1.**Consider the following schema for Bank Database:**
BRANCH(Branch_id, Bank_name, Branch_name, Assets)
ACCOUNT(Acc_no, Branch_id, Account_Type, Account_Balance, Customer_id)
CUSTOMER(Customer_id, Customer_name, Customer_age, Customer_address, Customer_phone) LOAN(Loan_number, Branch_id, Amount, Customer_id)
Note: Account_Type may be of following: Savings, Recurrent, Fixed Deposit
Write SQL queries to

1. Find all the Customers who have at least one account at the "Mangaluru" branch.
2. Find names of the depositors who have deposited highest amount among all the customers.
3. Retrieve the Customer name and loan amount of a customer who borrowed a loan more than 5,00,000.
4. Retrieve the details of bank branch with maximum and minimum assets among the various branches.
5. Demonstrate how you delete all account tuples at every branch located in a specific city.

SOLUTIONS:

i) Create the above tables by properly specifying the primary keys and the foreign keys.
ii) Enter at least five tuples for each relation.

```
CREATE TABLE BRANCH(
        BRANCH_ID VARCHAR(10),
        BANK_NAME VARCHAR(15),
        BRANCH_NAME VARCHAR(20),
        ASSETS INT NOT NULL,
        PRIMARY KEY(BRANCH_ID));

INSERT INTO BRANCH VALUES('B1','CANARA','MANGALURU',60000000)

SELECT * FROM BRANCH;
```

BRANCH_ID	BANK_NAME	BRANCH_NAME	ASSETS
B1	CANARA	MANGALURU	60000000
B2	BANK OF BARODA	MANGALURU	70000000
B3	CANARA	KASARAGOD	50000000
B4	SBI	BENGALURU	30000000
B5	UNION BANK	DELHI	20000000

CREATE TABLE ***CUSTOMER***(

CUSTOMER_ID VARCHAR(10),
CUSTOMER_NAME VARCHAR(20),
CUSTOMER_AGE INT,
CUSTOMER_ADDRESS VARCHAR(20),
CUSTOMER_PHONE INT,
PRIMARY KEY(CUSTOMER_ID));

INSERT INTO CUSTOMER VALUES('C1','RAVI',22,'MANGALURU',8745263);

SELECT * FROM CUSTOMER;

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_AGE	CUSTOMER_ADDRESS	CUSTOMER_PHONE
C1	RAVI	22	MANGALURU	8745263
C2	ASHA	26	DELHI	98745641
C3	VARUN	23	KASARGOD	78954623
C4	ARPITHA	22	MANGALURU	9856325
C5	SACHIN	23	BENGALORE	78541365

TABLE ACCOUNT(

ACC_NO INT,
BRANCH_ID VARCHAR(10),
ACCOUNT_TYPE VARCHAR(10),
ACCOUNT_BALANCE INT,
CUSTOMER_ID VARCHAR(10),
PRIMARY KEY(ACC_NO),
FOREIGN KEY (BRANCH_ID) REFERENCES
BRANCH(BRANCH_ID) ON DELETE CASCADE,
FOREIGN KEY(CUSTOMER_ID) REFERENCES
CUSTOMER(CUSTOMER_ID) ON DELETE CASCADE);

INSERT INTO ACCOUNT VALUES(123,'B1','SAVINGS',10000,'C1');
SELECT * FROM ACCOUNT;

ACC_NO	BRANCH_ID	ACCOUNT_TYPE	ACCOUNT_BALANCE	CUSTOMER_ID
123	B1	SAVINGS	10000	C1
456	B5	RECURRING	20000	C2
789	B1	SAVINGS	30000	C1
1122	B2	FD	5000	C3

1334	B1	SAVINGS	10000	C4
1234	B3	FD	90000	C5
5876	B4	RECURRING	80000	C3

```
CREATE TABLE LOAN(
        LOAN_NUMBER VARCHAR2(5),
        BRANCH_ID VARCHAR(10),
        AMOUNT INT,
        CUSTOMER_ID VARCHAR(10),
        PRIMARY KEY(LOAN_NUMBER),
        FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH(BRANCH_ID)
        ON DELETE CASCADE,
        FOREIGN KEY(CUSTOMER_ID) REFERENCES
        CUSTOMER(CUSTOMER_ID) ON DELETE CASCADE);

INSERT INTO LOAN VALUES('L1','B1',500000,'C1');

SELECT * FROM LOAN;
```

LOAN_NUMBER	BRANCH_ID	AMOUNT	CUSTOMER_ID
L1	B1	500000	C1
L2	B2	50000	C2
L3	B3	40000	C3
L4	B2	565000	C4
L5	B4	955000	C5
L6	B5	20000	C2

1.Find all the customers who have at least one account at the “Mangaluru” branch.

```
Select c.customer_id,c.customer_name from customer c,account a,branch b where
b.branch_name='mangaluru'and b.branch_id=a.branch_id and a.customer_id=c.customer_id;
```

CUSTOMER_ID	CUSTOMER_NAME
C1	RAVI
C1	RAVI
C3	VARUN
C4	ARPITHA

2.Find names of the depositors who have deposited highest amount among all the customers.

```
Select c.customer_id,c.customer_name,a.account_balance from
customer c,account a where c.customer_id=a.customer_id and
account_balance=(select max(account_balance)from account);
```

CUSTOMER_I	CUSTOMER_NAME	ACCOUNT_BALANCE

C5	SACHIN	90000

3.**Retrieve the Customer name and loan amount of a customer who borrowed a loan more than 5,00,000**.

```
Select c.customer_name c,l.amount from customer c,loan l
where c.customer_id=l.customer_id and amount>500000;
```

CUSTOMER_NAME	AMOUNT
ARPITHA	565000
ARPITHA	955000

4.Retrieve the details of bank branch with maximum and minimum assets among the various branches.

```
Select bank_name,branch_name,assets from branch
where assets=(select max(assets)from branch)
union
select bank_name,branch_name,assets from branch
where assets=(select min(assets)from branch);
```

