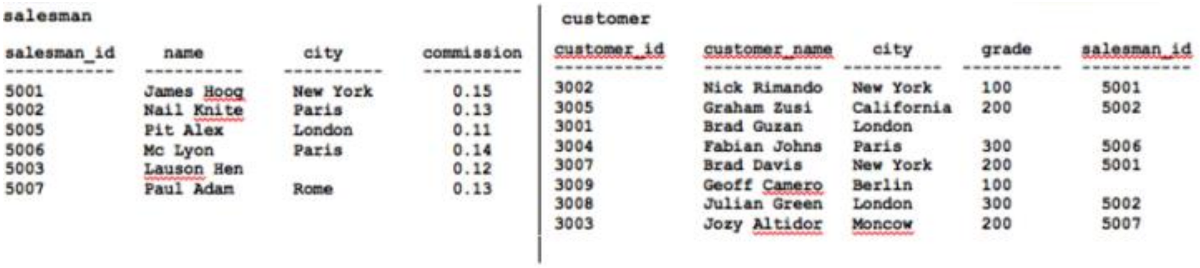
Name – Aditya Kamble Class – MSC DSAI PART i
Roll No – L011

Practical No 1
Database Management System

Aim : DDL operations on Relational Schema

Design the following schema and execute the following queries on it:

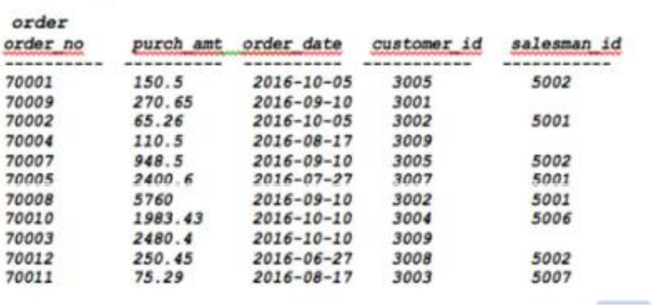


salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

customer

customer_id	customer_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London		
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moncow	200	5007

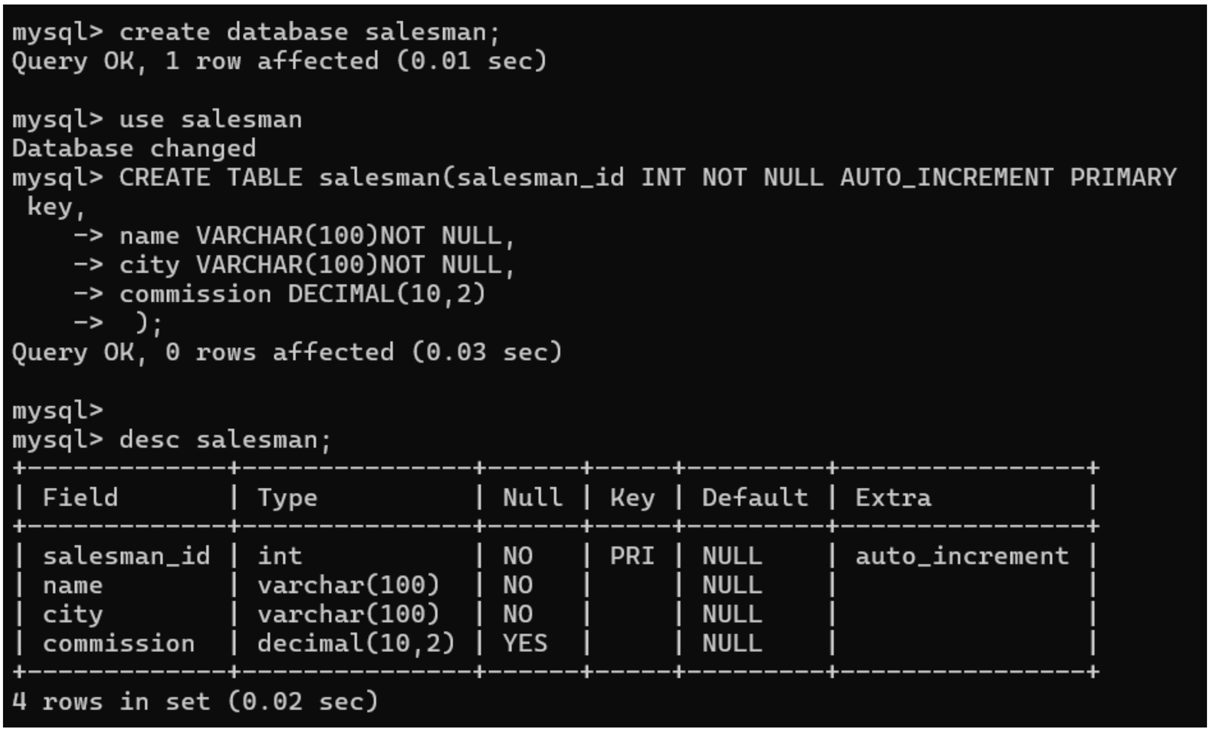


order

order_no	purch_amt	order_date	customer_id	salesman_id
70001	150.5	2016-10-05	3005	5002
70009	270.65	2016-09-10	3001	
70002	65.26	2016-10-05	3002	5001
70004	110.5	2016-08-17	3009	
70007	948.5	2016-09-10	3005	5002
70005	2400.6	2016-07-27	3007	5001
70008	5760	2016-09-10	3002	5001
70010	1983.43	2016-10-10	3004	5006
70003	2480.4	2016-10-10	3009	
70012	250.45	2016-06-27	3008	5002
70011	75.29	2016-08-17	3003	5007

Code:

