**Task1**

**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.StudentInfo 2. CoursesInfo 3.EnrollmentInfo. 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:**

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

**Query:**

CREATE TABLE StudentInfo (

STU\_ID INT PRIMARY KEY,

STU\_NAME VARCHAR(100) NOT NULL,

DOB DATE,

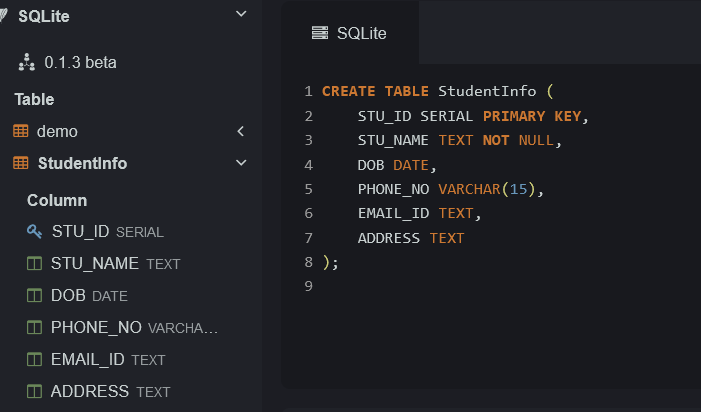
PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(100),

ADDRESS VARCHAR(255)

);

**Output:**



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

**Query:**

CREATE TABLE CoursesInfo (

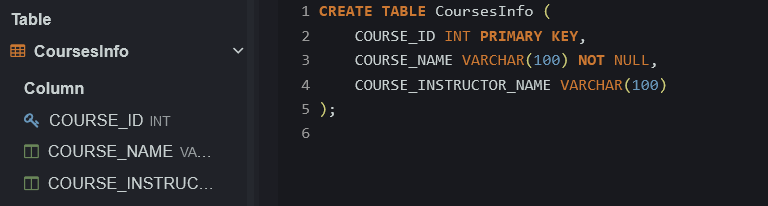
COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(100) NOT NULL,

COURSE\_INSTRUCTOR\_NAME VARCHAR(100)

);

**Output:**



c) Create the EnrollmentInfo with columns ENROLLMENT\_ID, STU\_ ID, COURSE\_ID,

ENROLL\_STATUS(Enrolled/Not Enrolled). The FOREIGN KEY constraint in the EnrollmentInfo table references the STU\_ID column in the StudentInfo table and the COURSE\_ID column in the CoursesInfo table.

**Query:**

CREATE TABLE EnrollmentInfo (

ENROLLMENT\_ID INT PRIMARY KEY,

STU\_ID INT,

COURSE\_ID INT,

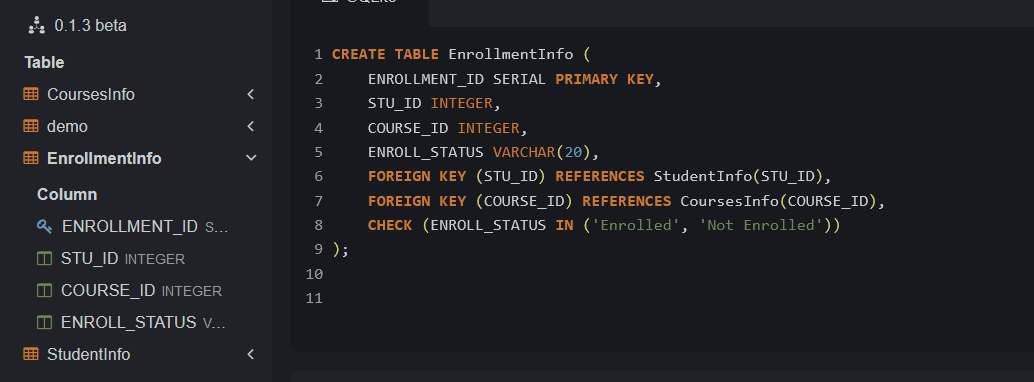
ENROLL\_STATUS VARCHAR(15) CHECK (ENROLL\_STATUS IN ('Enrolled', 'Not Enrolled')),

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

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

);

**Output:**



**2. Data Creation:**

Insert some sample data for StudentInfo table , CoursesInfo table, EnrollmentInfo with respective fields.

