



Схемы



Понятие схемы

Управление схемами и объектами в них

Поиск объектов при наличии нескольких схем

Специальные схемы

Пространство имен для объектов

разделение объектов на логические группы

предотвращение конфликта имен между приложениями

Принадлежит базе данных

Схема и пользователь — разные сущности

Специальные схемы

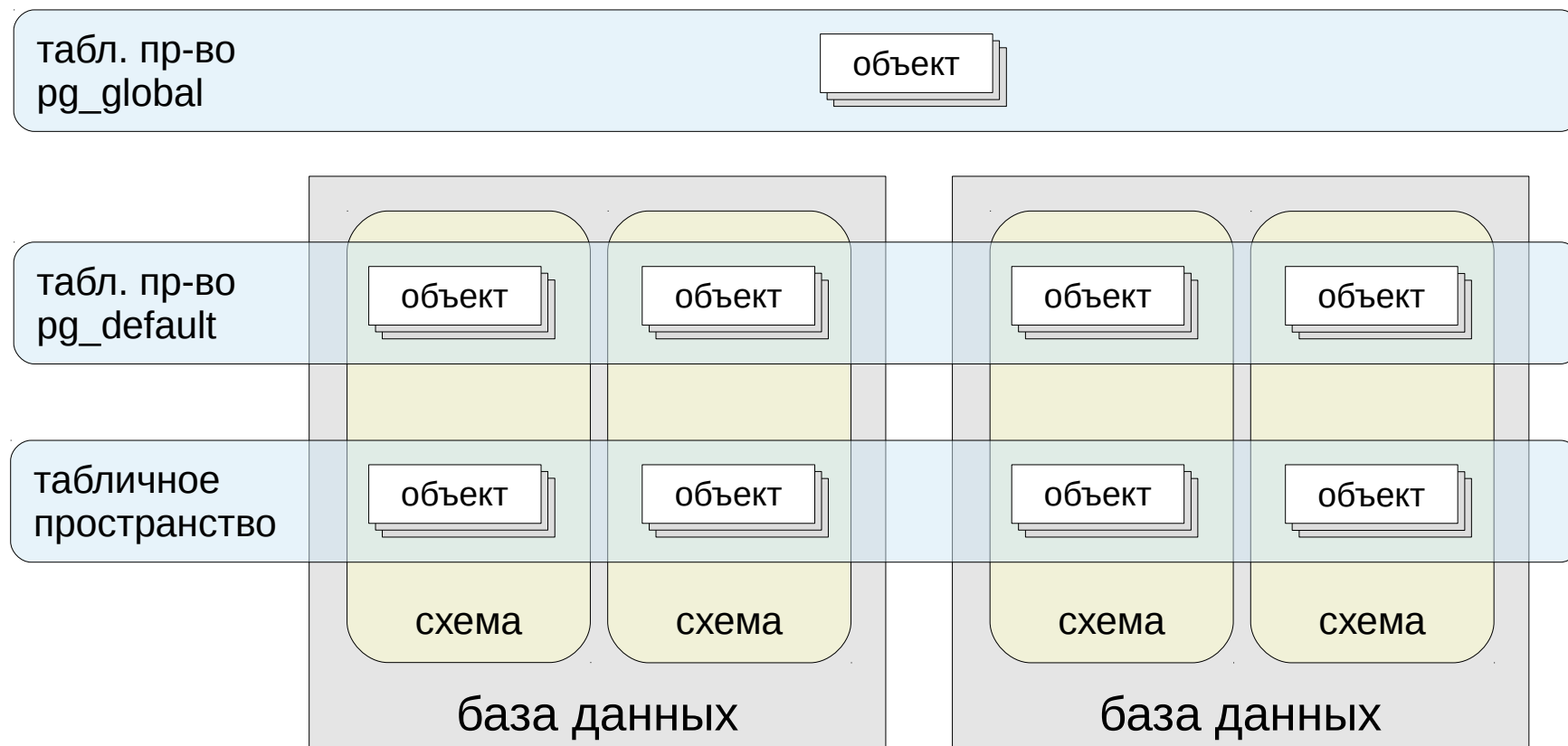
`public` — по умолчанию в ней создаются все объекты

`pg_catalog` — системные таблицы

`information_schema` — вариант системного каталога

`pg_temp` — ссылка на временную схему сеанса

кластер



Создание

```
CREATE SCHEMA schema;
```

Изменение

```
ALTER SCHEMA schema RENAME TO new_schema;
```

Удаление

```
DROP SCHEMA schema [CASCADE];  
(CASCADE удаляет все объекты)
```

Список схем

```
select * from pg_namespace;  
\dn (\dnS покажет также служебные схемы)
```

