ASSIGNMENT STUDENT INFORMATION SYSTEM (SIS)

Task 3. Aggregate functions, Having, Order By, GroupBy and Joins:

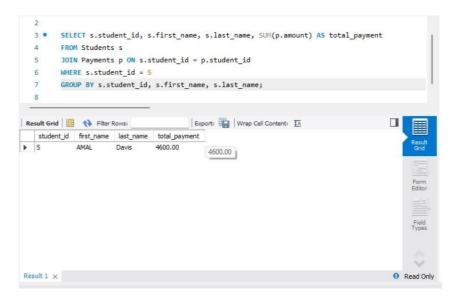
1. Write an SQL query to calculate the total payments made by a specific student. You will need to join the "Payments" table with the "Students" table based on the student's ID.

use sisdb;

SELECT s.student_id, s.first_name, s.last_name, SUM(p.amount) AS total_payment FROM Students s
JOIN Payments p ON s.student_id = p.student_id

WHERE s.student_id = p.student_id WHERE s.student_id = 5

GROUP BY s.student_id, s.first_name, s.last_name;



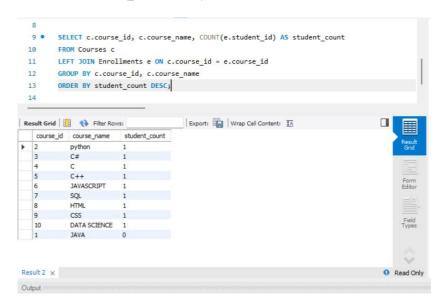
2. Write an SQL query to retrieve a list of courses along with the count of students enrolled in each course. Use a JOIN operation between the "Courses" table and the "Enrollments" table.

SELECT c.course_id, c.course_name, COUNT(e.student_id) AS student_count FROM Courses c

LEFT JOIN Enrollments e ON c.course_id = e.course_id

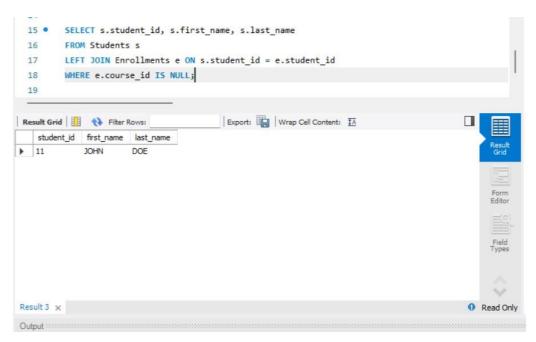
GROUP BY c.course_id, c.course_name

ORDER BY student_count DESC;



3. Write an SQL query to find the names of students who have not enrolled in any course. Use a LEFT JOIN between the "Students" table and the "Enrollments" table to identify students without enrollments.

SELECT s.student_id, s.first_name, s.last_name FROM Students s LEFT JOIN Enrollments e ON s.student_id = e.student_id WHERE e.course_id IS NULL;



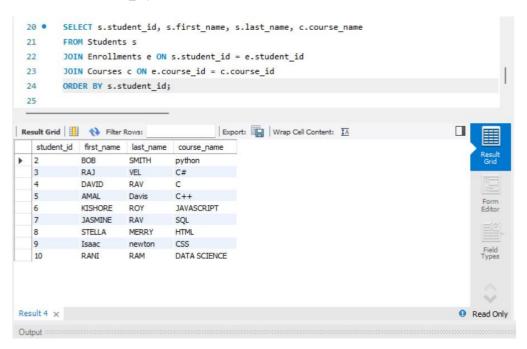
4. Write an SQL query to retrieve the first name, last name of students, and the names of the courses they are enrolled in. Use JOIN operations between the "Students" table and the "Enrollments" and "Courses" tables.

SELECT s.student_id, s.first_name, s.last_name, c.course_name FROM Students s

JOIN Enrollments e ON s.student_id = e.student_id

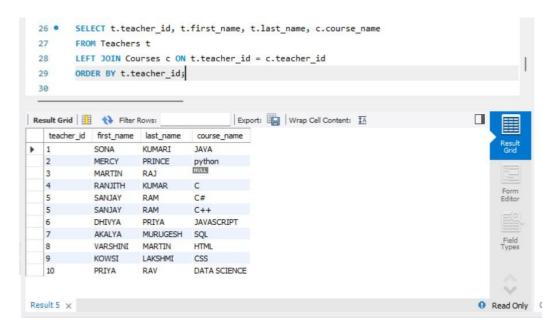
JOIN Courses c ON e.course_id = c.course_id

ORDER BY s.student_id;



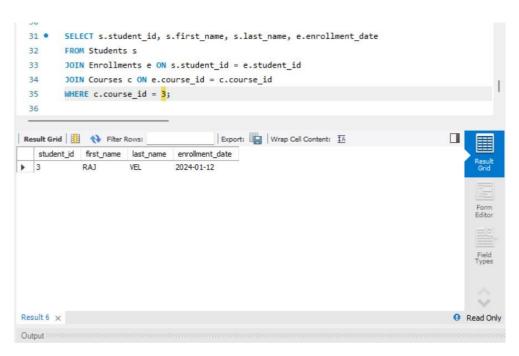
5. Create a query to list the names of teachers and the courses they are assigned to. Join the "Teacher" table with the "Courses" table.

