1. **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, Branch\_id, No-of\_Copies*)**

**BOOK\_LENDING (*Book\_id, Branch\_id, Card\_No, Date\_Out, Due\_Date*)**

**LIBRARY\_BRANCH (*Branch\_id, Branch\_Name, Address*)**

**CARD (*Card\_no*)**

**Write SQL queries to**

1. **Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.**
2. **Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017**
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.**

**Table Creation**

CREATE TABLE PUBLISHER

(NAME VARCHAR2 (20) PRIMARY KEY,

PHONE INTEGER,

ADDRESS VARCHAR2 (20));

CREATE TABLE BOOK

(BOOK\_ID INTEGER PRIMARY KEY,

TITLE VARCHAR2 (20),

PUB\_YEAR VARCHAR2 (20),

PUBLISHER\_NAME REFERENCES PUBLISHER (NAME) ON DELETE CASCADE);

CREATE0 TABLE BOOK\_AUTHORS

(AUTHOR\_NAME VARCHAR2 (20),

BOOK\_ID REFERENCES BOOK (BOOK\_ID) ON DELETE CASCADE,

PRIMARY KEY (BOOK\_ID, AUTHOR\_NAME));

CREATE TABLE LIBRARY\_BRANCH

(BRANCH\_ID INTEGER PRIMARY KEY,

BRANCH\_NAME VARCHAR2 (50),

ADDRESS VARCHAR2 (50));

CREATE TABLE BOOK\_COPIES

(NO\_OF\_COPIES INTEGER,

BOOK\_ID REFERENCES BOOK (BOOK\_ID) ON DELETE CASCADE,

BRANCH\_ID REFERENCES LIBRARY\_BRANCH (BRANCH\_ID) ON DELETE CASCADE,

PRIMARY KEY (BOOK\_ID, BRANCH\_ID));

CREATE TABLE CARD

(CARD\_NO INTEGER PRIMARY KEY);

CREATE TABLE BOOK\_LENDING

(DATE\_OUT DATE,

DUE\_DATE DATE,

BOOK\_ID REFERENCES BOOK (BOOK\_ID) ON DELETE CASCADE,

BRANCH\_ID REFERENCES LIBRARY\_BRANCH (BRANCH\_ID) ON DELETE CASCADE,

CARD\_NO REFERENCES CARD (CARD\_NO) ON DELETE CASCADE,

PRIMARY KEY (BOOK\_ID, BRANCH\_ID, CARD\_NO));

**Insertion of Values to Tables**

INSERT INTO PUBLISHER VALUES (‘MCGRAW-HILL’, 9989076587, ‘BANGALORE’);

INSERT INTO PUBLISHER VALUES (‘PEARSON’, 9889076565, ‘NEWDELHI’);

INSERT INTO PUBLISHER VALUES (‘RANDOM HOUSE’, 7455679345, ‘HYDRABAD’);

INSERT INTO PUBLISHER VALUES (‘HACHETTE LIVRE’, 8970862340, ‘CHENAI’);

INSERT INTO PUBLISHER VALUES (‘GRUPO PLANETA’, 7756120238, ‘BANGALORE’);

INSERT INTO BOOK VALUES (1,’DBMS’,’JAN-2017’, ‘MCGRAW-HILL’);

INSERT INTO BOOK VALUES (2,’ADBMS’,’JUN-2016’, ‘MCGRAW-HILL’);

INSERT INTO BOOK VALUES (3,’CN’,’SEP-2016’, ‘PEARSON’);

INSERT INTO BOOK VALUES (4,’CG’,’SEP-2015’, ‘GRUPO PLANETA’);

INSERT INTO BOOK VALUES (5,’OS’,’MAY-2016’, ‘PEARSON’);

INSERT INTO BOOK\_AUTHORS VALUES (’NAVATHE’, 1);

INSERT INTO BOOK\_AUTHORS VALUES (’NAVATHE’, 2);

INSERT INTO BOOK\_AUTHORS VALUES (’TANENBAUM’, 3);

INSERT INTO BOOK\_AUTHORS VALUES (’EDWARD ANGEL’, 4);

INSERT INTO BOOK\_AUTHORS VALUES (’GALVIN’, 5);

INSERT INTO LIBRARY\_BRANCH VALUES (10,’RR NAGAR’,’BANGALORE’);

INSERT INTO LIBRARY\_BRANCH VALUES (11,’RNSIT’,’BANGALORE’);

INSERT INTO LIBRARY\_BRANCH VALUES (12,’RAJAJI NAGAR’, ’BANGALORE’);

INSERT INTO LIBRARY\_BRANCH VALUES (13,’NITTE’,’MANGALORE’);

INSERT INTO LIBRARY\_BRANCH VALUES (14,’MANIPAL’,’UDUPI’);

INSERT INTO BOOK\_COPIES VALUES (10, 1, 10);

INSERT INTO BOOK\_COPIES VALUES (5, 1, 11);

INSERT INTO BOOK\_COPIES VALUES (2, 2, 12);

INSERT INTO BOOK\_COPIES VALUES (5, 2, 13);

INSERT INTO BOOK\_COPIES VALUES (7, 3, 14);

INSERT INTO BOOK\_COPIES VALUES (1, 5, 10);

INSERT INTO BOOK\_COPIES VALUES (3, 4, 11);

INSERT INTO CARD VALUES (100);

INSERT INTO CARD VALUES (101);

INSERT INTO CARD VALUES (102);

INSERT INTO CARD VALUES (103);

INSERT INTO CARD VALUES (104);

INSERT INTO BOOK\_LENDING VALUES (’01-JAN-17’,’01-JUN-17’, 1, 10, 101);

INSERT INTO BOOK\_LENDING VALUES (’11-JAN-17’,’11-MAR-17’, 3, 14, 101);

INSERT INTO BOOK\_LENDING VALUES (’21-FEB-17’,’21-APR-17’, 2, 13, 101);

INSERT INTO BOOK\_LENDING VALUES (’15-MAR-17’,’15-JUL-17’, 4, 11, 101);

INSERT INTO BOOK\_LENDING VALUES (‘12-APR-17’,’12-MAY-17’, 1, 11, 104);

**Queries:**

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

SELECT B.BOOK\_ID, B.TITLE, B.PUBLISHER\_NAME, A.AUTHOR\_NAME, C.NO\_OF\_COPIES, L.BRANCH\_ID

FROM BOOK B, BOOK\_AUTHORS A, BOOK\_COPIES C, LIBRARY\_BRANCH L

WHERE B.BOOK\_ID=A.BOOK\_ID

AND B.BOOK\_ID=C.BOOK\_ID

AND L.BRANCH\_ID=C.BRANCH\_ID;

**2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.**

SELECT CARD\_NO

FROM BOOK\_LENDING

WHERE DATE\_OUT BETWEEN ’01-JAN-2017’ AND ’01-JUL-2017’

GROUP BY CARD\_NO

HAVING COUNT (\*)>3;

**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=3;

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

CREATE VIEW V\_PUBLICATION AS

SELECT PUB\_YEAR

FROM BOOK;

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

CREATE VIEW V\_BOOKS AS

SELECT B.BOOK\_ID, B.TITLE, C.NO\_OF\_COPIES

FROM BOOK B, BOOK\_COPIES C, LIBRARY\_BRANCH L

WHERE B.BOOK\_ID=C.BOOK\_ID

AND C.BRANCH\_ID=L.BRANCH\_ID;

1. **Consider the following schema for Order Database:**

**SALESMAN (*Salesman\_id, Name, City, Commission*)**

**CUSTOMER (*Customer\_id, Cust\_Name, City, Grade, Salesman\_id*)**

**ORDERS (*Ord\_No, Purchase\_Amt, Ord\_Date, Customer\_id, Salesman\_id*)**

**Table Creation**

CREATE TABLE SALESMAN

(SALESMAN\_ID NUMBER (4),

NAME VARCHAR2 (20),

CITY VARCHAR2 (20),

COMMISSION VARCHAR2 (20),

PRIMARY KEY (SALESMAN\_ID));

CREATE TABLE CUSTOMER1

(CUSTOMER\_ID NUMBER (4),

CUST\_NAME VARCHAR2 (20),

CITY VARCHAR2 (20),

GRADE NUMBER (3),

PRIMARY KEY (CUSTOMER\_ID),

SALESMAN\_ID REFERENCES SALESMAN (SALESMAN\_ID) ON DELETE SET NULL);

CREATE TABLE ORDERS

(ORD\_NO NUMBER (5),

PURCHASE\_AMT NUMBER (10, 2),

ORD\_DATE DATE,

PRIMARY KEY (ORD\_NO),

CUSTOMER\_ID REFERENCES CUSTOMER1 (CUSTOMER\_ID) ON DELETE CASCADE,

SALESMAN\_ID REFERENCES SALESMAN (SALESMAN\_ID) ON DELETE CASCADE);

**Insertion of Values to Tables**

INSERT INTO SALESMAN VALUES (1000, ‘JOHN’,’BANGALORE’,’25 %’);

INSERT INTO SALESMAN VALUES (2000, ‘RAVI’,’BANGALORE’,’20 %’);

INSERT INTO SALESMAN VALUES (3000, ‘KUMAR’,’MYSORE’,’15 %’);

INSERT INTO SALESMAN VALUES (4000, ‘SMITH’,’DELHI’,’30 %’);

INSERT INTO SALESMAN VALUES (5000, ‘HARSHA’,’HYDRABAD’,’15 %’);

INSERT INTO CUSTOMER1 VALUES (10, ‘PREETHI’,’BANGALORE’, 100, 1000);

INSERT INTO CUSTOMER1 VALUES (11, ‘VIVEK’,’MANGALORE’, 300, 1000);

INSERT INTO CUSTOMER1 VALUES (12, ‘BHASKAR’,’CHENNAI’, 400, 2000);

INSERT INTO CUSTOMER1 VALUES (13, ‘CHETHAN’,’BANGALORE’, 200, 2000);

INSERT INTO CUSTOMER1 VALUES (14, ‘MAMATHA’,’BANGALORE’, 400, 3000);

INSERT INTO ORDERS VALUES (50, 5000, ‘04-MAY-17’, 10, 1000);

INSERT INTO ORDERS VALUES (51, 450, ‘20-JAN-17’, 10, 2000);

INSERT INTO ORDERS VALUES (52, 1000, ‘24-FEB-17’, 13, 2000);

INSERT INTO ORDERS VALUES (53, 3500, ‘13-APR-17’, 14, 3000);

INSERT INTO ORDERS VALUES (54, 550, ‘09-MAR-17’, 12, 2000);

**Queries:**

1. **Count the customers with grades above Bangalore’s average.**

SELECT GRADE, COUNT (DISTINCT CUSTOMER\_ID)

FROM CUSTOMER1

GROUP BY GRADE

HAVING GRADE > (SELECT AVG(GRADE)

FROM CUSTOMER1

WHERE CITY='BANGALORE');

1. **Find the name and numbers of all salesmen who had more than one customer.**

SELECT SALESMAN\_ID, NAME

FROM SALESMAN A

WHERE 1 < (SELECT COUNT (\*)

FROM CUSTOMER1

WHERE SALESMAN\_ID=A.SALESMAN\_ID);

1. **List all salesmen and indicate those who have and don’t have customers in their cities (Use UNION operation.)**

SELECT SALESMAN.SALESMAN\_ID, NAME, CUST\_NAME, COMMISSION

FROM SALESMAN, CUSTOMER1

WHERE SALESMAN.CITY = CUSTOMER1.CITY

UNION

SELECT SALESMAN\_ID, NAME, 'NO MATCH', COMMISSION

FROM SALESMAN

WHERE NOT CITY = ANY

(SELECT CITY

FROM CUSTOMER1)

ORDER BY 2 DESC;

1. **Create a view that finds the salesman who has the customer with the highest order of a day.**

CREATE VIEW ELITSALESMAN AS

SELECT B.ORD\_DATE, A.SALESMAN\_ID, A.NAME

FROM SALESMAN A, ORDERS B

WHERE A.SALESMAN\_ID = B.SALESMAN\_ID

AND B.PURCHASE\_AMT=(SELECT MAX (PURCHASE\_AMT)

FROM ORDERS C

WHERE C.ORD\_DATE = B.ORD\_DATE);

1. **Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.**

Use ON DELETE CASCADE at the end of foreign key definitions while creating child table orders and then execute the following:

Use ON DELETE SET NULL at the end of foreign key definitions while creating child table customers and then executes the following:

DELETE FROM SALESMAN

WHERE SALESMAN\_ID=1000;

**Program 3**

**Consider the schema for Movie Database:**

**ACTOR (*Act\_id, Act\_Name, Act\_Gender*)**

**DIRECTOR (*Dir\_id, Dir\_Name, Dir\_Phone*)**

**MOVIES (*Mov\_id, Mov\_Title, Mov\_Year, Mov\_Lang, Dir\_id*)**

**MOVIE\_CAST (*Act\_id, Mov\_id, Role*)**

**RATING (*Mov\_id, Rev\_Stars*)**

**Write SQL queries to**

1. **List the titles of all movies directed by ‘Hitchcock’.**
2. **Find the movie names where one or more actors acted in two or more movies.**
3. **List all actors who acted in a movie before 2000 and also in a movie after**

**2015 (use JOIN operation).**

1. **Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.**
2. **Update rating of all movies directed by ‘Steven Spielberg’ to 5.**

**Table Creation**

CREATE TABLE ACTOR (

ACT\_ID NUMBER (3),

ACT\_NAME VARCHAR (20),

ACT\_GENDER CHAR (1),

PRIMARY KEY (ACT\_ID));

CREATE TABLE DIRECTOR (

DIR\_ID NUMBER (3),

DIR\_NAME VARCHAR (20),

DIR\_PHONE NUMBER (10),

PRIMARY KEY (DIR\_ID));

CREATE TABLE MOVIES (

MOV\_ID NUMBER (4),

MOV\_TITLE VARCHAR (25),

MOV\_YEAR NUMBER (4),

MOV\_LANG VARCHAR (12),

DIR\_ID NUMBER (3),

PRIMARY KEY (MOV\_ID),

FOREIGN KEY (DIR\_ID) REFERENCES DIRECTOR (DIR\_ID));

CREATE TABLE MOVIE\_CAST (

ACT\_ID NUMBER (3),

MOV\_ID NUMBER (4),

ROLE VARCHAR (10),

PRIMARY KEY (ACT\_ID, MOV\_ID),

FOREIGN KEY (ACT\_ID) REFERENCES ACTOR (ACT\_ID),

FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

CREATE TABLE RATING (

MOV\_ID NUMBER (4),

REV\_STARS VARCHAR (25),

PRIMARY KEY (MOV\_ID),

FOREIGN KEY (MOV\_ID) REFERENCES MOVIES (MOV\_ID));

**Insertion of Values to Tables**

INSERT INTO ACTOR VALUES (301,’ANUSHKA’,’F’);

INSERT INTO ACTOR VALUES (302,’PRABHAS’,’M’);

INSERT INTO ACTOR VALUES (303,’PUNITH’,’M’);

INSERT INTO ACTOR VALUES (304,’JERMY’,’M’);

INSERT INTO DIRECTOR VALUES (60,’RAJAMOULI’, 8751611001);

INSERT INTO DIRECTOR VALUES (61,’HITCHCOCK’, 7766138911);

INSERT INTO DIRECTOR VALUES (62,’FARAN’, 9986776531);

INSERT INTO DIRECTOR VALUES (63,’STEVEN SPIELBERG’, 8989776530);

INSERT INTO MOVIES VALUES (1001,’BAHUBALI-2’, 2017, ‘TELAGU’, 60);

INSERT INTO MOVIES VALUES (1002,’BAHUBALI-1’, 2015, ‘TELAGU’, 60);

INSERT INTO MOVIES VALUES (1003,’AKASH’, 2008, ‘KANNADA’, 61);

INSERT INTO MOVIES VALUES (1004,’WAR HORSE’, 2011, ‘ENGLISH’, 63);

INSERT INTO MOVIE\_CAST VALUES (301, 1002, ‘HEROINE’);

INSERT INTO MOVIE\_CAST VALUES (301, 1001, ‘HEROINE’);

INSERT INTO MOVIE\_CAST VALUES (303, 1003, ‘HERO’);

INSERT INTO MOVIE\_CAST VALUES (303, 1002, ‘GUEST’);

INSERT INTO MOVIE\_CAST VALUES (304, 1004, ‘HERO’);

INSERT INTO RATING VALUES (1001, 4);

INSERT INTO RATING VALUES (1002, 2);

INSERT INTO RATING VALUES (1003, 5);

INSERT INTO RATING VALUES (1004, 4);

**Queries:**

1. **List the titles of all movies directed by ‘Hitchcock’.**

SELECT MOV\_TITLE

FROM MOVIES

WHERE DIR\_ID IN (SELECT DIR\_ID

FROM DIRECTOR

WHERE DIR\_NAME = ‘HITCHCOCK’);

1. **Find the movie names where one or more actors acted in two or more movies.**

SELECT MOV\_TITLE

FROM MOVIES M, MOVIE\_CAST MV

WHERE M.MOV\_ID=MV.MOV\_ID AND ACT\_ID IN (SELECT ACT\_ID

FROM MOVIE\_CAST GROUP BY ACT\_ID

HAVING COUNT (ACT\_ID)>1)

GROUP BY MOV\_TITLE

HAVING COUNT (\*)>1;

1. **List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).**

SELECT ACT\_NAME, MOV\_TITLE, MOV\_YEAR

FROM ACTOR A

JOIN MOVIE\_CAST C

ON A.ACT\_ID=C.ACT\_ID

JOIN MOVIES M

ON C.MOV\_ID=M.MOV\_ID

WHERE M.MOV\_YEAR NOT BETWEEN 2000 AND 2015;

OR

SELECT A.ACT\_NAME, A.ACT\_NAME, C.MOV\_TITLE, C.MOV\_YEAR

FROM ACTOR A, MOVIE\_CAST B, MOVIES C

WHERE A.ACT\_ID=B.ACT\_ID

AND B.MOV\_ID=C.MOV\_ID

AND C.MOV\_YEAR NOT BETWEEN 2000 AND 2015;

1. **Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.**

SELECT MOV\_TITLE, MAX (REV\_STARS)

FROM MOVIES

INNER JOIN RATING USING (MOV\_ID)

GROUP BY MOV\_TITLE

HAVING MAX (REV\_STARS)>0

ORDER BY MOV\_TITLE;

1. **Update rating of all movies directed by ‘Steven Spielberg’ to 5**

KL

UPDATE RATING

SET REV\_STARS=5

WHERE MOV\_ID IN (SELECT MOV\_ID FROM MOVIES

WHERE DIR\_ID IN (SELECT DIR\_ID

FROM DIRECTOR

WHERE DIR\_NAME = ‘STEVEN SPIELBERG’));

**Program 4**

**Consider the schema for 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*)**

**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 ‘1BI15CS101’ in all subjects.**
4. **Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.**
5. **Categorize students based on the following criterion:**

**If FinalIA = 17 to 20 then CAT = ‘Outstanding’**

**If FinalIA = 12 to 16 then CAT = ‘Average’**

**If FinalIA< 12 then CAT = ‘Weak’**

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

**Table Creation**

CREATE TABLE STUDENT (

USN VARCHAR (10) PRIMARY KEY,

SNAME VARCHAR (25),

ADDRESS VARCHAR (25),

PHONE NUMBER (10),

GENDER CHAR (1));

CREATE TABLE SEMSEC (

SSID VARCHAR (5) PRIMARY KEY,

SEM NUMBER (2),

SEC CHAR (1));

CREATE TABLE CLASS (

USN VARCHAR (10),

SSID VARCHAR (5),

PRIMARY KEY (USN, SSID),

FOREIGN KEY (USN) REFERENCES STUDENT (USN),

FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));

CREATE TABLE SUBJECT (

SUBCODE VARCHAR (8),

TITLE VARCHAR (20),

SEM NUMBER (2),

CREDITS NUMBER (2),

PRIMARY KEY (SUBCODE));

CREATE TABLE IAMARKS (

USN VARCHAR (10),

SUBCODE VARCHAR (8),

SSID VARCHAR (5),

TEST1 NUMBER (2),

TEST2 NUMBER (2),

TEST3 NUMBER (2),

FINALIA NUMBER (2),

PRIMARY KEY (USN, SUBCODE, SSID),

FOREIGN KEY (USN) REFERENCES STUDENT (USN),

FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE),

FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));

**Insertion of values to tables**

INSERT INTO STUDENT VALUES ('1GA13CS020','AKSHAY','BELAGAVI', 8877881122,'M');

INSERT INTO STUDENT VALUES ('1GA13CS062','SANDHYA','BENGALURU', 7722829912,'F');

INSERT INTO STUDENT VALUES ('1GA13CS091','TEESHA','BENGALURU', 7712312312,'F');

INSERT INTO STUDENT VALUES ('1GA13CS066','SUPRIYA','MANGALURU', 8877881122,'F');

INSERT INTO STUDENTVALUES ('1GA14CS010','ABHAY','BENGALURU', 9900211201,'M');

INSERT INTO STUDENT VALUES ('1GA14CS032','BHASKAR','BENGALURU', 9923211099,'M');

INSERT INTO STUDENTVALUES ('1GA14CS025','ASMI','BENGALURU', 7894737377,'F');

INSERT INTO STUDENT VALUES ('1GA15CS011','AJAY','TUMKUR', 9845091341,'M');

INSERT INTO STUDENT VALUES ('1GA15CS029','CHITRA','DAVANGERE', 7696772121,'F');

INSERT INTO STUDENT VALUES ('1GA15CS045','JEEVA','BELLARY', 9944850121,'M');

INSERT INTO STUDENT VALUES ('1GA15CS091','SANTOSH','MANGALURU', 8812332201,'M');

INSERT INTO STUDENT VALUES ('1GA16CS045','ISMAIL','KALBURGI', 9900232201,'M');

INSERT INTO STUDENT VALUES ('1GA16CS088','SAMEERA','SHIMOGA', 9905542212,'F');

INSERT INTO STUDENT VALUES ('1GA16CS122','VINAYAKA','CHIKAMAGALUR', 8800880011,'M');

INSERT INTO SEMSEC VALUES ('ISE8A', 8,'A');

INSERT INTO SEMSEC VALUES (‘ISE8B', 8,'B');

INSERT INTO SEMSEC VALUES (‘ISE8C’, 8,’C’);

INSERT INTO SEMSEC VALUES ('ISE7A', 7,’A’);

INSERT INTO SEMSEC VALUES (‘ISE7B’, 7,'B’);

INSERT INTO SEMSEC VALUES ('ISE7C', 7,'C');

INSERT INTO SEMSEC VALUES (‘ISE6A', 6,'A');

INSERT INTO SEMSEC VALUES (‘ISE6B’, 6,’B’);

INSERT INTO SEMSEC VALUES ('ISE6C’, 6,’C’);

INSERT INTO SEMSEC VALUES (‘ISE5A’, 5,'A’);

INSERT INTO SEMSEC VALUES ('ISE5B', 5,'B');

INSERT INTO SEMSEC VALUES (‘ISE5C', 5,'C');

INSERT INTO SEMSEC VALUES (‘ISE4A’, 4,’A’);

INSERT INTO SEMSEC VALUES ('ISE4B', 4,’B’);

INSERT INTO SEMSEC VALUES (‘ISE4C’, 4,'C’);

INSERT INTO SEMSEC VALUES ('ISE3A', 3,'A');

INSERT INTO SEMSEC VALUES (‘ISE3B', 3,'B');

INSERT INTO SEMSEC VALUES (‘ISE3C’, 3,’C’);

INSERT INTO SEMSEC VALUES ('ISE2A', 2,’A’);

INSERT INTO SEMSEC VALUES (‘ISE2B’, 2,'B’);

INSERT INTO SEMSEC VALUES ('ISE2C', 2,'C');

INSERT INTO SEMSEC VALUES (‘ISE1A', 1,'A');

INSERT INTO SEMSEC VALUES (‘ISE1B’, 1,’B’);

INSERT INTO SEMSEC VALUES ('ISE1C', 1,’C’);

INSERT INTO CLASS VALUES (‘1GA13CS020’,’ISE8A’);

INSERT INTO CLASS VALUES (‘1GA13CS062’,’ISE8A’);

INSERT INTO CLASS VALUES (‘1GA13CS066’,’ISE8B’);

INSERT INTO CLASS VALUES (‘1GA13CS091’,’ISE8C’);

INSERT INTO CLASS VALUES (‘1GA14CS010’,’ISE7A’);

INSERT INTO CLASS VALUES (‘1GA14CS025’,’ISE7A’);

INSERT INTO CLASS VALUES (‘1GA14CS032’,’ISE7A’);

INSERT INTO CLASS VALUES (‘1GA15CS011’,’ISE4A’);

INSERT INTO CLASS VALUES (‘1GA15CS029’,’ISE4A’);

INSERT INTO CLASS VALUES (‘1GA15CS045’,’ISE4B’);

INSERT INTO CLASS VALUES (‘1GA15CS091’,’ISE4C’);

