

xml schema support

Bob Beauchemin



Overview

- **XML Schema Collections**
 - Built-in XML Schema Collection and schemas
- **Storing and Using Schemas**
 - Schema Collection Security
- **Support for XML Schema Productions**
- **Versioning XML Schemas**

XML Schema Collections

- **XML schemas can be associated with an XML type**
 - insures that data inserted into column complies with schemas
 - data called “typed XML” vs. “untyped XML”
- **Set of schemas known as a XML schema collection**
 - schema valid XML can disallow fragments
- **User schema collections have no interdependencies**
 - Predefined XML schemas live in the sys schema collection
 - Useable from any user schema collections

Sys XML Schema Collection

Schema Namespace	Prefix
http://www.w3.org/2001/XMLSchema	xs
http://www.w3.org/2001/XMLSchema-instance	xsi
http://www.w3.org/XML/1998/namespace	xml
http://www.w3.org/2004/07/xpath-datatypes	xdt
http://www.w3.org/2004/07/xpath-functions	fn
http://schemas.microsoft.com/sqlserver/2004/sqltypes	sqltypes
urn:schemas-microsoft-com:xml-sql	
http://schemas.microsoft.com/sqlserver/2004/SOAP	

XML Schemas in Databases

- XML schemas used by XML data types must be in database
- **Create XML SCHEMA COLLECTION**
 - requires literal schemas (literal string or XML variable)
- **Collection name associated with XML instance**
 - column, parameter, or variable

Using XML Schema Collections

literal schema
targetNamespace

```
CREATE XML SCHEMA COLLECTION invcol
AS '<xs:schema ...
    targetNamespace="urn:invoices">
    ...
</xs:schema>'
```

reference to
schema collection

```
CREATE TABLE Invoices (
    int id IDENTITY PRIMARY KEY,
    invoice XML(invcol)
)
```

Typical XML Schema

for namespace

urn:geo →

type definition →

structured
type definition →

element that might
appear in xml data →

```
<xs:schema xmlns:xs=
  "http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:geo"
  xmlns:tns="urn:geo">
  <xs:simpleType name="dim">
    <xs:restriction base="xs:int"/>
  </xs:simpleType>
  <xs:complexType name="Point">
    <xs:sequence>
      <xs:element name="X"
        type="tns:dim"/>
      <xs:element name="Y"
        type="tns:dim"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Point"
    type="tns:Point"/>
</xs:schema>
```

Using strongly typed XML

```
CREATE TABLE point_tab(  
  id int IDENTITY primary key,  
  -- geocoll include schema for 'urn:geo'  
  thepoint xml(CONTENT geocoll)  
GO  
  
-- this works, schema-valid Point  
INSERT INTO point_tab VALUES(  
  '<Point  
xmlns="urn:geo"><X>10</X><Y>20</Y></Point>')  
  
-- this insert fails, foo is not an integer  
INSERT INTO point_tab VALUES(  
  '<Point  
xmlns="urn:geo"><X>10</X><Y>foo</Y></Point>')
```


XML Schema Storage

- **XML schema itself it not stored in database**
 - content stored in various system tables
 - only major schema information stored
- **Some schema information is not stored in database**
 - comments, processing instructions
 - <xs:annotation>

Retrieving Schemas

- **xml_schema_namespace** retrieves schemas
 - input is relational schema, collection name
 - optional parameter to return single schema namespace
 - returns xml type

```
DECLARE @s xml
SELECT @s = xml_schema_namespace(
    N'dbo', N'GeoSchemaColl')

-- or return schema for single namespace
SELECT @s = xml_schema_namespace(
    N'dbo', N'GeoSchemaColl', 'urn:geo')
PRINT CONVERT(VARCHAR(max), @s)
GO
```

Schema Collection Permissions

- Permission required to create/alter a schema collection
 - dbo has this permission
- Permission on schema valid items
 - EXECUTE and REFERENCES

```
-- can create
GRANT CREATE XML SCHEMA COLLECTION TO joe

-- can create items that use collection
GRANT REFERENCES ON
    XML SCHEMA COLLECTION::colname TO fred

-- can query structure of XML items from collection
GRANT EXECUTE ON
    XML SCHEMA COLLECTION::colname TO fred
```

Some XML Schema Limitations

- **Lax validation of <xs:any>, <xs:anyAttribute> supported in SQL Server 2008**
 - Only strict or skip validation in SQL Server 2005
- **Support for <xs:list> and <xs:union> added in 2008**
- **<xs:import> requires schema already in collection**
 - schemaLocation ignored
- **<xs:include>, <xs:redefine>, <xs:notation>,**
- **<xs:key>, <xs:keyref>, <xs:unique>**
 - not supported or allowed

Altering Schema Collections and Columns

- **XML Schema Collections can be ALTERed**
 - ADD new schema to an existing collection
 - ADD (but not change) types in an existing schema
 - you cannot DROP an existing schema from a collection
- **Tables with XML columns can be altered**
 - well-formed to schema valid
 - schema valid to well-formed
 - (N)VARCHAR(MAX), (N)VARCHAR(n) to
 - well-formed XML
 - schema valid XML
 - XML column to
 - (N)VARCHAR(MAX), (N)VARCHAR(n)

Versioning Through ALTER

- **ALTER allows versioning of SCHEMA COLLECTIONs & content**
 - ALTER from schema valid to well-formed
 - drop and recreate, repopulate schema collection
 - update content if necessary
 - ALTER from well-formed to schema valid
 - this causes revalidation against new collection

Review

- **Correlation between SQL Server data types and XML Schema type**
 - Built-in XML Schemas
- **SQL Server serves as a schema repository**
 - XML Schema Collections contain one or more XML Schemas
 - XML Schemas never downloaded from file system or internet
 - XML Schemas are decomposed when cataloged
- **XML Schema Collections associated with**
 - Columns
 - Parameters
 - Variables
- **The XML Schema Collection allows XML schema versioning**

More References

- **A Developer's Guide To SQL Server 2005 - Ch 9, Addison-Wesley, Bob Beauchemin and Dan Sullivan**
- **Essential XML Quick Reference, Addison-Wesley, Aaron Skonnard and Martin Gudgin**
- **The Art of XSD - SQL Server XML Schema Collections, Red Gate Books, Jacob Sebastian**

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**