PART - A

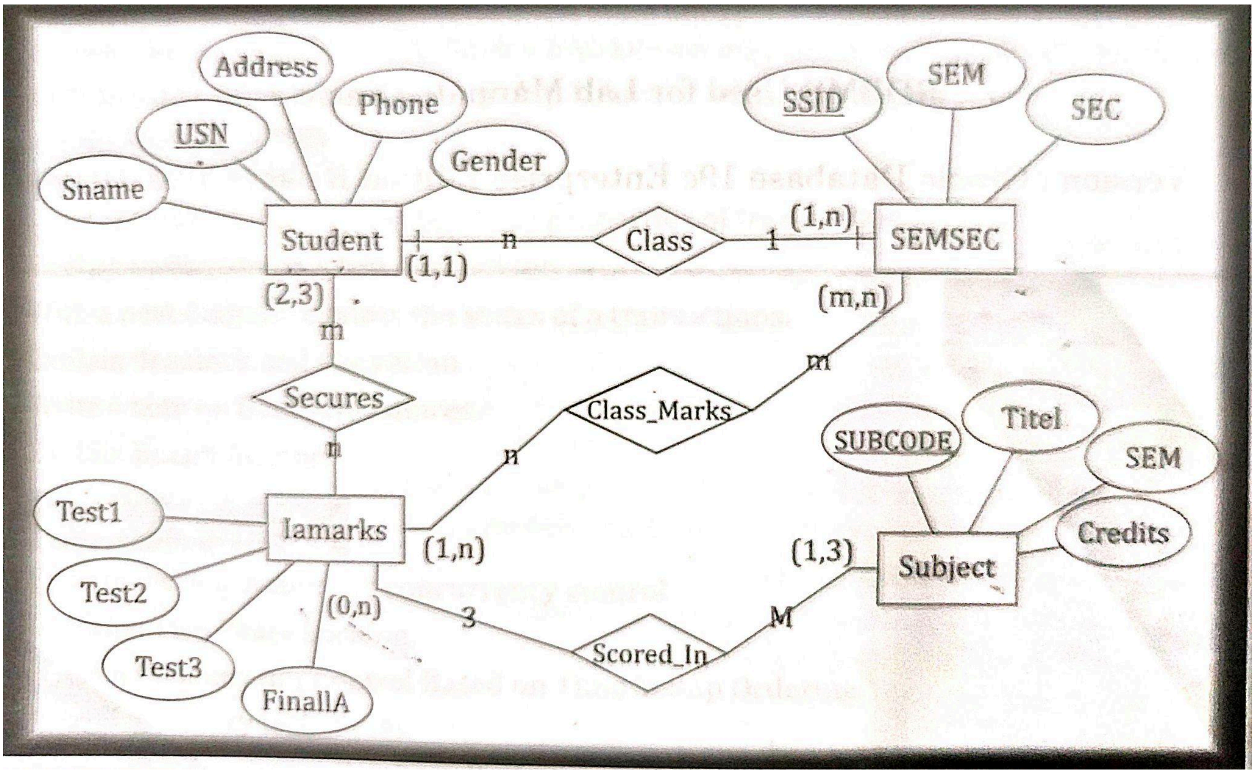
PROGRAM 01: Draw E-R diagram and convert entities and relationships to relation tables for a given scenario.

Solution: Consider College and Company Database

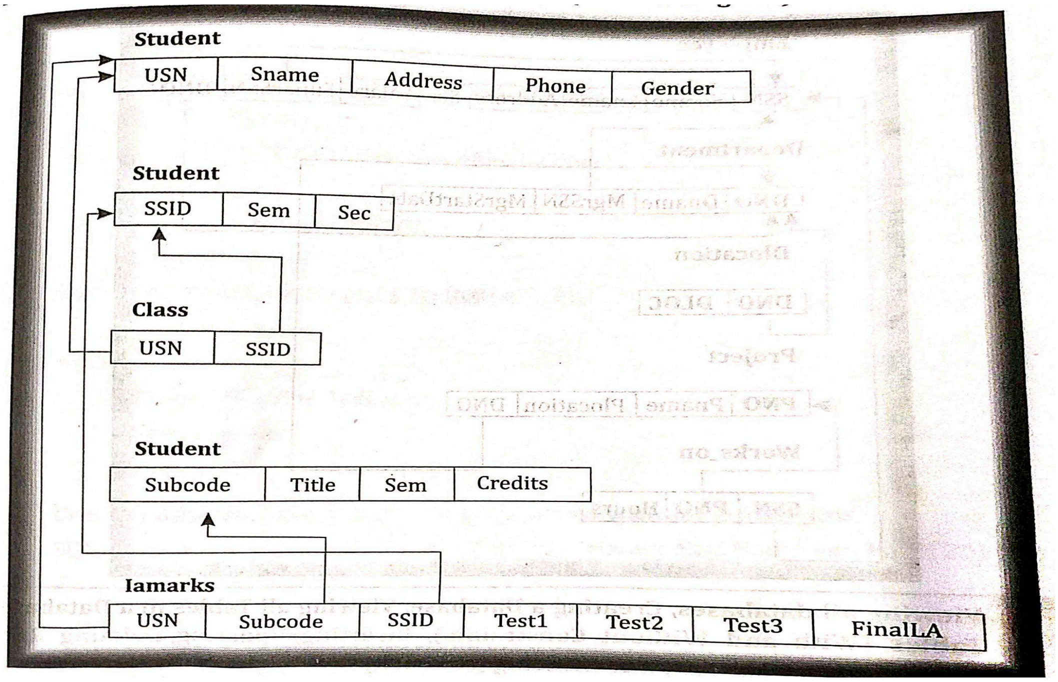
1. **COLLEGE DATABASE:**
   STUDENT (USN, SNAME, ADDRESS, PHONE, GENDER)
   SEMSEC (SSID, SEC)
   CLASS (USN, SSID)
   SUBJECT (SUBCODE, TITLE, SEM, CREDITS)
   IAMARKS(USN, SUBCODE, SSSID, TEST1, TEST2, TEST3, FINALIA)

2. **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)

COLLEGE DATABASE: E-R DIAGRAM



MAPPING ENTITIES AND RELATIONSHIPS TO RELATION TABLE (SCHEMA DIAGRAM)



Student

USN	Sname	Address	Phone	Gender

Student

SSID	Sem	Sec

Class

USN	SSID

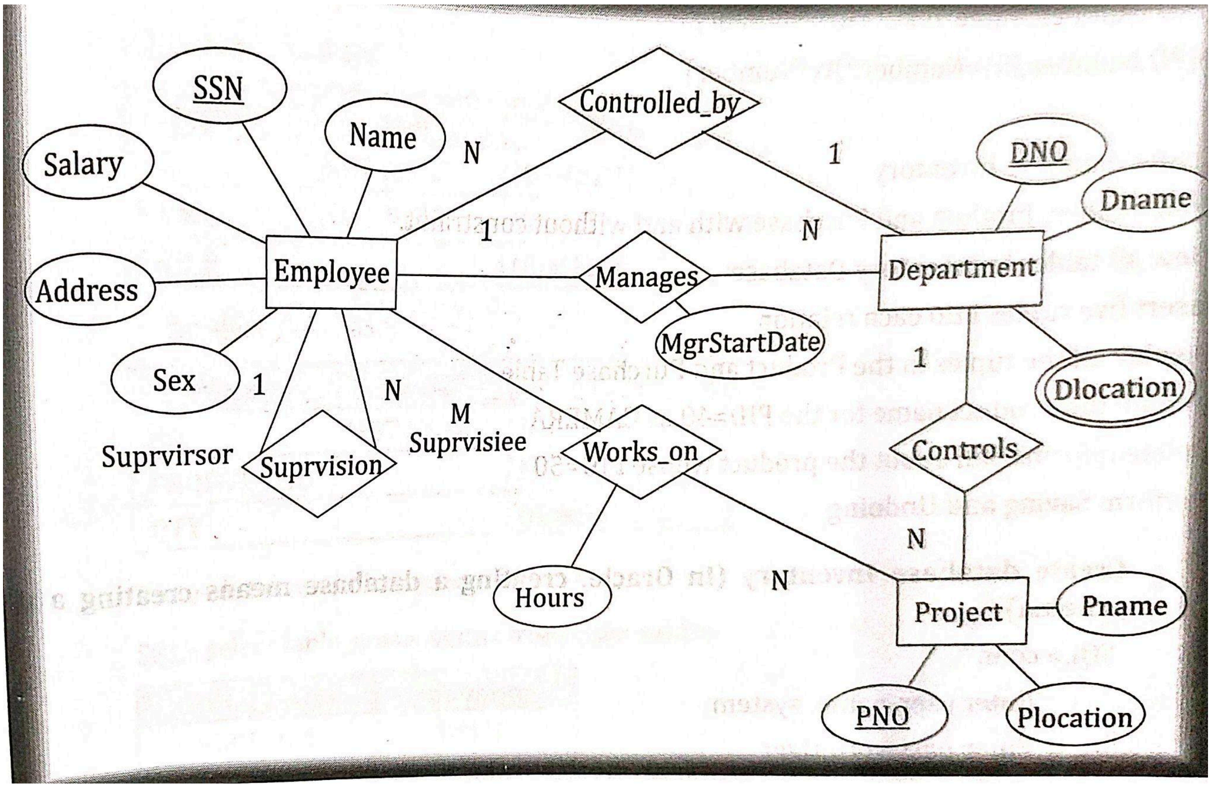
Student

Subcode	Title	Sem	Credits

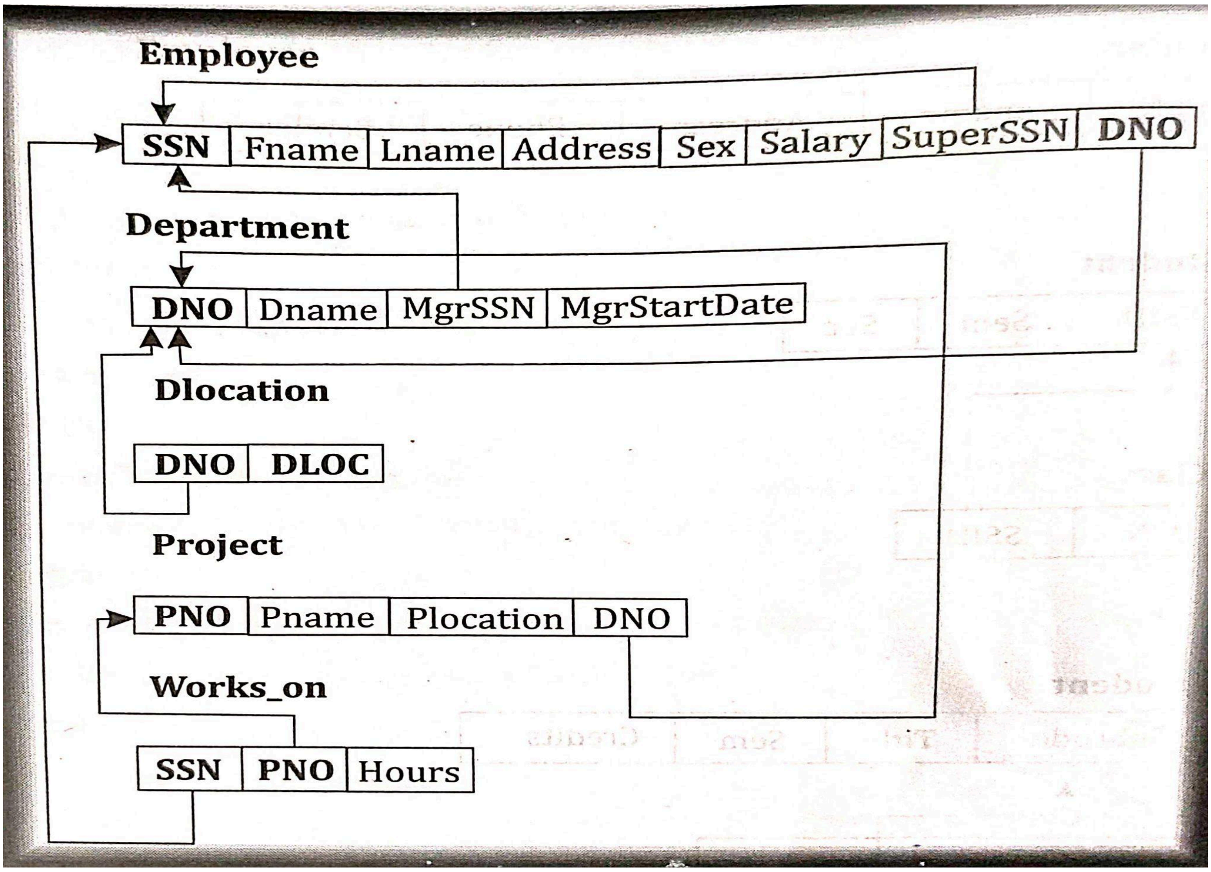
Iamarks

USN	Subcode	SSID	Test1	Test2	Test3	FinalLA

COMPANY DATABASE: E-R DIAGRAM



COMPANY DATABASE: SCHEMA DIAGRAM



PROGRAM 02: Viewing all the Databases, Creating a Database, Viewing all the Tables in a Database, Creating Tables(with and without constraints), Inserting/Updating/Deleting Records in a Table, Saving(Commit) and Undoing(rollback).

STEP-1:

```
SQL> connect
Enter user-name: bca1
Enter password:
Connected.
```

STEP 2: CREATE TABLES PRODUCT AND PURCHASE WITH AND WITHOUT CONSTRAINT.

```
SQL> create table Product(PID Number(10) Primary Key, Name varchar2(20) Not Null, Price Number(8,2));
Table created.

SQL> create table Purchase(PO Number(10) Primary Key, PRODUCT_ID Number(10) references Product(PID), Qty Number(5));

Table created.

SQL> desc Product;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PID                                       NOT NULL NUMBER(10)
 NAME                                      NOT NULL VARCHAR2(20)
 PRICE                                              NUMBER(8,2)

SQL> desc Purchase;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PO                                        NOT NULL NUMBER(10)
 PRODUCT_ID                                         NUMBER(10)
 QTY                                                NUMBER(5)
```

STEP 3: VIEW ALL TABLES IN INVENTORY DATABASE

```
SQL> select table_name, status from User_tables;

TABLE_NAME                     STATUS
------------------------------ --------
PRODUCT                        VALID
PURCHASE                       VALID
```

STEP 4: INSERT FIVE TUPLES INTO EACH RELATION

```
SQL> INSERT INTO product(pid,name,price) VALUES  (10,'PRINTER',20000);
```

```
1 row created.

SQL> INSERT INTO product(pid,name,price) VALUES  (200,'KEYBOARD',20000);
1 row created.

SQL> INSERT INTO product(pid,name,price) VALUES  (30,'MONITOR',15000);
1 row created.

SQL> INSERT INTO product(pid,name,price) VALUES  (40,'TABLE',25000);
1 row created.

SQL> INSERT INTO product(pid,name,price) VALUES  (50,'SCANNER',14000);
1 row created.


SQL> INSERT INTO Purchase(po,product_id,qty) VALUES(101,10,25);
1 row created.

SQL> INSERT INTO Purchase(po,product_id,qty) VALUES(102,40,20);
1 row created.

SQL> INSERT INTO Purchase(po,product_id,qty) VALUES(107,30,40);
1 row created.

SQL> INSERT INTO Purchase(po,product_id,qty) VALUES(104,40,50);
1 row created.

SQL> INSERT INTO Purchase(po,product_id,qty) VALUES(105,40,10);
1 row created.
```

STEP 5: DISPLAY ALL THE TUPLES IN THE PRODUCT AND PURCHASE TABLE

```
SQL> SELECT * FROM Product;

       PID NAME                      PRICE
---------- -------------------- ----------
        10 PRINTER                   20000
       200 KEYBOARD                  20000
        30 MONITOR                   15000
        40 TABLE                     25000
        50 SCANNER                   14000

SQL> SELECT * FROM Purchase;

        PO PRODUCT_ID        QTY
---------- ---------- ----------
```

```
       101        10                  25
       102        40                  20
       107        30                  40
       104        40                 50
       105        40                 10
```

STEP 6: UPDATE THE PRODUCT NAME FOR THE PID=40 AS CAMERA

```
SQL> UPDATE Product SET Name='CAMERA' WHERE PID=40;
1 row updated.

SQL> SELECT * FROM Product;

       PID NAME                 PRICE
---------- -------------------- ----------
        10 PRINTER              20000
       200 KEYBOARD              20000
        30 MONITOR                15000
        40 CAMERA                 25000
        50 SCANNER                14000
```

STEP 7: DELETE INFORMATION ABOUT THE PRODUCT WHOSE PID=50

```
SQL> DELETE FROM PRODUCT WHERE PID=50;

1 row deleted.

SQL> SELECT * FROM Product;

       PID NAME                 PRICE
---------- -------------------- ----------
        10 PRINTER             20000
       200 KEYBOARD             20000
        30 MONITOR               15000
        40 CAMERA                25000
```

STEP 8: PERFORM SAVING AND UNDOING

```
SQL> INSERT INTO product(pid,name,price) VALUES  (50,'MOBILE',35000);

1 row created.

SQL> INSERT INTO product(pid,name,price) VALUES  (60,'LAPTOP',70000);

1 row created.

SQL> COMMIT;

Commit complete.
```

```
SQL> SELECT * FROM PRODUCT;

       PID NAME                      PRICE
---------- -------------------- ----------
        10 PRINTER                   20000
       200 KEYBOARD                  20000
        30 MONITOR                   15000
        40 CAMERA                    25000
        50 MOBILE                    35000
        60 LAPTOP                    70000

6 rows selected.

SQL> SAVEPOINT S1;

Savepoint created.

SQL> INSERT INTO product(pid,name,price) VALUES  (70,'TABLE',50000);

1 row created.

SQL> INSERT INTO product(pid,name,price) VALUES  (80,'CHAIR',25000);

1 row created.

SQL> ROLLBACK TO S1;

Rollback complete.

SQL> SELECT * FROM PRODUCT;

       PID NAME                      PRICE
---------- -------------------- ----------
        10 PRINTER                   20000
       200 KEYBOARD                  20000
        30 MONITOR                   15000
        40 CAMERA                    25000
        50 MOBILE                    35000
        60 LAPTOP                    70000

6 rows selected.
```

PROGRAM 03: Altering a Table, Dropping/Truncating/Renaming a Table, Backing up/Restoring a Database

Steps:

1. **Create Lib Table by properly specifying the constraint.**
2. **Rename Lib as Library**
3. **Add a new column Price with NOT NULL constraints to the existing table Library**
4. **All constraints and views that reference the column are dropped automatically along with column**
5. **Rename the BID to BookId in the library table**
6. **Change the datatype of the column Year_Of_Publication as Text with size 15**
7. **Insert data into Library table**
8. **Truncate table to delete records**
9. **Drop table**

STEP 1: CREATE LIB TABLE BY PROPERLY SPECIFYING THE CONSTRAINT.

```
SQL> CONNECT
Enter user-name: BCA1
Enter password:
Connected.

SQL> CREATE TABLE LIB(
  2  BID VARCHAR(8) PRIMARY KEY,
  3  TITLE VARCHAR(20) NOT NULL,
  4  AUTHOR VARCHAR(20),
  5  PUBLICATION VARCHAR(20),
  6  YEAR_OF_PUBLICATION NUMBER(4)
  7  );
Table created.

SQL> DESC LIB;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BID                                       NOT NULL VARCHAR2(8)
 TITLE                                     NOT NULL VARCHAR2(20)
 AUTHOR                                             VARCHAR2(20)
 PUBLICATION                                        VARCHAR2(20)
 YEAR_OF_PUBLICATION                                NUMBER(4)
```

STEP 2: RENAME LIB AS LIBRARY

```
SQL> ALTER TABLE LIB RENAME TO LIBRARY;
Table altered.

SQL> DESC LIBRARY;
```

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BID                                      NOT NULL VARCHAR2(8)
 TITLE                                     NOT NULL VARCHAR2(20)
 AUTHOR                                             VARCHAR2(20)
 PUBLICATION                                        VARCHAR2(20)
 YEAR_OF_PUBLICATION                                NUMBER(4)
```

STEP 3: ADD A NEW COLUMN PRICE WITH NOT NULL CONSTRAINTS TO THE EXISTING TABLE LIBRARY

```
SQL> ALTER TABLE LIBRARY ADD PRICE NUMBER(8,2) NOT NULL;
Table altered.

