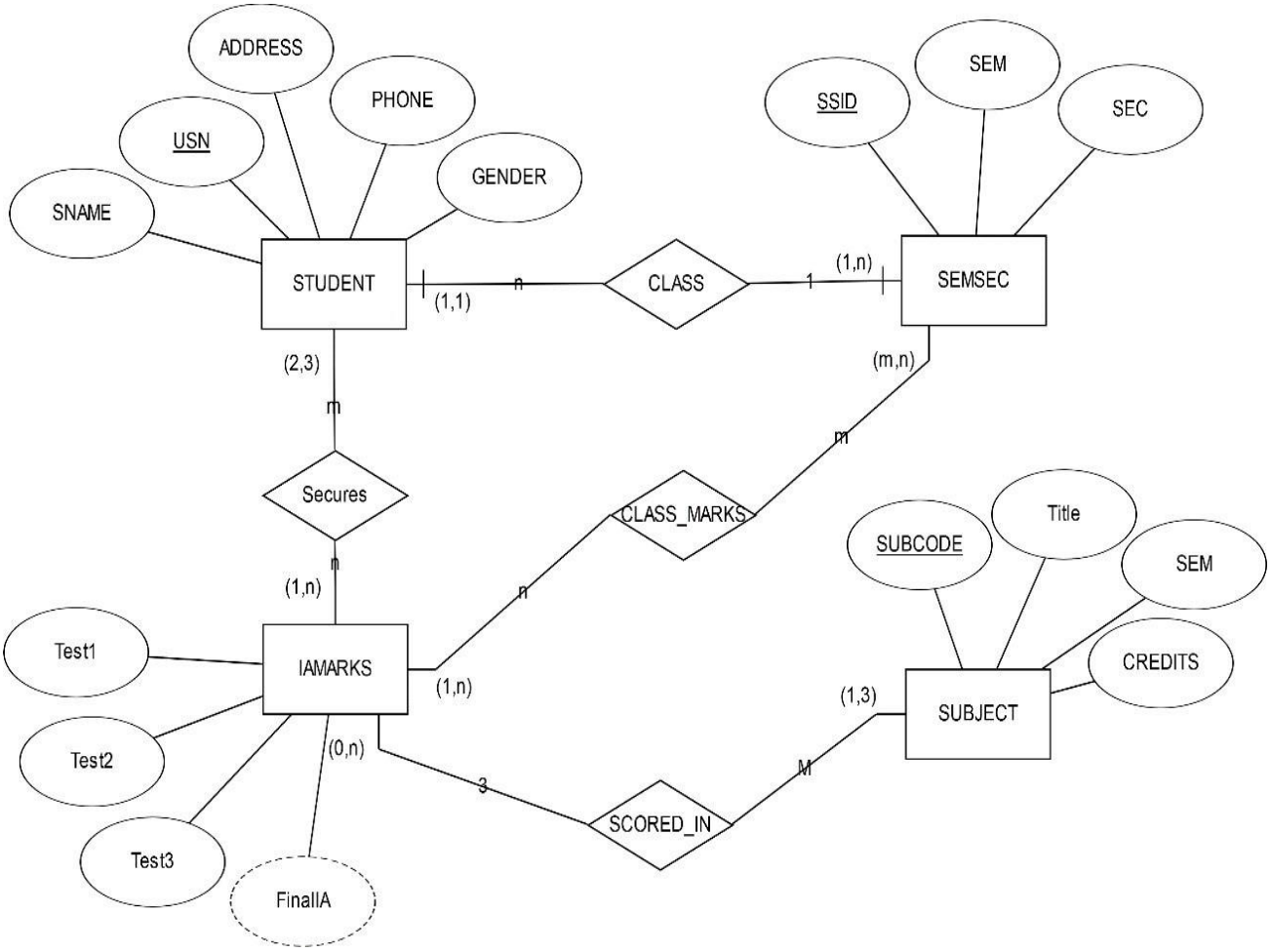
1.Consider the schema for CollegeDatabase:
STUDENT (USN, SName, Address, Phone, Gender)
SEMSEC (SSID, Sem, Sec)
CLASS (USN, SSID)
SUBJECT (Subcode, Title, Sem, Credits)
IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)
Write SQL queries to
1. List all the student details studying in fourth semester 'C'section.
2. Compute the total number of male and female students in each semester and in each section.
3. Create a view of Test1 marks of student USN '1DC20AI001' in all subjects.
4. Calculate the Final IA (average of best two test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:
If Final IA = 17 to 20 then CAT ='Outstanding'
If Final IA = 12 to 16 then CAT = 'Average'
If Final IA< 12 then CAT = 'Weak'

