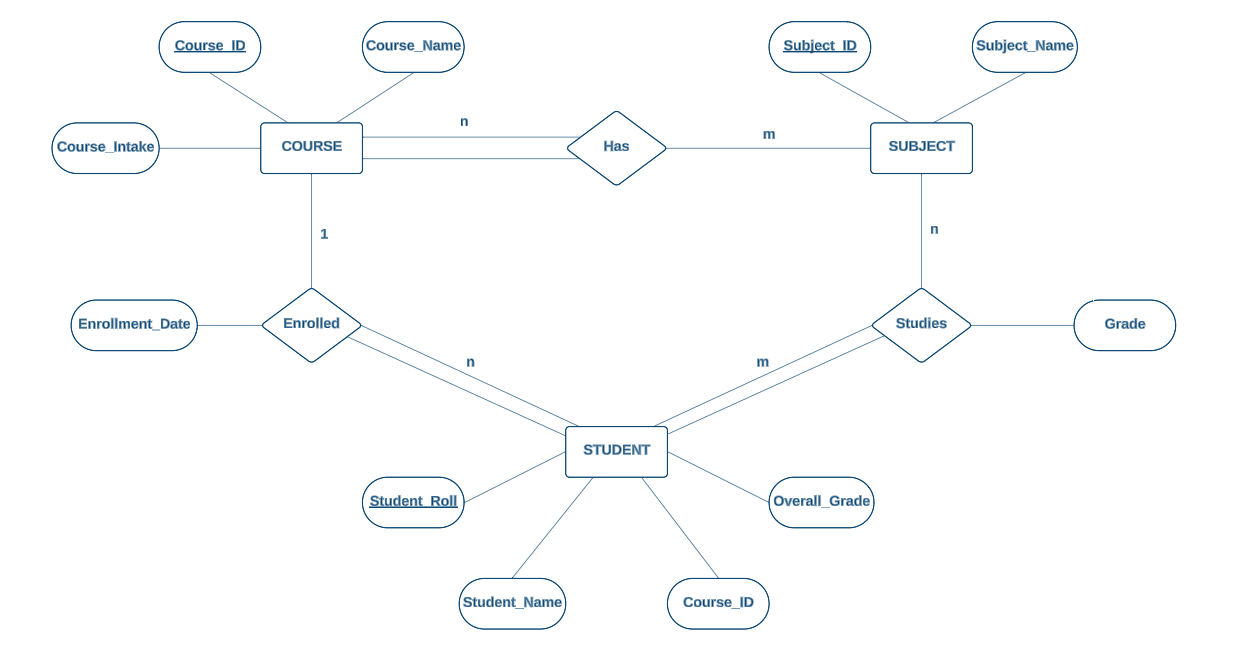
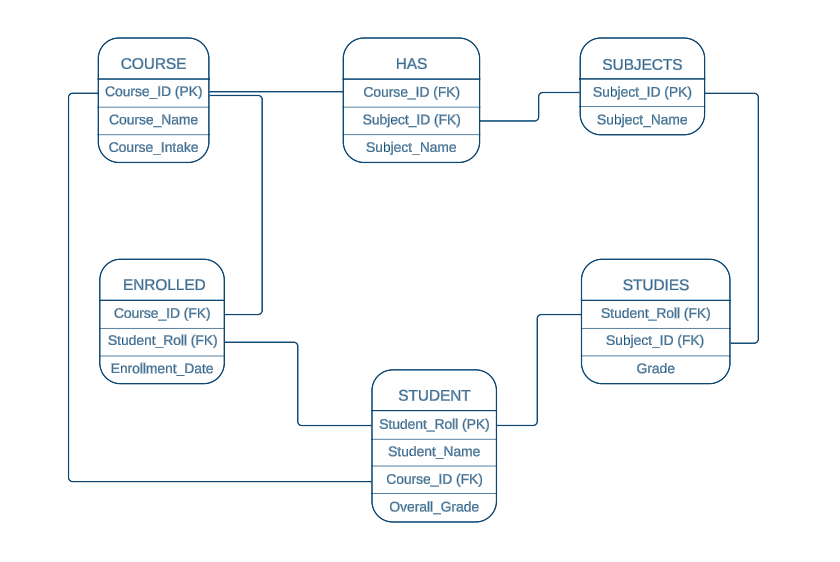
**ASSIGNMENT – 3**

