

Änderungen mit Change Tracking verfolgen

Der MS SQL Server bietet ein Feature mit dem Namen Change Tracking an, welches die Änderungen von Zeilen in einer Tabelle mittels Versionierung verfolgbar machen kann. Im Gegensatz zum Change Data Capture (CDC) werden jedoch nicht die Änderungen an sich erfasst und gespeichert, sondern es wird eine Versionsnummer über die Änderungen vergeben, sodass z. B. nur alle geänderten Abfragen seit einer bestimmten Version zurückgegeben werden können.

Dieses Feature bietet sich an, um inkrementelle SQL-Abfragen zu gestalten, da so z. B. jede Stunde ein SELECT ausgeführt werden kann, welches alle geänderten Zeilen zurückgibt, um diese in eine andere Datenbank zu kopieren oder mittels einer BI-Software zu verarbeiten.

Damit die Änderungen erfasst werden können, muss zuerst Change Tracking in der Datenbank und dann für die zu überwachenden Tabellen aktiviert werden.

Um Change Tracking in einer Datenbank verfügbar zu machen, kann folgendes SQL verwendet werden.

```
ALTER DATABASE ÄndereDenDatenbanknamen  
SET CHANGE_TRACKING = ON  
(CHANGE_RETENTION = 7 DAYS, AUTO_CLEANUP = ON)
```

In diesem SQL wurde eine Verfallszeit von 7 Tagen angegeben und dass die Logs ab dann abgeschnitten werden. Es können auch kürzere oder längere Zeiträume definiert oder sogar das Aufräumen komplett deaktiviert werden. Jedoch können bei vielen Änderungen eine Menge Daten anfallen, sodass ein Aufräumen zu empfehlen ist.

Mit dem folgenden SQL kann nun das Tracking für eine Tabelle aktiviert werden.

```
ALTER TABLE ÄndereDenNamenDerTabelle  
ENABLE CHANGE_TRACKING  
WITH (TRACK_COLUMNS_UPDATED = ON)
```

Nun werden alle Änderungen in der Tabelle überwacht und führen zu einer Aktualisierung der Version. Außerdem werden auch die Spalten überwacht, sodass nicht nur festgestellt werden kann, welche Zeilen geändert wurden, sondern auch welche Spalten. Diese Option sollte auf OFF gesetzt werden, wenn nur die geänderten Zeilen benötigt werden und sowieso immer bestimmte Spalten oder sogar alle Spalten ausgegeben werden.

Die Versionierung startet bei 0 und wird mit jeder Änderung um 1 hochgezählt, die aktuelle Version lässt sich mit folgendem SQL ermitteln.

```
SELECT CHANGE_TRACKING_CURRENT_VERSION();
```

Wenn nun Änderungen an den Daten vorgenommen werden, wird die Version automatisch erhöht. Mit Hilfe des folgenden SQL können nun nur die Änderungen ab einer bestimmten Version zurückgegeben werden.

```
SELECT aenderungen.SYS_CHANGE_VERSION, emp.*
FROM CHANGETABLE (CHANGES HumanResources.Employee, 0) AS aenderungen
INNER JOIN HumanResources.Employee emp on emp.[BusinessEntityID] =
aenderungen.[BusinessEntityID];
```

Als Beispiel wurde die Tabelle *HumanResources.Employee* aus der AdventureWorks Beispieeltabelle verwendet.

Hier wurde die Version auf 0 gesetzt, wodurch alle Zeilen zurückgegeben werden, welche jemals verändert wurden (innerhalb der letzten 7 Tage, da diese die Löschezit ist).

Nachdem das Change Tracking aktiviert wurde, sind 2 Zeilen in der Tabelle verändert worden, wodurch die Abfragen nun folgende Ergebnisse liefern.

```
11 SELECT CHANGE_TRACKING_CURRENT_VERSION() AS 'Version';
12
13 SELECT aenderungen.SYS_CHANGE_VERSION, emp.*
14 FROM CHANGETABLE (CHANGES HumanResources.Employee, 0) AS aenderungen
15 INNER JOIN HumanResources.Employee emp on emp.[BusinessEntityID] = aenderungen.[BusinessEntityID];
```

Results Messages

Version
1
2

	SYS_CHANGE_VERSION	BusinessEntityID	NationalIDNumber	LoginID	OrganizationNode	OrganizationLevel	JobTitle	BirthDate	MaritalStatus	Gender	HireDate	SalariedFlag	Vacat
1	1	5	695256988	adventure-works\gail0	0xSADA	3	Senior Design Engineer	1952-09-27	M	F	2008-01-06	1	5
2	2	6	998320692	adventure-works\jossef0	0xSADE	3	Senior Design Engineer	1959-03-11	M	M	2008-01-24	1	6