“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
Experiment No.	: 02
Experiment Date	: 23 September, 2024
Submission Date	: 30 September, 2024

Submitted To-

Oishi Jyoti

Assistant Professor

Department of ECE, RUET

Submitted By-

Sanjana Islam Orpa

Roll : 2110039

2.1 Experiment No. : 02

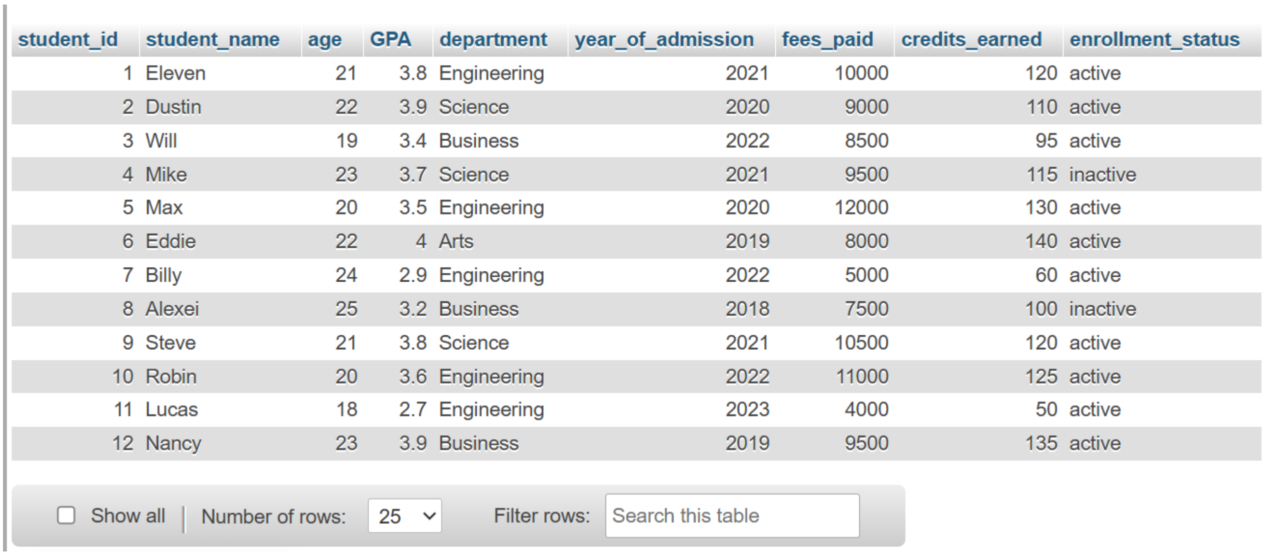
2.2 Experiment Name: Implementation of student data management and analysis using SQL

2.3 Objective:

- Efficient data retrieval.
- Decision making based on financial and academic data.
- To sort student records by their marks.

2.4 Query and Output:

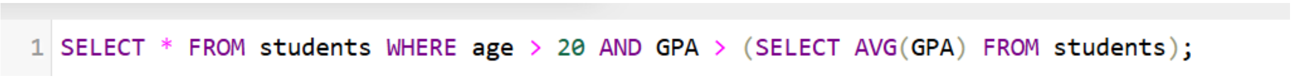
Students Table



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

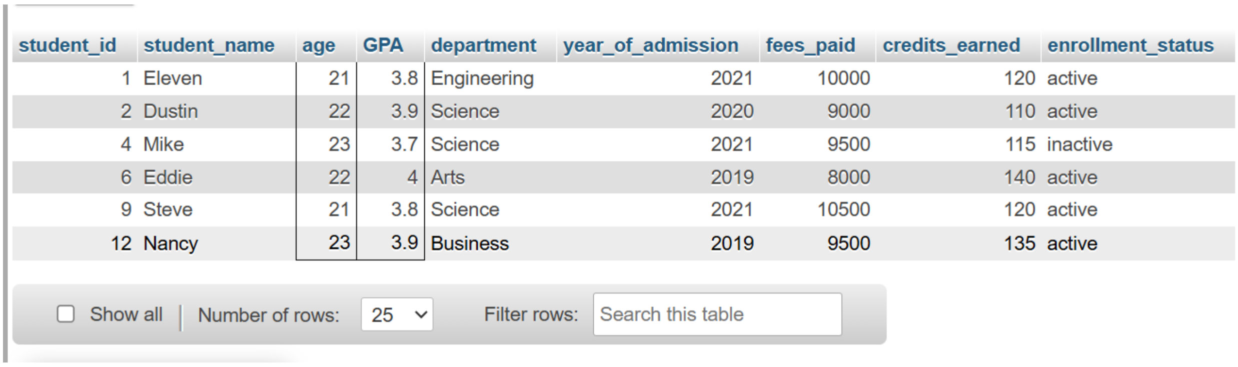
2.4.1 Find students who are older than 20 and have a GPA above the average GPA of all students.

Query:



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

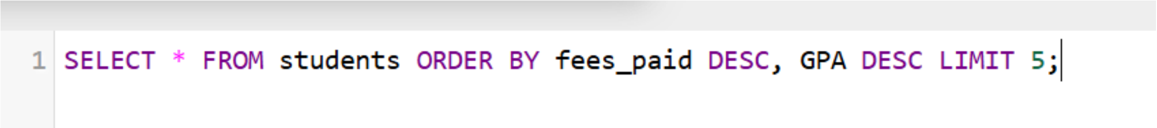
Output:



student_id	student_name	age	GPA	department	year_of_admission	fees_paid	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

