

YEDEKLEMEK VE YEDEKTEN DÖNMEK

20

Yazılımcıların da zaman zaman gerçekleştirdiği bir görev olan yedekleme işlemi, veritabanı yöneticisi görevlerinde ilk sıralarda yer alır. Veritabanı yöneticisi, yedekleme ve yedekleri yönetme görevini yerine getirir.

Bu bölümde, yedek türleri, veritabanı strateji ve planı oluşturma, SSMS ve T-SQL ile yedekleme işlemini gerçekleştirmek, alınan yedekleri geri yüklemek gibi işlemleri ele alacağız.

VERİTABANINI YEDEKLEMEK

Veritabanı projelerinde en önemli ve korunması gereken öncelikle sistem verinin kendisidir. Veritabanında bulunan verinin korunabilmesi, birçok farklı tehlikeye karşı gerçekleştirilir.

Bu tehlikeleri aşağıdaki gibi sıralayabiliriz.

YAZILIMSAL SORUNLAR

İnsan ve yazılım faktörünün bulunduğu her yerde olası sorun ve hatalara karşı hazırlıklı olunmalıdır. Veritabanı üzerinde çalışan yazılımların algoritma hataları, veritabanı geliştiricisi ya da yöneticisinin kazara veri silmesi, veritabanına izinsiz erişim ile gerçekleştirilen hacking saldırıları, işletim sistemi hataları, VTYS yönetim yazılımının hataları gibi bir çok farklı sebepten kaynaklanan hataları kapsar.

DONANIMSAL SORUNLAR

Veritabanı ve yazılımsal fiziksel donanımlar üzerinde çalışır. Bu durumda, sadece veritabanı ve yazılımdaki hataların giderilmesi tek başına yeterli değildir. Sunucu sistemindeki işlemci, disk ve belki donanımsal RAID sistemindeki olası hataları kapsar.

FİZİKSEL SORUNLAR

Yazılım ve donanımsal hatalar önemli ve en çok rastlanabilecek olası sorunlar arasındadır. Ancak yazılım ve veritabanının yer aldığı donanımların, gerçek hayattaki fiziksel konumları da çeşitli sorunlarla karşılaşabilir. Veri merkezi ya da sunucu sistemin bulunduğu farklı bir mekanın doğal afet, silahlı saldırı (sanal ortamdaki hacking saldırısının gerçek hayattaki karşılığıdır) gibi sorunları kapsar.

Kitapta verdiğim bir örneği hatırlayalım. Büyük bir GSM operatörünün veri merkezinde meydana gelen sel baskını sonucu, büyük çapta veri kaybı yaşanmış ve GSM operatörü, veri kaybının kapsamındaki müşterilerinin operatör kullanım ücreti verilerini kaybettiği için müşterilere hediye (!) etmek zorunda kalması doğal afetlere örnek verilebilir.

SQL SERVER OTOMATİK KURTARMA İŞLEMİ

SQL Server sunucusu, hata ya da elektrik kesilmesi gibi olağan olmayan sebeplerle kapandığında, **Buffer Cache** ve **Log Buffer** gibi bellek üzerinde yer alan veriler kaybedilir. SQL Server servisi tekrar çalışmaya başladığında, loglara kaydedilemeyen verileri kurtarmak için SQL Server otomatik kurtarma işlemi devreye girer. Bu işlemle birlikte, veritabanı kararlı hale getirilmek için gerekli işlemler gerçekleştirilmiş olur.

VERİTABANI LOGLAMA SEÇENEKLERİ

Veritabanı loglarının önem seviyesi ile alakalı 3 farklı loglama seçeneği vardır. Bunlar, transaction log dosyasına yansıtılacak logların seviyelerini ayarlamak için kullanılır. SQL Server da varsayılan seviye **Full Recovery**'dir. Ancak, gerekli durumlarda bu seçenek değiştirilerek performans ve log veri hacmi yönetilebilir.

SİMPLE RECOVERY MODEL

Veri kaybı önemli olmayan veritabanları için kullanılabilir. Bu model ile, log dosyası belli bir boyutun üzerine çıkmaz ve veritabanında belli bir zamana dönülemez. Loglar sadece işlemin tamamlandığına dair işaretlenene kadar tutulur ve sonrasında bu loglar silinir. Kritik öneme sahip veritabanlarında kullanılmaması gerekir.

FULL RECOVERY MODEL

Bu seçenek ile, tüm değişimler loglanır ve istenilen zamana dönülebilir. Log veri boyutu olarak en çok yer kaplayan modeldir. Loglar düzenli olarak yedeklenmezse, yedeklenene kadar bekletilir. Bu durumda, log dosyası büyük hacimlere ulaşabilir. Bu seçeneğin tam verimli ve doğru kullanılabilmesi için loglar hatasız olarak yedeklenmelidir. Arada herhangi bir verinin kaybolursa, istenen zamana dönülemeyecektir. Sağlıklı bir veritabanı modeli için veritabanı yedeklemenin yanı sıra, log dosyaları da düzenli olarak yedeklenmelidir. Bunun için zamanlanmış plan kullanılabilir.

BULK-LOGGED RECOVERY MODEL

Adından da anlaşılacağı gibi, BULK işlemlerinde tercih edilebilecek bir modeldir. BCP ile yapılan yüklemeler, **SELECT INTO** gibi toplu işlemler loglanmaz. Full Recovery moduna sahip sistemde, log dosyasının büyümemesi için Full Recovery ile Bulk-Logged arasında geçiş yapılabilir.

Veritabanı loglama seviyesi SSMS ya da T-SQL ile ayarlanabilir.

T-SQL ile loglama seviyesini ayarlamak için aşağıdaki yöntem kullanılabilir.

```
USE master
GO
ALTER DATABASE AdventureWorks SET RECOVERY FULL;
```

SSMS ile loglama seviyesini ayarlamak için aşağıdaki yöntem kullanılabilir.

