**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

در بررسی Performance ما باید چند مورد رو مورد بررسی قرار بدیم

1. سخت افزار Server
2. سخت افزار Client
3. تنظیمات اصلی خود SQL Server
4. پهنای باند شبکه
5. ساختار اصلی خود Database
6. بررسی Tempdb Database
7. بررسی Query ها، Stored Procedure ها ، View ها ،Function ها و Trigger ها
8. بررسی نرم افزار و قسمت های مشکل دار برنامه که دچار کندی می شوند
9. استفاده از Profiler و Activity Monitor
10. بررسی ساختار جداول و Index ها

مقدمه   
امروزه با افزایش نرم افزارهای تحت بانک های اطلاعاتی و پا به سن گذاشتن آنها ، یکی از مسائل مهم و دغدغه های شرکتها و سازمانهای بزرگ و کوچک ، نگهداری و مراقبت از بانک های اطلاعاتی هست. در این زمینه یکی از اساسی ترین مباحث پیچیده و گسترده، Performance Tuning یا بهبود و میزان سازی کارایی بانک های اطلاعاتی هست.

در این زمینه راهکارها و روشهای گوناگونی برای بررسی مشکل سرعت ، ترافیک شبکه ، مشکلات همزمانی و غیره تجزیه و تحلیل خواهند شد تا با ابزارهای گوناگون و تکنیکهای مختلف ، سبب کاهش مشکلات سرعت و بازدهی بانک های اطلاعاتی باشیم.

در سلسله مباحثی که مطرح خواهد شد ، ابزارهای گوناگون بررسی شده و روشهای گوناگون برای بررسی مشکلات کارایی سیستم مطرح و به تفصیل توضیح داده خواهد شد. امید است در پایان این سلسله مطالب آموزشی شاهد افزایش کارایی بانک های اطلاعاتی و افزایش عملکرد بهتر برنامه نویسان بانک های اطلاعاتی باشیم.

درس اول :

آشنایی با ساختار فایلهای Data.

هر دیتابیس حداقل شامل 2 فایل و در بعضی موارد فایلهای بیشتر می باشد.

به طور معمول هر دیتابیس شامل یک MDF  (فایل اصلی اطلاعات) و یک LDF  فایل LOG  هست که بعضی مواقع میتوان

فایلهای Data بیشتری با پسوند NDF به دیتابیس اضافه کرد. همچنین میتوان فایلهای Log  بیشتری نیز اضافه کرد که معمولا کاربردی ندارند.  
هر فایل MDF  از یک سری PageDataو Index Page تشکیل شده است.

13 نوع از  disk pages ها رو به کار میبرد.

Data pages، دو نوع از LOB pages ، row-overflow pages ، index pages ، Page Free Space(PFS) ، Global Allocation Map و Shared Global Allocation Map (GAM) و (SGAM)، Index Allocation Map(IAM)، Bulk Changed Map(BCM) و Differential Changed Map(DCM) ، File Header Page ، Boot Page .

تمام اطلاعات کاربر روی data یا LOB Page  ها ذخیره می شوند و تمام ردیفهای ایندکسها بر روی Index Page ها ذخیره می شود. PFS Page ها اطلاعات  page های قابل دسترس در دیتابیس را برای داده های جدید نگهداری میکنند Allocation Page

(GAM,SGAM,IAM) اطلاعات مابقی Page ها رانگهداری می کنند.اینها شامل هیچ ردیفی از دیتابیس نیستند و فقط برای کارهای داخلی  SQL Server به کار می روند.

BCM , DCM برای کارآمد عمل کردن Backup,Restore  به کار می روند.

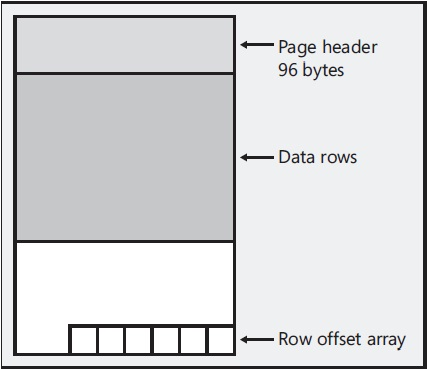
حال به بررسی Data Page ها می پردازیم

Data Pages

ساختاری هست که شامل اطلاعاتی می باشد که کاربر وارد جداول دیتابیس می کند.در SQL Server سه نوع گوناگون از data page ها داریم که هرکدام از آنها داده ها را به شیوه های گوناگون ذخیره میکنند که شامل in-row data , overflow-data , LOB ها هستند.

Data Page ها یک حجم ثابت 8KB یا 8192 bytes دارند که شامل 3 قسمت اصلی

Page Header , Data Row , Row Offset Array هست.



Page Header

همانطور که در تصویر بالا ملاحظه میکنید ، 96 بایت اول هر Data Page شامل Page Header هست و مابقی به Data row, row offset array اختصاص می یابد.

در جدول زیر اطلاعات موجود در Page Headerرا به ما نمایش میدهد.



ردیفهای اطلاعات در: In-Row Data

در ادامه Page Header ، ناحیه ای هست که داده ها درون آن ذخیره می شوند. بیشترین مقدار یک ردیف واحد اطلاعاتی برای ذخیره سازی در In-Row برابر 8060 byte می باشد.تعداد ردیفهایی که میتوانند بر روی یک Page ذخیره شوند وابستگی زیادی به ساختار جداول و نوع داده هایی که میخواهند ذخیره شوند دارند.به عنوان مثال ، یک جدول با فیلدهایی با طول ثابت غیر متغیرهایی که با var شروع شده اند بر روی هر Page تعداد ردیفهای ثابتی را ذخیره می کنند، در صورتی که داده های با طول متغیر ، بسته به میزان داده های وارد شده می توانند درون یک Page ذخیره شوند.نگهداری داده های با طول کمتر، اجازه ذخیره سازی ردیفهای بیشتری بر روی یک Page را می دهد که باعث کاهش I/O شده و منجربه قرار گرفتن تعداد Page های بیشتری در Buffer می شود که در نهایت افزایش سرعت را به همراه دارد.

Row Offset Array

شامل یک  Block متشکل از واحدهای 2 byte  می باشد.هر کدام از قسمتها شامل Offset  شروع ردیف مورد نظر در Page  می باشد.هر ردیف یک فضای 2 بایتی در این قسمت دارد .این آدرس درون هر ردیف ذخیره نمی شود چون فضای ذخیره سازی هر ردیف در Page کاهش پیدا می کند.

این Block  شامل یک مرتب سازی منطقی نیز می باشد.به عنوان مثال ، اگر یک جدول شامل Clustered Index باشد ، SQL Server به ترتیب Clustered Index Key ،مقادیر رو در این جدول ذخیره خواهد کرد.این به این معنی نیست که ردیفهایی که درون Page  ذخیره می شوند، به صورت فیزیکی براساس  Clustered Index Keyذخیره خواهند شد ، بلکه در slot 0  آدرس ردیف اول Clustered index key  ذخیره می شود و در slot 1  آدرس ردیف بعدی و الی آخر. همانطور که در ادامه خواهیم دید ،هرردیف میتواند در هر جایی از یک Page به صورت فیزیکی ذخیره شود.

بررسی: Data Page

توسط دستور DBCC Page میتوان محتویات یک Page از قبیل Page Header ، Data rows و row offset array رو برای یک Page در دیتابیس ملاحظه کرد.فقط یک کاربر با دسترسی Administrator یا همان System Administrator میتواند این دستور را اجرا کند.

این دستور Undocumented هست و توضیحات آن در Books Online نمی باشد .در ادامه این دستور کامل توضیح داده خواهد شد.

|  |
| --- |
| DBCC PAGE ({dbid | dbname}, filenum, pagenum[, printopt]) |

نتیجه این دستور به کاربر ارسال خواهد شد.

نکته مهم : توسط دستور

|  |
| --- |
| DBCC TraceOn(3604) |