ГБУ ДО ДЮТТ

**Инструкция по подключению к базе данных**

2024 г.

**Создание базы данных**

**1. Установите PostgreSQL и необходимые библиотеки**

* **PostgreSQL**: Убедитесь, что PostgreSQL установлен и настроен на вашем сервере. Установите его с официального сайта.
* **Python и библиотека psycopg2**: Убедитесь, что у вас установлен Python (рекомендуется версия 3.8 или выше). Установите библиотеку psycopg2 с помощью команды:

pip install psycopg2

**2. Настройте сервер PostgreSQL**

1. **Разрешите подключения:**

* Откройте файл postgresql.conf и убедитесь, что параметр listen\_addresses установлен, например:

***listen\_addresses = '\*'***

* Перейдите в файл pg\_hba.conf и добавьте запись для доступа:

***host all all 0.0.0.0/0 md5***

1. **Перезапустите сервер PostgreSQL:**

***sudo systemctl restart postgresql***

**3. Запустите скрипт**

1. **Откройте терминал или командную строку и выполните команду:**

***python “название файла”.py***

1. **Скрипт запросит следующие данные:**

* **Имя администратора:** Укажите имя пользователя PostgreSQL с правами администратора (например, postgres).
* **Пароль администратора:** Введите пароль этого пользователя (вводится при установке PostgreSQL если имя пользователя postgres).
* **Адрес сервера:** Если вы работаете локально, нажмите Enter (по умолчанию используется localhost).
* **Порт:** Нажмите Enter, чтобы использовать стандартный порт PostgreSQL 5432.
* **Имя нового пользователя:** Укажите имя нового пользователя, которого вы хотите создать.
* **Пароль нового пользователя:** Укажите пароль для нового пользователя.
* **Имя базы данных:** Укажите название базы данных, которую вы хотите создать.

**4. Описание работы скрипта**

1. **Подключение к базе данных postgres:**

* Скрипт подключается к базе данных postgres от имени администратора для выполнения административных команд.

1. **Проверка существования пользователя:**

* Если пользователь уже существует, вы получите сообщение об этом.
* Если пользователь отсутствует, скрипт создаст его и назначит ему права на создание баз данных.

1. **Проверка существования базы данных:**

* Если база данных уже существует, вы получите сообщение об этом.
* Если база данных отсутствует, скрипт создаст её и назначит нового пользователя владельцем.

1. **Закрытие подключения:**

* После завершения всех операций скрипт закроет соединение с базой данных.

#### ****5. Пример выполнения****

**Вот как может выглядеть работа скрипта:**

Введите параметры подключения к PostgreSQL для администратора:

Имя администратора: postgres

Пароль администратора: \*\*\*\*\*\*

Адрес сервера (по умолчанию localhost):

Порт (по умолчанию 5432):

Введите данные для нового пользователя:

Имя нового пользователя: new\_user

Пароль нового пользователя: new\_password

Введите имя создаваемой базы данных: new\_database

Пользователь 'new\_user' успешно создан.

Пользователю 'new\_user' даны права на создание баз данных.

База данных 'new\_database' успешно создана с владельцем 'new\_user'.

**Устранение ошибок**

* **Ошибка подключения (connection timed out)**:
  + Проверьте параметры подключения: IP-адрес, порт, и доступность сервера.
* **Ошибка авторизации (authentication failed)**:
  + Убедитесь, что вы ввели правильный пароль и логин администратора.
* **Пользователь или база уже существуют**:
  + Это сообщение означает, что пользователь или база данных уже созданы ранее. Скрипт их не дублирует.

**Создание таблицы в базе данных**

**1. Настройте базу данных**

**Перед выполнением скрипта:**

* Создайте базу данных, если она еще не существует (используйте предыдущую инструкцию).

**2. Запустите скрипт**

1. **Выполните скрипт в терминале/командной строке:**

python create\_table.py

1. **Программа запросит следующие параметры подключения:**

* **Имя базы данных**: Имя существующей базы данных, к которой нужно подключиться.
* **Имя пользователя**: ”Имя нового пользователя”, которое создали в ***пункте 3*** (или по умолчанию postgres).
* **Пароль**: Пароль PostgreSQL для указанного пользователя.
* **Адрес сервера**: Если сервер работает локально, нажмите Enter (по умолчанию будет использован localhost).
* **Порт**: Введите порт сервера (или нажмите Enter, чтобы использовать стандартный порт PostgreSQL 5432).

**3. Описание работы кода**

1. **Создание таблицы:**

* Скрипт проверяет, существует ли таблица в базе данных. Если её нет, создаётся новая таблица.
* Таблица содержит несколько столбцов с различными типами данных, включая уникальный идентификатор, текстовые данные и дату.
* Устанавливается ограничение уникальности для определённых столбцов, чтобы предотвратить дублирование записей.

1. **Добавление данных:**

* Программа запрашивает у пользователя ввод значений для нескольких полей.
* Введённые данные сохраняются в таблице. Если запись с такими же уникальными характеристиками уже существует, вместо дублирования выполняется обновление одной из колонок.

1. **Вывод данных:**

* По завершении работы все записи из таблицы извлекаются и отображаются в удобном для чтения формате.

**4. Пример выполнения**

**Пример взаимодействия с программой:**

Введите параметры подключения к PostgreSQL:

Имя базы данных: test\_db

Имя пользователя: postgres

Пароль: \*\*\*\*\*\*

Адрес сервера (по умолчанию localhost):

Порт (по умолчанию 5432):

Таблица 'employees' создана или уже существует.

Введите данные для добавления в таблицу. Чтобы завершить, введите 'stop'.

Имя сотрудника (или 'stop' для завершения): Иван Иванов

Должность: Менеджер

Дата найма (YYYY-MM-DD, или пусто для текущей даты): 2024-01-01

Данные сотрудника Иван Иванов добавлены или обновлены.

Имя сотрудника (или 'stop' для завершения): stop

Данные в таблице 'employees':

(1, 'Иван Иванов', 'Менеджер', '2024-01-01')

**5. Устранение ошибок**

1. **Ошибка подключения:**

* Проверьте параметры подключения: логин, пароль, имя базы, IP-адрес и порт.
* Убедитесь, что PostgreSQL настроен для удаленного доступа, если вы подключаетесь не с локального компьютера.

1. **Ошибка OperationalError:**

* Эта ошибка возникает при проблемах с сетью или сервером. Проверьте, работает ли сервер PostgreSQL.

1. **Ошибка в SQL-запросах:**

* Проверьте корректность введённых данных (например, формат даты).

1. **Ошибка уникальности:**

* Если вы добавляете дублирующую запись, она будет обновлена(можно исправить в ручную).