DATABASE MANAGEMENT SYSTEMS LAB

LAB MANUAL

PART – A

1. Draw E-R diagram and convert entities and relationships to relation table for a given scenario.

a. COLLEGE DATABASE

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)

b. COMPANY DATABASE

EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO)

DEPARTMENT (DNAME, DNUMBER, MGRSSN, MSRSTARTDATE) DEPT_LOCATIONS (DNUMBER, DLOCATION)

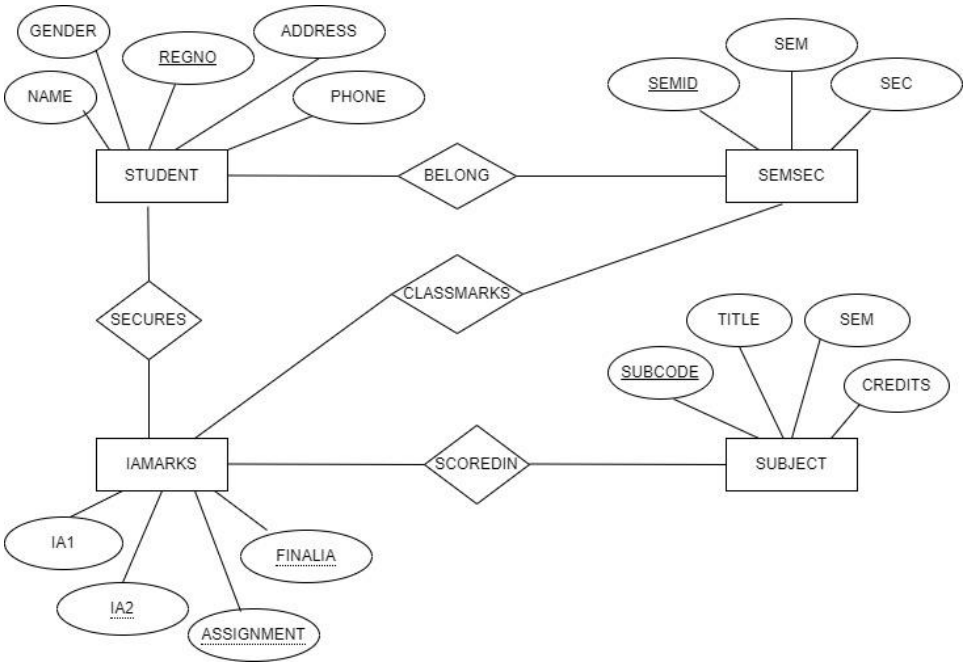
PROJECT (PNAME, PNUMBER, PLOCATION, DNUM)

WORKS_ON (ESSN, PNO, HOURS)

DEPENDENT (ESSN, DEPENDENT_NAME, SEX, BDATE, RELATIONSHIP)

