

# Организация данных Логическая структура



## **Авторские права**

© Postgres Professional, 2017 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](mailto:edu@postgrespro.ru)

## **Отказ от ответственности**

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

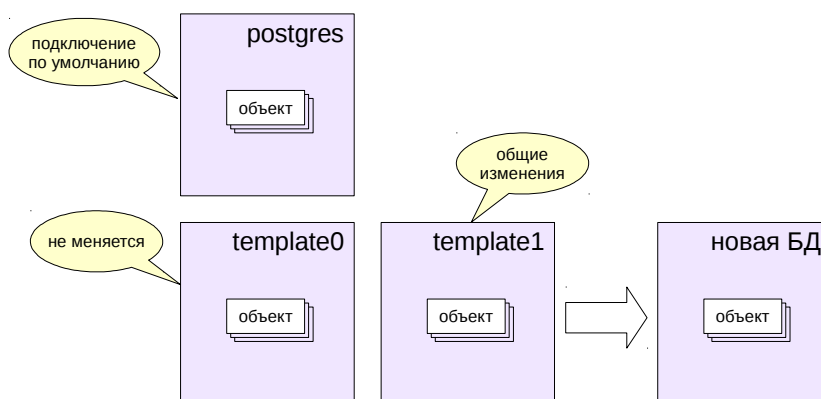
Базы данных и шаблоны

Схемы и путь поиска

Специальные схемы

Системный каталог

Инициализация кластера создает три базы данных  
Новая база всегда копируется из существующей



3

Экземпляр PostgreSQL управляет несколькими базами данных — кластером. При инициализации кластера создаются три одинаковые базы данных. Все остальные БД, создаваемые пользователем, копируются из какой-либо существующей.

Шаблон `template1` используется по умолчанию для создания новых БД. В него можно добавить объекты и расширения, которые будут копироваться в каждую новую базу данных.

Шаблон `template0` не должен изменяться. Он нужен как минимум в двух ситуациях. Во-первых, для восстановления БД из резервной копии, выполненной `pg_dump` (так как в эту копию попадут не только объекты данной БД, но и объекты, установленные в `template1`). Во-вторых, при создании новой БД с кодировкой, отличной от указанной при инициализации кластера. Подробнее эти вопросы рассматриваются в курсах по администрированию.

База данных `postgres` используется при подключении по умолчанию пользователем `postgres`. Она не является обязательной, но некоторые утилиты предполагают ее наличие, поэтому ее не рекомендуется удалять, даже если она не нужна.

<https://postgrespro.ru/docs/postgresql/9.6/manage-ag-templatedbs.html>

## Пространство имен для объектов

разделение объектов на логические группы  
предотвращение конфликта имен между приложениями

## Схема и пользователь — разные сущности

### Специальные схемы

`public` — по умолчанию в ней создаются все объекты  
`pg_catalog` — системный каталог  
`information_schema` — вариант системного каталога  
`pg_temp` — для временных таблиц  
...

Схемы представляют собой пространства имен для объектов БД. Они позволяют разделить объекты на логические группы для управления ими, предотвратить конфликты имен при работе нескольких пользователей или при установке приложений.

В PostgreSQL схема и пользователь — разные сущности (хотя настройки по умолчанию позволяют пользователям удобно работать с одноименными схемами).

Существует несколько специальных схем, обычно присутствующих в каждой базе данных.

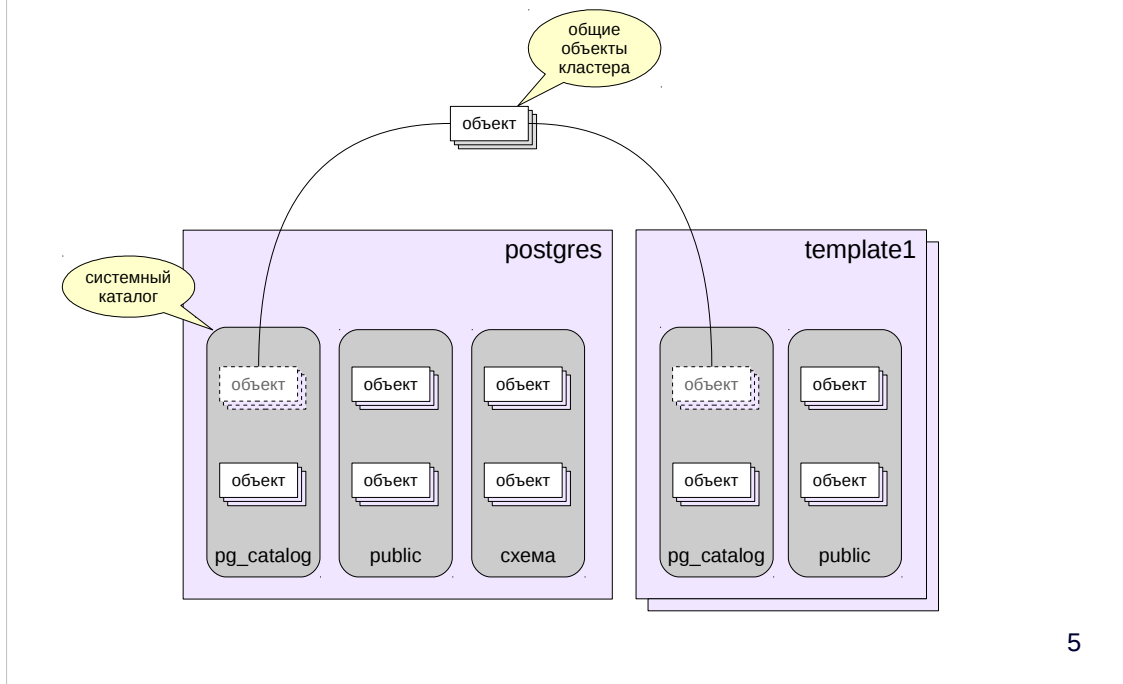
Схема `public` используется по умолчанию для хранения объектов, если не выполнены иные настройки.

