xquery
processing xml with xquery methods

Bob Beauchemin

Overview

- **Querying XML**
  - Query methods of XML data type
  - Queries, Functions, and Schema
- **Querying XML in System Structures**
- **Updating XML**

SQL Server 2005 XQuery

- **XQuery is supported through methods on the XML type**
  - xml.exist - returns bit
  - xml.value - returns scalar
  - xml.query - returns XML data type instance
  - xml.nodes - returns one column rowset w/XML column
  - xml.modify - modifies an instance (uses XQuery DML)

xml.exist

- **xml.exist uses XML type and XQuery expression**
  - returns false if query returns empty sequence
  - returns true otherwise
  - returns NULL if XML instance is NULL
- **Usable in XML type check constraints**
  - xml method usage must be encapsulated as UDF

xml.value

- **xml.value return a SQL Server scalar type**
  - xml data type instance, XQuery as input, SQL data type to be returned
  - returns scalar type or NULL
  - cannot return XML type instance
- **Usable within SQL statements**
  - hoist element/attribute values as computed columns
    - UDF must be defined WITH SCHEMABINDING if persisted
  - in predicates (WHERE and HAVING), in GROUP BY, in ORDER BY
  - as column result values in SELECT

Using value method in a view

- **XML data type can be used in SQL View**
  - source or target column in view
  - view columns can be based on XML method results
    - computed columns
  - not usable in distributed or indexed views

computed columns →

```
CREATE VIEW invoice_summary
AS
SELECT
 doc.invoiceid,
 doc.value('count(//invoice)', 'int') AS count,
 doc.value('sum(//value)', 'float') AS value
FROM documents
```

XML column and primary key

- **Sometimes desirable to have xml column be primary key**
  - specifically not supported
- **Persisted computed column based on function using XQuery can be primary key**
- **Secondary column can be primary key**
  - must duplicate information in xml column
  - duplication insured by check based on xml value data type method

xml.query

- **xml.query returns XML data type**
  - XML type instance and XQuery are inputs
  - output is always untyped XML (or NULL)
  - can return scalar as atomic type (XML data type)
  - can use constructors to compose new data

```
DECLARE @xmldoc xml
SET @xmldoc = '<people>...</people>'

SELECT @xmldoc.query('
for $p in /people/person
return $p/name
')
```

Data Accessors

- **XQuery data accessor functions include data() and string()**
  - data() is a function that return the atomized, typed value of zero-or-more items specified as an argument
    - returns typed value of an attribute or element
    - returns string value of a text node
    - returns string value for comment, processing instruction, document node
    - if more than one string value, the values are concatenated
    - with typed complex content as input, it will return an error
  - string() is a function that returns the string value of zero-or-one node
    - when called on a document node ('/') returns concatenated value of all text nodes
    - returns an error when called on more than one node

Using text()

- **text() is an XPath node test**
  - returns text nodes
- **text() behaves differently with typed and untyped XML**
  - with untyped XML, it returns the text node
  - with typed XML, it returns an error on simple content
    - will returns a value with complex content

Relational Data in XQuery

- **SQL Server XQuery can combine relational and XML**
  - sql:variable - use TSQL variable in XQuery
  - sql:column - use column value in XQuery
    - same row as XML type column
    - two-part name (tablename.columnname) required

xml.nodes method

- **nodes can decompose an XML data type**
  - similar to OpenXML
    - usually better performance, XML column is already parsed
  - similar to value(), but produces
    - references (pointers) to XML nodes that can be used a context node in subsequent XML methods
    - many rows rather than single scalar value or single row
    - result is one-column rowset that can only be used as input to XML methods
    - each row contains a pointer to a different context node in the same document
  - T-SQL CROSS/OUTER APPLY can be use to produce rowset from all of the rows in a table
    - similar to OpenXML
  - CROSS APPLY can combine nodes from different nesting levels in document

SQL Server Internal Use of XML

- **XML in SQL Server 2005**
  - Showplan XML, including USE PLAN query hint
  - Deadlock Graph
  - Blocked Process Report
  - Eventdata() function - DDL Triggers
  - Event Notifications Messages
  - Query Notifications - Notifications and Trace Events
  - Trace Events - OLEDB provider information and Execution Warnings/Memory Grant list
  - Bulk Copy
    - XML Format File
    - Bulk Copying XML
  - Surface Area Configuration Tool - Input and Output
  - Database Tuning Advisor
  - SSIS jobs - Reporting Services report files – SSMS/BI Studio Projects
  - XMLA
  - Web Service Interface to SQL Server
  - Web Service Interface to Reporting Services
  - Notification Services - Instance and Application Definition files
- **XML in SQL Server 2008**
  - Format for Serialized Policies in Policy-Based Management
  - Extended Events

Querying XML in System Objects

- **XML is used for data or documents**
- **XML as data in:**
  - EVENTDATA()
  - Query notification data
- **SQL Server 2005 document formats are XML**
  - Query Plans
  - Deadlock graphs, Blocked process report
  - DTA, SAC - input/output parms
  - BCP format files
- **SQL Server projects and queries**
  - SSMS projects
  - SQLRS report definitions
  - SSIS jobs

XML DML

- **xml.modify can mutate the XML instance in place**
  - used in an UPDATE or SET SQL statement
  - modify fails on a NULL XML instance
- **XQuery has no standard DML**
- **SQL Server uses proposed DML extensions within XQuery expressions**
  - insert
  - delete
  - replace value-of

Modification Operators

- **XML DML operators**
  - insert point must be a single node
    - can insert before or after an element
    - can insert first or last sibling
    - can insert subelements or attribute
    - can insert SQL variable (in SQL Server 2008)
  - delete removes all matching nodes
    - if elements have children, removes trees
    - can also remove attributes
  - replace can replace elements or attributes
    - only one replacement point per XML instance
    - only simple types
    - uses "value-of" clause to update value

Actions On Multiple Nodes

- **Insert or replace value-of allow single "action node"**
- **Multi-node insert/replace with T-SQL loop**
  - loop counter is a T-SQL variable

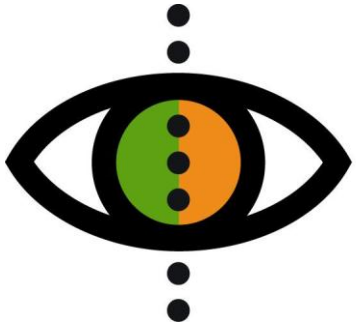
Review

- XQuery be used against XML column instances, parameters, variables
- Exposed through XML methods
  - xml.exist
  - xml.value
  - xml.query
  - xml.nodes
- Relational data can be used in XQuery
- XML DML enables update in place
  - via xml.modify

References

- A Developer's Guide to SQL Server 2005 - Ch 10, Addison- Wesley, Bob Beauchemin and Dan Sullivan
- Pro SQL Server 2008 XML, APress, Michael Coles
- XQuery Labs and Tutorials
  - http://beyondrelational.com/blogs/jacob/archive/2008/06/26/xquery-labs-a-collection-of-xquery-sample-scripts.aspx
- XQuery in Relational Database Systems - Michael Rys
- XQuery Implementation in a Relational Database System, Shankar Pal, et al., Conference on VLDB

For more in-depth online developer training visit



pluralsight

see what you can learn

on-demand content from authors you trust