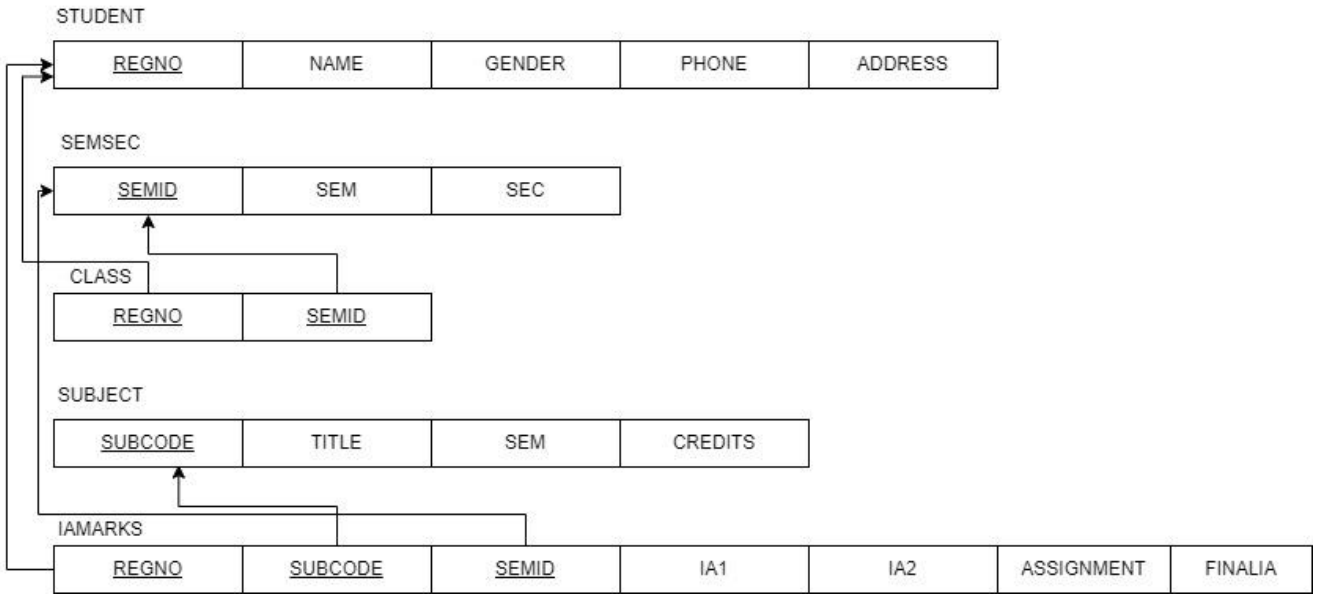
SOLUTION:

E-R Diagram for College Database

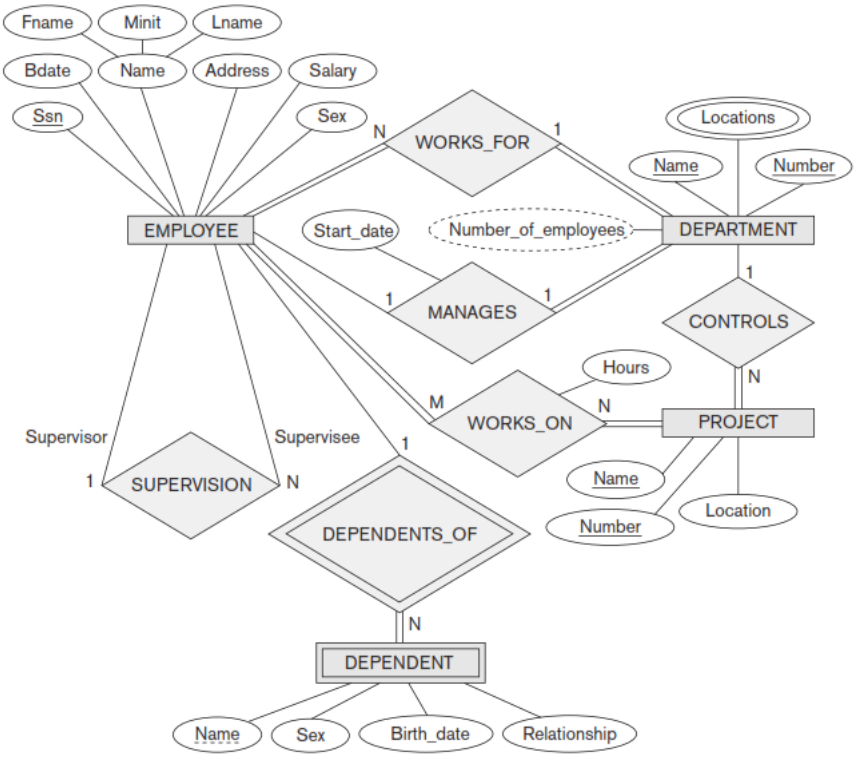


Schema Diagram

Mapping entities and relationships to relation table (Schema Diagram)

