5. Consider the schema for **Company Database**:

employee(ssn, name, address, sex, salary, superssn, dno)

department(dno, dname, mgrssn, mgrstartdate)

dlocation(dno,dloc)

project(pno, pname, plocation, dno)

works_on(ssn, pno, hours)

Write SQL queries to

1. Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.

2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.

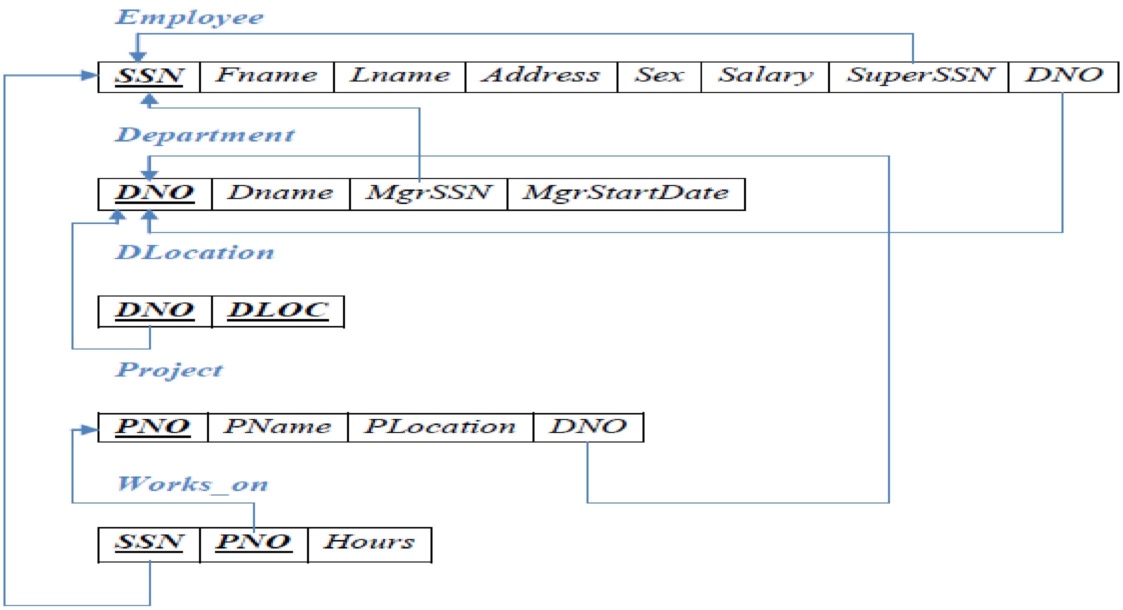
3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department

4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).

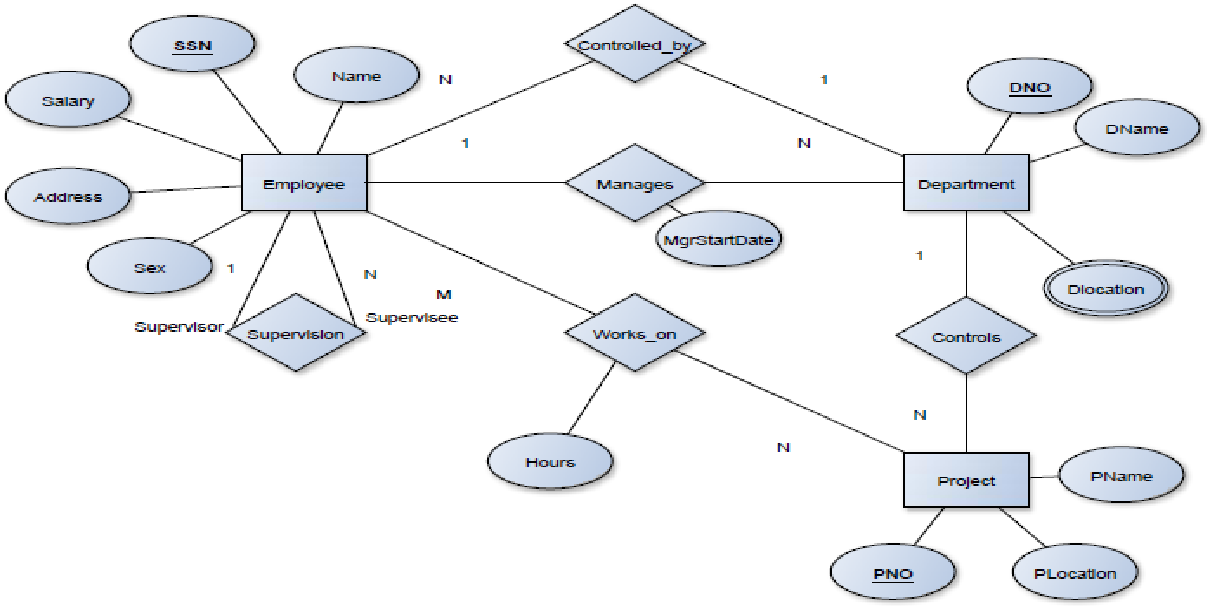
5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

Solution:

Schema Diagram:



Entity – Relationship Diagram



Creating Tables (Relations):

- create table **department** (dno varchar (20) primary key, dname varchar (20), mgrstartdate date);
- create table **employee** (ssn varchar (20) primary key, fname varchar (20), lname varchar (20), address varchar (20), sex char (1), salary integer, superssn references employee (ssn), dno references department (dno));

NOTE: Once DEPARTMENT and EMPLOYEE tables are created, we must alter department table to add foreign constraint MGRSSN using SQL command

- alter table **department** add mgrssn varchar (20);
- alter table **department** add foreign key (mgrssn) references employee (ssn);

- create table **dlocation** (dloc varchar (20), dno references department (dno), primary key (dno, dloc));
- create table **project** (pno integer primary key, pname varchar (20), plocation varchar (20), dno references department (dno));
- create table **works_on** (hours int, ssn references employee (ssn), pno references project(pno), primary key (ssn, pno));

Insertion of Values to Tables

employee

```
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatece01’,’john’,’scott’,’bangalore’,’m’, 450000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatise01’,’james’,’smith’,’bangalore’,’m’, 500000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatise02’,’hearn’,’baker’,’bangalore’,’m’, 700000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatise03’,’edward’,’scott’,’mysore’,’m’, 500000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatise04’,’pavan’,’hegde’,’mangalore’,’m’, 650000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatise05’,’girish’,’malya’,’mysore’,’m’, 450000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatise06’,’neha’,’sn’,’bangalore’,’f’, 800000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatacc01’,’ahana’,’k’,’mangalore’,’f’, 350000);
insert into employee (ssn, fname, lname, address, sex, salary) values (’gatacc02’,’santhosh’,’kumar’,’mangalore’,’m’, 300000);
```

department

```
insert into department values (’1’,’accounts’,’01-jan-01’,’ gatece01’);
insert into department values (’2’,’it’,’01-aug-16’,’ gatise01’);
insert into department values (’3’,’ece’,’01-jun-08’,’ gatece01’);
insert into department values (’4’,’ise’,’01-aug-15’,’ gatise01’);
insert into department values (’5’,’ise’,’01-jun-02’,’ gatise05’);
```

Note: update entries of employee table to fill missing fields superssn and dno

```
        update employee set  superssn=null, dno=’3’  where ssn=’gatece01’;
        update employee set  superssn=’ gatise02’, dno=’5’  where ssn=’gatcse01’;
        update employee set superssn=’ gatise03’, dno=’5’  where ssn=’gatise02’;
        update employee set  superssn=’ gatise04’, dno=’5’  where ssn=’gatise03’;
        update employee set  dno=’5’, superssn=’ gatise05’  where ssn=’gatise04’;
        update employee set  dno=’1’, superssn=’ gatise06’  where ssn=’gatise05’;
```

dlocation

```
        insert into dlocation values (’bangalore’, ’1’);
        insert into dlocation values (’bangalore’, ’2’);
        insert into dlocation values (’bangalore’,’3’);
        insert into dlocation values (’mangalore’, ’4’);
        insert into dlocation values (’mangalore’, ’5’);
```

project

```
        insert into project values (100,’iot’,’bangalore’,’5’);
        insert into project values (101,’cloud’,’bangalore’,’5’);
        insert into project values (102,’bigdata’,’bangalore’,’5’);
        insert into project values (103,’sensors’,’bangalore’,’3’);
        insert into project values (104,’bank management’,’bangalore’,’1’);
        insert into project values (105,’account’,’bangalore’,’1’);
```

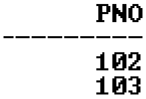
work_on

```
        insert into works_on values (4,’gatise01’, 100);
        insert into works_on values (6,’gatise01’, 101);
        insert into works_on values (8,’gatise01’, 102);
        insert into works_on values (10,’gatise02’, 100);
        insert into works_on values (3,’gatise04’, 100);
        insert into works_on values (4,’gatise05’, 101);
        insert into works_on values (6,’gatise03’, 102);
        insert into works_on values (7,’gatece01’, 103);
```

Queries:

1. Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.

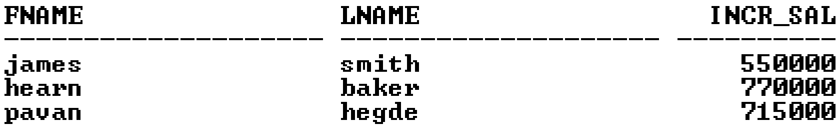
```
(select distinct p.pno
from project p, department d, employee e
where e.dno=p.dno and d.mgrssn=e.ssn and e.lname=’scott’)
union
(select distinct p1.pno
from project p1, works_on w, employee e1
where p1.pno=w.pno and e1.ssn=w.ssn and e1.lname=’scott’);
```



```
      PNO
---------
      102
      103
```

2. Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.

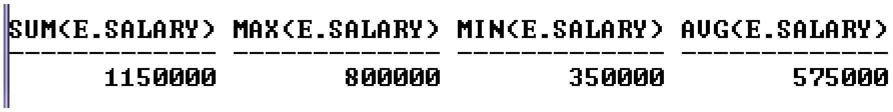
```
select e.fname, e.lname, 1.1*e.salary as incr_sal
from employee e, works_on w, project p
where e.ssn=w.ssn and w.pno=p.pno and p.pname=’iot’;
```



```
FNAME                LNAME                  INCR_SAL
-------------------- -------------------- ----------
james                smith                    550000
hearn                baker                    770000
pavan                hegde                    715000
```

3. Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department

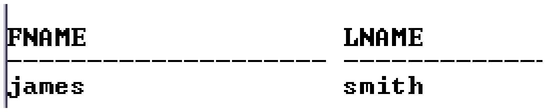
```
select sum (e.salary), max (e.salary), min (e.salary), avg (e.salary)
from employee e, department d
where e.dno=d.dno and d.dname=’accounts’;
```



```
SUM(E.SALARY) MAX(E.SALARY) MIN(E.SALARY) AVG(E.SALARY)
------------- ------------- ------------- -------------
      1150000        800000        350000        575000
```

4. Retrieve the name of each employee who works on all the projects Controlled by department number 5 (use NOT EXISTS operator).

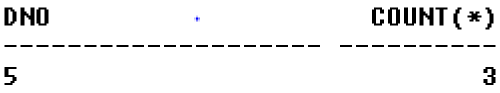
```
select e.fname, e.lname
from employee e
where not exists((select pno from project
where dno='5')
minus
(select pno
from works_on
where e.ssn=ssn));
```



```
FNAME                LNAME
-------------------- -------------
james                smith
```

5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6, 00,000.

```
select d.dno, count (*)
from department d, employee e
where d.dno=e.dno and e.salary>600000 and d.dno in (select e1.dno
from employee e1
group by e1.dno
having count (*)>5)
group by d.dno;
```



```
DNO                    COUNT(*)
-------------------- ----------
5                             3
```