Базы данных

Лекция 12 – Администрирование сервера. Роли.

Преподаватель: Поденок Леонид Петрович, 505а-5

+375 17 293 8039 (505а-5)
+375 17 320 7402 (ОИПИ НАНБ)
prep@lsi.bas-net.by

ftp://student:2ok*uK2@Rwox@lsi.bas-net.by

Кафедра ЭВМ, 2024

2024.12.23

Оглавление

Роли базы данных....3
CREATE ROLE — создать роль в базе данных....5
DROP ROLE – удаления роли....12
SET ROLE — установить идентификатор текущего пользователя....13
GRANT — определить права доступа....15
GRANT для объектов баз данных....15
GRANT для ролей....16

Роли базы данных

Для управления разрешениями на доступ к базе данных Postgres использует концепцию ролей (roles). Роль — это сущность, которая может владеть объектами и иметь определённые права в базе.

Каждое подключение к серверу базы данных выполняется под именем конкретной роли, которая определяет начальные права доступа для команд, выполняемых в этом соединении.

Роль может представлять «пользователя», «группу» или и то, и другое, в зависимости от варианта использования. Роли базы данных концептуально полностью отличаются от пользователей операционной системы. Поддержание соответствия между ними может быть удобным, но не является обязательным.

Роли могут не только могут владеть объектами базы данных, но и выдавать другим ролям разрешения на доступ к этим объектам.

Роль может быть членом другой роли, используя, таким образом, ее права.

Роли определяются на уровне кластера баз данных и действуют для всех баз в кластере.

Для начальной настройки кластера базы данных система сразу после инициализации всегда содержит одну предопределённую роль с возможностью подключения.

Эта роль всегда является суперпользователем («superuser») и по умолчанию имеет такое же имя, как и пользователь операционной системы, инициализирующий кластер баз данных командой `initdb`, если при её запуске не указано другое имя.

Обычно, но необязательно, эта роль называется **`postgres`**.

Для создания других ролей вначале нужно подключиться с этой ролью.

Имя роли для конкретного подключения к базе данных указывается клиентской программой характерным для неё способом, таким образом инициируя запрос на подключение.

Программа psql для указания роли использует аргумент командной строки -U.

Многие приложения предполагают, что по умолчанию нужно использовать имя пользователя операционной системы (включая createuser и psql). Поэтому часто бывает удобным поддерживать соответствие между именами ролей и именами пользователей операционной системы.

CREATE ROLE — создать роль в базе данных

```
CREATE ROLE name [ [ WITH ] option [ ... ] ]
CREATE USER name
```

Добавляет новую роль в кластер баз данных Postgres.

Чтобы выполнить эту команду, необходимо быть суперпользователем или иметь право CREATEROLE.

Во время создания роли можно сразу назначить создаваемую роль членом существующей роли, а также назначить существующие роли членами создаваемой роли.

Правила, по которым включаются начальные параметры членства в роли, описаны ниже в предложениях IN ROLE, ROLE и ADMIN.

Все атрибуты, заданные в CREATE ROLE, могут быть изменены позднее командами ALTER ROLE.

Команды GRANT и REVOKE позволяют управлять назначением членов ролей и дают возможность изменять параметры после создания новой роли.

name – имя создаваемой роли. Должно соответствовать правилам именования идентификаторов SQL – либо обычное, без специальных символов, либо в двойных кавычках.

option – опции:

SUPERUSER | NOSUPERUSER – определяют, будет ли эта роль «суперпользователем», который может переопределять все ограничения доступа в базе данных.

Суперпользователь базы данных обходит все проверки прав доступа, за исключением права на вход в систему. Создать нового суперпользователя может только суперпользователь. По умолчанию подразумевается NOSUPERUSER.

LOGIN | NOLOGIN – определяют, разрешается ли новой роли вход на сервер, то есть, может ли эта роль стать начальным авторизованным именем при подключении клиента.

Можно считать, что роль с атрибутом LOGIN соответствует пользователю.

Роли без этого атрибута бывают полезны для управления доступом в базе данных, но это не пользователи в обычном понимании.

По умолчанию подразумевается вариант NOLOGIN, за исключением вызова CREATE ROLE в виде CREATE USER.

CREATEDB | NOCREATEDB – имеет ли право создавать базы данных.

По умолчанию подразумевается NOCREATEDB.

Указывать CREATEDB могут только суперпользователи и роли с правом CREATEDB.

Обычно роль должна явно иметь разрешение на создание базы данных (за исключением суперпользователей, которые пропускают все проверки).

CREATEROLE | NOCREATEROLE – имеет ли право создавать, изменять и удалять другие роли, добавлять для них комментарии и изменять метку безопасности.

Обычно роль должна явно иметь разрешение на создание других ролей.

Роль с правом CREATEROLE может изменять и удалять только те роли, которые были назначены пользователю с правами CREATEROLE и ADMIN OPTION.

Изменение роли включает большинство действий команды ALTER ROLE, например смену пароля, а также действия команд COMMENT и SECURITY LABEL в отношении ролей.

По сравнению с ролью SUPERUSER роль с правами CREATEROLE имеет определенныn ограничения (в отношении пользователей REPLICATION и BYPASSRLS, например).

REPLICATION | NOREPLICATION – определяет, будет ли роль ролью репликации.

Чтобы роль могла подключаться к серверу в режиме репликации (в режиме физической или логической репликации) и создавать/удалять слоты репликации, у неё должен быть этот атрибут (либо это должна быть роль суперпользователя).

Роль, имеющая атрибут REPLICATION, обладает очень большими привилегиями и поэтому этот атрибут должны иметь только роли, фактически используемые для репликации.

По умолчанию подразумевается вариант NOREPLICATION.

Выдавать атрибут REPLICATION могут только суперпользователи или роли с REPLICATION.

BYPASSRLS | NOBYPASSRLS – Эти предложения определяют, будут ли для роли игнорироваться все политики защиты на уровне строк (RLS).

Политики защиты строк (RLS, Row Level Security) позволяют управлять доступом к таблице на уровне отдельных строк. Выдавать атрибут BYPASSRLS могут только суперпользователи или роли BYPASSRLS.

По умолчанию подразумевается вариант NOBYPASSRLS.

ACHTUNG!!! pg_dump по умолчанию отключает row_security (устанавливает значение OFF), чтобы гарантированно было выгружено всё содержимое таблицы.

Если пользователь, запускающий pg_dump, не будет иметь необходимых прав, он получит ошибку.

Тем не менее, владелец выгружаемой таблицы всегда обходит защиту RLS.

INHERIT | NOINHERIT – влияет на статус наследования членства, когда данная роль добавляется как член другой роли, как в текущей, так и в будущих командах.

Управляет статусом наследования членства, добавленного этой командой с помощью предложений IN ROLE и ROLE.

Также используется по умолчанию для статуса наследования при добавлении данной роли в качестве члена с помощью команды GRANT.

По умолчанию используется INHERIT, при этом роль наследует права ролей, членом которых она является.

Иногда бывает нужно создать роль, которая наследовать права по умолчанию не будет.

Определённые атрибуты роли не наследуются, т. е. членство в роли, например, с правом CREATEDB, не позволит её участнику создавать новые базы данных, даже если членство было выдано с атрибутом INHERIT.

Если в выдаче членства есть атрибут SET, роль участника сможет выполнить команду SET ROLE для роли createdb, а затем создать новую базу данных.

CONNECTION LIMIT – Если роли разрешён вход, этот параметр определяет, сколько параллельных подключений может установить роль LOGIN.

```
CONNECTION LIMIT connlimit
```

connlimit – количество параллельных подключений. -1 снимает ограничение (устанавливается по умолчанию).

Под это ограничение подпадают только обычные подключения.

Подготовленные транзакции и соединения фоновых рабочих процессов не попадают.

PASSWORD – задает пароль для роли.

Пароль полезен только для ролей с атрибутом LOGIN, но задать его можно и для ролей без такого атрибута. PASSWORD можно опустить, если проверка подлинности по паролю не будет использоваться.

```
[ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL
```

password – строка пароля.

