

# **CREATE USER/ROLE to Add Users**

by Sophia

WHAT'S COVERED

This tutorial explores creating users in a database in two parts:

- 1. CREATE ROLE
- 2. Other Attributes

### **1. CREATE ROLE**

For many databases, there is a difference between user accounts and groups. User accounts are created and added to groups. As a best practice, permissions are set to a group and are applied to the user accounts that are added to the group. In PostgreSQL, roles are used for both users and groups. Roles that can log into the database, which are also called login roles, are the same as users.

When we have roles that contain other roles, they are called group roles, which are the same as groups in other databases.

The basic way to create a role is by using the CREATE ROLE statement:

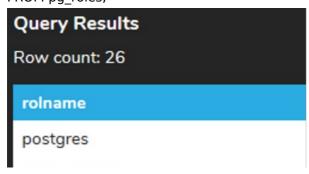
CREATE ROLE <rolename>;

For example, if we wanted to create a role named "newaccount", we can run the following:

CREATE ROLE newaccount;

We can see which roles exist in our database by running the following command:

SELECT rolname FROM pg\_roles;



 $\odot$  2022 SOPHIA Learning, LLC. SOPHIA is a registered trademark of SOPHIA Learning, LLC.

pg\_monitor

pg\_read\_all\_settings

pg\_read\_all\_stats

pg\_stat\_scan\_tables

pg\_read\_server\_files

pg\_write\_server\_files

pg\_execute\_server\_program

pg\_signal\_backend

sophia

developer

sales

security

davida

southwest

testuser1

temporary\_user

master\_user

marketing

technology

database\_admin

database\_test

regular\_user

db\_role

administrator

newaccount

Note that the column is rolname and not rolename. We can see the newaccount is the last role in the list. All of the role names that start with the prefix pg\_ are system roles within the database.

This newaccount cannot log in, as we have not defined the LOGIN attribute to that role. To create a login role, or a user, we must use the LOGIN attribute and the initial password. Let us go ahead and create one:

## CREATE ROLE myaccount

#### PASSWORD 'mypassword';

This will create a login role, or user, named myaccount with the password mypassword. Note that we have to use single quotes around the password. Also note that even with this account created, we will not be able to switch to it in our web interface. However, if we wanted to log in through the psql client tool, the command would look like this:

#### psql -U myaccount -W postgres

The command would then prompt you to enter in the password.

### 2. Other Attributes

With the creation of the login role, there are additional attributes that can be used. The superuser attribute, for example, is one that can override all access restrictions in the database. It is a role that should be used only when truly needed, as it can basically do anything in a database. It would look like:

CREATE ROLE adminaccount SUPERUSER LOGIN PASSWORD 'secretpassword'; Another type of role is one that can create other databases in PostgreSQL. For this, the CREATEDB attribute is

CREATE ROLE dbaccount CREATEDB LOGIN PASSWORD 'securePass1'; The VALID UNTIL attribute can also be used. This allows you to enter in a date and time after which the role's password is no longer valid and the user can no longer log in. This can be useful for individuals that may only work at the company for a short period of time, such as a contractor.

CREATE ROLE contractaccount LOGIN PASSWORD 'securePass1' VALID UNTIL '2025-01-01'; After January 1st, 2025, the password for the role contractaccount would no longer be valid.



needed:

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.

### SUMMARY

The CREATE ROLE statement will allow us to create users with a variety of attributes.

Source: Authored by Vincent Tran