BANK_NAME	BRANCH_NAME	ASSETS
BANK OF BARODA	MANGALURU	70000000
UNION BANK	DELHI	20000000

5. Demonstrate how you delete all account tuples at every branch located in a specific city.

```
Delete from branch where branch_name='delhi';
```

Schema diagram:

BRANCH

Branch_id	Bank_name	Branch_name	Assets

ACCOUNT

Acc_no	Branch_id	Account_type	Account_bal	Customer_id

CUSTOMER

Customer_id	Customer_name	Customer_age	Customer_address	Customer_phone

LOAN

Loan_number	Branch_id	Amount	Customer_id

2.**Consider the following schema for a Library Database**

BOOK(Book_id, Title, Publisher_Name,

Pub_Year) BOOK_AUTHORS(Book_id,

Author_Name) PUBLISHER(Name, Address,

Phone) BOOK_COPIES(Book_id,

Programme_id, No-of_Copies)

BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date)

LIBRARY_PROGRAMME(Programme_id, Programme_Name,Address)

Write SQL queries to

1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each Programme, etc.
2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2023 to Jun 2023.
3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.
4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
5. Create a view of all books and its number of copies that are currently available in the Library.

SOLUTIONS:

```
CREATE TABLE PUBLISHER(
        NAME VARCHAR2(20) PRIMARY KEY,
        ADDRESS VARCHAR2(20),
        PHONE NUMBER(10));

INSERT INTO PUBLISHER VALUES('PEARSON','LONDON',9874522224);
INSERT INTO PUBLISHER VALUES('TATAMCGRAW','NEWYORK',9858523565);
INSERT INTO PUBLISHER VALUES('OXFORD','UK',9885121112);
INSERT INTO PUBLISHER VALUES('CAMBRIDGE','UK',9785634615);
INSERT INTO PUBLISHER VALUES('OREILLY','CALIFORNIA',9994125455);


SELECT * FROM PUBLISHER;
```

NAME	ADDRESS	PHONE
PEARSON	LONDON	874522224
TATAMCGRAW	NEWYORK	858523565
OXFORD	UK	9885121112
CAMBRIDGE	UK	9785634615
OREILLY	CALIFORNIA	9994125455

```
CREATE TABLE BOOK (
        BOOK_ID VARCHAR2(20) PRIMARY KEY,
        TITLE VARCHAR2(40),
        PUBLISHER_NAME VARCHAR2(20) references
        PUBLISHER(NAME) on delete cascade,
        PUB_YEAR INT);

INSERT INTO BOOK VALUES ('B101','DBMS','Pearson',2017);
INSERT INTO BOOK VALUES ('B102','AIML','TataMcGraw',2009);
INSERT INTO BOOK VALUES ('B103','DCN','Pearson',2017);
INSERT INTO BOOK VALUES ('B104','ATC','Oxford',2017);
INSERT INTO BOOK VALUES ('B105','PYTHON','OReilly',2014);
INSERT INTO BOOK VALUES ('B106','HADOOP','Pearson',2000);

SELECT * FROM BOOK;
```

BOOK_ID	TITLE	PUBLISHER_NAME	PUB_YEAR
B101	DBMS	PEARSON	2017
B102	AIML	TATAMCGRAW	2009
B103	DCN	PEARSON	2017
B104	ATC	OXFORD	2017
B105	PYTHON	OREILLY	2014
B106	HADOOP	PEARSON	2000

```
CREATE  TABLE  BOOK_AUTHORS(
        BOOK_ID varchar(20),
        AUTHOR_NAME VARCHAR2(20),
        PRIMARY KEY(BOOK_ID, AUTHOR_NAME),
        FOREIGN KEY(BOOK_ID) REFERENCES BOOK ON DELETE CASCADE);

INSERT INTO BOOK_AUTHORS VALUES('B101', 'ELMARSI');
INSERT INTO BOOK_AUTHORS VALUES('B101', 'NAVATHE');
INSERT INTO BOOK_AUTHORS VALUES('B101', 'RAMAKRISHNAN');
INSERT INTO BOOK_AUTHORS VALUES('B106', 'DOUGLAS');
INSERT INTO BOOK_AUTHORS VALUES('B102', 'ELAINE');
INSERT INTO BOOK_AUTHORS VALUES('B105', 'SRINIVASAN');

SELECT * FROM BOOK_AUTHORS;
```

BOOK_ID	AUTHOR_NAME
B101	ELMARSI
B101	NAVATHE
B101	RAMAKRISHNAN

B102	ELAINE
B105	SRINIVASAN
B106	DOUGLAS

```
CREATE TABLE LIBRARY_PROGRAMME(
        PROGRAMME_ID VARCHAR(20) PRIMARY_KEY,
        PROGRAMME_NAME VARCHAR(10),
        ADDRESS VARCHAR(20));

INSERT INTO LIBRARY_PROGRAMME VALUES ('L1','SAHYADRI','MANGALORE');
INSERT INTO LIBRARY_PROGRAMME VALUES ('L2','SAPNA','MANGALORE');
INSERT INTO LIBRARY_PROGRAMME VALUES ('L3','SANKALP','BANGALORE');
INSERT INTO LIBRARY_PROGRAMME VALUES ('L4','PENGUIN','CHENNAI');
INSERT INTO LIBRARY_PROGRAMME VALUES ('L5','AGNES','CHENNAI');

SELECT * FROM LIBRARY_PROGRAMME;
```

