DBMS LAB Program -2

2 Consider the following schema for **Order Database**:

salesman(salesman_id, name, city, commission)
customer(customer_id, cust_name, city, grade, salesman_id)
orders(ord_no, purchase_amt, ord_date, customer_id, salesman_id)

Write SQL queries to

1. Count the customers with grades above Bangalore’s average.

2. Find the name and numbers of all salesman who had more than one customer.

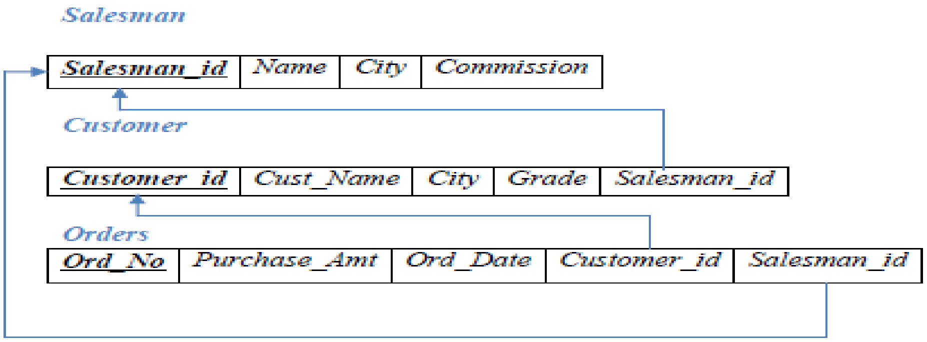
3. List all the salesman and indicate those who have and don’t have customers in their cities (Use UNION operation.)

4. Create a view that finds the salesman who has the customer with the highest order of a day.

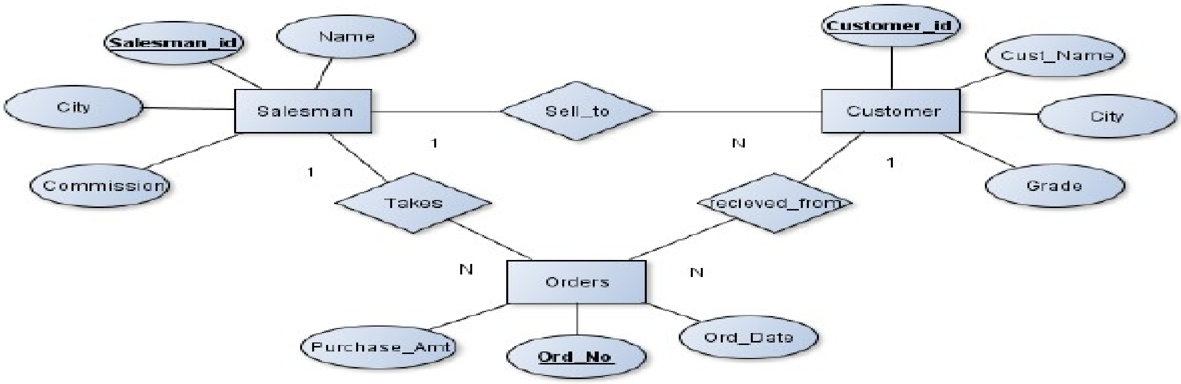
5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.

Solution:

Schema Diagram:



Entity –Relationship Diagram



Creating tables (Relations) and Insertion values into the tables:

create database orderdb;

use orderdb;

salesman(salesman_id, name, city, commission)

mysql> create table salesman (salesman_id int, name varchar(20), city varchar(20), commission varchar(20), primary key (salesman_id));

