Program 6:

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)

mysql> create database orders;

Query OK, 1 row affected (0.14 sec)

mysql> use orders;

Database changed

mysql> create table salesman(

-> salesman\_id int primary key,

-> name varchar(20),

-> city varchar(20),

-> commission varchar(10));

Query OK, 0 rows affected (2.18 sec)

mysql> describe salesman;

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

| Field | Type | Null | Key | Default | Extra |

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

| salesman\_id | int | NO | PRI | NULL | |

| name | varchar(20) | YES | | NULL | |

| city | varchar(20) | YES | | NULL | |

| commission | varchar(10) | YES | | NULL | |

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

4 rows in set (0.43 sec)

mysql> create table customer(

-> customer\_id int primary key,

-> cust\_name varchar(20),

-> city varchar(20),

-> grade int,

-> salesman\_id int,

-> foreign key(salesman\_id) references salesman(salesman\_id) on update cascade on delete cascade);

Query OK, 0 rows affected (1.27 sec)

mysql> describe customer;

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

| Field | Type | Null | Key | Default | Extra |

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

| customer\_id | int | NO | PRI | NULL | |

| cust\_name | varchar(20) | YES | | NULL | |

| city | varchar(20) | YES | | NULL | |

| grade | int | YES | | NULL | |

| salesman\_id | int | YES | MUL | NULL | |

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

5 rows in set (0.10 sec)

mysql> create table orders(

-> ord\_no int primary key,

-> purchase\_amt real,

-> ord\_date date,

-> customer\_id int,

-> salesman\_id int,

-> foreign key(customer\_id) references customer(customer\_id) on update cascade on delete cascade,

-> foreign key(salesman\_id) references salesman(salesman\_id) on update cascade on delete cascade);

Query OK, 0 rows affected (2.14 sec)

mysql> describe orders;

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

| Field | Type | Null | Key | Default | Extra |

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

| ord\_no | int | NO | PRI | NULL | |

| purchase\_amt | double | YES | | NULL | |

| ord\_date | date | YES | | NULL | |

| customer\_id | int | YES | MUL | NULL | |

| salesman\_id | int | YES | MUL | NULL | |

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

5 rows in set (0.02 sec)

mysql> insert into salesman values

-> (1000,'john','bangalore','25%'),

-> (2000,'ravi','bangalore','20%'),

-> (3000,'kumar','mysore','15%'),

-> (4000,'smith','delhi','30%'),

-> (5000,'harsha','hyderabad','15%');

Query OK, 5 rows affected (0.29 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> 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 | hyderabad | 15% |

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

5 rows in set (0.29 sec)

mysql> insert into customer values

-> (10,'preethi','bangalore',100,1000),

-> (11,'vivek','mangalore',300,1000),

-> (12,'chethan','chennai',400,2000),

-> (13,'bhaskar','bangalore',200,2000),

-> (14,'mamatha','bangalore',400,3000);

Query OK, 5 rows affected (0.33 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> select \* from customer;

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

| customer\_id | cust\_name | city | grade | salesman\_id |

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

| 10 | preethi | bangalore | 100 | 1000 |

| 11 | vivek | mangalore | 300 | 1000 |

| 12 | chethan | chennai | 400 | 2000 |

| 13 | bhaskar | bangalore | 200 | 2000 |

| 14 | mamatha | bangalore | 400 | 3000 |

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

5 rows in set (0.00 sec)

mysql> insert into orders values

-> (50,5000,'2017-05-05',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);

Query OK, 5 rows affected (0.18 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> select \* from orders;

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

| ord\_no | purchase\_amt | ord\_date | customer\_id | salesman\_id |

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

| 50 | 5000 | 2017-05-05 | 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 |

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

5 rows in set (0.03 sec)

Write SQL queries to

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

mysql> select grade ,count(customer\_id) from customer group by grade having grade >(select avg(grade) from customer where city='bangalore');

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

| grade | count(customer\_id) |

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

| 300 | 1 |

| 400 | 2 |

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

2 rows in set (0.05 sec)

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

mysql> select salesman\_id,name from salesman s where 1<(select count(\*) from customer c where s.salesman\_id=c.salesman\_id);

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

| salesman\_id | name |

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

| 1000 | john |

| 2000 | ravi |

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

2 rows in set (0.05 sec)

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

-> union

-> select s.salesman\_id,s.name,'no match',s.commission from salesman s where city not in(select city from customer) order by 1 asc;

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

| salesman\_id | name | cust\_name | commission |

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

| 1000 | john | preethi | 25% |

| 1000 | john | bhaskar | 25% |

| 1000 | john | mamatha | 25% |

| 2000 | ravi | preethi | 20% |

| 2000 | ravi | bhaskar | 20% |

| 2000 | ravi | mamatha | 20% |

| 3000 | kumar | no match | 15% |

| 4000 | smith | no match | 30% |

| 5000 | harsha | no match | 15% |

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

9 rows in set (0.06 sec)

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

mysql> create view v\_salesman as

-> select o.ord\_date,s.salesman\_id,s.name

-> from salesman s,orders o

-> where s.salesman\_id=o.salesman\_id

-> and o.purchase\_amt=(select max(purchase\_amt) from orders c

-> where o.ord\_date=c.ord\_date);

Query OK, 0 rows affected (0.26 sec)

mysql> select \* from v\_salesman;

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

| ord\_date | salesman\_id | name |

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

| 2017-05-05 | 1000 | john |

| 2017-01-20 | 2000 | ravi |

| 2017-02-24 | 2000 | ravi |

| 2017-04-13 | 3000 | kumar |

| 2017-03-09 | 2000 | ravi |

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

5 rows in set (0.05 sec)

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

mysql> delete from salesman where salesman\_id=1000;