



PART-A Lab Program 5

Department of
Information Science & Engineering

www.cambridge.edu.in



A. Consider the schema for CompanyDatabase:

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.**

```
CREATE TABLE DEPARTMENT
```

```
(  
  DNO VARCHAR2 (20) PRIMARY KEY,  
  DNAME VARCHAR2 (20),  
  MGRSTARTDATE DATE  
);
```

```
CREATE TABLE EMPLOYEE
```

```
(SSN VARCHAR2 (20) PRIMARY KEY,  
  FNAME VARCHAR2 (20),  
  LNAME VARCHAR2 (20),  
  ADDRESS VARCHAR2 (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 REFERENCES EMPLOYEE (SSN);

```
CREATE TABLE DLOCATION  
(DLOC VARCHAR2 (20),  
DNO REFERENCES DEPARTMENT (DNO),  
PRIMARY KEY (DNO, DLOC)  
);
```

```
CREATE TABLE PROJECT  
(  
PNO INTEGER PRIMARY KEY,  
PNAME VARCHAR2(20),  
PLOCATION VARCHAR2 (20),  
DNO REFERENCES DEPARTMENT (DNO)  
);
```

```
CREATE TABLE WORKS_ON  
(  
HOURS NUMBER (2),  
SSN REFERENCES EMPLOYEE (SSN),  
PNO REFERENCES PROJECT(PNO),  
PRIMARY KEY (SSN, PNO)  
);
```

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSECE01','JOHN','SCOTT','BANGALORE','M', 450000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSCSE01','JAMES','SMITH','BANGALORE','M', 500000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSCSE02','HEARN','BAKER','BANGALORE','M', 700000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSCSE03','EDWARD','SCOTT','MYSORE','M', 500000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSCSE04','PAVAN','HEGDE','MANGALORE','M', 650000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSCSE05','GIRISH','MALYA','MYSORE','M', 450000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSCSE06','NEHA','SN','BANGALORE','F', 800000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSACC01','AHANA','K','MANGALORE','F', 350000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSACC02','SANTHOSH','KUMAR','MANGALORE','M', 300000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSISE01','VEENA','M','MYSORE','M', 600000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES ('RNSIT01','NAGESH','HR','BANGALORE','M', 500000);

```
INSERT INTO DEPARTMENT VALUES ('1','ACCOUNTS','01-JAN-01','RNSACC02');  
INSERT INTO DEPARTMENT VALUES ('2','IT','01-AUG-16','RNSIT01');  
INSERT INTO DEPARTMENT VALUES ('3','ECE','01-JUN-08','RNSECE01');  
INSERT INTO DEPARTMENT VALUES ('4','ISE','01-AUG-15','RNSISE01');  
INSERT INTO DEPARTMENT VALUES ('5','CSE','01-JUN-02','RNSCSE05');
```

Note: update entries of employee table to fill missing fields SUPERSSN and DNO

```
UPDATE EMPLOYEE SET SUPERSSN=NULL, DNO='3' WHERE SSN='RNSECE01';
```

```
UPDATE EMPLOYEE SET SUPERSSN='RNSCSE02', DNO='5' WHERE SSN='RNSCSE01';
```

```
UPDATE EMPLOYEE SET SUPERSSN='RNSCSE03', DNO='5' WHERE SSN='RNSCSE02';
```

```
UPDATE EMPLOYEE SET SUPERSSN='RNSCSE04', DNO='5' WHERE SSN='RNSCSE03';
```

UPDATE EMPLOYEE SET DNO='5', SUPERSSN='RNSCSE05' WHERE SSN='RNSCSE04';

UPDATE EMPLOYEE SET DNO='5', SUPERSSN='RNSCSE06' WHERE SSN='RNSCSE05';

UPDATE EMPLOYEE SET DNO='5', SUPERSSN=NULL WHERE SSN='RNSCSE06';

UPDATE EMPLOYEE SET DNO='1', SUPERSSN='RNSACC02' WHERE SSN='RNSACC01';

UPDATE EMPLOYEE SET DNO='1', SUPERSSN=NULL WHERE SSN='RNSACC02';

UPDATE EMPLOYEE SET DNO='4', SUPERSSN=NULL WHERE SSN='RNSISE01';

UPDATE EMPLOYEE SET DNO='2', SUPERSSN=NULL WHERE SSN='RNSIT01';

```
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');
```

```
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,'SALARY MANAGEMENT','BANGALORE','1');  
INSERT INTO PROJECT VALUES (106,'OPENSTACK','BANGALORE','4');  
INSERT INTO PROJECT VALUES (107,'SMART CITY','BANGALORE','2');
```



```
INSERT INTO WORKS_ON VALUES (4, 'RNSCSE01', 100);  
INSERT INTO WORKS_ON VALUES (6, 'RNSCSE01', 101);  
INSERT INTO WORKS_ON VALUES (8, 'RNSCSE01', 102);  
INSERT INTO WORKS_ON VALUES (10, 'RNSCSE02', 100);  
INSERT INTO WORKS_ON VALUES (3, 'RNSCSE04', 100);  
INSERT INTO WORKS_ON VALUES (4, 'RNSCSE05', 101);  
INSERT INTO WORKS_ON VALUES (5, 'RNSCSE06', 102);  
INSERT INTO WORKS_ON VALUES (6, 'RNSCSE03', 102);  
INSERT INTO WORKS_ON VALUES (7, 'RNSECE01', 103);  
INSERT INTO WORKS_ON VALUES (5, 'RNSACC01', 104);  
INSERT INTO WORKS_ON VALUES (6, 'RNSACC02', 105);  
INSERT INTO WORKS_ON VALUES (4, 'RNSISE01', 106);  
INSERT INTO WORKS_ON VALUES (10, 'RNSIT01', 107);
```

Queries:

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 control the project.

```
(SELECT DISTINCT P.PNO
FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
WHERE E.DNO=D.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

100

101

102

103

104

105

106

107

2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 per cent 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) |
|---------------|---------------|---------------|---------------|
| 650000 | 350000 | 300000 | 325000 |

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 |