

SQL Server Backup Strategien

Thomas Reinwart

Einleitung

Einen Datenverlust kann sich heutzutage keiner mehr leisten, zu schnell ist man in den Medien mit negative Schlagzeilen konfrontiert. Die Szenarien eines Datenverlustes können Vielfältig sein, eines klassischen Hardware Defekts bis zum Hackerangriff mit Datenmanipulation. Im Falle des Falls muss ein Notfallplan parat stehen, der muss funktionieren. Was der SQL Server für Möglichkeiten von Backup und Recovery bietet und welche zusätzlichen Tool, dafür angeboten werden sehen wir uns nun an.

Was bedeutet „downtime“

Wenn die Datenbank verfügbar ist, man online aber nicht erreichbar, sei es durch Netzwerk oder Server Ausfälle, ist man aus Anwender Sicht nicht Online. Es muss festgelegt sein, ob ein Wartungsfenster zur Verfügbarkeit zählt. Oft wird eine hohe Verfügbarkeit von 90% und mehr verlangt, was bedeutet das umgerechnet in Tagen auf das Jahr? Eine Verfügbarkeit von 99% entsprechen 3 ½ Tage Downtime im Jahr. Militärische Systeme kommen bis zu 99,9999% Verfügbarkeit.

Datenbank Sicherung Strategien

Um ein Sicherungskonzept festzulegen spielen 2 Punkte eine Rolle: Datenmenge und Wiederherstellungszeitraum. Auch sollte man Abwiegen, ob der Wiederherstellungszeitraum tatsächlich so enorm wichtig ist, ist es das Tagesgeschäft eines Unternehmens und steht man damit im online im Internet, die Buchhaltung die auf jeden Cent stimmen sollte oder handelt es sich um eine Applikation bei der eben ein paar Datensätze einfach nochmals erfasst werden müssen.

Abhängig von der Applikation bevorzugt man wohl immer den kleinsten möglichen Ausfall, was aber eben bei großen Datenmengen unter Umständen nicht mehr so einfach zu bewerkstelligen ist und auch mit Kosten an zusätzlicher Hardware, wie Platz auf neuen Medien oder Ausbau des Netzwerks, um die Datenmengen erst überhaupt transportieren zu können, verbunden sein kann.

Ablage, Auswahl des Sicherungsmediums

Im Zusammenhang mit der Backup Strategie ist auch die Frage der Aufbewahrung zu klären, wiederrum geht es um die Datenmenge, die in einem Zeitraum übertragen werden muss. Die Datensicherung am selben Rechner und Festplatte abzulegen ist eine ganz schlechte Idee. Vernünftiger ist es, die Daten übers Netzwerk auf eine andere Festplatte oder auf ein zentrales Bandlaufwerk zu sichern.

Vollständige Sicherung

Ist eine Sicherung der kompletten Datenbank, dabei werden die Datenbank und Teile des Transaktionsprotokolls gesichert. Diese Sicherung ist die Voraussetzung für die differenzielle Datenbanksicherung. Zur Wiederherstellung wird aus diesem Backup die komplette Datenbank restored.

Differenzielle Sicherung

Mit dieser Sicherung speichert man jene Datenblöcke ab, die sich seit der letzten vollständigen Sicherung geändert haben. Die differenzielle Sicherung ist kleiner als eine Vollsicherung. Bei der Wiederherstellung benötigt man auch die passende Vollsicherung, um die Datenbank wieder vollständig herzustellen. Es müssen die Medien der passenden vollständigen und die der differenziellen Sicherung bereitstehen.

Sicherung der Transaktionsprotokolle

Das Transaktionsprotokoll beinhaltet alle Veränderung auf der Datenbank. Die Sicherung des Transaktionsprotokolls ist viel schneller als eine Vollsicherung möglich. Details zum Transaktionsprotokoll siehe Infobox.

Tail-Log Sicherung

Hier wird auch jener Teil gesichert, der noch nicht im Transaktionsprotokoll gesichert wurde, wenn die Datenbank nicht mehr erreichbar ist. Damit kann man das Protokollfragment zusätzlich sichern.

Wiederherstellungs Modelle (Recovery model)

Legt fest, wie der SQL Server die Daten zum Transaktionsprotokoll speichert und verwaltet, wenn die Transaktionen abgeschlossen sind.

Simple / Einfach

Der SQL Server verwaltet das zugehörige Transaktionsprotokoll automatisch. Inaktive Teile des Transaktionsprotokolls werden automatisch gelöscht. Das Transaktionsprotokoll wird somit immer klein gehalten, es kann nicht unbegrenzt wachsen.

Der Nachteil ist, dass man in diesem Modus das Transaktionsprotokoll nicht extra sichern kann. Es bleibt nur eine Vollständige oder differentielle Sicherung über.

Full / Vollständig

Es werden alle Transaktionen im Transaktionsprotokoll gehalten und der inaktive Teil wird nicht gelöscht. Erst wenn die Transaktionsprotokollsicherung ausgeführt wird, wird der inaktive Teil gelöscht. Im Worst Case einer Wiederherstellung der Datenbank würde daher zuerst die DB Sicherung selber und danach die Transaktionsprotokollsicherung eingespielt werden. Denn nur mit der Transaktionsprotokollsicherung erreicht man, dass alles bis zur letzten Datenbankänderung restored wurde.

Wo ein Vorteil, da nicht auch ein Nachteil: Wenn nicht laufend ein Transaktionsprotokollsicherung durchgeführt wurde, wird das Transaktionsprotokoll schnell recht groß, solange bis auf der Platte kein Platz mehr ist. Dann ist ein Arbeiten am SQL Server nicht mehr möglich.

Bulk-logged / Massenprotokolliert

Ähnlich wie vollständig, aber bestimmte Operation werden nur in Minimalform ins Transaktionsprotokoll geschrieben. Das Transaktionsprotokoll nimmt daher nicht so schnell viel Speicherplatz in Anspruch. Man setzt es ein, wenn *BULK INSERTS* auf der Datenbank durchgeführt werden. Der Nachteil hier im Gegensatz zum Vollständigen Wiederherstellungsmodell ist, dass hier keine Wiederholung bis zu einem Zeitpunkt durchgeführt werden kann.

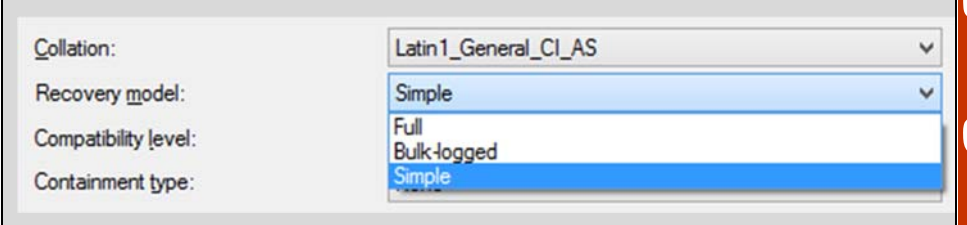
Was muss ich am SQL Server überhaupt sichern?

Die *MASTER Datenbank* enthält die gesamte Konfiguration des SQL Servers. Ohne die Master Datenbank gibt es keinen lauffähigen SQL Server. In dieser Datenbank werden für die SQL Server Instanz

Jede SQL Server Datenbank besteht aus min. zwei Files unterschiedlichen Files. Die Datendatei hat die Extension *mdf*, darin befinden sich die eigentlichen Daten in den Tables. Zusätzlich sind darin auch die Indizes, View, Stores Procedures, Trigger usw.

Das Transaction Log File die Extension *ldf*, hier werden alle Veränderungen aufgezeichnet, die zum Wiederherstellen aller Transaktionen in der Datenbank notwendig sind.

Das Transaktion File kann je nach eingestellten Recovery model der einzelnen Datenbank unterschiedlich groß werden, wobei es im Full Modus am größten wird.



alle Informationen gespeichert, daher muss diese unbedingt gesichert werden.

Der *SQL Server Agent* speichert seine Jobs in der MSDB Datenbank. Darin sind unter anderen die Zeitpläne, SSIS Rollen, Datenbank und Email Rollen, User Berechtigungen. Die MSDB Sicherung ist ebenso notwendig.

Der SQL Server verwendet die MODEL Datenbank als Vorlage bei der Anlage von neuen Datenbanken. Führt man Änderungen an der Model Datenbank durch, ist es ratsam diese auch zu sichern.

Die TEMPDB dient als temporäre Datenbank im SQL Server, etwa zum Speichern von Zwischenergebnisse bei SQL Abfragen. Die TempDB wird bei jedem Neustart einer SQL Server Instanz neu angelegt.

Snapshot

Die Erstellung von Datenbank Snapshots ist seit dem SQL Server 2005 möglich. Ein Snapshot beinhaltet einen konsistenten Zustand der Datenbank zu einem bestimmten Zeitpunkt. Technisch funktioniert ein Snapshot nach der *Copy-on-Write* Technologie.

Ein Snapshot ist keine vollständige Kopie der Datenbank. Dieser funktioniert nur, wenn die Basisdatenbank vorhanden und voll funktionstüchtig ist.

Log Shipping

Bei dieser Technik werden zwei oder mehrere SQL Server Instanzen verwendet, um die Transaction Log Files zu verteilen. Es basiert auf SQL Server Jobs. Möglich wird dies mit einer höheren Version als der Express Version. Der Vorteil ist, dass es relativ einfach ist, dies einzurichten. **(Bild 1,2)**

Log shipping umfasst das Kopieren des Datenbank Backups und der Transaktionsprotokollsicherungen vom *primary (source) server* und der Wiederherstellung der Datenbank und Transaktionsprotokollsicherungen auf einem oder mehreren sekundären Server. **(Tabellen rechts)**

Database Mirroring / Always on

Hierbei geht es um die Hochverfügbarkeit von Datenbanken. Database mirroring (DBM) gab es bereits ab SQL Server 2005 SP1, es ist am Datenbanklevel implementiert. Hierbei werden *Transaction Log Records* zwischen dem *principal* und dem DBM Server gesendet. Dabei gibt es noch den synchronen (*high safety mode*) und asynchronen (*high performance mode*) Modus. **(Bild 3, 4)**

Mit 2012 kam das neue Feature „Always on“ – AO zum SQL Server, die neue Generation von *database mirroring* für die Hochverfügbarkeit und Notfallwiederherstellung. In Zukunft wird AO dann DBM ablösen. Die Komplexität einer geclusterten SQL Server Instanz wird bei AO durch lokal installierte SQL Server Instanzen mit lokal angebundenem Storage vermieden. Die Datenbank ist wie bei DBM auf jedem Server lokal vorhanden. Im Gegensatz zu DBM lässt sich eine über AO abgesicherte

Snapshot erstellen

```
Create database MyTestDB_SnapShot on
```

```
(name=MyTestDB, filename = 'C:\SnapShots\snap1.ss')
```

```
as snapshot of MyTestDB
```

Wiederherstellen

```
Restore database MyTestDB from database_snapshot = 'MyTestDB_SnapShot'
```

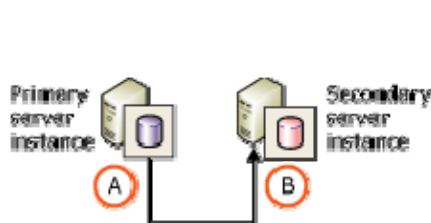


Bild 1: Log Shipping mit einer secondary Instanz

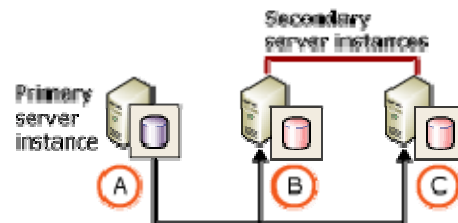


Bild 2: Log Shipping mit zwei secondary Instanzen

Log Shipping verfügbar für

2005	Enterprise, Standard, and Workgroup
2008	Enterprise, Standard, Web, and Workgroup
2008R2	Datacenter, Enterprise, Standard, Web, and Workgroup
2012	Enterprise, Business Intelligence, Standard, and Web
2014	Enterprise, Business Intelligence, Standard, and Web

Log Shipping kompatibel mit backup compression

2005	Nicht verfügbar
2008	Enterprise
2008R2	Datacenter, Enterprise, or Standard
2012	Enterprise, Business Intelligence, oder Standard
2014	Enterprise, Business Intelligence, oder Standard

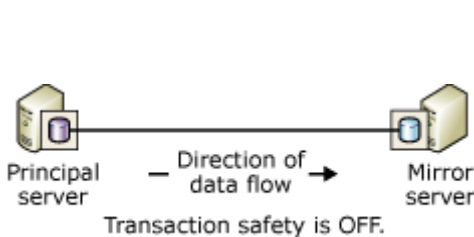


Bild 3: Asynchrones database mirroring

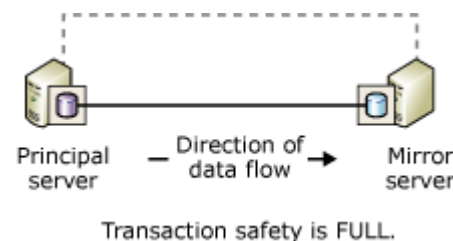


Bild 4: Synchrones database mirroring

Datenbank über einen einheitlichen DNS-Namen / IP-Adresse ansprechen. Die Applikation, die die Datenbank nutzt, hat somit Transparenz über einen eventuell erfolgenden Failover und muss nicht zur Laufzeit einen anderen Datenbankserver ansprechen.

Cloud Backup

Das Microsoft® SQL Server® Backup to Microsoft Azure® Tool erlaubt eine verschlüsselte und komprimierte Sicherung in die Microsoft Cloud. Möglich ab Windows 7 SP1 und Windows Server 2008 R2 SP1. Voraussetzung dafür ist der Microsoft Cloud Zugang, setzt man auf die Backup Strategie, ist man mit den Bestimmungen einverstanden und die aktuellen Kosten (*Azure pricing calculator for database management* verwenden) stehen im Verhältnis zum Nutzen.

SQL Server Maintainance Plan

Natürlich verfügt der SQL Server über einen Wizzard gesteuerten Wartungsplan, bei dem am Ende aller getroffenen Einstellungen ein Script vom Agent ausgeführt wird. Bei wem funktioniert der DB CleanUp Task?

Für den professionellen Einsatz würde ich dann eher die mehrfach ausgezeichneten *SQL Server SQL Stored Procedures* von **Ola Hallengren** [1], den schwedischen SQL DBA, empfehlen. Sie bietet mehr Optionen an und geht sparsamer mit den Ressourcen um.

Disaster Recovery

RPO - Recovery Point Objective

Wie viel Datenverlust kann verkraftet werden.

Es handelt sich um den Zeitraum, der zwischen zwei Datensicherungen liegen darf. Also wie viele Datentransaktionen dürfen zwischen der letzten Sicherung und des möglichen Systemausfalles verloren gehen. Sollen keine Daten verloren gehen, liegt der Wert für RPO bei 0 Sekunden.

RTO - Recovery Time Objective

Wie lange darf ein Geschäftsprozess ausfallen.

Es handelt sich um jene Zeit von Eintritt des Schadens bis zur vollständigen Wiederherstellung des Geschäftsprozesses. Der Zeitpunkt kann von 0 Minuten bis Tage oder Wochen dauern.

Auswahl seines passenden Backup Plans

RPO > 0

Hier kann *Backup und Restore* zum Einsatz kommen, das ist auch schon mit der SQL Express Version möglich. Es funktioniert auch im *Recovery Modus Simple*. RPO und RTO sind dabei von der Datenmenge abhängig. Auch der Einsatz eines Hot Standby Systems ist möglich.

Setzt man *Log Shipping* ein, hat dies den Vorteil einer kurzen RTO, man ist aber von der Datenmenge abhängig. Man hat weniger Einfluss auf das Primärsystem, und es können mehrere Hot Standby Systeme eingesetzt werden. Allerdings benötigt

man dazu min. die SQL Server Standard Edition und hier muss der Recovery Modus auf Full eingestellt sein. Es ist ein manueller Eingriff für den DB Admin notwendig, der ähnlich wie bei Backup und Restore vor sich geht. Ein RPO von wenigen Minuten kann damit erreicht werden.

