

## Практика 2

```
postgres=# SELECT pg_read_file('/etc/postgresql/16/main/postgresql.conf', 1516, 861)
ошибка: не удалось открыть файл "/etc/postgresql/16/main/postgresql.conf" для чтения
postgres=# SHOW config_file \g
    config_file
-----
C:/PostgreSQL/data/postgresql.conf
(1 строка)

postgres=# SELECT sourceline, name, setting, applied, error
postgres-# FROM pg_file_settings \g
   sourceline |          name          |      setting      | applied | error
-----+-----+-----+-----+-----+-----+
      65 | max_connections |      100       |      t
     132 | shared_buffers  | 128MB        |      t
    155 | dynamic_shared_memory_type | windows
    267 | max_wal_size    |      1GB       |      t
    268 | min_wal_size    |     80MB       |      t
    501 | log_file_mode   |      0640      |      t
    641 | log_timezone    | Asia/Qyzylorda |      t
    683 | autovacuum_worker_slots |      16
    777 | datestyle       | iso, dmy
    779 | timezone        | Asia/Qyzylorda |      t
    793 | lc_messages     | Russian_Russia.1251 |      t
    795 | lc_monetary     | Russian_Russia.1251 |      t
    796 | lc_numeric       | Russian_Russia.1251 |      t
    797 | lc_time          | Russian_Russia.1251 |      t
    803 | default_text_search_config | pg_catalog.russian |      t
(15 строк)
```

```
postgres=# SELECT name, unit, setting, boot_val, reset_val,
:postres-#           source, sourcefile, sourceline, pending_restart, context
postgres-# FROM pg_settings
postgres-# WHERE name = 'work_mem' \gx
-[ RECORD 1 ]-----+
name        | work_mem
unit        | kB
setting     | 4096
boot_val    | 4096
reset_val   | 4096
source      | default
sourcefile  |
sourceline  |
pending_restart | f
context     | user

postgres=# SELECT pg_reload_conf() \g
pg_reload_conf
-----
t
(1 строка)
```

```

postgres=# ALTER SYSTEM SET work_mem TO '16MB' \g
ALTER SYSTEM
postgres=# SELECT pg_read_file('postgresql.auto.conf') \g
          pg_read_file
-----
# Do not edit this file manually! +
# It will be overwritten by the ALTER SYSTEM command.+ +
work_mem = '16MB' +
(1 行)

postgres=# SHOW work_mem \g
  work_mem
-----
4MB
(1 行)

postgres=# SELECT pg_reload_conf() \g
      pg_reload_conf
-----
t
(1 行)

```

```

postgres=# SELECT name, unit, setting, boot_val, reset_val,
postgres-#           source, sourcefile, sourceline, pending_restart, context
postgres-# FROM pg_settings
postgres-# WHERE name = 'work_mem' \gx
-[ RECORD 1 ]-----+
name        | work_mem
unit        | kB
setting     | 16384
boot_val    | 4096
reset_val   | 16384
source      | configuration file
sourcefile  | C:/PostgreSQL/data/postgresql.auto.conf
sourceline  | 3
pending_restart | f
context     | user

postgres=# ALTER SYSTEM RESET work_mem \g
ALTER SYSTEM
postgres=# SELECT pg_read_file('postgresql.auto.conf') \g
          pg_read_file
-----
# Do not edit this file manually! +
# It will be overwritten by the ALTER SYSTEM command.+ +
(1 行)

```

```
postgres=# SELECT name, unit, setting, boot_val, reset_val,
postgres-#           source, sourcefile, sourceline, pending_restart, context
postgres-# FROM pg_settings
postgres-# WHERE name = 'work_mem' \gx
-[ RECORD 1 ]-----+
name          | work_mem
unit          | kB
setting       | 4096
boot_val      | 4096
reset_val     | 4096
source        | default
sourcefile    |
sourceline   |
pending_restart | f
context       | user

postgres=# SET work_mem TO '24MB' \g
SET
postgres=# SELECT set_config('work_mem', '32MB', false) \g
set_config
-----
32MB
(1 ё€Ёюр)
```

```
postgres=# SHOW work_mem \g
work_mem
-----
32MB
(1 ё€Ёюр)

postgres=# SELECT current_setting('work_mem') \g
current_setting
-----
32MB
(1 ё€Ёюр)

postgres=# SELECT name, setting, unit FROM pg_settings WHERE name = 'work_mem' \g
 name   | setting | unit
-----+-----+-----+
work_mem | 32768   | kB
(1 ё€Ёюр)
```

```
postgres=# RESET work_mem \g
RESET
postgres=# BEGIN \g
BEGIN
postgres=# SET work_mem TO '64MB' \g
SET
postgres=# SHOW work_mem \g
 work_mem
-----
 64MB
(1 行)


```

```
postgres=# ROLLBACK \g
ROLLBACK
postgres=# SHOW work_mem \g
 work_mem
-----
 4MB
(1 行)


