Formatting Output - Exercises, Practice, Solution

1. From the following table, write a SQL query to select all the salespeople. Return salesman_id, name, city, commission with the percent sign (%). Go to the editor

Sample table: salesman

SELECT salesman_id,name,city,'%',commission*100

FROM salesman;

Sample Output:

```
salesman_id     name       city       ?column?   ?column?
5001       James Hoog New York   %          15.00
5002       Nail Knite Paris      %          13.00
5005       Pit Alex   London          %          11.00
5006       Mc Lyon         Paris      %          14.00
5007       Paul Adam  Rome       %          13.00
5003       Lauson Hen San Jose   %          12.00
```

Click me to see the solution with pictorial presentation

2. From the following table, write a SQL query to find the number of orders booked for each day. Return the result in a format like "For 2001-10-10 there are 15 orders".". Go to the editor

Sample table: orders

SELECT ' For',ord_date,',there are',

COUNT (ord_no),'orders.'

FROM orders

GROUP BY ord_date;

Sample Output:

```
?column?   ord_date   ?column?   count ?column?
For        2012-04-25 ,there are 1    orders.
For        2012-06-27 ,there are 1    orders.
For        2012-07-27 ,there are 1     orders.
For        2012-08-17 ,there are 2     orders.
For        2012-09-10 ,there are 3     orders.
For        2012-10-05 ,there are 2     orders.
For        2012-10-10 ,there are 2     orders.
```

Click me to see the solution with pictorial presentation

3. From the following table, write a SQL query to find all the orders. Sort the result-set in ascending order by ord_no. Return all fields. Go to the editor

Sample table: orders

```
SELECT *

FROM orders

ORDER BY ord_no;
```

Sample Output:

```
ord_no     purch_amt  ord_date   customer_id     salesman_id
70001 150.50          2012-10-05 3005       5002
70002 65.26      2012-10-05 3002       5001
70003 2480.40         2012-10-10 3009       5003
70004 110.50          2012-08-17 3009       5003
70005 2400.60         2012-07-27 3007       5001
70007 948.50          2012-09-10 3005       5002
70008 5760.00         2012-09-10 3002       5001
70009 270.65          2012-09-10 3001       5005
70010 1983.43         2012-10-10 3004       5006
70011 75.29      2012-08-17 3003       5007
70012 250.45          2012-06-27 3008       5002
70013 3045.60         2012-04-25 3002       5001
```

Click me to see the solution with pictorial presentation

4. From the following table, write a SQL query to find all the orders. Sort the result-set in descending order by ord_date. Return all fields. Go to the editor

Sample table: orders

```
SELECT *
```

FROM orders

ORDER BY ord_date DESC;

Sample Output:

```
ord_no     purch_amt  ord_date   customer_id      salesman_id
70010 1983.43         2012-10-10 3004       5006
70003 2480.40         2012-10-10 3009       5003
70002 65.26      2012-10-05 3002       5001
70001 150.50          2012-10-05 3005       5002
70009 270.65          2012-09-10 3001       5005
70008 5760.00         2012-09-10 3002       5001
70007 948.50          2012-09-10 3005       5002
70011 75.29      2012-08-17 3003       5007
70004 110.50          2012-08-17 3009       5003
70005 2400.60         2012-07-27 3007       5001
70012 250.45          2012-06-27 3008       5002
70013 3045.60         2012-04-25 3002       5001
```

Click me to see the solution with pictorial presentation

5. From the following table, write a SQL query to find all the orders. Sort the result-set in descending order by ord_date and purch_amt. Return all fields.   Go to the editor

Sample table: orders

SELECT *

FROM orders

ORDER BY ord_date ,purch_amt DESC;

