

# XML

## 1. XML

- Prvky, atribúty, správne štruktúrovaný
- SQL SERVER a XML

## 2. Úvod do XPATH a XQUERY

## 3. XSD (XML Schema Definition)

- Prvky, Atribúty, Typy a Indikátory správny dokument

## 1. XML

### a) Definícia

### b) Značkovacie jazyky: HTML, ...

### c) Prvky, atribúty, správne štruktúrovaný a správny dokument

### d) SQL SERVER a XML

### e) Úvod do XPATH a XQUERY

### a) Definícia

**XML ako metajazyk slúži na popis štruktúry, výmenu a ukladanie dát.**

XML (eXtensible Markup Language - rozšíriteľný značkovací jazyk) - W3C (World Wide Web Consortium)

- ako jazyk dovoľuje popísať (aj hierarchickú) štruktúru dokumentu, vytvárať štruktúrované dokumenty
- ako formát súboru, obsahujúci dáta, umožňuje výmenu údajov medzi aplikáciami a ich uloženie
- umožňuje oddelenie obsahu od dizajnu dát

### b) Značkovacie jazyky: HTML, ...

XML je pokračovanie jazyka SGML, je jeho jednoduchšia verzia.

Ale má prísnejšiu syntax ako HTML.

```
<html>
```

```
<body>
```

```
Blava
```

```
  <table style="width: 24%;">
```

```
    <tr>
```

```
      <td>1</td>
```

```
      <td>2</td>
```

```
    </tr>
```

```
    <tr>
```

```
      <td>3</td>
```

```
      <td style="background-color: blue">4</td>
```

```
    </tr>
```

```
  </table>
```

```
</body>
```

```
</html>
```

Ďalšie jazyky pre podporu práce s XML:

**XSD** (XML Schema Definition), **XPath**, **XQuery** + **XML DML**, **XSLT**

### c) Prvky, atribúty a správne štruktúrovaný dokument

XML dokument **sa skladá**

- z užívateľom pomenovaných prvkov a
- z malého počtu preddefinovaných prvkov.

**Prvok/element** je kombinácia užívateľom pomenovaných značiek (tagov <...>) a dát. Každý prvok má svoj typ/názov, uvedený v značke. Značka slúži na označenie, popis prvku. Prvok môže obsahovať aj **atribúty**.

XML dokument je **správne štruktúrovaný/dobre vytvorený** ak dodržiava isté pravidlá - spĺňa špecifikáciu XML: [http://www.w3schools.com/xml/xml\\_validator.asp](http://www.w3schools.com/xml/xml_validator.asp)

- dokument XML musí obsahovať koreňový prvok, v ktorom sú všetky ostatné prvky vložené
- každý prvok musí mať začiatočnú aj koncovú značku <prvok1> </prvok1>
  - existuje skrátenejší zápis pre prvky, neobsahujúce data <prvok1 />
  - názvy prvkov sú citlivé na malé a veľké písmená
  - hodnoty atribútov píšeme do úvodzoviek/ľavých apostrofov (a do začiatočných tagov)
- značky prvkov musia byť správne vnorené (vnútorný prvok má byť úplne obsiahnutý vo vonkajšom)

```
<?xml version="1.0" ?>
<!-- 1.priklad s komentarom -->
<senat>
  <student rocnik="1"> Elizabet </student>
  <student />
  <![CDATA[<prv1> 1 &lt; 2 </prv1>]]>
</senat>
```

<	&lt;
>	&gt;
&	&amp;
'	&apos;
"	&quot;

Špeciálne znaky:

### MS Visual Studio (.NET)

Dáta môžeme vložiť do/popísať v XML dokumentu/e ako **prvok alebo atribút**.

1	2	3	
4	5	6	70

```
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Matrix>
  <row Col1="1" Col2="2" Col3="3" />
  <row Col1="4" Col2="5" Col3="6" Col4="70" />
</Matrix>
```

```

<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<?xml-stylesheet type='text/xsl' href='MatrixElements.xsl' ?>
<Matrix>
  <row>
    <col>1</col>
    <col>2</col>
    <col>3</col>
  </row>
  <row>
    <col>4</col>
    <col>5</col>
    <col>6</col>
    <col>70</col>
  </row>
</Matrix>

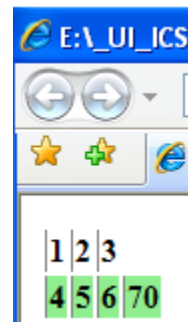
```

Na tento dokument `MatrixElements.xsl` sa odvoláva predchádzajúci. `xsl:template` prvok popisuje množinu prvkov a ako ich spracovať.

```

<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html" />
<xsl:template match="/">
<HTML>
<BODY>
<table>
<xsl:for-each select="//Matrix/row">
<tr>
<xsl:if test="position() mod 2 = 0">
<xsl:attribute name="bgcolor">lightgreen</xsl:attribute>
</xsl:if>
<xsl:for-each select="col">
<td style="text-align:right;font-weight:bold;border-left:solid 0.5pt gray">
<xsl:value-of select="." />
</td>
</xsl:for-each>
</tr>
</xsl:for-each>
</table>
</BODY>
</HTML>
</xsl:template>
</xsl:stylesheet>

