




ОНЛАЙН-ОБРАЗОВАНИЕ

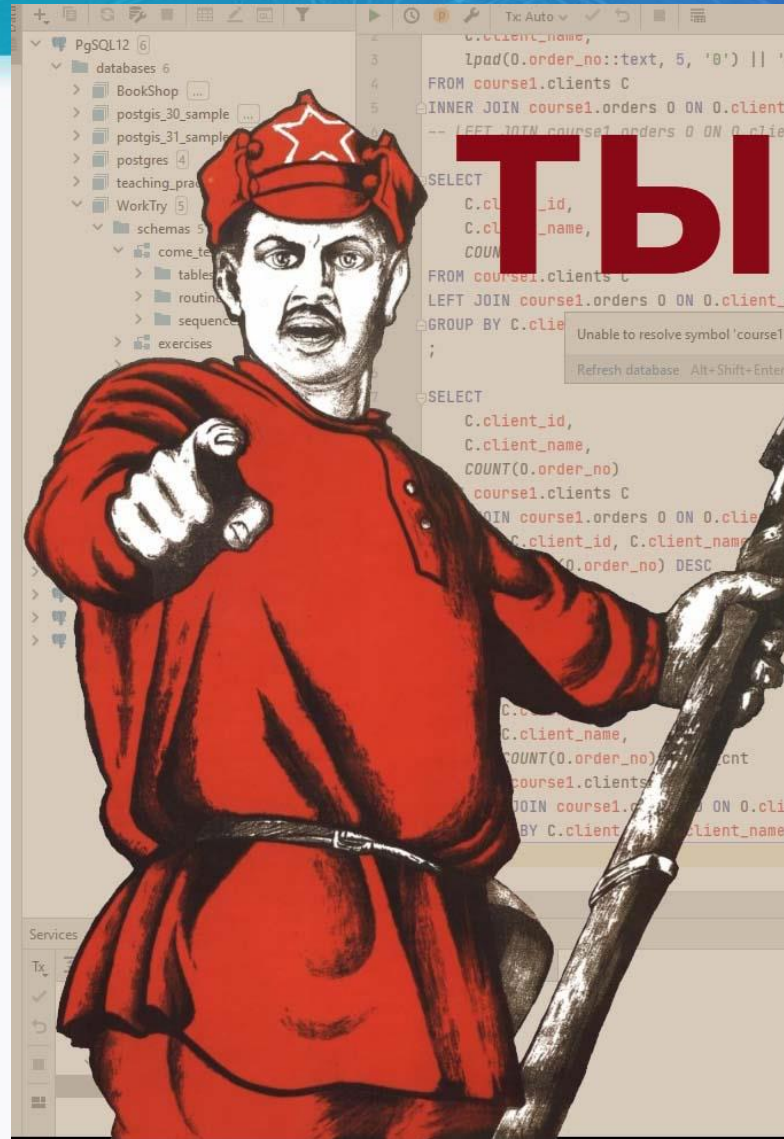
Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы
заодно проверяем, включена ли запись занятия

...а включена ли запись?



ВКЛЮЧИЛ ЗАПИСЬ?

DDL: Создание, изменение и удаление объектов в PostgreSQL



АНТОН ЗОЛотов

telegram @AVZolotov

Преподаватель



Антон Золотов

Программист, разработчик баз данных.
Общий стаж разработки программного обеспечения
более 35 лет, разработкой БД занимаюсь около 20 лет.

Предметные области:

- ERP;
- складская логистика;
- системы анализа медиаданных;
- системы поддержки принятия решений.

