

(Automatisierte) Migration einer SQL Server Datenbank

Kategorie
java, SQL Server

Aaron
Priesterroth

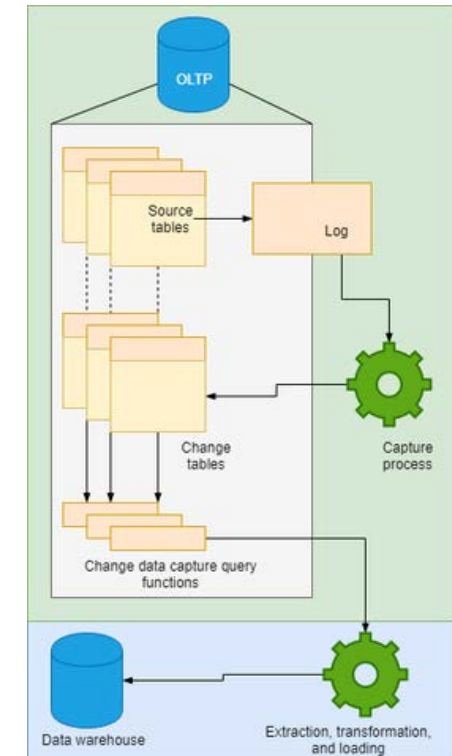
Die Migration einer SQL Server Datenbank kann, je nach äußeren Umständen, eine kritische Operation darstellen. Vor allem dann, wenn die Instanz oder die auf der Instanz befindlichen Datenbanken zeit-kritischen Abläufen unterliegen und sich eine "Downtime" nicht leisten können.

Die herkömmliche Migration besteht oft aus dem Erstellen und Einspielen von Backups. Da die Datenbanken nach dem Erstellen der Backups jedoch weiter zur Verfügung stehen müssen, entsteht eine Differenz zwischen dem Echtzeit-System und der erstellten Backup-Datei. Diese Differenzen müssen anschließend mühselig über den Transaktionslog der Datenbank aufgearbeitet werden, um die Äquivalenz der Zustände der Datenbanken wiederherzustellen.

Change Data Capture

Ein von Microsoft zur Verfügung stehendes Werkzeug mit dem Namen **Change-Data-Capture** (CDC) kann dabei einen entscheidenden Vorteil bieten: Wird CDC für eine Datenbank eingeschaltet, werden alle zukünftig getätigten Transaktionen pro Tabelle in einer zusätzlichen Systemtabelle (eine sog. Änderungstabelle) registriert. Dabei wird der Typ der Operation sowie die zur Operation zugehörigen Informationen erfasst.

Wird also beispielsweise eine Zeile in einer Tabelle gelöscht, wird die Zeile gespeichert und im Feld der zugehörigen Operation eine "1" gespeichert. Im Falle eines Updates werden sogar zwei Transaktionen erfasst, eine mit der Operation "3" und den Werten der Zeile vor dem Update, eine zweite mit der Operation "4" und den Werten der Zeile nach dem Update.



Basierend auf diesem Werkzeug haben wir uns zum Ziel gesetzt mit dem **SQL Syncer** eine vollautomatische Echtzeit-Datenintegration zu entwerfen, dessen Natur es möglich macht, in jeder denkbaren Umgebung eine zuverlässige Lösung zur Migration zu bieten. Eine automatisierte Migration ist ein ambitioniertes Projekt. Worauf geachtet werden muss und welchen Problemen man begegnen kann, möchten wir im folgenden Abschnitt durchleuchten.

Automatisierte Migration & SQL Syncer

Für die automatisierte Migration bildet CDC ein solides Grundgerüst, bringt den Stein jedoch noch nicht ins Rollen: Sobald CDC für eine Datenbank eingeschaltet ist, kann mit dem Übertragen der Tabellen begonnen werden. Aber wie sollen die Daten übertragen werden? Hier gilt es zu entscheiden, ob eine weitere Abstraktionsstufe in Form von Backup-Dateien oder ähnlichem gewollt ist.

Der **SQL Syncer**, dessen Augenmerk auf der Echtzeit-Datenintegration liegt, überträgt Daten in einer direkten (wahlweise auch in einer indirekten) Peer-To-Peer Verbindung von der Daten-Quelle in die Daten-Senke. Dabei ist es wichtig, die Struktur und den Aufbau der Datenbank selbst zu beachten:

Wenn Tabellen übertragen werden, gibt es benutzerdefinierte Typen, Abhängigkeiten, Einschränkungen, Schemas, Indexe, Full-Text-Indexe, Trigger, etc., die berücksichtigt werden müssen.

Von einem rein logischen Ansatz bilden die Benutzer und Benutzer-Gruppen die obere Hierarchie der Instanz. Beim Übertragen und Erstellen der Tabellen und Datenbanken, können die Berechtigungen direkt übernommen werden, ohne dass sie nachträglich eingefügt werden müssen. Wie man Benutzer-Logins überträgt, können Sie [hier](#) in einem Artikel von Microsoft zu lesen.

Bezüglich der Datenbank bildet die oberste Hierarchie das Schema. Ohne die passenden Schemas können die benötigten Tabellen nicht übertragen werden, sie müssen als Erstes übernommen werden. Ohne die Benutzer-definierten Typen können jedoch wiederum die Tabellen, die diese Typen verwenden, nicht übertragen werden. Indexe können wiederum nicht übertragen werden bis die Tabellen transferiert wurden. Der Rattenschwanz zieht sich durch den gesamten Prozess.

Weitere kleine Details zum Migrieren etwa von System-versionierten Tabellen (temporale Tabellen) können Sie [hier](#) in einem unserer Artikel nachlesen.

Interessant wird es wieder, wenn die ersten Datenbank-Objekte übertragen wurden. Datenintegration in Echtzeit ist ein iterativer Prozess, der fortlaufend Änderungen an der Daten-Quelle in die Daten-Senke übernimmt. Das heißt, nachdem die grundlegende Struktur der Instanz bzw. der Datenbank übertragen wurde, kann mit der Phase der kontinuierlichen Synchronisation begonnen werden. Hier kommt CDC zum Einsatz.

Mit Hilfe der zuvor vom System erstellten Änderungstabellen der zu übertragenden Datenbank, kann nun spielend leicht jede Transaktion nachverfolgt und so die beiden Instanzen kontinuierlich, iterativ abgeglichen werden.

Mit Hilfe dieser Strategie kann der Benutzer nun selbst die Migration in die Hand nehmen und nach Belieben die beiden Datenbanken synchronisieren. Der Vorgang kann beliebig lange durchgeführt werden, bis das System sich einem nicht-kritischen Zustand befindet, sodass die erstellte Kopie der Datenbank als Live-System übernommen werden kann.

Weitere spannende Einblicke hinter die Kulissen der Entwicklung des **SQL Syncer** oder eines unserer anderen Software-Projekte gibt es auf [unserer Github Seite](#) zu finden.