**Query:**

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

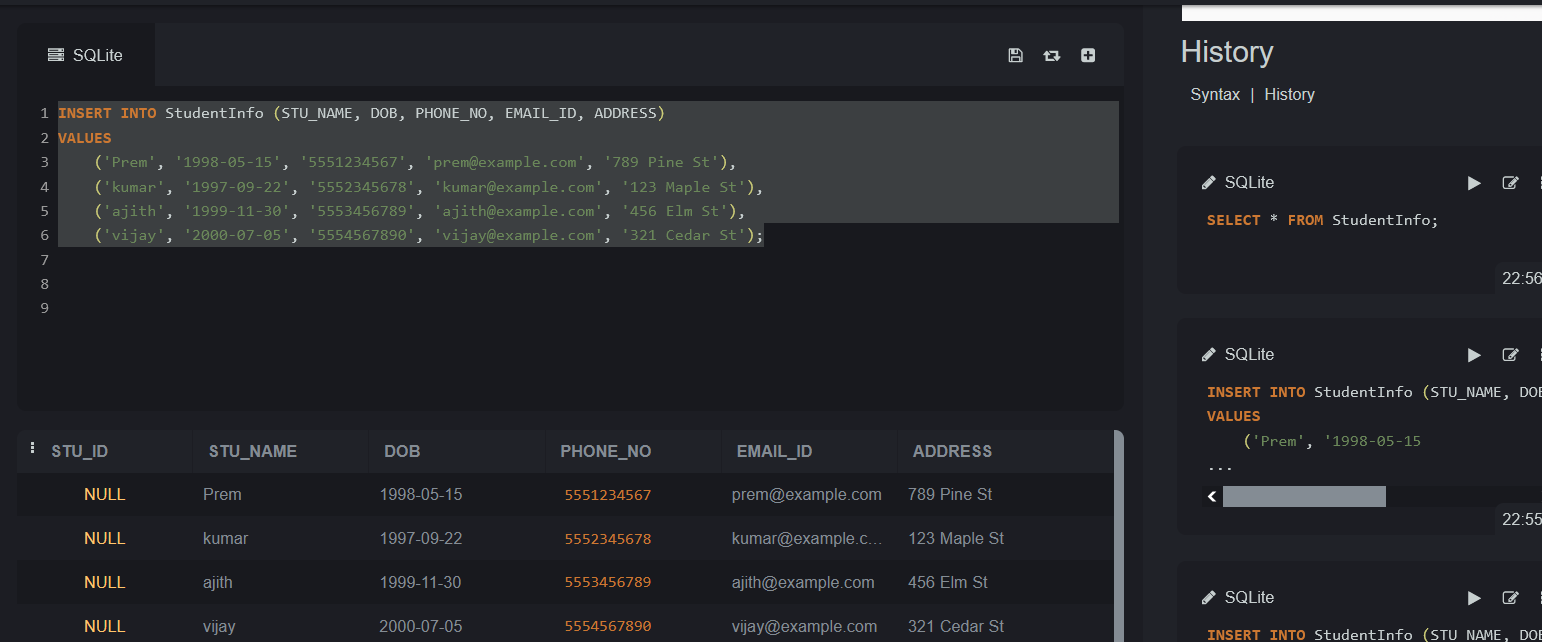
VALUES

(‘1’,'Prem', '1998-05-15', '5551234567', 'prem@example.com', '789 Pine St'),

(‘2’,'kumar', '1997-09-22', '5552345678', 'kumar@example.com', '123 Maple St'),

(‘3’,'ajith', '1999-11-30', '5553456789', 'ajith@example.com', '456 Elm St'),

(‘4’,'vijay', '2000-07-05', '5554567890', 'vijay@example.com', '321 Cedar St');

**Output:**

**Insert Sample Data into CoursesInfo Table:**

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

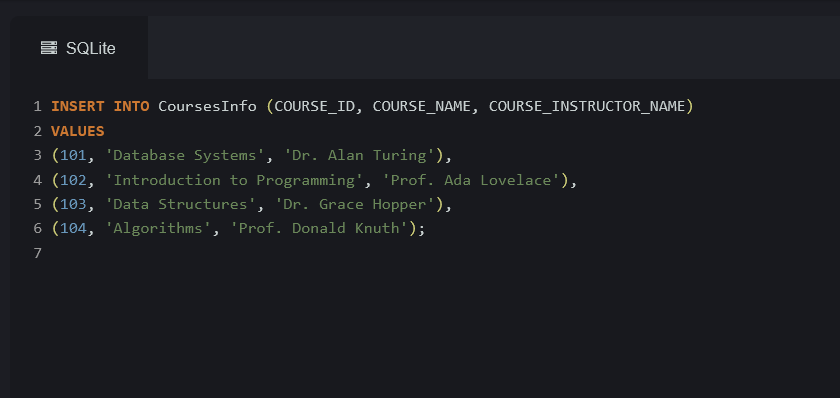
VALUES

(101, 'Database Systems', 'Dr. Alan Turing'),

(102, 'Introduction to Programming', 'Prof. Ada Lovelace'),

(103, 'Data Structures', 'Dr. Grace Hopper'),

(104, 'Algorithms', 'Prof. Donald Knuth');



