**TASK-1 ACADEMIC MANAGEMENT SYSTEM**

**1) Database Creation:**

**a) Create the Studentlnfo table with columns STU\_ ID,STU\_NAME,DOB, PHONE\_NO, EMAIL\_ID,ADDRESS**

-- Create StudentInfo table

CREATE TABLE StudentInfo (

STU\_ID INT PRIMARY KEY,

STU\_NAME VARCHAR(50),

DOB DATE,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(50),

ADDRESS VARCHAR(100)

);

**b) Create the Courseslnfo table with columns COURSE\_ID, COURSE\_NAME,COURSE\_INSTRUCTOR NAME.**

-- Create CoursesInfo table

CREATE TABLE CoursesInfo (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(50),

COURSE\_INSTRUCTOR\_NAME VARCHAR(50)

);

**c) Create the EnrollmentInfo withcolumns ENROLLMENT\_ ID, STU\_ ID,COURSE\_ID,**

**ENROll\_STATUS(EnroHed/Not Enrolled).The FOREIGN KEY constraint In the Enrollmentlnfo**

**table references the STU\_ID column in the Studentlnfotable and the COURSE\_ID column in the Coursestnfo table**

-- Create EnrollmentInfo table

CREATE TABLE EnrollmentInfo (

ENROLLMENT\_ID INT PRIMARY KEY,

STU\_ID INT,

COURSE\_ID INT,

ENROLL\_STATUS VARCHAR(20),

FOREIGN KEY (STU\_ID) REFERENCES StudentInfo(STU\_ID),

FOREIGN KEY (COURSE\_ID) REFERENCES CoursesInfo(COURSE\_ID)

);

**2) Data Creation:**

**Insert some sample data for Studentlnfo table, Courseslnfotable, Enrollmentlnfo with respective fields.**

-- Insert sample data into StudentInfo table

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

VALUES

(1, 'John Doe', '1998-05-10', '1234567890', 'john.doe@example.com', '123 Main St'),

(2, 'Jane Smith', '1999-09-15', '9876543210', 'jane.smith@example.com', '456 Elm St');

-- Insert sample data into CoursesInfo table

INSERT INTO CoursesInfo (COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME)

VALUES

(1, 'Mathematics', 'Professor Johnson'),

(2, 'History', 'Professor Thompson');

-- Insert sample data into EnrollmentInfo table

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

VALUES

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

(2, 1, 2, 'Enrolled'),

(3, 2, 1, 'Enrolled');

**3) Retrieve the Student information**

**a) Write a query to retrieve student details, such as student name, contact information and**

**Enrollment status.**

SELECT

SI.STU\_NAME,

SI.PHONE\_NO,

SI.EMAIL\_ID,

SI.ADDRESS,

EI.ENROLL\_STATUS

FROM

StudentInfo SI

JOIN

EnrollmentInfo EI ON SI.STU\_ID = EI.STU\_ID;



**b) Write a query to retrieve a list of courses in which a specific student Is enrolled.**

SELECT

CI.COURSE\_NAME

FROM

CoursesInfo CI

JOIN

EnrollmentInfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

JOIN

StudentInfo SI ON EI.STU\_ID = SI.STU\_ID

WHERE

SI.STU\_NAME = 'John Doe';



**c) Write a query to retrieve course information, including course name. instructor information.**

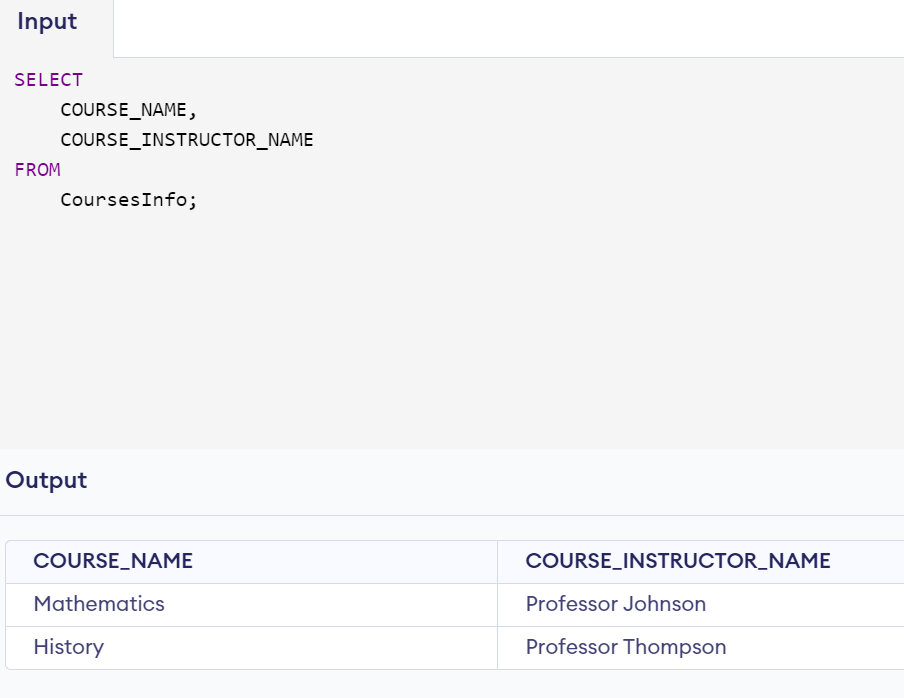
SELECT

COURSE\_NAME,

COURSE\_INSTRUCTOR\_NAME

FROM

CoursesInfo;



