

ACCA7002 Database Systems

Tutorial 12

Database Security in PostgreSQL using pgAdmin 4

Exercise 12.1: Implement basic security measures in PostgreSQL database using pgAdmin 4.

Tasks:

Open pgAdmin 4:

- Open pgAdmin 4 and connect to your PostgreSQL server.
- Create a New Database:
 - In pgAdmin 4, right-click on "Databases" and select "Create > Database..."
 - Name the database as lab_security and click "Save."

Create a User:

- In pgAdmin 4, right-click on "Login/Group Roles" and select "Create > Login/Group Role..."
- Name the role as lab_user and set a secure password in the "Definition" tab.

Grant Permissions:

- Right-click on the lab_security database, navigate to "Tools," and select "Query Tool."
- Execute the following SQL query to grant all privileges to lab_user:

```
GRANT ALL PRIVILEGES ON DATABASE lab_security TO lab_user;
```

Test User Access:

- Log in as lab_user and test access to the lab_security database.

Implement Role-Based Access Control:

- In pgAdmin 4, navigate to the lab_security database.
- Go to "Tools > Query Tool" and execute the following SQL queries to create a new role (read_only) and grant SELECT privileges on all tables in the public schema:

```
CREATE ROLE read_only;
```

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO read_only;
```

Encrypt Passwords:

- In pgAdmin 4, navigate to "Login/Group Roles," right-click on lab_user, and select "Properties."
- In the "Definition" tab, set a new password and choose the MD5 option.

Audit Logging:

- In pgAdmin 4, navigate to "File > Options > Logging" and enable logging options.

Set Database Connection Limits:

- In pgAdmin 4, navigate to "Databases > lab_security > Properties."
- Set the "Max Connections" property to limit the maximum number of connections.

SSL/TLS Configuration:

- Configure SSL/TLS for secure connections. In pgAdmin 4, navigate to "File > Options > SSL" and configure the required options.

Backup and Recovery:

- Use pgAdmin 4 to perform a database backup. Right-click on the database, select "Backup," and configure the backup options.

Exercise 11.2: Creating Roles and Group Roles in PostgreSQL using pgAdmin 4

Objective: Create roles and group roles in PostgreSQL using pgAdmin 4 to manage access control.

Tasks:

Open pgAdmin 4:

- Open pgAdmin 4 and connect to your PostgreSQL server.

Create a New Role:

- In pgAdmin 4, right-click on "Login/Group Roles" and select "Create > Login/Group Role..."
- Name the role as lab_role and set a secure password in the "Definition" tab.
- Configure other role properties as needed (e.g., set a specific login role, limit connection settings).

Create a Group Role:

- In pgAdmin 4, right-click on "Login/Group Roles" and select "Create > Login/Group Role..."
- Name the role as lab_group and set a secure password in the "Definition" tab.
- In the "Role Membership" tab, add lab_role to the "Member of" list. This makes lab_group a group role containing lab_role.

Grant Permissions:

- Right-click on the lab_role role, and select "Properties."
- In the "Role Privileges" tab, grant necessary privileges on databases, schemas, or other objects.

Test Role Access:

- Log in as lab_role and test access to the database and objects for which privileges were granted.

Group Role Access:

- Log in as lab_group and test access. Since lab_group includes lab_role, it should inherit the privileges granted to lab_role.

Additional Permissions:

- Explore additional permissions such as login, superuser, replication, etc., in the "Definition" tab of role properties.

Audit Logging:

- In pgAdmin 4, navigate to "File > Options > Logging" and enable logging options for the roles created.

Additional Challenges (Optional):

- Explore advanced settings for roles, such as password expiration, connection limits, and role attributes.
- Create roles with specific login restrictions (e.g., allowed IP addresses).
- Experiment with different membership scenarios for group roles.

-----End-----