تعریف یک Clustered Index

ما احتمالا به شاخص هایی نیاز خواهیم داشت که در بررسی ها کمک می کنند. یکی از مهمترین تصمیمات نمایه سازی (indexing) تعیین ستون هایی است که clustered index را تشخیص خواهند داد.

توضیحات:

ممکن است آن را قبلا خوانده باشید و یا ممکن است در مورد آن چیزی ندانید، به هرحال من دوباره آن را در اینجا مطرح می کنم. شما در هر جدول تنها یک clustered index دارید، زیرا clustered index ترتیب منطقی داده در جدول را تعیین می کند و جدول حقیقی را تشکیل می دهد. نداشتن clustered index باید یک تصمیم آگاهانه باشد که با دقت مورد بررسی قرار بگیرد.

یک clustered index چگونه می تواند کمک کند و جرا به آن احتیج دارید؟ اگر شما در حال گروه بندی، مرتب سازی، و جستجوی دامنه ی مجموعه های بزرگ می باشید، دسته بندی (clustering) ستون های مربوط به بررسی شما می تواند بررسی را به طور قابل ملاحظه ای افزایش داده و از مخازن کمتری (مانند CPU و حافظه) استفاده کند. داده ای که یک clustered index را تشکیل می دهد، به عنوان ساختار درخت B با خود داده ذخیره می شود.

به این مثال توجه کنید که در آن محدوده ای از ردیف ها را از جدول AdventureWorks انتخاب میکنیم، Production.TransactionHistory با استفاده از دسته بندی، کلید اصلی مجزا روی TransactionID:

```
set statistics io on
go
select *
from Production.TransactionHistory
where TransactionID between 211000 and 211200
go
```

با امتحان شمارش های I/O مشاهده می کنیم که 6 صفحه داده لازم است خوانده شود تا درخواست کامل شود.

(201 row(s) affected)))))) Table 'TransactionHistory'. Scan count 1, logical reads 6, physical reads 0, read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.

اکنون اجازه بدهید کلید اولیه را به عنوان یک non-clustered دوباره ایجاد کنیم.

```
alter table Production.TransactionHistory
drop constraint PK_TransactionHistory_TransactionID
```

```
go
alter table Production.TransactionHistory
add constraint PK_TransactionHistory_TransactionID primary key nonclustered (TransactionID)
go
```

با اجرای دوباره ی همان بررسی و امتحان مجدد شمارش های مشاهده می کنیم که اکنون 205 صفحه داده برای تکمیل همان درخواست لازم است.

```
(201 row(s) affected)))))) Table 'TransactionHistory'. Scan count 1, logical reads 205, physical reads 0,
read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob read-ahead reads 0.
```

SQL Server به طور خودکار کلید اصلی را یک clustered index می سازد. اگر ستون های دیگر بهتر می توانند بررسی های انتقادی را پشتیبانی کنند، کلید اولیه لازم نیست clustered index از جدول باشد.

انتخاب clustered index یک عمل تعادلی بین اجرای بررسی در مقابل سلامت جدول می باشد. جدول های به شدت volatile (به عنوان مثال جدول هایی که ورودی های بسیاری را تجربه می کنند)، می توانند تقسیم بندی منطقی شدیدی را تجربه کنند. اگر clustered index به طور یکواخت در حال افزایش داده نمی باشد و جدول به شدت volatile باشد، شما تقسیم بندی سریع جدول را تجربه خواهید کرد که می تواند منجر به I/O اضافه برای بازیابی داده در تکمیل درخواست بررسی، شود. اگر هیچکدام از ستون ها کاندید خوبی برای یک clustered index نباشند، معمولا افزایش یکنواختی در مقادیر ایجاد می کنم.