

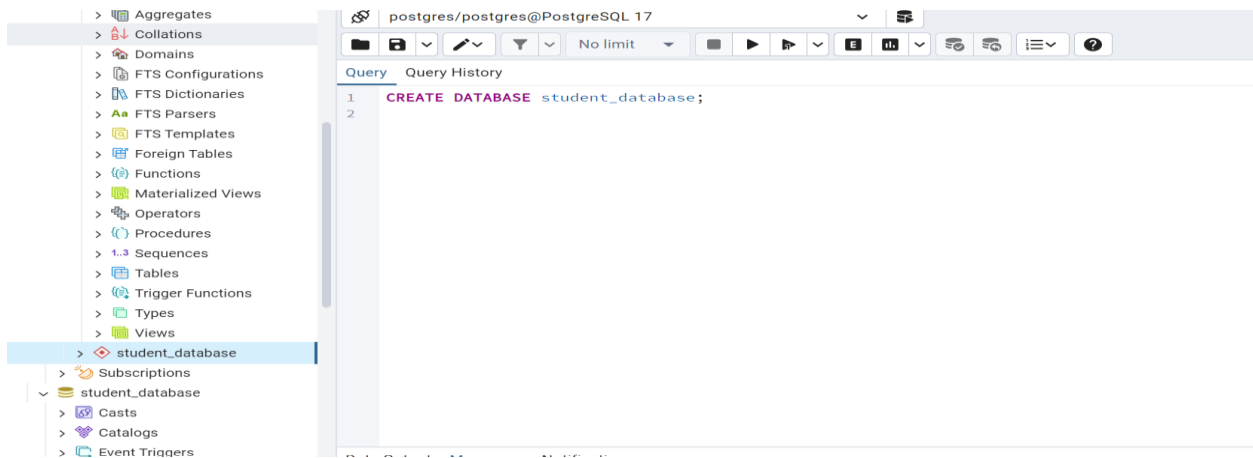
# QUERY SHEET

## Project: Student Database Management System(PostgreSQL)

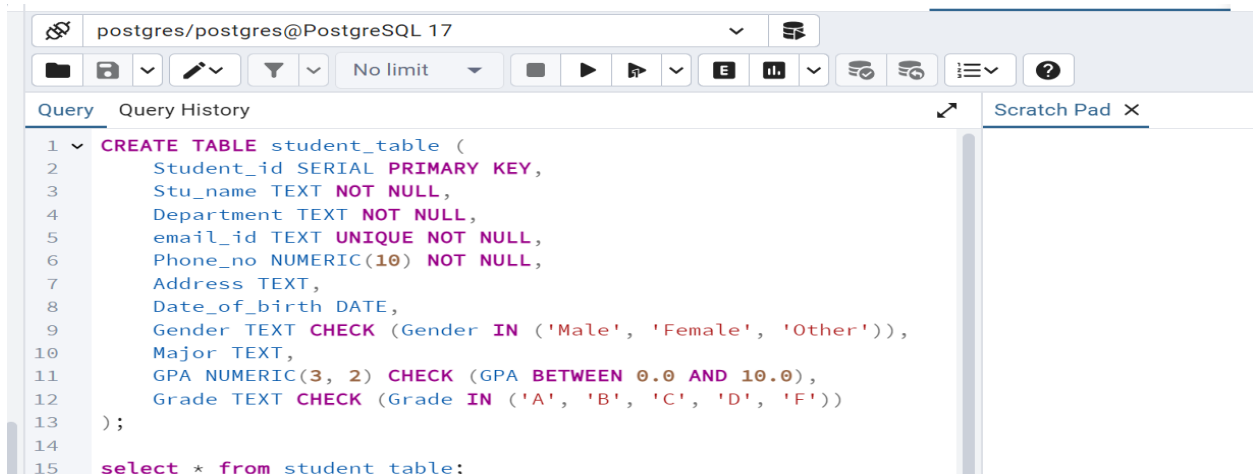
Objective: Design and implement a student database management system using PostgreSQL that allows storing and retrieving student information efficiently. The project will include the following tasks:

### 1. Database Setup

a. Create a database named "student\_database."



