



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Fall,Year:2024),B.Sc.in CSE (Day)

LAB REPORT NO - 03

Course Title: Database System

Course Code: CSE210

Section:222-D3

Lab Experiment Name : SQL Joins with Aggregates in dummydb Database

Student Details

Name		ID
1.	MD.SHAJALAL	223002088

Lab Date : 06 - 11 - 2024

Submission Date : 13 – 11 – 2024

Course Teacher's Name : Farhana Akter Sunny

Lab Report Status

Marks:

Signature:.....

Comments:.....

Date:.....

1. TITLE OF THE LAB REPOT EXPERIMENT

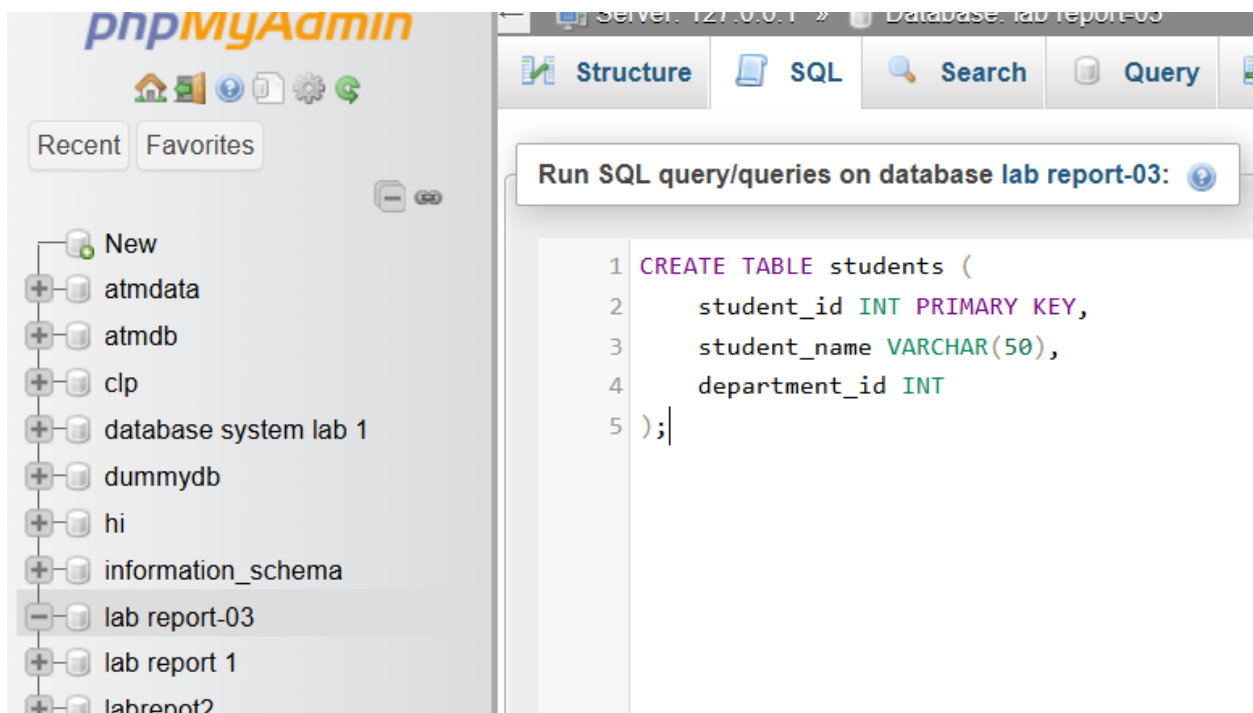
SQL Joins with Aggregates in dummydb Database

2. OBJECTIVES

- Implement and analyze SQL joins (INNER JOIN, LEFT JOIN, FULL OUTER JOIN) with aggregate functions, GROUP BY clauses, and arithmetic operations.
- Generate meaningful reports based on relational data across multiple tables.