ER DIAGRAM



Schema Diagram

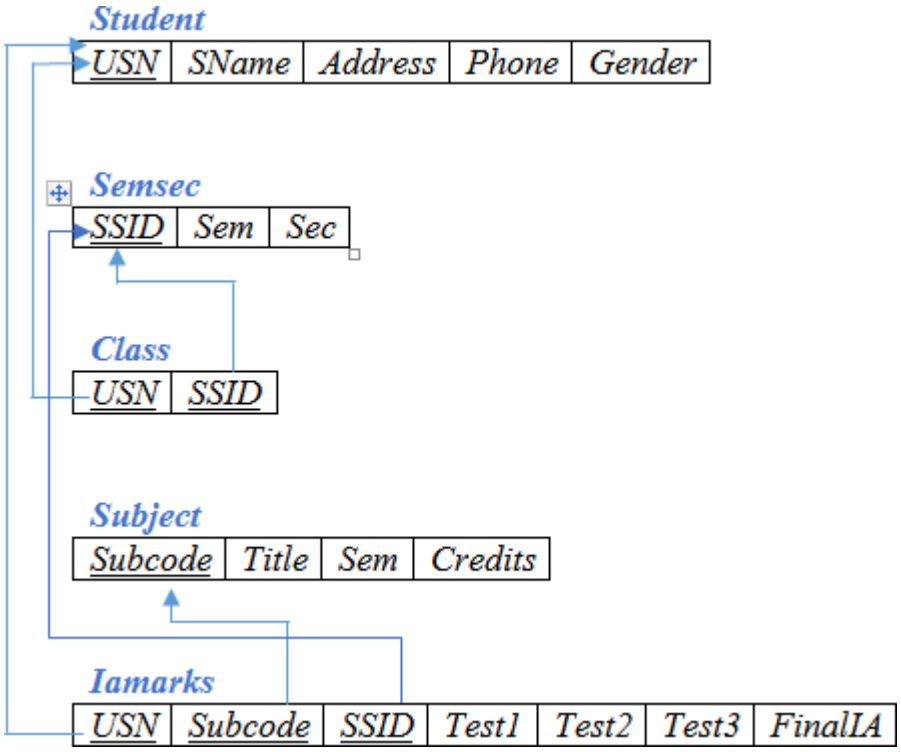


Table Creation

```
CREATE TABLE STUDENT (
 USN VARCHAR (10) PRIMARY
 KEY, SNAME VARCHAR (25),
 ADDRESS VARCHAR (25),
 PHONE NUMBER (10),
 GENDER CHAR (1));

CREATE TABLE SEMSEC (
SSID VARCHAR (5) PRIMARY
KEY, SEM NUMBER (2),
SEC CHAR (1));

CREATE TABLE
```

```
CLASS ( USN
VARCHAR (10),
SSID VARCHAR (5),
PRIMARY KEY (USN,
SSID),
FOREIGN KEY (USN) REFERENCES STUDENT
(USN), FOREIGN KEY (SSID) REFERENCES
SEMSEC (SSID));
CREATE TABLE SUBJECT ( SUBCODE
VARCHAR (8),
TITLE VARCHAR (20),
SEM NUMBER (2),
CREDITS NUMBER (2), PRIMARY KEY
(SUBCODE));

 CREATE TABLE IAMARKS ( USN VARCHAR
 (10),
 SUBCODE VARCHAR (8), SSID VARCHAR(5), TEST1
 NUMBER(2), TEST2 NUMBER(2), TEST3 NUMBER(2), FINALIA
 NUMBER (2),
 PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
 FOREIGN KEY (SUBCODE) REFERENCES SUBJECT
 (SUBCODE), FOREIGN KEY (SSID) REFERENCES SEMSEC
 (SSID));
```

