



عنوان مقاله: موضوعی نهفته در Sort Operator

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: مهرماه ۱۴۰۱

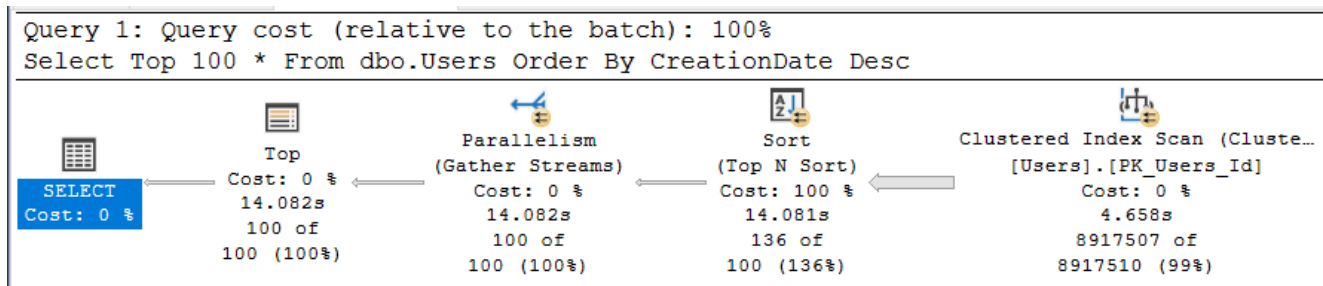
منبع: <https://nikamooz.com/sort-operator>

SQL Server بر اساس تعداد رکوردهای درخواستی از دو الگوریتم متفاوت جهت مرتب سازی داده‌ها استفاده می‌نماید. در این مقاله ما قصد داریم نشان دهیم که چگونه تعداد رکوردهای بازگشتی در یک Query عملکرد SQL Server را در مرتب سازی داده‌ها تغییر می‌دهد. جهت نمایش داده‌ها به صورت مرتب شده بر اساس یک یا چند ستون خاص از عبارت Order By استفاده می‌گردد. استفاده از این عبارت ممکن است اجتناب ناپذیر باشد مانند هنگامی که از Paging استفاده می‌نمائیم.

در این Demo ما از دیتابیس StackOverflow و جدول Users این دیتابیس استفاده می‌نمائیم. جدول Users دارای یک Clustered Index است که بر روی ستون Id می‌باشد. می‌خواهیم اطلاعات ۱۰۰ کاربری را به دست بیاوریم که اخیراً در سایت ثبت نام نموده‌اند. کوئری زیر این کار را انجام می‌دهد:

```
Select Top 100 * From dbo.Users  
Order By CreationDate Desc
```

تصویر زیر Plan اجرائی کوئری را نشان می‌دهد:

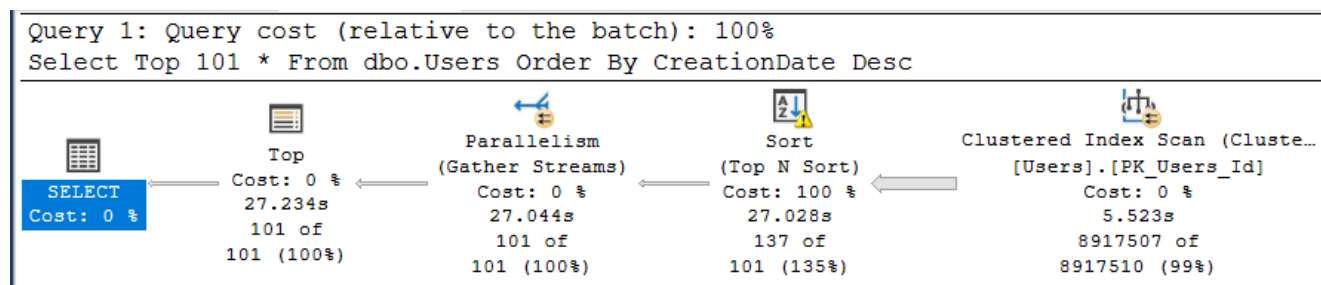


تصویر نشان می دهد که SQL Server از اپراتور Sort استفاده نموده و علامت Warning بر روی این اپراتور وجود ندارد.

حالا می خواهیم اطلاعات ۱۰۱ کاربر را به دست بیاوریم که اخیرا در سایت ثبت نام نموده اند. در واقع نسبت به کوئری قبلی یک ردیف بیشتر را واکنشی می نمائیم. کوئری زیر این کار را انجام می دهد:

```
Select Top 101 * From dbo.Users
Order By CreationDate Desc
```

تصویر زیر Plan اجرائی کوئری جدید را نشان می دهد با وجود اینکه فقط یک ردیف بیشتر واکنشی شده است اما علامت Warning بر روی اپراتور Sort وجود دارد:



تصویر بعدی متن علامت Warning را نمایش می دهد:

Warnings

Operator used tempdb to spill data during execution with spill level 1 and 4 spilled thread(s), Sort wrote 145522 pages to and read 145522 pages from tempdb with granted memory 1459744KB and used memory 1459744KB

SQL Server مجبور شده است تقریباً ۱۴۵ هزار Page هشت کیلوبایتی داده را در دیتابیس سیستمی TempDB بنویسد و این کار سرعت اجرای کوئری را بسیار کاهش می دهد. مدت زمان اجرای کوئری دوم از مدت زمان اجرای کوئری اول بسیار بیشتر است، به عبارت دیگر کوئری اول خیلی سریع تر از کوئری دوم اجرا می شود.

