**Иркутский национальный исследовательский технический университет**

**Институт кибернетики им Е.И. Попова**

**Кафедра автоматизированных систем**

**Методические указания**

к лабораторным работам по курсу   
«**Администрирование информационных систем**»

Редакция 3.2020

**Лабораторная работа №3 (2 семестр)**

**«Администрирование баз данных в UbuntuServer»**

(4 часа)

**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.

http://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE\_%D0%BF%D0%BE\_ubuntu\_server/%D0%B1%D0%B0%D0%B7%D1%8B\_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85

<http://help.ubuntu.ru/wiki/руководство_по_ubuntu_server>

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.

**1 Базы данных**

Ubuntu предоставляет два популярных сервера баз данных:

**MySQL**

**PostgreSQL**

**1.1** **MySQL**

MySQL - это быстрый, многопоточный, многопользовательский и устойчивый сервер SQL базы данных. Он предназначен как для ответственных сильно загруженных производственных систем, так и для встраивания в массовое программное обеспечение.

### Установка

Для установки MySQL запустите следующую команду из терминала:

sudo apt-get install mysql-server

В процессе установки у вас запросят пароль для пользователя root под 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

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

### Настройка

Вы можете отредактировать файл /etc/mysql/my.cnf для настройки основных параметров - файл журнала, номер порта и пр. Например, чтобы настроить MySQL на ожидание подключений от компьютеров в сети, измените параметр **bind-address** на IP адрес сервера:

bind-address = 192.168.0.5

Замените 192.168.0.5 на реальное значение адреса вашего сервера.

После изменений в /etc/mysql/my.cnf 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-5.5

Сервис MySQL будет остановлен и вас попросят ввести новый пароль.

### Драйверы базы данных

Хотя конфигурация по умолчанию для MySQL, предоставляемая пакетами Ubuntu, имеет великолепную функциональность и работает достаточно хорошо, есть некоторые вещи, которые вы можете решить до того как продолжить.

**MySQL разработан так, что позволяет хранить данные по-разному. Эти варианты относятся к драйверам (управляющим модулям - engines) как баз данных, так и хранилищ**. Существует два основных драйвера, которые вам могут быть интересны: **InnoDB** и **MyISAM**. Драйверы хранилищ прозрачны (незаметны) конечным пользователям. MySQL управляет событиями по-разному на нижнем уровне, но независимо от того, какая система хранения данных используется, вы будете взаимодействовать с базой одним и тем же способом.

Каждый драйвер имеет свои преимущества и недостатки.

Хотя смешивание и связывание драйверов баз данных на уровне таблиц разрешается и может быть привлекательным, это снижает эффективность настройки производительности, которую вы смогли бы провести при разделении ресурсов между двумя системами вместо замешивания их в одно целое.

1. **MyISAM** более старая из двух. Она может быть быстрее InnoDB при определенных обстоятельствах и предпочтительна при рабочей нагрузке, ориентированной на чтение данных. Некоторые интернет приложения настроены на использование именно MyISAM (однако это не означает, что они будут медленнее под InnoDB). MyISAM также поддерживает тип данных **FULLTEXT**, который позволяет осуществлять очень быстрый поиск по большому количеству текстовых данных. Однако MyISAM поддерживает блокировку записи только на уровне таблиц. Это означает, что только один процесс может изменять данные в таблице в один момент времени. Поскольку некоторые приложения, использующие таблицу, могут масштабироваться (работать несколькими экземплярами - scales), это может стать серьезной помехой. Здесь также отсутствует журналирование, что может усложнить восстановление данных после сбоя. Следующая ссылка предоставляет некоторые соображения по использованию [MyISAM в работающей базе данных](http://www.mysqlperformanceblog.com/2006/06/17/using-myisam-in-production/" \o "http://www.mysqlperformanceblog.com/2006/06/17/using-myisam-in-production/).
2. **InnoDB** - более современный драйвер, созданный по принципам [ACID](http://ru.wikipedia.org/wiki/ACID), что гарантирует надежную обработку транзакций базы данных. Блокировка записи производится на уровне одной записи в таблице. Это означает возможность нескольких изменений в одной таблице одновременно. Кэширование данных происходит также и в оперативной памяти внутри драйвера базы данных, позволяя кэшировать более эффективно, чем на уровне блоков файлов. В соответствии с **ACID** все транзакции журналируются независимо от основных таблиц. Это позволяет намного более надежно восстанавливать данные при проверке целостности данных.

Начиная с MySQL 5.5, **InnoDB** является драйвером по умолчанию и настоятельно рекомендуется вместо **MyISAM**, если только у вас нет специфических потребностей, уникальных для этого драйвера.

### 

### Расширенные настройки

### Создание настроенного файла my.cnf

Существует ряд параметров, которые могут быть указаны в файле настроек MySQL, что со временем позволит вам повысить производительность вашего сервера. Для начальной настройки вам может пригодиться [Percona's my.cnf generating tool](http://tools.percona.com/members/wizard" \o "http://tools.percona.com/members/wizard). Этот инструмент позволит вам создать файл my.cnf более оптимизированный под специфические возможности вашего сервера и ваши требования.

**Не меняйте** ваш существующий файл my.cnf на созданный утилитой, если у вас уже загружены данные в вашу базу данных. Некоторые изменения в этом файле могут быть несовместимы, если вы поменяли то, как хранятся данные на диске, после чего вы не сможете запустить MySQL. Если вы все же решили это сделать и у вас есть данные в базе, вам потребуется выполнить **mysqldump** и загрузить данные повторно:

mysqldump --all-databases --all-routines -u root -p > ~/fulldump.sql

Эта операция запросит пароль пользователя root перед созданием копии данных. Желательно убедиться, что другие пользователи и процессы не используют базу во время этой операции. В зависимости от количества данных в вашей базе, это может занять длительное время. Вы не увидите ничего на экране во время этого процесса.

Как только файл выгрузки создан, остановите MySQL:

sudo service mysql stop

Теперь сохраните оригинальный файл my.cnf и замените его на новый:

sudo cp /etc/my.cnf /etc/my.cnf.backup

sudo cp /path/to/new/my.cnf /etc/my.cnf

Затем удалите и создайте заново пространство базы данных, а также убедитесь в корректности его владельца перед запуском MySQL:

sudo rm -rf /var/lib/mysql/\*

sudo mysql\_install\_db

sudo chown -R mysql: /var/lib/mysql

sudo service start mysql

Теперь все, что осталось - это импортировать ваши данные. Чтобы дать нам представление как идет процесс импорта, полезно использовать утилиту 'Pipe Viewer' (pv). Далее показано как установить и использовать **pv** для этого случая, но если вы не хотите ее использовать, просто замените **pv** на **cat** в соответствующей команде. Не обращайте внимание на время ETA (расчетное время окончания), генерируемое pv, поскольку оно основывается на среднем времени обработки каждой записи при сохранении в файл, однако скорость может сильно меняться от записи к записи при использовании **mysqldumps**:

sudo apt-get install pv

pv ~/fulldump.sql | mysql

Когда все завершится, это будет означать что все получилось!

Эта операция не обязательна для всех изменений my.cnf. Многие значения, которые вы захотите поменять для улучшения производительности сработают даже на работающем сервере. Но как всегда не забудьте сделать надежную копию файлов настроек и данных перед внесением изменений.

**MySQL Tuner**

**MySQL Tuner** - это полезный инструмент, который подсоединяется к работающему MySQL и предлагает варианты, как можно улучшить настройки для вашей рабочей нагрузки. Чем дольше работает сервер, тем лучше рекомендации предоставит **mysqltuner**. Для рабочего окружения подождите как минимум 24 часа прежде чем запускать утилиту. Вы можете установить **mysqltuner** из хранилища Ubuntu:

sudo apt-get install mysqltuner

После установки запустите ее:

mysqltuner

и ждите ее финального отчета. Верхняя секция предоставляет общую информацию о сервере баз данных, а нижняя часть содержит рекомендации по настройке, необходимые для изменения вашего my.cnf. Многие из них могут быть поправлены вживую на сервере без перезагрузки. Смотрите официальную документацию MySQL для перечня параметров, изменяемых "налету". Далее часть примерного отчета по работающей базе, который показывает, что можно извлечь некоторую пользу от увеличения размера кэша запросов:

-------- Recommendations -----------------------------------------------------

General recommendations:

Run OPTIMIZE TABLE to defragment tables for better performance

Increase table\_cache gradually to avoid file descriptor limits

Variables to adjust:

key\_buffer\_size (> 1.4G)

query\_cache\_size (> 32M)

table\_cache (> 64)

innodb\_buffer\_pool\_size (>= 22G)

## 1.2 PostgreSQL

**PostgreSQL** - это объектно-реляционная система баз данных, которая обладает признаками традиционной коммерческой базы данных, с расширениями, которые будут доступны следующему поколению СУБД (систем управления базами данных).

### Установка

Для установки PostgreSQL выполните следующую команду в терминале:

sudo apt-get install postgresql

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

### 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. Обратитесь к [PostgreSQL Administrator's Guide](http://www.postgresql.org/docs/8.4/static/admin.html" \o "http://www.postgresql.org/docs/8.4/static/admin.html), если вы собираетесь использовать какую-либо альтернативу типа Kerberos.

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**. Например, если вы установили PostgreSQL 8.4, файлы настроек сохранятся в каталоге /etc/postgresql/8.4/main.

Для настройки аутентификации **ident** добавьте записи в файл /etc/postgresql/8.4/main/pg\_ident.conf. В файле содержатся подробные комментарии чтобы направлять вас.

Чтобы разрешить соединения по TCP/IP, отредактируйте файл /etc/postgresql/8.4/main/postgresql.conf. Найдите строку

#listen\_addresses = 'localhost'

и замените ее на:

listen\_addresses = 'localhost'

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.

Вы можете также редактировать любые другие параметры, если знаете, что вы делаете. Для подробностей смотрите комментарии файла настроек или документацию по PostgreSQL.

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';

После настройки пароля, измените файл /etc/postgresql/8.4/main/pg\_hba.conf на использование MD5 аутентификации для пользователя postgres:

local all postgres md5

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

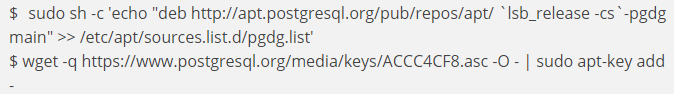
sudo /etc/init.d/postgresql-8.4 restart

Настройка выше в любом случае неполная. Пожалуйста обратитесь к руководству [PostgreSQL Administrator's Guide](http://www.postgresql.org/docs/8.4/static/admin.html" \o "http://www.postgresql.org/docs/8.4/static/admin.html) для настройки других параметров.

## 2. НАСТРОЙКА POSTGRESQL В UBUNTU

## 2.1 Альтернативная УСТАНОВКА POSTGRESQL В UBUNTU

Если вы хотите получить самую новую версию, то придется добавить в систему PPA. Для этого выполните команды:



Затем обновите списки пакетов, чтобы получить самую новую доступную версию:



Установка Postgresql Ubuntu из PPA или официальных репозиториев выглядит одинаково:

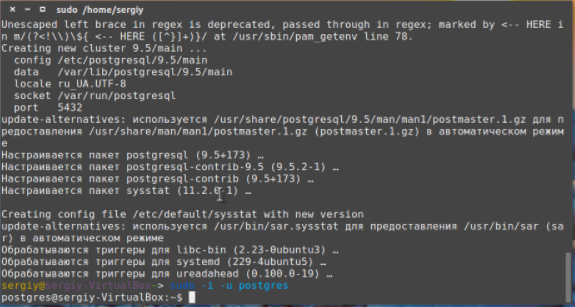


Когда установка будет завершена, можно переходить к настройке.

## 2.2 НАСТРОЙКА POSTGRESQL В UBUNTU

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

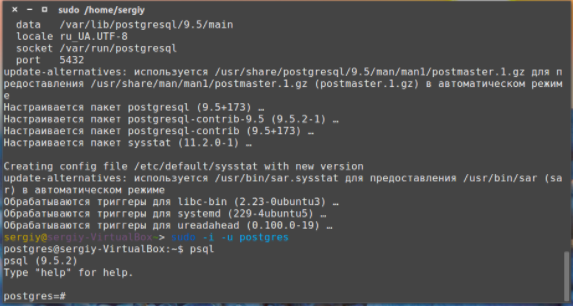




Эта учетная запись создается во время установки программы, и на данный момент вы можете получить доступ к системе баз данных только с помощью нее. По умолчанию postgress использует концепцию ролей для аутентификации и авторизации.

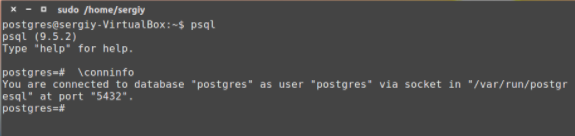
Это очень похоже на учетные записи Unix, но программа не различает пользователей и групп, есть только роли. Сразу после установки Postgresql пытается связать свои роли с системными учетными записями, если для имени системной учетной записи существует роль, то пользователь может войти в консоль управления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



****

**И посмотреть информацию о соединении:**

****

****

**Чтобы выйти наберите:**

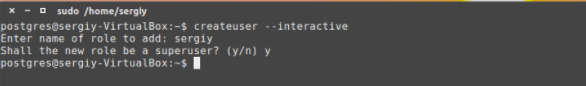
****

Теперь давайте рассмотрим, как создать другие роли и базы данных.

### 2.3 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**:

****



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

### 2.4 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 segiy, то по умолчанию система попытается получить доступ к базе данных segiy. **Мы можем ее очень просто создать:**

****

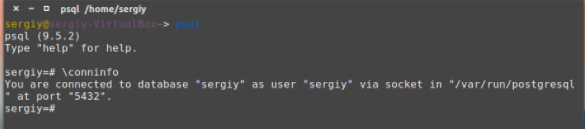
Дальше, чтобы подключиться к этой базе данных нам **нужно войти от имени одноименного пользователя:**

****

****

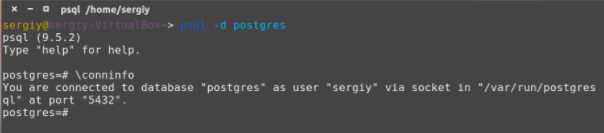
**Заходим в консоль и смотрим информацию о подключении:**

****



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

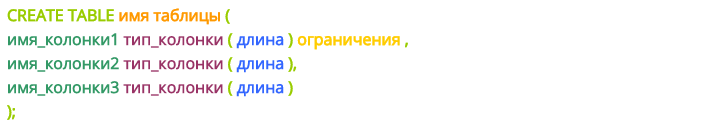
****



Все сработало верно, при условии, что все компоненты были настроены, как описано выше.

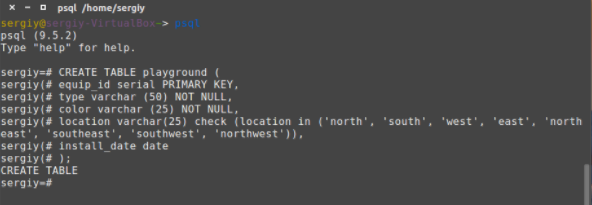
### 2.5 СОЗДАНИЕ ТАБЛИЦ

Теперь, когда вы знаете, как подключиться 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 playground (  
equip\_id serial PRIMARY KEY,  
type varchar (50) NOT NULL,  
color varchar (25) NOT NULL,  
location varchar(25) check (location in (‘north’, ‘south’, ‘west’, ‘east’, ‘northeast’, ‘southeast’, ‘southwest’, ‘northwest’)),  
install\_date date  
);**

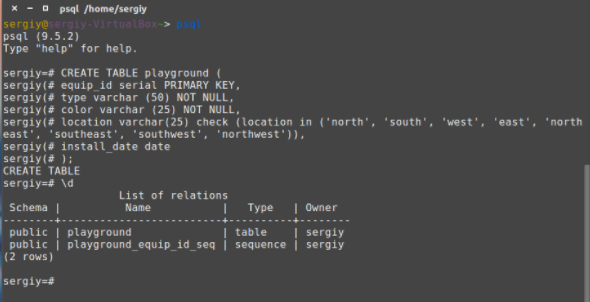
****

Мы создали таблицу детской площадки для описания оборудования, которое на ней есть. Сначала идет идентификатор equip\_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 создания.

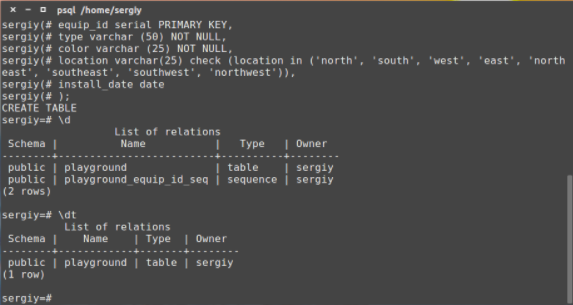
**Вы можете вывести все таблицы, выполнив команду:**

****

****

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

****

****

### 2.6 ВСТАВКА И УДАЛЕНИЕ ДАННЫХ

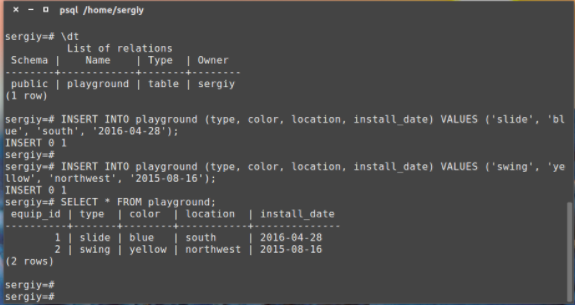
У нас есть таблица, но пока что она пуста. Мы можем добавить в нее некоторые данные. **Давайте добавим пункт, описывающий качели. Для этого используются такие команды:**

****

****

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

****

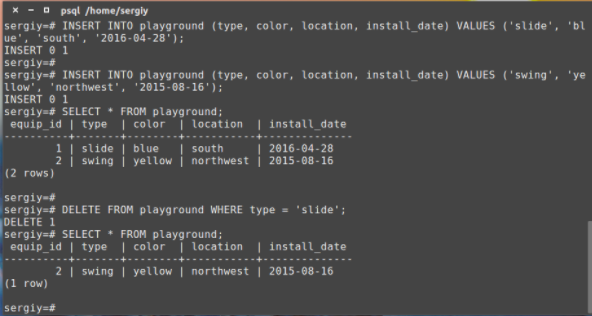
****

Удалять записи можно по любому критерию, например, **удалим записи, поле type которых имеет значение slide:**

****

**И снова смотрим что получилось:**

****

****

## 3 УСТАНОВКА PHPPGADMIN

Не всегда удобно управлять базой данных из терминала. Иногда нужно получить **доступ ко всему через веб-интерфейс**. Для этого есть программа phppgadmin, но для ее работы нужен веб-сервер Apache. Для установки программы наберите:

****

Когда установка будет завершена откройте файл /etc/apache2/conf-available/phppgadmin.conf и закомментируйте строку:

****

А вместо нее добавьте:

****

Это необходимо, чтобы открыть доступ к этому адресу не только с локального компьютера, но и их других устройств сети. Заметьте, что вы не сможете войти под учетной записью postgres, это сделано из соображений безопасности. Когда завершите, перезагрузите Apahce:

****

**Источник: https://losst.ru/ustanovka-postgresql-ubuntu-16-04**

**При создании роли PostgreSQL, базы данных, таблиц, вставки и удалении данных используйте собственные примеры, данные!!!**