

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБ-
РАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВА-
НИЯ «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

Клименко Н.Д.

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

В рамках лабораторной работы был использован сервер с установленной системой управления базами данных PostgreSQL. Вся работа проводилась в среде PostgreSQL, включая создание базы данных, таблиц, добавление записей, создание резервной копии (дампа) и ее восстановление.

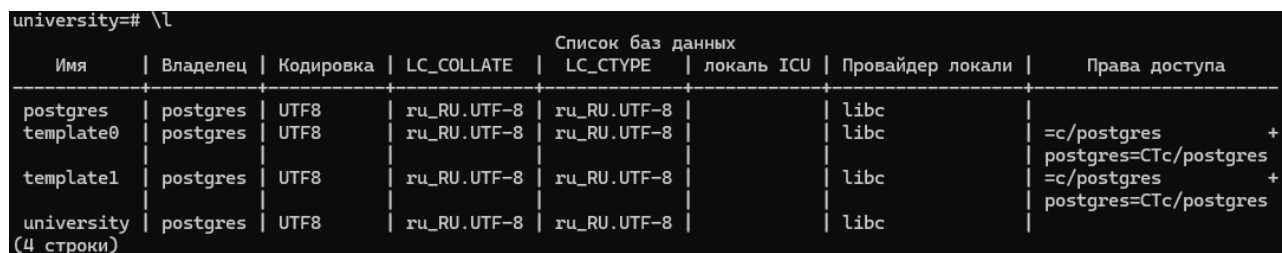
Для начала работы необходимо авторизоваться в учетной записи пользователя PostgreSQL. Это можно сделать следующими командами:

- `sudo -i -u postgres;`
- `psql` (запускает инструмент командной строки `psql` для работы с базами данных).

Для хранения данных создадим новую базу данных `university` с помощью SQL-запроса:

- `CREATE DATABASE university;`

Результат создания базы данных представлен на рисунке 1.



Имя	Владелец	Кодировка	LC_COLLATE	Список баз данных		Провайдер локали	Права доступа
				LC_STYPE	локаль ICU		
postgres	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	
template0	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	=c/postgres +
template1	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	=c/postgres +
university	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8		libc	postgres=CtC/postgres

(4 строки)

Рисунок 1 – Базы данных

Для работы с созданной базой данных подключимся к ней, используя команду `\c <название БД>`.

В базе данных `university` создадим таблицу `students`, которая будет содержать данные о студентах. Для этого выполним следующий SQL-запрос:

```
CREATE TABLE students (  
    id SERIAL PRIMARY KEY,  
    full_name VARCHAR(50),  
    student_card VARCHAR(11),  
    scholarship INT  
);
```

Добавим данные о студентах в таблицу:

```
INSERT INTO students (full_name, student_card, scholarship)
VALUES
('Ivan Ivanov', 's1221060101', 5194),
('Petr Petrov', 's1221060202', 3800);
```

Проверим содержимое созданной таблицы, результат представлен на рисунке 2.

```
university=# \c university
Вы подключены к базе данных "university" как пользователь "postgres".
university=# SELECT * FROM students;
 id | full_name | student_card | scholarship
-----+-----+-----+-----
  1 | Ivan Ivanov | s1221060101 |      5194
  2 | Petr Petrov | s1221060202 |      3800
(2 строки)
```

Рисунок 2 – Таблица students

Для создания резервной копии базы данных используем утилиту pg_dump.

Выполним команду:

```
- pg_dump -U postgres -F c -f dump.tar university
```

Где "-F c" – формат архива (сжатый дамп) и "-f dump.tar" – имя файла, в который сохраняется дамп.

Результат представлен на рисунке 3.

```
postgres@deb-server:~$ pg_dump -U postgres -F c -f dump.tar university
postgres@deb-server:~$ ls -l
итого 8
drwxr-xr-x 3 postgres postgres 4096 дек 29 00:08 15
-rw-r--r-- 1 postgres postgres 2871 дек 29 00:15 dump.tar
postgres@deb-server:~$ |
```

Рисунок 3 – Создание дампа

Создадим новую базу данных university_new для восстановления данных:

```
- CREATE DATABASE university_new;
```

После этого восстановим данные из дампа с помощью утилиты pg_restore:

```
- pg_restore -U postgres -d university_new -F c dump.tar
```

Подключаемся к созданной базе данных (\c) и смотрим содержимое (\dt). Также, чтобы убедиться в корректном восстановлении, выведем содержимое таблицы students. Как видно, содержимое таблицы students базы данных

university_new идентично содержимому таблицы students базы данных university, от куда можно сделать вывод, что восстановление прошло удачно. Данные действия показаны на рисунке 4.

```
postgres@deb-server:~$ pg_restore -U postgres -d university_new -F c dump.tar
postgres@deb-server:~$ psql
psql (15.10 (Debian 15.10-0+deb12u1))
Введите "help", чтобы получить справку.

postgres=# \c university_new
Вы подключены к базе данных "university_new" как пользователь "postgres".
university_new=# \dt
          Список отношений
Схема |   Имя   | Тип  | Владелец
-----+-----+-----+-----
public | students | таблица | postgres
(1 строка)

university_new=# SELECT * FROM students;
 id | full_name | student_card | scholarship
----+-----+-----+-----
  1 | Ivan Ivanov | s1221060101 |         5194
  2 | Petr Petrov | s1221060202 |         3800
(2 строки)

university_new=# |
```

Рисунок 4 – Восстановление и проверка данных

Вывод

В ходе выполнения данной лабораторной работы я приобрел навыки в создании резервных копий (дампов) баз данных с использованием утилиты `pg_dump`, а также в восстановлении баз данных из дампов с помощью утилиты `pg_restore`.