

"Heaven's Light is Our Guide"
Rajshahi University of Engineering & Technology
Rajshahi, Bangladesh



Department of Electrical & Computer Engineering
(ECE-21)

Course Code: ECE 2216

Course Title: Data Base Systems Sessional

Experiment No: 02

Date of Submission: 01-10-2024

<u>Submitted To:</u>	<u>Submitted By:</u>
Oishi Jyoti Assistant Professor Department of ECE Rajshahi University of Engineering & Technology	Mushfiqur Rahman Roll: 2110028 ECE-21 Series

Experiment No: 02

Experiment Name: Database Query using MySQL.

Theory:

One of the most effective tools for maintaining and working with relational databases is SQL (Structured Query Language). Users may carry out a variety of tasks with it, including obtaining, adding, updating, and removing data. SQL is a standard approach for working with structured data, which makes it indispensable for searching databases. Simple queries, including those that get data, can be followed by more complex ones that filter, group, and aggregate data. With regard to database administration, SQL may be divided into several components, such as Data Query Language (DQL), Data Manipulation Language (DML), Data Definition Language (DDL), and Data Control Language (DCL).

The fact that SQL can handle complicated data jobs with straightforward instructions is one of its main advantages. Direct data manipulation is possible with simple commands like SELECT, UPDATE, INSERT, and DELETE, while more complex operations like JOIN, GROUP BY, HAVING, and ORDER BY aid in efficient information organization and filtering. To glean insights from data, common aggregate methods like SUM(), AVG(), and COUNT() are employed. Subqueries and conditional statements are also supported by SQL, allowing for deeper investigation. Because of its adaptability, SQL is a vital tool in domains such as scientific research, web development, and corporate intelligence.[1], [2]

Objective:

- Gaining knowledge about aggregate functions.
- Understanding subqueries and learning how to apply them.

Student Table:

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
1	Eleven	21	3.8	Engineering	2021	10000	120	active
2	Dustin	22	3.9	Science	2020	9000	110	active
3	Will	19	3.4	Business	2022	8500	95	active
4	Mike	23	3.7	Science	2021	9500	115	inactive
5	Max	20	3.5	Engineering	2020	12000	130	active
6	Eddie	22	4.0	Arts	2019	8000	140	active
7	Billy	24	2.9	Engineering	2022	5000	60	active
8	Alexei	25	3.2	Business	2018	7500	100	inactive
9	Steve	21	3.8	Science	2021	10500	120	active
10	Robin	20	3.6	Engineering	2022	11000	125	active
11	Lucas	18	2.7	Engineering	2023	4000	50	active
12	Nancy	23	3.9	Business	2019	9500	135	active

Task:

- Find students who are older than 20 and have a GPA above the average GPA of all students.
- Find the top 5 students with the highest fees paid, ordered by GPA (in descending order) as a tiebreaker.
- List students who belong to the "Engineering" department, have a GPA greater than 3.5, and are enrolled after 2020.
- Find students who are not active (i.e., enrollment_status = 'inactive') and have not paid any fees (fees_paid = 0).
- Calculate the total fees paid and average GPA for each department, but only for departments with more than 10 students.

Query to create table:

Run SQL query/queries on server "127.0.0.1":

```
1 CREATE DATABASE students_db;
2 USE students_db;
3
```

Run SQL query/queries on server "127.0.0.1":

```
1 USE students_db;
2
3 CREATE TABLE students (
4     student_id INT PRIMARY KEY,
5     student_name VARCHAR(50),
6     age INT,
7     GPA DECIMAL(3, 2),
8     department VARCHAR(50),
9     year_of_admission INT,
10    fees_paid DECIMAL(10, 2),
11    credits_earned INT,
12    enrollment_status VARCHAR(10)
13 );
14
```

Run SQL query/queries on database students_db:

```
1 INSERT INTO students (student_id, student_name, age, GPA, department, year_of_admission, fees_paid, credits_earned, enrollment_status)
2 VALUES
3 (1, 'Eleven', 21, 3.8, 'Engineering', 2021, 18000, 120, 'active'),
4 (2, 'Dustin', 23, 3.9, 'Science', 2020, 5000, 110, 'active'),
5 (3, 'Will', 19, 3.4, 'Science', 2022, 2500, 95, 'active'),
6 (4, 'Mike', 23, 2.7, 'Science', 2021, 9500, 115, 'inactive'),
7 (5, 'Max', 20, 3.5, 'Engineering', 2020, 12000, 130, 'active'),
8 (6, 'Eddie', 20, 4.0, 'Arts', 2019, 2000, 140, 'active'),
9 (7, 'Billy', 20, 2.8, 'Engineering', 2022, 5000, 100, 'active'),
10 (8, 'Alex', 25, 3.3, 'Business', 2018, 7500, 100, 'inactive'),
11 (9, 'Steve', 20, 3.0, 'Science', 2021, 10000, 120, 'active'),
12 (10, 'Robin', 20, 3.4, 'Engineering', 2022, 12000, 115, 'active'),
13 (11, 'Lucas', 18, 2.7, 'Engineering', 2023, 4000, 100, 'active'),
14 (12, 'Nancy', 23, 3.9, 'Business', 2019, 9500, 135, 'active');
```

Output:

	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
<input type="checkbox"/> Edit Copy Delete	1	Eleven	21	3.80	Engineering	2021	18000.00	120	active
<input type="checkbox"/> Edit Copy Delete	2	Dustin	23	3.90	Science	2020	5000.00	110	active
<input type="checkbox"/> Edit Copy Delete	3	Will	19	3.40	Science	2022	2500.00	95	active
<input type="checkbox"/> Edit Copy Delete	4	Mike	23	2.70	Science	2021	9500.00	115	inactive
<input type="checkbox"/> Edit Copy Delete	5	Max	20	3.50	Engineering	2020	12000.00	130	active
<input type="checkbox"/> Edit Copy Delete	6	Eddie	20	4.00	Arts	2019	2000.00	140	active
<input type="checkbox"/> Edit Copy Delete	7	Billy	20	2.80	Engineering	2022	5000.00	100	active
<input type="checkbox"/> Edit Copy Delete	8	Alex	25	3.30	Business	2018	7500.00	100	inactive
<input type="checkbox"/> Edit Copy Delete	9	Steve	20	3.00	Science	2021	10000.00	120	active
<input type="checkbox"/> Edit Copy Delete	10	Robin	20	3.40	Engineering	2022	12000.00	115	active
<input type="checkbox"/> Edit Copy Delete	11	Lucas	18	2.70	Engineering	2023	4000.00	100	active
<input type="checkbox"/> Edit Copy Delete	12	Nancy	23	3.90	Business	2019	9500.00	135	active

Solution of Task 1:

```
Run SQL query/queries on database students_db: ⓘ

1 SELECT *
2 FROM students
3 WHERE age > 20
4 AND GPA > (SELECT AVG(GPA) FROM students);
```

Output:

	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
<input type="checkbox"/> Edit Copy Delete	1	Eleven	21	3.80	Engineering	2021	18000.00	120	active
<input type="checkbox"/> Edit Copy Delete	2	Dustin	23	3.90	Science	2020	5000.00	110	active
<input type="checkbox"/> Edit Copy Delete	12	Nancy	23	3.90	Business	2019	9500.00	135	active

↑ ☐ Check all With selected Edit Copy Delete Export

Solution of Task 2:

```
Run SQL query/queries on database students_db: ⓘ

1 SELECT *
2 FROM students
3 ORDER BY fees_paid DESC, GPA DESC
4 LIMIT 5;
```

Output:

	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
<input type="checkbox"/> Edit Copy Delete	1	Eleven	21	3.80	Engineering	2021	18000.00	120	active
<input type="checkbox"/> Edit Copy Delete	5	Max	20	3.50	Engineering	2020	12000.00	130	active
<input type="checkbox"/> Edit Copy Delete	10	Robin	20	3.40	Engineering	2022	12000.00	115	active
<input type="checkbox"/> Edit Copy Delete	9	Steve	20	3.00	Science	2021	10000.00	120	active
<input type="checkbox"/> Edit Copy Delete	12	Nancy	23	3.90	Business	2019	9500.00	135	active

Solution of Task 3:

Run SQL query/queries on database students_db: ⓘ

```
1 SELECT *
2 FROM students
3 WHERE department = 'Engineering'
4 AND GPA > 3.5
5 AND year_of_admission > 2020;
```

Output:

	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Eleven	21	3.80	Engineering	2021	18000.00	120	active
<input type="checkbox"/> Check all	With selected: <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete <input type="checkbox"/> Export								

Solution of Task 4:

Run SQL query/queries on database students_db: ⓘ

```
1 SELECT *
2 FROM students
3 WHERE enrollment_status = 'inactive'
4 AND fees_paid = 0;
```

Output:

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
SELECT * FROM students WHERE enrollment_status = 'inactive' AND fees_paid = 0;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
------------	--------------	-----	-----	------------	-------------------	-----------	----------------	-------------------

Query results operations

Solution of Task 5:

Run SQL query/queries on database students_db: ⓘ

```
1 SELECT department, SUM(fees_paid) AS total_fees, AVG(GPA) AS average_GPA
2 FROM students
3 GROUP BY department
4 HAVING COUNT(*) > 10;
```

Output:

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0118 seconds.)

SELECT department, SUM(fees_paid) AS total_fees, AVG(GPA) AS average_GPA FROM students GROUP BY department HAVING COUNT(*) > 10;

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

department total_fees average_GPA

Query results operations
```

Discussion: This experiment used MySQL with XAMPP to examine simple database operations. First, we made a table called "students" and a database called "student_db." We renamed a column "favorite_subject" to "major" in order to demonstrate how to handle database modifications. To maintain the data's relevance, we removed the records of students who received less than 30 marks. Additionally, we created a new column named "log" and entered data in it according to the semester. Overall, this experiment made the database more accurate and valuable by assisting us in comprehending critical activities like adding, editing, and maintaining data in a straightforward manner.

References:

- [1] A. W. & N. Eng, "Chapter 16 SQL Data Manipulation Language," Oct. 24, 2014, *BCcampus*.
- [2] "SQL Cheat Sheet for SELECT, INSERT, DELETE and UPDATE Commands." Accessed: Oct. 01, 2024. [Online]. Available: <https://www.mssqltips.com/sqlservertip/7511/sql-cheat-sheet-for-select-insert-delete-and-update-commands/>