SELECT t.teacher_id, t.first_name, t.last_name, c.course_name FROM Teachers t LEFT JOIN Courses c ON t.teacher_id = c.teacher_id ORDER BY t.teacher_id;



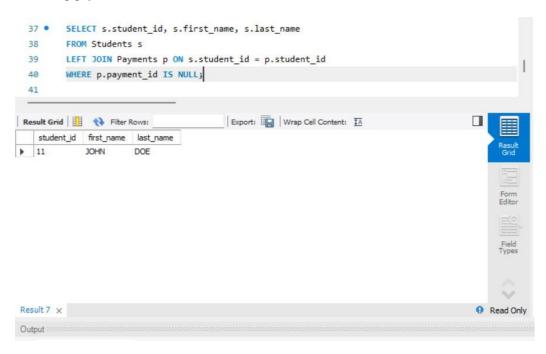
6. Retrieve a list of students and their enrollment dates for a specific course. You'll need to join the "Students" table with the "Enrollments" and "Courses" tables.

SELECT s.student_id, s.first_name, s.last_name, e.enrollment_date FROM Students s
JOIN Enrollments e ON s.student_id = e.student_id
JOIN Courses c ON e.course_id = c.course_id
WHERE c.course_id = 3;



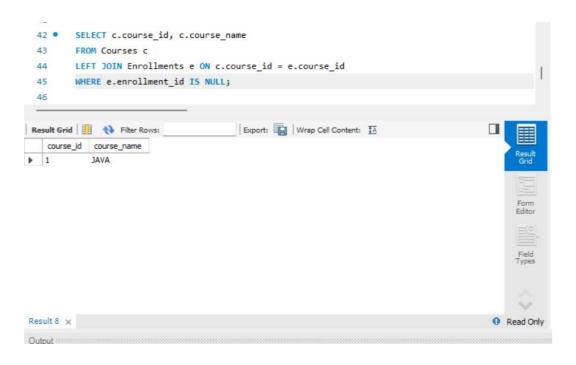
7. Find the names of students who have not made any payments. Use a LEFT JOIN between the "Students" table and the "Payments" table and filter for students with NULL payment records.

SELECT s.student_id, s.first_name, s.last_name FROM Students s LEFT JOIN Payments p ON s.student_id = p.student_id WHERE p.payment_id IS NULL;



8. Write a query to identify courses that have no enrollments. You'll need to use a LEFT JOIN between the "Courses" table and the "Enrollments" table and filter for courses with NULL enrollment records.

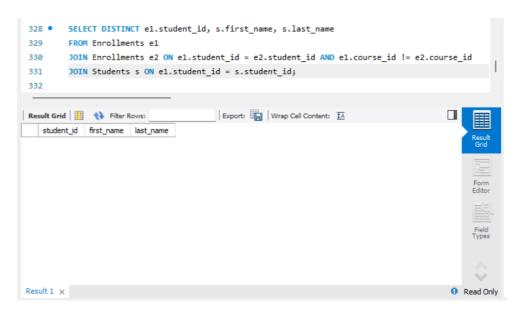
SELECT c.course_id, c.course_name FROM Courses c LEFT JOIN Enrollments e ON c.course_id = e.course_id WHERE e.enrollment_id IS NULL;



9. Identify students who are enrolled in more than one course. Use a self-join on the "Enrollments" table to find students with multiple enrollment records.

SELECT DISTINCT e1.student_id, s.first_name, s.last_name FROM Enrollments e1

JOIN Enrollments e2 ON e1.student_id = e2.student_id AND e1.course_id != e2.course_id JOIN Students s ON e1.student_id = s.student_id;



10. Find teachers who are not assigned to any courses. Use a LEFT JOIN between the "Teacher" table and the "Courses" table and filter for teachers with NULL course assignments.

SELECT t.teacher_id, t.first_name, t.last_name FROM Teachers t LEFT JOIN Courses c ON t.teacher_id = c.teacher_id WHERE c.course_id IS NULL;