**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 enrolment 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.

**E-R Diagram:**

****

**Database Schema:**

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

-- COURSE Table

Create Table COURSE (

COURSE\_ID number (2), COURSE\_NAME varchar2(10), COURSE\_INTAKE number (3), primary key(COURSE\_ID));

Insert into COURSE values (10, 'CS', 100);

Insert into COURSE values (20, 'IT', 100);

Insert into COURSE values (30, 'ECE', 120);

Insert into COURSE values (40, 'AI', 75);

Insert into COURSE values (50, 'WEB DEV', 90);

-- SUBJECT Table

Create Table SUBJECT (

SUBJECT\_ID number (3), SUBJECT\_NAME varchar2(15), primary key(SUBJECT\_ID));

Insert into SUBJECT values (101, 'DSA');

Insert into SUBJECT values (102, 'OOP');

Insert into SUBJECT values (103, 'DBMS');

Insert into SUBJECT values (104, 'ML');

Insert into SUBJECT values (105, 'BIG DATA');

Insert into SUBJECT values (106, 'COA');

Insert into SUBJECT values (107, 'GRAPH THEORY');

Insert into SUBJECT values (108, 'NETWORKING');

-- STUDENT Table

Create Table STUDENT (

STUDENT\_ROLL number (5), STUDENT\_NAME varchar2(25), COURSE\_ID number (2), OVERALL\_GRADE varchar2(1), primary key(STUDENT\_ROLL), foreign key(COURSE\_ID) REFERENCES course(COURSE\_ID) ON DELETE CASCADE);

Insert into STUDENT values (1050, 'RAGHAV', 10, 'B');

Insert into STUDENT values (1051, 'JASON', 20, 'A');

Insert into STUDENT values (1052, 'SOHAM', 30, 'C');

Insert into STUDENT values (1053, 'SRIPARNA', 20, 'B');

Insert into STUDENT values (1054, 'SHREYA', 10, 'A');

Insert into STUDENT values (1055, 'FAAIZ', 40, 'A');

Insert into STUDENT values (1056, 'TRIJOY', 20, 'c');

Insert into STUDENT values (1057, 'SHRAMAN', 30, 'B');

Insert into STUDENT values (1058, 'NEIL', 20, 'A');

Insert into STUDENT values (1069, 'ROHIT', 20, 'A');

-- ENROLLED Table

Create Table ENROLLED (

STUDENT\_ROLL number (5), COURSE\_ID number (2), foreign key(COURSE\_ID) REFERENCES course(COURSE\_ID) ON DELETE CASCADE, foreign key (STUDENT\_ROLL) REFERENCES student (STUDENT\_ROLL) ON DELETE CASCADE);

Insert into ENROLLED values (1050, 10);

Insert into ENROLLED values (1051, 20);

Insert into ENROLLED values (1052, 30);

Insert into ENROLLED values (1053, 20);

Insert into ENROLLED values (1054, 10);

Insert into ENROLLED values (1055, 40);

Insert into ENROLLED values (1056, 20);

Insert into ENROLLED values (1057, 30);

Insert into ENROLLED values (1058, 20);

Insert into ENROLLED values (1069, 40);

-- HAS Table

Create Table HAS (COURSE\_ID number (2), SUBJECT\_ID number (3), SUBJECT\_NAME varchar (15), foreign key(COURSE\_ID) REFERENCES course(COURSE\_ID) ON DELETE CASCADE, foreign key(SUBJECT\_ID) REFERENCES subject(SUBJECT\_ID) ON DELETE CASCADE);

