# XML Workshop VII - Validating values with SCHEMA

## Introduction

In the previous workshop we have seen how to work with *TYPED XML* and *XML SCHEMA*s. By using a *TYPED XML* which is bound to an *XML SCHEMA*, you can make sure that the value being stored to your XML column or variable is EXACTLY as per the format that you want them to be. SQL Server will perform this validation on your behalf.

Most of the times, you will be using an *XML* column or variable, when you expect an EXTERNAL application to supply a piece of information to your App. So there is a certain *XML* structure that the EXTERNAL application need to follow. When it sends the data to your application, it has to make sure that the data is as per the required format. Everything will work well, as long as you get the correct data. But what if the EXTERNAL application starts sending you data in an incorrect format? You certainly need to validate the data and reject it, because it does not abide with the required format. But how do you validate it? Well, SQL Server will do the validation for you, if you are using a *TYPED XML* column or variable. You can specify the structure of the data that you expect in an *XML* schema and bind the *XML* variable or column to that *SCHEMA*. The rest will be done by SQL Server.

In the previous workshop we have seen how to do this kind of validation. But the validation that we did previously were only on the structure of the *XML*, not on the value (actual data). For example, we have seen how to make sure that the data that we receive has the required *ELEMENTS* and *ATTRIBUTES*. However, that is not enough. We need to make sure that the value stored in the *ELEMENTS* and *ATTRIBUTES* are correct too. If the *age* element has a value of 500 or -3, then what sense does it make? Or it could be that your application is expecting a 6 digit *employee number* but you receive an *employee number* that is 9 characters long alpha-numeric value. Huh, we really need a way to validate the values!

In this session, we will look into a few examples that explain how to validate the values of *ELEMENTS* and *ATTRIBUTES* by using an *XML SCHEMA*.

## *Data Type* Validation

One of the most important validations that we need to make is that of the data types. *BirthDate* should be a valid date value. *Salary* should be a Number and *Age* cannot have decimals. Let us now write a schema that performs data type validations. [code] [schema]

```
/*
Let us create a SCHEMA for storing Employee Information. The Schema
is pretty simple and the elements are self explanatory. Please note that
each element is defined with a specific data type. While storing the value
to the XML variable or column which is bound to this SCHEMA, SQL Server will
validate the elements as per the given SCHEMA.
*/
CREATE XML SCHEMA COLLECTION EmployeeSchema AS '
<schema xmlns="http://www.w3.org/2001/XMLSchema">
```

```
  <element name="Employee">
    <complexType>
      <sequence>
        <element name="FullName" type="string" />
        <element name="Salary" type="decimal" />
        <element name="Age" type="integer" />
        <element name="Married" type="boolean" />
        <element name="BirthDate" type ="date" />
        <element name="ReportingTime" type="time" />
      </sequence>
    </complexType>
  </element>
</schema>
'
GO

/*
If there is no XML schema, the external application might be sending data in the
following format, which looks correct at first glance. But, since we have a strict
XML Schema, the following XML will not be accepted.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>$10,000</Salary>
    <Age>Thirty</Age>
    <Married>Yes</Married>
    <BirthDate>1975-03-14</BirthDate>
    <ReportingTime>10:00 AM</ReportingTime>
</Employee>
'

/*
OUTPUT:

Msg 6926, Level 16, State 1, Line 4
XML Validation: Invalid simple type value: '$10,000'. Location: /:Employee[1]/:Salary[1]

"Salary" is defined as "decimal" and it cannot take the "$" sign. Let us correct that.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10,000</Salary>
    <Age>Thirty</Age>
    <Married>Yes</Married>
    <BirthDate>1975-03-14</BirthDate>
    <ReportingTime>10:00 AM</ReportingTime>
</Employee>
'

/*
```

```
OUTPUT:

Msg 6926, Level 16, State 1, Line 2
XML Validation: Invalid simple type value: '10,000'. Location: /:Employee[1]/:Salary[1]

Hmmmm..Doesn't it accept comma (,) too? OK, no arguments. Let us correct it.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>Thirty</Age>
    <Married>Yes</Married>
    <BirthDate>1975-03-14</BirthDate>
    <ReportingTime>10:00 AM</ReportingTime>
</Employee>
'

/*
OUTPUT:

Msg 6926, Level 16, State 1, Line 2
XML Validation: Invalid simple type value: 'Thirty'. Location: /:Employee[1]/:Age[1]

The SCHEMA defines "Age" as a numeric value. Hence it should be a number. Let us
correct it too.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>Yes</Married>
    <BirthDate>1975-03-14</BirthDate>
    <ReportingTime>10:00 AM</ReportingTime>
</Employee>
'
/*
OUTPUT:

Msg 6926, Level 16, State 1, Line 2
XML Validation: Invalid simple type value: 'Yes'. Location: /:Employee[1]/:Married[1]

"Married" is defined as "Boolean". It cannot take "Yes"/"No" or "True"/"False". The
"Boolean" field can store only "1" or "0"
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14</BirthDate>
    <ReportingTime>10:00 AM</ReportingTime>
</Employee>
'

/*
OUTPUT:

Msg 6926, Level 16, State 1, Line 2
XML Validation: Invalid simple type value: '1975-03-14'. Location: /:Employee
[1]/:BirthDate[1]

Why does SQL Server reject the date value? Well, the XSD implementation of
"date" data type in
SQL Server expects Time Zone information too. In the following example, I
have put GMT + 5:30
as my Time Zone.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime>10:00 AM</ReportingTime>
</Employee>
'
/*
OUTPUT:
Msg 6926, Level 16, State 1, Line 2
XML Validation: Invalid simple type value: '10:00 AM'. Location: /:Employee
[1]/:ReportingTime[1]

Well, what is the problem here? Probably, it did not like "AM". Let us change
that.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime>10:00:00</ReportingTime>
</Employee>
'
/*
OUTPUT:

Msg 6926, Level 16, State 1, Line 2
XML Validation: Invalid simple type value: '10:00:00'. Location: /:Employee
[1]/:ReportingTime[1]
```

```
SQL Server's implementation of "time" data type expects the Time Zone
information too. The following
is the correct XML which validates with our EmployeeSchema.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime>10:00:00+05:30</ReportingTime>
</Employee>
'
```

## Using *fixed* attribute with elements

Some times we would come across situations where we always expect a fixed value. For example, the *gender* of the employee should be *male* or Marital Status should be *Married*. The following example shows how to do this type of validations by using the *fixed* attribute. [code] [schema]

```
DROP XML SCHEMA COLLECTION EmployeeSchema
GO

/*
Note the usage of attribute "fixed" which restricts the values to be always
"1"
*/

CREATE XML SCHEMA COLLECTION EmployeeSchema AS '
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="Employee">
    <complexType>
      <sequence>
        <element name="FullName" type="string" />
        <element name="Salary" type="decimal" />
        <element name="Age" type="integer" />
        <element name="Married" type="boolean" fixed="1"/>
        <element name="BirthDate" type ="date" />
        <element name="ReportingTime" type="time" />
      </sequence>
    </complexType>
  </element>
</schema>
'
GO

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>0</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime>10:00:00+05:30</ReportingTime>
</Employee>
'

/*
Msg 6921, Level 16, State 1, Line 3
XML Validation: Element or attribute 'Married' was defined as fixed, the
element value has to be
equal to value of 'fixed' attribute specified in definition.
Location: /:Employee[1]/:Married[1]
*/
```

## Validating elements with *default* attribute

The *default* attribute allows to specify a default value for the given element or attribute. When *default* is specified the element or attribute becomes optional. The following example demonstrates the usage of *default* attribute. [code] [schema]

```
DROP XML SCHEMA COLLECTION EmployeeSchema
GO

/*
This example adds the "default" attribute to specify a certain value
if the element is not present.
*/

CREATE XML SCHEMA COLLECTION EmployeeSchema AS '
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="Employee">
    <complexType>
      <sequence>
        <element name="FullName" type="string" />
        <element name="Salary" type="decimal" />
        <element name="Age" type="integer" />
        <element name="Married" type="boolean" fixed="1"/>
        <element name="BirthDate" type ="date" />
        <element name="ReportingTime" type="time" default="10:00:00+05:30"/>
      </sequence>
    </complexType>
  </element>
</schema>
'
GO

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime />
</Employee>
'
/*
This XML validates correctly because there is a default value specified in
the
```

```
SCHEMA for the element <Reportingtime>. If the default value is removed from
the SCHEMA definition, SQL Server will generate the following error while validating
the above XML value.

Msg 6926, Level 16, State 1, Line 3
XML Validation: Invalid simple type value: ''. Location: /:Employee[1]/:ReportingTime[1]
*/
```

## Validating attributes with *use*

Attributes are optional by default. To make an attribute mandatory, you need to add the *use* attribute and set the value to *required*. The following example demonstrates it. [code] [schema]

```
DROP XML SCHEMA COLLECTION EmployeeSchema
GO

/*
This version of the SCHEMA defines a new attribute in the "Employee" node.
The attribute is defined as "required". By default an attribute is "optional".
To make it mandatory, you need to use the "use" attribute and specify the value
"required"
*/

CREATE XML SCHEMA COLLECTION EmployeeSchema AS '
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="Employee">
    <complexType>
      <sequence>
        <element name="FullName" type="string" />
        <element name="Salary" type="decimal" />
        <element name="Age" type="integer" />
        <element name="Married" type="boolean" fixed="1"/>
        <element name="BirthDate" type ="date" />
        <element name="ReportingTime" type="time" default="10:00:00+05:30"/>
      </sequence>
      <attribute name="EmployeeNumber" type="integer" use="required" />
    </complexType>
  </element>
</schema>
'
GO

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee>
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime />
</Employee>
'

```

```
/*
The above code will generate the following error because the attribute
"EmployeeNumber" is a required attribute.

Msg 6906, Level 16, State 1, Line 3
XML Validation: Required attribute 'EmployeeNumber' is missing. Location: /:Employee[1]

If the "required" modifier is not specified, then "optional" is assumed. Hence the above
XML will validate correctly.

The correct XML for the above SCHEMA is given below.
*/

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee EmployeeNumber="1001">
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime />
</Employee>
'

/*
The above XML validates correctly. You will not be able to use an alpha-numeric
employee number, because the SCHEMA defines it to be an INTEGER.
*/
```

## Validating attributes with *fixed* and *default*

The following example demonstrates how to write a schema which validates attributes with *fixed* and *default*. [code] [schema]

```
DROP XML SCHEMA COLLECTION EmployeeSchema
GO

/*
This version of the SCHEMA demonstrates how to implement "fixed" and "default" restrictions
with attributes.
*/
CREATE XML SCHEMA COLLECTION EmployeeSchema AS '
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="Employee">
    <complexType>
      <sequence>
        <element name="FullName" type="string" />
        <element name="Salary" type="decimal" />
        <element name="Age" type="integer" />
        <element name="Married" type="boolean" fixed="1"/>
        <element name="BirthDate" type ="date" />
        <element name="ReportingTime" type="time" default="10:00:00+05:30"/>
      </sequence>
      <attribute name="EmployeeNumber" type="integer" use="required" />
      <attribute name="Language" type="string" fixed="EN" />
      <attribute name="Nationality" type="string" default="Alien" />
    </complexType>
  </element>
</schema>
'
GO

DECLARE @emp AS XML(EmployeeSchema)
SET @emp = '
<Employee EmployeeNumber="1001" Language="EN" >
    <FullName>Jacob</FullName>
    <Salary>10000</Salary>
    <Age>30</Age>
    <Married>1</Married>
    <BirthDate>1975-03-14+05:30</BirthDate>
    <ReportingTime />
</Employee>
'
/*
Language should always be "EN" because a fixed value is specified in the SCHEMA.
If you enter any other language, you will get the following error.

Msg 6921, Level 16, State 2, Line 2
XML Validation: Element or attribute 'Language' was defined as fixed, the element value has to be equal
to value of 'fixed' attribute specified in definition. Location: /:Employee[1]/@:Language

"Nationality" can be ignored. If you ignore this attribute, the default value of "Alien" is assumed.
*/
```

## Conclusions

This session presented a few examples that shows how to do data type validations using an *XML Schema*. I will come up with a few more workshops which explains *XML Schemas* in much more detail.