При указании пустого значения будет задан пароль NULL, и пользователь не пройдет проверку подлинности по паролю. Пароль NULL можно установить явно, указав PASSWORD NULL.

ENCRYPTED – может быть опущено, поскольку пароль в системных каталогах всегда хранится в зашифрованном виде (данное ключевое слово для обратной совместимости).

Метод шифрования определяется параметром конфигурации **password_encryption**.

Если представленная строка пароля уже зашифрована с применением MD5 или SCRAM, она сохраняется как есть вне зависимости от значения password_encryption (система не может расшифровать переданный зашифрованной пароль, чтобы зашифровать его по другому алгоритму). Это позволяет пересохранять зашифрованные пароли при выгрузке/восстановлении.

VALID UNTIL – устанавливает дату и время, после которого пароль роли перестаёт действовать. Если это предложение отсутствует, срок действия пароля будет неограниченным.

```
VALID UNTIL 'timestamp'
```

VALID UNTIL определяет срок действия только пароля, но не роли как таковой.

При входе пользователя без проверки подлинности по паролю ограничение срока пароля не действует.

IN ROLE – новая роль автоматически становится членом указанных существующих ролей.

Для нового члена будет включён атрибут SET и отключён атрибут ADMIN.

Атрибут SET позволяет выполнить команду SET ROLE, которая меняет идентификатор текущего пользователя в активном сеансе на указанное имя роли.

Если не указан атрибут NOINHERIT, будет включён атрибут INHERIT.

```
IN ROLE role_name [, ...]
IN GROUP role_name [, ...]
```

role_name – имя роли, членом которой создаваемая роль является.

ROLE – одна или нескольких указанных существующих ролей автоматически добавляются как члены новой роли с включённым атрибутом SET.

Фактически это делает новую роль «группой».

Для ролей, указанных в этом предложении с атрибутом INHERIT, будет включён атрибут INHERIT в новом членстве.

Для новых участников атрибут ADMIN будет отключён.

```
ROLE role_name [, ...]
```

ADMIN – работает подобно ROLE, но перечисленные в нём роли добавляются как члены новой роли с включённым атрибутом ADMIN, что даёт им право включать в новою роль другие роли, а также другие роли исключать из неё. Без этого указания обычные пользователи это делать не могут.

```
ADMIN role_name [, ...]
```

DROP ROLE – удаления роли

```
DROP ROLE имя;
```

В составе утилит есть программы `createuser` и `dropuser`, которые являются обёртками для этих команд SQL и вызываются из оболочки ОС:

```
createuser имя
dropuser имя
```

Список существующих можно получить, обратившись к представлению pg_roles:

```
SELECT rolname FROM pg_roles;
```

Чтобы проверить, какие роли могут подключаться к базе данных:

```
SELECT rolname FROM pg_roles WHERE rolcanlogin;
```

Или метакоманда `\du` из `psql`.

SET ROLE — установить идентификатор текущего пользователя

```
SET [ SESSION | LOCAL ] ROLE role_name
SET [ SESSION | LOCAL ] ROLE NONE
RESET ROLE
```

Команда меняет идентификатор текущего пользователя в активном сеансе на `role_name`.

Имя роли может быть записано в виде идентификатора или строковой константы.

После SET ROLE, права доступа для команд SQL проверяются так, как если бы сеанс изначально был установлен с этим именем роли.

Текущий пользователь должен либо непосредственно обладать атрибутом SET для указанного `role_name`, либо опосредованно по цепочке членства в роли с атрибутом SET.

LOCAL – смена идентификатора действует только до конца текущей транзакции, в которой была вызвана SET ROLE, независимо от того, фиксируется она или нет.

SESSION – если команда SET ROLE выполняется внутри транзакции, которая затем прерывается, эффект команды SET ROLE пропадает, когда транзакция откатывается.

Если же окружающая транзакция фиксируется, этот эффект сохраняется до конца сеанса, если его не переопределит другая команда SET ROLE.

