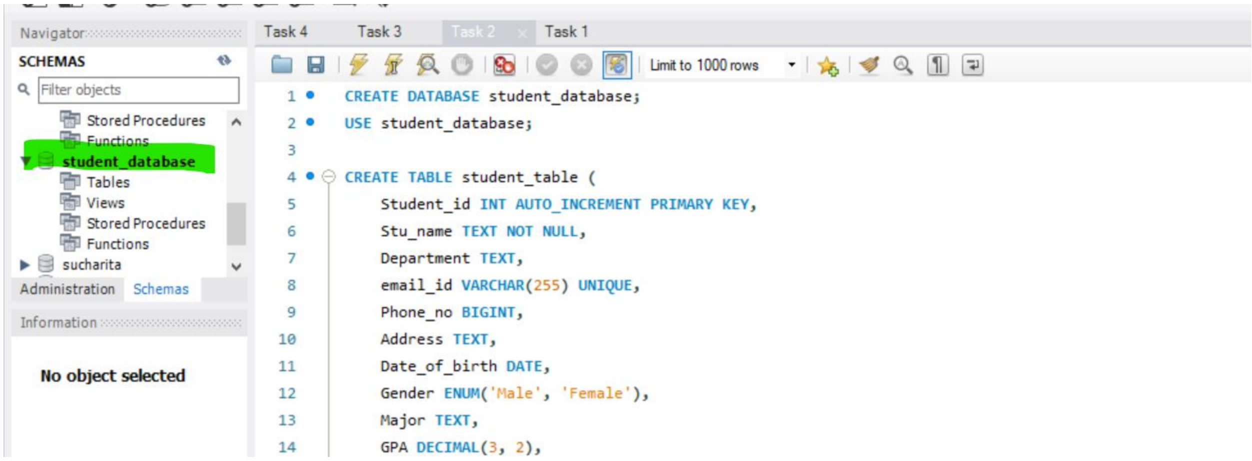
OUTPUT SHEET

Project: Student Database Management System(MySQL)

Objective: Design and implement a student database management system using PostgreSQL that allows storing and retrieving student information efficiently. The project will include the following tasks:

1. Database Setup

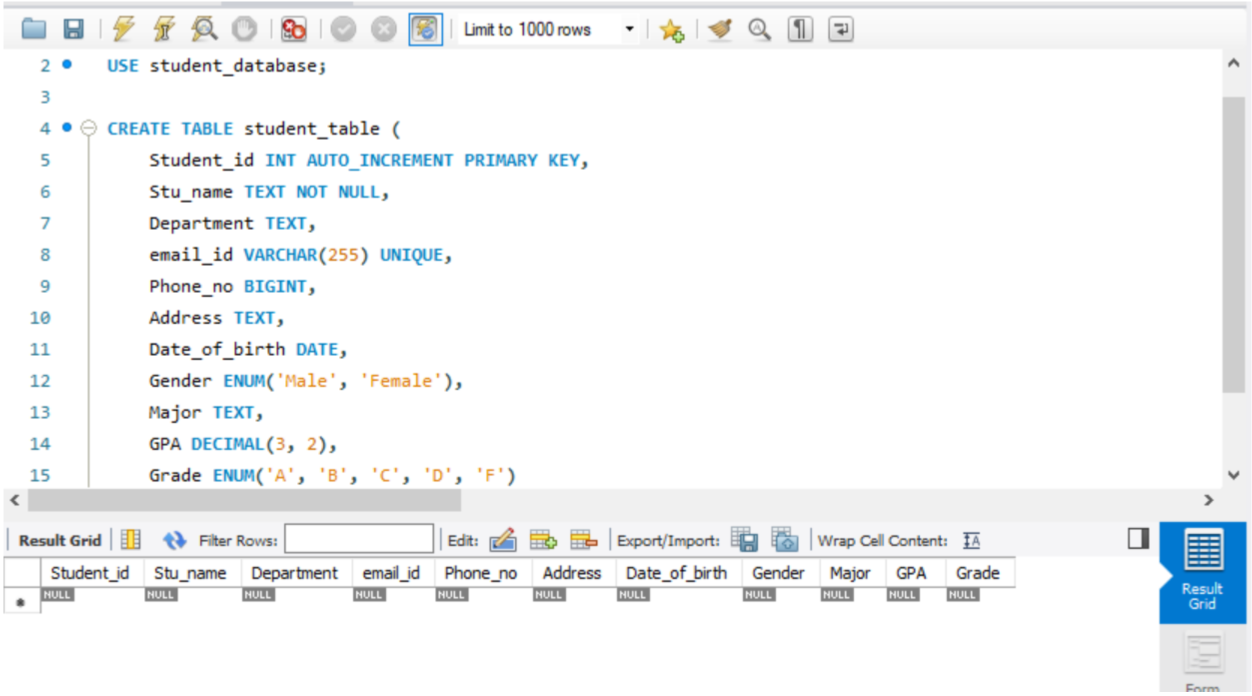
Create a database named "student_database."



```
CREATE DATABASE student_database;
USE student_database;

CREATE TABLE student_table (
    Student_id INT AUTO_INCREMENT PRIMARY KEY,
    Stu_name TEXT NOT NULL,
    Department TEXT,
    email_id VARCHAR(255) UNIQUE,
    Phone_no BIGINT,
    Address TEXT,
    Date_of_birth DATE,
    Gender ENUM('Male', 'Female'),
    Major TEXT,
    GPA DECIMAL(3, 2),
```