**Insert Sample Data into EnrollmentInfo Table**

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

VALUES

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

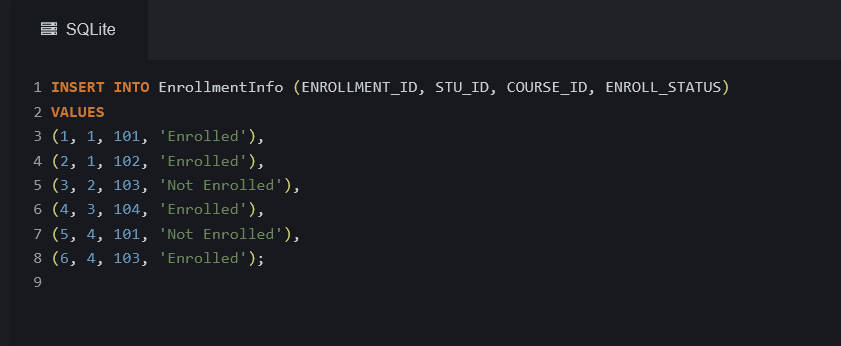
(2, 1, 102, 'Enrolled'),

(3, 2, 103, 'Not Enrolled'),

(4, 3, 104, 'Enrolled'),

(5, 4, 101, 'Not Enrolled'),

(6, 4, 103, 'Enrolled');



**3) Retrieve the Student Information**

a) Write a query to retrieve student details, such as student name, contact information’s, and Enrollment status.

**Query:**

SELECT

si.STU\_NAME AS StudentName,

si.PHONE\_NO AS Phone,

si.EMAIL\_ID AS Email,

ei.ENROLL\_STATUS AS EnrollmentStatus

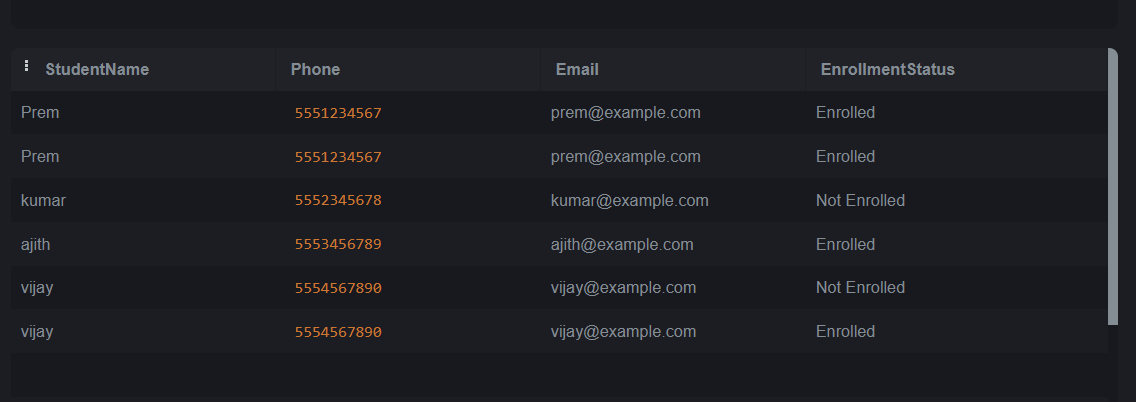
FROM

StudentInfo si

JOIN

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

**Output:**



b) Write a query to retrieve a list of courses in which a specific student is enrolled

**Query:**

SELECT

ci.COURSE\_NAME AS CourseName,

ci.COURSE\_INSTRUCTOR\_NAME AS InstructorName

FROM

EnrollmentInfo ei

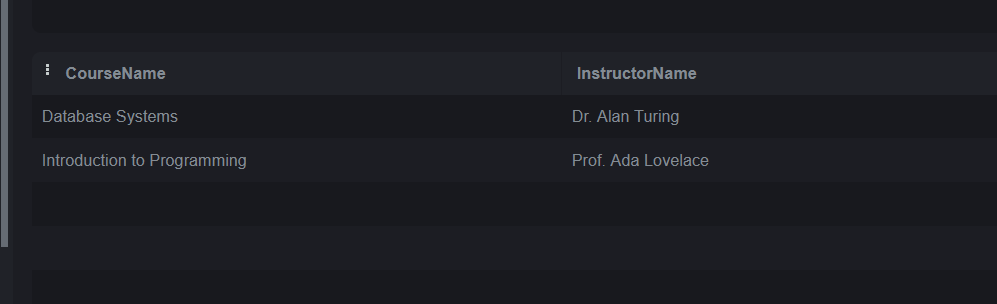
JOIN

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

WHERE

ei.STU\_ID = 1;

**output:**



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

**Query:**

SELECT

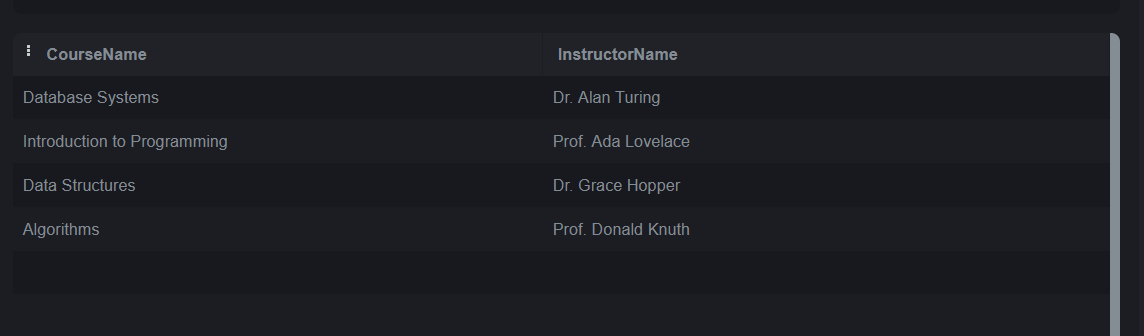
COURSE\_NAME AS CourseName,

COURSE\_INSTRUCTOR\_NAME AS InstructorName

FROM

CoursesInfo;

Output:



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

**Query:**

SELECT

COURSE\_NAME AS CourseName,

COURSE\_INSTRUCTOR\_NAME AS InstructorName

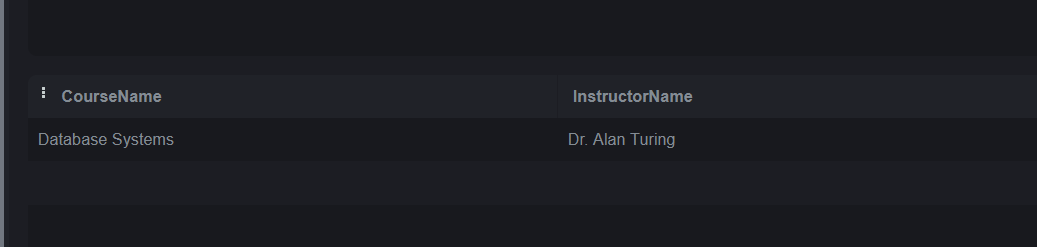
FROM

CoursesInfo

WHERE

COURSE\_ID = 101;