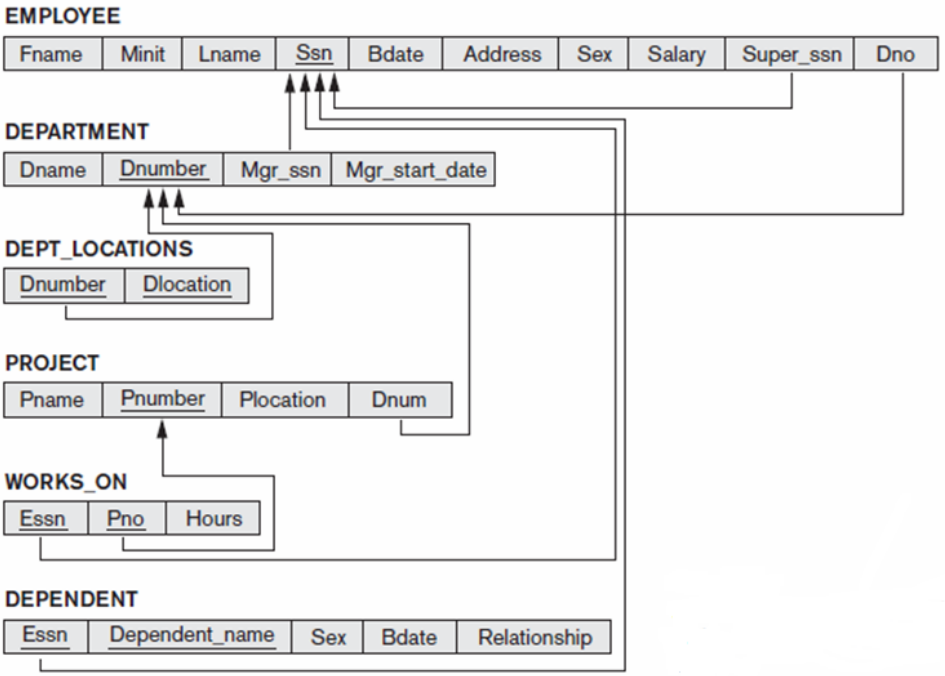


E-R Diagram for Company Database



Schema Diagram

Mapping entities and relationships to relation table (Schema Diagram)



II Consider the Company database with following Schema

EMPLOYEE (SSN, FNAME, MINIT, LNAME, BDATE, ADDRESS, SEX, SALARY, DNO)

DEPARTMENT (DNAME, DNUMBER, MGRSSN, MSRSTARTDATE)

DEPT_LOCATIONS (DNUMBER, DLOCATION)

PROJECT (PNAME, PNUMBER, PLOCATION, DNUM)

WORKS_ON (ESSN, PNO<HOURS)

DEPENDENT (ESSN, DEPENDENT_NAME, SEX, BDATE, RELATIONSHIP)

2. Perform the following:
a. Create company database, Viewing all databases, Viewing all Tables in a Database,
b. Creating Tables (With and Without Constraints)
c. Inserting/Updating/Deleting Records in a Table
d. Saving (Commit) and Undoing (rollback)

Solution:

a. Creating Company Database

```
CREATE DATABASE Company;
```

Viewing all databases

```
SHOW DATABASES;
```

Viewing all Tables in a Database

```
SHOW tables;
```

Creating Tables (With and Without Constraints)

CREATE TABLE DEPARTMENT (DNO VARCHAR2 (20) PRIMARY KEY, DNAME VARCHAR2 (20), MGRSTARTDATE DATE);

CREATE TABLE EMPLOYEE (SSN VARCHAR2 (20) PRIMARY KEY, FNAME VARCHAR2 (20), MINIT VARCHAR2(20), LNAME VARCHAR2 (20), BDATE DATE, ADDRESS VARCHAR2 (20), SEX CHAR (1), SALARY INTEGER, DNO REFERENCES DEPARTMENT (DNO));

CREATE TABLE PROJECT(PNAME VARCHAR2(10), PNUMBER INTEGER PRIMARY KEY, PLOCATION VARCHAR2(20), DNO REFERENCES DEPARTMENT (DNO));

