**SQL Developer Internship Documentation**

**Objective:**

To understand advanced SQL queries involving joins and filtering techniques to extract meaningful insights from the database.

**Queries:**

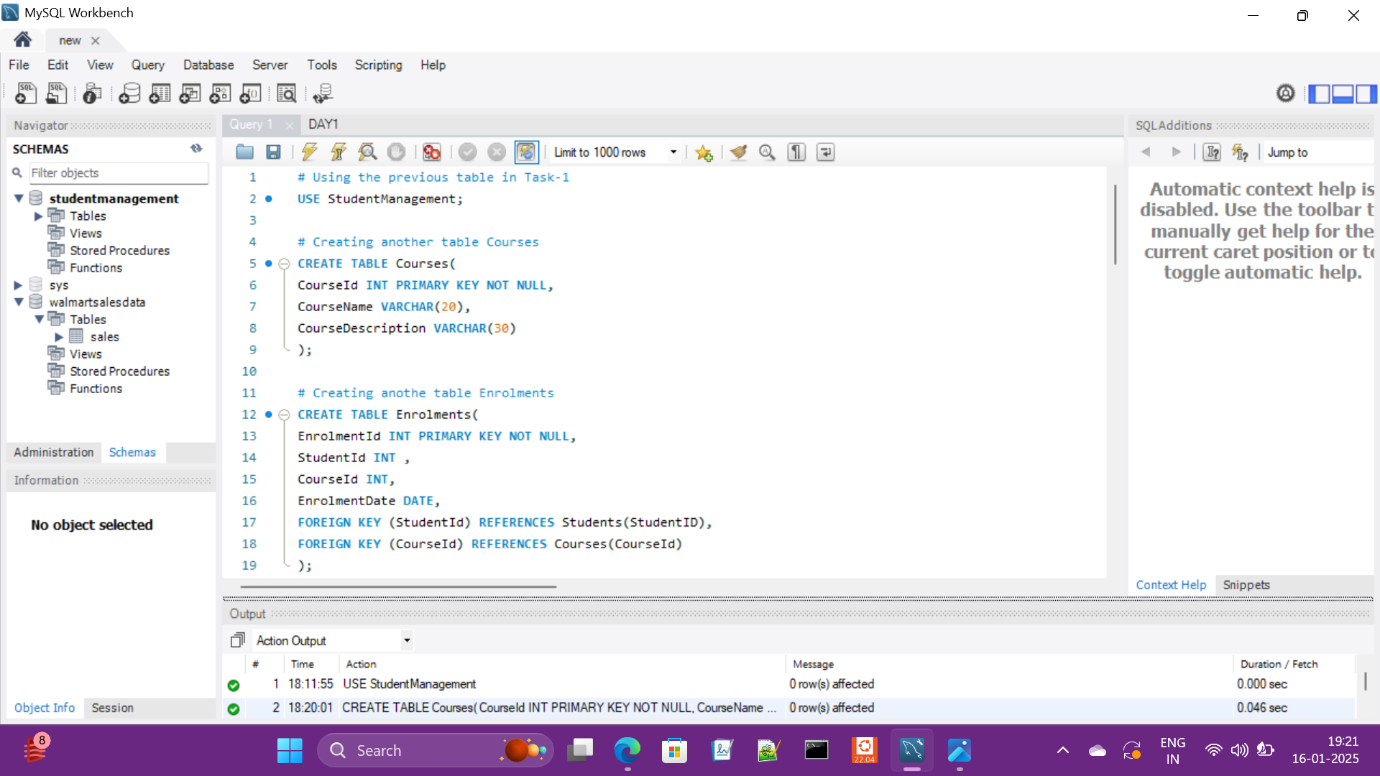
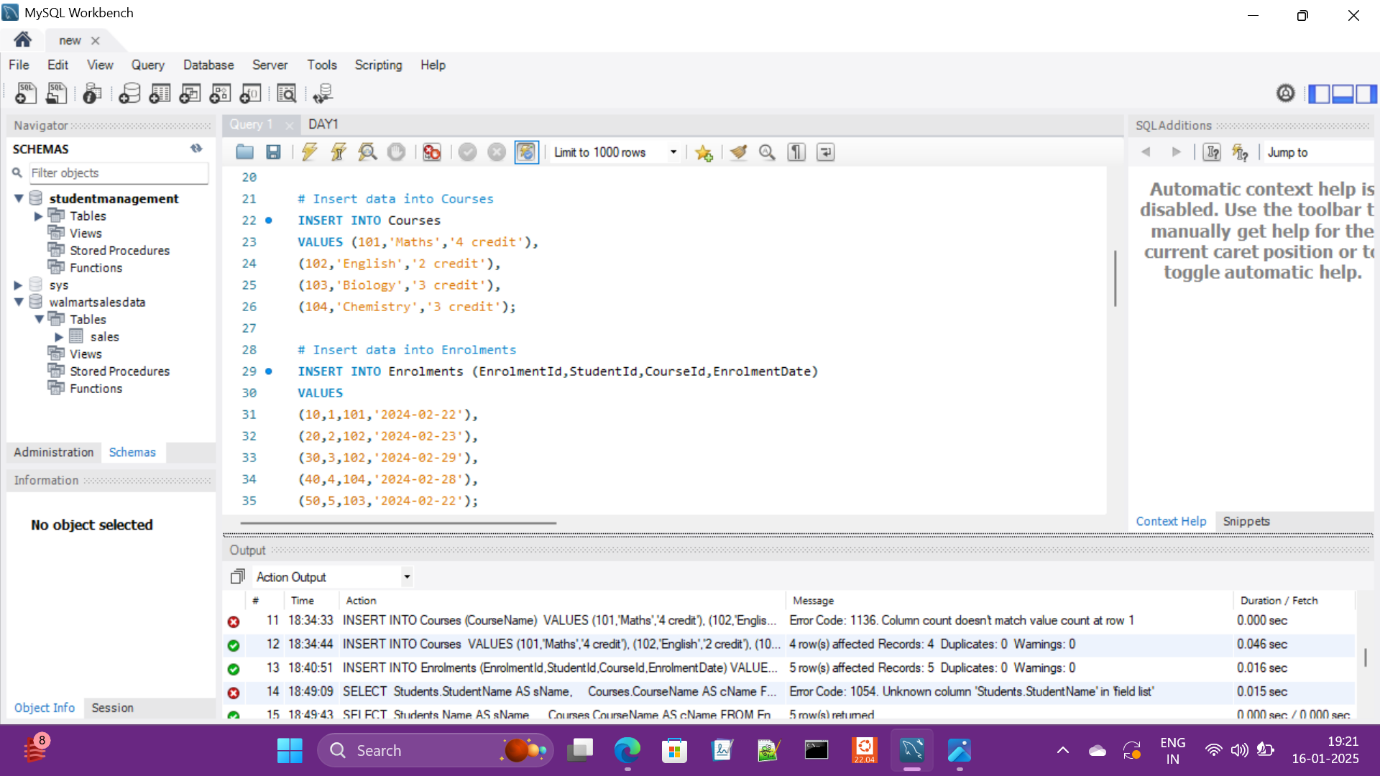
**Database Setup**

