

PostgreSQL

Разработка серверной части приложений PostgreSQL 16 **(dev-1)**



Управление доступом Обзор

PROFESSIONAL
Posgres



Роли и атрибуты

Подключение к серверу

Парольная аутентификация

Привилегии и управление ими

Категории ролей

Групповые и предопределенные роли

Привилегии по умолчанию

Привилегии и подпрограммы

Роль может быть пользователем СУБД

не связана с пользователем ОС

Роли можно включать друг в друга

удобно при настройке доступа

Свойства роли определяются атрибутами

LOGIN

возможность подключения

SUPERUSER

суперпользователь

CREATEDB

возможность создавать базы данных

CREATEROLE

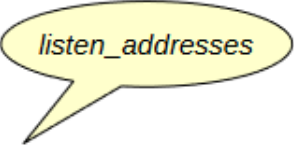
возможность создавать роли

и другие

Подключение к серверу

1. Строки pg_hba.conf просматриваются сверху вниз
2. Выбирается первая запись, соответствующая параметрам подключения (тип, база, пользователь и адрес)

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	local	all	postgres		peer
	local	all	all		peer
	host	all	all	127.0.0.1/32	scram-sha-256
	host	all	all	:::1/128	scram-sha-256
...					
	local — сокет		all — любая роль		
	host — TCP/IP		имя роли		
		all — любая БД			
		имя БД			
				all — любой IP	
				IP/маска	
				доменное имя	



3. Выполняется аутентификация указанным методом
4. Удачно — доступ разрешается, иначе — запрещается
(если не подошла ни одна запись — доступ запрещается)

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	local	all	postgres		peer
	local	all	all		peer
	host	all	all	127.0.0.1/32	scram-sha-256
	host	all	all	:::1/128	scram-sha-256

trust — разрешить

reject — отказать

scram-sha-256 и md5 — запросить пароль

peer — спросить ОС

На сервере

- пароль устанавливается при создании роли или позже
- пользователю без пароля будет отказано в доступе
- пароль хранится в системном каталоге `pg_authid`

Ввод пароля на клиенте

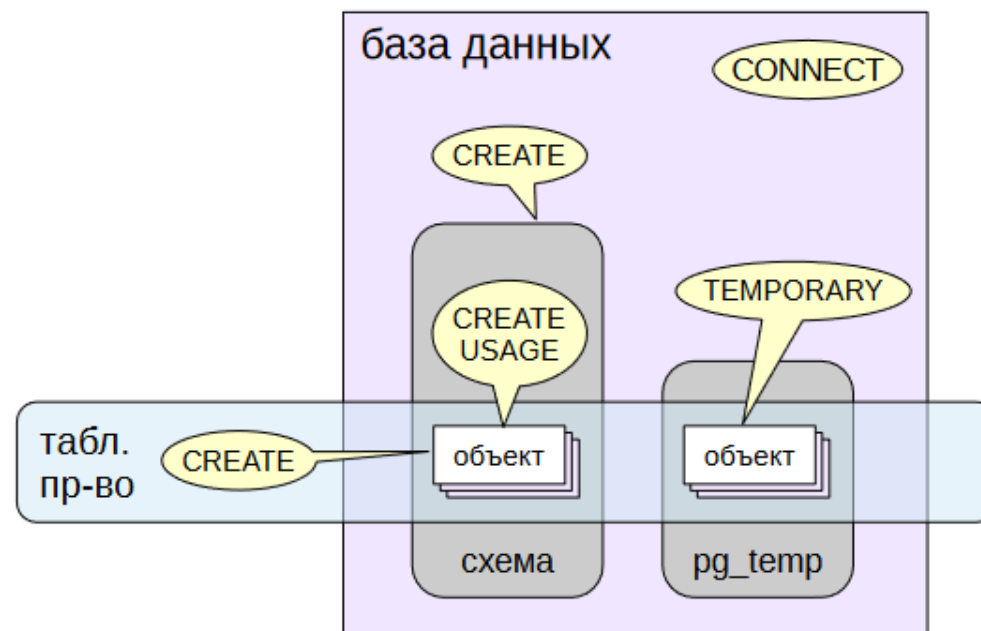
- вручную
- из переменной окружения `PGPASSWORD`
- из файла `~/.pgpass` (строки в формате *узел:порт:база:роль:пароль*)

Привилегии определяют права доступа ролей к объектам
Таблицы и представления

SELECT	чтение данных	}	можно на уровне столбцов
INSERT	вставка строк		
UPDATE	изменение строк		
REFERENCES	внешний ключ (для таблиц)		
DELETE	удаление строк		
TRUNCATE	опустошение (для таблиц)		
TRIGGER	создание триггеров		

Привилегии

Табличные пространства,
базы данных, схемы



Последовательности

SELECT
UPDATE
USAGE

currval
currval

nextval
nextval

setval

Суперпользователи

полный доступ ко всем объектам — проверки не выполняются

Владельцы

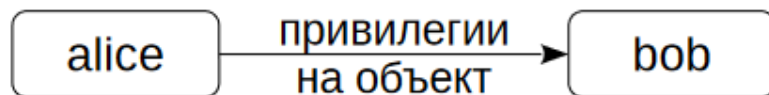
изначально все привилегии для объекта (можно отозвать)
действия со своими объектами, не регламентируемые привилегиями,
например: удаление, выдача и отзыв привилегий и т. п.

