БАЗЫ ДАННЫХ

Лекция 16 XML

XML

- XML – Extensible Markup Language
- Является подмножеством языка SGML – Standard Generalized Markup Language – метаязыка для определения языков разметки

W3C – стандартизация

- Консорциум Всемирной паутины – World Wide Web Consortium – организация, разрабатывающая и внедряющая технологические стандарты для web
- Глава – Тимоти Джон Бернерс-Ли
- Ок. 15 стандартов утверждены для XML:
  - XML Schema
  - XPath
  - XSLT
  - XQuery

XML-документ

- Текстовый файл
- Древовидная структура

- Теги
- Атрибуты

- Данные

XML

- Элементы
- Символьные данные
- Древовидная структура документа
- Корневой элемент, дочерние элементы и листья
- Атрибуты

```
<?xml version="1.0" encoding="UTF-8"?>
<PersonList Type="Employee">
    <Title> Value="Employee List"></Title>
    <Contents>
        <Employee>
            <Name>Ann Heathers</Name>
            <No>12202</No>
            <Deptno>d3</Deptno>
            <Address>
                <City>Dallas</City>
                <Street>Main St</Street>
            </Address>
        </Employee>
        <Employee>
            <Name>John Dow</Name>
            <No>12216</No>
            <Deptno>d1</Deptno>
            <Address>
                <City>Seattle</City>
                <Street>Abbey Rd</Street>
            </Address>
        </Employee>
    </Contents>
</PersonList>
```

XML-документ

- Начинается специальным тегом с именем **?xml** (объявление XML
- Каждый элемент начинается открывающим тегом и завершается закрывающим
- Могут быть вложенные теги

XML

- Правильно построенный документ – well-formed – соответствует синтаксическим правилам XML
- Валидный документ – valid – соответствует правилам описания типа документа (Document Type Definition, DTD)

XML

- наличие корневого элемента
- каждый открывающий тег имеет соответствующий закрывающий тег
- правильное вложение элементов документа
- атрибут должен иметь значение, которое берется в кавычки

```
<?xml version="1.0" encoding="UTF-8"?>
<PersonList Type="Employee">
    <Title> Value="Employee List"></Title>
    <Contents>
        <Employee>
            <Name>Ann Heathers</Name>
            <No>12202</No>
            <Deptno>d3</Deptno>
            <Address>
                <City>Dallas</City>
                <Street>Main St</Street>
            </Address>
        </Employee>
        <Employee>
            <Name>John Dow</Name>
            <No>12216</No>
            <Deptno>d1</Deptno>
            <Address>
                <City>Seattle</City>
                <Street>Abbey Rd</Street>
            </Address>
        </Employee>
    </Contents>
</PersonList>
```

Языки схем

- язык DTD - Document Type Definition
  - Набор правил для структурирования XML
  - Внутренний DTD – часть XML-документа
  - Внешний DTD – адрес URL
- язык XML Schema

DTD – Document Type Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PersonList SYSTEM "D:\DTDS\EMP4.dtd">
<!ELEMENT EmployeeList (Title, Contents)>
<!ELEMENT Title EMPTY>
<!ELEMENT Contents (Employee*)>
<!ELEMENT Employee (Name, No, Deptno, Address)>
<!ELEMENT Name (Fname, Lname)>
<!ELEMENT Fname (#PCDATA)>
<!ELEMENT Lname (#PCDATA)>
<!ELEMENT No (#PCDATA)>
<!ELEMENT Deptno (#PCDATA)>
<!ELEMENT Address (City, Street) >
<!ELEMENT City (#PCDATA)>
<!ELEMENT Street (#PCDATA)>
<!ATTLIST EmployeeList Type CDATA #IMPLIED
Date CDATA #IMPLIED>
<!ATTLIST Title Value CDATA #REQUIRED>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<PersonList Type="Employee">
    <Title> Value="Employee List"></Title>
    <Contents>
        <Employee>
            <Name>Ann Heathers</Name>
            <No>12202</No>
            <Deptno>d3</Deptno>
            <Address>
                <City>Dallas</City>
                <Street>Main St</Street>
            </Address>
        </Employee>
        <Employee>
            <Name>John Dow</Name>
            <No>12216</No>
            <Deptno>d1</Deptno>
            <Address>
                <City>Seattle</City>
                <Street>Abbey Rd</Street>
            </Address>
        </Employee>
    </Contents>
</PersonList>
```

DTD – Document Type Definition

- Имя DTD должно соответствовать имени тега корневого элемента XML-документа
- XML-документ нужно связать с соответствующим файлом DTD
- Объявления типов элементов должны начинаться с инструкции ELEMENT
- Порядок элементов XML-документа
- Элементы без подчиненных - #PCDATA
- * наличие элементов (от нуля и больше)
- ? наличие не более одного элемента
- + наличие по крайней мере одного элемента
- Объявление атрибута <!ATTLIST имя атрибута и тип данных>
- #IMPLIED атрибут необязательный, #REQUIRED обязательный

DTD – Document Type Definition

- Атрибут типа ID - определение уникального значения
- Атрибут типа IDREF должен ссылаться на действительный идентификатор, объявленный в этом же документе
- Атрибут типа IDREFS задает список разделенных пробелами строк, на которые ссылаются значения атрибута типа ID

XML Schema

- XML Schema — язык описания структуры XML-документа – предназначен для определения правил, которым должен подчиняться документ
- Создается модель данных документа:
  - словарь (названия элементов и атрибутов)
  - модель содержания (отношения между элементами и атрибутами и их структура)
  - типы данных

Задачи

- преобразование XML в строки реляционных таблиц
- преобразование данных в таблицах в XML

Декомпозиция XML

- sp_xml_ prepare document
- sp_xml_ remove document

```
DECLARE @hdoc INT
DECLARE @doc VARCHAR(1000)
SET @doc ='<ROOT>
<Employee>
    <Name>Ann Heathers</Name>
    <No>12202</No>
    <Deptno>d3</Deptno>
    <Address>
        <City>Dallas</City>
        <Street>Main St</Street>
    </Address>
</Employee>
<Employee>
    <Name>John Dow</Name>
    <No>12216</No>
    <Deptno>d1</Deptno>
    <Address>
        <City>Seattle</City>
        <Street>Abbey Rd</Street>
    </Address>
</Employee>
</ROOT>'
EXEC sp_xml_preparedocument @hdoc OUTPUT, @doc
```

Декомпозиция XML

```
SELECT * FROM OPENXML (@hdoc, '/ROOT/Employee', 1)
WITH (name VARCHAR(20) 'Name',
no INT 'No',
deptno VARCHAR(6) 'Deptno',
address VARCHAR(50) 'Address');
```

Results | Messages

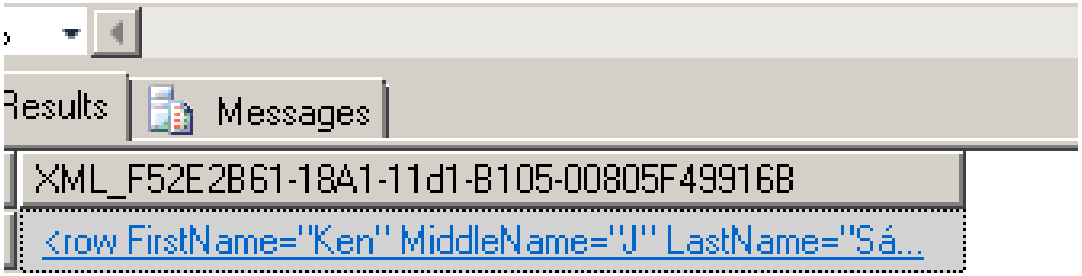
	name	no	deptno	address
	Ann Heathers	12202	d3	Dallas Main St
	John Dow	12216	d1	Seattle Abbey Rd

Представление данных в XML

- RAW – каждая строка РН в строку XML
- AUTO – каждая строка РН в XML-элемент с подчиненными
- PATH – сочетание атрибутной и элементной форм
- EXPLICIT – расширенная форма РН

RAW

```
SELECT
 [Title],
 [FirstName],
 [MiddleName],
 [LastName],
 [AdditionalContactInfo]
 FROM [Person].[Person]
 FOR XML RAW;
```



RAW

```
<row FirstName="Ken" MiddleName="J" LastName="Sánchez" />
<row FirstName="Terri" MiddleName="Lee" LastName="Duffy" />
<row FirstName="Roberto" LastName="Tamburello" />
<row FirstName="Rob" LastName="Walters" />
<row Title="Ms." FirstName="Gail" MiddleName="A" LastName="Erickson" />
<row Title="Mr." FirstName="Jossef" MiddleName="H" LastName="Goldberg" />
<row FirstName="Dylan" MiddleName="A" LastName="Miller" />
<row FirstName="Diane" MiddleName="L" LastName="Margheim" />
<row FirstName="Gigi" MiddleName="N" LastName="Matthew" />
<row FirstName="Michael" LastName="Raheem" />
<row FirstName="Ovidiu" MiddleName="V" LastName="Cracium" />
<row FirstName="Thierry" MiddleName="B" LastName="D'Hers" />
<row Title="Ms." FirstName="Janice" MiddleName="M" LastName="Galvin" />
<row FirstName="Michael" MiddleName="I" LastName="Sullivan" />
<row FirstName="Sharon" MiddleName="B" LastName="Salavaria" />
<row FirstName="David" MiddleName="M" LastName="Bradley" />
<row FirstName="Kevin" MiddleName="F" LastName="Brown" />
<row FirstName="John" MiddleName="L" LastName="Wood" />
<row FirstName="Mary" MiddleName="A" LastName="Dempsey" />
<row FirstName="Wanida" MiddleName="M" LastName="Benshoof" />
<row FirstName="Terry" MiddleName="J" LastName="Eminhizer" />
<row FirstName="Sariya" MiddleName="E" LastName="Harnpadoungsataya" />
<row FirstName="Mary" MiddleName="E" LastName="Gibson" />
<row Title="Ms." FirstName="Jill" MiddleName="A" LastName="Williams" />
<row FirstName="James" MiddleName="R" LastName="Hamilton" />
<row FirstName="Peter" MiddleName="J" LastName="Krebs" />
<row FirstName="Jo" MiddleName="A" LastName="Brown" />
<row FirstName="Guy" MiddleName="R" LastName="Gilbert" />
```

AUTO

```
SELECT
Title,
FirstName,
MiddleName,
LastName,
AdditionalContactInfo,
DepartmentID,
ShiftID,
EndDate
FROM Person.Person inner join HumanResources.EmployeeDepartmentHistory
on Person.Person.BusinessEntityID= HumanResources.EmployeeDepartmentHistory.BusinessEntityID
FOR XML AUTO;
```

AUTO

```
    <HumanResources.EmployeeDepartmentHistory DepartmentID="16" ShiftID="1" />
  </Person.Person>
- <Person.Person FirstName="Terri" MiddleName="Lee" LastName="Duffy">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="1" ShiftID="1" />
  </Person.Person>
- <Person.Person FirstName="Roberto" LastName="Tamburello">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="1" ShiftID="1" />
  </Person.Person>
- <Person.Person FirstName="Rob" LastName="Walters">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="1" ShiftID="1" EndDate="2004-06
  </Person.Person>
- <Person.Person FirstName="Rob" LastName="Walters">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="2" ShiftID="1" />
  </Person.Person>
- <Person.Person Title="Ms." FirstName="Gail" MiddleName="A" LastName="Erickson">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="1" ShiftID="1" />
  </Person.Person>
- <Person.Person Title="Mr." FirstName="Jossef" MiddleName="H" LastName="Goldberg">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="1" ShiftID="1" />
  </Person.Person>
- <Person.Person FirstName="Dylan" MiddleName="A" LastName="Miller">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="6" ShiftID="1" />
  </Person.Person>
- <Person.Person FirstName="Diane" MiddleName="L" LastName="Margheim">
    <HumanResources.EmployeeDepartmentHistory DepartmentID="6" ShiftID="1" />
```

PATH

```
SELECT DepartmentID "@Department",
Title "EmpName/Title",
FirstName "EmpName/FirstName",
MiddleName "EmpName/MiddleName",
LastName "EmpName/LastName"

FROM Person.Person inner join HumanResources.EmployeeDepartmentHistory
on Person.Person.BusinessEntityID= HumanResources.EmployeeDepartmentHistory.BusinessEntityID
FOR XML path;
```

Шаблон	Описание
eN	XML-элемент eN
@aN	XML-атрибут aN
eN1/eN2	XML-элемент eN1 и вложенный элемент eN2
eN/@aN	XML-элемент eN и атрибут aN

PATH

```
<row Department="1">
  <EmpName>
    <FirstName>Roberto</FirstName>
    <LastName>Tamburello</LastName>
  </EmpName>
</row>
<row Department="1">
  <EmpName>
    <FirstName>Rob</FirstName>
    <LastName>Walters</LastName>
  </EmpName>
</row>
<row Department="2">
  <EmpName>
    <FirstName>Rob</FirstName>
    <LastName>Walters</LastName>
  </EmpName>
</row>
<row Department="1">
  <EmpName>
    <Title>Ms.</Title>
    <FirstName>Gail</FirstName>
    <MiddleName>A</MiddleName>
    <LastName>Erickson</LastName>
  </EmpName>
</row>
```

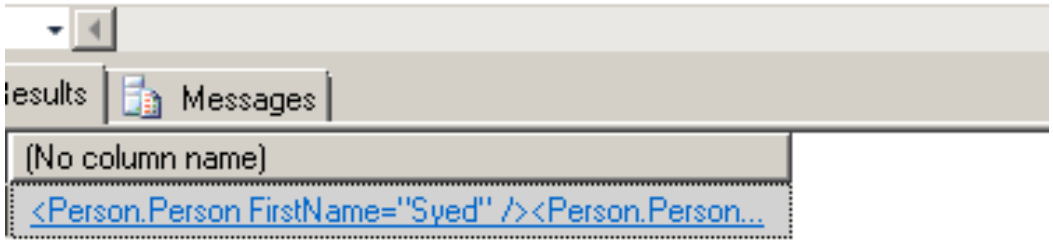
Директивы

- TYPE
- ROOT
- ELEMENTS

Директива TYPE

- сохранять результат реляционного запроса как XML-документ или фрагмент типа данных XML

```
DECLARE @x xml;
SET @x = (SELECT FirstName FROM Person.Person
FOR XML AUTO, TYPE);
SELECT @x;
```




```
<Person.Person FirstName="Kim" />
<Person.Person FirstName="Kim" />
<Person.Person FirstName="Kim" />
<Person.Person FirstName="Hazem" />
<Person.Person FirstName="Sam" />
<Person.Person FirstName="Humberto" />
<Person.Person FirstName="Gustavo" />
<Person.Person FirstName="Pilar" />
<Person.Person FirstName="Pilar" />
<Person.Person FirstName="Aaron" />
<Person.Person FirstName="Adam" />
<Person.Person FirstName="Alex" />
<Person.Person FirstName="Alexandra" />
<Person.Person FirstName="Allison" />
<Person.Person FirstName="Amanda" />
<Person.Person FirstName="Amber" />
<Person.Person FirstName="Andrea" />
<Person.Person FirstName="Angel" />
<Person.Person FirstName="Bailey" />
<Person.Person FirstName="Ben" />
<Person.Person FirstName="Blake" />
<Person.Person FirstName="Carla" />
<Person.Person FirstName="Carlos" />
<Person.Person FirstName="Charles" />
<Person.Person FirstName="Chloe" />
<Person.Person FirstName="Connor" />
<Person.Person FirstName="Courtney" />
<Person.Person FirstName="Dalton" />
<Person.Person FirstName="Devin" />
```

Директива ROOT

- Добавление к результирующему набору XML одного элемента верхнего уровня

```
DECLARE @x xml;
SET @x = (SELECT FirstName FROM Person.Person
FOR XML AUTO, ROOT ('AllPersons'));
SELECT @x;
```

```
<AllPersons>
  <Person.Person FirstName="Syed" />
  <Person.Person FirstName="Catherine" />
  <Person.Person FirstName="Kim" />
  <Person.Person FirstName="Kim" />
  <Person.Person FirstName="Kim" />
  <Person.Person FirstName="Hazem" />
  <Person.Person FirstName="Sam" />
  <Person.Person FirstName="Humberto" />
  <Person.Person FirstName="Gustavo" />
  <Person.Person FirstName="Pilar" />
  <Person.Person FirstName="Pilar" />
  <Person.Person FirstName="Aaron" />
```

Директива ELEMENTS

```
DECLARE @x xml;
SET @x = (SELECT FirstName FROM Person.Person
FOR XML AUTO, elements);
SELECT @x;
```

```
  <FirstName>Syed</FirstName>
</Person.Person>
<Person.Person>
  <FirstName>Catherine</FirstName>
</Person.Person>
<Person.Person>
  <FirstName>Kim</FirstName>
</Person.Person>
<Person.Person>
  <FirstName>Kim</FirstName>
</Person.Person>
<Person.Person>
  <FirstName>Kim</FirstName>
</Person.Person>
<Person.Person>
  <FirstName>Hazem</FirstName>
</Person.Person>
<Person.Person>
  <FirstName>Sam</FirstName>
</Person.Person>
<Person.Person>
  <FirstName>Humberto</FirstName>
</Person.Person>
```

Тип данных XML

- столбцы таблицы
- переменные
- входные или выходные параметры в хранимых процедурах или функциях

Нельзя использовать UNIQUE, PRIMARY KEY или FOREIGN KEY

```
CREATE TABLE xmltab(id INTEGER NOT NULL PRIMARY KEY,
                    xml_column XML);
```

XML Schema

- встроенный механизм, позволяющий проверять на корректность XML-документы
- XML SCHEMA COLLECTION
- может содержать один или более XML-SCHEMA-документов
- для XML-столбца можно указать имя коллекции схем

XML Schema

```
CREATE XML SCHEMA COLLECTION EmployeeSchema AS
N'<?xml version="1.0" encoding="UTF-16"?>
<xsd:schema elementFormDefault="unqualifled"
attributeFormDefault="unqualified"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
<xsd:element name="employees">
<xsd:complexType mixed="false">
<xsd:sequence>
<xsd:element name="fname" type="xsd:string"/>
<xsd:element name="lname" type="xsd:string"/>
<xsd:element name="department" type="xsd:string"/>
<xsd:element name="salary" type="xsd:integer"/>
<xsd:element name="comments" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>';
```

Индексирование XML

- Первичный XML-индекс:
  - Индексируются все теги, значения и пути
  - Используется для возвращения скалярных значений или поддеревьев

```
CREATE PRIMARY XML INDEX index_xml_column ON xmltab(xml_column);
```

Индексирование XML

- Три типа вторичных типа XML-индексов:
  - FOR PATH — по структуре
  - FOR VALUE — по значениям элементов и атрибутов
  - FOR PROPERTY — по свойствам

```
CREATE XML INDEX i_xmlcolumn_path ON xmltab(xml_column)
USING XML INDEX index_xml_column FOR PATH;
```

Индексирование XML

- Не могут быть составными
- Не могут быть кластеризованными

Индексирование XML

- sys.xml_indexes

```
SELECT USING_XML_INDEX_ID, SECONDARY_TYPE FROM SYS.XML_INDEXES;
```

Results | Messages

using_xml_index_id	secondary_type
NULL	NULL
256000	P

Представления каталога

```
SELECT * FROM SYS.XML_SCHEMA_ATTRIBUTES;

SELECT * FROM SYS.XML_SCHEMA_ELEMENTS;

SELECT * FROM SYS.XML_SCHEMA_COMPONENTS;
```

Results | Messages

xml_component_id	xml_collection_id	xml_namespace_id	is_qualified	name	symbo
65554	65548	1	1	employees	E
65557	65548	1	0	fname	E
65558	65548	1	0	lname	E
65559	65548	1	0	department	E
65560	65548	1	0	salary	E
65561	65548	1	0	comments	E

Запрос данных из XML

- язык запросов Xpath
  - XML Path Language — язык запросов к элементам XML-документа
- язык запросов Xquery
  - XQuery — язык запросов, разработанный для обработки данных в формате XML

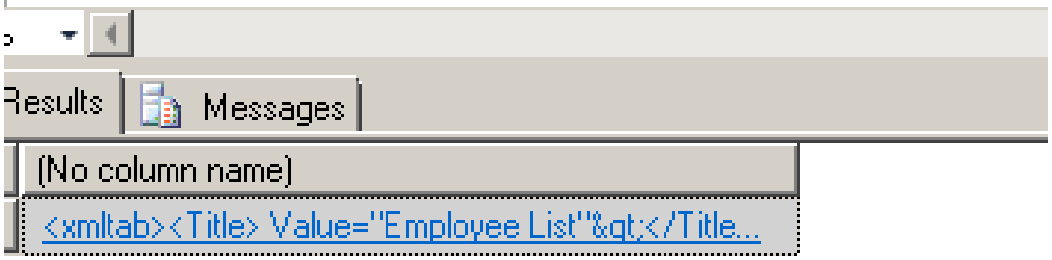
Запрос данных

- query() – выполнение запроса
- exist() – проверка существования
- value() – возвращает значение атрибута
- nodes() – возвращает набор узлов
- modify() – изменение документа
  - insert
  - delete
  - replace value of

Запрос данных

```
------------------------------
SELECT xml_column.query('/PersonList/Title')
FROM xmltab
FOR XML AUTO, TYPE;

------------------------------
```

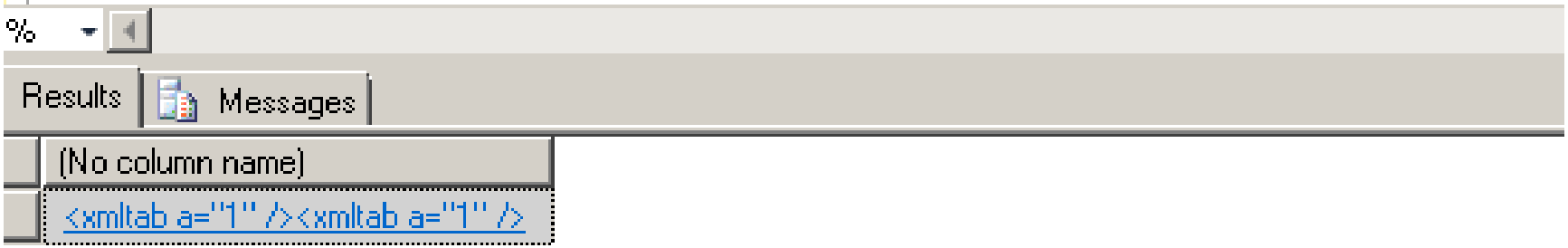



```
<xmltab>
  <Title> Value="Employee List"&gt;</Title>
</xmltab>
<xmltab>
  <Title> Value="Employee List"&gt;</Title>
</xmltab>
```

Запрос данных

```
SELECT xml_column.exist('/PersonList/Title/@Value=EmployeeList') AS a
FROM xmltab
FOR XML AUTO, TYPE;

-----------------------------------------------
```



Изменение данных

```
DECLARE @myXMLDoc xml;
SET @myXMLDoc = '<Root>
    <ProductDescription ProductID="1" ProductName="Road Bike">
        <Features>
        </Features>
    </ProductDescription>
        <ProductDescription ProductID="2" ProductName="Mountain Bike">
        <Features>
        </Features>
    </ProductDescription>
</Root>'  ;
SELECT @myXMLDoc;

-- insert
SET @myXMLDoc.modify('
insert <Maintenance>3 year parts and labor extended maintenance is available</Maintenance>
into (/Root/ProductDescription/Features)[1]') ;
SELECT @myXMLDoc ;

-- update
SET @myXMLDoc.modify('
  replace value of (/Root/ProductDescription/@ProductName)[1]
  with     "The new bike"  ');
SELECT @myXMLDoc;

-- delete
SET @myXMLDoc.modify('
  delete /Root/ProductDescription/@ProductName
')
SELECT @myXMLDoc;
```

Вопросы?