**Определения (Лекция 16)**

|  |  |
| --- | --- |
| **XML** (e**X**tensible **M**arkup **L**anguage)  <person>  <firstName>Иван</firstName>  <lastName>Иванов</lastName>  <address>  <streetAddress>Московское ш., 101, кв.101</streetAddress>  <city>Ленинград</city>  <postalCode>101101</postalCode>  </address>  <phoneNumbers>  <phoneNumber>812 123-1234</phoneNumber>  <phoneNumber>916 123-4567</phoneNumber>  </phoneNumbers>  </person> | **JSON** (**J**ava**S**cript **O**bject **N**otation)  {  "firstName": "Иван",  "lastName": "Иванов",  "address": {  "streetAddress": "Московское ш., 101, кв.101",  "city": "Ленинград",  "postalCode": 101101  },  "phoneNumbers": [  "812 123-1234",  "916 123-4567"  ]  } |

**Как сконструировать XML:**

|  |  |
| --- | --- |
| DECLARE @x xml  SET @x=  '<Empl EmplID = "2">  Name and Features Empl  <Name>Ivanov Ivan</Name>  <Features>  <Dep>IT</Dep>  <Salary>50000</Salary>  </Features>  </Empl>'  SELECT @x | <Empl EmplID="2">  Name and Features Empl  <Name>Ivanov Ivan</Name>  <Features>  <Dep>IT</Dep>  <Salary>50000</Salary>  </Features>  </Empl> |

**Структура XML:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DECLARE @hdoc int  EXEC sp\_xml\_preparedocument  @hdoc OUTPUT, @x  SELECT @hdoc  SELECT \*  FROM OPENXML (@hdoc, '/',1)  ORDER BY parentid asc  EXEC sp\_xml\_removedocument @hdoc | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **id** | **parentid** | **nodetype** | **localname** | **text** | | 0 |  | 1 | Empl |  | | 2 | 0 | 2 | EmplID |  | | 3 | 0 | 3 | #text | Name and Features Empl | | 4 | 0 | 1 | Name |  | | 5 | 0 | 1 | Features |  | | 8 | 2 | 3 | #text | 2 | | 9 | 4 | 3 | #text | Ivanov Ivan | | 6 | 5 | 1 | Dep |  | | 7 | 5 | 1 | Salary |  | | 10 | 6 | 3 | #text | IT | | 11 | 7 | 3 | #text | 50000 | |

**Как генерировать xml на основе параметров:**

|  |  |
| --- | --- |
| DECLARE @x xml  SET @x='<DepID>5</DepID><XID>8</XID>'  SELECT @x.query('<Empl><Features><Dep> **{/DepID}** </Dep></Features></Empl>') | <Empl>  <Features>  <Dep>  <DepID>5</DepID>  </Dep>  </Features>  </Empl> |

**Вариант 2 (выделить только значение параметра)**

|  |  |
| --- | --- |
| DECLARE @x xml  SET @x='**<DepID>5</DepID>**<XID>8</XID>'  SELECT @x.query('<Empl><Features><Dep> **{ data(/DepID) }** </Dep></Features></Empl>') | <Empl>  <Features>  <Dep>5</Dep>  </Features>  </Empl> |

**Как создать атрибут у узла:**

|  |  |
| --- | --- |
| DECLARE @x xml  SET @x='<EmplID>5</EmplID>'  SELECT @x.query('<Empl EmplID="{ /EmplID }" ></Empl>') | <Empl EmplID="5" /> |

**Пример XQuery (создание одного узла из нескольких):**

|  |  |
| --- | --- |
| DECLARE @x xml  SET @x = '  <News>  <page>First page of news</page>  <page>Second page of news</page>  <page>Third page of news</page>  </News>'  select @x.query('  <FullNews>  { string(/News[1]/page[1]) }  { string(/News[1]/page[2]) }  { string(/News[1]/page[3]) }  </FullNews>') | <FullNews>First page of newsSecond page of newsThird page of news</FullNews> |

**Пример XQuery:**

|  |  |
| --- | --- |
| DECLARE @x xml  SET @x = '  <News>  <page>First page of news</page>  <page>Second page of news</page>  <page>Third page of news</page>  </News>'  select @x.query('  <FullNews>  {  for $i in /News[1]/page  return string($i)  }  </FullNews>') | <FullNews>First page of news Second page of news Third page of news</FullNews> |

