عنوان: آشنایی با Column Store Index در SQL Server 2012
نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۳۹۱/۰۷/۲۹ ۱۶:۹
آدرس: www.dotnettips.info
گروه‌ها: SQL Server, SQL Server 2012

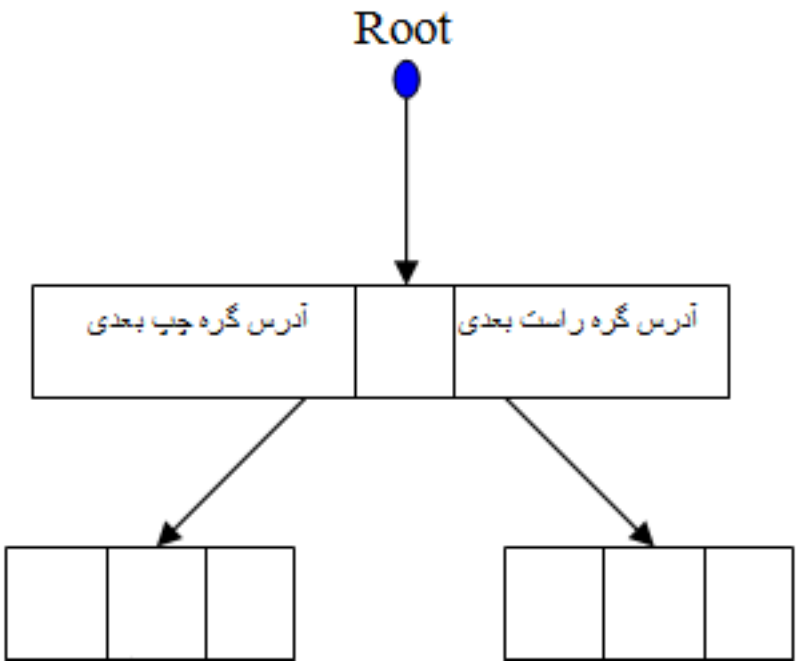
Column Store Index یکی از ویژگیهای جدید SQL Server 2012 می باشد، که کارایی Query های قایل اجرا روی دیتابیس‌های با حجم داده ای بسیار بالا را (که اصطلاحا به آنها Data Warehouse یا انبار داده گویند)، چندین برابر بهبود بخشیده است .

قبل از توضیح در مورد Column Store مختصری در مورد نحوه ذخیره سازی داده‌ها در SQL Server می پردازیم. می‌توان گفت در SQL Server دو روش ذخیره سازی وجود دارد،یکی بصورت ردیفی که اصطلاحا به آن Row Store یا Row-Wise گویند، و دیگری بصورت ستونی که اصطلاحا به آن Column Store گویند .

در روش ذخیره سازی Row Store، مقادیر ستونها در یک سطر بصورت متوالی ذخیره می‌شوند، در این روش ذخیره سازی از ساختار B-Tree یا Heap استفاده می‌شود.

یادآوری: در ساختار B-Tree، یک گره Root وجود دارد، و گره بعد از Root گره ای است که آدرس گره راست بعدی و آدرس گره چپ بعدی را در خود نگه می‌دارد.

شکل زیر نمای یک درخت B-Tree می‌باشد:

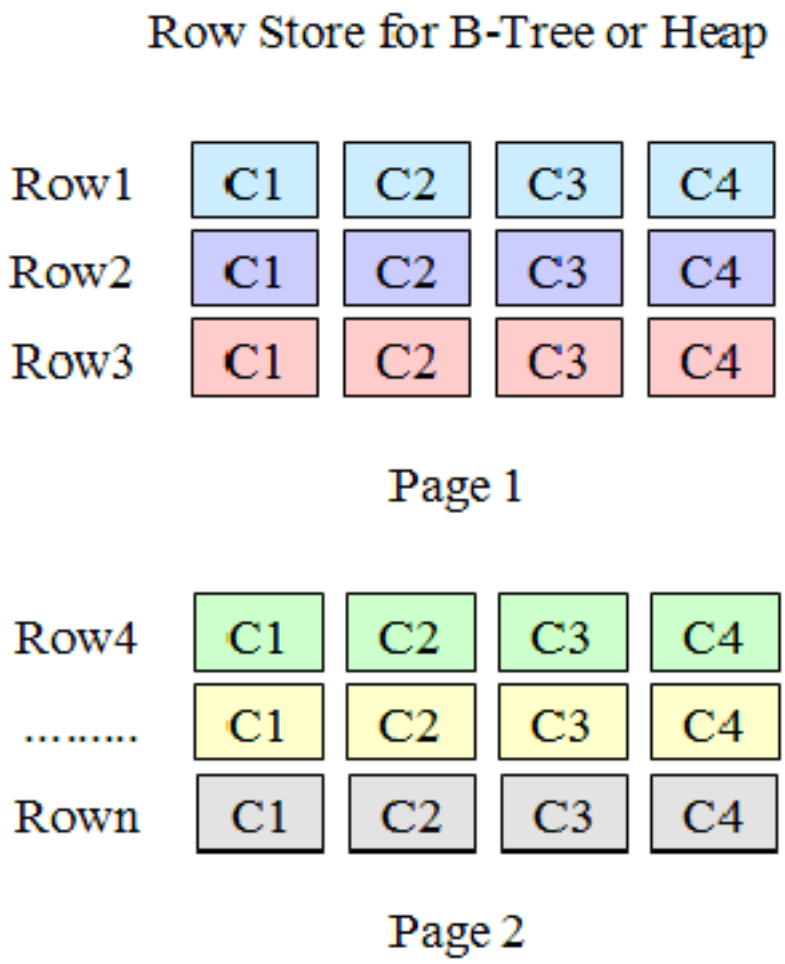


جهت کسب اطلاعات بیشتر درمورد ساختار B-Tree

یادآوری: وقتی در یک جدول، ایندکسی از نوع Clustered ایجاد نماییم، SQL Server، در ابتدا یک کپی از جدول ایجاد و داده‌های جدول را از نو مرتب می‌نماید، و ساختار صفحه ریشه و دیگر صفحات را ایجاد می‌کند و سپس جدول اصلی را حذف می‌نماید. به جدولی که Clustered Index ندارد، اصطلاحا Heap گویند.

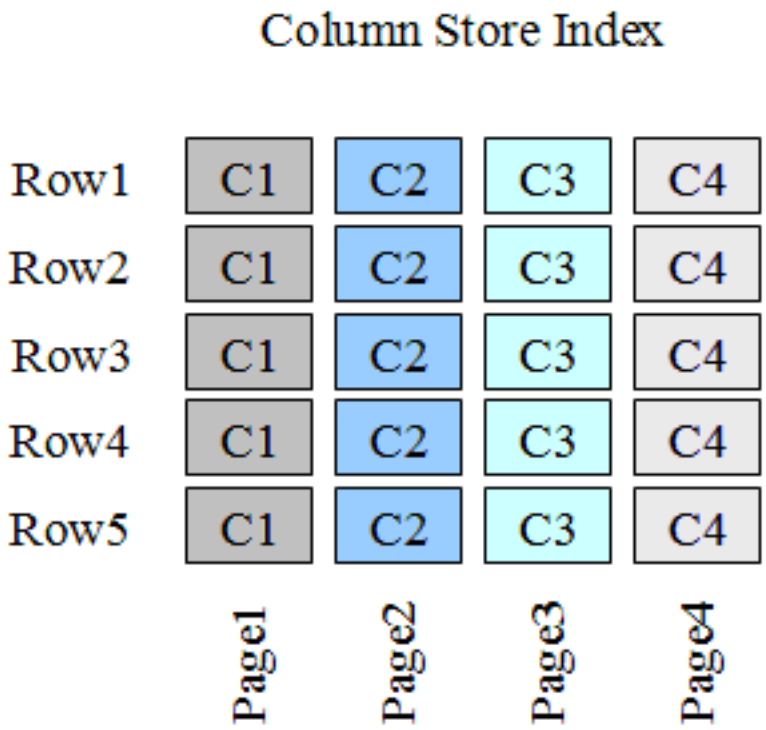
برخلاف ذخیره سازی Row Store، در ذخیره سازی Column Store، داده‌ها بصورت ستونی ذخیره می‌شوند،در این روش داده ها، فشرده سازی می‌شوند و اینکار باعث می‌شود،در زمان درخواست یک Query، نیاز به Disk I/o به حداقل برسد، در نتیجه، زمان و سرعت پاسخگویی به پرس و جوها بسیار افزایش می‌یابد.

