

GRANT to Assign Users

by Sophia Tutorial



WHAT'S COVERED

This tutorial explores the GRANT and REVOKE commands to assign user roles to a group role in two parts:

1. Introduction
2. Examples

1. Introduction

It can be useful to create group roles, as we did in the prior tutorial, so that it is easier to manage privileges and permissions. This is especially important as an organization has more users. This way, privileges and permissions can be granted to or revoked from a group as a whole rather than from the individual users.

As we have covered, typically a group role would not have the LOGIN attribute, although logically it can. However, it does not make sense to have it defined as such. Remember as well that in PostgreSQL, there is not a distinction between group roles or non-group roles. As such, you can grant membership to other group roles rather than just to user roles.

2. Examples

Let us first create a user role for an admin account:

```
CREATE ROLE myadmin  
LOGIN  
PASSWORD 'mypassword';
```

We can then create a group role called adminrole that has the various admin privileges:

```
CREATE ROLE adminrole  
CREATEDB  
CREATEROLE;
```

If we wanted to grant the adminrole role to the myadmin user role, we would do so with the GRANT command like:

```
GRANT adminrole TO myadmin;
```

If we wanted to grant to more than one user at a time, we could include the list of users separated by commas like:

```
GRANT adminrole TO myadmin1, myadmin2, myadmin3;
```

Say we wanted to separate out the admin role into one that could create roles and a separate one that could create databases:

```
CREATE ROLE adminrole_cr  
CREATEROLE;
```

```
CREATE ROLE adminrole_db  
CREATEDB;
```

We can grant permissions separately to myadmin by doing:

```
GRANT adminrole_cr TO myadmin;  
GRANT adminrole_db TO myadmin;
```

If we wanted to take away the permission to create databases, we can use the REVOKE command like:

```
REVOKE adminrole_db FROM myadmin;
```

Since both of those roles are group roles, you could grant them to each other, but only in one direction because the database will not allow you to set up circular membership loops:

```
GRANT adminrole_cr TO adminrole_db;  
GRANT adminrole_db TO adminrole_cr;
```

Query Results

Query failed because of: error: role "adminrole_db" is a member of role "adminrole_cr"



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

We can GRANT and REVOKE group roles to and from user roles and other group roles.

Source: Authored by Vincent Tran

