

عنوان:	بررسی Select For XML
نویسنده:	وحید نصیری
تاریخ:	۱۳۹۲/۱۲/۰۵
آدرس:	www.dotnettips.info
گروه‌ها:	NoSQL, SQL Server, xml

تعدادی افزونه‌ی T-SQL، از نگارش‌های پیشین SQL Server، جهت تولید خروجی XML از یک بانک اطلاعاتی رابطه‌ای، به همراه آن بوده‌اند که در این قسمت آن‌ها را بررسی خواهیم کرد.

پیشنیاز بحث

در ادامه، از بانک اطلاعاتی معروف northwind برای تهیه کوئری‌ها استفاده خواهیم کرد. بنابراین فرض بر این است که این بانک اطلاعاتی را پیشتر به وهله‌ی جاری SQL Server خود افزوده‌اید.

بررسی FOR XML RAW

از نگارش 2005 به بعد، Select for XML علاوه بر خروجی متنی XML، توانایی تولید خروجی از نوع XML را نیز یافته است. در ادامه 4 حالت مختلف خروجی آن را بررسی خواهیم کرد.

```
SELECT Customers.CustomerID, Orders.OrderID
FROM Customers, Orders
WHERE Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerID
FOR XML RAW
```

خروجی For XML Raw فوق به نحو ذیل است:

```
<row CustomerID="ALFKI" OrderID="10643" />
<row CustomerID="ALFKI" OrderID="10692" />
```

Select for XML در اینجا به صورت خودکار، هر ردیف کوئری را تبدیل به یک المان row نموده و همچنین هر ستون کوئری را تبدیل به ویژگی‌های این المان (attributes) کرده‌است. همچنین باید دقت داشت که خروجی آن یک fragment است و دارای یک root element مشخص نیست.

برای تغییر حالت خروجی آن می‌توان از حالت ELEMENTS استفاده کرد:

```
SELECT Customers.CustomerID, Orders.OrderID
FROM Customers, Orders
WHERE Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerID
FOR XML RAW, ELEMENTS
```

اینبار مقادیر هر ردیف خروجی، بجای ظاهر شدن در ویژگی‌ها، به صورت یک المان نمایش داده می‌شود:

```
<row>
  <CustomerID>ALFKI</CustomerID>
  <OrderID>10643</OrderID>
</row>
```

حالت پیشرفته‌تر FOR XML RAW را در ادامه ملاحظه می‌کنید:

```
SELECT Customers.CustomerID,
Orders.OrderID
FROM Customers,
Orders
WHERE Customers.CustomerID = Orders.CustomerID
```

```
ORDER BY
Customers.CustomerID
FOR XML RAW('Customer'), ELEMENTS XSINIL, ROOT('Customers'), XMLSCHEMA('http://MyCustomers')
```

با استفاده از Root می‌توان Fragment حاصل را تبدیل به Document با یک Root element مشخص کرد. در قسمت Raw نیز می‌توان مقدار پیش فرض row را مقدار دهی کرد.

```
<Customers>
  <Customer xmlns="http://MyCustomers">
    <CustomerID>ALFKI</CustomerID>
    <OrderID>10643</OrderID>
  </Customer>
```

از XSINIL برای مشخص سازی المان‌های نال استفاده می‌شود. اگر XSINIL ذکر نشود، المان‌های نال در خروجی وجود نخواهند داشت.

ذکر XMLSCHEMA، سبب می‌شود تا SQL Server به صورت خودکار XML Schema را بر اساس اطلاعات ستون‌های رابطه‌ای مورد استفاده تولید کند.

این نکات را برای FOR XML AUTO نیز می‌توان بکار برد.

بررسی FOR XML AUTO

حالت دوم بکارگیری Select for XML به همراه عبارت Auto است:

```
SELECT Customers.CustomerID, Orders.OrderID
FROM Customers, Orders
WHERE Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerID
FOR XML AUTO, ELEMENTS
```

با خروجی ذیل:

```
<Customers>
  <CustomerID>ALFKI</CustomerID>
  <Orders>
    <OrderID>10643</OrderID>
  </Orders>
  <Orders>
    <OrderID>10692</OrderID>
  </Orders>
</Customers>
```

در اینجا ابتدا شماره مشتری و سپس اطلاعات تمام خریدهای او ذکر می‌شوند.

بررسی For XML Explicit

اگر بخواهیم خروجی را تبدیل به ترکیبی از المان‌ها و ویژگی‌ها کنیم، می‌توان از For XML Explicit استفاده کرد:

```
SELECT 1 AS Tag,
NULL AS Parent,
Customers.CustomerID AS [Customers!1!CustomerID],
NULL AS [Order!2!OrderId]
FROM Customers
UNION ALL
SELECT 2,
1,
Customers.CustomerID,
Orders.OrderID
FROM Customers,
Orders
WHERE Customers.CustomerID = Orders.CustomerID
```

```
ORDER BY
[Customers!1!CustomerID]
FOR XML EXPLICIT
```

با خروجی:

```
<Customers CustomerID="ALFKI">
  <Order OrderId="10643" />
  <Order OrderId="10692" />
  <Order OrderId="10702" />
  <Order OrderId="10835" />
  <Order OrderId="10952" />
  <Order OrderId="11011" />
</Customers>
```

برای استفاده از FOR XML EXPLICIT، باید به ازای هر سطح از سلسله مراتب مورد نظر، یک عبارت select را تهیه کرد که اینها نهایتاً باید با هم UNION ALL شوند. به علاوه دو ستون اضافی Tag و Parent نیز باید ذکر شوند. از این دو برای مشخص سازی سلسله مراتب استفاده می‌شوند. 1! سبب تولید یک ویژگی در سطح اول می‌شود و 2! سبب تولید ویژگی دیگری در سطح دوم.

بررسی FOR XML PATH

همانطور که مشاهده می‌کنید، نوشتن FOR XML EXPLICIT نسبتاً طولانی و پیچیده است. برای ساده سازی آن از نگارش 2005 به بعد، روش For XML Path معرفی شده است:

```
WITH XMLNAMESPACES('http://somens' AS au)
SELECT
  CustomerID AS [@au:CustomerID],
  CompanyName AS [Company/Name],
  ContactName AS [Contact/Name]
FROM Customers
FOR XML PATH('Customer')
```

با خروجی:

```
<Customer xmlns:au="http://somens" au:CustomerID="ALFKI">
  <Company>
    <Name>Alfreds Futterkiste</Name>
  </Company>
  <Contact>
    <Name>Maria Anders</Name>
  </Contact>
</Customer>
```

در اینجا با استفاده از WITH XMLNAMESPACES یک فضای نام جدید را تعریف کرده و سپس نحوه‌ی استفاده از آن را توسط یک Alias مشاهده می‌کنید. در اینجا همچنین توسط Aliasها می‌توان یک مسیر مشخص را نیز تعریف کرد. رشته‌ای که در قسمت Path مشخص می‌شود، بیانگر نام المان‌های خروجی است.

یک نکته: اگر کوئری FOR XML PATH را اجرا کنید، نام ستون خروجی به صورت خودکار به XML_F5..6 تنظیم می‌شود. علت اینجاست که در حالت پیش فرض، نوع خروجی این افزونه، استریم است و نه XML. برای تبدیل آن به نوع XML باید یک Type را اضافه کرد:

```
FOR XML PATH('Customer'), Type
```

در این حالت خروجی FOR XML PATH قابل انتساب به یک متغیر T-SQL از نوع XML خواهد بود.