**ASSIGNMENT TASK 3-SISDB**

mysql> use SISDB;

Database changed

**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\_payment FROM Students s JOIN Payments p ON s.student\_id = p.student\_id WHERE s.student\_id = 14 GROUP BY s.student\_id, s.first\_name, s.last\_name;

+------------+------------+-----------+---------------+

| student\_id | first\_name | last\_name | total\_payment |

+------------+------------+-----------+---------------+

| 14 | Divya | Natarajan | 2100.00 |

+------------+------------+-----------+---------------+

1 row in set (0.01 sec)

**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;

+-----------+----------------------+---------------+

| course\_id | course\_name | student\_count |

+-----------+----------------------+---------------+

| 1001 | Python Basics | 2 |

| 1002 | Data Structures | 1 |

| 1003 | Database Systems | 1 |

| 1004 | Networking | 1 |

| 1006 | AI & ML | 1 |

| 1008 | Software Engineering | 1 |

| 1009 | Web Development | 1 |

| 1010 | Cloud Computing | 1 |

| 1005 | Operating Systems | 0 |

| 1007 | Cyber Security | 0 |

+-----------+----------------------+---------------+

10 rows in set (0.01 sec)

**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 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.student\_id IS NULL;

+------------+------------+-----------+

| student\_id | first\_name | last\_name |

+------------+------------+-----------+

| 15 | Ravi | Chandran |

+------------+------------+-----------+

1 row in set (0.00 sec)

**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;

+------------+-------------+----------------------+

| first\_name | last\_name | course\_name |

+------------+-------------+----------------------+

| Arun | Kumar | Python Basics |

| Divya | Natarajan | Networking |

| John | Doe | Python Basics |

| Keerthi | Rajagopal | Cloud Computing |

| Lakshmi | Subramanian | AI & ML |

| Meera | Venkatesh | Data Structures |

| Priya | Madhavan | Software Engineering |

| Santhosh | Sivam | Web Development |

| Vikram | Reddy | Database Systems |

+------------+-------------+----------------------+

9 rows in set (0.03 sec)

**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 Teachers t LEFT JOIN Courses c ON t.teacher\_id = c.teacher\_id ORDER BY t.first\_name, t.last\_name;

+------------+--------------+----------------------+

| first\_name | last\_name | course\_name |

+------------+--------------+----------------------+

| Anitha | Selvam | AI & ML |

| Deepa | Balakrishnan | Software Engineering |

| Ganesh | Prasad | Database Systems |

| Karthik | Srinivasan | Web Development |

| Lakshmi | Raghavan | NULL |

| Mohan | Vasudevan | Cyber Security |

| Rajesh | Varma | Operating Systems |

| Revathi | Iyer | Cloud Computing |

| Sundar | Krishnan | Python Basics |

| Uma | Shankar | Data Structures |

| Uma | Shankar | Networking |

+------------+--------------+----------------------+

11 rows in set (0.05 sec)

**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 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 = 1001 ORDER BY e.enrollment\_date;

+------------+-----------+-----------------+

| first\_name | last\_name | enrollment\_date |

+------------+-----------+-----------------+

| Arun | Kumar | 2024-03-01 |

| John | Doe | 2025-03-26 |

+------------+-----------+-----------------+

2 rows in set (0.00 sec)

**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.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;

+------------+------------+-----------+

| student\_id | first\_name | last\_name |

+------------+------------+-----------+

| 15 | Ravi | Chandran |

+------------+------------+-----------+

1 row in set (0.00 sec)

**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.course\_id IS NULL;

+-----------+-------------------+

| course\_id | course\_name |

+-----------+-------------------+

| 1005 | Operating Systems |

| 1007 | Cyber Security |

+-----------+-------------------+

2 rows in set (0.00 sec)

**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;

+------------+------------+-----------+--------------+

| student\_id | first\_name | last\_name | course\_count |

+------------+------------+-----------+--------------+

| 18 | Priya | Madhavan | 2 |

+------------+------------+-----------+--------------+

1 row in set (0.00 sec)

**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 Teachers t LEFT JOIN Courses c ON t.teacher\_id = c.teacher\_id WHERE c.course\_id IS NULL;

+------------+------------+-----------+

| teacher\_id | first\_name | last\_name |

+------------+------------+-----------+

| 2 | Lakshmi | Raghavan |

+------------+------------+-----------+

1 row in set (0.00 sec)