- сертифицированный тренер PostgreSQL
- преподаватель курса "PostgreSQL" в OTUS

Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом

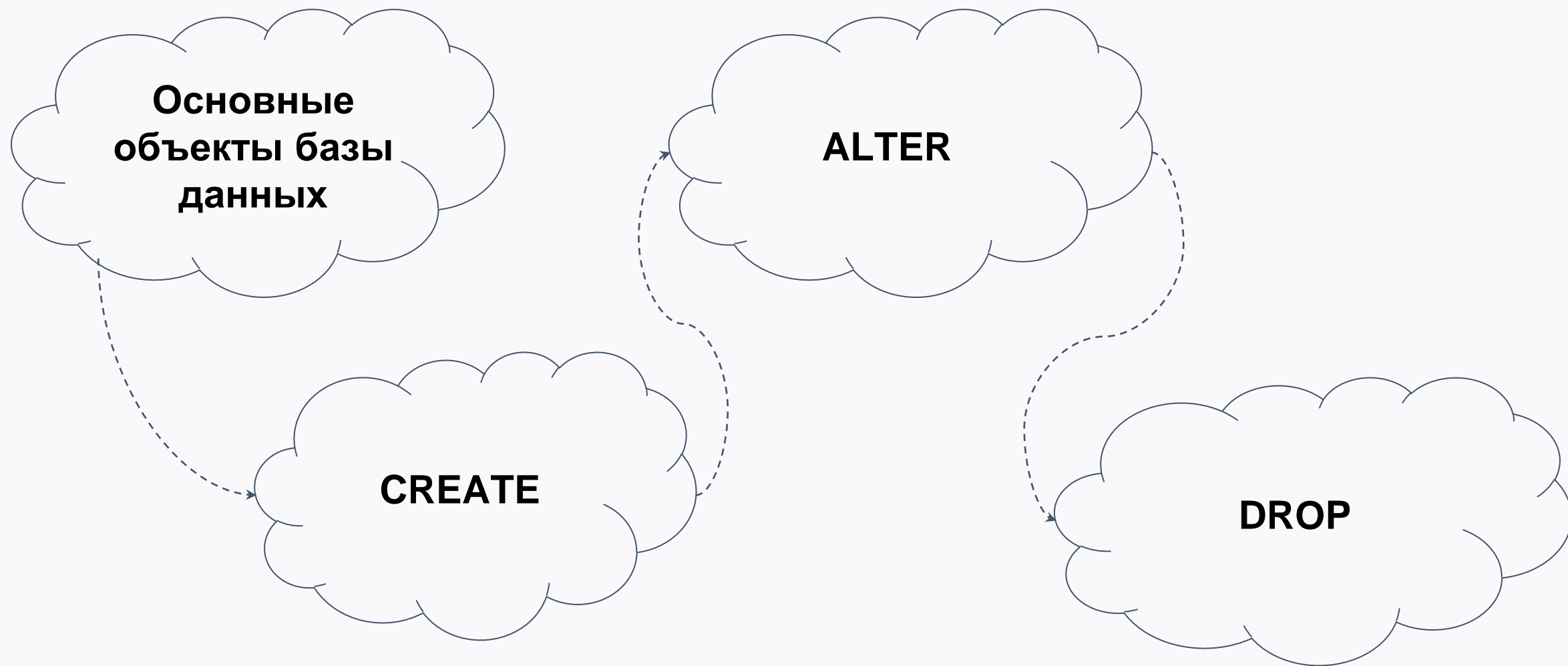