SQL> DESC LIBRARY;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BID                                      NOT NULL VARCHAR2(8)
 TITLE                                     NOT NULL VARCHAR2(20)
 AUTHOR                                             VARCHAR2(20)
 PUBLICATION                                        VARCHAR2(20)
 YEAR_OF_PUBLICATION                                NUMBER(4)
 PRICE                                     NOT NULL NUMBER(8,2)
```

STEP 4: ALL THE CONSTRAINTS AND VIEWS THAT REFERENCE THE COLUMN ARE DROPPED AUTOMATICALLY ALONG WITH COLUMN

```
SQL> ALTER TABLE LIBRARY DROP COLUMN AUTHOR CASCADE CONSTRAINTS;
Table altered.

SQL> DESC LIBRARY;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BID                                      NOT NULL VARCHAR2(8)
 TITLE                                     NOT NULL VARCHAR2(20)
 PUBLICATION                                        VARCHAR2(20)
 YEAR_OF_PUBLICATION                                NUMBER(4)
 PRICE                                     NOT NULL NUMBER(8,2)
```

STEP 5: RENAME THE BID TO BOOKID IN THE LIBRARY TABLE

```
SQL> ALTER TABLE LIBRARY RENAME COLUMN BID TO BOOKID;
Table altered.

SQL> DESC LIBRARY;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKID                                    NOT NULL VARCHAR2(8)
 TITLE                          NOT NULL VARCHAR2(20)
 PUBLICATION                             VARCHAR2(20)
 YEAR_OF_PUBLICATION                     NUMBER(4)
 PRICE                          NOT NULL NUMBER(8,2)
```

STEP 6: CHANGE THE DATATYPE OF THE COLUMN YEAR_OF_PUBLICATION AS TEXT WITH SIZE 15

```
SQL> ALTER TABLE LIBRARY MODIFY YEAR_OF_PUBLICATION VARCHAR2(15);
Table altered.

SQL> DESC LIBRARY;
 Name                           Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKID                         NOT NULL VARCHAR2(8)
 TITLE                          NOT NULL VARCHAR2(20)
 PUBLICATION                             VARCHAR2(20)
 YEAR_OF_PUBLICATION                     VARCHAR2(15)
 PRICE                          NOT NULL NUMBER(8,2)
```

STEP 7: INSERT DATA INTO LIBRARY TABLE

```
SQL>INSERT INTO LIBRARY VALUES('SP001','DBMS','SKYWARD PUBLISHERS','2022',300);
1 row created.

SQL> SELECT * FROM LIBRARY;

BOOKID   TITLE                PUBLICATION          YEAR_OF_PUBLICA      PRICE
-------- -------------------- -------------------- --------------- ----------
SP001    DBMS                 SKYWARD PUBLISHERS   2022                   300

SQL> DESC LIBRARY;
 Name                           Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKID                         NOT NULL VARCHAR2(8)
 TITLE                          NOT NULL VARCHAR2(20)
 PUBLICATION                             VARCHAR2(20)
 YEAR_OF_PUBLICATION                     VARCHAR2(15)
 PRICE                          NOT NULL NUMBER(8,2)
```

STEP 8: TRUNCATE TABLE TO DELETE RECORDS

```
SQL> TRUNCATE TABLE LIBRARY;
Table truncated.

SQL> SELECT * FROM LIBRARY;
no rows selected
```

```
SQL> desc library;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKID                                    NOT NULL VARCHAR2(8)
 TITLE                                     NOT NULL VARCHAR2(20)
 PUBLICATION                                        VARCHAR2(20)
 YEAR_OF_PUBLICATION                                VARCHAR2(15)
 PRICE                                     NOT NULL NUMBER(8,2)
```

STEP 9: DROP TABLE

```
SQL> DROP TABLE LIBRARY;

Table dropped.

SQL> desc library;
ERROR:
ORA-04043: object library does not exist
```

PROGRAM 04: For given set of relation schemas, create tables and perform the following Simple Queries, Simple Queries with Aggregate functions, Queries with Aggregate functions(group by and having clause)

Consider the Salary database and execute the following simple queries
SALARYDB(EID:String; NAME:Text;DEPT:String,DOJ:Date;SALARY:Number)

Steps:

1. **Create Table Salary.**
2. **Enter five tuples into the table.**
3. **Display Employee Number and their Salary.**
4. **Find the sum of salaries of all employees.**
5. **Find the sum and average salaries of employees of a particular department.**
6. **Find the number of employees working for each department.**
7. **Display employee information in ascending and descending order of their date of joining.**
8. **Find the highest salary that an Employee draws.**
9. **Find the least salary that an Employee draws.**
10. **Display the details of employee whose name is Rushank and salary is greater than 50000.**

STEP 1: CREATE TABLE SALARY.

```
SQL> CREATE TABLE SALARYDB(
  2  ENO VARCHAR2(8) PRIMARY KEY,
  3  NAME VARCHAR2(15) NOT NULL,
  4  DEPT VARCHAR2(10),
  5  DOJ DATE,
  6  SALARY NUMBER(10,2));

Table created.

SQL> DESC SALARYDB;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ENO                                       NOT NULL VARCHAR2(8)
 NAME                                      NOT NULL VARCHAR2(15)
 DEPT                                               VARCHAR2(10)
 DOJ                                                DATE
 SALARY                                             NUMBER(10,2)
```

STEP 2: ENTER FIVE TUPLES INTO THE TABLE.

```
SQL> INSERT INTO SALARYDB VALUES(&ENO,&NAME,&DEPT,&DOJ,&SALARY);
Enter value for eno: 'SC1010'
Enter value for name: 'AHANA'
Enter value for dept: 'HR'
Enter value for doj: '15-FEB-2010'
Enter value for salary: 60000
old   1: INSERT INTO SALARYDB VALUES(&ENO,&NAME,&DEPT,&DOJ,&SALARY)
new   1: INSERT INTO SALARYDB
VALUES('SC1010','AHANA','HR','15-FEB-2010',60000)

1 row created.

SQL> /
Enter value for eno: 'SC1011'
Enter value for name: 'RAMESH'
Enter value for dept: 'FINANCE'
Enter value for doj: '10-MAR-2012'
Enter value for salary: 45000
old   1: INSERT INTO SALARYDB VALUES(&ENO,&NAME,&DEPT,&DOJ,&SALARY)
new   1: INSERT INTO SALARYDB
VALUES('SC1011','RAMESH','FINANCE','10-MAR-2012',45000)

1 row created.

SQL> /
Enter value for eno: 'SC1013'
Enter value for name: 'NAVEEN'
Enter value for dept: 'MARKETING'
Enter value for doj: '8-JAN-2009'
Enter value for salary: 55000
old   1: INSERT INTO SALARYDB VALUES(&ENO,&NAME,&DEPT,&DOJ,&SALARY)
new   1: INSERT INTO SALARYDB
VALUES('SC1013','NAVEEN','MARKETING','8-JAN-2009',55000)

1 row created.

SQL> /
Enter value for eno: 'SC1014'
Enter value for name: 'ANAGHA'
Enter value for dept: 'HR'
Enter value for doj: '14-APR-2012'
Enter value for salary: 35000
old   1: INSERT INTO SALARYDB VALUES(&ENO,&NAME,&DEPT,&DOJ,&SALARY)
new   1: INSERT INTO SALARYDB
VALUES('SC1014','ANAGHA','HR','14-APR-2012',35000)

1 row created.

SQL> /
Enter value for eno: 'SC1015'
Enter value for name: 'RUSHANK'
Enter value for dept: 'ADMIN'
Enter value for doj: '16-MAY-2011'
Enter value for salary: 55000
old   1: INSERT INTO SALARYDB VALUES(&ENO,&NAME,&DEPT,&DOJ,&SALARY)
new                    1:          INSERT          INTO          SALARYDB
VALUES('SC1015','RUSHANK','ADMIN','16-MAY-2011',55000)

1 row created.

SQL> /
Enter value for eno: 'SC1016'
Enter value for name: 'RUSHANK'
Enter value for dept: 'FINANCE'
Enter value for doj: '4-JUN-2008'
Enter value for salary: 25000
old   1: INSERT INTO SALARYDB VALUES(&ENO,&NAME,&DEPT,&DOJ,&SALARY)
new                    1:          INSERT          INTO          SALARYDB
VALUES('SC1016','RUSHANK','FINANCE','4-JUN-2008',25000)

1 row created.

SQL> SELECT * FROM SALARYDB;

ENO      NAME            DEPT       DOJ           SALARY
-------- --------------- ---------- --------- ----------
SC1010   AHANA           HR         15-FEB-10      60000
SC1011   RAMESH          FINANCE    10-MAR-12      45000
SC1013   NAVEEN          MARKETING  08-JAN-09      55000
SC1014   ANAGHA          HR         14-APR-12      35000
SC1015   RUSHANK         ADMIN      16-MAY-11      55000
SC1016   RUSHANK         FINANCE    04-JUN-08      25000

6 rows selected.
```

STEP 3: DISPLAY EMPLOYEE NUMBER AND THEIR SALARY.

```
SQL> SELECT ENO,SALARY
  2  FROM SALARYDB;

ENO          SALARY
-------- ----------
SC1010        60000
SC1011        45000
SC1013        55000
SC1014        35000
SC1015        55000
```

```
SC1016      25000

6 rows selected.
```

STEP 4: FIND THE SUM OF SALARIES OF ALL EMPLOYEES.

```
SQL> SELECT SUM(SALARY) AS "TOTAL_SALARY" FROM SALARYDB;

TOTAL_SALARY
------------
      275000
```

STEP 5: FIND THE SUM AND AVERAGE SALARIES OF EMPLOYEES OF A PARTICULAR DEPARTMENT.

```
SQL> SELECT DEPT,SUM(SALARY) AS "TOTAL_SALARY", AVG(SALARY) AS "AVERAGE_SALARY"
  2  FROM SALARYDB
  3  GROUP BY DEPT;

DEPT       TOTAL_SALARY AVERAGE_SALARY
---------- ------------ --------------
ADMIN             55000          55000
HR                95000          47500
FINANCE           70000          35000
MARKETING         55000          55000
```

STEP 6: FIND THE NUMBER OF EMPLOYEES WORKING FOR EACH DEPARTMENT.

```
SQL> SELECT DEPT,
  2  COUNT(*) AS "NUMBER_OF_EMPLOYEES"
  3  FROM SALARYDB
  4  GROUP BY DEPT;

DEPT       NUMBER_OF_EMPLOYEES
---------- -------------------
ADMIN                        1
HR                           2
FINANCE                      2
MARKETING                    1
```

STEP 7: DISPLAY EMPLOYEE INFORMATION IN ASCENDING AND DESCENDING ORDER OF THEIR DATE OF JOINING.

```
SQL> SELECT * FROM SALARYDB
  2  ORDER BY DOJ ASC;

ENO     NAME          DEPT      DOJ         SALARY
```

```
-------- --------------- ---------- --------- ----------
SC1016   RUSHANK       FINANCE    04-JUN-08      25000
SC1013   NAVEEN        MARKETING  08-JAN-09      55000
SC1010   AHANA         HR         15-FEB-10      60000
SC1015   RUSHANK       ADMIN      16-MAY-11      55000
SC1011   RAMESH        FINANCE    10-MAR-12      45000
SC1014   ANAGHA        HR         14-APR-12      35000

6 rows selected.
SQL> SELECT * FROM SALARYDB
  2  ORDER BY DOJ DESC;

ENO      NAME            DEPT       DOJ           SALARY
-------- --------------- ---------- --------- ----------
SC1014   ANAGHA        HR         14-APR-12      35000
SC1011   RAMESH        FINANCE    10-MAR-12      45000
SC1015   RUSHANK       ADMIN      16-MAY-11      55000
SC1010   AHANA         HR         15-FEB-10      60000
SC1013   NAVEEN        MARKETING  08-JAN-09      55000
SC1016   RUSHANK       FINANCE    04-JUN-08      25000

6 rows selected.
```

STEP 8: FIND THE HIGHEST SALARY THAT AN EMPLOYEE DRAWS.

```
SQL> SELECT MAX(SALARY) AS "HIGHEST_SALARY"
  2  FROM SALARYDB;

HIGHEST_SALARY
--------------
         60000
```

STEP 9: FIND THE LEAST SALARY THAT AN EMPLOYEE DRAWS.

```
SQL> SELECT MIN(SALARY) AS "LEAST_SALARY"
  2  FROM SALARYDB;

LEAST_SALARY
------------
       25000
```

STEP 10: DISPLAY THE DETAILS OF EMPLOYEE WHOSE NAME IS RUSHANK AND SALARY IS GREATER THAN 50000.

```
SQL> SELECT *
  2  FROM SALARYDB
  3  WHERE NAME='RUSHANK' AND SALARY>50000;

ENO      NAME            DEPT       DOJ           SALARY
-------- --------------- ---------- --------- ----------
SC1015   RUSHANK       ADMIN      16-MAY-11      55000
```

Program 05: Execute the following queries.

a. **How the resulting salaries if every employee working on the 'Research' Departments is given a 10 percent(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 these departments.**

Solution:

Consider the Company database with the following schema.

EMP(ENO:;ENAME:String;EBDATE:Date;ADDRESS:Text;GENDER:Text;SALARY:Number;DEPTNO:Number)
DEPT(DNO:Number;DNAME:String;DLOCATION:String)
PROJECT(PNO:Number;PNAME:string;DNUM:Number)
WORKS_ON(ENO:Number;PNUM:Number;Hours:Number)

STEP 1: CREATING TABLE BY PROPERLY BY SPECIFYING PRIMARY AND FOREIGN KEYS

```
SQL> CONNECT
Enter user-name: bca1
Enter password:
Connected.

SQL> CREATE TABLE DEPT(
  2  DNO NUMBER(4) PRIMARY KEY,
  3  DNAME VARCHAR2(20) NOT NULL,
  4  DLOCATION VARCHAR2(20));

Table created.

SQL> DESC DEPT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DNO                                       NOT NULL NUMBER(4)
 DNAME                                     NOT NULL VARCHAR2(20)
 DLOCATION                                          VARCHAR2(20)

SQL> CREATE TABLE EMP1(
   2   ENO number(6)primary key,
   3   ENAME VARCHAR2(20) NOT NULL,
  4   EBDATE DATE,
  5   ADDRESS VARCHAR2(20),
  6   GENDER CHAR,
  7   SALARY NUMBER(10) NOT NULL,
  8  DEPTNO NUMBER(4) REFERENCES DEPT);

Table created.

SQL> DESC EMP1
 Name                        Null?                    Type
 ----------------------------------------- -------- ----------------------------
 ENO                      NOT NULL                 NUMBER(6)
 ENAME                    NOT NULL                 VARCHAR2(20)
 EBDATE                                            DATE
 ADDRESS                                           VARCHAR2(20)
 GENDER                                            CHAR(1)
 SALARY                   NOT NULL                 NUMBER(10)
 DEPTNO                                            NUMBER(4)

SQL> CREATE TABLE PROJECT(
  2  PNO NUMBER(10) PRIMARY KEY,
  3  PNAME VARCHAR2(20) NOT NULL,
  4  DNUM NUMBER(4) REFERENCES DEPT);

Table created.

SQL> DESC PROJECT;
 Name                      Null?                      Type
 ----------------------------------------- -------- ----------------------------
 PNO                    NOT NULL                   NUMBER(10)
 PNAME                  NOT NULL                   VARCHAR2(20)
 DNUM                                              NUMBER(4)

SQL> CREATE TABLE WORKS_ON(
  2  ENO NUMBER(6) NOT NULL,
  3  PNUM NUMBER(10) REFERENCES PROJECT NOT NULL,
  4  HOURS NUMBER(3,1) NOT NULL,
  5  PRIMARY KEY(ENO,PNUM));

Table created.

SQL> DESC WORKS_ON;
 Name                      Null?                      Type
 ----------------------------------------- -------- ----------------------------
 ENO                   NOT NULL                    NUMBER(6)
 PNUM                  NOT NULL                    NUMBER(10)
 HOURS                 NOT NULL                    NUMBER(3,1)
```

STEP 2: INSERT VALUES IN TO THE TABLES

```
SQL> INSERT INTO DEPT (DNO, DNAME, DLOCATION)
VALUES (2,'ACCOUNTS','JAYANAGAR');

1 row created.

SQL>INSERT INTO DEPT(DNO,DNAME,DLOCATION) VALUES
(4,'RESEARCH','KENGERI');

1 row created.

SQL>INSERT INTO DEPT(DNO, DNAME, DLOCATION) VALUES (5, 'ADMIN' ,
'SOUTHEND');

1 row created.

SQL>INSERT INTO EMP1 (ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1001,'ANIRUDH','14-JAN-1990','BANGALORE','M',45000,4);

1 row created.

SQL>INSERT INTO EMP1(ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1004,'LAKSHMI','4-MAR-1998','MYSORE','F',55000,4);

1 row created.

SQL>INSERT INTO EMP1(ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1002,'SINCHANA','22-DEC-1990','MANGALORE','F',50000,2);

1 row created.

SQL> INSERT INTO EMP1(ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1007,'PRASHANT','26-JAN-1989','DHARWAD','M',20000,4);

1 row created.

SQL> INSERT INTO EMP1(ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1003,'VINAY','26-NOV-1990','HUBLI','M',30000,2);

1 row created.

SQL> INSERT INTO EMP1(ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1005,'VIDYA','26-NOV-1978','HUBLI','F',35000,4);

1 row created.

SQL> INSERT INTO EMP1(ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1006,'PRAJWAL','2-FEB-1974','BANGALORE','M',65000,5);

1 row created.

SQL> INSERT INTO EMP1(ENO, ENAME, EBDATE, ADDRESS, GENDER, SALARY,
DEPTNO) VALUES (1008,'RAJESH','2-FEB-2010','BANGALORE','M',25000,5);

1 row created.

SQL> SELECT*FROM EMP1;

       ENO ENAME      EBDATE    ADDRESS          G    SALARY          DEPTNO
---------- ------------------- --------- ------------------- - ----------------------------------------------
      1001 ANIRUDH        14-JAN-90 BANGALORE          M      45000                  4

      1004 LAKSHMI          04-MAR-98 MYSORE           F      55000                   4

      1002 SINCHANA          22-DEC-90 MANGALORE          F      50000                  2

      1007 PRASHANT          26-JAN-89 DHARWAD           M      20000                 4


      1003 VINAY            26-NOV-90 HUBLI            M      30000                  2

      1005 VIDYA            26-NOV-78 HUBLI            F      35000                  4

      1006 PRAJWAL           02-FEB-74 BANGALORE          M      65000                 5

      1008 RAJESH           02-FEB-10 BANGALORE         M      25000                 2


8 rows selected.

SQL> SELECT*FROM DEPT;

     DNO       DNAME              DLOCATION
---------- ------------------- --------------------
       2          ACCOUNTS          JAYANAGAR
       4          RESEARCH          KENGERI
```

```
         5 ADMIN                SOUTHEND

SQL> INSERT INTO PROJECT(PNO,PNAME,DNUM)VALUES(10,'ERP',5);

1 row created.

SQL> INSERT INTO PROJECT(PNO,PNAME,DNUM)VALUES(20,'BANKING',2);

1 row created.

SQL> INSERT INTO
PROJECT(PNO,PNAME,DNUM)VALUES(30,'CONNECT_TECH',4);

1 row created.

SQL> INSERT INTO PROJECT(PNO,PNAME,DNUM)VALUES(40,'SMART_SEEK',4);

1 row created.

SQL> INSERT INTO PROJECT(PNO,PNAME,DNUM)VALUES(50,'FINANCE',2);

1 row created.

SQL> INSERT INTO PROJECT(PNO,PNAME,DNUM)VALUES(60,'ANALYTICA',4);

1 row created.

SQL> INSERT INTO PROJECT(PNO, PNAME, DNUM)VALUES(70,
'MARKET_RESEARCH', 4);

1 row created.

SQL> INSERT INTO
PROJECT(PNO,PNAME,DNUM)VALUES(80,'SMART_SEARCH',4);

1 row created.

SQL> SELECT * FROM PROJECT;

       PNO PNAME                     DNUM
---------- -------------------- ----------
        10 ERP                          5
        20 BANKING                      2
        30 CONNECT_TECH                 4
```

```
      40 SMART_SEEK                       4
      50 FINANCE                          2
      60 ANALYTICA                        4
      70 MARKET_RESEARCH                  4
      80 SMART_SEARCH                     4

8 rows selected.


SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1001,10,4.5) ;

1 row created.

SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1002,10,6);

1 row created.

SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1008,10,4);

1 row created.

SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1006,20,4);

1 row created.

SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1004,20,8);

1 row created.

SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1005,40,8);

1 row created.

SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1003,50,8);

1 row created.

SQL> INSERT INTO WORKS_ON(ENO,PNUM,HOURS)VALUES(1007,60,5);

1 row created.

SQL> SELECT * FROM WORKS_ON;

   ENO   PNUM     HOURS
```

```
---------- ---------- ----------
      1001         10        4.5
      1002         10          6
      1008         10          4
      1006         20          4
      1004         20          8
      1005         40          8
      1003         50          8
      1007         60          5

8 rows selected.
```

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

```
SQL> SELECT E.ENO,E.ENAME,D.DNAME,1.1*E.SALARY AS "INC_SALARY"
  2  FROM EMP1 E , DEPT D
  3  WHERE E.DEPTNO=D.DNO AND D.DNAME='RESEARCH';

       ENO ENAME                DNAME                INC_SALARY
---------- -------------------- -------------------- ----------
      1001 ANIRUDH              RESEARCH                  49500
      1004 LAKSHMI              RESEARCH                  60500
      1007 PRASHANT             RESEARCH                  22000
      1005 VIDYA                RESEARCH                  38500
```

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 these departments.

```
SQL> SELECT MAX(E.SALARY),MIN(E.SALARY),SUM(E.SALARY),AVG(E.SALARY)
  2  FROM EMP1 E, DEPT D
  3  WHERE E.DEPTNO = D.DNO AND D.DNAME = 'ACCOUNTS';

MAX(E.SALARY) MIN(E.SALARY) SUM(E.SALARY) AVG(E.SALARY)
------------- ------------- ------------- -------------
        50000         30000         80000         40000
```

PROGRAM 06: 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**

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

```
SQL> SELECT E.ENAME FROM EMP1 E
2    WHERE EXISTS(SELECT D.DNO FROM DEPT D WHERE
E.DEPTNO=D.DNO AND E.DEPTNO = 5);

ENAME
--------------------
PRAJWAL
```

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

```
SQL> SELECT D.DNAME,COUNT(*)FROM EMP1 E, DEPT D
  2  WHERE E.DEPTNO = D.DNO
  3  GROUP BY D.DNAME HAVING COUNT(*)>=2;

DNAME                  COUNT(*)
-------------------- --------
ACCOUNTS                  3
RESEARCH                  4
```

PROGRAM 07: EXECUTE THE FOLLOWING QUERIES.

a. **For each project, retrieve the project number, the project name, and the number of employees who work on that project.(use GROUP BY)**
b. **Retrieve the name of employees who born in the year 1990's**

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

```
SQL> SELECT P.PNO,P.PNAME,COUNT(*)AS "NO_OF_EMP1"
  2  FROM PROJECT P, WORKS_ON W
  3  WHERE P.PNO = W.PNUM GROUP BY P.PNO,P.PNAME;

       PNO PNAME                NO_OF_EMP1
---------- -------------------- ----------
        10 ERP                           3
        60 ANALYTICA                     1
        20 BANKING                       2
        50 FINANCE                       1
        40 SMART_SEEK                    1
```

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

```
SQL> SELECT ENAME,EBDATE
  2  FROM EMP1
  3  WHERE EBDATE LIKE '%-%-90';

ENAME                EBDATE
-------------------- ---------
ANIRUDH              14-JAN-90
SINCHANA             22-DEC-90
VINAY                26-NOV-90
```

PROGRAM 08: For each Department that has more than five employees, retrieve the department number and number of employees who are making salaries more than 40000.

SOLUTION:

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

```
SQL> SELECT D.DNAME, D.DNO, COUNT (*) AS "NO_OF_EMP1"
  2  FROM EMP1 E, DEPT D
  3  WHERE E.DEPTNO=D.DNO AND E.SALARY > 40000 AND
  4  D.DNO IN (SELECT DEPTNO FROM EMP1  GROUP BY DEPTNO HAVING
COUNT(*) > = 5)
  5  GROUP BY D.DNO, D.DNAME;

no rows selected
```

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

```
SQL> SELECT D.DNAME, D.DNO,COUNT (*) AS "NO_OF_EMP1"
  2  FROM EMP1 E, DEPT D
  3  WHERE E.DEPTNO=D.DNO AND E.SALARY > 40000 AND
  4  D.DNO IN (SELECT DEPTNO FROM EMP1  GROUP BY DEPTNO HAVING
COUNT(*)>=2)
  5  GROUP BY D.DNO, D.DNAME;

DNAME                       DNO        NO_OF_EMP1
--------------------      ----------   ----------
ACCOUNTS                     2                1
RESEARCH                     4                2
```

PROGRAM 09: 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.

```
SQL> SELECT P.PNO,P.PNAME,COUNT(*) AS "NO_OF_EMP_WORKING"
  2  FROM PROJECT P, WORKS_ON W
  3  WHERE P.PNO = W.PNUM GROUP BY P.PNO,P.PNAME HAVING COUNT(*)>2;

       PNO PNAME                NO_OF_EMP_WORKING
---------- -------------------- -----------------
        10 ERP                                  3
```

PROGRAM 10: For a given set of relation tables perform the following: Creating Views (with and without check option), Dropping views, Selecting from a view

1. **WITHOUT CHECK OPTION:**

STEP 1: CREATE VIEW

```
SQL> CREATE VIEW EMP_DEPT AS(SELECT E.ENO, E.ENAME, E.SALARY,
E.DEPTNO, D.DNAME
  2  FROM EMP1 E, DEPT D WHERE E.DEPTNO=D.DNO);

View created.
```

STEP 2: DISPLAY ALL THE ROWS OF A VIEW

```
SQL> SELECT*FROM EMP_DEPT;

       ENO ENAME                   SALARY     DEPTNO DNAME
---------- ------------------- ---------- ---------- --------------------
      1001 ANIRUDH                  45000          4 RESEARCH
      1004 LAKSHMI                  55000          4 RESEARCH
      1002 SINCHANA                 50000          2 ACCOUNTS
      1007 PRASHANT                 20000          4 RESEARCH
      1003 VINAY                    30000          2 ACCOUNTS
      1005 VIDYA                    35000          4 RESEARCH
      1006 PRAJWAL                  65000          5 ADMIN
      1008 RAJESH                   25000          2 ADMIN

8 rows selected.
```

STEP 3 : INSERT RECORDS IN TO A VIEW

```
SQL> INSERT INTO EMP_DEPT(ENO,ENAME,SALARY,DEPTNO) VALUES
(1009,'SRIKANTH',90000,5);

1 row created.
```

STEP 4: DISPLAY ALL THE ROWS OF A VIEW

```
SQL> SELECT*FROM EMP_DEPT;

       ENO ENAME                   SALARY     DEPTNO DNAME
---------- ------------------- ---------- ---------- --------------------
      1001 ANIRUDH                  45000          4 RESEARCH
      1004 LAKSHMI                  55000          4 RESEARCH
      1002 SINCHANA                50000         2 ACCOUNTS
      1007 PRASHANT                20000         4 RESEARCH
      1003 VINAY              30000         2 ACCOUNTS
      1005 VIDYA              35000         4 RESEARCH
      1006 PRAJWAL                65000         5 ADMIN
      1008 RAJESH              25000         2 ADMIN
      1009 SRIKANTH                90000         5 ADMIN
```

9 rows selected.

STEP 5: DROP VIEW

```
SQL> DROP VIEW EMP_DEPT;
```

View dropped.

2. **WITH CHECK OPTION**

STEP 1: CREATE A VIEW ON EMP TABLE WITH CHECK OPTION OF SALARY LESS THAN 50000 IN WHERE CONDITION

```
SQL> CREATE VIEW EMP_VIEW AS
  2  (SELECT ENO,ENAME,SALARY FROM EMP1
  3  WHERE SALARY <=50000)WITH CHECK OPTION;
```

View created.

STEP 2 : DISPLAY ALL THE ROWS OF A VIEW.