Insertion of values totables

```
INSERT INTO STUDENT VALUES
('1RN13CS020','AKSHAY','BELAGAVI', 8877881122,'M');
INSERT INTO STUDENT
VALUES('1RN13CS062','SANDHYA','BENGALURU', 7722829912,'F');
INSERT INTO STUDENT
VALUES('1RN13CS091','TEESHA','BENGALURU', 7712312312,'F');
INSERT INTO STUDENT
VALUES('1RN13CS066','SUPRIYA','MANGALURU', 8877881122,'F');
INSERT INTO STUDENTVALUES('1RN14CS010','ABHAY','BENGALURU',
9900211201,'M');
```

```
INSERT INTO STUDENT
VALUES('1RN14CS032','BHASKAR','BENGALURU', 9923211099,'M');
INSERT INTO STUDENTVALUES ('1RN14CS025','ASMI','BENGALURU',
7894737377,'F'); INSERT INTO STUDENT VALUES
('1RN15CS011','AJAY','TUMKUR', 9845091341,'M');


INSERT INTO STUDENT VALUES
('1RN15CS029','CHITRA','DAVANGERE', 7696772121,'F');
INSERT INTO STUDENT VALUES ('1RN15CS045','JEEVA','BELLARY',
9944850121,'M'); INSERT INTO STUDENT VALUES
('1RN15CS091','SANTOSH','MANGALURU', 8812332201,'M');
INSERT INTO STUDENT
VALUES('1RN16CS045','ISMAIL','KALBURGI', 9900232201,'M');
INSERT INTO STUDENT VALUES
('1RN16CS088','SAMEERA','SHIMOGA', 9905542212,'F');
INSERT INTO STUDENT VALUES
('1RN16CS122','VINAYAKA','CHIKAMAGALUR', 8800880011,'M');

INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A');
INSERT INTO SEMSEC VALUES (_CSE8B', 8,'B');
INSERTINTOSEMSECVALUES(_CSE8C‘,8,‘C‘);

INSERT INTO SEMSEC VALUES ('CSE7A', 7,‘A‘);
INSERTINTOSEMSECVALUES(_CSE7B‘,7,'B‘);
INSERT INTO SEMSEC VALUES ('CSE7C',7,'C');

INSERT INTO SEMSEC VALUES (_CSE6A', 6,'A');
INSERT INTO SEMSEC VALUES (_CSE6B‘, 6,‘B‘);
INSERT INTO SEMSEC VALUES ('CSE6C‘, 6,‘C‘);

INSERT INTO SEMSEC VALUES (_CSE5A‘, 5,'A‘);
INSERT INTO SEMSEC VALUES ('CSE5B', 5,'B');
INSERT INTO SEMSEC VALUES (_CSE5C', 5,'C');

INSERTINTOSEMSECVALUES(_CSE4A‘,4,‘A
‘); INSERT INTO SEMSEC VALUES ('CSE4B',
4,‘B‘);
INSERTINTOSEMSECVALUES(_CSE4C‘,4,'C‘
);

INSERT INTO SEMSEC VALUES ('CSE3A', 3,'A');
```

```
INSERT INTO SEMSEC VALUES (_CSE3B', 3,'B');
INSERTINTOSEMSECVALUES(_CSE3C‘,3,‘C‘);

INSERT INTO SEMSEC VALUES ('CSE2A',
2,‘A‘); INSERT INTO SEMSEC VALUES
(_CSE2B‘, 2,'B‘); INSERT INTO SEMSEC
VALUES ('CSE2C', 2,'C'); INSERT INTO
SEMSEC VALUES (_CSE1A', 1,'A');


INSERT INTO SEMSEC VALUES (_CSE1B‘, 1,‘B‘);
INSERT INTO SEMSEC VALUES ('CSE1C', 1,‘C‘);

INSERTINTOCLASSVALUES(_1RN13CS020‘,‘CSE8A
‘);
INSERTINTOCLASSVALUES(_1RN13CS062‘,‘CSE8A
‘);
INSERTINTOCLASSVALUES(_1RN13CS066‘,‘CSE8B
‘);
INSERTINTOCLASSVALUES(_1RN13CS091‘,‘CSE8C
‘);

INSERTINTOCLASSVALUES(_1RN14CS010‘,‘CSE7A
‘);
INSERTINTOCLASSVALUES(_1RN14CS025‘,‘CSE7A
‘);
INSERTINTOCLASSVALUES(_1RN14CS032‘,‘CSE7A
‘);

INSERTINTOCLASSVALUES(_1RN15CS011‘,‘CSE4A
‘);
INSERTINTOCLASSVALUES(_1RN15CS029‘,‘CSE4A
‘);
INSERTINTOCLASSVALUES(_1RN15CS045‘,‘CSE4B
‘);
INSERTINTOCLASSVALUES(_1RN15CS091‘,‘CSE4C
‘);

INSERTINTOCLASSVALUES(_1RN16CS045‘,‘CSE3A
‘);
INSERTINTOCLASSVALUES(_1RN16CS088‘,‘CSE3B
```

```
‘);
INSERTINTOCLASSVALUES(_1RN16CS122‘,‘CSE3C
‘);

INSERT INTO SUBJECT VALUES ('10CS81','ACA', 8,
4); INSERT INTO SUBJECT VALUES ('10CS82','SSM',
8, 4); INSERT INTO SUBJECT VALUES
('10CS83','NM', 8, 4); INSERT INTO SUBJECT
VALUES ('10CS84','CC', 8, 4); INSERT INTO
SUBJECT VALUES ('10CS85','PW', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS71','OOAD', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS72','ECS', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS73','PTW', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS74','DWDM', 7, 4);
INSERT INTO SUBJECT VALUES (_10CS75','JAVA', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS76','SAN', 7, 4);

INSERT INTO SUBJECT VALUES ('15CS51', 'ME', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS52','CN', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS53','DBMS', 5,
4); INSERT INTO SUBJECT VALUES ('15CS54','ATC',
5, 4); INSERT INTO SUBJECT VALUES
('15CS55','JAVA', 5, 3); INSERT INTO SUBJECT
VALUES ('15CS56','AI', 5, 3);



INSERT INTO SUBJECT VALUES ('15CS41','M4', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS42','SE', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS43','DAA', 4,
4); INSERT INTO SUBJECT VALUES
('15CS44','MPMC', 4, 4); INSERT INTO SUBJECT
VALUES ('15CS45','OOC', 4, 3); INSERT INTO
SUBJECT VALUES ('15CS46','DC', 4, 3);

INSERT INTO SUBJECT VALUES ('15CS31','M3', 3,
4); INSERT INTO SUBJECT VALUES ('15CS32','ADE',
3, 4); INSERT INTO SUBJECT VALUES
('15CS33','DSA', 3, 4); INSERT INTO SUBJECT
VALUES ('15CS34','CO', 3, 4); INSERT INTO
```

```
SUBJECT VALUES ('15CS35','USP', 3, 3); INSERT
INTO SUBJECT VALUES ('15CS36','DMS', 3, 3);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS81','CSE8C', 15, 16,18);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS82','CSE8C', 12, 19,14);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS83','CSE8C', 19, 15,20);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS84','CSE8C', 20, 16,19);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS85','CSE8C', 15, 15,12);
```

