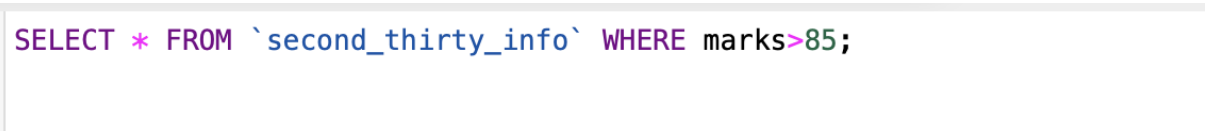
Experiment No:02
Experiment Name: Querying and Manipulating Student Marks Data
Tasks:
1.List students with marks greater than 85
2.Find the average marks in department EEE
3.Count the number of students in each department
4.Calculate the total marks given in ETC department
5.List the top 3 students
6.Find the students where marks are between 70 to 90
7.List student in ECE department limited to first one
8.Count the number of students having less than 75 marks
Theory: SQL (Structured Query Language) plays a pivotal role in managing and querying data within relational databases. It empowers users to efficiently filter, aggregate, and organize information. For example, common SQL operations might involve listing students who scored above 85, calculating the average marks within specific departments, counting the number of students in each department, or summing total marks for particular departments like ETC. These tasks are achieved through essential SQL functions such as `SELECT`, `WHERE`, `AVG()`, `SUM()`, and `COUNT()`, streamlining data retrieval and summarization.

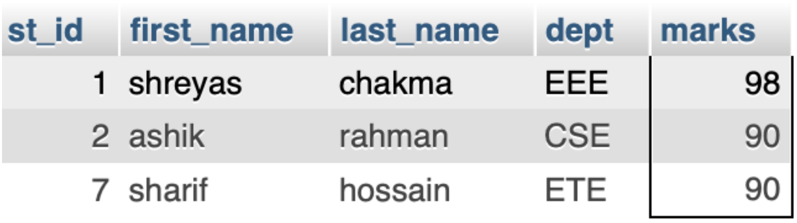
SQL also enables more sophisticated queries, such as ranking students using `ORDER BY`, limiting query results with `LIMIT`, and filtering marks within specific ranges via the `BETWEEN` operator. These features allow for identifying high-performing students, extracting department-specific insights, and finding students who fall within certain score brackets. Mastering these SQL techniques is crucial for analyzing academic datasets and making informed decisions in both educational and professional environments.

By leveraging SQL's powerful capabilities, users can transform raw academic data into actionable insights, improving decision-making processes and enabling a deeper understanding of performance across departments and student groups.
Code:



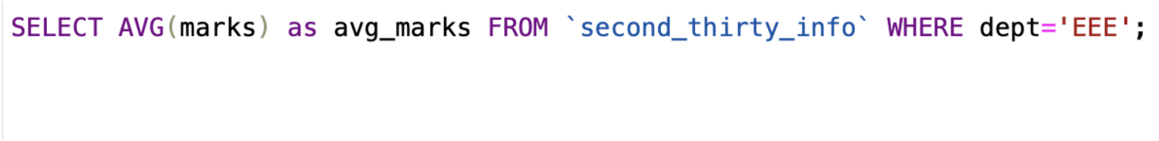
```
SELECT * FROM `second_thirty_info` WHERE marks>85;
```

Output:



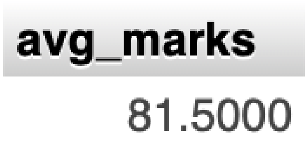
st_id	first_name	last_name	dept	marks
1	shreyas	chakma	EEE	98
2	ashik	rahman	CSE	90
7	sharif	hossain	ETE	90

Code:



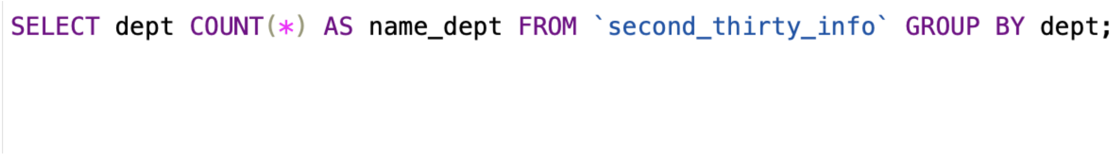
```
SELECT AVG(marks) as avg_marks FROM `second_thirty_info` WHERE dept='EEE';
```

