



PostgreSQL для администраторов баз данных и разработчиков



Проверить, идет ли запись

Меня хорошо видно & слышно?

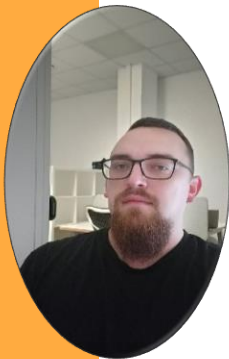


Ставим "+", если все хорошо
"-", если есть проблемы



Тема вебинара

DDL: создание, изменение и удаление объектов в PostgreSQL



Игорь Тоескин

Ведущий разработчик СУБД

Специалист в области разработки и проектировании витрин данных в PostgreSQL, а также в области разработки хранимых процедур в таких СУБД как PostgreSQL и Oracle

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в учебной группе
#канал группы



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Маршрут вебинара



Основные объекты базы данных

CREATE

ALTER

DROP

Цели вебинара

К концу занятия вы сможете

Использовать команды DDL для создания,
изменения и удаления объектов БД

Смысл

Зачем вам это уметь

Для разработки и администрирования баз
данных

Основные объекты БД

Основные объекты БД

- Кластер
 - База данных
 - Табличное пространство
 - Роль (пользователь)
- База данных
 - Схема
 - Таблица
 - Индекс
 - Материализованное представление
 - Последовательность
 - Функция, процедура
 - ...
 - Представление •
 - ...

Создание базы данных

```
CREATE DATABASE имя
    [ WITH ] [ OWNER [=] имя_пользователя ]
    [ TEMPLATE [=] шаблон ]
    [ ENCODING [=] кодировка ]
    [ LOCALE [=] локаль ]
    [ LC_COLLATE [=] категория_сортировки ]
    [ LC_STYPE [=] категория_типов_символов ]
    [ TABLESPACE [=] табличное пространство ]
    [ ALLOW_CONNECTIONS [=] разрешение подключения ]
    [ CONNECTION LIMIT [=] предел подключений ]
    [ IS_TEMPLATE [=] это_шаблон ]
```

Изменение атрибутов базы данных

```
ALTER DATABASE имя RENAME TO новое_имя
```

```
ALTER DATABASE имя OWNER TO { новый_владелец | CURRENT_USER |  
SESSION_USER }
```

```
ALTER DATABASE имя SET TABLESPACE новое_табл_пространство
```

```
ALTER DATABASE имя SET параметр_конфигурации { TO | = } {  
значение | DEFAULT }
```

```
ALTER DATABASE имя SET параметр_конфигурации FROM CURRENT
```

```
ALTER DATABASE имя RESET параметр_конфигурации
```

```
ALTER DATABASE имя RESET ALL
```

LIVE

Табличные пространства

```
CREATE TABLESPACE табл_пространство  
[ OWNER { новый_владелец | CURRENT_USER | SESSION_USER } ]  
LOCATION 'каталог'  
[ WITH ( параметр_табличного_пространства = значение [, ... ] ) ]
```

LIVE

Роли/Пользователи

```
CREATE ROLE имя [ [ WITH ] параметр [ ... ] ]
```

Здесь параметр:

```
    SUPERUSER | NOSUPERUSER  
| CREATEDB | NOCREATEDB  
| CREATEROLE | NOCREATEROLE  
| INHERIT | NOINHERIT  
| LOGIN | NOLOGIN  
| REPLICATION | NOREPLICATION  
| BYPASSRLS | NOBYPASSRLS  
| CONNECTION LIMIT предел_подключений  
| [ ENCRYPTED ] PASSWORD 'пароль' | PASSWORD NULL  
| VALID UNTIL 'дата_время'  
| IN ROLE имя_роли [, ...]  
| IN GROUP имя_роли [, ...]  
| ROLE имя_роли [, ...]  
| ADMIN имя_роли [, ...]  
| USER имя_роли [, ...]  
| SYSID uid
```