Схема `pg_catalog` хранит объекты *системного каталога*. Системный каталог — это метаданные об объектах, принадлежащих кластеру, которая хранится в самом кластере в виде таблиц. Альтернативное представление системного каталога (определенное в стандарте SQL) дает схема `information_schema`.

Схема `pg_temp` служит для хранения временных таблиц. (На самом деле таблицы создаются в схемах `pg_temp_1`, `pg_temp_2` и т. п. — у каждого пользователя своя схема. Но обращаются все пользователи к ней как к `pg_temp`.)

Есть и другие схемы, но они носят технический характер.

<https://postgrespro.ru/docs/postgresql/9.6/ddl-schemas.html>



Схемы принадлежат базам данных, объекты БД распределены по схемам.

Однако несколько таблиц системного каталога хранят информацию, общую для всего кластера. Это список баз данных, список пользователей и некоторые другие сведения. Эти таблицы хранятся вне какой-либо базы данных, но при этом одинаково видны из каждой БД.

## Определение схемы объекта

квалифицированное имя (*схема.имя*) явно определяет схему  
имя без квалификатора проверяется в схемах, указанных в пути поиска

## Путь поиска

определяется параметром *search\_path*, реальное значение  
*current\_schemas()*

не включаются несуществующие схемы и схемы, к которым нет доступа  
схемы *pg\_temp* и *pg\_catalog* неявно включены первыми,  
если не указаны в *search\_path*

первая явно указанная в пути схема используется для создания объектов

При указании объекта надо определить, о какой схеме идет речь, поскольку в разных схемах могут храниться объекты с одинаковыми именами.

Если имя объекта квалифицировано именем схемы, то все просто — используется явно указанная схема. Если имя использовано без квалификатора, PostgreSQL пытается найти имя в одной из схем, перечисленных в пути поиска, который определяется конфигурационным параметром *search\_path*.

Путь поиска может отличаться от значения параметра *search\_path*. В путь поиска не включаются несуществующие схемы из *search\_path*, а также схемы, к которым у пользователя нет доступа (разграничению доступа посвящена одна из последующих тем курса). Кроме того, в начало пути поиска неявно добавляются:

- схема *pg\_catalog*, чтобы всегда иметь доступ к системному каталогу
- схема *pg\_temp*, если пользователь создавал временные объекты

Реальный путь поиска, включая неявные схемы, возвращает вызов функции: *current\_schemas(true)*. Схемы перебираются в указанном в пути поиска порядке, слева направо. Если в схеме нет объекта с нужным именем, поиск продолжается в следующей схеме.

При создании нового объекта с именем без квалификатора он попадает в первую явно указанную в пути схему.

Можно провести аналогию между путем поиска *search\_path* и путем PATH в операционных системах.

<https://postgrespro.ru/docs/postgresql/9.6/runtime-config-client.html#guc-search-path>

## Описание всех объектов кластера

набор таблиц в каждой базе данных (схема `pg_catalog`)  
и несколько глобальных объектов кластера  
набор представлений для удобства

## Доступ

запросы SQL, специальные команды `psql`

## Правила организации

названия таблиц начинаются с `pg_`  
имена столбцов содержат трехбуквенный префикс  
в качестве ключей используются скрытые поля `oid` типа `OID`  
названия объектов хранятся в нижнем регистре

Системный каталог хранит метаинформацию обо всех объектах кластера. Он состоит из набора таблиц в каждой БД и из нескольких таблиц, общих для всего кластера. Для удобства над таблицами также определены несколько представлений.

<https://postgrespro.ru/docs/postgresql/9.6/catalogs.html>

К системному каталогу можно обращаться с помощью обычных запросов SQL, а команды DDL приводят к изменению данных системного каталога. Кроме того, `psql` имеет целый ряд команд, позволяющих удобно просматривать системный каталог.

<https://postgrespro.ru/docs/postgresql/9.6/app-psql.html>

Все имена таблиц системного каталога начинаются с `pg_`, например, `pg_database`. Столбцы таблиц начинаются с префикса, обычно соответствующего имени таблицы, например, `datname`. Имена объектов хранятся в нижнем регистре, например, `'postgres'`.

Таблицы системного каталога имеют первичные ключи и связаны внешними ключами. Столбец с первичным ключом всегда называется `oid` и имеет специальный тип `OID` — Object Identifier (целое 32-битное число). Это скрытое поле, которое можно увидеть, только явно указав его имя.

<https://postgrespro.ru/docs/postgresql/9.6/datatype-oid>





## Логически

кластер содержит базы данных,  
базы данных — схемы,  
схемы — конкретные объекты (таблицы, индексы и т. п.)

Базы данных создаются клонированием существующих

Схема объекта определяется по пути поиска

Полное описание содержимого кластера баз данных  
хранится в системном каталоге

1. Создайте новую базу данных и подключитесь к ней.
2. Создайте схему, названную так же, как и пользователь.
3. Создайте схему app.
4. Создайте несколько таблиц в обеих схемах.
5. Получите в rsq1 описание созданных схем и список всех таблиц в них.
6. Установите путь поиска так, чтобы при подключении к БД таблицы из обеих схем были доступны по неквалифицированному имени; приоритет должна иметь «пользовательская» схема.
7. Проверьте правильность настройки.