**Taskl**

**Project Title: Academic Management System ( using SQL) Project Description:**

Design and develop an Academic Management System using SQL. The projects should involve three tables 1.Studentlnfo 2. Courseslnfo 3.Enrollmentlnfo .The Aim is to create a system that allows for managing student information and course enrollment. The project will include the following tasks:

1. **Database Creation:**
2. Create the Studentlnfo table with columns STU\_ ID, STU\_NAME, DOB, PHONE\_NO, EMAIL\_ID,ADDRESS.

CREATE TABLE StudentInfo (

STU\_ID INT PRIMARY KEY,

STU\_NAME VARCHAR(255),

DOB DATE,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(255),

ADDRESS VARCHAR(255)

);

1. Create the Courseslnfo table with columns COURSE\_ID, COURSE\_NAM E,COURSE\_INSTRUCTOR NAME.

CREATE TABLE CoursesInfo (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(255),

COURSE\_INSTRUCTOR\_NAME VARCHAR(255)

);

1. Create the Enrollmentlnfo with columns ENROLLMENT\_ID, STU\_ ID, COURSE\_ID,

ENROLL\_STATUS(Enrolled/Not Enrolled). The FOREIGN KEY constraint in the Enrollmentlnfo

table references the STU-ID column in the Studentlnfo table and the COURSE-ID column in the Courseslnfo table.

CREATE TABLE EnrollmentInfo (

ENROLLMENT\_ID INT PRIMARY KEY,

STU\_ID INT,

COURSE\_ID INT,

ENROLL\_STATUS varchar(255) CHECK (ENROLL\_STATUS IN ('Enrolled', 'Not Enrolled')),

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

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

);

1. **Data Creation:**

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

-- Insert data into StudentInfo table

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

VALUES

(1, 'student 1', '2000-01-15', '1234567890', 'student1@gmail.com','no.1 Malleshwaram Bengaluru'),

(2, 'student 2', '1999-03-20', '987-654-3210', 'student2@gmail.com', 'no.2, Malleshwaram Bengaluru');

-- Insert data into table

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

VALUES

(101,'Math','Prof.1'),

(102, 'History', 'Prof.2 ');

-- Insert data into EnrollmentInfo table

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

VALUES

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

(2, 1, 102, 'Enrolled'),

(3, 2, 101, 'Enrolled'),

(4, 2, 102, ‘Not Enrolled');

1. **Retrieve the Student Information**
   1. Write a query to retrieve student details, such as student name, contact informations, and Enrollment status.

Select a.STU\_NAME

, a.PHONE\_NO

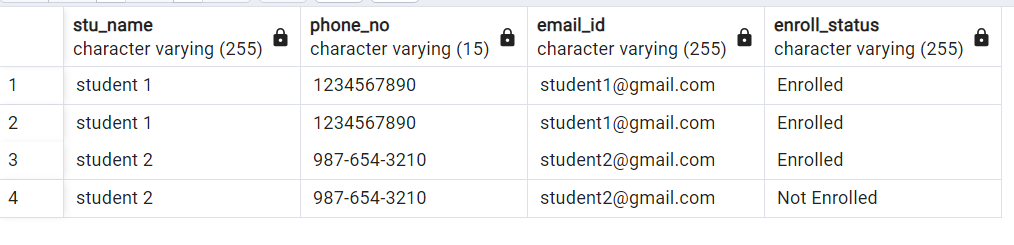
,a. EMAIL\_ID

, b.ENROLL\_STATUS

From StudentInfo a

Join EnrollmentInfo b

On a. STU\_ID = b. STU\_ID;



* 1. Write a query to retrieve a list of courses in which a specific student is enrolled.

