

XML Schema چیست؟

XML Schema معرف ساختار، نوع داده‌ها و المان‌های یک سند XML است. البته باید در نظر داشت که تعریف XML Schema کاملاً اختیاری است و اگر تعریف شود مزیت اعتبارسنجی داده‌های در حال ذخیره سازی در بانک اطلاعاتی را به صورت خودکار به همراه خواهد داشت. در این حالت به نوع داده‌ای XML دارای اسکیمای، typed XML و به نوع بدون اسکیمای، untyped XML گفته می‌شود.

به یک نوع XML، چندین اسکیمای مختلف را می‌توان نسبت داد و به آن XML schema collection نیز می‌گویند.

XML schema collections پیش فرض و سیستمی

تعدادی XML Schema پیش فرض در SQL Server تعریف شده‌اند که به آن‌ها sys schema collections گفته می‌شود.

```
Prefix - Namespace
xml = http://www.w3.org/XML/1998/namespace
xs = http://www.w3.org/2001/XMLSchema
xsi = http://www.w3.org/2001/XMLSchema-instance
fn = http://www.w3.org/2004/07/xpath-functions
sqltypes = http://schemas.microsoft.com/sqlserver/2004/sqltypes
xdt = http://www.w3.org/2004/07/xpath-datatypes
(no prefix) = urn:schemas-microsoft-com:xml-sql
(no prefix) = http://schemas.microsoft.com/sqlserver/2004/SOAP
```

در اینجا پیشنوندها و فضاهای نام sys schema collections را ملاحظه می‌کنید. از این اسکیمایها برای تعاریف strongly typed امکانات موجود در SQL Server کمک گرفته شده‌است. اگر علاقمند باشید تا این تعاریف را مشاهده کنید به مسیر Program Files\Microsoft SQL Server\version\Tools\Binn\schemas\sqlserver\2006\11\events فایل events.xsd قابل مشاهده است و یا در مسیر Tools\Binn\schemas\sqlserver\2004\07 Tools\Binn\schemas\sqlserver\2004\sqltypes اسکیمای ابزارهای query processor و show plan قابل بررسی می‌باشد. مهم‌ترین آن‌ها را در پوشه Tools\Binn\schemas\sqlserver\2004\sqltypes در فایل sqltypes.xsd می‌توانید ملاحظه کنید. اگر به محتوای آن دقت کنید، قسمتی از آن به شرح ذیل است:

```
<xsd:simpleType name="char">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="nchar">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="varchar">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="nvarchar">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="text">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:simpleType name="ntext">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
```

در اینجا نوع‌های توکار char تا ntext به xsd:string نگاشت شده‌اند و برای اعتبارسنجی datetime و نگاشت آن، از الگوی ذیل استفاده می‌شود؛ به همراه حداقل و حداکثر قابل تعریف:

```
<xsd:simpleType name="datetime">
  <xsd:restriction base="xsd:dateTime">
    <xsd:pattern value="((000[1-9])|(00[1-9][0-9])|(0[1-9][0-9]{2})|([1-9][0-9]{3}))-((0[1-9])|(1[012]))-((0[1-9])|([12][0-9])|(3[01]))T((0[1][0-9])|(2[0-3]))(:[0-5][0-9]){2}(\.[0-9]{2}[037])?">
    <xsd:maxInclusive value="9999-12-31T23:59:59.997"/>
    <xsd:minInclusive value="1753-01-01T00:00:00.000"/>
  </xsd:restriction>
</xsd:simpleType>
```

ادیتور SQL Server management studio به خوبی، گشودن، ایجاد و یا ویرایش فایل‌هایی با پسوند xsd را پشتیبانی می‌کند.

تعریف XML Schema و استفاده از آن جهت تعریف یک strongly typed XML

XML Schema مورد استفاده در SQL Server حتما باید در بانک اطلاعاتی ذخیره شود و برای خواندن آن، برای مثال از فایل سیستم استفاده نخواهد شد.

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

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

در اینجا نحوه‌ی تعریف کلی یک XML Schema collection و سپس انتساب آن را به یک ستون XML ملاحظه می‌کنید. ستون invoice که از نوع XML تعریف شده، ارجاعی را به اسکیمای تعریف شده دارد. در ادامه نحوه‌ی تعریف یک اسکیمای نمونه قابل مشاهده است:

```
CREATE XML SCHEMA COLLECTION geocol AS
'<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:geo"
  elementFormDefault="qualified"
  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" minOccurs="0" maxOccurs="unbounded" />
      <xs:element name="Y" type="tns:dim" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Point" type="tns:Point" />
</xs:schema>'
```

در این اسکیمای ساده به نام dim تعریف شده است که محدودیت آن، ورود اعداد صحیح می‌باشد. همچنین امکان تعریف نوع‌های پیچیده نیز در اینجا وجود دارد. برای مثال نوع پیچیده Point دارای دو المان X و Y از نوع dim در ادامه تعریف شده است. المانی که نهایتاً بر این اساس در XML ظاهر خواهد شد توسط xs:element تعریف شده است. اکنون برای آزمایش اسکیمای تعریف شده، جدول geo_tab را به نحو ذیل تعریف می‌کنیم و سپس سعی در رکورد در آن خواهیم کرد:

```
declare @geo_tab table(
  id int identity primary key,
  point xml(content geocol)
)

insert into @geo_tab values('<Point xmlns="urn:geo"><X>10</X><Y>20</Y></Point>')
insert into @geo_tab values('<Point xmlns="urn:geo"><X>10</X><Y>test</Y></Point>')
```

در اینجا اگر دقت کنید، برای تعریف نام اسکیمای مورد استفاده، واژه content نیز ذکر شده است. Content مقدار پیش فرض است و در آن پذیرش XML Fragments یا محتوای XML ایی با بیش از یک Root element مجاز است. حالت دیگر آن document است که تنها یک Root element را می پذیرد.

در این مثال، insert اول با موفقیت انجام خواهد شد؛ اما insert دوم با خطای ذیل متوقف می شود:

```
XML Validation: Invalid simple type value: 'test'. Location: /*:Point[1]/*:Y[1]
```

همانطور که ملاحظه می کنید، چون در insert دوم، در المان عددی Y، مقدار test وارد شده است و تطابق با اسکیمای تعریف شده ندارد، insert آن مجاز نخواهد بود.

یافتن محل ذخیره سازی اطلاعات اسکیمای در SQL Server

اگر علاقمند باشید تا با محل ذخیره سازی اطلاعات اسکیمای، نوع های تعریف شده و حتی محل استفاده از آن ها در بانک های اطلاعاتی مختلف موجود آشنا شوید و گزارشی از آن ها تهیه کنید، می توانید از کوئری های ذیل استفاده نمایید:

```
select * from sys.xml_schema_collections
select * from sys.xml_schema_namespaces
select * from sys.xml_schema_elements
select * from sys.xml_schema_attributes
select * from sys.xml_schema_types
select * from sys.column_xml_schema_collection_usages
select * from sys.parameter_xml_schema_collection_usages
```

باید دقت داشت زمانیکه یک schema در حال استفاده است (یک رکورد ثبت شده مقید به آن تعریف شده باشد)، امکان drop آن نخواهد بود. حتما باید اطلاعات و ستون مرتبط، ارجاعی را به schema نداشته باشند تا بتوان آن schema را حذف کرد. محتوای اسکیمای ذخیره شده به شکل xsd تعریف شده، ذخیره سازی نمی شود. بلکه اطلاعات آن تجزیه شده و سپس در جداول سیستمی SQL Server ذخیره می گردند. هدف از اینکار، بالا بردن سرعت اعتبارسنجی XML typedها است. بنابراین بدیهی است در این حالت اطلاعاتی مانند comments موجود در xsd تهیه شده در بانک اطلاعاتی ذخیره نمی گردند. برای بازیابی اطلاعات اسکیمای ذخیره شده می توان از متد xml_schema_namespace استفاده کرد:

```
declare @x xml
select @x = xml_schema_namespace(N'dbo', N'geocol')
print convert(varchar(max), @x)
```

برای تعریف و یا تغییر یک XML Schema نیاز به دسترسی مدیریتی یا dbo است (به صورت پیش فرض). همچنین برای استفاده از Schema تعریف شده، کاربر متصل به SQL Server باید دسترسی Execute و References نیز داشته باشد.

نحوه ی ویرایش یک schema collection موجود

چند نکته:

- امکان alter یک schema collection وجود دارد.
- می توان یک schema جدید را به collection موجود افزود.
- امکان افزودن (و نه تغییر) نوع های یک schema موجود، میسر است.
- امکان drop یک اسکیمای از collection موجودی وجود ندارد. باید کل collection را drop کرد و سپس آن را تعریف نمود.
- جداولی با فیلدهای nvarchar را می توان به فیلدهای XML تبدیل کرد و برعکس.
- امکان تغییر یک فیلد XML به حالت untyped و برعکس وجود دارد.

فرض کنید که می خواهیم اسکیمای متناظر با یک ستون XML را تغییر دهیم. ابتدا باید آن ستون XML ایی را Alter کرده و قید اسکیمای آن را برداریم. سپس باید اسکیمای موجود را drop و مجددا ایجاد کرد. همانطور که پیشتر ذکر شد، اگر اسکیمایی در حال

استفاده باشد، قابل drop نیست. در ادامه مجدداً باید ستون XML ایی را تغییر داده و اسکیمای آن را معرفی کرد. روش دوم مدیریت این مساله، اجازه دادن به حضور بیش از یک اسکیمای در مجموعه است. به عبارتی نگارش‌بندی اسکیمای که به نحو ذیل قابل انجام است:

```
alter XML SCHEMA COLLECTION geocol add @x
```

در اینجا به collection موجود، یک اسکیمای جدید (برای مثال نگارش دوم اسکیمای فعلی) اضافه می‌شود. در این حالت geocol، هر دو نوع اسکیمای موجود را پشتیبانی خواهد کرد.

نحوه‌ی import یک فایل xsd و ذخیره آن به صورت اسکیمای

اگر بخواهیم یک فایل xsd موجود را به عنوان xsd معرفی کنیم می‌توان از دستورات ذیل کمک گرفت:

```
declare @x xml
set @x = (select * from openrowset(bulk 'c:\path\file.xsd', single_blob) as x)
CREATE XML SCHEMA COLLECTION geocol2 AS @x
```

در اینجا به کمک openrowset فایل xsd موجود، در یک متغیر xml بارگذاری شده و سپس در دستور ایجاد یک اسکیمای کالکشن جدید استفاده می‌شود. از openrowset برای خواندن یک فایل xml موجود، جهت insert محتوای آن در بانک اطلاعاتی نیز می‌توان استفاده کرد.

محدودیت‌های XML Schema در SQL Server

تمام استاندارد XML Schema در SQL Server پشتیبانی نمی‌شود و همچنین این مورد از نگارشی به نگارشی دیگر نیز ممکن است تغییر یافته و بهبود یابد. برای مثال در SQL Server 2005 از xs:any پشتیبانی نمی‌شود اما در SQL Server 2008 این محدودیت برطرف شده‌است. همچنین مواردی مانند xs:key, xs:keyref, xs:notation, xs:redefine, xs:include و xs:unique در SQL Server پشتیبانی نمی‌شوند.

یک نکته‌ی تکمیلی

برنامه‌ای به نام xsd.exe به همراه Visual Studio ارائه می‌شود که قادر است به صورت خودکار از یک فایل XML موجود، XML Schema تولید کند. [اطلاعات بیشتر](#)