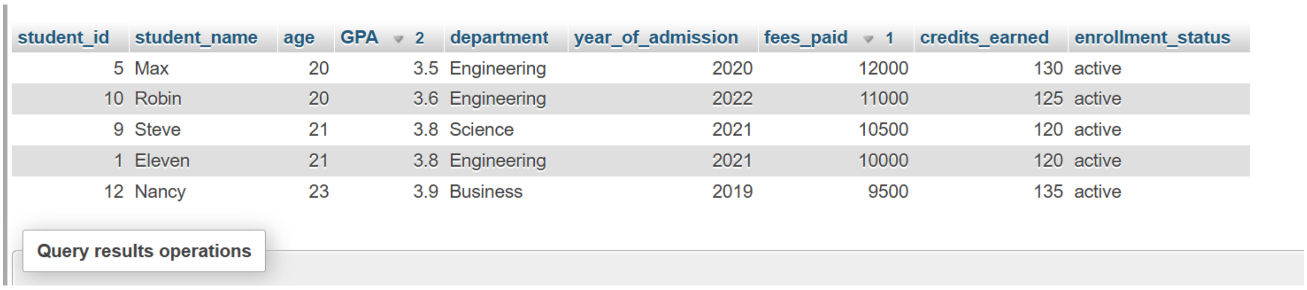
2.4.2 Find the top 5 students with the highest fees paid, ordered by GPA (in descending order) as a tiebreaker

Query:



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

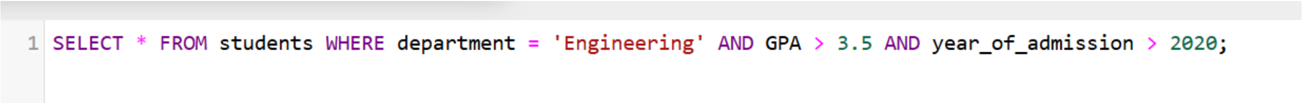
Output:



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

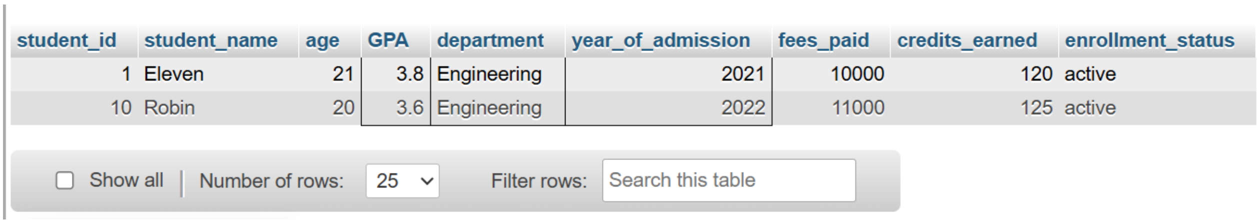
2.4.3 List students who belong to the "Engineering" department, have a GPA greater than 3.5, and are enrolled after 2020

Query:



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

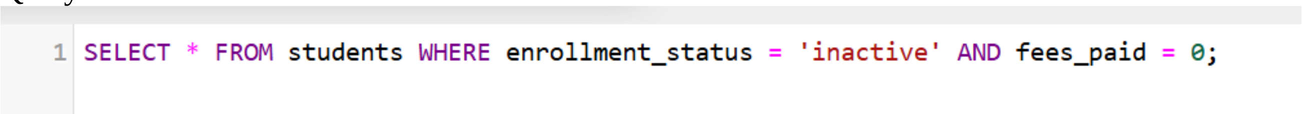
Output:



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

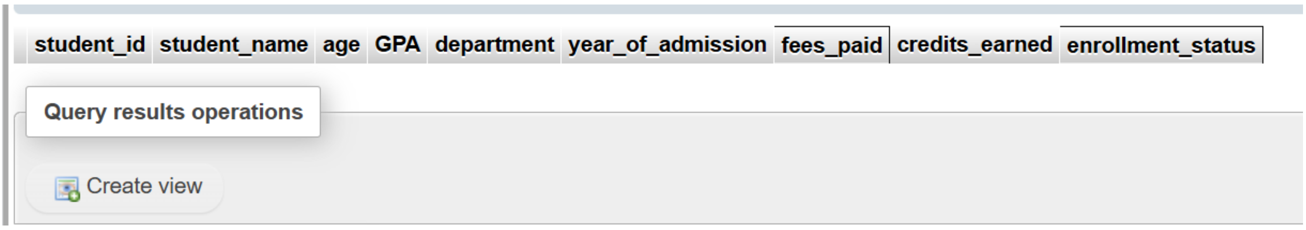
2.4.4 Find students who are not active (i.e., enrollment_status = 'inactive') and have not paid any fees (fees_paid = 0)

Query:



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

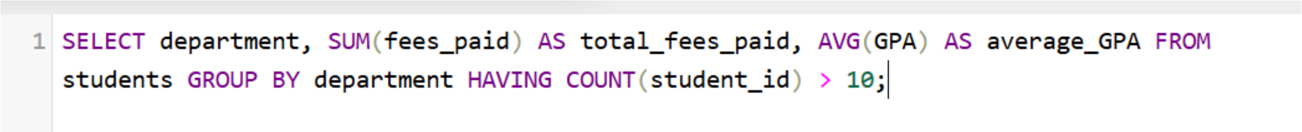
Output:



student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status

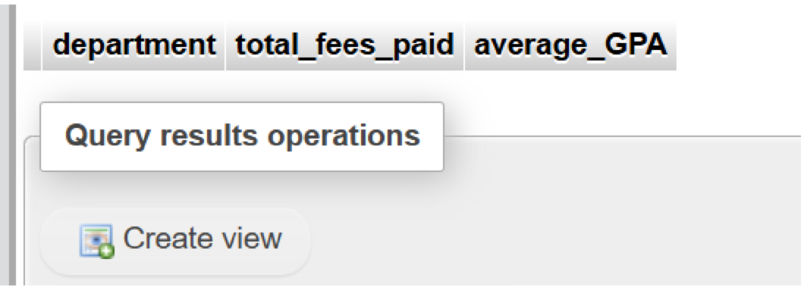
2.4.5 Calculate the total fees paid and average GPA for each department, but only for departments with more than 10 students

