

HANSER

# **Microsoft SQL Server 2005 für Administratoren**

Irene Bauder

ISBN 3-446-22800-4

Leseprobe

Weitere Informationen oder Bestellungen unter  
<http://www.hanser.de/3-446-22800-4> sowie im Buchhandel

### 10.2.4 Die Transact-SQL-Anweisung „BACKUP“

Natürlich ist es auch möglich, mit Transact-SQL-Anweisungen eine Sicherung durchzuführen. Die Anweisung „BACKUP DATABASE“ kann sowohl eine vollständige als auch eine differenzielle Sicherung für die gesamte Datenbank, für Datengruppen oder für Dateien erstellen.

Die Datenbank „Firma“ wird bei Ausführung der ersten Zeile vollständig in die Datei „Firma\_1.dat“ archiviert. Die zweite Zeile verwendet für die Sicherung ein zuvor eingerichtetes logisches Sicherungsmedium.

```
BACKUP DATABASE Firma TO DISK = 'c:\sqlbackup\Firma_1.dat'  
BACKUP DATABASE Firma TO FullBackupdevice
```

Das nächste Beispiel führt eine differenzielle Archivierung durch. Die Änderungen werden in das zuvor angelegte Sicherungsmedium „FirmaMedium“ geschrieben.

```
BACKUP DATABASE Firma TO FirmaMedium WITH DIFFERENTIAL
```

Die Informationen am Anfang des Sicherungsmediums werden über den Zusatz „FORMAT“ festgelegt. Dadurch wird der komplette bisherige Inhalt gelöscht. Eine kurze Beschreibung des Sicherungssatzes wird über „DESCRIPTION“ bestimmt. Der Fortschritt der Sicherung („STATS“) wird alle 25 Prozent angezeigt.

```
BACKUP DATABASE AdventureWorks TO FullBackupdevice  
WITH FORMAT,  
MEDIANAME = 'Medium für AdventureWorks',  
MEDIADescription = 'Vollst. Sicherungen',  
DESCRIPTION = 'Sicherungssatz Nr1',  
STATS = 25
```

Die Meldungen, die während und am Ende der Sicherung ausgegeben werden, können folgendermaßen aussehen

```
25 Prozent verarbeitet.  
50 Prozent verarbeitet.  
75 Prozent verarbeitet.  
21024 Seiten wurden für die 'AdventureWorks'-Datenbank, Datei 'AdventureWorks_Data' für  
Datei 1, verarbeitet.  
100 Prozent verarbeitet.  
5 Seiten wurden für die 'AdventureWorks'-Datenbank, Datei 'AdventureWorks_Log' für Datei 1,  
verarbeitet.  
BACKUP DATABASE hat erfolgreich 21029 Seiten in 34.849 Sekunden verarbeitet (4.943  
MB/s).
```

Um zwei Sicherungsgeräte parallel zu nutzen, definieren Sie am besten zuerst die beiden logischen Sicherungsmedien und verwenden diese dann in der „BACKUP“-Anweisung.

```
EXEC sp_addumpdevice 'disk', 'AWGeraet1', 'C:\SQLBackup\AWGeraet1.bak'  
EXEC sp_addumpdevice 'disk', 'AWGeraet2', 'C:\SQLBackup\AWGeraet2.bak'
```

```
BACKUP DATABASE AdventureWorks TO AWGeraet1, AWGeraet2
```

Durch die nächste Anweisung wird nicht die komplette Datenbank, sondern nur eine Dateigruppe mit dem logischen Namen „Zweite\_Gruppe“ als Backup abgelegt.

```
BACKUP DATABASE Firma FILEGROUP = 'Zweite_Gruppe' TO FirmaMedium
```

Anstelle einer kompletten Dateigruppe ist es auch möglich, nur eine einzelne Datei zu archivieren. In diesem Fall muss der logische Name der Datei genannt werden.

```
BACKUP DATABASE Bestellverwaltung FILE = 'Bestell2' TO FullBackupdevice
```

Daneben können Sie natürlich über diverse Parameter auch alle Optionen angeben, die Sie im Archivierungs-Dialogfeld des SQL Server Management Studios gesehen haben.

Um eine Sicherung zu spiegeln, schreiben Sie zusätzlich die „MIRROR TO“-Klausel. Im nächsten Beispiel werden auf vier verschiedene Bandgeräte dieselben Daten archiviert.

```
BACKUP DATABASE Firma
TO TAPE = '\\.\tape0'
MIRROR TO TAPE = '\\.\tape1'
MIRROR TO TAPE = '\\.\tape2'
MIRROR TO TAPE = '\\.\tape3'
WITH FORMAT, MEDIANAME = 'FirmaSatz1'
```



Um unautorisierten Zugriff auf die Backup-Dateien zu verhindern, können Sie zusätzlich über das Argument „PASSWORT“ ein Kennwort für den Sicherungssatz vergeben. In diesem Fall muss auch das Kennwort bei der Wiederherstellung angegeben werden.