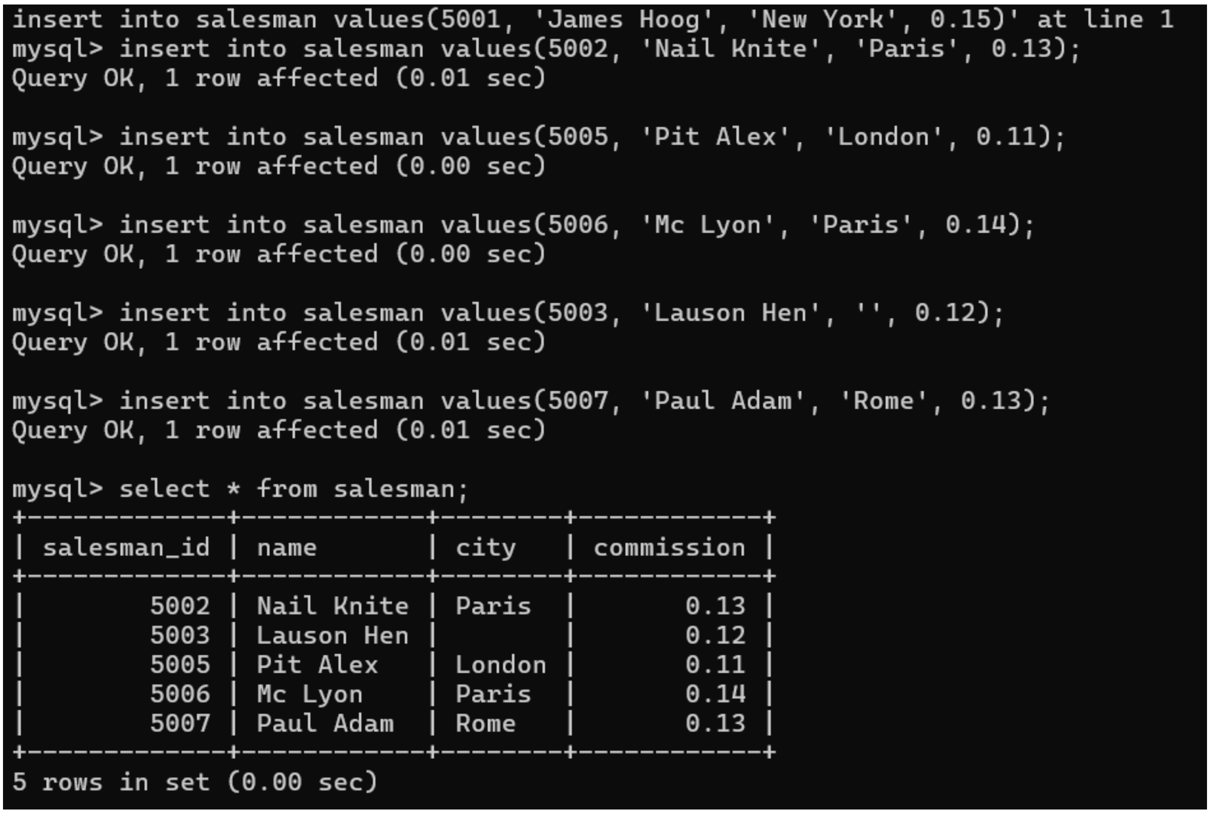
```
create database salesman;

use salesman

CREATE TABLE salesman(salesman_id INT NOT NULL
AUTO_INCREMENT PRIMARY key,

name VARCHAR(100)NOT NULL,

city VARCHAR(100)NOT NULL,

commission DECIMAL(10,2)

 );


desc salesman;
```



```
mysql> create database salesman;
Query OK, 1 row affected (0.01 sec)

mysql> use salesman
Database changed
mysql> CREATE TABLE salesman(salesman_id INT NOT NULL AUTO_INCREMENT PRIMARY
 key,
    -> name VARCHAR(100)NOT NULL,
    -> city VARCHAR(100)NOT NULL,
    -> commission DECIMAL(10,2)
    ->  );
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> desc salesman;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| salesman_id | int           | NO   | PRI | NULL    | auto_increment |
| name        | varchar(100)  | NO   |     | NULL    |                |
| city        | varchar(100)  | NO   |     | NULL    |                |
| commission  | decimal(10,2) | YES  |     | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
4 rows in set (0.02 sec)
```

Code:

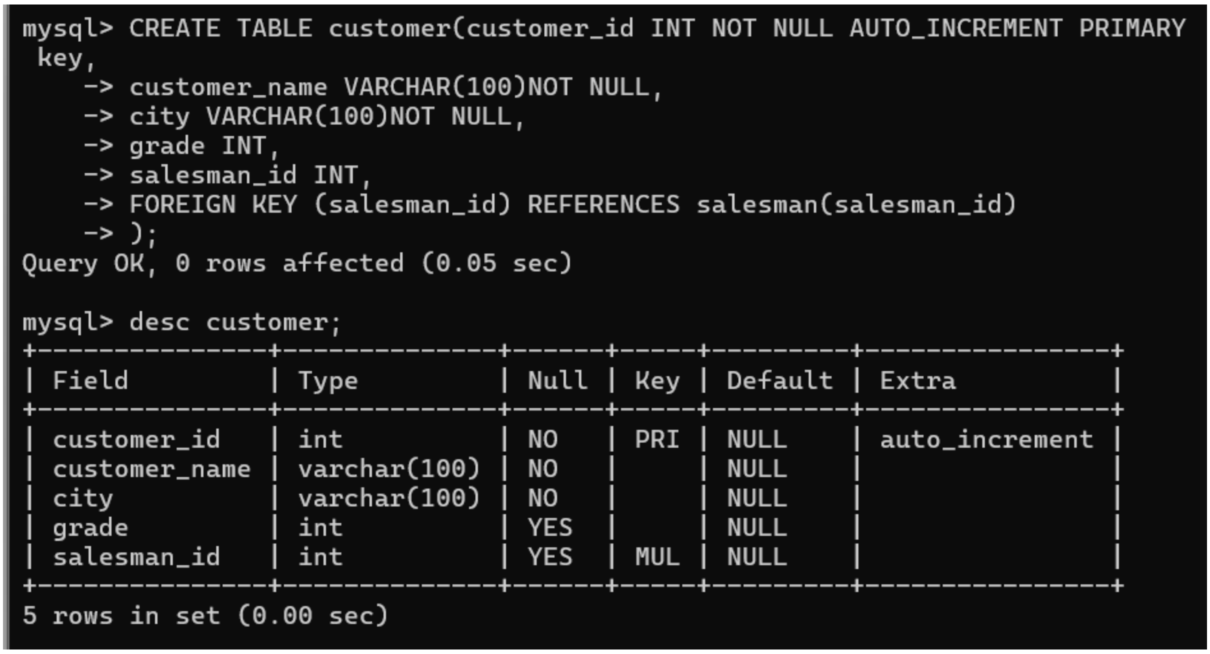
```
insert into salesman values(5001, 'James Hoog', 'New York', 0.15);
insert into salesman values(5002, 'Nail Knite', 'Paris', 0.13);
insert into salesman values(5005, 'Pit Alex', 'London', 0.11);
insert into salesman values(5006, 'Mc Lyon', 'Paris', 0.14);
insert into salesman values(5003, 'Lauson Hen', '', 0.12);
insert into salesman values(5007, 'Paul Adam', 'Rome', 0.13);
select * from salesman;
```



```
insert into salesman values(5001, 'James Hoog', 'New York', 0.15)' at line 1
mysql> insert into salesman values(5002, 'Nail Knite', 'Paris', 0.13);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5005, 'Pit Alex', 'London', 0.11);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5006, 'Mc Lyon', 'Paris', 0.14);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5003, 'Lauson Hen', '', 0.12);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5007, 'Paul Adam', 'Rome', 0.13);
Query OK, 1 row affected (0.01 sec)

mysql> select * from salesman;
+-------------+------------+--------+------------+
| salesman_id | name       | city   | commission |
+-------------+------------+--------+------------+
|        5002 | Nail Knite | Paris  |       0.13 |
|        5003 | Lauson Hen |        |       0.12 |
|        5005 | Pit Alex   | London |       0.11 |
|        5006 | Mc Lyon    | Paris  |       0.14 |
|        5007 | Paul Adam  | Rome   |       0.13 |
+-------------+------------+--------+------------+
5 rows in set (0.00 sec)
```