PROGRAMME_ID	PROGRAMME_NAME	ADDRESS
L1	SAHYADRI	MANGALORE
L2	SAPNA	MANGALORE
L3	SANKALP	BANGALORE
L4	PENGUIN	CHENNAI
L5	AGNES	CHENNAI

```
CREATE TABLE BOOK_COPIES(
        BOOK_ID VARCHAR(20)
        REFERENCES BOOK(BOOK_ID)
        ON DELETE CASCADE,
        PROGRAMME_ID VARCHAR(20)
        REFERENCES
        LIBRARY_PROGRAMME(PROGRAMME_ID) ON DELETE CASCADE,
        NO_OF_COPIES NUMBER(2),
        PRIMARY KEY(BOOK_ID, PROGRAMME_ID));

INSERT INTO BOOK_COPIES VALUES ('B101','L1',99);
INSERT INTO BOOK_COPIES VALUES ('B102','L1',99);
INSERT INTO BOOK_COPIES VALUES ('B103','L2',99);
INSERT INTO BOOK_COPIES VALUES ('B103','L1',99);
```

SELECT * FROM BOOK_COPIES;

BOOK_ID	PROGRAMME_ID	NO_OF_COPIES
B101	L1	99
B101	L2	100
B102	L1	99
B102	L2	100
B103	L2	10
B103	L1	9

```
CREATE TABLE  BOOK_LENDING(
        BOOK_ID VARCHAR(20),
        PROGRAMME_ID VARCHAR(20),
        CARD_NO VARCHAR(20),
        DATE_OUT DATE,
        DUE_DATE DATE,
        PRIMARY KEY(PROGRAMME_ID, BOOK_ID, CARD_NO),
        FOREIGN KEY(BOOK_ID)
        REFERENCES book(BOOK_ID) ON DELETE CASCADE,
        FOREIGN KEY(PROGRAMME_ID)
        REFERENCES LIBRARY_PROGRAMME(PROGRAMME_ID) ON DELETE
       CASCADE,
      CONSTRAINT CK1 CHECK (DUE_DATE > DATE_OUT));
```

```
INSERT INTO BOOK_LENDING VALUES('B101','L1','FA101','02-JAN-21','09-JAN-21');
INSERT INTO BOOK_LENDING VALUES('B101','L1','FA102','02-MAR-23','09-MAR-23');
INSERT INTO BOOK_LENDING VALUES('B102','L1','FA102','02-MAR-23','09-MAR-23');
INSERT INTO BOOK_LENDING VALUES('B101','L2','FA102','02-MAR-23','09-MAR-23');
INSERT INTO BOOK_LENDING VALUES('B101','L1','S103','04-APR-22','30-JUN-22');
```

SELECT * FROM BOOK_LENDING;

BOOK_ID	PROGRAMME_ID	CARD_NO	DATE_OUT	DUE_DATE
B101	L1	FA101	02-JAN-21	09-JAN-21
B101	L1	FA102	02-MAR-23	09-MAR-23
B102	L1	FA102	02-MAR-23	09-MAR-23
B101	L2	FA102	02-MAR-23	09-MAR-23
B101	L1	S103	04-APR-22	30-JUN-22

1. **Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each Programme, etc.**

```
SELECT B.BOOK_ID, TITLE, PUBLISHER_NAME,AUTHOR_NAME,
NO_OF_COPIES FROM BOOK B, BOOK_AUTHORS A, BOOK_COPIES BC
WHERE B.BOOK_ID = BC.BOOK_ID AND  B.BOOK_ID = A.BOOK_ID;
```

BOOK_ID	TITLE	PUBLISHER_NAME	AUTHOR_NAME	NO_OF_COPIES
B101	DBMS	PEARSON	ELMARSI	99
B101	DBMS	PEARSON	NAVATHE	99
B101	DBMS	PEARSON	RAMAKRISHNAN	99
B102	AIML	TATAMCGRAW	ELAINE	99

2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2023 to Jun 2023

```
SELECT CARD_NO  FROM BOOK_LENDING WHERE DATE_OUT  BETWEEN '01-
 JAN-2023' AND '30-JUN-2023' GROUP BY CARD_NO HAVING COUNT(*) >= 3 ;
```

CARD_NO
FA102

3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.

```
DELETE FROM BOOK WHERE BOOK_ID = B103;
```

4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

```
CREATE TABLE BOOK1 (
        BOOK_ID VARCHAR2(20) PRIMARY KEY,
        TITLE VARCHAR2(40),
        PUBLISHER_NAME VARCHAR2(20) references
        PUBLISHER(NAME) on delete cascade,
        PUB_YEAR INT)
        PARTITION BY RANGE(pub_year)
        (PARTITION p1 VALUES LESS THAN(2001),
        PARTITION p2 VALUES LESS THAN(2005),
        PARTITION P3 VALUES LESS THAN (2010),
        PARTITION P4 VALUES LESS THAN(MAXVALUE));

INSERT INTO BOOK1 VALUES ('B101','DBMS','PEARSON',2017);
INSERT INTO BOOK1 VALUES ('B102','AIML','TATAMCGRAW',2009);
INSERT INTO BOOK1 VALUES ('B103','DCN','PEARSON',2017);
INSERT INTO BOOK1 VALUES ('B104','ATC','OXFORD',2017);
INSERT INTO BOOK1 VALUES ('B105','PYTHON','OREILLY',2014);
INSERT INTO BOOK1 VALUES ('B106','HADOOP','PEARSON',2000);

SELECT * FROM BOOK1 PARTITION(P1);
```

