Heavens Light is Our Guide

Rajshahi University of Engineering & Technology

Course Title

Data Base Systems Sessional

Course No: ECE 2216

Lab Report No: 02

Date of submission: 01 October 2024

Submitted by:	**Submitted to:**
Zanifa Islam Roll: 2110008 Reg. No.:1062/2021-2022 Department of ECE	Oishi Jyoti Assistant Professor, Department of ECE, Rajshahi University of Engineering Technology, Rajshahi.

Experiment No: 02

Experiment Name: Different Operations On Table.

Objective:

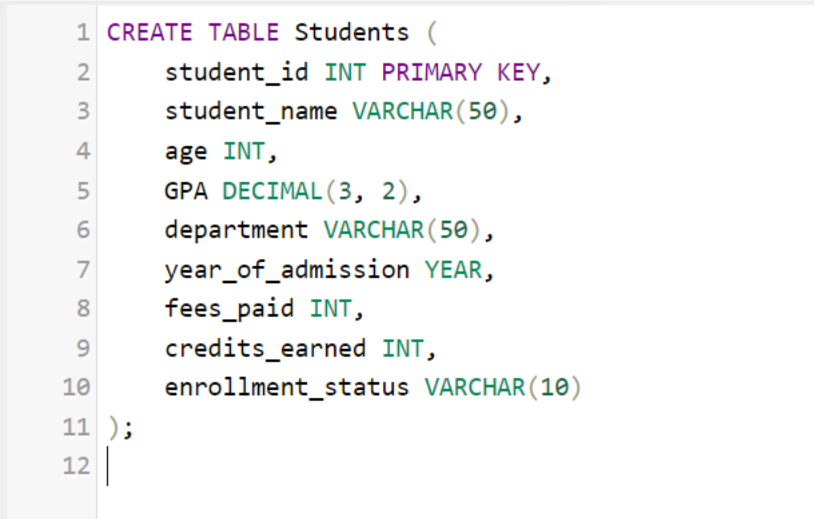
- Query and filter student data based on specific conditions such as age, GPA, and department.
- Analyze student performance by comparing GPA with the average GPA of all students.
- Rank students by fees paid and use GPA as a tie-breaking factor.
- Filter students based on department, GPA, and enrollment year.
- Identify inactive students and those who have not paid any fees.
- Calculate department-wise statistics, such as total fees paid and average GPA, while filtering for departments with more than 10 students.
- Demonstrate effective use of SQL for data extraction, analysis, and aggregation in a database system.

Tasks:

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.

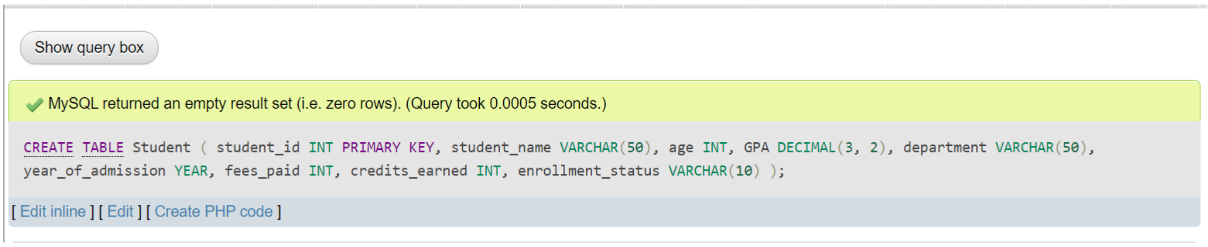
Query And Outputs:

Query For Creating Table:



```sql
CREATE TABLE Students (
    student_id INT PRIMARY KEY,
    student_name VARCHAR(50),
    age INT,
    GPA DECIMAL(3, 2),
    department VARCHAR(50),
    year_of_admission YEAR,
    fees_paid INT,
    credits_earned INT,
    enrollment_status VARCHAR(10)
);
```