Off-topic обсуждаем вSlack



Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара



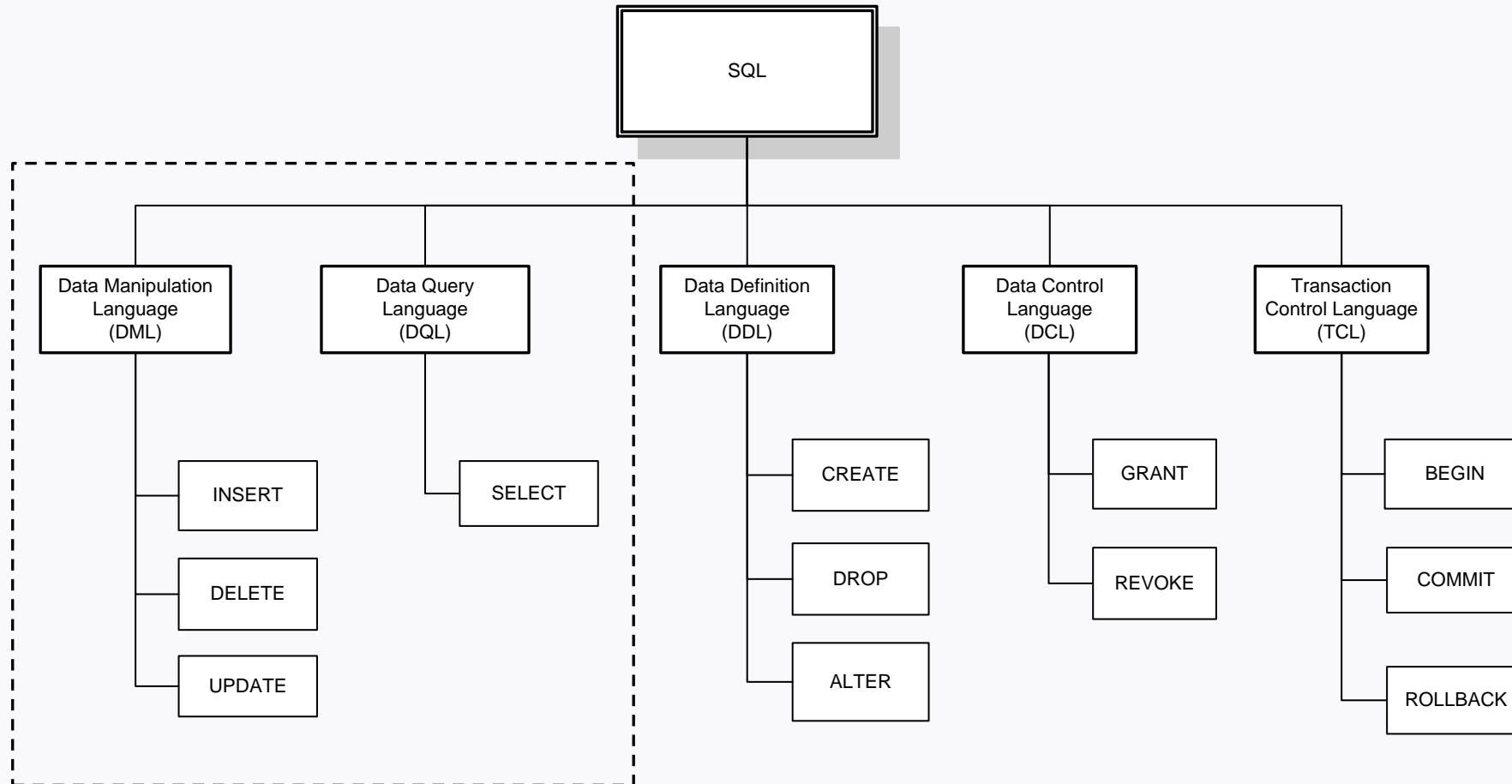
Цели вебинара | После занятия вы сможете

