**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.**

mysql> select s.student\_id, s.first\_name, s.last\_name, sum(p.amount) as total\_payments

-> from students s

-> join payments p on s.student\_id = p.student\_id

-> where s.student\_id = 102

-> 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.**

mysql> 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.**

mysql>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 ;

**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.**

mysql> select 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.first\_name, s.last\_name;

**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**

mysql> select t.first\_name, t.last\_name, c.course\_name

-> from teacher t

-> join courses c on t.teacher\_id = c.teacher\_id

-> order by t.first\_name, t.last\_name;

**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.**

mysql> select s.first\_name, s.last\_name, e.enrollment\_date, 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

-> where c.course\_id = 203

-> order by e.enrollment\_date;

**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**

mysql> select 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

-> order by s.first\_name, s.last\_name;

**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.**

mysql> 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

-> order by c.course\_name;

**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.**

mysql> select s.student\_id, s.first\_name, s.last\_name, count(e.course\_id) as course\_count

-> from students s

-> join enrollments e on s.student\_id = e.student\_id

-> group by s.student\_id, s.first\_name, s.last\_name

-> having count(e.course\_id) > 1

-> order by course\_count;

**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.**

mysql> select t.teacher\_id, t.first\_name, t.last\_name

-> from teacher t

-> left join courses c on t.teacher\_id = c.teacher\_id

-> where c.course\_id is null

-> order by t.first\_name, t.last\_name;