Code:

```
CREATE TABLE customer(customer_id INT NOT NULL
AUTO_INCREMENT PRIMARY key,

customer_name VARCHAR(100)NOT NULL,

city VARCHAR(100)NOT NULL,

grade INT,

salesman_id INT,

FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)

);

desc customer;
```



```
mysql> CREATE TABLE customer(customer_id INT NOT NULL AUTO_INCREMENT PRIMARY
 key,
    -> customer_name VARCHAR(100)NOT NULL,
    -> city VARCHAR(100)NOT NULL,
    -> grade INT,
    -> salesman_id INT,
    -> FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc customer;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| customer_id   | int          | NO   | PRI | NULL    | auto_increment |
| customer_name | varchar(100) | NO   |     | NULL    |                |
| city          | varchar(100) | NO   |     | NULL    |                |
| grade         | int          | YES  |     | NULL    |                |
| salesman_id   | int          | YES  | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Code:

insert into customer values(3002, 'Nick Rimando', 'New York', 100, 5001);

insert into customer values(3005, 'Graham Zusi', 'California', 200, 5002);

insert into customer values(3001, 'Brad Guzan', 'Londan', 100, 5005);

insert into customer values(3004, 'Fabian Johns', 'Paris', 300, 5006);

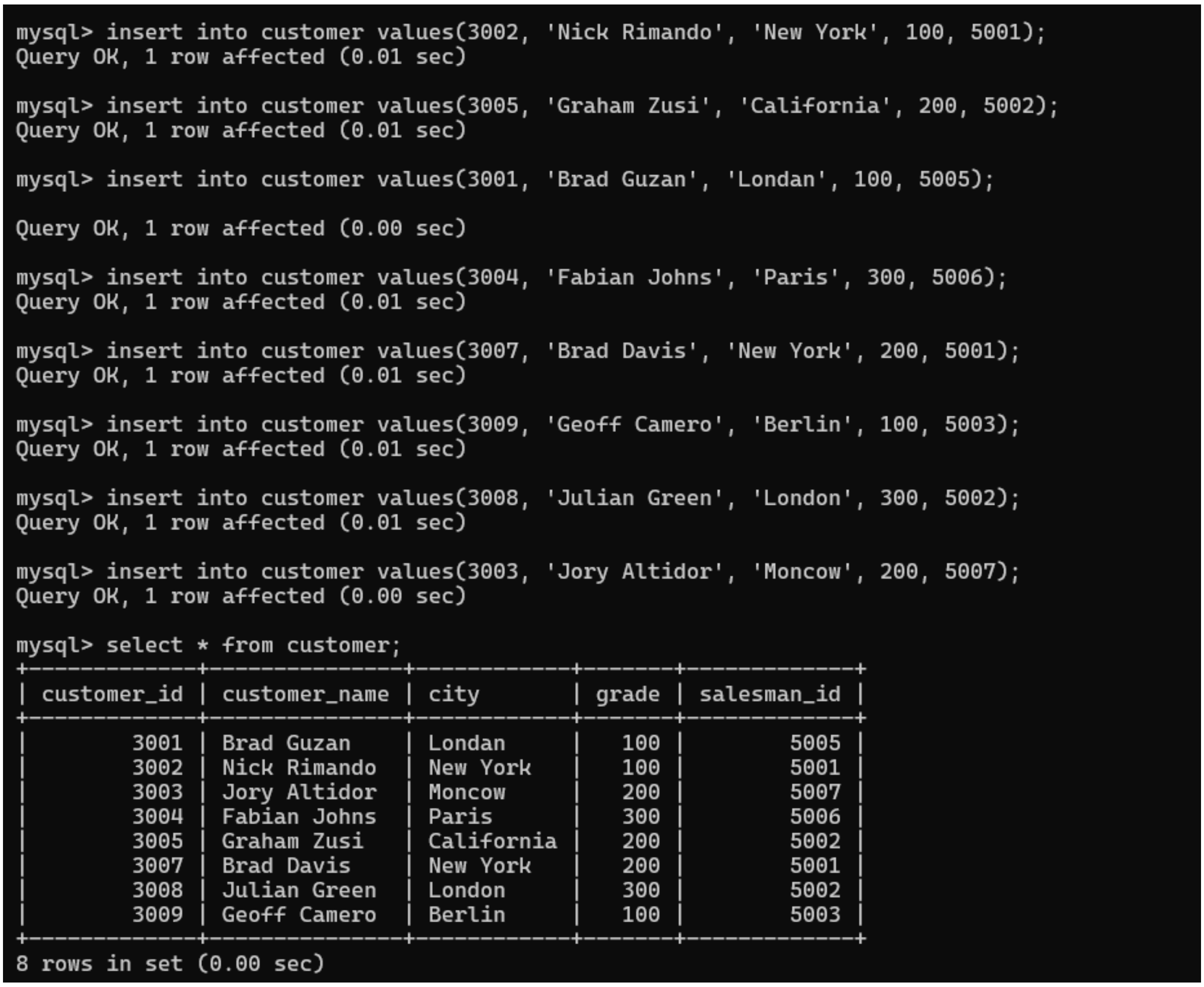
insert into customer values(3007, 'Brad Davis', 'New York', 200, 5001);

insert into customer values(3009, 'Geoff Camero', 'Berlin', 100, 5003);

insert into customer values(3008, 'Julian Green', 'London', 300, 5002);

insert into customer values(3003, 'Jory Altidor', 'Moncow', 200, 5007);

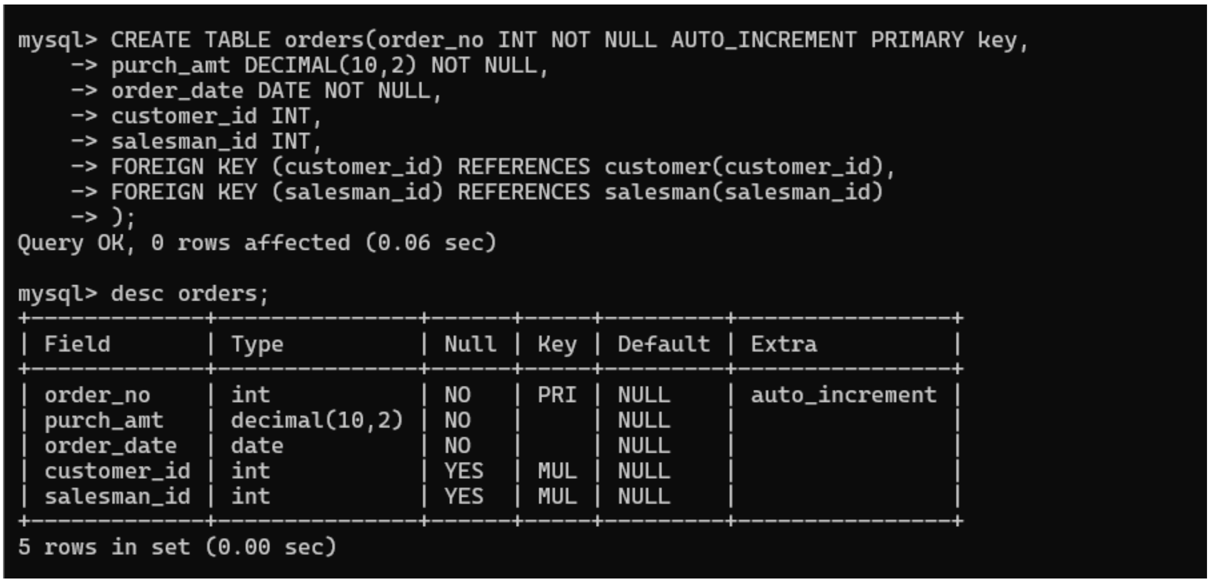
select * from customer;



```
mysql> insert into customer values(3002, 'Nick Rimando', 'New York', 100, 5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3005, 'Graham Zusi', 'California', 200, 5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3001, 'Brad Guzan', 'Londan', 100, 5005);

Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3004, 'Fabian Johns', 'Paris', 300, 5006);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3007, 'Brad Davis', 'New York', 200, 5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3009, 'Geoff Camero', 'Berlin', 100, 5003);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3008, 'Julian Green', 'London', 300, 5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3003, 'Jory Altidor', 'Moncow', 200, 5007);
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer;
+-------------+---------------+------------+-------+-------------+
| customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3001 | Brad Guzan    | Londan     |   100 |        5005 |
|        3002 | Nick Rimando  | New York   |   100 |        5001 |
|        3003 | Jory Altidor  | Moncow     |   200 |        5007 |
|        3004 | Fabian Johns  | Paris      |   300 |        5006 |
|        3005 | Graham Zusi   | California |   200 |        5002 |
|        3007 | Brad Davis    | New York   |   200 |        5001 |
|        3008 | Julian Green  | London     |   300 |        5002 |
|        3009 | Geoff Camero  | Berlin     |   100 |        5003 |
+-------------+---------------+------------+-------+-------------+
8 rows in set (0.00 sec)
```

Code:

```
CREATE TABLE orders(order_no INT NOT NULL AUTO_INCREMENT PRIMARY key,

purch_amt DECIMAL(10,2) NOT NULL,

order_date DATE NOT NULL,

customer_id INT,

salesman_id INT,

FOREIGN KEY (customer_id) REFERENCES customer(customer_id),

FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)

);

desc orders;
```



```
mysql> CREATE TABLE orders(order_no INT NOT NULL AUTO_INCREMENT PRIMARY key,
    -> purch_amt DECIMAL(10,2) NOT NULL,
    -> order_date DATE NOT NULL,
    -> customer_id INT,
    -> salesman_id INT,
    -> FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
    -> FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc orders;
+-------------+---------------+------+-----+---------+----------------+
| Field       | Type          | Null | Key | Default | Extra          |
+-------------+---------------+------+-----+---------+----------------+
| order_no    | int           | NO   | PRI | NULL    | auto_increment |
| purch_amt   | decimal(10,2) | NO   |     | NULL    |                |
| order_date  | date          | NO   |     | NULL    |                |
| customer_id | int           | YES  | MUL | NULL    |                |
| salesman_id | int           | YES  | MUL | NULL    |                |
+-------------+---------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Code:

insert into orders values(70001, 150.5, '2016-10-05', 3005, 5002);

insert into orders values(70009, 270.65, '2016-09-10', 3001, NULL);

insert into orders values(70002, 65.26, '2016-10-05', 3002, 5001);

insert into orders values(70004, 110.5, '2016-08-17', 3009, NULL);

insert into orders values(70007, 948.5, '2016-09-10', 3005, 5002);

insert into orders values(70005, 2400.6, '2016-07-27', 3007, 5001);

insert into orders values(70008, 5760, '2016-09-10', 3002, 5001);

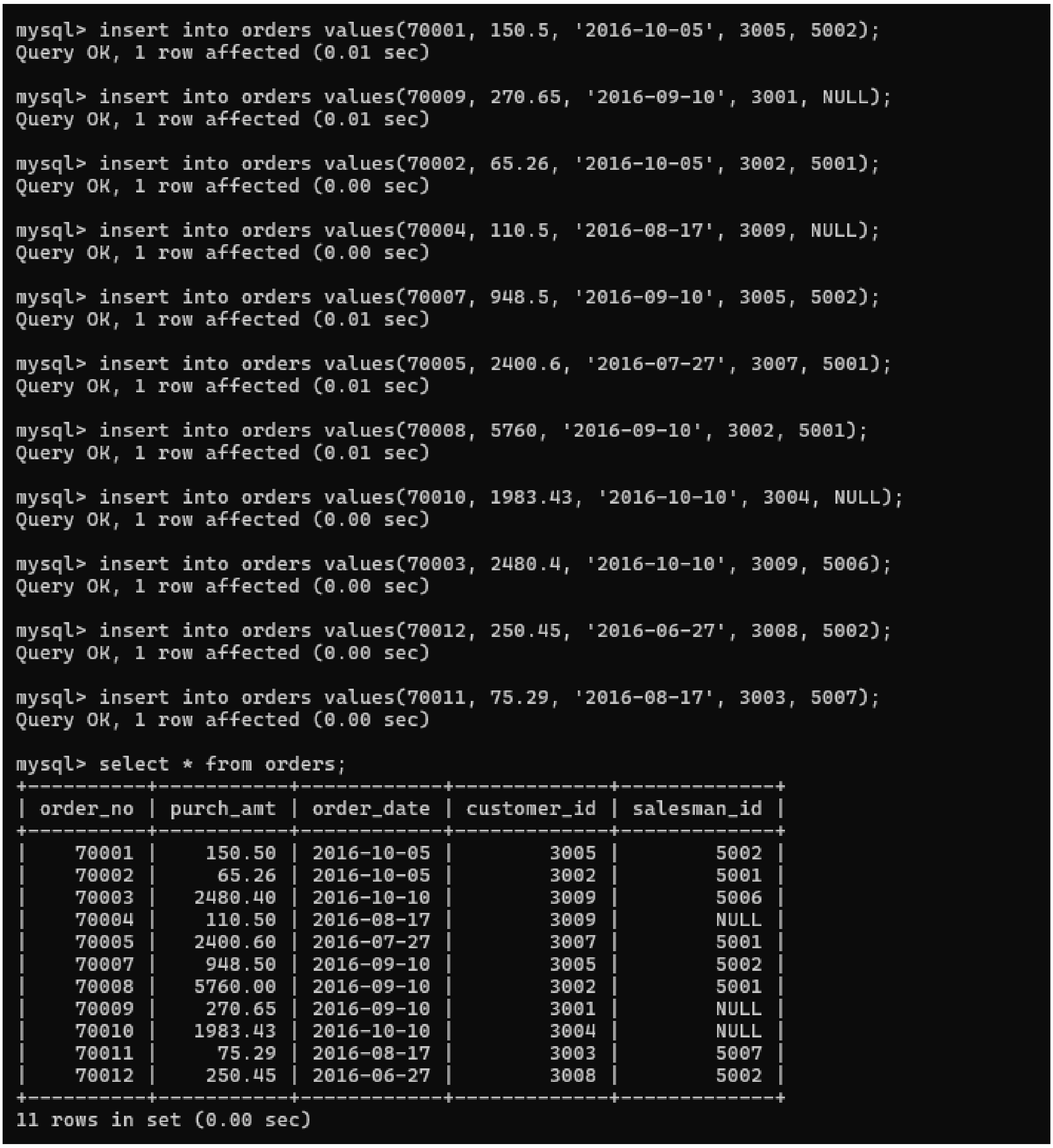
insert into orders values(70010, 1983.43, '2016-10-10', 3004, NULL);

insert into orders values(70003, 2480.4, '2016-10-10', 3009, 5006);

insert into orders values(70012, 250.45, '2016-06-27', 3008, 5002);

insert into orders values(70011, 75.29, '2016-08-17', 3003, 5007);

select * from orders;