Использовать команды DDL для создания, изменения и удаления объектов БД

Смысл | Зачем вам это уметь

Для разработки и администрирования баз данных

Структура языка SQL (Structured Query Language) и ОСНОВНЫЕ КОМАНДЫ



Основные объекты

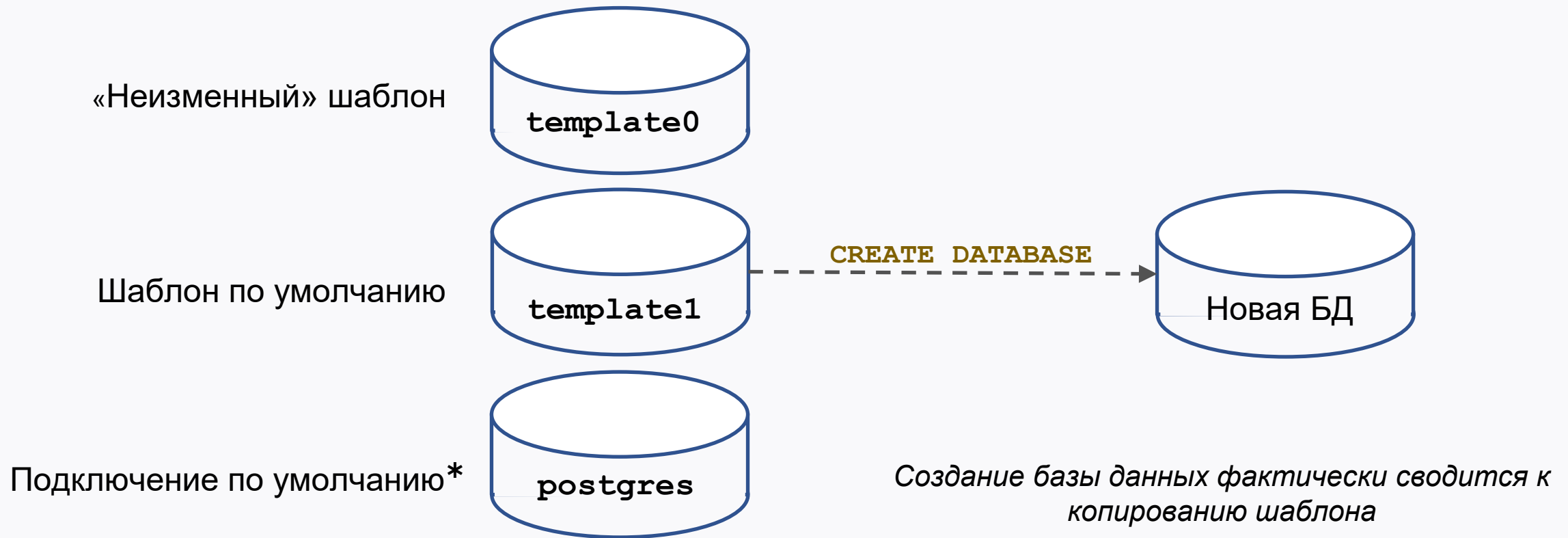
Кластер	<ul style="list-style-type: none">• База данных• Табличное пространство• Роль (пользователь)
База данных	<ul style="list-style-type: none">• Схема<ul style="list-style-type: none">• Таблица• Индекс• Материализованное представление• Последовательность• Функция, процедура• ...• Представление• ...

Создание базы данных

CREATE DATABASE имя	
[[WITH] [OWNER [=] имя_пользователя]	Если не указано - пользователь, выполняющий команду
[TEMPLATE [=] шаблон]	По умолчанию - template1
[ENCODING [=] кодировка]	По умолчанию - кодировка шаблона
[LOCALE [=] локаль]	Краткий вариант определения значения для двух параметров LC_COLLATE и LC_CTYPE. Исключает отдельное указание любого из этих параметров
[LC_COLLATE [=] категория_сортировки]	Порядок сортировки. По умолчанию используется LC_COLLATE шаблона
[LC_CTYPE [=] категория_типов_символов]	Классификация символов (принадлежность символов категориям, например: строчные, заглавные, цифры и т. п.). По умолчанию используется классификация символов, установленная в шаблоне.
[TABLESPACE [=] табл_пространство]	Табличное пространство, которое будет использоваться по умолчанию для объектов, создаваемых в этой базе. По умолчанию - табличное пространство шаблона.
[ALLOW_CONNECTIONS [=] разр_подключения]	
[CONNECTION LIMIT [=] предел_подключений]	
[IS_TEMPLATE [=] это_шаблон]]	

Кластер баз данных

При установке кластера PostgreSQL создаются 3 базы данных:



* Непосредственно после создания кластера, пока не созданы пользователи

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
```


Табличные пространства

Табличные пространства определяют физическое размещение данных в файловой системе

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

Имя табличного пространства (табл_пространство) не должно начинаться с pg_

*Каталог, который будет использован для этого табличного пространства. Этот каталог должен уже существовать - команда **CREATE TABLESPACE** не создает каталог*

Параметр табличного пространства – в настоящее время поддерживаются параметры, позволяющие переопределить оценки стоимости чтения данных в этом табличном пространстве:

```
seq_page_cost,
random_page_cost,
effective_io_concurrency,
maintenance_io_concurrency.
```

Табличные пространства

Объекты одной базы данных могут быть размещены в нескольких табличных пространствах
В одном табличном пространстве могут быть размещены объекты нескольких баз данных

При установке кластера создаются два табличных пространства:

- ✓ **pg_default** – используется по умолчанию при создании объектов БД (если не было задано другое ТС по умолчанию). Располагается в каталоге PGDATA/base, внутри этого каталога объекты размещаются в подкаталогах баз данных
- ✓ **pg_global** – в этом ТС размещаются объекты, общие для всего кластера. Располагается в каталоге PGDATA/global

Нельзя удалить табличное пространство до тех пор, пока в нем существует хотя бы один объект (выражение **CASCADE** не определено для команды **DROP TABLESPACE**)

Схемы

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

имя_схемы не должно начинаться с pg_

Схема – пространство имен для объектов, разнесение объектов по различным схемам позволяет создавать более гибкие системы безопасности, разделять объекты на логические группы, использовать одинаковые имена объектов в различных схемах

Схема принадлежит конкретной базе данных

При создании базы данных создаются схемы:

- public
- pg_catalog
- information_schema

Временные таблицы размещаются в схемах pg_temp_*

<https://postgrespro.ru/docs/postgresql/13/ddl-schemas>

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 ]
```

<https://postgrespro.ru/docs/postgresql/13/sql-createtable>

Таблицы

Создание таблицы из результатов 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 ]
```

<https://postgrespro.ru/docs/postgresql/13/sql-createtableas>

Временные таблицы

Временные таблицы автоматически удаляются в конце сеанса или могут удаляться в конце текущей транзакции

Временные таблицы создаются в специальной схеме, так что при создании таких таблиц имя схемы задать нельзя

Для временных таблиц можно задать параметр ON COMMIT.
Возможные значения:

PRESERVE ROWS	никакие действия по завершении транзакции не выполняются (по умолчанию)
DELETE ROWS	при фиксации или отмене транзакции таблица очищается
DROP	при фиксации или отмене транзакции таблица удаляется

Перенос таблиц

Изменение схемы: перемещение данных не производится, изменения затрагивают только системные каталоги.

```
ALTER TABLE имя_таблицы SET SCHEMA имя_схемы;
```

Изменение табличного пространства: производится физическое перемещение данных. Это требует исключительной (EXCLUSIVE) блокировки таблицы – «дорогая» операция!

```
ALTER TABLE имя_таблицы SET tablespace имя_табличного пространства;
```

Секционированные таблицы

Секционирование таблиц – разделение таблиц на отдельные части с отдельными параметрами физического хранения

Виды секционирования

- по диапазонам;
- по списку;
- по хэшу.

Представления. Материализованные 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  
[ (имя_столбца [, ...] ) ]  
[ USING метод ]  
[ WITH ( параметр_хранения [= значение] [, ... ] ) ]  
[ TABLESPACE табл_пространство ]  
AS запрос  
[ WITH [ NO ] DATA ]
```

Заданный *запрос* выполняется и наполняет представление в момент вызова команды (если только не указано **WITH NO DATA**).

Команда CREATE MATERIALIZED VIEW подобна CREATE TABLE AS, за исключением того, что она «запоминает» запрос, порождающий представление, так что это представление можно позже обновить по требованию.

```
REFRESH MATERIALIZED VIEW [ CONCURRENTLY ] имя  
[ WITH [ NO ] DATA ]
```