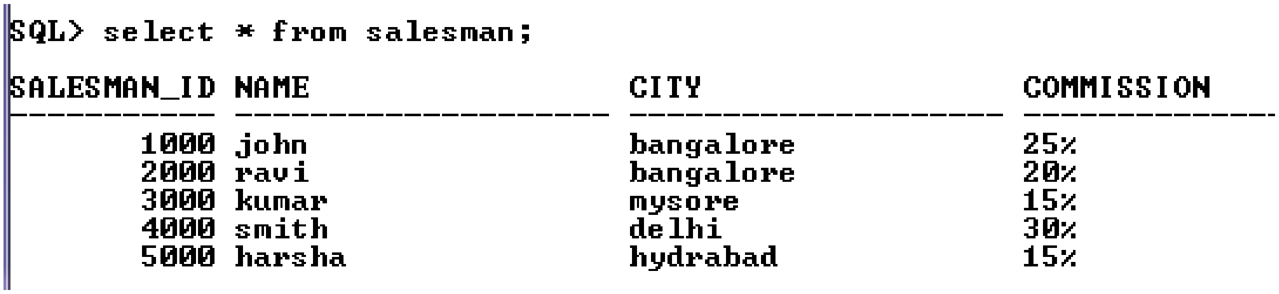
mysql>insert into salesman values (1000, "john","bangalore","25%");

mysql>insert into salesman values (2000, “ravi”,”bangalore”,”20%”);

mysql>insert into salesman values (3000, “kumar”,”mysore”,’”15%”);

mysql>insert into salesman values (4000, “smith”,”delhi”,’30%”);

mysql>insert into salesman values (5000,”harsha”,”hydrabad”,”15%”);



```
SQL> select * from salesman;

SALESMAN_ID NAME                 CITY                 COMMISSION
----------- -------------------- -------------------- --------------
       1000 john                 bangalore            25%
       2000 ravi                 bangalore            20%
       3000 kumar                mysore               15%
       4000 smith                delhi                30%
       5000 harsha               hydrabad             15%
```

customer(customer_id, cust_name, city, grade, salesman_id)

- **mysql>** create table customer (customer_id int, cust_name varchar(20), city varchar(20), grade int, salesman_id int, primary key (customer_id), Foreign Key (salesman_id) references salesman (salesman_id) on delete set null);

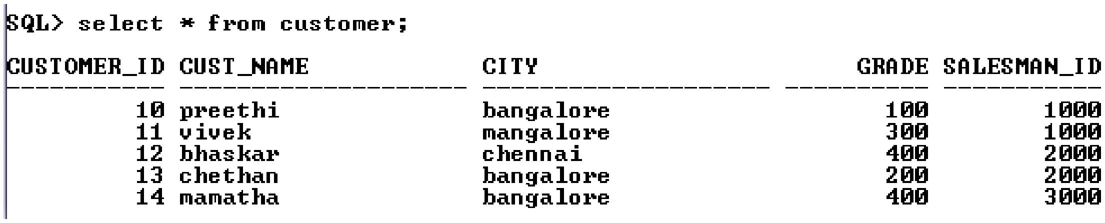
mysql>insert into customer values (10, ’preethi’,’bangalore’, 100, 1000);

mysql>insert into customer values (11, ’vivek’,’mangalore’, 300, 1000);

mysql>insert into customer values (12, ’bhaskar’,’chennai’, 400, 2000);

mysql>insert into customer values (13, ’chethan’,’bangalore’, 200, 2000);

mysql>insert into customer values (14, ’mamatha’,’bangalore’, 400, 3000);



```
SQL> select * from customer;

CUSTOMER_ID CUST_NAME            CITY                      GRADE SALESMAN_ID
----------- -------------------- -------------------- ---------- -----------
         10 preethi              bangalore                   100        1000
         11 vivek                mangalore                   300        1000
         12 bhaskar              chennai                     400        2000
         13 chethan              bangalore                   200        2000
         14 mamatha              bangalore                   400        3000
```

orders(ord_no, purchase_amt, ord_date, customer_id, salesman_id)

