

Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology



Course Title

Database Systems Sessional

Course No.

ECE 2216

Lab Report

02

Submitted By

Shafayetul Huda Sadi

Roll: 2110057

Reg. No.:1111/2021-2022

Department of ECE

RUET

Submitted To

Oishi Jyoti

Assistant Professor

Department of ECE

RUET

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

Table 1: Students Table

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 Data Definition Language (DDL).
- Studying various operations of Data Manipulation Language (DML).

3 Theory

Structured Query Language (SQL) is a powerful and standardized programming language used for managing and manipulating relational databases. It allows users to perform various operations, including querying, updating, inserting, and deleting data within a database. SQL provides a robust framework for retrieving specific information through the use of various commands such as SELECT, INSERT, UPDATE, and DELETE.

A relational database organizes data into tables, which consist of rows and columns. Each table represents a specific entity, with each row corresponding to an individual record and each column representing a particular attribute of that entity. The relationships between tables can be established using foreign keys, enabling complex data interactions and integrity. SQL supports various functions for filtering and sorting data, such as the WHERE clause to specify conditions, the ORDER BY clause to arrange results, and aggregate functions like SUM, AVG, and COUNT to summarize data effectively.

4 Tools

1. Computer
2. MySQL
3. Internet
4. Reference Book

5 Solution

Creating a database name ECE21:

```
1 mysql> CREATE DATABASE ECE21;
2 Query OK, 1 row affected (0.01 sec)
3
4 mysql>
```

Creating a table for students:

```
1 mysql> CREATE TABLE Students (
2     ->     st_id INT PRIMARY KEY,
3     ->     st_name VARCHAR(255),
4     ->     age INT,
5     ->     gpa DECIMAL(3,2),
6     ->     department VARCHAR(255),
7     ->     year INT,
8     ->     fees_paid DECIMAL(10,2),
9     ->     credits_earned INT,
10    ->     enrollment_status VARCHAR(50)
11    -> );
12 Query OK, 0 rows affected (0.04 sec)
13
14 mysql>
```

Inserting data into students table:

```
1 mysql> INSERT INTO Students (st_id, st_name, age, gpa, department, year, fees_paid,
2     credits_earned, enrollment_status)
3     -> VALUES
4     -> (1, 'Eleven', 21, 3.8, 'Engineering', 2021, 10000, 120, 'active'),
5     -> (2, 'Dustin', 22, 3.9, 'Science', 2020, 9000, 110, 'active'),
6     -> (3, 'Will', 19, 3.4, 'Business', 2022, 8500, 95, 'active'),
7     -> (4, 'Mike', 23, 3.7, 'Science', 2021, 9500, 115, 'inactive'),
8     -> (5, 'Max', 20, 3.5, 'Engineering', 2020, 12000, 130, 'active'),
9     -> (6, 'Eddie', 22, 4.0, 'Arts', 2019, 8000, 140, 'active'),
10    -> (7, 'Billy', 24, 2.9, 'Engineering', 2022, 5000, 60, 'active'),
11    -> (8, 'Alexei', 25, 3.2, 'Business', 2018, 7500, 100, 'inactive'),
12    -> (9, 'Steve', 21, 3.8, 'Science', 2021, 10500, 120, 'active'),
13    -> (10, 'Robin', 20, 3.6, 'Engineering', 2022, 11000, 125, 'active'),
14    -> (11, 'Lucas', 18, 2.7, 'Engineering', 2023, 4000, 50, 'active'),
15    -> (12, 'Nancy', 23, 3.9, 'Business', 2019, 9500, 135, 'active');
16 Query OK, 12 rows affected (0.01 sec)
17 Records: 12  Duplicates: 0  Warnings: 0
18 mysql>
```