1. List all the student details studying in fourth semester 'C' section.

```
SELECT S.*, SS.SEM, SS.SEC
 FROM STUDENT S, SEMSEC SS,
 CLASS C WHERE S.USN = C.USN AND
 SS.SSID = C.SSID
 AND SS.SEM = 4 AND
 SS.SEC='C';
```

USN	SNAME	ADDRESS	PHONE	G	SEM	S
1RN15CS091	SANTOSH	MANGALURU	8812332201	M	4	C

2. Compute the total number of male and female students in each semester and in each section.

```
SELECT SS.SEM, SS.SEC, S.GENDER, COUNT (S.GENDER) AS COUNT
FROM STUDENT S, SEMSEC SS, CLASS C
WHERES.USN =
C.USNAND SS.SSID
=C.SSID
GROUP BY SS.SEM, SS.SEC, S.GENDER
ORDER BYSEM;
```

SEM	S	G	COUNT
3	A	M	1
3	B	F	1
3	C	M	1
4	A	F	1
4	A	M	1
4	B	M	1
4	C	M	1
7	A	F	1
7	A	M	2
8	A	F	1
8	A	M	1
8	B	F	1
8	C	F	1

3. Create a view of Test1 marks of student USN '1BI15CS101' in allsubjects.

```
CREATE VIEW
STU_TEST1_MARKS_VIEW AS SELECT
TEST1, SUBCODE
FROM IAMARKS
WHERE USN = '1RN13CS091';
```

TEST1	SUBCODE
15	10CS81
12	10CS82
19	10CS83
20	10CS84
15	10CS85

4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for allstudents.

```
CREATE OR REPLACE PROCEDURE
AVGMARKS IS CURSOR C_IAMARKS IS
SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B,
GREATEST(TEST3,TEST2) ASC
FROM IAMARKS
WHERE FINALIA IS
NULL FOR UPDATE;

C_ANUMBER;
C_BNUMBER;
```

```
  C_CNUMBER;
  C_SMNUMBER;
  C_AVNUMBER;

BEGIN
  OPEN
  C_IAMARKS;
  LOOP
  FETCH C_IAMARKS INTO C_A, C_B,
    C_C; EXIT WHEN
    C_IAMARKS%NOTFOUND;
    --DBMS_OUTPUT.PUT_LINE(C_A || ' ' || C_B || ' ' ||
    C_C); IF (C_A != C_B) THEN
 C_SM:=C_A+C_B;
    ELSE
 C_SM:=C_A+C_C;
    END IF;

    C_AV:=C_SM/2;
    --DBMS_OUTPUT.PUT_LINE('SUM = '||C_SM);
    --DBMS_OUTPUT.PUT_LINE('AVERAGE = '||C_AV);
    UPDATE IAMARKS SET FINALIA=C_AV WHERE CURRENT OF C_IAMARKS;

  END LOOP;
  CLOSE
C_IAMARKS; END;
/
```

Note: Before execution of PL/SQL procedure, IAMARKS table contents are:

SELECT * FROM IAMARKS;

```
SQL> SELECT * FROM IAMARKS;

USN         SUBCODE  SSID       TEST1      TEST2      TEST3    FINALIA
---------- -------- ----- ---------- ---------- ---------- ----------
1RN13CS091 10CS81   CSE8C         15         16         18
1RN13CS091 10CS82   CSE8C         12         19         14
1RN13CS091 10CS83   CSE8C         19         15         20
1RN13CS091 10CS84   CSE8C         20         16         19
1RN13CS091 10CS85   CSE8C         15         15         12
```

Below SQL code is to invoke the PL/SQL stored procedure from the command line:

```
BEGIN
AVGMARKS;
END;
```

```
SQL> select * from IAMARks;

USN        SUBCODE  SSID       TEST1      TEST2      TEST3    FINALIA
---------- -------- ----- ---------- ---------- ---------- ----------
1RN13CS091 10CS81   CSE8C          15         16         18         17
1RN13CS091 10CS82   CSE8C          12         19         14         17
1RN13CS091 10CS83   CSE8C          19         15         20         20
1RN13CS091 10CS84   CSE8C          20         16         19         20
1RN13CS091 10CS85   CSE8C          15         15         12         15
     .
```

5. Categorize students based on the followingcriterion: If FinalIA = 17 to 20 then CAT=‘Outstanding’
   If FinalIA = 12 to 16 then CAT = ‘Average’ If FinalIA< 12 then CAT = ‘Weak’
   Give these details only for 8th semester A, B, and C section students.

```
SELECT
    S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER,
    (CASE
      WHEN IA.FINALIA BETWEEN 17 AND 20 THEN'OUTSTANDING'
      WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
      ELSE 'WEAK'
    END) AS CAT
FROM STUDENT S, SEMSEC SS, IAMARKS IA,
SUBJECT SUB WHERE S.USN = IA.USN AND
SS.SSID = IA.SSID AND
SUB.SUBCODE = IA.SUBCODE
AND SUB.SEM = 8;
```

2.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.