**d) Write a query to retrieve course information for a specific course**

SELECT

COURSE\_NAME,

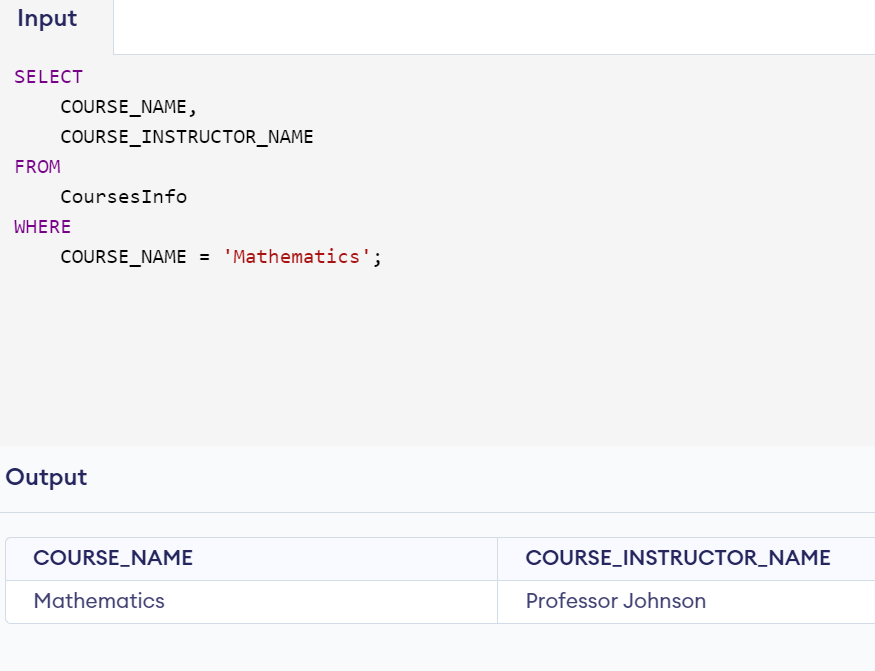
COURSE\_INSTRUCTOR\_NAME

FROM

CoursesInfo

WHERE

COURSE\_NAME = 'Mathematics';



**e) Write a query to retrieve course information for multiple courses.**

SELECT

COURSE\_NAME,

COURSE\_INSTRUCTOR\_NAME

FROM

CoursesInfo

WHERE

COURSE\_NAME IN ('Mathematics', 'History', 'Science');



**4. Reporting and Analytics**

**a) Write a query to retrieve the number of students enroll in each course**

SELECT

CI.COURSE\_NAME,

COUNT(EI.STU\_ID) AS ENROLLMENT\_COUNT

FROM

CoursesInfo CI

JOIN

EnrollmentInfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

GROUP BY

CI.COURSE\_NAME;



