



Пользователи и роли

Концепция управления доступом

Создание ролей и их атрибуты

Участие ролей в группах

Изменение и удаление ролей

Просмотр информации

Роли

пользователь или группа
определяются на уровне кластера

Права доступа к объектам БД определяются ролью

владелец объекта или выданные привилегии

Роль не связана с пользователем ОС

некоторые приложения (например, psql) берут имя пользователя ОС
как имя роли по умолчанию

Изначально одна суперпользовательская роль

обычно postgres (можно задать в initdb)

Пользователь — роль с правом входа

```
CREATE ROLE role LOGIN;
```

или

```
CREATE USER role;
```

или

```
$ createuser role
```

Группа — роль без права входа (как правило)

```
CREATE ROLE role;
```

или

```
$ createuser --no-login role
```

Помимо LOGIN можно указывать и другие атрибуты

`CREATE ROLE role [WITH] option ...`

<code>SUPERUSER</code>	суперпользователь
<code>CREATEDB</code>	право создавать базы данных
<code>CREATEROLE</code>	право создавать роли
<code>CONNECTION LIMIT <i>conlim</i></code>	ограничение количества сеансов
<code>VALID UNTIL '<i>timestamp</i>'</code>	ограничение срока действия
<code>...</code>	

Участие в групповой роли

Включение роли в группу

```
role1: GRANT group TO role2;
```



Исключение роли из группы

```
role1: REVOKE group FROM role2;
```

Право управления участием в групповой роли

роль с атрибутом SUPERUSER — в любой

роль с атрибутом CREATEROLE — кроме суперпользовательской

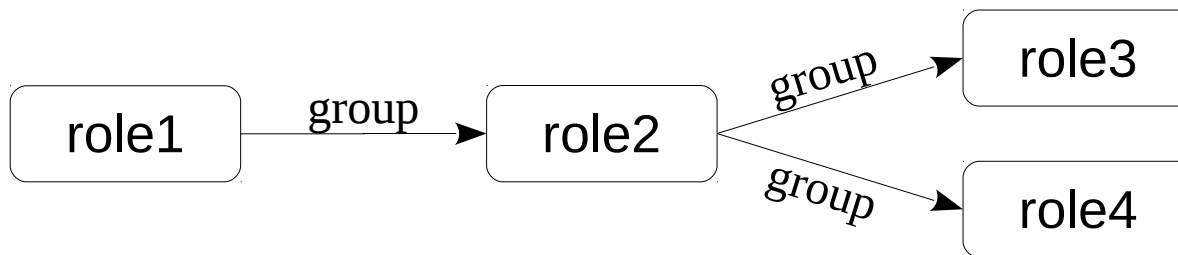
Включение с передачей права управления

```
role1: GRANT group TO role2 WITH ADMIN OPTION;
```

теперь *role2* может управлять *group*, включая передачу права управления

```
role2: GRANT group TO role3 WITH ADMIN OPTION;
```

```
role2: GRANT group TO role4 WITH ADMIN OPTION;
```



Отзыв права передачи управления

```
role1: REVOKE ADMIN OPTION FOR group FROM role2;
```

В качестве сокращений

```
CREATE ROLE role2 IN group;
```

```
    CREATE ROLE role2;  
    GRANT group TO role2;
```

```
CREATE ROLE group ROLE role2, role3;
```

```
    CREATE ROLE group;  
    GRANT group TO role2;  
    GRANT group TO role3;
```

```
CREATE ROLE group ADMIN role1, role2;
```

```
    CREATE ROLE group;  
    GRANT group TO role1 WITH ADMIN OPTION;  
    GRANT group TO role2 WITH ADMIN OPTION;
```


Изменение атрибутов роли

```
ALTER ROLE role [WITH] option ...;
```

Переименование роли

```
ALTER ROLE role RENAME TO newname;
```

Информация о ролях

```
select * from pg_roles;  
\du
```

Удаление роли

```
DROP ROLE role;
```

или

```
$ dropuser role
```

При удалении роли не должно остаться:

объектов, для которых эта роль является владельцем

```
REASSIGN OWNED BY role TO new_role;
```

выданных привилегий (не считая включения в группы)

```
DROP OWNED BY role;
```

Познакомились с концепцией управления доступом

Узнали, как создавать роли с разными правами

Научились включать роли друг в друга

Научились изменять и удалять роли

1. Завести роль creator без права входа в систему, но с правом создания баз данных и ролей.
2. Завести пользователя usr с правом входа в систему.
3. Убедиться, что usr не может создать базу данных.
4. Включить пользователя usr в группу creator.
5. Создать базу данных DB9 под пользователем usr.
6. Удалить базу данных и роли.



Авторские права

Курс «Администрирование PostgreSQL 9.4. Базовый курс» разработан в компании Postgres Professional (2015 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

Отказ от ответственности

Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или косвенным, 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.

Концепция управления доступом

Создание ролей и их атрибуты

Участие ролей в группах

Изменение и удаление ролей

Просмотр информации

Роли

пользователь или группа
определяются на уровне кластера

Права доступа к объектам БД определяются ролью

владелец объекта или выданные привилегии

Роль не связана с пользователем ОС

некоторые приложения (например, `psql`) берут имя пользователя ОС
как имя роли по умолчанию

Изначально одна суперпользовательская роль

обычно `postgres` (можно задать в `initdb`)

Роль объединяет в себе понятия пользователей и групп (в зависимости от настройки).

Управление доступом определяется в терминах ролей (подробно рассмотрено в следующей теме).

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

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

<http://www.postgresql.org/docs/current/static/user-manag.html>

<http://www.postgresql.org/docs/current/static/database-roles.html>

Пользователь — роль с правом входа

```
CREATE ROLE role LOGIN;  
или  
CREATE USER role;  
или  
$ createuser role
```

Группа — роль без права входа (как правило)

```
CREATE ROLE role;  
или  
$ createuser --no-login role
```

Можно считать, что «пользователь» в обычном понимании — это роль, допускающая вход в систему, а «группа» — роль без права входа.

Тем не менее, никто не запрещает иметь «групповую роль» с правом входа. Настройка ролей и полномочий сделана весьма гибко с тем, чтобы в каждом конкретном случае администратор мог выбрать наиболее удобную схему управления.

<http://www.postgresql.org/docs/current/static/database-roles.html>

<http://www.postgresql.org/docs/current/static/app-createuser.html>

Помимо LOGIN можно указывать и другие атрибуты

`CREATE ROLE role [WITH] option ...`

<code>SUPERUSER</code>	суперпользователь
<code>CREATEDB</code>	право создавать базы данных
<code>CREATEROLE</code>	право создавать роли
<code>CONNECTION LIMIT <i>conlim</i></code>	ограничение количества сеансов
<code>VALID UNTIL '<i>timestamp</i>'</code>	ограничение срока действия
<code>...</code>	

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

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

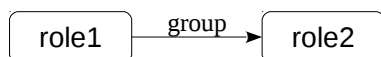
В таблице перечислены лишь некоторые из атрибутов. Атрибут `inherit` рассматривается в теме «Привилегии».

<http://www.postgresql.org/docs/current/static/role-attributes.html>

<http://www.postgresql.org/docs/current/static/sql-createrole.html>

Включение роли в группу

```
role1: GRANT group TO role2;
```



Исключение роли из группы

```
role1: REVOKE group FROM role2;
```

Право управления участием в групповой роли

роль с атрибутом SUPERUSER — в любой

роль с атрибутом CREATEROLE — 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 привилегии, которыми обладает групповая роль, и таким образом можно управлять правами на более крупном уровне. Это подробно рассматривается в теме «Привилегии».

Роль, включенную в группу, можно исключить из нее.