**Tables to Create**

1. **Students:**
   * **Already created in Task 1.**
   * **Contains student details such as student\_id, name, and email.**
2. **Courses:**
   * **Fields:**
     + **course\_id: Primary Key.**
     + **course\_name: Name of the course.**
     + **course\_description: Optional field for details.**
3. **Enrollments:**
   * **Fields:**
     + **enrollment\_id: Primary Key.**
     + **student\_id: Foreign Key referencing the Students table.**
     + **course\_id: Foreign Key referencing the Courses table.**
     + **enrollment\_date: Date of enrolment.**

**Purpose of Creating Tables and Inserting Data:**

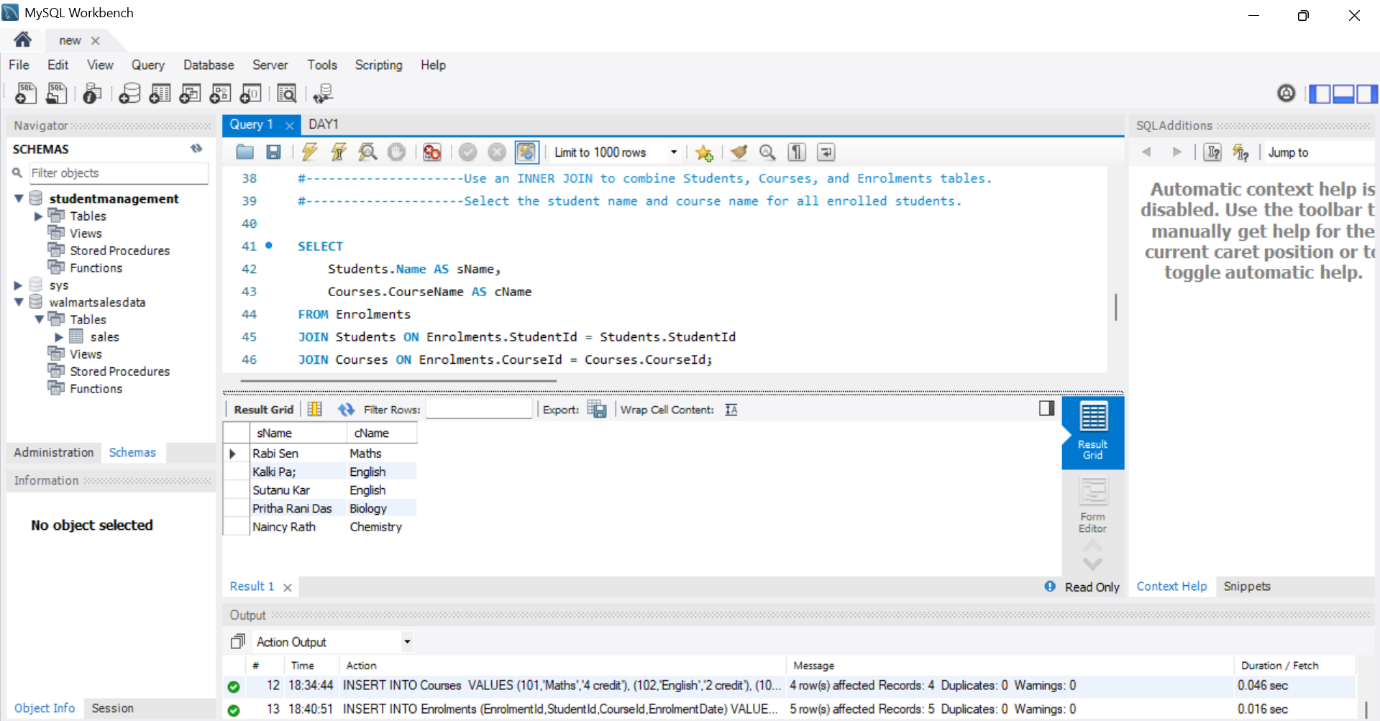
**Creating tables and inserting data helps in organizing and managing information efficiently. It establishes relationships between different data entities, ensuring data integrity and enabling meaningful analysis through queries. This foundation supports effective data storage, retrieval, and manipulation.**

****

**Task 1: List all students and the courses they are enrolled in**

**Output:** This query provides a list of all students along with the names of the courses they are enrolled in. By joining the Students, Enrolments, and Courses tables, we can extract relevant information such as the student's name and their corresponding course names. This output helps in understanding the specific courses each student is taking.

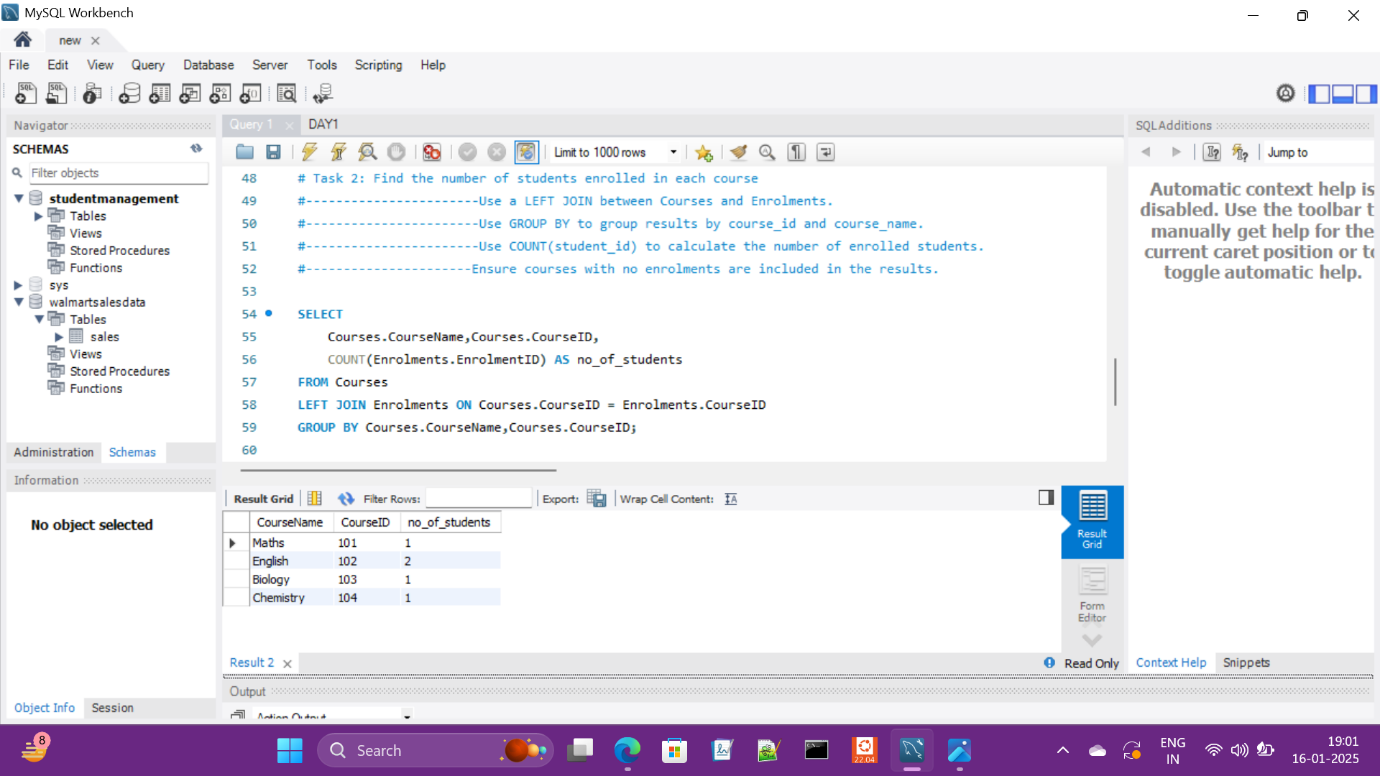
**Explanation:** The query uses an INNER JOIN to combine the Students and Enrolments tables on the student\_id column, and the Enrolments and Courses tables on the course\_id column. This ensures that only records with matching student and course IDs in all three tables are included in the result. The result provides clear visibility of the student-to-course relationships.



**Task 2: Find the number of students enrolled in each course**

**Output:** This query returns a count of students for each course. The result includes the course names and the number of students enrolled in each course. Courses with no enrolled students are also included in the result, showing a count of zero.

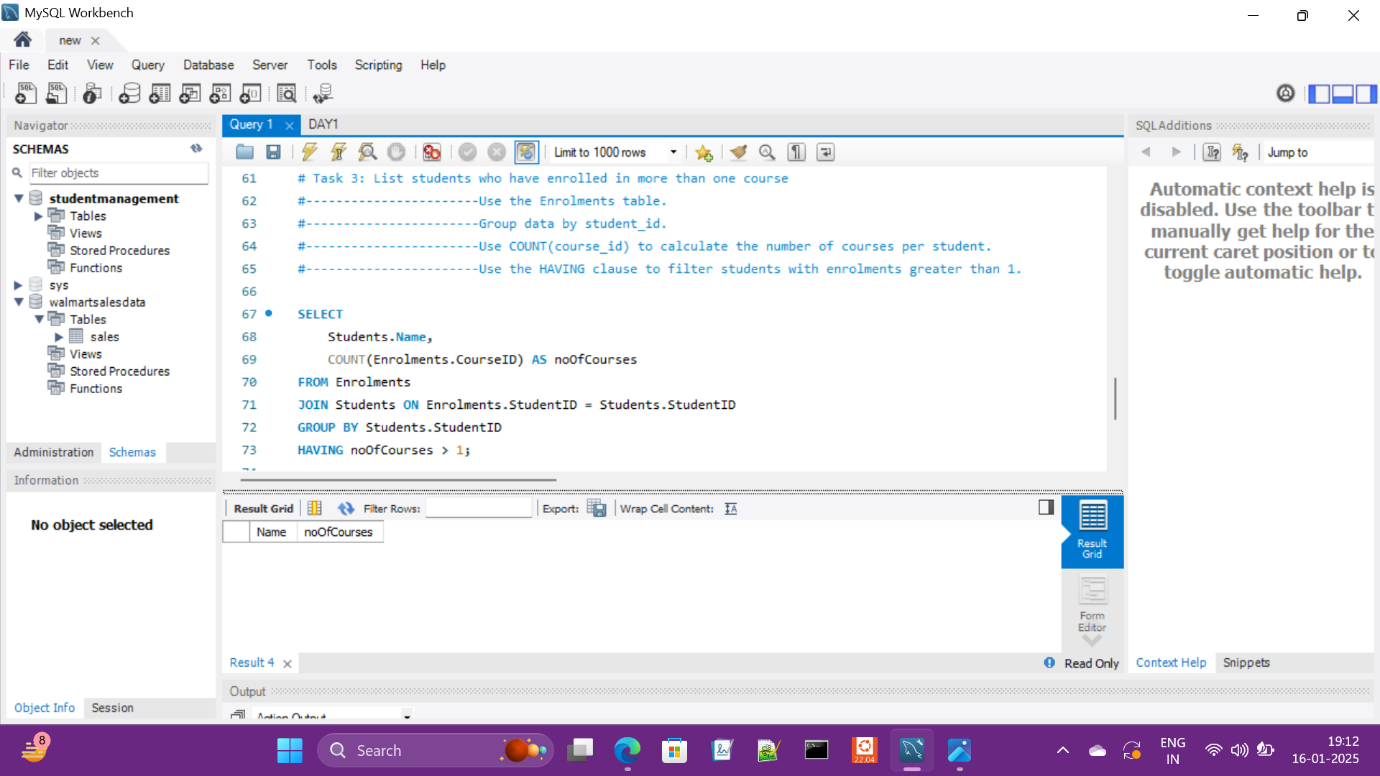
**Explanation:** The query uses a LEFT JOIN between the Courses and Enrolments tables to include all courses, even those without any enrolments. The GROUP BY clause groups the results by course\_id and course\_name, while the COUNT function calculates the number of students in each course. This output allows us to identify the popularity of each course and those that require more promotion or engagement.



**Task 3: List students who have enrolled in more than one course**

**Output:** The query lists students who are enrolled in more than one course. Each student's name is accompanied by the count of courses they are taking, providing insight into the students who have a higher academic load or are pursuing multiple interests.

