SASI KUMAR.S

batch ; 10.am | 28-09-2024

[SQL PROJECT ]

[ STUDENT MANAGEMENT ]

* **Project Overview:**

The project is a Student Management System (SMS) designed to manage student information, course enrollment, instructor assignments, and related academic activities. The system aims to streamline academic operations, enhance data management, and provide insights for informed decision-making.

* **Core Features:**

**1. Student Information Management:**

* Store and manage student details (personal info, contact info, etc.)
* Track student enrollment history

**2. Course Management:**

* Create, update, and delete courses
* Assign credit hours and descriptions to courses

**3. Enrollment Management:**

* Enroll students in courses
* Track enrollment dates and course assignments

**4. Instructor Management:**

* Store and manage instructor information (contact info, etc.)
* Assign instructors to courses

**5. Course Assignment Management:**

* Assign courses to instructors
* Track multiple instructors for a single course (if needed)

**6. Reporting and Analytics:**

* Generate reports on student enrollment, course popularity, and instructor workload
* Provide insights for academic planning and resource allocation

**7. Data Validation and Constraints:**

* + Ensure data consistency through primary keys, unique constraints, and enumeration

**SOLUTIONS**

1. **Retrieve the full names and email addresses of all students enrolled in the Computer Science course.**

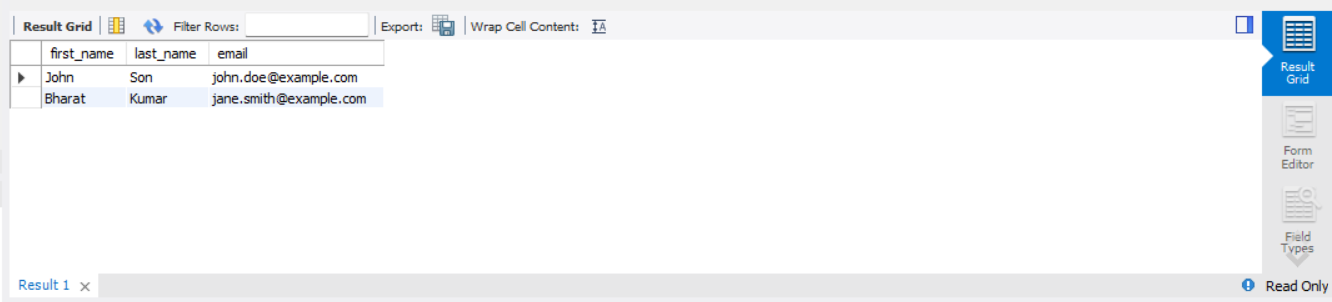
select s.first\_name,s.last\_name,s.email

from students\_details as s

join enrollment as e on s.student\_id = e.student\_id

join courses as c on e.course\_id = c.course\_id

where c.course\_name = "Computer Science";



**2. List all courses along with the number of students enrolled in each course**.

select c.course\_name,count(e.enrollment\_id) as enrollment\_student

from courses as c

join enrollment as e on c.course\_id = e.course\_id

group by c.course\_name

order by enrollment\_student desc;

|  |  |  |
| --- | --- | --- |
|  | |  |
|  | |  |
| **3.Find the instructors who are teaching more than 2 courses.**  select i.first\_name,i.last\_name  from instructors as i  join courseassignments as ca on i.instructor\_id = ca.instructor\_id  group by i.first\_name,i.last\_name  having count(ca.course\_id) > 2; | |  |
| **4. Retrieve the details of students (name, email) who are enrolled in a course taught by Instructor Name.**  SELECT S.first\_name, S.last\_name, S.email  FROM Students\_details S  JOIN Enrollment E ON S.student\_id = E.student\_id  JOIN Courses C ON E.course\_id = C.course\_id  JOIN CourseAssignments CA ON C.course\_id = CA.course\_id  JOIN Instructors I ON CA.instructor\_id = I.instructor\_id  WHERE I.first\_name = 'David' AND I.last\_name = 'Kim'; | |  |
| **5. List all students who have not enrolled in any course.**  select sd.first\_name,sd.last\_name,sd.email  from students\_details as sd  join enrollment as e on sd.student\_id = e.student\_id  where e.student\_id is null; | |  |
|  | |  |
|  |  | | |  |
| **6.** **Get the list of courses with no assigned instructor.**  select c.course\_name  from courses as c  join courseassignments as ca on c.course\_id = ca.course\_id  where ca.course\_id is null; |  | | |  |
|  |  | | |  |
| **7. Retrieve the full details of the Students table, sorted by last name in alphabetical order.** |  | | |  |
| Select \* from students\_details order by last\_name asc; |  | | |  |
|  |  | | |  |
| **8. Find the total number of credit hours for a student enrolled in the Database Systems and Web Development courses.**  SELECT SUM(C.credit\_hours)  FROM Students\_details S  JOIN Enrollment E ON S.student\_id = E.student\_id  JOIN Courses C ON E.course\_id = C.course\_id  AND C.course\_name IN ('Database Systems', 'Web Development')  GROUP BY S.student\_id  HAVING COUNT(DISTINCT C.course\_name) = 2; |  | | |  |
|  |  | | |  |
| **9. Identify the student with the maximum number of enrollments.**  select sd.first\_name,sd.last\_name,sd.email,count(e.enrollment\_id) as totalenrolledstudents  from students\_details sd  join enrollment e on sd.student\_id = e.student\_id  group by sd.student\_id  order by totalenrolledstudents desc limit 2; |  | | |  |
|  |  | | |  |
| **10. Get the details of students whose phone numbers are missing.**  select \* from students\_details where phone\_number is null; |  | | |  |
|  |  | | |  |
|  |  | | |  |
|  |  | | |  |