

استفاده از یک جدول مشتق شده (derived table) به جای IN predicate با عملکردهای aggregate

استفاده از یک جدول مشتق شده به جای **IN predicate** زمانیکه در حال گردآوری داده هستیم، به ما اجازه می دهد تا فقط رکوردهای خاصی از جدول را پردازش کنیم، بنابراین کاهش منابع نیاز به اجرای یک **query** دارد.

توضیحات

وقتی که از **IN** استفاده می کنیم، ابتدا باید **query** را در زیرمجموعه ی خود پردازش کرده و سپس داده های مشابه دیگری را مجدداً در **query** اصلی خود (بسته به عبارت **WHERE**) پردازش کنیم. اگر برای بیشترین قسمت کار می توانیم از جدول مشتق شده استفاده کنیم، می توانیم از پردازش دوباره ی داده اجتناب کنیم. قبل از اینکه به مثالی برای توضیح این نکته پردازیم، لازم است به جدول اصلی خود یک ایندکس اضافه کنیم، بنابراین نتایج با اسکن کردن جدول تغییر نمی کنند. در اینجا کد ایجاد این ایندکس را مشاهده می کنید:

```
CREATE NONCLUSTERED INDEX idxParentID_IntDataColumnParentID  
ON [dbo].[Parent] ([IntDataColumn],[ParentID])
```

```
-- cleanup statements
```

```
DROP INDEX Parent.idxParentID_IntDataColumnParentID
```

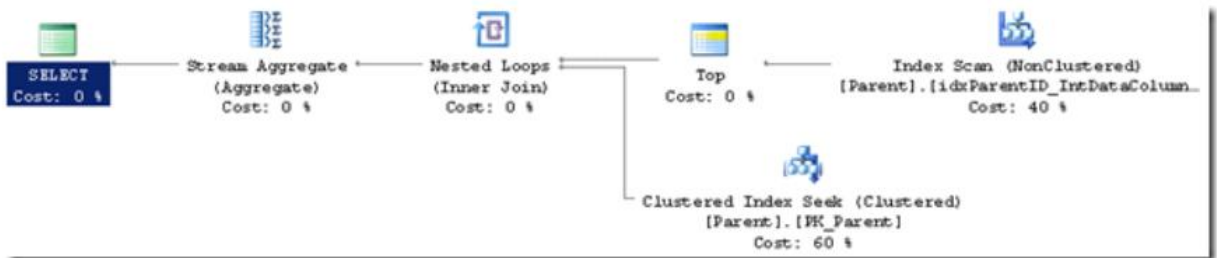
اجازه بدهید به یک **query** که از **IN** برای گزارش دومین مقدار بزرگ یک جدول استفاده می کند، نگاهی داشته باشیم. یک راه برای انجام این کار مانند زیر می باشد:

```
SELECT MIN(IntDataColumn)  
FROM [dbo].[Parent]  
WHERE ParentID IN (SELECT TOP 2 ParentID  
FROM [dbo].[Parent]  
ORDER BY IntDataColumn DESC)
```

تنها با نگاه به **query** متوجه می شویم که قرار است دوباره به جدول اصلی دسترسی داشته باشیم تا این نتیجه را دریافت کنیم. در **explain plan** مشاهده می کنیم که دومین دستیابی از جستجوی ایندکس (**index seek**) استفاده کرده، بنابراین ممکن است موضوع بزرگی نباشد.

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

88146323 - 88446780 - 88146330



اکنون اجازه بدهید که **query** را بازنویسی کرده و از یک جدول مشتق شده برای تولید نتیجه استفاده می کند. در اینجا وضعیت **SQL** را مشاهده می کنید:

```
SELECT MIN(IntDataColumn)
FROM (SELECT TOP 2 IntDataColumn
FROM [dbo].[Parent]
ORDER BY IntDataColumn DESC) AS A
```

توجه داشته باشید که از جدول اصلی تنها یک بار به عنوان مرجع استفاده کردیم و **explain plan** نیز تایید می کند که ما برای بار دوم نیاز به دسترسی به جدول اصلی نخواهیم داشت، حتی با یک ایندکس.



همچنین از نتایج **SQL Profiler** در زیر مشاهده می کنیم که صرفه جویی قابل توجه در منابع داشته ایم، حتی در مورد این **query**. گرچه **CPU** و مدت زمان کل یکی هستند، ما مجبوریم فقط دو تا از خواندن ها را انجام دهیم که با **query** اصلی در مقابل 8 قرار می گیرد.

	CPU	Reads	Writes	Duration
IN Predicate	0	8	0	0
Derived Table	0	2	0	0

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/>