Перемещение объектов

```
ALTER TABLE name SET SCHEMA schema;  
(связанные объекты также перемещаются)
```

Список объектов в схеме

```
\d schema.*
```

Поиск объектов

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

Путь поиска

```
SHOW search_path;                                (по умолчанию "$user",public)  
select current_schemas(true);  
SET search_path TO ...;
```

схемы перебираются в указанном порядке

"\$user" обозначает схему с именем текущего пользователя

если схема не существует, она игнорируется

первая явно указанная в пути (и существующая) схема
называется текущей и используется для создания объектов

Схема pg_temp

для временных таблиц создаются схемы pg_temp_N

pg_temp — ссылка на временную схему данного сеанса

если pg_temp нет в пути, она неявно подразумевается первой

по окончании сеанса все объекты временной схемы удаляются, сама схема остается и будет повторно использоваться в других сеансах.

Схема pg_catalog

схема для объектов системного каталога

если pg_catalog нет в пути, она неявно подразумевается первой (но после pg_temp)

функция `current_schemas(true)` покажет полностью развернутый путь поиска

Схема information_schema

альтернативный взгляд на системный каталог

определена стандартом SQL

стабильна, но не содержит специфики PostgreSQL

по умолчанию не входит в путь поиска

определяет набор представлений: `\dv information_schema.*`

представления определены в `share/information_schema.sql`

(путь можно уточнить с помощью `pg_config --sharedir`)

Рассмотрели организацию данных на уровне схем

Научились создавать, изменять и удалять схемы

Научились переносить объекты из схемы в схему

Узнали про путь поиска для неквалифицированных имен

Познакомились с информационной схемой

1. Создать пользователя USR с правом создания БД.
2. Создать базу данных DB10 под пользователем USR и подключиться к ней.
3. Создать таблицу T. В какой схеме она создалась?
4. Создать схему USR.
5. Снова создать таблицу T. В какой схеме она создалась?
6. Изменить путь поиска так, чтобы по умолчанию имя T соответствовало первой таблице.



Авторские права

Курс «Администрирование PostgreSQL 9.4. Базовый курс» разработан в компании Postgres Professional (2015 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

Отказ от ответственности

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

Понятие схемы

Управление схемами и объектами в них

Поиск объектов при наличии нескольких схем

Специальные схемы

Пространство имен для объектов

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

Принадлежит базе данных

Схема и пользователь — разные сущности

Специальные схемы

- public — по умолчанию в ней создаются все объекты
- pg_catalog — системные таблицы
- information_schema — вариант системного каталога
- pg_temp — ссылка на временную схему сеанса

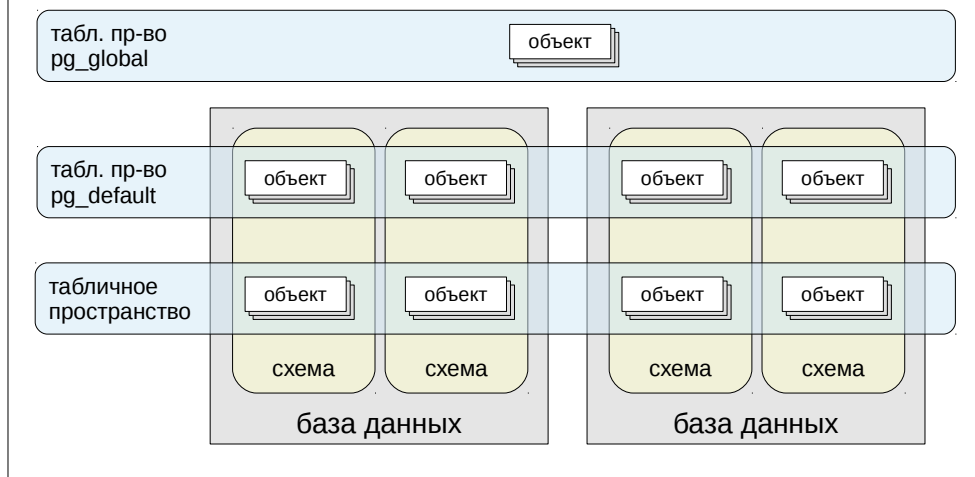
Схемы представляет собой пространства имен для объектов БД. Они позволяют разделить объекты на логические группы для управления ими, предотвратить конфликты имен при работе нескольких пользователей или при установке приложения.

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

Существует три специальных схемы, присутствующих в каждой базе данных. Схема public используется по умолчанию для хранения объектов, если только не выполнены иные настройки. Схема pg_catalog хранит системные объекты (словарь данных). Схема information_schema дает альтернативных доступ к словарю данных, регламентируемый стандартом SQL.

