JADAVPUR UNIVERSITY
Department of Information Technology
DATABASE MANAGEMENT SYSTEM LAB

IT UG-2

ASSIGNMENT 3

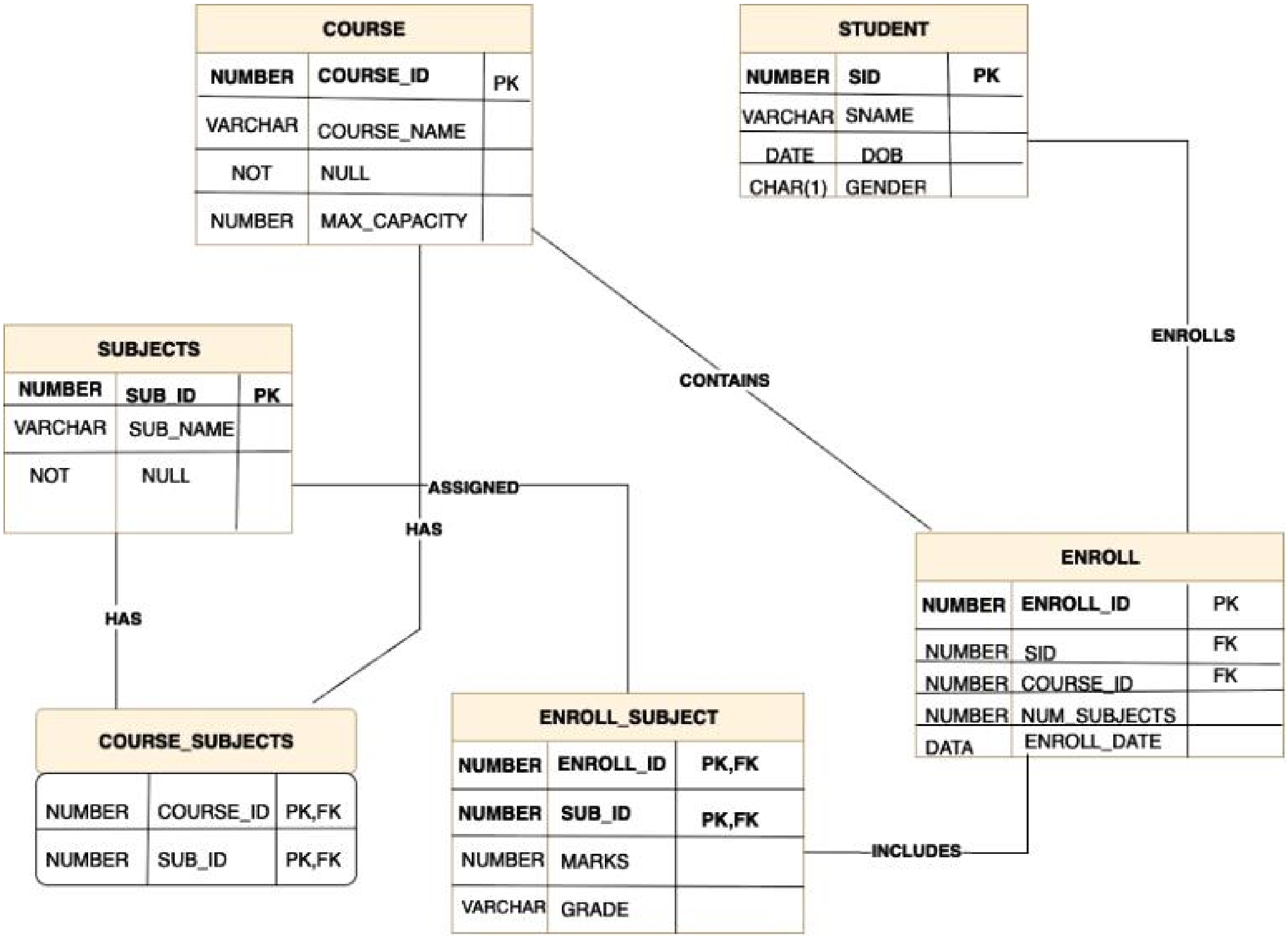
Name: ***Ankit Roy***

Roll : ***002411001028***

Year: 2024-25

Sec : ***A1***

1)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.



COURSE

NUMBER	COURSE_ID	PK
VARCHAR	COURSE_NAME	
NOT	NULL	
NUMBER	MAX_CAPACITY	

STUDENT

NUMBER	SID	PK
VARCHAR	SNAME	
DATE	DOB	
CHAR(1)	GENDER	

SUBJECTS

NUMBER	SUB_ID	PK
VARCHAR	SUB_NAME	
NOT	NULL	

ENROLL

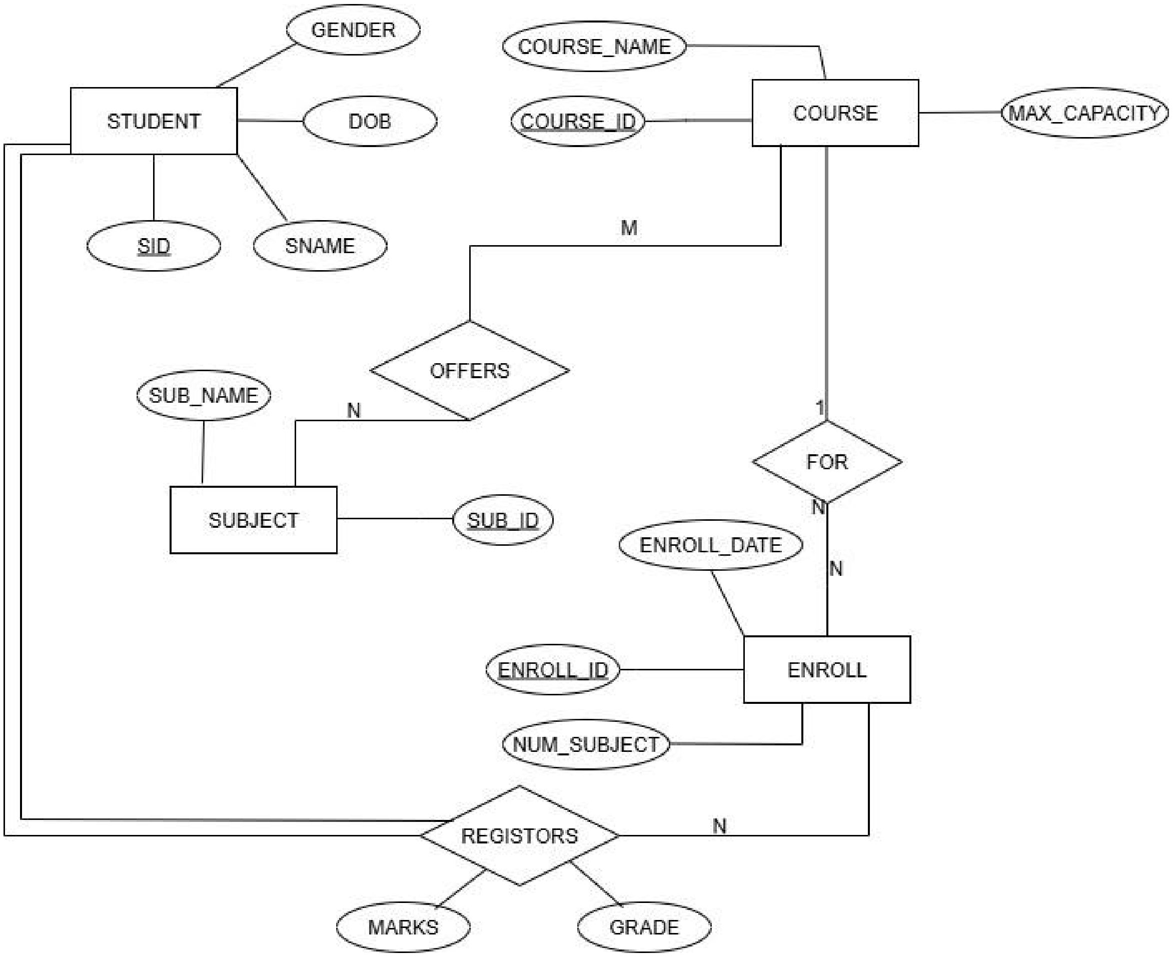
NUMBER	ENROLL_ID	PK
NUMBER	SID	FK
NUMBER	COURSE_ID	FK
NUMBER	NUM_SUBJECTS	
DATA	ENROLL_DATE	

