# Assignment 3

Name: Soumitri Chattopadhyay

Roll: 001911001083

JU, IT’23

**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 a ER diagram and database schema for the system. Specify the primary key, foreign key and other constraints for all required tables.**

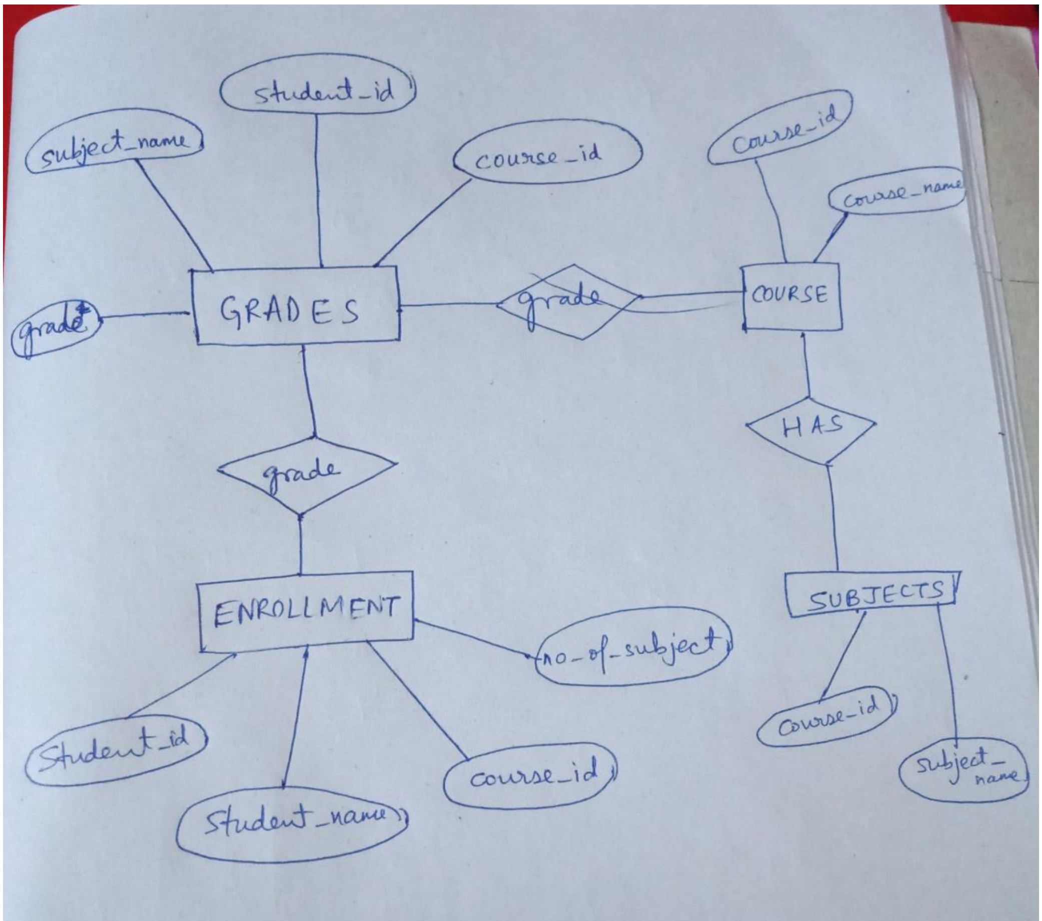

Fig: ER Diagram

```
CREATE TABLE COURSE (
    course_id INT(10) NOT NULL,
    course_name VARCHAR(255),
    max_intake INT(5) NOT NULl,
    PRIMARY KEY(course_id)
 );

CREATE TABLE SUBJECTS (
    course_id INT(10) NOT NULL,
    subject_name VARCHAR(255) NOT NULL,
    FOREIGN KEY(course_id) REFERENCES COURSE(course_id)
 );

CREATE TABLE ENROLLMENT (
    student_id INT(10) NOT NULL,
    student_name VARCHAR(255) NOT NULL,
    course_id INT(10) NOT NULL,
    no_of_subjects INT(1) NOT NULL,
    PRIMARY KEY(student_id),
    FOREIGN KEY(course_id) REFERENCES COURSE(course_id),
    CHECK (no_of_subjects >= 5 AND no_of_subjects <= 6)
 );

CREATE TABLE GRADES (
    student_id INT(10) NOT NULL,
    course_id INT(10) NOT NULL,
    subject_name VARCHAR(255) NOT NULL,
    grade CHAR(1) NOT NULL,
    FOREIGN KEY(student_id) REFERENCES ENROLLMENT(student_id),
    FOREIGN KEY(course_id) REFERENCES COURSE(course_id)
 );
```

1. **Insert at least five tuples in each table.**

```
INSERT INTO COURSE VALUES(10,"CSE",86);
INSERT INTO COURSE VALUES(20,"ETCE",86);
INSERT INTO COURSE VALUES(30,"EE",143);
INSERT INTO COURSE VALUES(40,"IT",92);
INSERT INTO COURSE VALUES(50,"MECH",118);

INSERT INTO SUBJECTS VALUES(10,"CPNM");
INSERT INTO SUBJECTS VALUES(10,"DBMS");
INSERT INTO SUBJECTS VALUES(10,"OOPS");
INSERT INTO SUBJECTS VALUES(10,"MACHINE LEARNING");
INSERT INTO SUBJECTS VALUES(10,"CRYPTOGRAPHY");

INSERT INTO SUBJECTS VALUES(20,"VLSI");
INSERT INTO SUBJECTS VALUES(20,"DIGITAL CIRCUIT");
INSERT INTO SUBJECTS VALUES(20,"NANOELECTRONICS");
INSERT INTO SUBJECTS VALUES(20,"INTEGRATE CIRCUITS");
INSERT INTO SUBJECTS VALUES(20,"MICROPROCESSORS");

INSERT INTO SUBJECTS VALUES(30,"KIRCHOFF LAWS");
INSERT INTO SUBJECTS VALUES(30,"MICROELECTRONICS");
INSERT INTO SUBJECTS VALUES(30,"ALTERNATING CURRENT");
INSERT INTO SUBJECTS VALUES(30,"POWER ENGINEERING");
