

قسمت اول را در [این آدرس](#) می‌توانید مطالعه نمائید.

در ادامه قسمت اول، اگر بخواهیم نود جدیدی را به فیلد XML موجود اضافه کنیم، روش انجام آن به صورت زیر است (یکی از روش‌ها البته):

```
DECLARE @tblTest AS TABLE (xmlField XML)

INSERT INTO @tblTest
(
    xmlField
)
VALUES
(
    '<Sample>
      <Node1>Value1</Node1>
      <Node2>Value2</Node2>
      <Node3>OldValue</Node3>
    </Sample>'
)

DECLARE @Name NVARCHAR(50)
SELECT @Name = 'Vahid'

UPDATE @tblTest
SET    xmlField.modify(
    'insert element Node4 {sql:variable("@Name")} as last into
    (/Sample)[1]'
)

SELECT tt.xmlField
FROM   @tblTest tt
```

که حاصل آن (افزوده شدن یک المان جدید به نام Node4 بر اساس مقدار متغیر Name در انتهای لیست) به صورت زیر می‌باشد:

```
<Sample>
<Node1>Value1</Node1>
<Node2>Value2</Node2>
<Node3>OldValue</Node3>
<Node4>Vahid</Node4>
</Sample>
```

سؤال 1 :

اگر بخواهیم نام Node4 نیز متغیر باشد به چه صورتی باید مساله را حل کرد؟
در این حالت باید از کوئری‌های داینامیک استفاده کرد. باید یک رشته را ایجاد (کل عبارت update باید یک رشته شود) و سپس از دستور exec کمک گرفت و البته باید دقت داشت در این حالت کار encoding کارکترهای غیرمجاز در XML را باید خودمان انجام دهیم.

سؤال 2:

اگر بخواهیم نام نودها و مقادیر آن‌ها را به صورت یک جدول معمولی بازگشت دهیم به چه صورتی باید عمل کرد؟

```
DECLARE @XML AS XML
```

```
SELECT @XML = tt.xmlField
FROM   @tblTest tt

SELECT t.c.value('local-name(..)', 'varchar(max)') AS ParentNodeName,
       t.c.value('local-name(.)', 'varchar(max)') AS NodeName,
       t.c.value('text()[1]', 'varchar(max)') AS NodeText
FROM   @XML.nodes('/*/*') AS t(c)
```

که پس از اجرای آن خواهیم داشت:

ParentNodeName	NodeName	NodeText
Sample	Node1	Value1
Sample	Node2	Value2
Sample	Node3	OldValue
Sample	Node4	Vahid