BOOK_ID	TITLE	PUBLISHER_NAME	PUB_YEAR
B106	HADOOP	PEARSON	2000

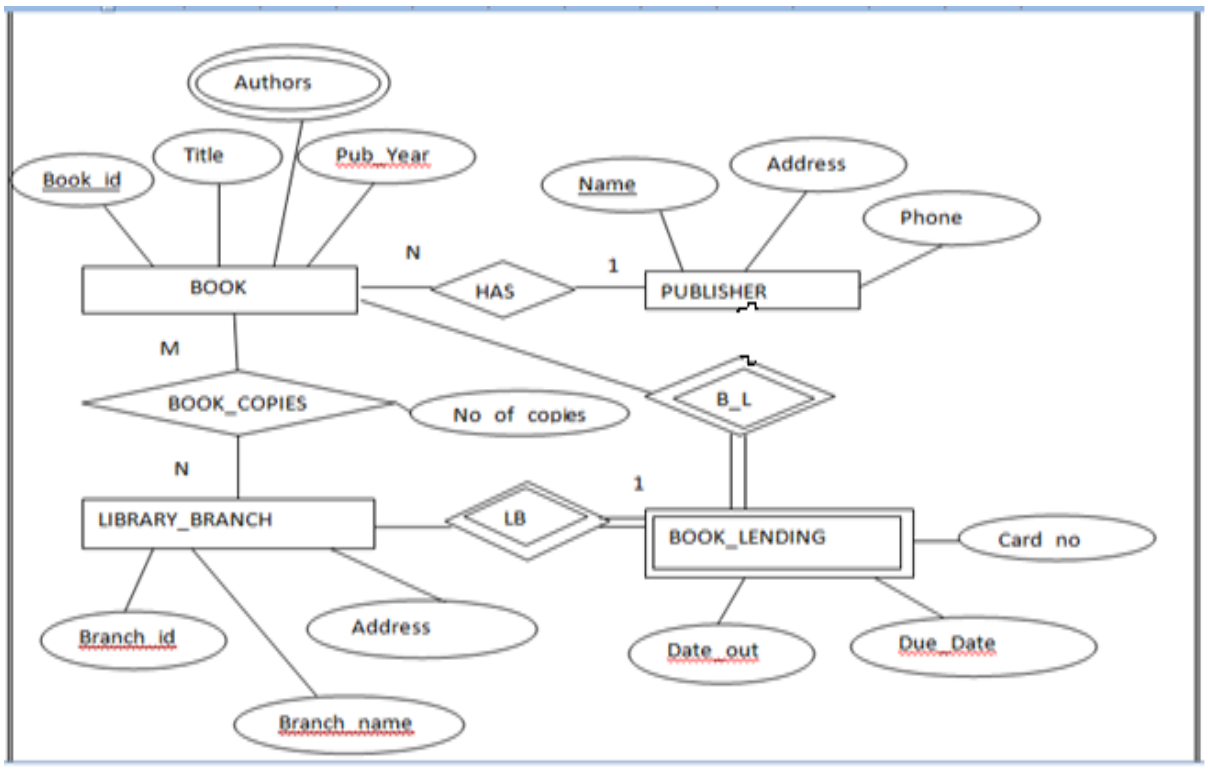
5. **Create a view of all books and its number of copies that are currently available in the Library.**

Create view available_book As Select b.book_id, b.title,
sum(bc.no_of_copies)- (select count(*) from book_lending bl where bl.book_id= b. book_id group by bl.book_id) as books_available from book b, book_copies bc where b.book_id=bc.book_id group by b.book_id,b.title;

SELECT * FROM AVILABLE_BOOK;

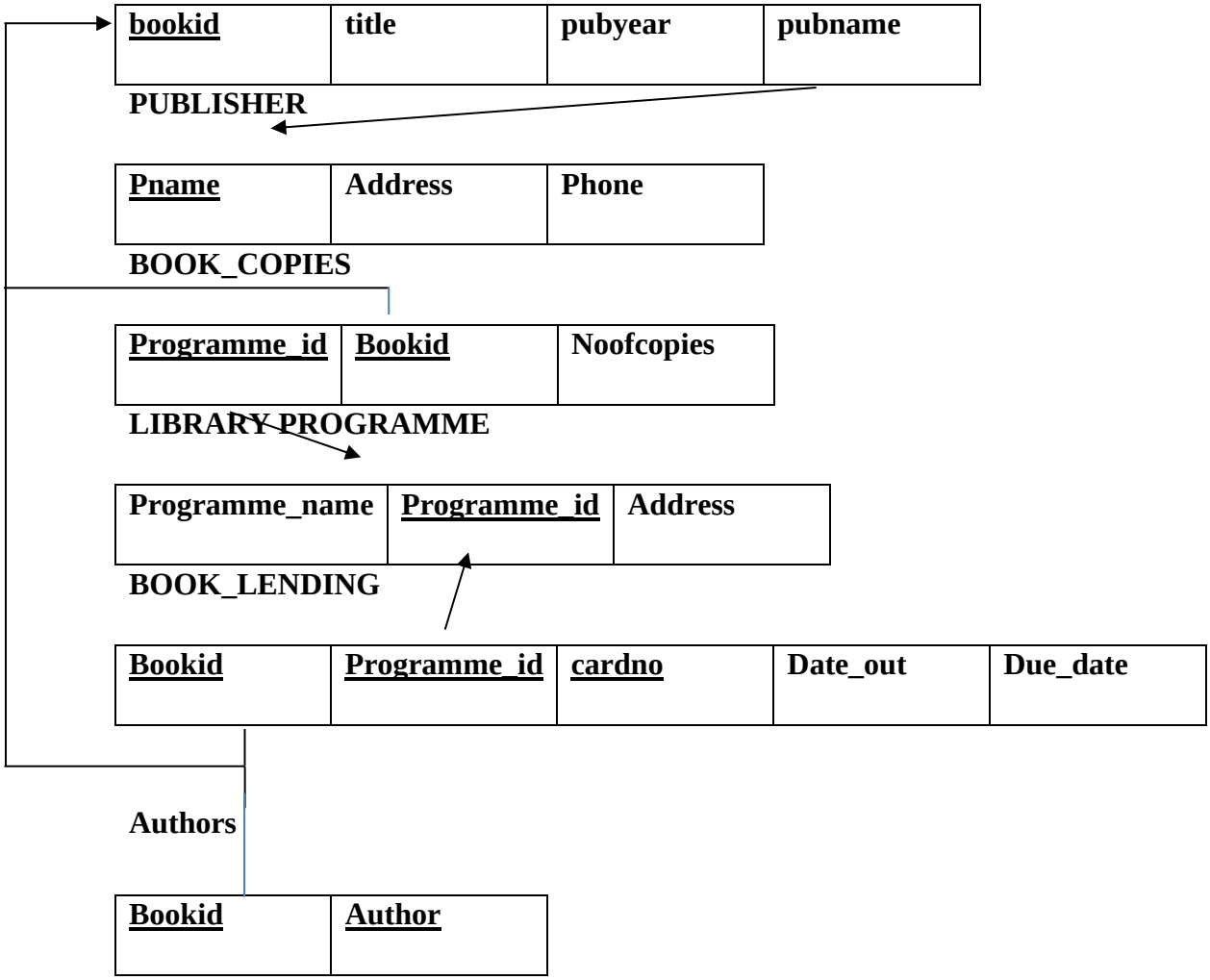
BOOK_ID	TITLE	BOOKS_AVAILABLE
B101	DBMS	195
B102	AIML	198
B103	DCN	

