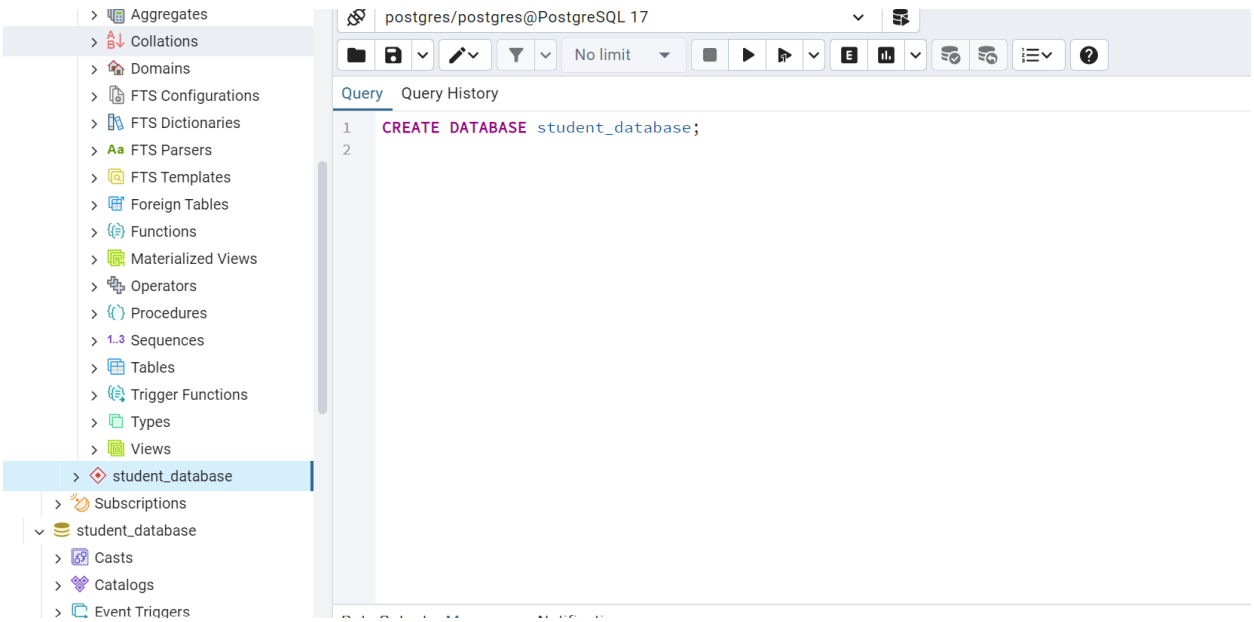
OUTPUT SHEET

Project: Student Database Management System(PostgreSQL)

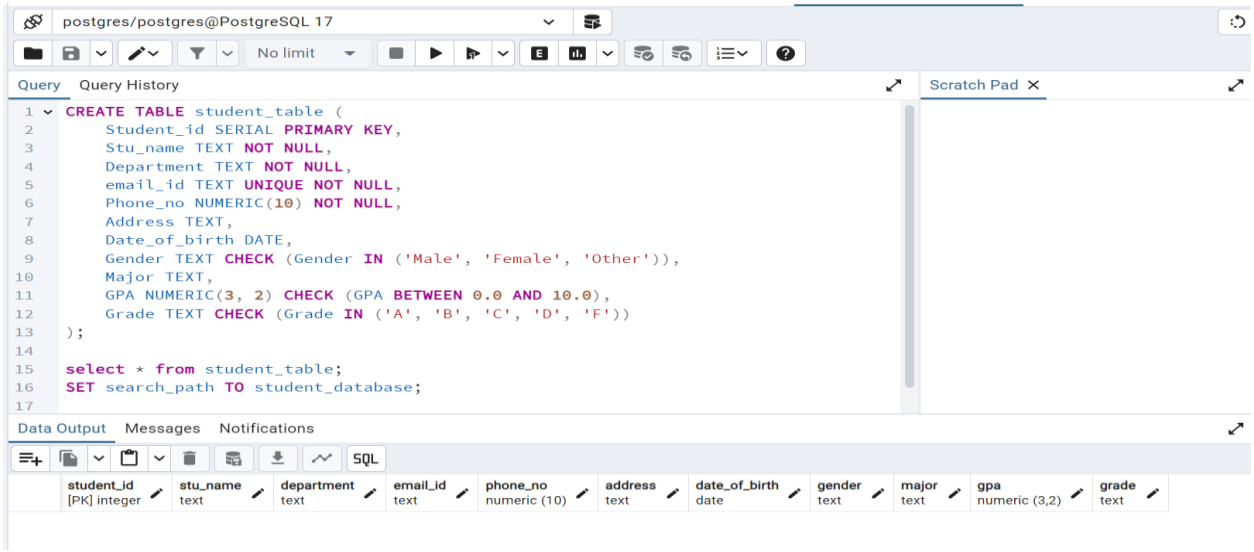
Objective: Design and implement a student database management system using PostgreSQL that allows storing and retrieving student information efficiently. The project will include the following tasks:

1. Database Setup

Create a database named "student_database."

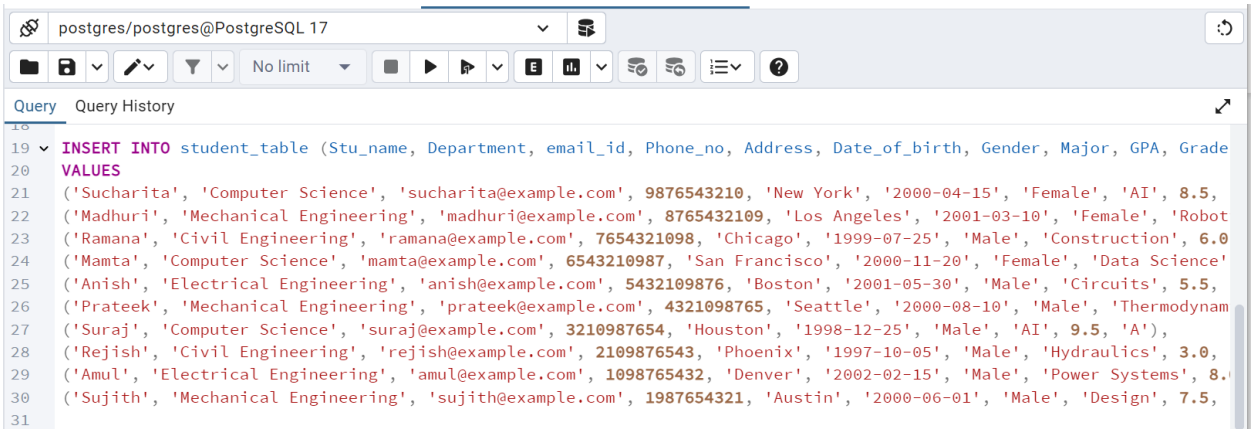


Create a table called " student_table " with the following columns: Student_id (integer), Stu_name (text), Department (text), email_id (text),Phone_no (numeric), Address (text), Date_of_birth (date), Gender (text), Major (text), GPA (numeric),Grade (text) should be A,B,C etc.