**Пример Constructed XML:**

|  |  |
| --- | --- |
| DECLARE @x xml  SET @x=''  SELECT @x.query(  'element Empl  {  attribute EmplID { 2 },  text {"Name and Features Empl"},  element Name  { text {"Ivanov Ivan"} },  element Features  {  element Dep  { text{"IT"} },  element Salary  { text{"50000"} }  } }' ) | <Empl EmplID="2">  Name and Features Empl  <Name>Ivanov Ivan  </Name>  <Features>  <Dep>IT</Dep>  <Salary>50000</Salary>  </Features>  </Empl> |

**Как сгенерировать XML по данным в базе:**

|  |  |
| --- | --- |
| DECLARE @empl TABLE (id int, name nvarchar(50), depID int)  INSERT INTO @empl VALUES  (1, 'Ivanov', 1), (2, 'Petrov', 2)  DECLARE @orders TABLE (Oid int, Data datetime, emplID int)  INSERT INTO @orders VALUES  (1002, '20160415', 1), (1003, '20160416', 1),  (1004, '20160417', 2), (1005, '20160418', 1)  SELECT empl.id, empl.name, empl.depID, orders.Oid, orders.Data  FROM @empl empl INNER JOIN @orders orders ON  empl.id = orders.emplID  **FOR XML AUTO** | <empl id="1" name="Ivanov" depID="1">  <orders Oid="1002" Data="2016-04-15T00:00:00" />  <orders Oid="1003" Data="2016-04-16T00:00:00" />  <orders Oid="1005" Data="2016-04-18T00:00:00" />  </empl>  <empl id="2" name="Petrov" depID="2">  <orders Oid="1004" Data="2016-04-17T00:00:00" />  </empl> |

**Как залить xml-файл на сервер:**

DECLARE @docs TABLE (id int, st xml)

INSERT INTO @docs

SELECT 10, xCol

FROM

(

SELECT \*

FROM OPENROWSET (BULK 'Путь к файлу\test.xml', SINGLE\_CLOB) AS xCol

) AS R(xCol)

SELECT \* FROM @docs

**Методы XML:**

CREATE TABLE #Countries (id int primary key, name nvarchar(100), xmlData XML)

INSERT INTO #Countries (id, name, xmlData) VALUES

(1, 'Russian Federation',

'<indicators>

<indicator name = "Population">

<comments><comment>Total population</comment></comments>

<IndicatorCode>SP.POP.TOTL</IndicatorCode>

<data year = "2011" size = "142956460"/>

<data year = "2012" size = "143178000"/>

<data year = "2013" size = "143499861"/>

</indicator>

<indicator name = "UnemploymentRate">

<comments><comment>Unemployment, total (% of total labor force) (modeled ILO estimate)</comment></comments>

<IndicatorCode>SL.UEM.TOTL.ZS</IndicatorCode>

<data year = "2011" size = "6.5"/>

<data year = "2012" size = "5.5"/>

<data year = "2013" size = "5.6"/>

</indicator>

</indicators> ')

**Метод query()**

|  |  |
| --- | --- |
| SELECT name,  xmlData.query('/indicators/indicator[2]/data') AS Xml1  FROM #Countries; | <data year="2011" size="6.5" />  <data year="2012" size="5.5" />  <data year="2013" size="5.6" /> |
| SELECT name,  xmlData.query('/indicators/indicator[2]/data[2]') AS Xml1  FROM #Countries; | <data year="2012" size="5.5" /> |
| SELECT name, xmlData.query ('/indicators/indicator/comments  [comment [contains(., "total") ] ]') AS ContainsComment  FROM #Countries | <comments>  <comment>Unemployment, total (% of total labor force) (modeled ILO estimate)</comment>  </comments> |
| SELECT name, xmlData.query  ('/indicators/indicator/comments  [comment [contains(lower-case(.), "total") ] ]') AS ContainsComment  FROM #Countries | <comments><comment>Total population</comment></comments>  <comments><comment>Unemployment, total (% of total labor force) (modeled ILO estimate)</comment>  </comments> |
| SELECT name,  xmlData.query('/indicators/indicator/data  /.[@year = "2012"]') as Data  FROM #Countries | <data year="2012" size="143178000" />  <data year="2012" size="5.5" /> |

