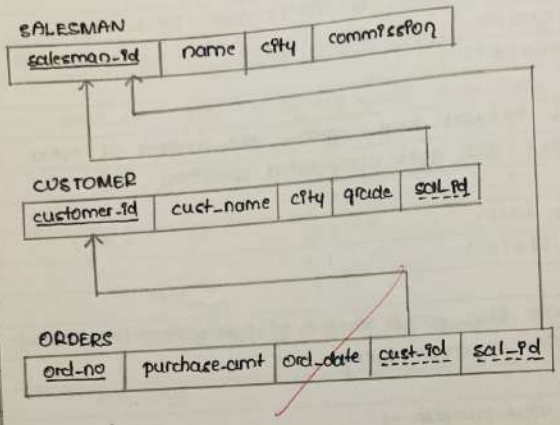
Schema Diagram:



Date .....

Expt. No. 2 Page No. 2

LAB PROGRAM - 02

Consider the following schema for ordered database

SALESMAN (salesman_id, name, city, commission)

CUSTOMER (customer_id, cust_name, city, grade, sal_id)

ORDERS (ord_no, purchase_amt, ord_date, cust_id, sal_id)

Write SQL queries to

1. Count the customers with grades above 'Banglore' 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 highest order of a day
5. Demonstrate the 'delete' operation by removing salesman with id 1000. All his orders must also be deleted.

Table created

Table created

Table created

Date .....

Expt. No. 2 Page No. 9

Table Creations:

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

create table customer(
customer_id int primary key,
cust_name varchar(20),
city varchar(20),
grade int,
sal_id int,
foreign key(sal_id) references salesman(salesman_id) on delete cascade);

create table orders(
ord_no int primary key,
purchase_amt int,
ord_date date,
cust_id int,
sal_id int,
foreign key(cust_id) references customer(customer_id) on delete cascade);
foreign key(sal_id) references salesman(salesman_id) on delete cascade);
```

select * from salesman;

SALESMAN_ID	NAME	CITY	COMMISSION
1000	John	banglore	25
2000	ravi	banglore	20
3000	kumar	mysore	15
4000	sumith	delhi	30
5000	harsha	banglore	15

select * from customer;

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SAL_ID
10	preethi	banglore	100	1000
11	vivek	manglore	300	1000
12	bhasker	chennai	400	2000
13	chethan	banglore	200	2000
14	Mamatha	banglore	400	3000

Date .....

Expt. No. 2 Page No. 10

Values Insertion:

1. insert into salesman values
('1000', 'John', 'Banglore', '25');
insert into salesman values
('2000', 'ravi', 'Banglore', '20');
insert into salesman values
('3000', 'kumar', 'Mysore', '15');
insert into salesman values
('4000', 'sumith', 'delhi', '30');
insert into salesman values
('5000', 'harsha', 'banglore', '15');

2. insert into customer values
('10', 'Preethi', 'banglore', '100', '1000');
insert into customer values
('11', 'vivek', 'manglore', '300', '1000');
insert into customer values
('12', 'bhaskar', 'chennai', '400', '2000');
insert into customer values
('13', 'chethan', 'banglore', '200', '2000');
insert into customer values
('14', 'mamatha', 'banglore', '400', '3000');

select * from orders;

ORD_NO	PURCHASE_AMT	ORD_DATE	CUST_ID	SAL_ID
50	5000	04-May-19	10	1000
51	450	20-Jan-19	10	1000
52	1000	24-feb-19	13	2000
53	3500	13-Apr-19	14	3000
54	350	09-mar-19	12	2000

Output:

GRADE	COUNT(CUSTOMER_ID)
400	2
200	1

Date .....

Expt. No. 2 Page No. 11

3. insert into orders values
('50', '5000', '04-may-19', '10', '1000');
insert into orders values
('51', '450', '20-Jan-19', '10', '1000');
insert into orders values
('52', '1000', '24-feb-19', '13', '2000');
insert into orders values
('53', '3500', '13-apr-19', '14', '3000');
insert into orders values
('54', '350', '09-mar-19', '12', '2000');

Queries:

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

```
select grade, count(customer_id)
from customer
group by grade
having grade >
        (select avg(grade)
          from customers
          where city = 'banglore');
```

NAME	SALESMAN_ID
John	1000
Ravi	2000

SALESMAN_ID	NAME	CUST_NAME	CITY
1000	john	chethan	banglore
1000	john	mamatha	banglore
1000	john	no-match	manglore
1000	john	preethi	banglore
2000	ravi	chethan	banglore
2000	ravi	mamatha	banglore
2000	ravi	no-match	~~no-match~~ chennai
2000	ravi	preethi	~~preethi~~ banglore
3000	kumar	no-match	banglore

Date .....

Expt. No. 7 Page No. 12

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

```
select name, salesman_id
from salesman
where salesman_id IN
        (select sal_id
         from customer
         group by sal_id
         having count(*) > 1);
```

3. List all the salesman and indicate those who have and don't have customers in their cities (use union operation).

```
select salesman_id, s.name, c.cust_name, c.city
from salesman s, customer c
where c.city = s.city
UNION
select s.salesman, s.name, 'no-match', c.city
from salesman. s, customer c
where c.city <> s.city and s.salesman_id = c.sal_id;
```

SALESMAN_ID	NAME	ORD_NO
1000	john	50
1000	John	51
2000	ravi	52
3000	kumar	53
4000	ravi	54

```
select *
from salesman
```

SALESMAN_ID	NAME	CITY	COMMISSION
2000	ravi	banglore	20
3000	kumar	Mysore	15
4000	sumith	delhi	80
5000	harsha	pune	15

Expt. No. 2 Date

Page No. 13

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

```
create view view_orders as
select s.salesman_id, s.name, o1.ord_no
from salesman s, orders o1
where s.salesman_id = o1.sal_id and o1.purchase_amt =
    ( select max(purchase_amt)
      from orders o2
      where o2.ord_date = o1.ord.date);
```

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;
```

PA 10/10