Create database:

Create database practice2;

Use practice2;

Create table:

create table orders(ord\_no int primary key,purch\_amt decimal(10,2),ord\_date date,customer\_id int, salesman\_id int);

insert vales in table:

insert into orders values(70001,150.5,'2012-10-05',3005,5002);

insert into orders values(70009,270.65,'2012-09-10',3001,5005);

insert into orders values(70002,65.26,'2012-10-05',3002,5001);

insert into orders values(70004,110.5,'2012-08-17',3009,5003);

insert into orders values(70007,948.5,'2012-09-10',3005,5002);

insert into orders values(70005,2400.6,'2012-07-27',3007,5001);

insert into orders values(70008,5760,'2012-09-10',3002,5001);

insert into orders values(70010,1983.43,'2012-10-10',3004,5006);

insert into orders values(70003,2480.4,'2012-10-10',3009,5003);

insert into orders values(700012,250.45,'2012-06-27',3008,5002);

insert into orders values(700011,75.29,'2012-08-27',3003,5007);

insert into orders values(700013,3045.6,'2012-04-25',3002,5001);

show table:

select \* from orders;

+--------+-----------+------------+-------------+-------------+

| 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 |

| 700011 | 75.29 | 2012-08-27 | 3003 | 5007 |

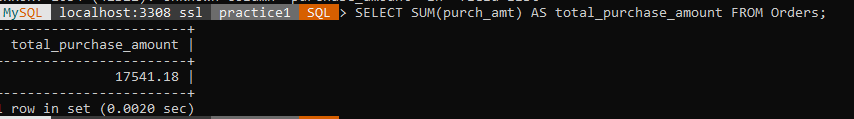
| 700012 | 250.45 | 2012-06-27 | 3008 | 5002 |

| 700013 | 3045.60 | 2012-04-25 | 3002 | 5001 |

+--------+-----------+------------+-------------+-------------+

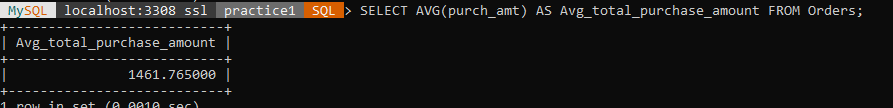
1. From the following table, write a SQL query to calculate total purchase amount of all orders. Return total purchase amount.

SELECT SUM(purch\_amt) AS total\_purchase\_amount FROM Orders;



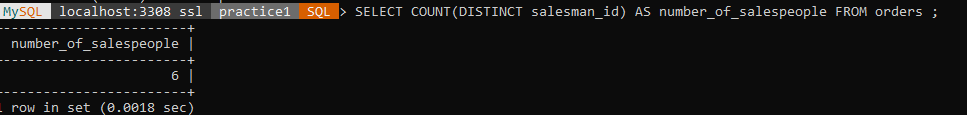
1. From the following table, write a SQL query to calculate the average purchase amount of all orders. Return average purchase amount.

SELECT AVG(purch\_amt) AS Avg\_total\_purchase\_amount FROM Orders;



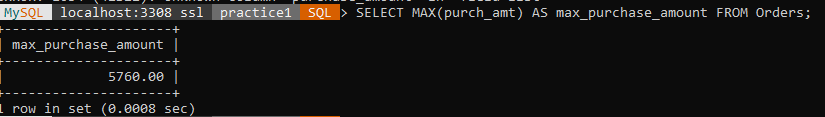
1. From the following table, write a SQL query that counts the number of unique salespeople. Return number of salespeople.

SELECT COUNT(DISTINCT salesman\_id) AS number\_of\_salespeople FROM orders ;



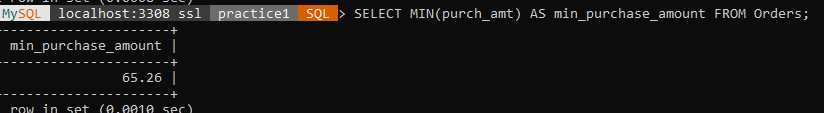
1. From the following table, write a SQL query to find the maximum purchase amount.

SELECT MAX(purch\_amt) AS max\_purchase\_amount FROM Orders;



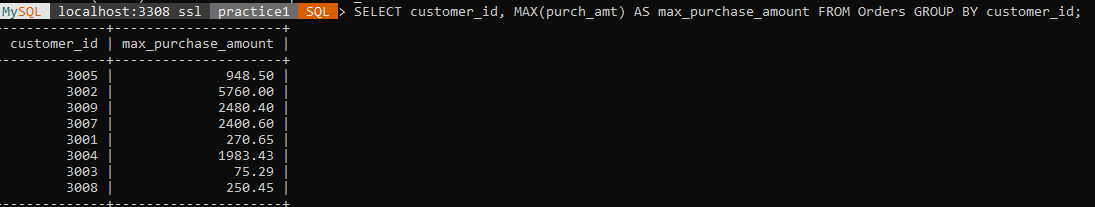
1. From the following table, write a SQL query to find the minimum purchase amount.

SELECT MIN(purch\_amt) AS min\_purchase\_amount FROM Orders;



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

SELECT customer\_id, MAX(purch\_amt) AS max\_purchase\_amount FROM Orders GROUP BY customer\_id;

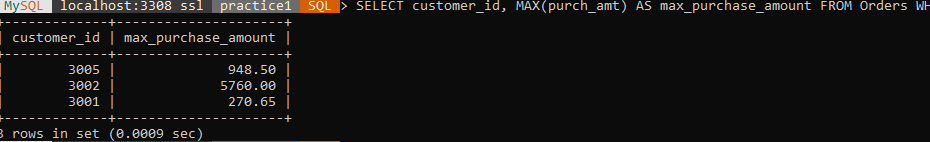


7. From the following table, write a SQL query to find the highest purchase

amount ordered by each customer on a particular date.

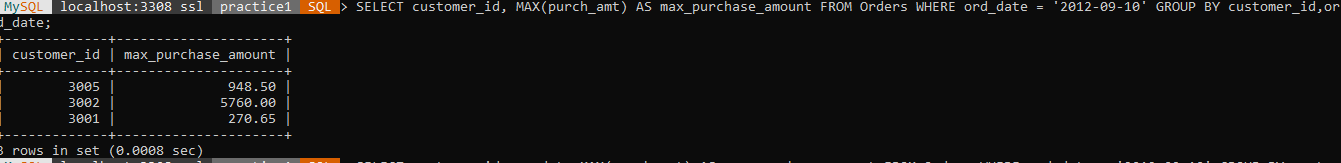
Return, order date and highest purchase amount.

SELECT customer\_id, MAX(purch\_amt) AS max\_purchase\_amount FROM Orders WHERE ord\_date = '2012-09-10' GROUP BY customer\_id;



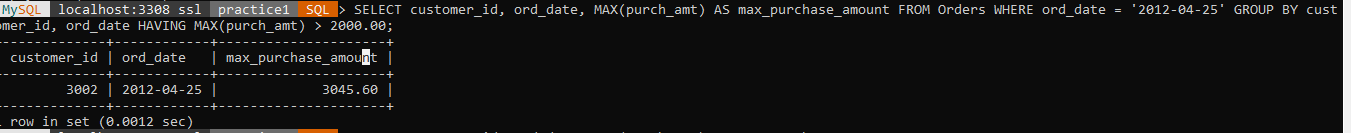
8.From the following table, write a SQL query to determine the highest purchase amount made by each salesperson on '2012-08-17'. Return salesperson ID, purchase amount.

SELECT customer\_id, MAX(purch\_amt) AS max\_purchase\_amount FROM Orders WHERE ord\_date = '2012-09-10' GROUP BY customer\_id,ord\_date;



9. 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.

SELECT customer\_id, ord\_date, MAX(purch\_amt) AS max\_purchase\_amount FROM Orders WHERE ord\_date = '2012-04-25' GROUP BY customer\_id, ord\_date HAVING MAX(purch\_amt) > 2000.00;



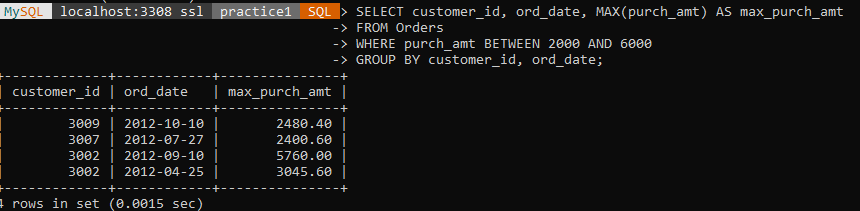
10. From the following table, write a SQL query to find the maximum order (purchase) amount in the range 2000 - 6000 (Begin and end values are included.) by combination of each customer and order date. Return customer id, order date and maximum purchase amount.

SELECT customer\_id, ord\_date, MAX(purch\_amt) AS max\_purch\_amt

-> FROM Orders

-> WHERE purch\_amt BETWEEN 2000 AND 6000

-> GROUP BY customer\_id, ord\_date;



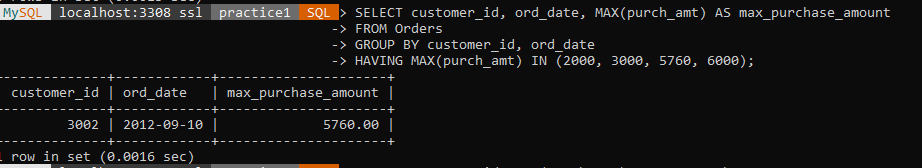
11. From the following table, write a SQL query to find the maximum order (purchase) amount based on the combination of each customer and order date. Filter the rows for maximum order (purchase) amount is either 2000, 3000, 5760, 6000. Return customer id, order date and maximum purchase amount.

SELECT customer\_id, ord\_date, MAX(purch\_amt) AS max\_purchase\_amount

-> FROM Orders

-> GROUP BY customer\_id, ord\_date

-> HAVING MAX(purch\_amt) IN (2000, 3000, 5760, 6000);



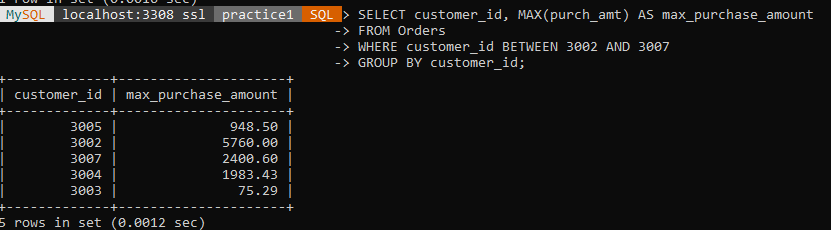
12. From the following table, write a SQL query to determine the maximum order amount for each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Return customer id and maximum purchase amount.

SELECT customer\_id, MAX(purch\_amt) AS max\_purchase\_amount

-> FROM Orders

-> WHERE customer\_id BETWEEN 3002 AND 3007

-> GROUP BY customer\_id;



13. 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.

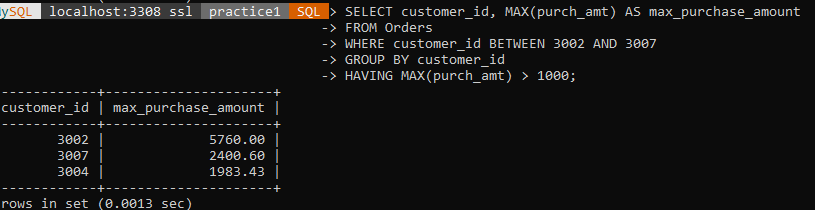
SELECT customer\_id, MAX(purch\_amt) AS max\_purchase\_amount

-> FROM Orders

-> WHERE customer\_id BETWEEN 3002 AND 3007

-> GROUP BY customer\_id

-> HAVING MAX(purch\_amt) > 1000;



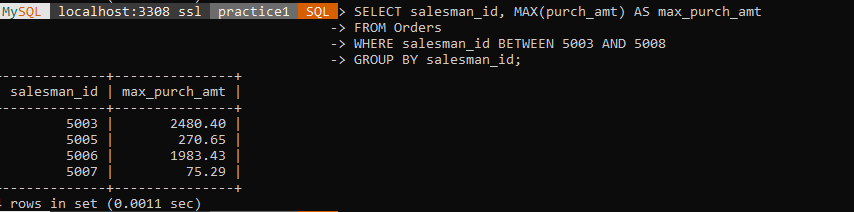
14. From the following table, write a SQL query to determine the maximum order (purchase) amount generated by each salesperson. Filter the rows for the salesperson ID is in the range 5003 and 5008 (Begin and end values are included.). Return salesperson id and maximum purchase amount.

SELECT salesman\_id, MAX(purch\_amt) AS max\_purch\_amt

-> FROM Orders

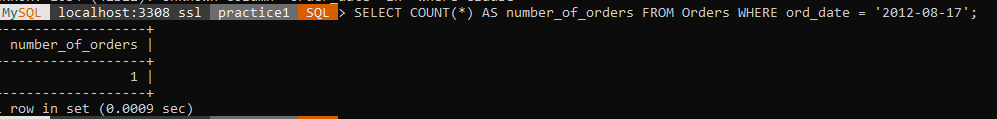
-> WHERE salesman\_id BETWEEN 5003 AND 5008

-> GROUP BY salesman\_id;