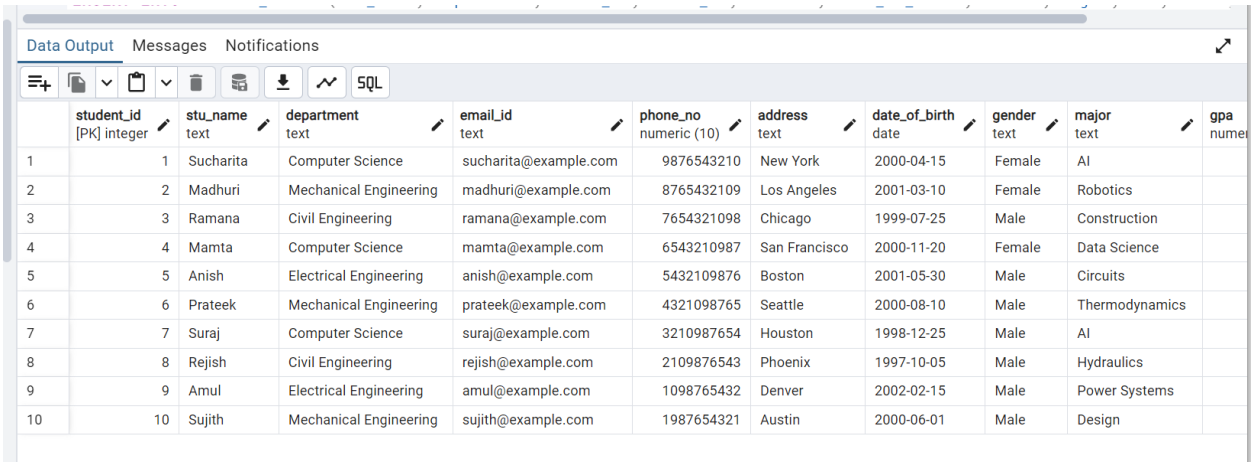


2. Data Entry

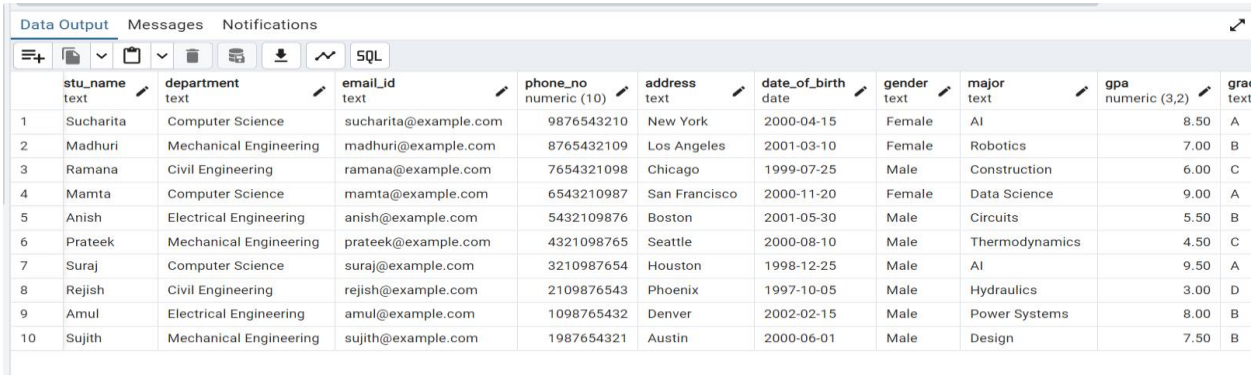
Insert 10 sample records into the "student_table" using INSERT command.



```
INSERT INTO student_table (Stu_name, Department, email_id, Phone_no, Address, Date_of_birth, Gender, Major, GPA, Grade
VALUES
('Sucharita', 'Computer Science', 'sucharita@example.com', 9876543210, 'New York', '2000-04-15', 'Female', 'AI', 8.5,
('Madhuri', 'Mechanical Engineering', 'madhuri@example.com', 8765432109, 'Los Angeles', '2001-03-10', 'Female', 'Robot
('Ramana', 'Civil Engineering', 'ramana@example.com', 7654321098, 'Chicago', '1999-07-25', 'Male', 'Construction', 6.0
('Mamta', 'Computer Science', 'mamta@example.com', 6543210987, 'San Francisco', '2000-11-20', 'Female', 'Data Science'
('Anish', 'Electrical Engineering', 'anish@example.com', 5432109876, 'Boston', '2001-05-30', 'Male', 'Circuits', 5.5,
('Prateek', 'Mechanical Engineering', 'prateek@example.com', 4321098765, 'Seattle', '2000-08-10', 'Male', 'Thermodynam
('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,
('Amul', 'Electrical Engineering', 'amul@example.com', 1098765432, 'Denver', '2002-02-15', 'Male', 'Power Systems', 8.
('Sujith', 'Mechanical Engineering', 'sujith@example.com', 1987654321, 'Austin', '2000-06-01', 'Male', 'Design', 7.5,
```



student_id [PK] integer	stu_name text	department text	email_id text	phone_no numeric (10)	address text	date_of_birth date	gender text	major text	gpa
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	Construction	
4	Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	
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	Thermodynamics	
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 Systems	
10	Sujith	Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	



stu_name text	department text	email_id text	phone_no numeric (10)	address text	date_of_birth date	gender text	major text	gpa numeric (3,2)	gra text
Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	8.50	A
Madhuri	Mechanical Engineering	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	7.00	B
Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00	C
Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	9.00	A
Anish	Electrical Engineering	anish@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50	B
Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50	A
Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
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

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.

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

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

	stu_name text	department text	email_id text	phone_no numeric (10)	address text	date_of_birth date	gender text	major text	gpa numeric (3,2)	grade text
1	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
2	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00	C
3	Prateek	Mechanical Engineer...	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
4	Sujith	Mechanical Engineer...	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50	B
5	Madhuri	Mechanical Engineer...	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	7.00	B
6	Anish	Electrical Engineering	anish@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50	B
7	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00	B
8	Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	9.00	A
9	Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	8.50	A
10	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50	A

4. Query for Male Students:

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

```
-- Retrieve information about male students:

SELECT * FROM student_table
WHERE Gender = 'Male';
```

	stu_name text	department text	email_id text	phone_no numeric (10)	address text	date_of_birth date	gender text	major text	gpa numeric (3,2)	grade text
1	3	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00	C
2	5	Anish	Electrical Engineering	anish@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50	B
3	6	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
4	7	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50	A
5	8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
6	9	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00	B
7	10	Sujith	Mechanical Engineering	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50	B

5. Query for Students with GPA less than 5.0

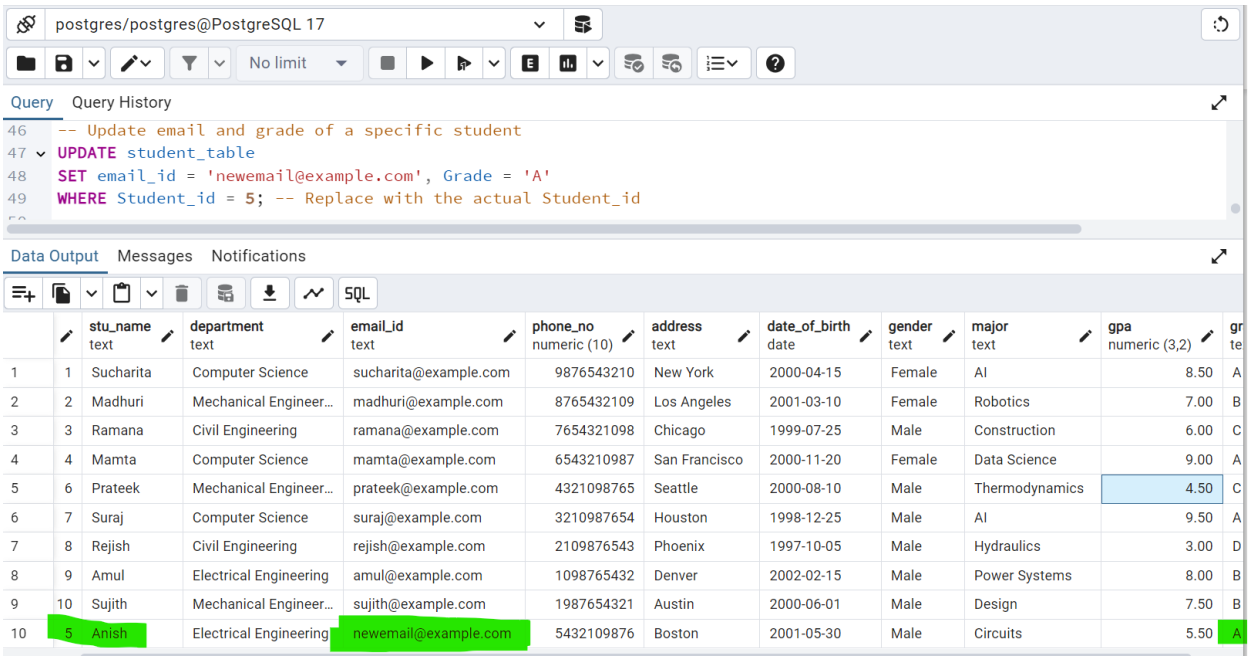
Create a query to fetch the details of students who have a GPA less than 5.0 from the "student_table."

```
-- Fetch details of students with GPA < 5.0
SELECT * FROM student_table
WHERE GPA < 5.0;
```