ER DIAGRAM:



SCHEMA DIAGRAM:

BOOK


bookid
title
pubyear
pubname
PUBLISHER
Pname
Address
Phone
BOOK_COPIES
Programme_id
Bookid
Noofcopies
LIBRARY PROGRAMME
Programme_name
Programme_id
Address
BOOK_LENDING
Bookid
Programme_id
cardno
Date_out
Due_date
Authors
Bookid
Author

3.Consider the schema for College Database:

STUDENT(USN, SName, Address, Phone,

Gender) SEMSEC(SSID, Sem, Sec)

CLASS(USN, SSID)

COURSE(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 "4SF20CD001" in all Courses.
4. Calculate the FinalIA (average of three test marks) and update the corresponding table for all students.
5. Categorize students based on the following criterion:

   If FinalIA = 45 to 50 then CAT = "Outstanding" If FinalIA= 40 to 45 then CAT= "Good"

   If FinalIA = 30 to 40 then CAT = "Average"

   If FinalIA< 30 then CAT = "Weak"

Give these details only for 8th semester A, B, and C section students.

SOLUTIONS:

```
CREATE TABLE STUDENT(
        USN     VARCHAR2(20),
        SNAME   VARCHAR2(10),
        ADDRESS VARCHAR2(10),
        PHONE   NUMBER(10),
        GENDER  VARCHAR2(10),
        PRIMARY KEY(USN));

INSERT INTO STUDENT VALUES('4SF20CS089','AJAY','MANGALORE',733825,'MALE');

SELECT * FROM STUDENT;
```

USN	SNAME	ADDRESS	PHONE	GENDER
4SF20CS089	AJAY	MANGALORE	733825	MALE
4SF20IS109	VARSHINI	BANTWAL	896523	FEMALE
4SF20CS098	NAVISH	UDUPI	9956258	MALE
4SF20CD001	SAHANA	SURATHKAL	8752683	FEMALE
4SF20CS088	KAVYA	MANGALORE	78965231	FEMALE

```
CREATE TABLE SEMSEC(
        SSID   NUMBER(5),
        SEM   NUMBER(2),
        SECTION  VARCHAR2(1),
        PRIMARY KEY(ssid));

INSERT INTO SEMSEC VALUES(1,4,'A');

SELECT * FROM SEMSEC;
```

SSID	SEM	SECTION
1	4	A
2	4	C
3	8	A
4	8	B
5	8	C
6	5	A

```
CREATE TABLE CLASS(
        SSID NUMBER(5),
        USN  VARCHAR2(20),
        PRIMARY KEY(USN),
        FOREIGN KEY(ssid) REFERENCES SEMSEC(SSID),
        FOREIGN KEY(USN) REFERENCES STUDENT(USN));

INSERT INTO CLASS VALUES(1,'4SF20CD001');


SELECT * FROM CLASS;
```

SSID	USN
1	4SF20CD001
2	4SF20CS088
3	4SF20CS089
4	4SF20IS109
5	4SF20CS098

```
CREATE TABLE SUBJECT(
        SUBCODE VARCHAR2(7) PRIMARY KEY,
        TITLE VARCHAR2(20),
        SEM NUMBER(4),
        CREDIT NUMBER(2));

INSERT INTO SUBJECT VALUES('20CS31','DATA STRUCTURE',4,4);
```

SELECT * FROM SUBJECT;

SUBCODE	TITLE	SEM	CREDIT
20CS31	DATA STRUCTURE	4	4
20CS32	UNIX	4	3
20CS33	DBMS	5	4
20CS34	DCN	5	3
20CS35	AIML	7	4

```
CREATE TABLE IAMARKS(
        USN  VARCHAR2(20),
        SUBCODE VARCHAR2(7),
        SSID NUMBER(5),
        TEST1 NUMBER(3),
        TEST2 NUMBER(3),
        TEST3 NUMBER(3),
        FINALIA NUMBER(3),
        PRIMARY KEY(USN,SUBCODE,SSID),
        FOREIGN KEY(USN) REFERENCES STUDENT(USN),
        FOREIGN KEY(SUBCODE) REFERENCES SUBJECT(SUBCODE),
        FOREIGN KEY(SSID) REFERENCES SEMSEC(SSID));
```

```
INSERT INTO IAMARKS VALUES('4SF20CD001',1,'20CS31',38,35,32,0);
```

SELECT * FROM IAMARKS;

USN	SUBCODE	SSID	TESTT1	TEST2	TEST3	FINALIA
4SF20CD001	20CS31	1	38	35	32	0
4SF20CD001	20CS32	1	28	26	29	0
4SF20CS088	20CS31	2	38	42	32	0
4SF20CS089	20CS33	3	42	46	41	0
4SF20IS109	20CS34	4	28	26	29	0
4SF20CS098	20CS35	5	48	46	50	0

