

اجتناب از استفاده از کاراکترهای Wildcraft برای شروع معیارهای جستجو

هنگامی که از اپراتور **LIKE** استفاده می کنید و اولین کاراکتر در رشته ی جستجو یک کاراکتر **wildcraft** باشد، % یا **SQL** **Optimizer** مجبور خواهند بود که اسکن یک جدول یا ایندکس را در هنگام اجرای **query** انجام دهند.

توضیحات

قبل از اینکه وارد جزئیات مربوط به توضیح خود شویم، اجازه بدهید یک ایندکس روی ستونی ایجاد کنیم که قصد داریم از آن در عبارت **WHERE** مربوط به **query** خود استفاده کنیم. در اینجا کدی را مشاهده می کنید که برای ایجاد ایندکس مورد نظر روی جدول **Child** استفاده می شود.

```
CREATE NONCLUSTERED INDEX idxChild_VarcharDataColumn
```

```
ON [dbo].[Child] ([VarcharDataColumn])
```

```
-- cleanup statements
```

```
--DROP INDEX Child.idxChild_VarcharDataColumn
```

بنابراین چرا باید اسکن جدول/ایندکس انجام شود؟ از آنجایی که همه ی ایندکس های **SQL Server** در یک **B-Tree structure** ذخیره شده اند، وقتی معیارهای جستجوی خود را با یک کاراکتر **Wildcraft** آغاز می کنیم، **optimizer** قادر به استفاده از ایندکس برای اجرای جستجو برای یافتن سریع داده نمی باشد. بنابراین اگر همه ی ستون های لازم برای **query** بخشی از ایندکس باشند، یا اسکن از یک جدول را اجرا می کند و یا اسکن از یک ایندکس را. اکنون متوجه شدیم که برخی موارد وجود دارند که در آنها براساس نیازهایتان این امر ممکن نیست، اما مثال زیر نشان می دهد که چرا باید هرچا ممکن است از انجام چنین کاری اجتناب شود. اجازه بدهید یک **query** ساده بنویسیم که یک جستجو روی ستونی که در بالا ایندکس کردیم، انجام می دهد. در اینجا کد وضعیت **SQL** را مشاهده می کنید.

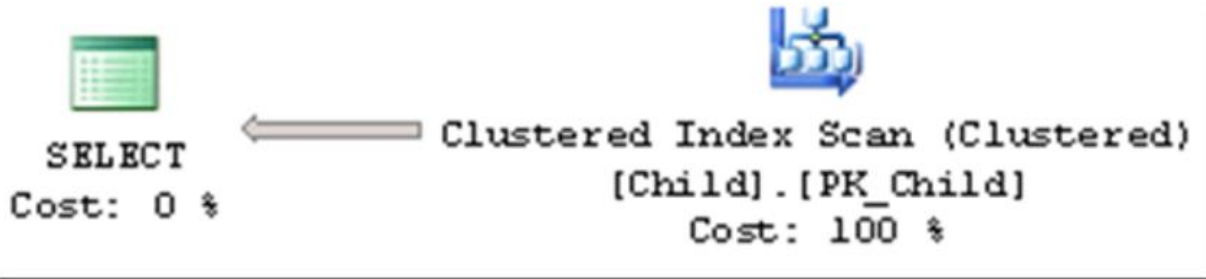
```
SELECT * FROM [dbo].[Child]
```

```
WHERE VarcharDataColumn LIKE '%EST5804%'
```

با نگاه کردن به **explain plan** برای این **query** متوجه می شویم که ایندکس روی **VarcharDataColumn** که ایجاد کرده ایم، نادیده گرفته شده و یک **clustered index** (لزوما اسکن از جدول) باید اجرا شود.

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

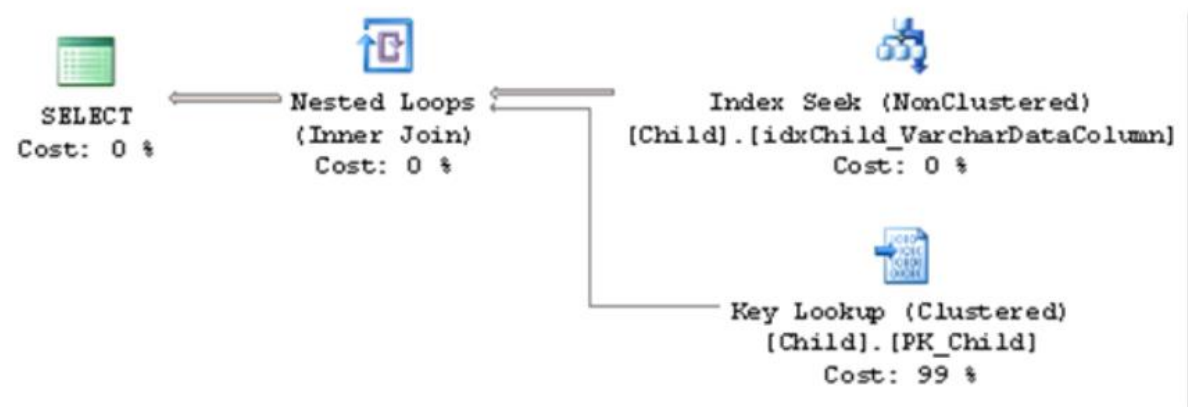
88146323 - 88446780 - 88146330



اکنون اجازه بدهید رشته ی بدهید رشته ی جستجو را در این **query** تغییر دهیم تا **wildcard** را حذف کنیم، بنابراین رشته ای که در حال جستجوی آن هستید با یک کاراکتر معتبر آغاز می شود. در اینجا وضعیت آپدیت شده ی **SQL** را مشاهده می کنید. نکته: من معیار جستجو را طوری انتخاب کردم که هر دو **query** یک نتیجه را ارائه بدهند و نتایج به وسیله ی یک **query** تغییر نمی یابند و یک مجموعه ی نتیجه ی بزرگتر را گزارش می دهد.

```
SELECT * FROM [dbo].[Child]
WHERE VarcharDataColumn LIKE 'TEST5804%'
```

با نگاه کردن به **explain plan** برای این **query** متوجه می شویم که **optimizer** در حال استفاده از ایندکسی می باشد که ایجاد کردیم و یک جستجو به جای یک اسکن انجام می دهد:



گرچه فقط با مقایسه ی **explain plans** باید قادر به تشخیص اجرای بهتر **query** دوم باشیم، اجازه بدهید تایید کنیم که در واقع این **query** از منابع کمتری استفاده می کند و اجرای آن از **query** آغازین سریعتر می باشد. در زیر مشاهده می کنیم که با حذف کاراکتر **wildcard** از ابتدای **query**، در واقع یک پیشرفت خیلی بزرگ مشاهده می کنیم:

	CPU	Reads	Writes	Duration
Wildcard at Start	328	7042	0	404
No Wildcard at Start	0	670	0	64



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/>