COURSE_SUBJECTS

NUMBER	COURSE_ID	PK,FK
NUMBER	SUB_ID	PK,FK

ENROLL_SUBJECT

NUMBER	ENROLL_ID	PK,FK
NUMBER	SUB_ID	PK,FK
NUMBER	MARKS	
VARCHAR	GRADE	



Insert at least five tuples in each table.

```
CREATE TABLE COURSE(
  COURSE_ID NUMBER(2),
  COURSE_NAME VARCHAR2(10)
);

INSERT INTO COURSE VALUES (10, 'CSE');
INSERT INTO COURSE VALUES (20, 'IT');
INSERT INTO COURSE VALUES (30, 'EE');
INSERT INTO COURSE VALUES (40, 'ML');
INSERT INTO COURSE VALUES (50, 'WEBDEV');


CREATE TABLE SUBJECTS(
  SUBJECT_ID NUMBER(3),
  SUBJECT_NAME VARCHAR2(15)
);

INSERT INTO SUBJECTS VALUES (101, 'COA');
INSERT INTO SUBJECTS VALUES (102, 'DBMS');
INSERT INTO SUBJECTS VALUES (103, 'DSA');
INSERT INTO SUBJECTS VALUES (104, 'ML');
INSERT INTO SUBJECTS VALUES (105, 'LLD');
INSERT INTO SUBJECTS VALUES (106, 'CN');
INSERT INTO SUBJECTS VALUES (107, 'GRAPH THEORY');
INSERT INTO SUBJECTS VALUES (108, 'OS');
```

```sql
CREATE TABLE STUDENT(
  STUDENT_ID NUMBER(5),
  STUDENT_NAME VARCHAR2(25),
  COURSE_ID NUMBER(2),
  GRADE VARCHAR2(1)
);

INSERT INTO STUDENT VALUES (18066, 'SAP', 10, 'B');
INSERT INTO STUDENT VALUES (18067, 'RAJESH', 20, 'A');
INSERT INTO STUDENT VALUES (18068, 'SHREYA', 20, 'C');
INSERT INTO STUDENT VALUES (18069, 'SOHAM', 20, 'B');
INSERT INTO STUDENT VALUES (18070, 'KAUSTAV', 20, 'A');
INSERT INTO STUDENT VALUES (18071, 'BITESH', 40, 'A');
INSERT INTO STUDENT VALUES (18072, 'ANKIT', 50, 'B');
INSERT INTO STUDENT VALUES (18073, 'PRIYA', 30, 'A');
INSERT INTO STUDENT VALUES (18074, 'ROHAN', 10, 'C');


CREATE TABLE ENROLLED(
  STUDENT_ID NUMBER(5),
  COURSE_ID NUMBER(2)
);

INSERT INTO ENROLLED VALUES (18066, 10);
INSERT INTO ENROLLED VALUES (18067, 20);
INSERT INTO ENROLLED VALUES (18068, 20);
INSERT INTO ENROLLED VALUES (18069, 20);
INSERT INTO ENROLLED VALUES (18070, 20);
INSERT INTO ENROLLED VALUES (18073, 30);
INSERT INTO ENROLLED VALUES (18074, 10);


CREATE TABLE HAVE(
  COURSE_ID NUMBER(2),
  SUBJECT_ID NUMBER(3),
  SUBJECT_NAME VARCHAR2(15)
);

INSERT INTO HAVE VALUES (40, 101, 'COA');
INSERT INTO HAVE VALUES (40, 102, 'DSA');
INSERT INTO HAVE VALUES (40, 103, 'DBMS');
INSERT INTO HAVE VALUES (40, 104, 'ML');
INSERT INTO HAVE VALUES (40, 105, 'BIG DATA');
INSERT INTO HAVE VALUES (40, 107, 'GRAPH THEORY');


CREATE TABLE SELECTED(
  STUDENT_ID NUMBER(5),
  SUBJECT_ID NUMBER(3),
  GRADE VARCHAR2(1)
);

INSERT INTO SELECTED VALUES (18071, 101, 'A');
INSERT INTO SELECTED VALUES (18071, 102, 'B');
INSERT INTO SELECTED VALUES (18071, 103, 'A');
INSERT INTO SELECTED VALUES (18071, 104, 'B');
INSERT INTO SELECTED VALUES (18071, 105, 'C');
INSERT INTO SELECTED VALUES (18071, 107, 'A');
INSERT INTO SELECTED VALUES (18073, 108, 'B');
INSERT INTO SELECTED VALUES (18074, 102, 'A');
```