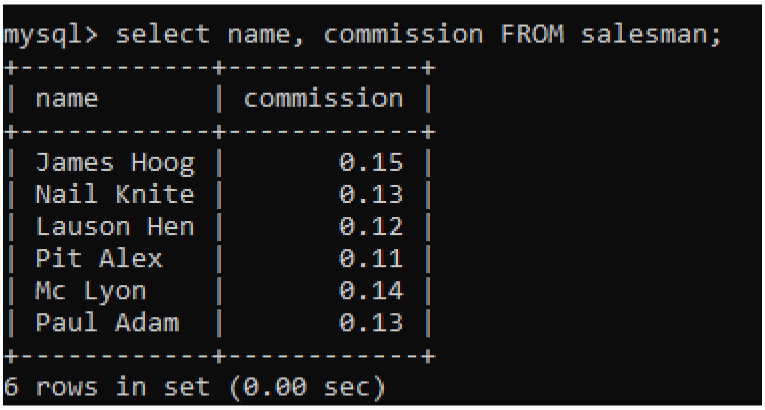
```
mysql> insert into orders values(70001, 150.5, '2016-10-05', 3005, 5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70009, 270.65, '2016-09-10', 3001, NULL);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70002, 65.26, '2016-10-05', 3002, 5001);
Query OK, 1 row affected (0.00 sec)

mysql> insert into orders values(70004, 110.5, '2016-08-17', 3009, NULL);
Query OK, 1 row affected (0.00 sec)

mysql> insert into orders values(70007, 948.5, '2016-09-10', 3005, 5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70005, 2400.6, '2016-07-27', 3007, 5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70008, 5760, '2016-09-10', 3002, 5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70010, 1983.43, '2016-10-10', 3004, NULL);
Query OK, 1 row affected (0.00 sec)

mysql> insert into orders values(70003, 2480.4, '2016-10-10', 3009, 5006);
Query OK, 1 row affected (0.00 sec)

mysql> insert into orders values(70012, 250.45, '2016-06-27', 3008, 5002);
Query OK, 1 row affected (0.00 sec)

mysql> insert into orders values(70011, 75.29, '2016-08-17', 3003, 5007);
Query OK, 1 row affected (0.00 sec)

mysql> select * from orders;
+----------+-----------+------------+-------------+-------------+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+----------+-----------+------------+-------------+-------------+
|    70001 |    150.50 | 2016-10-05 |        3005 |        5002 |
|    70002 |     65.26 | 2016-10-05 |        3002 |        5001 |
|    70003 |   2480.40 | 2016-10-10 |        3009 |        5006 |
|    70004 |    110.50 | 2016-08-17 |        3009 |        NULL |
|    70005 |   2400.60 | 2016-07-27 |        3007 |        5001 |
|    70007 |    948.50 | 2016-09-10 |        3005 |        5002 |
|    70008 |   5760.00 | 2016-09-10 |        3002 |        5001 |
|    70009 |    270.65 | 2016-09-10 |        3001 |        NULL |
|    70010 |   1983.43 | 2016-10-10 |        3004 |        NULL |
|    70011 |     75.29 | 2016-08-17 |        3003 |        5007 |
|    70012 |    250.45 | 2016-06-27 |        3008 |        5002 |
+----------+-----------+------------+-------------+-------------+
11 rows in set (0.00 sec)
```

1. Display name and commission for all the salesmen.

Code:

select name, commission FROM salesman;

Output:



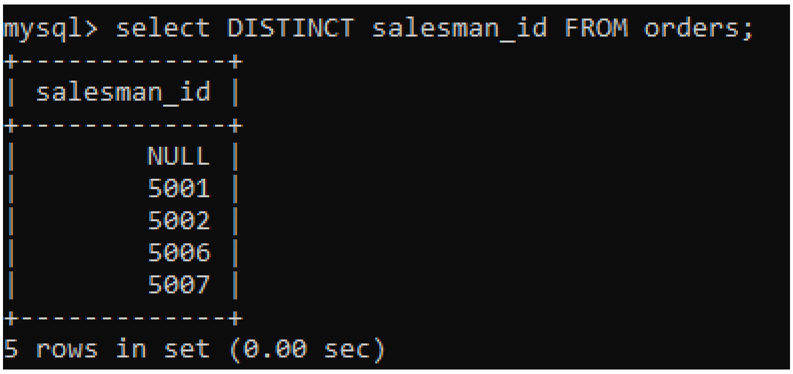
```
mysql> select name, commission FROM salesman;
+------------+------------+
| name       | commission |
+------------+------------+
| James Hoog |       0.15 |
| Nail Knite |       0.13 |
| Lauson Hen |       0.12 |
| Pit Alex   |       0.11 |
| Mc Lyon    |       0.14 |
| Paul Adam  |       0.13 |
+------------+------------+
6 rows in set (0.00 sec)
```

2. Retrieve salesman id of all salesmen from orders table without any repeats.

Code:

select DISTINCT salesman_id FROM orders;

Output:



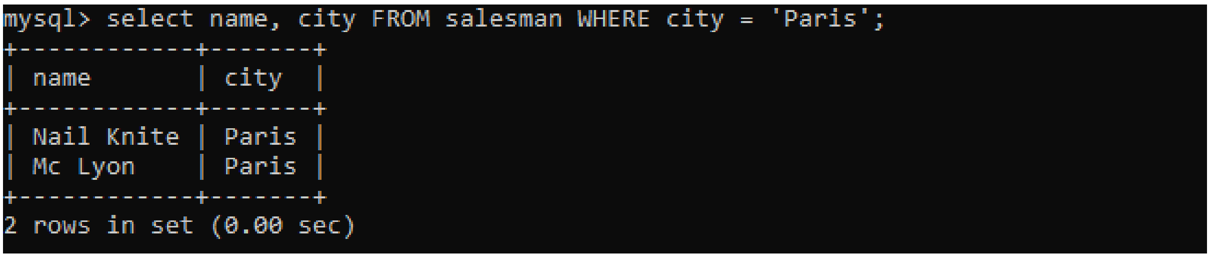
```
mysql> select DISTINCT salesman_id FROM orders;
+-------------+
| salesman_id |
+-------------+
|        NULL |
|        5001 |
|        5002 |
|        5006 |
|        5007 |
+-------------+
5 rows in set (0.00 sec)
```

3. Display names and city of salesman, who belongs to the city of Paris.

Code:

select name, city FROM salesman WHERE city = 'Paris';

Output:



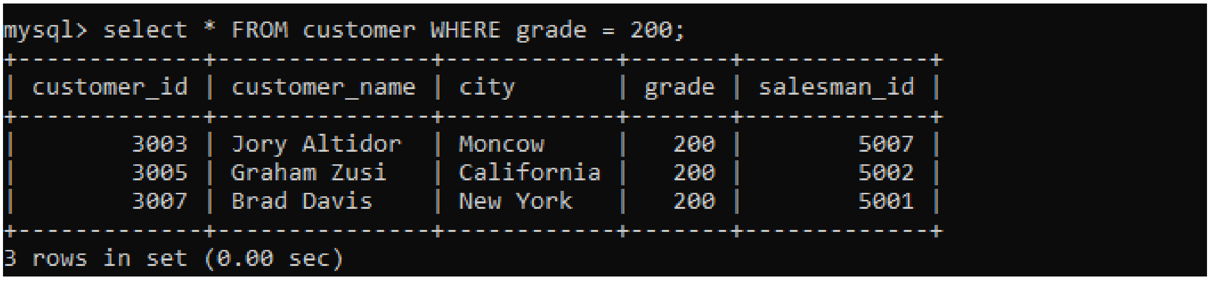
```
mysql> select name, city FROM salesman WHERE city = 'Paris';
+------------+-------+
| name       | city  |
+------------+-------+
| Nail Knite | Paris |
| Mc Lyon    | Paris |
+------------+-------+
2 rows in set (0.00 sec)
```

4. Display all the information for those customers with a grade of 200.

Code:

select * FROM customer WHERE grade = 200;

Output:



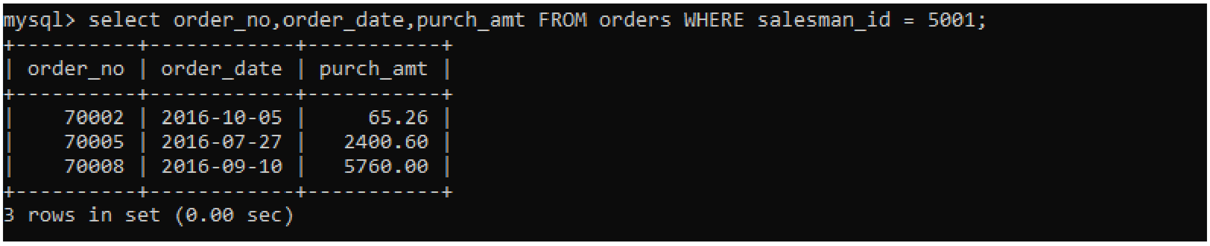
```
mysql> select * FROM customer WHERE grade = 200;
+-------------+---------------+------------+-------+-------------+
| customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3003 | Jory Altidor  | Moncow     |   200 |        5007 |
|        3005 | Graham Zusi   | California |   200 |        5002 |
|        3007 | Brad Davis    | New York   |   200 |        5001 |
+-------------+---------------+------------+-------+-------------+
3 rows in set (0.00 sec)
```