- Ayar yapmak için ilgili veritabanının üzerine fare ile sağ tıklayın.
- Properties menüsünden veritabanı özellikleri ekranına geçin.
- Options menüsü ile veritabanı ayarlarına girin.

Collation:	SQL_Latin1_General_CP1_CI_AS
Recovery model:	Full
Compatibility level:	Full Bulk-logged Simple
Containment type:	

YEDEKLEME TÜRLERİ

SQL Server farklı veritabanı yedekleme yöntemleri ile verinin yedeklenmesi gerçekleştirilebilir. Bu yedekleme yöntemleri aşağıdaki gibi üçe ayrılır.

TAM VERİTABANI YEDEĞİ (FULL DATABASE BACKUP)

Tüm veritabanının yedeğini almak için kullanılır. Yedekleme için çok kullanılan yöntemdir. Bir veritabanını Full Backup olarak yedekledikten sonra, tek bir yedek dosyası ile veritabanı tekrar kullanılabilir hale getirilebilir. Ancak, herhangi bir zamana geri dönmek için tek başına yeterli değildir.

FARK YEDEĞİ (DIFFERENTIAL BACKUP)

Tam yedeklemeden sonra yapılan değişikliklerin yedeğini almak için kullanılır. Bir veritabanının tam yedeği alındıktan sonra, her defasında tam yedek alınarak büyük veri alanı kaplamak yerine, bir kez ya da uzun aralıklarla tam yedek alınarak, sonraki yedeklerin tam yedek ile sonraki zaman arasındaki fark yedeği olması sağlanabilir.

Büyük hacimli bir veritabanında aylık tam yedek alındığı bir senaryo için durum şöyle gerçekleştirilebilir. Ayın son günü alınan tam yedekten sonra, her gün için bir fark yedeği alınır. Daha sonra bu fark yedeğinin bulunduğu günün verisine dönmek istendiğinde tam yedek ve ilgili günün fark yedeği yeterli olacaktır.

LOG YEDEĞİ (LOG BACKUP)

Veritabanında belli bir zamana dönebilmek için log yedekleri büyük öneme sahiptir. Full Backup işleminden sonraki tüm fark yedeklerinin bulunması gerekir. Fark yedeklerinin birinin bile bulunmaması durumunda, bu tarihten sonraki log yedeklerinin işlevini yitirmesine neden olur. Özetlemek ve kavramı anlamak için veritabanı yedeği ile log yedeklerinin bir bütünü oluşturduğunun bilinmesinde fayda var. İstenen tarihe dönebilmek için loglar büyük öneme sahiptir.

YEDEKLEME VE KURTARMA PLANI OLUŞTURMAK

Veri güvenliği için veritabanı ve logların yedeklenmesi gerekir. Yedekleme işlemiyse bir strateji ve planlama olmadan yapılmamalıdır. Strateji ve planlama olmadan yapılan yedekleme ve kurtarma planı, ani veri kayıplarında yapılan plansız yedeklemenin anlaşılması ve veri kaybını geri getirememe gibi büyük sorunlara yol açar.

Veritabanının başlıca görevlerinden biri olan yedekleme ve kurtarma planı oluşturmak, veritabanı yöneticisini iş hayatında seçkin hale getirebilecek kadar önemlidir. Bir geliştirici ya da veritabanı yöneticisi, diğer birçok alanda ne kadar uzman olursa olsun, yedekleme ve kurtarma için gerekli ileri seviye bilgi ve planlama yeteneğine sahip değilse, iş hayatında büyük sorunlarla karşılaşabilecektir.

BAŞLANGIÇ YEDEKLEME VE KURTARMA PLANLAMASI

Veritabanı planlama stratejisi geliştirmek ve planlamayı teknik anlamda hazırlamak belirli bir süreç içerisinde gerçekleşir. Planlamanın yapılabilmesi için veri, veritabanı ve donanım alt yapısını iyi tanımak gerekir. Aynı zamanda, veritabanı hakkında aşağıdaki soruların cevabını bilmek, doğru planlama için gereklidir.

VERİ NE KADAR ÖNEMLİ?

Veritabanının önemini ve ne sıklıkla yedeklenmesi gerektiğini belirleyen şey verinin kendisidir. Verinin önemi arttıkça, daha sık yedekleme ihtiyacı doğar. Geliştirme ortamındaki bir veritabanını, haftada bir ve log yedeği gerektirmeden yedekleyebilirken, anlık veri işlemi yüksek olan bir veritabanının günde bir ya da iki kez yedeklenmesi gerekebilir. Bazen bu da yeterli gelmeyerek aradaki zaman dilimi için fark yedekleri de almak gerekebilir. Veri önemi arttıkça yedek, fark yedeği, log yedeği ve bu yedeklerin önemi de artmaktadır.

YEDEKLENEN VERİTABANLARININ TÜRÜ NEDİR?

Yedeklenecek veritabanının türü ve hangi amaç için kullanıldığı planlama stratejisi oluşturmak için önemlidir. Örneğin; sık kullanılan bir veritabanı

ile yedek ya da test olarak kullanılan bir veritabanı kopyasının yedekleme stratejisi farklıdır. Hatta bazı veritabanları için yedekleme gereksinimi bile duyulmayabilir.

Veritabanı türüne göre kritiklik seviyesi değişecektir. Örneğin; bir kullanıcı veritabanı çok sık güncellenip üzerinde işlem yapılır. Bu durumda, veritabanı ve log yedeklemesi için farklı bir planlama yapılmalıdır. En kötü ihtimalle de veri kaybı gerçekleşir. Ancak, master veritabanı çok sık güncellenmemesine rağmen, bozulma ya da bir hata ile karşılaşırsa sonucu başarısız olur. Bazı durumlarda, master veritabanının onarılması ya da yedeğinin kullanılması gerekir.

