

چگونگی ایجاد یک روش ذخیره شده ی SQL Server با پارامترها

قدرت واقعی روش های ذخیره شده توانایی انتقال پارامترها و واداشتن روش ذخیره شده به بررسی درخواست های متفاوتی که انجام می شود، می باشد. در این قسمت انتقال مقادیر پارامتر را به یک روش ذخیره شده بررسی می کنیم.

توضیحات:

همانطور که می توانید پارامترها را با کد SQL خود استفاده کنید، همچنین می توانید روش های ذخیره شده ی خود را طوری تنظیم کنید که یک یا دو مقدار پارامتر را بپذیرد.

یک پارامتر:

در این مثال جدول **Person.Address** را از دیتابیس **AdventureWorks** تنظیم می کنیم، اما به جای بازگشت همه ی رکوردها، آن را تنها به یک شهر خاص محدود می کنیم. در این مثال فرض بر این است که روی مقدار **City** که انتقال داده می شود، یک هماهنگی دقیق وجود دارد.

```
CREATE PROCEDURE uspGetAddress @City nvarchar(30)
AS
SELECT *
FROM AdventureWorks.Person.Address
WHERE City = @City
GO
```

برای فراخوانی این برنامه ی ذخیره شده آن را مانند زیر اجرا می کنیم:

```
EXEC uspGetAddress @City = 'New York'
```

می توانیم همین کار را انجام دهیم اما به یوزرها اجازه دهیم تا برای جستجوی داده یک نقطه ی شروع ارائه می دهید. در اینجا می توانیم "=" را به یک **LIKE** تبدیل کنیم و از "%" استفاده کنیم.

```
CREATE PROCEDURE uspGetAddress @City nvarchar(30)
AS
SELECT *
FROM AdventureWorks.Person.Address
WHERE City LIKE @City + '%'
GO
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

در هر دو مثال قبل فرض بر این است که مقدار هر پارامتر همیشه منتقل خواهد شد. اگر سعی کنید برنامه را بدون انتقال یک پارامتر اجرا کنید، یک پیغام خطا مانند پیغام زیر دریافت خواهید کرد:

Msg 201, Level 16, State 4, Procedure uspGetAddress, Line 0

Procedure or function 'uspGetAddress' expects parameter '@City', which was not supplied.

مقادیر پارامتر پیش فرض

در بسیاری موارد انتقال همه ی مقادیر پارامتر تمرین خوبی است، اما گاهی اوقات این کار غیرممکن است. بنابراین در این مورد از گزینه ی **NULL** استفاده می کنیم تا به شما اجازه دهیم مقدار یک پارامتر را انتقال ندهید. اگر این روش ذخیره شده را ایجاد کرده و اجرا کنیم، هیچ داده ای را باز نخواهد گرداند، زیرا در جستجوی مقادیر **City** است که برابر **NULL** می باشد.

```
CREATE PROCEDURE uspGetAddress @City nvarchar(30) = NULL
AS
SELECT *
FROM AdventureWorks.Person.Address
WHERE City = @City
GO
```

ما می توانیم این روش ذخیره شده را تغییر داده و از تابع **ISNULL** استفاده کنیم تا به این مرحله برسیم. بنابراین اگر یک مقدار انتقال داده شود، این مقدار برای محدود کردن مجموعه ی نتیجه استفاده خواهد شد و اگر یک مقدار انتقال داده نشود، همه ی رکوردها را گزارش خواهد داد. (نکته: اگر ستون **City** دارای مقادیر **NULL** باشد، چنین مقادیری را در خود ندارد. شما مجبور خواهید بود منطق های دیگری مانند **IS NULL City** را به آن اضافه کنید.)

```
CREATE PROCEDURE uspGetAddress @City nvarchar(30) = NULL
AS
SELECT *
FROM AdventureWorks.Person.Address
WHERE City = ISNULL(@City, City)
GO
```

پارامترهای چندگانه:

برقراری پارامترهای چندگانه بسیار آسان می باشد. فقط کافیست هر پارامتر را لیست کرده و نوع داده به وسیله ی یک کاما مجزا می شود، مانند زیر:

```
CREATE PROCEDURE uspGetAddress @City nvarchar(30) = NULL, @AddressLine1 nvarchar(60) =  
NULL  
AS  
SELECT *  
FROM AdventureWorks.Person.Address  
WHERE City = ISNULL(@City, City)  
AND AddressLine1 LIKE '%' + ISNULL(@AddressLine1, AddressLine1) + %'  
GO
```

برای اجرای این، هرکدام از کارهای زیر را می توانید انجام دهید:

```
EXEC uspGetAddress @City = 'Calgary'  
--or  
EXEC uspGetAddress @City = 'Calgary', @AddressLine1 = 'A'  
--or  
EXEC uspGetAddress @AddressLine1 = 'Acardia'  
-- etc...
```

آدرس آموزشگاه : تهران - خیابان شریعتی - بالا تر از خیابان ملک - جنب بانک صادرات - پلاک 651 طبقه دوم - واحد 7

88146323 - 88446780 - 88146330

<http://www.tahlildadeh.com/>