**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

Extended events ای که در قسمت قبل بررسی شدند، جهت جمع آوری اطلاعات آماری تک کوئری‌ها مورد استفاده قرار می‌گیرند؛ اما Dynamic management objects یا به اختصار DMO's، تجمعی عمل می‌کنند برای مثال جهت محاسبه‌ی میانگین logical reads چند کوئری مانند هم. متن یک کوئری و پلن آن، توسط DMO's مختلفی قابل استخراج هستند. متن یک کوئری توسط sys.dm\_exec\_sql\_text قابل استخراج است و برای دسترسی به کوئری پلن‌ها از sys.dm\_exec\_query\_plan، sys.dm\_exec\_cached\_plans و sys.dm\_exec\_text\_query\_plan استفاده می‌شود. در این حالت برای دسترسی به اطلاعات آماری از sys.dm\_exec\_query\_stats و sys.dm\_exec\_function\_stats کمک گرفته خواهد شد.

**استفاده از Dynamic management objects برای جمع آوری اطلاعات آماری کوئری‌ها**

در ادامه در طی چند مثال، روش استخراج اطلاعات آماری کوئری‌ها را توسط DMO's بررسی می‌کنیم.

**دریافت متن کوئری‌های در حال اجرا**

توسط کوئری زیر که توسط تابع sys.dm\_exec\_sql\_text اجرا می‌شود، می‌توان لیست کوئری‌های در حال اجرای بر روی بانک‌های اطلاعاتی جاری را بدست آورد:

|  |
| --- |
| SELECT  [r].[session\_id],  DB\_NAME([r].[database\_id]) [DatabaseName],  [t].[text]  FROM sys.dm\_exec\_requests [r]  CROSS APPLY sys.dm\_exec\_sql\_text([r].sql\_handle) [t];  GO |

در اینجا text، همان متن کوئری است و هربار که این کوئری اجرا می‌شود، نتیجه‌ی متفاوتی را بر اساس کوئری‌هایی که در آن لحظه در حال اجرا هستند، دریافت خواهیم کرد.

تابع sys.dm\_exec\_sql\_text برای اجرا نیاز به یک sql\_handle دارد که آن‌را از طریق sys.dm\_exec\_requests می‌توان تامین کرد.  
**دریافت پلن کوئری‌های در حال اجرا**

توسط کوئری زیر که توسط تابع sys.dm\_exec\_query\_plan اجرا می‌شود، می‌توان لیست پلن کوئری‌های در حال اجرای بر روی بانک‌های اطلاعاتی جاری را بدست آورد:

|  |
| --- |
| SELECT  [r].[session\_id],  DB\_NAME([r].[database\_id]) [DatabaseName],  [t].[text],  [p].[query\_plan]  FROM sys.dm\_exec\_requests [r]  CROSS APPLY sys.dm\_exec\_sql\_text([r].sql\_handle) [t]  CROSS APPLY sys.dm\_exec\_query\_plan([r].[plan\_handle]) [p];  GO |

تابع sys.dm\_exec\_query\_plan برای اجرا نیاز به یک plan\_handle دارد که آن‌را از طریق sys.dm\_exec\_requests می‌توان تامین کرد.

حاصل این کوئری، به همراه text یا اصل متن کوئری‌های در حال اجرا و همچنین query\_plan، یا همان اطلاعات XML ای پلن که در قسمت اول، نمونه‌ای از آن‌را بررسی کردیم، می‌باشد که با کلیک بر روی هر کدام در management studio، نمایش گرافیکی آن‌ها ظاهر خواهد شد. البته این پلن‌ها، تنها تخمین‌ها را به همراه دارند؛ چون از کش خوانده می‌شوند.

**دریافت لیست پلن‌های کش شده**

توسط Viewای به نام sys.dm\_exec\_cached\_plans می‌توان به لیست پلن‌های کش شده‌ی در سیستم دسترسی یافت:

|  |
| --- |
| SELECT \* FROM sys.dm\_exec\_cached\_plans; |

البته خروجی آن، آنچنان جالب نیست. چون یکی از ستون‌های آن، فقط حاوی همان plan\_handle ای است که در مثال قبل بررسی کردیم و به خودی خود، حاوی اطلاعات قابل مشاهده‌ای نیست. به همین جهت اگر بخواهیم آن‌را با کوئری‌هایی که تاکنون نوشتیم، ترکیب کنیم به کوئری زیر خواهیم رسید:

|  |
| --- |
| SELECT  [r].[session\_id],  DB\_NAME([r].[database\_id]) [DatabaseName],  [cp].[objtype],  [cp].[size\_in\_bytes],  [t].[text],  [p].[query\_plan]  FROM sys.dm\_exec\_requests [r]  CROSS APPLY sys.dm\_exec\_sql\_text([r].sql\_handle) [t]  CROSS APPLY sys.dm\_exec\_query\_plan([r].[plan\_handle]) [p]  JOIN sys.dm\_exec\_cached\_plans [cp] ON [r].[plan\_handle] = [cp].[plan\_handle]; |

مزیت این کوئری نسبت به موارد قبلی، وجود ستون‌های جدید objtype و size\_in\_bytes است که بیانگر نوع کوئری، مانند AdHoc و اندازه‌ی پلن در کش هستند.

**دریافت متن پلن‌های تو در تو و عمیق**

با استفاده از تابع sys.dm\_exec\_text\_query\_plan می‌توان به متن پلن‌های عمیق دسترسی یافت. در این حالت خروجی کوئری در management studio به صورت یک لینک قابل کلیک ظاهر نمی‌شود و صرفا یک متن قابل کپی است که می‌توان آن‌را با پسوند sqlplan برای بررسی‌های بعدی، ذخیره کرد:

|  |
| --- |
| SELECT  [r].[session\_id],  DB\_NAME([r].[database\_id]) [DatabaseName],  [tq].[query\_plan]  FROM sys.dm\_exec\_requests [r]  CROSS APPLY sys.dm\_exec\_text\_query\_plan([r].plan\_handle, 0, -1) [tq];  GO |

در اینجا اعداد 0 و 1- به معنای ابتدا و انتهای batch هستند.

**دریافت اطلاعات آماری کوئری‌های درحال اجرا**

توسط viewای به نام sys.dm\_exec\_query\_stats می‌توان به اطلاعات آماری کوئری‌های در حال اجرا دسترسی یافت:

|  |
| --- |
| SELECT \* FROM sys.dm\_exec\_query\_stats; |

این کوئری تعداد ستون‌های قابل توجهی را به همراه دارد مانند Physical reads، logical reads و .... به همین جهت نیاز است اطلاعات مفید آن‌را فیلتر کرد:

|  |
| --- |
| SELECT  [qs].[last\_execution\_time],  [qs].[execution\_count],  [qs].[total\_logical\_reads]/[qs].[execution\_count] [AvgLogicalReads],  [qs].[max\_logical\_reads],  [t].[text],  [p].[query\_plan]  FROM sys.dm\_exec\_query\_stats [qs]  CROSS APPLY sys.dm\_exec\_sql\_text([qs].sql\_handle) [t]  CROSS APPLY sys.dm\_exec\_query\_plan([qs].[plan\_handle]) [p]  WHERE [qs].[execution\_count] > 25 OR [qs].[total\_logical\_reads] > 10000  ORDER BY [qs].[total\_logical\_reads]/[qs].[execution\_count] DESC; |

این کوئری در حقیقت ترکیبی است از کوئری‌هایی که تاکنون نوشتیم و در آن text و query\_plan از sys.dm\_exec\_sql\_text و sys.dm\_exec\_query\_plan تامین شده‌اند، به همراه تعدادی ستون مفید sys.dm\_exec\_query\_stats مانند

last\_execution\_time و AvgLogicalReads به علاوه در اینجا کوئری‌هایی که بیشتر از 25 بار اجرا شده‌اند و یا total\_logical\_reads آن‌ها بیش از 10 هزار بوده، در خروجی ظاهر خواهند شد مفهوم تجمعی بودن DMO's

از SQL Server 2016 به بعد، امکان دریافت اطلاعات آماری توابع نیز میسر شده‌است:

|  |
| --- |
| SELECT \* FROM sys.dm\_exec\_function\_stats; |

یک نکته: قابلیت جدیدی تحت عنوان Query Store از زمان SQL Server 2016 معرفی شد‌ه‌است و کار آن دریافت تمام اطلاعاتی است که تاکنون بررسی کردیم و تفاوت آن، در ذخیره شده بودن آن است. یعنی این اطلاعات را داخل بانک اطلاعاتی در حال بررسی ذخیره می‌کند که شامل متن و پلن کوئری و همچنین اطلاعات آماری آن است که توسط DMO's تهیه می‌شود.