xml – intro
the xml data type

Bob Beauchemin

Overview

- **XML data type**
  - Storage
  - Implementation
  - Conversion
- **Composition and decomposition overview**
- **Loading XML from a file**
- **Standards**

Why XML in a relational database?

- **XML is *lingua franca* of interop**
  - data transmitted and received as XML
  - widely used in business applications
  - many technologies support it
- **Database will have to deal with XML**
  - better to deal with it directly in the database
  - integrated backup/restore, recovery
  - security

XML in a relational database

- **XML can be stored as text**
  - loses much of value of XML representation
- **XML can be decomposed into multiple relational tables**
  - allows use of relational technologies
  - relational to XML composition can happen in database or middle-tier/client
- **XML can be stored as an xml data type**
  - allows use of XML technologies

XML Support in SQL Server Versions

- **SQL Server 2000**
  - XML Decomposition to Relational – OpenXML
  - Relational Composition to XML – SELECT... FOR XML
  - Server-side XPath Queries
- **SQLXML Web Releases 1-3**
  - Released separately after server release
  - Additional Client and Middle-tier Functionality
- **SQL Server 2005**
  - XML Data Type
  - XML Schema Collections
  - XQuery Support
  - Web Release Functionality combined into SQLXML4
- **SQL Server 2008**
  - Incremental Improvements

Processing XML using SQLXML4

- **Functionality set using IIS and libraries**
- **Installs by default with SQL Server 2005**
  - But not with SQL Server 2008
- **Includes**
  - SQLXML Bulk Loader
  - Client-side SELECT FOR XML formatting
    - Using templates
  - XPath queries using XSD Mapping Schemas
  - Transformation using midtier XSLT transforms
  - Updategrams for data modification
  - Client-side components
    - SQLXML Managed Classes
    - Diffgrams

Using XML through SQLCLR

- **SQLCLR sprocs and functions can use XML**
  - Any code in System.Xml is available
- **SQL Server 2008 adds LINQ to XML**
  - System.Core.dll, System.Xml.Linq available as "approved" assemblies
- **XML data type maps to System.Data.SqlTypes.SqlXml**
  - Use CreateXmlReader static method

XML as a data type

- **The XML data type is native database type**
  - used as type of column in table
  - used as type of parameter in stored procedure
  - used as type of return value of a user-defined function
  - used as type of a variable

Using the XML data type

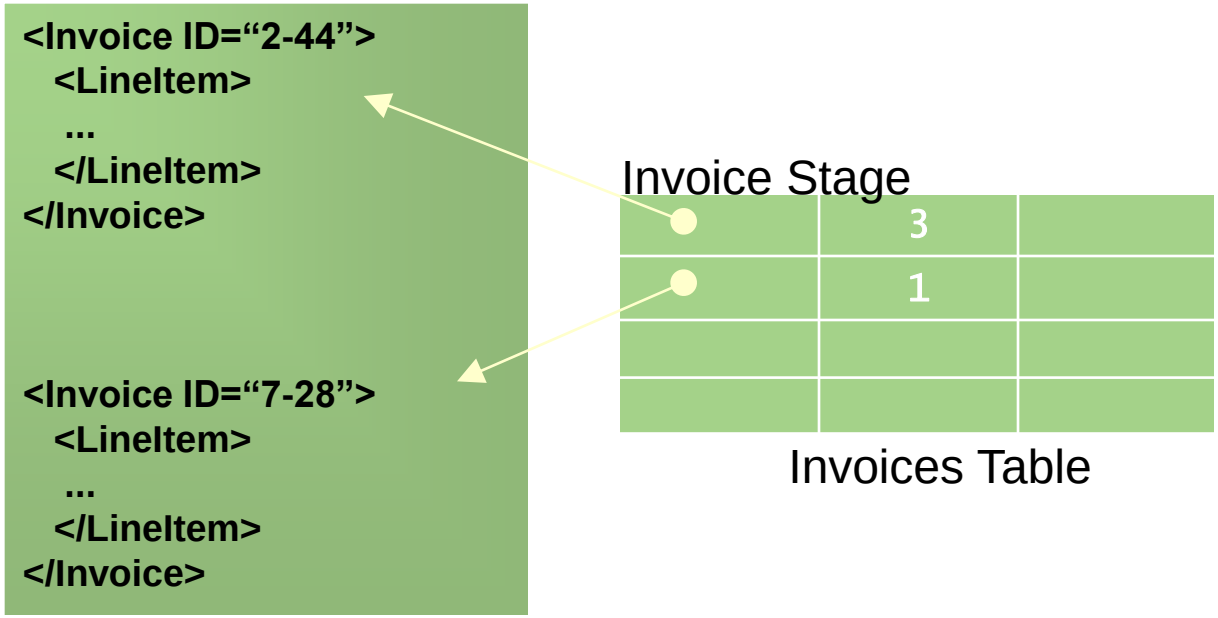
```
CREATE TABLE xml_tab (
    id INTEGER,
    xml_col XML)
```

```
CREATE PROCEDURE transform (
   @xform  XML,
   @indoc  XML,
   @outdoc XML OUTPUT)
AS
...
```

```
CREATE FUNCTION simple (
   @x NVARCHAR(max))
RETURNS XML
AS
DECLARE @a XML
SET @a = @x
...
RETURN @a
```

XML column

- **XML column can store well-formed XML**
  - XML 1.0 recommendation
  - documents or fragments
  - maintains infoset integrity
    - not lexical integrity



Encodings

```
-- This works correctly
INSERT INTO xml_tab VALUES(4,
'<?xml version="1.0" encoding="utf-8"?>
<doc1>
   <row au_id="111-11-1111"/>
</doc1>')

-- This fails, encoding does not agree w/variable
type
INSERT INTO xml_tab VALUES(5,
N'<?xml version="1.0" encoding="utf-8"?>
<doc1>
   <row au_id="111-11-1111"></row>
</doc1>')
```

