Lab Sheet 8: Programming Language with DBMS

Department of Computer Science and Engineering

Pre-requisites

- Basic understanding of SQL (DDL, DML, and query operations)
- Familiarity with a programming language (Python recommended but anything will do)
- Docker installed on the local machine

Lab Environment Setup

1. Install Docker

1. Visit https://docs.docker.com/get-docker/ and install Docker Desktop (Windows/Mac) or Docker Engine (Linux).
2. Verify installation:

```
docker --version
```

2. Setup PostgreSQL using Docker

1. Pull the official PostgreSQL image:

```
docker pull postgres
```

2. Run a PostgreSQL container:

```
docker run -e POSTGRES_PASSWORD=admin123 -p 5432:5432 -d postgres
```

3. Access the PostgreSQL shell:

```
docker exec -it pg_lab psql -U postgres
```

4. Using psql, connect to PostgreSQL as above and create a new database named `studentdb`.

Lab Task Overview

Each student must write a program that:

- Connects to the PostgreSQL database using appropriate drivers/libraries
- Accepts user input via the terminal
- Executes SQL commands for creating tables (DDL)
- Inserts, updates, and deletes records (DML)
- Performs queries and displays results

Lab Tasks

Write a program in your favourite language which has one function to do each of the following.

Task 1: Table Creation (DDL)

1. Within `studentdb`, create a table using the name and column description provided as parameters to the function.
2. Confirm creation by querying the `information_schema.tables`.

Task 2: Data Insertion and Manipulation (DML)

1. Prompt the user (through the terminal) to enter details of multiple students.
2. Insert the entered data into the `Students` table.
3. Allow the user to:
    - Update a student's department based on name.
    - Delete a student record by ID.

Task 3: Query Operations

1. Write a function that:
    - Retrieves and displays all records from the `Students` table.
    - Displays students belonging to a user-specified department.
    - Displays average age of students in each department (using GROUP BY).
    - Finds all students whose names start with a user-provided letter (pattern matching).

Task 4: User Interaction and Menu Design

Design a simple text-based menu to allow the user to choose operations, for example:

```
1. Create Table
2. Insert Student
3. Update Student
4. Delete Student
5. Query Data
6. Exit
Enter your choice:
```

Task 5: Program Termination and Cleanup

- On exit, close all database connections gracefully.
- Ensure that exceptions (like invalid input or connection errors) are handled properly.

Submission Instructions

- Submit the program files in a zipped file along with the following:
    1. source file(s)
    2. `requirements.txt` (if applicable)
    3. `README.md` explaining setup and execution steps
- Ensure that your code runs correctly inside the PostgreSQL Docker environment.