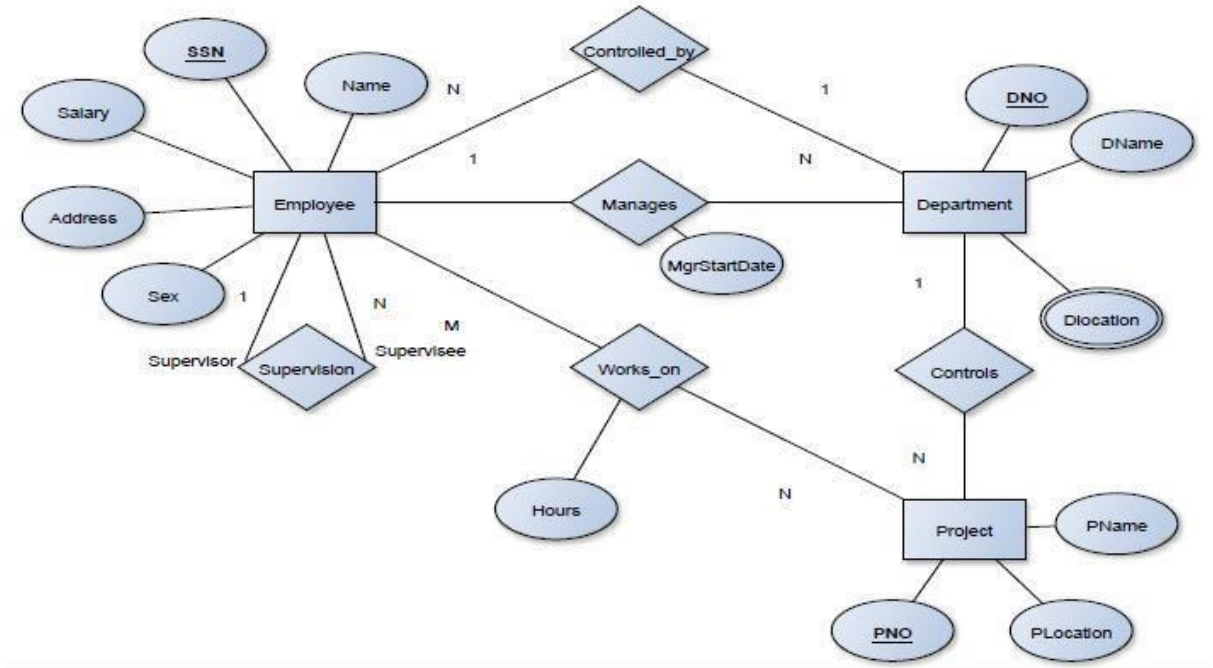
2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise.

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

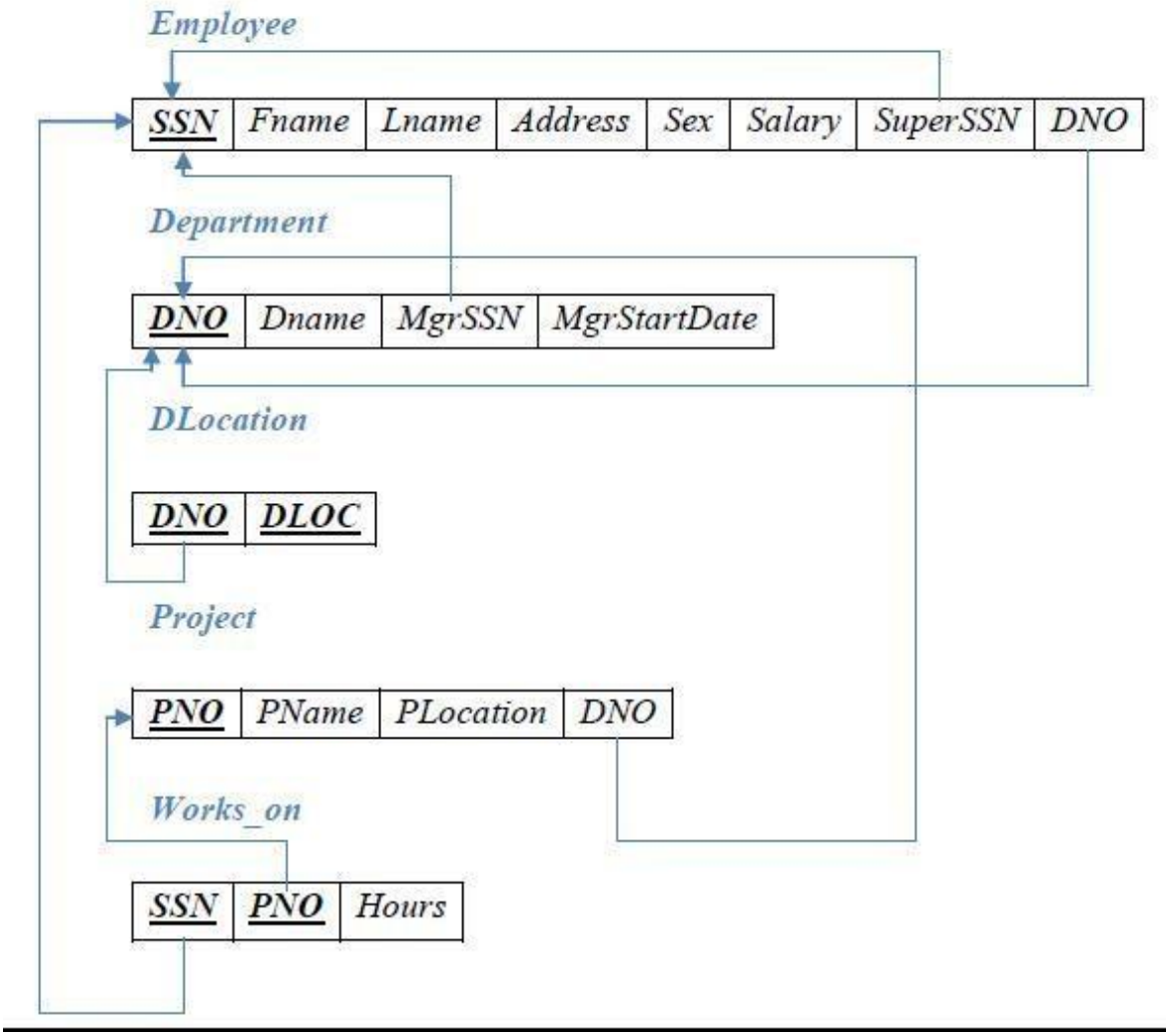
4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator).

