# Capstone Task 1

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

## Database Creation

### Create the StudentInfo table with columns STU\_ ID, STU\_NAME, DOB, PHONE\_NO,

### EMAIL\_ID, ADDRESS.

CREATE TABLE StudentInfo (

STU\_ID INT PRIMARY KEY,

STU\_NAME VARCHAR(100),

DOB DATE,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(100),

ADDRESS VARCHAR(255)

);

### Create the CoursesInfo table with columns COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR NAME

CREATE TABLE CoursesInfo (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(100),

COURSE\_INSTRUCTOR\_NAME VARCHAR(100)

);

### 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.

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 StudentInfo table , CoursesInfo table, EnrollmentInfo with

### respective fields.

For StudentInfo Table

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

(1, 'Alice Smith', '1998-05-12', '123-456-7890', 'alice.smith@example.com', '123 Maple Street'),

(2, 'Bob Johnson', '1997-08-23', '234-567-8901', 'bob.johnson@example.com', '456 Oak Avenue'),

(3, 'Charlie Davis', '1999-12-02', '345-678-9012', 'charlie.davis@example.com', '789 Pine Road'),

(4, 'Diana Evans', '2000-03-15', '456-789-0123', 'diana.evans@example.com', '101 Birch Lane'),

(5, 'Eve Foster', '1998-07-29', '567-890-1234', 'eve.foster@example.com', '202 Cedar Boulevard');

For CourseInfo Table

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

(101, 'Mathematics', 'Dr. Alan Turing'),

(102, 'Physics', 'Dr. Marie Curie'),

(103, 'Chemistry', 'Dr. Rosalind Franklin'),

(104, 'Biology', 'Dr. Charles Darwin'),

(105, 'Computer Science', 'Dr. Ada Lovelace');

For 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, 2, 104, 'Enrolled'),

(5, 3, 105, 'Enrolled'),

(6, 3, 101, 'Not Enrolled'),

(7, 4, 102, 'Enrolled'),

(8, 4, 103, 'Enrolled'),

(9, 5, 104, 'Not Enrolled'),

(10, 5, 105, 'Enrolled');

## 3. Retrieve the Student Information

### a. Write a query to retrieve student details, such as student name, contact informations, and Enrollment status.

Query:

SELECT

SI.STU\_NAME

,SI.PHONE\_NO

,SI.EMAIL\_ID

,SI.ADDRESS

,EI.COURSE\_ID

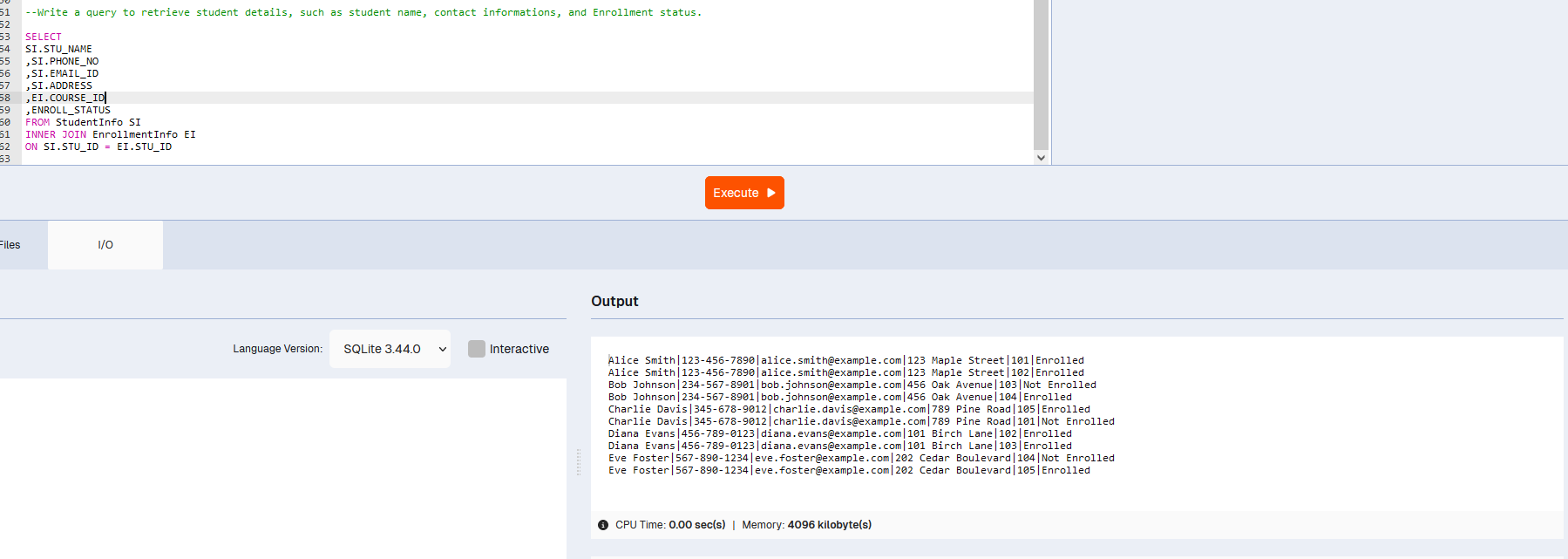
,ENROLL\_STATUS

FROM StudentInfo SI

INNER 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\_ID

, CI.COURSE\_NAME

, CI.COURSE\_INSTRUCTOR\_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 = 'Alice Smith' AND EI.ENROLL\_STATUS = 'Enrolled';

Output:



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

Query:

SELECT

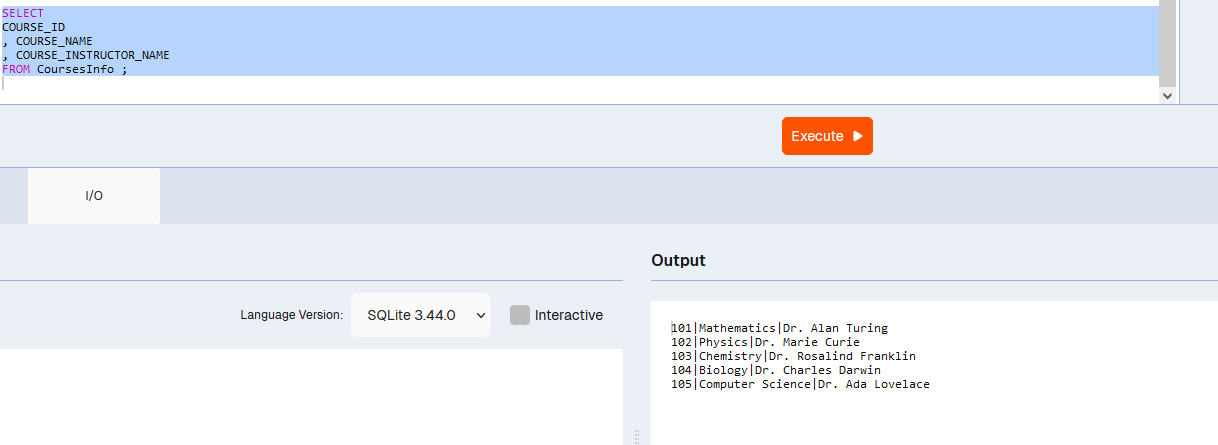
COURSE\_ID

, COURSE\_NAME

, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo ;

Output:



### Write a query to retrieve course information for a specific course

Query:

SELECT

COURSE\_ID

, COURSE\_NAME

, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

WHERE COURSE\_ID = 101;

OR

SELECT

COURSE\_ID

, COURSE\_NAME

, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

WHERE COURSE\_NAME = 'Mathematics';

Output:



### Write a query to retrieve course information for multiple courses

Query:

Using Course id

SELECT

COURSE\_ID

, COURSE\_NAME

, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

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

OR

Using Course name

SELECT

COURSE\_ID

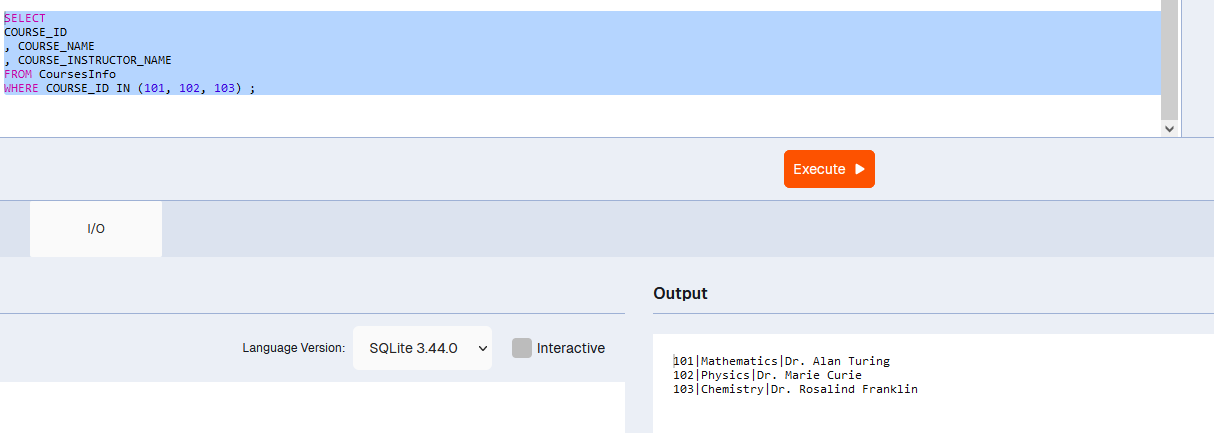
, COURSE\_NAME

, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

WHERE COURSE\_NAME IN ('Mathematics', 'Physics', 'Chemistry') ;

