

Санкт-Петербургский Национальный Исследовательский
Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №3 по LLP
Вариант libXML

Юнусов Роман Ильдарович
Группа Р33102
Преподаватель Юрий Кореньков

Г.Санкт-Петербург

2024 г.

Задание

Объединить лабораторные работы 1 и 2 в клиент-серверную архитектуру, сериализовав данные с использованием xml.

Протестировать работу с использованием реальной базы данных neo4j.

Ход работы.

Было принято решение использовать Libxml2, библиотека подключалась через графический интерфейс visual studio 2019, для реализации серверной части использовалась виндовская реализация сокетов.

Пример базы данных -

<https://demo.neo4jlabs.com:7473/browser/?dbms=neo4j://northwind@demo.neo4jlabs.com&db=northwind>

Логин и пароль – northwind

Т.к. у меня реляционная БД, то нужно было отдельно вытащить данные каждой таблицы

Для покупателей

MATCH (c:Customer)

RETURN DISTINCT c.customerID as id, c.contactName as name

Для заказов

MATCH (o:Order) WHERE toInteger(o.shipPostalCode) is not null

RETURN DISTINCT o.customerID as id, toInteger(o.shipPostalCode) as zip, o.shipCountry as country

В качестве тестов были использованы select и join

Select

MATCH (o:Order) WHERE toInteger(o.shipPostalCode) is not null and

toInteger(o.shipPostalCode) > 15000 and toInteger(o.shipPostalCode) < 50000

RETURN DISTINCT o.customerID as id, toInteger(o.shipPostalCode) as zip, o.shipCountry as country

Оригинальный вывод

	id	zip	country
1	"TOMSP"	44087	"Germany"
2	"MAGAA"	24100	"Italy"
3	"REGGC"	42100	"Italy"
4	"LAMAI"	31000	"France"
5	"WILMK"	21240	"Finland"

Мой вывод

```
<?xml version="1.0"?>
<root>
  <raw_data 0="'WILMK'" 1="21240" 2="'Finland'"/>
  <raw_data 0="'LAMAI'" 1="31000" 2="'France'"/>
  <raw_data 0="'REGGC'" 1="42100" 2="'Italy'"/>
  <raw_data 0="'MAGAA'" 1="24100" 2="'Italy'"/>
  <raw_data 0="'TOMSP'" 1="44087" 2="'Germany'"/>
</root>
```

JOIN

Тест

join query

MATCH (c: Customer)-[PURCHASED]->(o:Order) WHERE toInteger(o.shipPostalCode)
is not null and toInteger(o.shipPostalCode) < 90000 and o.shipCountry = "USA"
RETURN distinct c.contactName as name, toInteger(o.shipPostalCode) as zip, o.shipCountry
as country

Оригинальный вывод

	name	zip	country
1	"Paula Wilson"	87110	"USA"
2	"Art Braunschweiler"	82520	"USA"
3	"Jose Pavarotti"	83720	"USA"
4	"Liu Wong"	59801	"USA"

rted streaming 4 records after 9 ms and completed after 12 ms.

Мой вывод

```
<?xml version="1.0"?>
<root>
  <raw_data 0="'RATTC'" 1="'Paula Wilson'" 2="'RATTC'" 3="87110" 4="'USA'"/>
  <raw_data 0="'SAVEA'" 1="'Jose Pavarotti'" 2="'SAVEA'" 3="83720" 4="'USA'"/>
  <raw_data 0="'SPLIR'" 1="'Art Braunschweiler'" 2="'SPLIR'" 3="82520" 4="'USA'"/>
  <raw_data 0="'THECR'" 1="'Liu Wong'" 2="'THECR'" 3="59801" 4="'USA'"/>
</root>

> _
```

Также было принято решение протестировать решение с помощью db-fiddle и

https://github.com/ptom/northwind_psql/blob/master/northwind.sql

В клиентской части приложения есть фильтр, внутри которого файлы, с помощью которых можно построить как базы данных в моём приложении, так и базу данных на db-fiddle, и есть оба типа запросов.