**Assignment 2: Student Information System**

**TASK : 3**

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**

***ANSWER:***

INSERT INTO Payments(student\_id,amount,payment\_date) VALUES(9,5000,'2024-02-15'); -- inserted this line for output purpose

SELECT s.student\_id, s.first\_name, s.email, SUM(p.amount) AS total\_payment FROM payments p INNER JOIN students s ON s.student\_id=p.student\_id WHERE s.student\_id=9 GROUP BY s.student\_id; -- answer

1. **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.**

***ANSWER:***

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

1. **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 enrolment.**

***ANSWER:***

INSERT INTO Students (first\_name, last\_name, date\_of\_birth,

Email, phone\_number) VALUES ("Joe","Shang",'1995-08-14',"joeshang@gmail.com",'123456765');

SELECT s.student\_id, s.first\_name FROM students s LEFT JOIN enrollments e ON s.student\_id=e.student\_id WHERE e.student\_id IS NULL;

1. **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.**

***ANSWER:***

SELECT s.first\_name, s.last\_name, c.course\_name FROM students s INNER JOIN enrollments e ON s.student\_id=e.student\_id INNER JOIN courses c ON c.course\_id=e.course\_id;

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

***ANSWER:***

SELECT CONCAT (t.first\_name,' ',t.last\_name) AS teacher\_name, c.course\_name FROM teacher t INNER JOIN courses c ON t.teacher\_id=c.teacher\_id;

1. **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.**

***ANSWER:***

SELECT s.student\_id, c.course\_name, e.enrollment\_date FROM students s INNER JOIN enrollments e ON s.student\_id=e.student\_id INNER JOIN courses c ON c.course\_id=e.course\_id WHERE c.course\_name="Computer Science";

1. **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.**

**ANSWER:**

SELECT CONCAT (s.first\_name, ' ', s.last\_name) AS non\_paying\_student\_name FROM students s LEFT JOIN payments p ON s.student\_id=p.payment\_id WHERE p.student\_id IS NULL;

1. **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.**

**ANSWER:**

SELECT c.course\_id, c.course\_name FROM courses c LEFT JOIN enrollments e ON c.course\_id=e.course\_id WHERE e.course\_id IS NULL;

1. **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.**

**ANSWER:**

SELECT e1.student\_id, COUNT(e2.course\_id) as course\_count FROM enrollments e1 JOIN enrollments e2 ON e1.student\_id=e2.student\_id GROUP BY (e1.student\_id) HAVING course\_count>1;

**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.**

**ANSWER:**

SELECT t.teacher\_id, CONCAT (t.first\_name,' ',t.last\_name) AS teacher\_name FROM teacher t LEFT JOIN courses c ON t.teacher\_id=c.teacher\_id WHERE c.course\_id IS NULL;