

Почему при создании базы данных без предложения TABLESPACE табличным пространством по умолчанию становится pg_default?

1. Создайте новое табличное пространство.
2. Измените табличное пространство по умолчанию для базы данных template1 на созданное пространство.
3. Создайте новую базу данных.
Проверьте, какое табличное пространство по умолчанию установлено для новой базы данных.
4. Посмотрите в файловой системе символическую ссылку в PGDATA на каталог табличного пространства.
5. Удалите созданное табличное пространство.

1. Новое табличное пространство

```
student$ sudo -u postgres mkdir /var/lib/postgresql/ts_dir
=> CREATE TABLESPACE ts LOCATION '/var/lib/postgresql/ts_dir';
CREATE TABLESPACE
```

2. Табличное пространство по умолчанию для template1

```
=> ALTER DATABASE template1 SET TABLESPACE ts;
ALTER DATABASE
```

3. Новая база данных и проверка

```
=> CREATE DATABASE db;
CREATE DATABASE

=> SELECT spcname
FROM pg_tablespace
WHERE oid = (SELECT dattablespace FROM pg_database WHERE datname = 'db');

 spcname
-----
 ts
(1 row)
```

Табличное пространство по умолчанию — ts.

Вывод: если нет явного указания, табличное пространство по умолчанию определяется шаблоном, из которого клонируется новая база данных.

4. Символическая ссылка

```
=> SELECT oid AS tsoid FROM pg_tablespace WHERE spcname = 'ts';

 tsoid
-----
 16386
(1 row)
```

```
student$ sudo -u postgres ls -l /var/lib/postgresql/16/main/pg_tblspc/16386
lrwxrwxrwx 1 postgres postgres 26 июн 23 23:32
/var/lib/postgresql/16/main/pg_tblspc/16386 -> /var/lib/postgresql/ts_dir
```

5. Удаление табличного пространства

```
=> ALTER DATABASE template1 SET TABLESPACE pg_default;
ALTER DATABASE

=> DROP DATABASE db;
DROP DATABASE

=> DROP TABLESPACE ts;
DROP TABLESPACE

student$ sudo -u postgres rm -rf /var/lib/postgresql/ts_dir
```

1. Установите параметр *random_page_cost* в значение 1.1 для табличного пространства *pg_default*.

1. Используйте команду `ALTER TABLESPACE ... SET:`

<https://postgrespro.ru/docs/postgresql/16/sql-altertablespace>

Параметры *seq_page_cost* и *random_page_cost* используются планировщиком запросов и задают примерную стоимость чтения с диска одной страницы данных при последовательном и произвольном доступе соответственно.

Чем меньше соотношение между этими параметрами, тем чаще планировщик будет предпочитать индексный доступ последовательному сканированию таблицы.

Более подробно параметры **_cost*, и в частности *random_page_cost*, рассматриваются в курсе QPT «Оптимизация запросов».

1. Установка random_page_cost для табличного пространства

Значения по умолчанию параметров seq_page_cost и random_page_cost больше подходят для медленных HDD-дисков. Предполагается, что доступ к произвольной странице данных в четыре раза дороже последовательного:

```
=> \dconfig *page_cost
```

List of configuration parameters

Parameter	Value
random_page_cost	4
seq_page_cost	1

(2 rows)

Если используются диски с разными характеристиками, для них можно создать разные табличные пространства и настроить подходящие соотношения этих параметров. Например для быстрых SSD-дисков значение random_page_cost можно уменьшить практически до значения seq_page_cost.

```
=> ALTER TABLESPACE pg_default SET (random_page_cost = 1.1);
```

ALTER TABLESPACE

Настройки, сделанные командой ALTER TABLESPACE, сохраняются в таблице pg_tablespace. Их можно посмотреть в psql следующей командой:

```
=> \db+
```

List of tablespaces						
Name	Owner	Location	Access privileges	Options	Size	
pg_default	postgres			{random_page_cost=1.1}	29 MB	
pg_global	postgres				589 kB	

(2 rows)

Параметры *_cost можно установить и в postgresql.conf. Тогда они будут действовать для всех табличных пространств.