شکل زیر نحوه ذخیره سازی داده ها،بصورت Row Store را نمایش می‌دهد:



شکل بالا ذخیره سازی داده ها، در ساختار B-Tree یا Heap را نمایش می‌دهد، در شکل فوق یک جدول چهار ستونی با N سطر (Row) در نظر گرفته شده است.بطوریکه ستونهای هر Row بطور متوالی در یک صفحه (Page) یکسان ذخیره می‌شوند.

شکل زیر نحوه ذخیره سازی داده ها،بصورت Column Store را نمایش می‌دهد:



مطابق شکل،ستونهای مربوط به هر Row،همگی در یک صفحه (Page) یکسان ذخیره شده اند. به عنوان مثال ستون C1 که مربوط به سطر اول (Row1) می‌باشد، با ستون C1 که مربوط به سطر دوم (Row2) می‌باشد، در یک ستون و در یک صفحه (Page1) ذخیره شده اند، و الی آخر ...

سئوال: یکبار دیگر به هردو شکل با دقت نگاهی بیاندازید، عمده تفاوت آنها در چیست؟

جواب: درست حدس زدید، تفاوت بارز بین دو روش Column Store و Row Store در نحوه ذخیره سازی داده‌ها می‌باشد. بطور مثال، فرض کنید،در روش ذخیره سازی Row Store، به دنبال مقادیری از ستون C2 می‌باشید، SQL Server می‌بایست کل رکوردهای جدول (منظور همه Rowها در همه Page ها)را Scan نماید، تا مقادیر مربوط به ستون C2 را بدست آورد.درحالیکه در روش ذخیره سازی Column Store، جهت یافتن مقادیر ستون C2، نیازی به Scan نمودن کل جدول نیست،بلکه SQL Server فقط به Scan نمودن ستون دوم (C2) یا Page2 بسنده می‌نماید.همین امر باعث افزایش چندین برابری، زمان پاسخگویی به هر Query می‌شود.

سئوال: در روش ذخیره سازی Column Store، چگونه مصرف حافظه بهینه می‌شود؟

جواب: واضح است، که در روش Row Store، SQL Server مجبور است، برای بدست آوردن داده‌های مورد نظرتان،کل اطلاعات جدول را وارد حافظه نماید(اطلاعات اضافه ای که به هیچ وجه بدرد، نتیجه پرس و جوی شما نمی‌خورد)، و شروع به Scan داده‌های مد نظر شما می‌نماید.بطوریکه در روش Column Store، SQL Server، فقط ستون داده‌های مورد پرس و جو را در حافظه قرار می‌دهد.(در واقع فقط داده هایی را در حافظه قرار می‌دهد، که شما به آن نیاز دارید)،بنابراین،طبیعی است که در روش Column Store مقدار حافظه کمتری نسبت به روش Row Store در هنگام اجرای Query استفاده می‌شود. به عبارت دیگر می‌توان گفت که در روش Column Store به دلیل، به حداقل رساندن استفاده از Disk I/O سرعت و زمان پاسخگویی به پرس و جوها چندین برابر می‌شود.

