

# **NET3000**

## **Database Concepts and SQL**

Instructor: Phil Kaufman

### **Lecture 10**

#### **Publish XML to the Web**

# Agenda

- What is XML?
- Attributes of XML
- XML Structure
- XML Syntax
  - ◆ XML Elements
  - ◆ XML Attributes
- Displaying raw XML in a browser
  - ◆ CSS
  - ◆ JavaScript
  - ◆ XSLT

# Agenda (con't)

- Generating XML from database
  - ◆ FOR XML

# What is XML

- What is XML?
  - XML is a markup language
  - XML stands for **EX**tensible **M**arkup **L**anguage
  - It is used to store and transport data
  - XML was designed to store and transport data
  - XML focus is on what the data is / what the data means
  - XML Separates Data from Presentation

# XML example

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44.95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
    with XML.</description>
  </book>
  <book id="bk102">
    <author>Ralls, Kim</author>
    <title>Midnight Rain</title>
    <genre>Fantasy</genre>
    <price>5.95</price>
    <publish_date>2000-12-16</publish_date>
    <description>A former architect battles corporate zombies,
    an evil sorceress, and her own childhood to become queen
    of the world.</description>
  </book>
</catalog>
```

# Attributes of XML

- XML does not use pre-defined tag?
  - The author of the XML document creates/defines tags
- XML is extensible
  - Adding and removing XML data still renders the XML document valid
- XML is simple
  - XML tools (such as parsers) are numerous
  - XML uses plain text to store, transport, and share data
  - XML is a software and hardware independent

# XML Structure

- XML structure is that of a tree
- XML documents start with a document declaration / prolog
  - ♦ Declarations gives parsers details for preparation to parse

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
```

- ♦ Declarations must be the first line of document, if you provide one (it is technically optional, but recommended)

# XML Structure (con't)

- All XML documents data begin with a “root” element
- XML documents then branch out to “leaves” or “child” elements

```
<root>  
  <child>  
    <subchild>.....</subchild>  
  </child>  
</root>
```



# XML Syntax Rules

- XML syntax rules are very easy (and few)
  - XML documents must contain one root element
  - All XML elements must have a closing tag
  - XML tags are case sensitive
  - XML tags must be properly nested

correct nesting:

```
<root><child><child2> ..... </child2></child></root>
```

incorrect nesting:

```
<root><child><child2> ..... </child></child2></root>
```

# XML Syntax Rules (con't)

- XML attributes are quoted

correct quoting:

```
<team name="teamABC">  
  <jersey_colour>Blue</jersey_colour>  
  <roster_count>12</roster_count>  
</team>
```

incorrect quoting:

```
<team name=teamABC>  
  <jersey_colour>Blue</jersey_colour>  
  <roster_count>12</roster_count>  
</team>
```

# XML Syntax Rules (con't)

- Must use Entity References where applicable

Entity reference	Symbol	Description
&lt;	<	Less than
&gt;	>	Greater than
&amp;	&	Ampersand
&apos;	'	apostrophe
&quot;	“	Quotation mark

# XML Syntax Rules (con't)

- XML comments

```
<!-- This is a comment -->
```

- White-space (spaces) are preserved
- XML documents use LF for new lines

# XML Elements

## ■ What is an XML Element?

- Elements include everything between a start tag and end tag
  - text
  - attributes
  - other elements
  - or a mix of the above
- Empty Elements have no content

`<element></element>`    OR    `<element />`

# XML Elements (con't)

- Elements must be extendable

```
<team name="teamABC">  
  <jersey_colour>Blue</jersey_colour>  
  <roster_count>12</roster_count>  
</team>
```

can extend to

```
<team name="teamABC">  
  <jersey_colour>Blue</jersey_colour>  
  <sponsor>Forever Pizza</sponsor>  
  <roster_count>12</roster_count>  
  <coach>Scott Bowman</coach>  
</team>
```

# XML Attributes

## ■ XML Element can have attributes

- Attributes provide data for a particular element

```
<car manufacturer="audi">
```

- Remember that attributes need to be quoted
- Element or Attribute?
  - There are no rules when to use attribute versus element

```
<team name="teamABC">  
  <jersey_colour>Blue</jersey_colour>  
  <roster_count>12</roster_count>  
</team>
```

```
<team>  
  <name>teamABC</name>  
  <jersey_colour>Blue</jersey_colour>  
  <roster_count>12</roster_count>  
</team>
```

# Displaying raw XML in browser

- All major browsers will display XML
- Raw XML does not appear as HTML style pages
  - ♦ Remember why: XML does not define HOW the data will be presented
- Option 1: use CSS
  - ♦ OK for simple XML, but a complicated XML file cannot be handled with straight CSS
- Option 2: use Javascript
  - ♦ Code required to transpose XML to HTML is verbose
  - ♦ Various Toolkits available to process and beautify XML



# Displaying raw XML in browser (con't)

## ■ Option 3: use XSLT

- XSLT transforms XML into HTML for browser
- XSLT stands for e**X**tensible **S**tylesheet **L**anguage **T**ransformations
- XSLT allows you to add and remove elements and attributes from the outputted file
- XSLT allows you to rearrange, sort, make decisions, conduct tests concerning what elements to hide and show

# XSLT example

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">
<xsl:for-each select="league/team">
  <div style="background-color:teal;color:white;padding:4px">
    <span style="font-weight:bold"><xsl:value-of select="name"/> - </span>
    <xsl:value-of select="jersey_colour"/>
  </div>
  <div style="margin-left:20px;margin-bottom:1em;font-size:10pt">
    <p>
      <xsl:value-of select="coach"/>
      <span style="font-style:italic"> Sponsored by: <xsl:value-of
select="sponsor"/></span>
    </p>
  </div>
</xsl:for-each>
</body>
</html>
```

# XSLT example (con't)

## ■ Reference XSLT stylesheet in your XML document

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="formatteams.xsl"?>
<league>
  <team name="teamABC">
    <jersey_colour>Blue</jersey_colour>
    <sponsor>Forever Pizza</sponsor>
    <roster_count>12</roster_count>
    <coach>Scott Bowman</coach>
  </team>
  .
  .
</league>
```

# XSLT elements

xsl:analyze-string  
xsl:apply-imports  
xsl:apply-templates  
xsl:attribute  
xsl:attribute-set  
xsl:call-template  
xsl:character-map  
xsl:choose  
xsl:comment  
xsl:copy  
xsl:copy-of  
xsl:decimal-format  
xsl:document  
xsl:element  
xsl:fallback  
xsl:for-each  
xsl:for-each-group

xsl:function  
xsl:if  
xsl:include  
xsl:import  
xsl:import-schema  
xsl:key  
xsl:matching-substring  
xsl:message  
xsl:namespace  
xsl:next-match  
xsl:namespace-alias  
xsl:non-matching-substring  
xsl:number  
xsl:otherwise  
xsl:output  
xsl:output-character

xsl:param  
xsl:perform-sort  
xsl:processing-instruction  
xsl:preserve-space  
xsl:result-document  
xsl:sequence  
xsl:sort  
xsl:strip-space  
xsl:stylesheet  
xsl:template  
xsl:text  
xsl:value-of  
xsl:variable  
xsl:when  
xsl:with-param

# SQL Server FOR XML

- Transact-SQL has built in function to send result set to XML formatting
- Example 1

```
SELECT name, jersey_colour, sponsor, roster_count, coach
FROM team
ORDER BY team.name
FOR XML AUTO
;
```

```
<team name="The Crush" jersey_colour="Blue" sponsor="Harrold Carpeting"
roster_count="17" coach="Donald Cherry" />
<team name="Enforcers" jersey_colour="Red" sponsor="Enertech Industries"
roster_count="19" coach="Allan Arbour" />
```

# SQL Server FOR XML (con't)

## ■ Example 2

```
SELECT name, jersey_colour, sponsor, roster_count, coach
FROM team
ORDER BY team.name
FOR XML RAW ('team'), ROOT ('name'), ELEMENTS XSINIL
;
```

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <team>
    <name>The Crush</name>
    <jersey_colour>Blue</jersey_colour>
    <sponsor>Harrold Carpeting</sponsor>
    <roster_count>17</roster_count>
    <coach>Donald Cherry</coach>
  </team>
  <team>
    <name>Enforcers</name>
    <jersey_colour>Red</jersey_colour>
    <sponsor>Enertech Industries</sponsor>
    <roster_count>19</roster_count>
    <coach>Allan Arbour</coach>
  </team>
</name>
```

# SQL Server FOR XML (con't)

## ■ FOR XML syntax

```
[ FOR { BROWSE | <XML> } ]
<XML> ::=
XML
{
  { RAW [ ('ElementName') ] | AUTO }
  [
    <CommonDirectives>
    [ , { XMLDATA | XMLSCHEMA [ ('TargetNameSpaceURI') ] } ]
    [ , ELEMENTS [ XSINIL | ABSENT ] ]
  ]
| EXPLICIT
  [
    <CommonDirectives>
    [ , XMLDATA ]
  ]
| PATH [ ('ElementName') ]
  [
    <CommonDirectives>
    [ , ELEMENTS [ XSINIL | ABSENT ] ]
  ]
}

<CommonDirectives> ::=
[ , BINARY BASE64 ]
[ , TYPE ]
[ , ROOT [ ('RootName') ] ]
```

# Conclusion

We learned about:

- What is XML?
- Attributes of XML
- XML Structure
- XML Syntax (Elements, Attributes)
- Displaying raw XML in a browser (CSS, JavaScript, XSLT)
- XSLT elements
- FOR XML