Lab 2

Workflow with PostgreSQL

CSE 4308

DATABASE MANAGEMENT SYSTEMS LAB

Prepared by: Md. Tariquzzaman

Academic Year: 2024–2025

Objective

Learn to configure PostgreSQL for password authentication, create a PostgreSQL role, assign a password, create a dedicated database, load SQL files, and perform CRUD operations inside the student's own database.

Lab Workflow

Repeat this subsection for every single lab. If anything in this section fails in IUT lab PCs, check the commands given at the end of the lab manual after notes.

1. Connect to the internet by running nmtui in terminal, add a dsl connection, then go back and activate that dsl connection. Make sure that your net id is not being used in any other pc or router, if you face any issue, use any other net id from your friends which is not being used.

2. Download all the files posted in the google classroom.

3. Disconnect from the **dsl** connection via nmtui, connect to the ethernet/wired connection via nmtui. If no such connection exists, create an **ethernet** connection and then activate.

4. **Create Your Role and Database**

   Inside the PostgreSQL shell, create your own role (user) and database.

   1. Login to PostgreSQL as 'postgres':

   Log in again to PostgreSQL using 'postgres':

```
sudo -u postgres psql
```

   2. Create Your Role and Database

   Important: If the role or database already exists, we must drop them first to avoid conflicts.

```
-- Terminate active connections to the database
SELECT pg_terminate_backend(pid)
FROM pg_stat_activity
WHERE datname = 'cse4308_12345';

-- Drop database if it exists
DROP DATABASE IF EXISTS cse4308_12345;

-- Drop role if it exists
DROP ROLE IF EXISTS alice;

CREATE ROLE alice LOGIN;
ALTER ROLE alice WITH PASSWORD 'alice123';

CREATE DATABASE cse4308_12345 OWNER alice;

GRANT ALL PRIVILEGES ON DATABASE cse4308_12345 TO alice;
```

This script will:

- Terminate active sessions to allow safe dropping.
- Drop the existing `cse4308_12345` database and `alice` role if present.
- Create a new `alice` role with a password.
- Create the database `cse4308_12345` owned by `alice`.
- Grant all permissions to `alice`.

3. Exit PostgreSQL:

```
\q
```

5. **Connect Using Your Newly Created User**

Now connect to your own database as the ‘alice‘ user.

1. Connect to Your Database:

Use the following command to log in as the ‘alice‘ user:

```
psql -U alice -d cse4308_12345
```

2. Verify the Connection:

Verify that you are connected to the correct database with the following command:

```
\conninfo
```

Expected output:

```
You are connected to database "cse4308_12345" as user "alice".
```

3. Exit PostgreSQL:

```
\q
```

6. **Load the Lab SQL File (MUST be run outside psql)**

Load the SQL file that sets up the tables and inserts initial data.

1. Load the SQL File:

Run the following command to load the lab SQL file:

```
psql -U alice -d cse4308_12345 -f /home/cse/Downloads/lab2.sql
```

This will: - Drop old tables (if they exist). - Create new tables. - Insert initial rows of data.

7. **Reconnect and Verify Tables**

   After loading the SQL file, reconnect to PostgreSQL to verify that the tables have been created.

   1. Reconnect to Your Database:

```
psql -U alice -d cse4308_12345
```

   2. Check Tables:

   Verify the tables have been created by listing them:

```
\dt
```

   3. View Table Data:

   Verify the data by running the following SQL queries:

```
SELECT * FROM students;
SELECT * FROM courses;
```

8. Open a text editor and save it in the appropriate submission name format `fullid_CSE4308_L2_T1.sql`. You will use this text editor to write your solutions, make sure the save it after writing each solution. The structure of this file will be:

```
-- Task 1
INSERT INTO students (student_id, name, major, gpa)
VALUES (104, 'David', 'ME', 3.60);
--Task 2
SELECT name, gpa FROM students WHERE major = 'CSE';
```

Tasks

Perform the basic CRUD operations (Create, Read, Update, Delete) on the tables. 1. CREATE:

```
INSERT INTO students (student_id, name, major, gpa)
VALUES (104, 'David', 'ME', 3.60);

INSERT INTO courses (course_id, course_name, credits)
VALUES (4, 'Machine Learning', 3);
```

2. READ:

```
SELECT * FROM students;
SELECT * FROM courses;

SELECT name, gpa FROM students WHERE major = 'CSE';
SELECT course_name FROM courses WHERE credits = 3;
SELECT * FROM students ORDER BY gpa DESC;
```

3. UPDATE:

```
UPDATE students
SET gpa = 3.55
WHERE name = 'Bob';

UPDATE courses
SET credits = 4
WHERE course_name = 'Operating Systems';
```

4. DELETE:

```
DELETE FROM students WHERE name = 'Charlie';
DELETE FROM courses WHERE course_name = 'Digital Electronics';
```

5. Verify:

```
SELECT * FROM students;
SELECT * FROM courses;
```

1. **Exit psql**

   Once you've completed all operations, exit the PostgreSQL shell:

```
\q
```

2. **Submit Work**

   Save all successfully executed commands into a '.sql' file using a text editor.

   1. Save Your Work:

   Name the file as: `fullid_CSE4308_L2_T1.sql`

   2. Organize Your Files:

   Save your files in the following structure:

```
~/Documents/CSE4308_Labs/
                        Lab2/
                            lab2.sql
                            12345_CSE4308_L2_T1.sql
```

Notes

- All work must be performed inside **your** database and **your** user.
- Always run SQL file loading using `psql -U <user> -d <database> -f ...`.
- Save only commands that executed successfully. Logging in and running lab.sql file is not required.

Configure PostgreSQL to Use MD5 Authentication

Try only if you faced error while ". Create Your Role and Database" or "Connect Using Your Newly Created User"

You only need to do this once in a PC in any lab. Once done, **do not perform these again**. Note: your lab might have already been configured with the following by a previous user.

Before you start creating users and databases, we need to modify the PostgreSQL authentication method to allow password-based authentication.

1. Open the 'pg_hba.conf' File:

   Use a text editor to open the 'pg_hba.conf' file:

   ```
   sudo nano /etc/postgresql/14/main/pg_hba.conf
   ```

2. Modify Authentication Method to 'md5':

   Locate the following line:

   ```
   local   all             all                                     peer
   ```

   Change 'peer' to 'md5':

   ```
   local   all             all                                     md5
   ```

3. Save and Exit:

   Press 'CTRL + O' to save and 'CTRL + X' to exit the editor.

4. Restart PostgreSQL:

   To apply the changes, restart the PostgreSQL service:

   ```
   sudo systemctl restart postgresql
   ```

   Set Password for 'cse' User

   Now, you need to set or reset the password for the 'cse' user.

5. Login to PostgreSQL as 'postgres':

   If you are not logged in as 'postgres', log in as the superuser:

   ```
   sudo -u postgres psql
   ```

6. Set Password for 'cse':

   To set the password for the 'cse' user, run the following command inside PostgreSQL:

```
ALTER ROLE cse WITH PASSWORD '123';
```

7. Exit PostgreSQL:

   After setting the password, exit the PostgreSQL shell:

```
\q
```