Авторы: Егор Рогов, Павел Лузанов

Обратная связь
Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Темы

Базы данных и шаблоны

Схемы и путь поиска

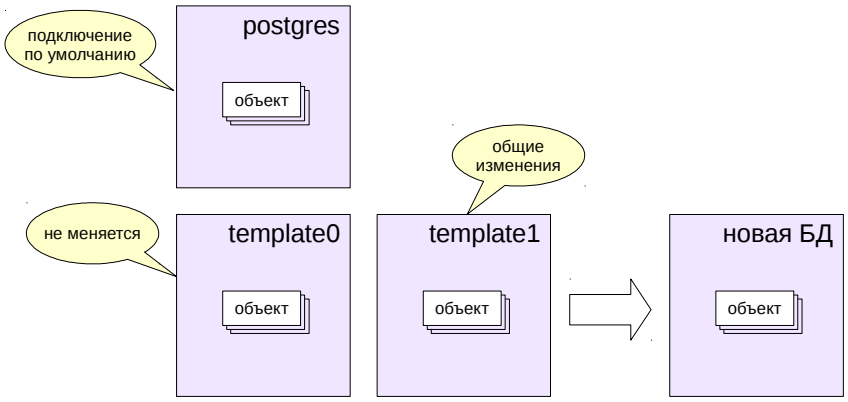
Специальные схемы, временные объекты

Управление базами, схемами и объектами в них

Кластер баз данных

Инициализация кластера создает три базы данных

Новая база всегда клонируется из существующей



3

Экземпляр PostgreSQL управляет несколькими базами данных — кластером. При инициализации кластера специальным образом создаются три одинаковые базы данных. Все остальные БД, создаваемые пользователем, клонируются 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.

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/10/manage-ag-templatedbs.html

Схемы

Пространство имен для объектов внутри базы данных

- каждый объект принадлежит какой-либо схеме

Задачи

- разделение объектов на логические группы
- предотвращение конфликта имен между приложениями

Схема и пользователь — разные сущности

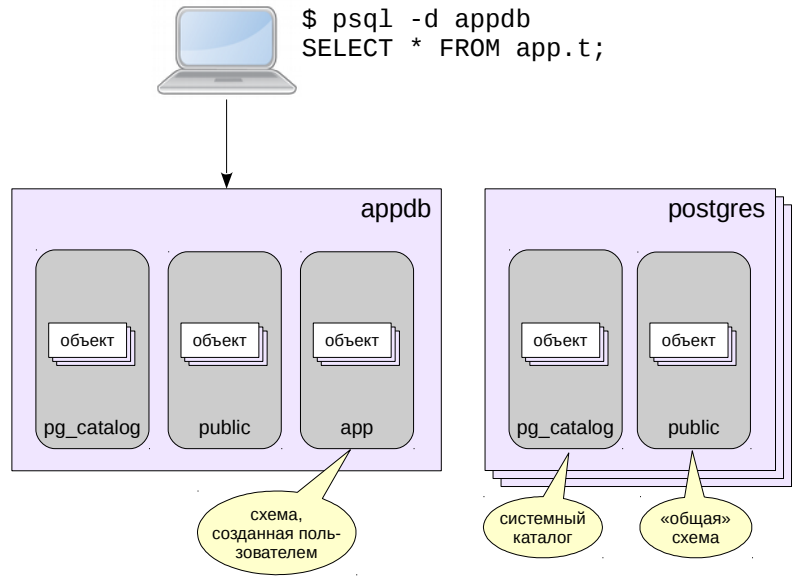
Схемы представляет собой пространства имен для объектов БД. Они позволяют разделить объекты на логические группы для управления ими, предотвратить конфликты имен при работе нескольких пользователей или при установке приложений.

Каждый объект, существующий в базе данных, принадлежит какой-либо схеме.

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

https://postgrespro.ru/docs/postgresql/10/ddl-schemas.html

Базы и схемы кластера



Кластер состоит из баз данных; база содержит различные схемы, по которым, в свою очередь, распределены объекты.

Есть некоторое количество стандартных схем, которые существуют в любой базе данных. Кроме того, пользователь может создавать свои собственные схемы.

Клиент подключается одновременно только к одной базе данных, но в этой базе может работать с объектами в любых схемах.

Путь поиска

Определение схемы объекта

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

Путь поиска

- задается параметром *search_path*
- исключаются несуществующие схемы и схемы, к которым нет доступа; подставляются неявно подразумеваемые схемы
- реальное значение показывает функция current_schemas
- 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 (как на рисунке на предыдущем слайде). Если имя использовано без квалификатора, PostgreSQL пытается найти имя в одной из схем, перечисленных в пути поиска, который определяется конфигурационным параметром search_path.

Реальный путь поиска может отличаться от значения параметра search_path. Из указанного пути исключаются несуществующие схемы, а также схемы, к которым у пользователя нет доступа (этому вопросу посвящен модуль «Разграничение доступа»). Кроме того, в начало пути поиска неявно добавляются некоторые специальные схемы.

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/10/runtime-config-client.html#guc-search-path

Специальные схемы

Схема public

- по умолчанию входит в путь поиска
- если ничего не менять, все объекты будут в этой схеме

Схема, совпадающая по имени с пользователем

- по умолчанию входит в путь поиска, но не существует
- если создать, объекты пользователя будут в этой схеме

Схема pg_catalog

- схема для объектов системного каталога
- если pg_catalog нет в пути, она неявно подразумевается первой

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 метаинформация об объектах, принадлежащих кластеру, которая хранится в самом кластере в виде таблиц. Альтернативное представление системного каталога (определенное в стандарте SQL) дает схема information_schema.

В схеме pg_catalog находятся объекты системного каталога (в частности, таблицы pg_*).

Если не указать в пути поиска pg_catalog, эта схема будет проверяться первой, чтобы системные объекты были видимы (но после pg_temp).

Специальные схемы

Временные таблицы

существуют на время сеанса или транзакции

не журналируются (невозможно восстановление после сбоя)

не попадают в общий буферный кэш

Схема pg_temp_*N*

автоматически создается для временных таблиц

pg_temp — ссылка на конкретную временную схему данного сеанса

если pg_temp нет в пути, она неявно подразумевается *самой* первой

по окончании сеанса все объекты временной схемы удаляются,
а сама схема остается и повторно используется для других сеансов

PostgreSQL умеет работать с временными таблицами. Такие таблицы предназначены для хранения данных, которые должны быть доступны только текущему сеансу (и только на время его жизни, или даже на время текущей транзакции).

Временные таблицы являются нежурналируемыми. Это означает, что в случае сбоя таблица не может быть восстановлена с помощью журнала, а вместо будет очищена. Кроме того, страницы таких таблиц не попадают в общий буферный кэш — работа с ними ведется во внутренней памяти обслуживающего процесса. Благодаря этому, работа с временной таблицей происходит несколько более эффективно, чем с обычной.

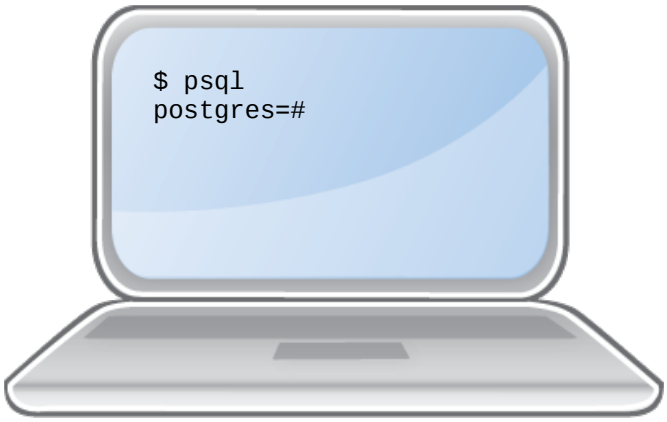
Временные таблицы организованы с помощью схем. Для сеанса создается временная схема с именем pg_temp_*N* (pg_temp_1, pg_temp_2 и т. п.). Обращаться к ней нужно по имени pg_temp (без номера) — для каждого сеанса это имя ссылается на конкретную временную схему.

Если pg_temp нет в пути, то эта схема просматривается перед всеми остальными. При желании можно указать схему pg_temp (как и pg_catalog) явно на нужном месте.

После окончания сеанса все объекты временной схемы удаляются, а сама схема остается для повторного использования.

Есть и другие специальные схемы; они носят более технический характер.

Демонстрация



Итоги

Логически

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

Базы данных создаются клонированием существующих

Схема указывается явно или определяется по пути поиска

Некоторые схемы имеют специальное назначение

Практика

1. Создайте новую базу данных и подключитесь к ней.
2. Проверьте размер базы данных.
3. Создайте две схемы: app и названную так же, как и пользователь.
4. Создайте несколько таблиц в обоих схемах и наполните их какими-нибудь данными.
5. Проверьте, на сколько увеличился размер базы данных.
6. Установите путь поиска так, чтобы при подключении к БД таблицы из обоих схем были доступны по неквалифицированному имени; приоритет должна иметь «пользовательская» схема.