عنوان مقاله: معرفی کتاب: افزایش کارایی بانک اطلاعاتی بوسیله Index
نویسنده مقاله: مسعود طاهری
تاریخ انتشار: آبان ماه ۱۳۹۳
منبع: https://nikamooz.com/increase-database-performance-by-index/

TDE چیست؟

معمولاً اطلاعات حساس موجود در بانک‌های اطلاعاتی سازمان‌ها به شکل Encryptشده ذخیره می‌شوند تا به راحتی امکان دسترسی به آنها وجود نداشته باشد.

اما یکی از چالش‌هایی که اغلب سازمان‌ها با آن مواجه هستند امکان دسترسی فیزیکی برخی از کاربران، هکرها و... به فایل‌های فیزیکی بانک‌های اطلاعاتی و نسخه پشتیبان آنها می‌باشد.

در این گونه مواقع چنانچه اطلاعات سازمان به بیرون دَرز پیدا کند ممکن است کسب و کار یک سازمان به علت لُو رفتن دارایی‌های اطلاعاتیش به خطر افتاده و شاید تا مرحله ورشکستگی پیش رود.

راه‌حل رفع این مشکل در SQL Server چیست؟

در SQL Server ۲۰۰۸,۲۰۱۲ امکانی وجود دارد که کلیه فایل‌های فیزیکی مربوط به بانک اطلاعاتی را به طور خودکار Encrypt و Dec , n rypt می‌کند این قابلیت در SQL Server با نام TDE (Transparent Data Encryption) شناخته می‌شود.

طی این مقاله نحوه پیاده‌سازی از این قابلیت به ازای یک بانک اطلاعاتی تحت SQL Server ۲۰۰۸,۲۰۱۲ را فرا خواهید گرفت.

تعاریف اولیه

Encryption چیست؟ Encryption در لغت به معنی پنهان کردن و یا رمز کردن می‌باشد. اما در علم کامپیوتر به معنی رمز کردن یا رمز گذاری داده‌ها از طریق یک الگوریتم خاص جهت محافظت از خود آنها است. معمولاً از یک کلید جهت انجام این عملیات استفاده می‌شود.

Decryption چیست؟ Decryption در لغت به معنی آشکار کردن و یا رمز گشایی می‌باشد. اما در علم کامپیوتر به معنی رمزگشایی داده‌ها با استفاده از کلیدی است که عملیات Encryption با آن انجام شده است.

Key چیست؟ Key یا کلید فرمول و یا روشی است که برای انجام عملیات Encryption و یا Decryption استفاده می‌شود.

آشنایی با فایل‌های فیزیکی بانک‌های اطلاعاتی در SQL Server

به طور کلی بانک‌های اطلاعاتی در SQL Server کلیه داده‌های خود را در فایل‌های فیزیکی بر روی دیسک ذخیره می‌کنند. هر بانک اطلاعاتی در SQL Server حداقل از دو نوع فایل زیر تشکیل شده است

Data File: در این نوع فایل‌ها کلیه اطلاعات موجود در بانک اطلاعاتی اعم از جداول، فیلدها، رکوردها و... ذخیره می‌شوند. در واقع این فایل دارایی اطلاعاتی سازمان شما می‌باشد. معمولاً پسوند این نوع فایل‌ها در بانک اطلاعاتی *.mdf و یا *.ndf می‌باشد.

Log File: کلیه وقایعی که به ازای یک بانک اطلاعاتی رخ می‌دهد در این فایل نوشته می‌شود. در این فایل کلیه عملیاتی که مربوط به تغییر داده‌ها بوده ثبت می‌شود تا SQL Server آنها را طی فرآیند خاص به Data File منتقل نماید.

سازمان‌هایی که از SQL Server به عنوان DBMS (Database Management System) استفاده می‌کنند باید از این نوع فایل‌ها (Data File, Log File) محافظت کامل را به عمل آورند.

معمولاً محافظت از این نوع فایل‌ها در سازمان‌ها به صورت زیر انجام می‌شود:

۱- پیاده‌سازی RAID برای دیسک‌هایی که فایل‌های بانک اطلاعاتی (Data File و Log File) در آن قرار دارند.

با قرار دادن این نوع فایل‌ها داخل دیسک‌های فیزیکی مختلف احتمال آسیب دیدن آنها کاهش یافته و در ضمن با استفاده از این روش سرعت دسترسی به اطلاعات موجود در فایل‌ها افزایش می‌یابد.

۲- جلوگیری از دسترسی کاربران به محل ذخیره‌سازی فایل‌های بانک‌اطلاعاتی (Data File و Log File).

در این حالت با تعریف سطح دسترسی در سطح سیستم عامل، از دسترسی کاربران به محل فیزیکی این نوع فایل‌ها جلوگیری می‌شود.

اما مشکل اصلی زمانی رخ می‌دهد که یک کاربر کنجکاو، هکر و... به نسخه‌ای از این فایل‌ها و یا نسخه پشتیبان بانک اطلاعاتی دست پیدا می‌کند. چنانچه اطلاعات حساسی داخل این نوع فایل‌ها داشته باشید ممکن است کسب و کار شما به خطر افتاده و یا شاید تا مرحله ورشکستگی پیش رود.

روش‌های Encrypt کردن اطلاعات در SQL Server

همانگونه که پیشتر اشاره شده اطلاعات حساس موجود در بانک‌های اطلاعاتی معمولاً به صورت Encrypt شده ذخیره می‌شوند. در SQL Server برای Encrypt کردن این نوع اطلاعات چهار روش وجود دارد.

۱- با استفاده از Passphrase (عبارت عبور) : در این روش عملیات Encryption و Decryption با استفاده از یک رمز عبور دریافتی از کاربر انجام می‌شود.

۲- با استفاده از Symmetric Key (کلید متقارن) : در این روش از یک کلید برای انجام عملیات Encryption و Decryption استفاده می‌شود. برای ایجاد این نوع کلیدها در SQL Server از الگوریتم‌هایی مانند DES، Triple DES،RC۴، DESX و... استفاده می‌شود.

۳- با استفاده از Asymmetric Key (کلید نامتقارن) : در این روش از یک Private Key (کلید خصوصی) برای انجام عملیات Encryption و یک Public Key (کلید خصوصی) برای انجام عملیات Decryption استفاده می‌شود. برای ایجاد این نوع کلیدها در SQL Server از الگوریتم‌هایی مانند RSA_۵۱۲، RSA_۱۰۲۴، و... استفاده می‌شود.

۴- با استفاده از Certificate (گواهینامه‌ها) : در SQL Server یک Public Key Certificate (گواهینامه کلید عمومی) به عنوان Certificate شناخته می‌شود. Certificate شامل یک Public Key (کلید عمومی) است که آن را به یک کاربر، دستگاه و یا سرویس نسبت می‌دهد و با استفاده از یک Private Key (کلید خصوصی) مرتبط به صورت انحصاری قابل شناسایی است.

سلسله مراتب Encryption در SQL Server

SQL Server از یک ساختار سلسله مراتبی برای انجام عملیات Encryption و مدیریت Key استفاده می‌کند.به طور کلی این ساختار در سه سطح زیر می‌باشد

۱- در سطح سیستم عامل (Windows Level)

۲- در سطح (SQL Server Level) SQL Server

۳- در سطح بانک اطلاعاتی (Database Level)

در پایین‌ترین سطح از این ساختار Data قرار دارد چیزی که هدف نهایی ما برای انجام عملیات Encryption می‌باشد. Data توسط Keyهای دیگر Encrypt می‌شود. همچنین Encrypt شدن این Key توسط سایر Keyهای دیگر نیز وجود دارد. در نهایت کلیه این کلیدها توسط Database Master Key یا DMK کد می‌شود. DMK نیز توسط Service Master Key یا SMK نگهداری می‌شود و در بالاترین سطح ویندوز از SMK توسط Data Protection API یا DPAPI محافظت به عمل می‌آورد.

TDE چیست؟

TDE (Transparent Data Encryption) به معنی رمزگذاری داده‌ها به صورت شفاف می‌باشد. با استفاده از این روش کلیه فایل‌های بانک اطلاعاتی به طور خودکار Encrypt و Decrypt می‌شود.

این قابلیت در SQL Server ۲۰۰۸,۲۰۱۲ موجود بوده و شامل مزایای زیر می‌باشد.