VERİ NE KADAR ÇABUK KURTARILMALI?

Veri yedeğinin ne kadar sürede aktif hale getirilmesi gerektiği bilinmelidir. Kritik bir ortamdaki önemli bir veri yedeğinin aktif edilmesi söz konusu ise, verinin disk okuma hızı bile çok önemlidir. Büyük veri kümelerinde uzun sürecek veri kurtarma senaryosu birçok soruna sebep olabilir.

YEDEKLEMİYİ GERÇEKLEŞTİRECEK DONANIM VAR MI?

Yedekleme için ek donanıma ihtiyaç vardır. Sağlıklı ve hızlı veri kurtarma işlemi için hızlı ve güçlü donanımlar olmalıdır. Bu bir yedek sunucusu olabileceği gibi, bir disk, teyp ya da farklı bir aygıtta olabilir.

YEDEKLEME İÇİN EN UYGUN ZAMAN NEDİR?

Yedekleme işlemleri genel olarak yayında olan ve üzerinde sürekli işlem yapılan canlı sistemler üzerinde gerçekleştirilir. Kurumsal veritabanlarında, hafta sonu akşam saatleri işlem yoğunluğunun en az olduğu zamanlardır. Yedekleme işlemi, büyük hacimli verilerin bulunduğu sistemlerde uzun süreceği için yoğunluğun en az olduğu zamanlarda gerçekleştirilir. Ancak, her zaman az yoğun olduğumu durumda yedekleme istenmez. Yoğun veri işlemi olan durumlarda da yedekleme işlemi gerekebilir. Yedekleme planının bu duruma göre hazırlanması, yedeklemeden kaynaklanan olası gecikme ve sorunlara çözüm olacaktır.

YEDEKLEME, SIKIŞTIRILABİLİR Mİ?

SQL Server, veritabanı yedeklerinin sıkıştırılmasını destekler. Sıkıştırılan veritabanı yedeği disk üzerinde daha az yer kaplar. Daha az yer kaplaması nedeniyle daha az disk I/O işlemine sebep olur. Taşınabilirlik ve veri aktarımı açısından da yedeği sıkıştırmak faydalıdır. Daha az yer kaplaması nedeniyle, yedekleme sunucuları ve kullanılan yedek aygıtları, disk ya da teyp gibi donanımsal kaynaklarda az yer kaplar. Bu nedenle, yedekleme donanımı ihtiyacı azalacağı için, donanım maliyeti de azalacaktır. Ancak, sıkıştırma işlemi disk üzerinde faydalı olsa da, işlemci açısından daha fazla iş yükü getireceği gibi, yoğun işlemci gücü harcanmasına neden olur.

YEDEKLEMELERİ ALAN DIŞINDA SAKLAMAK GEREKİYOR MU?

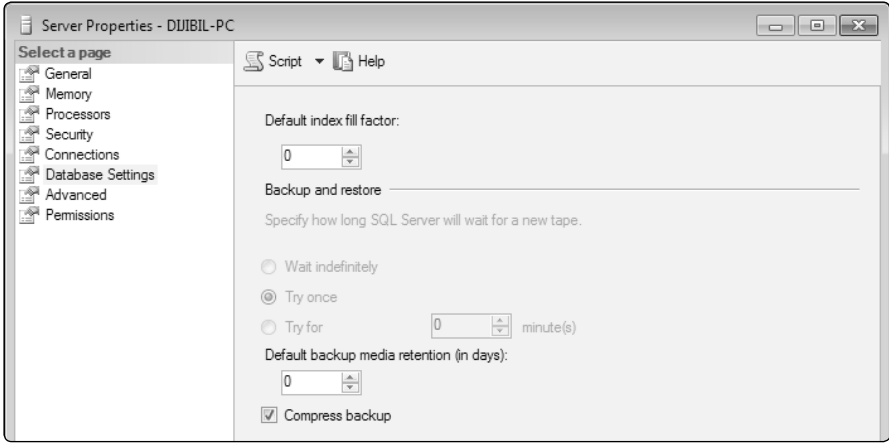
Performans ile ilgili bölümde detaylıca örneklendirdiğimiz bu konu fiziksel sorunlar ile ilgilidir. Verinin bulunduğu donanımsal sistemin doğal afet ya da benzeri fiziksel sorunlardan dolayı yayınına devam edememesi durumunda, ikizleme (*aynalama*) ya da yedekten veri kurtarma işlemlerinin gerçekleştirilmesi gerekir. Kritik verilerin bulunduğu sistemlerin karasal hat olarak farklı bağlantılarda, farklı şehir ya da ülkede yedeklerinin bulunması, olası birçok fiziksel sorunun önüne geçer. Fiziksel sorun meydana geldiğinde yedeklenen bu veritabanı ve yazılım yedekleri kullanılarak sistem tekrar ayağa kaldırılabilir.

YEDEKLEME SIKIŞTIRMASI PLANLAMAK

SQL Server'da yedek sıkıştırma ayarı varsayılan olarak devre dışıdır. Sıkıştırma işlemi iki şekilde gerçekleştirilebilir.

Management Studio ile yedek sıkıştırma özelliğini ayarlamak:

- **Management Studio** ekranındaki **Object Explorer** panelinden sunucu girdisini, farenin sağ düğmesiyle tıklayarak **Server Properties** iletişim kutusunu açın.
- Açılan ekranda **Database Settings** sayfasını açın. Sıkıştırma özelliğini açmak için, **Compress backup** seçeneğini seçili hale getirin. Aynı özelliği kapatmak için, aynı seçenekteki seçeneği kaldırın.



T-SQL ile yedek sıkıştırma özelliğini ayarlamak:

Yedek sıkıştırma ayarları programsal olarak yapılabilir. Bu işlem aşağıdaki gibi yapılabilir.

```
EXEC sp_configure 'backup compression default','1';
GO
RECONFIGURE WITH OVERRIDE;
GO
```

Yukarıdaki sorgu ile yedek sıkıştırma özelliği aktif edilir. İkinci parametre olarak verilen 1 değeri yerine, 0 değeri verilirse, yedek sıkıştırma özelliği pasif edilir.

