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)

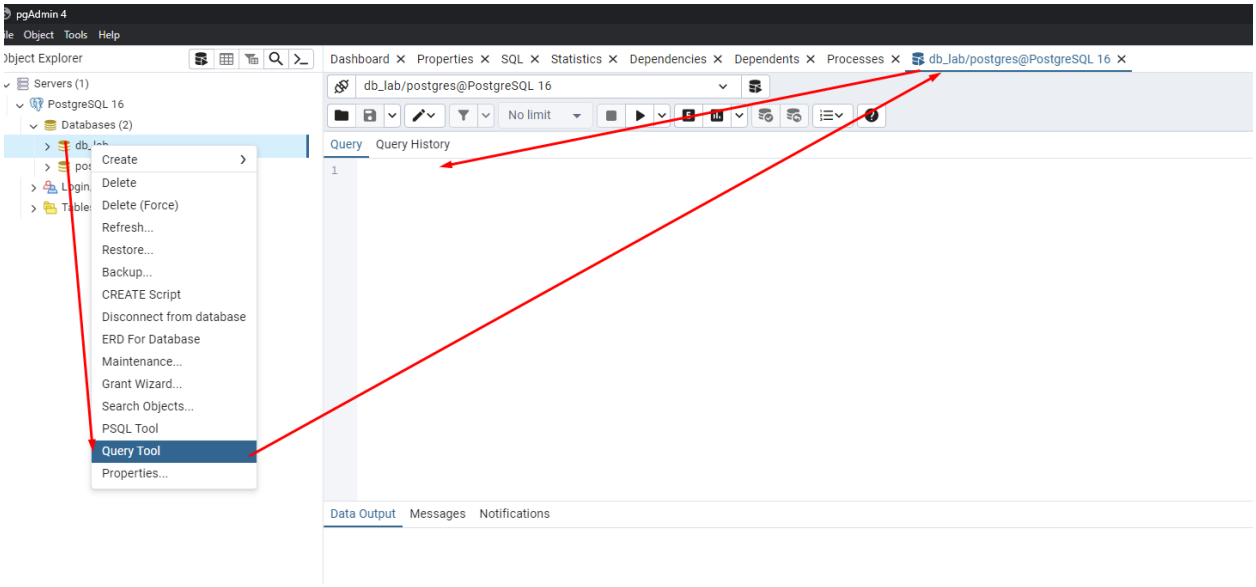

Рисунок 1.

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)

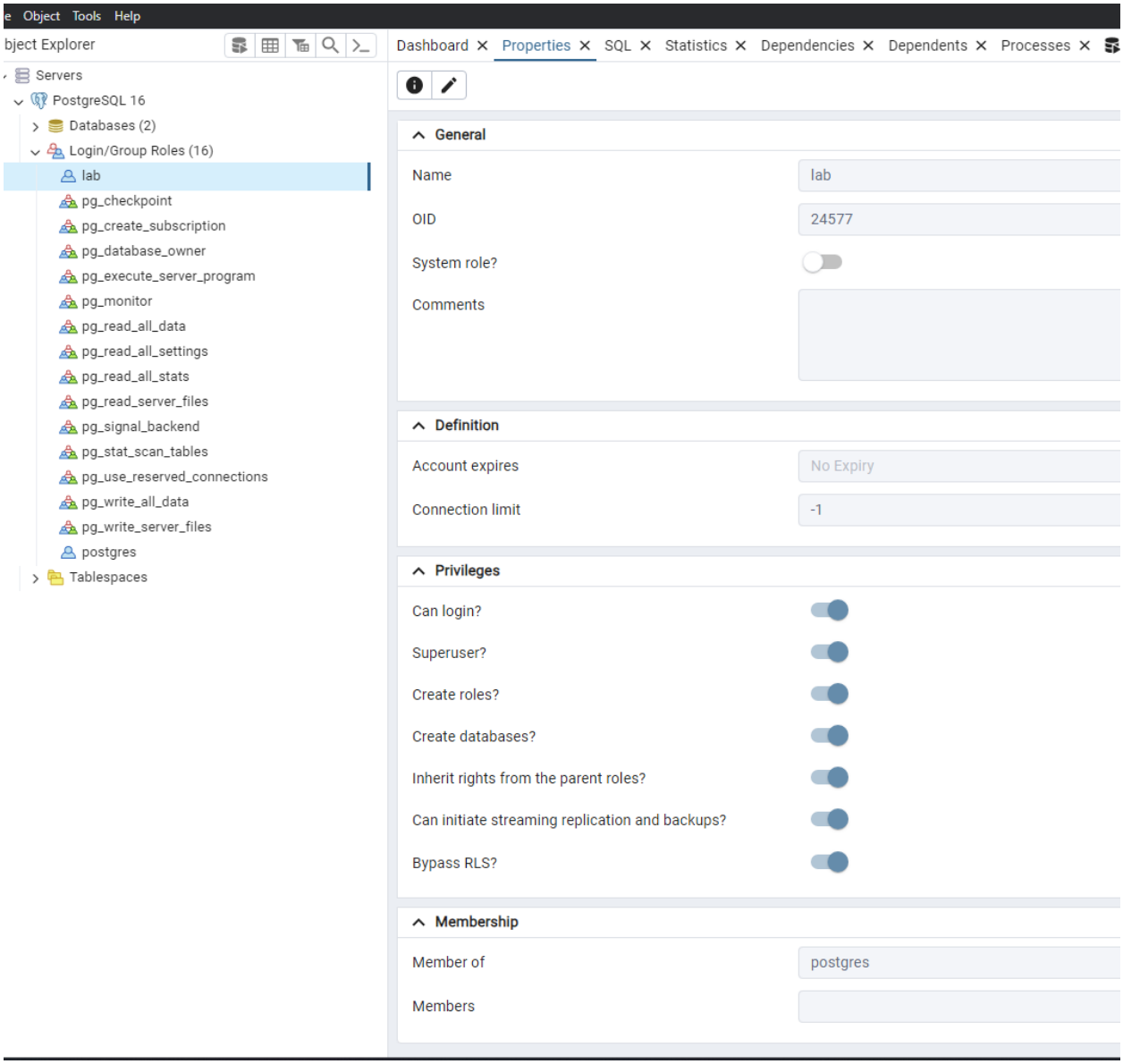

Рисунок 2

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

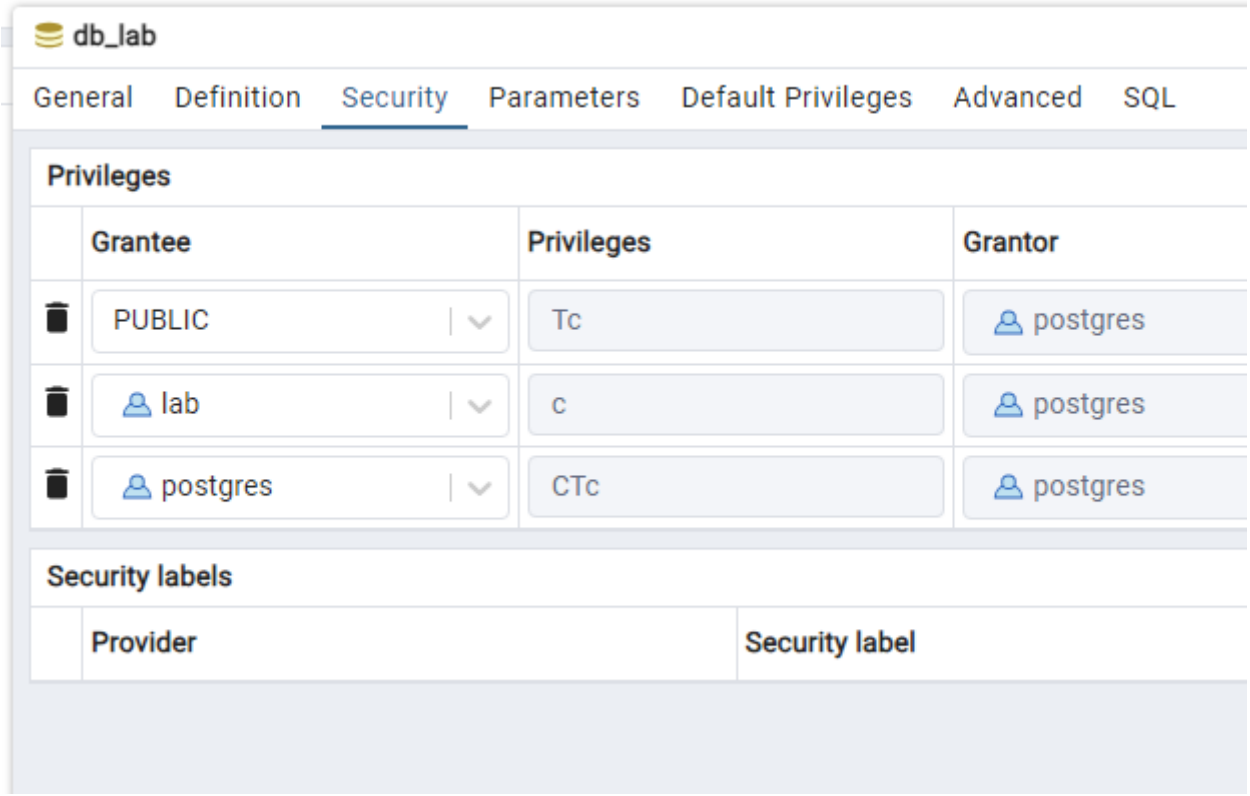

Рисунок 3

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