Project Title: Academic Management System (using SQL)

1. Database Creation

```
CREATE DATABASE student_database;
USE student_database;
```

a)

```
CREATE TABLE StudentInfo (
STU_ID int PRIMARY KEY,
STU_NAME varchar(100),
DOB DATE,
PHONE_NO varchar(15),
EMAIL_ID varchar(50),
ADDRESS varchar(250)
);
```

b)

```
CREATE TABLE CourseInfo (
COURSE_ID INT,
COURSE_NAME VARCHAR(100),
COURSE_INSTRUCTOR_NAME VARCHAR(100),
PRIMARY KEY (COURSE_ID)
);
```

c)

```
CREATE TABLE EnrollmentInfo (
ENROLLMENT_ID INT,
STU_ID INT,
COURSE_ID INT,
ENROLL_STATUS VARCHAR(20),
PRIMARY KEY (ENROLLMENT_ID),
FOREIGN KEY (STU_ID) REFERENCES StudentInfo(STU_ID),
FOREIGN KEY (COURSE_ID) REFERENCES CourseInfo(COURSE_ID)
);
```

2. Data Creation:

```
INSERT INTO StudentInfo (STU_ID, STU_NAME, DOB, PHONE_NO, EMAIL_ID, ADDRESS) VALUES
(101, 'Amal Crampire', '1994-11-25', '9999999991', 'amal01@gmail.com', 'Chennai'),
(102, 'Philip Ron', '1994-06-23', '9999999992', 'phil02@gmail.com', 'Bangalore'),
(103, 'Sam Issac', '1993-08-25', '9999999993', 'sam03@gmail.com', 'Hyderabad'),
(104, 'Tony Stark', '1993-10-23', '9999999994', 'tony04@gmail.com', 'Pune'),
(105, 'Anees Shaik', '1997-10-10', '9999999995', 'anee05@gmail.com', 'Delhi'),
(106, 'Jason Santhom', '1991-04-21', '9999999996', 'jason06@gmail.com', 'Indore');

INSERT INTO CourseInfo (COURSE_ID, COURSE_NAME, COURSE_INSTRUCTOR_NAME) VALUES
(1, 'SQL', 'Sheeba'),
(2, 'Python', 'Arun'),
(3, 'AWS', 'Merry'),
(4, 'JAVA', 'Henry'),
(5, 'CSS', 'Hong');

INSERT INTO EnrollmentInfo (ENROLLMENT_ID, STU_ID, COURSE_ID, ENROLL_STATUS) VALUES
(10001, 101, 1, 'ENROLLED'),
(10002, 103, 2, 'ENROLLED'),
(10003, 104, 4, 'ENROLLED'),
```

(10004, 102, 3, 'ENROLLED'),
(10005, 105, 3, 'NOT ENROLLED'),
(10006, 106, 5, 'ENROLLED')
(10007, 101, 5, 'NOT ENROLLED');

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

```
SELECT
s.STU_NAME,
s.PHONE_NO,
s.ADDRESS,
e.ENROLL_STA
TUS

FROM
 StudentInfo s

JOIN
EnrollmentInfo
e
ON s.STU_ID =
e.STU_ID

ORDER BY
e.ENROLL_STA
TUS ASC;
```

STU_NAME	PHONE_NO	ADDRESS	ENROLL_STATUS
Amal Crampire	9999999991	Chennai	ENROLLED
Philip Ron	9999999992	Bangalore	ENROLLED
Sam Issac	9999999993	Hyderabad	ENROLLED
Tony Stark	9999999994	Pune	ENROLLED
Jason Santhom	9999999996	Indore	ENROLLED
Amal Crampire	9999999991	Chennai	NOT ENROLLED
Anees Shaik	9999999995	Delhi	NOT ENROLLED

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

```
SELECT
c.COURSE_NA
ME,
s.STU_NAME
FROM
EnrollmentInfo
e
JOIN
CourseInfo c
ON
e.COURSE_ID =
c.COURSE_ID
JOIN
StudentInfo s
ON s.STU_ID =
e.STU_ID
WHERE
e.STU_ID = 101
AND
e.ENROLL_STA
TUS =
'ENROLLED';
```