Остальные роли

доступ в рамках выданных привилегий

Выдача привилегий

alice: GRANT привилегии ON объект TO *bob*;

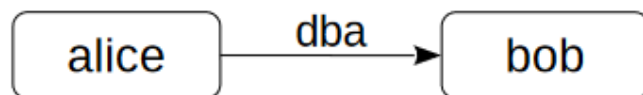


Отзыв привилегий

alice: REVOKE привилегии ON объект FROM *bob*;

Включение в роль

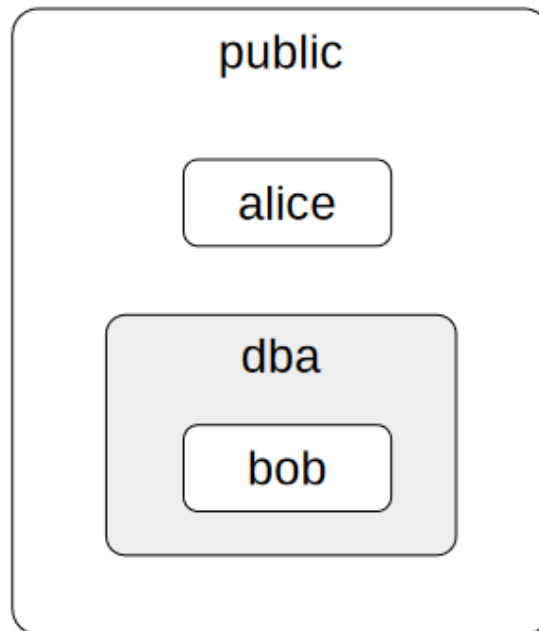
alice: GRANT *dba* TO *bob*;



псевдороль `public` неявно включает в себя все остальные роли

Исключение из роли

alice: REVOKE *dba* FROM *bob*;



Предопределенные роли

pg_read_all_settings

чтение всех параметров сервера

pg_read_all_stats

доступ к статистике

pg_stat_scan_tables

мониторинг и блокировки таблиц

pg_read_all_data

чтение данных из всех таблиц

pg_write_all_data

изменение данных во всех таблицах

pg_read_server_files

чтение файлов на сервере

pg_write_server_files

запись в файлы на сервере

pg_execute_server_programs

выполнение программ на сервере

...

pg_monitor

Единственная привилегия для функций и процедур

EXECUTE

выполнение

Характеристики безопасности

SECURITY INVOKER

выполняется с правами вызывающего
(по умолчанию)

SECURITY DEFINER

выполняется с правами владельца

Привилегии псевдороди public

подключение к любой базе данных

доступ к системному каталогу

выполнение любых подпрограмм

привилегии выдаются автоматически для каждого нового объекта

Настраиваемые привилегии по умолчанию

возможность дополнительно выдать или отозвать привилегии

для вновь создаваемого объекта

Роли, атрибуты и привилегии — гибкий механизм,
позволяющий по-разному организовать работу

можно легко разрешить все всем

можно строго разграничить доступ, если это необходимо

При создании новых ролей надо позаботиться
о возможности их подключения к серверу



1. Создайте две роли (пароль должен совпадать с именем):
 - employee — сотрудник магазина,
 - buyer — покупатель.

Убедитесь, что созданные роли могут подключиться к БД.
2. Отзовите у роли public права выполнения всех функций и подключения к БД.
3. Разграничьте доступ таким образом, чтобы:
 - сотрудник мог только заказывать книги, а также добавлять авторов и книги,
 - покупатель мог только приобретать книги.

Проверьте выполненные настройки в приложении.

Настройте привилегии таким образом, чтобы одни пользователи имели полный доступ к таблицам, а другие могли только запрашивать, но не изменять информацию.

1. Создайте новую базу данных и двух пользователей: `writer` и `reader`.
2. Отзовите у роли `public` все привилегии на схему `public`, выдайте роли `writer` обе привилегии, а роли `reader` — только `usage`.
3. Настройте привилегии по умолчанию так, чтобы роль `reader` получала доступ на чтение к таблицам, принадлежащим `writer` в схеме `public`.
4. Создайте пользователя `w1`, включив его в роль `writer`, и пользователя `r1`, включив его в `reader`.
5. Под ролью `writer` создайте таблицу.
6. Убедитесь, что `r1` имеет доступ к таблице только на чтение, а `w1` имеет к ней полный доступ, включая удаление.