Пользователи

```
CREATE ROLE имя [ [ WITH ] параметр [ ... ] ]
```

Здесь параметр:

```
    SUPERUSER | NOSUPERUSER  
| CREATEDB | NOCREATEDB  
| CREATEROLE | NOCREATEROLE  
| INHERIT | NOINHERIT  
| LOGIN | NOLOGIN  
| REPLICATION | NOREPLICATION  
| BYPASSRLS | NOBYPASSRLS  
| CONNECTION LIMIT предел_подключений  
| [ ENCRYPTED ] PASSWORD 'пароль' | PASSWORD NULL  
| VALID UNTIL 'дата_время'  
| IN ROLE имя_роли [, ...]  
| IN GROUP имя_роли [, ...]  
| ROLE имя_роли [, ...]  
| ADMIN имя_роли [, ...]  
| USER имя_роли [, ...]  
| SYSID uid
```


LIVE

Схемы

```
CREATE SCHEMA имя_схемы [ AUTHORIZATION указание_роли ] [  
элемент_схемы [ ... ] ]  
CREATE SCHEMA AUTHORIZATION указание_роли [ элемент_схемы [ ... ] ]  
CREATE SCHEMA IF NOT EXISTS имя_схемы [ AUTHORIZATION указание_роли ]  
CREATE SCHEMA IF NOT EXISTS AUTHORIZATION указание_роли
```

Здесь указание_роли:

```
    имя_пользователя  
| CURRENT_USER  
| SESSION_USER
```

LIVE

Расширения

```
CREATE EXTENSION [ IF NOT EXISTS ] имя_расширения  
    [ WITH ] [ SCHEMA имя_схемы ]  
        [ VERSION версия ]  
        [ CASCADE ]
```

<https://postgrespro.ru/docs/postgresql/15/contrib>

LIVE

Домены

```
CREATE DOMAIN имя [ AS ] тип_данных  
    [ COLLATE правило_сортировки ]  
    [ DEFAULT выражение ]  
    [ ограничение [ ... ] ]
```

Здесь ограничение:

```
[ CONSTRAINT имя_ограничения ]  
{ NOT NULL | NULL | CHECK (выражение) }
```

Таблицы

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ] TABLE [
IF NOT EXISTS ] имя_таблицы ( [
    { имя_столбца тип_данных [ COLLATE правило_сортировки ] [
ограничение_столбца [ ... ] ]
    | ограничение_таблицы
    | LIKE исходная_таблица [ вариант_копирования ... ] }
[, ... ]
] )
[ INHERITS ( таблица_родитель [, ... ] ) ]
[ PARTITION BY { RANGE | LIST | HASH } ( { имя_столбца | ( выражение )
} [ COLLATE правило_сортировки ] [ класс_операторов ] [, ... ] ) ]
[ USING метод ]
[ WITH ( параметр_хранения [= значение] [, ... ] ) | WITHOUT OIDS ]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE табл_пространство ]
```

Типизированные таблицы

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ] TABLE [
IF NOT EXISTS ] имя_таблицы
    OF имя_типа [ (
    { имя_столбца [ WITH OPTIONS ] [ ограничение_столбца [ ... ] ]
    | ограничение_таблицы }
    [, ... ]
) ]
[ PARTITION BY { RANGE | LIST | HASH } ( { имя_столбца | ( выражение )
} [ COLLATE правило_сортировки ] [ класс_операторов ] [, ... ] ) ]
[ USING метод ]
[ WITH ( параметр_хранения [= значение] [, ... ] ) | WITHOUT OIDS ]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE табл_пространство ]
```


Секционированные таблицы

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ] TABLE [
IF NOT EXISTS ] имя_таблицы
    PARTITION OF таблица_родитель [ (
    { имя_столбца [ WITH OPTIONS ] [ ограничение_столбца [ ... ] ]
    | ограничение_таблицы }
    [, ... ]
) ] { FOR VALUES указание_границ_секции | DEFAULT }
[ PARTITION BY { RANGE | LIST | HASH } ( { имя_столбца | ( выражение )
} [ COLLATE правило_сортировки ] [ класс_операторов ] [, ... ] ) ]
[ USING метод ]
[ WITH ( параметр_хранения [= значение] [, ... ] ) | WITHOUT OIDS ]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE табл_пространство ]
```

Таблицы как результат запроса

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ]  
TABLE [ IF NOT EXISTS ] имя_таблицы  
    [ (имя_столбца [, ...] ) ]  
    [ USING метод ]  
    [ WITH ( параметр_хранения [= значение] [, ...] ) | WITHOUT  
OIDS ]  
    [ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]  
    [ TABLESPACE табл_пространство ]  
AS запрос  
    [ WITH [ NO ] DATA ]
```

