

Национальный исследовательский ядерный университет «МИФИ»

(Московский Инженерно-Физический Институт)

Кафедра №42 «Криптология и кибербезопасность»

Лабораторная работа №2-3

«Базовые задачи администрирования СУБД»

Тимин Александр Б21-515 (2024г.)

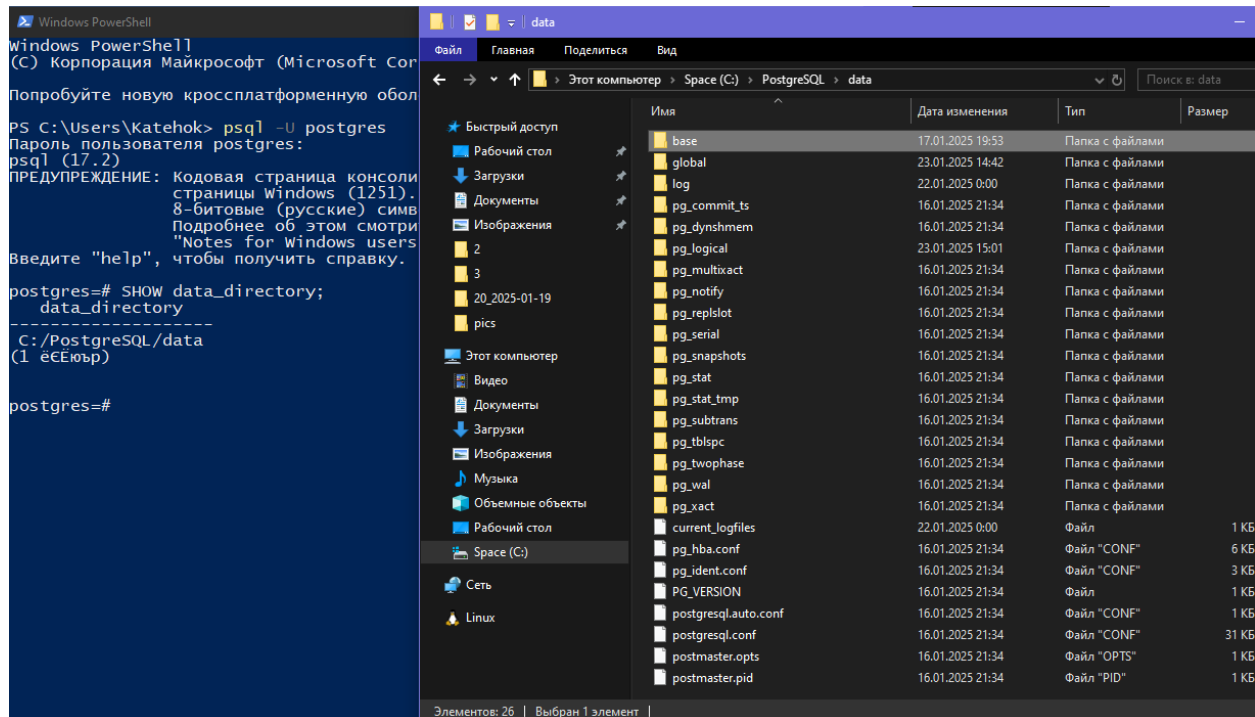
1 Где БД?

Подключимся к БД, используя PowerShell:

```
psql -U postgres
```

Запросим путь до директории с данными:

```
SHOW data_directory;
```



У меня данные БД postgres хранятся по пути C:\PostgreSQL\data.

2 Что в директории?

Основные файлы и папки:

- **base/**: Содержит данные всех баз данных в кластере. Каждая база данных представлена подкаталогом с уникальным идентификатором.
- **global/**: Хранит глобальные метаданные кластера, такие как роли, таблицы настройки и системные каталоги.
- **pg_wal/**: Каталог для WAL-журналов (Write-Ahead Logging), которые используются для восстановления базы данных в случае сбоя.