3 Database Structure

The following tables in dummydb represent a sample educational system with students, courses, enrollments, and departments.



phpMyAdmin

Server: 127.0.0.1 » Database: lab report-03

Structure SQL Search Query

Run SQL query/queries on database lab report-03: ?

```
1 CREATE TABLE courses (  
2     course_id CHAR(5) PRIMARY KEY,  
3     course_name VARCHAR(50),  
4     credits INT  
5 );
```

Recent Favorites

- New
- atmdata
- atmdb
- clp
- database system lab 1
- dummydb
- hi
- information_schema
- lab report-03
 - New
 - students

phpMyAdmin

Server: 127.0.0.1 » Database: lab report-03

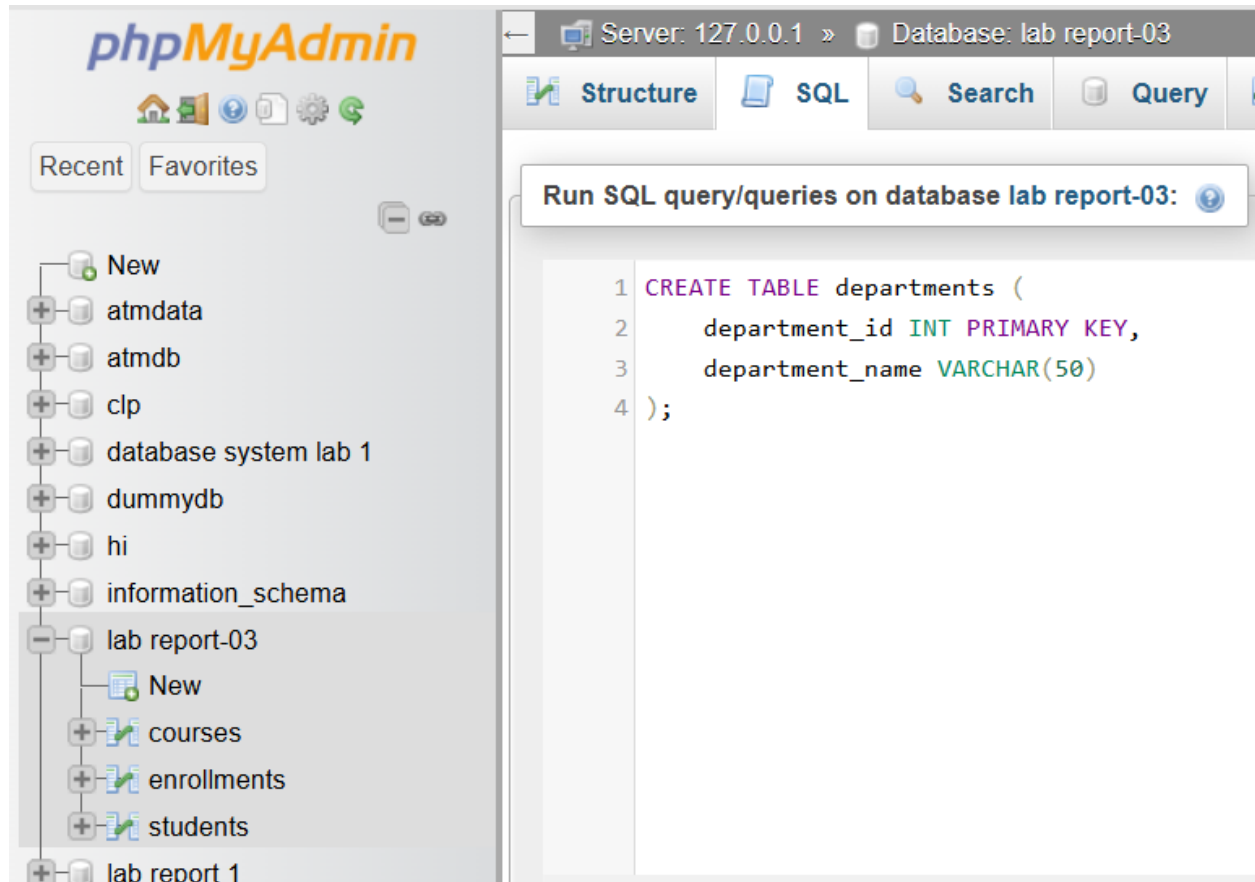
Structure SQL Search Query Export Import