XML Data Model

- **XML data type loosely based on XQuery 1.0/XPath 2.0 data model**
  - Fragments are valid XML
  - Bare text nodes are allowed
  - NULL value and empty strings are allowed
  - Sequences can be produced by XQuery
- **Not all XML processors handle these**
- **XML is considered data, not documents**
  - CDATA sections are entitized
  - Namespace prefixes are irrevelent
    - May not be retained on XQuery output

Values and conversion

- **XML type can be NULL or empty string**
  - these are not the same
  - XML type can also be "bare text"
- **CAST/CONVERT works**
  - With (n)char and (n)varchar (including MAX)
  - From TEXT/NTEXT to XML
    - But not XML to TEXT/NTEXT
  - From sql_variant (first convert to (n)varchar)
  - With SQLCLR UDT
    - Produces structured XML type
    - Must be CAST/CONVERT for SELECT...FOR XML

XML Type Usage

- **XML data type is not just a text type**
- **XML technologies supported**
  - the contents can validated using XML Schema
    - XML Schema Collections are securables
  - XML-aware indexes are supported
  - XQuery and XPath 2.0 supported
  - In-database XML-related functionality works on the type
    - FOR XML
    - OpenXML

XML data type methods

- **XML data type can use XML-specific methods**
  - all methods use XQuery as input
  - exist - true or false if nodes returned
  - value - single SQL scalar value from query
  - query - returns XML type
  - modify - change XML data in place
  - nodes - return alternate context nodes
    - used as input to "sub-XQuery"

Composition and Decomposition

- **XML data can be stored as relational**
  - Incoming XML broken into columns
    - Multiple tables - one XML parse per table
    - One or more rows per table per document
  - xml.nodes added in SQL Server 2005
  - OpenXML since SQL Server 2000
    - less memory efficient
    - convenient "overflow column"
- **Relational data can be delivered as XML**
  - SELECT...FOR XML
    - RAW, AUTO, EXPLICIT modes
    - PATH mode introduced in SQL Server 2005
  - Many enhancements in SQL Server 2005

XML Import From Files

- **BULK Rowset Provider can import XML**
  - inserts from file to XML column or variable
  - SINGLE_BLOB/CLOB/NCLOB option inserts one XML doc
    - SINGLE_CLOB, SINGLE_NCLOB also work
    - SINGLE_BLOB avoids encoding problems
  - format file required for multi-row insert

```
CREATE TABLE invoices (
  rowid int primary key identity,
  invoice xml )
INSERT invoices
  SELECT * FROM OPENROWSET
  (BULK 'c:\invoice.txt', SINGLE_BLOB)
   as X
```

XML type limitations

- **XML type is not treated like character types**
  - does not support comparison (except to NULL)
    - no equality comparison
    - no ORDER BY, GROUP BY
    - no built-in functions except ISNULL and COALESCE
  - cannot be used as a KEY column
  - cannot be used in a UNIQUE constraint
  - cannot be declared with COLLATE
    - uses XML encodings
    - always stored as UNICODE UCS-2

XML Type & ANSI SQL 2003

- **ANSI SQL part 14 defines standard for XML data type**
  - XML Schema for SQL/XML
    - http://standards.iso.org/iso/9075/2003/sqlxml
  - SQL Server XML data type modeled on current standard
  - types mostly correspond to SQL -> XML mappings
  - standard composition functions not used
    - predated by "SELECT ... FOR XML"

Review

- **XML is a data type in SQL Server**
  - useable in columns, variables, and parameters
  - must adhere to XML 1.0 and Namespaces specs
  - goes beyond those specs – XQuery/XPath 2.0 data model
- **SQL Server functions and XML type**
  - SELECT… FOR XML
  - OpenXML, xml.nodes
- **Database functions support XML**
  - New BULK provider allows insert from files

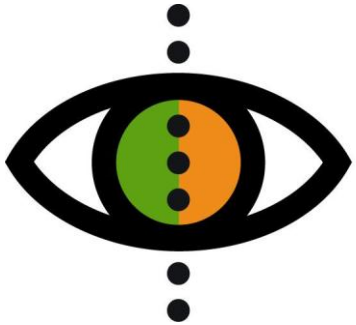
References

- XML Best Practices for Microsoft SQL Server 2005, Shankar Pal et al., MSDN Online
- What's New in FOR XML in Microsoft SQL Server 2005, Michael Rys, MSDN Online
- XML Support in Microsoft SQL Server 2005, Shankar Pal, Mark Fussell, and Irwin Dolobowsky, MSDN Online
- Performance Optimizations for the XML Data Type in SQL Server, Shankar Pal, et al., MSDN Online
- Webcast: Introducing XML in SQL Server 2005

More References

- A Developer's Guide To SQL Server 2005 for Developers - Ch 9, Addison-Wesley, Bob Beauchemin and Dan Sullivan
- Michael Rys' Blog: http://sqljunkies.com/WebLog/mrys
- Shankar Pal Blog: http://blogs.msdn.com/spal/
- My Blog: http://www.SQLskills.com/blogs/bobb
- Denis Ruckebusch's blog: http://blogs.msdn.com/denisruc/default.aspx
- Galex's Semi-Structured Blog http://blogs.msdn.com/galexy/
- John Gallardo's Weblog: http://blogs.msdn.com/jgalla/default.aspx
- Mike Rorke's Blog: http://blogs.msdn.com/mrorke/default.aspx
- Jinghao Liu's Blog (Chinese): http://blogs.msdn.com/jinghaol/

For more in-depth online developer training visit



pluralsight

see what you can learn

on-demand content from authors you trust