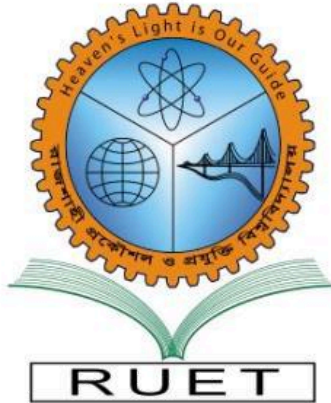


Heaven's Light is Our Guide

Rajshahi University of Engineering and Technology



Department of Electrical & Computer Engineering

Course Code: ECE 2216

Course Title: Database Management System Sessional

Submitted by

Name: Fardin shahriar chowdhury

Roll: 2110058

Submitted to

Oishi Jyoti

Assistant Professor

Department of Electrical and Computer

Engineering, RUET

Experiment No: 03

Experiment Name: Database query using sql

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

Code:

```
SELECT student_name, age, GPA
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	4.0
	Steve	21	3.8
	Nancy	23	3.9

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

Code:

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

Output:

Result Grid			
Filter Rows:			
	student_name	fees_paid	GPA
▶	Max	12000.00	3.5
	Robin	11000.00	3.6
	Steve	10500.00	3.8
	Eleven	10000.00	3.8
	Nancy	9500.00	3.9


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


Code:

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

Output:

Result Grid

 Filter Rows:

Export:  Wrap

	student_name	department	GPA	year_of_admission
▶	Eleven	Engineering	3.8	2021
	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)

Code:

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

Output:

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

Code:

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

Output:

