

# CREATE ROLE to Create Groups

by Sophia



## WHAT'S COVERED

This tutorial explores using the CREATE ROLE to create groups in a database in two parts:

1. Introduction
2. Exploring CREATE ROLE Options

## 1. Introduction

The CREATE ROLE statement can be used to create a user or a group, depending on how it is used. This is something unique to PostgreSQL, as other databases typically have a different concept to separate the user and group. We have already looked at the creation of a role for a user in the prior tutorial. This role needs the LOGIN attribute with a password that is set. Group roles, on the other hand, do not have a LOGIN attribute or password, as users are meant to inherit the permissions from the group role as a best practice.

## 2. Exploring CREATE ROLE Options

The structure of the CREATE ROLE command looks like this:

```
CREATE ROLE <rolename>  
<WITH>  
[SUPERUSER]  
[CREATEDB]  
[CREATEROLE]  
[INHERIT]  
[LOGIN]  
[CONNECTION LIMIT]  
[IN ROLE];
```

The keyword WITH is an optional keyword in the CREATE ROLE command. Notice that we have a lot of different options here that can be used. From the SUPERUSER to the BYPASSRLS option, there is a default option with NO in front of it like NOSUPERUSER or NOBYPASSRLS that is available. By default, it is used if the specific parameter is not used.

We looked at the SUPERUSER, CREATEDB, and LOGIN options in the prior tutorial. As a reminder, the SUPERUSER is one that has the ability to override all access restrictions. It can be a dangerous status as the

role can drop any objects or access any data. To create a role using the SUPERUSER attribute, the account must be a SUPERUSER.

The CREATEDB attribute just defines if the role is able to create databases. Typically a database administrator role would benefit from this attribute.

The CREATEROLE attribute allows you to create new roles, ALTER roles and DROP other roles.

The INHERIT option in this case is the default choice. There is the NOINHERIT attribute. This attribute determines if the role inherits the permissions and privileges of roles that it is a member of. A role that has the INHERIT attribute has the ability to use the database permissions that have been granted to all of the roles that it is directly or indirectly a member of.

The CONNECTION LIMIT determines how many concurrent connections the role can make. The default is set to -1 meaning that there is no limit.

The IN ROLE will list one or more existing roles in which the new role would be immediately added as a member. This could be a login role or another group role. For example, you could have an executive role that is set up to be part of a management role so that the executive role will get all of the permissions that the management role has.

There are other attributes that can be used with the CREATE ROLE that are very specific to a given scenario that we will not cover.

We can use the different attributes together in a single CREATE ROLE statement. For example, you could have created an admin role that has the ability to create databases and roles:

```
CREATE ROLE adminrole
```

```
CREATEDB
```

```
CREATEROLE;
```

This would create a group role called adminrole.



TRY IT

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 can be used to create group roles. Login roles can be a part of these.

Source: Authored by Vincent Tran