CREATE TABLE WORKS_ON(ESSN REFERENCES EMPLOYEE(SSN), PNO REFERENCES PROJECT (PNUMBER), HOURS DECIMAL(4,1));

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);

c. Inserting/Updating/Deleting Records in a Table

Insert

INSERT INTO DEPARTMENT (DNO, DNAME, MGRSTARTDATE) VALUES (5, 'RESEARCH', '22-MAY-1988', '33445');

INSERT INTO DEPARTMENT VALUES(4, 'ADMINISTRATION', '01-JAN-1995', '98765');

INSERT INTO DEPARTMENT VALUES(1,'HEADQUARTERS', '19-JUN-1981', '88665');

ALTER TABLE DEPARTMENT ADD MGRSSN REFERENCES EMPLOYEE (SSN);

INSERT INTO EMPLOYEE(SSN, FNAME, MINIT, LNAME, BDATE, ADDRESS, SEX, SALARY, DNO)VALUES ('12345','JOHN','B','SMITH','09-JAN-1965', 'BANGALORE', 'M',45000,'5');

INSERT INTO EMPLOYEE VALUES ('88665','JAMES','E','BORG', '10-NOV-1937', 'BANGALORE', 'M', 50000, '1');

INSERT INTO EMPLOYEE VALUES ('33445','FRANKLIN','T','WONG','12-DEC-1960', 'BANGALORE', 'M', 70000, '4');

INSERT INTO EMPLOYEE VALUES ('66884','RAMESH','K','NARAYAN','15-AUG-1970', 'MYSORE', 'M',51000,'4');

INSERT INTO EMPLOYEE VALUES ('98765', 'JENNI','S','WALLACE','02-MAR-972','MANGALORE', 'M', 65000,'5');

INSERT INTO EMPLOYEE VALUES ('98798','ROY','M', 'ANDREW','16-APR-1980','MYSORE', 'M', 40000, '5');

INSERT INTO EMPLOYEE VALUES ('34798','BROY','M', 'REW','6-APR-1981','CHENNAI', 'M', 40000, '2');

INSERT INTO EMPLOYEE VALUES ('99887','ALICIA','N','ZABIA','07-AUG-1966', 'BANGALORE', 'F',80000,'2');

INSERT INTO PROJECT VALUES ('PRODUCTX',1, 'BELL',5);
INSERT INTO PROJECT VALUES ('PRODUCTY',2, 'BELL',4);

INSERT INTO WORKS_ON VALUES(33445,1,38.5);
INSERT INTO WORKS_ON VALUES(12345,2,24.5);
INSERT INTO WORKS_ON VALUES(88665,1,34.5);
INSERT INTO WORKS_ON VALUES(98765,2,20.5);

Update

UPDATE DEPARTMENT SET MGRSSN='98765' WHERE DNO=5;
UPDATE DEPARTMENT SET MGRSSN='33445' WHERE DNO=4;
UPDATE DEPARTMENT SET MGRSSN='88665' WHERE DNO=1;

UPDATE EMPLOYEE SET DNO =‘4‘, SUPER_SSN=33445 WHERE SSN=66884;

Delete

DELETE ENTRIES OF EMPLOYEE TABLE WHERE DNO =1;

DELETE FROM EMPLOYEE WHERE SSN='98798';

d. COMMIT and ROLLBACK

COMMIT;
On execution of this command all changes to the database made by you are made permanent and cannot be undone.

ROLLBACK;

A group of related SQL commands that all have to complete successfully or otherwise be rolled back, is called a transaction.

3. Perform the following:

a. Altering a Table
b. Dropping
c. Truncating
d. Renaming Tables
e. Backing up
f. Restoring a Database.