Правом на управление ролями (включение и исключение) обладает роль с атрибутом superuser (суперпользовательская роль), а также роль с атрибутом createrole. Последняя может управлять любой ролью, кроме суперпользовательских.

<http://www.postgresql.org/docs/current/static/role-membership.html>

Передача права управления

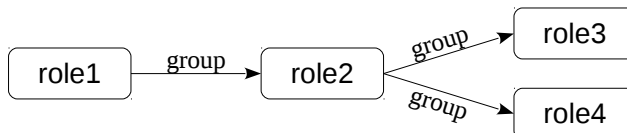
Включение с передачей права управления

```
role1: GRANT group TO role2 WITH ADMIN OPTION;
```

теперь role2 может управлять group, включая передачу права управления

```
role2: GRANT group TO role3 WITH ADMIN OPTION;
```

```
role2: GRANT group TO role4 WITH ADMIN OPTION;
```



Отзыв права передачи управления

```
role1: REVOKE ADMIN OPTION FOR group FROM role2;
```

7

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.

<http://www.postgresql.org/docs/current/static/sql-revoke.html>

В качестве сокращений

```
CREATE ROLE role2 IN group;  
    CREATE ROLE role2;  
    GRANT group TO role2;  
  
CREATE ROLE group ROLE role2, role3;  
    CREATE ROLE group;  
    GRANT group TO role2;  
    GRANT group TO role3;  
  
CREATE ROLE group ADMIN role1, role2;  
    CREATE ROLE group;  
    GRANT group TO role1 WITH ADMIN OPTION;  
    GRANT group TO role2 WITH ADMIN OPTION;
```

Для создания групповых ролей можно воспользоваться удобными сокращениями, приведенными на слайде.

Изменение атрибутов роли

```
ALTER ROLE role [WITH] option ...;
```

Переименование роли

```
ALTER ROLE role RENAME TO newname;
```

Информация о ролях

```
select * from pg_roles;  
\du
```

Можно изменить права роли, указав новые атрибуты в `alter role`.

Роль может быть переименована (ссылки на роль в системном каталоге — так же, как и на другие объекты — используют идентификаторы, а не имя).

<http://www.postgresql.org/docs/current/static/sql-alterrole.html>

Информацию о ролях можно увидеть в системном каталоге в таблице `pg_roles` или (более наглядно) с помощью `\du`.

Удаление роли

```
DROP ROLE role;  
или  
$ dropuser role
```

При удалении роли не должно остаться:

```
объектов, для которых эта роль является владельцем  
REASSIGN OWNED BY role TO new_role;  
  
выданных привилегий (не считая включения в группы)  
DROP OWNED BY role;
```

Прежде, чем роль может быть удалена, надо избавиться от всех объектов, владельцем которых является эта роль, и отозвать все выданные роли привилегии. Исключать роль из групп не требуется.

Первая задача может быть решена с помощью команды `reassign owned`, вторая — с помощью `drop owned` (так как эта команда удаляет не только объекты, принадлежащие роли, но и отзывает все привилегии, выданные этой роли).

<http://www.postgresql.org/docs/current/static/sql-droprole.html>

Познакомились с концепцией управления доступом

Узнали, как создавать роли с разными правами

Научились включать роли друг в друга

Научились изменять и удалять роли

1. Завести роль creator без права входа в систему, но с правом создания баз данных и ролей.
2. Завести пользователя usr с правом входа в систему.
3. Убедиться, что usr не может создать базу данных.
4. Включить пользователя usr в группу creator.
5. Создать базу данных DB9 под пользователем usr.
6. Удалить базу данных и роли.

Решение

```
# create role creator with createdb createrole;

# create role usr with login;

# \c - usr
# create database db;
-- ERROR:  permission denied to create database

# \c - postgres
# grant creator to usr;

# \c - usr
# set role creator; -- становимся на время creator-ом
# create database db9;
-- CREATE DATABASE

# \c - postgres
# drop database db9;
# drop role usr;
# drop role creator;
```