Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

кафедра Автоматизированных систем

ОТЧЕТ

к лабораторной работе №3 по дисциплине

АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

|  |
| --- |
| «Администрирование баз данных в UbuntuServer» |
|  |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | ИСМб-17-1 |  |  |  | Лескова Т.С. |
|  | шифр группы |  | подпись |  | Фамилия И.О. |
| Проверил |  |  |  |  | Хритова М.А. |
|  | должность |  | подпись |  | Фамилия И.О. |

**Содержание**

[Введение 3](#_Toc25369806)

[1 Базы данных - MySQL и PostgreSQL. Основные понятия. 4](#_Toc25369807)

[1.1 MySQL 4](#_Toc25369808)

[1.2 PostgreSQL 5](#_Toc25369809)

[2 Установка СУБД MySQL и PostgreSQL в ОС UbuntuServer 6](#_Toc25369810)

[2.1 MySQL 6](#_Toc25369811)

[2.1.1 Установка 6](#_Toc25369812)

[2.1.2 Настройка 7](#_Toc25369813)

[2.2 PostgreSQL 9](#_Toc25369814)

[2.2.1 Установка 9](#_Toc25369815)

[2.2.2 Настройка 11](#_Toc25369816)

[3 Настройка PostgreSQL для UbuntuServer 14](#_Toc25369817)

[3.1 Создание роли Postgresql 16](#_Toc25369818)

[3.2 Создание базы данных 16](#_Toc25369819)

[3.3 Создание таблиц 17](#_Toc25369820)

[3.4 Вставка и удаление данных 18](#_Toc25369821)

[Заключение 20](#_Toc25369822)

Введение

Чтобы стать более опытным пользователем Ubuntu нужно разобраться в работе приложений для взаимодействия с базами данных. Также умения администрирования баз данных являются необходимыми для администраторов Ubuntu.

**Цель работы:** Получить начальные навыки практического администрирования серверной операционной системы на примере UbuntuServer.

Задание:

1. Прочитать теоретическую информацию об управлении пакетами и по настройке сети для операционной системы UbuntuServer.
2. ИнсталлироватьUbuntuServer, продемонстрировать навыки работы с пакетами, произвести настройку сети.
3. Выполнить анализ возможностей.
4. Написать отчет.
5. Базы данных - MySQL и PostgreSQL. Основные понятия.

В операционной системе Ubuntu существует два популярных сервера баз данных: **MySQL и PostgreSQL.**

* 1. MySQL

**MySQL** — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

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.
  1. PostgreSQL

**PostgreSQL** — свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux и т.д.

Сильными сторонами PostgreSQL считаются **высокопроизводительные и надёжные механизмы транзакций** (группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными.) **и репликации**; **расширяемая система встроенных языков программирования; наследование**- концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения; **легкая расширяемость**.

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.

1. Установка СУБД MySQL и PostgreSQL в ОС UbuntuServer
   1. MySQL
      1. Установка

Для установки MySQL запустите следующую команду из терминала:

sudo apt-get install mysql-server

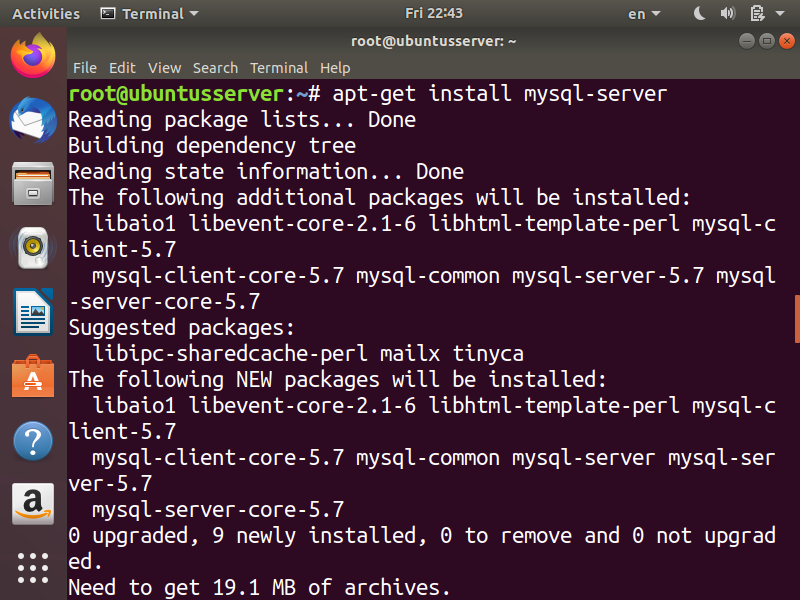


Рисунок 1 - Установка MySQL

Как только установка завершится, сервер MySQL запустится автоматически. Вы можете использовать следующую команду в терминале для проверки, что сервер MySQL запущен:

sudo netstat -tap | grep mysql

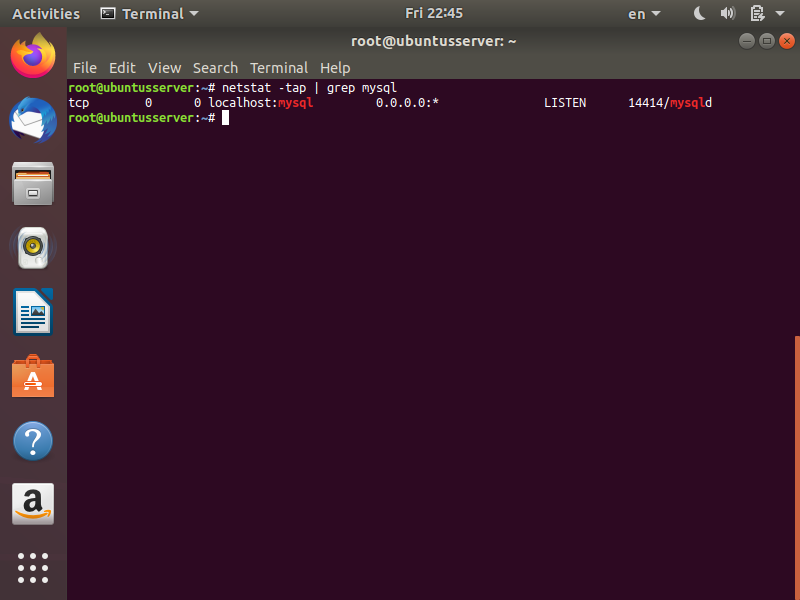


Рисунок 2 - Проверка запуска сервера mySQL

Когда вы запустите эту команду, вы сможете увидеть что-то похожее на следующую строку:

tcp 0 0 localhost:mysql \*:\* LISTEN 2556/mysqld

Если сервер не работает, вы можете набрать следующую команду для его запуска:

sudo service mysql restart

* + 1. **Настройка**

Вы можете отредактировать файл конфигурации для настройки основных параметров - файл журнала, номер порта и прочего. Расположение файла конфигурации MySQL зависит от дистрибутива. В Ubuntu и Debian файл находится по адресу /etc/mysql/mysql.conf.d/mysqld.cnf, чтобы настроить MySQL на ожидание подключений от компьютеров в сети, измените параметр **bind-address**. По умолчанию установлено значение 127.0.0.1 (прослушивается только на локальном хосте).

В этой работе мы настроим сервер MySQL на прослушивание всех интерфейсов, изменив значение на 0.0.0.0

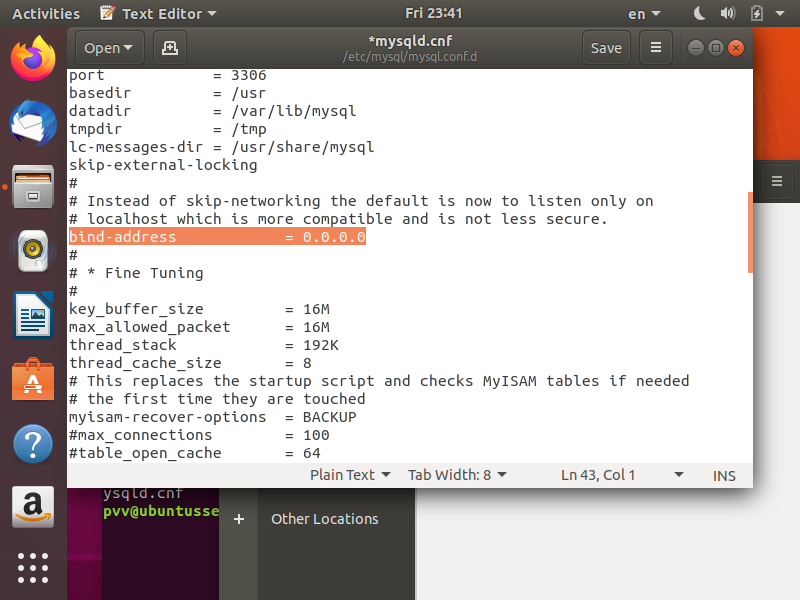


Рисунок 3 - Изменение файла my.cnf

После изменений файла конфигурации сервис MySQL нужно перезагрузить:

sudo service mysql restart

Если вам потребовалось сменить пароль пользователя **root** в MySQL, введите в терминале:

sudo dpkg-reconfigure mysql-server-\*

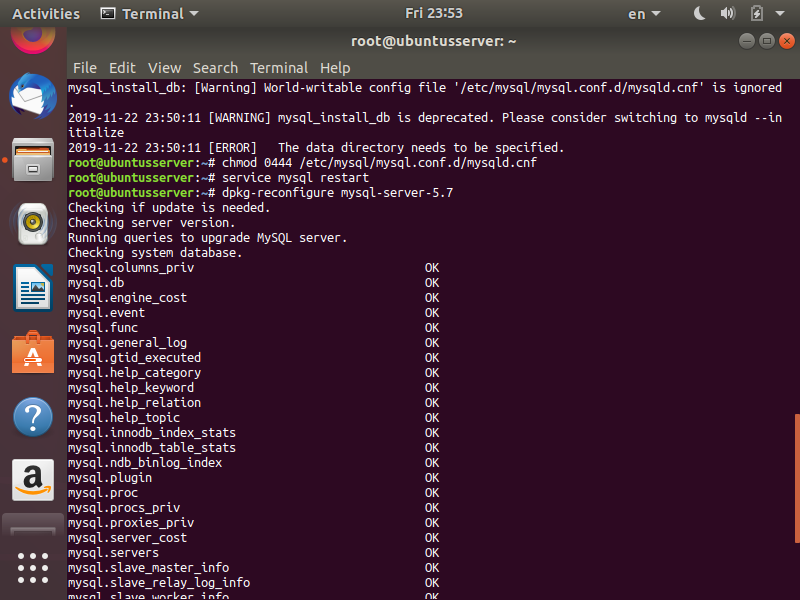


Рисунок 4 - Изменение пароля

Сервис MySQL будет остановлен и вас попросят ввести новый пароль.

* 1. PostgreSQL
     1. **Установка**

Для установки PostgreSQL выполните следующую команду в терминале:

sudo apt-get install postgresql

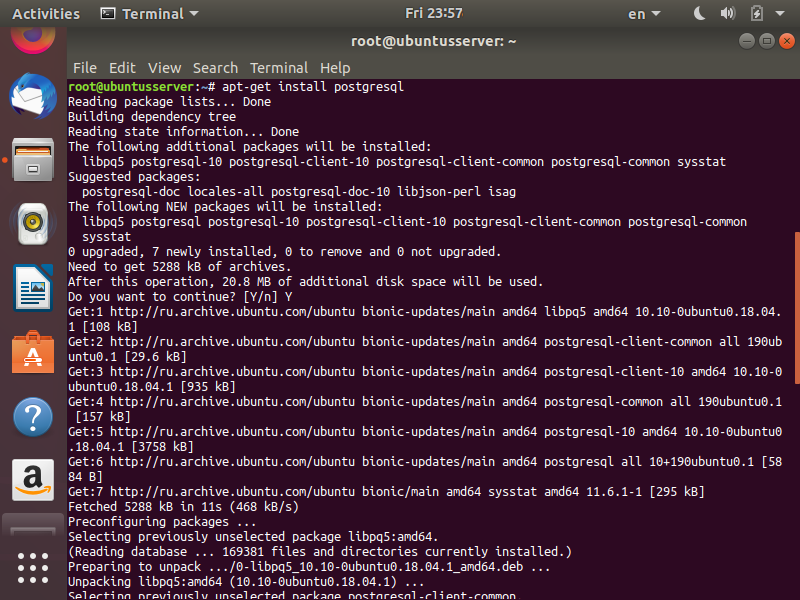


Рисунок 5 - Установка PostgreSQL

По окончанию процесса установки проверяем, запущен ли сервер PostgreSQL:

systemctl status postgresql

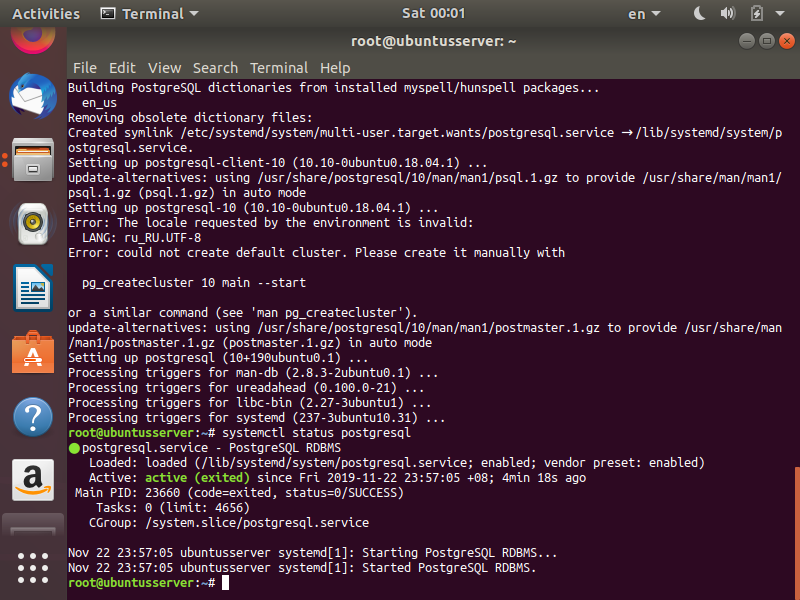


Рисунок 6 – Проверка сервера PostgreSQL

Сразу после установки вы можете настроить сервер PostgreSQL по вашим потребностям, хотя стандартная настройка вполне жизнеспособна.

* + 1. **Настройка**

По умолчанию соединения через TCP/IP заблокированы. PostgreSQL поддерживает множество методов аутентификации. Метод аутентификации **IDENT** используется для **postgres** и локальных пользователей пока не настроено что-то еще.

Дальнейшее обсуждение предполагает, что вы собираетесь разрешить соединения по TCP/IP и используете аутентификацию клиентов на основе метода MD5. Файлы настроек PostgreSQL хранятся в каталоге **/etc/postgresql/<version>/main**.

Для настройки аутентификации **ident** необходимо добавить записи в файл /etc/postgresql/10/main/pg\_ident.conf. В файле содержатся подробные комментарии чтобы направлять вас.

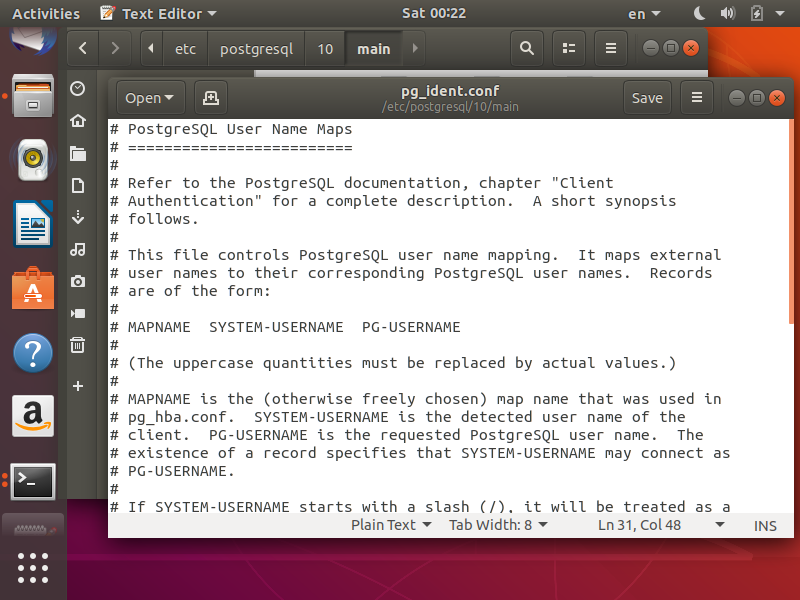


Рисунок 7 – Файл pg\_ident.conf

Чтобы разрешить соединения по TCP/IP, отредактируйте файл /etc/postgresql/10/main/postgresql.conf. Найдите строку

#listen\_addresses = 'localhost'

и замените ее на:

listen\_addresses = 'localhost'

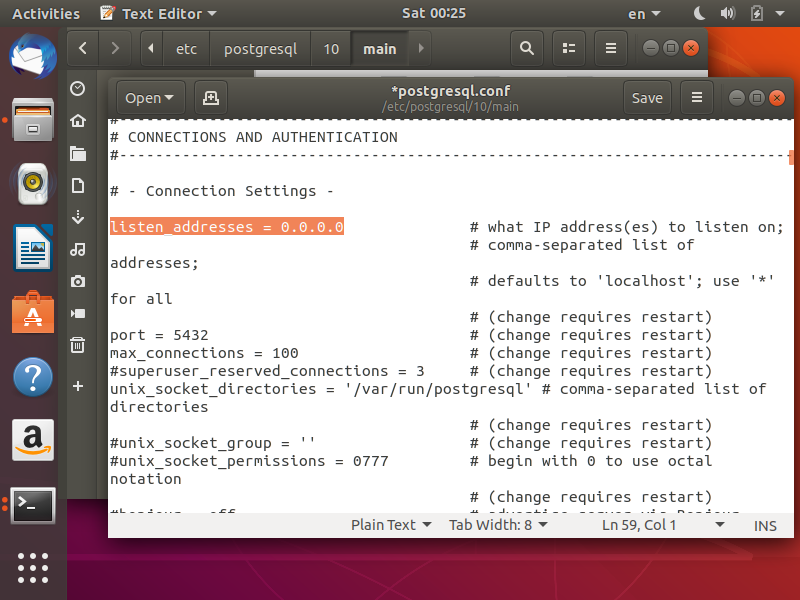


Рисунок 8 - Редактирование файла

Чтобы разрешить другим компьютерам соединяться с вашим PostgreSQL сервером, замените 'localhost' на IP адрес вашего сервера или в качестве альтернативы на 0.0.0.0, чтобы подключить все интерфейсы.

Теперь, поскольку мы можем подключиться к нашему серверу PostgreSQL, следующим шагом будет установка пароля для пользователя **postgres**. Выполните следующую команду в терминале для соединения со стандартной базой шаблонов PostgreSQL:

sudo -u postgres psql template1

Эта команда подключится к PostgreSQL базе данных template1 как пользователь **postgres**. После подключения к серверу PostgreSQL вы окажетесь в SQL консоли. Вы можете выполнить следующую SQL команду в консоли **psql** для настройки пароля пользователя **postgres**:

ALTER USER postgres with encrypted password 'your\_password'

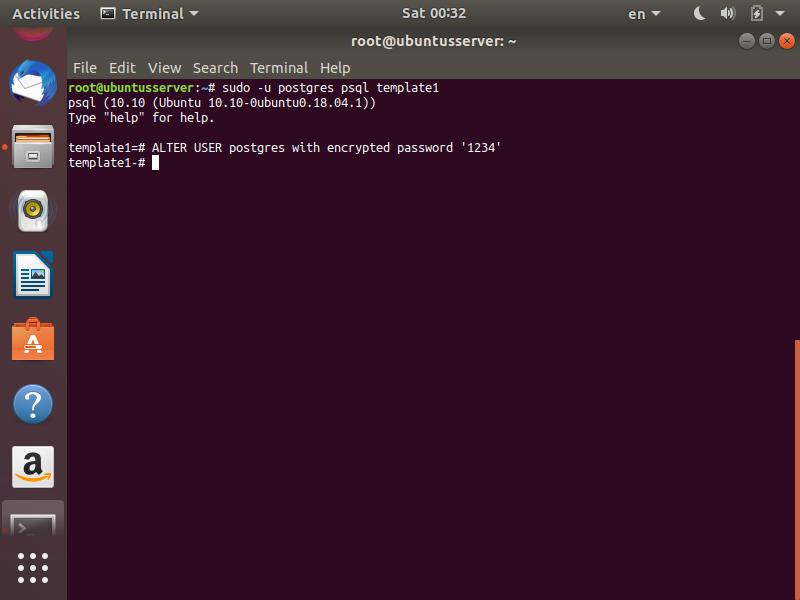


Рисунок 9 - Установка пароля для пользователя postgres

После настройки пароля, измените файл /etc/postgresql/10/main/pg\_hba.conf на использование MD5 аутентификации для пользователя **postgres**:

local all postgres md5

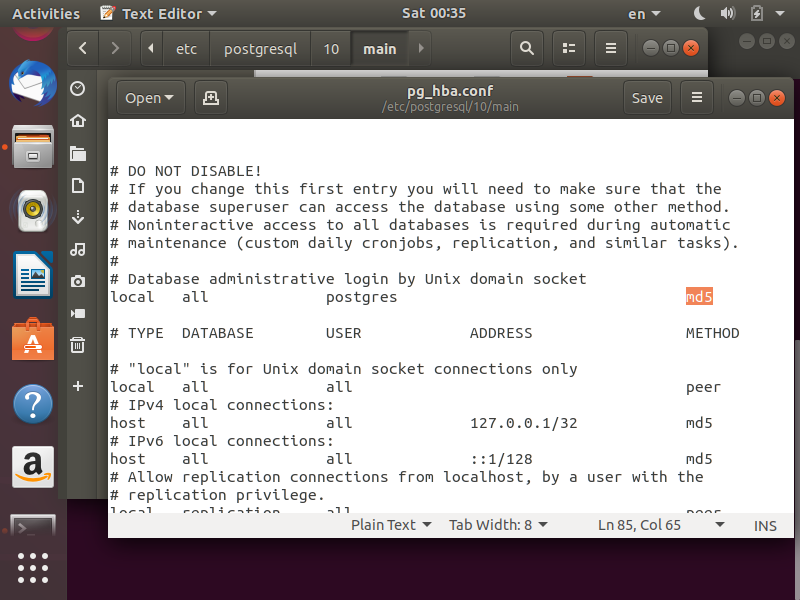


Рисунок 10 - Применение MD5 аутентификации

Под конец вам потребуется перезапустить сервис PostgreSQL для применения новых настроек. Из терминала выполните следующее для перезапуска PostgreSQL:

sudo service postgresql restart

1. **Настройка PostgreSQL для UbuntuServer**

Вы знаете, как установить Postgresql Ubuntu, но этого недостаточно для начала полноценной работы. Первым делом, откройте терминал и переключите его на пользователя **postgres** с помощьюкоманды:

sudo su - postgres

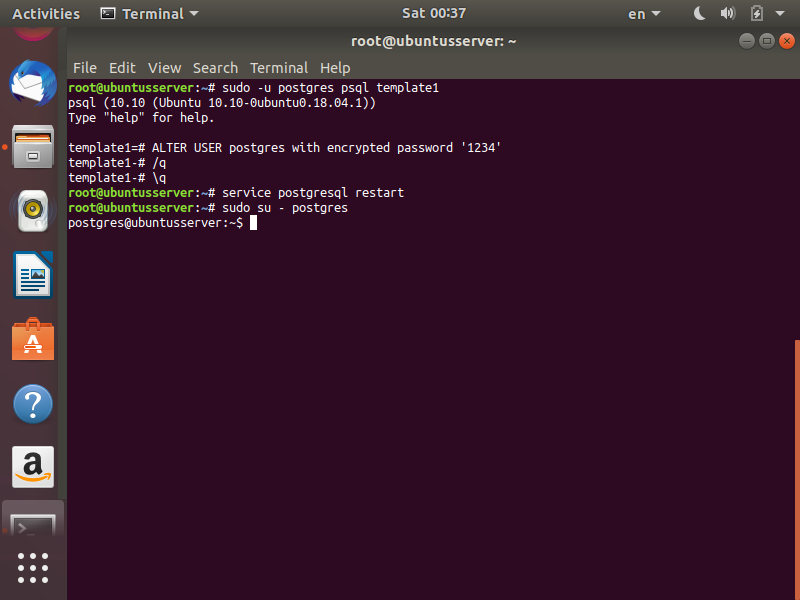


Рисунок 11 - Переключение на пользователя postgrespsql

Эта учетная запись создается во время установки программы и на данный момент вы можете получить доступ к системе баз данных только с помощью нее. По умолчанию **postgres** использует концепцию ролей для аутентификации и авторизации.

Это очень похоже на учетные записи Unix, но программа не различает пользователей и групп, есть только роли. Сразу после установки Postgresql пытается связать свои роли с системными учетными записями, если для имени системной учетной записи существует роль, то пользователь может войти в консоль управления и выполнять позволенные ему действия. Таким образом, после переключения на пользователя **postgres** вы можете войти в консоль управления:

psql

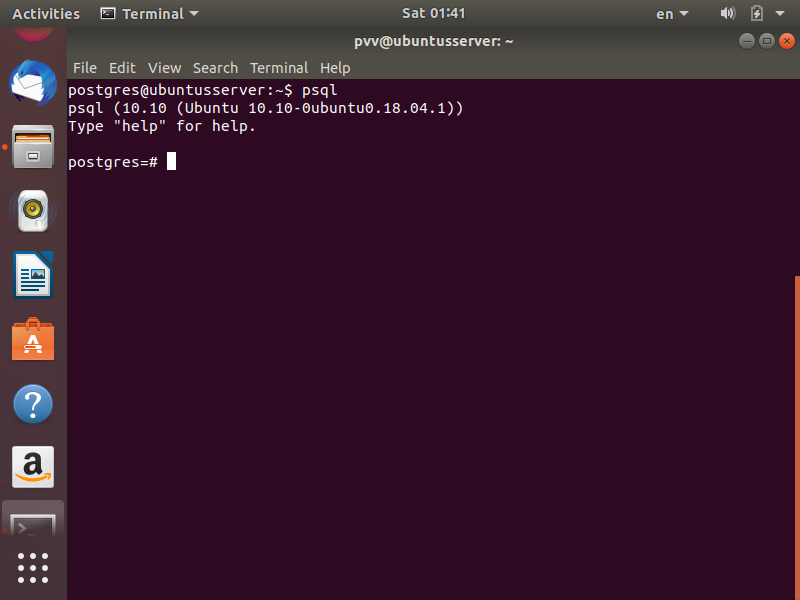


Рисунок 12 - Консоль управления

И посмотреть информацию о соединении**:**

\conninfo

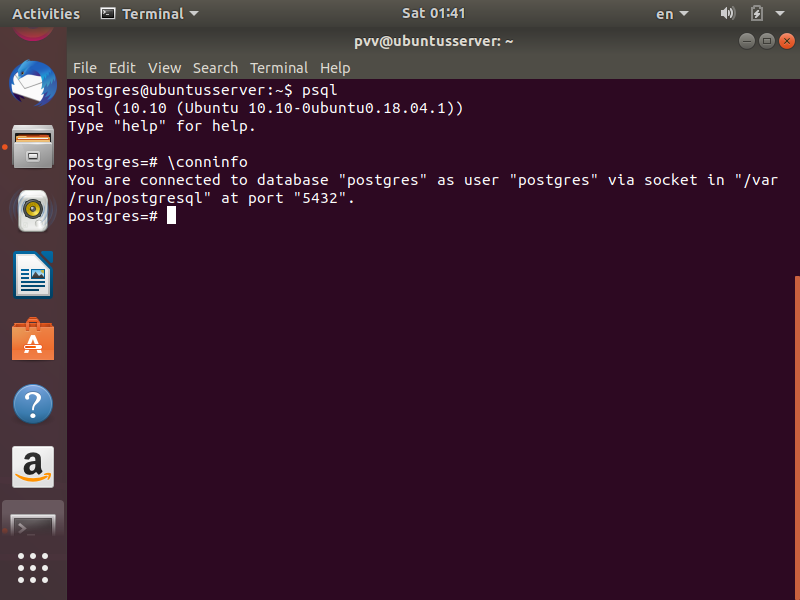


Рисунок 13 - Информация о соединении

Чтобы выйти наберите:

\q

Теперь рассмотрим, как создать другие роли и базы данных.

* 1. Создание роли Postgresql

Вы уже можете полноценно работать с базой данных с помощью учетной записи **postgres**, но давайте создадим дополнительную роль. Учетная запись **postgres** является администратором, поэтому имеет доступ к функциям управления. Для создания пользователя выполните:

createuser --interactive

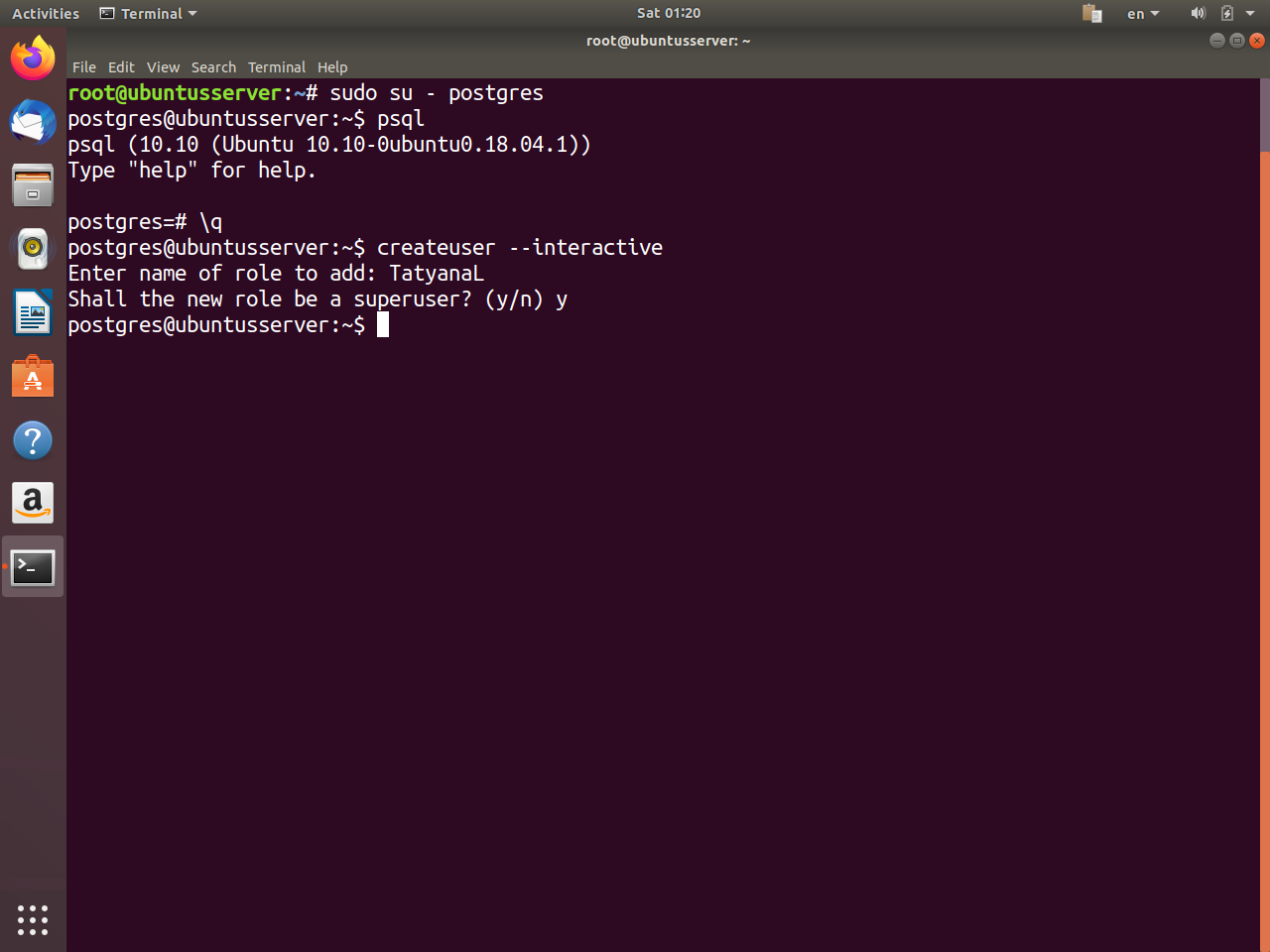


Рисунок 14 - Создание роли Postgresql

Скрипт задаст лишь два вопроса, имя новой роли и нужно ли делать ее суперпользователем.

* 1. Создание базы данных

Точно также как имена ролей сопоставляются с системными пользователями, имя базы данных будет подбираться по имени пользователя. Например, если мы создали пользователя **TatyanaL**, то по умолчанию система попытается получить доступ к базе данных **TatyanaL**. Мы можем ее очень просто создать:

createdb TatyanaL

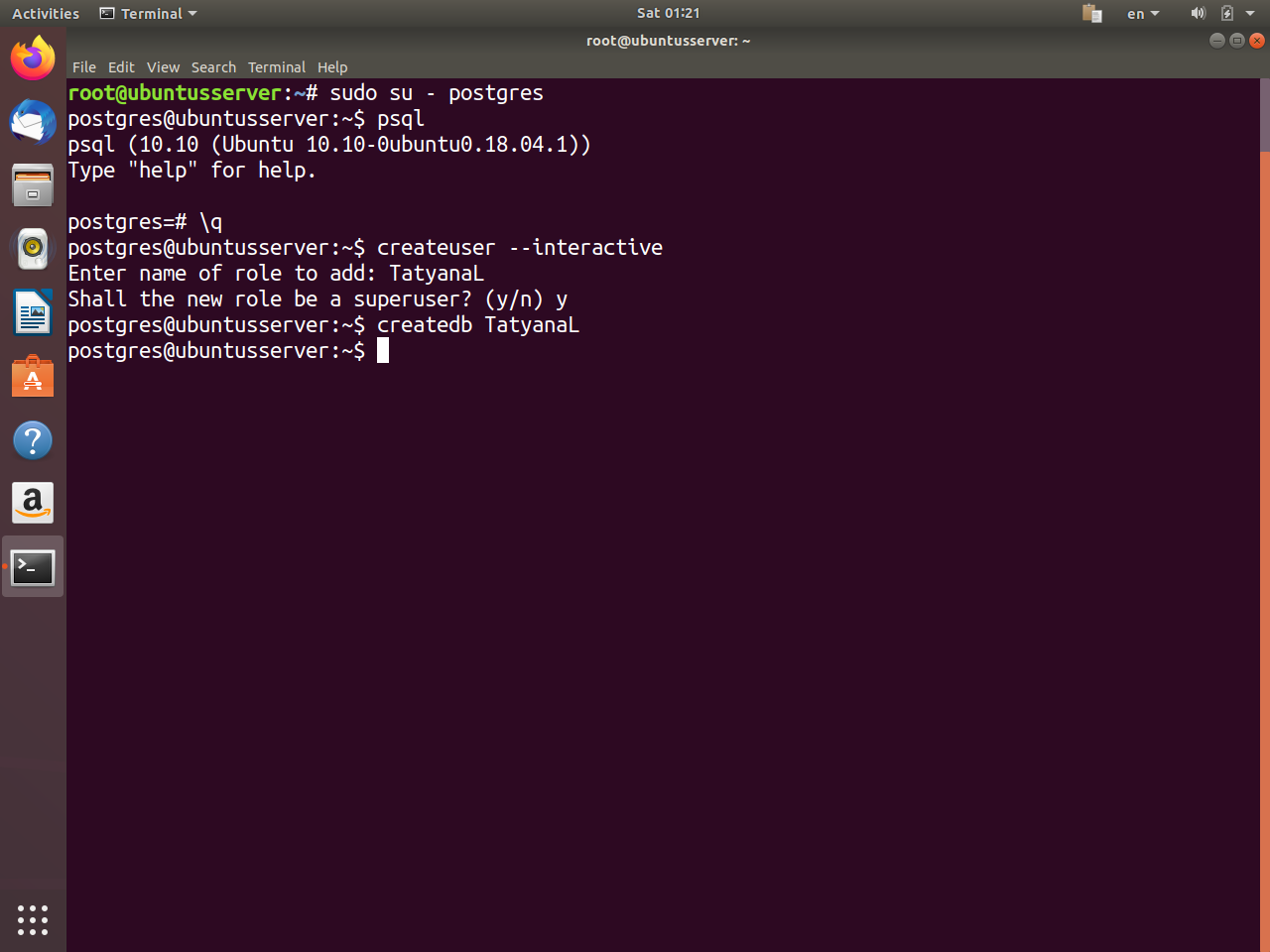


Рисунок 15 - Создание базы данных

Дальше, чтобы подключиться к этой базе данных нам нужно войти от имени одноименного пользователя:

sudo su - TatyanaL

Заходим в консоль и смотрим информацию о подключении:

psql

затем

\conninfo

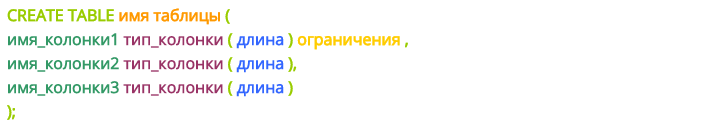
Все верно сработало. Мы подключились с помощью роли **TatyanaL** к базе **TatyanaL**. Если нужно указать другую базу данных, вы можете сделать это с помощью опции -d, например:

psql -d postgres

Все сработало верно, при условии, что все компоненты были настроены правильно.

* 1. Создание таблиц

Теперь, когда вы знаете как подключится к базе данных Postgresql давайте рассмотрим, как выполняются основные задачи. Сначала разберем создание таблиц для хранения некоторых данных. Для создания таблицы Postgresql используется такой синтаксис:

****

Как видите, сначала мы задаем имя таблицы, затем описываем каждый столбец. Столбец должен иметь имя, тип и размер, также можно задать ограничения для данных, которые там будут содержаться. Например:

CREATE TABLE session (

exam\_id serial PRIMARY KEY,

subject varchar (50) NOT NULL,

teacher varchar (25) NOT NULL,

type varchar (4) check (type in ('test','exam')),

exam\_date date

);

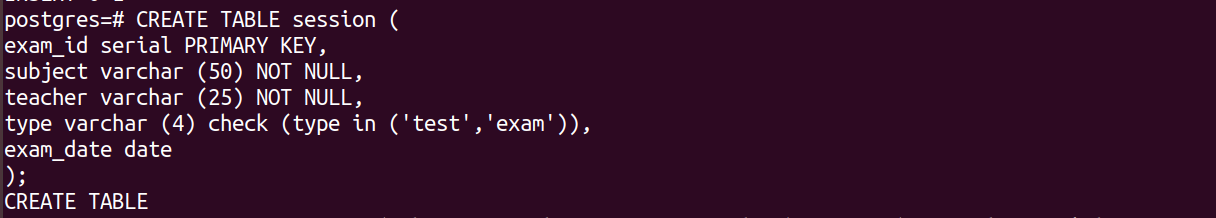


Рисунок 16 - Создание таблицы

Мы создали таблицу сессии, с входящими в нее экзаменами и зачетами. Сначала идет идентификатор **exam\_id**, который имеет тип **serial**, это значит, что его значение будет автоматически увеличиваться, ключ **primary key** значит, что значения должны быть уникальны.

Следующие колонки — обычные строки, для них мы задаем длину поля, они не могут быть пустыми (NOT NULL). Следующий столбец тоже строка, но она может содержать только одно из указанных значений, последний столбец — дата экзамена с типом данных – «дата».

Вы можете вывести все таблицы, выполнив команду:

\d

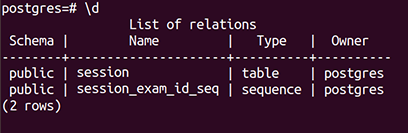


Рисунок 17 - Вывод таблицы на экран

Здесь мы видим, что кроме нашей таблицы, существует еще одна переменная — **session\_exam\_id\_seq**. В ней содержится последнее значение этого поля. Если нужно вывести только таблицы, выполните:

\dt

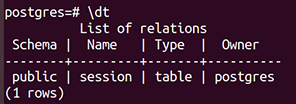


Рисунок 18 - Вывод на экран конкретной таблицы

* 1. Вставка и удаление данных

У нас есть таблица, но пока что она пуста. Мы можем добавить в нее некоторые данные. Давайте добавим пункты, описывающий разные книги. Для этого используются такие команды:

INSERT INTO session (subject, teacher, type, exam\_date) VALUES ('English', 'Yakoba I. A.', 'test', '2019-12-26');

INSERT INTO session (subject, teacher, type, exam\_date) VALUES ('Information Systems Administration', 'Hritova M. A.', 'exam', '2020-01-14');

INSERT INTO session (subject, teacher, type, exam\_date) VALUES ('Parallel Programming', 'Ippolitov A. A.', 'exam', '2020-01-17');

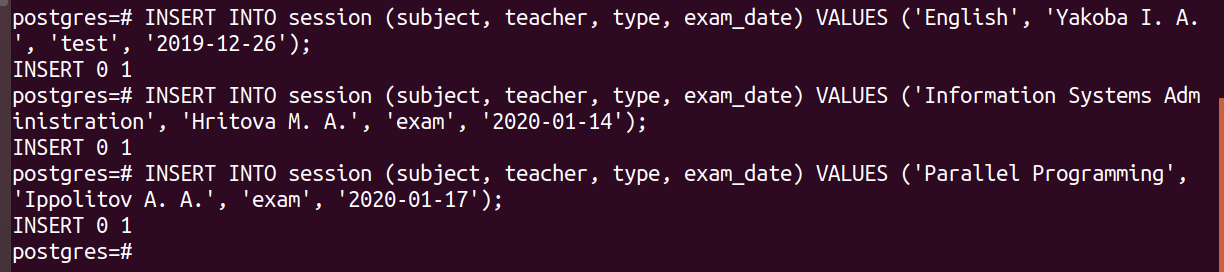


Рисунок 19 - Добавление записей в таблицу и их просмотр

Заметьте, что имена столбцов не обязательно заключать в кавычки, а вот имена значений — обязательно. Теперь смотрим что получилось:

SELECT \* FROM session;

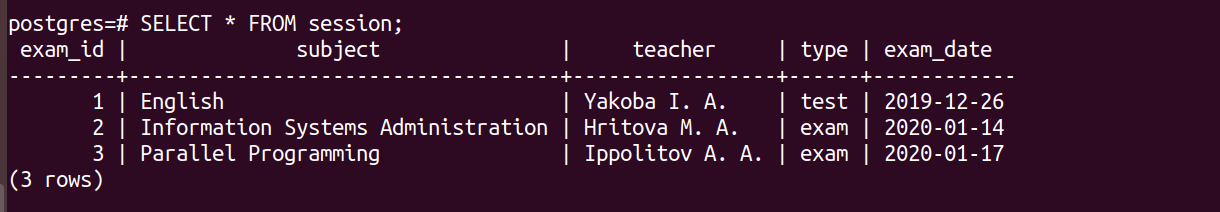


Рисунок 20 – Просмотр добавленных записей

Удалять записи можно по любому критерию, например, удалим записи, поле **type** которых - имеет значение‘**test**’**:**

DELETE FROM session WHERE type = 'test';

И снова смотрим что получилось:

SELECT \* FROM session;

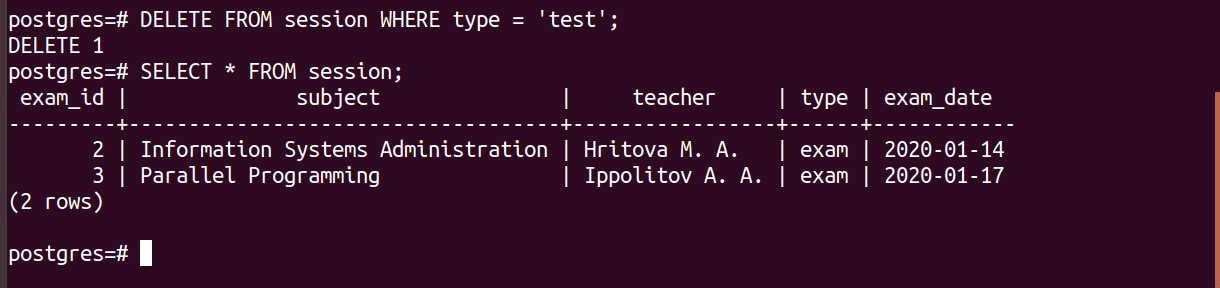


Рисунок 21 - Удаление записи и проверка ее удаления

Заключение

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

Все знания и полученные навыки должны помочь в будущем, более глубоком освоении в плане администрирования ОС UbuntuServer.