Solution:

CREATE TABLE DEPT(DEPTNO INTEGER, DNAME VARCHAR(10),LOC VARCHAR(4), PRIMARY KEY(DEPTNO));

DESC DEPT;

a. Altering a table

i. Alter table dept as department

ALTER TABLE DEPT RENAME TO DEPARTMENT1;

DESC DEPARTMENT1;

ii. **Add a new column PINCODE with not null constraints to the existing table DEPT**

ALTER TABLE DEPARTMENT1 ADD(PINCODE NUMBER(6) NOT NULL);

iii. DESC DEPARTMENT1;

a. Rename

Rename the column DNAME to DEPT_NAME in dept table

ALTER TABLE DEPARTMENT1 RENAME COLUMN DNAME TO DEPT_NAME;

b. Drop

DROP TABLE DEPARTMENT1;

c. Truncate

TRUNCATE TABLE DEPTARTMENT1;

A truncate SQL statement is used to remove all rows (complete data) from a table. It is similar to the DELETE statement with no WHERE clause. Truncate table is faster and uses lesser resources than DELETE TABLE command.

4. For a given set of relation schemes, create tables and perform the following
a. Simple Queries,
b. Simple Queries with Aggregate functions
c. Queries with Aggregate functions (group by and having clause).

SOLUTION:

Create table employee

CREATE TABLE EMPLOYEE (SSN VARCHAR2 (20) PRIMARY KEY, FNAME VARCHAR2 (20), MINIT VARCHAR2(20), LNAME VARCHAR2 (20), BDATE DATE, ADDRESS VARCHAR2 (20), SEX CHAR (1), SALARY INTEGER, DNO REFERENCES DEPARTMENT (DNO));

INSERT INTO EMPLOYEE VALUES ('87655', 'JENNI','E','WALL','12-APR-1972','MANGALORE', 'M', 60000,'5');
INSERT INTO EMPLOYEE VALUES ('77988','ROY','G', 'WALL','26-SEP-1980','MYSORE', 'M', 45000, '4');

a. **SELECT EMPLOYEE**

SELECT * FROM EMPLOYEE;

b. Retrieve employee number and their salary

SELECT SSN, SALARY FROM EMPLOYEE;

c. Retrieve average salary of all employee

SELECT AVG(SALARY) FROM EMPLOYEE;

d. Retrieve number of employee

SELECT COUNT(*) FROM EMPLOYEE;

e. Retrieve distinct number of employee

SELECT COUNT(DISTINCT DNO) FROM EMPLOYEE;

f. Retrieve total salary of employee group by employee name and count similar names

SELECT FNAME, SUM(SALARY), COUNT(*) FROM EMPLOYEE GROUP BY(FNAME);

g. Retrieve total salary of employee which is greater than >120000

SELECT FNAME, SUM(SALARY) FROM EMPLOYEE GROUP BY(FNAME) HAVING SUM(SALARY)>120000;

h. Display name of employee in descending order

SELECT FNAME FROM EMPLOYEE ORDER BY FNAME DESC;

5. Execute the following queries

a. How the resulting salaries if every employee working on the 'Research' Departments is given a 10%raise.

b. 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.

SOLUTION:

a. SELECT E.FNAME, E.LNAME, E.SALARY, 1.1*E.SALARY AS INCR_SAL FROM EMPLOYEE E, DEPARTMENT D, EMPLOYEE W WHERE E.SSN=W.SSN AND E.DNO=D.DNO AND D.DNAME='RESEARCH';

b. 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';

6. **Execute the following queries**

a. Retrieve the name of each employee Controlled by Department number 5 (use EXISTS operator).

b. Retrieve the name of each dept and number of employees working in each Department which has at least 2 employees.

