



Табличные пространства

Табличные пространства

Создание, изменение и удаление табличных пространств

Хранение данных в файловой системе

Перемещение данных

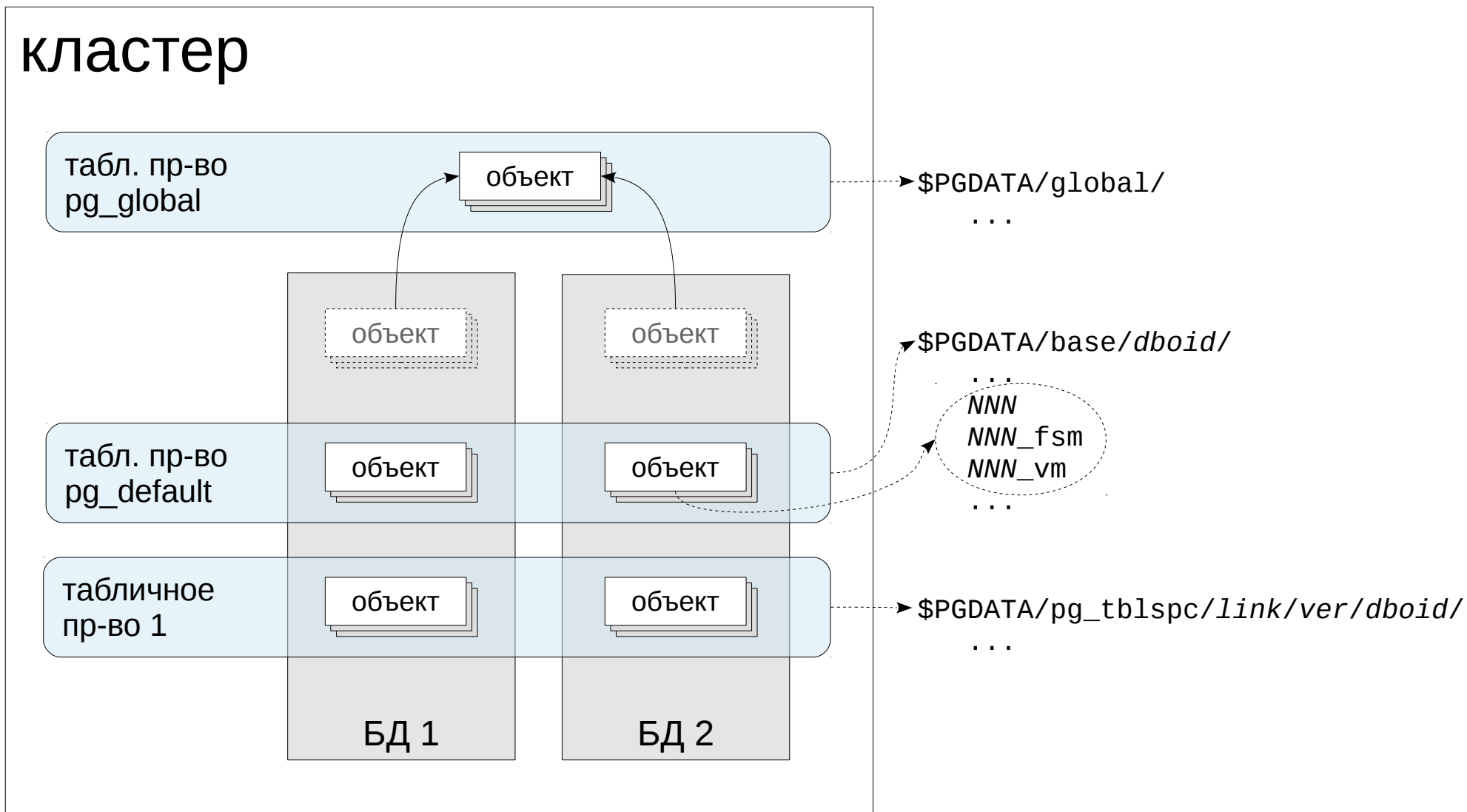
Табличные пространства – средство для организации физического хранения данных