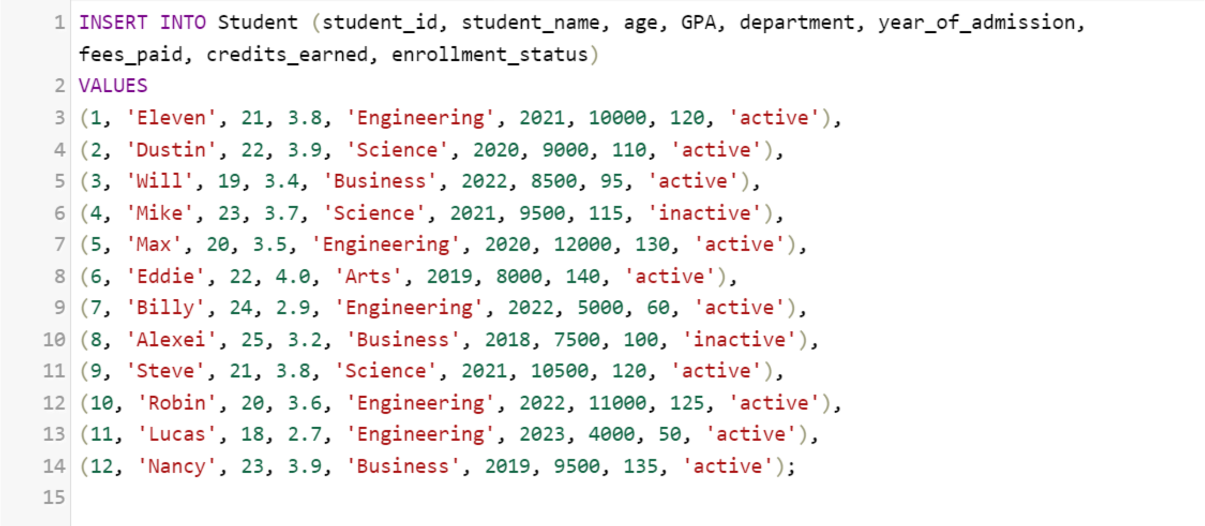
Output:



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```sql
CREATE TABLE Student ( student_id INT PRIMARY KEY, student_name VARCHAR(50), age INT, GPA DECIMAL(3, 2), department VARCHAR(50),
year_of_admission YEAR, fees_paid INT, credits_earned INT, enrollment_status VARCHAR(10) );
```

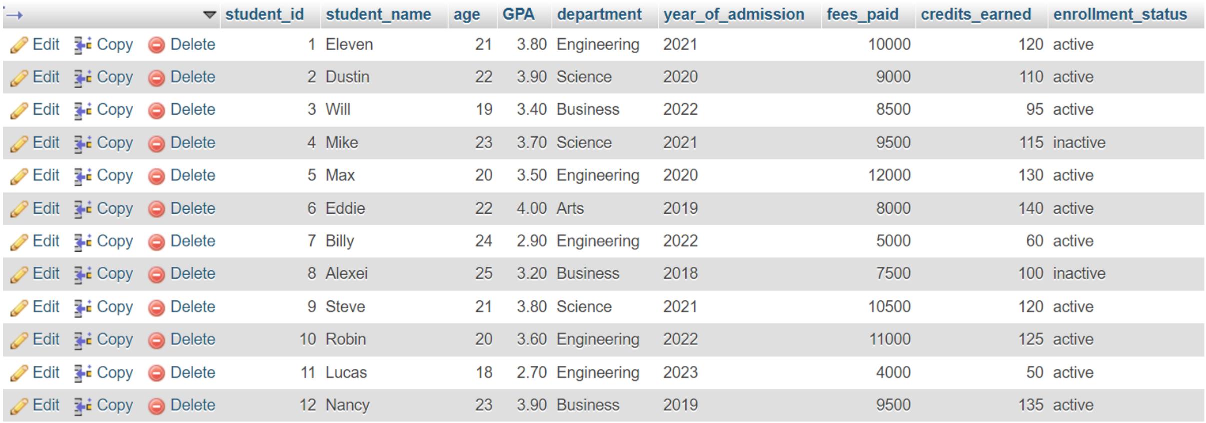
Inserting Information Into Table:



```sql
INSERT INTO Student (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');

```

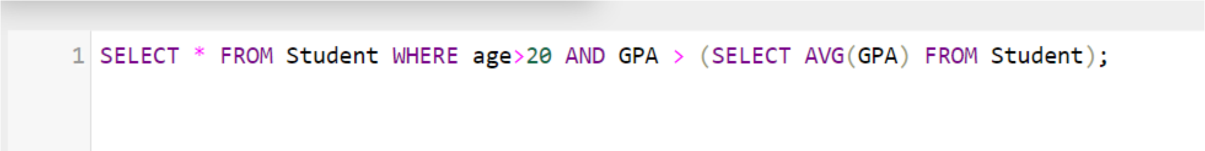
Output:



← T →			student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
Edit	Copy	Delete	1	Eleven	21	3.80	Engineering	2021	10000	120	active
Edit	Copy	Delete	2	Dustin	22	3.90	Science	2020	9000	110	active
Edit	Copy	Delete	3	Will	19	3.40	Business	2022	8500	95	active
Edit	Copy	Delete	4	Mike	23	3.70	Science	2021	9500	115	inactive
Edit	Copy	Delete	5	Max	20	3.50	Engineering	2020	12000	130	active
Edit	Copy	Delete	6	Eddie	22	4.00	Arts	2019	8000	140	active
Edit	Copy	Delete	7	Billy	24	2.90	Engineering	2022	5000	60	active
Edit	Copy	Delete	8	Alexei	25	3.20	Business	2018	7500	100	inactive
Edit	Copy	Delete	9	Steve	21	3.80	Science	2021	10500	120	active
Edit	Copy	Delete	10	Robin	20	3.60	Engineering	2022	11000	125	active
Edit	Copy	Delete	11	Lucas	18	2.70	Engineering	2023	4000	50	active
Edit	Copy	Delete	12	Nancy	23	3.90	Business	2019	9500	135	active

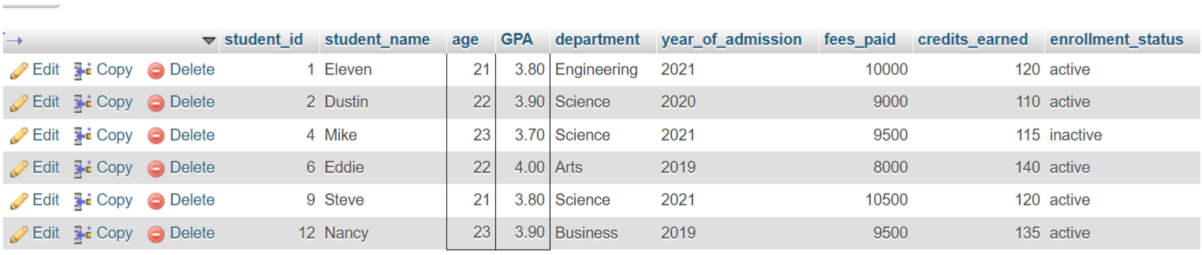
Task-1:

Code:



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

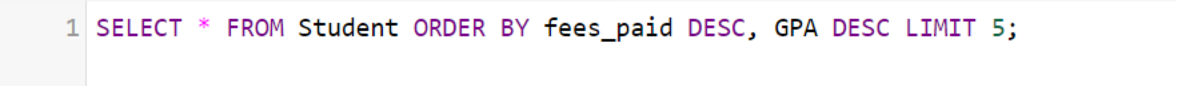
Output:



	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
Edit Copy Delete	1	Eleven	21	3.80	Engineering	2021	10000	120	active
Edit Copy Delete	2	Dustin	22	3.90	Science	2020	9000	110	active
Edit Copy Delete	4	Mike	23	3.70	Science	2021	9500	115	inactive