Output:

```
1 mysql> SELECT st_id, st_name, age, gpa, department, year FROM students;
2 +-----+-----+-----+-----+-----+-----+
3 | st_id | st_name | age | gpa | department | year |
4 +-----+-----+-----+-----+-----+-----+
5 | 1 | Eleven | 21 | 3.80 | Engineering | 2021 |
6 | 2 | Dustin | 22 | 3.90 | Science | 2020 |
7 | 3 | Will | 19 | 3.40 | Business | 2022 |
8 | 4 | Mike | 23 | 3.70 | Science | 2021 |
9 | 5 | Max | 20 | 3.50 | Engineering | 2020 |
10 | 6 | Eddie | 22 | 4.00 | Arts | 2019 |
11 | 7 | Billy | 24 | 2.90 | Engineering | 2022 |
12 | 8 | Alexei | 25 | 3.20 | Business | 2018 |
13 | 9 | Steve | 21 | 3.80 | Science | 2021 |
14 | 10 | Robin | 20 | 3.60 | Engineering | 2022 |
15 | 11 | Lucas | 18 | 2.70 | Engineering | 2023 |
16 | 12 | Nancy | 23 | 3.90 | Business | 2019 |
17 +-----+-----+-----+-----+-----+-----+
18 12 rows in set (0.00 sec)
19
20 mysql>
```

Task 01: Find students who are older than 20 and have a GPA above the average GPA of all students.

```
1 mysql> SELECT st_name, age, GPA
2     -> FROM students WHERE age > 20 AND GPA > (SELECT AVG(GPA) FROM Students);
3
4 +-----+-----+-----+
5 | st_name | age | GPA |
6 +-----+-----+-----+
7 | Eleven | 21 | 3.80 |
8 | Dustin | 22 | 3.90 |
9 | Mike | 23 | 3.70 |
10 | Eddie | 22 | 4.00 |
11 | Steve | 21 | 3.80 |
12 | Nancy | 23 | 3.90 |
13 +-----+-----+-----+
14 6 rows in set (0.00 sec)
15
16 mysql>
```

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

```
1 mysql> SELECT st_name, fees_paid, GPA
2     -> FROM Students ORDER BY fees_paid DESC, GPA DESC LIMIT 5;
3
4 +-----+-----+-----+
5 | st_name | fees_paid | GPA |
6 +-----+-----+-----+
7 | Max | 12000.00 | 3.50 |
8 | Robin | 11000.00 | 3.60 |
9 | Steve | 10500.00 | 3.80 |
10 | Eleven | 10000.00 | 3.80 |
11 | Nancy | 9500.00 | 3.90 |
12 +-----+-----+-----+
13 5 rows in set (0.00 sec)
```

```
14
15 mysql>
```

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

```
1 mysql> SELECT st_name, department, GPA, year
2     -> FROM Students WHERE department = 'Engineering' AND GPA > 3.5 AND year > 2020;
3
4 +-----+-----+-----+-----+
5 | st_name | department | GPA  | year |
6 +-----+-----+-----+-----+
7 | Eleven  | Engineering | 3.80 | 2021 |
8 | Robin   | Engineering | 3.60 | 2022 |
9 +-----+-----+-----+-----+
10 2 rows in set (0.00 sec)
11
12 mysql>
```

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

```
1 mysql> SELECT st_name, enrollment_status, fees_paid
2     -> FROM Students WHERE enrollment_status = 'inactive' AND fees_paid = 0;
3
4 Empty set (0.00 sec)
5
6 mysql>
```

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

```
1 mysql> SELECT department, SUM(fees_paid) AS total_fees, AVG(GPA) AS avg_GPA
2     -> FROM Students GROUP BY department HAVING COUNT(st_id) > 10;
3
4 Empty set (0.00 sec)
5
6 mysql>
```

6 Conclusion

In this lab, we successfully explored and applied various SQL operations to manipulate and analyze a structured database system. By utilizing Data Definition Language (DDL), we created the schema for the student data table, effectively organizing the information into relevant attributes such as student ID, name, age, GPA, department, and enrollment status. Using Data Manipulation Language (DML), we executed a series of queries to insert, update, and retrieve data, demonstrating our understanding of essential database operations.

This exercise provided valuable hands-on experience with SQL, reinforcing the significance of querying data to extract insights, filtering records based on specific criteria, and aggregating results for comprehensive analysis. We gained practical knowledge about the importance of database design, the relationships between different entities, and the need for maintaining data integrity through constraints. These skills are crucial for managing and analyzing databases in real-world applications, enabling effective decision-making and data-driven strategies.

7 Reference

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.