Университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1.5 по Системному программному обеспечению

Группа: P34112

Выполнили: Горшков Артем Владимирович

Рябикин Илья Леонидович

Преподаватель: Кореньков Юрий Дмитриевич

Санкт-Петербург

2021 год

Задание

Разработать способ организации данных в файле, позволяющий хранить, выбирать и гранулярно обновлять наборы записей общим объёмом от 10GB соответствующего варианту вида. Реализовать модуль или библиотеку для работы с ним в режиме курсора.

Используя данный способ сериализации, воспользоваться существующей библиотекой для описания схемы и реализации модуля, обеспечивающего функционирование протокола обмена запросами создания, выборки, модификации и удаления данных, и результатами их выполнения.

Использовать средство синтаксического анализа по выбору, реализовать модуль для разбора некоторого подмножества языка запросов по выбору в соответствии с вариантом формы данных. Должна быть обеспечена возможность описания команд создания, выборки, модификации и удаления данных.

Используя созданные модули разработать в виде консольного приложения две программы: клиентскую и серверную части. Серверная часть – получающая по сети запросы и операции описанного формата и выполняющая их над файлом, организованным в соответствии с разработанным способом. Имя файла данных для работы получать с аргументами командной строки, создавать новый в случае его отсутствия.Клиентская часть – получающая от пользователя команду, пересылающая её на сервер, получающая ответ и выводящая его в человекопонятном виде.

Исходный код

https://gitlab.se.ifmo.ru/ArtemGorshkov/lab1_5_spo

Описание и пример работы программы

Для запуска сервера: `./server file.db`

Для запуска клиента: `./client`

```
> create table two (col1 str, col2 num, col3 int);
Table was created.
> insert into two values ('qwe', 4.3, 4);
Row was inserted.
> insert into two values ('qdase', 12.2, 4);
Row was inserted.
```

```
> select * from two
+---------+------+------+
|    col1 | col2 | col3 |
+---------+------+------+
| 'qdase' | 12.2 |    4 |
+---------+------+------+
|   'qwe' |  4.3 |    4 |
+---------+------+------+
> insert into two values ('qswe', 1.2, 16);
Row was inserted.
> select * from two
+---------+------+------+
|    col1 | col2 | col3 |
+---------+------+------+
|  'qswe' |  1.2 |   16 |
+---------+------+------+
| 'qdase' | 12.2 |    4 |
+---------+------+------+
|   'qwe' |  4.3 |    4 |
+---------+------+------+
>
```

```
> create table tab1 (col1 int, col2 int)
Table was created.
> create table tab2 (col3 int, col4 int)
Table was created.
> insert into tab1 values (1,2)
Row was inserted.
> insert into tab2 values (2,2)
Row was inserted.
> insert into tab2 values (100,2)
Row was inserted.
> select * tab1
Parsing error: syntax error.
> select * from tab1
+------+------+
| col1 | col2 |
+------+------+
|    1 |    2 |
+------+------+
> select * from tab2
+------+------+
| col3 | col4 |
+------+------+
|  100 |    2 |
+------+------+
|    2 |    2 |
+------+------+
> select * from tab2 join tab1 on col4=col2 where col3>50
+------+------+
| col3 | col4 |
+------+------+
|  100 |    2 |
+------+------+
```

Логирование со стороны сервера:

```
Request:
 <?xml version="1.0"?>
<request>
  <action>4</action>
  <table>tab2</table>
  <joins>
    <join>
      <table>tab1</table>
      <t_column>col4</t_column>
      <s_column>col2</s_column>
    </join>
  </joins>
  <where>
    <op>3</op>
    <column>col3</column>
    <value type="0">50</value>
  </where>
</request>

Response:
 <?xml version="1.0"?>
<response>
  <columns>
    <column>col3</column>
    <column>col4</column>
  </columns>
  <values>
    <row>
      <value type="0">100</value>
      <value type="0">2</value>
    </row>
  </values>
</response>
```

Вывод

В результате работы было разработано консольное приложение поддерживающее базовые функции работы с таблицами с помощью SQL подобного языка запросов. В качестве протокола взаимодействия клиента и сервера была использован XML. К плюсам данного протокола можно отнести большое количество проверенных временем библиотек, которые позволяют удобно работать с DOM. В нашем случае выбор пал на библиотеку libxml. Минусом является тот факт, что все значения между тегами передаются и принимаются со стороны сервера только как строки. Решением этой проблемы стало явное указание типа данных в атрибуте (пример можно увидеть выше на скрине). В хранилище основной используемой структурой данных стали связные списки, так как это наиболее простой и понятный способ хранения.