

1. Создайте новую базу данных и подключитесь к ней.
2. Проверьте размер созданной базы данных.
3. Создайте две схемы: `app` и названную так же, как пользователь.

Создайте несколько таблиц в обеих схемах и наполните их какими-нибудь данными.

4. Проверьте, на сколько увеличился размер базы данных.
5. Установите путь поиска так, чтобы при подключении к БД таблицы из обеих схем были доступны по невалифицированному имени; приоритет должна иметь «пользовательская» схема.

## 1. База данных

```
=> CREATE DATABASE data_databases;
```

```
CREATE DATABASE
```

```
=> \c data_databases
```

```
You are now connected to database "data_databases" as user "student".
```

## 2. Размер БД

```
=> SELECT pg_size_pretty(pg_database_size('data_databases'));
```

```
pg_size_pretty
-----
7516 kB
(1 row)
```

Запомним значение в переменной psql:

```
=> SELECT pg_database_size('data_databases') AS oldsize \gset
```

## 3. Схемы и таблицы

```
=> CREATE SCHEMA app;
```

```
CREATE SCHEMA
```

```
=> CREATE SCHEMA student;
```

```
CREATE SCHEMA
```

В какой схеме будут созданы таблицы без явного указания?

```
=> SELECT current_schema();
```

```
current_schema
-----
student
(1 row)
```

Таблицы для схемы student:

```
=> CREATE TABLE a(s text);
```

```
CREATE TABLE
```

```
=> INSERT INTO a VALUES ('student');
```

```
INSERT 0 1
```

```
=> CREATE TABLE b(s text);
```

```
CREATE TABLE
```

```
=> INSERT INTO b VALUES ('student');
```

```
INSERT 0 1
```

Таблицы для схемы app:

```
=> CREATE TABLE app.a(s text);
```

```
CREATE TABLE
```

```
=> INSERT INTO app.a VALUES ('app');
```

```
INSERT 0 1
```

```
=> CREATE TABLE app.c(s text);
```

```
CREATE TABLE
```

```
=> INSERT INTO app.c VALUES ('app');
```

```
INSERT 0 1
```

#### 4. Изменение размера БД

```
=> SELECT pg_size_pretty(pg_database_size('data_databases'));
```

```
pg_size_pretty
-----
7612 kB
(1 row)
```

```
=> SELECT pg_database_size('data_databases') AS newsize \gset
```

Размер изменился на:

```
=> SELECT pg_size_pretty(:newsize::bigint - :oldsize::bigint);
```

```
pg_size_pretty
-----
96 kB
(1 row)
```

#### 5. Путь поиска

С текущими настройками пути поиска видны таблицы только схемы student:

```
=> SELECT * FROM a;
```

```
      s
-----
student
(1 row)
```

```
=> SELECT * FROM b;
```

```
      s
-----
student
(1 row)
```

```
=> SELECT * FROM c;
```

```
ERROR:  relation "c" does not exist
LINE 1: SELECT * FROM c;
                        ^
```

Изменим путь поиска:

```
=> ALTER DATABASE data_databases SET search_path = "$user",app,public;
```

```
ALTER DATABASE
```

```
=> \c
```

You are now connected to database "data\_databases" as user "student".

```
=> SHOW search_path;
```

```
search_path
-----
"$user", app, public
(1 row)
```

Теперь видны таблицы из обеих схем, но приоритет остается за student:

```
=> SELECT * FROM a;
```

```
      s
-----
student
(1 row)
```

```
=> SELECT * FROM b;
```

```
      s
-----
student
(1 row)
```

```
=> SELECT * FROM c;
```

```
s
```

```
-----
```

```
app
```

```
(1 row)
```

1. Создайте базу данных. Сделайте так, чтобы в каждом новом сеансе, подключенном к этой базе данных, устанавливалось значение параметра *temp\_buffers*, в четыре раза превышающее значение по умолчанию.

1. Используйте команду ALTER DATABASE ... SET

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

Подробнее о параметре *temp\_buffers*:

<https://postgrespro.ru/docs/postgresql/16/runtime-config-resource#GUC-TEMP-BUFFERS>

## 1. Установка temp\_buffers

```
=> CREATE DATABASE data_databases;
```

```
CREATE DATABASE
```

```
=> \c data_databases
```

You are now connected to database "data\_databases" as user "student".

Параметр temp\_buffers определяет объем памяти, выделяемый в каждом сеансе под локальный кеш для временных таблиц. Если данные временных таблиц не помещаются в temp\_buffers, страницы вытесняются, как это происходит в обычном буферном кеше. Недостаточное значение может привести к деградации производительности при активном использовании временных таблиц.

Значение по умолчанию для temp\_buffers составляет 8 Мбайт:

```
=> SELECT name, setting, unit, boot_val, reset_val
FROM pg_settings
WHERE name = 'temp_buffers' \gx
```

```
-[ RECORD 1 ]-----
name       | temp_buffers
setting    | 1024
unit       | 8kB
boot_val   | 1024
reset_val  | 1024
```

Установим для всех новых сеансов базы данных значение 32 Мбайта:

```
=> ALTER DATABASE data_databases SET temp_buffers = '32MB';
```

```
ALTER DATABASE
```

```
=> \c
```

You are now connected to database "data\_databases" as user "student".

```
=> SHOW temp_buffers;
```

```
temp_buffers
-----
32MB
(1 row)
```

Настройки, сделанные командой ALTER DATABASE, сохраняются в таблице pg\_db\_role\_setting. Их можно посмотреть в psql следующей командой:

```
=> \drds
```

```
                List of settings
Role | Database | Settings
-----+-----+-----
      | data_databases | temp_buffers=32MB
(1 row)
```

Конечно, параметр temp\_buffers не обязательно настраивать на уровне базы данных. Например, его можно настроить в postgresql.conf для всего кластера.