**TASK 1 – OUTPUT SCREENSHOTS**

**-- 1. Database Creation**

**-- Create the tables**

CREATE TABLE StudentInfo (

STU\_ID INT PRIMARY KEY,

STU\_NAME VARCHAR(50),

DOB DATE,

PHONE\_NO VARCHAR(15),

EMAIL\_ID VARCHAR(50),

ADDRESS VARCHAR(100)

);

CREATE TABLE CoursesInfo (

COURSE\_ID INT PRIMARY KEY,

COURSE\_NAME VARCHAR(50),

COURSE\_INSTRUCTOR\_NAME VARCHAR(50)

);

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 sample data**

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

VALUES

(1, 'Rahul Verma', '1998-08-15', '9876543215', 'rahul.verma@email.com', 'Delhi, India'),

(2, 'Priya Patel', '1999-04-22', '7890123450', 'priya.patel@email.com', 'Mumbai, India'),

(3, 'Amit Kumar', '1997-11-05', '9988776658', 'amit.kumar@email.com', 'Kolkata, India');

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

VALUES

(101, 'Mathematics 101', 'Dr. Rajesh Seth'),

(102, 'History 101', 'Dr. Sneha Gupta'),

(103, 'Computer Science 101', 'Prof. Priyanka Sharma');

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

VALUES

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

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

(3, 3, 103, 'Enrolled');

**-- 3. Retrieve Student Information**

**-- a. Retrieve student details**

SELECT STU\_NAME, PHONE\_NO, EMAIL\_ID, ENROLL\_STATUS

FROM StudentInfo

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

A screenshot of a computer

Description automatically generated

**-- b. List of courses for a specific student**

SELECT CoursesInfo.COURSE\_NAME

FROM CoursesInfo

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

WHERE EnrollmentInfo.STU\_ID = 1;

A screenshot of a computer

Description automatically generated

**-- c. Retrieve course information**

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo;

A screenshot of a computer

Description automatically generated

**-- d. Course information for a specific course**

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

WHERE COURSE\_NAME = 'Mathematics 101';

**A screenshot of a computer

Description automatically generated**

**-- e. Course information for multiple courses**

SELECT COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME

FROM CoursesInfo

WHERE COURSE\_NAME IN ('Mathematics 101', 'History 101');

A screenshot of a computer

Description automatically generated

**-- f. Test the queries by executing them**

**-- 4. Reporting and Analytics (using joining queries)**

**-- a. Number of students enrolled in each course**

SELECT CoursesInfo.COURSE\_NAME, COUNT(EnrollmentInfo.STU\_ID) AS Num\_Students

FROM CoursesInfo

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

GROUP BY CoursesInfo.COURSE\_NAME;

A screenshot of a computer

Description automatically generated

**-- b. List of students enrolled in a specific course**

SELECT StudentInfo.STU\_NAME

FROM StudentInfo

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

WHERE EnrollmentInfo.COURSE\_ID = 101;

A screenshot of a computer

Description automatically generated

**-- c. Count of enrolled students for each instructor**

SELECT CoursesInfo.COURSE\_INSTRUCTOR\_NAME, COUNT(EnrollmentInfo.STU\_ID) AS Num\_Students

FROM CoursesInfo

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

GROUP BY CoursesInfo.COURSE\_INSTRUCTOR\_NAME;

A screenshot of a computer

Description automatically generated

**-- d. List of students enrolled in multiple courses**

SELECT StudentInfo.STU\_NAME, COUNT(EnrollmentInfo.STU\_ID) AS Num\_Courses

FROM StudentInfo

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

GROUP BY StudentInfo.STU\_NAME

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

A screenshot of a computer

Description automatically generated

**-- e. Courses with the highest number of enrolled students**

SELECT CoursesInfo.COURSE\_NAME, COUNT(EnrollmentInfo.STU\_ID) AS Num\_Students

FROM CoursesInfo

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

GROUP BY CoursesInfo.COURSE\_NAME

ORDER BY Num\_Students DESC;

A screenshot of a computer

Description automatically generated