**Lab Exercise 13- Role-Based Access Control in PostgreSQL**

**Objective**

* Understand how to create roles and assign privileges
* Implement user and group roles
* Restrict access to specific database objects
* Apply RBAC concepts practically

**Prerequisites**

* PostgreSQL installed on Windows
* Access to SQL Shell or pgAdmin
* A database for testing access control

**Step 1: Create a Test Database and Table**

Connect to the PostgreSQL server as a superuser (such as postgres) and run:

CREATE DATABASE rbac\_test;

\c rbac\_test

CREATE TABLE department (

id SERIAL PRIMARY KEY,

name TEXT,

head TEXT

);

**Step 2: Create Roles (Users and Groups)**

Create a group role for read-only users

CREATE ROLE readonly;

Create individual users and assign passwords

CREATE ROLE alice LOGIN PASSWORD 'alicepass';

CREATE ROLE bob LOGIN PASSWORD 'bobpass';

Grant the readonly group role to these users

GRANT readonly TO alice;

GRANT readonly TO bob;

**Step 3: Set Permissions**

By default, users cannot access any tables unless granted

Grant specific permissions to the readonly group

GRANT CONNECT ON DATABASE rbac\_test TO readonly;

GRANT USAGE ON SCHEMA public TO readonly;

GRANT SELECT ON department TO readonly;

**Step 4: Verify Access**

Connect as user alice and try selecting from the table

\c rbac\_test alice

SELECT \* FROM department;

You should see results or an empty table, depending on whether data is present

Try inserting data (should fail)

INSERT INTO department (name, head) VALUES ('HR', 'Tom');

This should return a permission denied error

**Step 5: Create a Role with Write Access**

Create a group role for writers

CREATE ROLE writer;

Create another user

CREATE ROLE charlie LOGIN PASSWORD 'charliepass';

GRANT writer TO charlie;

Grant write permissions

GRANT INSERT, UPDATE, DELETE ON department TO writer;

GRANT CONNECT ON DATABASE rbac\_test TO writer;

GRANT USAGE ON SCHEMA public TO writer;

**Step 6: Test Write Access**

Connect as user charlie

\c rbac\_test charlie

INSERT INTO department (name, head) VALUES ('IT', 'Sam');

You should see success confirming insert access

**Step 7: Revoke Access (Optional)**

To remove access from a role

REVOKE SELECT ON department FROM readonly;

Or remove a user from a role

REVOKE readonly FROM alice;

**Summary Table**

| **Action** | **Command Example** |
| --- | --- |
| Create group role | CREATE ROLE readonly; |
| Create user | CREATE ROLE alice LOGIN PASSWORD 'x'; |
| Grant role to user | GRANT readonly TO alice; |
| Grant table permission | GRANT SELECT ON table TO readonly; |
| Revoke table permission | REVOKE SELECT ON table FROM readonly; |
| Check user privileges | \du or SELECT \* FROM pg\_roles; |