Create a table called " student_table " with the following columns: Student_id (integer), Stu_name (text), Department (text), email_id (text),Phone_no (numeric), Address (text), Date_of_birth (date), Gender (text), Major (text), GPA (numeric),Grade (text) should be A,B,C etc.



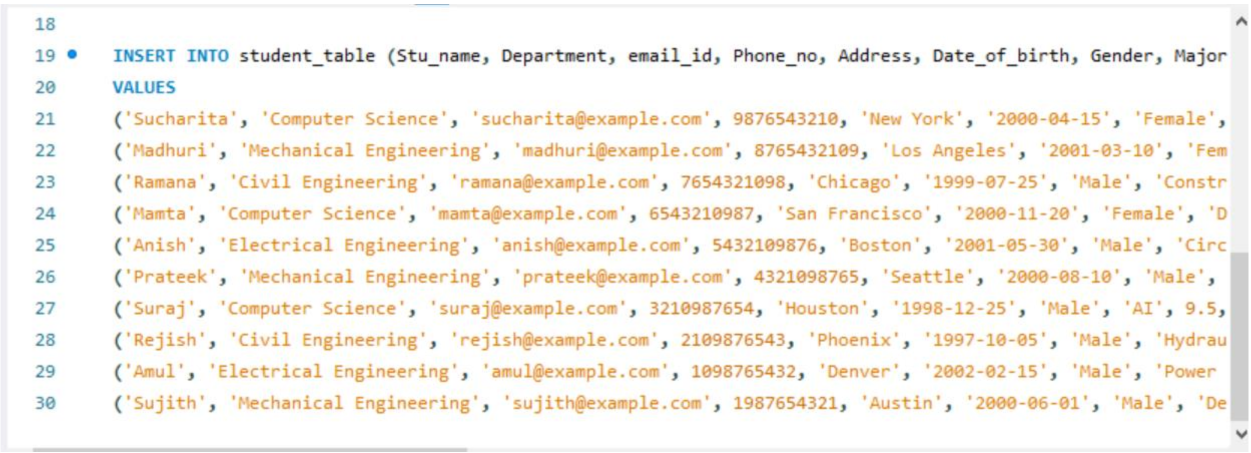
```
USE student_database;

CREATE TABLE student_table (
    Student_id INT AUTO_INCREMENT PRIMARY KEY,
    Stu_name TEXT NOT NULL,
    Department TEXT,
    email_id VARCHAR(255) UNIQUE,
    Phone_no BIGINT,
    Address TEXT,
    Date_of_birth DATE,
    Gender ENUM('Male', 'Female'),
    Major TEXT,
    GPA DECIMAL(3, 2),
    Grade ENUM('A', 'B', 'C', 'D', 'F')
```

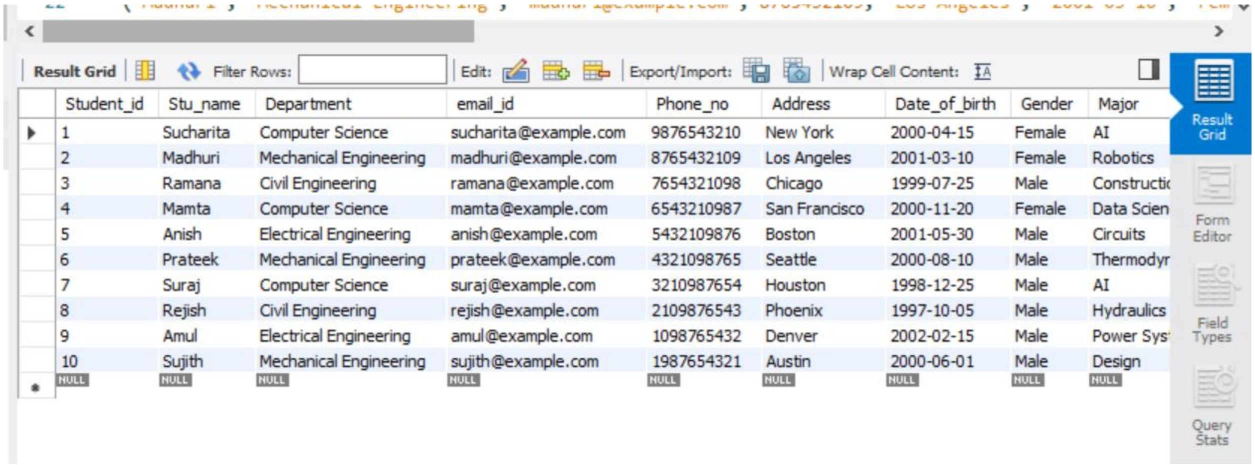
	Student_id	Stu_name	Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major	GPA	Grade
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Data Entry

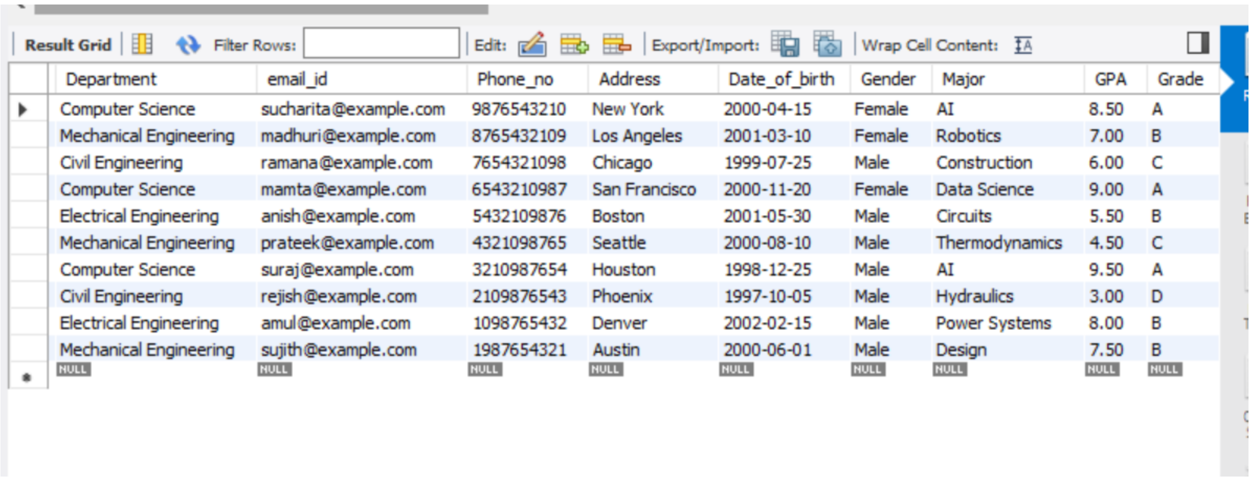
Insert 10 sample records into the "student_table" using INSERT command.



```
INSERT INTO student_table (Stu_name, Department, email_id, Phone_no, Address, Date_of_birth, Gender, Major
VALUES
('Sucharita', 'Computer Science', 'sucharita@example.com', 9876543210, 'New York', '2000-04-15', 'Female',
('Madhuri', 'Mechanical Engineering', 'madhuri@example.com', 8765432109, 'Los Angeles', '2001-03-10', 'Fem
('Ramana', 'Civil Engineering', 'ramana@example.com', 7654321098, 'Chicago', '1999-07-25', 'Male', 'Constr
('Mamta', 'Computer Science', 'mamta@example.com', 6543210987, 'San Francisco', '2000-11-20', 'Female', 'D
('Anish', 'Electrical Engineering', 'anish@example.com', 5432109876, 'Boston', '2001-05-30', 'Male', 'Circ
('Prateek', 'Mechanical Engineering', 'prateek@example.com', 4321098765, 'Seattle', '2000-08-10', 'Male',
('Suraj', 'Computer Science', 'suraj@example.com', 3210987654, 'Houston', '1998-12-25', 'Male', 'AI', 9.5,
('Rejish', 'Civil Engineering', 'rejish@example.com', 2109876543, 'Phoenix', '1997-10-05', 'Male', 'Hydrau
('Amul', 'Electrical Engineering', 'amul@example.com', 1098765432, 'Denver', '2002-02-15', 'Male', 'Power
('Sujith', 'Mechanical Engineering', 'sujith@example.com', 1987654321, 'Austin', '2000-06-01', 'Male', 'De
```