5. Display the order number, order date and the purchase amount for order(s) which will be delivered by the salesman with ID 5001

Code:

select order_no,order_date,purch_amt FROM orders WHERE salesman_id = 5001;

Output:



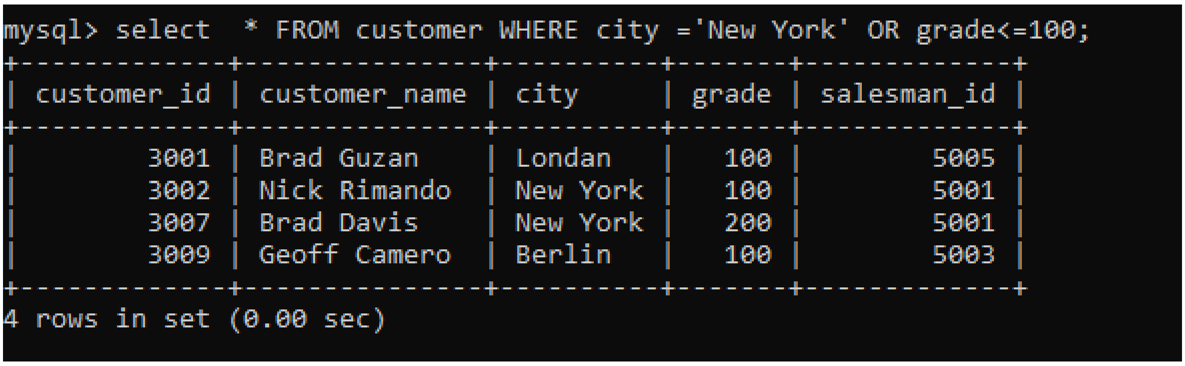
```
mysql> select order_no,order_date,purch_amt FROM orders WHERE salesman_id = 5001;
+----------+------------+-----------+
| order_no | order_date | purch_amt |
+----------+------------+-----------+
|    70002 | 2016-10-05 |     65.26 |
|    70005 | 2016-07-27 |   2400.60 |
|    70008 | 2016-09-10 |   5760.00 |
+----------+------------+-----------+
3 rows in set (0.00 sec)
```

12. Display all the customers, who are either belongs to the city New York or not had a grade above 100.

Code:

select * FROM customer WHERE city ='New York' OR grade<=100;

Output:



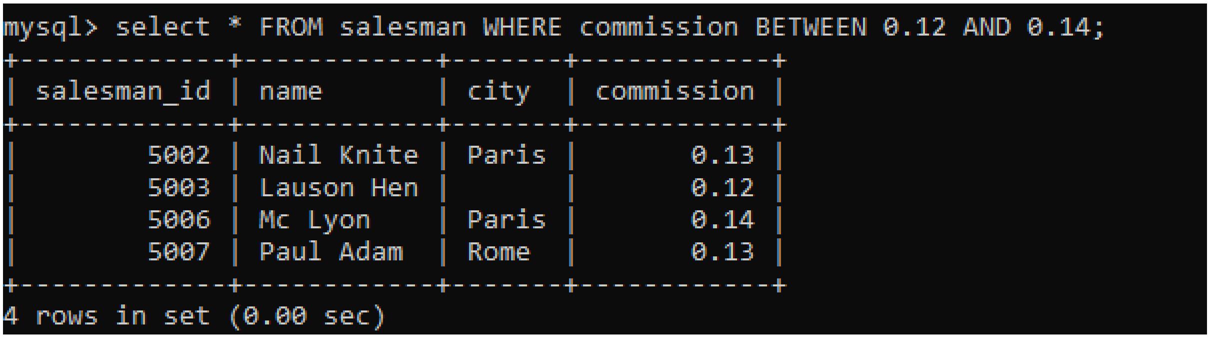
```
mysql> select  * FROM customer WHERE city ='New York' OR grade<=100;
+-------------+---------------+----------+-------+-------------+
| customer_id | customer_name | city     | grade | salesman_id |
+-------------+---------------+----------+-------+-------------+
|        3001 | Brad Guzan    | Londan   |   100 |        5005 |
|        3002 | Nick Rimando  | New York |   100 |        5001 |
|        3007 | Brad Davis    | New York |   200 |        5001 |
|        3009 | Geoff Camero  | Berlin   |   100 |        5003 |
+-------------+---------------+----------+-------+-------------+
4 rows in set (0.00 sec)
```

13.Find those salesmen with all information who gets the commission within a range of 0.12 and 0.14.

Code:

select * FROM salesman WHERE commission BETWEEN 0.12 AND 0.14;

Output:



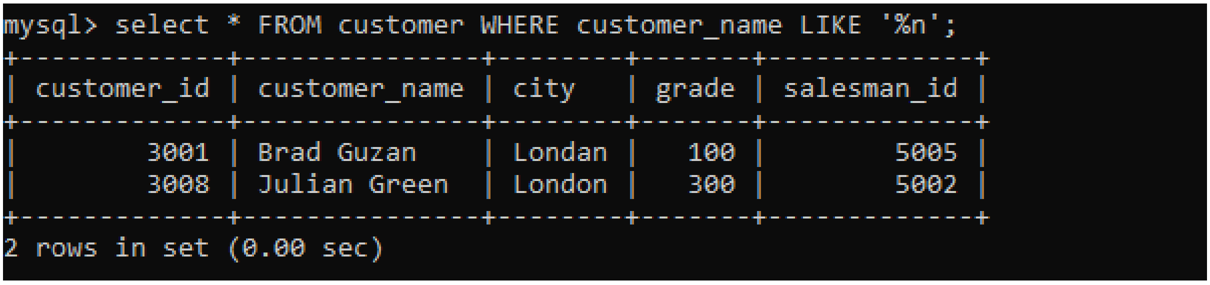
```
mysql> select * FROM salesman WHERE commission BETWEEN 0.12 AND 0.14;
+-------------+------------+-------+------------+
| salesman_id | name       | city  | commission |
+-------------+------------+-------+------------+
|        5002 | Nail Knite | Paris |       0.13 |
|        5003 | Lauson Hen |       |       0.12 |
|        5006 | Mc Lyon    | Paris |       0.14 |
|        5007 | Paul Adam  | Rome  |       0.13 |
+-------------+------------+-------+------------+
4 rows in set (0.00 sec)
```

14. Find all those customers with all information whose names are ending with the letter 'n'.

Code:

select * FROM customer WHERE customer_name LIKE '%n';

Output:



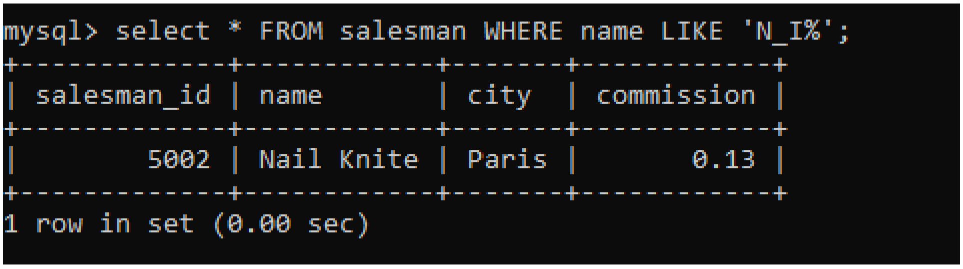
```
mysql> select * FROM customer WHERE customer_name LIKE '%n';
+-------------+---------------+--------+-------+-------------+
| customer_id | customer_name | city   | grade | salesman_id |
+-------------+---------------+--------+-------+-------------+
|        3001 | Brad Guzan    | Londan |   100 |        5005 |
|        3008 | Julian Green  | London |   300 |        5002 |
+-------------+---------------+--------+-------+-------------+
2 rows in set (0.00 sec)
```

15. Find those salesmen with all information whose name containing the 1st character is 'N' and the 4th character is 'l' and rests may be any character.

Code:

select * FROM salesman WHERE name LIKE 'N_l%';

Output:



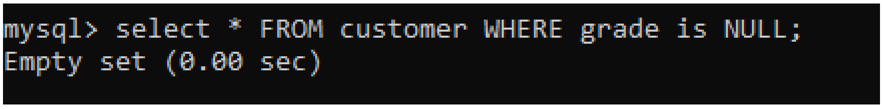
```
mysql> select * FROM salesman WHERE name LIKE 'N_I%';
+-------------+------------+-------+------------+
| salesman_id | name       | city  | commission |
+-------------+------------+-------+------------+
|        5002 | Nail Knite | Paris |       0.13 |
+-------------+------------+-------+------------+
1 row in set (0.00 sec)
```

16.Find that customer with all information who does not get any grade except NULL.

Code:

select * FROM customer WHERE grade is NULL;

Output:



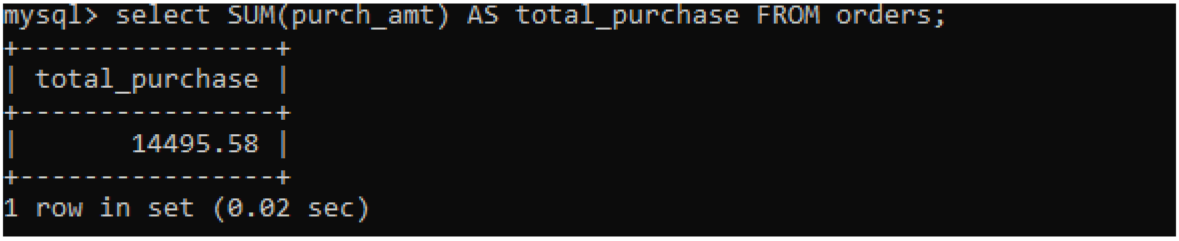
```
mysql> select * FROM customer WHERE grade is NULL;
Empty set (0.00 sec)
```

17. Find the total purchase amount of all orders.

Code:

select SUM(purch_amt) AS total_purchase FROM orders;

Output:



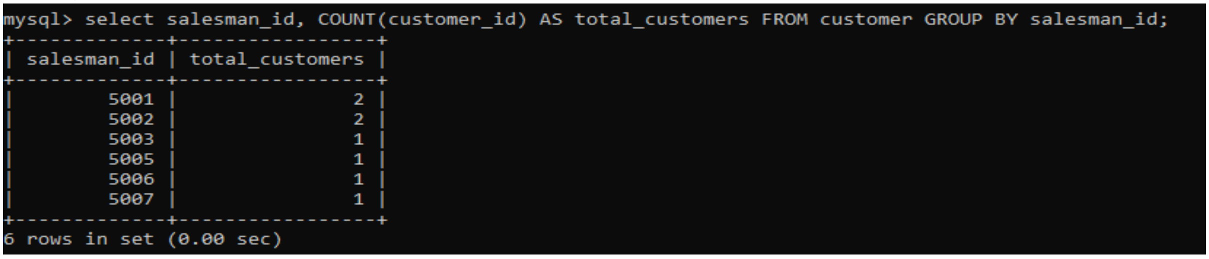
```
mysql> select SUM(purch_amt) AS total_purchase FROM orders;
+----------------+
| total_purchase |
+----------------+
|       14495.58 |
+----------------+
1 row in set (0.02 sec)
```

18. Find the number of salesman currently listing for all of their customers.

Code:

select salesman_id, COUNT(customer_id) AS total_customers FROM customer GROUP BY salesman_id;

Output:



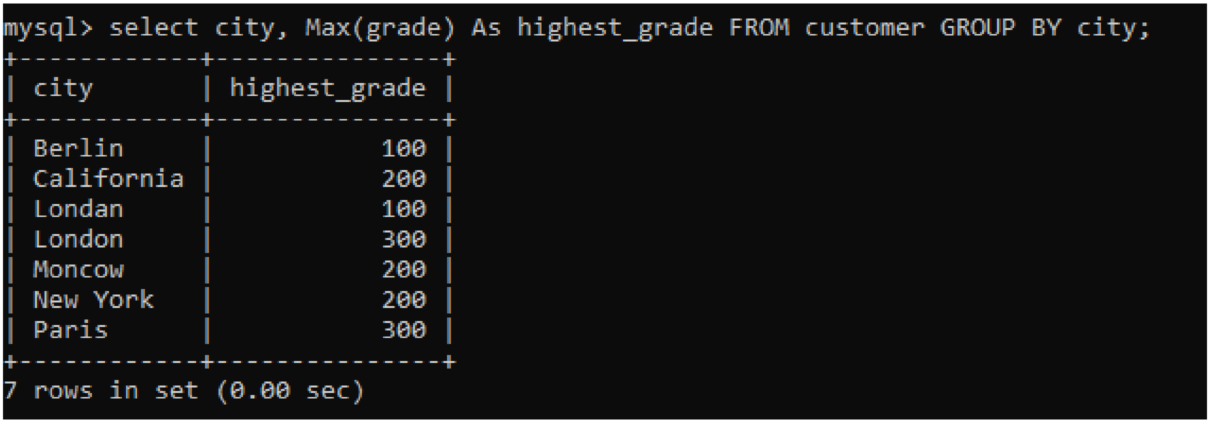
```
mysql> select salesman_id, COUNT(customer_id) AS total_customers FROM customer GROUP BY salesman_id;
+-------------+-----------------+
| salesman_id | total_customers |
+-------------+-----------------+
|        5001 |               2 |
|        5002 |               2 |
|        5003 |               1 |
|        5005 |               1 |
|        5006 |               1 |
|        5007 |               1 |
+-------------+-----------------+
6 rows in set (0.00 sec)
```

19.Find the highest grade for each of the cities of the customers.

