Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology

Course Title
Database System Sessional
Course No: ECE 2216

Lab Report- 02

Date of submission: September 30,2024

Submitted By:	Submitted To:
Jannatul Ferdous Iqra Roll: 2110030 Registration: 1084/2021-2022 Department of ECE	Oishi Jyoti Assistant Professor Department of ECE, RUET

Exp No.: 02

Exp Name: Database Management with MySQL: Query Operations on Students Table.

Objective: To develop a database system and get knowledge about the SQL query that can analyze student's information on different kinds of criteria.

Query And Output:

SQL Command:

```
INSERT INTO student_table (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', 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:

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	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 01: Find students older than 20 with a GPA above the average GPA.

SQL Command:

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

Output:

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
4	Mike	23	3.7	Science	2021	9500	115	inactive
6	Eddie	22	4	Arts	2019	8000	140	active
9	Steve	21	3.8	Science	2021	10500	120	active
12	Nancy	23	3.9	Business	2019	9500	135	active

Task 02: Find the top 5 students with the highest fees paid, ordered by GPA (in descending order) as a tiebreaker

SQL Command:

```
1 SELECT * FROM student_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	3.5	Engineering	2020	12000	130	active
10	Robin	20	3.6	Engineering	2022	11000	125	active
9	Steve	21	3.8	Science	2021	10500	120	active
1	Eleven	21	3.8	Engineering	2021	10000	120	active
12	Nancy	23	3.9	Business	2019	9500	135	active

Task 03: List students who belong to the "Engineering" department, have a GPA greater than 3.5, and are enrolled after 2020

SQL Command:

```
1 SELECT * FROM student_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	3.8	Engineering	2021	10000	120	active
10	Robin	20	3.6	Engineering	2022	11000	125	active

Task 04: Find students who are not active (i.e., enrollment_status = 'inactive') and have not paid any fees (fees_paid = 0)

SQL Command:

```sql
SELECT * FROM student_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

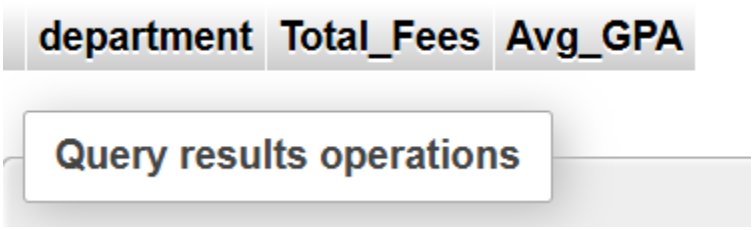
Query results operations

Task 05: Calculate the total fees paid and average GPA for each department, but only for departments with more than 10 students

SQL Command:

```sql
SELECT department, SUM(fees_paid) AS Total_Fees, AVG(GPA) AS Avg_GPA FROM student_table GROUP BY department HAVING COUNT(*) > 10;
```

Output:



Discussions:

This experiment showcased the effective use of SQL queries to analyze student data. We explored various queries to filter students by age, GPA, fees paid, and enrollment status. Additionally, we computed aggregate values to assess the financial contributions of different departments. These operations are crucial for managing academic databases and making data-driven decisions.

References:

- https://www.w3schools.com/mysql/mysql_sql.asp
- https://www.javatpoint.com/mysql-queries