**b) Write a query to retrieve the list of students enrolled in a specific course**

SELECT

SI.STU\_NAME,

SI.PHONE\_NO,

SI.EMAIL\_ID,

SI.ADDRESS

FROM

StudentInfo SI

JOIN

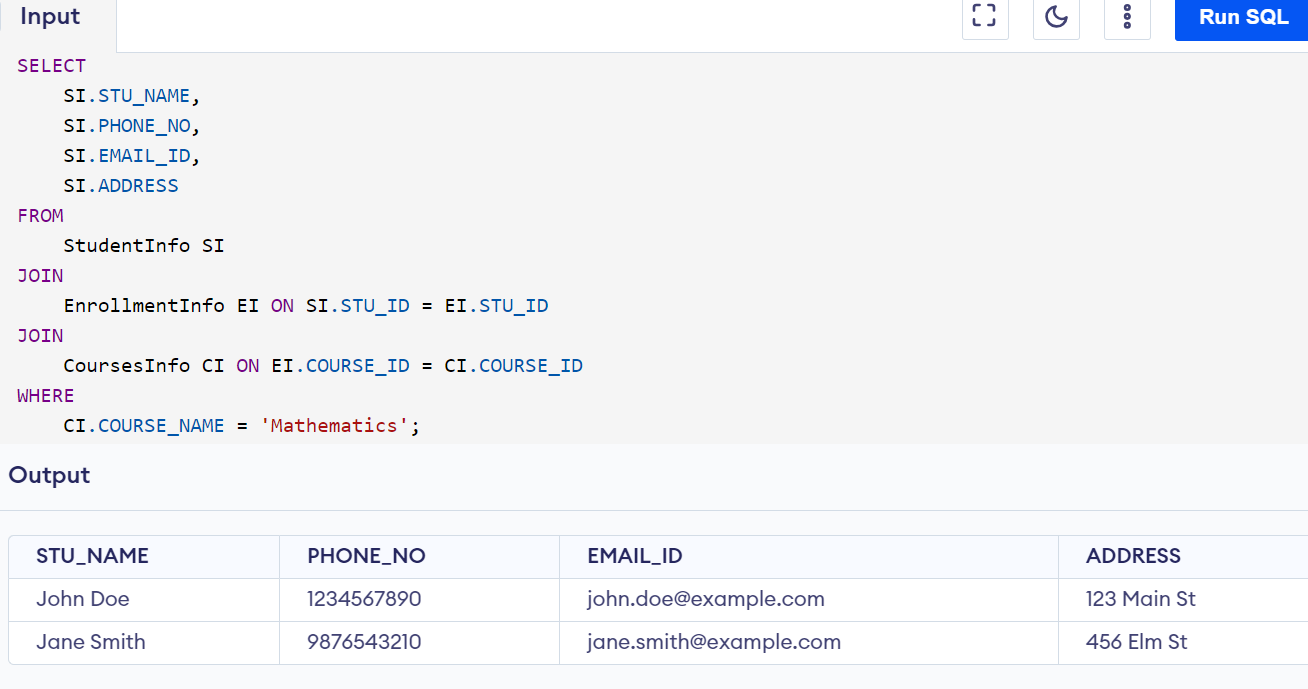
EnrollmentInfo EI ON SI.STU\_ID = EI.STU\_ID

JOIN

CoursesInfo CI ON EI.COURSE\_ID = CI.COURSE\_ID

WHERE

CI.COURSE\_NAME = 'Mathematics';



**c) Write a query to retrieve the count of enrolled students for each instructor.**

SELECT

CI.COURSE\_INSTRUCTOR\_NAME,

COUNT(EI.STU\_ID) AS ENROLLMENT\_COUNT

FROM

CoursesInfo CI

JOIN

EnrollmentInfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

GROUP BY

CI.COURSE\_INSTRUCTOR\_NAME;



**d) Write a query to retrieve the list of students who are enrolled in multiple courses**

SELECT

SI.STU\_ID,

SI.STU\_NAME

FROM

StudentInfo SI

JOIN

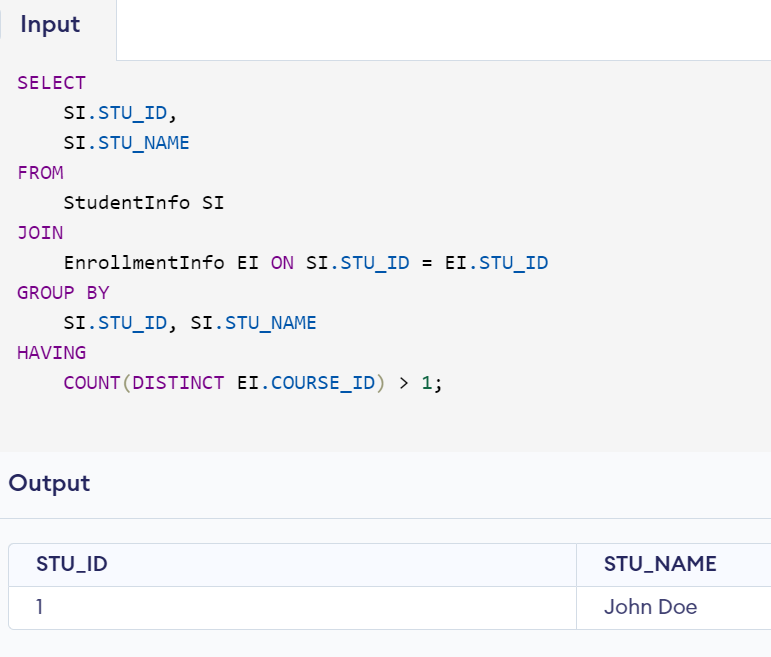
EnrollmentInfo EI ON SI.STU\_ID = EI.STU\_ID

GROUP BY

SI.STU\_ID, SI.STU\_NAME

HAVING

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



**e) Write a query to retrieve the course that have the highest number of enrolled students(arranging from highest to lowest)**

SELECT

CI.COURSE\_NAME,

COUNT(EI.STU\_ID) AS ENROLLMENT\_COUNT

FROM

CoursesInfo CI

JOIN

EnrollmentInfo EI ON CI.COURSE\_ID = EI.COURSE\_ID

GROUP BY

CI.COURSE\_NAME

ORDER BY

ENROLLMENT\_COUNT DESC;

