**Тема занятия №29: Работа с базой данных. Big data**

**1. Установка PostgreSQL**

PostgreSQL — это бесплатная объектно-реляционная СУБД с мощным функционалом, который позволяет конкурировать с платными базами данных, такими как Microsoft SQL, Oracle. PostgreSQL поддерживает пользовательские данные, функции, операции, домены и индексы. В данной статье мы рассмотрим установку и краткий обзор по управлению базой данных PostgreSQL. Мы установим СУБД PostgreSQL в Windows 10, создадим новую базу, добавим в неё таблицы и настроим доступа для пользователей. Также мы рассмотрим основы управления PostgreSQL с помощью SQL shell и визуальной системы управления PgAdmin. Надеюсь эта статья станет хорошей отправной точкой для обучения работы с PostgreSQL и использованию ее в разработке и тестовых проектах.

Для установки PostgreSQL перейдите на сайт https://www.postgresql.org и скачайте последнюю версию дистрибутива для Windows, на сегодняшний день это версия PostgreSQL 11 (в 11 версии PostgreSQL поддерживаются только 64-х битные редакции Windows). После загрузки запустите инсталлятор.



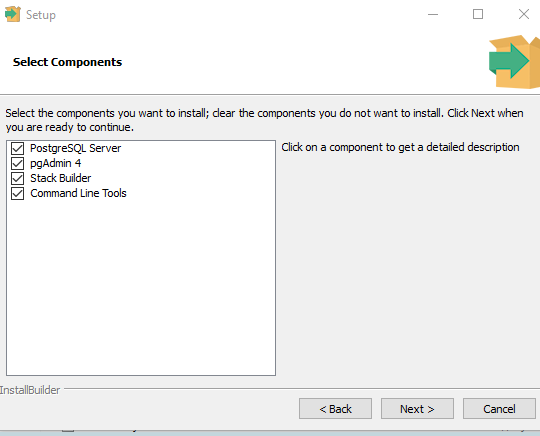
В процессе установки установите галочки на пунктах:

PostgreSQL Server – сам сервер СУБД

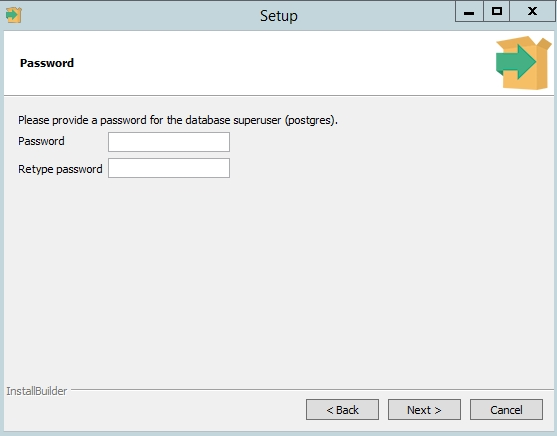
PgAdmin 4 – визуальный редактор SQL

Stack Builder – дополнительные инструменты для разработки (возможно вам они понадобятся в будущем)

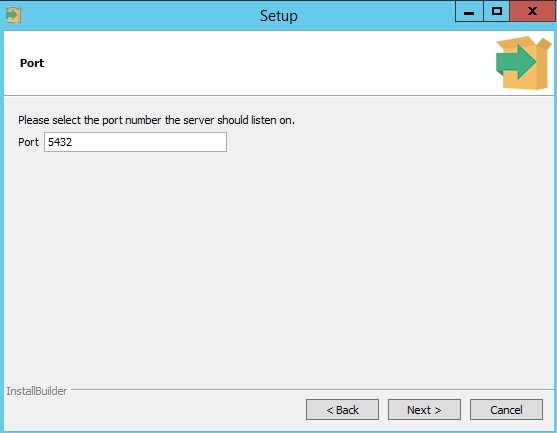
Command Line Tools – инструменты командной строки.



Установите пароль для пользователя postgres (он создается по умолчанию и имеет права суперпользователя).



По умолчанию СУБД слушает на порту 5432, который нужно будет добавить в исключения в правилах фаерволла.



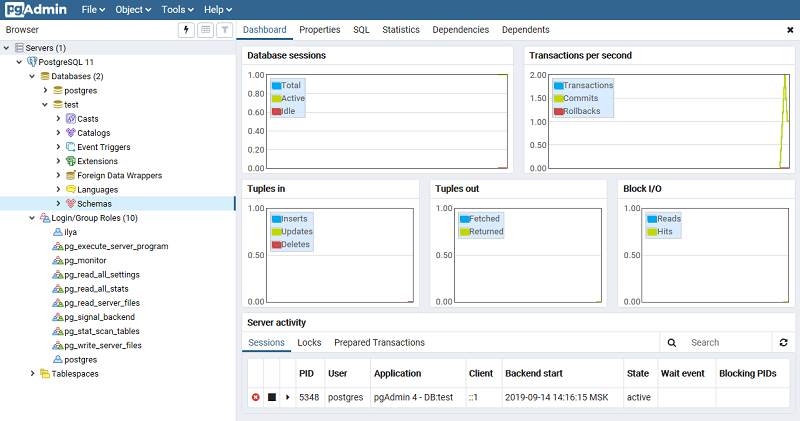
PgAdmin: Визуальный редактор для PostgresSQL

Редактор PgAdmin служит для упрощения управления базой данных PostgresSQL в понятном визуальном режиме.

Для запуска редактора запустите PgAdmin 4 в меню Пуск.

Для доступа нужно ввести пароль суперпользователя postgres.

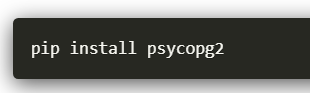
В панели Servers вы можете раскрыть список активных БД.



**2. Развёртывание адаптера и подключение к базе данных**

Установка Psycopg2 с помощью pip

Для начала нужно установить текущую версию Psycopg2 для использования PostgreSQL в Python. С помощью команды pip можно установить модуль в любую операцию систему: Windows, macOS, Linux:



Подключение к базе данных PostgreSQL из Python

Вот какие аргументы потребуются для подключения:

Имя пользователя: значение по умолчанию для базы данных PostgreSQL – postgres.

Пароль: пользователь получает пароль при установке PostgreSQL.

Имя хоста: имя сервера или IP-адрес, на котором работает база данных. Если она запущена локально, то нужно использовать localhost или 127.0.0.0.

Имя базы данных: в этом руководстве будем использовать базу postgres\_db.

Шаги для подключения:

Использовать метод connect() с обязательными параметрами для подключения базы данных.

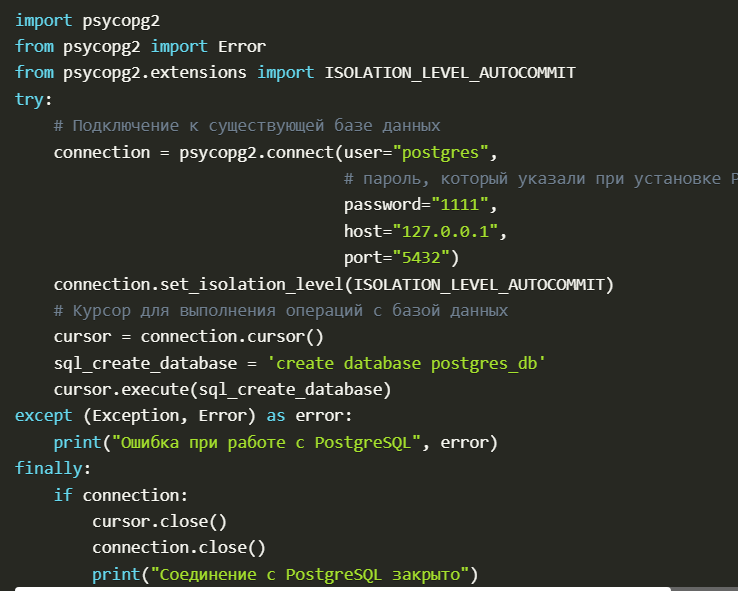
Создать объект cursor с помощью объекта соединения, который возвращает метод connect. Он нужен для выполнения запросов.

Закрыть объект cursor и соединение с базой данных после завершения работы.

Перехватить исключения, которые могут возникнуть в процессе.

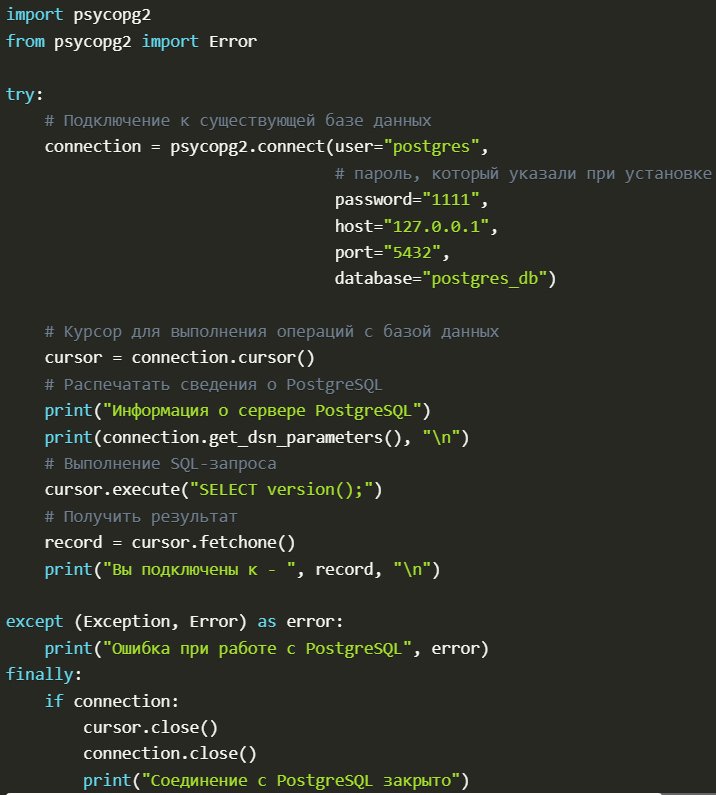
Создание базы данных PostgreSQL с Psycopg2

Для начала создадим базу данных на сервере. Во время установки PostgreSQL вы указывали пароль, его нужно использовать при подключении.



Пример кода для подключения к базе данных PostgreSQL из Python

Для подключения к базе данных PostgreSQL и выполнения SQL-запросов нужно знать название базы данных. Ее нужно создать прежде, чем пытаться выполнить подключение.



**1. Использование блоков try-except для работы с ошибки при работе с базой данных. Транзакции**

В большинстве случаев выполнять операцию в базе данных нужно лишь после завершения некой другой операции. Например, в банковской системе перевести сумму со счета А на счет Б можно только после снятия средств со счета А.

В транзакции или все операции выполняются, или не выполняется ни одна из них. Таким образом требуется выполнить все операции в одной транзакции, чтобы она была успешной.

Транзакции PostgreSQL обрабатываются объектом соединения. Он отвечает за сохранение изменений или откат в случае неудачи.

С помощью объекта cursor выполняются операции в базе данных. Можно создать неограниченное количество объектов cursor из объекта connection. Если любая из команд объекта cursor завершается неудачно, или же транзакция отменяется, то следующие команды не выполняются вплоть до вызова метода connection.rollback().

Объект соединения отвечает за остановку транзакции. Это можно сделать с помощью методов commit() ли rollback().

После использования метода commit() изменения сохраняются в базе данных.

С помощью метода rollback() можно откатить изменения.

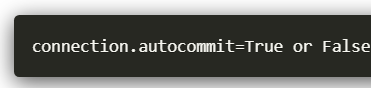
Примечание: вызов метода closе() или любого другого, отвечающего за уничтожение объекта соединения, приводит к неявному вызову rollback(), вследствие чего все изменения откатываются.

Connection.autocommit

По умолчанию соединение работает в режиме автоматического сохранения, то есть свойство auto-commit равно True. Это значит, что при успешном выполнении запроса изменения немедленно сохраняются в базу данных, а откат становится невозможным.

Для выполнения запросов в транзакции это свойство нужно отключить. Для этого нужно сделать connection.autocommit=False. В этом случае будет возможность откатить выполненный запрос к оригинальному состоянию в случае неудачи.

Синтаксис autocommit:



Connection.commit

Если все операции в транзакции завершены, используйте connection.commit() для сохранения изменений в базе данных. Если метод не использовать, то все эффекты взаимодействия с данными не будут применены.

Синтаксис commit:



Connection.rollback

Если в транзакции хотя бы одна операция завершается неудачно, то отменить изменения можно с помощью connection.rollback().

Синтаксис rollback:



Пример управления транзакциями PostgreSQL из Python

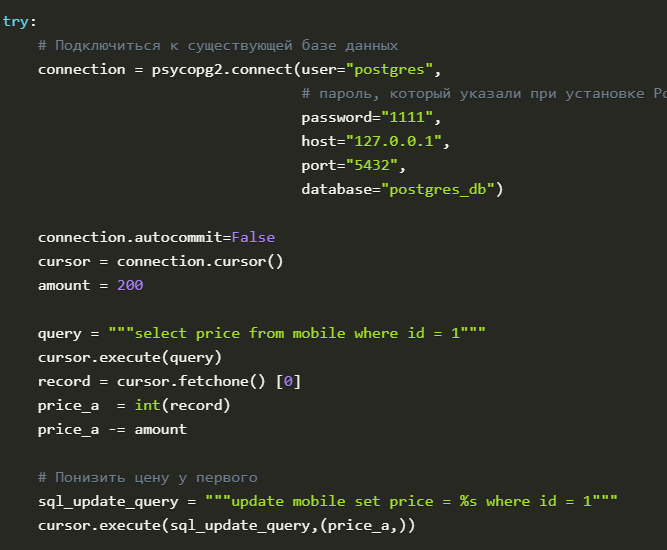
Отключите режим автосохранения (auto-commit).

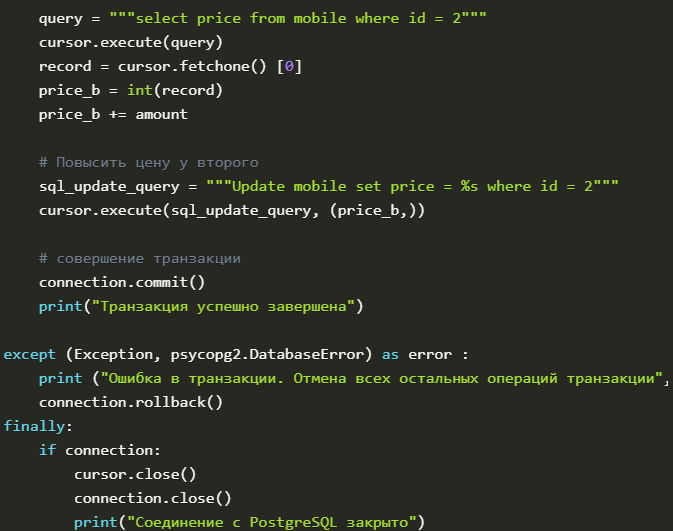
Если все операции были выполнены успешно, используйте connection.commit() для их сохранения в базе данных.

Если какая-то из операций была завершена неудачно, то откатиться к последнему состоянию можно с помощью connection.rollback().

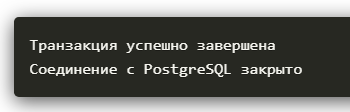
Примечание: транзакция остается открытой до явного вызова commit() или rollback().

Давайте понизим цену на один телефон и повысим на второй.

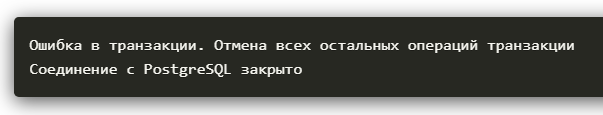




Вы получите следующий вывод после успешного завершения всех операций в транзакции.



Если хотя бы одна из операций будет завершена с ошибкой, то вывод будет таким.



Инструкция With для управления транзакциями

Для создания транзакции внутри определенного блока в коде используйте инструкцию with.

Что делает инструкция with?Соединения и объекта cursor в psycopg2 — это всего лишь контекстные менеджеры, которые можно использовать с with. Основное преимущество в том, что это позволяет не писать явно commit или rollback.

Синтаксис:

