create table salesman(salesman\_id varchar(8),

name varchar(20),

city varchar(20),

commission varchar2(10),

constraint pks primary key(salesman\_id));

create table customer(customer\_id varchar(8),

cust\_name varchar2 (20),

city varchar2 (20),

grade number (3),

salesman\_id varchar(8),

constraint pkc primary key(customer\_id),

constraint fkc foreign key(salesman\_id) references salesman(salesman\_id) on delete set null);

create table orders (ord\_no varchar(8),

purchase\_amt number(10, 2),

ord\_date date,

\

customer\_id varchar(8),

salesman\_id varchar(8),

constraint pko primary key (ord\_no),

constraint fkoc foreign key (customer\_id) references customer (customer\_id) on delete cascade,

constraint fkos foreign key (salesman\_id) references salesman (salesman\_id) on delete cascade);

**queries**

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

select grade, count (distinct customer\_id) as no\_of\_customer

from customer

group by grade

having grade > (select avg(grade)

from customer

where city='bangalore');

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

select salesman\_id, name

from salesman s

where ((select count (\*)

from customer

where salesman\_id=s.salesman\_id)>1);

5.Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted. Use ON DELETE CASCADE at the end of foreign key definitions while creating child tableorders and then execute the following: Use ON DELETE SET NULL at the end of foreign key definitions while creating child tablecustomers and then execute

delete from salesman

where salesman\_id=1000;

select \* from salesman;

select \* from customer;

select \* from orders;

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

select distinct s.salesman\_id,s.ord\_date from orders s

where (select sum(purchase\_amt) from orders where

salesman\_id=s.salesman\_id and ord\_date=s.ord\_date and s.customer\_id=customer\_id)

=(select max(sum(purchase\_amt))

from orders s1 where s1.ord\_date=s.ord\_date group by s1.ord\_date,s1.salesman\_id,s1.customer\_id);

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

select s.salesman\_id, s.city

from salesman s

where exists (select city from customer where s.city=city and s.salesman\_id=salesman\_id)

union

select salesman\_id,'no match of cities'

from salesman s

where not exists (select city from customer where s.city=city and s.salesman\_id=salesman\_id);