Output:



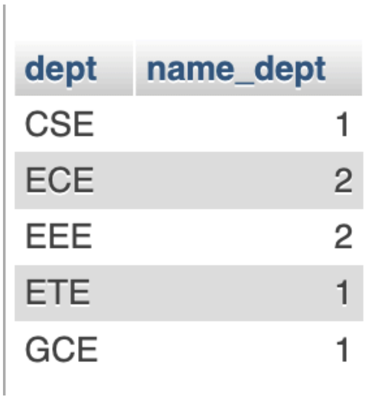
avg_marks
81.5000

Code:



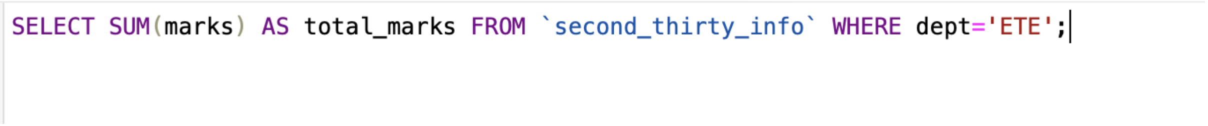
```
SELECT dept COUNT(*) AS name_dept FROM `second_thirty_info` GROUP BY dept;
```

Output:



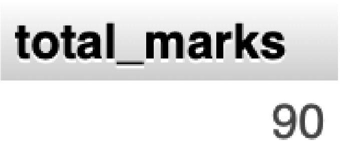
dept	name_dept
CSE	1
ECE	2
EEE	2
ETE	1
GCE	1

Code:



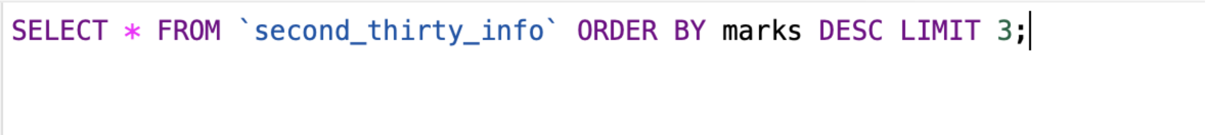
```
SELECT SUM(marks) AS total_marks FROM `second_thirty_info` WHERE dept='ETE';
```

Output:



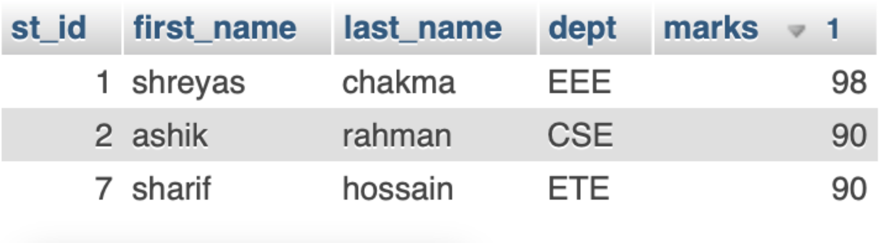
total_marks
90

Code:



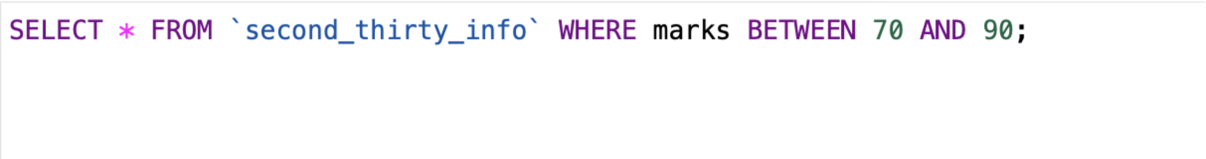
```
SELECT * FROM `second_thirty_info` ORDER BY marks DESC LIMIT 3;
```

Output:



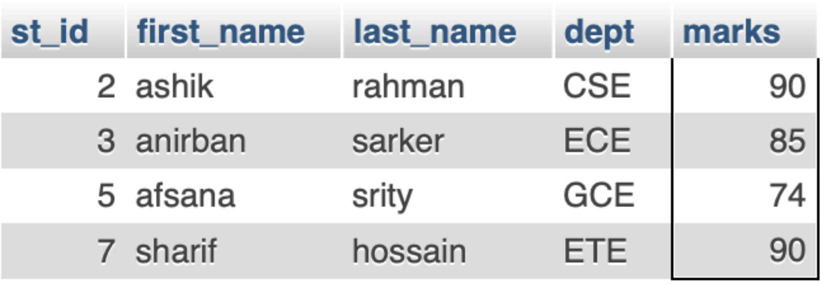
st_id	first_name	last_name	dept	marks ▼ 1
1	shreyas	chakma	EEE	98
2	ashik	rahman	CSE	90
7	sharif	hossain	ETE	90

Code:



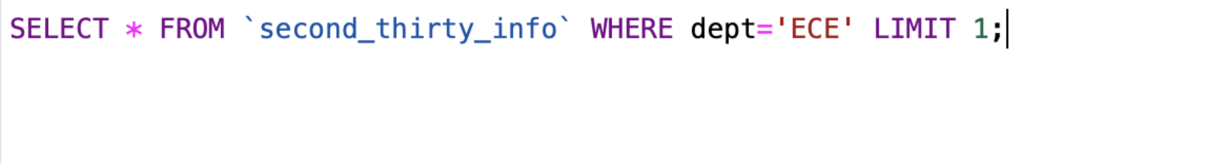
```
SELECT * FROM `second_thirty_info` WHERE marks BETWEEN 70 AND 90;
```

Output:



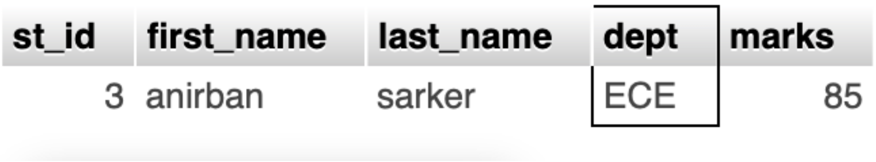
st_id	first_name	last_name	dept	marks
2	ashik	rahman	CSE	90
3	anirban	sarker	ECE	85
5	afsana	srity	GCE	74
7	sharif	hossain	ETE	90

Code:



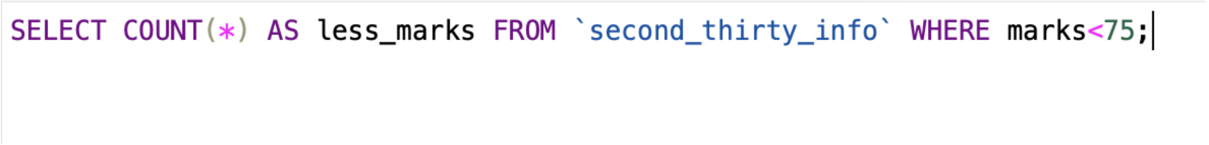
```
SELECT * FROM `second_thirty_info` WHERE dept='ECE' LIMIT 1;
```

Output:



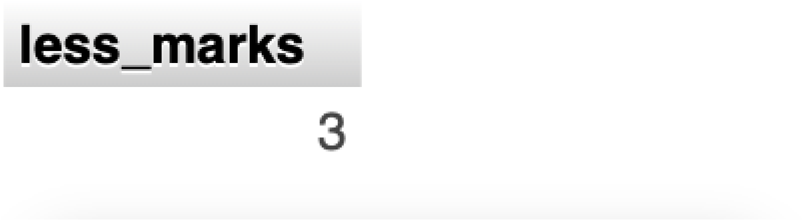
st_id	first_name	last_name	dept	marks
3	anirban	sarker	ECE	85

Code:



```
SELECT COUNT(*) AS less_marks FROM `second_thirty_info` WHERE marks<75;
```

Output:



less_marks
3