**Метод value()**

|  |  |
| --- | --- |
| SELECT name, xmlData.value  (' (/indicators/indicator[2]/  IndicatorCode/**text()** )[1]', 'varchar(100)') as val  FROM #Countries | name val  Russian Federation SL.UEM.TOTL.ZS |
| SELECT name, xmlData.value  (' (/indicators/indicator[1]/  data[2]/**@size** ) [1]', 'int') AS size  FROM #Countries | name size  Russian Federation 143178000 |

**Метод nodes()**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT name, T.query('.') AS val, T.value('@year', 'int') Year, T.value('@size', 'float') Pop  FROM #Countries C CROSS APPLY C.xmlData.nodes  ('/indicators/indicator/data') as data(T) | |  |  |  | | --- | --- | --- | | **val** | **Year** | **Pop** | | <data year="2011" size="142956460" /> | 2011 | 142956460 | | <data year="2012" size="143178000" /> | 2012 | 143178000 | | <data year="2013" size="143499861" /> | 2013 | 143499861 | | <data year="2011" size="6.5" /> | 2011 | 6.5 | | <data year="2012" size="5.5" /> | 2012 | 5.5 | | <data year="2013" size="5.6" /> | 2013 | 5.6 | |

**Метод exists()**

|  |  |
| --- | --- |
| SELECT name, xmlData.exist  ('/indicators/indicator[2]/data  [@size="142956460"]') AS SizeExists1  FROM #Countries | SELECT name, xmlData.exist  ('/indicators/indicator[1]/data[1]  [@size = 142956460]') AS SizeExists2  FROM #Countries |
| SELECT name, xmlData.exist  ('/indicators/indicator[2]/data  [@size > "45.4"]') AS SizeExists3  FROM #Countries | SELECT name, xmlData.exist  ('/indicators/indicator[2]/data  [@size[xs:float(.) > xs:float("45.4")]]') AS SizeExistsType4  FROM #Countries |

**Метод modify()**

CREATE TABLE #test\_mdf (name nvarchar(100), xmlData xml)

INSERT INTO #test\_mdf VALUES

('test',

'<Empl>

<data> <num>1</num><val>15</val> </data>

<data> <num>2</num><val>20</val> </data>

</Empl>')

|  |  |
| --- | --- |
| UPDATE #test\_mdf  SET xmlData.modify  ('replace value of  (/Empl/data [num[text()="1"]]  /val /text()) [1] with "10"') | <Empl>  <data><num>1</num><val>10</val></data>  <data><num>2</num><val>20</val></data>  </Empl> |
| UPDATE #test\_mdf  SET xmlData.modify  ('insert (/Empl/data[1]) as last  into (/Empl)[1]') | <Empl>  <data><num>1</num><val>15</val></data>  <data><num>2</num><val>20</val></data>  <data><num>1</num><val>15</val></data>  </Empl> |
| UPDATE #test\_mdf  SET xmlData.modify  ('delete (/Empl/data[1])')  (последовательно после 2-ого) | <Empl>  <data><num>2</num><val>20</val></data>  <data><num>1</num><val>15</val></data>  </Empl> |

CREATE TABLE #test\_modify

(name nvarchar(100), xmlData xml)

INSERT INTO #test\_modify

VALUES

('test',

'<Empl>

<data num = "1" val = "15"/>

<data num = "2" val = "20"/>

</Empl>')

|  |  |
| --- | --- |
| UPDATE #test\_modify  SET xmlData.modify  ('replace value of  (/Empl/ data[@num = "1"] /@val)[1]  with "10"') | <Empl>  <data num="1" val="10" />  <data num="2" val="20" />  </Empl> |
| UPDATE #test\_modify  SET xmlData.modify('  insert  (  attribute EmplID { "8" }  )  into (/Empl)[1]  ') | <Empl EmplID="8">  <data num="1" val="10" />  <data num="2" val="20" />  </Empl> |