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.

ER-Diagram:


SSN
Name
Salary
Address
Sex
Supervisor
Supervision
Supervisee
Employee
Controlled_by
N
1
1
N
Manages
MgrStartDate
Department
DNO
DName
Dlocation
1
Controls
N
Works_on
Hours
N
Project
PName
PNO
PLocation
1
N
M

SCHEMA:


Employee
SSN
Fname
Lname
Address
Sex
Salary
SuperSSN
DNO
Department
DNO
Dname
MgrSSN
MgrStartDate
DLocation
DNO
DLOC
Project
PNO
PName
PLocation
DNO
Works_on
SSN
PNO
Hours

```
CREATE TABLE DEPARTMENT
(DNO
VARCHAR2 (20)
PRIMARY KEY,
DNAME
VARCHAR2 (20),
MGRSTARTDAT
E DATE);

CREATE TABLE EMPLOYEE
(SSN VARCHAR2 (20) PRIMARYKEY, FNAME
VARCHAR2(20),LNME VARCHAR2(20), ADDRESS
VARCHAR2 (20), SEX CHAR (1), SALARY INTEGER,
SUPERSSN REFERENCES EMPLOYEE (SSN),
DNO REFERENCES DEPARTMENT (DNO));

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 PRIMARYKEY, 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));
```

Insertion of values totables

```
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=N
ULL,DNO=‘3‘
WHERESSN=‘
RNSECE01‘;

UPDATE EMPLOYEE SET
SUPERSSN=‘RNSCSE02‘,DN
O=‘5‘
WHERESSN=‘RNSCSE01‘;

UPDATE EMPLOYEE SET
SUPERSSN=‘RNSCSE03‘,DN
O=‘5‘
WHERESSN=‘RNSCSE02‘;

UPDATE EMPLOYEE SET
SUPERSSN=‘RNSCSE04‘,DN
O=‘5‘
WHERESSN=‘RNSCSE03‘;

UPDATE EMPLOYEE SET DNO=‘5‘,
SUPERSSN=‘RNSCSE05‘
WHERESSN=‘RNSCSE04‘;



UPDATE EMPLOYEE
SET DNO=‘5‘,
SUPERSSN=‘RNSCSE06‘
WHERESSN=‘RNSCSE05‘
```

```
UPDATE EMPLOYEE
SET DNO='5',
SUPERSSN=NULL
WHERESSN='RNSCSE06'
;

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

UPDATE EMPLOYEE SET
DNO='1', SUPERSSN=NULL
WHERESSN='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');
INSERTINTODLOCATIONVALUES('MANGALORE',_4');
INSERTINTODLOCATIONVALUES('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,'SALARYMANAGEMENT','BANGALORE','1');
INSERT INTO PROJECT
VALUES(106,'OPENSTACK','BANGALORE','4');
INSERT INTO PROJECT VALUES (107,'SMARTCITY','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:

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=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 a10 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 thisdepartment

```
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
```

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

```
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
```

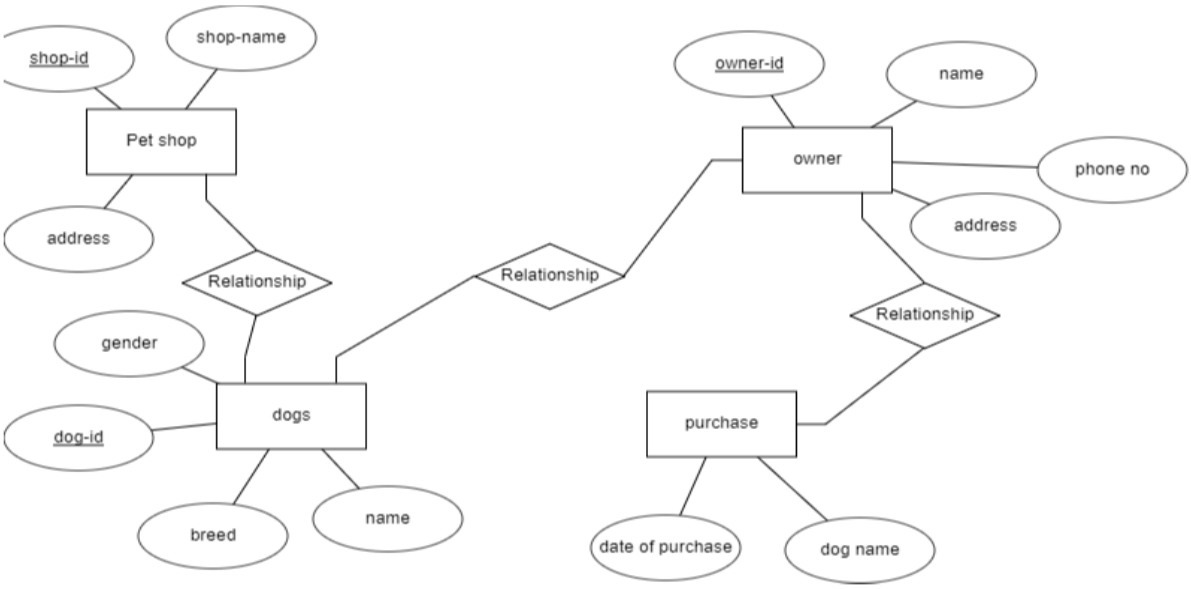
5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

```
SELECT D.DNO, COUNT (*)
FROM DEPARTMENT D,
EMPLOYEE E WHERE
D.DNO=E.DNO
AND E.SALARY>600000
AND .DNO IN (SELECT E1.DNO
    FROM EMPLOYEE E1
    GROUP BY E1.DNO
    HAVING COUNT (*)>5)
            GROUP BY D.DNO;
```

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

3.Puppy pet shop wants to keep track of dogs and their owners. The person can buy maximum three pet dogs. We store person's name, SSN and address and dog's name, date of purchase and sex. The owner of the pet dogs will be identified by SSN since the dog's names are not distinct.

a) Establish the database by normalizing up to 3NF and considering all schema level constraints
b) Write SQL insertion query to insert few tuples to all the relations
c) List all pets owned by a person 'Abhiman'.
d) List all persons who are not owned a single pet
e) Write a trigger to check the constraint that the person can buy maximum three pet dogs
f) Write a procedure to list all dogs and owner details purchased on the specific date.



Create four tables Petstore,Dog,Owner,Purchase.
Insert atleast 5 values for each.

c)List all pets owned by a person 'Abhiman'.

```
Select petid
From owner o,purchase p,pet p1
Where o.name="Abhiman" abnd o.ssn=p.ssn and p.petid=p1.petid
```

d) List all persons who are not owned a single pet

```
select ssn,name
from owner
    minus
select ssn
```

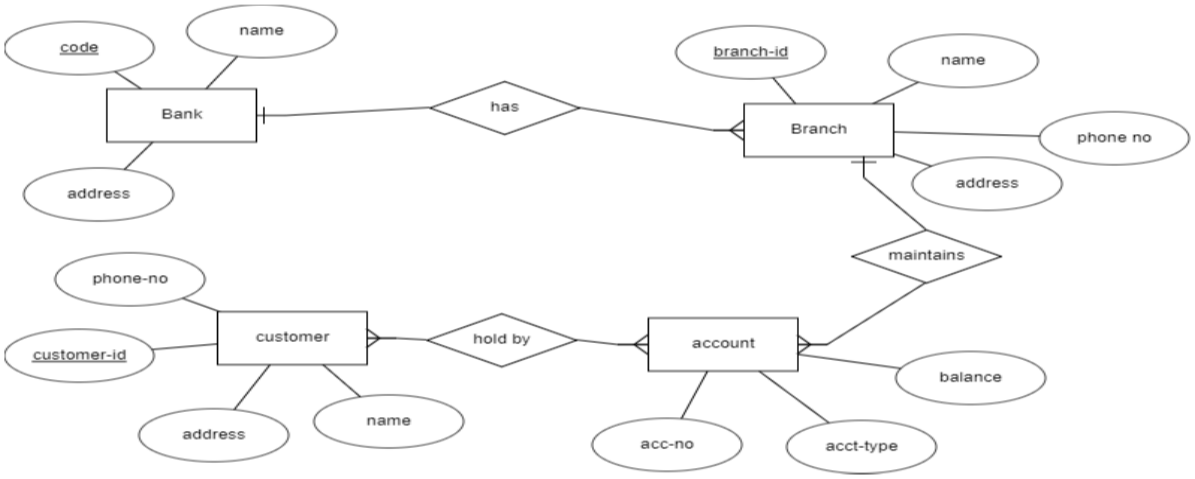
from purchase

e)Write a trigger to check the constraint that the person can buy maximum three pet dogs

```
create trigger maxcount
after insert of
count(*) > 3 on
purchase p,owner o
where p.ssn=o.ssn

maxcount("you cannot purchase dog")
```

4.The commercial bank wants keep track of the customer's account information. The each customer may have any number of accounts and account can be shared by any number of customers. The system will keep track of the date of last transaction. We store the following details.
a) Account: unique account-number, type and balance
b) Customer: unique customer-id, name and several addresses composed of street, city and state

a) Establish the database by normalizing up to 3NF and considering all schema level constraints
b) Write SQL insertion query to insert few tuples to all the relations
c) Add 5% interest to the customer who have less than 10000 balances and 6% interest to remaining customers.
d) List joint accounts involving more than three customers
e) Write a insertion trigger to allow only current date for date of last transaction field.
f) Write a procedure to find the customer who has highest number of accounts, the customer who involved in most of joint accounts.



Queries:

c) Add 5% interest to the customer who have less than 10000 balances and 6% interest to remaining customers.

