**PSQL DB Security**

**Subject:** PostgreSQL

**Students:** BOURBAI Ismail

**CREATE ROLE:**

PostgreSQL uses the roles concept to manage database access permissions. A role can be a user or a group, depending on how you setup the role. A role that has login right is called user. A role may be a member of other roles, which are known as groups.

To create a new role, you use the CREATE ROLE statement as follows:

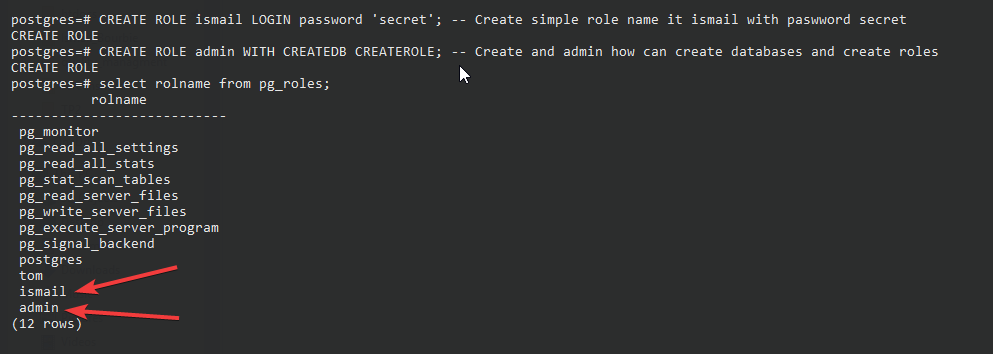


Figure 1 - Create new Role

Make **ismail** the **owner** of test\_base database:

ALTER DATABASE test\_base OWNER TO ismail;

Now the user ismail can connect to test\_base:

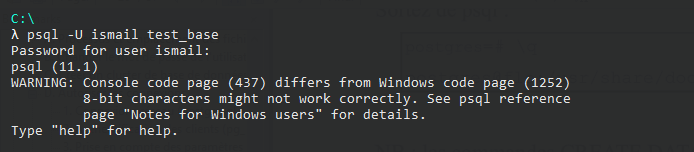


Figure 2 - Connect With User

**Privileges:**

When an object is created, it is assigned an owner. The owner is normally the role that executed the creation statement. For most kinds of objects, the initial state is that only the owner (or a superuser) can do anything with the object. To allow other roles to use it, privileges must be granted.

There are different kinds of privileges: SELECT, INSERT, UPDATE, DELETE, TRUNCATE, REFERENCES, TRIGGER, CREATE, CONNECT, TEMPORARY, EXECUTE, and USAGE. The privileges applicable to a particular object vary depending on the object's type (table, function, etc). For complete information on the different types of privileges supported by PostgreSQL.

To assign privileges, the GRANT command is used. For example, if “ismail” is an existing role, and “test\_table” is an existing table, the privilege to update the table can be granted with:

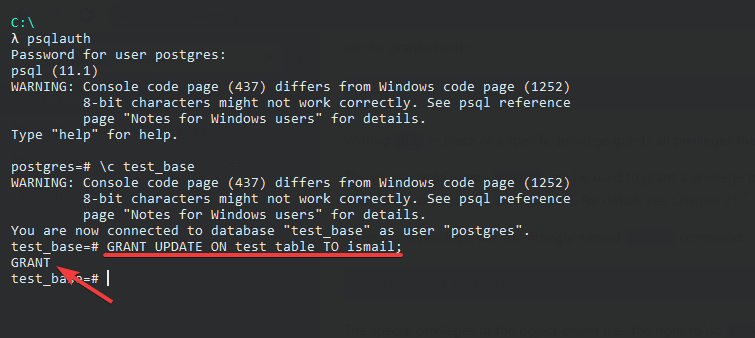


Figure 3 - GRANT example

To revoke a privilege, use REVOKE command:

* Revoke all privilege:

REVOKE ALL PRIVILEGES ON test\_table FROM ismail;

* Revoke specific privilege from the public (for ex: INSERT):

REVOKE INSERT ON test\_table FROM PUBLIC;