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

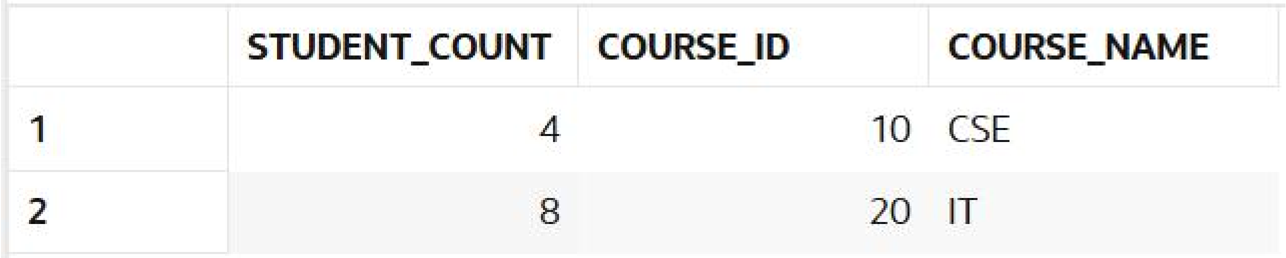
```
ALTER TABLE ENROLLED ADD (ENROLL_DATE DATE);

UPDATE ENROLLED SET ENROLL_DATE = TO_DATE('19-DEC-20', 'DD-MON-YY') WHERE STUDENT_ID =
18066;
UPDATE ENROLLED SET ENROLL_DATE = TO_DATE('17-JUN-19', 'DD-MON-YY') WHERE STUDENT_ID =
18067;
UPDATE ENROLLED SET ENROLL_DATE = TO_DATE('14-JUL-21', 'DD-MON-YY') WHERE STUDENT_ID =
18068;
UPDATE ENROLLED SET ENROLL_DATE = TO_DATE('21-JUN-18', 'DD-MON-YY') WHERE STUDENT_ID =
18069;
UPDATE ENROLLED SET ENROLL_DATE = TO_DATE('04-SEP-20', 'DD-MON-YY') WHERE STUDENT_ID =
18070;
UPDATE ENROLLED SET ENROLL_DATE = TO_DATE('11-AUG-21', 'DD-MON-YY') WHERE STUDENT_ID =
18073;
UPDATE ENROLLED SET ENROLL_DATE = TO_DATE('25-NOV-22', 'DD-MON-YY') WHERE STUDENT_ID =
18074;
```

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

```
ALTER TABLE COURSE
MODIFY COURSE_NAME VARCHAR(10) NOT NULL;
```

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

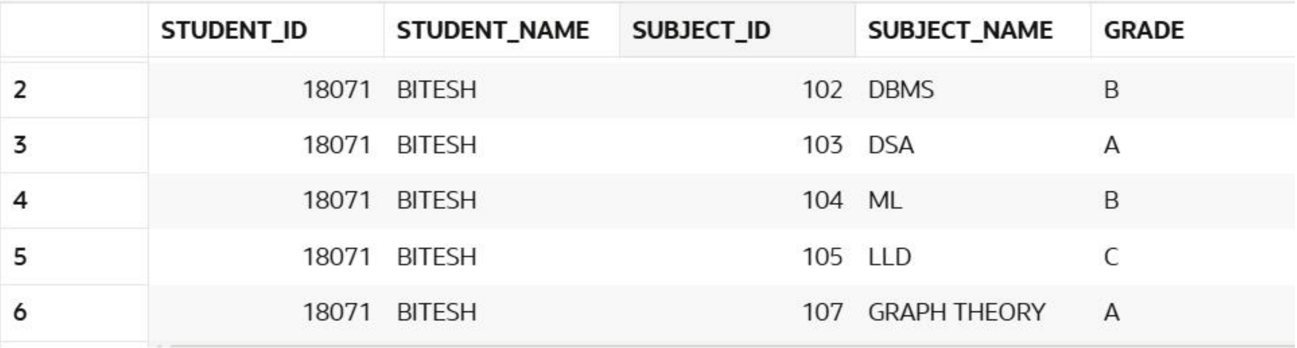
```
SELECT * FROM (
SELECT
COUNT(inner_query.COURSE_ID) AS STUDENT_COUNT,
inner_query.COURSE_ID, inner_query.COURSE_NAME
FROM (
SELECT
ENROLLED.COURSE_ID, COURSE.COURSE_NAME
FROM ENROLLED
JOIN COURSE ON ENROLLED.COURSE_ID = COURSE.COURSE_ID
) inner_query
GROUP BY inner_query.COURSE_ID, inner_query.COURSE_NAME
)
WHERE STUDENT_COUNT > 3;
```



