

```
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 table orders and then execute the following: Use ON DELETE SET NULL at the end of foreign key definitions while creating child table customers 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);
```