- **mysql>** create table orders (ord_no int, purchase_amt int, ord_date date, customer_id int, salesman_id int, primary key (ord_no), Foreign Key (customer_id) references customer (customer_id) on delete cascade, Foreign Key (salesman_id) references salesman (salesman_id) on delete cascade);

mysql>insert into orders values (50, 5000, '2017-05-04', 10, 1000);

mysql>insert into orders values (51, 450, ’2017-01-20’, 10, 2000);

mysql>insert into orders values (52, 1000, ’2017-02-24’, 13, 2000);

mysql>insert into orders values (53, 3500, ’2017-04-13’, 14, 3000);

mysql>insert into orders values (54, 550, ’2017-03-17’, 12, 2000);

mysql> select * from orders;

```
+--------+-----------------+------------+----------------+-----------------+
| ord_no | purchase_amt | ord_date   | customer_id    | salesman_id  |
+--------+----------------+------------+-----------------+-----------------+
|    50  |       5000      | 2017-05-04 |        10          |       1000       |
|    51  |        450      | 2017-01-20 |        10          |       2000       |
|    52  |       1000       | 2017-02-24 |         13         |       2000       |
|    53  |       3500      | 2017-04-13 |        14          |       3000       |
|    54  |        550      | 2017-03-09 |        12          |       2000       |
```

Queries

1. Count the customers with grades above Bangalore's average.

```
mysql> select avg(grade) from customer where city ='bangalore';
```

```
+------------+
| avg(grade) |
+------------+
|   233.3333 |
+------------+
```

```
mysql> select grade, count(*) from customer group by grade having grade > (select
           avg(grade) from customer where city="bangalore");
```

```
+-------+----------+
| grade | count(*) |
+-------+----------+
|   300 |       1      |
|   400 |       2      |
+-------+----------+
```

```
mysql> select grade, count(distinct customer_id) from customer group by grade having
           grade > (select avg(grade) from customer where city="bangalore");
```

```
+-------+----------------------------------+
| grade | count(distinct customer_id)          |
+-------+----------------------------------+
|   300  |                          1                  |
|   400  |                          2                  |
+-------+----------------------------------+
```

2. Find the name and numbers of all salesman who had more than one customer.

```
mysql> select s.salesman_id, name from salesman s, customer c
           where s.salesman_id=c.salesman_id group by s.salesman_id, name
            having count(*)>1;
```

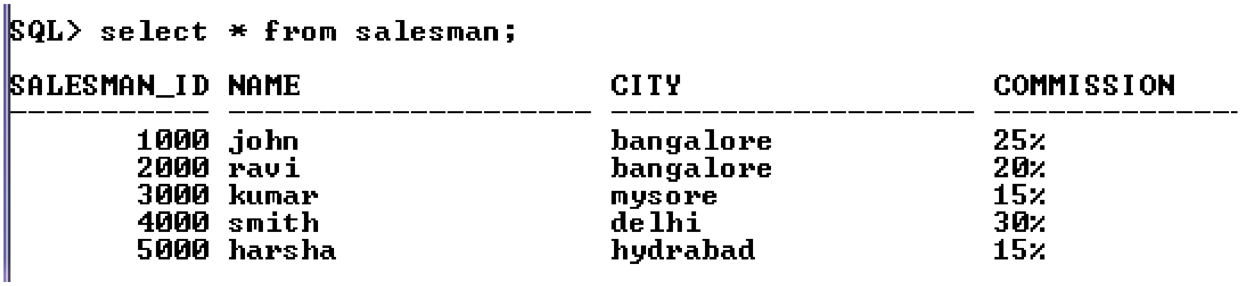
```
 +----------------+------+
 | salesman_id | name |
 +----------------+------+
|        1000       | john |
|        2000       | ravi   |
+-----------------+--------+
```

3. List all salesmen and indicate those who have and don’t have customers in their cities (Use UNION operation.)

mysql> select **s.salesman_id, s.name, c.cust_name, s.commission** from **salesman s, customer c** where **s.city = c.city and s.salesman_id=c.salesman_id**

