

SQL Server Case Study (University Management System)

Problem Statement: University Management System

Introduction

This assignment is based on the Student University Management System case study, which involves a database for managing students, courses, departments, professors, and enrolments. The database schema includes the following tables:

The assignment case study consists of 30 SQL query questions designed to test your understanding of SQL concepts, including SELECT, JOINS, subqueries, Common Table Expressions (CTEs), GROUP BY, ORDER BY, WHERE, HAVING, and more.

Database Schema

The database consists of the following tables:

1. Students

- **StudentID** (Primary Key): Unique identifier for each student.
- **FirstName**: Student's first name.
- **LastName**: Student's last name.
- **DateOfBirth**: Student's date of birth.
- **DepartmentID** (Foreign Key): References the department the student belongs to.

2. Departments

- **DepartmentID** (Primary Key): Unique identifier for each department.
- **DepartmentName**: Name of the department (e.g., Computer Science, Mathematics).
- **HeadProfessorID** (Foreign Key): References the professor who heads the department.

3. Courses

- **CourseID** (Primary Key): Unique identifier for each course.
- **CourseName**: Name of the course (e.g., Database Systems, Calculus).
- **DepartmentID** (Foreign Key): References the department offering the course.
- **Credits**: Number of credits for the course.

4. Enrollments

- **EnrollmentID** (Primary Key): Unique identifier for each enrollment record.
- **StudentID** (Foreign Key): References the student enrolled in the course.

- **CourseID** (Foreign Key): References the course the student is enrolled in.
- **Grade**: Student's grade in the course (e.g., A, B, F).
- **EnrollmentDate**: Date when the student enrolled in the course.

5. Professors

- **ProfessorID** (Primary Key): Unique identifier for each professor.
- **FirstName**: Professor's first name.
- **LastName**: Professor's last name.
- **DepartmentID** (Foreign Key): References the department the professor is associated with.

Assignment Questions

1. Write a query to retrieve all records from the Students table.
2. Write a query to list the first and last names of students who belong to DepartmentID 1.
3. Write a query to display course names and their corresponding department names using an INNER JOIN.
4. Write a query to count the number of students in each department, including departments with zero students, and display the department name and student count.
5. Write a query to list all students' first and last names, sorted alphabetically by last name in ascending order.
6. Write a query to find the first and last names of students who have received an 'A' grade in any course, along with the grade.
7. Write a query to retrieve the names of courses that have more than 3 credits.
8. Write a query to list professors' first and last names along with their department names, including professors who are not assigned to any department.
9. Write a query using a subquery to find the first and last names of students enrolled in the course with CourseID 101.
10. Write a query using a CTE to list students' first and last names along with their total credits earned from enrolled courses.
11. Write a query to identify departments that have no students enrolled and display their department names.
12. Write a query to list the top 5 students by their average grade, displaying their first name, last name, and average grade, sorted in descending order of average grade.
13. Write a query to find courses that have no enrollments and display their course names.

14. Write a query using a subquery to retrieve the first and last names of professors who are department heads.
15. Write a query to list students' first and last names along with their enrollment dates for enrollments made after January 1, 2023.
16. Write a query using a CTE to calculate the average credits per course for each department and display the department name and average credits, sorted by average credits in descending order.
17. Write a query to find students who are enrolled in more than one course, displaying their first name, last name, and the count of courses.
18. Write a query to list course names along with the first and last names of the professor who heads the department offering the course.
19. Write a query using a subquery to find students whose grades are above the overall average grade across all enrollments.
20. Write a query to identify departments with more than 10 students and display the department name and student count.
21. Write a query to find students who are not enrolled in any courses and display their first and last names.
22. Write a query using a CTE to list all courses and the number of enrollments in each, sorted by enrollment count in descending order.
23. Write a query to retrieve the first name, last name, and date of birth of students born after the year 2000.
24. Write a query to list all courses and the total number of students enrolled in each, including courses with zero enrollments.
25. Write a query using a subquery to find the names of courses offered by the Computer Science department.
26. Write a query to list the first and last names of professors who are not heading any department.
27. Write a query to display students' first and last names, their department names, and the number of courses they are enrolled in, including students with zero enrollments.
28. Write a query using a CTE to find students who have received an 'F' grade in any course, displaying their first name, last name, and grade.
29. Write a query to identify courses that are offered by multiple departments and display the course name and the count of departments offering it.
30. Write a query to list students' first and last names along with their most recent enrollment date.

Instructions

- Write the SQL query for each question, ensuring it is syntactically correct and produces the expected output based on the provided schema.
- Use appropriate SQL clauses (e.g., JOIN, GROUP BY, WHERE) as required by the question.
- For questions involving CTEs or subqueries, ensure the logic is clear and optimized.
- Test your queries on a sample database if possible to verify correctness.
- Submit your answers in a single SQL file or document, clearly labeling each query with its corresponding question number.