Query:



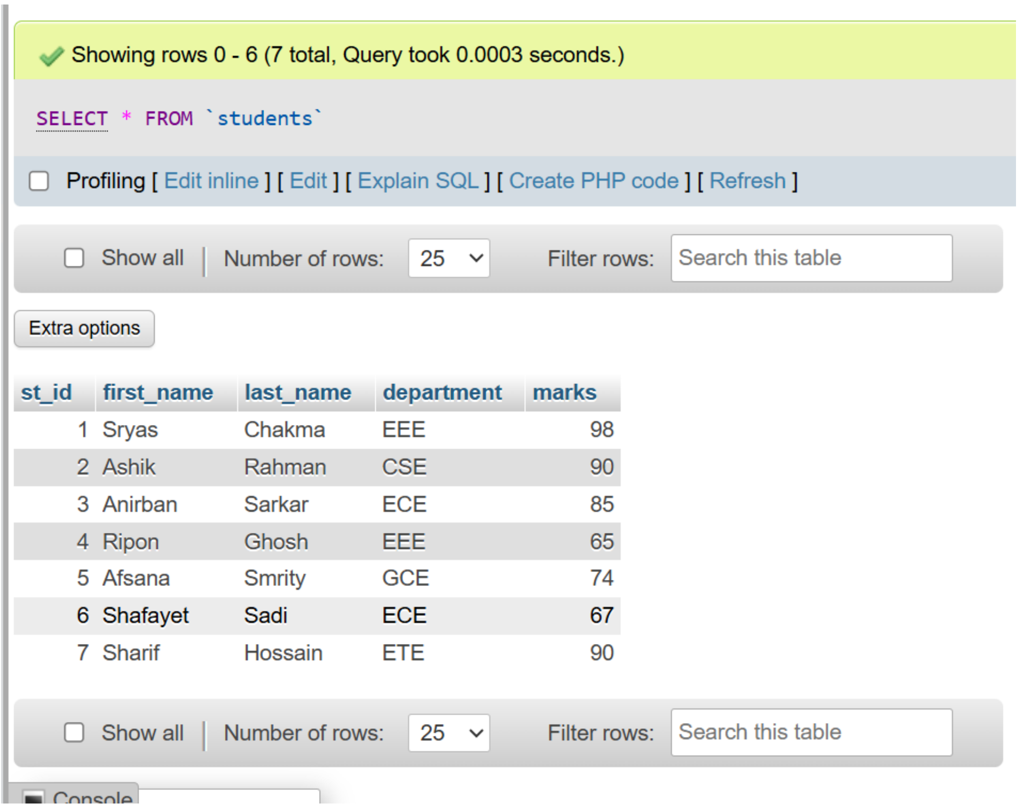
```sql
SELECT department, SUM(fees_paid) AS total_fees_paid, AVG(GPA) AS average_GPA FROM
students GROUP BY department HAVING COUNT(student_id) > 10;
```

Output:



department	total_fees_paid	average_GPA

Student Information Table:



Showing rows 0 - 6 (7 total, Query took 0.0003 seconds.)

```sql
SELECT * FROM `students`
```

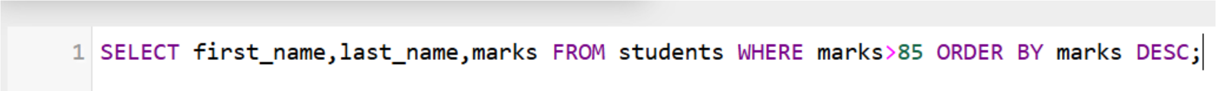
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

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

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

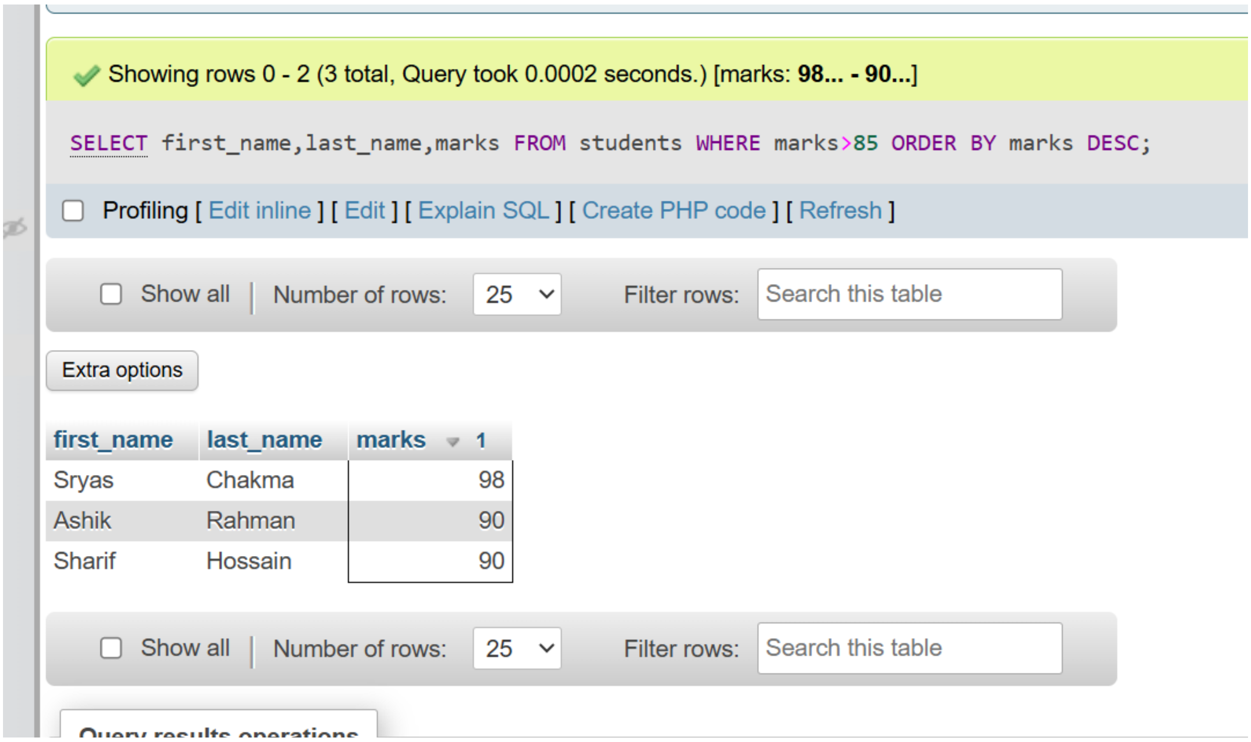
2.4.6 List students with marks greater than 85

Query:



```
SELECT first_name,last_name,marks FROM students WHERE marks>85 ORDER BY marks DESC;
```

Output:



Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.) [marks: **98... - 90...**]

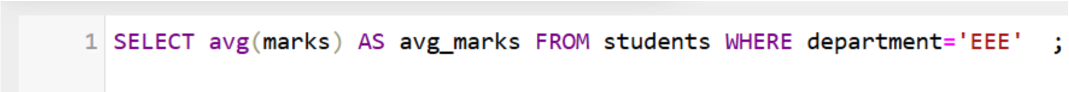
```
SELECT first_name,last_name,marks FROM students WHERE marks>85 ORDER BY marks DESC;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options
□ Show all
Number of rows: 25
Filter rows: Search this table
Query results operations

first_name	last_name	marks ▼ 1
Sryas	Chakma	98
Ashik	Rahman	90
Sharif	Hossain	90

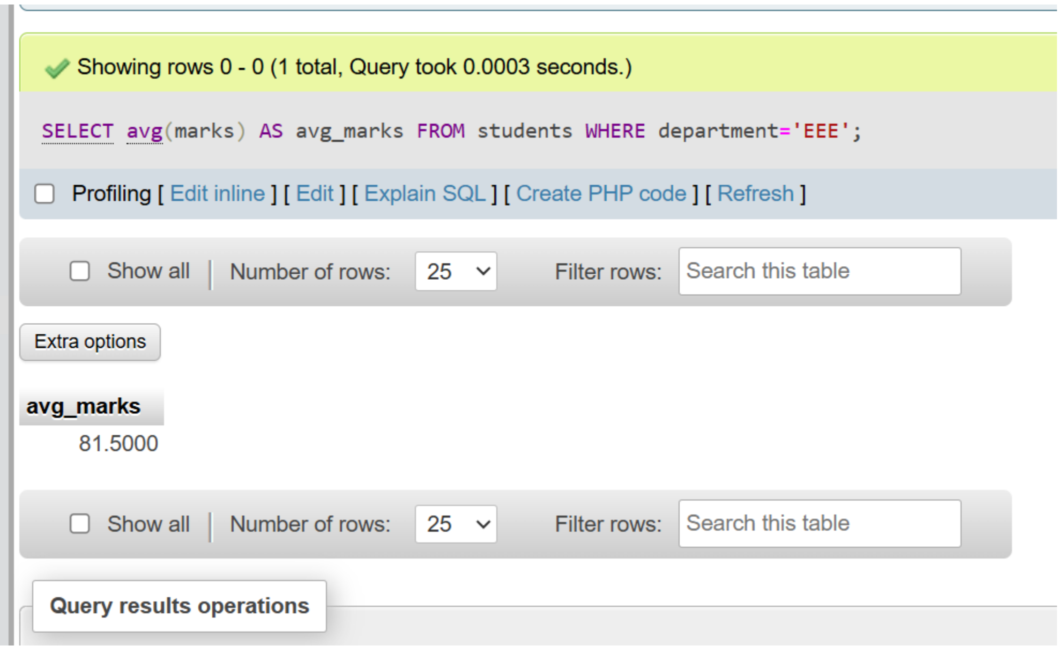
2.4.7 Find the average marks in department EEE

Query:



```
SELECT avg(marks) AS avg_marks FROM students WHERE department='EEE'  ;
```

Output:



✓ Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
SELECT avg(marks) AS avg_marks FROM students WHERE department='EEE';
```

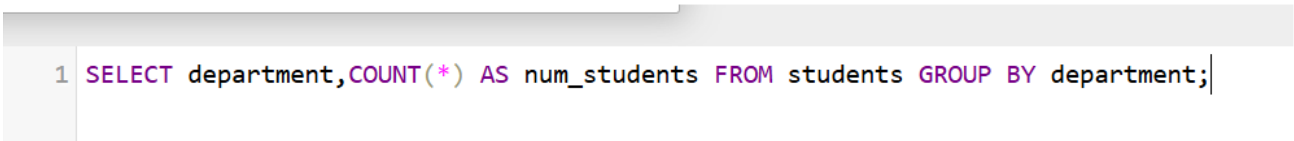
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

avg_marks
81.5000

□ Show all | Number of rows: 25 Filter rows: Search this table
Query results operations

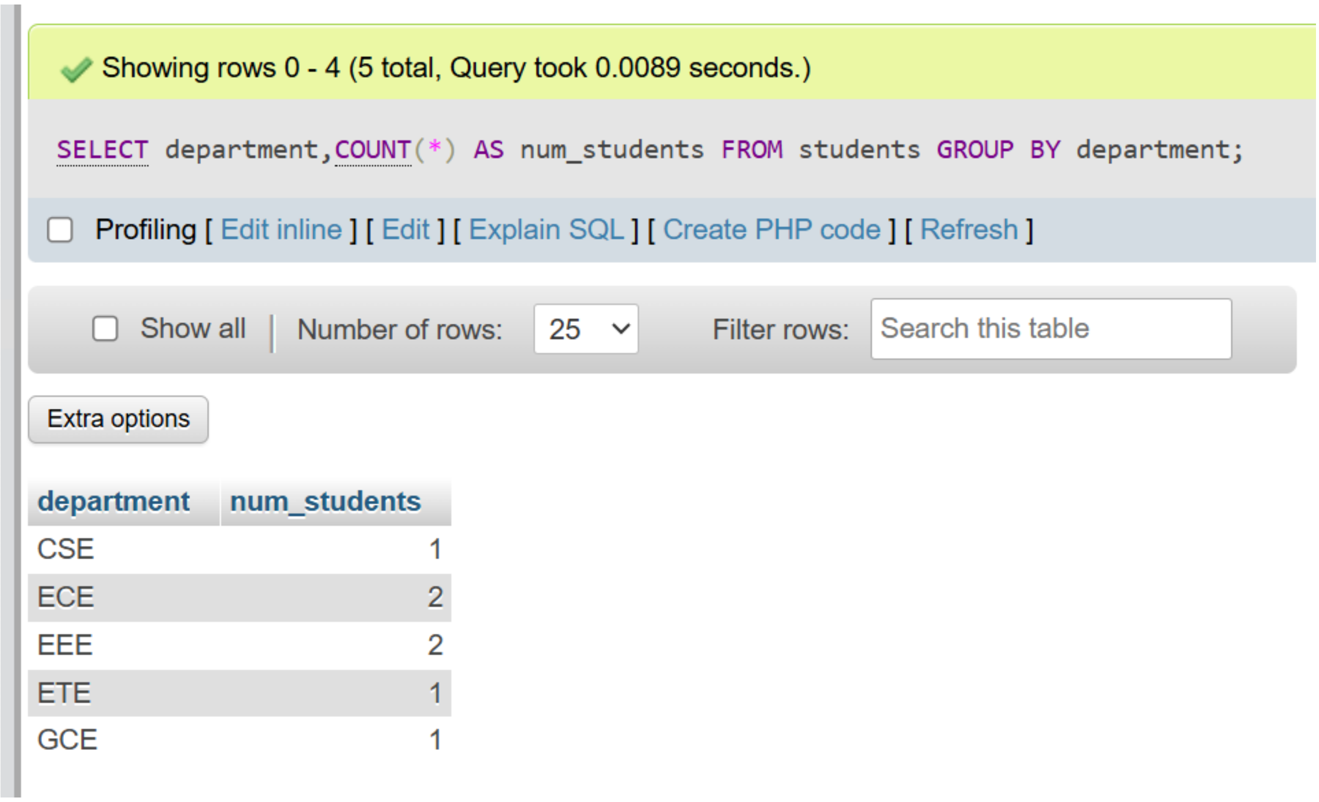
2.4.8 Count the number of students in each department

Query:



```
SELECT department,COUNT(*) AS num_students FROM students GROUP BY department;
```

Output:



✓ Showing rows 0 - 4 (5 total, Query took 0.0089 seconds.)

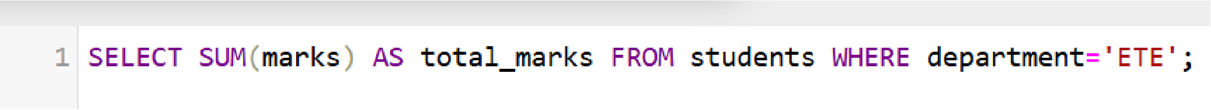
```
SELECT department,COUNT(*) AS num_students FROM students GROUP BY department;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

department	num_students
CSE	1
ECE	2
EEE	2
ETE	1
GCE	1

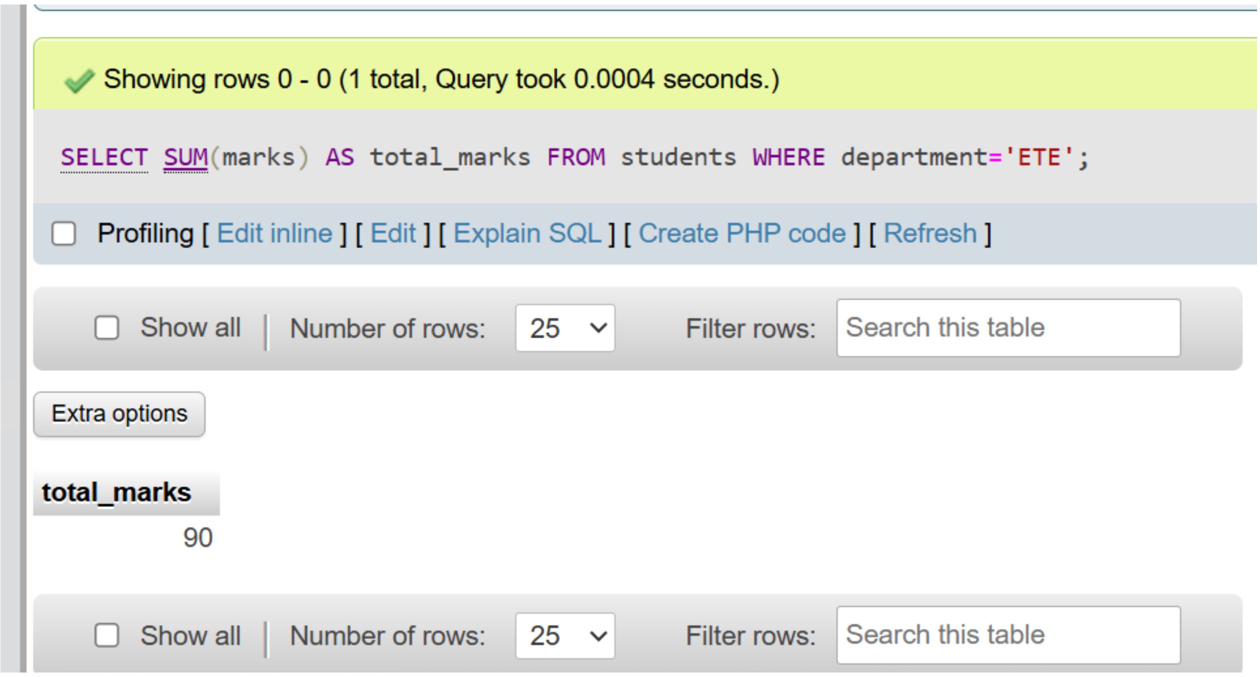
2.4.9 Calculate the total marks given in ETE department

Query:



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

Output:



✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

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

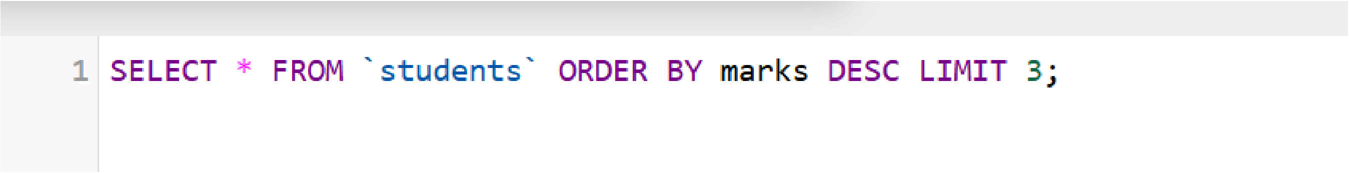
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

total_marks
90

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

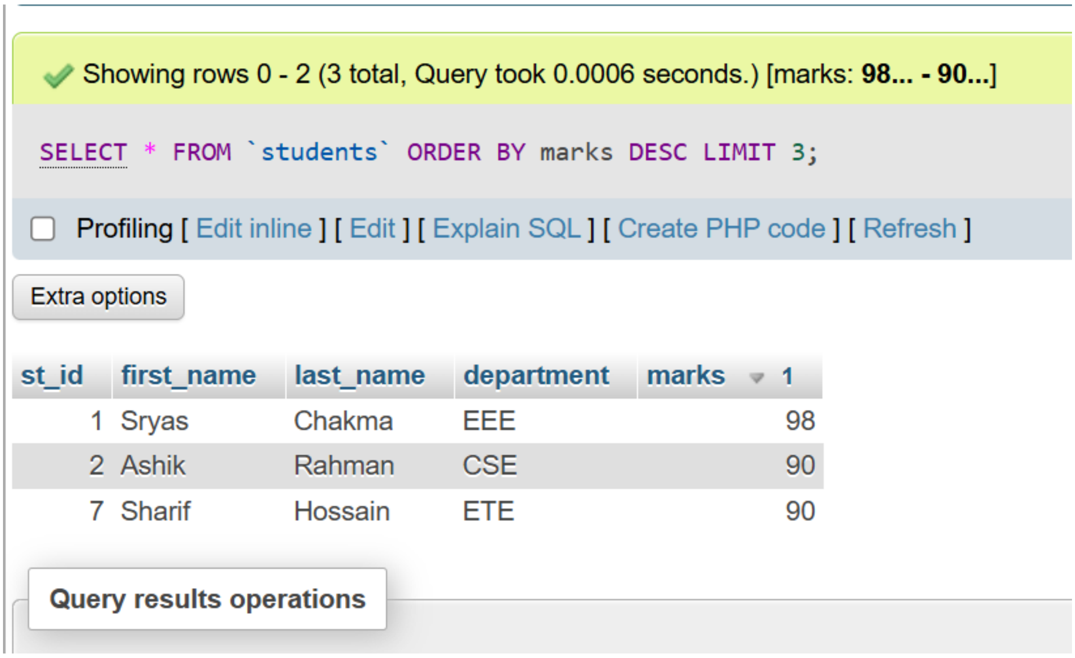
2.4.10 List the top three students

Query:



```sql
SELECT * FROM `students` ORDER BY marks DESC LIMIT 3;
```

Output:



✓ Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.) [marks: **98...** - **90...**]

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

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

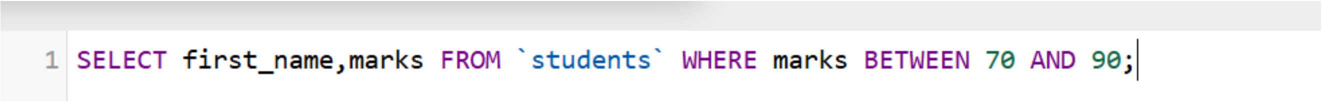
Extra options

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

Query results operations

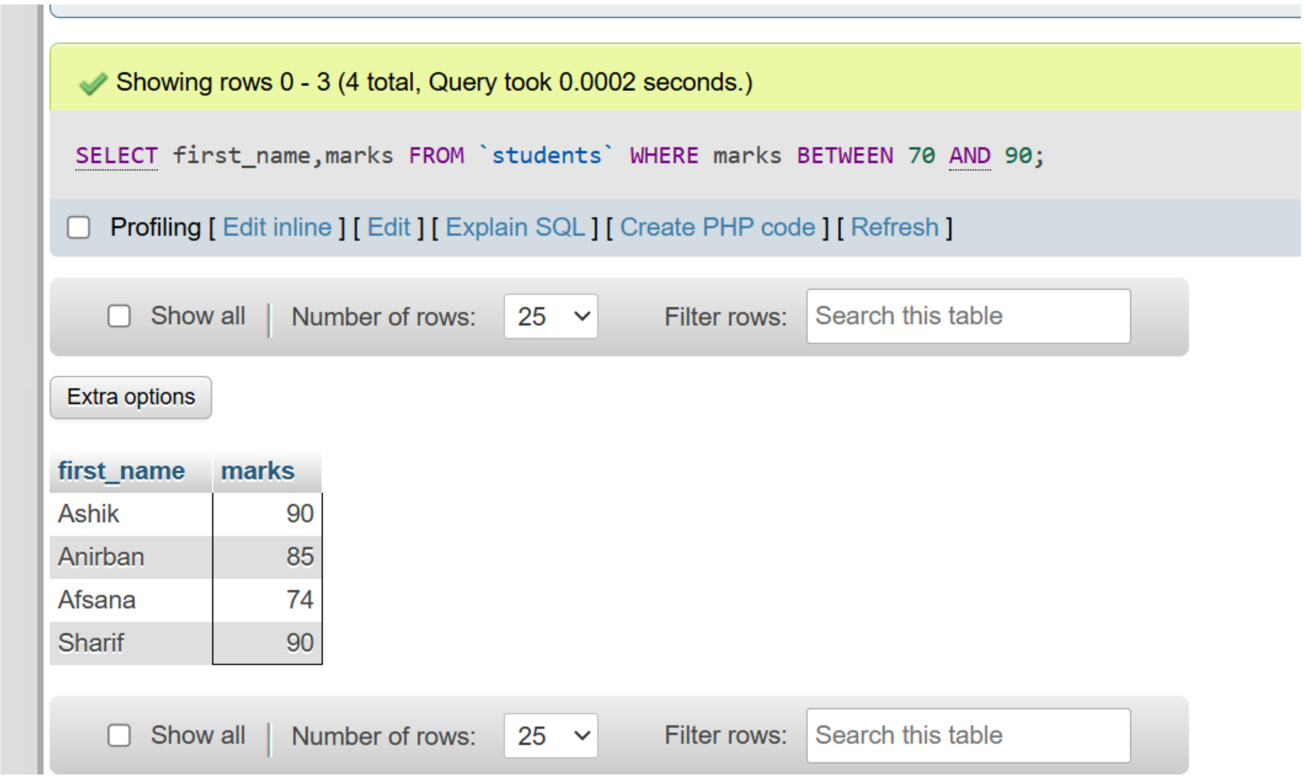
2.4.11 Find students whose marks are between 70 to 90

Query:



```
SELECT first_name,marks FROM `students` WHERE marks BETWEEN 70 AND 90;
```

Output:



✓ Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

```
SELECT first_name,marks FROM `students` WHERE marks BETWEEN 70 AND 90;
```

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

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

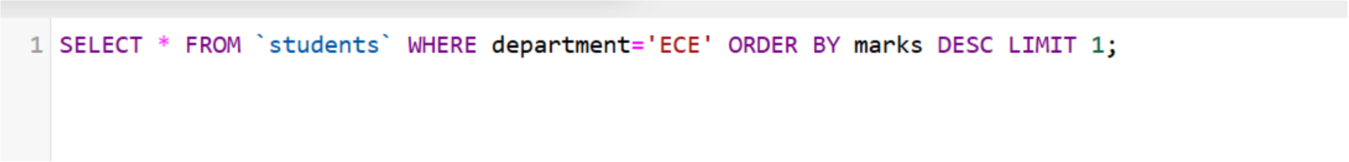
Extra options

first_name	marks
Ashik	90
Anirban	85
Afsana	74
Sharif	90

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

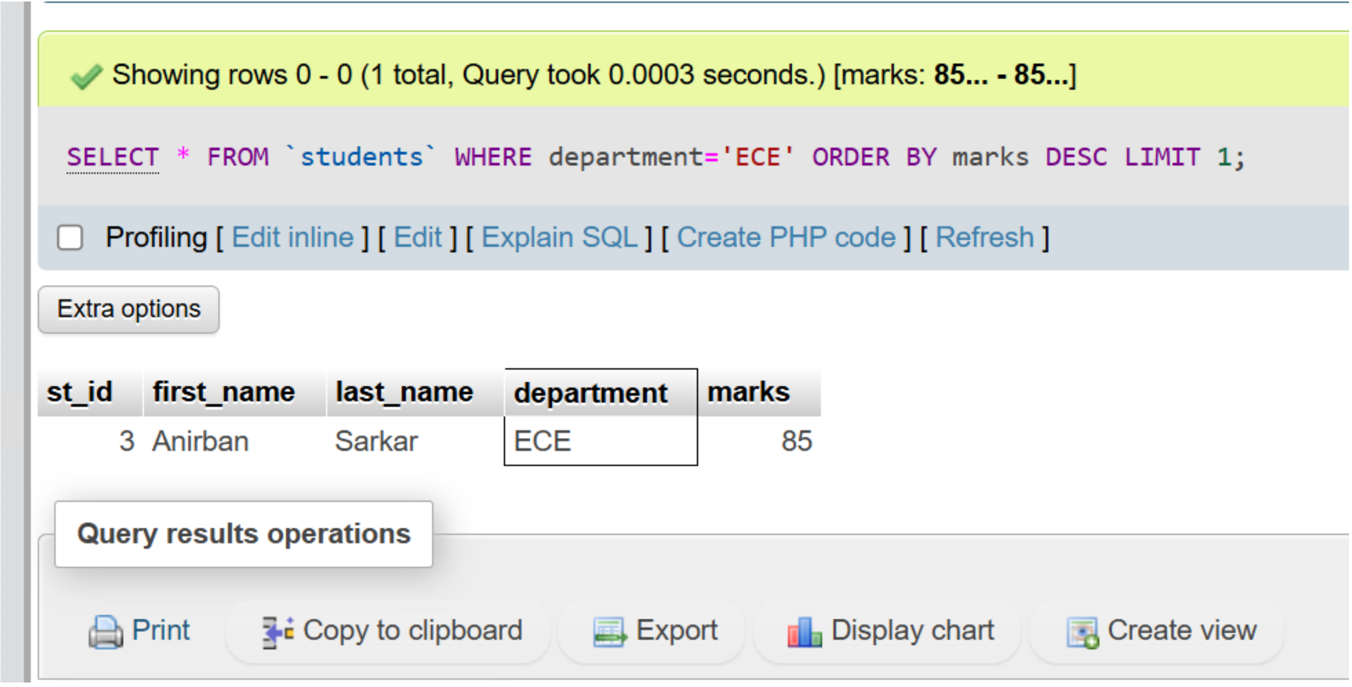
2.4.12 List student in ECE department, limited to first one

Query:



```
SELECT * FROM `students` WHERE department='ECE' ORDER BY marks DESC LIMIT 1;
```

Output:



✓ Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.) [marks: **85... - 85...**]

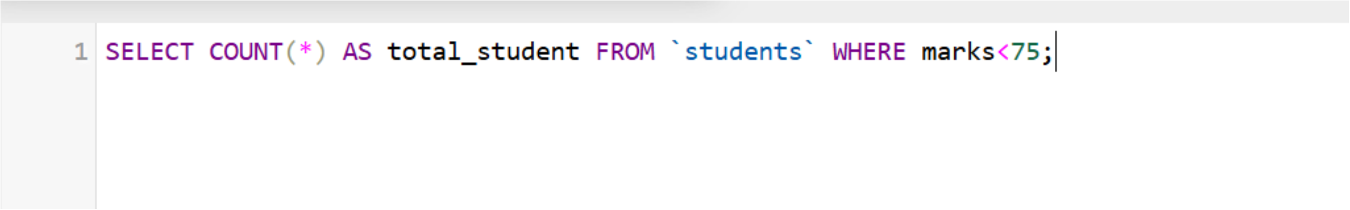
```
SELECT * FROM `students` WHERE department='ECE' ORDER BY marks DESC LIMIT 1;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

st_id	first_name	last_name	department	marks
3	Anirban	Sarkar	ECE	85

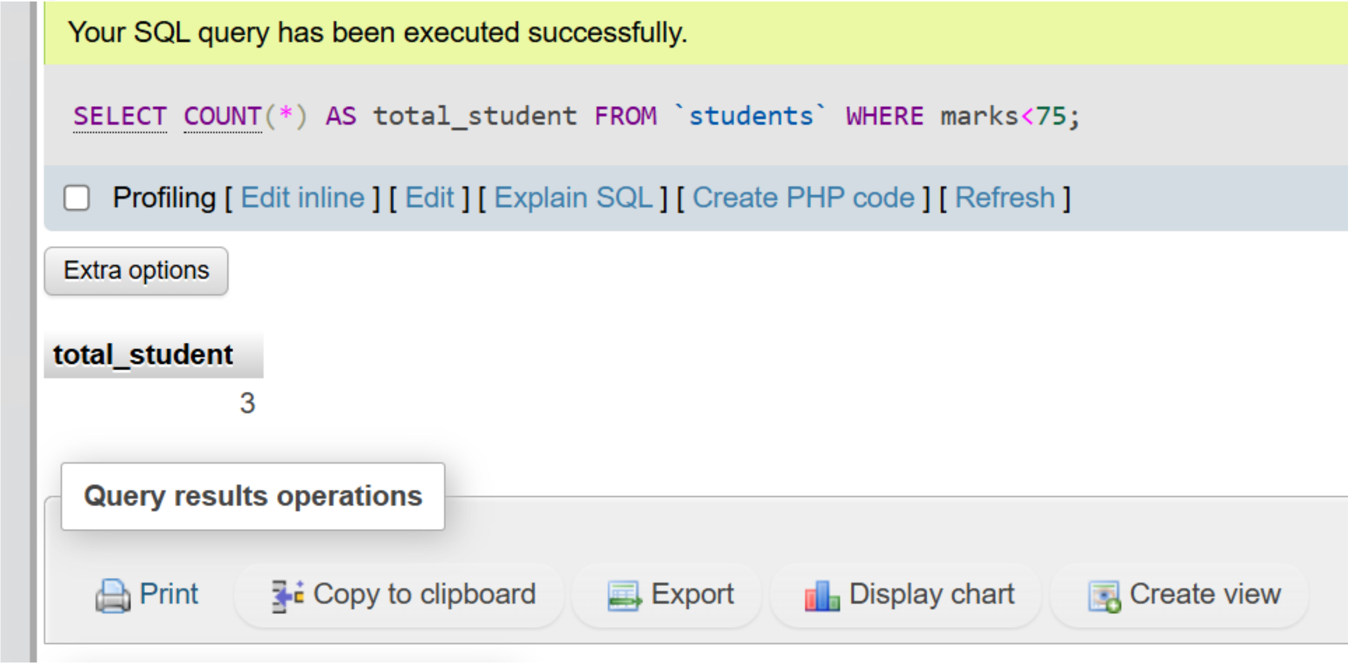
2.4.13 Count the number of student having less than 75 marks

Query:



```sql
SELECT COUNT(*) AS total_student FROM `students` WHERE marks<75;
```

Output:



Your SQL query has been executed successfully.

```sql
SELECT COUNT(*) AS total_student FROM `students` WHERE marks<75;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options

total_student
3