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

```
SELECT a.*,b.sem,b.section FROM student a,semsec b,class c WHERE a.usn=c.usn and
b.ssid=c.ssid AND b.sem=4 AND b.section='c';
```

USN	SNAME	ADRESS	PHONE	GENDER	SEM	SECTION
4SF20CS088	KAVYA	MANGALORE	78965231	FEMALE	4	C

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

SELECT sem,section,gender,count(*) FROM student s, semsec s1,class c WHERE s.usn=c.usn AND s1.ssid=c.ssid GROUP BY (gender,sem,section) ORDER BY(sem);

SEM	SECTION	GENDER	COUNT(*)
4	A	FEMALE	1
4	C	FEMALE	1
8	B	FEMALE	1
8	A	MALE	1
8	C	MALE	1

3. Create a view of Test1 marks of student USN "4SF20CD001" in all Courses

Create view test_marks As select subcode ,testt1 from iamarks
Where usn='4sf20cd001';

SELECT * FROM TEST_MARKS;

SUBCODE	TESTT1
20CS31	38
20CS32	28

4.Calculate the FinalIA (average of three test marks) and update the corresponding table for all students.

UPDATE iamarks SET finalia=GREATEST((testT1+test2),(testT1+test3),(test3+test2))/2;

USN	SUBCODE	SSID	TESTT1	TEST2	TEST3	FINALIA
4SF20CD001	20CS31	1	38	35	32	37
4SF20CD001	20CS32	1	28	26	29	29
4SF20CS088	20CS31	2	38	42	32	40
4SF20CS089	20CS33	3	42	46	41	44
4SF20IS109	20CS34	4	28	26	29	29
4SF20CS098	20CS35	5	48	46	50	49

5. Categorize students based on the following criterion:

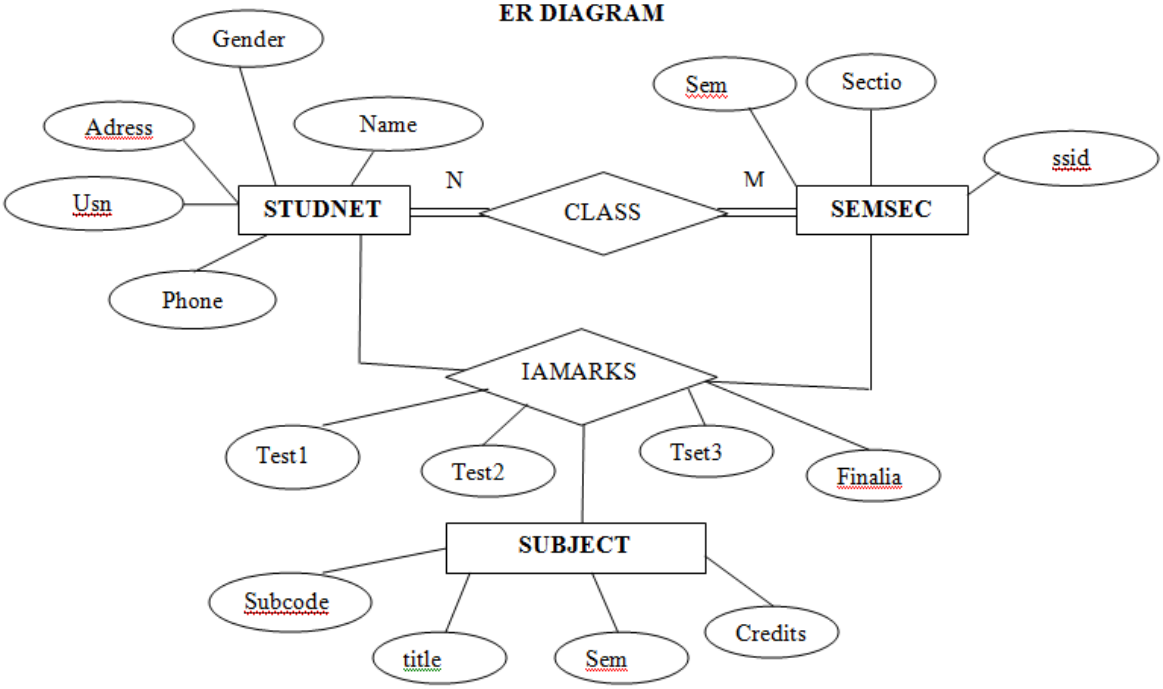
If FinalIA = 45 to 50 then CAT = "Outstanding" If FinalIA= 40 to 45 then CAT= "Good"
If FinalIA = 30 to 40 then CAT = "Average"
If FinalIA< 30 then CAT = "Weak"

```
SELECT usn,finalia,
    CASE
    WHEN finalia BETWEEN 45 AND 50 THEN 'OUTSTANDING'
    WHEN finalia BETWEEN 40 AND 45 THEN 'GOOD'
    WHEN finalia BETWEEN 30 AND 40 THEN 'AVERAGE'
    WHEN finalia <30  THEN 'WEAK'
```

