Практика 2

Работа с форматами данных в C++

Загрузка дампа в PostgreSQL

Цель практики:

1. Научиться работать с форматами данных (csv, xml, json) в C++
2. Научиться создавать скрипт для загрузки данных в PostgreSQL
3. Научиться создавать и разворачивать бэкапы в PostgreSQL

Задание:

1. На портале открытых данных выбрать и скачать файл, на основе которого будет создан дамп. Формат файла: csv, xml, json (один на выбор)

   Портал открытых данных: https://data.gov.ru/opendata
2. Написать программу, которая будет читать данные из файла и на их основе формировать SQL-скрипт
3. Развернуть созданный дамп в СУБД

Что нужно сдать для зачёта по практике:

1. Ссылка на набор данных (или сам файл с данными)
2. Текст программы
3. Созданный программой скрипт

Подсказки для выполнения практики

Дополнительные библиотеки и примеры использования

1. Для работы с csv никакие дополнительные библиотеки не используются
2. Для работы с XML используется библиотека tinyxml2

   https://github.com/leethomason/tinyxml2

   Для работы нужно добавить файлы tinyxml2.h и tinyxml2.cpp в папку с программой
3. Для работы с json используется библиотека nlohmann/json

   https://github.com/nlohmann/json/

   Для работы нужно добавить файл json.hpp (single_include/nlohmann/json.hpp) в папку с программой

Пример работы с tinyxml2

Файл, который будет парситься

```
<?xml version="1.0" encoding="UTF-8"?>
<register>
   <record>
      <smiName>Электрические сети и системы</smiName>
      <certificateNum>РП № 52</certificateNum>
   </record>
   <record>
      <smiName>Ухтышка</smiName>
      <certificateNum>РП № 61</certificateNum>
   </record>
</register>
```

Пример работы с tinyxml2: чтение файла и вывод данных на экран

```
#include "tinyxml2.h"
#include <iostream>
#include <Windows.h>

int main() {
      SetConsoleOutputCP( CP_UTF8 );

      tinyxml2::XMLDocument doc;
      doc.LoadFile( "data.xml" );
      const tinyxml2::XMLElement* response, *items, *post;

      response = doc.FirstChildElement("register");

      for (
            post = response->FirstChildElement("record");
        post;
        post = post->NextSiblingElement("record")
        ) {
        const tinyxml2::XMLElement* smiName, *certificateNum;
        smiName = post->FirstChildElement("smiName");
        certificateNum = post->FirstChildElement("certificateNum");
```

```
        std::cout << smiName->GetText() << std::endl;
        std::cout << certificateNum->GetText() << std::endl;
    }

      return 0;
}
```

Пример работы с nlohmann/json

Файл, который будет парситься

```
{
 "elements":[
   {
      "global_id":2812434,
      "Name":"Мемориальная доска памяти ополченцев Народного комиссариата
иностранных дел РФ",
   },
   {
      "global_id":2812437,
      "Name":"Мемориальная доска Бугримовой Ирине Николаевне",
   ]
}
```

Пример работы с nlohmann/json: чтение файла и вывод данных на экран

```
#include "json.hpp"
#include <iostream>
#include <fstream>
#include <Windows.h>

using json = nlohmann::json;

int main() {
      SetConsoleOutputCP(CP_UTF8);

      std::ifstream json_file("data2.json");
      json json_object;
      json_file >> json_object;

      std::cout << json_object.at("elements") << '\n';
      for(auto &array : json_object["elements"]) {
            int global_id = array["global_id"];
            std::cout << global_id << std::endl;
            std::string name = array["Name"];
            std::cout << name << std::endl;
      }

      return 0;
}
```

Итоговый скрипт

В скрипте должно быть:

- создание таблицы, куда потом должны добавляться данные (CREATE TABLE)
- добавление данных (INSERT)

Пример скрипта (добавление одной записи):

```
CREATE TABLE article (
    id bigint NOT NULL,
    link_id bigint,
    title text NOT NULL,
    author text NOT NULL,
);
CREATE SEQUENCE article_id_seq START WITH 1 INCREMENT BY 1 NO MINVALUE NO MAXVALUE CACHE 1;
ALTER SEQUENCE article_id_seq OWNED BY article.id;
INSERT INTO article(link_id, title, author) VALUES (1, 'Заголовок 1', 'Автор 1');
```

CREATE SEQUENCE - команда, которая создаёт последовательность для автоматического увеличения id для каждой добавленной строчки

ALTER SEQUENCE article_id_seq OWNED BY article.id; - назначается "владелец" последовательности, т.е. для какого столбца применяется выбранная последовательность

Как запустить скрипт?

База данных, для которой будет запускаться скрипт, должна быть предварительно создана.

(Как создаётся БД можно посмотреть в слайдах к практике 1)

1. Для Windows
    1. Перейти в папку с командой psql.exe

       cd c:\programs\postgres12\bin
    2. Запустить выбранный скрипт

       psql.exe -U postgres -d my_database -f c:\backup\backup_14_01_4.sql

       -U postgres - запуск идёт под пользователем postgres

       -d my_database - скрипт запускается для БД my_database

       -f c:\backup\backup_14_01_4.sql - путь к скрипту
2. Для Linux

   sudo -u postgres psql

   psql my_database < backup_file