```



## Menné priestory MP

MP/Name Space v XML je unikátna kolekcia disjunktných mien definovaných pomocou URI referencie a atribútu `xmlns`. Menné priestory môžu byť definované implicitne alebo explicitne.

```
<matice xmlns = "http://torok.csaba.com/csaba">
```

## d) SQL SERVER a XML

*Stĺpce, premenné a parametre* procedúr a funkcií môžu byť typu **XML**. XML môžeme

1. písať
2. načítať
3. premapovať
4. generovať

## 1. písat' premennú a stĺpec

a) Premennú:

```
DECLARE @var XML
SET @var = '
    <AAA>
        <BBB />
        <BBB nazov="hm"> "baba" </BBB>
    </AAA> '
SELECT @var Ako
```

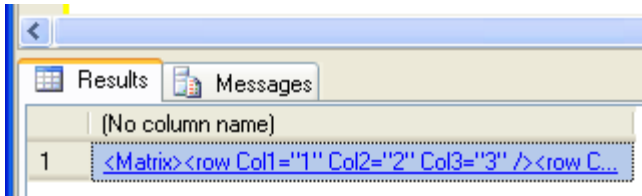


b) Stĺpec:

```
IF OBJECT_ID('T') IS NOT NULL DROP TABLE T
GO
CREATE TABLE T (k int, XmlColumn xml default
                N'<kor><a /><prv2>222</prv2></kor>')
GO
INSERT T(k) VALUES (1);
SELECT * FROM T
```

## 2. načítat' - FROM openrowset(bulk 'cesta', single\_clob)

```
DECLARE @xmlPr xml -- s premennou neide
SET @xmlPr = (
    select * from openrowset
        (bulk N'C:\ MatrixAttributes.xml', single_clob)
    AS a )
SELECT @xmlPr
```



## 3. premapovat' - FROM Osoba FOR XML AUTO

```
USE OsobaVztah
DECLARE @xmlDoc xml
SET @xmlDoc = (SELECT * FROM Osoba
                FOR XML AUTO) -- , ELEMENTS XSINIL
SELECT @xmlDoc
```

Poznamenáme, že v prípade:

```
SET @xmlDoc = (SELECT * FROM Osoba FOR XML AUTO )
```

- z výsledku chýbajú NULL hodnoty, riadky neobsahujú rovnaký počet atribútov a vygenerovaná XSD schéma (pozri nižšie) sa opiera o atribúty

```
SET @xmlDoc = (SELECT * FROM Osoba FOR XML AUTO , ELEMENTS XSINIL )
```

- výsledok obsahuje NULL hodnoty a XSD schéma sa konštruuje pomocou prvkov a nie atribútov.

#### 4. generovať (pomocou query a for)

```
declare @x xml
```

```
set @x= ''
```

```
SELECT @x.query('
                for $a in (1, 2, 3)
                return $a*10
            ') -- = 1 2 3
```

```
declare @x xml
```

```
set @x= ''
```

```
SELECT @x.query('
                for $a in (1, 2, 3)
                return <a10> {$a*10} </a10>
            ')
```

```
--<a10>10</a10>
```

```
--<a10>20</a10>
```

```
--<a10>30</a10>
```

```
-- TOTO ani netreba tu ukazat.
```

```
declare @x xml
```

```
set @x='<ROOT><a>789</a></ROOT>'
```

```
SELECT @x
```

```
SELECT @x.query('
    for $a in (xs:string( "haha"), xs:double( "-1.23" ),
data(/ROOT/a ))
    return $a') -- result test -1.23 789
```

## 2. Úvod do XPATH a XQUERY

[http://msdn.microsoft.com/en-us/library/ms254471\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms254471(v=vs.110).aspx)

XML sa líši od relačných dát, preto potrebuje vlastný dopytovací jazyk.

### Príklady:

```
declare @x xml
set @x='<root>
        <Person Name="B" />
        <Person />
        <Person Name="A" />
    </root>'
select @x
--select @x.query('<Person NAME="A"/>')
----<Person />
----<Person Name="B" />
```

```
select @x.query(
    'for $person in //Person
        order by $person/@Name
        return $person')
```

Alebo:

```
... //Person[3] ...
... //Person[last()] ...
... //Person[@Name="B"] ...
... //Person[attribute::Name="B"] ...
```

```
declare @x xml
set @x='<root>
        <Person Name="A" />
        <Person />
        <Person Name="B" />
    </root>'
select T.jaj.query(
    'for $person in //Person[attribute::Name]
        order by $person/@Name
        return $person')
FROM @x.nodes('.') AS T(jaj)
```