Für die Archivierung des Transaktionsprotokolls gibt es eine eigene SQL-Anweisung, die „BACKUP LOG“ lautet. Die Syntaxen von „BACKUP LOG“ und „BACKUP DATABASE“ sind sehr ähnlich.

```
BACKUP LOG Firma TO FirmaMedium
```

```
BACKUP LOG Firma TO FirmaMedium WITH NO_TRUNCATE
```

Wenn eine Platte, auf der Daten der Datenbank gespeichert sind, defekt ist, und auf diese Daten kein Zugriff mehr erfolgen kann, sollte vor der Wiederherstellung unbedingt erst noch das aktuelle Transaktionsprotokoll gesichert werden. In diesem Fall darf es aber nicht gekürzt werden. Dies kann durch die zusätzlich Angabe „WITH NO\_TRUNCATE“ erreicht werden.

### Beispiel zum Sicherungsablauf

Nachfolgend finden Sie eine typische Reihenfolge von Sicherungen, die für eine Datenbank ausgeführt werden. Über mehrere „INSERT“-Befehle werden zwischendurch neue Datensätze eingefügt. In der Praxis würde das Bearbeiten der Datensätze von verschiedenen Anwendern durchgeführt werden. Es wird mit einer vollständigen Sicherung begonnen, das Transaktionsprotokoll wird regelmäßig gesichert, und es erfolgt auch eine differenzielle Archivierung.

```
-- logisches Sicherungsmedium einrichten
EXEC sp_addumpdevice 'disk', 'FirmaAdminSG', 'C:\SQLBackup\FirmaAdmin1.bak'
```

```
-- vollständige Sicherung mit Initialisierung des Mediums
BACKUP DATABASE FirmaAdmin TO FirmaAdminSG
WITH FORMAT,
MEDIANAME = 'Medium für FirmaAdmin',
MEDIADescription = 'verschiedene Sicherungen für DB FirmaAdmin',
DESCRIPTION = 'Sicherungssatz Voll1'

-- Neue Daten werden durch Anwender eingefügt
USE FirmaAdmin
INSERT INTO Mitarbeiter VALUES('601','Huber','Otto','supp',2300,NULL,NULL)
INSERT INTO Mitarbeiter VALUES('602','Huber','Werner','supp',2300,NULL,NULL)
INSERT INTO Mitarbeiter VALUES('603','Huber','Peter','supp',2300,NULL,NULL)
INSERT INTO Mitarbeiter VALUES('604','Huber','Paul','supp',2300,NULL,NULL)

-- Sicherung des Transaktionsprotokolls
BACKUP LOG FirmaAdmin TO FirmaAdminSG
WITH DESCRIPTION = 'Sicherungssatz Log1'

-- Neue Daten werden durch Anwender eingefügt
INSERT INTO Mitarbeiter VALUES('701','Meier','Otto','kuec',2300,NULL,NULL)
INSERT INTO Mitarbeiter VALUES('702','Meier','Werner','kuec',2300,NULL,NULL)
INSERT INTO Mitarbeiter VALUES('703','Meier','Peter','kuec',2300,NULL,NULL)
INSERT INTO Mitarbeiter VALUES('704','Meier','Paul','kuec',2300,NULL,NULL)

-- differenzielle Sicherung
BACKUP DATABASE FirmaAdmin TO FirmaAdminSG
WITH DIFFERENTIAL, DESCRIPTION = 'Sicherungssatz Diff1'

-- Neue Daten werden durch Anwender eingefügt
INSERT INTO Mitarbeiter VALUES('801','Schmitt','Otto','vert',2300,NULL,NULL)

-- Sicherung des Transaktionsprotokolls
BACKUP LOG FirmaAdmin TO FirmaAdminSG
WITH DESCRIPTION = 'Sicherungssatz Log2'

-- Neue Daten werden durch Anwender eingefügt
INSERT INTO Mitarbeiter VALUES('901','Kenner','Otto','vert',2300,NULL,NULL)

-- Sicherung des Transaktionsprotokolls
BACKUP LOG FirmaAdmin TO FirmaAdminSG
WITH DESCRIPTION = 'Sicherungssatz Log3'
```

### 10.2.5 Sichern unter Einsatz des massenprotokollierten Wiederherstellungsmodells

Mithilfe des massenprotokollierten Wiederherstellungsmodells, das bestimmte Operationen nur minimal protokolliert, können einige Operationen effizienter durchgeführt werden als mit dem vollständigen Wiederherstellungsmodell, das jede Operation in voller Länge aufzeichnet.

