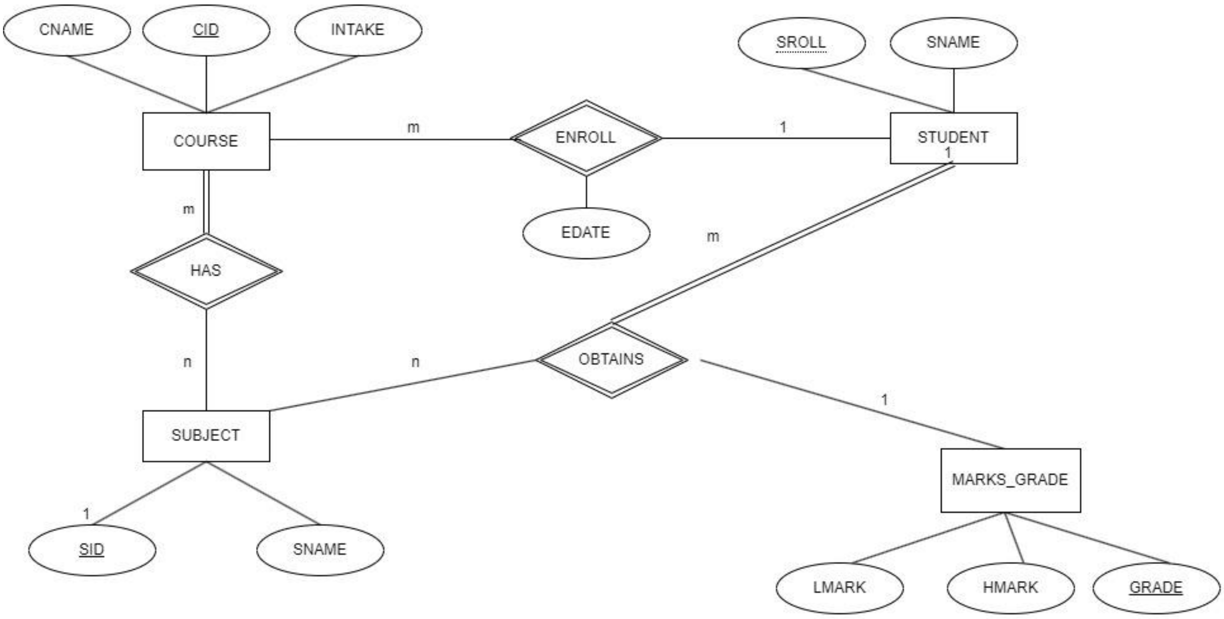
ATRIJ ROY JU IT A3 UG2
ROLL NO :- 002311001086
ASSIGNMENT 3

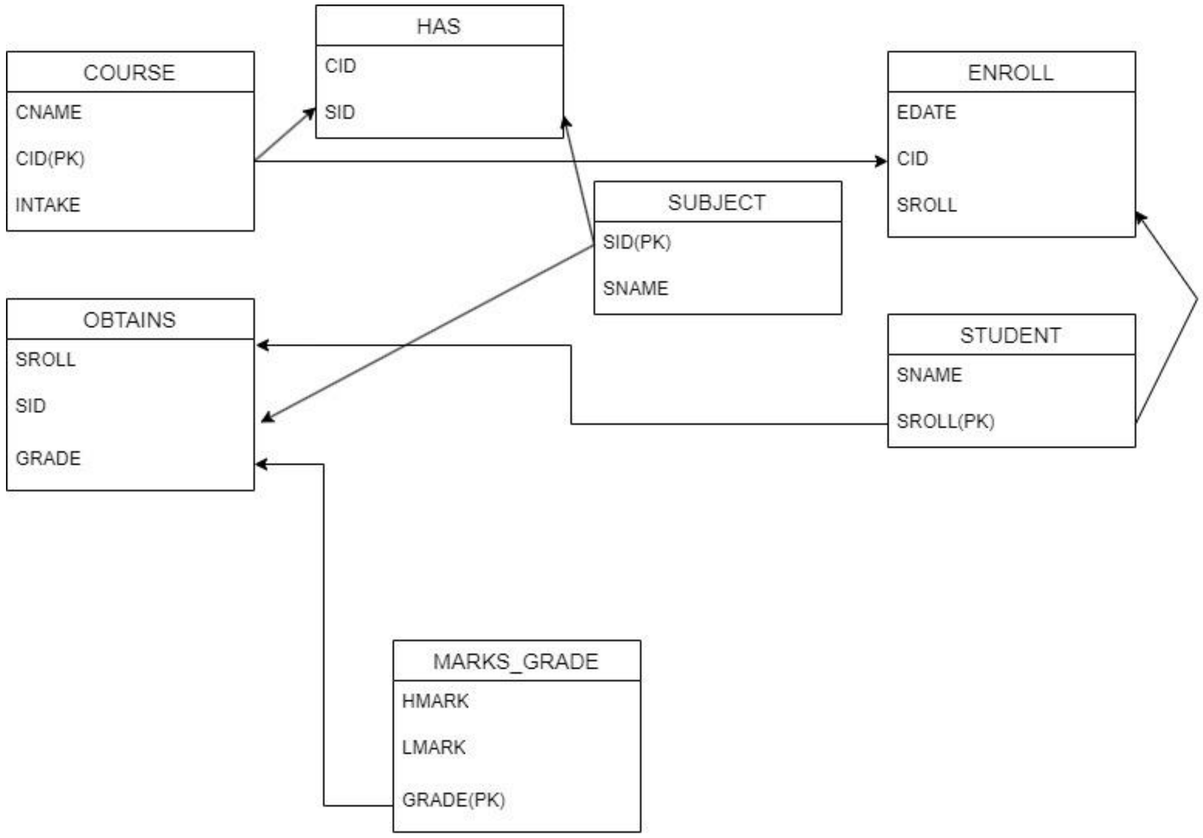
Question:-

In an educational institute, various numbers of courses are offered. In each course, 7 numbers of subjects are taught. One student can select minimum 5 and maximum 6 numbers of subjects for that course. Each course has maximum intake capacity. The same subject may be taught in various courses. The system must be able to handle course, subject, student, marks grade and enrollment information. Assumptions also can be made. Design an ER diagram and database schema for the system. Specify the primary key, foreign key and other constraints for all required tables. Draw the ER diagram in MS Word.

ER DIAGRAM



RELATION SCHEMA



1. Insert at least five tuples in each table.

   CREATE TABLE COURSE(

   CID NUMBER(3) primary key,

   CNAME VARCHAR2(10) not null,

   INTAKE NUMBER(3) not null);

   CREATE TABLE SUBJECT (

   SID NUMBER(2) PRIMARY KEY,

```
SNAME VARCHAR2(50) NOT NULL
);
CREATE TABLE STUDENT(
   SNAME VARCHAR2(10) NOT NULL,
   SROLL NUMBER(2) PRIMARY KEY
   );

CREATE TABLE HAS(
   CID NUMBER(1),
   SID NUMBER(2),
   PRIMARY KEY (CID,SID),
   FOREIGN KEY (CID) REFERENCES COURSE(CID),
   FOREIGN KEY (SID) REFERENCES SUBJECT(SID)
   )

CREATE TABLE MARKS_GRADE(
   GRADE VARCHAR(1) PRIMARY KEY,
   HMARK NUMBER(2) NOT NULL,
   LMARK NUMBER(2) NOT NULL


);
CREATE TABLE OBTAINS(
   SID NUMBER(2) NOT NULL,
   SROLL NUMBER(2)NOT NULL,
   GRADE VARCHAR2(1) NOT NULL,
    PRIMARY KEY (SID,SROLL,GRADE),
    FOREIGN KEY(SID) REFERENCES SUBJECT(SID),
    FOREIGN KEY(SROLL) REFERENCES STUDENT(SROLL),
    FOREIGN KEY(GRADE) REFERENCES MARKS_GRADE(GRADE)
);

CREATE TABLE ENROLL
(
    EDATE DATE NOT NULL,
    CID NUMBER(2) NOT NULL,
    SROLL NUMBER(2)NOT NULL,
    PRIMARY KEY (CID,SROLL),
    FOREIGN KEY(CID) REFERENCES COURSE(CID),
    FOREIGN KEY(SROLL) REFERENCES STUDENT(SROLL)
);


INSERT INTO COURSE VALUES('CSE',1,80);
INSERT INTO COURSE VALUES('IT',2,90);
INSERT INTO COURSE VALUES('ETCE',3,90);
INSERT INTO COURSE VALUES('IEE',4,75);
INSERT INTO COURSE VALUES('EE',5,98);


INSERT INTO SUBJECT VALUES(1,'DSA');
INSERT INTO SUBJECT VALUES(2,'DBMS');
INSERT INTO SUBJECT VALUES(3,'OOPS');
INSERT INTO SUBJECT VALUES(4,'WIRELESS');
INSERT INTO SUBJECT VALUES(5,'GRAPHS');
INSERT INTO SUBJECT VALUES(6,'ANA_DIGITAL');
INSERT INTO SUBJECT VALUES(7,'CIRCUIT_TH');
INSERT INTO SUBJECT VALUES(8,'COA');
INSERT INTO SUBJECT VALUES(9,'SIGNAL');
INSERT INTO SUBJECT VALUES(10,'AC-DC');

INSERT INTO HAS VALUES(1,1);
INSERT INTO HAS VALUES(1,2);
INSERT INTO HAS VALUES(1,3);
INSERT INTO HAS VALUES(1,5);
INSERT INTO HAS VALUES(1,6);
INSERT INTO HAS VALUES(2,1);
INSERT INTO HAS VALUES(2,2);
INSERT INTO HAS VALUES(2,3);
INSERT INTO HAS VALUES(2,4);
INSERT INTO HAS VALUES(2,5);
INSERT INTO HAS VALUES(3,1);
INSERT INTO HAS VALUES(3,5);
INSERT INTO HAS VALUES(3,6);
INSERT INTO HAS VALUES(3,7);
INSERT INTO HAS VALUES(3,8);
INSERT INTO HAS VALUES(4,1);
INSERT INTO HAS VALUES(4,8);
INSERT INTO HAS VALUES(4,9);
INSERT INTO HAS VALUES(4,4);
INSERT INTO HAS VALUES(4,7);
INSERT INTO HAS VALUES(5,10);
INSERT INTO HAS VALUES(5,7);
INSERT INTO HAS VALUES(5,8);
INSERT INTO HAS VALUES(5,9);
INSERT INTO HAS VALUES(5,6);

INSERT INTO STUDENT VALUES('HARI',1);
INSERT INTO STUDENT VALUES('RAM',2);
INSERT INTO STUDENT VALUES('SHYAM',3);
INSERT INTO STUDENT VALUES('JODU',4);
INSERT INTO STUDENT VALUES('MADHU',5);
INSERT INTO STUDENT VALUES('GOPAL',6);
INSERT INTO STUDENT VALUES('MOHAN',7);
INSERT INTO STUDENT VALUES('RADHA',8);

INSERT INTO ENROLL VALUES ('12-AUG-24',1,1);
INSERT INTO ENROLL VALUES ('12-AUG-24',1,2);
INSERT INTO ENROLL VALUES ('12-AUG-24',2,3);
INSERT INTO ENROLL VALUES ('05-SEP-24',2,4);
INSERT INTO ENROLL VALUES ('05-SEP-24',2,5);
```

```sql
INSERT INTO ENROLL VALUES ('05-SEP-24',3,6);
INSERT INTO ENROLL VALUES ('03-OCT-24',4,7);
INSERT INTO ENROLL VALUES ('03-OCT-24',5,8);

INSERT INTO MARKS_GRADE VALUES ('A',99,91);
INSERT INTO MARKS_GRADE VALUES ('B',90,81);
INSERT INTO MARKS_GRADE VALUES ('C',80,71);
INSERT INTO MARKS_GRADE VALUES ('D',70,61);
INSERT INTO MARKS_GRADE VALUES ('E',60,51);
INSERT INTO MARKS_GRADE VALUES ('F',50,0);

INSERT INTO OBTAINS VALUES (1,3,'A');
INSERT INTO OBTAINS VALUES (2,3,'B');
INSERT INTO OBTAINS VALUES (3,3,'A');
INSERT INTO OBTAINS VALUES (4,3,'D');
INSERT INTO OBTAINS VALUES (5,3,'C');
INSERT INTO OBTAINS VALUES (1,4,'A');
INSERT INTO OBTAINS VALUES (2,4,'B');
INSERT INTO OBTAINS VALUES (3,4,'A');
INSERT INTO OBTAINS VALUES (4,4,'B');
INSERT INTO OBTAINS VALUES (5,4,'A');
INSERT INTO OBTAINS VALUES (1,5,'A');
INSERT INTO OBTAINS VALUES (2,5,'B');
INSERT INTO OBTAINS VALUES (3,5,'A');
INSERT INTO OBTAINS VALUES (4,5,'A');
INSERT INTO OBTAINS VALUES (5,5,'D');
```

2. At the time of creation if we forget to create a field enrollment date (ENROLL_DATE) in ENROLL table so add the field.

```sql
ALTER TABLE ENROLL ADD (ENROLL_DATE DATE);
```

3. Course name cannot be blank, therefore add the criteria in the specific table.

```sql
ALTER TABLE COURSE MODIFY CNAME VARCHAR2(10) NOT NULL;
```

4. Find the Course which has more than 3 students.

```sql
SELECT * FROM ( SELECT COUNT(CID) AS STUD_COUNT,CID,CNAME FROM

   (SELECT ENROLL.CID, COURSE.CNAME FROM ENROLL, COURSE WHERE ENROLL.CID=COURSE.CID)

   GROUP BY CID,CNAME) WHERE STUD_COUNT>3;
```

5. Give the details of a STUDENT with all Subjects and Grade where he/she enroll (Enter the sid value as input).

```sql
SELECT STUDENT.SROLL, STUDENT.SNAME, SUBJECT.SID, SUBJECT.SNAME, OBTAINS.GRADE

FROM STUDENT INNER JOIN OBTAINS ON STUDENT.SROLL=OBTAINS.SROLL

INNER JOIN SUBJECT ON OBTAINS.SID= SUBJECT.SID

WHERE STUDENT.SROLL = 3;
```

6. Display the course where the maximum number of students enrolls.

```sql
SELECT STUDENT_COUNT, CNAME FROM (SELECT  COUNT(CID) AS STUDENT_COUNT, CID, CNAME FROM

   (SELECT ENROLL.CID, COURSE.CNAME FROM ENROLL,COURSE

   WHERE ENROLL.CID=COURSE.CID) GROUP BY CID, CNAME)

   WHERE STUDENT_COUNT= (SELECT MAX(STUDENT_COUNT) FROM (SELECT COUNT(CID) AS STUDENT_COUNT, CID, CNAME

   FROM ENROLL, COURSE WHERE ENROLL.CID=COURSE.CID) GROUP BY CID, CNAME);
```

7. Find out the course where no student is enrolled.

```sql
SELECT COURSE.CID, CNAME FROM COURSE LEFT JOIN ENROLL ON COURSE.CID= ENROLL.CID WHERE SID IS NULL;
```

8. Delete Course no 30 from COURSE table.

DELETE FROM COURSE WHERE CID=30;

9. Rename the COURSE table as DEPARTMENT.

   RENAME COURSE TO DEPARTMENT;