COURSE_NAME	STU_NAME
SQL	Amal Crampire

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

```
SELECT *
FROM
CourseInfo;
```

COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME
1	SQL	Sheeba
2	Python	Arun
3	AWS	Merry
4	JAVA	Henry
5	CSS	Hong
NULL	NULL	NULL

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

```
SELECT *
FROM
CourseInfo
WHERE
COURSE_NAM
E = 'SQL';
```

COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME
1	SQL	Sheeba
NULL	NULL	NULL

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

```
SELECT *
FROM
CourseInfo
WHERE
COURSE_NAM
E IN ('SQL',
'Python');
```

COURSE_ID	COURSE_NAME	COURSE_INSTRUCTOR_NAME
1	SQL	Sheeba
2	Python	Arun
NULL	NULL	NULL

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

```
SELECT *
FROM
StudentInfo;
```

STU_ID	STU_NAME	DOB	PHONE_NO	EMAIL_ID
101	Amal Crampire	1994-11-25	9999999991	amal01@gmail.com
102	Philip Ron	1994-06-23	9999999992	phil02@gmail.com
103	Sam Issac	1993-08-25	9999999993	sam03@gmail.com
104	Tony Stark	1993-10-23	9999999994	tony04@gmail.com
105	Anees Shaik	1997-10-10	9999999995	anee05@gmail.com
106	Jason Santhom	1991-04-21	9999999996	jason06@gmail.com
NULL	NULL	NULL	NULL	NULL

4. Reporting and Analytics (Using joining queries):

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

```
SELECT c.Course_Name,
COUNT(e.course_id) AS
numberofStud
FROM CourseInfo c
JOIN EnrollmentInfo e ON
c.course_id = e.course_ID
WHERE e.enroll_status =
'ENROLLED'
GROUP BY c.Course_Name;
```

Course_Name	numberofStud
SQL	1
Python	1
JAVA	1
AWS	1
CSS	1

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

```
SELECT e.COURSE_ID,
c.COURSE_NAME, s.STU_NAME
FROM CourseInfo c
JOIN EnrollmentInfo e ON
c.course_id = e.course_ID
JOIN StudentInfo s ON s.STU_ID
= e.STU_ID
WHERE e.enroll_status =
'ENROLLED';
```

COURSE_ID	COURSE_NAME	STU_NAME
1	SQL	Amal Crampire
2	Python	Sam Issac
4	JAVA	Tony Stark
3	AWS	Philip Ron
5	CSS	Jason Santhom

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

```
SELECT
c.COURSE_INSTRUCTOR_NAME,
COUNT(e.STU_ID) AS
numberofStud
FROM CourseInfo c
JOIN EnrollmentInfo e ON
c.course_id = e.course_ID
WHERE e.enroll_status =
'ENROLLED'
GROUP BY
c.COURSE_INSTRUCTOR_NAME;
```

COURSE_INSTRUCTOR_NAME	numberofStud
Sheeba	1
Arun	1
Henry	1
Merry	1
Hong	1

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

```
SELECT e.stu_id,
COUNT(c.course_id) AS
numberofStud
FROM CourseInfo c
JOIN EnrollmentInfo e ON
c.course_id = e.course_ID
```

stu_id	numberofStud

```
WHERE e.enroll_status =
'ENROLLED'
GROUP BY e.stu_id
HAVING COUNT(c.course_id) >
1;
```

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

```
SELECT c.COURSE_ID,
c.COURSE_NAME,
COUNT(e.STU_ID) AS
numberofStud
FROM CourseInfo c
JOIN EnrollmentInfo e ON
c.course_id = e.course_ID
WHERE e.enroll_status =
'ENROLLED'
GROUP BY c.COURSE_ID,
c.COURSE_NAME
ORDER BY COUNT(e.STU_ID)
DESC;
```

COURSE_ID	COURSE_NAME	numberofStud
1	SQL	1
2	Python	1
4	JAVA	1
3	AWS	1
5	CSS	1