- `pg_hba.conf`: Файл настройки правил доступа к базам данных (Host-Based Authentication).
- `postgresql.conf`: Основной файл конфигурации PostgreSQL. Здесь задаются параметры, такие как размер буферов, количество соединений, настройки логирования и многое другое.
- `pg_ident.conf`: Используется для сопоставления системных пользователей и ролей PostgreSQL.
- `log/`: Папка, в которой хранятся журналы работы PostgreSQL (если логирование настроено).
- `PG_VERSION`: Текстовый файл, содержащий номер версии PostgreSQL, установленной в этом кластере.
- `postmaster.pid`: Указывает, что кластер запущен, и содержит информацию о процессе `postmaster`.

```

Windows PowerShell
PS C:\Users\Katehok> cd "C:/PostgreSQL/data"
PS C:\PostgreSQL\data> dir

Каталог: C:\PostgreSQL\data

Mode                LastWriteTime         Length Name
----                -
d-----         17.01.2025         19:53         base
d-----         23.01.2025         14:42        global
d-----         22.01.2025          0:00         log
d-----         16.01.2025         21:34    pg_commit_ts
d-----         16.01.2025         21:34    pg_dynshmem
d-----         23.01.2025         15:01    pg_logical
d-----         16.01.2025         21:34    pg_multixact
d-----         16.01.2025         21:34    pg_notify
d-----         16.01.2025         21:34    pg_replslot
d-----         16.01.2025         21:34    pg_serial
d-----         16.01.2025         21:34    pg_snapshots
d-----         16.01.2025         21:34    pg_stat
d-----         16.01.2025         21:34    pg_stat_tmp
d-----         16.01.2025         21:34    pg_subtrans
d-----         16.01.2025         21:34    pg_tblspc
d-----         16.01.2025         21:34    pg_twophase
d-----         16.01.2025         21:34    pg_wal
d-----         16.01.2025         21:34    pg_xact
-a----         22.01.2025          0:00          45 current_logfiles
-a----         16.01.2025         21:34        5639 pg_hba.conf
-a----         16.01.2025         21:34        2712 pg_ident.conf
-a----         16.01.2025         21:34           3 PG_VERSION
-a----         16.01.2025         21:34          90 postgresql.auto.conf
-a----         16.01.2025         21:34       31565 postgresql.conf
-a----         16.01.2025         21:34          57 postmaster.opts
-a----         16.01.2025         21:34          54 postmaster.pid

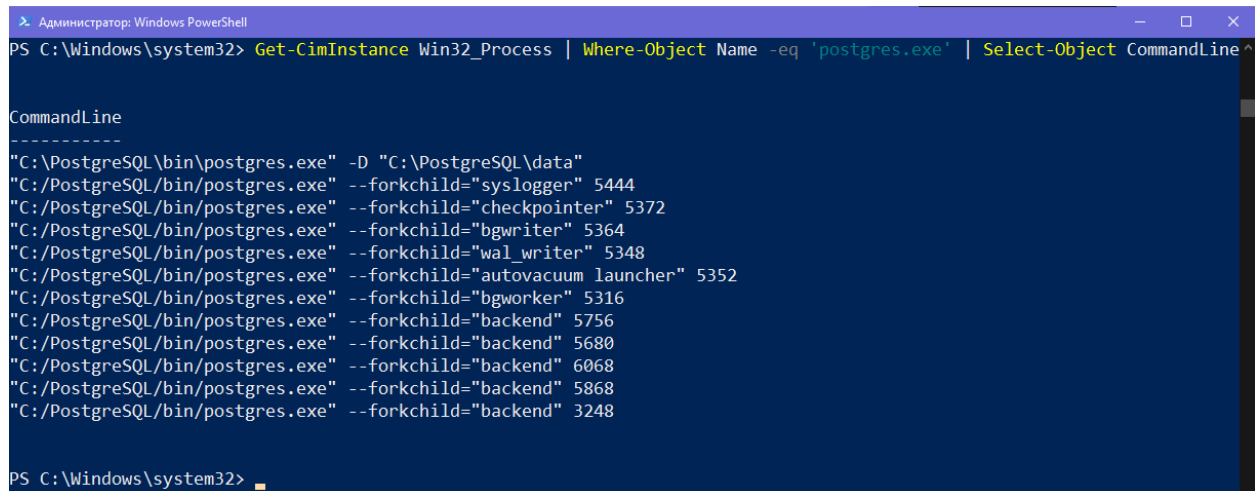
PS C:\PostgreSQL\data>

```

3 Как запущен экземпляр PostgreSQL?