Insert into HAS values (20, 101, 'DSA');

Insert into HAS values (20, 102, 'OOP');

Insert into HAS values (20, 103, 'DBMS');

Insert into HAS values (20, 104, 'ML');

Insert into HAS values (20, 105, 'BIG DATA');

Insert into HAS values (20, 106, 'COA');

Insert into HAS values (20, 108, 'NETWORKING');

Insert into HAS values (40, 101, 'DSA');

Insert into HAS values (40, 102, 'OOP');

Insert into HAS values (40, 103, 'DBMS');

Insert into HAS values (40, 104, 'ML');

Insert into HAS values (40, 105, 'BIG DATA');

Insert into HAS values (40, 108, 'NETWORKING');

Insert into HAS values (40, 107, 'GRAPH THEORY');

-- STUDIES Table

Create Table STUDIES (STUDENT\_ROLL number (5), SUBJECT\_ID number (3), GRADE varchar2(1), foreign key (STUDENT\_ROLL) REFERENCES student (STUDENT\_ROLL) ON DELETE CASCADE, foreign key(SUBJECT\_ID) REFERENCES subject(SUBJECT\_ID) ON DELETE CASCADE);

Insert into STUDIES values (1053, 101, 'A');

Insert into STUDIES values (1053, 102, 'B');

Insert into STUDIES values (1053, 103, 'A');

Insert into STUDIES values (1053, 104, 'C');

Insert into STUDIES values (1053, 106, 'A');

Insert into STUDIES values (1055, 104, 'C');

Insert into STUDIES values (1055, 105, 'B');

Insert into STUDIES values (1055, 107, 'A');

Insert into STUDIES values (1055, 103, 'C');

Insert into STUDIES values (1055, 108, 'A');

1. **At the time of creation if we forget to create a field enrollment date (ENROLL\_DATE)**

**in the ENROLL table so add the field.**

Alter Table ENROLLED

Add (ENROLL\_DATE date);

Update ENROLLED set ENROLL\_DATE='19-NOV-21' where STUDENT\_ROLL=1050;

Update ENROLLED set ENROLL\_DATE='17-JUL-20' where STUDENT\_ROLL=1051;

Update ENROLLED set ENROLL\_DATE='14-DEC-22' where STUDENT\_ROLL=1052;

Update ENROLLED set ENROLL\_DATE='21-JUL-21' where STUDENT\_ROLL=1053;

Update ENROLLED set ENROLL\_DATE='04-JUN-21' where STUDENT\_ROLL=1054;

Update ENROLLED set ENROLL\_DATE='08-AUG-20' where STUDENT\_ROLL=1055;

Update ENROLLED set ENROLL\_DATE='13-SEP-22' where STUDENT\_ROLL=1056;

Update ENROLLED set ENROLL\_DATE='02-NOV-21' where STUDENT\_ROLL=1057;

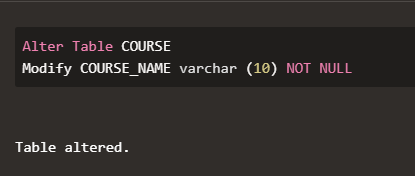
Update ENROLLED set ENROLL\_DATE='23-JUL-20' where STUDENT\_ROLL=1058;

Update ENROLLED set ENROLL\_DATE='18-OCT-22' where STUDENT\_ROLL=1069;

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

Alter Table COURSE

Modify COURSE\_NAME varchar (10) NOT NULL;



