**Task 1-Academic Management System**

**1 (a**) Create table StudentInfo (

STU\_ID INT Primary key,

STU\_NAME varchar(25),

DOB date,

PHONE\_NO BIGINT,

EMAIL\_ID varchar(200),

Address varchar(300));

**(b)** CREATE TABLE coursesinfo (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(255),

COURSE\_INSTRUCTOR\_NAME VARCHAR(255));

**(c)** CREATE TABLE EnrollmentInfo (

ENROLLMENT\_ID INT PRIMARY KEY,

STU\_ID INT,

COURSE\_ID INT,

ENROLLMENT\_STATUS VARCHAR(20),

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

FOREIGN KEY (COURSE\_ID) REFERENCES coursesinfo(COURSE\_ID));

**(2)--inserting value into StudentInfo table**

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

VALUES

(1, 'Alice Johnson', '2000-05-15', 1234567890, 'alice@example.com', '123 Main St, City, State, 12345'),

(2, 'Bob Smith', '1999-08-22', 9876543210, 'bob@example.com', '456 Elm St, Town, State, 54321'),

(3, 'Charlie Brown', '2001-02-10', 5555555555, 'charlie@example.com', '789 Oak St, Village, State, 98765'),

(4, 'David Lee', '2000-11-30', 1112223333, 'david@example.com', '321 Pine St, Hamlet, State, 67890'),

(5, 'Eva Wilson', '1998-04-05', 74598887777, 'eva@example.com', '654 Cedar St, Borough, State, 23456');

**--inserting value into CoursesInfo table**

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

VALUES

(1, 'Introduction to Programming', 'John Smith'),

(2, 'Database Management', 'Emily Johnson'),

(3, 'Web Development Fundamentals', 'Michael Brown'),

(4, 'Data Analysis with Python', 'Sarah Davis'),

(5, 'Artificial Intelligence', 'James Wilson');

**--inserting value into EnrollmentInfo table**

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

VALUES

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

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

(3, 2, 1, 'Not Enrolled'),

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

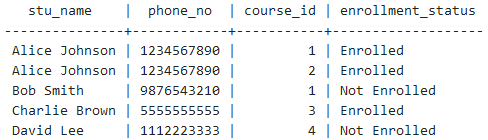
(5, 4, 4, 'Not Enrolled');

**3 (a)** **Query to Display name, contact information & Enrollment status**

Select StudentInfo.STU\_NAME, StudentInfo.PHONE\_NO,EnrollmentInfo.course\_id, EnrollmentInfo.ENROLLMENT\_STATUS

from StudentInfo

JOIN EnrollmentInfo on StudentInfo.STU\_ID = EnrollmentInfo.stu\_id;



**(b) Query to retrieve list of courses in which a specific student enrolled**

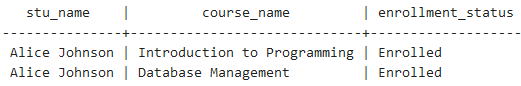
Select StudentInfo.STU\_NAME, CoursesInfo.COURSE\_NAME,enrollmentInfo.enrollment\_status

from enrollmentInfo

LEFT JOIN studentinfo on StudentInfo.STU\_ID = enrollmentInfo.stu\_ID

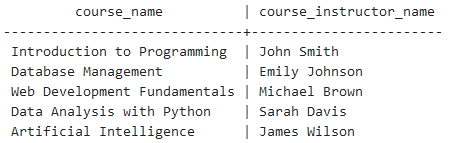
left join coursesinfo on coursesinfo.course\_Id= enrollmentInfo.course\_Id

where stu\_name='Alice Johnson';



**(c) Query to retrieve course information including course name and instructor information**

Select COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME from CoursesInfo;



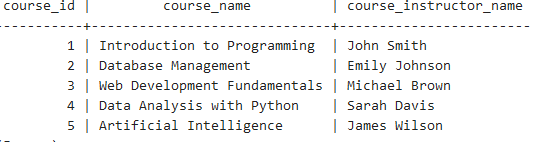
**(d) Query to retrieve course information of specific course**

Select \* from CoursesInfo where COURSE\_NAME = 'Database Management';



**(e) Query to retrieve course information for multiple course**

Select \* from CoursesInfo;



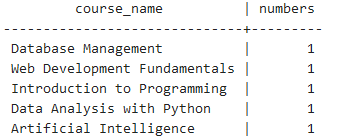
**(f) Executed all the queries listed above and the results are verified too with the sample data and the output data**

**4(a)Query to retrieve number of student enrolled in each course**

Select COURSE\_NAME, count(\*) AS Numbers

from CoursesInfo

Group By COURSE\_NAME;



**(b)Query to retrieve list of students enrolled in specific course**

Select StudentInfo.STU\_NAME

from StudentInfo

Inner Join EnrollmentInfo on StudentInfo.STU\_ID = EnrollmentInfo.stu\_ID

Inner Join CoursesInfo on EnrollmentInfo.ENROLLMENT\_ID=CoursesInfo.COURSE\_ID

Where CoursesInfo.COURSE\_NAME = 'Introduction to Programming';



**(c)Query to retrieve the count of enrolled students for each instructors**

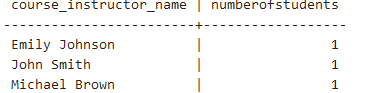
SELECT COURSE\_INSTRUCTOR\_NAME,COUNT(\*) AS NumberOfStudents

FROM coursesinfo

left join enrollmentinfo on EnrollmentInfo.COURSE\_ID = coursesinfo.COURSE\_ID

where ENROLLMENT\_STATUS ='Enrolled'

GROUP BY COURSE\_INSTRUCTOR\_NAME;



**(d)Query to retrieve list of students enrolled in multiple courses**

SELECT StudentInfo.STU\_NAME, COUNT(EnrollmentInfo.COURSE\_ID) AS num\_of\_courses\_enrolled

FROM StudentInfo

JOIN EnrollmentInfo ON StudentInfo.STU\_ID = EnrollmentInfo.STU\_ID

GROUP BY StudentInfo.STU\_ID

HAVING COUNT(EnrollmentInfo.COURSE\_ID) > 1;



**(e)Query to retrieve course that have highest number of enrolled students**

SELECT CoursesInfo.COURSE\_ID, CoursesInfo.COURSE\_NAME, COUNT(EnrollmentInfo.STU\_ID) AS num\_of\_courses\_enrolled

FROM coursesInfo

LEFT JOIN EnrollmentInfo ON CoursesInfo.COURSE\_ID = EnrollmentInfo.COURSE\_ID

GROUP BY CoursesInfo.COURSE\_ID, CoursesInfo.COURSE\_NAME

ORDER BY num\_of\_courses\_enrolled DESC;