**Output:**



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

**Query:**

SELECT

COURSE\_ID,

COURSE\_NAME AS CourseName,

COURSE\_INSTRUCTOR\_NAME AS InstructorName

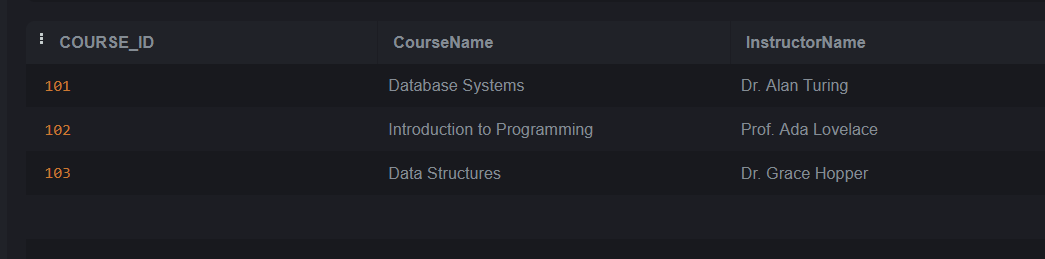
FROM

CoursesInfo

WHERE

COURSE\_ID IN (101, 102, 103); -- Replace with the actual course IDs

**Output:**



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

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

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

**Query:**

SELECT

ci.COURSE\_NAME AS CourseName,

COUNT(ei.STU\_ID) AS NumberOfStudents

FROM

EnrollmentInfo ei

JOIN

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

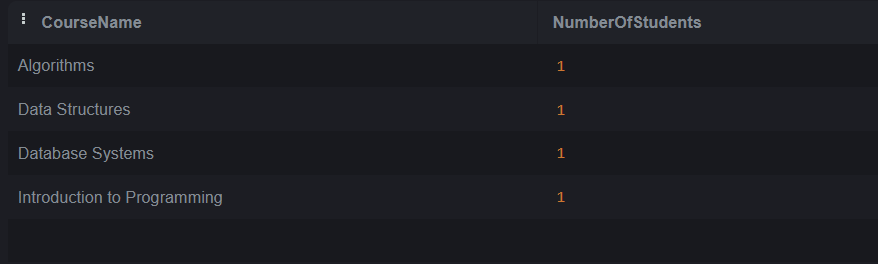
WHERE

ei.ENROLL\_STATUS = 'Enrolled'

GROUP BY

ci.COURSE\_NAME;

**output:**



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

SELECT

si.STU\_ID AS StudentID,

si.STU\_NAME AS StudentName,

si.PHONE\_NO AS PhoneNumber,

si.EMAIL\_ID AS Email

FROM

EnrollmentInfo ei

JOIN

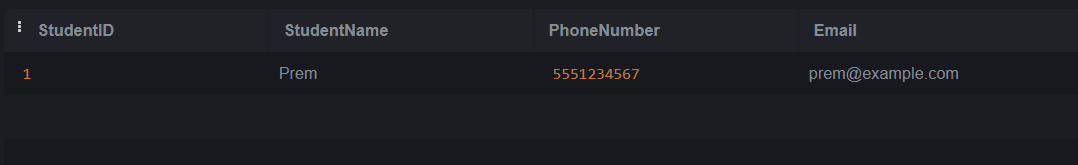
StudentInfo si ON ei.STU\_ID = si.STU\_ID

WHERE

ei.COURSE\_ID = 101 AND

ei.ENROLL\_STATUS = 'Enrolled';

output:



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

SELECT

ci.COURSE\_INSTRUCTOR\_NAME AS InstructorName,

COUNT(ei.STU\_ID) AS NumberOfEnrolledStudents

FROM

EnrollmentInfo ei

JOIN

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

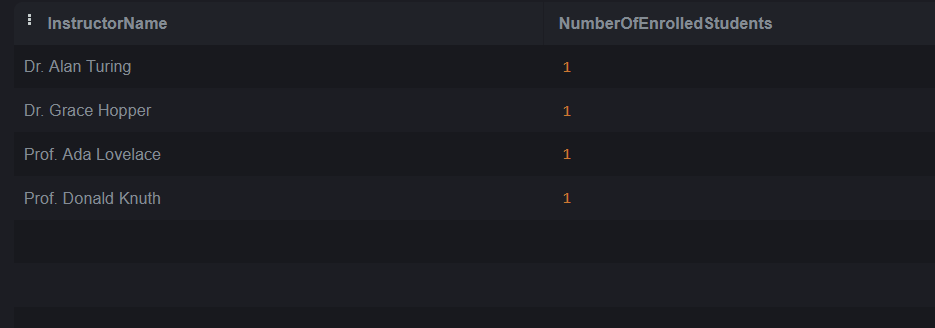
WHERE

ei.ENROLL\_STATUS = 'Enrolled'

GROUP BY

ci.COURSE\_INSTRUCTOR\_NAME;

**Output:**



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

SELECT

si.STU\_ID AS StudentID,

si.STU\_NAME AS StudentName,

COUNT(DISTINCT ei.COURSE\_ID) AS NumberOfCoursesEnrolled

FROM

EnrollmentInfo ei

JOIN

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

WHERE

ei.ENROLL\_STATUS = 'Enrolled'

GROUP BY

si.STU\_ID, si.STU\_NAME

HAVING

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

**Output:**



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

code:

SELECT

ci.COURSE\_NAME AS CourseName,

COUNT(ei.STU\_ID) AS NumberOfStudents

FROM

EnrollmentInfo ei

JOIN

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

WHERE

ei.ENROLL\_STATUS = 'Enrolled'

GROUP BY

ci.COURSE\_NAME

ORDER BY

NumberOfStudents DESC;

**Output:**

