Student Database Management System (PostgreSQL)

Objective: Design and implement a student database management system using PostgreSQL that allows storing and retrieving student information efficiently.

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.

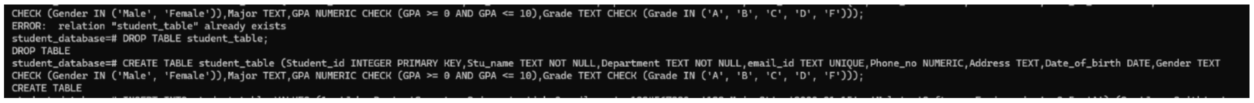
Query:

CREATE DATABASE student_database;

\c student_database

CREATE TABLE student_table (Student_id INTEGER PRIMARY KEY,   Stu_name TEXT NOT NULL, Department TEXT NOT NULL, email_id TEXT UNIQUE,Phone_no NUMERIC, Address TEXT, Date_of_birth DATE,Gender TEXT CHECK (Gender IN ('Male', 'Female', 'Other')),Major TEXT, GPA NUMERIC CHECK (GPA >= 0 AND GPA <= 10),Grade TEXT CHECK (Grade IN ('A', 'B', 'C', 'D', 'F')));

Output:



```
CHECK (Gender IN ('Male', 'Female')),Major TEXT,GPA NUMERIC CHECK (GPA >= 0 AND GPA <= 10),Grade TEXT CHECK (Grade IN ('A', 'B', 'C', 'D', 'F')));
ERROR:  relation "student_table" already exists
student_database=# DROP TABLE student_table;
DROP TABLE
student_database=# CREATE TABLE student_table (Student_id INTEGER PRIMARY KEY,Stu_name TEXT NOT NULL,Department TEXT NOT NULL,email_id TEXT UNIQUE,Phone_no NUMERIC,Address TEXT,Date_of_birth DATE,Gender TEXT
CHECK (Gender IN ('Male', 'Female')),Major TEXT,GPA NUMERIC CHECK (GPA >= 0 AND GPA <= 10),Grade TEXT CHECK (Grade IN ('A', 'B', 'C', 'D', 'F')));
CREATE TABLE
```

2.Data Entry

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

Query:

INSERT INTO student_table VALUES (1, 'John Doe', 'Computer Science', 'john@email.com', 1234567890, '123 Main St', '2000-01-15', 'Male', 'Software Engineering', 8.5, 'A'), (2, 'Jane Smith', 'Electronics', 'jane@email.com', 2345678901, '456 Oak Ave', '2001-03-20', 'Female', 'Electronics', 7.8, 'B'), (3, 'Mike Johnson', 'Mechanical', 'mike@email.com', 3456789012, '789 Pine Rd', '2000-07-10', 'Male', 'Robotics', 4.5, 'C'), (4, 'Sarah Williams', 'Computer Science', 'sarah@email.com', 4567890123, '321 Elm St', '2001-11-05', 'Female', 'AI', 9.0, 'A'), (5, 'Tom Brown', 'Electronics', 'tom@email.com', 5678901234, '654 Maple Dr', '2000-09-25', 'Male', 'Communications', 4.8, 'C'), (6, 'Emily Davis', 'Mechanical', 'emily@email.com', 6789012345, '987 Cedar Ln', '2001-05-30', 'Female', 'Manufacturing', 8.2, 'B'), (7, 'David Wilson', 'Computer Science', 'david@email.com', 7890123456, '147 Birch Rd', '2000-12-12', 'Male', 'Cybersecurity', 7.5, 'B'), (8, 'Lisa Anderson', 'Electronics', 'lisa@email.com', 8901234567, '258 Pine St', '2001-08-18', 'Female', 'VLSI', 6.9, 'B'), (9, 'James Taylor', 'Mechanical', 'james@email.com', 9012345678, '369 Oak Rd', '2000-04-22', 'Male', 'Automotive', 3.8, 'D'), (10, 'Amy Martin', 'Computer Science', 'amy@email.com', 0123456789, '741 Maple Ave', '2001-02-28', 'Female', 'Data Science', 9.5, 'A');

