

## Управление доступом Роли и атрибуты



### **Авторские права**

© Postgres Professional, 2017 год.

Авторы: Егор Рогов, Павел Лузанов

### **Использование материалов курса**

Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

### **Обратная связь**

Отзывы, замечания и предложения направляйте по адресу:

[edu@postgrespro.ru](mailto:edu@postgrespro.ru)

### **Отказ от ответственности**

Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или непрямым, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

Роли

Атрибуты

Участие в групповых ролях

Владельцы объектов

## Роль

пользователь СУБД  
может включать в себя другие роли — быть «групповой ролью»  
никак не связана с пользователем ОС (хотя некоторые программы берут имя пользователя ОС как имя роли по умолчанию)  
определяется на уровне кластера

## Псевдороль public

неявно включает в себя все остальные роли

Любая роль может рассматриваться и как пользователь СУБД, и в то же время может включать в себя другие роли.

Роли никак не связаны с именами пользователей ОС, хотя некоторые программы это предполагают, выбирая значения по умолчанию.

Роли являются общими объектами кластера. Например, одна роль может подключаться к разным базам данных и быть владельцем объектов в разных БД.

При создании кластера определяется одна начальная роль, имеющая суперпользовательский доступ. В дальнейшем роли можно создавать, изменять и удалять.

Также существует псевдороль public, в которую всегда неявно включены все остальные роли.

<https://postgrespro.ru/docs/postgresql/10/database-roles>

Атрибуты определяют свойства роли

`CREATE ROLE` роль [WITH] атрибут [атрибут ...]

LOGIN	возможность подключения
SUPERUSER	суперпользователь
CREATEDB	возможность создавать базы данных
CREATEROLE	возможность создавать роли
REPLICATION	использование протокола репликации
и другие	

Роль обладает некоторыми атрибутами, определяющими ее общие особенности и права (не связанные с правами доступа к объектам).

Обычно атрибуты имеют два варианта, например, `CREATEDB` (дает право на создание БД) и `NOCREATEDB` (не дает такого права). Как правило, по умолчанию выбирается ограничивающий вариант.

Если у роли нет атрибута `LOGIN`, она не сможет подключиться к серверу. Такие роли тоже имеют смысл в качестве групповых.

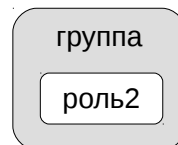
В таблице перечислены лишь некоторые из атрибутов. Атрибуты `INHERIT` и `BYPASSRLS` рассматривается дальше в этом модуле.

<https://postgrespro.ru/docs/postgresql/10/role-attributes>

<https://postgrespro.ru/docs/postgresql/10/sql-createrole>

## Включение роли в группу

*роль1*: GRANT группа TO *роль2*;



## Исключение роли из группы

*роль1*: REVOKE группа FROM *роль2*;

## Право управления участием в групповой роли имеют:

- любая роль — в самой себе
- роль с атрибутом SUPERUSER — в любой роли
- роль с атрибутом CREATEROLE — в любой, кроме суперпользовательской

Роль может быть включена в другую роль подобно тому, как пользователь Unix может быть включен в группу.

Однако PostgreSQL не делает различий между ролями-пользователями и ролями-группами. Поэтому любая роль может быть включена в любую другую. При этом возможно появление цепочек включений (но циклы не допускаются).

Смысл такого включения состоит в том, что для роли становятся доступны атрибуты (и привилегии, о которых пойдет речь дальше), которыми обладает групповая роль. Чтобы воспользоваться правами, которые дают атрибуты групповой роли, необходимо переключиться в нее командой SET ROLE.

Правом на включение и исключение других ролей в данную роль обладают:

- сама эта роль,
- роль с атрибутом SUPERUSER,
- роль с атрибутом CREATEROLE (если это не суперпользовательская роль).

<https://postgrespro.ru/docs/postgresql/10/role-membership>

# Передача права управления

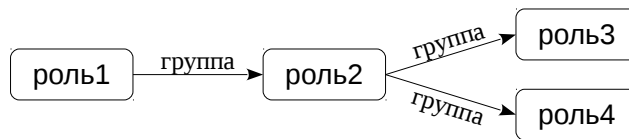
## Включение в группу с передачей права управления

*роль1*: GRANT группа TO *роль2* WITH ADMIN OPTION;

теперь *роль2* управляет группой, включая передачу права управления:

*роль2*: GRANT группа TO *роль3* WITH ADMIN OPTION;

*роль2*: GRANT группа TO *роль4* WITH ADMIN OPTION;



## Отзыв права передачи управления

*роль1*: REVOKE ADMIN OPTION FOR группа FROM *роль2*;

6

При включении роли в группу можно передать ей право управления (право на дальнейшее включение других ролей в эту группу и на исключение их из группы). Такие роли образуют иерархию включений.

Это право можно отобрать с помощью REVOKE ADMIN OPTION FOR, не исключая роль из группы.

<https://postgrespro.ru/docs/postgresql/10/sql-revoke>

## Владелец объекта

роль, создавшая объект  
(а также роли, включенные в нее)

может быть изменен командой `ALTER ... OWNER TO роль`

Когда роль создает в базе данных какие-либо объекты, она становится их *владельцем*. На самом деле владельцами считаются также и роли, включенные в создавшую объект роль.

При необходимости владельца объекта можно сменить. Для этого используется команда `ALTER` для соответствующего объекта с фразой `OWNER TO`.

Понятие владельца будет важно для следующей темы модуля — привилегий.





Роли объединяют концепции пользователей и групп

Атрибуты определяют свойства ролей

Роли можно включать друг в друга

1. Создайте роль creator без права входа в систему, но с правом создания баз данных и ролей.
2. Создайте пользователя weak с правом входа в систему.
3. Убедитесь, что weak не может создать базу данных.
4. Включите пользователя weak в группу creator.
5. Создайте новую базу данных под пользователем weak.