Запустив PowerShell с правами администратора, выполнил:

```
Get-CimInstance Win32_Process | Where-Object Name -eq 'postgres.exe' | Select-Object CommandLine
```



```
Администратор: Windows PowerShell
PS C:\Windows\system32> Get-CimInstance Win32_Process | Where-Object Name -eq 'postgres.exe' | Select-Object CommandLine

CommandLine
-----
"C:\PostgreSQL\bin\postgres.exe" -D "C:\PostgreSQL\data"
"C:/PostgreSQL/bin/postgres.exe" --forkchild="syslogger" 5444
"C:/PostgreSQL/bin/postgres.exe" --forkchild="checkpointer" 5372
"C:/PostgreSQL/bin/postgres.exe" --forkchild="bgwriter" 5364
"C:/PostgreSQL/bin/postgres.exe" --forkchild="wal_writer" 5348
"C:/PostgreSQL/bin/postgres.exe" --forkchild="autovacuum launcher" 5352
"C:/PostgreSQL/bin/postgres.exe" --forkchild="bgworker" 5316
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5756
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5680
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 6068
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5868
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 3248

PS C:\Windows\system32>
```

```
CommandLine
-----
"C:\PostgreSQL\bin\postgres.exe" -D "C:\PostgreSQL\data"
"C:/PostgreSQL/bin/postgres.exe" --forkchild="syslogger" 5444
"C:/PostgreSQL/bin/postgres.exe" --forkchild="checkpointer" 5372
"C:/PostgreSQL/bin/postgres.exe" --forkchild="bgwriter" 5364
"C:/PostgreSQL/bin/postgres.exe" --forkchild="wal_writer" 5348
"C:/PostgreSQL/bin/postgres.exe" --forkchild="autovacuum
launcher" 5352
"C:/PostgreSQL/bin/postgres.exe" --forkchild="bgworker" 5316
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5756
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5680
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 6068
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 5868
"C:/PostgreSQL/bin/postgres.exe" --forkchild="backend" 3248
```

Первый выведенный процесс – основной, запущен с указанием директории данных.

Остальные процессы, начинающиеся с `--forkchild`, — это рабочие процессы PostgreSQL. Вот их назначения:

- `syslogger`: Запись журналов.

- `checkpointer`: Обеспечение сохранности данных через контрольные точки.
- `bgwriter`: Запись данных на диск в фоновом режиме.
- `wal_writer`: Работа с WAL (Write-Ahead Logging).
- `autovacuum launcher`: Удаление "мусорных" данных и предотвращение переполнения транзакций.
- `bgworker`: Фоновые задачи.
- `backend`: Процессы клиентов, подключающихся к серверу (каждый клиент имеет свой backend-процесс).

4 Штатное завершение работы

С помощью `pg_ctl` – утилиты для управления сервером PostgreSQL завершим его работу:

```
pg_ctl stop -D "C:\PostgreSQL\data"
```

Эта команда:

- Завершает работу сервера PostgreSQL.
- Гарантирует, что все данные будут записаны в файловую систему.
- Закрывает все клиентские соединения.

```

Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Process -Name postgres

Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI ProcessName
-----  -
350      10      2968   13700   0,09    7560  0 postgres
356      10      4288   21096   0,28    8844  0 postgres
348      10      2624   9404    0,06   15524  0 postgres
352      10      3580   10340   1,50   16916  0 postgres
352      10      3536   10104   0,06   21224  0 postgres
349      10      3028   11152   0,08   23624  0 postgres
398      13      4516   21884   45,11  24700  0 postgres

PS C:\Windows\system32> pg_ctl stop -D "C:\PostgreSQL\data"
ожидание завершения работы сервера.... готово
сервер остановлен
PS C:\Windows\system32> Get-Process -Name postgres
Get-Process : Не удалось найти процесс с именем "postgres". Убедитесь, что имя процесса введено правильно, и повторите
вызов командлета.
строка:1 знак:1
+ Get-Process -Name postgres
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (postgres:String) [Get-Process], ProcessCommandException
+ FullyQualifiedErrorId : NoProcessFoundForGivenName,Microsoft.PowerShell.Commands.GetProcessCommand
PS C:\Windows\system32>

```

Как видно из скриншота, больше нет активных процессов СУБД.

5 Запуск вручную

С помощью все той же утилиты запустим процесс, указав нужную директорию:

```
pg_ctl start -D "C:\PostgreSQL\data"
```

```
Администратор: Windows PowerShell
PS C:\Windows\system32> Get-Process -Name postgres
Get-Process : Не удалось найти процесс с именем "postgres". Убедитесь, что имя процесса введено правильно, и повторите
вызов командлета.
строка:1 знак:1
+ Get-Process -Name postgres
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (postgres:String) [Get-Process], ProcessCommandException
+ FullyQualifiedErrorId : NoProcessFoundForGivenName,Microsoft.PowerShell.Commands.GetProcessCommand

PS C:\Windows\system32> pg_ctl start -D "C:\PostgreSQL\data"
ожидание запуска сервера....2025-01-23 18:28:42.495 MSK [14960] СООБЩЕНИЕ: передача вывода в протокол процессу сбора пр
отоколов
2025-01-23 18:28:42.495 MSK [14960] ПОДСКАЗКА: В дальнейшем протоколы будут выводиться в каталог "log".
готово
сервер запущен
PS C:\Windows\system32> Get-Process -Name postgres

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
359 10 3176 11208 0,02 7284 10 postgres
359 10 3184 11172 0,03 10676 10 postgres
362 10 3728 11800 0,02 13476 10 postgres
406 13 4380 23932 0,06 14960 10 postgres
359 10 3172 11164 0,02 20116 10 postgres
362 10 3740 11920 0,05 28436 10 postgres
359 10 2816 10896 0,00 30152 10 postgres

PS C:\Windows\system32>
```

6 Проверка

Подключимся к БД и выполним запрос.

```
Администратор: Windows PowerShell
PS C:\Windows\system32> pg_ctl start -D "C:\PostgreSQL\data"
ожидание запуска сервера....2025-01-23 18:28:42.495 MSK [14960] СООБЩЕНИЕ: передача вывода в протокол процессу сбора пр
отоколов
2025-01-23 18:28:42.495 MSK [14960] ПОДСКАЗКА: В дальнейшем протоколы будут выводиться в каталог "log".
готово
сервер запущен
PS C:\Windows\system32> Get-Process -Name postgres

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
359 10 3176 11208 0,02 7284 10 postgres
359 10 3184 11172 0,03 10676 10 postgres
362 10 3728 11800 0,02 13476 10 postgres
406 13 4380 23932 0,06 14960 10 postgres
359 10 3172 11164 0,02 20116 10 postgres
362 10 3740 11920 0,05 28436 10 postgres
359 10 2816 10896 0,00 30152 10 postgres

PS C:\Windows\system32> psql -U postgres
Пароль пользователя postgres:
psql (17.2)
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=# SELECT * FROM accounts;
 id | balance
----+-----
  1 |    1000
  2 |    1800
  3 |    2950
(3 æ€Ёюьш)

postgres=#
```

7 Создание новой БД

Создать новую БД (при необходимости можно явно указать владельца):

```
CREATE DATABASE test_db;
```

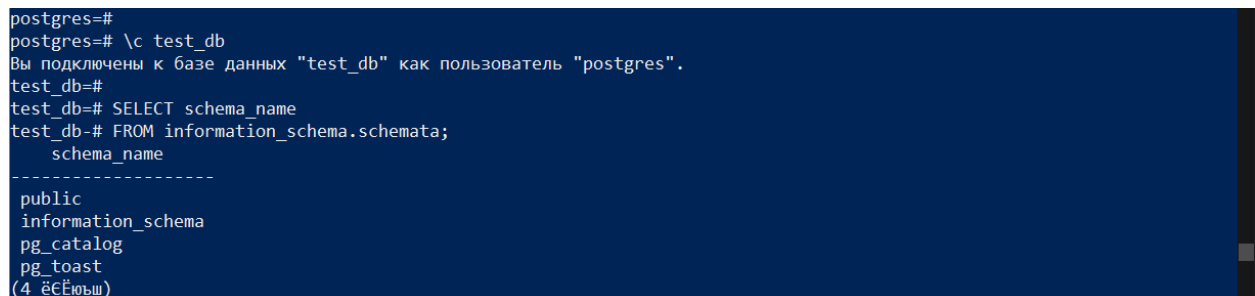
Посмотрим владельца:

```
SELECT d.datname, r.rolname AS owner
FROM pg_database d
JOIN pg_roles r ON d.datdba = r.oid
WHERE d.datname = 'test_db';
```



```
Администратор: Windows PowerShell
postgres=# CREATE DATABASE test_db;
CREATE DATABASE
postgres=# SELECT d.datname, r.rolname AS owner
postgres=# FROM pg_database d
postgres=# JOIN pg_roles r ON d.datdba = r.oid
postgres=# WHERE d.datname = 'test_db';
 datname | owner
-----+-----
 test_db | postgres
(1 ð€Ёюþ)
```

Подключимся к новой БД и проверим, какие в ней есть схемы.



```
postgres=#
postgres=# \c test_db
Вы подключены к базе данных "test_db" как пользователь "postgres".
test_db=#
test_db=# SELECT schema_name
test_db=# FROM information_schema.schemata;
 schema_name
-----
 public
 information_schema
 pg_catalog
 pg_toast
(4 ð€Ёюш)
```

В новой БД есть 4 схемы, 3 из которых – для системных объектов.

Посмотрим, какие в `public` схеме есть таблицы.



```
test_db=#
test_db=# SELECT table_name
test_db=# FROM information_schema.tables
test_db=# WHERE table_schema = 'public'
test_db=# AND table_type = 'BASE TABLE';
 table_name
-----
(0 ð€Ёюþ)
```

Ожидаемо, так как БД новая, в паблик-схеме пока нет никаких таблиц.

8 Создание пробных объектов

Создадим таблицу, и добавим в нее одну запись.

```
test_db=#
test_db=# CREATE TABLE test_table (id SERIAL PRIMARY KEY, name TEXT);
CREATE TABLE
test_db=# INSERT INTO test_table (name) VALUES ('Example');
INSERT 0 1
test_db=# SELECT * FROM test_table;
 id | name
----+-----
  1 | Example
(1 ð€Ёюþ)
```

Создадим схему, в ней – таблицу, добавим в таблицу запись.

```
test_db=#
test_db=# CREATE SCHEMA schema AUTHORIZATION postgres;
CREATE SCHEMA
test_db=# CREATE TABLE schema.test (id SERIAL PRIMARY KEY, name TEXT);
CREATE TABLE
test_db=# INSERT INTO schema.test (name) VALUES ('Abobus');
INSERT 0 1
test_db=# SELECT * FROM schema.test;
 id | name
----+-----
  1 | Abobus
(1 ð€Ёюþ)
```

9 Без доступа к старой

Попробуем получить доступ к таблице `account`, созданной в одной из предыдущих работ.

```
test_db=#
test_db=# SELECT * FROM account;
ОШИБКА: отношение "account" не существует
СТРОКА 1: SELECT * FROM account;
      ^

test_db=#
test_db=# SELECT * FROM public.account;
ОШИБКА: отношение "public.account" не существует
СТРОКА 1: SELECT * FROM public.account;
      ^

test_db=#
```

Безуспешно: в данной БД такой таблицы не существует.

10 Без доступа из старой

Подключимся к `postgres` и попробуем получить доступ к одной из новых таблиц.

Безуспешно: мы не можем обращаться к объекту в базе данных, не будучи подключенными к ней.


```
test_db=#
test_db=# \c postgres
Вы подключены к базе данных "postgres" как пользователь "postgres".
postgres=#
postgres=# SELECT * FROM test_table;
ОШИБКА: отношение "test_table" не существует
СТРОКА 1: SELECT * FROM test_table;
                        ^

postgres=#
postgres=# SELECT * FROM public.test_table;
ОШИБКА: отношение "public.test_table" не существует
СТРОКА 1: SELECT * FROM public.test_table;
                        ^

postgres=#
```

Заключение

В ходе работы успешно выполнены задачи по управлению кластером баз данных PostgreSQL и проверке его работоспособности.

Выяснено местоположение кластера баз данных (`C:/PostgreSQL/data`) и проанализировано его содержимое, включая конфигурационные файлы и служебные данные.

Определена командная строка запуска PostgreSQL и выполнено штатное завершение работы сервера. Сервер успешно перезапущен вручную и проверен на работоспособность.

Создана новая база данных, проверено её содержание и подтверждена изоляция между базами данных. Пробные объекты, созданные в новой базе, оказались недоступны из других баз, что соответствует принципам PostgreSQL.

Данная работа позволила закрепить навыки по настройке, управлению и проверке изоляции данных в PostgreSQL. Все задачи выполнены успешно.

Приложение

- отчет (docx);
- отчет (pdf).