LIVE

Представления

```
CREATE [ OR REPLACE ] [ TEMP | TEMPORARY ] [ RECURSIVE ] VIEW имя  
[ ( имя_столбца [, ...] ) ]  
    [ WITH ( имя_параметра_представления [=  
значение_параметра_представления] [, ... ] ) ]  
    AS запрос  
    [ WITH [ CASCADED | LOCAL ] CHECK OPTION ]
```

Материализованные представления

```
CREATE MATERIALIZED VIEW [ IF NOT EXISTS ] имя_таблицы  
    [ (имя_столбца [, ...] ) ]  
    [ USING метод ]  
    [ WITH ( параметр_хранения [= значение] [, ... ] ) ]  
    [ TABLESPACE табл_пространство ]  
AS запрос  
    [ WITH [ NO ] DATA ]
```

LIVE

Индексы

```
CREATE [ UNIQUE ] INDEX [ CONCURRENTLY ] [ [ IF NOT EXISTS ] имя  
] ON [ ONLY ] имя_таблицы [ USING метод ]  
    ( { имя_столбца | ( выражение ) } [ COLLATE  
правило_сортировки ] [ класс_операторов [ ( параметр_класса_оп =  
значение [, ... ] ) ] ] [ ASC | DESC ] [ NULLS { FIRST | LAST } ]  
[, ...] )  
    [ INCLUDE ( имя_столбца [, ...] ) ]  
    [ WITH ( параметр_хранения [= значение] [, ... ] ) ]  
    [ TABLESPACE табл_пространство ]  
    [ WHERE предикат ]
```

LIVE

Последовательности

```
CREATE [ TEMPORARY | TEMP ] SEQUENCE [ IF NOT EXISTS ] имя
    [ AS тип_данных ]
    [ INCREMENT [ BY ] шаг ]
    [ MINVALUE мин_значение | NO MINVALUE ] [ MAXVALUE
макс_значение | NO MAXVALUE ]
    [ START [ WITH ] начало ] [ CACHE кеш ] [ [ NO ] CYCLE ]
    [ OWNED BY { имя_таблицы.имя_столбца | NONE } ]
```

LIVE

Функции

```
CREATE [ OR REPLACE ] FUNCTION
    имя ( [ [ режим_аргумента ] [ имя_аргумента ] тип_аргумента [ { DEFAULT | = }
выражение_по_умолчанию ] [, ...] ] )
    [ RETURNS тип_результата
      | RETURNS TABLE ( имя_столбца тип_столбца [, ...] ) ]
{ LANGUAGE имя_языка
  | TRANSFORM { FOR TYPE имя_типа } [, ... ]
  | WINDOW
  | { IMMUTABLE | STABLE | VOLATILE }
  | [ NOT ] LEAKPROOF
  | { CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT }
  | { [ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER }
  | PARALLEL { UNSAFE | RESTRICTED | SAFE }
  | COST стоимость_выполнения
  | ROWS строк_в_результате
  | SUPPORT вспомогательная_функция
  | SET параметр_конфигурации { TO значение | = значение | FROM CURRENT }
  | AS 'определение'
  | AS 'объектный_файл', 'объектный_символ'
} ...
```

Процедуры

```
CREATE [ OR REPLACE ] PROCEDURE
    имя ( [ [ режим_аргумента ] [ имя_аргумента ] тип_аргумента [
{ DEFAULT | = } выражение_по_умолчанию ] [, ...] ] )
    { LANGUAGE имя_языка
      | TRANSFORM { FOR TYPE имя_типа } [, ... ]
      | [ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY
DEFINER
      | SET параметр_конфигурации { TO значение | = значение | FROM
CURRENT }
      | AS 'определение'
      | AS 'объектный_файл', 'объектный_символ'
    } ...
```