<http://www.postgresql.org/docs/current/static/ddl-schemas.html>

кластер



4

Табличные пространства служат для *физической* организации данных, определяя место хранения информации в файловой системе. Они относятся в целом к кластеру.

Базы данных служат для *логической* организации информации. Схемы позволяют дополнительно организовать информацию внутри каждой конкретной базы.

Создание

```
CREATE SCHEMA schema;
```

Изменение

```
ALTER SCHEMA schema RENAME TO new_schema;
```

Удаление

```
DROP SCHEMA schema [CASCADE];  
(CASCADE удаляет все объекты)
```

Список схем

```
select * from pg_namespace;  
\dn (\dnS покажет также служебные схемы)
```

В базе данных можно создавать новые схемы, переименовывать их, и при необходимости удалять. При удалении в схеме не должно остаться объектов, либо надо явно указать, что объекты также подлежат удалению.

Перемещение объектов

```
ALTER TABLE name SET SCHEMA schema;  
(связанные объекты также перемещаются)
```

Список объектов в схеме

```
\d schema.*
```

Объекты можно перемещать между схемами. При перемещении таблицы перемещаются и связанные объекты (индексы, последовательности). Также можно перемещать представления, последовательности, функции...

Поиск объектов

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

Путь поиска

```
SHOW search_path;                (по умолчанию "$user",public)
select current_schemas(true);
SET search_path TO ...;
```

схемы перебираются в указанном порядке
"\$user" обозначает схему с именем текущего пользователя
если схема не существует, она игнорируется
первая явно указанная в пути (и существующая) схема
называется текущей и используется для создания объектов

7

При указании имени объекта надо определить, о какой схеме идет речь.

Если имя объекта квалифицировано именем схемы, то используется явно указанная схема.

Если имя использовано без квалификатора, то PostgreSQL пытается найти имя в одной из схем, перечисленных в пути поиска.

Схемы перебираются в указанном порядке, слева направо. Если схема не существует, или если в ней нет объекта с нужным именем, поиск продолжается в следующей схеме.

При создании объекта (а значит, имя еще не существует) он попадает в первую указанную в пути схему.

Настройки по умолчанию подсказывают два сценария.

1. Все пользователи работают со схемой public.
2. Если создать схему с именем пользователя, этот пользователь будет работать со своей схемой.

Разумеется, можно использовать и другие сценарии, перенастроив путь поиска.

Путь поиска можно посмотреть с помощью `show search_path`, а установить на время сеанса с помощью `set search_path`. Установка конфигурационных параметров рассматривается в одной из следующих тем.

Схема pg_temp

для временных таблиц создаются схемы pg_temp_N

pg_temp — ссылка на временную схему данного сеанса

если pg_temp нет в пути, она неявно подразумевается первой

по окончании сеанса все объекты временной схемы удаляются, сама схема остается и будет повторно использоваться в других сеансах.

Схема pg_catalog

схема для объектов системного каталога

если pg_catalog нет в пути, она неявно подразумевается первой (но после pg_temp)

функция `current_schemas(true)` покажет полностью развернутый путь поиска

Временные таблицы организованы с помощью схем. Для сеанса создается временная схема с именем pg_temp_N. Имя pg_temp для каждого сеанса является ссылкой на его временную схему.

Если pg_temp нет в пути, то эта схема просматривается перед всеми остальными.

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

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).

При желании можно указать схемы pg_temp и pg_catalog явно на нужном месте.

<http://www.postgresql.org/docs/current/static/functions-info.html>

<http://www.postgresql.org/docs/current/static/runtime-config-client.html>

Схема information_schema

альтернативный взгляд на системный каталог
определена стандартом SQL
стабильна, но не содержит специфики PostgreSQL
по умолчанию не входит в путь поиска
определяет набор представлений: `\dv information_schema.*`
представления определены в `share/information_schema.sql`
(путь можно уточнить с помощью `pg_config --sharedir`)

Рассмотрели организацию данных на уровне схем

Научились создавать, изменять и удалять схемы

Научились переносить объекты из схемы в схему

Узнали про путь поиска для невалифицированных имен

Познакомились с информационной схемой

1. Создать пользователя USR с правом создания БД.
2. Создать базу данных DB10 под пользователем USR и подключиться к ней.
3. Создать таблицу T. В какой схеме она создалась?
4. Создать схему USR.
5. Снова создать таблицу T. В какой схеме она создалась?
6. Изменить путь поиска так, чтобы по умолчанию имя T соответствовало первой таблице.

Решение

```
# create user usr createdb;

# \c - usr
# create database db10;
# \c db10

# create table t(description text);
# insert into t values ('public');

# create schema usr;

# create table t(description text);
# insert into t values ('usr');

# select * from t;
-- usr
# set search_path to public;
# select * from t;
-- public
```