

# Липецкий государственный технический университет

Отчет по Лабораторной работе № 8  
по дисциплине «Операционная система Linux»  
на тему «Создание дампа БД и восстановление в Postgresql»

Студент

Руководитель

доцент, к.п.н.

учёная степень, учёное звание

подпись, дата

подпись, дата

Елфимова Д.А.

фамилия, инициалы

Кургасов В.В.

фамилия, инициалы

Липецк 2019 г.

## **Задание**

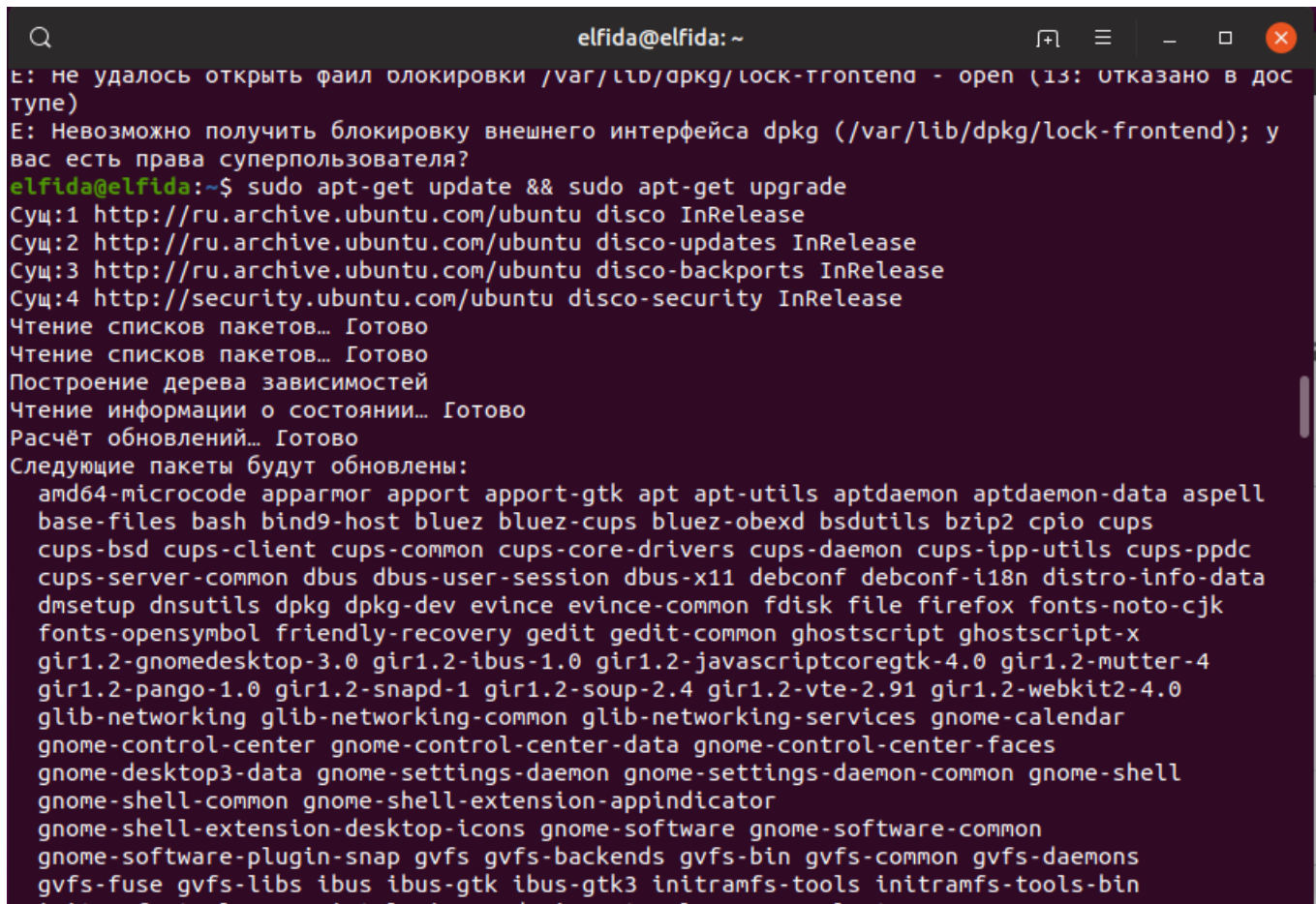
PostgreSQL является современной системой управления базами данных, часто используемая для хранения и обработки информации, связанной с веб-сайтами или сторонними приложениями. У любой базы данных, важно вовремя реализовать резервное копирование, чтобы избежать возможную потерю данных. В данной лабораторной необходимо реализовать дамп базы данных и ее восстановление.

