

فرض کنید که می‌خواهیم خروجی از جدول خود را به صورت XML نمایش یا از طریق وب سرویس در برنامه مان استفاده نماییم. اولین راهی که به ذهنمان می‌رسد خودمان رشته xml را با حلقه ای ایجاد نماید یا استفاده از فضای نام System.Xml و کلاس‌های نوشته شده برای این کار. اما خود Sql Server امکانات ویژه ای برای کار با ساختار xml مهیا نموده که براحتی می‌توانید خروجی xml از داده هایتان ایجاد نمایید.

برای این کار از عبارت [For XML](#) در Select می‌توان استفاده نمود. برای مثال برای بدست آوردن ساختار ساده از For Xml Auto استفاده نمایید

```
SELECT BusinessEntityID, PersonType, Title, FirstName, MiddleName, LastName
FROM Person
WHERE BusinessEntityID = 10001
FOR XML AUTO
```

که خروجی بصورت **node attribute** زیر می‌باشد:

```
<Person.Person BusinessEntityID="10001" PersonType="IN" FirstName="Carolyn" LastName="Alonso" />
```

اما اگر بخواهیم خروجی به صورت node Elements باشد کفایت از پارامتر **Elements** استفاده نمایید

```
SELECT BusinessEntityID, PersonType, Title, FirstName, MiddleName, LastName
FROM Person
WHERE BusinessEntityID = 10001
FOR XML AUTO, ELEMENTS
```

خروجی بصورت زیر می‌باشد:

```
<Person.Person>
  <BusinessEntityID>10001</BusinessEntityID>
  <PersonType>IN</PersonType>
  <FirstName>Carolyn</FirstName>
  <LastName>Alonso</LastName>
</Person.Person>
```

اگر بخواهیم node attributes و node elements با هم ترکیب کنیم بصورت زیر عمل می‌کنیم:

```
SELECT BusinessEntityID AS '@ID', PersonType, Title, FirstName, MiddleName, LastName
FROM Person
WHERE BusinessEntityID = 10001
FOR XML ELEMENTS
```

خروجی بصورت زیر است:

```
<Person ID="10001">
  <PersonType>IN</PersonType>
  <FirstName>Carolyn</FirstName>
  <LastName>Alonso</LastName>
</Person>
```

حال می‌خواهیم همه nodeها را یک node ریشه قرار دهیم برای این کار از پارامتر ROOT در کنار AUTO به صورت زیر استفاده نماییم:

```
SELECT *
FROM Person
WHERE BusinessEntityID = 15291
FOR XML AUTO , ROOT('Persons')
```

اما اگر بخواهیم نام جدول را با نام دلخواه خود تغییر دهیم از پارامتر PATH به جای AUTO به صورت زیر استفاده نماییم:

```
SELECT *
FROM Person
WHERE BusinessEntityID = 15291
FOR XML PATH('P') , ROOT('Persons')
```

## نظرات خوانندگان

نویسنده: احسان میرسعیدی  
تاریخ: ۰۲۶ ۱۳۹۱/۰۹/۱۹

بسیار تکنیک کارامدی بود. واقعا متشکرم

نویسنده: فرهاد فرهمندخواه  
تاریخ: ۷:۲۹ ۱۳۹۱/۰۹/۱۹

سلام  
مرسی از مطلب مفید تان

نویسنده: بهراد  
تاریخ: ۱۱:۸ ۱۳۹۱/۰۹/۱۹

بسیار عالی بود ، خیلی برام مفید واقع شد  
راه مشابه ای برای خروجی Json نیست؟  
ممنون

نویسنده: مجتبی کاویانی  
تاریخ: ۱۱:۵۷ ۱۳۹۱/۰۹/۱۹

ممنون. هنوز به صورت native نه اما از تیم microsoft در این [لینک](#) خواسته شده که For Json را هم اضافه کند.

نویسنده: وحید نصیری  
تاریخ: ۱۱:۵۹ ۱۳۹۱/۰۹/۱۹

« [استفاده از JSON در SQL Server](#) »

نویسنده: مهدی ناظم السادات  
تاریخ: ۱۸:۵۸ ۱۳۹۲/۰۹/۱۷

حالا دستور Insert نداریم که بشه یه فایل xml رو مثل فایل backup روی دیتابیس restore کنیم؟

نویسنده: محسن خان  
تاریخ: ۰:۵۹ ۱۳۹۲/۰۹/۱۸

- می‌تونم با کدنویسی اینکار رو انجام بدم:

```
var reportData = new DataSet();
reportData.ReadXml("yourfile.xml");
var connection = new SqlConnection("DB ConnectionString");
var sbc = new SqlBulkCopy(connection);
sbc.DestinationTableName = "yourXMLTable";
```

- یا می‌تونم از [import و export](#) خود SQL Server استفاده کنی.

- و یا از [OPENXML](#) میشه استفاده کرد:

```
INSERT Customers
SELECT *
FROM OPENXML ...
```

نویسنده: ناصر نیازی  
تاریخ: ۱۳۹۲/۱۱/۲۹ ۲۱:۵۱

تشکر فراوان از این مطلب فوق العاده کاربردی. یه نکته کوچک به ذهنم رسید. سومین قطعه کدی که نوشتید در جدول من کار نکرد به نظرم اومد که شاید اینچنین بوده باشه :  
کد شما

FOR XML ELEMENTS

کدی که من مد نظرم هست و در جدول من کار می کنه :

for xml auto, ELEMENTS