	stu_name text	department text	email_id text	phone_no numeric (10)	address text	date_of_birth date	gender text	major text	gpa numeric (3,2)	grade text
1	6	Prateek	Mechanical Engineering	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
2	8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D

6. Update Student Email and Grade

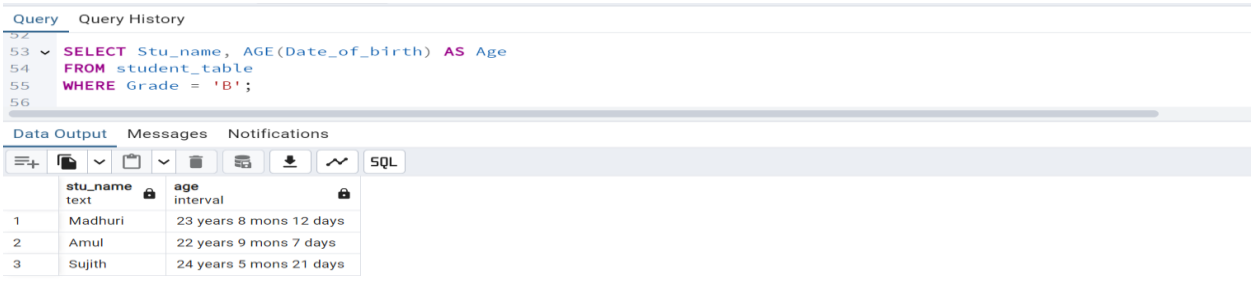
Write an update statement to modify the email and grade of a student with a specific ID in the "student_table."



	stu_name text	department text	email_id text	phone_no numeric (10)	address text	date_of_birth date	gender text	major text	gpa numeric (3,2)	gr te
1	1 Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	8.50	A
2	2 Madhuri	Mechanical Engineer...	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	7.00	B
3	3 Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	6.00	C
4	4 Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	9.00	A
5	6 Prateek	Mechanical Engineer...	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	4.50	C
6	7 Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	9.50	A
7	8 Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	3.00	D
8	9 Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	8.00	B
9	10 Sujith	Mechanical Engineer...	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	7.50	B
10	5 Anish	Electrical Engineering	newemail@example.com	5432109876	Boston	2001-05-30	Male	Circuits	5.50	A

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."



	stu_name text	age interval
1	Madhuri	23 years 8 mons 12 days
2	Amul	22 years 9 mons 7 days
3	Sujith	24 years 5 mons 21 days

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.

Query | Query History

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

Data Output | Messages | Notifications

	department text	gender text	avg_gpa numeric
1	Computer Science	Female	8.7500000000000000
2	Mechanical Engineering	Male	6.0000000000000000
3	Electrical Engineering	Male	6.7500000000000000
4	Civil Engineering	Male	4.5000000000000000
5	Mechanical Engineering	Female	7.0000000000000000
6	Computer Science	Male	9.5000000000000000