LIVE

Изменение объектов

```
ALTER [вид объекта] [название объекта] [что меняем]
```

Изменение таблицы

```
ALTER TABLE [ IF EXISTS ] [ ONLY ] имя [ * ]  
    действие [, ... ]  
ALTER TABLE [ IF EXISTS ] [ ONLY ] имя [ * ]  
    RENAME [ COLUMN ] имя_столбца TO новое_имя_столбца  
ALTER TABLE [ IF EXISTS ] [ ONLY ] имя [ * ]  
    RENAME CONSTRAINT имя_ограничения TO имя_нового_ограничения  
ALTER TABLE [ IF EXISTS ] имя  
    RENAME TO новое_имя  
ALTER TABLE [ IF EXISTS ] имя  
    SET SCHEMA новая_схема  
ALTER TABLE ALL IN TABLESPACE имя [ OWNED BY имя_роли [, ... ] ]  
    SET TABLESPACE новое_табл_пространство [ NOWAIT ]  
ALTER TABLE [ IF EXISTS ] имя  
    ATTACH PARTITION имя_секции { FOR VALUES указание_границ_секции | DEFAULT }  
ALTER TABLE [ IF EXISTS ] имя  
    DETACH PARTITION имя_секции
```

Изменение таблицы

```
ADD [ COLUMN ] [ IF NOT EXISTS ] имя_столбца тип_данных [ COLLATE правило_сортировки ] [
ограничение_столбца [ ... ] ]
DROP [ COLUMN ] [ IF EXISTS ] имя_столбца [ RESTRICT | CASCADE ]
ALTER [ COLUMN ] имя_столбца [ SET DATA ] TYPE тип_данных [ COLLATE правило_сортировки ] [ USING
выражение ]
ALTER [ COLUMN ] имя_столбца SET DEFAULT выражение
ALTER [ COLUMN ] имя_столбца DROP DEFAULT
ALTER [ COLUMN ] имя_столбца { SET | DROP } NOT NULL
ALTER [ COLUMN ] имя_столбца DROP EXPRESSION [ IF EXISTS ]
ALTER [ COLUMN ] имя_столбца ADD GENERATED { ALWAYS | BY DEFAULT } AS IDENTITY [ (
параметры_последовательности ) ]
ALTER [ COLUMN ] имя_столбца { SET GENERATED { ALWAYS | BY DEFAULT } | SET
параметр_последовательности | RESTART [ [ WITH ] перезапуск ] } [...]
ALTER [ COLUMN ] имя_столбца DROP IDENTITY [ IF EXISTS ]
ALTER [ COLUMN ] имя_столбца SET STATISTICS integer
ALTER [ COLUMN ] имя_столбца SET ( атрибут = значение [, ... ] )
ALTER [ COLUMN ] имя_столбца RESET ( атрибут [, ... ] )
ALTER [ COLUMN ] имя_столбца SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN }
ADD ограничение_таблицы [ NOT VALID ]
ADD ограничение_таблицы_по_индексу
ALTER CONSTRAINT имя_ограничения [ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY
IMMEDIATE ]
```


LIVE

Удаление объектов

```
DROP [вид объекта] [название объекта] {CASCADE}
```

LIVE

Вопросы?



Ставим "+",
если вопросы есть



Ставим "-",
если вопросов нет

Рефлексия

Вопросы для проверки

1. Кто что запомнил?
2. Хватило ли практики?
3. Для каких объектов нужно использовать ALTER, а для каких – CREATE OR REPLACE?

Рефлексия



С какими впечатлениями уходите с вебинара?



Как будете применять на практике то, что узнали на вебинаре?

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Тоескин Игорь

Ведущий разработчик СУБД

Специалист в области разработки и проектировании витрин данных в PostgreSQL, а также в области разработки хранимых процедур в таких СУБД как PostgreSQL и Oracle