Sample Output:

```
ord_no     purch_amt  ord_date   customer_id      salesman_id
70013 3045.60         2012-04-25 3002       5001
70012 250.45          2012-06-27 3008       5002
70005 2400.60         2012-07-27 3007       5001
70004 110.50          2012-08-17 3009       5003
70011 75.29      2012-08-17 3003       5007
70008 5760.00         2012-09-10 3002       5001
70007 948.50          2012-09-10 3005       5002
```

```
70009 270.65          2012-09-10 3001       5005
70001 150.50          2012-10-05 3005       5002
70002 65.26      2012-10-05 3002       5001
70003 2480.40         2012-10-10 3009       5003
70010 1983.43         2012-10-10 3004       5006
```

Click me to see the solution with pictorial presentation

6. From the following table, write a SQL query to find all the customers. Sort the result-set by customer_id. Return cust_name, city, grade.   Go to the editor

Sample table: customer

SELECT cust_name,city,grade

FROM customer

Order by customer_id;

Sample Output:

```
cust_name  city       grade
Brad Guzan London
Nick Rimando     New York   100
Jozy Altidor     Moscow          200
Fabian Johnson   Paris      300
Graham Zusi      California 200
Brad Davis New York   200
Julian Green     London          300
Geoff Cameron    Berlin          100
```

Click me to see the solution with pictorial presentation

7. From the following table, write a SQL query that calculates the maximum purchase amount generated by each salesperson for each order date. Sort the result-set by salesperson id and order date in ascending order. Return salesperson id, order date and maximum purchase amount.   Go to the editor

Sample table: orders

SELECT salesman_id,ord_date,max(purch_amt)

FROM orders

GROUP BY salesman_id,ord_date

Order by salesman_id,ord_date;

Sample Output:

```
salesman_id      ord_date   max
5001       2012-04-25 3045.60
5001       2012-07-27 2400.60
5001       2012-09-10 5760.00
5001       2012-10-05 65.26
5002       2012-06-27 250.45
5002       2012-09-10 948.50
5002       2012-10-05 150.50
5003       2012-08-17 110.50
5003       2012-10-10 2480.40
5005       2012-09-10 270.65
5006       2012-10-10 1983.43
5007       2012-08-17 75.29
```

Click me to see the solution with pictorial presentation

8. From the following table, write a SQL query to find all the customers. Sort the result-set in descending order on 3rd field. Return customer name, city and grade. Go to the editor

Sample table: customer

SELECT cust_name, city, grade

FROM customer

Order by 3 DESC;

Sample Output:

```
cust_name  city       grade
Brad Guzan London
Fabian Johnson   Paris      300
Julian Green     London          300
Brad Davis New York   200
Jozy Altidor     Moscow          200
Graham Zusi      California 200
Nick Rimando     New York   100
```

```
Geoff Cameron    Berlin          100
```

Click me to see the solution with pictorial presentation

9. From the following table, write a SQL query that counts the unique orders and the highest purchase amount for each customer. Sort the result-set in descending order on 2nd field. Return customer ID, number of distinct orders and highest purchase amount by each customer.   Go to the editor

Sample table: orders

```
SELECT customer_id,count(DISTINCT ord_no),max(purch_amt)

FROM orders

GROUP BY customer_id

Order by 2 DESC;
```

Sample Output:

```
customer_id      count      max
3002       3          5760.00
3009       2          2480.40
3005       2          948.50
3004       1          1983.43
3001       1          270.65
3007       1          2400.60
3008       1          250.45
3003       1          75.29
```

Click me to see the solution with pictorial presentation

10. From the following table, write a SQL query to calculate the summation of purchase amount, total commission (15% for all salespeople) by each order date. Sort the result-set on order date. Return order date, summation of purchase amount and commission.   Go to the editor

Sample table : orders

```
SELECT ord_date,sum(purch_amt), sum(purch_amt)*15

FROM orders
```

```
GROUP BY ord_date

Order by ord_date;
```

Sample Output:

```
ord_date   sum        ?column?
2012-04-25 3045.60         456.8400
2012-06-27 250.45          37.5675
2012-07-27 2400.60         360.0900
2012-08-17 185.79          27.8685
2012-09-10 6979.15         1046.8725
2012-10-05 215.76          32.3640
2012-10-10 4463.83         669.5745
```

Click me to see the solution with pictorial presentation

□