Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ОТЧЕТ

к лабораторной работе по дисциплине:

|  |
| --- |
| **Администрирование информационных систем** |
| Администрирование баз данных в UbuntuServer |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСМб-19-1 |  |  |  | Малиновцев И.А. |
|  |  | подпись |  | Фамилия И.О |
| Проверил: |  |  |  |  | Хритова М.А. |
|  | должность |  | подпись |  | Фамилия И.О |

Иркутск 2023 г.

Содержание

[Содержание 2](#_Toc127729664)

[1 Введение 3](#_Toc127729665)

[2 Основные понятия СУБД MySQL и PostgreSQL 4](#_Toc127729666)

[3 Установка СУБД 6](#_Toc127729667)

[3.1 Установка MySQL 6](#_Toc127729668)

[3.2 Установка PostgreSQL 7](#_Toc127729669)

[4 Настройка СУБД 8](#_Toc127729670)

[4.1 Настройка MySQL 8](#_Toc127729671)

[4.1.1 Конфигурирование файлов 8](#_Toc127729672)

[4.1.2 Создание базы данных 9](#_Toc127729673)

[4.1.3 Создание роли 11](#_Toc127729674)

[4.2 Настройка PostgreSQL 14](#_Toc127729675)

[4.2.1 Конфигурирование базы данных 14](#_Toc127729676)

[4.2.2 Создание базы данных 14](#_Toc127729677)

[4.2.3 Создание роли 16](#_Toc127729678)

[5 Заключение 19](#_Toc127729679)

1 Введение

**Цель работы:** получить начальные навыки практического администрирования базы данных в операционной системе UbuntuServer, изучить СУБД MySQL и PostgreSQL.

**Требования к отчетным документам:** отчетный документ представляется в печатном или электронном виде, в форматах pdf, doc(x).

Отчет должен содержать следующие главы:

1. Введение
2. Теоретическая часть. Основные понятия СУБД MySQL и PostgreSQL.
3. Практическая часть. Установка СУБД MySQL и PostgreSQL в ОС UbuntuServer (привести использованные команды и скриншоты).
4. Практическая часть. Настройка PostgreSQL. Работа с базой данных (привести использованные команды и скриншоты).
5. Заключение (полученные результаты).

Объем отчетного документа 7 - 10 страниц.

**Задание:**

1. Самостоятельно найти информацию и изучить СУБД MySQL и PostgreSQL.
2. Прочитать и изучить информацию об установке СУБД MySQL и PostgreSQL для операционной системы UbuntuServer.

3. Инсталлировать MySQL и PostgreSQL в UbuntuServer.

4. Выполнить настройку PostgreSQL в UbuntuServer (выполнить создание роли, базы данных, таблиц, произвести добавление и удаление данных).

5. Написать отчет.

2 Основные понятия СУБД MySQL и PostgreSQL

MySQL — самая популярная в мире база данных с открытым исходным кодом.

База данных MySQL — это клиент-серверная система, состоящая из многопоточного SQL-сервера, поддерживающего различные серверные части, нескольких различных клиентских программ и библиотек, инструментов администрирования и широкого спектра прикладных программных интерфейсов (API).

Основные понятия:

**GNU General Public License** — лицензия на свободное программное обеспечение, созданная в рамках проекта GNU в 1988 г., по которой автор передаёт программное обеспечение в общественную собственность.

**Репликация** - механизм синхронизации содержимого нескольких копий объекта (например, содержимого базы данных).

**WAMP** - комплектация пакетов программ, под соответствующую ОС, обозначаемую первой буквой.

**AppServ** — пакет инструментов, который поможет использовать системы управления контентом.

**LAMP** — набор (комплекс) серверного программного обеспечения, широко используемый во Всемирной паутине.

**XAMPP** — кроссплатформенная сборка веб-сервера (развитие LAMP), содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

**VertrigoServ** — профессиональный пакет программ, предназначенных для установки и управления полноценного веб-сервера Apache на домашнем компьютере под управлением Microsoft Windows.

**MyISAM** — систем хранения данных в СУБД MySQL.

**InnoDB** — систем хранения данных в СУБД MySQL.

PostgreSQL — это реляционная СУБД с открытым кодом и свободным распространением. База данных Postgre базируется на стандартном языке запросов SQL, а точнее, на его расширении — процедурном языке PL/pgSQL. PostgreSQL работает в системах Unix (FreeBSD и Linux: дистрибутивах Ubuntu, Debian, CentOS) и Windows.

Основные возможности:

1)**Функции** — являются блоками кода, исполняемыми на сервере, а не на клиенте БД.

2)**Триггеры** — определяются как функции, инициируемые DML-операциями. Например, операция INSERT может запускать триггер, проверяющий добавленную запись на соответствия определённым условиям. При написании функций для триггеров могут использоваться различные языки программирования.

3)**Индексы** — в PostgreSQL имеется поддержка индексов следующих типов: B-дерево, хэш и т.д.

4)**Многоверсионность** — PostgreSQL поддерживает одновременную модификацию БД несколькими пользователями с помощью механизма Multiversion Concurrency Control (MVCC). Благодаря этому соблюдаются требования ACID и практически отпадает нужда в блокировках чтения.

5)**Пользовательские объекты** — PostgreSQL может быть расширен пользователем для собственных нужд практически в любом аспекте. есть возможность добавлять собственные преобразования типов; типы данных; домены (пользовательские типы с изначально наложенными ограничениями); функции; индексы; операторы (включая переопределение уже существующих); процедурные языки и т.д.

Основные понятия:

**AIX** - UNIX-подобная операционная система компании IBM.

**BSD-система** — система распространения программного обеспечения в исходных кодах, созданная для обмена опытом между учебными заведениями.

**HP-UX** — проприетарная версия операционной системы UNIX фирмы HP.

**IRIX** — операционная система, используемая на рабочих станциях и серверах фирмы Silicon Graphics (SGI).

**DML** — это семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных.

**B-дерево** - структура данных, дерево поиска. С точки зрения внешнего логического представления, сбалансированное, сильно ветвистое дерево. Часто используется для хранения данных во внешней памяти.

**Хэш** — это структура данных, реализующая интерфейс ассоциативного массива, а именно, она позволяет хранить пары (ключ, значение) и выполнять три операции: операцию добавление новой пары, операцию поиска и операцию удаления пары по ключу.

**Multiversion** **Concurrency** **Control** — механизм обеспечения параллельного доступа к БД, заключающийся в предоставлении каждому пользователю так называемого «снимка» БД, обладающего тем свойством, что вносимые пользователем изменения в БД невидимы другим пользователям до момента фиксации транзакции.

**ACID** — описывает требования к транзакционной системе, обеспечивающие более надежную и предсказуемую ее работу.

**Grant** – gредоставляет разрешения на защищаемый объект участнику. Общий подход заключается в том, чтобы предоставить <некоторые разрешения> на <некоторый объект> <некоторому пользователю, имени входа или группе> (GRANT…ON…TO).

3 Установка СУБД

3.1 Установка MySQL

Для установки MySQL применялась команда: sudo apt-get install mysql-server.

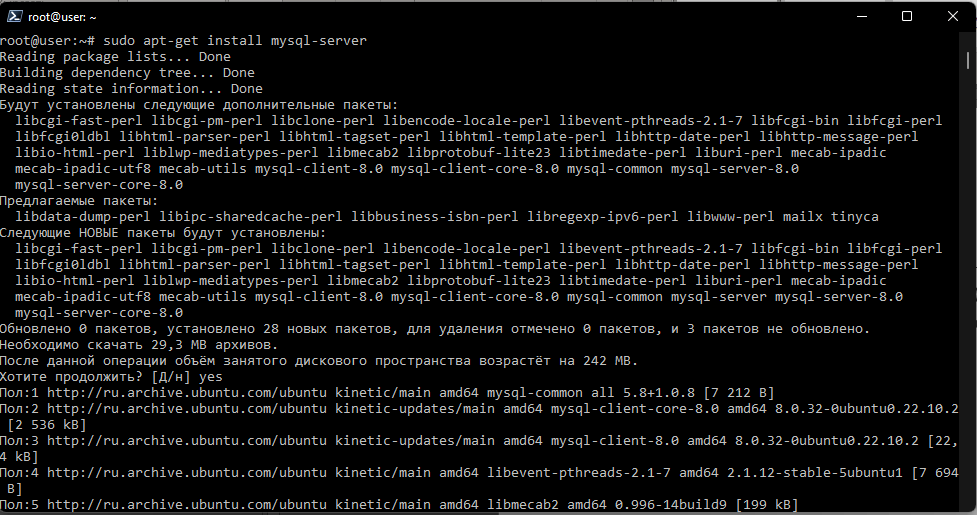


Рисунок 1 – Использование команды sudo apt-get install mysql-server

Для установки консольного клиента MySQL использовалась команда sudo apt-get install mysql-client.

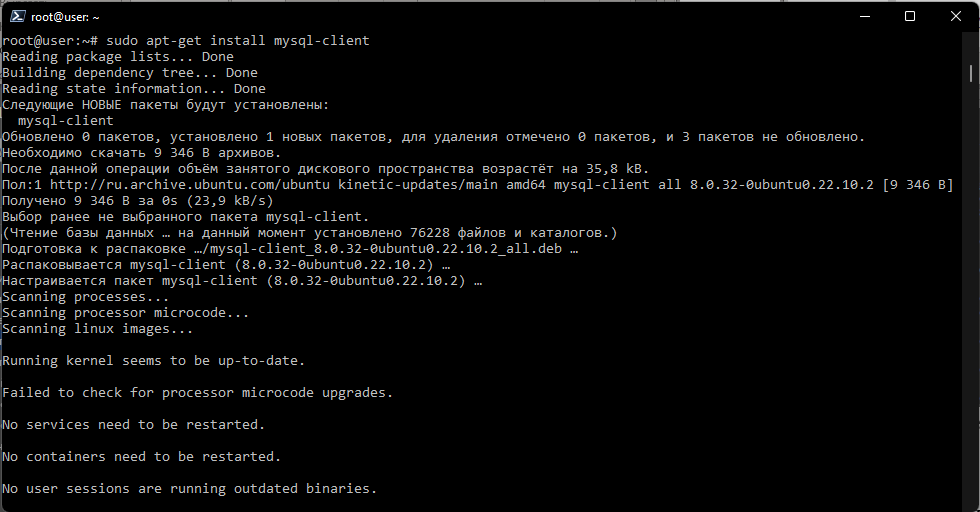


Рисунок 2 – Использование команды sudo apt-get install mysql-client

3.2 Установка PostgreSQL

Для установки PostgreSQL с набором часто используемых библиотек применялась команда sudo apt-get install postgresql postgresql-contrib.

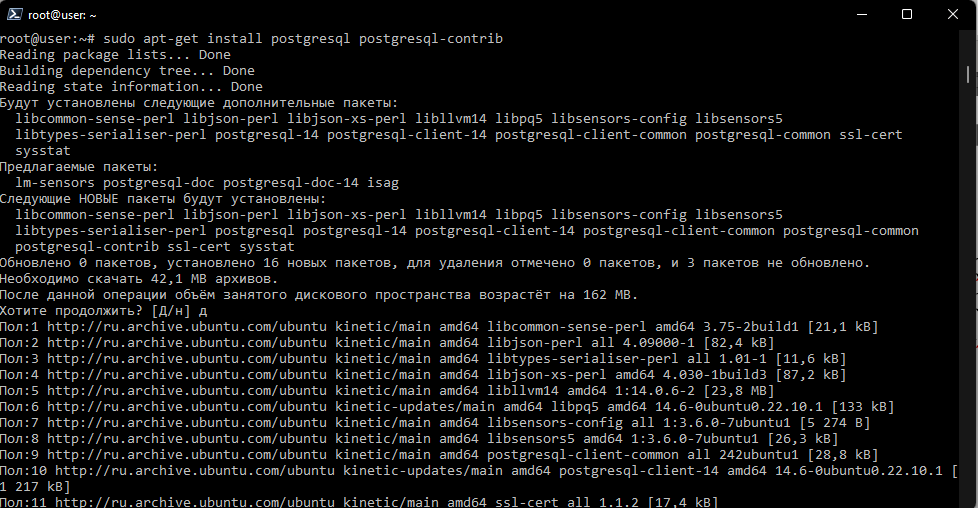


Рисунок 3 – Использование команды sudo apt-get install postgresql postgresql-contrib

4 Настройка СУБД

4.1 Настройка MySQL

4.1.1 Конфигурирование файлов

По умолчанию MySQL на платформе Ubuntu взаимодействует некорректно с кириллицей, так как установленной кодировкой по-умолчанию установлена latin1. Для изменения целевой кодировки по необходимо отредактировать файл my.cnf в операционной системе.

Первым этапом необходимо добавить в секцию [mysqld] следующие строки:

skip-character-set-client-handshake

character-set-server = utf8

init-connect='SET NAMES utf8'

collation-server=utf8\_general\_ci

Далее необходимо добавить в секциях [client] и [mysqldump] строчку

default-character-set=utf8

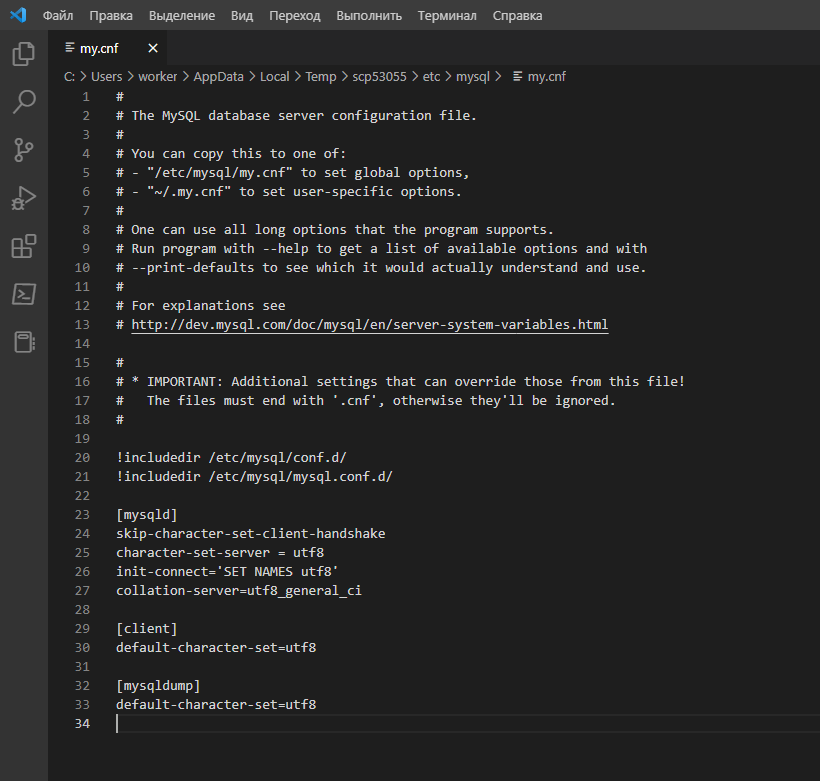


Рисунок 4 – Редактирование конфигурационного файла

Завершающим этапом выступает перезагрузка сервера MySQL командой sudo service mysql restart.

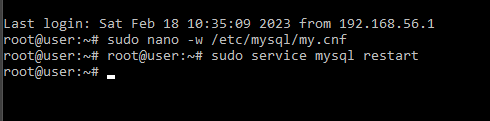


Рисунок 5 – Перезагрузка сервера

4.1.2 Создание базы данных

Для входа в оболочку MySQL используется команда sudo mysql.

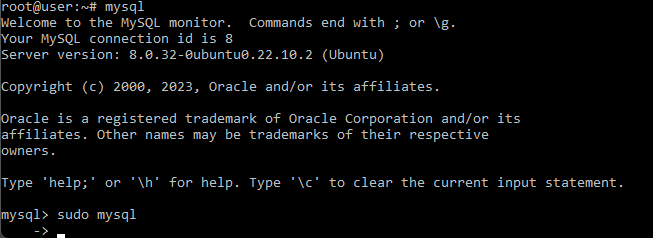


Рисунок 6 – Вход в оболочку MySQL

Чтобы создать новую базу данных MySQL используется команда CREATE DATABASE earthquakes\_database; earthquakes\_database в данной команде является названием создаваемой базы данных.

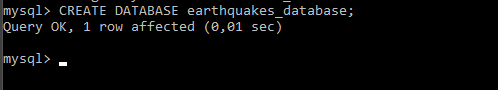


Рисунок 7 – Создание базы данных

Для просмотра всех существующих баз данных применяется команда SHOW DATABASES;

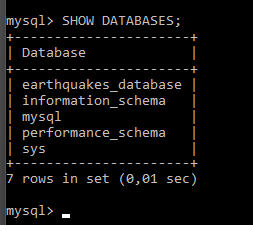


Рисунок 8 – Просмотр списка баз данных

Чтобы выбрать базу данных перед началом сеанса MySQL, используется команду USE earthquakes\_database;

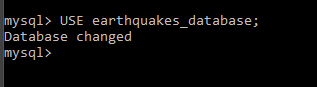


Рисунок 9 – Выбор базы данных

Создание таблицы осуществлялась командой

CREATE TABLE IF NOT EXISTS epicentres(id INT PRIMARY KEY AUTO\_INCREMENT, coordX int NOT NULL, coordY int NOT NULL, dateEpic VARCHAR(255) NOT NULL, isReal Boolean NOT NULL);

ALTER TABLE epicentres AUTO\_INCREMENT=1;

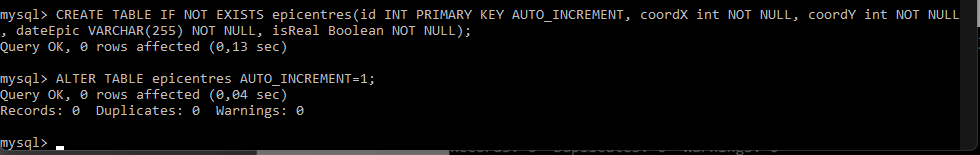


Рисунок 10 – Создание таблицы

Для просмотра списка колонок созданной таблицы использовалась команда show columns in epicentres;

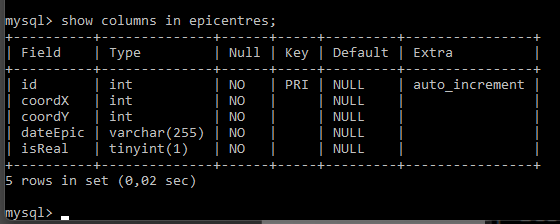


Рисунок 11 – Результат создания таблицы

Для внесения записей в таблицу применялась стандартная команда Insert.

INSERT INTO epicentres (coordX, coordY, dateEpic, isReal) VALUES (123, 47, '12.10.2019:13.00', true);

INSERT INTO epicentres (coordX, coordY, dateEpic, isReal) VALUES (59, 96, '23.02.2023:06.35', true);

INSERT INTO epicentres (coordX, coordY, dateEpic, isReal) VALUES (12, 35, '04.07.2003:22.17', false);

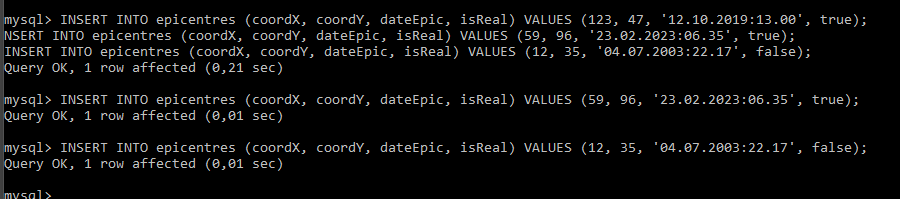


Рисунок 12 – Добавления записей

Просмотр записей в таблице epicentres осуществлялся с помощью команды select\* from epicentres;

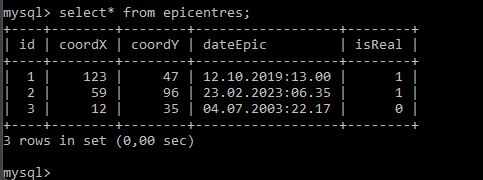


Рисунок 13 – Результат добавления записей

4.1.3 Создание роли

Новый пользователь в MySQL добавляется командой CREATE USER ' selector'@'localhost' IDENTIFIED BY '123';

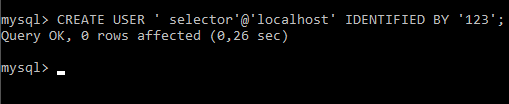


Рисунок 14 – Создание пользователя

Проверить права доступа роли можно с помощью команды show grants for ' selector '@'localhost';

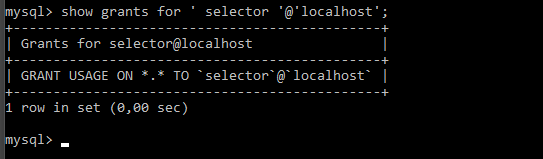


Рисунок 15 – Возможности пользователя

Для создания роли применяется команда create role selector\_userrole;

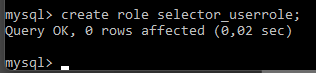


Рисунок 16 – Создание роли

Назначение прав пользования роли осуществляется с помощью команды grant select on earthquakes\_database.\* to selector\_userrole;

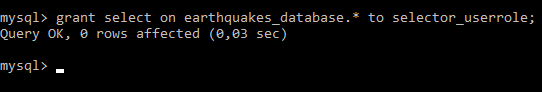


Рисунок 17 – Задание возможностей

Предоставление роли созданному пользователю осуществляется с помощью команды grant selector\_userrole to 'selector'@'localhost';

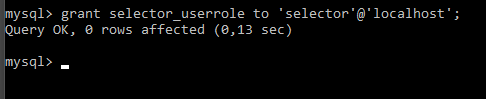


Рисунок 18 – Привязка пользователя к роли

В результате возможности пользователя были обновлены.

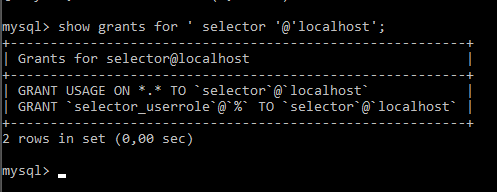


Рисунок 19 – Результат присвоения роли

Вход в созданного пользователя sudo mysql -u selector -p

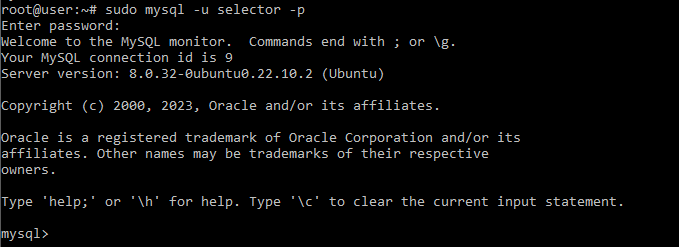


Рисунок 20 – Вход в пользователя Selector

Проверка установленной роли из доступных осуществляется с помощью вызова Select current\_role();

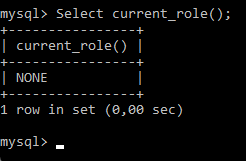


Рисунок 21 – Проверка роли

Установка роли для текущего пользователя осуществляется с помощью команды set role selector\_userrole;

Таким образом пользователь использует выбранную роль

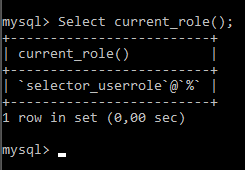


Рисунок 22 – Результат выбора роли

Далее была выбрана тестовая таблица, в которой просмотрены таблицы.

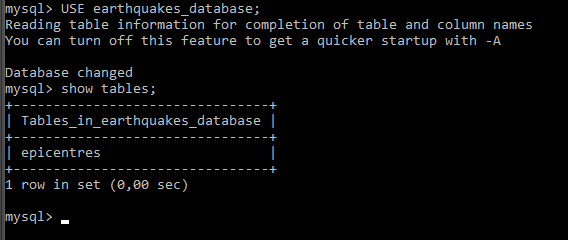


Рисунок 23 – Вызов таблиц базы данных

Так как при установлении прав роли было разрешено исключительно просматривать таблиц, то удаление данных из базы данных недоступно.

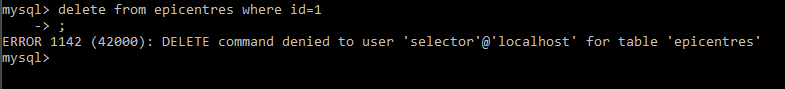


Рисунок 24 – Попытка удаления данных

Просмотр содержимого таблицы, в свою очередь, доступен для данной роли.

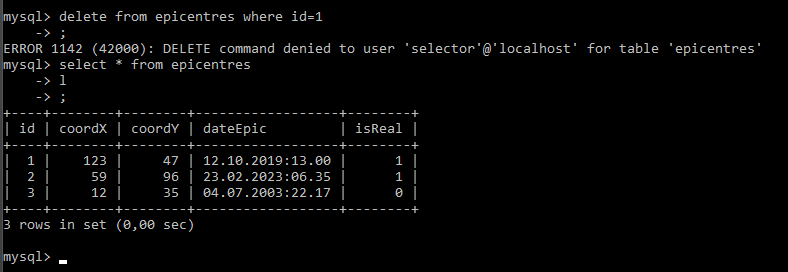


Рисунок 25 – Просмотр таблицы пользователем с заданной ролью

Удаление таблицы также заблокировано.

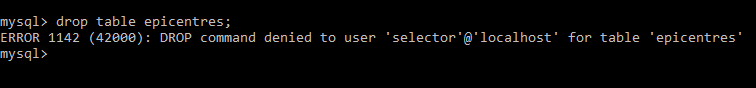


Рисунок 26 – Попытка удаления таблицы

Созданную роль можно присвоить пользователю по умолчанию. Для этого необходимо использовать команду set default role selector\_userrole to 'selector'@'localhost'; от имени администратора базы данных.

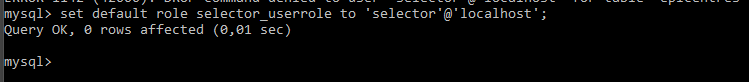


Рисунок 27 – Попытка удаления строки в таблице

4.2 Настройка PostgreSQL

4.2.1 Конфигурирование базы данных

Запуск сервиса PostgresSQL осуществляется с помощью команды sudo systemctl start postgresql.service

Проверить статус сервиса можно командой sudo systemctl status postgresql.service

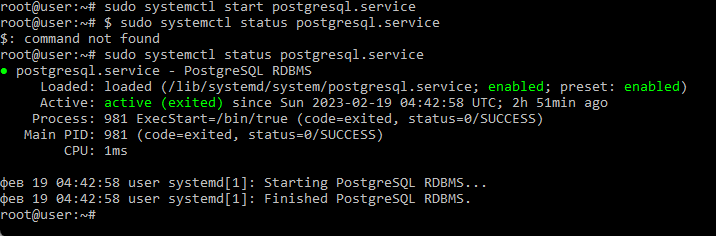


Рисунок 28 – Проверка статуса сервиса

Вход в администратора баз данных sudo -u postgres psql

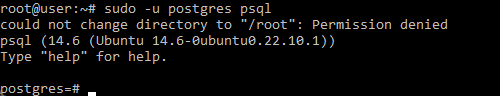


Рисунок 29 – Вход с помощью администратора

4.2.2 Создание базы данных

Чтобы создать новую базу данных PostgreSQL используется команда CREATE DATABASE earthquakes\_database; earthquakes\_database в данной команде является названием создаваемой базы данных.

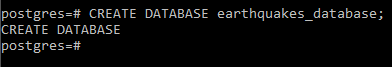


Рисунок 30 – Создание базы данных

Вывести список баз данных осуществляется командой \l. После вызова команды в списке баз данных созданная earthquakes\_database отображена на последней строке.

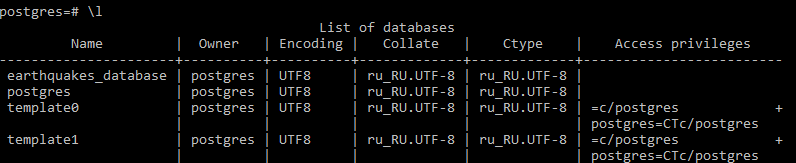


Рисунок 31 – Список баз данных

Подключение к созданной базе данных осуществляется командой \c earthquakes\_database

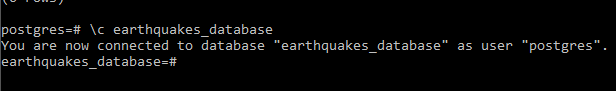


Рисунок 32 – Подключение к базе данных

Создание таблицы осуществляется командой CREATE TABLE IF NOT EXISTS epicentres(id SERIAL PRIMARY KEY, coordX int NOT NULL, coordY int NOT NULL, dateEpic VARCHAR(255) NOT NULL, isReal Boolean NOT NULL);

Для вывода имеющихся таблиц в базе данных применяется команда \dt

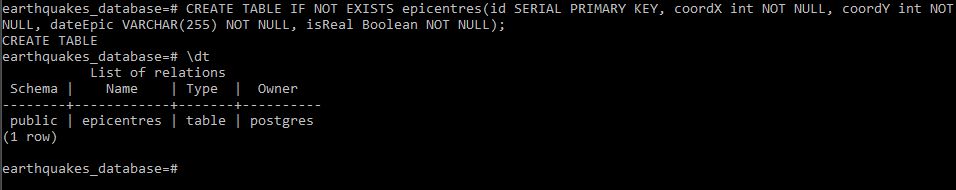


Рисунок 33 – Результат создания таблицы

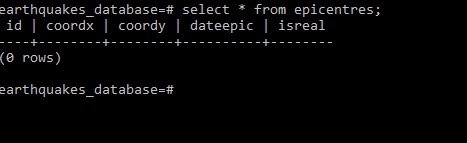


Рисунок 34 – Результат создания таблицы

Добавление данных в базу данных осуществляется с помощью команды Insert.

INSERT INTO epicentres (coordX, coordY, dateEpic, isReal) VALUES (123, 47, '12.10.2019:13.00', true);

INSERT INTO epicentres (coordX, coordY, dateEpic, isReal) VALUES (59, 96, '23.02.2023:06.35', true);

INSERT INTO epicentres (coordX, coordY, dateEpic, isReal) VALUES (12, 35, '04.07.2003:22.17', false);

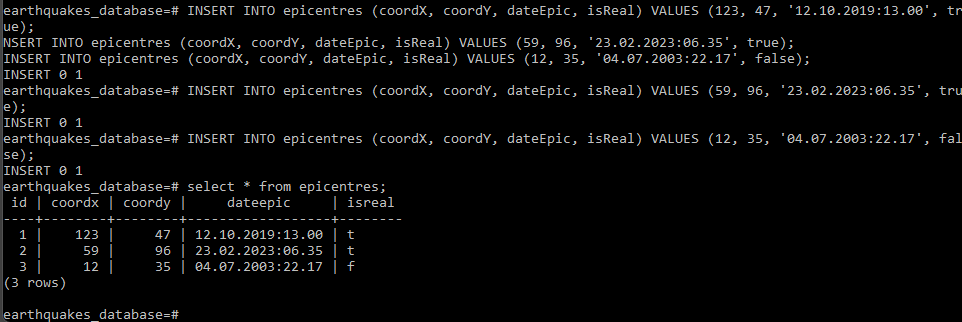


Рисунок 35 – Результат записи данных в таблицу

Удаление данных осуществляется командой delete from epicentres where id=3;

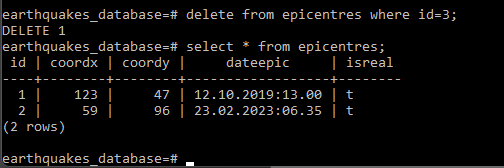


Рисунок 36 – Результат удаления записи из таблицы

4.2.3 Создание роли

Для создания пользователя используется команда CREATE USER selector WITH PASSWORD '123';

Для создания роли с возможностью «Только чтения» используется набор команд

CREATE ROLE selector\_userrole;

GRANT CONNECT ON DATABASE earthquakes\_database TO selector\_userrole;

GRANT SELECT ON TABLE epicentres TO selector\_userrole;

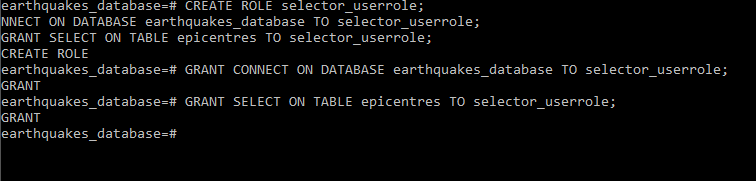


Рисунок 37 – Результат создания роли

Привязка роли к пользователю осуществляется командой GRANT selector\_userrole TO selector;

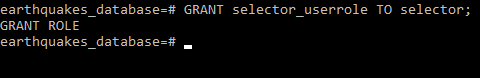


Рисунок 38 – Результат привязки роли

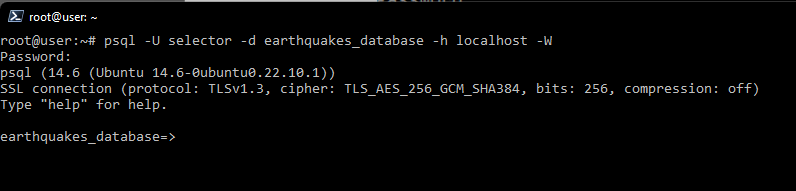


Рисунок 39 – Авторизация в созданного пользователя

Созданный пользователь обладает возможностью просмотра таблиц в закрепленной базе данных

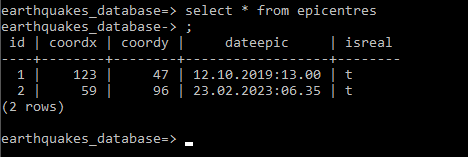


Рисунок 40 – Просмотр таблицы пользователем Selector

Операции удаления и добавления, в свою очередь, для пользователя с данной ролью недоступны.

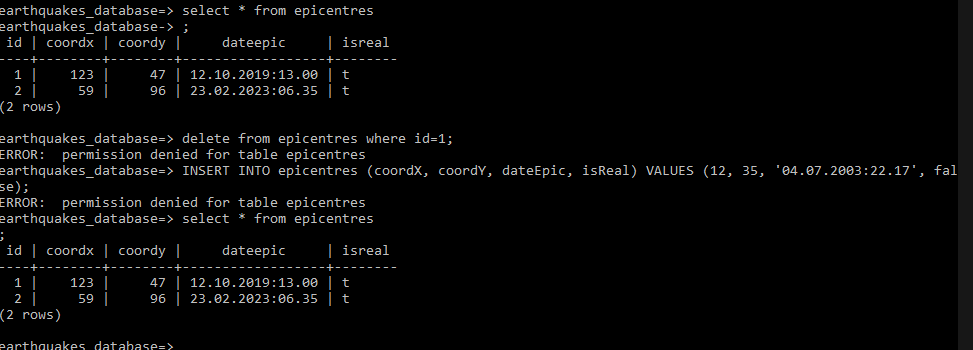


Рисунок 41 – Проверка удаления и добавления в таблицу

5 Заключение

В результате выполнения лабораторной работы были получены навыки практического администрирования баз данных в операционной системе UbuntuServer, изучены СУБД MySQL и PostgreSQL, команды и конфигурационные настройки, используемые для администрирования баз данных, имеющиеся базы данных были установлены и настроены для практического применения.