Beim massenprotokollierten Wiederherstellungsmodell werden nur die Blöcke protokolliert, die während einer Operation geändert wurden. Dadurch wird das Transaktionsprotokoll klein gehalten. Um die Operation wiederherstellen zu können, muss das Transaktionsprotokoll sofort nach der Fertigstellung einer massenprotokollierten Operation gesichert werden.

Wenn für die Datenbank das massenprotokollierte Wiederherstellungsmodell eingestellt ist, sichert der MS SQL Server das Transaktionsprotokoll in zwei Schritten.

- Es werden alle Blöcke, die durch die Massenkopieroperation modifiziert wurden, archiviert. Dies ist der große Unterschied zum vollständigen Wiederherstellungsmodell.
- Das Transaktionsprotokoll selber wird wie beim vollständigen Wiederherstellungsmodell gesichert.

Das massenprotokollierte Wiederherstellungsmodell ermöglicht bestimmten Operationen eine schnellere Durchführung mit minimaler Protokollierung, da sich die geänderten Blöcke nur über ein Bitmap gemerkt werden. Dadurch werden jedoch die Wiederherstellungsmöglichkeiten begrenzt. Es ist keine Wiederherstellung bis zu einem bestimmten Zeitpunkt möglich. Wenn auf den Datenteil einer Datenbank nicht mehr zugegriffen werden kann, weil zum Beispiel die Festplatte defekt ist, kann nach einer Massen-Operation keine Transaktionsprotokollsicherung erfolgen.

Durch das massenprotokollierte Wiederherstellungsmodell werden folgende Operationen nur minimal protokolliert:

- Erstellung oder Neuaufbau eines Index
- Massensladen von Daten mit der SQL-Anweisung „BULK INSERT“, mit der SQL Server Integration Services-Aufgabe „Bulk Insert“ und mit dem zeilenbasierten Programm Bcp.
- Ausführung der Anweisung „SELECT INTO“, wenn dadurch permanente Tabellen angelegt werden.
- Ausführung der Anweisungen „WRITETEXT“ und „UPDATETEXT“

### Beispiel zur Verdeutlichung

Häufig ist es am besten, wenn Sie bei bestimmten Operationen zwischen dem vollständigen und dem massenprotokollierten Wiederherstellungsmodell wechseln. Dabei darf nicht vergessen werden, sofort nach einer Massenkopieroperation das Transaktionsprotokoll zu archivieren, um sicherzustellen, dass alle Daten wiederherstellbar sind. Als Verdeutlichung soll das folgende Beispiel unter der Annahme, dass das dem massenprotokollierten Wiederherstellungsmodell eingestellt ist, dienen.

- 12:00 Uhr: Transaktionsprotokoll wird (stündlich) gesichert.
- 12:10 Uhr: Massenkopieroperation beginnt.
- 12:20 Uhr: Massenkopieroperation ist fertig.
- 12:47 Uhr: Datenbank wird aufgrund eines Festplattenfehlers fehlerverdächtig.
- 12:50 Uhr: Sie stellen fest, dass auf die Datenbank nicht mehr zugegriffen werden kann. Sie versuchen nun, das Transaktionsprotokoll zu sichern, erhalten aber die Fehlermeldung, dass die minimal mitgeschriebenen Operationen nicht archiviert werden können, da die Datenbank nicht verfügbar ist.

Um 12:50 Uhr können Sie somit nur die Datenbank und die Transaktionsprotokollsicherungen bis zu 12:00 Uhr zur Wiederherstellung verwenden. Alle Datenänderungen zwischen 12:00 Uhr und 12:50 Uhr sind verloren. Hätten Sie um 12:20 Uhr direkt nach der fertig ausgeführten Massenkopieroperation das Transaktionsprotokoll gesichert, wäre die Datenbank nun nicht in einem massenprotokollierten Zustand.

Nur falls keine Massenkopieroperation bei gesetztem massenprotokollierten Wiederherstellungsmodell stattgefunden hat, werden nur das Transaktionsprotokoll und nicht auch die modifizierten Blöcke archiviert.

Wenn das Transaktionsprotokoll sofort nach der Massenkopieroperation gesichert wird, setzt man genau genommen den massenprotokollierten Zustand zurück, wodurch Transaktionsprotokollsicherungen wieder ohne Zugriff auf den Datenteil der Datenbank ausgeführt werden können. Dann hätte auch noch das Transaktionsprotokoll um 12:50 Uhr archiviert werden können. Es wäre kein Datenverlust entstanden.

