# **Task1**

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

**1. Database Creation**

CREATE DATABASE student\_database;

USE student\_database;

|  |  |
| --- | --- |
| **a)** | **b)** |
| *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)*  *);* | 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, 'Tom Hardy', '1993-07-23', '9999999991', 'tom101@gmail.com', 'Chennai'),*  *(102, 'Sam Joseph', '1994-06-23', '9999999992', 'sam102@gmail.com', 'Bangalore'),*  *(103, 'Ben Issac', '1993-08-25', '9999999993', 'ben103@gmail.com', 'Bangalore'),*  *(104, 'Kane Lewis', '1993-10-23', '9999999994', 'kane104@gmail.com', 'Pune'),*  *(105, 'Ian Robert', '1994-06-14', '9999999995', 'ian105@gmail.com', 'Delhi'),*  *(106, 'John Austin', '1991-07-17', '9999999996', 'john106@gmail.com', 'Indore');* |
|  |
| *INSERT INTO CourseInfo (COURSE\_ID, COURSE\_NAME, COURSE\_INSTRUCTOR\_NAME) VALUES*  *(1, 'SQL', 'Hayden'),*  *(2, 'Python', 'Ashish'),*  *(3, 'AWS', 'Tim'),*  *(4, 'JAVA', 'Harry'),*  *(5, 'CSS', 'Nathan');* |
|  |
| *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');* |

3) **Retrieve the Student Information**

|  |  |
| --- | --- |
| 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\_STATUS  FROM StudentInfo s  JOIN EnrollmentInfo e  ON s.STU\_ID = e.STU\_ID  ORDER BY e.ENROLL\_STATUS ASC; |  |
|  |  |
| b) Write a query to retrieve a list of courses in which a specific student is enrolled. | |
| *SELECT c.COURSE\_NAME, 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\_STATUS = 'ENROLLED';* |  |
|  |  |
| c) Write a query to retrieve course information, including course name, instructor information. | |
| *SELECT \**  *FROM CourseInfo;* |  |
|  | |
| d) Write a query to retrieve course information for a specific course. | |
| *SELECT \**  *FROM CourseInfo*  *WHERE COURSE\_NAME = 'SQL';* |  |
|  |  |
| e) Write a query to retrieve course information for multiple courses. | |
| *SELECT \**  *FROM CourseInfo*  *WHERE COURSE\_NAME IN ('SQL', 'Python');* |  |
|  |  |
| 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;* |  |

**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;* |  |
|  |  |
| 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';* |  |
|  |  |
| 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;* |  |
|  |  |
| 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*  *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;* |  |
|  |  |