چرا این اتفاق افتاد؟ آیا واکنشی یک رکورد بیشتر می‌تواند سرعت اجرای کوئری را بسیار کاهش دهد؟

هنگامی که ۱۰۰ رکورد و کمتر به صورت مرتب شده واکنشی می‌شوند SQL Server جهت مرتب سازی از یک الگوریتم خاص استفاده می‌نماید و هنگامی که ۱۰۱ رکورد و بیشتر به شکل مرتب شده واکنشی می‌شوند، SQL Server الگوریتم دیگری را مورد استفاده قرار می‌دهد. در ادامه به بررسی بیشتر این موضوع می‌پردازیم. تصویر زیر نشان می‌دهد که در حالت واکنشی ۱۰۰ رکورد به شکل مرتب شده میزان Memory ایده آل SQL Server جهت اجرای کوئری برابر با 4 مگابایت است:

MemoryGrantInfo	
DesiredMemory	4096
GrantedMemory	4096
GrantWaitTime	0
IsMemoryGrantFeedbackAdjusted	NoAccurateGrant
LastRequestedMemory	4096
MaxQueryMemory	1479208
MaxUsedMemory	4032
RequestedMemory	4096
RequiredMemory	4032
SerialDesiredMemory	856
SerialRequiredMemory	816
Optimization Level	FULL

اما هنگامی که ۱۰۱ رکورد به شکل مرتب شده واکنشی شده است میزان حافظه ایده آل SQL Server جهت اجرای کوئری تقریباً به چهار گیگابایت افزایش یافته است! تصویر زیر این مسئله را نشان می‌دهد:

MemoryGrantInfo	
DesiredMemory	3963584
GrantedMemory	1460512
GrantWaitTime	0
IsMemoryGrantFeedbackAdjusted	YesAdjusting
LastRequestedMemory	1479208
MaxQueryMemory	1460512
MaxUsedMemory	1460512
RequestedMemory	1460512
RequiredMemory	2816
SerialDesiredMemory	3961272
SerialRequiredMemory	512
Optimization Level	FULL

تخمین میزان حافظه مورد نیاز جهت اجرای کوئری در حالت دوم که ۱۰۱ رکورد واکشی شده کلا بهم ریخته زیرا که الگوریتم مرتب سازی داده تغییر یافته است.

جهت برطرف نمودن مشکل فوق می توان از ایندکس گذاری مناسب استفاده نمود و اینکه فقط ستون های مورد نیاز باید واکشی شوند. اما باید به این نکته توجه داشت که در محیط عملیاتی کوئری ها به این سادگی نیستند و بخصوص در کوئری هایی که Join وجود دارد ممکن است ایندکس گذاری موجب حذف اپراتور Sort نشود. همواره تاکید بر این است که فقط ستون های مورد نیاز در کوئری در Select List قرار گیرند و از واکشی ستون های اضافی خودداری شود اما باید به این نکته توجه ویژه داشت که تاثیر تعداد رکوردهای بازگشتی در سرعت اجرای کوئری ها بسیار حائز اهمیت است.

در ادامه سوالاتی مرتبط با مرتب سازی داده ها مطرح می گردد:

بسیاری از Application ها از تکنیک صفحه بندی یا Paging جهت نمایش داده استفاده می نمایند. سوال مهم این است که آیا نیاز هست بیشتر از صد رکورد در هر صفحه به کاربر نمایش داده شود؟

سوال دیگر این است که آیا وب سرویس هایی که ریز رکوردها را واکشی می نمایند نیاز دارند داده ها را به صورت مرتب شده دریافت کنند؟

آیا این امکان وجود دارد که مرتب سازی داده در سمت گیرنده انجام شود و عبارت Order By از کوئری حذف گردد؟

آیا داده هایی که به یک جدول موقت جهت نگه داشتن نتایج میانی درج می شوند باید مرتب شده باشند؟ به صورت کلی در عبارت ... Select ... Insert Into آیا به عبارت Order By نیاز است؟

نوشتن عبارت Order By در کوئری هزینه دارد. همچنین ایندکس گذاری Free نیست. ایندکس سبب افزایش حجم دیتا می شود، حجم و مدت زمان عملیات پشتیبان گیری را افزایش می دهد، عملیات DML را کند می کند، در مدت زمان اجرای دستور CheckDB DBCC تاثیر می گذارد و در آخر ایندکس ها نیاز به نگهداری دارند.