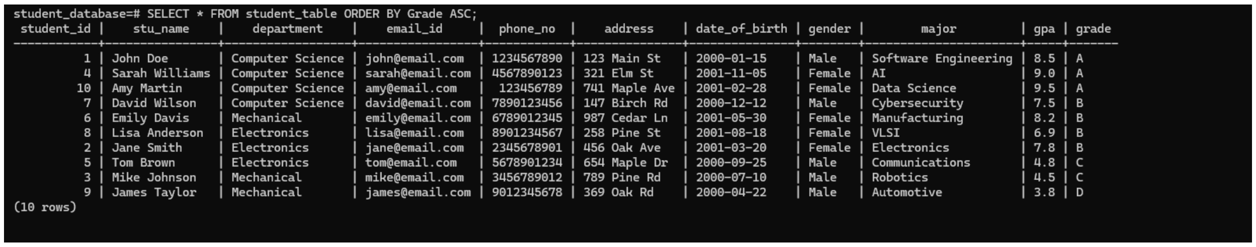
3.Student Information Retrieval

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

Query:

SELECT * FROM student_table ORDER BY Grade ASC;

Output:



```
student_database=# SELECT * FROM student_table ORDER BY Grade ASC;
 student_id |    stu_name    |    department    |     email_id     |  phone_no  |    address    | date_of_birth | gender |        major         | gpa | grade
------------+----------------+------------------+------------------+------------+---------------+---------------+--------+----------------------+-----+-------
          1 | John Doe       | Computer Science | john@email.com   | 1234567890 | 123 Main St   | 2000-01-15    | Male   | Software Engineering | 8.5 | A
          4 | Sarah Williams | Computer Science | sarah@email.com  | 4567890123 | 321 Elm St    | 2001-11-05    | Female | AI                   | 9.0 | A
         10 | Amy Martin     | Computer Science | amy@email.com    |  123456789 | 741 Maple Ave | 2001-02-28    | Female | Data Science         | 9.5 | A
          7 | David Wilson   | Computer Science | david@email.com  | 7890123456 | 147 Birch Rd  | 2000-12-12    | Male   | Cybersecurity        | 7.5 | B
          6 | Emily Davis    | Mechanical       | emily@email.com  | 6789012345 | 987 Cedar Ln  | 2001-05-30    | Female | Manufacturing        | 8.2 | B
          8 | Lisa Anderson  | Electronics      | lisa@email.com   | 8901234567 | 258 Pine St   | 2001-08-18    | Female | VLSI                 | 6.9 | B
          2 | Jane Smith     | Electronics      | jane@email.com   | 2345678901 | 456 Oak Ave   | 2001-03-20    | Female | Electronics          | 7.8 | B
          5 | Tom Brown      | Electronics      | tom@email.com    | 5678901234 | 654 Maple Dr  | 2000-09-25    | Male   | Communications       | 4.8 | C
          3 | Mike Johnson   | Mechanical       | mike@email.com   | 3456789012 | 789 Pine Rd   | 2000-07-10    | Male   | Robotics             | 4.5 | C
          9 | James Taylor   | Mechanical       | james@email.com  | 9012345678 | 369 Oak Rd    | 2000-04-22    | Male   | Automotive           | 3.8 | D
(10 rows)
```

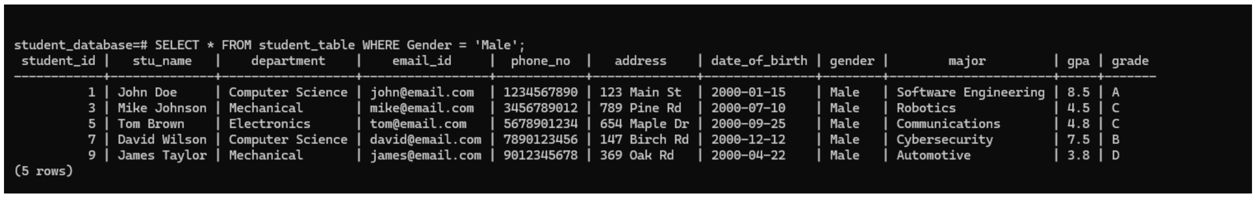
4.Query for Male Students:

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

Query:

SELECT * FROM student_table WHERE Gender = 'Male';

Output:



```
student_database=# SELECT * FROM student_table WHERE Gender = 'Male';
 student_id |   stu_name   |    department    |     email_id     |  phone_no  |    address    | date_of_birth | gender |        major         | gpa | grade
------------+--------------+------------------+------------------+------------+---------------+---------------+--------+----------------------+-----+-------
          1 | John Doe     | Computer Science | john@email.com   | 1234567890 | 123 Main St   | 2000-01-15    | Male   | Software Engineering | 8.5 | A
          3 | Mike Johnson | Mechanical       | mike@email.com   | 3456789012 | 789 Pine Rd   | 2000-07-10    | Male   | Robotics             | 4.5 | C
          5 | Tom Brown    | Electronics      | tom@email.com    | 5678901234 | 654 Maple Dr  | 2000-09-25    | Male   | Communications       | 4.8 | C
          7 | David Wilson | Computer Science | david@email.com  | 7890123456 | 147 Birch Rd  | 2000-12-12    | Male   | Cybersecurity        | 7.5 | B
          9 | James Taylor | Mechanical       | james@email.com  | 9012345678 | 369 Oak Rd    | 2000-04-22    | Male   | Automotive           | 3.8 | D
(5 rows)
```

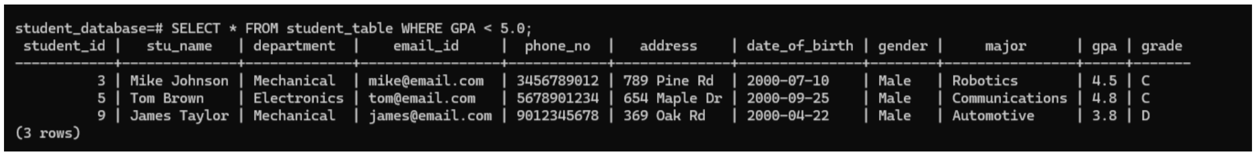
5.Query for Students with GPA less than 5.0

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

Query:

SELECT * FROM student_table WHERE GPA < 5.0;

Output:



```
student_database=# 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
------------+--------------+-------------+------------------+------------+--------------+---------------+--------+----------------+-----+-------
          3 | Mike Johnson | Mechanical  | mike@email.com   | 3456789012 | 789 Pine Rd  | 2000-07-10    | Male   | Robotics       | 4.5 | C
          5 | Tom Brown    | Electronics | tom@email.com    | 5678901234 | 654 Maple Dr | 2000-09-25    | Male   | Communications | 4.8 | C
          9 | James Taylor | Mechanical  | james@email.com  | 9012345678 | 369 Oak Rd   | 2000-04-22    | Male   | Automotive     | 3.8 | D
(3 rows)
```

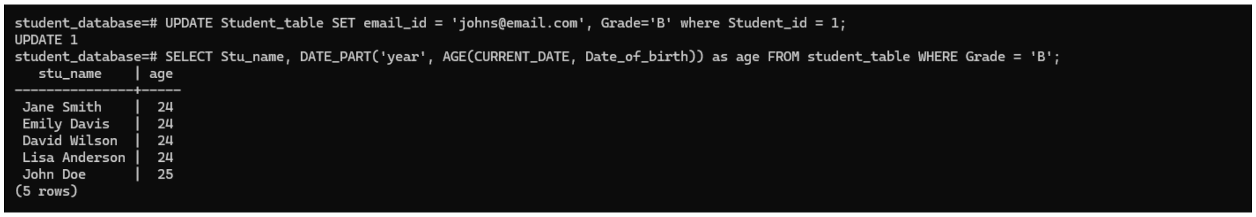
6.Update Student Email and Grade

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

Query:

UPDATE student_table SET email_id = johns@email.com', Grade = 'B' WHERE Student_id = 1;

Output:



```
student_database=# UPDATE Student_table SET email_id = 'johns@email.com', Grade='B' where Student_id = 1;
UPDATE 1
student_database=# SELECT Stu_name, DATE_PART('year', AGE(CURRENT_DATE, Date_of_birth)) as age FROM student_table WHERE Grade = 'B';
   stu_name    | age
---------------+-----
 Jane Smith    |  24
 Emily Davis   |  24
 David Wilson  |  24
 Lisa Anderson |  24
 John Doe      |  25
(5 rows)
```

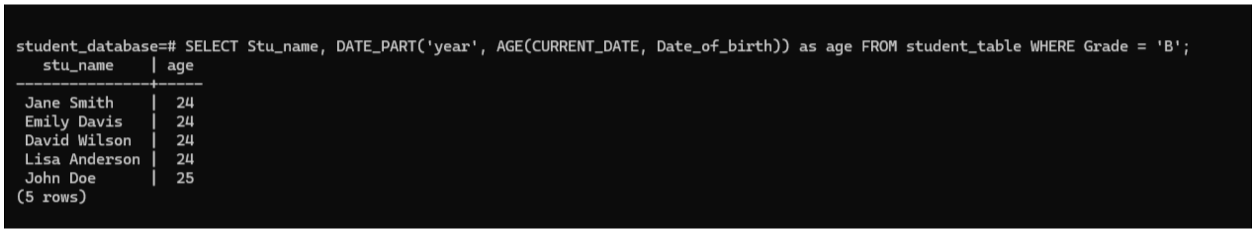
7.Query for Students with Grade "B"

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

Query:

SELECT Stu_name, DATE_PART('year', AGE(CURRENT_DATE, Date_of_birth)) as age FROM student_table WHERE Grade = 'B';

Output:



```
student_database=# SELECT Stu_name, DATE_PART('year', AGE(CURRENT_DATE, Date_of_birth)) as age FROM student_table WHERE Grade = 'B';
   stu_name    | age
---------------+-----
 Jane Smith    |  24
 Emily Davis   |  24
 David Wilson  |  24
 Lisa Anderson |  24
 John Doe      |  25
(5 rows)
```

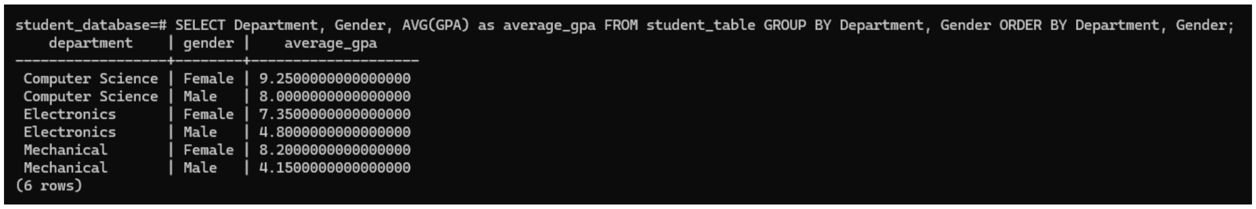
8.Grouping and Calculation

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

Query:

SELECT Department, Gender, AVG(GPA) as average_gpa FROM student_table GROUP BY Department, Gender ORDER BY Department, Gender;

Output:



```
student_database=# SELECT Department, Gender, AVG(GPA) as average_gpa FROM student_table GROUP BY Department, Gender ORDER BY Department, Gender;
    department    | gender |    average_gpa
------------------+--------+--------------------
 Computer Science | Female | 9.2500000000000000
 Computer Science | Male   | 8.0000000000000000
 Electronics      | Female | 7.3500000000000000
 Electronics      | Male   | 4.8000000000000000
 Mechanical       | Female | 8.2000000000000000
 Mechanical       | Male   | 4.1500000000000000
(6 rows)
```

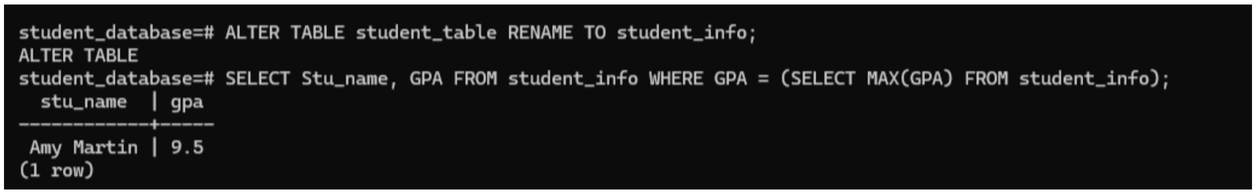
9.Table Renaming

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

Query:

ALTER TABLE student_table RENAME TO student_info;

Output:



```
student_database=# ALTER TABLE student_table RENAME TO student_info;
ALTER TABLE
student_database=# SELECT Stu_name, GPA FROM student_info WHERE GPA = (SELECT MAX(GPA) FROM student_info);
  stu_name  | gpa
------------+-----
 Amy Martin | 9.5
(1 row)
```

10.Retrieve Student with Highest GPA

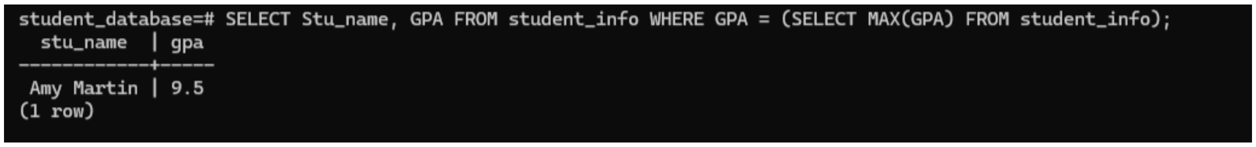
Write a query to retrieve the name of the student with the highest GPA from the

"student_info" table.

Query:

SELECT Stu_name, GPA FROM student_info WHERE GPA = (SELECT MAX(GPA) FROM student_info);

Output:



```
student_database=# SELECT Stu_name, GPA FROM student_info WHERE GPA = (SELECT MAX(GPA) FROM student_info);
  stu_name  | gpa
------------+-----
 Amy Martin | 9.5
(1 row)
```