برای درک بیشتر Row Store و Column Store مثالی می‌زنیم:

فرض کنید،قصد بدست آوردن ستونهای C1 و C2 از جدول A را داریم، بنابراین خواهیم داشت:

```
Select C1, C2 from A
```

روش Row Store:

در این روش همه صفحات دیسک (مربوط به جدول A) درون حافظه قرار داده می‌شود، یعنی علاوه بر ستونهای C1 و C2، اطلاعات مربوط به ستونهای C3 و C4 نیز درون حافظه قرار می‌گیرد،بطوریکه مقادیر ستونهای C3 و C4 به هیچ وجه مورد قبول ما نیست، و در خروجی پرس و جوی ما تاثیری ندارد، و فقط بی جهت حافظه اشغال می‌نماید.

روش Column Store:

در این روش فقط صفحات مروبط به ستون C1 و C2 در حافظه قرار می‌گیرد.(منظور Page1 و Page2 می‌باشد) بنابراین فقط اطلاعات مورد نیاز در خروجی، در حافظه قرار می‌گیرد.
از دیگر مزایای استفاده از روش Column Store، فشرده سازی داده می‌باشد،برای درک بیشتر توضیح می‌دهم:

همانطور که در اوایل مطلبم به عرض رساندم، در روش Row Store ، داده‌ها در یک سطر و در یک Page ذخیره می‌شوند، بنابراین امکان وجود داده‌های تکراری در یک سطر به حداقل می‌رسد، چرا که، اگر فرض کنیم چهار ستون به نام‌های ID،FirstName،LastName و City، داشته باشیم،در آن صورت بطور حتم،در یک سطر، داده تکراری وجود نخواهد داشت، اما ممکن است در تعداد سطرهای زیاد داده‌های تکراری مانند Firstname یا City و غیرو بوجود بیاید، این موضوع را بیان کردم، چون می‌خواستم عنوان کنم،بسیاری از الگوریتم‌های فشرده سازی از الگوی تکراری بودن داده، جهت فشرده سازی داده‌ها استفاده می‌کنند، به همین جهت فشرده سازی در روش Row Store به حداقل می‌رسد و فضای اشغال شده در حافظه دراین روش بسیار زیاد خواهد بود. اما در روش Column Store ، امکان تکراری بودن مقادیر یک ستون بسیار زیاد است، بطور مثال ممکن است تعداد افرادی را که نام شهر آنها "تهران" باشد مثلا 20 بار تکرار شده باشد، و چون در روش Column Store، ستون‌ها در یک Page ذخیره می‌شوند، بنابراین امکان استفاده از الگوریتمهای فشرده سازی در این روش بسیار بالا می‌باشد، در نتیجه مقدار فضایی را که در حافظه یا دیسک سخت توسط این روش اشغال می‌شود، بسیار کمتر از روش Row Store است.
چه موقع می‌توانیم از Column Store استفاده نماییم:

در تعریف Column Store گفته بودم، روش فوق، جهت بهبود بخشیدن به زمان و سرعت پاسخگویی به Queryهای اجرا شده روی دیتابیس‌های با حجم داده ای بسیار بالا(Data Warehouse) می‌باشد، به بیان ساده‌تر Column Store را روی دیتابیس‌های offline یا دیتابیسهایی که صرفا جهت گزارش گیری مورد استفاده قرار می‌گیرند، تنظیم می‌نمایند.در واقع با تنظیم Column Store Index روی Databaseهای بزرگ مانند Databaseهای بانک‌ها که حجم داده ای میلیونی در جداول آنها وجود دارد، سرعت پاسخگویی Query ها، چندین برابر افزایش می‌یابد.
در یک جدول می‌توانید، هم Column Store Index داشته باشید و هم یک Row Store Index (منظور یک Clustered Index می باشد)

Syntax برای ایجاد Column Store Index به شرح ذیل می‌باشد:

```
CREATE [ NONCLUSTERED ] COLUMNSTORE INDEX index_name
    ON <object> ( column  [ ,...n ] )
    [ WITH ( <column_index_option> [ ,...n ] ) ]
    [ ON {
           { partition_scheme_name ( column_name ) }
           | filegroup_name
           | "default"
         }
    ]
[ ; ]

<object> ::=
{
    [database_name. [schema_name ] . | schema_name . ]
     table_name
```

```
{

<column_index_option> ::=
{
      DROP_EXISTING = { ON | OFF }
    | MAXDOP = max_degree_of_parallelism
 }
```

یک Column Store Index می‌بایست از نوع NONCLUSTERED باشد.