```
Select interest *1.05 as newinterest
From account
Where balance < 10000 and
(select interest *1.06 as newinterest
From account
Where balance > 10,000);
```

d) List joint accounts involving more than three customers

```
Select accountno,count(*)
From account
Where accounttype='Joint'
Group by accountno
Having count(*)>3;
```

e)Write a insertion trigger to allow only current date for date of last transaction field.

```
Create trigger last_trans
After insert of
Date-of-trans on transaction
Where new_date_trans!='01/01/2022'


Last_trans("year should be current year")
```

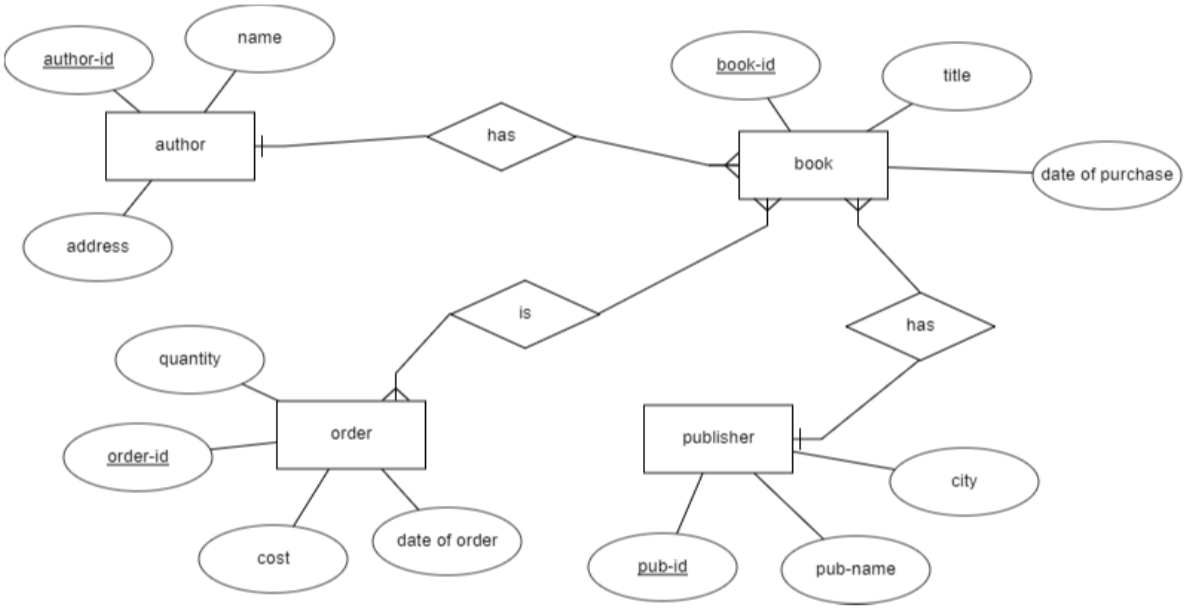
f)Write a procedure to find the customer who has highest number of accounts, the customer who has lowest balance, the customer who involved in most of joint accounts.

```
Create procedure customers
    Select max(count(*)) customer_name from
    Account  a,customer c
    Where a.cus_n=c.cus_nu;

    Select min(balance),c.name
    From account a,customer c
    Where a.cus_n=c.cus_nu;

Exec customers;
```

5.The XYZ Book shop wants keep track of orders of the book. The book is composed of unique id, title, year of publication, single author and single publisher. Each order will be uniquely identified by order-id and may have any number of books. We keep track of quantity of each book ordered. We store the following details for author and publisher. AUTHOR: unique author-id, name, city, country PUBLISHER: unique publisher-id, name, city, country.

a) Establish the database by normalizing up to 3NF
b) Write SQL insertion query to insert few tuples to all the relations
c) Find the author who has published highest number of books
d) List the books published by specific publisher during the year 2011.
e) Write before insertion trigger to book to check year of publication should allow current year only.

f)Write a procedure to list all the books published by a specific author during the specific year



c.Find the author who has published highest number of books

```
select a.name,max(count(*))
from book b,author a
where b.author-id=a.author-id
 group by b.author-id
```

d.List the books published by specific publisher during the year 2011.

```
Select b.title
From book b,publisher p
Where b.pub-id=p.pub-id
Group by p.pub-id
Having p.date=2011
```

e.Write before insertion trigger to book to check year of publication should allow current year only.

```
Create trigger year_pub
Before insert of
Date on book
New.date!='01/01/2022'

Year_pub("year should be current year")
```

f.Write a procedure to list all the books published by a specific author during the specific year

```
create procedure books(@author,@year)
select b.title from book b,author a
where  b.year=@year and a.name=@author and b.author-id=a.author-id;

exec books("abc','2/01/2022')
```