Aynı işlemin, hem SSMS hem de T-SQL ile nasıl yapıldığını inceledik. T-SQL ile yapılan ayar değişikliği **Management Studio** ortamındaki seçim kutusunun seçili halinin değişmesine sebep olacaktır.

YEDEKLEMİYİ GERÇEKLEŞTİRMEK

Yedekleme işlemini gerçekleştirmek için yedekleme türleri gibi mimari birçok özelliği inceledik. Veritabanı yedekleme işleminin öneminden ve hangi şartlarda ne tür yedeklemeler gerektiği gibi konulara değindik. Artık veritabanı yedekleme işleminin nasıl gerçekleştirileceği incelenebilir.

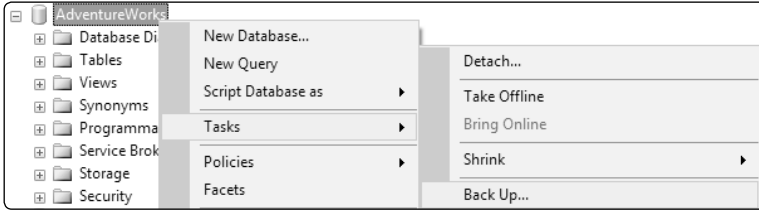
SQL SERVER MANAGEMENT STUDIO İLE YEDEK OLUŞTURMAK

Yedekleme işlemi, **Management Studio** ile kolay ve hızlı bir şekilde yapılabilir.

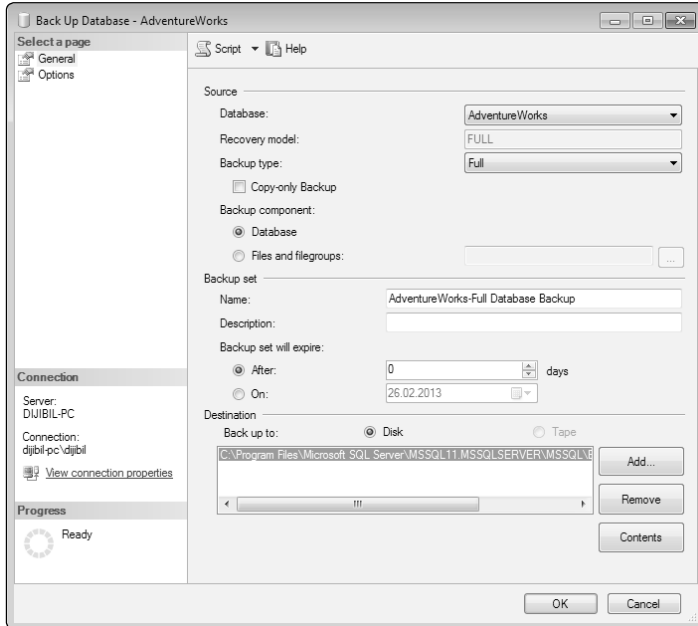
Management Studio ile yedekleme işlemi için aşağıdaki adımları takip edin.

Object Explorer paneliyle yedeklenmek istenen veritabanının üzerine fare ile sağ tıklayın.

- Açılan pencerede sırasıyla **Tasks > Back Up...** menülerini seçin.
- **Back Up Database** ekranında gerekli ayarlamaları yapıp, **OK** butonuna basılarak yedekleme başlatılabilir.



Yukarıdaki menüler seçildikten sonra, aşağıdaki **Back Up Database** paneli açılacaktır.



Management Studio ile oluşturulan yedekler, varsayılan olarak aşağıdaki dosya yolunda saklanır.

C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup

T-SQL İLE VERİTABANI YEDEĞİ OLUŞTURMAK

T-SQL ile veritabanını yedeklemek için aşağıdaki söz dizimi kullanılabilir.

Söz Dizimi:

```
BACKUP DATABASE veritabani_ismi
TO DISK = 'disk_yolu' [, DISK ='ikinci_disk_yolu',...]
[WITH MEDIANAME='medyaset_ismi'];
```

AdventureWorks veritabanının tam veritabanı yedeğini (**Full Database Backup**) alalım.

```
BACKUP DATABASE AdventureWorks
TO DISK='C:\Backups\AWorks1.bak';
```

Sorgunun başarılı çalışabilmesi için izin olarak belirtilen C:\ içerisinde Backups isimli klasör oluşturulmuş olmalıdır. Aynı zamanda, veritabanının ismi de tam olarak belirtilmelidir.

Bir sonraki alınan yedeğin, var olan dosyanın üstüne yazması için şu ifade kullanılabilir.

```
BACKUP DATABASE AdventureWorks
TO DISK='C:\Backups\AWorks1.BAK'
WITH INIT;
```

Tam veritabanı yedeği yerine, fark yedeği almak için şu ifade kullanılabilir.

```
BACKUP DATABASE AdventureWorks
TO DISK = 'C:\Backups\AWorks1.BAK'
WITH DIFFERENTIAL;
```

Fark yedeği almak istendiğinde, fark yedeği sorgusu sık kullanılacaktır.

Veritabanının birden fazla dosyaya yazdırılması için şu ifade kullanılabilir.

```
BACKUP DATABASE AdventureWorks
TO DISK='C:\Backups\AWorks1.BAK',
DISK='D:\Backups\AWorks2.BAK',
DISK='E:\Backups\AWorks3.BAK';
```

Veritabanının birden fazla parçaya ayrılması ve bu parçaların birden fazla fiziksel disk üzerinde saklanarak I/O performansı ve hızlı işlem için tercih edilebilecek bir yöntemdir.