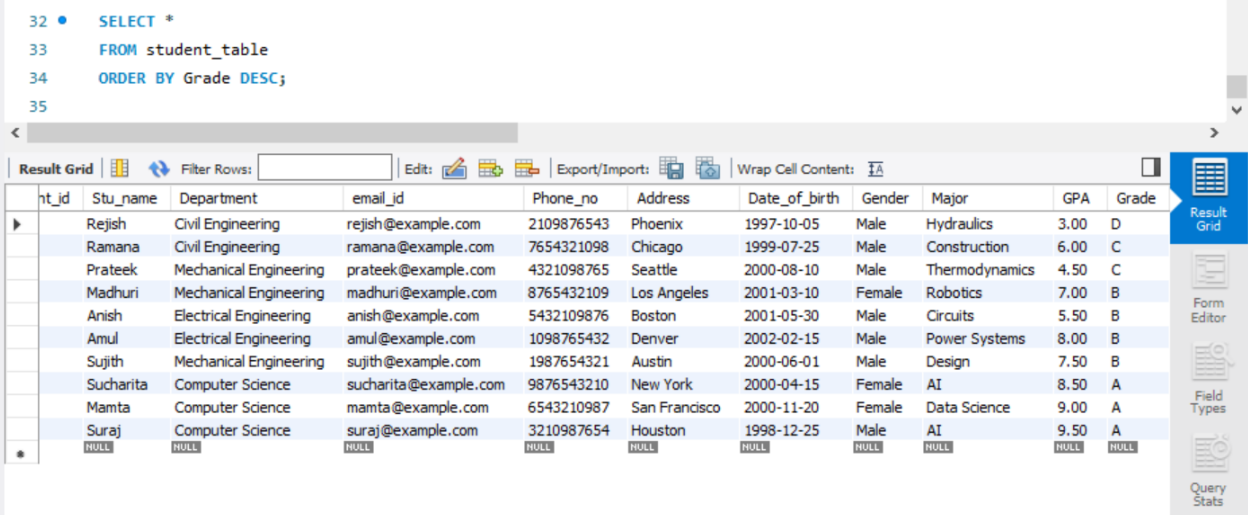
Student_id	Stu_name	Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major
1	Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI
2	Madhuri	Mechanical Engineering	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics
3	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Constructio
4	Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Scien
5	Anish	Electrical Engineering	anish@example.com	5432109876	Boston	2001-05-30	Male	Circuits
6	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodyr
7	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI
8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics
9	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Sys
10	Sujith	Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major	GPA	Grade
Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	8.50	A
Mechanical Engineering	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	7.00	B
Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00	C
Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	9.00	A
Electrical Engineering	anish@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50	B
Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50	A
Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00	B
Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50	B
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Student Information Retrieval

Develop a query to retrieve all students' information from the "student_table" and sort them in descending order by their grade.

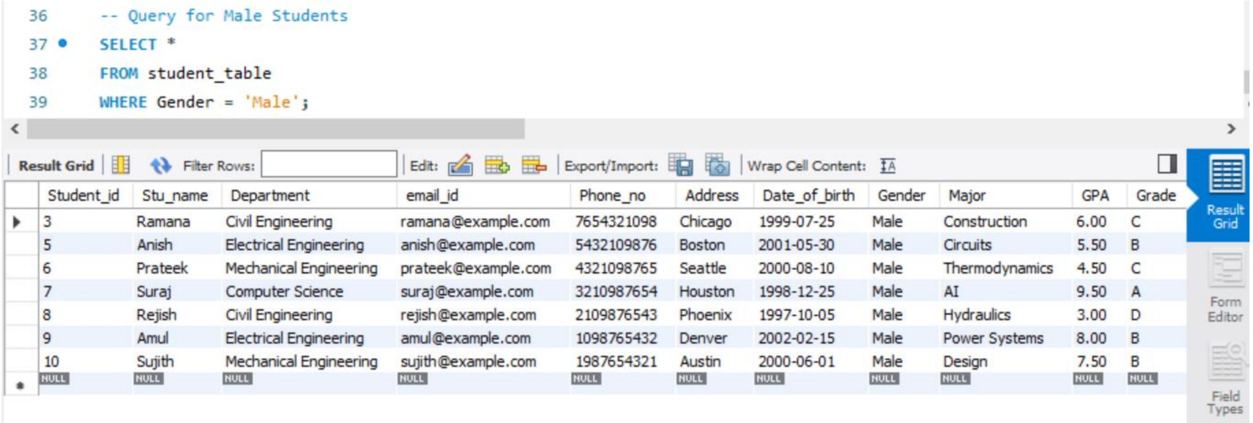


```
SELECT *
FROM student_table
ORDER BY Grade DESC;
```

nt_id	Stu_name	Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major	GPA	Grade
	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00	C
	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
	Madhuri	Mechanical Engineering	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	7.00	B
	Anish	Electrical Engineering	anish@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50	B
	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00	B
	Sujith	Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50	B
	Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	8.50	A
	Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	9.00	A
	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50	A
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Query for Male Students:

.Implement a query to retrieve information about all male students from the "student_table."

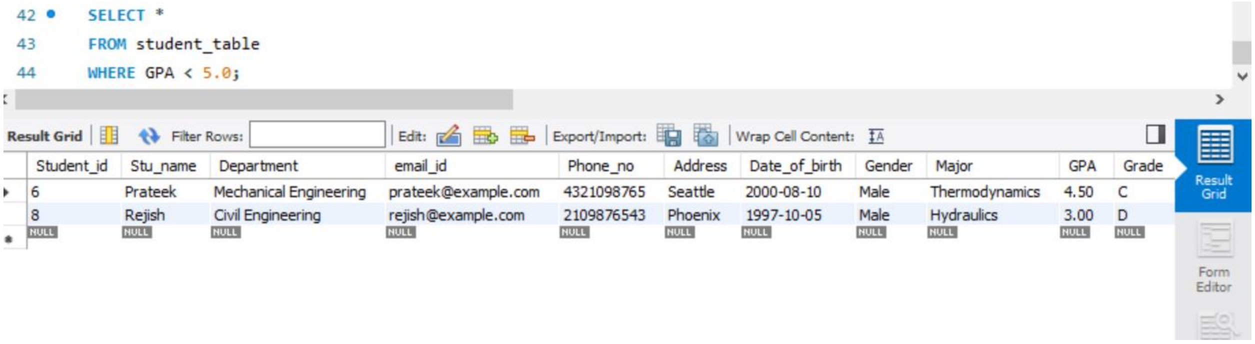


```
-- Query for Male Students
SELECT *
FROM student_table
WHERE Gender = 'Male';
```

Student_id	Stu_name	Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major	GPA	Grade
3	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00	C
5	Anish	Electrical Engineering	anish@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50	B
6	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
7	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50	A
8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
9	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00	B
10	Sujith	Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50	B
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

5. Query for Students with GPA less than 5.0

Create a query to fetch the details of students who have a GPA less than 5.0 from the "student_table."

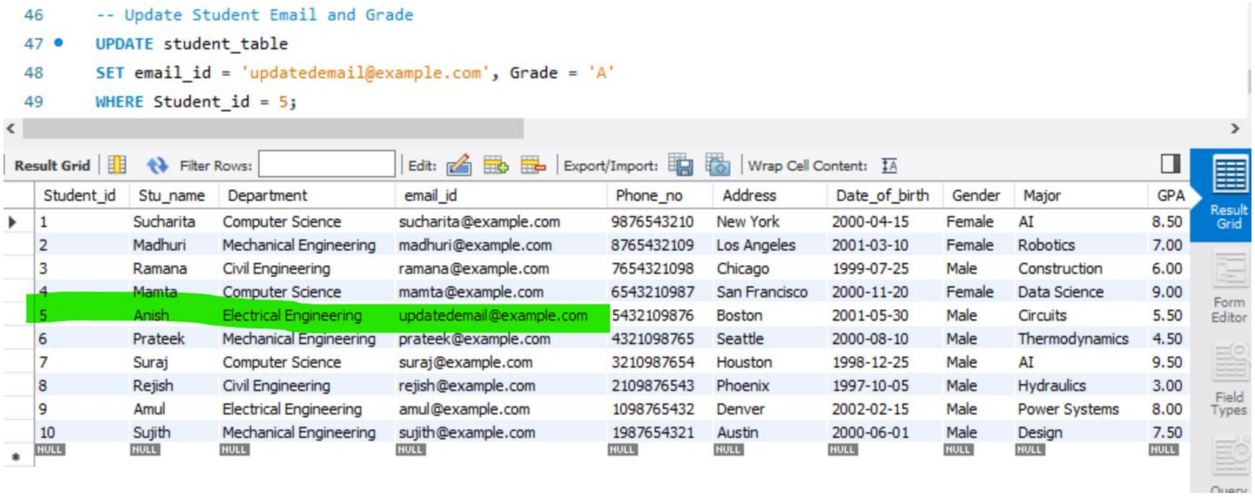


```
SELECT *
FROM student_table
WHERE GPA < 5.0;
```

Student_id	Stu_name	Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major	GPA	Grade
6	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. Update Student Email and Grade

Write an update statement to modify the email and grade of a student with a specific ID in the "student_table."

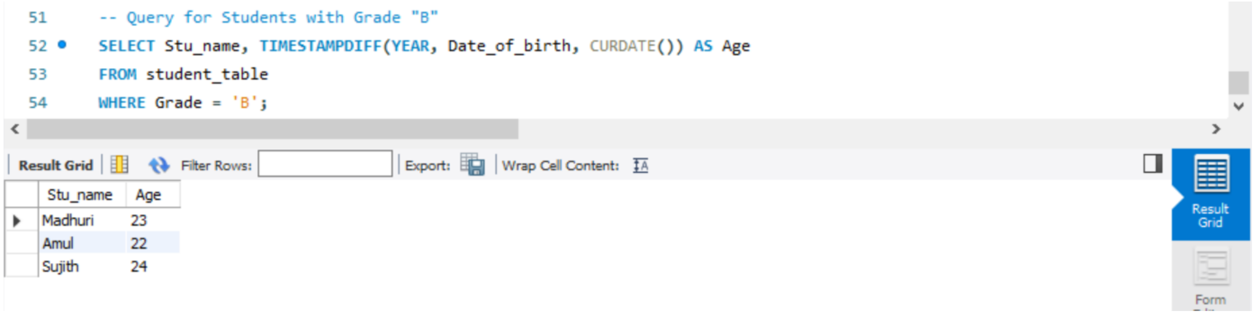


```sql
-- Update Student Email and Grade
UPDATE student_table
SET email_id = 'updatedemail@example.com', Grade = 'A'
WHERE Student_id = 5;
```

Student_id	Stu_name	Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major	GPA
1	Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	8.50
2	Madhuri	Mechanical Engineering	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	7.00
3	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00
4	Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	9.00
5	Anish	Electrical Engineering	updatedemail@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50
6	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50
7	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50
8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00
9	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00
10	Sujith	Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7. Query for Students with Grade "B"

Develop a query to retrieve the names and ages of all students who have a grade of "B" from the "student_table."

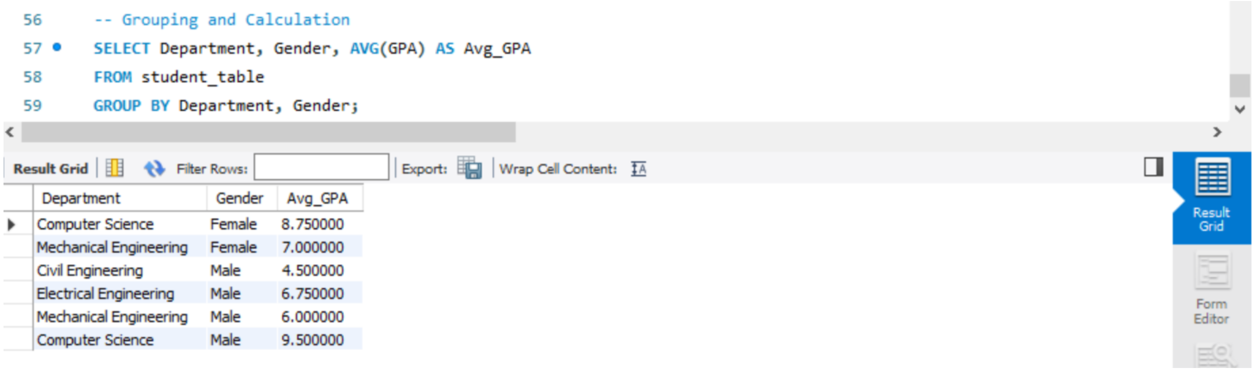


```sql
-- Query for Students with Grade "B"
SELECT Stu_name, TIMESTAMPDIFF(YEAR, Date_of_birth, CURDATE()) AS Age
FROM student_table
WHERE Grade = 'B';
```

Stu_name	Age
Madhuri	23
Amul	22
Sujith	24

8. Grouping and Calculation

Create a query to group the "student_table" by the "Department" and "Gender" columns and calculate the average GPA for each combination.

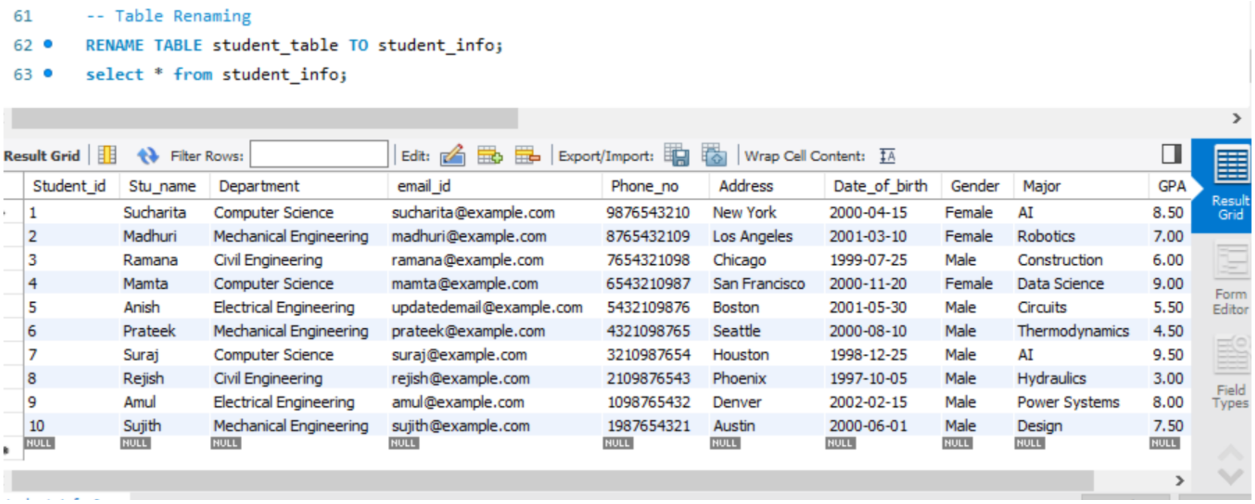


```sql
-- Grouping and Calculation
SELECT Department, Gender, AVG(GPA) AS Avg_GPA
FROM student_table
GROUP BY Department, Gender;
```

Department	Gender	Avg_GPA
Computer Science	Female	8.750000
Mechanical Engineering	Female	7.000000
Civil Engineering	Male	4.500000
Electrical Engineering	Male	6.750000
Mechanical Engineering	Male	6.000000
Computer Science	Male	9.500000

9. Table Renaming

Rename the "student_table" to "student_info" using the appropriate SQL statement.

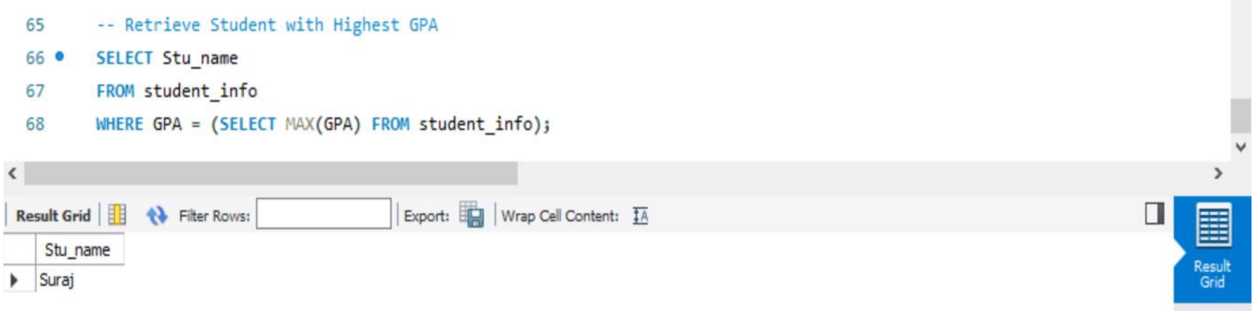


```
-- Table Renaming
RENAME TABLE student_table TO student_info;
select * from student_info;
```

Student_id	Stu_name	Department	email_id	Phone_no	Address	Date_of_birth	Gender	Major	GPA
1	Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	8.50
2	Madhuri	Mechanical Engineering	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	7.00
3	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00
4	Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	9.00
5	Anish	Electrical Engineering	updatedemail@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50
6	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50
7	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50
8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00
9	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00
10	Sujith	Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

10. Retrieve Student with Highest GPA

Write a query to retrieve the name of the student with the highest GPA from the "student_info" table.



```
-- Retrieve Student with Highest GPA
SELECT Stu_name
FROM student_info
WHERE GPA = (SELECT MAX(GPA) FROM student_info);
```

Stu_name
Suraj

Query Explanation

1. Database Setup

a. Create the Database:

```
CREATE DATABASE student_database;

USE student_database;
```

b. Create the Table:

```
CREATE TABLE student_table (

    Student_id INT AUTO_INCREMENT PRIMARY KEY,

    Stu_name TEXT NOT NULL,

    Department TEXT,

    email_id VARCHAR(255) UNIQUE,

    Phone_no BIGINT,

    Address TEXT,

    Date_of_birth DATE,

    Gender ENUM('Male', 'Female'),

    Major TEXT,

    GPA DECIMAL(3, 2),

    Grade ENUM('A', 'B', 'C', 'D', 'F')

);

select * from student_table;
```

Explanation:-

- AUTO_INCREMENT ensures unique Student_id values.
- ENUM restricts Gender and Grade to specific values.