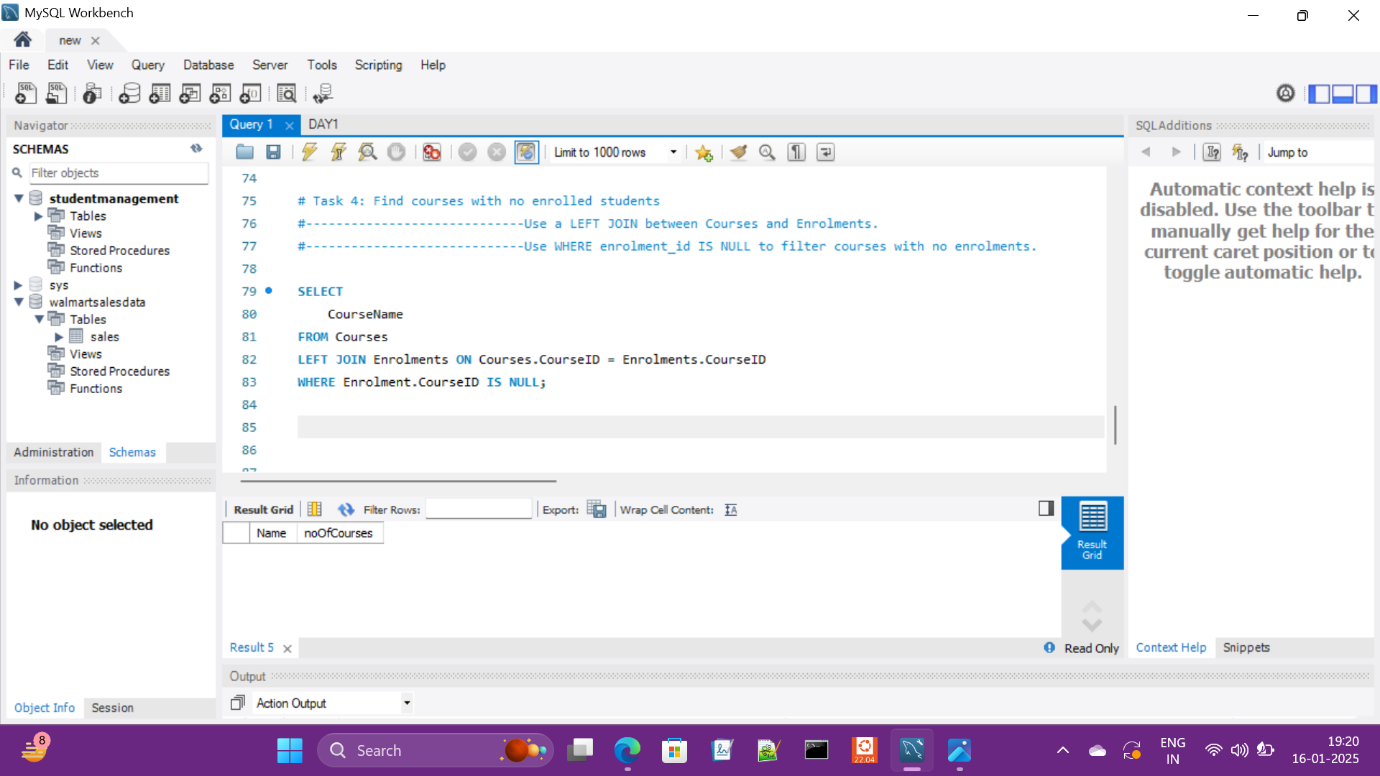
**Explanation:** The query involves an INNER JOIN between the Students and Enrolments tables based on the student\_id column. The GROUP BY clause groups the results by student\_id and student\_name, and the HAVING clause filters the grouped results to include only those with a count of course\_id greater than one. This output helps in identifying highly engaged students and those who may need additional support or resources.



**Task 4: Find courses with no enrolled students**

**Output:** The query returns a list of courses that have no enrolled students. By identifying these courses, we can take steps to understand why they are not attracting students and consider strategies to improve their enrolment.

**Explanation:** The query uses a LEFT JOIN between the Courses and Enrolments tables and filters the results using the WHERE clause to include only those courses where the enrollment\_id is NULL. This effectively identifies courses that are not associated with any students in the Enrollments table. The output provides a clear view of which courses need attention to increase student participation.



**Summary of Key Findings:**

1. **Student-Course Relationships:** The first task output shows detailed enrolment data, helping to understand which students are enrolled in which courses.
2. **Course Popularity:** The second task highlights the number of students enrolled in each course, identifying both popular and under-enrolled courses.
3. **Highly Engaged Students:** The third task identifies students taking multiple courses, providing insights into student engagement and potential support needs.
4. **Courses Needing Attention:** The fourth task highlights courses with no students enrolled, pointing out areas where promotional efforts or course content revisions may be needed.

**Main Flow Services and Technologies Pvt. Ltd.**

* **Contact No:** +91 8777467809
* **Email-Add:** 23053218@kiit.ac.in
* **Website:** www.mainflow.in