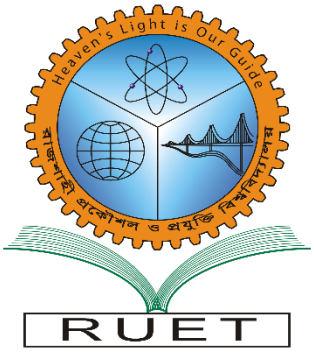
Heaven's Light is Our Guide

Rajshahi University of Engineering & Technology



Department of Electrical & Computer Engineering

Course Code : ECE 2216

Course Title : Data Base Systems

Experiment No. : 02

Submission Date : 30-09-2024

Submitted to:	**Submited by:**
Oishi Jyoti Assistant Professor Department of ECE ,RUET	Shrayas Chakma Roll: 2110061 Regi. No: 1115/2021-2022 ECE-21 Series

Experiment No: 02

Experiment Task:

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

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

Theory:

Structured query language (SQL) is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. You can use SQL statements to store, update, remove, search, and retrieve information from the database. You can also use SQL to maintain and optimize database performance.

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

Solution:

```
SELECT student_id, student_name, age, GPA, department, year_of_admission, fees_paid, credit_earned, enrollment_status FROM students WHERE age > 20
AND GPA > (SELECT AVG(GPA) FROM students);
```

Output:

	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credit_earned	enrollment_status
1	1	Eleven	21	3.8	Engineering	2021	10000	120	active
2	2	Dustin	22	3.9	Science	2020	9000	110	active
3	4	Mike	23	3.7	Science	2021	9500	115	inactive
4	6	Eddie	22	4	Arts	2019	8000	140	active
5	9	Steve	21	3.8	Science	2021	10500	120	active
6	12	Nancy	23	3.9	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

Solution:

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

Output:

	student_id	student_name	fees_paid	GPA
1	5	Max	12000	3.5
2	10	Robin	11000	3.6
3	9	Steve	10500	3.8
4	1	Eleven	10000	3.8
5	12	Nancy	9500	3.9

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

Solution:

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

Output:

	student_id	student_name	department	GPA	year_of_admission
1	1	Eleven	Engineering	3.8	2021
2	10	Robin	Engineering	3.6	2022

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

Solytion:

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

Output:

student_id	student_name	enrollment_status	fees_paid

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

Solution:

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

Output:

department	total_fees_paid	average_GPA	student_count

References:

1. https://aws.amazon.com/what-is/sql/
2. https://www.w3schools.com/MySQL/default.asp