```

```
postgres=# BEGIN \g
BEGIN
postgres=# SET LOCAL work_mem TO '64MB' \g
SET
postgres=# SHOW work_mem \g
 work_mem
-----
 64MB
(1 行)


```

```
postgres=# COMMIT \g
COMMIT
postgres=# SHOW work_mem \g
 work_mem
-----
 4MB
(1 行)


```

```
postgres=# SELECT CASE
postgres-#   WHEN current_setting('myapp.currency_code', true) IS NULL
postgres-#   THEN set_config('myapp.currency_code', 'RUB', false)
postgres-#   ELSE current_setting('myapp.currency_code')
postgres-# END \g
 current_setting
-----
 RUB
(1 行)


```

```
postgres=# SELECT current_setting('myapp.currency_code') \g;
некорректная команда \g;
Введите \? для получения справки.
postgres=# SELECT current_setting('myapp.currency_code') \g
ошибка:  syntax error at or near "current_setting"
строка 2: SELECT current_setting('myapp.currency_code')
          ^
postgres=# SELECT current_setting('myapp.currency_code')
postgres-# \g
  current_setting
-----
  RUB
(1 строка)
```

## Практика+



1. Установите параметр `work_mem = 32MB` в командной строке запуска утилиты `psql`.
2. В пакетном дистрибутиве для Ubuntu файл `postgresql.conf` находится не в каталоге PGDATA. Каким образом сервер находит этот файл конфигурации при запуске?

12

1. Используйте один из двух способов: ключ `options` в строке подключения или переменная среды `PGOPTIONS`.

Информация о формировании строки подключения:

<https://postgrespro.ru/docs/postgresql/16/libpq-connect#LIBPQ-CONNSTRING>

2. Расположение файла `postgresql.conf` можно посмотреть в параметре `config_file`.

Чтобы узнать, где задается этот параметр, посмотрите вывод команды `ps` для основного процесса `postgres`. Идентификатор процесса (PID) записан в первой строке файла `postmaster.pid`, который находится в каталоге с данными (PGDATA).

```
postgres=# \q
C:\PostgreSQL\pgsql\bin>set PGOPTIONS=-c work_mem=32MB
C:\PostgreSQL\pgsql\bin>psql -d postgres
psql (18.1)
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной
страницы Windows (1251).
8-битовые (русские) символы могут отображаться некорректно.
Подробнее об этом смотрите документацию psql, раздел
"Notes for Windows users".
Введите "help", чтобы получить справку.

postgres=# SHOW work_mem;
 work_mem
-----
 32MB
(1 строка)
```

```
C:\PostgreSQL\pgsql\bin>psql -d postgres -c "SHOW config_file"
 config_file
-----
 C:/PostgreSQL/data/postgresql.conf
(1 строка)

C:\PostgreSQL\pgsql\bin>type "C:\Program Files\PostgreSQL\16\data\postmaster.pid"
Системе не удается найти указанный путь.

C:\PostgreSQL\pgsql\bin>wmic process where processid=<PID> get commandline
Не удается найти указанный файл.

C:\PostgreSQL\pgsql\bin>-c config_file="C:\Program Files\PostgreSQL\16\data\postgresql.conf"
"-c" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.
```

2 часть

```
postgres=# CREATE TABLE t(s text);
CREATE TABLE
postgres=# INSERT INTO t VALUES ('Первая версия');
INSERT 0 1
postgres=# BEGIN;
BEGIN
postgres=# SELECT pg_current_xact_id();
 pg_current_xact_id
-----
 889
(1 ё€Ёюр)

postgres=# SELECT *, xmin, xmax FROM t;
      s      | xmin | xmax
-----+-----+
 Первая версия | 888 |    0
(1 ё€Ёюр)

postgres=# BEGIN;
ПРЕДУПРЕЖДЕНИЕ: транзакция уже выполняется
BEGIN
postgres=# SELECT pg_current_xact_id();
 pg_current_xact_id
-----
 889
(1 ё€Ёюр)

postgres=# \q

C:\PostgreSQL\pgsql\bin>psql.exe -d postgres
psql (18.1)
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается от основной
страницы Windows (1251).
8-битовые (русские) символы могут отображаться некорректно.
Подробнее об этом смотрите документацию psql, раздел
"Notes for Windows users".
Введите "help", чтобы получить справку.

postgres=# BEGIN;
BEGIN
postgres=# SELECT pg_current_xact_id();
 pg_current_xact_id
-----
 890
(1 ё€Ёюр)
```

```
postgres=# SELECT *, xmin, xmax FROM t;
      s      | xmin | xmax
-----+-----+
Первая версия |  888 |    0
(1 строка)

postgres=# UPDATE t SET s = 'Вторая версия';
UPDATE 1
postgres=#  SELECT *, xmin, xmax FROM t;
      s      | xmin | xmax
-----+-----+
Вторая версия |  890 |    0
(1 строка)

postgres=# SELECT *, xmin, xmax FROM t;
      s      | xmin | xmax
-----+-----+
Вторая версия |  890 |    0
(1 строка)

postgres=# COMMIT;
COMMIT
postgres=# SELECT *, xmin, xmax FROM t;
      s      | xmin | xmax
-----+-----+
Вторая версия |  890 |    0
(1 строка)

postgres# COMMIT;
ПРЕДУПРЕЖДЕНИЕ:  нет незавершённой транзакции
COMMIT
```

```
postgres=# COMMIT;
ПРЕДУПРЕЖДЕНИЕ:  нет незавершённой транзакции
COMMIT
postgres=# BEGIN;
BEGIN
postgres=# UPDATE t SET s = 'Третья версия' RETURNING *;
      s
-----
 Третья версия
(1 строка)

UPDATE 1
postgres=# BEGIN;
ПРЕДУПРЕЖДЕНИЕ:  транзакция уже выполняется
BEGIN
postgres=# UPDATE t SET s = 'Четвертая версия' RETURNING
      s
-----
 Четвертая версия
(1 строка)

UPDATE 1
postgres=# COMMIT;
COMMIT
postgres=# COMMIT;
ПРЕДУПРЕЖДЕНИЕ:  нет незавершённой транзакции
COMMIT
postgres=# SELECT * FROM t;
      s
-----
 Четвертая версия
(1 строка)

postgres=# ■
```

## Практика



1. Создайте таблицу с одной строкой.

Начните первую транзакцию на уровне изоляции Read Committed и выполните запрос к таблице.

Во втором сеансе удалите строку и зафиксируйте изменения. Сколько строк увидит первая транзакция, выполнив тот же запрос повторно? Проверьте.

Завершите первую транзакцию.

2. Повторите то же самое, но пусть теперь транзакция работает на уровне изоляции Repeatable Read:

```
BEGIN ISOLATION LEVEL REPEATABLE READ;
```

Объясните отличия.

```
postgres=# DROP TABLE IF EXISTS t CASCADE;
DROP TABLE
postgres=# CREATE TABLE t(id int);
CREATE TABLE
postgres=# INSERT INTO t VALUES (1);
INSERT 0 1
postgres=# BEGIN ISOLATION LEVEL READ COMMITTED;
BEGIN
postgres=#*
postgres=#*# SELECT * FROM t;
 id
 ----
  1
(1 строка)

postgres=#*# DELETE FROM t;
DELETE 1
postgres=#*# COMMIT;
COMMIT
postgres=# SELECT * FROM t;
 id
 ----
(0 строк)

postgres=# COMMIT;
ПРЕДУПРЕЖДЕНИЕ:  нет незавершённой транзакции
COMMIT
postgres=# BEGIN ISOLATION LEVEL REPEATABLE READ;
BEGIN
postgres=#*
postgres=#*# SELECT * FROM t;
 id
 ----
(0 строк)

postgres=#*# INSERT INTO t VALUES (1); -- если таблица пустая, можешь заново вставить
INSERT 0 1
postgres=#*# DELETE FROM t;           -- удалить строку
DELETE 1
postgres=#*# COMMIT;
COMMIT
postgres=# SELECT * FROM t;
 id
 ----
(0 строк)
```

1. Начните транзакцию и создайте новую таблицу с одной строкой. Не завершая транзакцию, откройте второй сеанс и выполните в нем запрос к таблице. Проверьте, что увидит транзакция во втором сеансе.  
Зафиксируйте транзакцию в первом сеансе и повторите запрос к таблице во втором сеансе.
2. Повторите задание 1, но откатите, а не зафиксируйте транзакцию в первом сеансе. Что изменилось?
3. В первом сеансе начните транзакцию и выполните запрос к созданной ранее таблице. Получится ли удалить эту таблицу во втором сеансе, пока первая транзакция не завершена? Проверьте.

```
postgres=# BEGIN;
BEGIN
postgres=# CREATE TABLE t1(id int);
CREATE TABLE
postgres=# INSERT INTO t1 VALUES (1);
INSERT 0 1
postgres=# SELECT * FROM t1;
 id
---
 1
(1 ё€Ёюр)

postgres=# COMMIT;
COMMIT
postgres=# SELECT * FROM t1;
 id
---
 1
(1 ё€Ёюр)

postgres=# BEGIN;
BEGIN
postgres=# CREATE TABLE t2(id int);
CREATE TABLE
postgres=# INSERT INTO t2 VALUES (10);
INSERT 0 1
postgres=# SELECT * FROM t2;
 id
---
 10
(1 ё€Ёюр)

postgres=# ROLLBACK;
ROLLBACK
postgres=# SELECT * FROM t2;
ОШИБКА:  отношение "t2" не существует
СТРОКА 1: SELECT * FROM t2;
^
postgres=# BEGIN;
BEGIN
postgres=# SELECT * FROM t1;
 id
---
 1
(1 ё€Ёюр)

postgres=# DROP TABLE t1;
DROP TABLE
postgres=# COMMIT;
COMMIT
postgres=# -
```