عنوان مقاله: نحوه صحیح ایجاد Nonclustered Index

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: تیرماه ۱۴۰۱

منبع: https://nikamooz.com/nonclustered-index

هنگامی که در شرط کوئری شما دو ستون وجود دارد و عملگر بین آن‌ها And هست، کدام ستون را در کلید Index اول قرار دهید، کوئری با Performance بهتری اجرا خواهد شد؟ لازم به ذکر است که، حالت‌های متعددی وجود دارد و ما در این مقاله یک نمونه جالب را بررسی خواهیم نمود.

**کوئری زیر را در نظر بگیرید:**

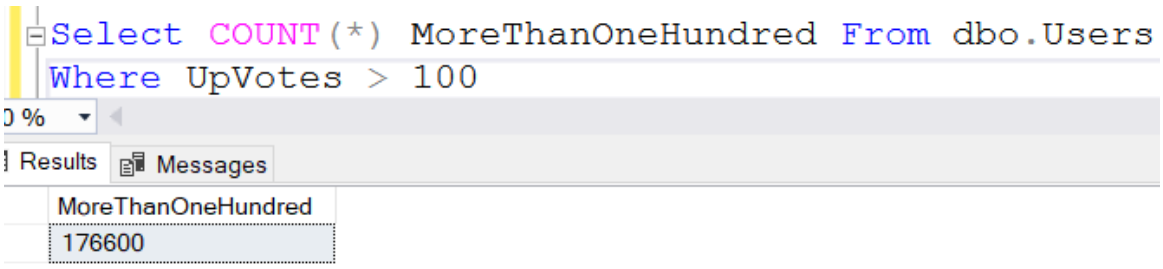
```
Select ID, Location, UpVotes, DownVotes, LastAccessDate
From dbo.Users
Where UpVotes > 100 And DownVotes = 0
```

این کوئری رکوردهایی را نمایش می دهد که UpVotes آن ها بزرگتر از ۱۰۰ و DownVotes آن‌ها برابر با صفر است. در اینجا هدف ما نمایش دادن اطلاعات کامل افراد نیست بنابراین فقط پنج ستون را در Select List قرار داده‌ایم. می‌خواهیم یک Nonclustered Index ایجاد نمائیم که سرعت اجرای کوئری مورد بحث را افزایش دهد.

**کدام ستون را در کلید ایندکس اول قرار دهیم کوئری Performance بهتری خواهد داشت؟**

**نکته:** جدول Users در StackOverFlow Database دارای یک Clustered Index بر روی ستون ID است.

راه حلی که در نگاه اول به ذهن می‌رسد این است که، تعداد رکوردهایی که UpVotes آن‌ها بیشتر از ۱۰۰ است چند تاست؟ تصویر زیر نشان می‌دهد که تعداد این رکوردها برابر با ۱۷۶۶۰۰ است.



همچنین تصویر زیر تعداد رکوردهایی که DownVotes آن‌ها برابر با صفر است را نمایش می‌دهد:

```
Select COUNT(*) ZeroDownVotes From dbo.Users
Where DownVotes = 0
```

Results | Messages

| ZeroDownVotes |
|---|
| 8609732 |

همان طور که در تصویر مشاهده می‌نمائید DownVotes بیش از هشت میلیون و شش صد هزار نفر برابر با صفر است.

خب، به نظر می‌رسد بهتر است که SQL Server ابتدا صد و هفتاد هزار رکوردی، که UpVotes آن‌ها بزرگتر از ۱۰۰ است را پیدا نماید و بعد در بین این رکوردها به دنبال آن‌هایی بگردد که DownVotesشان برابر با صفر است. یعنی برای ایجاد ایندکس ابتدا ستون UpVotes آورده شود، دستور زیر این ایندکس را ایجاد می‌نماید:

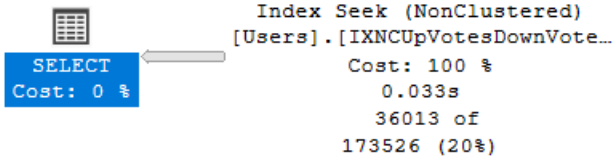
```
Create Index IXNCUpVotesDownVotes_Include On dbo.Users(UpVotes, DownVotes)
Include (Location, LastAccessDate)
```

**تصویر زیر Plan اجرائی کوئری را نمایش می‌دهد:**

```
Select ID, Location, UpVotes, DownVotes, LastAccessDa
From dbo.Users
Where UpVotes > 100 And DownVotes = 0
```

100 % | Results | Messages | Execution plan

Query 1: Query cost (relative to the batch): 100%
Select ID, Location, UpVotes, DownVotes, LastAccessDate From dbo.Users



همان طور که مشاهده می‌نمائید Query Optimizer از ایندکسی که ایجاد کردیم به خوبی استفاده نمود.

**تصویر بعدی تعداد Logical Reads را نمایش می‌دهد:**

```
Select ID, Location, UpVotes, DownVotes, LastAccessDate
From dbo.Users
Where UpVotes > 100 And DownVotes = 0
```

Results | Messages | Execution plan

```
(36013 rows affected)
Table 'Users'. Scan count 1, logical reads 1103, physical reads 0,
```

در واقع SQL Server برای اجرای این کوئری مجبور شده است که ۱۱۰۳ صفحه دیتا یا اصطلاحا Data Page را بخواند. آیا کوئری می‌تواند با Performance بهتری اجرا شود؟ پاسخ مثبت است.

