Building & Hosting a Student Database on Cloud

This Activity will challenge you to develop a database application to manage student information. You'll leverage the following technologies:

1. **Render:** Create a PostgreSQL database instance on the Render platform.
2. **Database Client:** Choose either pgAdmin or a VS Code extension to connect and interact with the PostgreSQL database.
3. **SQL Commands:** Write SQL statements to create a table for storing student data and perform CRUD (Create, Read, Update, Delete) operations on that table.

Tasks:

1. **Setting Up the Database:**
    - Sign up for a free Render account (if you don't have one already).
    - Create a new PostgreSQL database instance on Render.
    - Secure your database by potentially restricting external access (optional).
2. **Connecting to the Database:**
    - Choose your preferred tool: pgAdmin or a VS Code extension for PostgreSQL.
    - Establish a connection to your Render-hosted PostgreSQL database using the provided credentials.
3. **Schema Design:**
    - Plan the structure of your student data table. Consider attributes like student ID, name, program, and contact information.
    - Use SQL's CREATE TABLE statement to define the table schema in your PostgreSQL database.
4. **CRUD Operations:**
    - **Create:** Insert some sample student records into the newly created table using INSERT statements.
    - **Read:** Retrieve student information using SELECT statements. Try filtering data based on specific criteria (e.g., program, ID).
    - **Update:** Modify existing student data using UPDATE statements. Simulate scenarios like changing a student's program or contact details.
    - **Delete:** Remove unwanted student records using DELETE statements. Ensure you understand the implications of deleting data.

Deliverables:

- A well-defined PostgreSQL database schema for storing student information.
- SQL scripts demonstrating CRUD operations on the student data table.
- Documentation explaining each step of the process, including screenshots or visuals where relevant.

Bonus Challenge:

- Implement data validation rules within the database schema to ensure data integrity (e.g., enforce unique student IDs).