**Assignment 2: Student Information System**

**TASK : 4**

1. **Write an SQL query to calculate the average number of students enrolled in each course. Use aggregate functions and subqueries to achieve this.**

***ANSWER:***

SELECT AVG(student\_count) AS avg\_stud\_count FROM

(SELECT COUNT(student\_id) AS student\_count FROM enrollments e group by course\_id) AS stud\_count\_query;

***OUTPUT:***



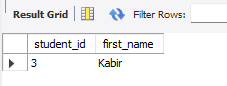
1. **Identify the student(s) who made the highest payment. Use a subquery to find the maximum payment amount and then retrieve the student(s) associated with that amount.**

***ANSWER:***

SELECT s.student\_id,s.first\_name FROM payments p INNER JOIN

students s ON s.student\_id=p.student\_id WHERE amount=(SELECT MAX(amount) FROM payments);

***OUTPUT:***



1. **Retrieve a list of courses with the highest number of enrollments. Use subqueries to find the course(s) with the maximum enrollment count.**

***ANSWER:***

SELECT c.course\_id,c.course\_name,course\_count

FROM (SELECT course\_id,COUNT(student\_id) AS course\_count FROM enrollments GROUP BY course\_id) AS course\_counts

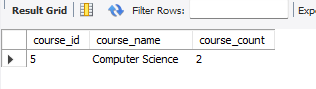
JOIN courses c ON course\_counts.course\_id=c.course\_id

WHERE course\_count=(SELECT MAX(course\_count) FROM

(SELECT COUNT(student\_id) AS course\_count FROM enrollments

GROUP BY course\_id) AS max\_count\_table);

***OUTPUT:***

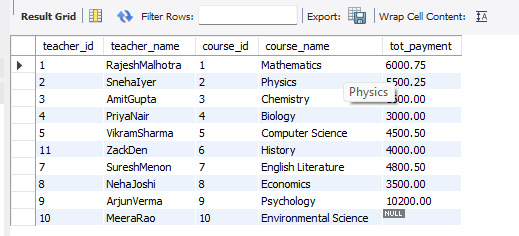


1. **Calculate the total payments made to courses taught by each teacher. Use subqueries to sum payments for each teacher's courses.**

***ANSWER:***

SELECT t.teacher\_id,CONCAT(t.first\_name,t.last\_name) as teacher\_name,c.course\_id,c.course\_name, (SELECT SUM(amount) FROM payments p JOIN enrollments e ON e.student\_id=p.student\_id JOIN courses c ON c.course\_id=e.course\_id WHERE t.teacher\_id=c.teacher\_id) AS tot\_payment FROM TEACHER t INNER JOIN courses c ON t.teacher\_id=c.teacher\_id;

***OUTPUT:***



1. **Identify students who are enrolled in all available courses. Use subqueries to compare a student's enrollments with the total number of courses.**

***ANSWER:***

SELECT s.student\_id,s.first\_name,s.last\_name

FROM students s WHERE (SELECT COUNT(\*) FROM enrollments e

WHERE e.student\_id=s.student\_id) = (SELECT COUNT(\*) FROM courses);

***OUTPUT:***



1. **Retrieve the names of teachers who have not been assigned to any courses. Use subqueries to find teachers with no course assignments.**

***ANSWER:***

SELECT t.teacher\_id,CONCAT(t.first\_name, ' ', t.last\_name) AS teacher\_name FROM teacher t WHERE t.teacher\_id NOT IN (SELECT DISTINCT c.teacher\_id FROM courses c);

***OUTPUT:***



1. **Calculate the average age of all students. Use subqueries to calculate the age of each student based on their date of birth.**

***ANSWER:***

SELECT AVG(YEAR(CURDATE())-YEAR(date\_of\_birth)) AS average\_age FROM students;

***OUTPUT:***

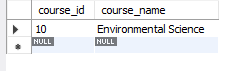


1. **Identify courses with no enrollments. Use subqueries to find courses without enrollment records.**

***ANSWER:***

SELECT c.course\_id,c.course\_name FROM courses c WHERE c.course\_id NOT IN (SELECT e.course\_id FROM enrollments e);

***OUTPUT:***

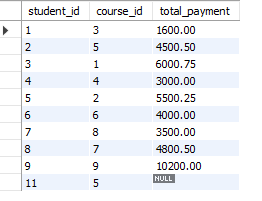


1. **Calculate the total payments made by each student for each course they are enrolled in. Use subqueries and aggregate functions to sum payments.**

***ANSWER:***

SELECT e.student\_id,e.course\_id,(SELECT SUM(p.amount) FROM payments p WHERE p.student\_id=e.student\_id) AS total\_payment FROM enrollments e JOIN students s ON e.student\_id=s.student\_id;

***OUTPUT:***



1. **Identify students who have made more than one payment. Use subqueries and aggregate functions to count payments per student and filter for those with counts greater than one.**

***ANSWER:***

SELECT student\_id,(SELECT COUNT(\*) FROM payments p WHERE p.student\_id=s.student\_id) AS payment\_count FROM students s WHERE (SELECT COUNT(\*) FROM payments p WHERE p.student\_id=s.student\_id) > 1;

***OUTPUT:***

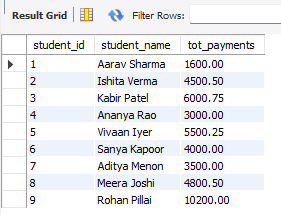


1. **Write an SQL query to calculate the total payments made by each student. Join the "Students" table with the "Payments" table and use GROUP BY to calculate the sum of payments for each student.**

***ANSWER:***

SELECT s.student\_id,CONCAT(s.first\_name,' ',s.last\_name) AS student\_name,SUM(p.amount) AS tot\_payments FROM students s INNER JOIN payments p ON p.student\_id=s.student\_id GROUP BY s.student\_id;

***OUTPUT:***

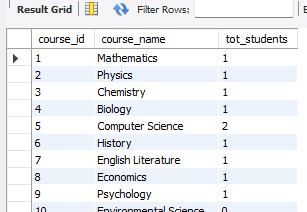


1. **Retrieve a list of course names along with the count of students enrolled in each course. Use JOIN operations between the "Courses" table and the "Enrollments" table and GROUP BY to count enrollments.**

***ANSWER:***

SELECT c.course\_id,c.course\_name,COUNT(e.student\_id) AS tot\_students FROM courses c LEFT JOIN enrollments e ON c.course\_id=e.course\_id GROUP BY c.course\_id;

***OUTPUT:***



1. **Calculate the average payment amount made by students. Use JOIN operations between the "Students" table and the "Payments" table and GROUP BY to calculate the average.**

***ANSWER:***

SELECT s.student\_id,CONCAT(s.first\_name,' ',s.last\_name) AS student\_name,ROUND(AVG(p.amount),2) AS avg\_amt

FROM students s INNER JOIN payments p ON s.student\_id = p.student\_id GROUP BY s.student\_id;

***OUTPUT:***