Yedeklenen veritabanında, geri yükleme işleminde kullanılacak bir şifre oluşturmak için şu ifade kullanılabilir.

```
BACKUP DATABASE AdventureWorks
TO DISK = 'C:\Backups\AWorks1.BAK'
WITH PASSWORD = 'D!İ@J#İ$B#İ$L';
```

Veritabanı yedekleme işleminin yüzdesel olarak tamamlanma oranının gösterilmesini sağlamak için şu ifade kullanılabilir.

```
BACKUP DATABASE AdventureWorks
TO DISK = 'C:\Backups\AWorks4.BAK'
WITH STATS;
```

```
10 percent processed.
20 percent processed.
30 percent processed.
40 percent processed.
50 percent processed.
60 percent processed.
70 percent processed.
80 percent processed.
90 percent processed.
Processed 24792 pages for database 'AdventureWorks', file 'AdventureWorks2012_Data' on file 1.
100 percent processed.
Processed 2 pages for database 'AdventureWorks', file 'AdventureWorks2012_Log' on file 1.
BACKUP DATABASE successfully processed 24794 pages in 6.591 seconds (29.389 MB/sec).
```

Yukarıdaki sorgu çalıştırıldığında, **SSMS** sorgu ekranının **Messages** kısmında, varsayılan olarak her tamamlanan **%10** işlem için bildirim yapılır. **WITH STATS** aşağıdaki gibi kullanıldığı taktirde, farklı yüzde oranına göre de mesaj bildirimi yapılabilir.

```
BACKUP DATABASE AdventureWorks
TO DISK = 'C:\Backups\AWorks5.BAK'
WITH STATS = 2;
```

```
80 percent processed.
82 percent processed.
84 percent processed.
86 percent processed.
88 percent processed.
90 percent processed.
92 percent processed.
94 percent processed.
96 percent processed.
98 percent processed.
Processed 24792 pages for database 'AdventureWorks', file 'AdventureWorks2012_Data' on file 1.
100 percent processed.
Processed 2 pages for database 'AdventureWorks', file 'AdventureWorks2012_Log' on file 1.
BACKUP DATABASE successfully processed 24794 pages in 7.347 seconds (26.364 MB/sec).
```

Veritabanı yedeği için bir açıklama satırı eklenebilir. Bu işlem için şu ifade kullanılabilir.

```
BACKUP DATABASE AdventureWorks
TO DISK = 'C:\Backups\AWorks6.BAK'
WITH DESCRIPTION = 'AdventureWorks için Tam Yedek';
```

Yedekleme sırasında, birden fazla özellik tanımlayabilmek için şu ifade kullanılabilir. Bu sorguda, hem Mirror işlemi, hem de birden fazla özellik tanımlaması yapalım.

```
BACKUP DATABASE AdventureWorks
TO DISK = 'C:\Backups\AWorks7.BAK'
MIRROR TO DISK = 'C:\Backups\AdventureWorks_MIRROR.BAK'
WITH FORMAT, STATS, PASSWORD = 'D!i@J#i$B#i$L';
```

Mirror, hazırlanan yedeğin bir kopyasının daha oluşturulması için kullanılır. Bir yedek bozulursa, Mirror ile alınan diğer yedek kullanılabilir.

Veritabanı yedekleme işlemleri için gerekli söz dizimi oldukça farklı özellik ve seçeneklere sahiptir. İhtiyaçlara göre değişecek söz dizimi ile birçok farklı işlem kolaylıkla yapılabilir. Ancak hızlı yedekleme işlemlerinde Management Studio gibi yönetim araçları daha faydalı olacaktır.

Bir veritabanı yedeği dosyası hakkında bilgi almak gerekebilir. Yedek içerisinde bulunan, veritabanına ait dosya adları, uzantıları ve boyutu gibi bilgileri elde etmek için aşağıdaki sorgu kullanılabilir.

```
RESTORE FILELISTONLY
FROM DISK = 'C:\Backups\AWorks1.BAK';
```

	LogicalName	PhysicalName	Type	FileGroupName	Size	MaxSize	FileId	CreateLSN	DropLSN	Uniqueld
1	AdventureWorks2012_Data	C:\Program Files\Microsoft SQL Server\MSSQL11.MSS...	D	PRIMARY	214958080	35184372080640	1	0	0	40FA46CD-DFA9-40E1-90E5-5BE7CA6783EA
2	AdventureWorks2012_Log	C:\Program Files\Microsoft SQL Server\MSSQL11.MSS...	L	NULL	1835008	35184372080640	2	0	0	E16FAF18-F2EA-4BFB-9B8A-7F30D4F87B84

Genel olarak kullanılacak ve birçok farklı parametreyi içeren bir yedekleme sorgusu oluşturalım. Yedekleme işlemleri genel olarak, bir script olarak hazırlanır ve sürekli kullanılır.

```
BACKUP DATABASE AdventureWorks
TO DISK = N'C:\Backups\AWorks10.BAK'
WITH NOFORMAT, NOINIT,
NAME = N'AdventureWorks - Full Database Backup',
SKIP, NOREWIND, NOUNLOAD, COMPRESSION,
STATS = 10, CHECKSUM, CONTINUE_AFTER_ERROR
GO
DECLARE @BackupSetID AS INT SELECT @BackupSetID = position
FROM msdb..backupset
WHERE database_name = N'AdventureWorks'
AND
backup_set_id = (SELECT MAX(backup_set_id)
FROM msdb..backupset
WHERE database_name = N'AdventureWorks')
IF @BackupSetID IS NULL
BEGIN RAISERROR(N'Doğrulama başarısız! 'AdventureWorks2012'
bilgisi bulunamadı.', 16, 1)
END
RESTORE VERIFYONLY FROM DISK = N'C:\Backups\AWorks10.BAK'
WITH FILE = @BackupSetID,
NOUNLOAD,
NOREWIND
GO
```