## Оглавление

1. Создание дампа с нуля . . . . .	4
2. Восстановление БД из дампа . . . . .	6
Заключение . . . . .	8

## 1. Создание дампа с нуля

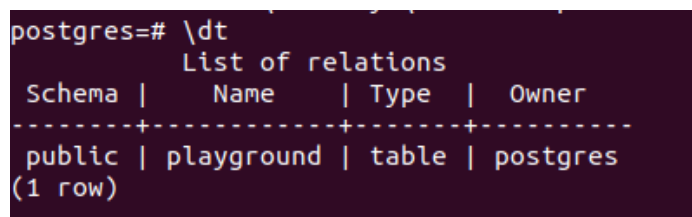
1. Обновить список репозиторий и систему, а также произвести перезагрузку системы



```
elfida@elfida: ~  
E: не удалось открыть файл блокировки /var/lib/dpkg/lock-frontend - open (13: отказано в доступе)  
E: Невозможно получить блокировку внешнего интерфейса dpkg (/var/lib/dpkg/lock-frontend); у вас есть права суперпользователя?  
elfida@elfida:~$ sudo apt-get update && sudo apt-get upgrade  
Суц:1 http://ru.archive.ubuntu.com/ubuntu disco InRelease  
Суц:2 http://ru.archive.ubuntu.com/ubuntu disco-updates InRelease  
Суц:3 http://ru.archive.ubuntu.com/ubuntu disco-backports InRelease  
Суц:4 http://security.ubuntu.com/ubuntu disco-security InRelease  
Чтение списков пакетов... Готово  
Чтение списков пакетов... Готово  
Построение дерева зависимостей  
Чтение информации о состоянии... Готово  
Расчёт обновлений... Готово  
Следующие пакеты будут обновлены:  
amd64-microcode apparmor appport appport-gtk apt apt-utils aptdaemon aptdaemon-data aspell  
base-files bash bind9-host bluez bluez-cups bluez-obexd bsutils bzip2 cpio cups  
cups-bsd cups-client cups-common cups-core-drivers cups-daemon cups-ipp-utils cups-ppdc  
cups-server-common dbus dbus-user-session dbus-x11 debconf debconf-i18n distro-info-data  
dmsetup dnsutils dpkg dpkg-dev evince evince-common fdisk file firefox fonts-noto-cjk  
fonts-opensymbol friendly-recovery gedit gedit-common ghostscript ghostscript-x  
gir1.2-gnomedesktop-3.0 gir1.2-ibus-1.0 gir1.2-javascriptcoregtk-4.0 gir1.2-mutter-4  
gir1.2-pango-1.0 gir1.2-snapd-1 gir1.2-soup-2.4 gir1.2-vte-2.91 gir1.2-webkit2-4.0  
glib-networking glib-networking-common glib-networking-services gnome-calendar  
gnome-control-center gnome-control-center-data gnome-control-center-faces  
gnome-desktop3-data gnome-settings-daemon gnome-settings-daemon-common gnome-shell  
gnome-shell-common gnome-shell-extension-appindicator  
gnome-shell-extension-desktop-icons gnome-software gnome-software-common  
gnome-software-plugin-snap gvfs gvfs-backends gvfs-bin gvfs-common gvfs-daemons  
gvfs-fuse gvfs-libs ibus ibus-gtk ibus-gtk3 initramfs-tools initramfs-tools-bin
```