	STUDENT_COUNT	COURSE_ID	COURSE_NAME
1	4	10	CSE
2	8	20	IT

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

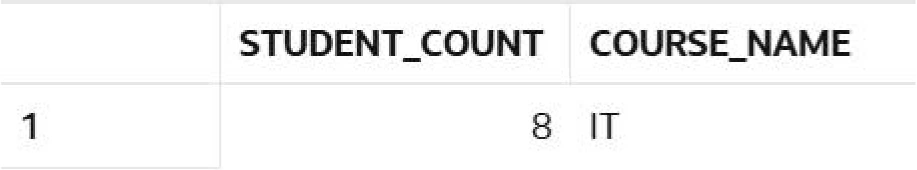
```
SELECT
STUDENT.STUDENT_ID, STUDENT.STUDENT_NAME, SUBJECTS.SUBJECT_ID, SUBJECTS.SUBJECT_NAME,
SELECTED.GRADE
FROM STUDENT
INNER JOIN SELECTED
ON STUDENT.STUDENT_ID = SELECTED.STUDENT_ID INNER JOIN SUBJECTS
ON SELECTED.SUBJECT_ID = SUBJECTS.SUBJECT_ID WHERE STUDENT.STUDENT_ID = &sid;
```



	STUDENT_ID	STUDENT_NAME	SUBJECT_ID	SUBJECT_NAME	GRADE
2	18071	BITESH	102	DBMS	B
3	18071	BITESH	103	DSA	A
4	18071	BITESH	104	ML	B
5	18071	BITESH	105	LLD	C
6	18071	BITESH	107	GRAPH THEORY	A

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

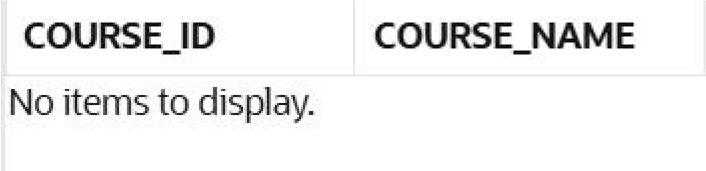
```
SELECT STUDENT_COUNT, COURSE_NAME FROM (
SELECT COUNT(COURSE_ID) AS STUDENT_COUNT ,COURSE_ID, COURSE_NAME FROM (
SELECT ENROLLED.COURSE_ID , COURSE.COURSE_NAME
FROM ENROLLED,COURSE
WHERE ENROLLED.COURSE_ID=COURSE.COURSE_ID) GROUP BY COURSE_ID,COURSE_NAME)
WHERE STUDENT_COUNT = (
SELECT MAX(STUDENT_COUNT) 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));
```



	STUDENT_COUNT	COURSE_NAME
1	8	IT

6. 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_ID IS NULL;
```



COURSE_ID	COURSE_NAME

No items to display.

7. Delete Course no 30 from COURSE table.

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

8. Rename the COURSE table as DEPARTMENT.

```
RENAME COURSE TO DEPARTMENT;
```

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

```
UPDATE SELECTED SET GRADE='B'
WHERE SUBJECT_ID =103 AND GRADE='A';
```

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

```
DELETE FROM STUDENT WHERE COURSE_ID = 20;
```

11. 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;
```