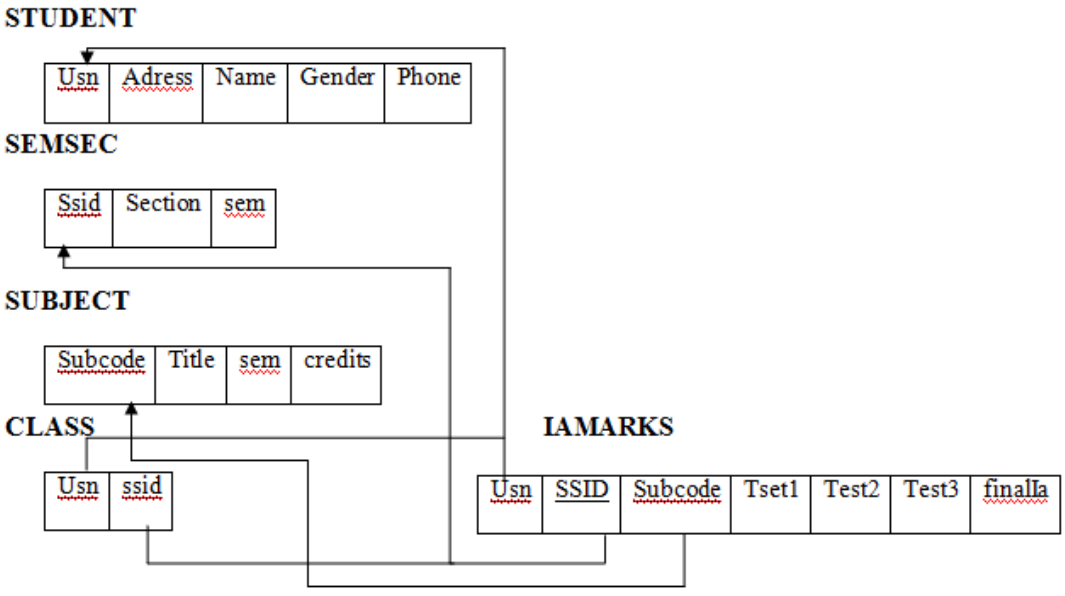
END
AS CATEGORY FROM iamarks i,semsec s WHERE i.ssid=s.ssid
AND sem=8 AND section IN('A','B','C');

USN	FINALIA	CATEGORY
4SF20CS089	44	GOOD
4SF20IS109	29	WEAK
4SF20CS098	49	OUTSTANDING

ER DIAGRAM



SCHEMA DIAGRAM



4.Consider the schema for Company Database:

EMPLOYEE (Eid, Name, Address, Gender, Salary,

SuperEid, Dno) DEPARTMENT (Dnum, Dname,

DMgr_id, Mgr_start_date) DLOCATION (Dno,

Dlocation)

PROJECT (Pnum, Pname, Plocation,Dno)

WORKS_ON (Eid, Pno, Hours)

DEPENDENT (Empid, Dep_name, Gender, Bdate, Relationship)

Write SQL queries to

1. Make a list of all project numbers for projects that involve an employee whose name is "Rahul",
   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. Create a view Dept_info that gives details of department name, Number of employees and total
   salary of each department.

SOLUTIONS :

```
CREATE TABLE EMPLOYEE(
        EID INT PRIMARY KEY,
        NAME VARCHAR2(20),
        ADDRESS VARCHAR2(20),
        GENDER CHAR(1) CHECK(GENDER ='M' OR GENDER ='F'),
        SALARY NUMBER(6),
        SUPEREID REFERENCES EMPLOYEE(EID),
        DNO NUMBER);

INSERT INTO EMPLOYEE VALUES(1, 'RAHUL' , 'MANGALURU', 'M',35000,1,NULL);
INSERT INTO EMPLOYEE VALUES(2, 'SAHANA', 'MANGALURU', 'F',35000,1,NULL);
INSERT INTO EMPLOYEE VALUES(3, 'SAGAR', 'BENGALURU', 'M',35000,1,NULL);
INSERT INTO EMPLOYEE VALUES(4, 'SAGARIK',  'MANGALURU',M',35000,1,NULL);
INSERT INTO EMPLOYEE VALUES(5, 'SAJAAN', 'MYSORE', 'M',600000,1,NULL);

SELECT * FROM EMPLOYEE;
```

EID	NAME	ADDRESS	GENDER	SALARY	SUPEREID	DNO

1	Rahul	Mangaluru	M	35000	1	NULL
2	Sahana	Mangaluru	F	35000	1	NULL
3	Sagar	Bengaluru	M	35000	1	NULL
4	Sagarik	Mangaluru	M	35000	1	NULL
5	Sajaan	Mysore	M	600000	1	NULL

```
CREATE TABLE DEPARTMENT (
        DNUM NUMBER(5) PRIMARY KEY,
        DNAME VARCHAR2(10) ,
        DMGR_ID REFERENCES EMPLOYEE(EID),
        MGR_START_DATE DATE);
```

```
INSERT INTO department VALUES(1,'CSE',1,'2-Nov-2007');
INSERT INTO department VALUES(2,'IOT',2,'2-Nov-2007');
INSERT INTO department VALUES(3,'Account',2,'2-Nov-2017');
INSERT INTO department VALUES(4,'ISE',1,'2-Nov-2000');
INSERT INTO department VALUES(5,'Finance',1,'3-Nov-2001');
```

```
SELECT * FROM DEPARTMENT;
```

DNUM	DNAME	DMGR_ID	MGR_START
1	CSE	1	02-NOV-07
2	IOT	2	02-NOV-07
3	Account	2	02-NOV-17
4	ISE	1	02-NOV-00
5	Finance	1	03-NOV-01

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT FK FOREIGN KEY(DNO)
REFERENCES
DEPARTMENT(DNUM);
```

```
UPDATE EMPLOYEE SET DNO=4 WHERE EID=1;
UPDATE EMPLOYEE SET DNO=1 WHERE EID=2;
UPDATE EMPLOYEE SET DNO=3 WHERE EID=3;
UPDATE EMPLOYEE SET DNO=3 WHERE EID=4;
UPDATE EMPLOYEE SET DNO=3 WHERE EID=5;
```

```
SELECT * FROM EMPLOYEE;
```

EID	NAME	ADDRESS	GENDER	SALARY	SUPEREID	DNO
1	Rahul	Mangaluru	M	35000	1	4
2	Sahana	Mangaluru	F	35000	1	1
3	Sagar	Bengaluru	M	35000	1	3
4	Sagarik	Mangaluru	M	35000	1	3
5	Sajaan	Mysore	M	600000	1	3

