

حذف فراخوانی عملکردها (Function Calls) از لیست SELECT

استفاده از عملکردها به عنوان نوعی از برنامه نویسی معمولاً تمرین خوبی است، چرا که به طور کل کد را خواناتر کرده و به شما اجازه می دهد که بارها بارها از آن استفاده کنید. این مسئله در مورد **query** های **SQL** هم درست میباشد، به جز در این مورد که مواردی وجود دارند که در هنگام اجراهای پشت سر هم یک وضعیت، ممکن است راه موثری برای رسیدن به نتیجه نباشد.

توضیحات

اجازه بدهید به سطح بالای اتفاقاتی که در هنگام استفاده از یک عملکرد در لیست **SELECT** از یک **query** می افتد، نگاهی داشته باشیم. اساساً لازم است این عملکرد برای هر رکورد گزارش شده ای توسط **query**، فراخوانده شود. اگر این عملکرد حاوی اتصال یک جدول چندگانه برای انجام نوعی **lookup** باشد، این فرایند می تواند بسیار پرهزینه باشد. در برخی موارد بهتر است فراخوانی این جدول را حذف کرده و آن را به سادگی مستقیماً به دیگر جدول ها در **query** خود متصل کنید. برای توضیح این نکته اجازه بدهید که یک **query** ساده را بگیریم که یک **lookup** روی یک جدول دیگر انجام می دهد. در زیر چند **SQL** را می بینید که قبل از تست به اجرای آنها نیاز خواهیم داشت تا عملکرد خود را اجرا کرده و یک ایندکس به ستون **lookup** اضافه کنیم.

```
CREATE FUNCTION fn_getParentDate (@ParentID bigint) RETURNS datetime AS
BEGIN
DECLARE @DateData datetime
SELECT @DateData = DateDataColumn from [dbo].[Parent] where ParentID=@ParentID
RETURN @DateData
END
GO
```

```
CREATE NONCLUSTERED INDEX idxChild_ParentID
ON [dbo].[Child] ([ParentID])
```

```
-- cleanup statements
--DROP INDEX Child.idxChild_ParentID
--DROP FUNCTION fn_getParentDate
```

اکنون می توانیم یک **query** ساده بنویسیم که این عملکرد را فرا می خواند و یک **lookup** روی جدول اصلی انجام می دهد. در اینجا وضعیت را مشاهده می کنید:

```
SELECT dbo.fn_getParentDate(ParentID),ChildID
```

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

88146323 - 88446780 - 88146330

FROM [dbo].[Child]

با نگاه به **explain plan** برای این **query** مشاهده می کنیم که قرار است یک اسکن بر روی جدول **Child** انجام شود که معنی دار می شود، زیرا عبارت **WHERE** وجود ندارد و برای هر ردیف گزارش شده از ایندکس روی جدول اصلی استفاده می کند تا یک جستجو برای **lookup** انجام دهد.

اکنون اجازه بدهید این **query** را بازنویسی کرده و به جای استفاده از فراخوانی عملکرد، جدول اصلی را به **query** خود متصل کرده و **DateDataColumn** را به لیست **SELECT** خود اضافه کنیم. در اینجا وضعیت را مشاهده می کنید:

```
SELECT P.DateDataColumn,ChildID
FROM [dbo].[Parent] P INNER JOIN
[dbo].[Child] C ON P.ParentID=C.ParentID
```

با مشاهده ی **explain plan** برای این **query** متوجه می شویم که تنها باید به جدول اصلی دسترسی داشته باشد، اما اکنون باید یک اسکن از این جدول، قبل از اجرای اتصال، انجام دهد.

فقط نگاه کردن به طرح های توضیحی بالا مشخص نمی کند که وضعیت بهتر انجام می شود یا نه. جستجوی ایندکس در **query** با عملکرد ممکن است باعث شود به این نتیجه برسید که اینطور سریعتر است اما اجازه بدهید وضعیت ها را اجرا کرده و نگاهی به نتایج **SQL Profiler** در زیر ببندازیم. از این نتایج مشاهده می کنیم که در واقع **query** بدون عملکرد بیشتر از دو برابر سریعتر اجرا شد و از منابع بسیار کمتری استفاده کرد نسبت به آن **query** که با فراخوانی عملکرد اجرا شد.

	CPU	Reads	Writes	Duration
Function	14985	5705126	0	25982
No Function	578	5933	0	11964

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

88146323 - 88446780 - 88146330