"Heaven's Light is Our Guide"

Rajshahi University of Engineering & Technology, Rajshahi

Department of Electrical & Computer Engineering

Course Code : ECE 2216

Course Title : Data Base Systems Sessional

Submission Date : 30 September 2024

Submitted To- Oishi Jyoti Associate Professor ECE RUET	Submitted By- Mehnaz Rashid Mohona Roll:2110035 Reg No.1089/2021-22 ECE

Lab No. 2

Lab Tasks:

Create the following Table:

st_id	first_name	last_name	department	marks
1	Shreyas	Chakma	EEE	98
2	Ashik	Rahman	CSE	90
3	Anirban	Sarkar	ECE	85
4	Ripon	Ghosh	EEE	65
5	Afsana	Srmity	GCE	74
6	Shafayet	Sadi	ECE	67
7	Sharif	Hossain	ETE	90

Problems:

1. List students with marks greater than 85.
2. Find the average marks in the department EEE.
3. Count the numbers of students in each department.
4. Count the total marks given in ETE department.
5. List the top 3 students.
6. Find the students whose marks are between 70 to 90.
7. List students in ECE department, limited to the first one.
8. Count the number of student having less than 75 marks.

Lab Report Tasks:

Create the following Table:

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

Problems:

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.

Objectives:

1. To learn about different SQL keywords: SELECT, WHERE, DISTINCT, ORDER BY, GROUP BY, HAVING, SUM, AVG, COUNT, DESC, ASC, LIMIT, BETWEEN and AS.
2. To use these keywords to perform different queries on a database table and solve given problems.

Lab Tasks Solution

Creating table and inserting values:

Query:

```sql
CREATE TABLE st_data (
    st_id INT,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    department VARCHAR(10),
    marks INT
);

INSERT INTO st_data (st_id, first_name, last_name, department, marks) VALUES
(1, 'Shreyas', 'Chakma', 'EEE', 98),
(2, 'Ashik', 'Rahman', 'CSE', 90),
(3, 'Anirban', 'Sarkar', 'ECE', 85),
(4, 'Ripon', 'Ghosh', 'EEE', 65),
(5, 'Afsana', 'Srmity', 'GCE', 74),
(6, 'Shafayet', 'Sadi', 'ECE', 67),
(7, 'Sharif', 'Hossain', 'ETE', 90);
```

Output:

st_id	first_name	last_name	department	marks
1	Shreyas	Chakma	EEE	98
2	Ashik	Rahman	CSE	90
3	Anirban	Sarkar	ECE	85
4	Ripon	Ghosh	EEE	65
5	Afsana	Srmity	GCE	74
6	Shafayet	Sadi	ECE	67
7	Sharif	Hossain	ETE	90

Problem 1: Listing students with marks greater than 85.

Query:

```
SELECT * FROM st_data WHERE marks > 85;
```

Output:

st_id	first_name	last_name	department	marks
1	Shreyas	Chakma	EEE	98
2	Ashik	Rahman	CSE	90
7	Sharif	Hossain	ETE	90

Problem 2: Finding the average marks in the department EEE.

Query:

```
SELECT AVG(marks) FROM st_data WHERE department = 'EEE';
```

Output:

AVG(marks)
81.5000

Problem 3: Counting the numbers of students in each department.

Query:

```
SELECT department, COUNT(*) AS number_of_student FROM st_data GROUP BY department;
```

Output:

department	number of student
CSE	1
ECE	2
EEE	2
ETE	1
GCE	1

Problem 4: Counting the total marks given in ETE department .

Query:

```
SELECT SUM(marks) AS total_marks FROM st_data WHERE department = 'ETE';
```

Output:

total_marks
90

Problem 5: Listing the top 3 students.

Query:

```
SELECT * FROM st_data ORDER BY marks DESC LIMIT 3;
```

Output:

st_id	first name	last name	department	marks ⬍ 1
1	Shreyas	Chakma	EEE	98
2	Ashik	Rahman	CSE	90
7	Sharif	Hossain	ETE	90

Problem 6: Finding the students whose marks are between 70 to 90.

Query:

```
SELECT * FROM st_data WHERE marks BETWEEN 70 AND 90;
```

Output:

st_id	first_name	last_name	department	marks
2	Ashik	Rahman	CSE	90
3	Anirban	Sarkar	ECE	85
5	Afsana	Srmity	GCE	74
7	Sharif	Hossain	ETE	90

Problem 7: Listing students in ECE department, limited to the first one.

Query:

```
SELECT * FROM st_data WHERE department = 'ECE' LIMIT 1;
```

Output:

st_id	first_name	last_name	department	marks
3	Anirban	Sarkar	ECE	85

Problem 8: Counting the number of student having less than 75 marks.

Query:

```
SELECT COUNT(*) AS students_getting_less_marks FROM st_data WHERE marks < 75;
```

Output:

students_getting_less_marks
3

Lab Report Tasks Solution

Creating table and inserting values:

```
CREATE TABLE students_table (
    student_id INT,
    student_name TEXT,
    age INT,
    GPA FLOAT,
    department TEXT,
    year_of_admission INT,
    fees_paid INT,
    credits_earned INT,
    enrollment_status TEXT
);
```

```
INSERT INTO students_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

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

Query:

```
SELECT * FROM students_table WHERE age > 20 AND GPA > (SELECT AVG(GPA) FROM students_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

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

Query:

```
SELECT * FROM students_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

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

Query:

```
SELECT * FROM students_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

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

Query:

```
SELECT * FROM students_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

N.B. The table is empty because there is no such student whose enrollment status is inactive and at the same time did not pay any fees.

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

Query:

```
SELECT department, SUM(fees_paid) AS total_fees, AVG(GPA) AS avg_GPA FROM students_table
GROUP BY department HAVING COUNT(*) > 10;
```

Output:

department	total_fees	avg_GPA

N.B. The table is empty because there is no department who has more than 10 students.