```
CREATE TABLE DLOCATION
        (DNO REFERENCES DEPARTMENT(DNUM),
        LOCATION  VARCHAR2(10),
        PRIMARY KEY(DNO,LOCATION));

INSERT INTO DLOCATION VALUES(1,'MANGALURU');
INSERT INTO DLOCATION VALUES(1,' MYSORE');
INSERT INTO DLOCATION VALUES(2,'MANGALURU');
INSERT INTO DLOCATION VALUES(3,' BENGALURU');
INSERT INTO DLOCATION VALUES(4,'MANGALURU');
INSERT INTO DLOCATION VALUES(5,'MANGALURU');

SELECT * FROM DLOCATION;
```

DNO	LOCATION
1	MANGALURU
1	MYSORE
2	MANGALURU
3	BENGALURU
4	MANGALURU
5	MANGALURU

```
CREATE TABLE PROJECT(
        PNUM NUMBER(2) PRIMARY KEY,
        PNAME VARCHAR2(20),
        PLOCATION VARCHAR2(20),
        DNO NUMBER  REFERENCES DEPARTMENT(DNUM));

INSERT INTO PROJECT VALUES(1,'IOT,'MANAGLURU',1);
INSERT INTO PROJECT VALUES(2,'DATA MINING','MANAGLURU',1);
INSERT INTO PROJECT VALUES(3,'CC,'HUBLI',3);
INSERT INTO PROJECT VALUES(4,'IMAGE PROCESSING,'MANAGLURU',4);
```

INSERT INTO PROJECT VALUES(5,’RESEARCH’,'MANAGLURU',5);

SELECT * FROM PROJECT;

PNUM	PNAME	PLOCATION	DNO
1	IOT	MANAGLURU	1
2	DATA MINING	MANAGLURU	1
3	CC	HUBLI	3
4	IMAGE PROCESSING	MANAGLURU	4
5	RESEARCH	MANAGLURU	5

```
CREATE TABLE  WORKSON (
        EID  NUMBER(5) REFERENCES EMPLOYEE(EID),
        PNO  NUMBER(2) REFERENCES   PROJECT(PNUM),
        HOURS NUMBER(5,2),
        PRIMARY KEY(EID,PNO));
```

```
INSERT INTO WORKSON VALUES(1,1,4);
INSERT INTO WORKSON VALUES(2,1,5);
INSERT INTO WORKSON VALUES(3,2,4);
INSERT INTO WORKSON VALUES(4,3,4);
INSERT INTO WORKSON VALUES(5,5,4);
```

SELECT * FROM WORKSON;

EID	PNO	HOURS
1	1	4
2	1	5
3	2	4
4	3	4
5	5	4

```
CREATE TABLE DEPENDENT(
                EMP_ID INT CONSTRAINT DEP_EMPID_PK PRIMARY KEY,
                DEPENDENT_NAME VARCHAR2(12),
                GENDER VARCHAR2(5),
                BDATE DATE,
                RELATIONSHIP VARCHAR2(12),
                FOREIGN KEY(EMP_ID)REFERENCES EMPLOYEE(EID) ON
        DELETE
                CASCADE);
```

INSERT INTO DEPENDENT VALUES(1,' RAMESH','M',' 01-APR-99',' BROTHER');
INSERT INTO DEPENDENT VALUES(2,' KAVYA','F',' 30-DEC-90',' SISTER');
INSERT INTO DEPENDENT VALUES(3,' AJITH,'M',' 24-FEB-87,' BROTHER');
INSERT INTO DEPENDENT VALUES(4,' ARUN','M',' 15-JUN-92,' FATHER');
INSERT INTO DEPENDENT VALUES(5,' MANVITHA','F',' 05-JAN-97',' MOTHER');

SELECT * FROM DEPENDENT;

EMP_ID	DEPENDENT_NAME	GENDER	BDATE	RELATIONSHIP
1	RAMESH	M	01-APR-99	BROTHER
2	KAVYA	F	30-DEC-90	SISTER
3	AJITH	M	24-FEB-87	BROTHER
4	ARUN	M	15-JUN-92	FATHER
5	MANVITHA	F	05-JAN-97'	MOTHER

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

```
Select pno from workson where eid in
(select eid from employee where name='rahul')
union
select pnum  from project where dno in
(select dnum from department where dmgr_id in
(select eid from employee where name='rahul'));
```

PNO
1
2
4
5

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

```
SELECT Eid,name, salary,salary+0.1*salary as updated_salary FROM employee   WHERE
Eid IN(SELECT Eid FROM workson WHERE pno IN(SELECT pnum FROM project
WHERE Pname='IOT'));
```

EID	NAME	SALARY	UPDATED_SALARY
1	RAHUL	35000	38500
2	SAHANA	35000	38500

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(salary), AVG(salary), MAX(salary), MIN(salary)
FROM employee e,department d WHERE d.dnum=e.dno AND dname='Account';
```

SUM(SALARY)	AVG(SALARY)	MAX(SALARY)	MIN(SALARY)
670000	223333.333	600000	35000

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

```
SELECT Eid,name FROM employee e WHERE NOT EXISTS((SELECT pnum FROM
project WHERE dno=5) MINUS (SELECT pno FROM workson w WHERE
w.Eid=e.Eid));
```

EID	NAME
5	Sajaan

5.Create a view Dept_info that gives details of department name, Number of employees and total salary of each department.

```
Create view dept_info(name,count_emp,sum_sal) as Select d.dname, count(*), sum(salary)
From department d inner join employee e ON e.Dno = d.Dnum Group by d.Dname;
```

```
SELECT * FROM DEPT_INFO;
```

NAME	COUNT_EMP	SUM_SAL
ISE	1	35000
ACCOUNT	3	670000
CSE	1	35000

ER DIAGRAM: