ASSIGNMENT

Sub: RDBMS Lab; Code MCAN192

Session: 2022-23, Odd Semester

Implicit cursor:

1)

a) Create the table : Student_fees(Roll,Name,Total_marks,CourseFees)

SOL:

>>>create table student_fees(roll number(3) primary key, name varchar2(20), total_marks number(7,1), coursefees number(9,2));

SCHEMA:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STUDENT_FEES	ROLL	Number	-	3	0	1	-	-	-
	NAME	Varchar2	20	-	-	-	☐	-	-
	TOTAL_MARKS	Number	-	7	1	-	☐	-	-
	COURSEFEES	Number	-	9	2	-	☐	-	-

b) Populate the table with at least five records.

SOL:

>>>insert into student_fees values(1, 'Rajendra Kumar Shaw', 957, 15000);

>>>insert into student_fees values(2, 'Rahul Kumar Shaw', 750, 15000);

>>>insert into student_fees values(3, 'Harry Potter', 988, 15000);

>>>insert into student_fees values(4, 'Atul Kumar Giri', 757, 15000);

>>>insert into student_fees values(5, 'Ronit Singh', 887, 15000);

ROLL	NAME	TOTAL_MARKS	COURSEFEES
1	Rajendra Kumar Shaw	957	15000
2	Rahul Kumar Shaw	750	15000
3	Harry Potter	988	15000
4	Atul Kumar Giri	757	15000
5	Ronit Singh	887	15000

c) Decrease the course fee by 50% if Total Marks obtained by the student is above 90% and display a message showing the number of students got this fee waiver or none.

SOL:

```
DECLARE

    X NUMBER(5);

    FULL_MARKS NUMBER(5):=1000;

BEGIN

    UPDATE STUDENT_FEES SET COURSEFEES = COURSEFEES*0.5 WHERE TOTAL_MARKS
> FULL_MARKS*0.9;

    X:=SQL%ROWCOUNT;

     DBMS_OUTPUT.PUT_LINE('Total ' || X || ' number of students got the
fees waiver.');

END;
```

OUTPUT:

Total 2 number of students got the fees waiver.

ROLL	NAME	TOTAL_MARKS	COURSEFEES
1	Rajendra Kumar Shaw	957	7500
2	Rahul Kumar Shaw	750	15000
3	Harry Potter	988	7500
4	Atul Kumar Giri	757	15000
5	Ronit Singh	887	15000

2) Create a table EMP(Eno, Ename, Designation, Deptnum) with at least five records.

SOL:

>>> CREATE TABLE EMP3(ENO NUMBER(3) PRIMARY KEY, ENAME VARCHAR2(20), DESIGNATION VARCHAR2(20), DEPTNUM NUMBER(3), SALARY NUMBER(9,2));

SCHEMA:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMP3	ENO	Number	-	3	0	1	-	-	-
	ENAME	Varchar2	20	-	-	-	☐	-	-
	DESIGNATION	Varchar2	20	-	-	-	☐	-	-
	DEPTNUM	Number	-	3	0	-	☐	-	-
	SALARY	Number	-	9	2	-	☐	-	-

POPULATING TABLE:

>>>INSERT INTO EMP3 VALUES(1, 'RAJENDRA', 'SDE', 5, 25000);

>>>INSERT INTO EMP3 VALUES(2, 'RAHUL', 'SOC', 3, 30000);

>>>INSERT INTO EMP3 VALUES(3, 'RONIT', 'DBA', 5, 35000);

>>>INSERT INTO EMP3 VALUES(4, 'ATUL', 'ANALYST', 2, 45000);

>>>INSERT INTO EMP3 VALUES(5, 'HARRY', 'SDE', 5, 20000);

ENO	ENAME	DESIGNATION	DEPTNUM	SALARY
1	RAJENDRA	SDE	5	25000
2	RAHUL	SOC	3	30000
3	RONIT	DBA	5	35000
4	ATUL	ANALYST	2	45000
5	HARRY	SDE	5	20000

a) Check whether there is any employee in department number 5 having salary below ₹30000.

SOL:

```
DECLARE

    X NUMBER(5);

BEGIN

      SELECT COUNT(*) INTO X FROM EMP3 WHERE SALARY < 30000 AND
   DEPTNUM = 5;

    DBMS_OUTPUT.PUT_LINE('Total ' || X || ' number of employees.');

END;
```

b) Transfer those employees of department number 5 to department number 2 with a salary hike of 10%. Find how many employees are being transferred.

SOL:

```
DECLARE

    X NUMBER(5);

BEGIN

     UPDATE EMP3 SET DEPTNUM = 2, SALARY = SALARY + (SALARY * 0.1) WHERE
SALARY < 30000 AND DEPTNUM = 5;

    X:=SQL%ROWCOUNT;

        DBMS_OUTPUT.PUT_LINE('Total  '  ||  X  ||  '  number  of  employees
transferred.');

END;
```

OUTPUT:

Total 2 number of employees transferred.

ENO	ENAME	DESIGNATION	DEPTNUM	SALARY
1	RAJENDRA	SDE	2	27500
2	RAHUL	SOC	3	30000
3	RONIT	DBA	5	35000
4	ATUL	ANALYST	2	45000
5	HARRY	SDE	2	22000

Explicit cursor:

1)

a) Create the table: Attendance(Roll, MCA401, MCA402,MCA403) to keep the percentage of attendance of each student for three subjects and populate the table with at least 5 records.

SOL:

>>> CREATE TABLE ATTENDANCE(ROLL NUMBER(3) PRIMARY KEY, MCA401 NUMBER(3), MCA402 NUMBER(1),MCA403 NUMBER(1));

SCHEMA

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ATTENDANCE	ROLL	Number	-	3	0	1	-	-	-
	MCA401	Number	-	3	0	-	☐	-	-
	MCA402	Number	-	3	0	-	☐	-	-
	MCA403	Number	-	3	0	-	☐	-	-

POPULATING THE TABLE:

>>>INSERT INTO ATTENDANCE VALUES(1, 25, 27, 30);

>>>INSERT INTO ATTENDANCE VALUES(2, 23, 15, 20);

>>>INSERT INTO ATTENDANCE VALUES(3, 26, 24, 30);

>>>INSERT INTO ATTENDANCE VALUES(4, 15, 17, 19);

>>>INSERT INTO ATTENDANCE VALUES(5, 17, 14, 23);

ROLL	MCA401	MCA402	MCA403
1	25	27	30
2	23	15	20
3	26	24	30
4	15	17	19
5	17	14	23

b) Create a table NOT_ELIGIBLE(Roll, Avg_attendance).

SOL:

>>> CREATE TABLE NOT_ELIGIBLE(ROLL NUMBER(3) REFERENCES ATTENDANCE(ROLL), AVG_ATTENDANCE NUMBER(4,1));

SCHEMA

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
NOT_ELIGIBLE	ROLL	Number	-	3	0	-	☐	-	-
	AVG_ATTENDANCE	Number	-	4	1	-	☐	-	-

c) For each student calculate the average percentage of attendance and if it is less than 70% then insert a record in the table NOT_ELIGIBLE for that student.

SOL:

CONSIDERING THE ATTENDANCE RECORD IS FOR 30 SESSIONS FOR EACH SUBJECT.

```
DECLARE
    CURSOR C IS SELECT ROLL, MCA401, MCA402, MCA403 FROM ATTENDANCE;
    ROLL ATTENDANCE.ROLL%TYPE;
    MCA401 ATTENDANCE.MCA401%TYPE;
    MCA402 ATTENDANCE.MCA402%TYPE;
    MCA403 ATTENDANCE.MCA403%TYPE;
    AVG_ATTENDANCE NOT_ELIGIBLE.AVG_ATTENDANCE%TYPE;
    FULL_ATTENDANCE NUMBER(3):=90;
    ELIGIBILITY_CRITERIA NUMBER(4,1):= 90*0.7;
BEGIN
    OPEN C;
    LOOP
        FETCH C INTO ROLL, MCA401, MCA402, MCA403;
        EXIT WHEN C%NOTFOUND;
        AVG_ATTENDANCE:= ((MCA401+MCA402+MCA403)/FULL_ATTENDANCE)*100;
        IF AVG_ATTENDANCE < ELIGIBILITY_CRITERIA THEN
            INSERT INTO NOT_ELIGIBLE VALUES(ROLL, AVG_ATTENDANCE);
        END IF;
    END LOOP;
    CLOSE C;
END;
```

THE REFLECTION IN TABLE NOT_ELIGIBLE

ROLL	AVG_ATTENDANCE
4	56.7
5	60