۱- جهت استفاده از آن نیازی به تغییر در Source برنامه‌های کاربردی نداشته و کلیه تنظیمات آن در سطح بانک اطلاعاتی و SQL Server می‌باشد.

۲- انجام عملیات Encryption و Decryption به طور خودکار و بلادرنگ بر روی فایل‌های بانک اطلاعاتی.

۳- انجام عملیات Encryption بر روی کلیه Backupهای بانک اطلاعاتی اعم از Full Backup، Differential Backup، Log Backup و ...

۴- استفاده از حداقل منابع سرور برای انجام عملیات Encrypt و Decrypt کردن داده‌ها

۵- با استفاده از این تکنولوژی چنانچه فایل‌ها و نسخه‌های پشتیبان بانک اطلاعاتی شما به هر صورت از سازمان خارج شود امکان دسترسی به اطلاعات آن تا زمان ارائه Certificateهای ذخیره شده در پایگاه داده Master وجود نخواهد داشت.

TDE چگونه کار می‌کند؟

TDE با استفاده از یک کلید Encrypt شده (Database Encryption Key/DEK) که در Boot Record بانک‌اطلاعاتی وجود دارد کار می‌کند. کلید TDE با استفاده از یک Certicificate که در پایگاه داده Masterقرار دارد Encrypt شده است. همچنین کلید موجود در پایگاه داده Master توسط Service Master Key رمزگذاری شده است. در صورت به سرقت رفتن یکی از فایل‌ها و یا نسخه‌های پشتیبان مربوط به بانک اطلاعاتی به سرقت رود محتوی آن بدون داشتن Cerificate موجود در پایگاه داده Master ارزشی ندارد.

زمانی که SQL Server بخواهد اطلاعات را از دیسک به Buffer Pool (بخشی از حافظه RAM متعلق به SQL Server) منتقل نماید عملیات Decryption با استفاده از DEK انجام می‌شود و در صورتیکه SQL Server خواهان انتقال اطلاعات موجود در Buffer Pool به دیسک باشد با استفاده از DEK عملیات Encryption را انجام می‌دهد. نکته مهمی که در این باره وجود دارد این است که SQL Server به طور خودکار این عملیات را انجام می‌دهد.
بنابراین چنانچه به هر نحوه فایل‌ها و یا نسخه پشتیبان مربوط به بانک اطلاعاتی شما به بیرون دَرز کند حتماً به همان کلیدی که عملیات Encrypt و Decrypt با آن انجام شده نیاز خواهد داشت.

سلسله مراتب Encryption هنگام استفاده از TDE

در پایین‌ترین سطح این ساختار Data قرار دارد که عملیات Encryption بر روی آن انجام می‌شود.عملیات Encryption بوسیله DEK یا Database Encryption Key انجام می‌شود. محل قرارگیری Data و DEK بر روی بانک اطلاعاتی شما می‌باشد.

همچنین SQL Server با استفاده از یک Certificate از DEK محافظت می‌نماید. محافظت از Certificate با استفاده از DMK یا Database Master Key می‌باشد و در نهایت DMK با استفاده از یک Service Master Key یا SMK محافظت می‌شود. لازم به ذکر است که به ازای یک Instance فقط یک نسخه از SMK وجود دارد که ایجاد آن بر عهده SQL Server می‌باشد.

برای اینکه TDE عملیات Encryptionرا بدرستی انجام بدهد باید DMK و Certificate بر روی بانک اطلاعاتی Master ایجاد شود.
حال چنانچه فایل‌های بانک اطلاعاتی و یا نسخه پشتیبان بانک اطلاعاتی شما به سرقت رود بدون داشتن DMK و Certificate امکان دستیابی به اطلاعات موجود در آن امکان‌پذیر نخواهد بود.

راه‌اندازی TDE

برای راه‌اندازی TDE باید سلسله مراتب Encryption مربوط به آن را از بالا به پایین راه‌اندازی کنید. انجام اینکار طی ۴ مرحله امکان پذیر می‌باشد

۱- ایجاد DMK : Data Master Key یا Database Master Key به ازای هر بانک اطلاعاتی به طور جداگانه وجود دارد و اطمینان می‌دهد که از کلیه کلیدهای موجود در بانک اطلاعاتی (Symmetric Key, Asymmetric Key, Certificate) توسط آن محافظت خواهد شد. DMK با استفاده از الگوریتم Triple DES و Passwordی که کاربر تعیین می‌کند محافظت می‌شود.

با توجه به اینکه TDE نیاز به استفاده از DMK دارد باید آن را حتماً در بانک اطلاعاتی Master راه‌اندازی کنید.

```
USE Master
GO
-Master KEY ایجاد
CREATE MASTER KEY ENCRYPTION BY PASSWORD ='DMK_Password'
GO
```

۲- ایجاد Certificate: همانگونه که قبلاً اشاره شد Certificate مورد نیاز TDE باید در بانک اطلاعاتی Master باشد.

```
USE master
GO
-TDECert ایجاد Certificate با نام
CREATE CERTIFICATE TDECert WITH SUBJECT ='MyCert'
GO
```

۴- ایجاد DEK: Database Encryption Key کلید مخصوص Encryption داده‌ها می‌باشد. محل ایجاد DEK بر روی بانک اطلاعاتی مورد نظر شما می‌باشد. DEK با استفاده از یک الگوریتم که توسط کاربر تعیین می‌شود عملیات

Encryption را انجام می‌دهد.این الگوریتم می‌تواند یکی از الگوریتم‌های AES_۱۹۲, AES_۱۲۸, AES_۲۵۶, Triple_DES_۳KEY محافظت می‌شود.

```
USE YourDB
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM=AES_256 -AES_192,AES_128,TRIPLE_DES_3KEY
ENCRYPTION BY SERVER CERTIFICATE TDECert
GO
```

۵- فعال‌سازی TDE بر روی بانک اطلاعاتی: زمانیکه TDE را بر روی بانک اطلاعاتی خود فعال کنید SQL Serverبه طور خودکار کلیه اطلاعات موجود در بانک اطلاعاتی شما را با استفاده از Database Encryption Key (DEK)رمزگذاری می‌کند. مدت زمان Encryption با توجه به حجم بانک اطلاعاتی و قدرت سخت‌افزار شما ممکن است کوتاه و یا طولانی باشد. طی انجام عملیات Encryption باید در نظر داشته باشید که سرویس SQL را از دسترس خارج ننماید تا عملیات Encryption به طور کامل انجام شود. لازم به ذکر است در صورت انجام اینکار عملیات Encryption ادامه خواهد یافت.

```
USE YourDB
GO
-فعال سازی قابلیت TDEبرای بانک اطلاعاتی
ALTER DATABASE YourDB SET ENCRYPTION ON
GO
```

حال اگر فایل‌های (Backup, Data File, Log File) بانک اطلاعاتی که با استفاده از TDE رمزگذاری شده است را بر روی یک سرور دیگر مورد استفاده قرار دهید با خطای زیر مواجه خواهید شد.

```
USE master

GO
```

در این سرور وجود نداردCertificate, Database Master Key چون امکان استفاده فایل های بانک اطلاعاتی وجود ندارد.

```
EXEC sp_attach_db 'YourDB','C:\TDE_TEST\YourDB.dmf'
GO
```

نحوه جابجایی بانک‌های اطلاعاتی که بر روی آن TDE فعال می‌باشد.

برای جابجایی بانک‌های اطلاعاتی که TDE بر روی آنها فعال است باید مراحل زیر طی شود.

۱- تهیه نسخه پشتیبان از Service Master Key : با توجه به اینکه SMK به ازای یک Instance از SQL Server به وجود می‌آید تهیه نسخه پشتیبان از آن ضروری است. در صورتی که در Instance جدید SMK مربوط به Instance قدیمی موجود نباشد فایل‌های بانک اطلاعاتی (با فرض انجام مراحل دیگر تا انتها) در دسترس بوده اما به ازای هر

بار Restart سرویس SQL Server بانک اطلاعاتی از دسترس خارج خواهد شد. جهت تهیه نسخه پشتیبان از SMK از دستور زیر استفاده نمایید.

```
-تهیه نسخه پشتیبان از SMK در سرور مبدا
BACKUP SERVICE MASTER KEY TO FILE='C:\TDE_TEST\SMK.bak'
ENCRYPTION BY PASSWORD='SMK_Password'
GO
```

۲- تهیه نسخه پشتیبان از : Database Master Key برای استفاده از DMK در سرور مقصد باید Backup آن در سرور مبدا تهیه شود جهت تهیه آن می‌توانید از دستور زیر استفاده نمایید.

```
-تهیه نسخه پشتیبان از DMK در سرور مبدا
BACKUP MASTER KEY TO FILE ='C:\TDE_TEST\DMK.bak'
ENCRYPTION BY PASSWORD ='DMK_Password_Backup'
GO
```

تهیه نسخه پشتیبان از Certificate : جهت تهیه نسخه پشتیبان از : جهت تهیه نسخه پشتیبان از Certificate در سرور مبدا از دستور زیر استفاده کنید.

```
USE YourDB
GO
-تهیه نسخه پشتیبان از Certificate در سرور مبدا
BACKUP CERTIFICATE TDECert TO FILE='C:\TDE_Test\CertBackup.bak'
WITH PRIVATE KEY
(
FILE='C:\TDE_Test\PrivateKey.bak',
ENCRYPTION BY PASSWORD='Cert_Password_Backup'
)
GO
```

حال چنانچه بخواهید فایل‌های (Backup ,Data File, Log File) بانک اطلاعاتی را در سرور مقصد مورد استفاده قرار دهید به راحتی می‌توانید با انجام مراحل زیر آنها را در سرور مقصد مورد استفاده قرار دهید.

۱- بازیابی نسخه پشتیبان مربوط به : Service Master Key جهت بازیابی SMK از دستور زیر استفاده نمایید.

```
-بازیابی نسخه پشتیبان SMK در سرور مقصد
RESTORE SERVICE MASTER KEY FROM FILE='C:\TDE_Test\SMK.bak'
DECRYPTION BY PASSWORD='SMK_Password'
GO
```

۲- بازیابی نسخه پشتیبان مربوط به : Database Master Key جهت بازیابی DMK از دستور زیر استفاده نمایید.

```
-بازیابی نسخه پشتیبان DMK در سرور مقصد
RESTORE MASTER KEY FROM FILE='C:\TDE_Test\DMK.bak'
DECRYPTION BY PASSWORD='DMK_Password_Backup'
ENCRYPTION BY PASSWORD='DMK_Password'
GO
-باز کردن DMK در سرور مقصد
```

```
OPEN MASTER KEY DECRYPTION BY PASSWORD='DMK_Password'
```

۳- بازیابی نسخه پشتیبان مربوط به Certificate : جهت بازیابی نسخه پشتیبان Certificate از دستور زیر استفاده نمایید.

```
-بازیابی نسخه پشتیبان Certificate در سرور مقصد
CREATE CERTIFICATE TDECert FROM FILE ='C:\TDE_Test\CertBackup.bak'
WITH PRIVATE KEY
(
FILE='C:\TDE_Test\PrivateKey.bak',
DECRYPTION BY PASSWORD='Cert_Password_Backup'
)
GO
```

۴- استفاده از فایل‌های مربوط به بانک اطلاعاتی: حال چنانچه هر کدام از نسخه‌های پشتیبان تهیه شده مربوط به SMK، DMK و Certificate را در سرور مقصد بازیابی نمایید به راحتی می‌توانید فایل‌های مربوط به بانک اطلاعاتی را مورد استفاده قرار دهید.

```
EXEC sp_attach_db 'YourDB','C:\TDE_TEST\YourDB.dmf'
```

نکاتی مهم درباره استفاده از TDE

هنگام استفاده از TDE باید به نکات زیر توجه داشت.

۱- با توجه به اینکه عملیات Encrypt و Decrypt کردن داده‌ها بیشترین استفاده از CPU را می‌کند. ممکن است این عملیات باعث افزایش کارکرد CPU شود و در برخی از موارد کارایی Database پایین آید.

۲- هنگامی که بانک اطلاعاتی شما با استفاده از TDE رمزگذاری شده است. به طور خودکار کلیه Backupهای (Full Backup, Differential Backup, Log Backup) مربوط به بانک اطلاعاتی Encrypt خواهند شد.

این مقاله در شماره ۲۳۴ ماهنامه رایانه به چاپ رسیده است