Edit Copy Delete	6	Eddie	22	4.00	Arts	2019	8000	140	active
Edit Copy Delete	9	Steve	21	3.80	Science	2021	10500	120	active
Edit Copy Delete	12	Nancy	23	3.90	Business	2019	9500	135	active

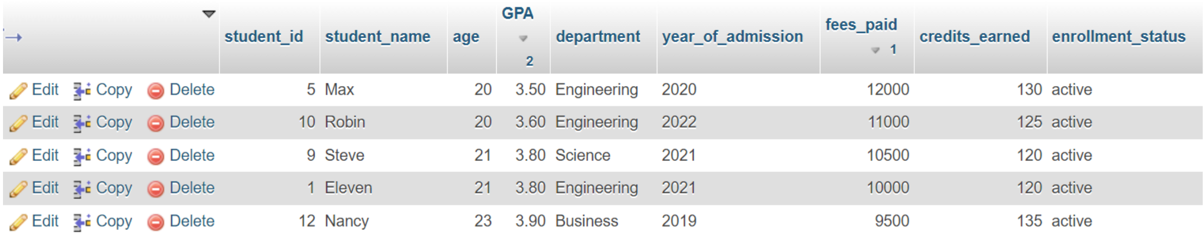
Task-2:

Code:



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

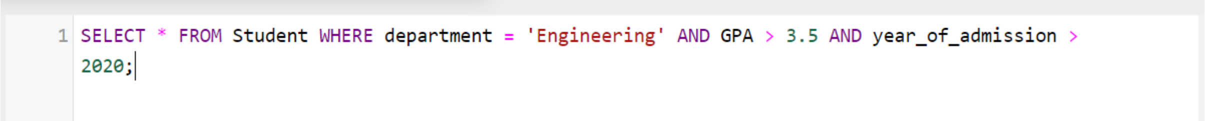
Output:



	student_id	student_name	age	GPA 2	department	year_of_admission	fees_paid 1	credits_earned	enrollment_status
Edit Copy Delete	5	Max	20	3.50	Engineering	2020	12000	130	active
Edit Copy Delete	10	Robin	20	3.60	Engineering	2022	11000	125	active
Edit Copy Delete	9	Steve	21	3.80	Science	2021	10500	120	active
Edit Copy Delete	1	Eleven	21	3.80	Engineering	2021	10000	120	active
Edit Copy Delete	12	Nancy	23	3.90	Business	2019	9500	135	active

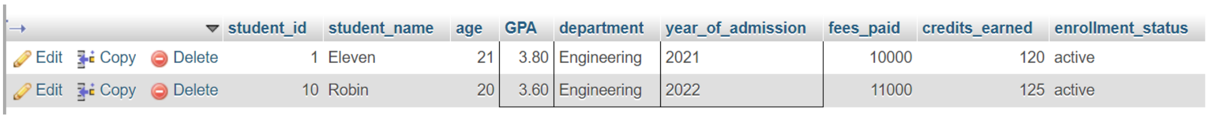
Task-3:

Code:



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

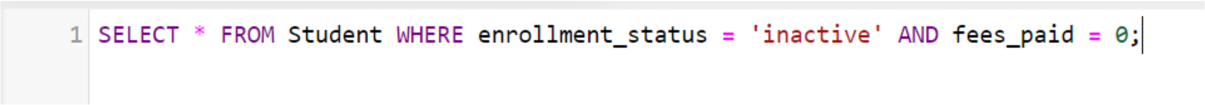
Output:



	student_id	student_name	age	GPA	department	year_of_admission	fees_paid	credits_earned	enrollment_status
Edit Copy Delete	1	Eleven	21	3.80	Engineering	2021	10000	120	active
Edit Copy Delete	10	Robin	20	3.60	Engineering	2022	11000	125	active

Task-4:

Code:

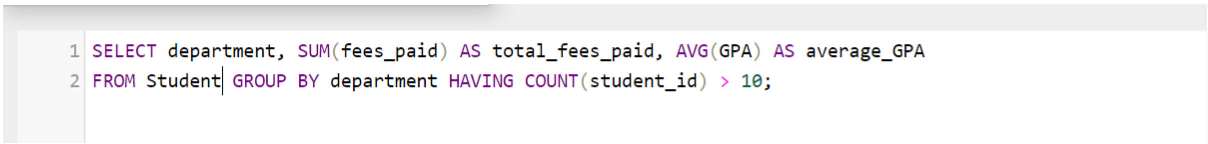


```sql
SELECT * FROM Student WHERE enrollment_status = 'inactive' AND fees_paid = 0;
```

Output:

Task-5:

Code:



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

Output: