# PostgreSQL Assignment

### Assignment Description

In this assignment, you will work with PostgreSQL, a powerful open-source relational database management system. Your task involves creating 03 tables based on the provided sample data and then writing and executing queries to perform various database operations such as creating, reading, updating, and deleting data. Additionally, you will explore concepts like LIMIT and OFFSET, JOIN operations, GROUP BY, aggregation and LIKE.

## Instructions:

### Database Setup:

* Create a fresh database titled **"university\_db"** or any other appropriate name.

## Table Creation:

Create a **"students"** table with the following fields:

* student\_id (Primary Key): Integer, unique identifier for students.
* student\_name: String, representing the student's name.
* age: Integer, indicating the student's age.
* email: String, storing the student's email address.
* frontend\_mark: Integer, indicating the student's frontend assignment marks.
* backend\_mark: Integer, indicating the student's backend assignment marks.
* status: String, storing the student's result status.

Create a **"courses"** table with the following fields:

* course\_id (Primary Key): Integer, unique identifier for courses.
* course\_name: String, indicating the course's name.
* credits: Integer, signifying the number of credits for the course.

Create an **"enrollment"** table with the following fields:

* enrollment\_id (Primary Key): Integer, unique identifier for enrollments.
* student\_id (Foreign Key): Integer, referencing student\_id in "Students" table.
* course\_id (Foreign Key): Integer, referencing course\_id in "Courses" table.

## Sample Data

* Insert the following sample data into the **"students"** table:

| **student\_id** | **student\_name** | **age** | **email** | **frontend\_mark** | **backend\_mark** | **status** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Alice | 22 | [alice@example.com](mailto:alice@example.com) | 55 | 57 | NULL |
| 2 | Bob | 21 | [bob@example.com](mailto:bob@example.com) | 34 | 45 | NULL |
| 3 | Charlie | 23 | [charlie@example.com](mailto:charlie@example.com) | 60 | 59 | NULL |
| 4 | David | 20 | [david@example.com](mailto:david@example.com) | 40 | 49 | NULL |
| 5 | Eve | 24 | [newemail@example.com](mailto:newemail@example.com) | 45 | 34 | NULL |
| 6 | Rahim | 23 | [rahim@gmail.com](mailto:rahim@gmail.com) | 46 | 42 | NULL |

* Insert the following sample data into the **"courses"** table:

| **course\_id** | **course\_name** | **credits** |
| --- | --- | --- |
| 1 | Next.js | 3 |
| 2 | React.js | 4 |
| 3 | Databases | 3 |
| 4 | Prisma | 3 |

* Insert the following sample data into the **"enrollment"** table:

| **enrollment\_id** | **student\_id** | **course\_id** |
| --- | --- | --- |
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 2 | 1 |
| 4 | 3 | 2 |

## Execute SQL queries to fulfill the ensuing tasks:

### Query 1:

Insert a new student record with the following details:

* Name: YourName
* Age: YourAge
* Email: YourEmail
* Frontend-Mark: YourMark
* Backend-Mark: YourMark
* Status: NULL

### Query 2:

Retrieve the names of all students who are enrolled in the course titled 'Next.js'.

**Sample Output:**

| **student\_name** |
| --- |
| Alice |
| Bob |

### Query 3:

Update the status of the student with the highest total (frontend\_mark + backend\_mark) mark to 'Awarded'

### Query 4:

Delete all courses that have no students enrolled.

### Query 5:

Retrieve the names of students using a limit of 2, starting from the 3rd student.

**Sample Output:**

| **student\_name** |
| --- |
| Charlie |
| David |

### Query 6:

Retrieve the course names and the number of students enrolled in each course.

**Sample Output:**

| **course\_name** | **students\_enrolled** |
| --- | --- |
| Next.js | 2 |
| React.js | 2 |

### Query 7:

Calculate and display the average age of all students.

**Sample Output:**

| **average\_age** |
| --- |
| 22.2857142857142857 |

### Query 8:

Retrieve the names of students whose email addresses contain 'example.com'.

**Sample Output:**

| **student\_name** |
| --- |
| Alice |
| Bob |
| Charlie |
| `David |

Prepare the SQL code for table creation, sample data insertion, and the seven queries in a text document or your preferred format. Include comments explaining each query's purpose and functionality. **Save your document as "PostgreSQL\_Assignment.sql" or any other appropriate name.**

**Based on the above table data explain the concept along with the example for below items**

1. Explain the primary key and foreign key concepts in PostgreSQL.
2. What is the difference between the VARCHAR and CHAR data types?
3. Explain the purpose of the WHERE clause in a SELECT statement.
4. What are the LIMIT and OFFSET clauses used for?
5. How can you perform data modification using UPDATE statements?
6. What is the significance of the JOIN operation, and how does it work in PostgreSQL?
7. Explain the GROUP BY clause and its role in aggregation operations.
8. How can you calculate aggregate functions like COUNT, SUM, and AVG in PostgreSQL?
9. What is the purpose of an index in PostgreSQL, and how does it optimize query performance?
10. Explain the concept of a PostgreSQL view and how it differs from a table.