Bu sorgu sonucunda, **Messages** alanındaki çıktı aşağıdaki gibi olacaktır.

```
10 percent processed.
20 percent processed.
30 percent processed.
40 percent processed.
50 percent processed.
60 percent processed.
70 percent processed.
80 percent processed.
90 percent processed.
Processed 24792 pages for database 'AdventureWorks', file 'AdventureWorks2012_Data' on file 4.
100 percent processed.
Processed 2 pages for database 'AdventureWorks', file 'AdventureWorks2012_Log' on file 4.
BACKUP DATABASE successfully processed 24794 pages in 3.716 seconds (52.126 MB/sec).
The backup set on file 4 is valid.
```

T-SQL İLE TRANSACTION LOG DOSYASI YEDEĞİ OLUŞTURMAK

Gerçek uygulamalarda, veritabanı yedeğinin yanı sıra, transaction log dosyasını da yedeklemeniz gereken birçok senaryo ile karşılaşılır. Söz dizimi olarak veritabanına benzer yapıdadır.

BACKUP LOG ile transaction log dosya yedeği oluşturabilmek için en az bir tam veritabanı yedeği (*Full Backup*) alınmış olmalıdır. Aşağıdaki sorguları çalıştırmadan önce bir tam veritabanı yedeği alın. Aksi halde **BACKUP LOG** sorguları hata üretecektir.

BACKUP LOG ile oluşturulan transaction log dosyasının varsayılan uzantısı **TRN**'dir.

AdventureWorks veritabanının transaction log dosya yedeğini oluşturalım.

```
BACKUP LOG AdventureWorks
TO DISK = 'C:\Backups\AWorks1.TRN';
```

Şifreli bir transaction log dosya yedeği oluşturalım.

```
BACKUP LOG AdventureWorks
TO DISK = 'C:\Backups\AWorks2.TRN'
WITH PASSWORD = 'D!i@J#i$B#i$L';
```

Transaction log dosya yedeği oluşturulurken, yedek işleminin tamamlanma yüzdesini görelim.

```
BACKUP LOG AdventureWorks
TO DISK = 'C:\Backups\AWorks3.TRN'
WITH STATS;
```

Yukarıdaki kullanım, varsayılan olarak %10 tamamlanma yüzdesi ile gerçekleşir.

%10 tamamlanma yüzdesini değiştirerek %1 yapalım.

```
BACKUP LOG AdventureWorks
TO DISK = 'C:\Backups\AWorks4.TRN'
WITH STATS = 1;
```

Transaction log dosyası yedeğine açıklama ekleyelim.

```
BACKUP LOG AdventureWorks
TO DISK = 'C:\Backups\AWorks5.TRN'
WITH DESCRIPTION = 'AdventureWorks transaction log yedeği';
```

Yedekleme sırasında, birden fazla özellik tanımlayabilmek için şu ifade kullanılabilir. Bu sorguda, hem Mirror işlemi, hem de birden fazla özellik tanımlaması yapalım.

```
BACKUP LOG AdventureWorks
TO DISK = 'D:\Backups\AWorks6.TRN'
MIRROR TO DISK = 'D:\AdventureWorks_mirror.TRN'
WITH FORMAT;
```

Mirror, hazırlanan yedeğin bir kopyasının daha oluşturulması için kullanılır. Bir yedek bozulursa, Mirror ile alınan diğer yedek kullanılabilir.

SQL AGENT İLE

OTOMATİK YEDEKLEME PLANI OLUŞTURMAK

Veritabanı yedekleme işlemi, insan faktörüne bırakılmayacak kadar önemli ve düzenli gerçekleştirilmesi gereken bir görevdir. SQL Agent ile yedekleme işini, veritabanının otomatik olarak gerçekleştirmesi sağlanabilir.

SQL Agent, SQL Server'dan ayrı olarak çalışan özel bir servise sahiptir. Bu servisi başlatmak için aşağıdaki adımları izleyin.

- **Başlat** menüsünde, SQL Server araçları arasında **SQL Server Configuration Manager** aracını açın.
- **SQL Server Agent** servisi üzerine fare ile sağ tıklayın ve özelliklerine girin.
- **Log On tab** menüsü içerisinde, fare ile **Start** butonuna tıklayın.
- **Service tab** menüsü içerisinde, fare ile **Start Mode** olarak **Automatic** seçili hale getirin.
- İlk olarak **Uygula**, daha sonra **Tamam** butonuna fare ile tıklayarak işlemi tamamlayın.

SQL Server Configuration Manager içerisinde değerler aşağıdaki gibi olmalıdır.

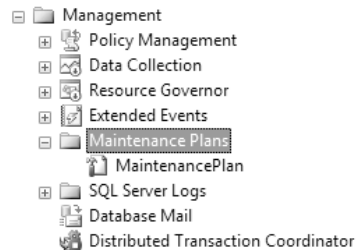
 SQL Server Agent (MSSQLSERVER) Running Automatic

Bu ayarları yaptıktan sonra, sistem yeniden başlasa bile, SQL Agent servisi sürekli ve otomatik olarak çalıştırılacaktır. Veritabanı yedekleme işlemini bu servisin otomatik olarak yapabilmesi için bu ayarın yapılmış olması gerekir.

SQL Agent servisini açtıktan sonra, **Agent XPs** ayarlarının da açılması gerekir. Açık değil ise, aşağıdaki sorgu ile bu ayarlar açılabilir.

```
EXEC sys.sp_configure N'show advanced options',N'1';
RECONFIGURE WITH OVERRIDE;
EXEC sys.sp_configure N'Agent XPs', N'1';
RECONFIGURE WITH OVERRIDE;
EXEC sys.sp_configure N'show advanced options', N'0';
RECONFIGURE WITH OVERRIDE;
```

SQL Agent ile çalıştırılacak bir yedekleme planı oluşturmak için, **Management Studio**'da **Object Explorer** panelindeki **Management** klasörü içerisinde bulunan Maintenance Plans sekmesi fare ile sağ tıklanarak açılır. Açılan menüden, Maintenance Plan Wizard seçilerek yedekleme planı oluşturmaya başlanır.



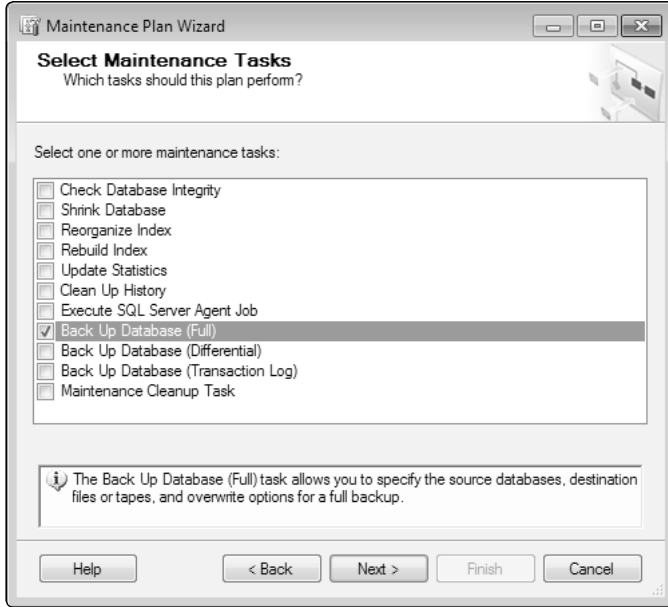
Açılan **Select Plan Properties** ekranını aşağıdaki gibi hazırlayın.

The screenshot shows the 'Maintenance Plan Wizard' window, specifically the 'Select Plan Properties' step. The window title is 'Maintenance Plan Wizard'. The main heading is 'Select Plan Properties' with the subtitle 'How do you want to schedule your maintenance tasks?'. The 'Name' field is 'MaintenancePlan1'. The 'Description' field is empty. The 'Run as' dropdown is set to 'SQL Server Agent service account'. There are two radio buttons: 'Separate schedules for each task' (unselected) and 'Single schedule for the entire plan or no schedule' (selected). The 'Schedule' dropdown is set to 'Not scheduled (On Demand)'. There is a 'Change...' button next to the schedule dropdown. At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

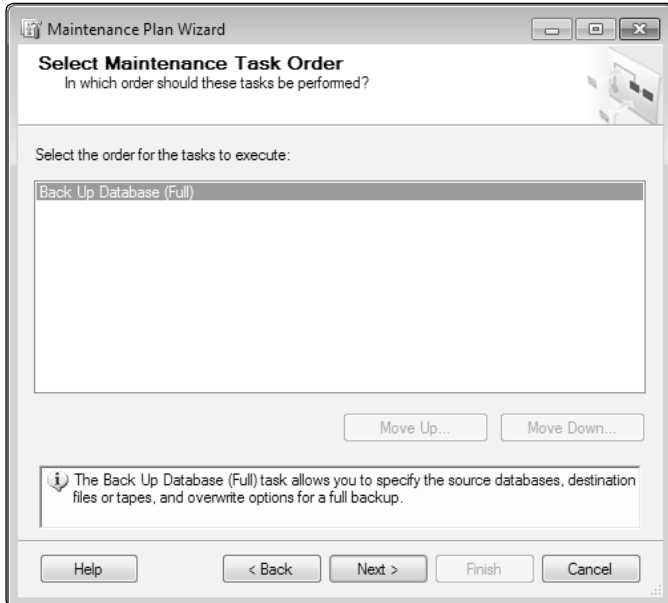
Ekranın Shedule kısmındaki **Change** butonuna tıklayarak, hangi zaman ve tarih aralığında yedekleme yapılacağını, yedeğin günlük, haftalık ya da aylık mı olduğunu ve hangi saatlerde yapılması gerektiği gibi detay bilgileri belirlenebilir.

The screenshot shows the 'New Job Schedule' dialog box. The 'Name' field is 'MaintenancePlan1'. The 'Schedule type' dropdown is set to 'Recurring'. The 'Jobs in Schedule' button is visible. The 'One-time occurrence' section has 'Date' set to '26.02.2013' and 'Time' set to '10:29:32'. The 'Frequency' section has 'Occurs' set to 'Weekly'. The 'Recurs every' section is set to '1 week(s) on'. The days of the week are listed with checkboxes: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday (checked). The 'Daily frequency' section has 'Occurs once at' set to '00:00:00'. The 'Duration' section has 'Start date' set to '26.02.2013' and 'End date' set to '26.02.2013'. The 'Summary' section has a description: 'Occurs every week on Sunday at 00:00:00. Schedule will be used starting on 26.02.2013.' At the bottom, there are buttons for 'OK', 'Cancel', and 'Help'.

Sonraki **Select Maintenance Tasks** ekranında **Back Up Database (Full)** seçim kutusu seçili hale getirilir.



Sonraki **Select Maintenance Task Order** ekranını **Next** ile geçin.



Sonraki, **Define Back Up Database (Full) Task** ekranındaki **Databases** seçim alanında **These databases**'i seçerek **AdventureWorks** veritabanını seçin ve **OK** düğmesine tıklayın.

Bu ekranda, yedeğin sıkıştırılma ayarı, dosya yolu, uzantısı, yedeğin diske mi teyp'e mi yapılacağı gibi birçok ayar değiştirilebilir.

Maintenance Plan Wizard

Define Back Up Database (Full) Task
Configure the maintenance task.

Backup type: Full

Database(s): <Select one or more>

Backup component

☒ Database

☐ Files and filegroups: ...

☐ Copy-only Backup

☐ For availability databases, ignore Replica Priority for Backup and Backup on Primary Settings

☐ Backup set will expire:

☒ After 14 days

☐ On 12.03.2013

Back up to: ☒ Disk ☐ Tape

☒ Back up databases across one or more files:

...

Add...

Remove

Contents

If backup files exist: Append

☒ Create a backup file for every database

☐ Create a sub-directory for each database

Folder: C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup

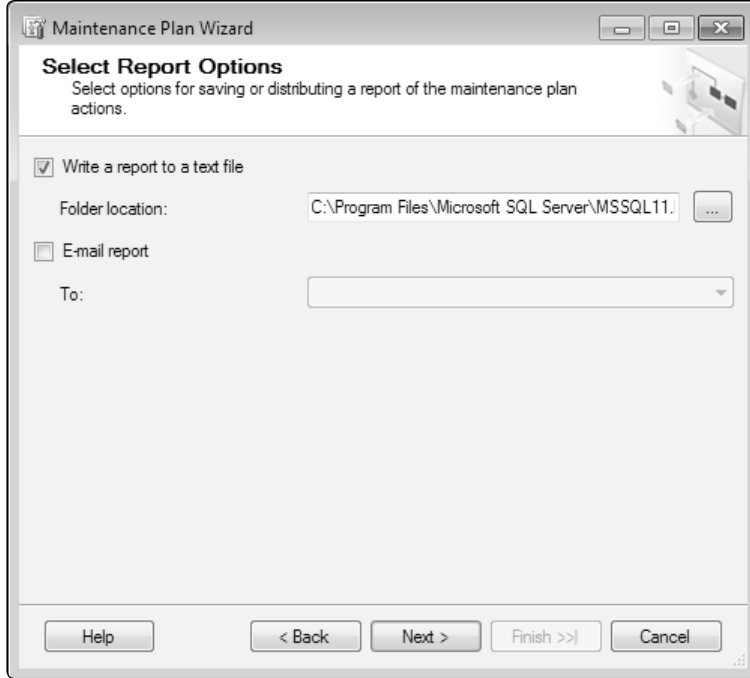
Backup file extension: bak

☐ Verify backup integrity

Set backup compression: Use the default server setting

Help < Back Next > Finish >> Cancel

Son olarak, **Select Report Options** ekranında “*Write a report to a text file*” seçeneği seçilerek, yedekleme işlemi raporunun bir metin dosyasına yazılması sağlanabilir. Eğer gerekli ayarlar yapıldıysa, raporun e-mail ile gönderilmesi de sağlanabilir.



Tüm işlemler tamamlandıktan sonra **Comple the Wizard** ekranında, yedekleme planı ile ilgili özet ayarlar incelenebilir.

Plan oluşturulduktan sonra, **Object Explorer**'daki **Maintenance Plans** içerisinde oluşturulan plan ismine sağ tıklayarak, View **History** menüsü seçildiğinde, yedekleme işleminin yapılıp yapılmadığı ve karşılaşılan sorunların takibi yapılabilir.

VERİTABANINI GERİ YÜKLEMELİK

Veritabanından alınan yedeklerin geri yüklemesi için **Management Studio** ve **T-SQL** olmak üzere iki yöntem kullanılabilir.

T-SQL GERİ YÜKLEME KOMUTLARINI KULLANMAK

T-SQL ile geri yükleme komutlarını kullanabilmek için söz dizimi özelliklerini bilmeniz gerekir. Temel anlamda geri yükleme komutları kullanımı basittir. Ancak, istek ve geri yükleme gereksinimleri bazen karmaşık sorgular oluşturmayı gerektirebilir.

- **FORMAT**: Yedek setinde bölme ve yedek dosyaları üzerinde yazma işlemini sağlar.
- **INIT** ve **NOINIT**: Yedek üzerine yazma ya da ekleme ayarları için kullanılır. SQL Server'da varsayılan özellik **NOINIT**'dir. **NOINIT** ile yedekleme, mevcut yedek dosyası ve setine ekleme olarak yapılır. **INIT** ile yedekleme, mevcut yedek verinin üzerine yazılmasını sağlar.
- **RESTART**: SQL Server'ın yedekleme işleminin kesildiği yerden devam etmesini sağlar.
- **UNLOAD**: Yedekleme işleminden sonra, teybi geriye sarıp çıkarır. Varsayılan kullanımdır.
- **NOUNLOAD**: Yedekleme işleminden sonra, teybi geriye sarmaz ve çıkarmaz. **UNLOAD**'ın tersidir.
- **BLOCKSIZE**: Teyp üzerindeki fiziksel blok büyüklüğünü ayarlar.
- **SKIP**: Teyp üzerinde yer alan ANSI etiket bilgilerini atlar. Bu etiketler, teyp kullanım zamanının dolması, yazma gibi bilgileri içerir.
- **NOSKIP**: Varsayılan olarak SQL Server, ANSI teyp etiketlerini okur. Bu özellik, etiketlerin okunmaması istendiğinde kullanılır.

AdventureWorks veritabanını silelim ve yedekten geriye dönerek tekrar elde edelim. Bu işlemten önce, ilk olarak, **AdventureWorks** veritabanına ait bir tam veritabanı yedeğine (Full Database Backup) sahip olduğunuza emin olun. Geri yükleme işlemi bu yedek ile gerçekleştirilecek.

Şimdi, **AdventureWorks** veritabanını silin.

```
DROP DATABASE AdventureWorks;
```

Veritabanı dosyalarının bulunduğu disk üzerindeki klasörde **AdventureWorks** veritabanına ait MDF ve LDF dosyalarının bulunmadığından emin olun. Dosyalar var ise, silin ya da farklı bir klasöre taşıyın.

```
RESTORE DATABASE AdventureWorks  
FROM DISK = N'C:\Backups\AWorks10.BAK';
```

Bu işlemden sonra **Management Studio**'daki **Object Explorer** panelinde, yeni bir veritabanının oluştuğu görülebilir.