2. Data Entry

a. Insert Sample Records:

```
INSERT INTO student_table (Stu_name, Department, email_id, Phone_no, Address, Date_of_birth, Gender, Major, GPA, Grade)

VALUES

('Sucharita', 'Computer Science', 'sucharita@example.com', 9876543210, 'New York', '2000-04-15', 'Female', 'AI', 8.5, 'A'),

('Madhuri', 'Mechanical Engineering', 'madhuri@example.com', 8765432109, 'Los Angeles', '2001-03-10', 'Female', 'Robotics', 7.0, 'B'),

('Ramana', 'Civil Engineering', 'ramana@example.com', 7654321098, 'Chicago', '1999-07-25', 'Male', 'Construction', 6.0, 'C'),

('Mamta', 'Computer Science', 'mamta@example.com', 6543210987, 'San Francisco', '2000-11-20', 'Female', 'Data Science', 9.0, 'A'),

('Anish', 'Electrical Engineering', 'anish@example.com', 5432109876, 'Boston', '2001-05-30', 'Male', 'Circuits', 5.5, 'B'),

('Prateek', 'Mechanical Engineering', 'prateek@example.com', 4321098765, 'Seattle', '2000-08-10', 'Male', 'Thermodynamics', 4.5, 'C'),

('Suraj', 'Computer Science', 'suraj@example.com', 3210987654, 'Houston', '1998-12-25', 'Male', 'AI', 9.5, 'A'),

('Rejish', 'Civil Engineering', 'rejish@example.com', 2109876543, 'Phoenix', '1997-10-05', 'Male', 'Hydraulics', 3.0, 'D'),

('Amul', 'Electrical Engineering', 'amul@example.com', 1098765432, 'Denver', '2002-02-15', 'Male', 'Power Systems', 8.0, 'B'),

('Sujith', 'Mechanical Engineering', 'sujith@example.com', 1987654321, 'Austin', '2000-06-01', 'Male', 'Design', 7.5, 'B');
```

Explanation:-

1. INSERT INTO student_table: Specifies the table where the new data will be inserted.
2. (Stu_name, Department, email_id, Phone_no, Address, Date_of_birth, Gender, Major, GPA, Grade): Lists the columns where data will be inserted.
3. VALUES: Introduces the data values to be inserted.

3. Student Information Retrieval

```
--To Retrieve all student records sorted in descending order by grade

SELECT * FROM student_table

ORDER BY Grade DESC;
```

Explanation:-

This retrieves all columns and sorts rows by Grade in descending order.

4. Query for Male Students

```
--To Retrieve information about male students:

SELECT * FROM student_table

WHERE Gender = 'Male';
```

Explanation:-

Filters rows where Gender is Male.

5. Query for Students with GPA less than 5.0

```
--To Fetch details of students with GPA < 5.0

SELECT * FROM student_table

WHERE GPA < 5.0;
```

Explanation:-

Filters rows where the GPA is less than 5.0.

6. Update Student Email and Grade

```
--To Update email and grade of a specific student:

UPDATE student_table

SET email_id = 'updatedemail@example.com', Grade = 'A'

WHERE Student_id = 5; -- Replace with the actual Student_id
```

Explanation:-

Updates email_id and Grade for the student with Student_id = 5.

7. Query for Students with Grade "B"

```
-- Retrieve names and ages of students with grade "B"
SELECT Stu_name, TIMESTAMPDIFF(YEAR, Date_of_birth, CURDATE()) AS Age
FROM student_table
WHERE Grade = 'B';
```

Explanation:-

Retrieves Stu_name and calculates age using TIMESTAMPDIFF for students with Grade = 'B'.

8. Grouping and Calculation

```
-- Group by Department and Gender, and calculate average GPA
SELECT Department, Gender, AVG(GPA) AS Avg_GPA
FROM student_table
GROUP BY Department, Gender;
```

Explanation:-

- Groups rows by unique combinations of Department and Gender.
- Calculates the average GPA (AVG(GPA)) for each group.

9. Table Renaming

```
-- Rename the table student_table to student_info
RENAME TABLE student_table TO student_info;
```

Explanation:-

Renames student_table to student_info.

10. Retrieve Student with Highest GPA

```
-- Retrieve the name of the student with the highest GPA

SELECT Stu_name

FROM student_info

WHERE GPA = (SELECT MAX(GPA) FROM student_info);
```

Explanation:-

Finds the student with the highest GPA using a subquery:

SELECT MAX(GPA) retrieves the highest GPA.

The outer query matches this GPA to the student.