Name:Neelam Godihal
USN:1BM19CS220

Program 6 : Order Database

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.

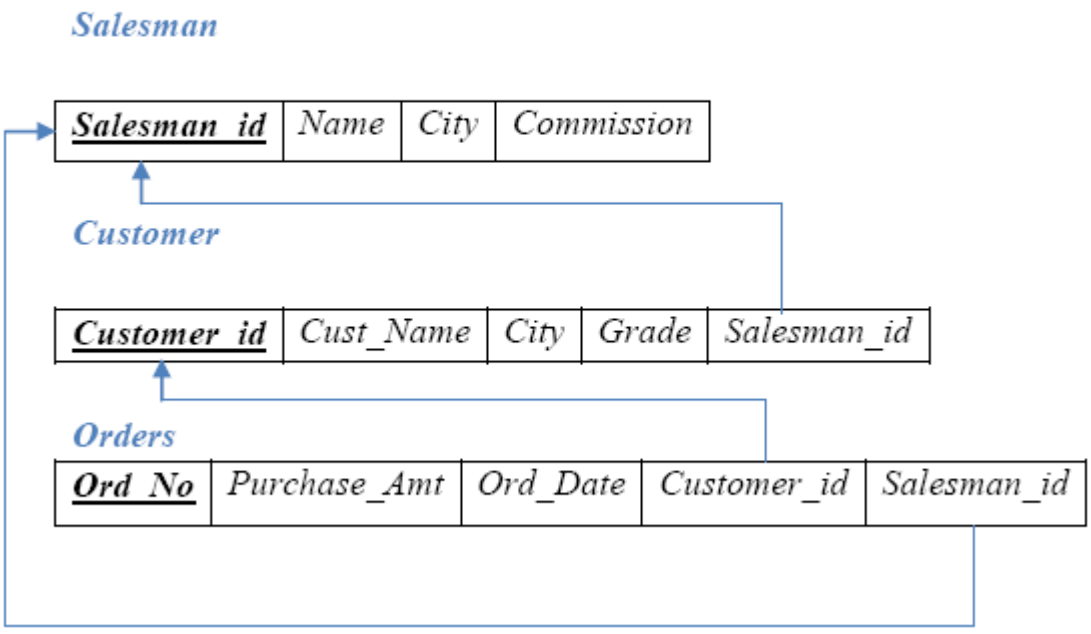
2. Find the name and numbers of all salesmen who had more than one customer.

3. List all salesmen 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.

Schema Diagram



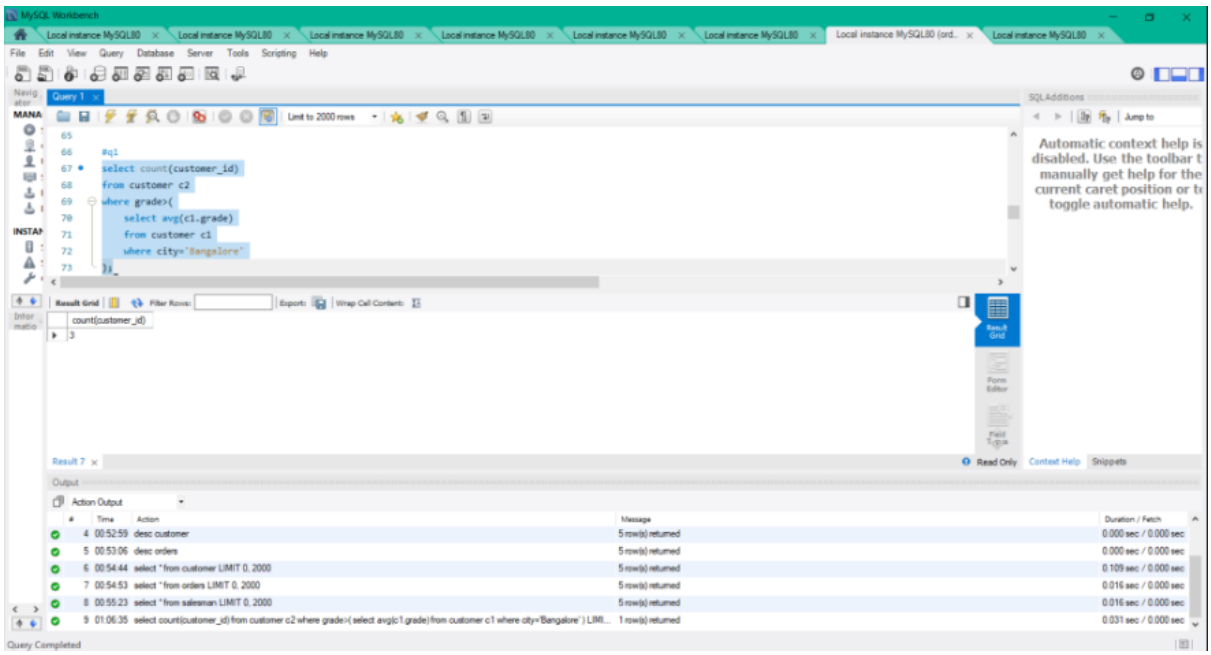
Creation of tables and entering tuple values:

```
show databases;

create database order1;

use order1;

create table salesman(
        salesman_id int primary key,
   name varchar(30),
   city varchar(30),
   commission varchar(10)
);

desc salesman;

create table customer(
        customer_id int primary key,
   cust_name varchar(30),
   city varchar(30),
   grade int(3),
   salesman_id int,
  foreign key(salesman_id) references salesman(salesman_id) on delete set NULL      #gonna
delete later thats why
);

desc customer;

create table orders(
        ord_no int(5) primary key,
   purchase_amt real,
   ord_date date,
   customer_id int,
   salesman_id int,
   foreign key(customer_id) references customer(customer_id),
   foreign key(salesman_id) references salesman(salesman_id)
);

desc orders;

insert into salesman
values  (1000,'John','Bangalore','25%'),
                (2000,'Ravi','Bangalore','20%'),
      (3000,'Kumar','Mysore','15%'),
      (4000,'Smith','Delhi','30%'),
      (5000,'Harsha','Hyderabad','15%');

select * from salesman;

insert into customer
values  (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);

select * from customer;

insert into orders
values  (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);

select * from orders;
```