Табличные пространства могут использоваться несколькими базами данных кластера

Каждая база данных имеет табличное пространство «по умолчанию»

Организация данных

кластер



Демонстрация

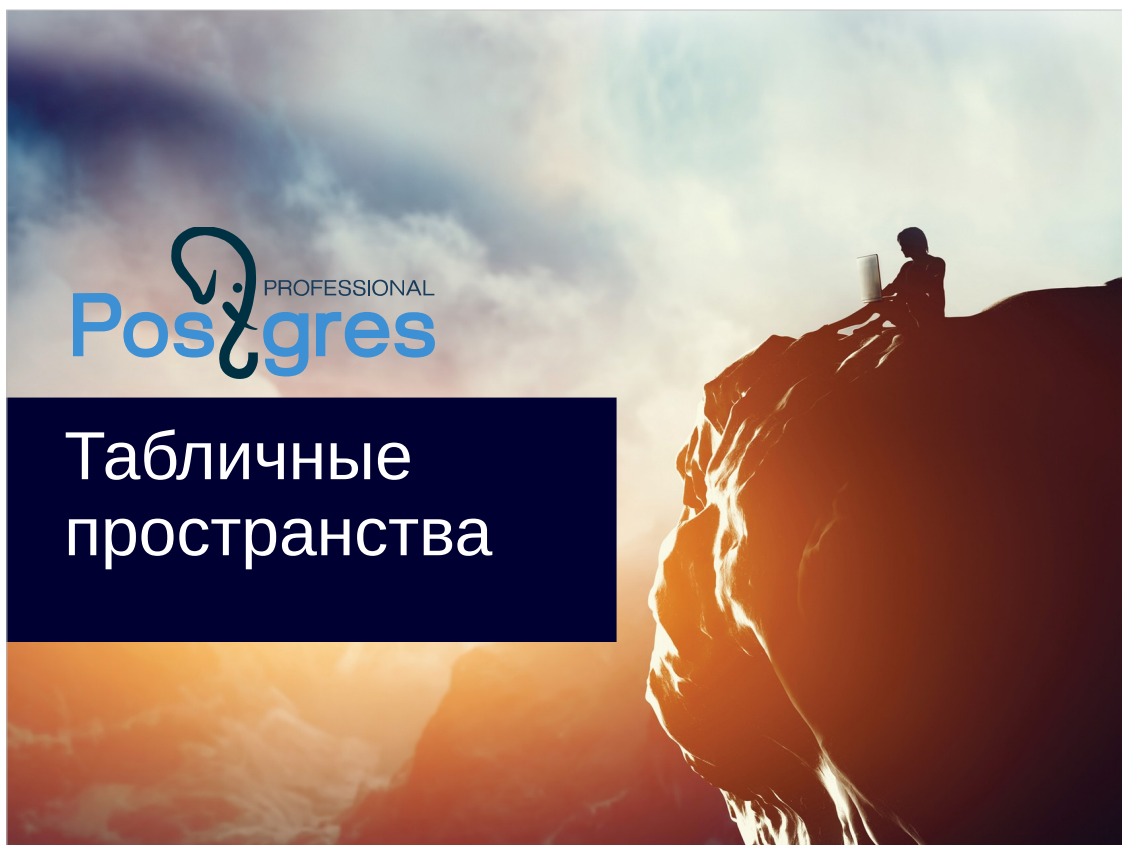


Познакомились с табличными пространствами,
их созданием и удалением

Посмотрели, как организованы данные в файловой системе

Научились перемещать данные между табличными
пространствами

1. Создать табличное пространство TS
2. Создать базу данных DB6 из шаблона template0 с табличным пространством TS по умолчанию
3. Создать в базе данных DB6 произвольную таблицу T в табличном пространстве pg_default
4. Найти файлы, относящиеся к таблице T
5. Переместить таблицу T в табличное пространство TS
6. Снова найти файлы, относящиеся к таблице T
7. Удалить табличное пространство TS



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.

