

Лабораторная работа №2-3: «Базовые задачи администрирования СУБД»

Выполнил студент группы Б20-505, Ландаков Павел

Выяснить, в каком месте файловой системы расположен установленный в предыдущих работах кластер баз данных PostgreSQL

Все исполняемые файлы:

```
admin@fhm5767oq8ffdmhua7r8:~$ pg_config --bindir
>>> /usr/lib/postgresql/14/bin
```

Все данные самой БД:

```
lab23> SHOW data_directory
+-----+
| data_directory          |
|-----|
| /var/lib/postgresql/14/main |
+-----+
```

Выяснить, какие файлы хранятся в директории кластера базы данных

```
postgres@fhm5767oq8ffdmhua7r8:~/14$ tree main/ -L 1
main/
├── base
├── global
├── pg_commit_ts
├── pg_dynshmem
├── pg_logical
├── pg_multixact
├── pg_notify
├── pg_replslot
├── pg_serial
├── pg_snapshots
├── pg_stat
├── pg_stat_tmp
├── pg_subtrans
├── pg_tblspc
├── pg_twophase
└── PG_VERSION
```

```
└── pg_wal
└── pg_xact
└── postgresql.auto.conf
└── postmaster.opts
└── postmaster.pid
```

- **base**: Хранит данные всех пользовательских баз данных.
- **global**: Содержит глобальные данные кластера.
- **pg_commit_ts**: Информация о временных метках фиксации транзакций.
- **pg_dynshmem**: Для динамической разделяемой памяти.
- **pg_logical**: Информация для логической репликации.
- **pg_multixact**: Информация о мульти-транзакциях.
- **pg_notify**: Информация о NOTIFY командах.
- **pg_replslot**: Информация о слотах репликации.
- **pg_serial**: Информация о сериализации транзакций.
- **pg_snapshots**: Экспортированные снимки данных.
- **pg_stat** и **pg_stat_tmp**: Статистика работы сервера.
- **pg_subtrans**: Управление подтранзакциями.
- **pg_tblspc**: Ссылки на таблицы в пространствах таблиц.
- **pg_twophase**: Данные о двухфазных фиксациях транзакций.
- **PG_VERSION**: Версия PostgreSQL.
- **pg_wal**: Журналы записи операций (WAL).
- **pg_xact**: Файлы журналов транзакций.
- **postgresql.auto.conf**: Параметры конфигурации через ALTER SYSTEM.
- **postmaster.opts** и **postmaster.pid**: Опции запуска и идентификатор процесса.

Выяснить, какой командной строкой запущен экземпляр PostgreSQL

```
admin@fhm5767oq8ffd़hua7r8:/usr/lib/postgresql/14$ ps -ef | grep postgres
postgres      4280      1  0 03:19 ?        00:00:00
/usr/lib/postgresql/14/bin/postgres -D /var/lib/postgresql/14/main -c
config_file=/etc/postgresql/14/main/postgresql.conf
```

Выполнить штатное завершение работы сервера PostgreSQL

```
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
  Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
  Active: active (exited) since Thu 2024-01-18 03:19:56 UTC; 15min ago
    Main PID: 4308 (code=exited, status=0/SUCCESS)
      CPU: 1ms

Jan 18 03:19:56 fhm5767oq8ffdmhua7r8 systemd[1]: Starting PostgreSQL RDBMS...
Jan 18 03:19:56 fhm5767oq8ffdmhua7r8 systemd[1]: Finished PostgreSQL RDBMS.
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ sudo systemctl stop postgresql
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
  Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Thu 2024-01-18 03:35:28 UTC; 1s ago
    Main PID: 4308 (code=exited, status=0/SUCCESS)
      CPU: 1ms

Jan 18 03:19:56 fhm5767oq8ffdmhua7r8 systemd[1]: Starting PostgreSQL RDBMS...
Jan 18 03:19:56 fhm5767oq8ffdmhua7r8 systemd[1]: Finished PostgreSQL RDBMS.
Jan 18 03:35:28 fhm5767oq8ffdmhua7r8 systemd[1]: postgresql.service: Deactivated successfully.
Jan 18 03:35:28 fhm5767oq8ffdmhua7r8 systemd[1]: Stopped PostgreSQL RDBMS.
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ █
```

Вновь запустить экземпляр PostgreSQL вручную

```
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
  Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Thu 2024-01-18 03:35:28 UTC; 40s ago
    Main PID: 4308 (code=exited, status=0/SUCCESS)
      CPU: 1ms

Jan 18 03:19:56 fhm5767oq8ffdmhua7r8 systemd[1]: Starting PostgreSQL RDBMS...
Jan 18 03:19:56 fhm5767oq8ffdmhua7r8 systemd[1]: Finished PostgreSQL RDBMS.
Jan 18 03:35:28 fhm5767oq8ffdmhua7r8 systemd[1]: postgresql.service: Deactivated successfully.
Jan 18 03:35:28 fhm5767oq8ffdmhua7r8 systemd[1]: Stopped PostgreSQL RDBMS.
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ sudo systemctl start postgresql
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
  Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
  Active: active (exited) since Thu 2024-01-18 03:36:14 UTC; 2s ago
    Process: 5448 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 5448 (code=exited, status=0/SUCCESS)
      CPU: 1ms

Jan 18 03:36:14 fhm5767oq8ffdmhua7r8 systemd[1]: Starting PostgreSQL RDBMS...
Jan 18 03:36:14 fhm5767oq8ffdmhua7r8 systemd[1]: Finished PostgreSQL RDBMS.
admin@fhm5767oq8ffdmhua7r8:/usr/lib/postgresql/14$ █
```

Подключиться к экземпляру и проверить его работоспособность

Все таблицы и данные сохранились:

```
admin@fhm5767qq8ffdmhua7r8:~$ pgcli -U admin -d lab23
Load your password from keyring returned:
No recommended backend was available. Install a recommended 3rd party backend package; or, install the keyrings.alt package if you want to use the non-recommended backends. See https://pypi.org/project/keyring for details.
To remove this message do one of the following:
- prepare keyring as described at: https://keyring.readthedocs.io/en/stable/
- uninstall keyring: pip uninstall keyring
- disable keyring in our configuration: add keyring = False to [main]
Server: PostgreSQL 14.10 (Ubuntu 14.10-0ubuntu0.22.04.1)
Version: 3.3.1
Home: http://pgcli.com
Lab23> \d
+-----+ | Schema | Name | Type | Owner |
+-----+
| public | locations | table | admin |
| public | users | table | admin |
| public | weatherconditions | table | admin |
| public | weatherhistory | table | admin |
+-----+
SELECT 4
Time: 0.011s
Lab23> 
```

Создать новую базу данных в кластере. Кто её владелец? Какие объекты в ней содержатся?

```
Lab23> CREATE DATABASE mydb;
CREATE DATABASE
Time: 0.108s
Lab23> 
```

| По умолчанию владельцем базы данных

будет тот, от чьего имени была выполнена команда, иначе можно указать другого владельца при помощи OWNER

Базово в свежей базе данных хранятся:

- Системные каталоги
- Системные таблицы
- Шаблоны template0 и template1

Подключиться к новой базе данных и создать в ней несколько пробных объектов

Подключение к mydb и создание пары новых таблиц:

```
lab23> \c mydb
You are now connected to database "mydb" as user "admin"
Time: 0.006s
mydb> CREATE TABLE Customers (
    CustomerID SERIAL PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    Phone VARCHAR(20)
);
CREATE TABLE Orders (
    OrderID SERIAL PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
CREATE TABLE
CREATE TABLE
Time: 0.032s
```

Убедиться, что из новой базы данных нет доступа к исходной

```
mydb> select * from locations;
relation "locations" does not exist
LINE 1: select * from locations
          ^
Time: 0.001s
mydb>
```

Убедиться, что из исходной базы данных нет доступа к новой

```
lab23> \d
+-----+-----+-----+-----+
| Schema | Name      | Type   | Owner  |
+-----+-----+-----+-----+
| public | locations | table  | admin   |
| public | users     | table  | admin   |
| public | weatherconditions | table  | admin   |
| public | weatherhistory  | table  | admin   |
+-----+-----+-----+-----+
SELECT 4
Time: 0.011s
lab23> select * from customers;
relation "customers" does not exist
LINE 1: select * from customers
          ^
Time: 0.001s
```

Заключение

В данной лабораторной работе было проведено знакомство с концептом "контейнерных баз данных", были освоены базовые задачи администрирования СУБД и изучены механизмы управления базами данных.