Рисунок 1. Обновления

2. Реализовать создание таблицы, добавление и удаление строк



```
postgres=# \dt  
List of relations  
Schema | Name | Type | Owner  
-----+-----+-----+-----  
public | playground | table | postgres  
(1 row)
```

Рисунок 2. Создание таблицы

```

postgres=# INSERT INTO playground (type, color, location, install_date) VALUES (
'slide', 'blue', 'south', '2014-04-28');
INSERT 0 1
postgres=# INSERT INTO playground (type, color, location, install_date) VALUES (
'swing', 'yellow', 'northwest', '2010-08-16');
INSERT 0 1
postgres=# SELECT * FROM playground
postgres=# ;

```

equip_id	type	color	location	install_date
1	slide	blue	south	2014-04-28
2	swing	yellow	northwest	2010-08-16

```

(2 rows)

```

Рисунок 3. Добавление строк

```

postgres=# DELETE FROM playground WHERE type = 'slide';
DELETE 1
postgres=# SELECT * FROM playground
postgres=# ;

```

equip_id	type	color	location	install_date
2	swing	yellow	northwest	2010-08-16

```

(1 row)

```

Рисунок 4. Удаление строк

### 3. Создание новой бд и выполнение ее дампа.

Дамп реализуется следующим образом:

```
pg_dump name_of_database > backup_file.bak
```

```

postgres=# CREATE DATABASE my_postgres_db OWNER pgelf;
CREATE DATABASE
postgres=# \q
postgres@elfida:~$ exit
выход
elfida@elfida:~$ sudo su - pgelf
pgelf@elfida:~$ psql my_postgres_db
psql (11.5 (Ubuntu 11.5-0ubuntu0.19.04.1))
Type "help" for help.

my_postgres_db=# CREATE TABLE playgr (
my_postgres_db(#      equip_id serial PRIMARY KEY,
my_postgres_db(#      type varchar (50) NOT NULL,
my_postgres_db(#      color varchar (25) NOT NULL,
my_postgres_db(#      location varchar(25) check (location in ('north', 'south', 'w
est', 'east', 'northeast', 'southeast', 'southwest', 'northwest')),
my_postgres_db(#      install_date date
my_postgres_db(# );
CREATE TABLE

```

Рисунок 5. Создание бд и таблицы в ней

```
pgelf@elfida:~$ pg_dump my_postgres_db > my_dump.bak
pgelf@elfida:~$ mc

pgelf@elfida:~$ ls -li
итого 16
523266 -rw-r--r-- 1 pgelf pgelf 8980 дек 17 01:25 examples.desktop
523273 -rw-rw-r-- 1 pgelf pgelf 2384 дек 17 01:48 my_dump.bak
```

Рисунок 6. Выполнение дампа и просмотр содержимого директория

## 2. Восстановление БД из дампа

1. Удаление БД и ее восстановление Восстановление БД реализуется следующим образом:

```
createdb empty_database
psql empty_database < backup_file.bak
```

```
pgelf@elfida:~$ dropdb my_postgres_db
pgelf@elfida:~$ psql my_postgres_db
psql: FATAL: database "my_postgres_db" does not exist
pgelf@elfida:~$ createdb my_postgres_db
pgelf@elfida:~$ psql my_postgres_db < my_dump.bak
SET
SET
SET
SET
SET
  set_config
-----
(1 row)

SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE
ALTER SEQUENCE
ALTER TABLE
COPY 0
  setval
-----
      1
(1 row)
```

Рисунок 7. Удаление БД и ее восстановление

2. Проверка

```
pgelf@elfida:~$ psql my_postgres_db
psql (11.5 (Ubuntu 11.5-0ubuntu0.19.04.1))
Type "help" for help.

my_postgres_db=# \d
               List of relations
Schema |           Name           |  Type   | Owner
-----+-----+-----+-----
public | playgr                   | table    | pgelf
public | playgr_equip_id_seq      | sequence | pgelf
(2 rows)
```

Рисунок 8. Проверка

## **Заключение**

В ходе данной лабораторной работы были изучены или повторно рассмотрены некоторые команды ОС Linux, было проведено ознакомление и анализ рекомендованной литературы, а также информации об использовании Postgresql и дампа баз данных.