

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 6
по дисциплине «Постреляционные базы данных»

Тема: «Работа с полуструктурированными данными
в формате XML»

ИСПОЛНИТЕЛЬ:

группа ИУ5-24М

Журавлев Н.В.

ФИО

подпись

"07" мая 2024 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В.

ФИО

подпись

" " 202_ г.

Москва - 2024

Цель работы

- Изучить языки запросов XPath и XQuery к XML-документам [1].
- Освоить методы работы с XML в постреляционных СУБД.
- Получить навыки экспорта в XML и запроса к XML-данным на примере СУБД PostgreSQL.

Задание

1. Через PgAdmin соединиться с PostgreSQL и создать базу данных. В БД создать две-три связанные таблицы по теме, выданной преподавателем. Открыть таблицы на редактирование и заполнить тестовыми данными. Не менее 5 записей в каждой таблице. Иметь поля с NULL, висящие относительно соединения записи в обеих таблицах и несколько дочерних записей к одной родительской.
2. В среде построения запросов PgAdmin продемонстрировать просмотр экспорт содержимого таблиц в xml в следующих вариантах:
 - a. все поля — элементы,
 - b. все поля — атрибуты,
 - c. добавление корневого элемента,
 - d. переименование строк,
 - e. получение xml-схемы по умолчанию,
 - f. отображение значений NULL.
3. В среде построения запросов PgAdmin продемонстрировать экспорт результата запроса с условием в XML и экспорт содержимого всех таблиц в xml произвольной структуры:
 - a. с атрибутами,
 - b. с дочерними элементами,
 - c. с атрибутами дочерних элементов.
4. В среде построения запросов создать сценарии для чтения xml из файла (взять xml документ сложной структуры, полученный ранее).
Выполнить запросы с условием на получение данных в виде таблиц.
5. В среде построения запросов создать сценарии для чтения xml из файла

(взять xml документ сложной структуры, полученный ранее). На языке XPath выполнить запросы:

- a. Проверки существования данных (атрибутов, элементов и их значений);
- b. Извлечения данных (атрибутов, элементов и содержимого);
- c. Получения фрагмента XML.

Ход работы

Создать БД (2-3 таблицы) и заполнить. Не менее 5 записей в каждой таблице. Иметь поля с NULL, висящие относительно соединения записи в обеих таблицах и несколько дочерних записей к одной родительской

Функцию `table_to_xml` можно рассматривать как средства экспорта в XML для созданных таблиц:

```
select table_to_xml('account',true,true,"")
select table_to_xml('diet',true,true,"")
select table_to_xml('history_diet',true,true,"")
```

Экспорт содержимого таблиц в xml: все поля – элементы

Следующий код извлекает строки из таблицы и создает XML-элементы, содержимое которых отражает значения полей таблицы:

```
select xmlelement(name account_info, login, password) from account
```

Экспорт содержимого таблиц в xml: все поля – атрибуты

Следующий код извлекает строки из таблицы и создает XML-элементы, в которых поля таблицы передаются как атрибуты элементов:

```
select xmlelement(name account, xmlattributes(height, weight, age)) from
account
```

Экспорт содержимого таблиц в xml: добавление корневого элемента

Следующий код извлекает строки из таблицы и создает XML-элементы с добавлением корневого элемента:

```
select xmlroot( xmlelement(name Root, xmlforest(height, weight, age)), version '1.1', standalone yes) from account
```

Экспорт содержимого таблиц в xml: переименование строк

Переименование делается с использованием слова as:

```
select xmlroot( xmlelement(name Root, xmlforest(login as height, weight, age)), version '1.1', standalone yes) from account
```

Экспорт содержимого таблиц в xml: получение xml схемы по умолчанию

Следующая функция выдаёт документы XML Schema, которые содержат схемы отображений:

```
select table_to_xml_and_xmlschema('account',true,true,"")
```

Экспорт содержимого таблиц в xml: отображение NULL выглядит следующим образом:

```
<id_diet xsi:nil="true"/>
```

XPath запрос: для проверки существования данных (атрибутов, элементов и их значений) используется xpath_exists:

```
select xpath_exists('/root/account/login[@name]', xmlroot(xmlelement(name Root, table_to_xml('account',true,true,"")), version '1.1', standalone yes))
select xpath_exists('/root/account/login', xmlroot(xmlelement(name Root, table_to_xml('account',true,true,"")), version '1.1', standalone yes))
select xpath_exists('/root/account/login[@name="name"]', xmlroot(xmlelement(name Root, table_to_xml('account',true,true,"")), version '1.1', standalone yes))
select xpath_exists('/root/account/login/text()', xmlroot(xmlelement(name Root, table_to_xml('account',true,true,"")), version '1.1', standalone yes))
```

XPath запрос: извлечения данных (атрибутов, элементов и содержимого)

```
select unnest(xpath('/root/account/@title', xmlelement(name Root,
table_to_xml('account',true,true,")), version '1.1', standalone yes)))

select unnest(xpath('/root/account/login/text()', xmlelement(name Root,
table_to_xml('account',true,true,")), version '1.1', standalone yes)))
```

XPath запрос: получение фрагмента XML с сайта

<https://www.postgresql.org/about/newsarchive/>

```
select unnest(xpath('//div[ @
class="container"]/a[ @href="/about/policies/" ]/text()', xmlparse(DOCUMENT
pg_read_file('test.html'))))
```

Продемонстрировать полнотекстовый поиск используя ts vector и ts query.

Поиск через ts vector и ts query. Использование операторов «и», «или»,
предшествование и соответствие

```
SELECT text, ts_rank(to_tsvector(text), to_tsquery('friend & best')) as rank from
texts where to_tsvector(text) @@ to_tsquery('friend & best') order by rank desc;
SELECT text, ts_rank(to_tsvector(text), to_tsquery('I <-> student')) as rank from
texts where to_tsvector(text) @@ to_tsquery('I <-> student') order by rank desc;
SELECT text, ts_rank(to_tsvector(text), to_tsquery('book | artists')) as rank from
texts where to_tsvector(text) @@ to_tsquery('book | artists') order by rank desc;
```

Запрос с условием в XML и экспорт содержимого всех таблиц в xml
произвольной структуры: с атрибутами, с дочерними элементами, с
атрибутами дочерних элементов.

```
select table_to_xml('account',true,true,"");
select xmlelement(name root, xmlelement(name account, xmlattributes(height,
weight, age))) from account;
select query_to_xml('select * from account where age = 20',true,true,"");
query_to_xml('select * from account where age = 20',true,true,"")
```

Продемонстрировать полнотекстовый поиск используя свои словари и файлы со стоп-словами

```
CREATE TEXT SEARCH DICTIONARY texts ( template =  
pg_catalog.simple, stopwords = stop )
```

Вывод

В результате выполнения работы были изучены языки запросов XPath и XQuery к XML-документам, синтаксис xml-документов, использование полнотекстового поиска и возможность создавать собственные словари для него.

Список используемой литературы

1. Виноградов В.И., Виноградова М.В. Постреляционные модели данных и языки запросов: Учебное пособие. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2017. — 100с. - ISBN 978-5-7038-4283-6.
2. PostgreSQL 14.2 Documentation. — Текст. Изображение: электронные // PostgreSQL : [сайт]. — URL: <https://www.postgresql.org/docs/14/index.html> (дата обращения: 12.05.2024)
3. pgAdmin 4 6.5 documentation. — Текст. Изображение: электронные // pgAdmin - PostgreSQL Tools : [сайт]. — URL: <https://www.pgadmin.org/docs/pgadmin4/6.5/index.html> (дата обращения: 12.05.2024)
4. Extensible Markup Language (XML) 1.0 (Fifth Edition). — Текст. Изображение : электронные // World Wide Web Consortium (W3C) : [сайт]. — URL: <https://www.w3.org/TR/xml/> (дата обращения: 12.05.2024)