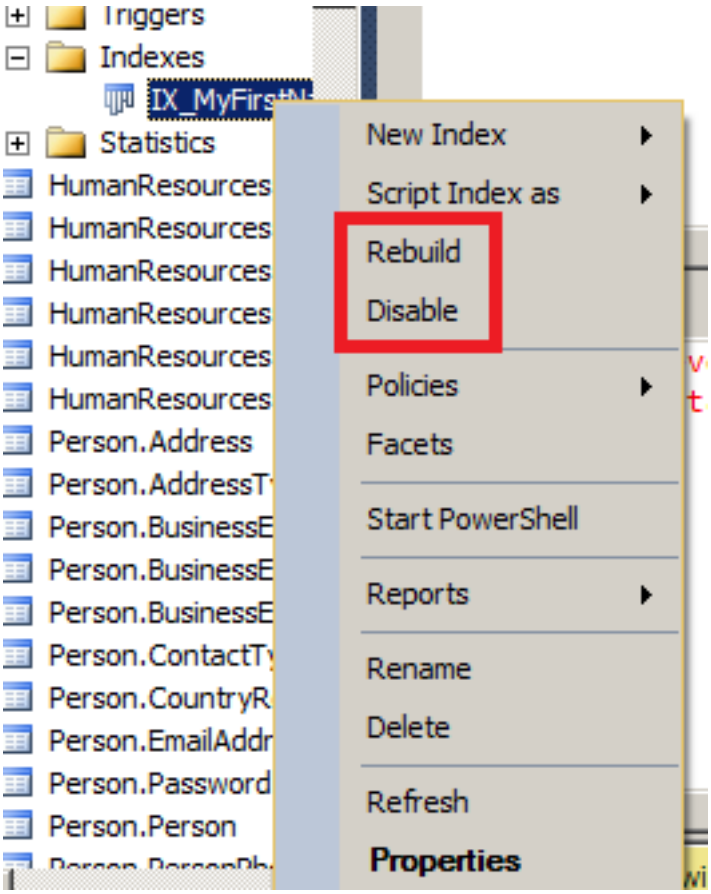
مثال از یک Column Store Index :

```
CREATE NONCLUSTERED COLUMNSTORE INDEX [IX_MyFirstName_ColumnStore]
ON [Test]
(Firstname)
```

در قطعه کد بالا، یک Column Store Index به نام Ix_MyFirstname_ColumnStore روی فیلد Firstname از جدول Test ایجاد شده است.

محدودیت‌های استفاده از Column Store Index به اختصار به شرح ذیل می‌باشد:

زمانی که در یک جدول، یک Column Store Index ایجاد نماییم، جدول ما در حالت Read-only قرار می‌گیرد، بطوریکه از آن پس اختیار Delete،Update و Insert روی جدول فوق را نخواهیم داشت. برای اینکه بتوانید عملیات Update، Insert یا Delete را انجام دهید، میبایست Column Store Index جدول مربوطه را Disable نمایید، و برای فعال نمودن Column Store Index، می‌بایست آن را Rebuild نمایید، با کلیک راست روی ایندکس ایجاد شده در SQL Server2012 موارد Disable و Rebuild قابل مشاهده می‌باشد.



یا بوسیله Scriptهای زیر می‌توانید، عملیات Disable یا Rebuild را روی Column Store Index انجام دهید:

```
ALTER INDEX [IX_MyFirstName_ColumnStore] ON [Test] DISABLE
ALTER INDEX [IX_MyFirstName_ColumnStore] ON [Test] Rebuild
```

بیشتر از یک Column Store Index نمی‌توانید روی یک جدول ایجاد نمایید.
در صورتی که تمایل داشته باشید بوسیله Alter ، نوع فیلدی (Type)، را که Column Store Index روی آنها اعمال گردیده است، تغییر دهید، در ابتدا می‌بایست Column Store Index، خود را Drop یا حذف نمایید، سپس عملیات Alter را اعمال کنید، در غیر اینصورت با خطای SQL Server مواجه می‌شوید.
یک Column Store Index می‌تواند روی 1024 ستون در یک جدول اعمال گردد.
یک Column Store Index نمی توانند، Unique باشد و نمی‌توان از آن به عنوان Primary Key یا Foreign Key استفاده نمود.

یاد آوری: با توجه به مزایای استفاده از Column Store Index، باید بگویم که در حجم‌های داده ای کم استفاده از Row Store Index بهتر می‌باشد. پیشنهاد مایکروسافت برای استفاده از Column Store Index برای دیتابیس‌های با حجم داده ای بسیار بالا می‌باشد.
موفق باشید

منابع: Columnstore Indexes: A New Feature in SQL Server known as Project “Apollo”

Columnstore Indexes

Fundamentals of Columnstore Index

SQL Server 2012 Column Store Index Example

نظرات خوانندگان

نویسنده: صالح
تاریخ: ۱۳۹۱/۰۷/۳۰ ۰:۵۷

درود بر شما
بسیار بهره جستم از مطلبتون. عالی بود.

نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۳۹۱/۰۷/۳۰ ۱۱:۱۶

ممنون از لطفا شما

نویسنده: علیرضا جهانشاهلو
تاریخ: ۱۳۹۱/۰۷/۳۰ ۲۲:۵۷

بسیار بهره بردیم از مطلب خوبتون
با تشکر

نویسنده: libertad
تاریخ: ۱۳۹۱/۰۸/۰۳ ۱۰:۳۴

آیا چنین امکانی در oracle هم وجود دارد؟

نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۳۹۱/۰۸/۰۳ ۱۲:۳۹

بله، چنین امکانی موجود است،برای اطلاعات بیشتر می‌توانید به آدرس زیر مراجعه کنید:
Compressing Columns

نویسنده: مهران زند
تاریخ: ۱۳۹۱/۰۸/۰۵ ۱۶:۲۸

بسیار عالی و ممنون. نتیجتا این گونه ایندکس گذاری برای دیتابیس‌های که درد حال رشد هستند یعنی مداوم در حال,update insert.. هستند به درد نخواهد خورد.

نویسنده: علی قمشلویی
تاریخ: ۱۳۹۱/۰۸/۰۵ ۲۰:۵۵

با سلام و تشکر به خاطر مطلب عالی و فوق العاده قابل فهمتون
یکی از سئوالاتی که همیشه برام هنگام کوئری گرفتن از دیتابیس پیش می‌آمد این بود که :

```
select c1 from A1 where c1=x
```

```
select * from A1 where c1=x
```

کدامیک دارای کارایی بیشتری می‌باشند که دراین مقاله به طور عالی توضیح داده شده بود.
البته جوابشو نمیدم تا دوستان این مقاله زیبا را از دست ندهند.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۸/۰۵ ۲۲:۵۸

البته موردی که در این مقاله عنوان شده یکی از مشکلات Select * هست...

نویسنده: ali
تاریخ: ۱۳۹۲/۰۹/۰۵ ۲۳:۴۵

با سلام یک جایی مطلب شما گنگ ماند " یعنی علاوه بر ستونهای c1 و c2، اطلاعات مربوط به ستونهای c3 و c4 نیز درون حافظه قرار می‌گیرد " پس چرا میگن تا آنجا که میشه برای بالا بردن performance در دستور select حتما تعداد فیلدهای آن مشخص باشد با آنچه شما گفتید c3 وc4 درون حاظه لود میشوند اینطور نیست ممنون