```
SQL> SELECT*FROM EMP_VIEW;

       ENO ENAME                    SALARY
---------- -------------------- ----------
      1001 ANIRUDH                45000
      1002 SINCHANA                50000
      1007 PRASHANT                20000
      1003 VINAY              30000
      1005 VIDYA              35000
      1008 RAJESH              25000
```

6 rows selected.

STEP 3: INSERT A ROW WHERE EMPLOYEE SALARY IS LESS THAN 50000

```
SQL> INSERT INTO EMP_VIEW(ENO,ENAME,SALARY) VALUES
(1011,'SNIGDHA',39000);

1 row created.
```

STEP 4: DISPLAY ALL THE ROWS OF A VIEW

```
SQL> SELECT*FROM EMP_VIEW;

       ENO ENAME                    SALARY
---------- -------------------- ----------
      1001 ANIRUDH                   45000
      1002 SINCHANA                  50000
      1007 PRASHANT                  20000
      1003 VINAY                     30000
      1005 VIDYA                     35000
      1008 RAJESH                    25000
      1011 SNIGDHA                   39000

7 rows selected.
```

STEP 5: INSERT A ROW WHERE EMPLOYEE SALARY IS GREATER THAN 50000. THIS WILL GIVE AN ERROR.

```
SQL> INSERT INTO EMP_VIEW(ENO,ENAME,SALARY) VALUES
(1012,'SMAYAN',99999);
INSERT INTO EMP_VIEW(ENO,ENAME,SALARY) VALUES
(1012,'SMAYAN',99999)
            *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

STEP 6: DROP A VIEW

```
SQL> DROP VIEW EMP_VIEW;

View dropped.
```

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)
STEP 1: CREATE COLLEGE DATABASE

```
SQL> connect collegedb
Enter password:
Connected.
```

STEP 2: CREATE ALL THE TABLES BY PROPERLY SPECIFYING KEY CONSTRAINTS.

```
SQL> CREATE TABLE BRANCH(
  2  BRID NUMBER(5) PRIMARY KEY,
  3  BRNAME VARCHAR2(15) NOT NULL,
  4  HOD VARCHAR2(10));
Table created.
```

SQL> DESC BRANCH;

```
 Name                    Null?           Type
 ----------------------------------------- -------- ----------------------------
 BRID                    NOT NULL        NUMBER(5)
 BRNAME                  NOT NULL        VARCHAR2(15)
 HOD                                     VARCHAR2(10)

SQL> CREATE TABLE STUDENT(
  2  USN VARCHAR2(15) PRIMARY KEY,
  3  NAME VARCHAR2(15) NOT NULL,
  4  ADDRESS VARCHAR2(15),
  5  BRID NUMBER(5) REFERENCES BRANCH,
  6  SEM VARCHAR2(15));

Table created.

SQL> DESC STUDENT;
 Name                    Null?           Type
 ----------------------------------------- -------- ----------------------------
 USN                     NOT NULL        VARCHAR2(15)
 NAME                    NOT NULL        VARCHAR2(15)
 ADDRESS                                 VARCHAR2(15)
 BRID                                    NUMBER(5)
 SEM                                     VARCHAR2(15)

SQL> CREATE TABLE AUTHOR(
  2  AID VARCHAR2(10) PRIMARY KEY,
  3  ANAME VARCHAR2(15) NOT NULL,
  4  COUNTRY VARCHAR2(15),
  5  AGE NUMBER(4));

Table created.

SQL> DESC AUTHOR;
 Name                     Null?                 Type
 ----------------------------------------- -------- ----------------------------
 AID                      NOT NULL              VARCHAR2(10)
 ANAME                    NOT NULL              VARCHAR2(15)
 COUNTRY                                        VARCHAR2(15)
 AGE                                            NUMBER(4)

SQL> CREATE TABLE BOOK(
  2  BKID VARCHAR2(10) PRIMARY KEY,
  3  BKNAME VARCHAR2(10) NOT NULL,
  4  AID VARCHAR2(10) REFERENCES AUTHOR,
  5  PUBLISHER VARCHAR2(20) NOT NULL,
  6  BRID NUMBER(5) REFERENCES BRANCH);

Table created.
SQL> DESC BOOK;
 Name                     Null?                 Type
 ----------------------------------------- -------- ----------------------------
 BKID                     NOT NULL              VARCHAR2(10)
 BKNAME                    NOT NULL             VARCHAR2(15)
 AID                                            VARCHAR2(10)
 PUBLISHER                NOT NULL              VARCHAR2(20)
 BRID                                           NUMBER(5)


SQL> CREATE TABLE BORROW(
  2  USN VARCHAR2(15) REFERENCES STUDENT,
  3  BKID VARCHAR2(10) REFERENCES BOOK,
  4  BORROW_DATE DATE,
  5  PRIMARY KEY(USN,BKID));

Table created.

SQL> DESC BORROW;
 Name                               Null?       Type
 ----------------------------------------- -------- ----------------------------
 USN                               NOT NULL     VARCHAR2(15)
 BKID                               NOT NULL    VARCHAR2(10)
 BORROW_DATE                                    DATE
```

PROGRAM 01:
VIEWING ALL DATABASES, CREATING A DATABASE, VIEWING ALL TABLES IN A DATABASE,CREATING TABLES (WITH AND WITHOUT CONSTRAINTS), INSERTING/UPDATING/DELETING RECORDS IN A TABLE, SAVING (COMMIT) AND UNDOING (ROLLBACK)

STEP 1: VIEW ALL TABLES IN COLLEGE DATABASE

```
SQL> SELECT TABLE_NAME, STATUS  FROM USER_TABLES;

TABLE_NAME                  STATUS
------------------------------ --------
BRANCH                    VALID
STUDENT                   VALID
AUTHOR                    VALID
BOOK                      VALID
BORROW                    VALID
```

STEP 2: INSERTING RECORDS INTO A TABLE

```
SQL> INSERT INTO BRANCH(BRID, BRNAME, HOD)
  2  VALUES(10, 'BCA', 'SANTOSH S');

1 row created.

SQL> INSERT INTO BRANCH VALUES(20, 'BBA', 'RASHMI E');

1 row created.

SQL> INSERT INTO BRANCH VALUES(30, 'BCOM', 'RAMESH A');

1 row created.

SQL> INSERT INTO BRANCH VALUES(40, 'BSC', 'ADITYA P');

1 row created.

SQL> INSERT INTO BRANCH VALUES(50, 'BA', 'ASHA P');

1 row created.

SQL> SELECT * FROM BRANCH;

      BRID BRNAME          HOD
---------- --------------- ----------
        10 BCA             SANTOSH S
        20 BBA             RASHMI E
        30 BCOM            RAMESH A
        40 BSC             ADITYA P
      50          BA               ASHA P

SQL> INSERT INTO STUDENT(USN, NAME, ADDRESS, BRID, SEM)
  2  VALUES('SCAS202201', 'ANURADHA', 'JAYANAGAR', 10, 'II SEM');

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCAS202202', 'MANULA', 'BASAVANGUDI',
10, 'II SEM');

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCAS202203', 'LAXMI', 'BASAVANGUDI',
10, 'IV SEM');

1 row created.

SQL> INSERT INTO STUDENT VALUES(‘SCAC202203’, 'RENUKA',
'HANUMANTHNAGAR', 20, 'II SEM');

1 row created.

SQL> INSERT INTO STUDENT VALUES(' SCAC202204', 'ARUN', 'JPNAGAR', 30, 'II
SEM');

1 row created.

SQL> INSERT INTO STUDENT VALUES(' SCAS202204', 'ABHI', 'GIRINAGAR', 40, 'II
SEM');

1 row created.

SQL> INSERT INTO STUDENT VALUES(' SCAA202201', 'DEEPTI', 'GIRINAGAR', 50,
'IV SEM');

1 row created.

SQL> SELECT * FROM STUDENT;

USN              NAME             ADDRESS               BRID          SEM
--------------- --------------- --------------- ---------- ---------------
SCAS202201     ANURADHA       JAYANAGAR             10            II SEM
SCAS202202     MANULA         BASAVANGUDI           10            II SEM
SCAS202203     LAXMI          BASAVANGUDI           10            IV SEM
SCAC202203     RENUKA         HANUMANTHNAGAR        20            II SEM
SCAC202204     ARUN           JPNAGAR               30            II SEM
SCAS202204     ABHI           GIRINAGAR             40            II SEM
SCAA202201     DEEPTI         GIRINAGAR             50            IV SEM
7 rows selected.
```

```
SQL> INSERT INTO AUTHOR(AID, ANAME, COUNTRY, AGE)
  2  VALUES('NEPCOMP01', 'ARUNA', 'INDIA', 36);

1 row created.

SQL> INSERT INTO AUTHOR VALUES('NEPCOMP02', 'SUMA', 'INDIA', 38);

1 row created.

SQL> INSERT INTO AUTHOR VALUES('NEPCOMM02', 'SANGEETA', 'INDIA', 42);

1 row created.

SQL> INSERT INTO AUTHOR VALUES('NEPCOMM01', 'DILIP', 'INDIA', 39);

1 row created.

SQL> INSERT INTO AUTHOR VALUES('NEPSCI01', 'SHEKAR', 'INDIA', 44);

1 row created.

SQL> SELECT * FROM AUTHOR;

AID        ANAME           COUNTRY                AGE
---------- --------------- --------------- ----------
NEPCOMP01  ARUNA           INDIA                   36
NEPCOMP02  SUMA            INDIA                   38
NEPCOMM02  SANGEETA        INDIA                   42
NEPCOMM01  DILIP           INDIA                   39
NEPSCI01   SHEKAR          INDIA                   44

SQL> INSERT INTO BOOK(BKID, BKNAME, AID, PUBLISHER,BRID) VALUES
('NEPDBMS', 'DBMS', 'NEPCOMP02' ,'SKYWARD',10);

1 row created.

SQL> INSERT INTO BOOK VALUES ('NEPSE', 'SE', 'NEPCOMP02' ,'SKYWARD',10);

1 row created.

SQL> INSERT INTO BOOK VALUES ('NEPJAVA', 'JAVA', 'NEPCOMM01'
,'OXFORD',20);

1 row created.

SQL> INSERT INTO BOOK VALUES ('NEPMATHS', 'MATHS', 'NEPSCI01'
,'OXFORD',30);

1 row created.
```

```
SQL> INSERT INTO BOOK VALUES ('NEPPHY', 'PHYSICS', 'NEPCOMM02' ,'SHREE',40);

1 row created.

SQL> SELECT * FROM BOOK;

BKID       BKNAME     AID                  PUBLISHER       BRID
---------- ---------- ---------- -------------------- ----------
NEPDBMS    DBMS       NEPCOMP02            SKYWARD           10
NEPSE      SE         NEPCOMP02            SKYWARD           10
NEPJAVA    JAVA       NEPCOMM01            OXFORD            20
NEPMATHS   MATHS      NEPSCI01             OXFORD            30
NEPPHY     PHYSICS    NEPCOMM02            SHREE             40

SQL> INSERT INTO BORROW(USN, BKID, BORROW_DATE) VALUES ('SCAS202201', 'NEPDBMS','20-MAY-2022');

1 row created.

SQL> INSERT INTO BORROW VALUES ('SCAS202201', 'NEPSE','28-MAY-2022');

1 row created.

SQL> INSERT INTO BORROW VALUES ('SCAC202204', 'NEPMATHS','28-JUN-2022');

1 row created.

SQL> INSERT INTO BORROW VALUES ('SCAA202201', 'NEPPHY','12-JUN-2022');

1 row created.

SQL> INSERT INTO BORROW VALUES ('SCAS202203', 'NEPPHY','18-JUN-2022');

1 row created.

SQL> INSERT INTO BORROW VALUES ('SCAS202201', 'NEPMATHS','5-JUN-2022');

1 row created.

SQL> INSERT INTO BORROW VALUES ('SCAS202201', 'NEPJAVA','5-JUN-2022');

1 row created.

SQL> SELECT * FROM BORROW;

USN             BKID       BORROW_DATE
--------------- ---------- ---------
SCAS202201      NEPDBMS    20-MAY-22
SCAS202201      NEPSE      28-MAY-22
```

```
SCAC202204     NEPMATHS   28-JUN-22
SCAA202201     NEPPHY     12-JUN-22
SCAS202203     NEPPHY     18-JUN-22
SCAS202201     NEPMATHS   05-JUN-22
SCAS202201     NEPJAVA    05-JUN-22

7 rows selected.
```

