord_date,MAX(purch_amt)

FROM orders

GROUP BY customer_id,ord_date

HAVING MAX(purch_amt) IN(2000 ,3000,5760, 6000) ;
```

Sample Output:

```
customer_id      ord_date   max
3002       2012-09-10 5760.00
```

Click me to see the solution with pictorial presentation

15. From the following table, write a SQL query to determine the maximum order amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Return customer id and maximum purchase amount. Go to the editor

Sample table: orders

SELECT MAX (purch_amt), customer_id

from orders

where customer_id BETWEEN 3002 and 3007

GROUP BY customer_id;

Sample Output:

```
customer_id      max
3002       5760.00
3007       2400.60
3004       1983.43
3003       75.29
3005       948.50
```

Click me to see the solution with pictorial presentation

16. From the following table, write a SQL query to find the maximum order (purchase) amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Filter the rows for maximum order (purchase) amount is higher than 1000. Return customer id and maximum purchase amount. Go to the editor

Sample table: orders

SELECT MAX (purch_amt), customer_id

from orders

where customer_id BETWEEN 3002 and 3007

```
GROUP BY customer_id

Having max(purch_amt)>1000;
```

Sample Output:

```
customer_id     max
3002        5760.00
3007        2400.60
3004        1983.43
```

Click me to see the solution with pictorial presentation

17. From the following table, write a SQL query to determine the maximum order (purchase) amount generated by each salesperson. Filter the rows for the salesperson ID is in the range 5003 and 5008 (Begin and end values are included.). Return salesperson id and maximum purchase amount. Go to the editor

Sample table: orders

```
SELECT MAX (purch_amt), salesman_id

from orders

where salesman_id BETWEEN 5003 and 5008

GROUP BY salesman_id;
```

Sample Output:

```
salesman_id     max
5005        270.65
5003        2480.40
5007        75.29
5006        1983.43
```

Click me to see the solution with pictorial presentation

18. From the following table, write a SQL query to count all the orders generated on '2012-08-17'. Return number of orders. Go to the editor

Sample table: orders

```
SELECT count(ord_no)

from orders

where ord_date='2012-08-17';
```

Sample Output:

```
count
2
```

Click me to see the solution with pictorial presentation

19. From the following table, write a SQL query to count the number of salespeople in a city. Return number of salespeople. Go to the editor

Sample table: salesman

```
SELECT COUNT(*)

FROM salesman

WHERE city IS NOT NULL;
```

Sample Output:

```
count
6
```

Click me to see the solution with pictorial presentation

20. From the following table, write a SQL query to count the number of orders based on the combination of each order date and salesperson. Return order date, salesperson id. Go to the editor

Sample table: orders

```
SELECT count(ord_no),ord_date,salesman_id

FROM orders

GROUP BY ord_date,salesman_id;
```

Sample Output:

```
ord_date   salesman_id     count
2012-07-27 5001       1
2012-08-17 5007       1
2012-04-25 5001       1
2012-09-10 5002       1
2012-10-05 5002       1
2012-10-10 5003       1
2012-09-10 5005       1
2012-08-17 5003       1
2012-06-27 5002       1
2012-09-10 5001       1
2012-10-05 5001       1
2012-10-10 5006       1
```

Click me to see the solution with pictorial presentation

21. From the following table, write a SQL query to calculate the average product price. Return average product price. Go to the editor

Sample table: item_mast

```
SELECT avg(PRO_PRICE)

FROM item_mast;
```

Sample Output:

```
Average Price
1435.0000000000000000
```

Click me to see the solution with results

22. From the following table, write a SQL query to count the number of products whose price are higher than or equal to 350. Return number of products. Go to the editor

Sample table: item_mast

```
SELECT count(PRO_NAME)

FROM item_mast
```

```
where pro_price>=350;
```

Sample Output:

```
Number of Products
8
```

Click me to see the solution with results

23. From the following table, write a SQL query to compute the average price for unique companies. Return average price and company id. Go to the editor

Sample table: item_mast

```
SELECT AVG(pro_price) AS "Average Price", pro_com AS "Company ID" FROM
item_mast GROUP BY pro_com;
```

Sample Output:

```
Average Price                Company ID
250.0000000000000000         14
650.0000000000000000         12
3200.0000000000000000        15
5000.0000000000000000        11
1475.0000000000000000        13
500.0000000000000000         16
```

Click me to see the solution with results

24. From the following table, write a SQL query to compute the sum of the allotment amount of all departments. Return sum of the allotment amount. Go to the editor

Sample table: emp_department

```
SELECT SUM(DPT_ALLOTMENT)

FROM emp_department;
```

Sample Output:

```
sum
450000
```

Click me to see the solution with results

25. From the following table, write a SQL query to count the number of employees in each department. Return department code and number of employees. Go to the editor

Sample table: emp_details

SELECT COUNT(EMP_IDNO),EMP_DEPT

FROM emp_details

Group by EMP_DEPT;

Sample Output:

```
emp_dept        count
27              2
57              5
47              3
63              3
```

Click me to see the solution with results

□