МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

Лабораторная работа №5
по курсу Постреляционные базы данных

Тема: «Работа с XML в постреляционных СУБД»

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-22М _____
Сметанкин К.И.
"__"_____2020 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В. _____
к.т.н., доцент
"__"_____2020 г.

Москва – 2020

Пункты задания для выполнения:

Задание 1. Преобразование XML и реляционных данных

1.1. Определить в схему БД (**базовая**)

Создать в среде **MS SQL Server** БД по своей теме**.** В БД создать таблицы, например:

"Person"(Персона), содержит свойства:

- **fio** - ФИО – строковое (КЛЮЧ),
- **age** – возраст – целое,
- **city** – город – строковое.

"Avto"(Автомобиль), содержит свойства:

- **type** - тип – строковое,
- **owner —** владелец — строковое (ВНЕШНИЙ КЛЮЧ)
- **number -** номер – строковое,
- **id** - идентификатор (PK) – целое, автоинкремент.

Открыть таблицы на редактирование и заполнить тестовыми данными.

1.2. Преобразовать реляционные данные в формат XML (**базовая**)

В среде построения запросов SQL Server Management Studio продемонстрировать просмотр содержимого таблиц, например, Персона и Авто в формате xml в следующих вариантах:

- автоматический формат (Персона),
- все поля — элементы (Персона),
- все поля - атрибуты (Персона),
- добавление корневого элемента (persons) (Персона),
- переименование строк (raw = Person) (Персона),
- получение xml-схемы по умолчанию (Персона),
- отображение значений NULL (Персона),
- получение произвольной структуры документа (Авто+Персона), например:

```
<persons>
    <person age=”20”>
            <fio> FIO </fio>
            <city title=”Mos” />
            <autos>
                    <auto num =”1234”> type </auto>
                    <auto num =”weq34”> type </auto>
                    ….
            </autos>
    </person>
    <person age=”40”>
            ……..
    </person>
    ……
```

<persons>

1.3. Преобразовать XML-документ в реляционную таблицу (**хорошо**):

В среде построения запросов создать сценарии для создания переменной типа xml и заполнения ее тестовыми данными (взять xml документ сложной структуры, полученный ранее). Выполнить:

- Просмотреть данные из xml переменной в виде следующих наборов:
    - ФИО, возраст, город персон
    - Типы и номера авто
    - Города без дубликатов.

Задание 2. Построение запросов к XML данным с помощью языка Xquery (**базовая**)

В среде построения запросов создать сценарии для создания переменной типа xml (со сложной структурой) и заполнения ее тестовыми данными (можно взять xml документы, полученные ранее). На языке XPath-XQuery выполнить запросы:

- Проверки существования данных (атрибутов, элементов и их значений) — exist(), например, Автомобиля у Иванова.
- Извлечения данных (атрибутов, элементов и содержимого) — value(), например, ФИО владельца авто 123.
- Получения фрагмента XML — query(), например, список автомобилей Иванова.
- **(хорошо)** Изменения содержимого XML документа - modify(): добавление, изменение и удаление элементов и атрибутов, например, Авто для Петрова.
- **(хорошо)** Построение таблицы на основе XML документа сложной структуры с переименованием полей – nodes(), например, номера и типы машин.

Реализация

Схема БД

```
CREATE TABLE users
(
    id       integer PRIMARY KEY,
    `login`    varchar(100) NOT NULL UNIQUE,
    `password` TEXT,
    avatar   varchar(100) NOT NULL UNIQUE,
    karma    int DEFAULT 0
);

INSERT INTO users (id, `login`, `password`, avatar, karma)
VALUES (1, 'user_1', 'password_1', '/static/default.jpg', 0);

CREATE TABLE posts
(
    id          integer PRIMARY KEY,
    header      varchar(100) NOT NULL UNIQUE,
    short_topic varchar(255) NOT NULL UNIQUE,
    main_topic  text         NOT NULL,
    user_id     integer      NOT NULL REFERENCES users (id)
);

INSERT INTO posts (id, header, short_topic, main_topic, user_id)
VALUES (1, 'header', 'short topic', 'main topic', 1),
       (2, 'header_2', 'short topic_2', 'main topic_2', 1),
       (3, 'header_3', 'short topic_3', 'main topic_3', 1);

CREATE TABLE comments
(
    id        integer PRIMARY KEY,
    parent_id integer REFERENCES comments (id),
    user_id   integer NOT NULL REFERENCES users (id),
    post_id   integer NOT NULL REFERENCES posts (id),
    payload   text    NOT NULL
);

INSERT INTO comments (id, parent_id, user_id, post_id, payload)
VALUES (1, NULL, 1, 3, N'отличная статья, больше не пиши');

INSERT INTO comments (id, parent_id, user_id, post_id, payload)
VALUES (2, 1, 1, 3, N’спасибо');
```

Авто

Реализация

```
select *
from pbd.dbo.comments
for xml auto;
```

Результат

```
<pbd.dbo.comments id="1" user_id="1" post_id="3" payload="nice post"/>
<pbd.dbo.comments id="2" parent_id="1" user_id="1" post_id="3" payload="thanks"/
>
```

Поля элементы

Реализация

```
select * from pbd.dbo.comments
for xml raw('comment'), elements;
```

Результат

```
<comment>
    <id>1</id>
    <user_id>1</user_id>
    <post_id>3</post_id>
    <payload>nice post</payload>
</comment>
<comment>
    <id>2</id>
    <parent_id>1</parent_id>
    <user_id>1</user_id>
    <post_id>3</post_id>
    <payload>thanks</payload>
</comment>
```

Поля атрибуты

Реализация

```
select * from pbd.dbo.comments
for xml raw(‘comment');
```

Результат

```
<comment id="1" user_id="1" post_id="3" payload="nice post"/>
<comment id="2" parent_id="1" user_id="1" post_id="3" payload="thanks"/>
```

Корневой элемент

Реализация

```
select * from pbd.dbo.comments
for xml raw('comment'), elements, root('DB');
```

Результат

```
<DB>
    <comment>
        <id>1</id>
        <user_id>1</user_id>
        <post_id>3</post_id>
        <payload>nice post</payload>
    </comment>
    <comment>
        <id>2</id>
        <parent_id>1</parent_id>
        <user_id>1</user_id>
        <post_id>3</post_id>
        <payload>thanks</payload>
    </comment>
</DB>
```

Схема по умолчанию

Реализация

```
select * from pbd.dbo.comments
for xml raw('comment'), elements, root('DB'), xmlschema;
```

Результат

```
<DB>
    <xsd:schema targetNamespace="urn:schemas-microsoft-com:sql:SqlRowSet5"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:sqltypes="http://schemas.microsoft.com/sqlserver/2004/sqltypes"
elementFormDefault="qualified">
        <xsd:import namespace="http://schemas.microsoft.com/sqlserver/2004/
sqltypes" schemaLocation="http://schemas.microsoft.com/sqlserver/2004/sqltypes/
sqltypes.xsd"/>
        <xsd:element name="comment">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="id" type="sqltypes:int"/>
                    <xsd:element name="parent_id" type="sqltypes:int"
minOccurs="0"/>
                    <xsd:element name="user_id" type="sqltypes:int"/>
                    <xsd:element name="post_id" type="sqltypes:int"/>
                    <xsd:element name="payload">
                        <xsd:simpleType>
                            <xsd:restriction base="sqltypes:text"
sqltypes:localeId="1033" sqltypes:sqlCompareOptions="IgnoreCase IgnoreKanaType
IgnoreWidth" sqltypes:sqlSortId="52"/>
                        </xsd:simpleType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:schema>
    <comment xmlns="urn:schemas-microsoft-com:sql:SqlRowSet5">
        <id>1</id>
        <user_id>1</user_id>
        <post_id>3</post_id>
        <payload>nice post</payload>
    </comment>
    <comment xmlns="urn:schemas-microsoft-com:sql:SqlRowSet5">
        <id>2</id>
        <parent_id>1</parent_id>
        <user_id>1</user_id>
        <post_id>3</post_id>
        <payload>thanks</payload>
    </comment>
</DB>
```

Null-значения

Реализация

```
select *
from pbd.dbo.comments
for xml raw('comment'), elements xsinil, root(‘DB');
```

Результат

```
<DB xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <comment>
        <id>1</id>
        <parent_id xsi:nil="true"/>
        <user_id>1</user_id>
        <post_id>3</post_id>
        <payload>nice post</payload>
    </comment>
    <comment>
        <id>2</id>
        <parent_id>1</parent_id>
        <user_id>1</user_id>
        <post_id>3</post_id>
        <payload>thanks</payload>
    </comment>
</DB>
```

Произвольная схема

Реализация

```
select p.id     as 'post/@id',
       p.header as 'post/header',
       u.id     as 'post/author/@id',
       u.login  as 'post/author/login'
from pbd.dbo.posts p
         join users u on p.user_id = u.id
for xml path('post_author'), root('DB')
```

Результат

```
<DB>
    <post_author>
        <post id="1">
            <header>header</header>
            <author id="1">
                <login>user_1</login>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="2">
            <header>header_2</header>
            <author id="1">
                <login>user_1</login>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="3">
            <header>header_3</header>
            <author id="1">
                <login>user_1</login>
            </author>
        </post>
    </post_author>
</DB>
```

```
declare @x xml
set @x = '
<DB>
    <post_author>
        <post id="1">
            <header>header</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="2">
            <header>header_2</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="3">
            <header>header_3</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
</DB>'
declare @doc int
exec sp_xml_preparedocument @doc output, @x
```

Авторы статей

Реализация

```
select distinct *
from openxml(@doc, '//author')
with (
    id varchar(100) '@id',
    login varchar(100) 'login/text()'
);
```

Результат

Id	Login
1	user_2

Проверка существования автора с id=2 и post id=3

Реализация

```
select @x.exist('//post[@id="3"]/author[@id="2"]') as ‘exist';
```

Результат

exist
False

Получение пользователя

Реализация

```
select @x.value('(//author[@id="1"])[1]', 'NVARCHAR(100)') as ‘value’;
```

Результат

Value
user_1/static/default.jpg

Получение фрагмента xml

Реализация

```
select @x.query('(//author[@id="1"])[1]/login/text()') as login;
```

Результат

Login
user_1

Добавление

Реализация

```
set @x.modify('insert <post_author><post id="4"><header>h1</header></post></
post_author> into (/DB)[1]');
select @x as ‘new';
```

Результат

```
<DB>
    <post_author>
        <post id="1">
            <header>header</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="2">
            <header>header_2</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="3">
            <header>header_3</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="4">
            <header>h1</header>
        </post>
    </post_author>
</DB>
```

Изменение

Реализация

```
set @x.modify('replace value of (//author[@id="1"]/login/text())[1]
               with "new user 1"');
select @x as 'after';
```

Результат

```
<DB>
    <post_author>
        <post id="1">
            <header>header</header>
            <author id="1">
                <login>new user 1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="2">
            <header>header_2</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="3">
            <header>header_3</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
</DB>
```

Удаление

Реализация

```
set @x.modify('delete //post[@id="1"]');
select @x as 'delete';
```

Результат

```
<DB>
    <post_author>
        <post id="2">
            <header>header_2</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="3">
            <header>header_3</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
</DB>
```

Таблица из xml

Реализация

```
select
    col.value('(@id)[1]','nvarchar(100)') as post_id,
    col.value('(header/text())[1]','nvarchar(100)') as post_header,
    col.value('(author/@id)[1]','nvarchar(100)') as author_id,
    col.value('(author/login)[1]','nvarchar(100)') as author_login,
    col.value('(author/avatar)[1]','nvarchar(100)') as author_avatar
from @x.nodes('//post') tab(col);
```

Результат

post_id	post_header	author_id	author_avatar
1	Header	user_1	/static/default.jpg
2	header_2	user_1	/static/default.jpg
3	header_3	user_1	/static/default.jpg

Дополнительное

```
declare @x xml
set @x = '
<DB>
    <post_author>
        <post id="1">
            <header>header</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="2">
            <header>header_2</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="3">
            <header>header_3</header>
            <author id="1">
                <login>user_1</login>
                <avatar>/static/default.jpg</avatar>
            </author>
        </post>
    </post_author>
    <post_author>
        <post id="4">
            <header>header_4</header>
            <author id="2">
                <login>user_2</login>
                <avatar>/static/new_avatar.jpg</avatar>
            </author>
        </post>
    </post_author>
</DB>'
declare @doc int
exec sp_xml_preparedocument @doc output, @x
```

Количество статей пользователя user_1

Реализация

```
select @x.value('count(for $p in //post where $p/author/login/text() = "user_1"
return $p)', 'int') as posts_count;
```

Результат

post_count
3

Количество статей по пользователям

Реализация

```
select @x.query('let $authors := distinct-values(for $u in //author return $u/
login) for $a in $authors return <posts_count>{attribute login {data($a)} }
{ count(for $p in //post where $p/author/login/text() = $a return $p) } </
posts_count>');
```

Результат

```
<posts_count login="user_1">3</posts_count>
<posts_count login="user_2">1</posts_count>
```

Измененный формат вывода

Реализация

```
select @x.query('for $post in //post return <post_author><post> {attribute login
{data($post/author/login)}} <header>{data($post/header)}</header>
<id>{data($post/@id)}</id> </post></post_author>');
```

Результат

```
<post_author>
    <post login="user_1">
        <header>header</header>
        <id>1</id>
    </post>
</post_author>
<post_author>
    <post login="user_1">
        <header>header_2</header>
        <id>2</id>
    </post>
</post_author>
<post_author>
    <post login="user_1">
        <header>header_3</header>
        <id>3</id>
    </post>
</post_author>
<post_author>
    <post login="user_2">
        <header>header_4</header>
        <id>4</id>
    </post>
</post_author>
```