برخلاف تصور فوق، اگر ابتدا ستون DownVotes در کلید ایندکس قرار گیرد تعداد Logical Reads مربوط به کوئری کاهش خواهد یافت، دستور زیر ایندکس دیگری روی جدول Users ایجاد می‌نماید:

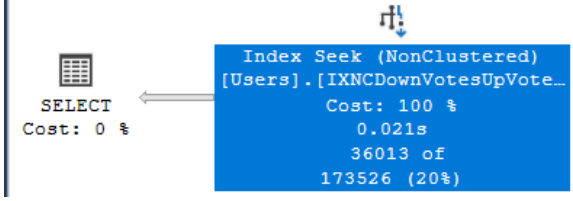
```
Create Index IXNCDownVotesUpVotes_Include On dbo.Users(DownVotes, UpVotes)
Include (Location, LastAccessDate)
```

تفاوت این ایندکس با ایندکس قبلی این است که ترتیب ستون‌ها در کلید ایندکس تغییر یافته است. تصویر زیر Plan اجرائی کوئری را نمایش می‌دهد:

```
Select ID, Location, UpVotes, DownVotes, LastAccessDate
From dbo.Users
Where UpVotes > 100 And DownVotes = 0
```

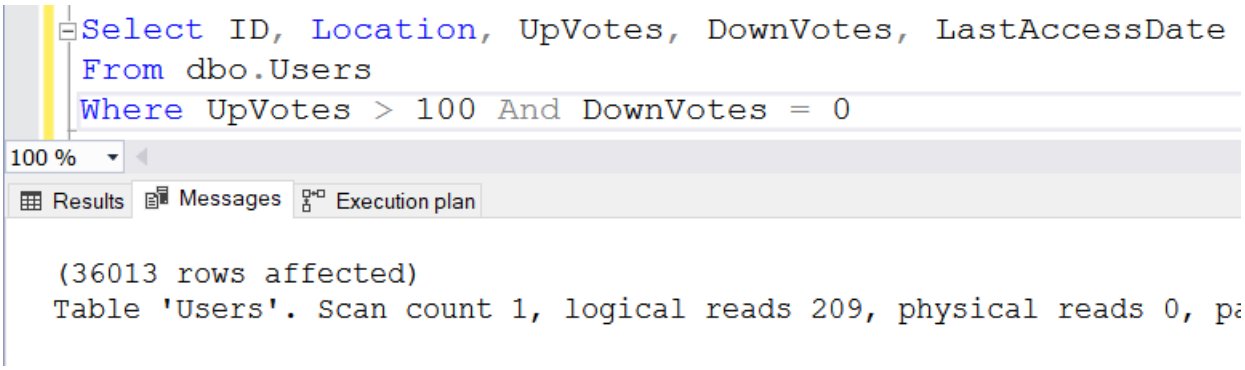
Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT [ID],[Location],[UpVotes],[DownVotes],[LastAccessDate] FROM [dbo].
```



تصویر نشان می‌دهد که Query Optimizer از ایندکس جدید استفاده نموده و این ایندکس را جهت اجرای کوئری مناسب‌تر از ایندکس قبلی تشخیص داده است.

**تصویر بعدی تعداد Logical Reads را نمایش می‌دهد:**




تصویر نشان می‌دهد که تعداد Logical Reads برابر با ۲۰۹ است و تقریبا به یک پنجم کاهش یافته است. اینکه چرا این اتفاق می‌افتد نیاز به یک مقاله جداگانه دارد.

هدف ما از این مقاله این است که نشان دهیم، عموما ایندکسی برای اجرای کوئری مناسب تر است که باعث شود SQL Server تعداد Data Page کمتری را بخواند و همان طور که در این مقاله اثبات شد اینکه کدام ستون در کلید ایندکس اول قرار بگیرد بسیار حائز اهمیت می‌باشد. نکته دیگر اینکه هر آنچه در نگاه اول منطقی به نظر می‌رسد لزوما بهترین حالت ممکن نیست و نیاز به تست دارد.

این نکته نیز ارزش بیان نمودن را دارد که اگر ما یک کوئری را به شکل Stored Procedure بنویسیم و مقادیری که در شرط کوئری استفاده می شوند را به پروسیجر پاس نمائیم، آیا ایندکسی که ایجاد شده است به ازای همه مقادیر ورودی پروسیجر مناسب خواهد بود؟ بستگی دارد، زیرا با وجود ایندکس مشاهده می‌شود که اجرای یک پروسیجر گاهی سریع و گاهی کند است .

بنابراین نباید انتظار داشت که وقتی یک ایندکس ایجاد می‌نمائیم اجرای پروسیجر همواره سریع باشد زیرا که مشکل مرموزی به نام Parameter Sniffing بر سرعت اجرای Stored Procedure ها و Parameterized Queries تاثیر منفی می‌گذارد. با این وجود Index گذاری مناسب یکی از مهم‌ترین راه‌های افزایش Performance کوئری‌ها می‌باشد.