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

Select \* from

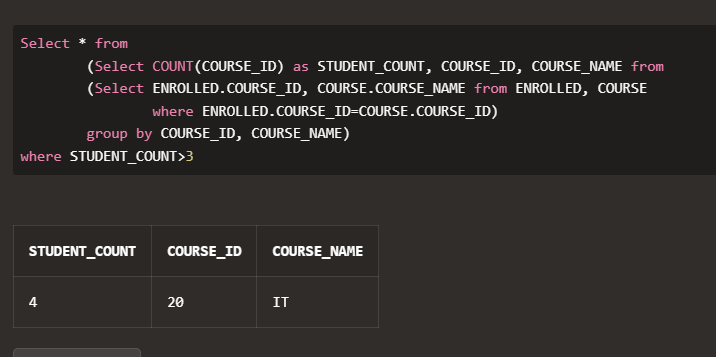
(Select COUNT(COURSE\_ID) as STUDENT\_COUNT, COURSE\_ID, COURSE\_NAME from

(Select ENROLLED.COURSE\_ID, COURSE.COURSE\_NAME from ENROLLED, COURSE

where ENROLLED.COURSE\_ID=COURSE.COURSE\_ID)

group by COURSE\_ID, COURSE\_NAME)

where STUDENT\_COUNT>3;



1. **Give the details of a STUDENT with all Subjects and Grade where he/she enrolls (Enter**

**the sid value as input).**

Select

STUDENT.STUDENT\_ROLL,

STUDENT.STUDENT\_NAME,

SUBJECT.SUBJECT\_ID,

SUBJECT.SUBJECT\_NAME,

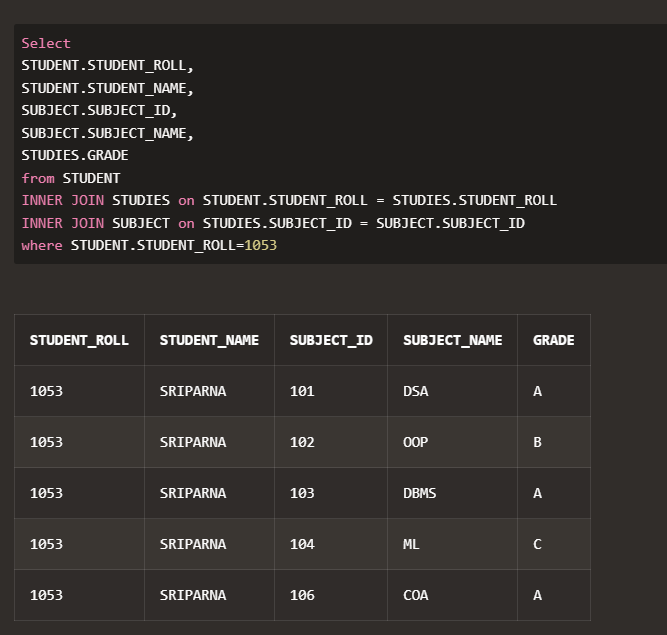
STUDIES.GRADE

from STUDENT

INNER JOIN STUDIES on STUDENT.STUDENT\_ROLL = STUDIES.STUDENT\_ROLL

INNER JOIN SUBJECT on STUDIES.SUBJECT\_ID = SUBJECT.SUBJECT\_ID

where STUDENT.STUDENT\_ROLL=1053;

****

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

Select STUDENT\_COUNT, COURSE\_NAME from (

select COUNT(COURSE\_ID) as STUDENT\_COUNT, COURSE\_ID, COURSE\_NAME

from (

select ENROLLED.COURSE\_ID, COURSE.COURSE\_NAME

from ENROLLED, COURSE

where ENROLLED.COURSE\_ID=COURSE.COURSE\_ID)

group by COURSE\_ID, COURSE\_NAME)