SOLUTION:

a. SELECT E.FNAME, E.LNAME FROM EMPLOYEE E WHERE EXISTS(SELECT DNO FROM EMPLOYEE WHERE E.DNO=5);

B. SELECT DNAME AS DEPARTMENT_NAME, COUNT(*) AS NO_OF_EMPLOYEES FROM DEPARTMENT INNER JOIN EMPLOYEE ON EMPLOYEE.DNO = DEPARTMENT.DNO GROUP BY DEPARTMENT.DNO, DNAME HAVING COUNT(*)>2 ORDER BY DNAME;

7. **Execute the following queries**

a. For each project, retrieve the project number, the project name, and the number of employee who work on that project. (use GROUP BY)

b. Retrieve the name of employees who born in the year 1990's

a. SELECT PNUMBER, PNAME, COUNT(*) FROM PROJECT, WORKS_ON WHERE PNUMBER = PNO GROUP BY PNUMBER, PNAME;

b. SELECT E.FNAME, E.LNAME, E.BDATE FROM EMPLOYEE E WHERE EXTRACT(YEAR FROM BDATE)='1990';

8. **For each Department that has more than five employees, retrieve the department number and number of employees who are making salary more than 40000.**

SELECT D.DNO, COUNT(*) FROM DEPARTMENT D, EMPLOYEE E WHERE D.DNO = E.DNO AND SALARY > 40000 GROUP BY D.DNO HAVING COUNT(*) > 5;

9. For each project on which more than two employees work, retrieve the project number, project name and the number of employees who work on that project.

SELECT PNUMBER, Pname, COUNT(*) FROM PROJECT P, WORKS_ON W WHERE PNUMBER=Pno GROUP BY PNUMBER, Pname HAVING COUNT(*) > 2;

10. For a given set of relation tables perform the following:

a. Creating Views (with and without check option),

b. Selecting from a view

c. Dropping views

SOLUTION:

EMPLOYEE TABLE

Creating View

a. Creating Views (With and Without Check Option)

CREATE VIEW SALES_STAFF AS SELECT FNAME, SSN, DNO FROM EMPLOYEE WHERE DNO = 5 WITH CHECK OPTION CONSTRAINT SALES_STAFF_CNST;

b. Selecting from a View

SELECT * FROM SALES_STAFF;

c. **Drop View**

DROP VIEW SALES_STAFF;

PART B

Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries.

BRANCH (BRANCHID, BRANCHNAME, HOD)
STUDENT (USN, NAME, ADDRESS, BRANCHID, SEM)
BOOK (BOOKID, BOOKNAME, AUTHORID, PUBLISHER, BRANCHID)
AUTHOR (AUTHORID, AUTHORNAME, COUNTRY, AGE)
BORROW (USN, BOOKID, BORROWED_DATE)

1. **Perform the following:**
a. Creating a Database, Viewing all databases, Viewing all Tables in a database
b. Creating Tables (With and Without Constraints)
c. Inserting/Updating/Deleting
d. Records in a Table, Saving (Commit) and Undoing (rollback)

SOLUTION:

a. Creating a Database

CREATE DATABASE STUDBOOK ;

Viewing all databases

SHOW DATABASES;

Viewing all Tables in a Database

SHOW TABLES;

b. Creating Tables

CREATE TABLE BRANCH (BRANCHID INT PRIMARY KEY, BNAME VARCHAR(10), HOD VARCHAR(10));

CREATE TABLE STUDENT (USN VARCHAR(10) PRIMARY KEY, NAME VARCHAR(10), ADDR VARCHAR(15), BRANCHID INT REFERENCES BRANCH(BRANCHID),SEM INT);

CREATE TABLE AUTHOR (AUTHORID INT PRIMARY KEY, ANAME VARCHAR(10), COUNTRY VARCHAR(10), AGE INT);

CREATE TABLE BOOK (BOOKID INT PRIMARY KEY, BNAME VARCHAR(20), AUTHORID INT REFERENCES AUTHOR(AUTHORID), PUBLISHER VARCHAR(10), BRANCHID INT REFERENCES BRANCH(BRANCHID));