INSERT INTO CLASS VALUES (‘1GA16CS045’,’ISE3A’);

INSERT INTO CLASS VALUES (‘1GA16CS088’,’ISE3B’);

INSERT INTO CLASS VALUES (‘1GA16CS122’,’ISE3C’);

INSERT INTO SUBJECT VALUES ('10CS81','ACA', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS82','SSM', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS83','NM', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS84','CC', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS85','PW', 8, 4);

INSERT INTO SUBJECT VALUES ('10CS71','OOAD', 7, 4);

INSERT INTO SUBJECT VALUES ('10CS72','ECS', 7, 4);

INSERT INTO SUBJECT VALUES ('10CS73','PTW', 7, 4);

INSERT INTO SUBJECT VALUES ('10CS74','DWDM', 7, 4);

INSERT INTO SUBJECT VALUES (‘10CS75','JAVA', 7, 4);

INSERT INTO SUBJECT VALUES ('10CS76','SAN', 7, 4);

INSERT INTO SUBJECT VALUES ('15CS51', 'ME', 5, 4);

INSERT INTO SUBJECT VALUES ('15CS52','CN', 5, 4);

INSERT INTO SUBJECT VALUES ('15CS53','DBMS', 5, 4);

INSERT INTO SUBJECT VALUES ('15CS54','ATC', 5, 4);

INSERT INTO SUBJECT VALUES ('15CS55','JAVA', 5, 3);

INSERT INTO SUBJECT VALUES ('15CS56','AI', 5, 3);

INSERT INTO SUBJECT VALUES ('15CS41','M4', 4, 4);

INSERT INTO SUBJECT VALUES ('15CS42','SE', 4, 4);

INSERT INTO SUBJECT VALUES ('15CS43','DAA', 4, 4);

INSERT INTO SUBJECT VALUES ('15CS44','MPMC', 4, 4);

INSERT INTO SUBJECT VALUES ('15CS45','OOC', 4, 3);

INSERT INTO SUBJECT VALUES ('15CS46','DC', 4, 3);

INSERT INTO SUBJECT VALUES ('15CS31','M3', 3, 4);

INSERT INTO SUBJECT VALUES ('15CS32','ADE', 3, 4);

INSERT INTO SUBJECT VALUES ('15CS33','DSA', 3, 4);

INSERT INTO SUBJECT VALUES ('15CS34','CO', 3, 4);

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

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

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1GA13CS091','10CS81','ISE8C', 15, 16, 18);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1GA13CS091','10CS82','ISE8C', 12, 19, 14);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1GA13CS091','10CS83','ISE8C', 19, 15, 20);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1GA13CS091','10CS84','ISE8C', 20, 16, 19);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3) VALUES ('1GA13CS091','10CS85','ISE8C', 15, 15, 12);

**Queries:**

1. **List all the student details studying in fourth semester ‘C’ section.**

SELECT S.\*, SS.SEM, SS.SEC

FROM STUDENT S, SEMSEC SS, CLASS C

WHERE S.USN = C.USN AND

SS.SSID = C.SSID AND

SS.SEM = 4 AND

SS.SEc=’C’;

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

SELECT SS.SEM, SS.SEC, S.GENDER, COUNT (S.GENDER) AS COUNT

FROM STUDENT S, SEMSEC SS, CLASS C

WHERES.USN = C.USN AND

SS.SSID = C.SSID

GROUP BY SS.SEM, SS.SEC, S.GENDER

ORDER BY SEM;

1. **Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.**

CREATE VIEW STU\_TEST1\_MARKS\_VIEW

AS

SELECT TEST1, SUBCODE

FROM IAMARKS

WHERE USN = '1RN13CS091';

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

CREATE OR REPLACE PROCEDURE AVGMARKS

IS

CURSOR C\_IAMARKS IS

SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B, GREATEST(TEST3,TEST2) AS C

FROM IAMARKS

WHERE FINALIA IS NULL

FOR UPDATE;

C\_A NUMBER;

C\_B NUMBER;

C\_C NUMBER;

C\_SM NUMBER;

C\_AV NUMBER;

BEGIN

OPEN C\_IAMARKS;

LOOP

FETCH C\_IAMARKS INTO C\_A, C\_B, C\_C;

EXIT WHEN C\_IAMARKS%NOTFOUND;

--DBMS\_OUTPUT.PUT\_LINE(C\_A || ' ' || C\_B || ' ' || C\_C);

IF (C\_A != C\_B) THEN

C\_SM:=C\_A+C\_B;

ELSE

C\_SM:=C\_A+C\_C;

END IF;

C\_AV:=C\_SM/2;

--DBMS\_OUTPUT.PUT\_LINE('SUM = '||C\_SM);

--DBMS\_OUTPUT.PUT\_LINE('AVERAGE = '||C\_AV);

UPDATE IAMARKS SET FINALIA=C\_AV WHERE CURRENT OF C\_IAMARKS;

END LOOP;

CLOSE C\_IAMARKS;

END;

/

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

SELECT \* FROM IAMARKS;

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

BEGIN

AVGMARKS;

END;

**5**. **Categorize students based on the following criterion:**

**If FinalIA = 17 to 20 then CAT = ‘Outstanding’**

**If FinalIA = 12 to 16 then CAT = ‘Average’**

**If FinalIA< 12 then CAT = ‘Weak’**

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

SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER,

(CASE

WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING'

WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'

ELSE 'WEAK'

END) AS CAT

FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB

WHERE S.USN = IA.USN AND

SS.SSID = IA.SSID AND

SUB.SUBCODE = IA.SUBCODE AND

SUB.SEM = 8;

|  |
| --- |
|  |

**Program 5**

**Consider the schema for Company Database:**

**EMPLOYEE (*SSN, Name, Address, Sex, Salary, SuperSSN, DNo*)**

**DEPARTMENT (*DNo, DName, MgrSSN, MgrStartDate*)**

**DLOCATION (*DNo,DLoc*)**

**PROJECT (*PNo, PName, PLocation, DNo*)**

**WORKS\_ON (*SSN, PNo, Hours*)**

**Write SQL queries to**

1. **Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.**
2. **Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.**
3. **Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department**
4. **Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator). For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.**

**Table Creation**

CREATE TABLE DEPARTMENT

(DNO VARCHAR2 (20) PRIMARY KEY,

DNAME VARCHAR2 (20),

MGRSTARTDATE DATE);

CREATE TABLE EMPLOYEE

(SSN VARCHAR2 (20) PRIMARY KEY,

FNAME VARCHAR2 (20),

LNAME VARCHAR2 (20),

ADDRESS VARCHAR2 (20),

SEX CHAR (1),

SALARY INTEGER,

SUPERSSN REFERENCES EMPLOYEE (SSN),

DNO REFERENCES DEPARTMENT (DNO));

**NOTE:** Once DEPARTMENT and EMPLOYEE tables are created we must alter department table to add foreign constraint MGRSSN using sql command

ALTER TABLE DEPARTMENT

ADD MGRSSN REFERENCES EMPLOYEE (SSN);

CREATE TABLE DLOCATION

(DLOC VARCHAR2 (20),

DNO REFERENCES DEPARTMENT (DNO),

PRIMARY KEY (DNO, DLOC));

CREATE TABLE PROJECT

(PNO INTEGER PRIMARY KEY,

PNAME VARCHAR2 (20),

PLOCATION VARCHAR2 (20),

DNO REFERENCES DEPARTMENT (DNO));

CREATE TABLE WORKS\_ON

(HOURS NUMBER (2),

SSN REFERENCES EMPLOYEE (SSN),

PNO REFERENCES PROJECT(PNO),

PRIMARY KEY (SSN, PNO));

**Insertion of values to tables**

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASECE01’,’JOHN’,’SCOTT’,’BANGALORE’,’M’, 450000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASCSE01’,’JAMES’,’SMITH’,’BANGALORE’,’M’, 500000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASCSE02’,’HEARN’,’BAKER’,’BANGALORE’,’M’, 700000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASCSE03’,’EDWARD’,’SCOTT’,’MYSORE’,’M’, 500000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASCSE04’,’PAVAN’,’HEGDE’,’MANGALORE’,’M’, 650000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASCSE05’,’GIRISH’,’MALYA’,’MYSORE’,’M’, 450000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASCSE06’,’NEHA’,’SN’,’BANGALORE’,’F’, 800000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASACC01’,’AHANA’,’K’,’MANGALORE’,’F’, 350000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASACC02’,’SANTHOSH’,’KUMAR’,’MANGALORE’,’M’, 300000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASISE01’,’VEENA’,’M’,’MYSORE’,’M’, 600000);

INSERT INTO EMPLOYEE (SSN, FNAME, LNAME, ADDRESS, SEX, SALARY) VALUES (‘GASIT01’,’NAGESH’,’HR’,’BANGALORE’,’M’, 500000);

INSERT INTO DEPARTMENT VALUES (‘1’,’ACCOUNTS’,’01-JAN-01’,’GATACC02’);

INSERT INTO DEPARTMENT VALUES (‘2’,’IT’,’01-AUG-16’,’GATIT01’);

INSERT INTO DEPARTMENT VALUES (‘3’,’ECE’,’01-JUN-08’,’GATECE01’);

INSERT INTO DEPARTMENT VALUES (‘4’,’ISE’,’01-AUG-15’,’GATISE01’);

INSERT INTO DEPARTMENT VALUES (‘5’,’CSE’,’01-JUN-02’,’GATCSE05’);

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

UPDATE EMPLOYEE SET

SUPERSSN=NULL, DNO=’3’

WHERE SSN=’GATECE01’;

UPDATE EMPLOYEE SET

SUPERSSN=’GATCSE02’, DNO=’5’

WHERE SSN=’GATCSE01’;

UPDATE EMPLOYEE SET

SUPERSSN=’GATCSE03’, DNO=’5’

WHERE SSN=’GATCSE02’;

UPDATE EMPLOYEE SET

SUPERSSN=’GATCSE04’, DNO=’5’

WHERE SSN=’GATCSE03’;

UPDATE EMPLOYEE SET

DNO=’5’, SUPERSSN=’GATCSE05’

WHERE SSN=’GATCSE04’;

UPDATE EMPLOYEE SET

DNO=’5’, SUPERSSN=’GATCSE06’

WHERE SSN=’GATCSE05’;

UPDATE EMPLOYEE SET

DNO=’5’, SUPERSSN=NULL

WHERE SSN=’GATCSE06’;

UPDATE EMPLOYEE SET

DNO=’1’, SUPERSSN=’RNSACC02’

WHERE SSN=’GATACC01’;

UPDATE EMPLOYEE SET

DNO=’1’, SUPERSSN=NULL

WHERE SSN=’GATACC02’;

UPDATE EMPLOYEE SET

DNO=’4’, SUPERSSN=NULL

WHERE SSN=’GATISE01’;

UPDATE EMPLOYEE SET

DNO=’2’, SUPERSSN=NULL

WHERE SSN=’GATIT01’;

INSERT INTO DLOCATION VALUES (’BANGALORE’, ‘1’);

INSERT INTO DLOCATION VALUES (’BANGALORE’, ‘2’);

INSERT INTO DLOCATION VALUES (’BANGALORE’, ‘3’);

INSERT INTO DLOCATION VALUES (’MANGALORE’, ‘4’);

INSERT INTO DLOCATION VALUES (’MANGALORE’, ‘5’);

INSERT INTO PROJECT VALUES (100,’IOT’,’BANGALORE’,’5’);

INSERT INTO PROJECT VALUES (101,’CLOUD’,’BANGALORE’,’5’);

INSERT INTO PROJECT VALUES (102,’BIGDATA’,’BANGALORE’,’5’);

INSERT INTO PROJECT VALUES (103,’SENSORS’,’BANGALORE’,’3’);

INSERT INTO PROJECT VALUES (104,’BANK MANAGEMENT’,’BANGALORE’,’1’);

INSERT INTO PROJECT VALUES (105,’SALARY MANAGEMENT’,’BANGALORE’,’1’);

INSERT INTO PROJECT VALUES (106,’OPENSTACK’,’BANGALORE’,’4’);

INSERT INTO PROJECT VALUES (107,’SMART CITY’,’BANGALORE’,’2’);

INSERT INTO WORKS\_ON VALUES (4, ‘GATCSE01’, 100);

INSERT INTO WORKS\_ON VALUES (6, ‘GATCSE01’, 101);

INSERT INTO WORKS\_ON VALUES (8, ‘GATCSE01’, 102);

INSERT INTO WORKS\_ON VALUES (10, ‘GATCSE02’, 100);

INSERT INTO WORKS\_ON VALUES (3, ‘GATCSE04’, 100);

INSERT INTO WORKS\_ON VALUES (4, ‘GATCSE05’, 101);

INSERT INTO WORKS\_ON VALUES (5, ‘GATCSE06’, 102);

INSERT INTO WORKS\_ON VALUES (6, ‘GATCSE03’, 102);

INSERT INTO WORKS\_ON VALUES (7, ‘GATECE01’, 103);

INSERT INTO WORKS\_ON VALUES (5, ‘GATACC01’, 104);

INSERT INTO WORKS\_ON VALUES (6, ‘GATACC02’, 105);

INSERT INTO WORKS\_ON VALUES (4, ‘GATISE01’, 106);

INSERT INTO WORKS\_ON VALUES (10, ‘GATIT01’, 107);

**Queries:**

1. **Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.**

(SELECT DISTINCT P.PNO

FROM PROJECT P, DEPARTMENT D, EMPLOYEE E

WHERE E.DNO=D.DNO

AND D.MGRSSN=E.SSN

AND E.LNAME=’SCOTT’)

UNION

(SELECT DISTINCT P1.PNO

FROM PROJECT P1, WORKS\_ON W, EMPLOYEE E1

WHERE P1.PNO=W.PNO

AND E1.SSN=W.SSN

AND E1.LNAME=’SCOTT’);

1. **Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.**

SELECT E.FNAME, E.LNAME, 1.1\*E.SALARY AS INCR\_SAL

FROM EMPLOYEE E, WORKS\_ON W, PROJECT P

WHERE E.SSN=W.SSN

AND W.PNO=P.PNO

AND P.PNAME=’IOT’;

1. **Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department**

SELECT SUM (E.SALARY), MAX (E.SALARY), MIN (E.SALARY), AVG (E.SALARY)

FROM EMPLOYEE E, DEPARTMENT D

WHERE E.DNO=D.DNO

AND D.DNAME=’ACCOUNTS’;

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

SELECT E.FNAME, E.LNAME

FROM EMPLOYEE E

WHERE NOT EXISTS((SELECT PNO

FROM PROJECT

WHERE DNO=’5’)

MINUS (SELECT PNO

FROM WORKS\_ON

WHERE E.SSN=SSN));

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

SELECT D.DNO, COUNT (\*)

FROM DEPARTMENT D, EMPLOYEE E

WHERE D.DNO=E.DNO

AND E.SALARY>600000

AND D.DNO IN (SELECT E1.DNO

FROM EMPLOYEE E1

GROUP BY E1.DNO

HAVING COUNT (\*)>5)

GROUP BY D.DNO;