### 3. XSD (XML Schema Definition)

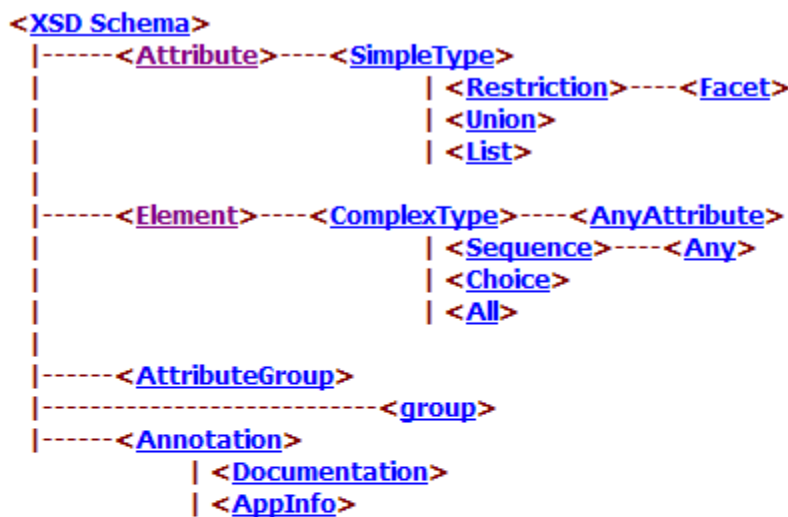
<http://msdn.microsoft.com/en-us/library/ms405151.aspx>

<http://msdn.microsoft.com/en-us/library/aa68549.aspx>

**XML Schema** je popis prípustných značiek, atribútov, ich štruktúry a typu. Ak XML dokument obsahuje iba značky danej schémy, hovoríme, že je **voči nej správny/validný**.

XML Schema, ktorá má viac možností ako DTD (Document Type Definition), nevyžaduje špecifickú syntax a umožňuje špecifikovať

- viachodnotové dátové typy
- počet výskytu prvkov
- maximálnu / minimálnu hodnotu prvku
- postupnosť a množinu prvkov
- OO prvky



XSD Schema Elements:

- [<XSD Schema>](#)
- [<Attribute>](#)
- [<Element>](#)
- [<ComplexType>](#)
- [<SimpleType>](#)
- [<Sequence>](#)
- [<Choice>](#)
- [<All>](#)
- [<Restriction>](#)
- [<Union>](#)
- [<List>](#)
- [<Facet>](#)
- [<Any>](#)
- [<AnyAttribute>](#)
- [<group>](#)
- [<AttributeGroup>](#)
- [<Annotation>](#)
- [<Documentation>](#)
- [<AppInfo>](#)

#### Príklad:

Vytvorte schému pre špeciálnu maticu s tromi stĺpcami, kde do prvého stĺpca môžeme uložiť iba hodnoty 1, 4, 7, 10.

#### Šablony:

```
<Matrix>  
  <row Col1="1" Col2="2" Col3="3" />  
  <row Col1="4" Col2="5" Col3="6" />  
  <row Col1="7" Col2="8" Col3="9" />  
  <row Col1="10" Col2="11" Col3="12" />  
</Matrix>
```

```
#####
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="int1_4_7_10">
    <xs:restriction base="xs:nonNegativeInteger">
      <!--
      <xs:maxInclusive value="100"></xs:maxInclusive>
      -->
      <xs:enumeration value="1"></xs:enumeration>
      <xs:enumeration value="4"/>
      <xs:enumeration value="7"/>
      <xs:enumeration value="10"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:element name="Matrix">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="row">
          <xs:complexType>
            <xs:attribute name="Col1" type="xs:int" use="required" />
            <xs:attribute name="Col2" type="xs:int" use="required" />
            <xs:attribute name="Col3" type="xs:int" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

```
#####
USE tempdb
GO

----IF OBJECT_ID('hahaTab') IS NOT NULL
DROP TABLE hahaTab
----IF OBJECT_ID('hahaSch') IS NOT NULL
DROP XML SCHEMA COLLECTION hahaSch
GO

CREATE XML SCHEMA COLLECTION hahaSch AS
'Sem pride Upravena Schema: 2b_Gener_Mati_Attr_Schema_Upraveny.xsd' ;
GO

----- Zoznam schema collections:
--SELECT * FROM sys.xml_schema_collections
----- Zoznam namespaces:
```



```
--SELECT name FROM sys.xml_schema_namespaces
```

```
CREATE TABLE hahaTab (  
    i int,  
    x xml (hahaSch))  
GO
```

```
INSERT INTO hahaTab VALUES(1,  
'Sem pride MatrixAttributes_NaScheme.xml (? alebo MatrixAttributes.xml)')
```

```
SELECT * FROM hahaTab
```

Example: [http://msdn.microsoft.com/en-us/library/ms256095\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms256095(v=vs.110).aspx)