**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**مزیت‌های استفاده از رویه‌های ذخیره شده؛ واقعیت یا توهم؟!**

متن زیر یک سری نکات و یا شاید توهماتی را مطرح می‌کند که در مورد رویه‌های ذخیره شده در اس کیوال سرور رایج هستند

**1. رویه‌های ذخیره شده در مقابل SQL Injection مقاوم هستند. کوئری‌های Ad hoc همیشه این آسیب پذیری را به همراه دارند.**

نادرست است! رویه‌های ذخیره شده‌ای که رشته‌ها را به صورت پارامتر دریافت کرده و آن‌ها را به صورت یک عبارت sql اجرا می‌کنند، آسیب پذیر هستند. اگر هنگام استفاده از کوئری‌های Ad hoc از پارامترها استفاده شود، در برابر حملات SQL Injection مصون خواهید بود.  
**2. execution plan رویه‌های ذخیره شده کش می‌شوند اما این Plan برای کوئری‌های Ad hoc هر بار محاسبه و تولید می‌گردد.**

نادرست است! اس کیوال سرور تا این اندازه بی هوش نیست! اگر execution plan ایی موجود باشد حتما استفاده خواهد شد و برای موتور اس کیوال سرور اصلا اهمیتی ندارد که کوئری در حال اجرا از یک رویه ذخیره شده صادر شده است یا از یک کوئری Ad hoc رویه‌های ذخیره شده پیش کامپایل شده نیستند و مانند تمامی کوئری‌های دیگر در زمان اجرا کامپایل می‌شوند.

**3. زمانیکه از رویه ذخیره شده استفاده می‌کنید همه چیز را در یک مکان به صورت متمرکز و مجتمع خواهید داشت (مدیریت بهتر)**

نادرست است! در یک مکان متمرکز در اختیار شما نیستند. برنامه جای خود را دارد و رویه‌های ذخیره شده در دیتابیس در جای دیگری قرار دارند و برای مثال اگر قرار باشد یک پارامتر را به رویه ذخیره شده خود اضافه کنید، کدهای شما نیز باید تغییر کنند.

**4. می‌توان از یک رویه ذخیره شده استفاده مجدد کرد (در نقاط مختلف یک کد) و اعمال تغییرات تنها در یک مکان (دیتابیس) باید صورت گیرد.**

هر چند این مورد درست است، اما باید دقت داشت که اگر چندین برنامه از این رویه ذخیره شده استفاده می‌کنند نباید تغییرات شما باعث از کار افتادن سایر برنامه‌ها شوند.

**5. می‌توان رویه ذخیره شده را بدون نیاز به توزیع مجدد برنامه تغییر داد.**

این مورد تا حدودی صحیح است. اگر تنها بحث بهینه سازی و امثال آن مطرح باشد صحیح است اما اگر واقعا نیاز به تغییر یک کوئری در رویه ذخیره شده وجود داشته باشد به احتمال زیاد برنامه نیز باید دستخوش تغییراتی گردد تا این دو با هم هماهنگ شوند.

**بهبود کارآیی برنامه‌های قدیمی که از SQL Server استفاده می‌کنند**

برنامه‌های قدیمی، الزاما خیلی قدیمی هم نیستند؛ برنامه‌هایی هستند پر از کوئری‌های ذیل:

|  |
| --- |
| SELECT \* FROM table1 WHERE OrderDate ='12 Mar 2004'  SET @SQL = 'SELECT \* FROM table2 WHERE OrderDate = ' + '''' + @Var + ''''  EXEC (@SQL) |

ویژگی مهم این نوع کوئری‌ها که با جمع زدن رشته‌ها و یا مقدار دهی مستقیم فیلدها تشکیل شده‌اند، «غیر پارامتری» بودن آ‌ن‌ها است.  
این نوع مشکلات با بکار گیری ORMها به نحو قابل توجهی کاهش یافته‌است؛ زیرا این نوع واسط‌ها در اغلب موارد، در آخر کار کوئری‌هایی پارامتری را تولید می‌کنند.

**مشکل کوئری‌های غیر پارامتری چیست؟**

استفاده‌ی وسیع از کوئری‌های غیرپارامتری با SQL Server، مشکلی را پدید می‌آورد به نام «Cache bloat» یا «کش پُف کرده» و این «پُف» به این معنا است که کش کوئری‌های اجرا شده‌ی بر روی SQL Server بیش از اندازه با Query plan ‌های مختلف حاصل از بررسی نحوه‌ی اجرای بهینه‌ی آن‌ها پر شده‌است. هر کوئری که به SQL Server می‌رسد، جهت اجرای بهینه، ابتدا پردازش می‌شود و دستور العملی خاص آن، تهیه و سپس در حافظه کش می‌شود. وجود این کش به این خاطر است که SQL Server هربار به ازای هر کوئری رسیده، این عملیات پردازشی را تکرار نکند. مشکل از زمانی شروع می‌شود که SQL Server کوئری‌هایی را که از نظر یک برنامه نویس مانند هم هستند را به علت عدم استفاده‌ی از پارامترها، یکسان تشخیص نداده و برای هر کدام یک Plan جداگانه را محاسبه و کش می‌کند. این مساله با حجم بالای کوئری‌های رسیده دو مشکل را ایجاد می‌کند:

1. مصرف حافظه‌ی بالای SQL Server که گاهی اوقات این حافظه‌ی اختصاص داده شده‌ی به کش کوئری‌ها به بالای یک گیگابایت نیز می‌رسد.

2. بالای سیستم CPU Usage

**سیستم قدیمی است؛ امکان تغییر کدها را نداریم.**

بدیهی است بهترین راه حلی که در اینجا وجود دارد، پارامتری ارسال کردن کوئری‌ها به SQL Server است تا به ازای هر تغییری در مقادیر آن‌ها، این کوئری‌ها باز هم یکسان به نظر برسند و SQL Server سعی در محاسبه‌ی مجدد Plan آن‌ها نکند. اما ... اگر این امکان را ندارید، خود SQL Server یک چنین قابلیت‌هایی را به صورت توکار تدارک دیده‌است که باید فعال شوند.

**فعال سازی پارامتری کردن خودکار کوئری‌ها در SQL Server**

اگر نمی‌توانید کدهای یک سیستم قدیمی را تغییر دهید، SQL Serverمی‌تواند به صورت خودکار این‌کار را برای شما انجام دهد. در این حالت فقط کافی است یکی از دو دستور ذیل را اجرا کنید:

|  |
| --- |
| --Forced  ALTER DATABASE dbName SET PARAMETERIZATION FORCED  --Simple  ALTER DATABASE dbName SET PARAMETERIZATION SIMPLE |

حالت simple بیشتر جهت پارامتری کردن خودکار کوئری‌های select بکار می‌رود. اگر می‌خواهید تمام کوئری‌های select, insert, update و deleteرا نیز پارامتری کنید، باید از حالت forced استفاده نمائید.

**فعال سازی بهبود کارآیی SQL Server با کوئری‌های Ad-Hoc زیاد**

به کوئری‌های غیرپارامتری، کوئری‌های Ad-Hoc نیز گفته می‌شود. اگر سیستم فعلی شما، تعداد زیادی کوئری Ad-Hoc تولید می‌کند، می‌توان فشار کاری SQL Server را برای این مورد خاص، تنظیم و بهینه سازی کرد.

فعال سازی گزینه‌ی ویژه‌ی «Optimize for Ad hoc Workloads» سبب می‌شود تا SQL Server پس از مدتی به صورت خودکار کش Plan کوئری‌هایی را که به ندرت استفاده می‌شوند، حذف کند. همین مساله سبب آزاد شدن حافظه و بهبود کارآیی کلی سیستم می‌گردد. همچنین باید درنظر داشت که کش Plan کوئری‌ها نامحدود نیست و سقفی دارد. به همین جهت آزاد شدن آن، کش کردن کوئری‌هایی را که بیشتر استفاده می‌شوند، ساده‌تر می‌کند.

برای اعمال آن به یک بانک اطلاعاتی خاص، نیاز است دستورات ذیل را اجرا کرد:

|  |
| --- |
| use dbName;  -- Optimizing for Ad hoc Workloads  exec sp\_configure 'show advanced options',1;  RECONFIGURE;  go  exec sp\_configure 'optimize for ad hoc workloads',1;  RECONFIGURE;  Go |