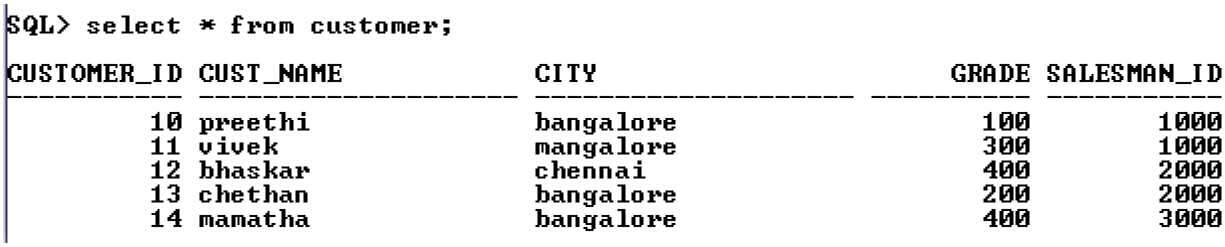
union

select s1.salesman_id, s1.name, "no match", s1.commission from salesman s1, customer c1 where **s1.city != c1.city and s1.salesman_id=c1.salesman_id;**



```
SQL> select * from salesman;

SALESMAN_ID NAME                 CITY                 COMMISSION
----------- -------------------- -------------------- --------------
       1000 john                 bangalore            25%
       2000 ravi                 bangalore            20%
       3000 kumar                mysore               15%
       4000 smith                delhi                30%
       5000 harsha               hydrabad             15%
```



```
SQL> select * from customer;

CUSTOMER_ID CUST_NAME            CITY                      GRADE SALESMAN_ID
----------- -------------------- -------------------- ---------- -----------
         10 preethi              bangalore                   100        1000
         11 vivek                mangalore                   300        1000
         12 bhaskar              chennai                     400        2000
         13 chethan              bangalore                   200        2000
         14 mamatha              bangalore                   400        3000
```

```
+----------------+--------+-------------+---------------+
| salesman_id  | name   | cust_name | commission |
+----------------+--------+-------------+---------------+
|       1000     | john     | preethi       | 25%            |
|       2000     | ravi      | chethan      | 20%            |
|       1000     | john     | no match    | 25%           |
|       2000     | ravi      | no match    | 20%           |
|       3000     | kumar  | no match    | 15%           |
+----------------+--------+--------------+-------------+
```

4. Create a view that finds the salesman who has the customer with the highest order of a day.

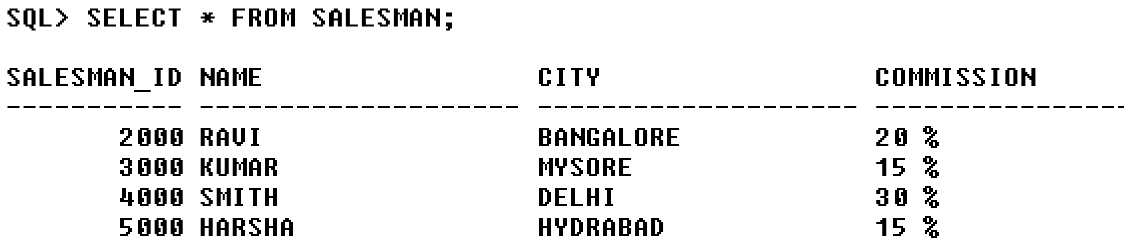
mysql> create view **salesman_view** as select **o.ord_date, s.salesman_id, s.name, o.purchase_amt** from **salesman s, orders o** where **s.salesman_id = o.salesman_id and o.purchase_amt** in (select max(purchase_amt) from orders o1 where o1.ord_date = o.ord_date);

mysql> select * from salesman_view;

```
+--------------+----------------+----------+-------------------+
| ord_date     | salesman_id | name      | purchase_amt |
+--------------+----------------+----------+-------------------+
| 2017-05-04 |       1000      | john     |        5000         |
| 2017-01-20 |       2000      | ravi      |         450          |
| 2017-02-24 |       2000      | ravi      |        1000         |
| 2017-04-13 |       3000      | kumar |        3500         |
| 2017-03-09 |       2000      | ravi      |         550          |
+--------------+----------------+----------+-------------------+
```

5. Demonstrate the DELETE operation by removing salesman with id 1000.
All his orders must also be deleted.

Delete from salesman where salesman_id=1000;



```
SQL> SELECT * FROM SALESMAN;

SALESMAN_ID NAME                 CITY                 COMMISSION
----------- -------------------- -------------------- ----------------
       2000 RAVI                 BANGALORE            20 %
       3000 KUMAR                MYSORE               15 %
       4000 SMITH                DELHI                30 %
       5000 HARSHA               HYDRABAD             15 %
```