TASK 1

Project : Academic Management System

1. Database creation

create table StudentInfo(STU\_ID int primary key,STU\_NAME VARCHAR(50),DOB DATE,

PHONE\_NO BIGINT,EMAIL\_ID VARCHAR(50),ADDRESS VARCHAR(100));

create table CourseInfo(COURSE\_ID INT primary key,COURSE\_NAME VARCHAR(50),

INSTRUCTOR\_NAME VARCHAR(50));

create table EnrollmentInfo(ENROLLMENT\_ID INT,STU\_ID INT,COURSE\_ID INT,

ENROLL\_STATUS varchar(50), FOREIGN KEY (STU\_ID) REFERENCES StudentInfo(STU\_ID), FOREIGN KEY (COURSE\_ID) REFERENCES CourseInfo(COURSE\_ID));

2. Data creation

INSERT INTO StudentInfo (STU\_ID, STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID, ADDRESS)

VALUES

(1, 'John Smith', '2000-05-15', 1234567890, 'john@example.com', '123 Main St, Cityville'),

(2, 'Emily Johnson', '2001-08-22', 9876543210, 'emily@example.com', '456 Elm St, Townsville'),

(3, 'Michael Brown', '1999-12-10', 5556667777, 'michael@example.com', '789 Oak St, Villageton');

INSERT INTO CourseInfo (COURSE\_ID, COURSE\_NAME, INSTRUCTOR\_NAME)

VALUES

(101, 'Introduction to Computer Science', 'Dr. Smith'),

(102, 'Advanced Mathematics', 'Prof. Johnson'),

(103, 'Literature and Writing', 'Dr. Martinez');

INSERT INTO EnrollmentInfo (ENROLLMENT\_ID, STU\_ID, COURSE\_ID, ENROLL\_STATUS)

VALUES

(1, 1, 101, 'Enrolled'),

(2, 2, 102, 'Enrolled'),

(3, 3, 103, 'Enrolled'),

(4, 1, 102, 'Enrolled'),

(5, 2, 101, 'Enrolled');

3. Retrieve the student information

A.\*/

SELECT

s.STU\_NAME,

s.PHONE\_NO,

s.EMAIL\_ID,

e.ENROLL\_STATUS

FROM

StudentInfo s

JOIN

EnrollmentInfo e ON s.STU\_ID = e.STU\_ID;

–B.

SELECT

ci.COURSE\_NAME,

ci.INSTRUCTOR\_NAME

FROM

EnrollmentInfo ei

JOIN

CourseInfo ci ON ei.COURSE\_ID = ci.COURSE\_ID

WHERE

ei.STU\_ID = 2 AND ei.ENROLL\_STATUS = 'Enrolled';

–C.

SELECT

COURSE\_NAME,

INSTRUCTOR\_NAME

FROM

CourseInfo;

–D.

SELECT

COURSE\_ID,

COURSE\_NAME,

INSTRUCTOR\_NAME

FROM

CourseInfo

WHERE

COURSE\_NAME = ‘Literature and Writing’;

–E.

SELECT

COURSE\_ID,

COURSE\_NAME,

INSTRUCTOR\_NAME

FROM

CourseInfo

WHERE

COURSE\_ID IN (101, 102, 103);

4. Reporting and analytics

A.\*/

SELECT

ci.COURSE\_NAME,

ci.INSTRUCTOR\_NAME,

COUNT(ei.STU\_ID) AS ENROLLED\_STUDENTS

FROM

CourseInfo ci

LEFT JOIN

EnrollmentInfo ei ON ci.COURSE\_ID = ei.COURSE\_ID

GROUP BY

ci.COURSE\_NAME, ci.INSTRUCTOR\_NAME;

–B.

SELECT

si.STU\_NAME

FROM

StudentInfo si

JOIN

EnrollmentInfo ei ON si.STU\_ID = ei.STU\_ID

WHERE

ei.COURSE\_ID = 102;

–C.

SELECT

ci.INSTRUCTOR\_NAME,

COUNT(DISTINCT si.STU\_ID) AS ENROLLED\_STUDENTS\_COUNT

FROM

CourseInfo ci

JOIN

EnrollmentInfo ei ON ci.COURSE\_ID = ei.COURSE\_ID

JOIN

StudentInfo si ON ei.STU\_ID = si.STU\_ID

GROUP BY

ci.INSTRUCTOR\_NAME;

–D.

SELECT

si.STU\_ID,

si.STU\_NAME,

COUNT(DISTINCT ei.COURSE\_ID) AS ENROLLED\_COURSES\_COUNT

FROM

StudentInfo si

JOIN

EnrollmentInfo ei ON si.STU\_ID = ei.STU\_ID WHERE ei.ENROLL\_STATUS = 'Enrolled'

GROUP BY

si.STU\_ID, si.STU\_NAME

HAVING

COUNT(DISTINCT ei.COURSE\_ID) > 1;

–E.

SELECT

ci.COURSE\_ID,

ci.COURSE\_NAME,

ci.INSTRUCTOR\_NAME,

COUNT(ei.STU\_ID) AS ENROLLED\_STUDENTS\_COUNT

FROM

CourseInfo ci

JOIN

EnrollmentInfo ei ON ci.COURSE\_ID = ei.COURSE\_ID

GROUP BY

ci.COURSE\_ID, ci.COURSE\_NAME, ci.INSTRUCTOR\_NAME

ORDER BY

ENROLLED\_STUDENTS\_COUNT DESC;