SET ROLE NONE устанавливает в качестве идентификатора текущего пользователя идентификатор текущего пользователя сеанса, выдаваемый функцией `session_user`.

RESET ROLE устанавливает в качестве идентификатора текущего пользователя значение, заданное во время подключения параметрами командной строки либо командой ALTER ROLE или ALTER DATABASE.

Если же такое значение не задано, в качестве идентификатора текущего пользователя так же устанавливается идентификатор текущего пользователя сеанса.

С помощью этой команды можно как добавить права, так и ограничить их.

Если роль пользователя сеанса получила членство в роли с WITH INHERIT TRUE, она автоматически получает права такой роли. В этом случае SET ROLE убирает все права, кроме принадлежащих этой роли и наследуемых ей. С другой стороны, если роль получила членство с WITH INHERIT FALSE, по умолчанию она не получает доступ к правам назначенной роли. Однако если членство в роли было выдано с WITH SET TRUE, пользователь сеанса может воспользоваться SET ROLE, чтобы убрать права, назначенные непосредственно пользователю сеанса, и вместо них назначить права, которые имеет указанная роль. Если роль получила членство с WITH INHERIT FALSE, SET FALSE, доступ к правам назначенной роли нельзя получить и с SET ROLE.

GRANT — определить права доступа

Команда GRANT имеет две основные разновидности:

первая назначает права для доступа к объектам баз данных – таблицам, столбцам, представлениям, сторонним таблицам, последовательностям, базам данных, обёрткам сторонних данных, сторонним серверам, функциям, процедурам, процедурным языкам, большим объектам, параметрам конфигурации, схемам, табличным пространствам или типам;

вторая назначает одни роли членами других.

GRANT для объектов баз данных

Эта разновидность команды GRANT даёт одной или нескольким ролям определённые права для доступа к объекту базы данных. Эти права добавляются к списку имеющихся, если роль уже наделена какими-то правами.

Права для доступа к объекту могут быть даны как конкретной роли, так и всем ролям, включая те, что могут быть созданы позже (PUBLIC).

Право, полученною ролью, может делегироваться другим ролям (WITH GRANT OPTION), либо нет. Группе PUBLIC право делегирования дать нельзя.

Нет необходимости явно давать права для доступа к объекту его владельцу, поскольку обычно это пользователь, создавший объект, и по умолчанию он имеет все права.

Однако владелец может лишить себя прав в целях безопасности.

Право удалять объект или изменять его определение произвольным образом не считается назначаемым – оно неотъемлемо связано с владельцем, так что отозвать это право или дать его кому-то другому нельзя.

Возможные права:
SELECT
INSERT
UPDATE
DELETE
TRUNCATE
REFERENCES
TRIGGER
CREATE
CONNECT
TEMPORARY
EXECUTE
USAGE
SET
ALTER SYSTEM

GRANT для ролей

Эта разновидность команды GRANT включает роль в члены одной или нескольких других ролей, а также изменяет параметры членства SET, INHERIT и ADMIN.

```
GRANT role_name [, ...] TO role_specification [, ...]
    [ WITH { ADMIN | INHERIT | SET } { OPTION | TRUE | FALSE } ]
    [ GRANTED BY role_specification ]
```

где *role_specification*:

[GROUP] role_name
PUBLIC
CURRENT_ROLE
CURRENT_USER
SESSION_USER

Чтобы изменить параметры существующего членства, следует выдать членство с обновлёнными значениями параметров.

Каждый из параметров, описанных ниже, может принимать значение TRUE или FALSE.

При изменении существующего членства отсутствие параметра приводит к сохранению текущего значения.

Получивший членство в роли с указанием ADMIN сможет, в свою очередь, включать в члены этой роли, а также исключать из неё другие роли. Без этого указания обычные пользователи не могут это делать.

Параметр INHERIT управляет статусом наследования нового членства.

Если для этого параметра установлено значение TRUE, новый член наследует от выданной роли. Если установлено значение FALSE, новый член не наследует от выданной роли.

Если при создании нового членства в роли этот параметр не указан, по умолчанию используется атрибут наследования добавляемого члена роли.