

## ۷. فصل هفتم: جابجایی داده ها (Import-Export)

### ۷,۱. مقدمه

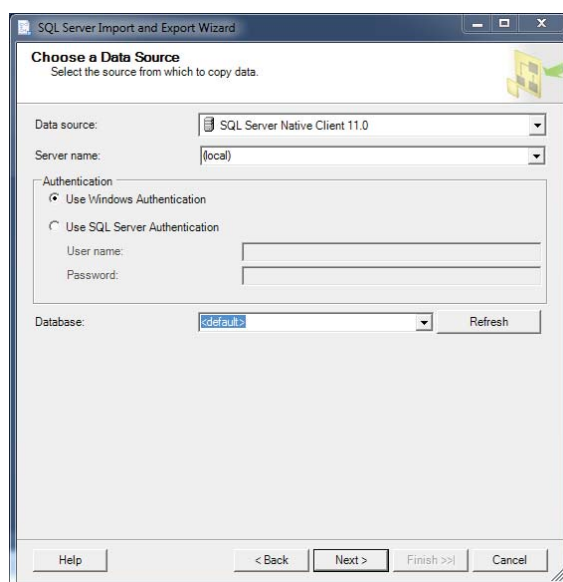
اهمیت بحث جابجایی داده‌ها قطعاً بر کسی پوشیده نیست. کافیت سناریوی ساده و بسیار متداولی را در نظر بگیرید که پایگاه داده‌ها روی یک server طراحی و پیاده‌سازی می‌شوند و روی یک server دیگر مورد استفاده قرار می‌گیرند. SQL Server راه‌های زیادی برای جابجایی داده‌ها در اختیار مدیران پایگاه داده‌ها قرار می‌دهد. یک مدیر پایگاه داده باید بر همه‌ی این راه‌ها تسلط کافی را داشته باشد تا بتواند در شرایط مختلف و با توجه به موارد مختلفی از قبیل حجم داده‌ای که باید جابجا شود، وجود یک لینک ارتباطی مورد اطمینان و سریع بین server ها و ... تصمیم درستی در رابطه با روش جابجایی داده‌ها بگیرد.

در این فصل به بررسی برخی از این روش‌های جابجا کردن داده‌ها می‌پردازیم.

### ۷,۲. روش‌های مختلف وارد کردن اطلاعات به SQL Server

#### ۷,۲,۱. استفاده از Import-Export Wizard

این امکان به شما اجازه جابجایی داده‌ها از منابع مختلف (Microsoft Excel, Microsoft Access, Flat File, ...) به SQL Server و بالعکس را می‌دهد. برای استفاده از این سرویس جهت ورود داده‌ها به SQL Server روی پایگاه داده موردنظر کلیک راست کنید و از بخش Tasks زیر منوی Import Data را انتخاب نمایید تا پنجره‌ای مطابق آنچه در شکل ۷-۱ دیده می‌شود، باز شود.



شکل ۷-۱

در قسمت Data Source منبعی را که حاوی داده‌های موردنظر است، انتخاب کنید و تنظیمات مربوطه را انجام دهید و next را بزنید. در مرحله جدید شما باید مقصد موردنظر برای داده‌ها را انتخاب کنید. پس از انتخاب مقصد و تنظیمات مربوط به آن، در مرحله بعد می‌توانید انتخاب کنید که داده‌ها از جداول یا view های منبع کپی شوند یا پرس و جویی طراحی شود که شما هم توانایی ویرایش آن را داشته باشید.

در مرحله بعدی جداول و view های قابل استفاده در مبدأ داده‌ها برای شما نشان داده شده که باید از میان آن‌ها منابع موردنظر را انتخاب کنید. همچنین با انتخاب گزینه Edit Mappings می‌توانید به تنظیمات بیشتری از جدول موردنظر دسترسی پیدا کنید (مثلاً نام ستون‌های جدول، قابلیت null بودن و ...). پس از این با تأیید تنظیمات مراحل قبل و پایان مراحل، داده‌ها از منبع به پایگاه داده موردنظر شما وارد خواهند شد.

## ۷.۲.۲. استفاده از دستور BULK INSERT

به کمک دستور BULK INSERT می‌توان داده را مستقیماً با اجرای این دستور T-SQL وارد SQL Server کرد. قالب کلی این دستور بصورت زیر است:

```
BULK INSERT
[ database_name . [ schema_name ] . [ schema_name . ] [ table_name | view_name ]
FROM 'data_file'
[ WITH
(
[ [ , ] BATCHSIZE = batch_size ]
[ [ , ] CHECK_CONSTRAINTS ]
[ [ , ] FIELDTERMINATOR = 'field_terminator' ]
[ [ , ] FIRSTROW = first_row ]
)]
```

در ادامه پارامترهای مهم این دستور را بررسی می‌کنیم:

در این دستور بعد از آوردن عبارت "BULK INSERT" باید مقصد موردنظر برای داده‌ها را انتخاب کرد و پس از انتخاب مقصد مناسب، باید منبع حاوی داده‌ها را تعریف کرد به این صورت که پس از کلمه‌ی کلیدی FROM باید آدرس کامل فایل موردنظر را قرار داد. و در آخر هم می‌توان با استفاده از کلمه‌ی کلیدی WITH برخی تنظیمات را برای نحوه‌ی وارد کردن اطلاعات، در نظر گرفت. مثلاً با تعریف BATCHSIZE داده‌ها را به چند بسته تقسیم می‌کنیم که هر بسته طی یک تراکنش به مقصد وارد می‌شود و امکان مدیریت در صورت بروز خطا را بالا می‌برد. (بصورت پیش‌فرض کل داده‌ها یک بسته در نظر گرفته می‌شوند) عبارت CHECK\_CONSTRAINTS اعمال شدن تمامی قیود جدول یا VIEW مقصد را بر روی داده‌ها، هنگام ورود به مقصد تضمین می‌کند. FIELDTERMINATOR جداکننده فیلدها در فایل‌های از نوع CHAR است که بصورت پیش‌فرض کاراکتر TAB (t) برای این منظور استفاده می‌شود. متغیر FIRSTROW بیانگر این است که انتقال داده‌ها در فایل مبدأ از کدام شماره سطر باید شروع شود.

برای کسب اطلاعات بیشتر در مورد دستور BULK INSERT به آدرس زیر مراجعه نمایید:

<http://msdn.microsoft.com/en-us/library/ms188365.aspx>

### ۷,۲,۳. استفاده از دستور OPENROWSET

OPENROWSET یک دستور T-SQL است که امکان پرس و جو گرفتن از منبع داده موردنظر را می دهد و وقتی این دستور را با عبارت INSERT INTO همراه می کنیم، می توانیم داده موردنظر را از منبع، مستقیماً وارد جدولی در SQL Server کنیم.

ساختار کلی این دستور بصورت زیر است:

```
OPENROWSET
( { 'provider_name' , { 'datasource' ; 'user_id' ; 'password'
  | 'provider_string' }
  , { [ catalog. ] [ schema. ] object
    | 'query'
  }

  | BULK 'data_file' ,
    { FORMATFILE = 'format_file_path' [ <bulk_options> ]
  }
) )
```

در ادامه به بررسی برخی پارامترهای مهم ساختار فوق می پردازیم:

- **provider\_name** یک رشته حاوی نام OLE DB provider موردنظر است. (لیست نام provider های قابل استفاده را می توانید در object explorer بباید: server objects را باز کنید، سپس linked servers و سپس providers)
- **datasource** یک رشته است که همانطور که از اسمش پیداست، مکان منبع داده موردنظر را باید مشخص کند که می تواند نام یک فایل، نام server یا هر رشته ای که provider به کمک آن بتواند مکان پایگاه داده های موردنظر را تشخیص دهد، باشد.
- **user\_id** و **password** اطلاعات امنیتی لازم برای برقراری ارتباط کاربر با datasource را مشخص می کنند.
- **Provider\_string** هم مشخص کننده provider و همچنین حاوی اطلاعات امنیتی لازم (مثل تعیین سطح دسترسی) است.
- **object [catalog.] [schema.]** هم نام جدول موردنظر را مشخص می کند که می تواند با یک query از آن جدول هم جایگزین شود.
- با استفاده از عبارت BULK می توانید با دستور OPENROWSET داده ها را از یک فایل بخوانید.

**توجه!** پیش از استفاده از این دستور در MS SQL Server، دستورات زیر را به همین ترتیب، اجرا کنید تا تنظیمات مربوطه در MS SQL Server فعال شوند.

```
sp_configure 'show advanced options', 1;  
RECONFIGURE;  
Go  
sp_configure 'Ad Hoc Distributed Queries', 1;  
RECONFIGURE;  
GO  
exec sp_configure 'Advanced', 1 RECONFIGURE  
exec sp_configure 'Ad Hoc Distributed Queries', 1  
RECONFIGURE  
EXEC master.dbo.sp_MSset_oledb_prop N'Microsoft.ACE.OLEDB.12.0',  
N'AllowInProcess', 1  
EXEC master.dbo.sp_MSset_oledb_prop N'Microsoft.ACE.OLEDB.12.0',  
N'DynamicParameters', 1  
GO
```

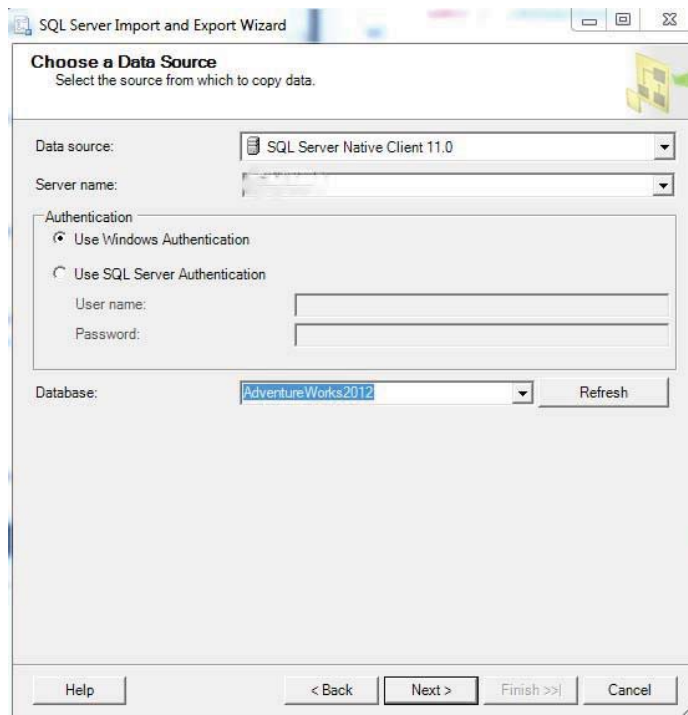
برای کسب اطلاعات بیشتر در مورد دستور OPENROWSET به آدرس زیر مراجعه نمایید:

<http://msdn.microsoft.com/en-us/library/ms190312.aspx>

## ۷,۳. روش‌های مختلف خارج کردن اطلاعات از SQL Server

### ۷,۳,۱. استفاده از Import-Export Wizard

همانطور که در بخش ۱-۲-۵ مشاهده کردید از این wizard می‌توان برای جابجایی داده‌ها به SQL Server و یا از SQL Server به سایر فرمت‌ها استفاده کرد. در این بخش نحوه استفاده از این امکان برای خارج کردن داده‌ها از SQL Server را بررسی می‌کنیم.



شکل ۷-۲

ابتدا روی پایگاه داده موردنظر کلیک راست کنید و از بخش Tasks زیر منوی Export Data را انتخاب نمایید تا پنجره‌ی زیر باز شود:

همانطور که مشاهده می‌کنید این بار منبع داده در گام اول مشخص است و فقط نیاز به تأیید دارد. همچنین می‌توانید در قسمت "Authentication" نحوه‌ی اتصال به منبع را هم مشخص کنید. در مرحله بعد، مقصد داده و تنظیمات مرتبط با آن را باید مشخص کنید که می‌تواند فایل متنی، Microsoft Excel، Microsoft Access یا سایر گزینه‌های موجود باشد.

در مراحل بعدی می‌توانید جداولی را که قصد خروج آن‌ها از SQL Server به مقصد تعیین شده را دارید انتخاب می‌نمایید. دقت کنید این خارج کردن داده‌ها به معنی پاک شدن آن‌ها از SQL Server نیست و یک کپی از داده‌ها به فرمت انتخاب شده از SQL Server خارج می‌شود. پس از طی مراحل wizard داده‌ها با فرمت انتخاب شده و در فایل‌ی که انتخاب کرده‌اید، قرار خواهند گرفت.

## ۷.۳.۲. استفاده از xp\_cmdshell

این دستور که ساختاری بسیار ساده دارد، یک رشته را به عنوان پارامتر می‌گیرد و آن را برای اجرا به Command Shell سیستم عامل ارسال می‌کند. ساختار این دستور بصورت زیر است:

```
xp_cmdshell { 'command_string' }
```

در این قسمت می‌خواهیم به کمک این دستور، یک کپی از داده‌های یک جدول را به یک فایل متنی منتقل کنیم. در ابتدا توجه کنید که معمولاً امکان استفاده از xp\_cmdshell برای server بصورت پیش فرض، فعال نیست. با استفاده از دستورات زیر ابتدا این امکان را برای server فعال می‌کنیم:

```
-- To enable the feature.  
EXEC sp_configure 'xp_cmdshell', 1  
GO  
-- To update the currently configured value for this feature.  
RECONFIGURE  
GO
```

با اجرای دستور زیر می‌توان نتیجه‌ی پرس و جوی نوشته شده ("QUERY") را در فایل متنی مشخص شده ("OUTPUT\_FILE") وارد کرد.

```
EXEC xp_cmdshell 'bcp "Query" queryout "OUTPUT_FILE" -T -c -t, '
```

همانطور که ملاحظه می‌کنید در دستور فوق از bcp استفاده شده، در ادامه کمی بیشتر به این دستور خواهیم پرداخت.

## ۷.۴ Bulk Copy Program (BCP)

به کمک bcp می‌توان حجم بزرگی از داده‌ها را بین SQL Server و یک فایل با ساختار مشخص شده توسط کاربر جابجا کرد. این دستور در windows command prompt اجرا می‌شود و ساختار کلی آن بصورت زیر است:

```
bcp [database_name.] schema.{  
table_name | view_name | "query"  
{ in data_file | out data_file | queryout data_file | format nul}  
}
```

در این ساختار:

- Data\_file آدرس کامل فایل موردنظر برای bcp است. (اگر داده‌هایی را می‌خواهیم به SQL Server

وارد کنیم، data\_file نام فایل مبدأ (فایل حاوی داده‌ها) است و اگر داده‌هایی را می‌خواهیم از SQL

Server به بیرون کپی کنیم، data\_file نام فایل مقصد است.)

- در خط سوم این ساختار جهت کپی داده‌ها بصورت زیر مشخص می‌شود:

in: از یک فایل مبدأ داده‌ها را در SQL Server کپی می‌کند.

out: داده‌ها را از SQL Server به یک فایل مشخص شده کپی می‌کند. اگر از فایلی که قبلاً وجود داشته

استفاده می‌کنید، توجه کنید که فایل overwrite می‌شود.

queryout: وقتی از یک query می‌خواهید داده‌ها را کپی کنید باید از این عبارت استفاده کنید.

format: با استفاده از این دستور، و به کمک option های موجود یک فایل ساختاری می‌سازد که در اینجا

مورد بحث ما نیست.

اطلاعات بیشتر درباره‌ی این دستور را در آدرس زیر بخوانید:

<http://msdn.microsoft.com/en-us/library/ms162802.aspx>

## ۷.۵. کار با داده‌های XML

XML یک زبان توصیف داده است که به دلیل کاربرد بالایش در مباحث مربوط به وب، بسیار محبوب شده است. مبحث XML مبحث بسیار وسیع و گسترده‌ای است که خارج از بحث کاری آزمایشگاه پایگاه داده است. در ادامه بطور اجمالی با برخی دستورات برای کار با داده‌های XML در SQL Server آشنا می‌شویم.

در صورتی که با XML آشنایی ندارید می‌توانید از این لینک استفاده کنید:

<http://www.w3schools.com/xml>

برای استفاده از داده‌های XML در SQL Server می‌توان از نوع داده‌ی xml برای تعریف شمای موردنظر، استفاده کرد. به عنوان مثال در AdventureWorks2012 جدول Product Model دارای دو فیلد با نوع داده‌ی xml است. به لینک زیر رجوع کنید:

[http://technet.microsoft.com/en-us/library/ms124873\(v=sql.100\).aspx](http://technet.microsoft.com/en-us/library/ms124873(v=sql.100).aspx)

برای طراحی پرس و جو در xml معمولاً از xquery استفاده می‌شود. ساختار کلی xquery بصورت زیر است:

```
For ...  
Let ...  
Where ...  
Order by ...  
Return ...
```

در این ساختار for تقریباً معادل from و return معادل select در پرس‌وجوهای معمولی است. و از let هم برای کمک گرفتن از متغیرهای موقت استفاده می‌شود. برای آشنایی بیشتر با xquery به آدرس زیر مراجعه کنید:

<http://www.w3schools.com/xquery/default.asp>

برای آشنایی بیشتر با برخی مثال‌های کاربردی xquery به آدرس زیر مراجعه کنید:

<http://technet.microsoft.com/en-us/library/ms186960.aspx>

در ادامه به کمک دستورهای که پیشتر در این فصل آموزش داده شد، مثالی برای جابجایی داده‌های xml در SQL Server را بررسی می‌کنیم:

در این مثال قصد داریم جدولی شامل فیلدهای XML را به فایل‌ی متنی منتقل کنیم.

برای این کار ابتدا یک جدول با سه فیلد بصورت زیر می‌سازیم:

```
CREATE TABLE xmlTable(  
    Name [nvarchar](250) NULL,  
    CatalogDescription [xml] NULL,  
    Instructions [xml] NULL  
)
```

سپس با xquery برای جدول ProductModel از پایگاه داده AdventureWorks2012 پرس و جویی طراحی می‌کنیم که نام کالاهایی را که با M شروع می‌شوند، همراه با دو فیلد features و specifications آنها بصورت xml برگرداند و نتیجه را وارد جدول xmltable می‌کنیم:

```
insert into xmlTable SELECT Name , CatalogDescription.query('
declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelDescription";
for $P in /ProductDescription
return
<Specifications>
{ $P/Specifications }
</Specifications>
') as Specifications , CatalogDescription.query('
declare default element namespace
"http://schemas.microsoft.com/sqlserver/2004/07/adventure-
works/ProductModelDescription";
for $P in /ProductDescription
return
<Features>
{ $P/Features }
</Features>
') as Features
FROM AdventureWorks2012.Production.ProductModel
where Name like 'M%'
```

سپس با اجرای یک دستور bcp بصورت زیر این داده‌ها را وارد یک فایل متنی می‌کنیم:

```
bcp <test_database_name>.xmlTable out xmltest.txt -T -c
```

برای مشاهده‌ی مثال‌های دیگری از جابجایی داده‌های xml در SQL Server به آدرس زیر مراجعه کنید:

<http://msdn.microsoft.com/en-us/library/ms191184.aspx>