b. 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.

```
postgres/postgres@PostgreSQL 17
Query History
18
19 INSERT INTO student_table (Stu_name, Department, email_id, Phone_no, Address, Date_of_birth, Gender, Major, GPA, Grade
20 VALUES
21 ('Sucharita', 'Computer Science', 'sucharita@example.com', 9876543210, 'New York', '2000-04-15', 'Female', 'AI', 8.5,
22 ('Madhuri', 'Mechanical Engineering', 'madhuri@example.com', 8765432109, 'Los Angeles', '2001-03-10', 'Female', 'Robot
23 ('Ramana', 'Civil Engineering', 'ramana@example.com', 7654321098, 'Chicago', '1999-07-25', 'Male', 'Construction', 6.0
24 ('Mamta', 'Computer Science', 'mamta@example.com', 6543210987, 'San Francisco', '2000-11-20', 'Female', 'Data Science'
25 ('Anish', 'Electrical Engineering', 'anish@example.com', 5432109876, 'Boston', '2001-05-30', 'Male', 'Circuits', 5.5,
26 ('Prateek', 'Mechanical Engineering', 'prateek@example.com', 4321098765, 'Seattle', '2000-08-10', 'Male', 'Thermodynam
27 ('Suraj', 'Computer Science', 'suraj@example.com', 3210987654, 'Houston', '1998-12-25', 'Male', 'AI', 9.5, 'A'),
28 ('Rejish', 'Civil Engineering', 'rejish@example.com', 2109876543, 'Phoenix', '1997-10-05', 'Male', 'Hydraulics', 3.0,
29 ('Amul', 'Electrical Engineering', 'amul@example.com', 1098765432, 'Denver', '2002-02-15', 'Male', 'Power Systems', 8.0
30 ('Sujith', 'Mechanical Engineering', 'sujith@example.com', 1987654321, 'Austin', '2000-06-01', 'Male', 'Design', 7.5,
31
```

## 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.

```
31 SELECT *
32 FROM student_table
33 ORDER BY Grade DESC;
```

## 4. Query for Male Students:

Implement a query to retrieve information about all male students from the "student\_table."

```
34
35 SELECT *
36 FROM student_table
37 WHERE Gender = 'Male';
```

## 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."

```
42 -- Fetch details of students with GPA < 5.0
43 SELECT * FROM student_table
44 WHERE GPA < 5.0;
45
```

## 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."

```
postgres/postgres@PostgreSQL 17
Query Query History
46 -- Update email and grade of a specific student
47 UPDATE student_table
48 SET email_id = 'newemail@example.com', Grade = 'A'
49 WHERE Student_id = 5; -- Replace with the actual Student_id
```

## 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."

```
Query Query History
52
53 SELECT Stu_name, AGE(Date_of_birth) AS Age
54 FROM student_table
55 WHERE Grade = 'B';
56
```

## 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;
```

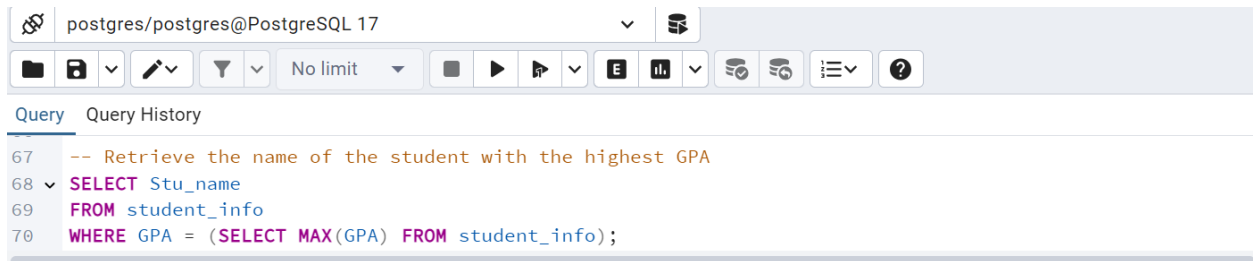
## 9. Table Renaming

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;
```

## 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.



The screenshot shows a PostgreSQL query editor interface. At the top, the connection is set to 'postgres/postgres@PostgreSQL 17'. Below the connection bar is a toolbar with various icons for file operations, query execution, and settings. The main area displays a SQL query with line numbers 67 through 70. The query is a SELECT statement that retrieves the student name with the highest GPA from the 'student\_info' table.

```
67 -- Retrieve the name of the student with the highest GPA
68 SELECT Stu_name
69 FROM student_info
70 WHERE GPA = (SELECT MAX(GPA) FROM student_info);
```

## FULL QUERY

### 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;
```

## 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');
```

## 3. Student Information Retrieval

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

```
SELECT * FROM student_table
```

```
ORDER BY Grade DESC;
```

#### **4. Query for Male Students**

--To Retrieve information about male students:

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

#### **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
```

#### **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';
```

#### **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;
```

#### **9. Table Renaming**

-- Rename the table student\_table to student\_info

```
ALTER TABLE student_table RENAME 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);