9. Table Renaming

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

Query | Query History

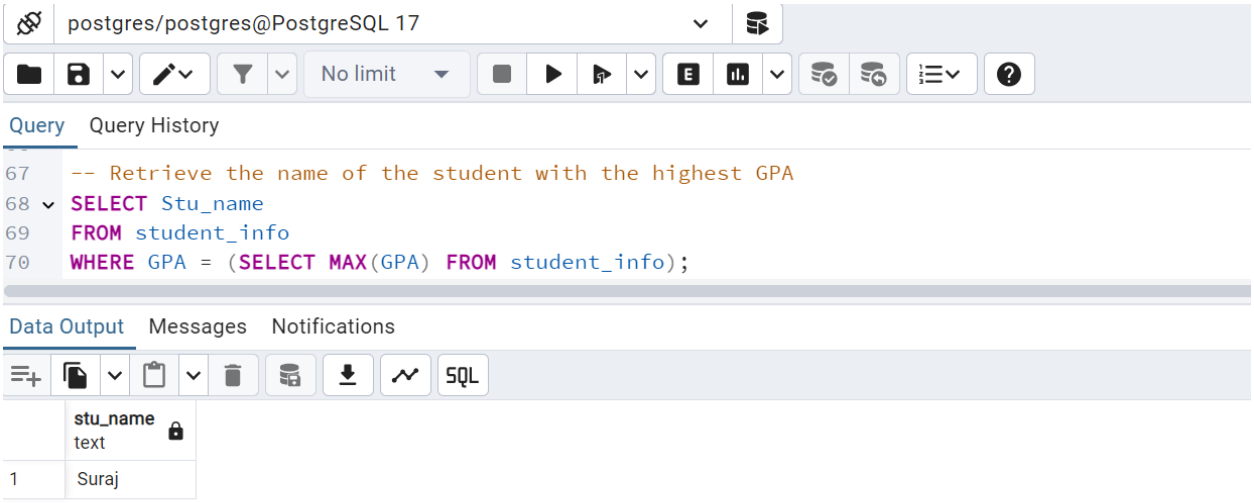
```
62  -- Rename the table student_table to student_info
63  ALTER TABLE student_table RENAME TO student_info;
64
65  select * from student_info;
```

Data Output | Messages | Notifications

	student_id [PK] integer	stu_name text	department text	email_id text	phone_no numeric (10)	address text	date_of_birth date	gender text	major text	gpa numeric
1	1	Sucharita	Computer Science	sucharita@example.com	9876543210	New York	2000-04-15	Female	AI	
2	2	Madhuri	Mechanical Engineer...	madhuri@example.com	8765432109	Los Angeles	2001-03-10	Female	Robotics	
3	3	Ramana	Civil Engineering	ramana@example.com	7654321098	Chicago	1999-07-25	Male	Construction	
4	4	Mamta	Computer Science	mamta@example.com	6543210987	San Francisco	2000-11-20	Female	Data Science	
5	6	Prateek	Mechanical Engineer...	prateek@example.com	4321098765	Seattle	2000-08-10	Male	Thermodynamics	
6	7	Suraj	Computer Science	suraj@example.com	3210987654	Houston	1998-12-25	Male	AI	
7	8	Rejish	Civil Engineering	rejish@example.com	2109876543	Phoenix	1997-10-05	Male	Hydraulics	
8	9	Amul	Electrical Engineering	amul@example.com	1098765432	Denver	2002-02-15	Male	Power Systems	
9	10	Sujith	Mechanical Engineer...	sujith@example.com	1987654321	Austin	2000-06-01	Male	Design	
10	5	Anish	Electrical Engineering	newemail@example.com	5432109876	Boston	2001-05-30	Male	Circuits	

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.



Query Explanation

1. Database Setup

a. Create the Database:

CREATE DATABASE student_database;

b. Create the Table:

```
CREATE TABLE student_table (
    Student_id SERIAL PRIMARY KEY,
    Stu_name TEXT NOT NULL,
    Department TEXT,
    email_id TEXT UNIQUE,
    Phone_no NUMERIC(10),
    Address TEXT,
    Date_of_birth DATE,
```

```
    Gender TEXT CHECK (Gender IN ('Male', 'Female')),
    Major TEXT,
    GPA NUMERIC(3, 2),
    Grade TEXT CHECK (Grade IN ('A', 'B', 'C', 'D', 'F'))
);
select * from student_table;
```

Explanation:-

1. CREATE TABLE student_table: This creates a new table called student_table in the database.
2. Student_id SERIAL PRIMARY KEY:

    - SERIAL: Automatically generates a unique integer value for each row.
    - PRIMARY KEY: Ensures this column uniquely identifies each row and prevents duplicate values.