where STUDENT\_COUNT = (

select MAX(STUDENT\_COUNT) from (

select COUNT(COURSE\_ID) as STUDENT\_COUNT, COURSE\_ID, COURSE\_NAME

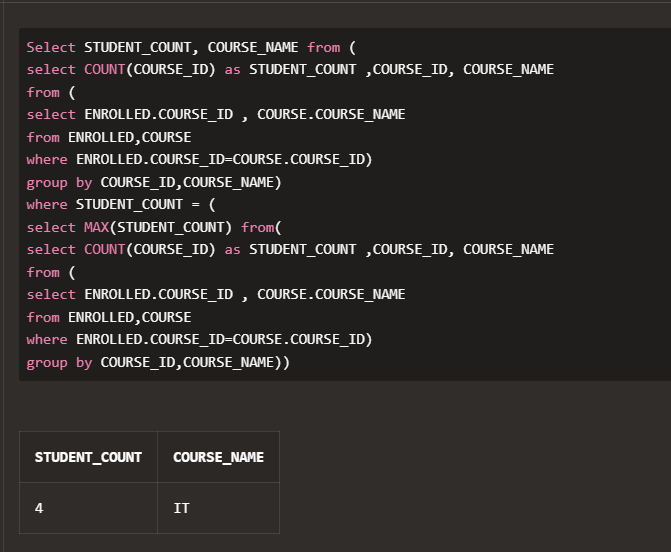
from (

select ENROLLED.COURSE\_ID, COURSE.COURSE\_NAME

from ENROLLED, COURSE

where ENROLLED.COURSE\_ID=COURSE.COURSE\_ID)

group by COURSE\_ID, COURSE\_NAME));

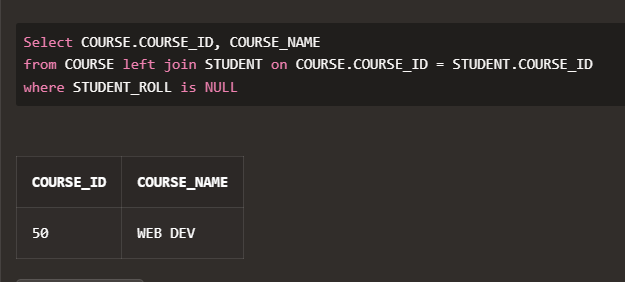


1. **Find out the course where no student is enrolled.**

Select COURSE.COURSE\_ID, COURSE\_NAME

from COURSE left join STUDENT on COURSE.COURSE\_ID = STUDENT.COURSE\_ID

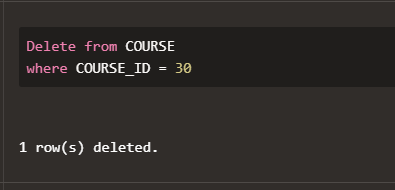
where STUDENT\_ROLL is NULL;



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

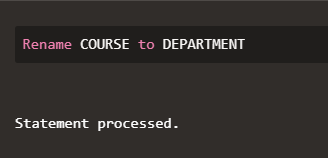
Delete from COURSE

where COURSE\_ID = 30;



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

Rename COURSE to DEPARTMENT;

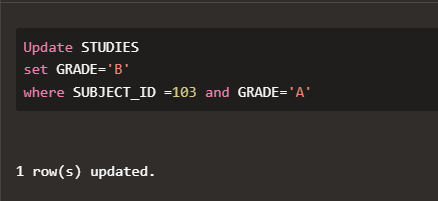


1. **Change the Marks Grade of Student “A” to “B” who is Enroll in the subject DBMS.**

Update STUDIES

set GRADE='B'

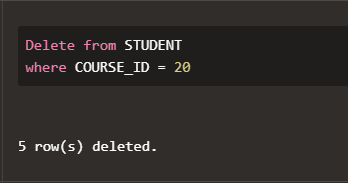
where SUBJECT\_ID =103 and GRADE='A';



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

Delete from STUDENT

where COURSE\_ID = 20;



1. **Change the enroll date to ‘16-08-2018’ whose student id is 18069 (first convert the**

**date into the default format).**

Update ENROLLED

set ENROLL\_DATE = '16-AUG-18'

where STUDENT\_ID = 18069;