Табличные пространства

Создание, изменение и удаление табличных пространств

Хранение данных в файловой системе

Перемещение данных

Табличные пространства – средство для организации физического хранения данных

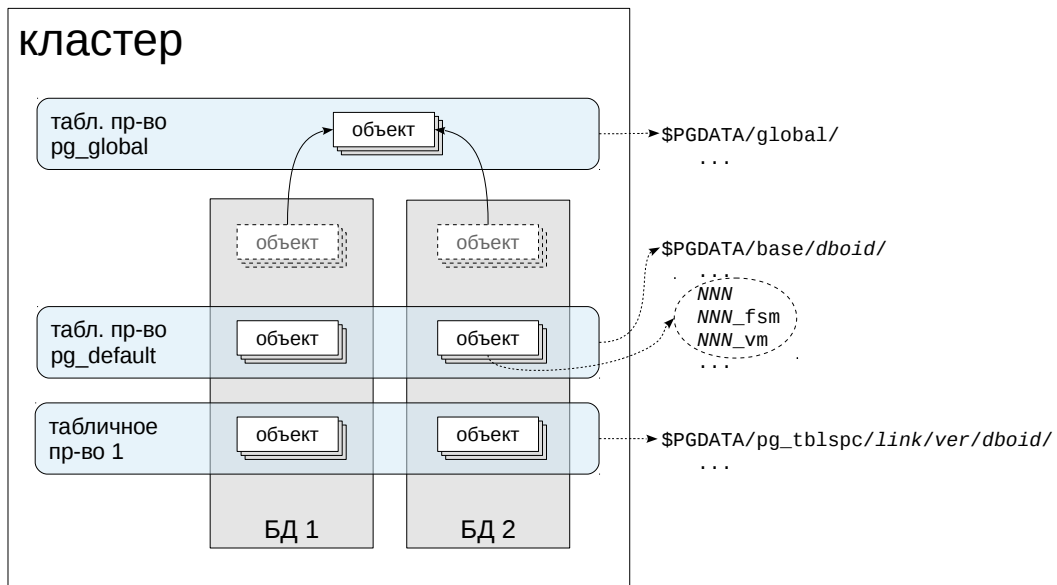
Табличные пространства могут использоваться несколькими базами данных кластера

Каждая база данных имеет табличное пространство «по умолчанию»

Табличное пространство определяет расположение данных в файловой системе.

Например, можно создать одно табличное пространство на медленных дисках для архивных данных, а другое – на быстрых для данных, с которыми идет активная работа.

Эти табличные пространства могут использоваться разными базами данных.



При создании кластера доступны два табличных пространства: `pg_global` для общих объектов кластера (`$PGDATA/global`) и `pg_default` для всех остальных (`$PGDATA/base`). При создании других табличных пространств ссылка на данные будет помещена в `$PGDATA/pg_tblspc`.

<http://www.postgresql.org/docs/9.4/static/storage-file-layout.html>



Познакомились с табличными пространствами,
их созданием и удалением

Посмотрели, как организованы данные в файловой системе

Научились перемещать данные между табличными
пространствами

1. Создать табличное пространство TS
2. Создать базу данных DB6 из шаблона template0 с табличным пространством TS по умолчанию
3. Создать в базе данных DB6 произвольную таблицу T в табличном пространстве pg_default
4. Найти файлы, относящиеся к таблице T
5. Переместить таблицу T в табличное пространство TS
6. Снова найти файлы, относящиеся к таблице T
7. Удалить табличное пространство TS

Решение

```
$ mkdir /home/postgres/ts
$ psql
# create tablespace ts location '/home/postgres/ts';

# create database db6 template template0 tablespace ts;

# \c db6
# create table t(n numeric) tablespace pg_default;

-- файлы $PGDATA/base/dboid/reliid*
-- dboid можно посмотреть oid2name
-- relid можно посмотреть oid2name -d db6 -t t

# alter table t set tablespace ts;

-- файлы $PGDATA/pg_tblspc/tsoid/ver/dboid/reliid*
-- tsoid можно посмотреть oid2name -d db6 -s
-- ver – версия PostgreSQL
-- dboid можно посмотреть oid2name
-- relid можно посмотреть oid2name -d db6 -t t
-- (relid будет отличаться от значения на шаге 4!)
```