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 ]
```

Методы индексов:

- В-дерево (по умолчанию) – для типа данных индексируемого столбца должны быть определены операции <, >, = <= и >=
- хеш – работают только с простыми условиями равенства
- GiST, SP-GiST – инфраструктуры, позволяющая реализовать различные стратегии индексирования (например GiST - для двумерных типов геометрических данных)
- GIN – определен для т.н. GIN-операторов
- BRIN – (Block Range Indexes), индексы, хранящие обобщённые сведения о значениях, находящихся в физически последовательно расположенных блоках таблицы.

Роли (пользователи)

Роль/пользователь – объект кластера, а не конкретной БД

Роль/пользователь кластера – это не пользователь ОС (но некоторые приложения используют имя пользователя ОС как роль кластера по умолчанию)

Роль может быть членом одной или нескольких других ролей.

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

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


Роли (пользователи)

Атрибут	Описание	Значение по умолчанию
SUPERUSER	Роль будет суперпользователем	NOSUPERUSER
CREATEDB	Роль сможет создавать БД	NOCREATEDB
CREATEROLE	Роль может создавать другие роли	NOCREATEROLE
INHERIT	Роль будет наследовать права ролей, членом которых она является	INHERIT
LOGIN	Роли разрешается вход на сервер	NOLOGIN для CREATE ROLE LOGIN для CREATE USER
REPLICATION	Роль сможет подключаться к серверу в режиме репликации	NOREPLICATION
BYPASSRLS	Для роли будут игнорироваться все политики защиты строк (RLS)	NOBYPASSRLS
CONNECTION LIMIT	Сколько параллельных подключений может установить роль	-1 (без ограничений)
[ENCRYPTED] PASSWORD 'пароль'	Пароль для роли	NULL
VALID UNTIL 'дата_время'	Дата и время окончания действия роли	
IN ROLE имя_роли [, ...]	Одна или несколько ролей, в которые будет включена новая роль	
IN GROUP имя_роли [, ...]	Устаревший синоним IN ROLE	
ROLE имя_роли [, ...]	Одна или несколько ролей, которые будут включены в создаваемую роль	
ADMIN имя_роли [, ...]	Аналогично ROLE, но с атрибутом WITH ADMIN OPTION	
USER имя_роли [, ...]	Устаревший синоним ROLE	

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 } ]
```

Команда создает специальный объект – таблицу, содержащую одну строку

Функции для работы с последовательностями:

<code>nextval (regclass)</code>	Продвигает объект к следующему значению и возвращает это значение
<code>setval (regclass, bigint [,boolean])</code>	Задаёт текущее значение и может устанавливать флаг <code>is_called</code> (при <code>true</code> последовательность сначала продвинется к следующему значению)
<code>currval (regclass)</code>	Возвращает значение, выданное при последнем вызове <code>nextval</code> для данной последовательности
<code>lastval ()</code>	Возвращает значение, выданное при последнем вызове <code>nextval</code>

Изменение структуры таблиц

```
ALTER TABLE имя_таблицы ADD COLUMN имя_столбца [options]
```

```
ALTER TABLE имя_таблицы DROP COLUMN имя_столбца
```

```
ALTER TABLE имя_таблицы ALTER COLUMN имя_столбца [options]
```


Удаление объектов

DROP *тип_объекта* [IF EXISTS] *имя_объекта* [CASCADE]

The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a web-like pattern that suggests connectivity or a digital network. The text "Вопросы?" is centered in the middle of the image, rendered in a bold, white, sans-serif font.


Вопросы?

The image features a high-angle, aerial view of a dense urban skyline, likely New York City, with numerous skyscrapers and buildings. The entire image is overlaid with a semi-transparent blue layer. A network of thin, light blue lines connects various points across the blue area, creating a digital or technological aesthetic. The Russian word "Порефлексируем" is centered in the middle of the image in a large, white, sans-serif font.

Порефлексируем

Вопросы?

- Кто что запомнил?)
- Хватило ли практики?



Заполните, пожалуйста,
опрос о занятии по ссылке в чате
<https://otus.ru/polls/25723/>

The background of the entire slide is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue overlay covers the entire image. Overlaid on this blue background is a network of thin, light blue lines connecting various points, creating a geometric, web-like pattern. The text is centered in the middle of the slide.

Спасибо за внимание!
Приходите на следующие вебинары

Золотов Антон