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

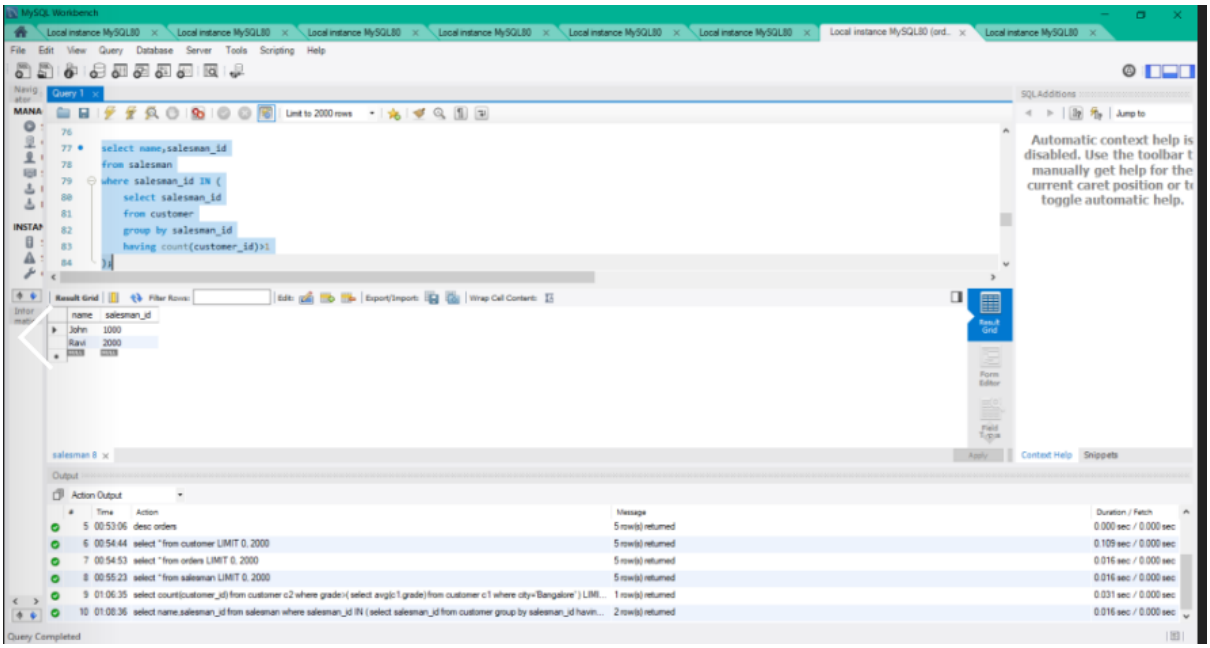
```
select count(customer_id)
from customer c2
where grade>(
        select avg(c1.grade)
        from customer c1
        where city='Bangalore'
);
```