Code:

select city, Max(grade) As highest_grade FROM customer GROUP BY city;

Output:



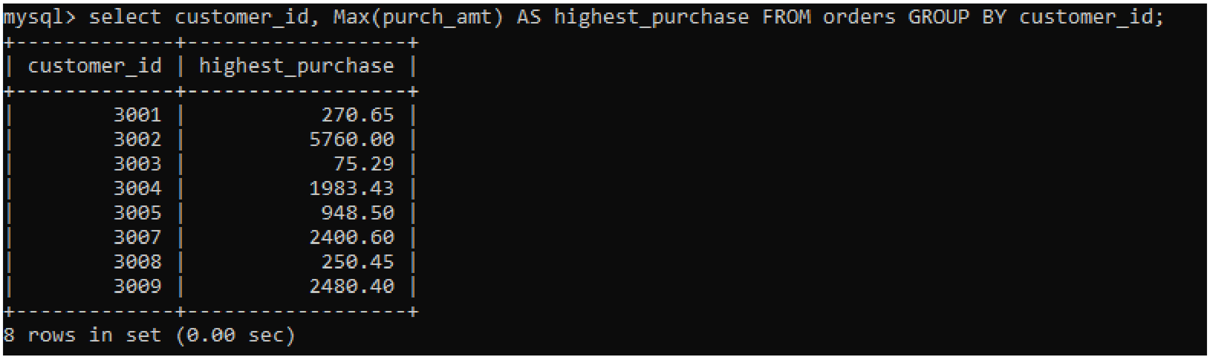
```
mysql> select city, Max(grade) As highest_grade FROM customer GROUP BY city;
+------------+---------------+
| city       | highest_grade |
+------------+---------------+
| Berlin     |           100 |
| California |           200 |
| Londan     |           100 |
| London     |           300 |
| Moncow     |           200 |
| New York   |           200 |
| Paris      |           300 |
+------------+---------------+
7 rows in set (0.00 sec)
```

20.Find the highest purchase amount ordered by each customer with their ID and highest purchase amount.

Code:

select customer_id, Max(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id;

Output:



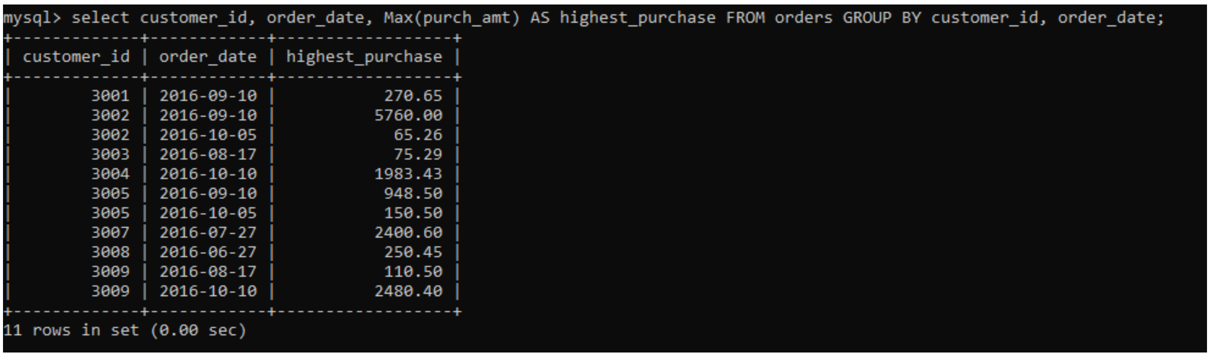
```
mysql> select customer_id, Max(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id;
+-------------+------------------+
| customer_id | highest_purchase |
+-------------+------------------+
|        3001 |           270.65 |
|        3002 |          5760.00 |
|        3003 |            75.29 |
|        3004 |          1983.43 |
|        3005 |           948.50 |
|        3007 |          2400.60 |
|        3008 |           250.45 |
|        3009 |          2480.40 |
+-------------+------------------+
8 rows in set (0.00 sec)
```

21.Find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount.

Code:

select customer_id, order_date, Max(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id, order_date;

Output:



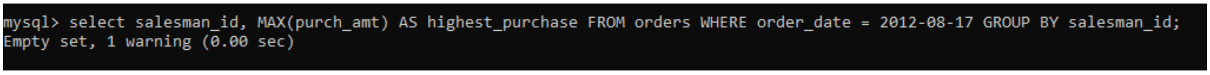
```
mysql> select customer_id, order_date, Max(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id, order_date;
+-------------+------------+------------------+
| customer_id | order_date | highest_purchase |
+-------------+------------+------------------+
|        3001 | 2016-09-10 |           270.65 |
|        3002 | 2016-09-10 |          5760.00 |
|        3002 | 2016-10-05 |            65.26 |
|        3003 | 2016-08-17 |            75.29 |
|        3004 | 2016-10-10 |          1983.43 |
|        3005 | 2016-09-10 |           948.50 |
|        3005 | 2016-10-05 |           150.50 |
|        3007 | 2016-07-27 |          2400.60 |
|        3008 | 2016-06-27 |           250.45 |
|        3009 | 2016-08-17 |           110.50 |
|        3009 | 2016-10-10 |          2480.40 |
+-------------+------------+------------------+
11 rows in set (0.00 sec)
```

22.Find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.

Code:

select salesman_id, MAX(purch_amt) AS highest_purchase FROM orders WHERE order_date = 2012-08-17 GROUP BY salesman_id;

Output:



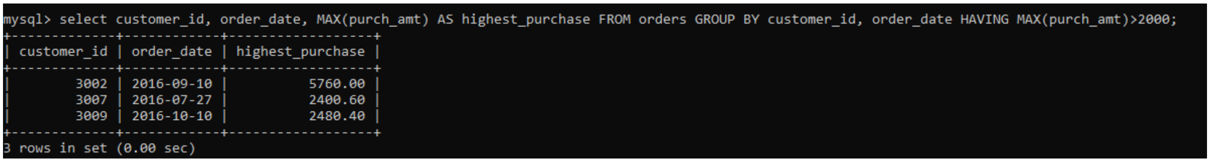
```
mysql> select salesman_id, MAX(purch_amt) AS highest_purchase FROM orders WHERE order_date = 2012-08-17 GROUP BY salesman_id;
Empty set, 1 warning (0.00 sec)
```

23. the highest purchase amount with their customer ID and order date, for only those customers who have the highest purchase amount in a day is more than 2000.

Code:

select customer_id, order_date, MAX(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id, order_date HAVING MAX(purch_amt)>2000;

Output:



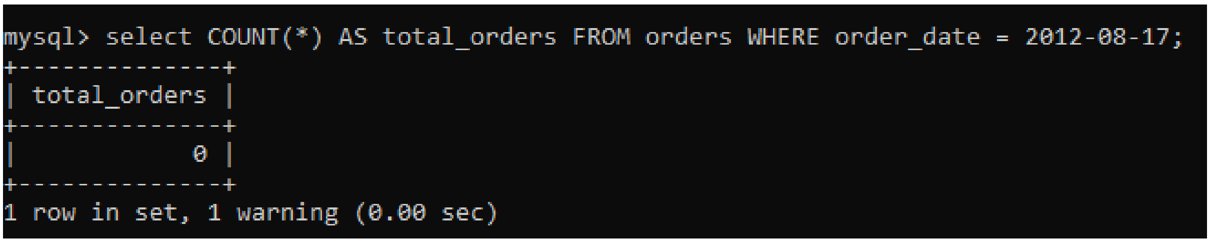
```
mysql> select customer_id, order_date, MAX(purch_amt) AS highest_purchase FROM orders GROUP BY customer_id, order_date HAVING MAX(purch_amt)>2000;
+-------------+------------+------------------+
| customer_id | order_date | highest_purchase |
+-------------+------------+------------------+
|        3002 | 2016-09-10 |          5760.00 |
|        3007 | 2016-07-27 |          2400.60 |
|        3009 | 2016-10-10 |          2480.40 |
+-------------+------------+------------------+
3 rows in set (0.00 sec)
```

24. Write a SQL statement that counts all orders for a date August 17th, 2012.

Code:

select COUNT(*) AS total_orders FROM orders WHERE order_date = 2012-08-17;

Output:



```
mysql> select COUNT(*) AS total_orders FROM orders WHERE order_date = 2012-08-17;
+--------------+
| total_orders |
+--------------+
|            0 |
+--------------+
1 row in set, 1 warning (0.00 sec)
```