Beim Einsatz von *Replication* ist der RTO defakto null. Am SQL Server funktioniert es auch im *Recovery Model Simple* ab der *Standard Edition*. Es sind mehrere Subscriber möglich, es ist auch als Scale Out System verwendbar. Aber der RPO ist unberechenbar, vor allem bei hoher Last. Es kann einen hohen Einfluss auf das Primärsystem haben, aber vor allem auf das Transaction Log File.

RPO = 0

SQL Failover Cluster kommt auf einen RTO von wenigen Sekunden. Es ist die einzige Lösung, bei der auch die Server Objekte, also *Linked Server* und die *Agent Jobs*, weiterhin funktionieren. Diese Lösung ist transparent für die Applikationen. Der SQL Server unterstützt dies aber der Standard Edition. Diese komplexe Lösung benötigt aber viel Infrastruktur wie *Active Directory* und *Shares Storage* und die Windows und SQL Server Versionen muss auf allen Nodes die gleiche sein.

Database mirroring kommt auf einen RTO von wenigen Sekunden. Es wird keine aufwändige Infrastruktur benötigt und ist aber der SQL Server Standard Edition möglich. Diese Technik ist in der MSDN bereits als deprecated beschrieben und wird in Zukunft durch AO ersetzt werden. Die Nachteile sind außerdem, dass ein spezieller Treiber am Client benötigt wird. Die Geschwindigkeit des Mirrors kann einen Einfluss auf den Primary haben. Am SQL Server ist Recovery Modus Full zu verwenden.

Availability Groups haben einen RTO von wenigen Sekunden. Es wird eine spezielle Infrastruktur vorausgesetzt, für den Client wirkt es transparent. Funktioniert leider nur in der SQL Server Enterprise Edition und den *Recovery Modus Full*. Es handelt sich um eine komplexe Lösung aufgrund mehrerer Komponenten. Auch hier müssen Windows und SQL Server Versionen auf allen Nodes die gleiche sein. Die Geschwindigkeit der Replica hat zudem Einfluss auf den Primary Server.

Virtualisierungcluster

Mittels eines Virtualisierungclusters kommt man auf einen RTO von null bei einem geplanten Failover und einem RPO von null. Es ist transparent für SQL Server und für den Client. Diese Lösung funktioniert für alle SQL Server Edition, auch bei der Express Version.

Nachteil ist hier eine längere Recovery Zeit bei einem ungeplanten Failover. Bei dieser komplexen Lösung ist viel Infrastruktur notwendig.

Virtualisierungsreplica

Mittels eines Virtualisierungsreplica kommt man auf einen RTO und RPO von null bei einem geplanten Failover und es ist ebenfalls transparent für SQL Server und für den Client. Allerdings kann es zu einem Datenverlust bei einem ungeplanten Failover kommen. Auch diese Lösung ist komplex durch die hohe Anzahl von Komponenten.

Durchspielen seines gewählten Szenarios

Der Beste theoretische Plan hilft nichts, wenn er nicht zumindest einmal durchgezogen wurde. Es gibt Fälle wo das Backup defekt oder nicht mehr kompatibel zum aktuellen Environment ist. Aber auch ganz triviale, wo es eine oder keine Urlaubsvertretung gibt und der Backup Job nicht läuft oder die passenden Medien nicht rechtzeitig gefunden oder darauf zugegriffen werden können.

Links & Quellen

[1] <https://ola.hallengren.com/>

Fazit

Es gibt unzählige Varianten der Datensicherung, wobei die passende Lösung von der Anwendung und von der Infrastruktur abhängig ist. Eine Hochverfügbarkeit führt zu einem höheren Wartungsaufwand und setzt auch Wissen über mehrere Themenbereiche voraus. Wird die kostenintensive Hochverfügbarkeit schlecht gewartet, führt dies am Ende zu einem noch größeren Problem.

Autorenbox

Thomas Reinwart verfügt über umfangreiche Berufserfahrung auf dem IT Sektor. In den letzten 20 Jahren war er in den Bereichen Softwareentwicklung, Software-design, Architekt und als Consultant tätig. Technischer Fokus ist derzeit Microsoft .net und SQL Server, wo er alle aktuellen Microsoft Zertifizierungen hat.

E-Mail: office@reinwart.com

