

**Rajshahi University of Engineering & Technology Department of Electrical and
Computer Engineering**



Course Code: ECE 2216

Course Title: Data Base Systems Sessional

Lab Report No : 02

Date of submission : 30-09-2024

Submitted To	Submitted By
Oishi Jyoti Assistant Professor Department of Electrical and Computer Engineering, Rajshahi University of Engineering & Technology	Afsana Islam Roll: 2110051 Registration No: 1105 Session: 2021-2022 Department of Electrical and Computer Engineering, Rajshahi University of Engineering & Technology

Experiment No: 02

Experiment Name: Implementing SQL commands

Theory:

Given student 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	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

Software Used:

- XAMPP

Task 1

Find students who are older than 20 and have a GPA above the average GPA of all Students

Code:

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

☐ Profiling [[Edit inline](#)] [[Edit](#)] [[Explain SQL](#)] [[Create PHP code](#)] [[Refresh](#)]

☐ Show all | Number of rows: Filter rows:

Extra options

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 2

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

Code :

```
SELECT * FROM st_info ORDER BY fees_paid DESC, GPA DESC LIMIT 5;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	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 3

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

Code :

```
SELECT * FROM st_info WHERE department = 'Engineering' AND GPA > 3.5 AND year_of_admission > 2020;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

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 4

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

Code :

```
SELECT * FROM st_info WHERE enrollment_status = 'inactive' AND fees_paid = 0;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
------------	--------------	-----	-----	------------	-------------------	-----------	----------------	-------------------

Query results operations

Task 5

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

Code:

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

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

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

department	total_fees_paid	avg_GPA	st_count
------------	-----------------	---------	----------

Query results operations