INSERT INTO SUBJECTS VALUES(30,"ELECTRICAL INSTRUMENTATION");

INSERT INTO SUBJECTS VALUES(40,"DSA HANDWRITTEN");
INSERT INTO SUBJECTS VALUES(40,"DBMS");
INSERT INTO SUBJECTS VALUES(40,"DATA SCIENCE");
INSERT INTO SUBJECTS VALUES(40,"ANDROID DEVELOPMENT");
INSERT INTO SUBJECTS VALUES(40,"BLOCKCHAINS");

INSERT INTO SUBJECTS VALUES(50,"ELASTICITY");
INSERT INTO SUBJECTS VALUES(50,"MACHINES");
INSERT INTO SUBJECTS VALUES(50,"ENGINEERING DRAWING");
INSERT INTO SUBJECTS VALUES(50,"HYDROSTATICS");
INSERT INTO SUBJECTS VALUES(50,"FREE BODY DIAGRAMS");

INSERT INTO ENROLLMENT VALUES(15059,"IVAN EGOROV",10,5);
INSERT INTO ENROLLMENT VALUES(18069,"DOMINIKA .",40,5);
INSERT INTO ENROLLMENT VALUES(18019,"SIMON SPIER",40,5);
INSERT INTO ENROLLMENT VALUES(19707,"PRADOSH MITTER",20,5);
INSERT INTO ENROLLMENT VALUES(16061,"ANAND BAKSHI",10,5);

INSERT INTO GRADES VALUES(15059,10,"CPNM",'A');
INSERT INTO GRADES VALUES(15059,10,"DBMS",'A');
INSERT INTO GRADES VALUES(15059,10,"OOPS",'B');
INSERT INTO GRADES VALUES(15059,10,"MACHINE LEARNING",'C');
INSERT INTO GRADES VALUES(15059,10,"CRYPTOGRAPHY",'A');

INSERT INTO GRADES VALUES(16061,10,"CPNM",'A');
INSERT INTO GRADES VALUES(16061,10,"DBMS",'C');
INSERT INTO GRADES VALUES(16061,10,"OOPS",'D');
INSERT INTO GRADES VALUES(16061,10,"MACHINE LEARNING",'A');
INSERT INTO GRADES VALUES(16061,10,"CRYPTOGRAPHY",'D');

INSERT INTO GRADES VALUES(18019,40,"DSA HANDWRITTEN",'C');
INSERT INTO GRADES VALUES(18019,40,"DBMS",'A');
INSERT INTO GRADES VALUES(18019,40,"BLOCKCHAINS",'D');
INSERT INTO GRADES VALUES(18019,40,"ANDROID DEVELOPMENT",'A');
INSERT INTO GRADES VALUES(18019,40,"DATA SCIENCE",'A');

INSERT INTO GRADES VALUES(18069,40,"DSA HANDWRITTEN",'A');
INSERT INTO GRADES VALUES(18069,40,"DBMS",'B');
INSERT INTO GRADES VALUES(18069,40,"BLOCKCHAINS",'B');
INSERT INTO GRADES VALUES(18069,40,"ANDROID DEVELOPMENT",'C');
INSERT INTO GRADES VALUES(18069,40,"DATA SCIENCE",'B');

INSERT INTO GRADES VALUES(19707,20,"VLSI",'A');
INSERT INTO GRADES VALUES(19707,20,"DIGITAL CIRCUIT",'D');
INSERT INTO GRADES VALUES(19707,20,"NANOELECTRONICS",'B');
INSERT INTO GRADES VALUES(19707,20,"INTEGRATE CIRCUITS",'A');
INSERT INTO GRADES VALUES(19707,20,"MICROPROCESSORS",'B');
```

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

```
ALTER TABLE ENROLLMENT

ADD enroll_date DATE;
```

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

```
ALTER TABLE COURSE

MODIFY course_name VARCHAR(255) NOT NULL;
```

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

```
SELECT course_id

FROM ENROLLMENT

WHERE course_id IN (

    SELECT course_id FROM ENROLLMENT

    GROUP BY course_id

    HAVING COUNT(*) >= 3

);
```

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

```
SELECT E.student_id AS STID, E.student_name AS NAME, E.course_id AS
COURSEID, C.course_name AS COURSE_NAME, G.subject_name AS
SUBJECT_NAME, G.grade AS GRADE
FROM ENROLLMENT E, GRADES G, COURSE C
WHERE E.student_id = G.student_id AND E.student_id = '&E.student_id'
AND E.course_id = C.course_id;
```

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

```
SELECT course_name
FROM COURSE
WHERE course_id IN (
SELECT course_id
                FROM ENROLLMENT
                GROUP BY course_id
                HAVING COUNT(course_id) = (SELECT
MAX(COUNT(course_id))
                                        FROM ENROLLMENT
                                        GROUP BY course_id)
);
```

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

```
SELECT course_name
FROM COURSE
WHERE course_id NOT IN (SELECT course_id
                FROM ENROLLMENT
                GROUP BY course_id) ;
```

8. **Delete Course no 30 from COURSE table.**

```
DELETE FROM COURSE WHERE course_id = 30 ;
```

9. **Rename the COURSE table as DEPARTMENT.**

```
ALTER TABLE COURSE
RENAME TO DEPARTMENT ;
```

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

```
UPDATE GRADES
SET grade = 'B'
WHERE subject_name = “DBMS”;
```

11. **Delete the record of the student who is enrolled in the course ‘IT’.**

```
DELETE FROM DEPARTMENT WHERE course_name = “IT”;
```

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

```
UPDATE ENROLLMENT
SET enroll_date = “2018-08-16”
WHERE student_id = 18069;
```