

**Московский государственный технический университет им.Н.Э.Баумана
кафедра "Системы обработки информации и управления"**



Постреляционные базы данных

к лабораторной работе №3

**Лабораторная работа «Модель полуструктурированных данных и языки запросов XPath, XQuery»
по дисциплине «Постреляционные базы данных»**

Инструктор : Мария Валерьевна

Email:2623859464@qq.com

Студент: Ван Чаочао

группа ИУ5И-22М

2022/04/01

Цель работы:

1. Изучить различные технологии работы с XML.
2. Освоить методы работы с XML в постреляционных СУБД.
3. Получить навыки работы с инструментальными средствами СУБД.

Средства выполнения:

СУБД MS SQL Server

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:

1 : Создать таблицу (创建表格)

Users Table :

DESKTOP-F4756N1.lab3 - dbo.Users			
列名	数据类型	允许 Null 值	
UserID	int	<input type="checkbox"/>	
Username	nchar(50)	<input type="checkbox"/>	
Password	nchar(20)	<input type="checkbox"/>	
RealName	nchar(50)	<input type="checkbox"/>	
Gender	nchar(10)	<input type="checkbox"/>	
DateOfBirth	date	<input type="checkbox"/>	
Email	nchar(20)	<input type="checkbox"/>	
Phone	nchar(11)	<input type="checkbox"/>	
City	nchar(10)	<input type="checkbox"/>	
Address	nvarchar(50)	<input type="checkbox"/>	
CarsID	int	<input checked="" type="checkbox"/>	

CarTable :

DESKTOP-F4756N1.lab3 - dbo.Car			
列名	数据类型	允许 Null 值	
CarsID	int	<input type="checkbox"/>	
CarName	nchar(50)	<input type="checkbox"/>	
CarDescription	nchar(50)	<input type="checkbox"/>	
Price	money	<input type="checkbox"/>	
Stock	int	<input type="checkbox"/>	
Sold	int	<input type="checkbox"/>	

2 : Заполнение таблицы данными (填充表格数据)

Users table data:

	UserID	Username	Password	RealName	Gender	DateOfBirth	Email	Phone	City	Address	CarsID
1	1	chaochao	123456789	wangchaochao	man	1980-08-27	2623859464@qq.com	18801035010	moscow	banstu	2
2	2	wei	265314789	zhang wei	man	1988-09-29	2638254969@qq.com	79269565693	moscow	banstu	1
3	3	cat	987654321	li li	woman	1999-08-12	3639859427@mail.com	79269815682	moscow	banstu	3
4	4	flower	321654987	hua hua	woman	1995-08-13	4569825236@qq.com	79279853569	moscow	С о р о ч а н ы	4

Car table data:

	CarsID	CarName	CarDescription	Price	Stock	Sold
1	1	BMW	red	2000000.00	100	20
2	2	Geely	black, five seats	1800000.00	80	10
3	3	Bentley	silver	10000000.00	50	8
4	4	JEEP	Wrangler	4290000.00	60	9

1.2. Преобразовать реляционные данные в формат XML (базовая)

1.Автоматическое преобразование

```
select * from Car
```

```
for xml auto
```

XML_F52E2B61-1...805F49916B3.xml	SQLQuery6.sql -...4756N1\asus (54))*	SQLQuery4.sql -...4756N1\asus (53))*
<Car CarsID="1" CarName="BMW	" CarDescription="red	" Price="2000000.0000" Stock="100" Sold="20" />
<Car CarsID="2" CarName="Geely	" CarDescription="black, five seats	" Price="1800000.0000" Stock="80" Sold="10" />
<Car CarsID="3" CarName="Bentley	" CarDescription="silver	" Price="10000000.0000" Stock="50" Sold="8" />
<Car CarsID="4" CarName="JEEP	" CarDescription="Wrangler	" Price="4290000.0000" Stock="60" Sold="9" />

2.Все поля выводятся как элементы(字段显示元素)

```
select * from Car
```

```
for xml raw('cars'), elements
```

XML_F52E2B61-1...805F49916B4.xml	XML_F52E2B61-1...805F49916B5.xml
<cars> <CarsID>1</CarsID> <CarName>BMW <CarDescription>red <Price>2000000.0000</Price> <Stock>100</Stock> <Sold>20</Sold> </cars> <cars> <CarsID>2</CarsID> <CarName>Geely <CarDescription>black, five seats <Price>1800000.0000</Price> <Stock>80</Stock> <Sold>10</Sold> </cars> <cars> <CarsID>3</CarsID> <CarName>Bentley <CarDescription>silver <Price>10000000.0000</Price> <Stock>50</Stock> <Sold>8</Sold> </cars> <cars> <CarsID>4</CarsID> <CarName>JEEP <CarDescription>Wrangler <Price>4290000.0000</Price> <Stock>60</Stock> <Sold>9</Sold> </cars>	

3. Все поля выводятся как атрибуты (显示属性)

```
select * from Car  
for xml raw('cars')
```

XML_F52E2B61-1...805F49916B5.xml	XML_F52E2B61-1...805F49916B4.xml	XML_F52E2B61-1...805F49916B3.xml
<cars CarsID="1" CarName="BMW	" CarDescription="red	" Price="2000000.0000" Stock="100" Sold="20" />
<cars CarsID="2" CarName="Geely	" CarDescription="black, five seats	" Price="1800000.0000" Stock="80" Sold="10" />
<cars CarsID="3" CarName="Bentley	" CarDescription="silver	" Price="10000000.0000" Stock="50" Sold="8" />
<cars CarsID="4" CarName="JEEP	" CarDescription="Wrangler	" Price="4290000.0000" Stock="60" Sold="9" />

4. Добавление корневого элемента (添加根元素)

```
select * from Car  
for xml auto, root('myroot')
```

XML_F52E2B61-1...805F49916B6.xml	XML_F52E2B61-1...805F49916B5.xml	XML_F52E2B61-1...805F49916B3.xml	SQLQu
<myroot> <Car CarsID="1" CarName="BMW	" CarDescription="red	" Price="2000000.0000" Stock="100" Sold="20" />	
<Car CarsID="2" CarName="Geely	" CarDescription="black, five seats	" Price="1800000.0000" Stock="80" Sold="10" />	
<Car CarsID="3" CarName="Bentley	" CarDescription="silver	" Price="10000000.0000" Stock="50" Sold="8" />	
<Car CarsID="4" CarName="JEEP	" CarDescription="Wrangler	" Price="4290000.0000" Stock="60" Sold="9" />	
</myroot>			

5. Добавление схемы документа (添加文档大纲)

```
select UserID, RealName, Phone from Users  
for xml raw('User'), root('information'), xmlschema
```

<information> <xsd:schema targetNamespace="urn:schemas-microsoft-com:sql:SqlRowSet1" 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="User"> <xsd:complexType> <xsd:attribute name="UserID" type="sqltypes:int" use="required" /> <xsd:attribute name="RealName" use="required" /> <xsd:simpleType> <xsd:restriction base="sqltypes:nchar" sqltypes:localId="2052" sqltypes:sqlCompareOptions="IgnoreCase IgnoreKanaType IgnoreWidth"> <xsd:maxLength value="20" /> </xsd:restriction> </xsd:simpleType> </xsd:attribute> <xsd:attribute name="Phone" use="required" /> <xsd:simpleType> <xsd:restriction base="sqltypes:nchar" sqltypes:localId="2052" sqltypes:sqlCompareOptions="IgnoreCase IgnoreKanaType IgnoreWidth"> <xsd:maxLength value="11" /> </xsd:restriction> </xsd:simpleType> </xsd:attribute> </xsd:complexType> </xsd:element> </xsd:schema> <User xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1" UserID="1" RealName="wangchao" Phone="18801035010" /> <User xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1" UserID="2" RealName="zhang wei" Phone="79269363693" /> <User xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1" UserID="3" RealName="li li" Phone="79269315682" /> <User xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1" UserID="4" RealName="hua hua" Phone="79279533569" /> </information>

6. Добавление значений NULL (添加 null 值)

```
select top(3) * from Car  
for xml raw('cars'), elements xsinil, root
```

XML_F52E2B61-1...05F49916B10.xml	SQLQuery6.sql ...4756N
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <cars> <CarsID>1</CarsID> <CarName>BMW <CarDescription xsi:nil="true" /> <Price>2000000.0000</Price> <Stock>100</Stock> <Sold>20</Sold> </cars> <cars> <CarsID>2</CarsID> <CarName>Geely <CarDescription>black, five seats <Price>1800000.0000</Price> <Stock>80</Stock> <Sold>10</Sold> </cars> <cars> <CarsID>3</CarsID> <CarName>Bentley <CarDescription>silver <Price>10000000.0000</Price> <Stock>50</Stock> <Sold>8</Sold> </cars> </root>	

7.Произвольный вывод (任意输出)

```
select UserID,
       Username,
       Phone,
       (select CarsID, CarName
        from Car
        where Owners=UserID
        for xml raw('cars'), elements, type)
from Users

for xml raw('user'), elements
```



XML_F52E2B61-1...05F49916B11.xml XML_F52E2B61-1...05F4

```
<user>
  <UserID>1</UserID>
  <Username>chaochao</Username>
  <Phone>18801035010</Phone>
  <cars>
    <CarsID>3</CarsID>
    <CarName>Bentley</CarName>
  </cars>
</user>
<user>
  <UserID>2</UserID>
  <Username>wei</Username>
  <Phone>79269565693</Phone>
  <cars>
    <CarsID>1</CarsID>
    <CarName>BMW</CarName>
  </cars>
</user>
<user>
  <UserID>3</UserID>
  <Username>cat</Username>
  <Phone>79269815682</Phone>
  <cars>
    <CarsID>2</CarsID>
    <CarName>Geely</CarName>
  </cars>
</user>
<user>
  <UserID>4</UserID>
  <Username>flower</Username>
  <Phone>79279853569</Phone>
  <cars>
    <CarsID>4</CarsID>
    <CarName>JEEP</CarName>
  </cars>
</user>
```


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

```
declare @x xml;
SET @x =( SELECT * FROM
            OPENROWSET( BULK 'D:\wcc.xml', SINGLE_BLOB) AS x
            );
declare @doc int
exec sp_xml_preparedocument @doc output, @x

select * from openxml(@doc, '/r/user')
with (uname nchar(20) 'Username',
      ucar nchar(20) 'cars/CarName');
```

结果	消息
	uname ucar
1	chaochao Bentley
2	wei BMW
3	cat Geely
4	flower JEEP

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.

```
declare @x xml;
SET @x =( SELECT * FROM
            OPENROWSET( BULK 'D:\wcc.xml', SINGLE_BLOB) AS x
            );
select @x.exist('/r/user');
select @x.exist('/r/user[UserID>2]');
select @x.exist('/r/user[Username="chaochao"]');
```

110 %	结果	消息
	(无列名)	
1	1	
	(无列名)	
1	1	
	(无列名)	
1	0	

- Извлечения данных (атрибутов, элементов и содержимого) — value(), например, ФИО владельца авто 123.

```
select @x.value('(/r/user/cars/CarName)[2]', 'nvarchar(10)');
```

	(无列名)
1	BMW

- Получения фрагмента XML — query(), например, список автомобилей Иванова.

```
select @x.query('/r/user[UserID>3]');
```

```
select @x.query('/r/user/cars[CarName="BMW"]');
```

result6.xml	xmlresult4.xml	wcc.xml	SQLC
<pre> <user> <UserID>4</UserID> <Username>flower</Username> <Phone>79279853569</Phone> <cars> <CarsID>4</CarsID> <CarName>JEEP</CarName> </cars> </user> </pre>			

xmlresult7.xml	wcc.xml	SQLQuery6.sql -...4
<pre> <cars> <CarsID>1</CarsID> <CarName>BMW</CarName> </cars> </pre>		

- (хорошо) Изменения содержимого XML документа - modify(): добавление, изменение и удаление элементов и атрибутов, например, Авто для Петрова.

```
declare @x xml;
```

```
SET @x = ( SELECT * FROM
            OPENROWSET( BULK 'D:\wcc.xml', SINGLE_BLOB) AS x
          );
```

```
set @x.modify('insert <user anotherPhone="79269852359"/> into (//user)[1]');
```

```
set @x.modify('insert <firstname>fname="wang"</firstname> as first into (//user)[1]');
```

```
set @x.modify('insert <colour>black</colour> as last into (//cars)[1]');
```

```
select @x;
```

<r>
<pre> <user> <firstname>fname="wang"</firstname> <UserID>1</UserID> <Username>chaochao</Username> <Phone>18801035010</Phone> <cars> <CarsID>3</CarsID> <CarName>Bentley</CarName> <colour>black</colour> </cars> <user anotherPhone="79269852359" /> </user> </pre>

```

SET @x.modify('
  replace value of (/r/user/Username[1]/text())[1]
  with "wang chaochao"
')
SELECT @x

```

```

<r>
  <user>
    <UserID>1</UserID>
    <Username>wang chaochao</Username>
    <Phone>18801035010</Phone>
    <cars>
      <CarsID>3</CarsID>
      <CarName>Bentley</CarName>
    </cars>
  </user>
</r>

```

delete an attribute

```

set @x.modify(' delete //Phone' );
select @x;

```

```

<r>
  <user>
    <UserID>1</UserID>
    <Username>chaochao</Username>
    <cars>
      <CarsID>3</CarsID>
      <CarName>Bentley</CarName>
    </cars>
  </user>
  <user>
    <UserID>2</UserID>
    <Username>wei</Username>
    <cars>
      <CarsID>1</CarsID>
      <CarName>BMW</CarName>
    </cars>
  </user>
  <user>
    <UserID>3</UserID>
    <Username>cat</Username>
    <cars>
      <CarsID>2</CarsID>
      <CarName>Geely</CarName>
    </cars>
  </user>
  <user>
    <UserID>4</UserID>
    <Username>flower</Username>
    <cars>
      <CarsID>4</CarsID>
      <CarName>JEEP</CarName>
    </cars>
  </user>
</r>

```

```

set @x.modify(' delete //user/cars[1]' ); #удаляет множество узлов
select @x;

```

```

<r>
  <user>
    <UserID>1</UserID>
    <Username>chaochao</Username>
    <Phone>18801035010</Phone>
  </user>
  <user>
    <UserID>2</UserID>
    <Username>wei</Username>
    <Phone>79269565693</Phone>
  </user>
  <user>
    <UserID>3</UserID>
    <Username>cat</Username>
    <Phone>79269815682</Phone>
  </user>
  <user>
    <UserID>4</UserID>
    <Username>flower</Username>
    <Phone>79279853569</Phone>
  </user>
</r>

```


- (хорошо) Построение таблицы на основе XML документа сложной структуры с переименованием полей – nodes(), например, номера и типы машин.

```
select car.value('(/cars/CarName)[1]', 'nchar(20)') as CName,
car.value('Username[1]', 'nchar(20)') as UNmae,
car.value('Phone[1]', 'nchar(20)') as Pnum
from @x.nodes('/r/user') tab(car);
```

110 %

结果 消息

	CName	UNmae	Pnum
1	Bentley	chaochao	18801035010
2	Bentley	wei	79269565693
3	Bentley	cat	79269815682
4	Bentley	flower	79279853569

Задание 3. Построить запросы на языке Xquery с применением FLWOR (отлично)

Использовать конструкции **FLWOR** на языке XQuery и выполнить запросы с применением Xpath (создание новых элементов с атрибутами и вложенными элементами),

```
declare @x xml;
SET @x =( SELECT * FROM
OPENROWSET( BULK 'D:\wcc.xml', SINGLE_BLOB) AS x
);
```

```
SELECT @x.query('for $e in /r/user/cars
order by $e descending
return $e
')
```

```

<cars>
  <CarsID>4</CarsID>
  <CarName>JEEP</CarName>
</cars>
<cars>
  <CarsID>3</CarsID>
  <CarName>Bentley</CarName>
</cars>
<cars>
  <CarsID>2</CarsID>
  <CarName>Geely</CarName>
</cars>
<cars>
  <CarsID>1</CarsID>
  <CarName>BMW</CarName>
</cars>

```

```

declare @x xml;
SET @x =( SELECT * FROM
            OPENROWSET( BULK 'D:\wcc.xml', SINGLE_BLOB) AS x
        );

SELECT @x.query('for $e in /r/user /cars
                order by $e ascending
                return $e/CarName
            ');

```

nlresult53.xml	xmlresult52.xml	xmlresult5
<CarName>BMW	</CarName>	
<CarName>Geely	</CarName>	
<CarName>Bentley	</CarName>	
<CarName>JEEP	</CarName>	

```

select @x.query('for $a in //user/Phone
                where (//user[UserID/text()='3']/Phone/@city = $a/@city)
                return string($a/@city)');

```

结果	消息
(无列名)	
1	london

```

select @x.query('for $a in //user
                where ($a/Phone//@city="moc")
                return $a/UserID/text()');

```

(无列名)	
1	london
(无列名)	
1	4

Литература:

1. Гапанюк Ю.Е., Ревунков Г.И. Введение в XML-технологии. Учебное пособие. – М.: МГТУ им. Н.Э. Баумана, 2010 г.
2. Каленик А. Использование новых возможностей Microsoft SQL Server 2005. - М.:Русская редакция; Спб.:Питер. - 2006 г.

3. Пирогов В. SQL Server 2005. Программирование клиент-серверных приложений. - Спб.: БХВ-Петербург. - 2006 г.
4. Уолтерс У. и др. SQL Server 2008: ускоренный курс для профессионалов: пер. с англ. – М.: ООО «И.Д. Вильямс», 2009. – 768с.