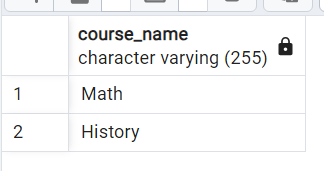
Select a.COURSE\_NAME

From CoursesInfo a

Join StudentInfo b

On a. STU\_ID = b. STU\_ID

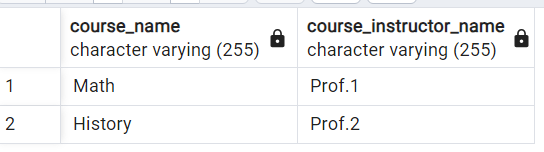
Where STU\_ID = 1;



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

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo ;

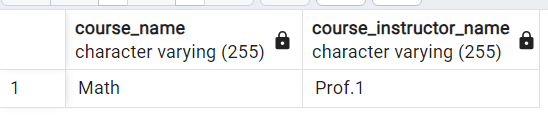


* 1. Write a query to retrieve course information for a specific course .

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

Where COURSE\_ID = 101;

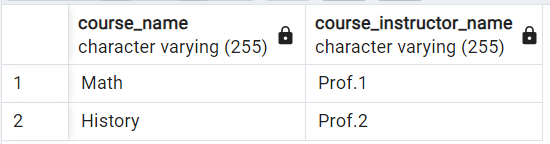


* 1. Write a query to retrieve course information for multiple courses.

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

Where COURSE\_ID in (101,102);



* 1. Test the queries to ensure accurate retrieval of student information.( execute the queries and verify the results against the expected output.)

Screenshot of results have been uploaded

**4. Reporting and Analytics {Using joining queries)**

1. Write a query to retrieve the number of students enrolled in each course

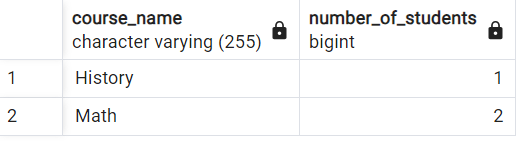
SELECT c.COURSE\_NAME, COUNT(e.STU\_ID) AS Number\_of\_Students

FROM CoursesInfo c

LEFT JOIN EnrollmentInfo e ON c.COURSE\_ID = e.COURSE\_ID

where e.enroll\_status = 'Enrolled'

GROUP BY c.COURSE\_NAME;



1. Write a query to retrieve the list of students enrolled in a specific course

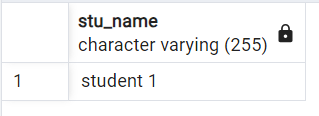
SELECT s.STU\_NAME

FROM StudentInfo s

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

WHERE e.COURSE\_ID = 102

And e.enroll\_status = 'Enrolled';



1. Write a query to retrieve the count of enrolled students for each instructor.

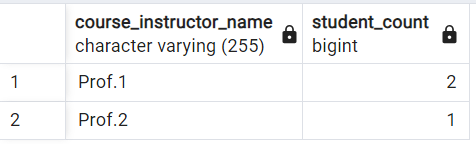
SELECT CI.COURSE\_INSTRUCTOR\_NAME, COUNT(DISTINCT EI.STU\_ID) AS Student\_Count

FROM CoursesInfo CI

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

Where ENROLL\_STATUS = 'Enrolled'

GROUP BY CI.COURSE\_INSTRUCTOR\_NAME;



1. Write a query to retrieve the list of students who are enrolled in multiple courses

SELECT s.STU\_NAME

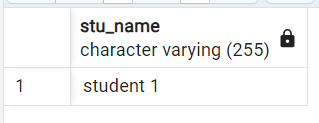
FROM StudentInfo s

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

Where ENROLL\_STATUS = 'Enrolled'

GROUP BY s.STU\_NAME

HAVING COUNT(e.ENROLLMENT\_ID) > 1;



1. Write a query to retrieve the courses that have the highest number of enrolled students(arranging from highest to lowest)

SELECT c.COURSE\_NAME, COUNT(e.STU\_ID) AS Number\_of\_Students

FROM CoursesInfo c

LEFT JOIN EnrollmentInfo e ON c.COURSE\_ID = e.COURSE\_ID

Where ENROLL\_STATUS = 'Enrolled'

GROUP BY c.COURSE\_NAME

ORDER BY Number\_of\_Students DESC;