CREATE TABLE BORROW (USN VARCHAR(10) REFERENCES STUDENT(USN), BOOKID INT REFERENCES BOOK(BOOKID), BORROWDATE DATE);

c. Inserting/Updating/Deleting Records in a Table

Insert

```
INSERT INTO BRANCH (BRANCHID, BNAME, HOD) VALUES (31,'BCA','RAVI');
INSERT INTO BRANCH (BRANCHID, BNAME, HOD) VALUES (32,'MCA','ROHAN');
INSERT INTO BRANCH (BRANCHID, BNAME, HOD) VALUES (33,'MBA','BALAJI');
INSERT INTO BRANCH (BRANCHID, BNAME, HOD) VALUES (34,'BBA','ROHIT');
```

```
INSERT INTO STUDENT(USN, NAME, ADDR, BRANCHID, SEM) VALUES('211BCA', 'SUNIL', 'BANGALORE', 31, 2);
INSERT INTO STUDENT VALUES('20MCA', 'KRISHNA', 'CHENNAI', 32, 1);
INSERT INTO STUDENT VALUES('20MBA', 'KAVITHA', 'MYSORE', 33, 3);
INSERT INTO STUDENT VALUES('18BBA', 'KIRAN', 'DELHI', 34, 6);
```

```
INSERT INTO AUTHOR (AUTHORID,ANAME,COUNTRY,AGE) VALUES(432, 'NAVATHE', 'INDIA', 50);
INSERT INTO AUTHOR VALUES(323, 'RITCHE', 'UK', 45);
INSERT INTO AUTHOR VALUES(254, 'RAM', 'INDIA', 35);
INSERT INTO AUTHOR VALUES(276, 'DENNIS', 'USA', 55);
INSERT INTO AUTHOR VALUES(256, 'RAMKRISHNA', 'INDIA', 42);
```

```
INSERT INTO BOOK( BOOKID,BNAME,AUTHORID,PUBLISHER,BRANCHID) VALUES(1001,'C', 432,'PEARSON',31);
INSERT INTO BOOK VALUES(1002,'DBMS', 323,'MGRAHILL',32);
INSERT INTO BOOK VALUES(1003,'OOAD', 254,'PEARSON',33);
INSERT INTO BOOK VALUES(1004,'UNIX', 432,'PEARSON',34);
```

```
INSERT INTO BORROW(USN, BOOKID,BORROWDATE)VALUES('211BCA', 1001, '10-JAN-2021');
INSERT INTO BORROW VALUES('20MCA',1002,'05-MAR-2021');
INSERT INTO BORROW VALUES('20MBA',1003,'19-MAY-2022');
INSERT INTO BORROW VALUES('20MBA', 1004,'22-FEB-2022');
INSERT INTO BORROW VALUES('20MBA', 1002,'23-FEB-2022');
```

Update

```
UPDATE BOOK SET BNAME='OOAD UML' WHERE BOOKID=1003;
```

Delete

```
DELETE FROM STUDENT WHERE USN='18BBA';
```

d. COMMIT and ROLLBACK

COMMIT;

ROLLBACK;

2. a. List the details of Students who are all studying in 2nd semester BCA.
b. List the students who are not borrowed any books.

SOLUTION:

A. SELECT * FROM STUDENT WHERE SEM=2 AND BRANCHID IN(SELECT BRANCHID FROM BRANCH WHERE BNAME='BCA');

B. SELECT * FROM STUDENT WHERE USN NOT IN (SELECT USN FROM BORROW);

3. a. Display the USN, Student name, Branch_name, Book_name, Author_name, Books_Borrowed_ Date of 2nd sem BCA Students who borrowed books.

b. Display the number of books written by each Author.

SOLUTION:

a. SELECT STUDENT.USN, STUDENT.NAME, BRANCH.BNAME, BOOK.BNAME, AUTHOR.ANAME, BORROW.BORROWDATE FROM STUDENT, BRANCH, BOOK, AUTHOR, BORROW WHERE STUDENT.USN=BORROW.USN AND BORROW.BOOKID=BOOK.BOOKID AND BOOK.AUTHORID =AUTHOR.AUTHORID AND STUDENT.BRANCHID=BRANCH.BRANCHID AND STUDENT.SEM=2 AND BRANCH.BNAME='BCA';

b. SELECT COUNT(*), AUTHORID FROM BOOK GROUP BY AUTHORID;

4. a. Display the student details who borrowed more than two books.
b. Display the student details who borrowed books of more than one Author.