3. Stu_name TEXT NOT NULL:

    - TEXT: Defines the column as text.
    - NOT NULL: Ensures this column cannot have empty or NULL values.

4. Department TEXT: Defines the Department column as text.
5. email_id TEXT UNIQUE:

    - TEXT: Defines the column as text.
    - UNIQUE: Ensures that no two rows can have the same email address.

6. Phone_no NUMERIC(10):

    - NUMERIC(10): Specifies a numeric column with up to 10 digits.

7. Address TEXT: Defines the Address column as text.
8. Date_of_birth DATE: Defines the Date_of_birth column to store dates.
9. Gender TEXT CHECK (Gender IN ('Male', 'Female')):

    - Restricts the Gender column to only accept Male or Female.

10. Major TEXT: Defines the Major column as text.
11. GPA NUMERIC(3, 2):

    - NUMERIC(3, 2): Specifies a numeric column with 3 digits in total, 2 of which are after the decimal point.

12. Grade TEXT CHECK (Grade IN ('A', 'B', 'C', 'D', 'F')):

- Restricts the Grade column to only accept A, B, C, D, or F.

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:-

1. SELECT *: Retrieves all columns from the student_table.
2. FROM student_table: Specifies the table from which to retrieve the data.
3. ORDER BY Grade DESC: Sorts the data by the Grade column in descending order (highest grade first).

4. Query for Male Students

--To Retrieve information about male students:

```
SELECT * FROM student_table

WHERE Gender = 'Male';
```

Explanation:-

1. SELECT *: Retrieves all columns from the student_table.
2. FROM student_table: Specifies the table from which to retrieve the data.
3. WHERE Gender = 'Male': Filters the rows where the Gender column is equal to 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:-

1. SELECT *: Retrieves all columns from the student_table.
2. FROM student_table: Specifies the table from which to retrieve the data.
3. WHERE GPA < 5.0: Filters the rows where the GPA column 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 = 'newemail@example.com', Grade = 'A'

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

Explanation:-

1. UPDATE student_table: Specifies the table to be updated.
2. SET email_id = 'newemail@example.com', Grade = 'A': Updates the email_id and Grade columns with the specified values.
3. WHERE Student_id = 5: Applies the update only to the row where Student_id equals 5.

7. Query for Students with Grade "B"

-- Retrieve names and ages of students with grade "B"

```
SELECT Stu_name, AGE(Date_of_birth) AS Age

FROM student_table

WHERE Grade = 'B';
```

Explanation:-

1. SELECT Stu_name, AGE(Date_of_birth) AS Age:
    - Retrieves the Stu_name and calculates the age from the Date_of_birth.
    - The AS Age renames the calculated column to Age.
2. FROM student_table: Specifies the table from which to retrieve the data.
3. WHERE Grade = 'B': Filters the rows where the Grade column equals 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:-

1. SELECT Department, Gender, AVG(GPA) AS Avg_GPA:

- Retrieves the Department, Gender, and calculates the average GPA (AVG(GPA)).
- Renames the calculated column as Avg_GPA.

2. FROM student_table: Specifies the table from which to retrieve the data.
3. GROUP BY Department, Gender: Groups the rows by unique combinations of Department and Gender.

9. Table Renaming

-- Rename the table student_table to student_info

ALTER TABLE student_table RENAME TO student_info;

Explanation:-

1. ALTER TABLE student_table: Modifies the student_table.
2. RENAME TO student_info: Renames the 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:-

1. SELECT Stu_name: Retrieves the Stu_name of the student with the highest GPA.
2. FROM student_info: Specifies the table from which to retrieve the data.
3. WHERE GPA = (SELECT MAX(GPA) FROM student_info):
    - Uses a subquery (SELECT MAX(GPA) FROM student_info) to find the highest GPA.
    - Filters the rows where GPA equals this maximum value.