10. Change the Marks Grade of Student "A" to "B" who is Enroll in the subject DBMS.

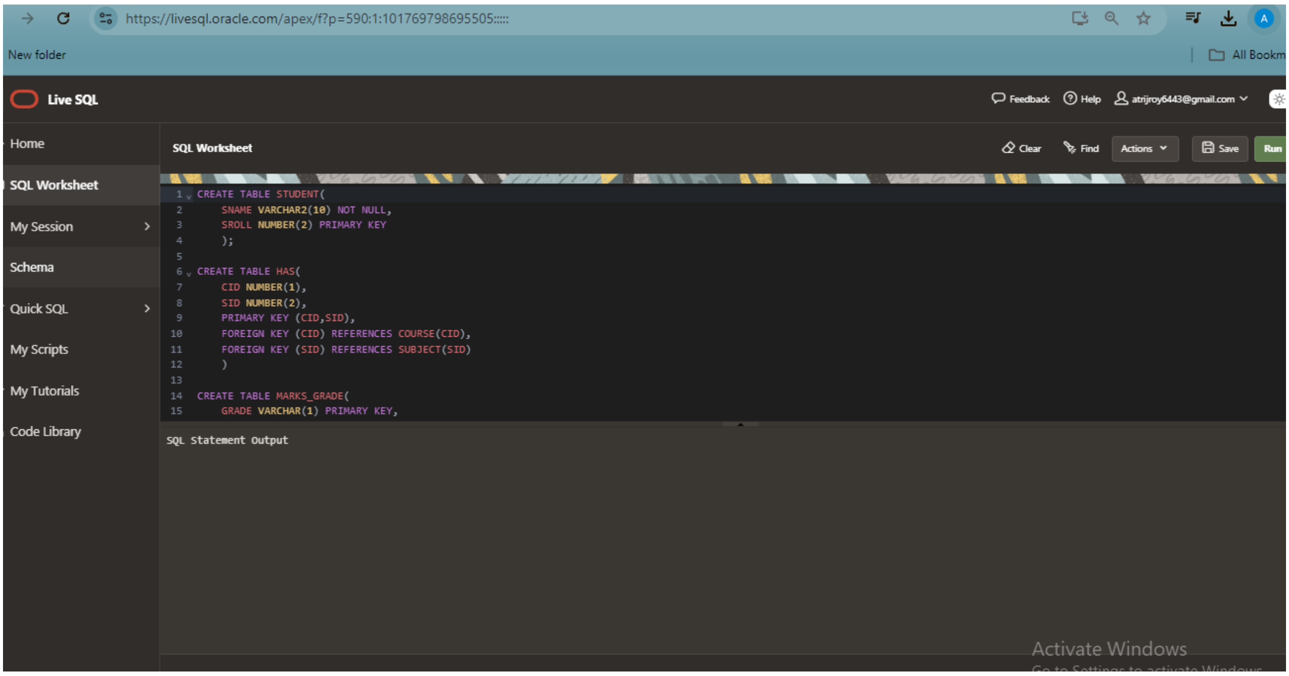
    UPDATE OBTAINS SET GRADE='B' WHERE SID=2 AND GRADE='A'

11. Delete the record of the student who is enrolled in the course 'IT'.

    DELETE FROM COURSE WHERE CNAME ="IT";

12. Change the enroll date to '16-08-2018' whose student id is 18069 (first convert the date into the default format).

UPDATE ENROLL SET EDATE = '16-AUG-18' WHERE SROLL=18069;



```
CREATE TABLE STUDENT(
    SNAME VARCHAR2(10) NOT NULL,
    SROLL NUMBER(2) PRIMARY KEY
    );

CREATE TABLE HAS(
    CID NUMBER(1),
    SID NUMBER(2),
    PRIMARY KEY (CID,SID),
    FOREIGN KEY (CID) REFERENCES COURSE(CID),
    FOREIGN KEY (SID) REFERENCES SUBJECT(SID)
    )

CREATE TABLE MARKS_GRADE(
    GRADE VARCHAR(1) PRIMARY KEY,
```

SQL Statement Output