SOLUTION:

a. SELECT * FROM STUDENT WHERE USN IN (SELECT USN FROM BORROW GROUP BY USN HAVING COUNT(USN) >=2);

b. SELECT * FROM STUDENT S WHERE EXISTS (SELECT BR.USN FROM BORROW BR JOIN BOOK BK ON BR.BOOKID = BK.BOOKID WHERE BR.USN=S.USN GROUP BY USN HAVING COUNT(DISTINCT AUTHORID)>1);

5. a. Display the Book names in descending order of their names.
b. List the details of students who borrowed the books which are all published by the same publisher.

SOLUTION:

a. SELECT BNAME FROM BOOK ORDER BY BNAME DESC;

b. SELECT * FROM STUDENT WHERE EXISTS (SELECT USN, PUBLISHER FROM BORROW JOIN BOOK ON BORROW.BOOKID=BOOK.BOOKID WHERE STUDENT.USN=BORROW.USN GROUP BY USN HAVING COUNT(DISTINCT PUBLISHER)=1);

Consider the following schema:

STUDENT1 (USN, name, date_of_birth, branch, mark1, mark2, mark3, total, GPA)

6. Perform the following:
Creating Tables (With and Without Constraints), Inserting/Updating/Deleting Records in a Table, Saving (Commit) and Undoing (rollback)

SOLUTION:

CREATE

CREATE TABLE STUDENT1(USN VARCHAR2(10) PRIMARY KEY, NAME VARCHAR2(20), DOB DATE, BRANCH VARCHAR2(20), MARK1 NUMBER(10), MARK2 NUMBER(10), MARK3 NUMBER(10), TOTAL NUMBER(10), GPA NUMBER(10));

INSERT

INSERT INTO STUDENT1(USN, NAME, DOB, BRANCH, MARK1, MARK2, MARK3) VALUES('21BCA','SUNIL','23-FEB-2001','BCA',67,80,55);

INSERT INTO STUDENT1 (USN, NAME, DOB, BRANCH, MARK1, MARK2, MARK3) VALUES('20BSC', 'KRISHNA','1-JAN-2000', 'B.Sc',78,90,70);

INSERT INTO STUDENT1 (USN, NAME, DOB, BRANCH, MARK1, MARK2, MARK3) VALUES('21BSC', 'KAVITHA', '14-FEB-2001','B.Sc', 55,45,40);

INSERT INTO STUDENT1 (USN, NAME, DOB, BRANCH, MARK1, MARK2, MARK3) VALUES('18BBA', 'KIRAN', '20-OCT-2003','BBA',75,87,90);

UPDATE

UPDATE STUDENT1 SET TOTAL=MARK1+MARK2+MARK3;

UPDATE STUDENT1 SET GPA=TOTAL/3;

NOTE: (4 ROWS WILL BE UPATED AT ONCE)

DELETE

DELETE FROM STUDENT1 WHERE USN='18BBA';

COMMIT and ROLLBACK

COMMIT;

ROLLBACK;

7. Execute the following queries:
a. Find the GPA (Grade Point Average) score of all the students.
b. Find the students who are born on a particular year of birth from the date_of_birth column.

SOLUTION:

a. SELECT SUM(GPA) AS GPA FROM STUDENT1;

b. SELECT * FROM STUDENT1 WHERE DOB='01-JAN-1999' ORDER BY DOB;

8. a. List the students who are studying in a particular branch of study. b. Find the maximum GPA score of the student branch-wise.

SOLUTION:

a. SELECT * FROM STUDENT1 WHERE BRANCH = 'BCA';
b. SELECT MAX(GPA), BRANCH FROM STUDENT1 GROUP BY BRANCH;