

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «ЛИПЕЦКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт

компьютерных наук

Кафедра

автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №7

По дисциплине "Операционные системы Linux"

На тему "Создание дампа БД и восстановление"

Студент

ПИ-22-1

подпись, дата

Кистерёв В.А.

Руководитель

канд.техн.наук, доцент

ученая степень, ученое звание

подпись, дата

Кургасов В.В.

Липецк, 2024 г.

Оглавление

Цель работы	3
Ход работы	4
Вывод.....	7

Цель работы

Освоение методов резервного копирования и восстановления баз данных.

Ход работы

Для выполнения лабораторной работы установим PostgreSQL:

– `sudo apt install postgresql postgresql-contrib`

Запустим postgresql:

– `sudo systemctl start postgresql`

Командой `sudo systemctl status postgresql` проверим статус сервера (рисунок 1).

```
user@labs:~$ sudo systemctl status postgresql
• postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; preset: enabled)
   Active: active (exited) since Thu 2025-01-02 16:42:21 UTC; 1min 13s ago
   Main PID: 2361 (code=exited, status=0/SUCCESS)
     CPU: 1ms

Jan 02 16:42:21 labs systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...
Jan 02 16:42:21 labs systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
```

Рисунок 1 – Проверка статуса сервера

Командой `sudo su - postgres` авторизуемся под учетной записью пользователя postgres. Запустим инструмент командной строки – `psql`, создадим новую роль и базу данных `test_db` (рисунок 2).

```
postgres@labs:~$ psql
psql (15.10 (Debian 15.10-0+deb12u1))
Type "help" for help.

postgres=# CREATE ROLE test WITH LOGIN PASSWORD '1234';
CREATE ROLE
postgres=# CREATE DATABASE test_db OWNER test;
CREATE DATABASE
postgres=# ALTER ROLE test CREATEDB;
ALTER ROLE
postgres=# ALTER ROLE test CREATEROLE;
ALTER ROLE
postgres=# \q
postgres@labs:~$
```

Рисунок 2 – Процесс создания базы данных

Командой `systemctl restart postgresql` перезапустим postgres.

Командой `\l` убедимся в том, что база данных `test_db` создана (рисунок 3).

```
postgres=# \l
```

Name	Owner	Encoding	Collate	Ctype	ICU Locale	Locale Provider	Access privileges
postgres	postgres	UTF8	C.UTF-8	C.UTF-8		libc	
template0	postgres	UTF8	C.UTF-8	C.UTF-8		libc	=c/postgres +
template1	postgres	UTF8	C.UTF-8	C.UTF-8		libc	=c/postgres +
test_db	test	UTF8	C.UTF-8	C.UTF-8		libc	postgres=Ct/postgres

(4 rows)

Рисунок 3 – Просмотр списка баз данных

Командой \с <название БД> подключимся к test_db.

Создадим таблицу books:

```
CREATE TABLE books (  
    id SERIAL PRIMARY KEY,  
    title VARCHAR(100),  
    author VARCHAR(100));
```

Добавим несколько записей в таблицу books:

```
INSERT INTO books (title, author) VALUES  
( '1984', 'George Orwell'),  
( 'To Kill a Mockingbird', 'Harper Lee');
```

На рисунке 4 представлено содержание таблицы books/

```
test_db=# SELECT * FROM books;  
 id |          title          | author  
-----+-----+-----  
  1 | 1984                    | George Orwell  
  2 | To Kill a Mockingbird   | Harper Lee  
(2 rows)
```

Рисунок 4 – Содержание таблицы books

Для создания резервной копии базы данных используем утилиту pg_dump.

Выполним команду: pg_dump -U postgres -F c -f dump.tar test_db ("-F c" – формат архива (сжатый дамп), "-f dump.tar" – имя файла, в который сохраняется дамп).

Пример использования команды представлен на рисунке 5.

```
postgres@labs:~$ pg_dump -U postgres -F c -f dump.tar test_db  
postgres@labs:~$ ls -l  
total 8  
drwxr-xr-x 3 postgres postgres 4096 Jan  2 16:42 15  
-rw-r--r-- 1 postgres postgres 2734 Jan  2 19:18 dump.tar  
postgres@labs:~$
```

Рисунок 5 – Создание дампа

Удалим созданную ранее базу данных test_db (рисунок 6).

```
postgres=# DROP DATABASE test_db;  
DROP DATABASE  
postgres=# \l  
  
          List of databases  
 Name      | Owner   | Encoding | Collate | Ctype   | ICU Locale | Locale Provider | Access privileges  
-----+-----+-----+-----+-----+-----+-----+-----  
 postgres  | postgres | UTF8     | C.UTF-8 | C.UTF-8 |             | libc            |  
 template0 | postgres | UTF8     | C.UTF-8 | C.UTF-8 |             | libc            | =c/postgres +  
                                     postgres=CTc/postgres  
 template1 | postgres | UTF8     | C.UTF-8 | C.UTF-8 |             | libc            | =c/postgres +  
                                     postgres=CTc/postgres  
(3 rows)
```

Рисунок 6 – Удаление test_db

Снова создадим базу данных test_db, восстановим данные из дампа с помощью утилиты pg_restore:

– pg_restore -U postgres -d test_db -F c dump.tar

Восстановление и проверка данных представлены на рисунке 7.

```
postgres=# CREATE DATABASE test_db OWNER postgres;
CREATE DATABASE
postgres=# \q
postgres@labs:~$ pg_restore -U postgres -d test_db -F c dump.tar
postgres@labs:~$ psql
psql (15.10 (Debian 15.10-0+deb12u1))
Type "help" for help.

postgres=# \c test_db
You are now connected to database "test_db" as user "postgres".
test_db=# \dt
      List of relations
Schema | Name  | Type  | Owner
-----+-----+-----+-----
public | books | table | postgres
(1 row)

test_db=# SELECT * FROM books;
 id | title | author
----+-----+-----
  1 | 1984  | George Orwell
  2 | To Kill a Mockingbird | Harper Lee
(2 rows)
```

Рисунок 7 – Восстановление данных

Вывод

В ходе выполнения данной лабораторной работы были приобретены навыки создания резервных копий (дампов) баз данных и их восстановлений с использованием утилит `pg_dump` и `pg_restore`.