“Heaven’s Light is Our Guide”

Rajshahi University of Engineering & Technology, Rajshahi

Department of Electrical & Computer Engineering

Lab report-2

Course Code : ECE 2216

Course Title : Database Systems- Sessional

Experiment No : 02

Submission Date : 01-09-2024

Submitted To	**Submitted By**
Oishi Jyoti	**Md.Shahibul Hasan Siam**
Assistant Professor	**ID:2110023**
Department of ECE	**Department of ECE**

2.1 Experiment No: 2

2.2 Experiment Name:

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 Sl:

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 student

2.3 Objective:

1. Learn Database System Using XAMPP:
2. Understand SQL and NoSQL:
3. Differentiate DML and DDL in SQL:
4. Learn SQL for Table Creation and Data Insertion:
5. Apply SQL Queries to Real-Life Scenarios:

2.4 Query:

```sql
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    student_name VARCHAR(50),
    age INT,
    GPA FLOAT,
    department VARCHAR(50),
    year_of_admission INT,
    fees_paid INT,
    credits_earned INT,
    enrollment_status VARCHAR(20)
);

INSERT INTO students (student_id, student_name, age, GPA, department, year_of_admission, fees_paid, credits_earned, enrollment_status)
VALUES
(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', 22, 2.9, 'Engineering', 2022, 5000, 60, 'active'),
(8, 'Alexei', 25, 2.8, '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');

```

2.5 Output:

	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
Edit Copy Delete	1	Eleven	21	3.8	Engineering	2021	10000	120	active
Edit Copy Delete	2	Dustin	22	3.9	Science	2020	9000	110	active
Edit Copy Delete	3	Will	19	3.4	Business	2022	8500	95	active
Edit Copy Delete	4	Mike	23	3.7	Science	2021	9500	115	inactive
Edit Copy Delete	5	Max	20	3.5	Engineering	2020	12000	130	active
Edit Copy Delete	6	Eddie	22	4	Arts	2019	8000	140	active
Edit Copy Delete	7	Billy	22	2.9	Engineering	2022	5000	60	active
Edit Copy Delete	8	Alexei	25	2.8	Business	2018	7500	100	inactive
Edit Copy Delete	9	Steve	21	3.8	Science	2021	10500	120	active
Edit Copy Delete	10	Robin	20	3.6	Engineering	2022	11000	125	active
Edit Copy Delete	11	Lucas	18	2.7	Engineering	2023	4000	50	active
Edit Copy Delete	12	Nancy	23	3.9	Business	2019	9500	135	active

2.6 Task:

Task-2.6.1: Find students who are older than 20 and have a GPA above the average GPA of all students.

Query:

```
SELECT `student_id`,`student_name` FROM `students` WHERE age>20 AND GPA > (SELECT AVG(GPA) FROM students);
```

Output:

student_name	age	GPA
Eleven	21	3.8
Dustin	22	3.9
Mike	23	3.7
Eddie	22	4
Steve	21	3.8
Nancy	23	3.9

Task-2.6.2: Find the top 5 students with the highest fees paid, ordered by GPA (in descending order) as a tiebreaker

Query:

```
SELECT student_name, fees_paid, GPA FROM students ORDER BY fees_paid DESC, GPA DESC LIMIT 5;
```

Output:

student_name	fees_paid 1	GPA 2
Max	12000	3.5
Robin	11000	3.6
Steve	10500	3.8
Eleven	10000	3.8
Nancy	9500	3.9

Task-2.6.3: List students who belong to the "Engineering" department, have a GPA greater than 3.5, and are enrolled after 2020

Query:

```
SELECT student_name, GPA, department, year_of_admission FROM students WHERE department = 'Engineering' AND GPA > 3.5 AND year_of_admission > 2020;
```

Output:

student_name	GPA	department	year_of_admission
Eleven	3.8	Engineering	2021
Robin	3.6	Engineering	2022

Task-4: Find students who are not active (i.e., enrollment_status = 'inactive') and have not paid any fees (fees_paid = 0)

Query:

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

Output: MySQL returned an empty result set (i.e. zero rows).

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

Query:

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

Output: MySQL returned an empty result set (i.e. zero rows).