ИИКС

Лабораторная работа №4: «PostgreSQL»

Грущин Илья

Б21-515

2024 г

Задание

1. Выполнить установку сервера PostgreSQL на реальную или виртуальную машину;
2. Выбрать клиент (psql,…) и осуществить подключение к установленному серверу;
3. Реализовать базу данных, аналогичную реализованной в рамках работ 1-1 … 1–3. Попробовать повторно выполнить все запросы. Отразить в отчёте все моменты, в которых поведение PostgreSQL не соответствует поведению SQLite3 (если они будут);

Установка PostgreSQL

В качестве клиента PostgreSQL выбран pgAdmin 4, установленный вместе с сервером PostgreSQL на реальную машину на базе ОС Windows 11.

Для исполнения результата SQL запросов вручную использован SQL Query Tool (рисунок 1)

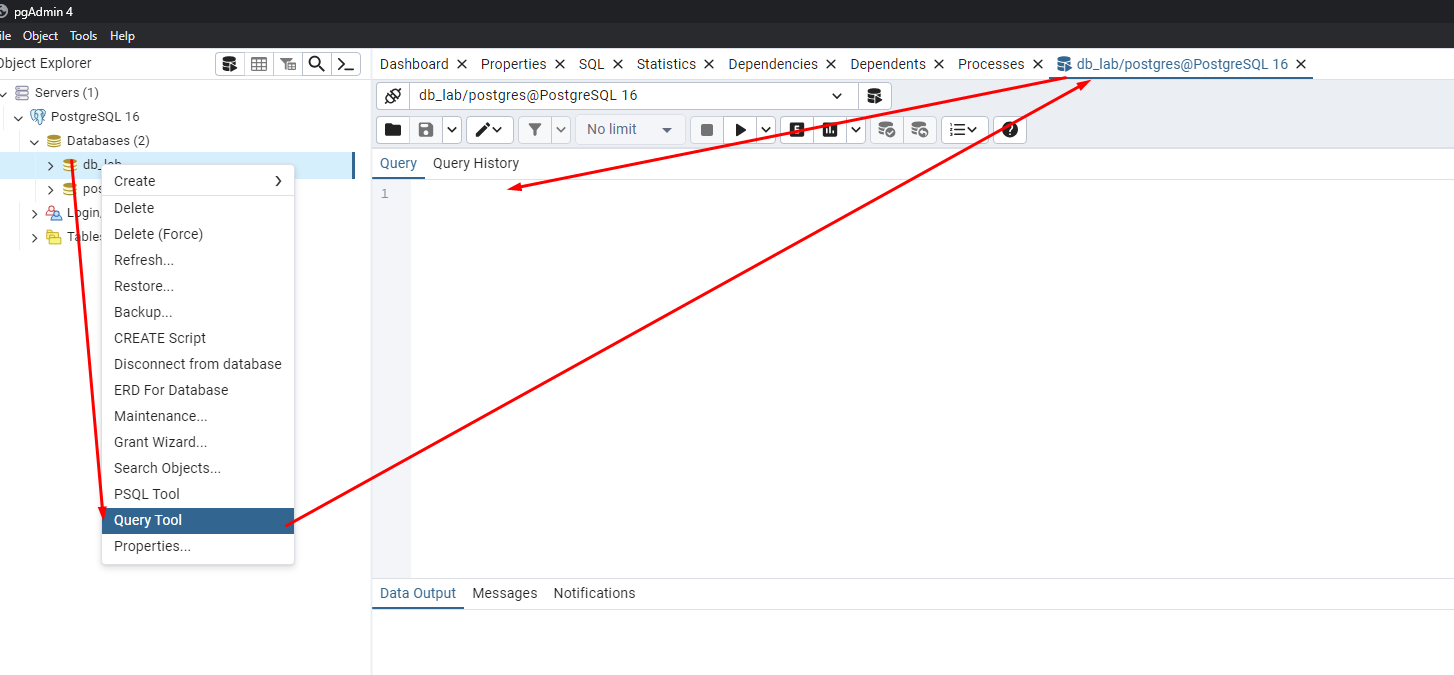


Рисунок 1.

Для автоматизированного создания, заполнения и сброса базы данных используется найденный на просторах сети и модифицированный под себя python-сценарий, приведенный в приложении. Данный сценарий действует на базе библиотеки pg8000.native

Для работы PostgreSQL со скриптом проведена настройка сервера и создание базы данных с помощью pgAdmin 4. Создан пользователь lab (рисунок 2)

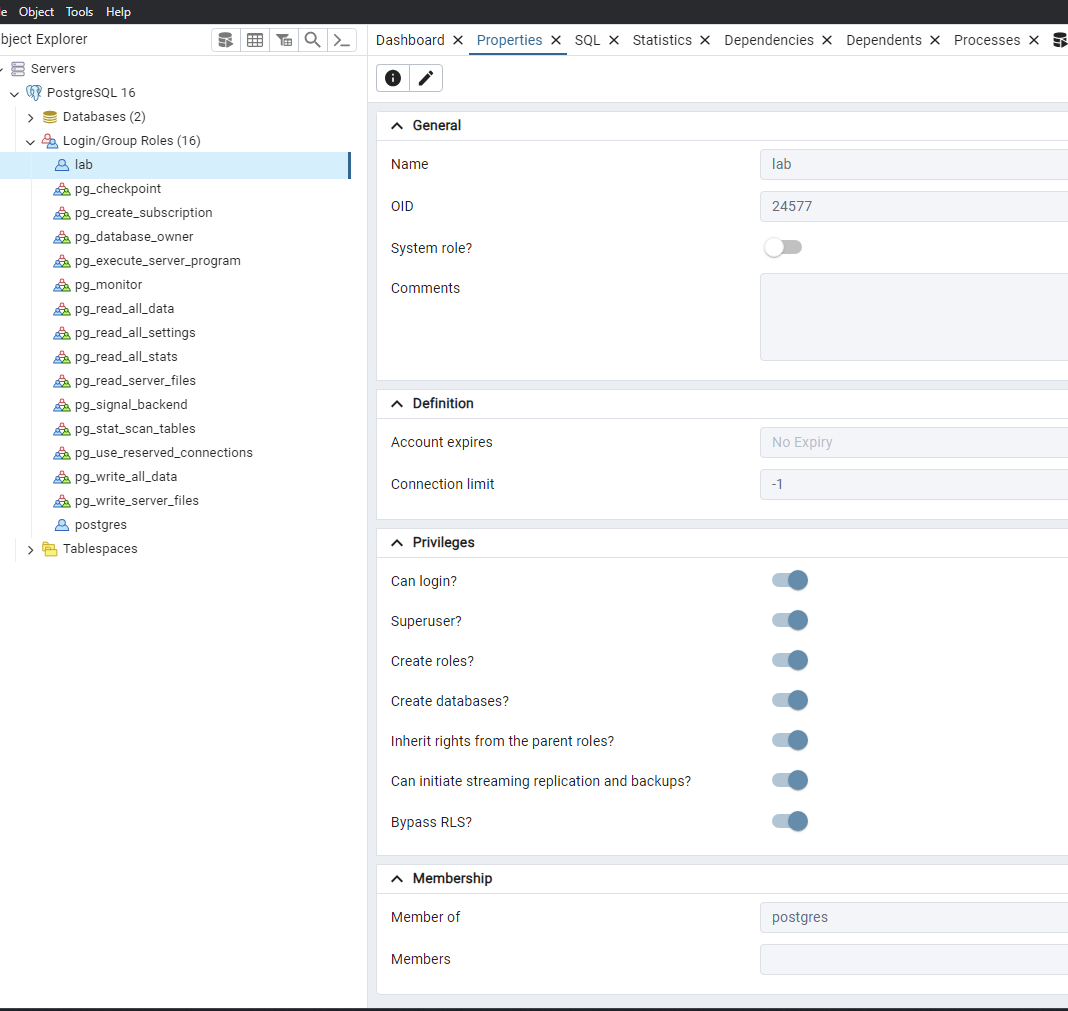


Рисунок 2

Создана база данных db\_lab, в которой наделен необходимыми правами пользователь lab (рисунок 3).

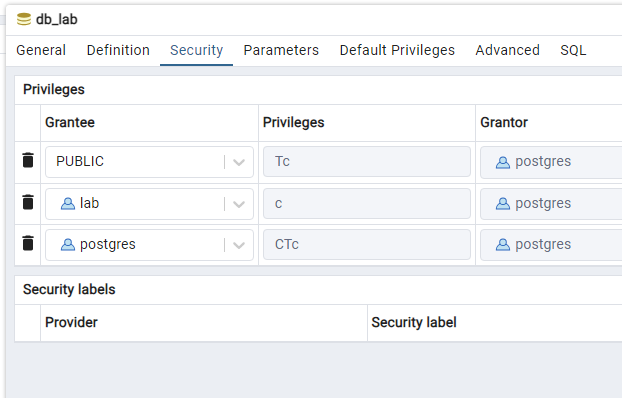


Рисунок 3

В дальнейшем работа производилась с помощью упомянутого сценария и SQL Query Tool.

Заключение

В данной работе проведена установка СУБД PostgreSQL (с клиентом pgAdmin 4) и её настройка для последующей работы с python-сценарием автоматизированного сброса, создания структуры и заполнения базы данных.

Впоследствии данная СУБД была использована при выполнении лабораторных работ 1-1, 1-2, 1-3, о чем свидетельствуют отчёты о них.

Приложение

**db\_lib.py**

import pg8000.native

def connect\_to\_db(host, port, database, user, password) -> pg8000.native.Connection:

    try:

        conn = pg8000.native.Connection(

            user=user,

            password=password,

            host=host,

            port=port,

            database=database

        )

        print("Connection to PostgreSQL DB successful")

        return conn

    except Exception as e:

        print(f"Error connecting to the database: {e}")

        return None

def execute\_sql\_file(conn: pg8000.native.Connection, file\_path: str):

    try:

        with open(file\_path, 'r') as file:

            sql\_script = file.read()

        print(conn.execute\_simple(sql\_script))

        print("SQL script executed successfully")

        print()

    except Exception as e:

        print(f"Error executing SQL script: {e}")

db\_config = {

    "host": "127.0.0.1",

    "port": "5432",

    "database": "db\_lab",

    "user": "lab",

    "password": "no-pass"

}

**init\_db.py**

from db\_lib import \*

def main():

    structure = "../sql/table\_structure/"

    data = "../sql/data/"

    sql\_file\_paths = [

        structure + "reset.sql",

        structure + "developers.sql",

        structure + "publishers.sql",

        structure + "genres.sql",

        structure + "games.sql",

        structure + "game\_genres.sql",

        structure + "users.sql",

        structure + "ratings.sql",

        structure + "developer\_publisher\_relationships.sql",

        structure + "relations.sql",

        data + "fill\_db.sql"

    ]

    conn = connect\_to\_db(\*\*db\_config)

    if conn:

        # Executing SQL commands from the file

        for sql\_file\_path in sql\_file\_paths:

            execute\_sql\_file(conn, sql\_file\_path)

        # Closing the database connection

        conn.close()

        print("Database connection closed.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Запуск осуществляется интерпретатором python:

PS C:\Users\USER\Documents\GitHub\Mephi\_42\_Databases\scripts> python .\init\_db.py