Es hätte auch passieren können, dass die Datenbank während der Massenkopieroperation um 12:15 Uhr fehlerhaft geworden wäre. Wie Sie wissen, können Sie das Transaktionsprotokoll in diesem Zustand nicht mehr sichern. Dies bedeutet, dass alle neuen und geänderten Daten zwischen 12:00 Uhr und 12:15 Uhr verloren sind. Um dies zu verhindern, hätten Sie eine Transaktionsprotokollsicherung um 12:10 Uhr direkt vor der Massenkopieroperation durchführen können. Die Daten zwischen 12:00 Uhr und 12:10 Uhr wären somit gesichert. Diese Sicherung könnte benutzt werden, um wieder auf den Stand 12:10 Uhr zu kommen. Anschließend könnte erneut die Massenkopieroperation ausgeführt werden.

### **Sinnvolle Vorgehensweise**

Aus diesen Beispielen ist klar zu erkennen, dass das Transaktionsprotokoll unbedingt direkt vor und sofort nach einer Massenkopieroperation gesichert werden sollte. Nur somit kann ein möglicher Datenverlust minimiert werden. Da Sie als Datenbankadministrator aber gar nicht wissen können, ob nicht noch jemand anders zum Beispiel die SQL-Anweisung „SELECT INTO“ benutzt, und Sie somit nicht wissen, wann eine solche Archivierung erfolgen soll, ist es am sinnvollsten, standardmäßig das vollständige Wiederherstellungsmodell einzustellen. Nur wenn Sie eine Massenkopieroperation starten, wechseln Sie in das massenprotokollierte Wiederherstellungsmodell.

Am besten ist es, einen Batch zu schreiben, in dem die Wechsel zwischen den beiden Wiederherstellungsmodellen und die Massenkopieroperation programmiert sind. Bevor in diesem Batch zum massenprotokollierten Wiederherstellungsmodell gewechselt wird, sollte das Transaktionsprotokoll archiviert werden. Nach der erfolgreich durchgeführten Kopieraktion erfolgt wieder ein Wechsel zum vollständigen Wiederherstellungsmodell und anschließend erneut eine Transaktionsprotokollsicherung.

### Beispielskript

Das nachfolgende Skript zeigt die eben beschriebene Vorgehensweise. Zuerst wird eine Test-Datenbank erstellt.

```
CREATE DATABASE WModellDB
```

1. Vollständiges Backup durchführen

```
BACKUP DATABASE WModellDB TO DISK = 'c:\backup\WModell.bak' WITH NOINIT
```

2. Transaktionsprotokollsicherung durchführen

```
BACKUP LOG WModellDB TO DISK = 'c:\backup\WModell.bak' WITH NOINIT
```

3. Wiederherstellungsmodell auf massenprotokolliert setzen

```
IF DATABASEPROPERTYEX('WModellDB', 'Recovery') = 'FULL'  
ALTER DATABASE WModellDB SET RECOVERY BULK_LOGGED
```

4. Massenkopieroperation durchführen

5. Wiederherstellungsmodell auf vollständig setzen

```
IF DATABASEPROPERTYEX('WModellDB', 'Recovery') = 'Bulk_Logged'  
ALTER DATABASE WModellDB SET RECOVERY FULL
```

6. Erneut eine Transaktionsprotokollsicherung durchführen, die recht groß sein kann

```
BACKUP LOG WModellDB TO DISK = 'c:\backup\WModell.bak' WITH NOINIT
```

## 10.3 Wiederherstellen von Daten

Beim MS SQL Server gibt es genau genommen zwei Wiederherstellungsprozesse:

- Automatische Wiederherstellung
- Manuelle Wiederherstellung

### Automatische Wiederherstellung

Die automatische Wiederherstellung erfolgt bei jedem Start des MS SQL Server-Systems. Sie können sie über bestimmte Ablaufverfolgungsflags nur teilweise deaktivieren (siehe 3.3.2). Es wird normalerweise dabei immer dieselbe Anzahl von Schritten ausgeführt, unabhängig davon, wie das System das letzte Mal beendet wurde. Dadurch wird garantiert, dass sich anschließend der MS SQL Server in einem logischen konsistenten Zustand befindet.

Im ersten Schritt des Wiederherstellungsprozesses lokalisiert der MS SQL Server die beiden Dateien der „Master“-Systemdatenbank, die „master.mdf“ und „mastlog.ldf“ heißen. Dazu sieht er in der Registrierungsdatenbank von Windows unter dem Schlüssel „HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\Parameters“ nach. (Die Nummer ist davon abhängig, wie viele Instanzen auf dem Rechner installiert sind.)