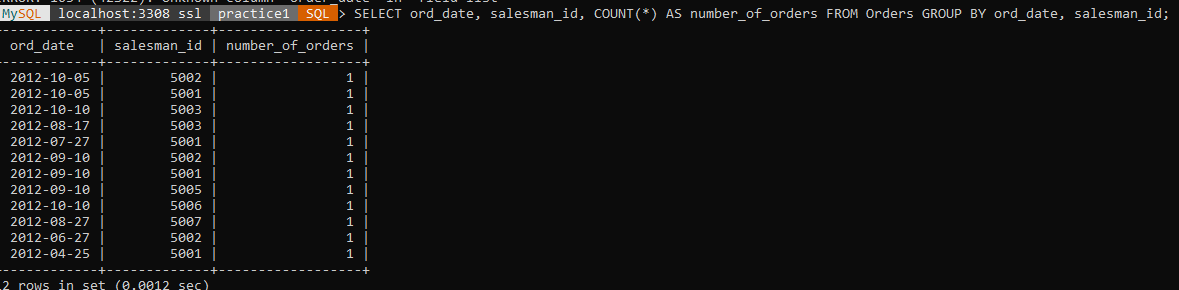
15. From the following table, write a SQL query to count all the orders generated on '2012-08-17'. Return number of orders.

SELECT COUNT(\*) AS number\_of\_orders FROM Orders WHERE ord\_date = '2012-08-17';



16. 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.

SELECT ord\_date, salesman\_id, COUNT(\*) AS number\_of\_orders FROM Orders GROUP BY ord\_date, salesman\_id;

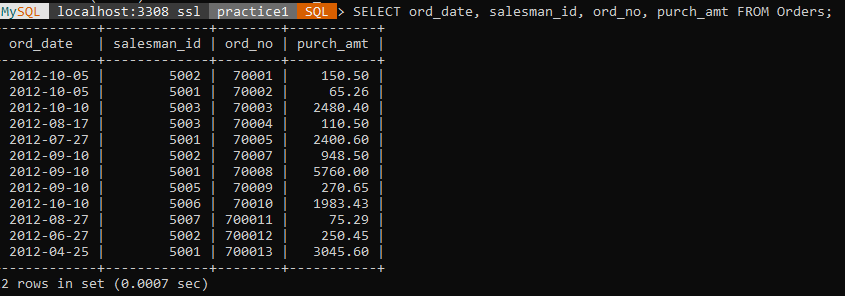


17. Write a query to display the columns in a specific order,

such as order date, salesman ID, order number, and purchase amount

for all orders

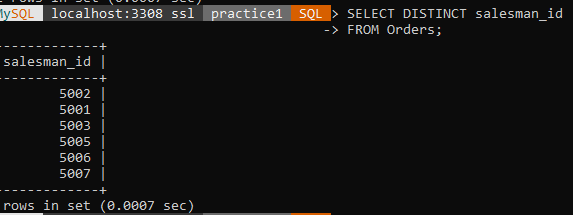
SELECT ord\_date, salesman\_id, ord\_no, purch\_amt FROM Orders;



18. From the following table, write a SQL query to identify the unique salespeople ID. Return salesman\_id.

SELECT DISTINCT salesman\_id

-> FROM Orders;

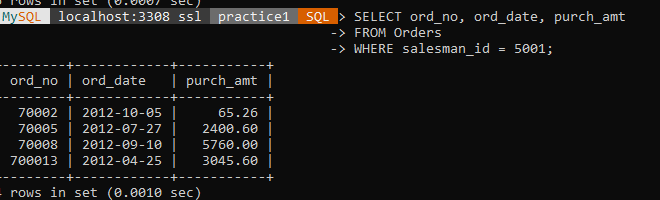


19. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.

SELECT ord\_no, ord\_date, purch\_amt

-> FROM Orders

-> WHERE salesman\_id = 5001;

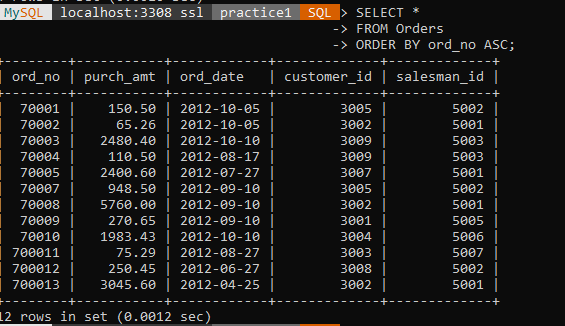


20. From the following table, write a SQL query to find all the orders. Sort the result-set in ascending order by ord\_no. Return all fields.

MySQL localhost:3308 ssl practice1 SQL > SELECT \*

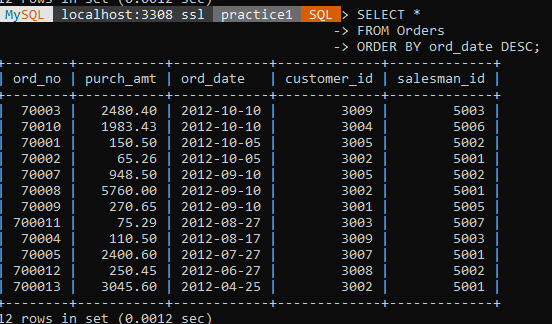
-> FROM Orders

-> ORDER BY ord\_no ASC;



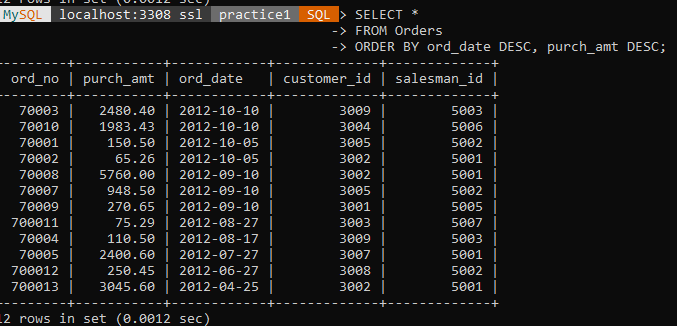
21. From the following table, write a SQL query to find all the orders. Sort the result-set in descending order by ord\_date. Return all fields.

SELECT \* FROM Orders ORDER BY ord\_date DESC;



22. 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.

SELECT \*FROM Orders ORDER BY ord\_date DESC, purch\_amt DESC;



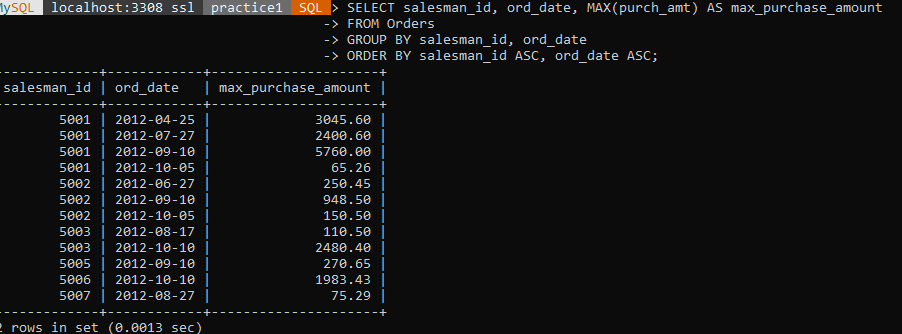
23. From the following table, write a SQL query that calculates the maximum purchase amount generated by each salesperson for each order date. Sort the result-set by salesperson id and order date in ascending order. Return salesperson id, order date and maximum purchase amount.

SELECT salesman\_id, ord\_date, MAX(purch\_amt) AS max\_purchase\_amount

-> FROM Orders

-> GROUP BY salesman\_id, ord\_date

-> ORDER BY salesman\_id ASC, ord\_date ASC;



24. 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.

SELECT customer\_id,

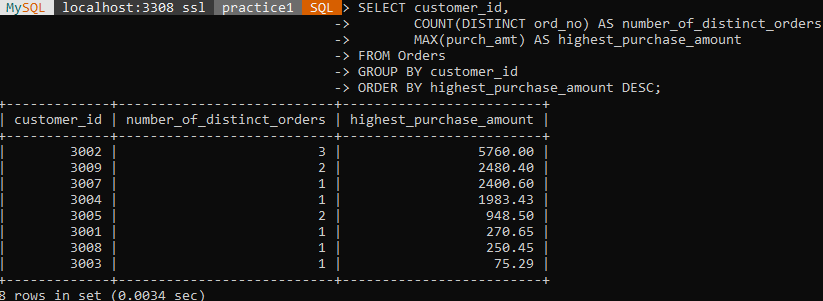
-> COUNT(DISTINCT ord\_no) AS number\_of\_distinct\_orders,

-> MAX(purch\_amt) AS highest\_purchase\_amount

-> FROM Orders

-> GROUP BY customer\_id

-> ORDER BY highest\_purchase\_amount DESC;



25. 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.

SELECT ord\_date, SUM(purch\_amt) AS total\_purchase\_amount, SUM(purch\_amt \* 0.15) AS total\_commission FROM Orders GROUP BY ord\_date;