Output:



## Reporting and Analytics (Using joining queries)

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

Query:

SELECT

CI.COURSE\_ID

,CI.COURSE\_NAME

,COUNT(EI.STU\_ID) AS NO\_OF\_STUDENT

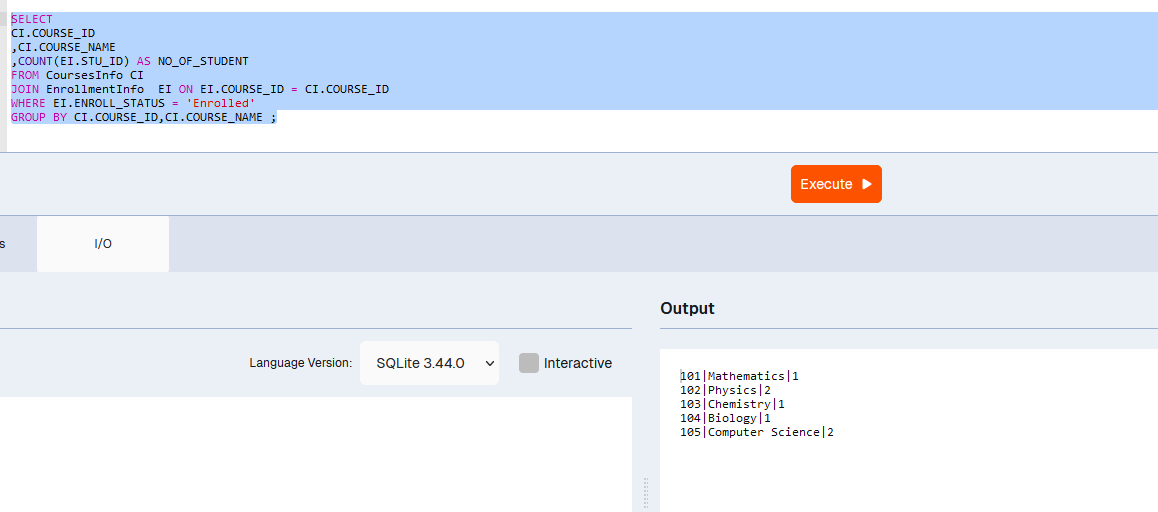
FROM CoursesInfo CI

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

WHERE EI.ENROLL\_STATUS = 'Enrolled'

GROUP BY CI.COURSE\_ID,CI.COURSE\_NAME ;

Output:



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

QUERY:

Using Course name

SELECT

s.STU\_ID,

s.STU\_NAME,

s.PHONE\_NO,

s.EMAIL\_ID,

s.ADDRESS

FROM CoursesInfo CI

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

JOIN StudentInfo s ON s.STU\_ID = EI. STU\_ID

WHERE EI.ENROLL\_STATUS = 'Enrolled' AND CI. COURSE\_NAME = 'Mathematics';

Using Course id

SELECT

s.STU\_ID,

s.STU\_NAME,

s.PHONE\_NO,

s.EMAIL\_ID,

s.ADDRESS

FROM

StudentInfo s

JOIN

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

WHERE

e.COURSE\_ID = 101 AND e.ENROLL\_STATUS = 'Enrolled';

Output:



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

QUERY:

SELECT

c.COURSE\_INSTRUCTOR\_NAME

, COUNT(s.STU\_ID) AS STUDENT\_COUNT

FROM

StudentInfo s

JOIN

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

JOIN

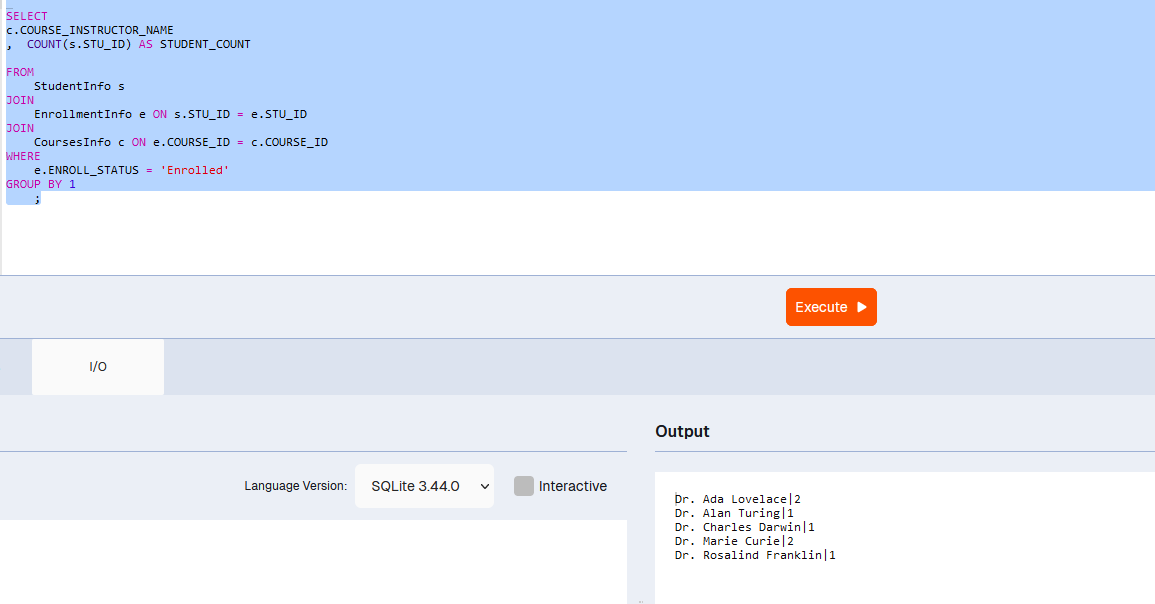
CoursesInfo c ON e.COURSE\_ID = c.COURSE\_ID

WHERE

e.ENROLL\_STATUS = 'Enrolled'

GROUP BY 1;

Output:



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

QUERY:

SELECT

s.STU\_ID,

s.STU\_NAME,

s.PHONE\_NO,

s.EMAIL\_ID,

s.ADDRESS

FROM

StudentInfo s

JOIN

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

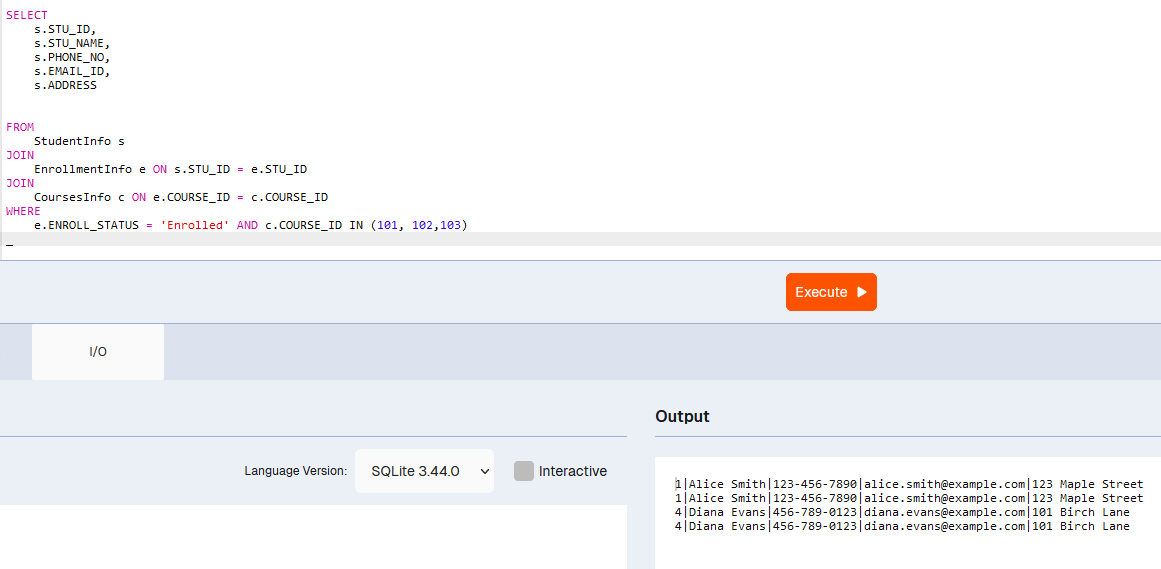
JOIN

CoursesInfo c ON e.COURSE\_ID = c.COURSE\_ID

WHERE

e.ENROLL\_STATUS = 'Enrolled' AND c.COURSE\_ID IN (101, 102,103)

Output:



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

QUERY:

SELECT

c. COURSE\_ID

, c.COURSE\_NAME

, COUNT (s.STU\_ID) AS STUDENT\_COUNT

FROM

StudentInfo s

JOIN

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

JOIN

CoursesInfo c ON e.COURSE\_ID = c.COURSE\_ID

WHERE

e.ENROLL\_STATUS = 'Enrolled'

GROUP BY 1,2

ORDER BY COUNT (s.STU\_ID) DESC

Output:

