

*“Heaven’s Light is Our Guide”*

Rajshahi University of Engineering & Technology  
Rajshahi, Bangladesh



Department of Electrical & Computer Engineering (ECE)

Course Code: ECE 2216

Course Title: Data Base Systems Sessional

Experiment No. 02

Experiment Date: 24 September 2024

Submission Date: September 2024

Submitted To

Oishi Jyoti  
Assistant Professor  
Department of ECE  
Rajshahi University of Engineering &  
Technology, Rajshahi

Submitted By

Sajidur Rahman  
Roll: 2110050  
Session: 2021-2022  
Reg: 1104/2021-2022

## Theory:

MySQL is a popular, open-source relational database management system. It stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one to one, one to many, unique, required, or optional, and “pointers” between different tables. The database enforces these rules so that with a well-designed database your application never sees data that’s inconsistent, duplicated, orphaned, out of date, or missing.

## Task:

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

**Tool:** XAMPP control panel

**Task 1: Find students who are older than 20 and have a GPA above the average GPA of all Students.**

## SQL:

```
>SELECT * FROM `students table`  
>SELECT * FROM `Students Table` WHERE age > 20 AND GPA > (SELECT AVG(GPA) FROM `Students Table`);  
>
```

## Output:

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
1	Eleven	21	4	Engineering	2021	10000	120	active
2	Dustin	22	4	Science	2020	9000	110	active
4	Mike	23	4	Science	2021	9500	115	inactive
6	Eddie	22	4	Arts	2019	8000	140	active
9	Steve	21	4	Science	2021	10500	120	active
12	Nancy	23	4	Business	2019	9500	135	active

**Task 2: Find the top 5 students with the highest fees paid, ordered by GPA (in descending order) as a tiebreaker**

## SQL:

```
>SELECT * FROM `Students Table` WHERE department = 'Engineering' AND GPA > 3.5 AND year_of_admission > 2020;  
>SELECT * FROM `Students Table` ORDER BY fees_paid DESC, GPA DESC LIMIT 5;
```

## Output:

student_id	student_name	age	GPA ▼ 2	department	year_of_admission	fees_paid ▼ 1	credits_earned	enrollment_status
5	Max	20	4	Engineering	2020	12000	130	active
10	Robin	20	4	Engineering	2022	11000	125	active
9	Steve	21	4	Science	2021	10500	120	active
1	Eleven	21	4	Engineering	2021	10000	120	active
4	Mike	23	4	Science	2021	9500	115	inactive

**Task 3: List students who belong to the "Engineering" department, have a GPA greater than 3.5, and are enrolled after 2020**

## SQL:

```
>SELECT * FROM `Students Table` WHERE age > 20 AND GPA > (SELECT AVG(GPA) FROM `Students Table`);  
>SELECT * FROM `Students Table` WHERE department = 'Engineering' AND GPA > 3.5 AND year_of_admission > 2020;
```

## Output:

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
1	Eleven	21	4	Engineering	2021	10000	120	active
10	Robin	20	4	Engineering	2022	11000	125	active

**Task 4: Find students who are not active (i.e., enrollment\_status = 'inactive') and have not paid any fees (fees\_paid = 0)**

## SQL:

```
>SELECT * FROM `Students Table` WHERE enrollment_status = 'inactive' AND fees_paid = 0;
```

### Output:

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

**Task 5: Calculate the total fees paid and average GPA for each department, but only for departments with more than 10 students**

### SQL:

```
SELECT *  
FROM `Students Table`  
WHERE age > 20  
AND GPA > (SELECT AVG(GPA) FROM `Students Table`);
```

### Output:

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

### Discussion:

This lab allowed us to practice a wide range of SQL functionalities, including filtering data, sorting and ranking records, aggregating results, and applying conditions on grouped data. Each task was a step toward gaining more proficiency in writing efficient SQL queries to extract useful insights from datasets. These skills are not only essential for database management but also for data analysis in various real-world applications.

### References

[1] MySQL Documentation, MySQL Reference Manual, [online]. Available: <https://dev.mysql.com/doc/>. Accessed: September 2024.

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