Run SQL query/queries on database lab report-03: ?

```
1 CREATE TABLE enrollments (  
2     enrollment_id INT PRIMARY KEY,  
3     student_id INT,  
4     course_id CHAR(5),  
5     grade INT,  
6     FOREIGN KEY (student_id) REFERENCES students(student_id),  
7     FOREIGN KEY (course_id) REFERENCES courses(course_id)  
8 );
```

Recent Favorites

- New
- atmdata
- atmdb
- clp
- database system lab 1
- dummydb
- hi
- information_schema
- lab report-03
 - New
 - courses
 - students















Sample Data

Let's insert some sample data into the tables:










```
INSERT INTO students (student_id, student_name, department_id) VALUES  
(1, 'Md.Shajalal', 101),  
(2, 'Sojib', 102),  
(3, 'Sakib', 101),  
(4, 'Ashraful Hridoy', NULL);
```

Output:

		student_id	student_name	department_id
<input type="checkbox"/>	 Edit  Copy  Delete	1	Md.Shajalal	101
<input type="checkbox"/>	 Edit  Copy  Delete	2	Sojib	102
<input type="checkbox"/>	 Edit  Copy  Delete	3	Sakib	101
<input type="checkbox"/>	 Edit  Copy  Delete	4	Ashraful Hridoy	NULL

















```
INSERT INTO courses (course_id, course_name, credits) VALUES
('C210', 'Database System', 4),
('C205', 'EEE', 3),
('C103', 'Functional Bengali', 2);
```

Output:

		course_id	course_name	credits
<input type="checkbox"/>	 Edit  Copy  Delete	C103	Functional Bengali	2
<input type="checkbox"/>	 Edit  Copy  Delete	C205	EEE	3
<input type="checkbox"/>	 Edit  Copy  Delete	C210	Database System	4

```
INSERT INTO enrollments (enrollment_id, student_id, course_id, grade)
VALUES
(1, 1, 'C210', 85),
(2, 1, 'C205', 90),
(3, 2, 'C210', 78),
(4, 3, 'C103', 88),
(5, 3, 'C210', 92);
```

Output:

	enrollment_id	student_id	course_id	grade
<input type="checkbox"/>  Edit  Copy  Delete	1	1	C210	85
<input type="checkbox"/>  Edit  Copy  Delete	2	1	C205	90
<input type="checkbox"/>  Edit  Copy  Delete	3	2	C210	78
<input type="checkbox"/>  Edit  Copy  Delete	4	3	C103	88
<input type="checkbox"/>  Edit  Copy  Delete	5	3	C210	92

```
INSERT INTO departments (department_id, department_name)
```











```
VALUES
```

```
(101, 'Science'),
```

```
(102, 'Arts'),
```

```
(103, 'Engineering');
```

Output:

	department_id	department_name
<input type="checkbox"/>  Edit  Copy  Delete	101	Science
<input type="checkbox"/>  Edit  Copy  Delete	102	Arts
<input type="checkbox"/>  Edit  Copy  Delete	103	Engineering

1: INNER JOIN with Aggregate Function and GROUP BY

Calculate the average grade for each course.

```
SELECT
    c.course_name,
    AVG(e.grade) AS average_grade
FROM
    courses c
INNER JOIN
    enrollments e ON c.course_id = e.course_id
GROUP BY
    c.course_name;
```

Expected Output:

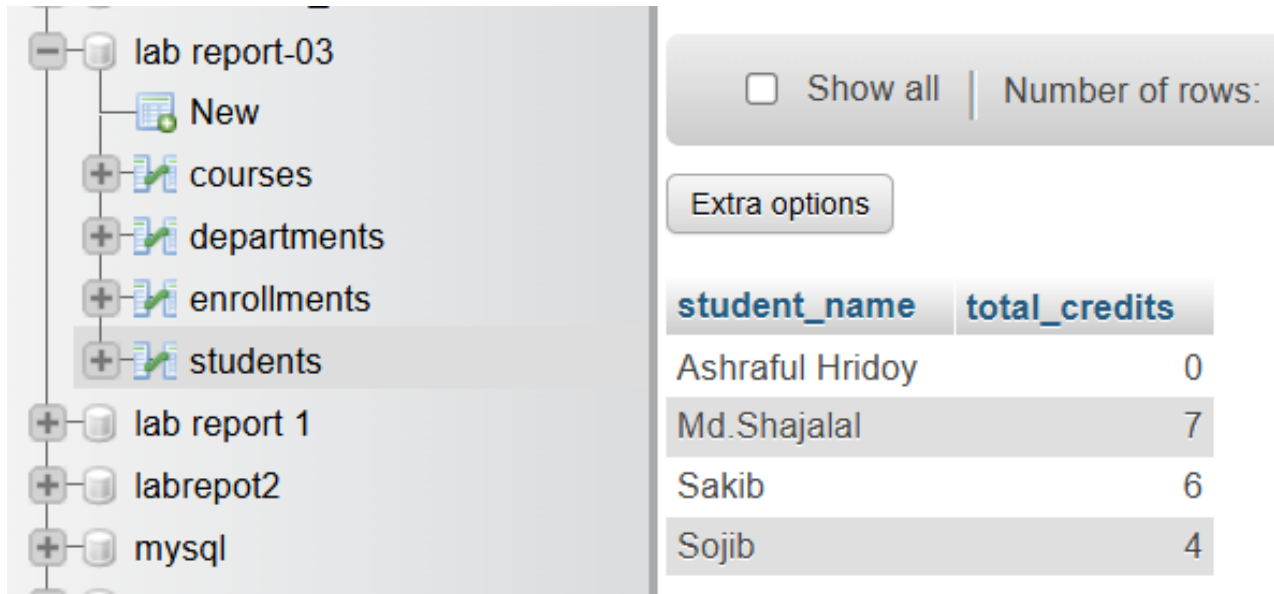
course_name	average_grade
Database System	85.0000
EEE	90.0000
Functional Bengali	88.0000

2: LEFT JOIN with Arithmetic Operation

List all students with the total number of credits they are enrolled in, displaying 0 if the student isn't enrolled in any course.

```
SELECT
    s.student_name,
    COALESCE(SUM(c.credits), 0) AS total_credits
FROM
    students s
LEFT JOIN
    enrollments e ON s.student_id = e.student_id
LEFT JOIN
    courses c ON e.course_id = c.course_id
GROUP BY
    s.student_name;
```


Expected Output:



student_name	total_credits
Ashraful Hridoy	0
Md.Shajalal	7
Sakib	6
Sojib	4

3: FULL OUTER JOIN with Aggregate Function

Display each department with the number of students enrolled in it, including departments with no students.

```
SELECT
    d.department_name,
    COUNT(s.student_id) AS num_students
FROM
    departments d
FULL OUTER JOIN
    students s ON d.department_id = s.department_id
GROUP BY
    d.department_name;
```

4: INNER JOIN with Arithmetic Operation and Filter

Find students who have earned more than 5 credits in total.

```
SELECT
    s.student_name,
    SUM(c.credits) AS total_credits
FROM
    students s
INNER JOIN
    enrollments e ON s.student_id = e.student_id
INNER JOIN
    courses c ON e.course_id = c.course_id
GROUP BY
    s.student_name
HAVING
    SUM(c.credits) > 5;
```

Expected Output:

student_name	total_credits
Md.Shajalal	7
Sakib	6

5: LEFT JOIN with COUNT and Aggregate Function

Display each department along with the number of courses offered, including departments that do not offer any courses.

```
SELECT
    d.department_name,
    COUNT(c.course_id) AS num_courses
FROM
    departments d
LEFT JOIN
    courses c ON d.department_id = c.department_id
GROUP BY
    d.department_name;
```

6: FULL OUTER JOIN with Aggregate Function

Show each student's name, their department, and average grade, including students who may not have any grades.

```
SELECT

    s.student_name,

    d.department_name,

    COALESCE(AVG(e.grade), 'No grades') AS average_grade

FROM

    students s

FULL OUTER JOIN

    enrollments e ON s.student_id = e.student_id

LEFT JOIN

    departments d ON s.department_id = d.department_id

GROUP BY

    s.student_name, d.department_name;
```

Conclusion:

This lab demonstrated SQL joins (INNER, LEFT, and FULL OUTER), aggregate functions, and GROUP BY clauses to generate comprehensive reports across relational tables, showing how SQL can effectively manage and analyze related data.