http://www.sqlservercentral.com/articles/XML/69633/
Printed 2010/03/22 07:13PM

XML Workshop 25 - Inserting elements and attributes to an XML document

By Jacob Sebastian, 2010/02/17

Introduction

In the previous session we examined the "delete" command supported by the "modify()" method of XML data type. In this session we will learn how to insert an element or attribute into an XML document using the XML data type method: modify().

Sample XML Document

Here is the sample XML document that we will use for the examples in this session.

Example 1

```
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
```

Inserting a new element as the first child of a node

The following example inserts a new "Employee" element with value "Steve" as as the first child element of the root node.

Example 2

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert element Employee {"Steve"}
    as first
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

The above example created a new XML element using a fixed value specified by a string literal. The next example shows how to create an XML element with value specified in a variable.

Example 3

```
DECLARE @x xml
SET @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

DECLARE @name VARCHAR(20)
SELECT @name = 'Steve'

SET @x.modify('
    insert element Employee {sql:variable("@name")}
    as first
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

Both the examples given above, used the 'insert element' command to create a new element. Another way of achieving the same result is by supplying the whole XML fragment to be inserted as a string.

Example 4

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert <Employee>Steve</Employee>
    as first
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

If the text value of the element is stored in a variable, the following code will help.

Example 5

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

DECLARE @name VARCHAR(20)
SELECT @name = 'Steve'

SET @x.modify('
    insert <Employee>{sql:variable("@name")}</Employee>
    as first
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

It is possible to insert the attribute values along with the elements declarations.

Example 6

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert <Employee Team="SQL server">Steve</Employee>
    as first
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL server">Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

Even if both the text value of the element as well as the value of the attribute are stored in variables, you still dont have anything to worry. The following code shows how to handle it.

Example 7

```
DECLARE @x XML
SELECT @x = '
```

```
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

DECLARE @name VARCHAR(20), @team VARCHAR(20)
SELECT @name = 'Steve', @team = 'SQL Server'

SET @x.modify('
    insert
                <Employee Team="{sql:variable("@team")}">
                        {sql:variable("@name")}
                </Employee>
    as first
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

Not always you would want to insert the new element as the first child. You can insert the new element as the last child by specifying the "as last" command instead of "as first".

Example 8

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert <Employee Team="SQL Server">Steve</Employee>
    as last
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
  <Employee Team="SQL Server">Steve</Employee>
</Employees>
*/
```

If you do not include "as first" and "as last" in your XQuery expression, SQL Server will assume "as last". The following code produces the same output as the above example.

Example 9

```
DECLARE @x XML
SELECT @x = '
```

```
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert <Employee Team="SQL Server">Steve</Employee>
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
  <Employee Team="SQL Server">Steve</Employee>
</Employees>
*/
```

We saw how to insert an element as first or last within a parent node. It is also possible to insert the child element at a specified position. the following example inserts an element as the second child, using the "insert after" command.

Example 10

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert <Employee Team="SQL Server">Steve</Employee>
    after (Employees/Employee[1])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

You could also use "insert before" to insert the new element above a specified child. The following example shows how to insert a new element as the second last child of a parent node.

Example 11

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
'
```

```
SET @x.modify('
    insert <Employee Team="XML">Bob</Employee>
    before (Employees/Employee[position()=last()])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="XML">Bob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

The position can also be specified by a variable. The following code shows how to insert an element at a position specified by a variable.

Example 12

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
'

DECLARE @pos INT
SELECT @pos = 3  -- insert as the third child

SET @x.modify('
    insert <Employee Team="XML">Bob</Employee>
    before (Employees/Employee[position()=sql:variable("@pos")])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="XML">Bob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

The following example shows how to insert a new element before "steve".

Example 13

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
'
```

```
SET @x.modify('
    insert <Employee Team="XML">Bob</Employee>
    before (Employees/Employee[. = "Steve"])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="XML">Bob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

The following example demonstrates how to insert an element right after another element having a text value specified by a variable.

Example 14

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
'

DECLARE @name VARCHAR(20)
SELECT @name = 'Steve' -- insert after steve

SET @x.modify('
    insert <Employee Team="XML">Bob</Employee>
    after (Employees/Employee[. = sql:variable("@name")])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="XML">Bob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

The examples we examined so far inserted relatively simple elements to the XML document. You could even use the "insert" command to add more complex XML fragments having child elements and or attributes.

Example 15

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
```

```
    insert
                <SpecialTeam name="SQL Azure">
                        <Employee id="1001">Steve</Employee>
                        <Employee id="1002">Mike</Employee>
                </SpecialTeam>
    as first
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <SpecialTeam name="SQL Azure">
    <Employee id="1001">Steve</Employee>
    <Employee id="1002">Mike</Employee>
  </SpecialTeam>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

It is also possible to insert more than one element in a single operation. The "insert" command can accept a set (comma separated list) of XML elements and insert them at the specified position. Here is an example.

Example 16

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert  (
                <Employee Team="SQL server">Steve</Employee>,
                <Employee Team="SQL server">Mike</Employee>
                )
    into (/Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
  <Employee Team="SQL server">Steve</Employee>
  <Employee Team="SQL server">Mike</Employee>
</Employees>
*/
```

In the previous examples, we saw how to insert new XML elements into an XML document using string literals. The next example shows how to creae a new XML element with the results of an XQuery expression. The following example creates a new XML element named "SQLGuys" and adds all the "SQL Server" people into the group.

Example 17

```
DECLARE @x XML
```

```
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="SQL Server">Mike</Employee>
</Employees>
'

SET @x.modify('
    insert <SQLGuys>{
                (/Employees/Employee)
    }
    </SQLGuys>
    into (/Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee Team="SQL Server">Mike</Employee>
  <SQLGuys>
    <Employee Team="SQL Server">Jacob</Employee>
    <Employee Team="ASP.NET">Smith</Employee>
    <Employee Team="SQL Server">Steve</Employee>
    <Employee Team="SQL Server">Mike</Employee>
  </SQLGuys>
</Employees>
*/
```

Inserting Attributes

The syntax for adding attributes to an XML node is slightly different from that of elements. While working with attributes, remember that the position of an attribute is not significant in XML. An XML element can not have more than one attribute having the same name. Let us see a few examples that deal with inserting attributes into XML nodes.

The following example adds an attribute named "Team" to the XML node representing employee "Jacob".

Example 18

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee>Steve</Employee>
  <Employee>Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
'

SET @x.modify('
    insert attribute Team {"SQL Server"}
    as first
    into (Employees/Employee[. = "Jacob"])[1]
    ')
```

```
SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
*/
```

The value of the attribute can be specified by a variable as given in the following example.

Example 19

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee>Steve</Employee>
  <Employee>Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
'

DECLARE @team VARCHAR(20)
SELECT @team = 'SQL Server'
SET @x.modify('
    insert attribute Team {sql:variable("@team")}
    as first
    into (Employees/Employee[. = "Jacob"])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
*/
```

It is also possible to create a new attribute with the value returned by an XQuery expression. The following example creates a new attribute named "Friend" in the element that represents the employee "Jacob". Interestingly the name of Jacob's friend is "Steve".

Example 20

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee>Steve</Employee>
  <Employee>Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
'

SET @x.modify('
    insert attribute Friend {data(/Employees/Employee[1])}
    as first
    into (Employees/Employee[. = "Jacob"])[1]
    ')
```

```
SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Friend="Steve">Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
*/
```

Well, the attribute value can also be a list of values returned by an XQuery expression. Jacob can have more friends, can't he?

Example 21

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee>Steve</Employee>
  <Employee>Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
'

SET @x.modify('
    insert attribute Friends {data(/Employees/Employee[. != "Jacob"])}
    as first
    into (Employees/Employee[. = "Jacob"])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Friends="Steve Smith">Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
*/
```

Multiple attributes can be created with a single query, by specifying a set containing the required number of attributes.

Example 22

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee>Steve</Employee>
  <Employee >Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
'

SET @x.modify('
    insert (
                attribute Team {"SQL"},
                attribute Category {"MVP"}
        )
    as first
    into (Employees/Employee[. = "Jacob"])[1]
    ')
```

```
SELECT @x
/*
<Employees>
  <Employee>Steve</Employee>
  <Employee Team="SQL" Category="MVP">Jacob</Employee>
  <Employee>Smith</Employee>
</Employees>
*/
```

The following example shows how to insert new attributes into XML documents stored in an XML colum. What is interesting here is that the values of the attributes are taken from another table by doing a relational join. The code below uses the "sql:column()" function to create attributes with the value stored in a column.

Example 23

```
DECLARE @team TABLE (
        EmpID INT,
        Team VARCHAR(20)
)
DECLARE @emp TABLE (
        EmpID INT,
        Data XML
)

INSERT INTO @emp (EmpID, Data)
SELECT 1, '<Employee>Jacob</Employee>'
INSERT INTO @emp (EmpID, Data)
SELECT 2, '<Employee>Steve</Employee>'
/*
EmpID       Data
----------- ---------------------------
1           <Employee>Jacob</Employee>
2           <Employee>Steve</Employee>
*/

INSERT INTO @team (EmpID, Team) SELECT 1, 'SQL Server'
INSERT INTO @team (EmpID, Team) SELECT 2, 'SQL Azure'
/*
EmpID       Team
----------- --------------------
1           SQL Server
2           SQL Azure
*/

UPDATE e
SET Data.modify('
        insert attribute Team {sql:column("Team")}
        into (/Employee)[1]
')
FROM @emp e
INNER JOIN @team t ON e.EmpID = t.EmpID

SELECT * FROM @emp
/*
EmpID       Data
----------- ---------------------------------------------
1           <Employee Team="SQL Server">Jacob</Employee>
2           <Employee Team="SQL Azure">Steve</Employee>
*/
```

Note that attribute names should be unique within an element. If the attribute you are trying to add already exists, you will get an error which says "XML well-formedness check: Duplicate attribute 'name'. Rewrite your XQuery so it returns well-formed XML."

However, it is possible to check for the existence of an attribute and perform a conditional insert. The following code inserts a "Team" attribute to the employee element specified in a variable. The insert operation takes place only if the employee element does not have a "Team" attribute.

Example 24

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee>Steve</Employee>
  <Employee>Smith</Employee>
</Employees>'

DECLARE @name VARCHAR(20), @team VARCHAR(20)
SELECT @name = 'Steve', @team = 'SQL Server'

SET @x.modify('
    insert
                if (/Employees/Employee[. = sql:variable("@name")]/@Team)
                then
                        ()
                else
                        attribute Team {sql:variable("@team")}
    as first into (Employees/Employee[. = sql:variable("@name")])[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
  <Employee>Smith</Employee>
</Employees>
*/
```

More fun!

The following example inserts a comment node.

Example 25

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert <!-- List of employees -->
    as first into (Employees)[1]
    ')
```

```
SELECT @x
/*
<Employees>
  <!-- List of employees -->
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

The following example adds an XSLT processing instruction to an XML document.

Example 26

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert
                <?xml-stylesheet href="emp.xsl" type="text/xsl"?>
    as first into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <?xml-stylesheet href="emp.xsl" type="text/xsl"?>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

The following example shows how to insert a text node into an XML element.

Example 27

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>'

SET @x.modify('
    insert text {"Best Employees of 2009"}
    as first into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
   Best Employees of 2009
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="ASP.NET">Smith</Employee>
</Employees>
*/
```

Inserting XML data type values

SQL Server 2005 does not allow XML data type variables in the "insert" operation using modify() method. SQL Server 2008 enhanced the modify() method to support XML data type values in the "insert" operation. The following code (runs only on SQL Server 2008 or above) inserts an XML variable into an XML document.

Example 28

```
DECLARE @x XML
SELECT @x = '
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
</Employees>'

DECLARE @emp XML
SELECT @emp = '<Employee Team="SQL Server">Steve</Employee>'

SET @x.modify('
    insert sql:variable("@emp")
    into (Employees)[1]
    ')

SELECT @x
/*
<Employees>
  <Employee Team="SQL Server">Jacob</Employee>
  <Employee Team="SQL Server">Steve</Employee>
</Employees>
*/
```

Conclusions

In this article, I tried to cover almost all scenarios of "insert" operations that I could quickly think of. If you have some scenarios that are not listed here, do let me know and I will try to add them to the list.

Additional reading: XML Support in SQL Server

About the author

Jacob Sebastian is a SQL Server MVP, MSDN/Technet Moderator, Regional Mentor for PASS Asia, Author, Speaker and Trainer. See Jacob's Blog|Profile|XML Resources