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

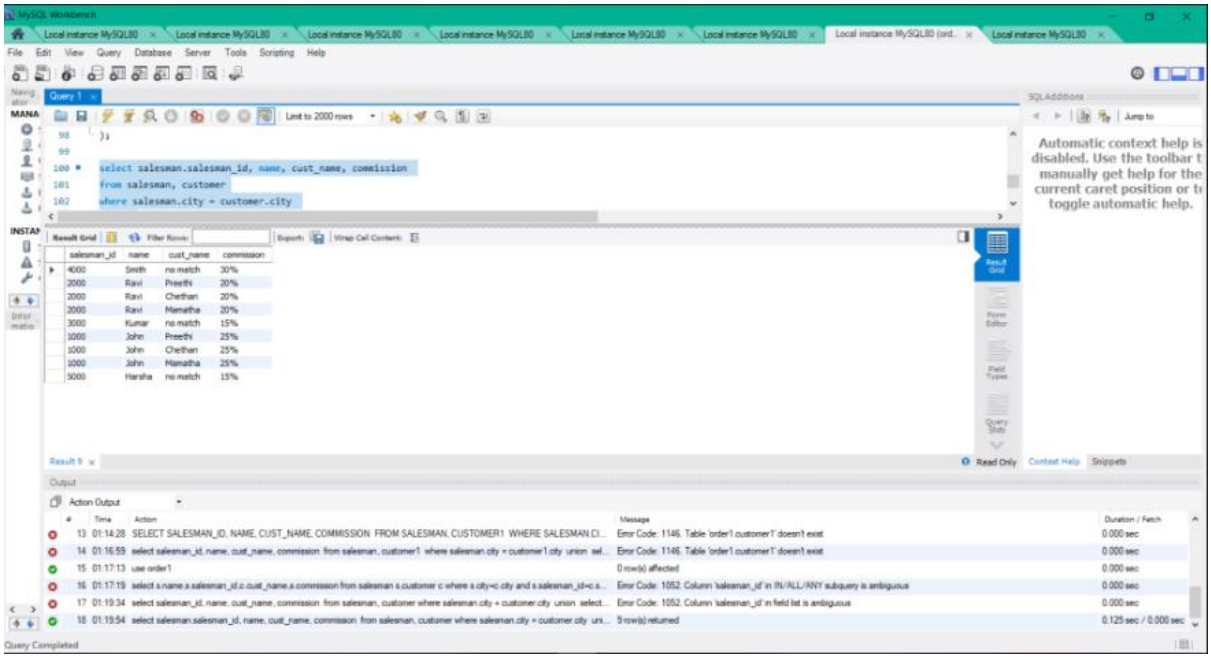
```
select name,salesman_id
from salesman
where salesman_id IN (
        select salesman_id
```

```
        from customer
        group by salesman_id
        having count(customer_id)>1
);
```



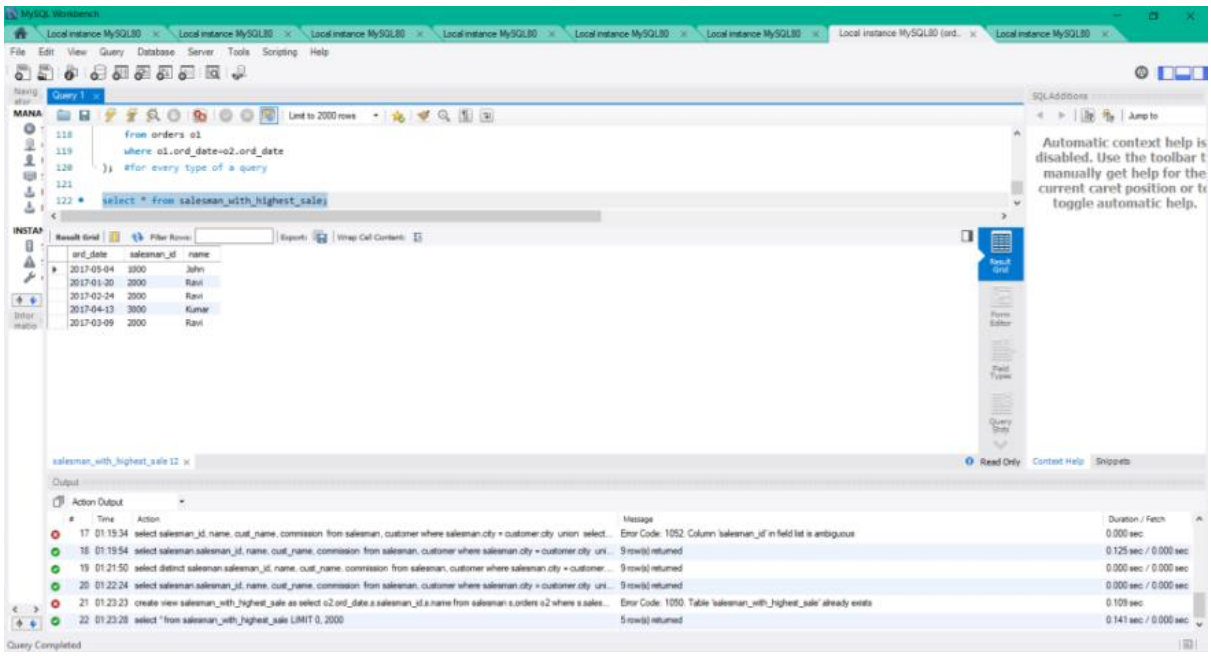
3. **List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)**

```
select salesman.salesman_id, name, cust_name, commission
from salesman, customer
where salesman.city = customer.city
union
select salesman_id, name, 'no match', commission
from salesman
where not city = any
(select city
from customer)
order by 2 desc;
```



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

```
create view salesman_with_highest_sale as
select o2.ord_date,s.salesman_id,s.name
from salesman s,orders o2
where s.salesman_id=o2.salesman_id and o2.purchase_amt=(
        select max(purchase_amt)
    from orders o1
    where o1.ord_date=o2.ord_date
);        #for every type of a query
```



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;

select * from Salesman;
select * from Orders
```

order_id	purchase_amt	order_date	customer_id	salesman_id
50	5000	2017-05-04	10	NULL
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
NULL	NULL	NULL	NULL	NULL