"Heaven's Light is Our Guide"

Rajshahi University of Engineering & Technology, Rajshahi

Department of Electrical & Computer Engineering (ECE-2111)

Course Code: ECE-2216
Course Title: Database Systems Sessional
Date of Submission: 01.10.2024
Experiment NO.: 02

Submitted To,
Oishi Jyoti
Assistant Professor
Dept. Of ECE, RUET

Submitted By,
Name: Md. Humayun Khalid
Roll: 2110015
Session: 2021-2022

1. Problem Statement:

Create a database system for ECE'21 with table containing following information of students:

St id	**St name**	Age	GPA	Department	Year	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

Tasks:

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

2. Objective:

- Creating, modifying and managing a database.
- Studying various operations of DDL and DML.

3. Theory:

SQL is a standard programming language used to manage and manipulate relational databases. It enables users to create, retrieve, update, and delete data stored in a database, as well as control access and manage the structure of the database. SQL consists of commands and queries that help interact with the data, allowing users to:

- **Query data**: Retrieve data based on specific criteria using SELECT statements.
- **Insert data**: Add new records to a table using INSERT INTO.
- **Update data**: Modify existing records using UPDATE.
- **Delete data**: Remove records from a table using DELETE.
- **Create and modify databases**: Use CREATE, ALTER, and DROP statements to define or modify tables and databases.
- **Control access**: Use GRANT and REVOKE to manage database permissions.

4. Tools:

- **Computer**
- **XAMPP Control Panel v3**
- **My SQL**

5. Queries & Output:

Creating a table:

```sql
CREATE TABLE students (
    st_id INT PRIMARY KEY,
    st_name VARCHAR(50),
    age INT,
    gpa DECIMAL(3, 2),
    department VARCHAR(50),
    year INT,
    fees_paid DECIMAL(10, 2),
    credits_earned INT,
    enrollment_status VARCHAR(20)
);
```

Inserting data into students table:

```sql
INSERT INTO students (st_id, st_name, age, gpa, department, year, 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', 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');
```

Output:

st_id	st_name	age	gpa	department	year	fees_paid	credits_earned	enrollment_status
1	Eleven	21	3.80	Engineering	2021	10000.00	120	active
2	Dustin	22	3.90	Science	2020	9000.00	110	active
3	Will	19	3.40	Business	2022	8500.00	95	active
4	Mike	23	3.70	Science	2021	9500.00	115	inactive
5	Max	20	3.50	Engineering	2020	12000.00	130	active
6	Eddie	22	4.00	Arts	2019	8000.00	140	active
7	Billy	24	2.90	Engineering	2022	5000.00	60	active
8	Alexei	25	3.20	Business	2018	7500.00	100	inactive
9	Steve	21	3.80	Science	2021	10500.00	120	active
10	Robin	20	3.60	Engineering	2022	11000.00	125	active
11	Lucas	18	2.70	Engineering	2023	4000.00	50	active
12	Nancy	23	3.90	Business	2019	9500.00	135	active

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

Code:

```
SELECT *
FROM students
WHERE age > 20
  AND gpa > (SELECT AVG(gpa) FROM students);

```

Output:

st_id	st_name	age	gpa	department	year	fees_paid	credits_earned	enrollment_status
1	Eleven	21	3.80	Engineering	2021	10000.00	120	active
2	Dustin	22	3.90	Science	2020	9000.00	110	active
4	Mike	23	3.70	Science	2021	9500.00	115	inactive
6	Eddie	22	4.00	Arts	2019	8000.00	140	active
9	Steve	21	3.80	Science	2021	10500.00	120	active
12	Nancy	23	3.90	Business	2019	9500.00	135	active

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

Code:

```
SELECT *
FROM students
ORDER BY fees_paid DESC, gpa DESC
LIMIT 5;

```

Output:

st_id	st_name	age	gpa ▾ 2	department	year	fees_paid ▾ 1	credits_earned	enrollment_status
5	Max	20	3.50	Engineering	2020	12000.00	130	active
10	Robin	20	3.60	Engineering	2022	11000.00	125	active
9	Steve	21	3.80	Science	2021	10500.00	120	active
1	Eleven	21	3.80	Engineering	2021	10000.00	120	active
12	Nancy	23	3.90	Business	2019	9500.00	135	active

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

Code:

```
SELECT *
FROM students
WHERE department = 'Engineering'
  AND gpa > 3.5
  AND year > 2020;

```

Output:

st_id	st_name	age	gpa	department	year	fees_paid	credits_earned	enrollment_status
1	Eleven	21	3.80	Engineering	2021	10000.00	120	active
10	Robin	20	3.60	Engineering	2022	11000.00	125	active

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

Code:

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

```

Output:

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

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

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

st_id	st_name	age	gpa	department	year	fees_paid	credits_earned	enrollment_status

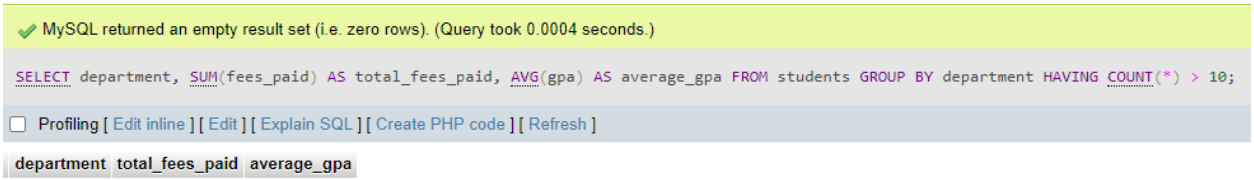
Task-5: Calculate the total fees paid and average GPA for each department, but only for departments with more than 10 students.

Code:

```sql
SELECT department,
       SUM(fees_paid) AS total_fees_paid,
       AVG(gpa) AS average_gpa
FROM students
GROUP BY department
HAVING COUNT(*) > 10;

```

Output:



6. Conclusion:

This SQL experiment demonstrated how to efficiently query and analyze student data using various SQL techniques. Through filtering with WHERE clauses, we identified specific students based on criteria like age, GPA, and enrollment status. Aggregation functions such as SUM() and AVG() allowed us to calculate total fees and average GPAs, while ORDER BY helped rank students based on fees and GPA. The use of GROUP BY and HAVING enabled us to focus on departments with more than 10 students, showcasing SQL's ability to summarize and sort data for meaningful insights. Overall, the experiment highlighted SQL's versatility in data retrieval and analysis.

7. References:

[1] C. J. Date, An Introduction to Database Systems, 8th ed. Boston: Addison-Wesley, 2003.

[2] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed. Boston: Pearson, 2016.

[3] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 6th ed. New York: McGraw-Hill, 2010.