STEP 3: UPDATING RECORDS IN A TABLE

```
SQL> UPDATE BOOK
  2  SET PUBLISHER='SKYWARD'
  3  WHERE BKID = 'NEPJAVA';

1 row updated.

SQL> SELECT * FROM BOOK;

BKID       BKNAME     AID        PUBLISHER            BRID
---------- ---------- ---------- -------------------- ----------
NEPDBMS    DBMS       NEPCOMP02  SKYWARD                10
NEPSE      SE         NEPCOMP02  SKYWARD                10
NEPJAVA    JAVA       NEPCOMM01  SKYWARD                20
NEPMATHS   MATHS      NEPSCI01   OXFORD                 30
NEPPHY     PHYSICS    NEPCOMM02  SHREE                  40
```

STEP 4: DELETING RECORDS FROM A TABLE.

```
SQL> DELETE BORROW
  2  WHERE BKID = 'NEPPHY';

2 rows deleted.

SQL> SELECT * FROM BORROW;

USN             BKID       BORROW_DA
--------------- ---------- ---------
SCAS202201      NEPDBMS    20-MAY-22
SCAS202201      NEPSE      28-MAY-22
SCAC202204      NEPMATHS   28-JUN-22
SCAS202201      NEPMATHS   05-JUN-22
SCAS202201      NEPJAVA    05-JUN-22
```

STEP 5: PERFORM SAVING(COMMIT)

```
SQL> INSERT INTO BRANCH(BRID, BRNAME, HOD) VALUES (60, 'MCA', 'BALAJI');

1 row created.

SQL> INSERT INTO BRANCH(BRID, BRNAME, HOD) VALUES (70, 'MBA', 'VENKAT');
```

```
1 row created.

SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM BRANCH;

   BRID BRNAME          HOD
---------- --------------- ----------
     10 BCA             SANTOSH S
     20 BBA             RASHMI E
     30 BCOM            RAMESH A
     40 BSC             ADITYA P
     50 BA              ASHA P
     60 MCA             BALAJI
     70 MBA             VENKAT

7 rows selected.
```

STEP 6: PERFORM UNDOING(ROLL BACK)

```
SQL> SAVEPOINT S1;

Savepoint created.

SQL> INSERT INTO BRANCH(BRID, BRNAME, HOD) VALUES (80, 'MATHS',
'VEENA');

1 row created.

SQL> INSERT INTO BRANCH(BRID, BRNAME, HOD) VALUES (90, 'CHEMISTRY',
'JALAJA');

1 row created.

SQL> ROLL BACK S1;
Rollback complete.
SQL> SELECT * FROM BRANCH;

   BRID     BRNAME      HOD
---------- --------------- ----------
     10     BCA         SANTOSH S
     20     BBA         RASHMI E
     30     BCOM        RAMESH A
     40     BSC         ADITYA P
     50     BA          ASHA P
     60     MCA         BALAJI
     70     MBA         VENKAT

7 rows selected.
```

PROGRAM 02:

A. LIST THE DETAILS OF STUDENTS WHO ARE ALL STUDYING IN 2ND SEM BCA.

B. LIST THE STUDENTS WHO ARE NOT BORROWED ANY BOOKS.

SOLUTION:

A. LIST THE DETAILS OF STUDENTS WHO ARE ALL STUDYING IN 2ND SEM BCA.

```
SQL> SELECT * FROM STUDENT S, BRANCH BR
  2  WHERE S.BRID = BR.BRID
  3  AND S.SEM = 'II SEM' AND BR.BRNAME = 'BCA';

USN             NAME            ADDRESS         BRID    SEM     BRID  BRNAME     HOD
--------------- --------------- --------------- ---------- ------------------------ --------------- ----------
SCAS202201   ANURADHA    JAYANAGAR       10     II SEM      10       BCA         SANTOSH S

SCAS202202   MANULA       BASAVANGUDI   10     II SEM      10       BCA         SANTOSH S
```

B. LIST THE STUDENTS WHO ARE NOT BORROWED ANY BOOKS.

```
SQL> SELECT * FROM STUDENT   S
  2  WHERE S.USN NOT IN ( SELECT B.USN FROM BORROW B);

USN            NAME           ADDRESS             BRID SEM
--------------- --------------- --------------- ---------- ---------------
SCAS202202     MANULA         BASAVANGUDI           10 II SEM
SCAS202203     LAXMI          BASAVANGUDI           10 IV SEM
SCAC202203     RENUKA         HANUMANTHNAGAR        20 II SEM
SCAS202204     ABHI           GIRINAGAR             40 II SEM
SCAA202201     DEEPTI         GIRINAGAR             50 IV SEM
```

PROGRAM 03:

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

```
SQL> SELECT S.USN, S.NAME, S.SEM, BR.BRNAME, BK.BKNAME, A.ANAME,
B.BORROW_DATE
  2  FROM STUDENT  S, BRANCH  BR, BOOK  BK, AUTHOR  A, BORROW  B
  3  WHERE S.BRID = BR.BRID  AND  S.BRID = BK.BRID  AND  A.AID = BK.AID
   4   AND B.USN = S.USN  AND  BK.BKID = B. BKID  AND  S.SEM = 'II SEM'  AND
BR.BRNAME = 'BCA';

USN           NAME          SEM      BRNAME     BKNAME ANAME    BORROW_DA
---------------------- -------------- -------------- ------------------------ ---------
SCAS202201    ANURADHA      II SEM     BCA          DBMS SUMA        20-MAY-22
SCAS202201    ANURADHA      II SEM     BCA          SE SUMA          28-MAY-22
```

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

```
SQL> SELECT A.ANAME,COUNT(DISTINCT BK.BKID) AS "NO_OF_BOOKS"

  2  FROM AUTHOR A,BOOK BK

  3  WHERE A.AID=BK.AID GROUP BY A.ANAME;

ANAME              NO_OF_BOOKS
--------------- -----------
DILIP                   1

SANGEETHA               1

SHEKAR                  1

SUMA                    2
```

PROGRAM 04

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. Display the student details who borrowed more than two books.

```
SQL> SELECT S.NAME
  2  FROM STUDENT S, BORROW B
  3  WHERE S.USN =B.USN
  4  GROUP BY S.NAME
  5  HAVING COUNT(DISTINCT B.BKID)>2;

NAME
---------------
ANURADHA
```

B. Display the student details who borrowed books of more than one Author.

```
SQL> SELECT S.NAME, COUNT(DISTINCT BK.AID)
  2  FROM STUDENT S, BOOK BK, BORROW B
  3  WHERE S.USN = B.USN AND B.BKID = BK.BKID
  4  GROUP BY S.NAME
  5  HAVING COUNT(DISTINCT BK.AID)>1;

NAME            COUNT(DISTINCTBK.AID)
--------------- ---------------------
ANURADHA                            3
```

PROGRAM 05:

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**

SOLUTION:

A. Display the Book names in descending order of their names.

```
SQL> SELECT * FROM BOOK
  2  ORDER BY BKNAME DESC;

BKID         BKNAME      AID       PUBLISHER              BRID
---------- ---------- ---------- -------------------- ----------
NEPSE        SE       NEPCOMP02  SKYWARD              10
NEPPHY       PHYSICS    NEPCOMM02  SHREE              40
NEPMATHS     MATHS      NEPSCI01   OXFORD             30
NEPJAVA      JAVA      NEPCOMM01  SKYWARD             20
NEPDBMS      DBMS       NEPCOMP02  SKYWARD            10
```

B. List the details of students who borrowed the books which are all published by the same

```
SQL> SELECT S.NAME,COUNT(BK.PUBLISHER)
  2  FROM STUDENT S, BOOK BK, BORROW B
  3  WHERE S.USN = B.USN AND B.BKID = BK.BKID
  4  GROUP BY S.NAME
  5  HAVING COUNT(BK.PUBLISHER)>1;

NAME            COUNT(BK.PUBLISHER)
--------------- -------------------
ANURADHA                      4
```

PROGRAM 06:Perform the following:

Creating Tables (With and Without Constraints), Inserting/ Updating/ Deleting Records in a Table, Saving (Commit) and Undoing (rollback)

SOLUTION:
STUDENT(USN,NAME,DATE_OF_BIRTH,BRANCH,MARK1,MARK2,MARK3,TOTAL, GPA)

```
SQL> connect
Enter user-name: bca1
Enter password:
Connected.
```

STEP1: CREATING TABLE(WITH AND WITHOUT CONSTRAINTS)

```
SQL> CREATE TABLE STUDENT(USN VARCHAR2(10) PRIMARY KEY,
  2  NAME VARCHAR2(20) NOT NULL,
  3  DOB DATE,
  4  BRANCH VARCHAR2(10) NOT NULL,
  5  MARK1 NUMBER(4) NOT NULL,
  6  MARK2 NUMBER(4) NOT NULL,
  7  MARK3 NUMBER(4) NOT NULL,
  8  TOTAL NUMBER(4),
  9  GPA NUMBER(4,2));

Table created.

SQL> DESC STUDENT;
 Name                 Null?              Type
 ----------------------------------------- -------- ----------------------------
 USN                  NOT NULL           VARCHAR2(10)
 NAME                 NOT NULL           VARCHAR2(20)
 DOB                                     DATE
 BRANCH               NOT NULL           VARCHAR2(10)
 MARK1                NOT NULL           NUMBER(4)
 MARK2                NOT NULL           NUMBER(4)
 MARK3                NOT NULL           NUMBER(4)
 TOTAL                                   NUMBER(4)
 GPA                                     NUMBER(4,2)
```

STEP2: INSERTING RECORDS INTO A TABLE.

```
SQL>INSERT INTO STUDENT (USN, NAME, DOB, BRANCH, MARK1, MARK2,
MARK3, TOTAL, GPA) VALUES ('SCA202201', 'SANJANA', '24-AUG-2004', 'BCA',
85,96,97,NULL,NULL);

1 row created.
```

```
SQL> INSERT INTO STUDENT VALUES('SCAC202201', 'ANIRUDH', '10-OCT-2004', 'BCOM', 75,85,65,NULL,NULL);

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCAB202201', 'AKASH', '11-NOV-2004', 'BBA', 75,85,83,NULL,NULL);

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCA202202', 'TANDRA', '1-DEC-2004', 'BCA',84,56,63,NULL,NULL);

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCA202203' ,'ANUSHA', '1-JAN-2005', 'BCA',68,72,78,NULL,NULL);

1 row created.

SQL> SELECT * FROM STUDENT;

USN          NAME         DOB          BRANCH     MARK1     MARK2   MARK3    TOTAL     GPA
---------- ---------- ------------------------------------------------------------------------------------------
SCA202201   SANJANA     24-AUG-04   BCA          85          96          97

SCAC202201 ANIRUDH     10-OCT-04   BCOM         75          85          65

SCAB202201 AKASH        11-NOV-04  BBA          75          85          83

SCA202202   TANDRA      01-DEC-04  BCA          84          56          63

SCA202203   ANUSHA      01-JAN-05  BCA          68          72          78
```

STEP 3:INSERTING RECORDS INTO A TABLE AND SAVING (COMMIT).

```
SQL> INSERT INTO STUDENT VALUES('SCAC202203', 'SAHAN', '15-JAN-2005', 'BCOM', 45,65,68,NULL,NULL);

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCAC202204', 'YOHAN', '15-FEB-2004', 'BCOM',55,76,78,NULL,NULL);

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCAC202205', 'SRIKANTH', '05-MAY-1997' ,'BCOM',99,89,79,NULL,NULL);

1 row created.
```

```
SQL> COMMIT;

Commit complete.
```

STEP 4: DELETING RECORDS FROM A TABLE AND SAVING (COMMIT)

```
SQL> DELETE STUDENT
  2  WHERE USN='SCAC202205';

1 row deleted.
```

STEP 5: UNDOING (ROLLBACK).

```
SQL> SAVEPOINT S1;
Savepoint created.

SQL> INSERT INTO STUDENT VALUES('SCAB202204', 'SATYA', '15-MAR-2004',
'BBA',68,77,74,NULL,NULL);

1 row created.

SQL> INSERT INTO STUDENT VALUES('SCAB202205', 'DHANU'', '25-MAR-2004'
,'BBA',68,85,90,NULL,NULL);

1 row created.

USN          NAME         DOB         BRANCH     MARK1     MARK2    MARK3    TOTAL     GPA
---------- ---------- ------------------------------------------------------------------------------------------
SCA202201   SANJANA     24-AUG-04   BCA          85          96          97

SCAC202201 ANIRUDH     10-OCT-04   BCOM         75          85          65

SCAB202201 AKASH        11-NOV-04  BBA          75          85          83

SCA202202    TANDRA      01-DEC-04  BCA          84          56          63

SCA202203    ANUSHA      01-JAN-05  BCA          68          72          78

SCAC202203  SAHAN        15-JAN-05  BCOM         45          65          68

SCAC202204 YOHAN        15-FEB-04  BCOM         55          76          78
7 rows selected.
```

STEP 5: UPDATING RECORDS IN A TABLE.

```
SQL> UPDATE STUDENT SET TOTAL = MARK1 + MARK2 + MARK3;

7 rows updated.

SQL> SELECT * FROM STUDENT;
```

```
USN          NAME         DOB          BRANCH      MARK1     MARK2    MARK3     TOTAL      GPA
---------- ---------- ------------------------------------------------------------------------------------------
SCA202201   SANJANA      24-AUG-04   BCA            85           96            97               278

SCAC202201 ANIRUDH     10-OCT-04    BCOM           75           85            65               225

SCAB202201 AKASH         11-NOV-04  BBA            75           85            83               243

SCA202202    TANDRA        01-DEC-04  BCA            84           56            63               203

SCA202203    ANUSHA        01-JAN-05  BCA            68           72            78               218

SCAC202203  SAHAN          15-JAN-05  BCOM           45           65            68               178

SCAC202204 YOHAN           15-FEB-04  BCOM           55           76            78               209
```

7 rows selected.

PROGRAM 07. Execute the following queries:

A. Find the GPA score of all the students.

B. Find the students who were born on a particular year of birth from the date_of_birth column.

A. Find the GPA score of all the students.

```
SQL> UPDATE  STUDENT SET GPA = ((100 * TOTAL)/300)/10;

7 rows updated.

SQL> SELECT * FROM STUDENT;
USN          NAME          DOB          BRANCH      MARK1     MARK2   MARK3     TOTAL      GPA
---------- ---------- ------------------------------------------------------------------------------------------
SCA202201   SANJANA     24-AUG-04   BCA           85          96          97             278       9.27

SCAC202201 ANIRUDH     10-OCT-04   BCOM          75          85          65             225       7.5

SCAB202201 AKASH        11-NOV-04  BBA           75          85          83             243       8.1

SCA202202    TANDRA       01-DEC-04  BCA          84          56          63             203       6.77

SCA202203    ANUSHA       01-JAN-05  BCA          68          72          78             218       7.27

SCAC202203  SAHAN         15-JAN-05  BCOM         45          65          68             178       5.93

SCAC202204 YOHAN          15-FEB-04  BCOM         55          76          78             209       6.27

7 rows selected.
```

B. Find the students who were born on a particular year of birth from the date_of_birth column.

```
SQL> SELECT USN, NAME, BRANCH, DOB
  2  FROM STUDENT
  3  WHERE DOB LIKE '%-%-04';

USN                  NAME                 BRANCH      DOB
---------- -------------------- ---------- ---------
SCA202201            SANJANA              BCA         24-AUG-04
SCAC202201           ANIRUDH              BCOM        10-OCT-04
SCAB202201           AKASH                BBA         11-NOV-04
SCA202202            TANDRA               BCA         01-DEC-04
SCAC202204           YOHAN                BCOM        15-FEB-04
```

PROGRAM 08. Execute the following queries:

A. List the students who are studying in a particular branch of study.

B. Find the maximum GPA score of the student branch-wise.

A. List the students who are studying in a particular branch of study.

```
SQL> SELECT USN, NAME, BRANCH, DOB
  2  FROM STUDENT
  3  WHERE BRANCH = 'BCA';

USN        NAME                BRANCH     DOB
---------- ------------------- ---------- ---------
SCA202201  SANJANA             BCA        24-AUG-04
SCA202202  TANDRA              BCA        01-DEC-04
SCA202203  ANUSHA              BCA        01-JAN-05
```

B. Find the maximum GPA score of the student branch-wise.

```
SQL> SELECT BRANCH, MAX(GPA)
  2  FROM STUDENT
  3  GROUP BY BRANCH;

BRANCH       MAX(GPA)
---------- ----------
BCA          9.27
BCOM         7.5
BBA          8.1
```