Migration einer existierenden System-versionierten Tabelle (temporale Tabelle)

Kathegorie
SQL Server

Aaron
Priesterroth

Temporale Tabellen sind ein nützliches Werkzeug wenn es darum geht, voll automatisch den Verlauf von Datenänderungen nachzuverfolgen. Sie machen sich die Funktionalität von automatisch generierten Spalten innerhalb der Tabelle zu Nutzen und beschreiben so einen Zeitraum über den Zustand der Daten.
Diese automatisch generierten Spalten sorgen beim Anwender jedoch oft für Probleme, gerade in Hinsicht auf Server-Migration.

Eine temporale Tabelle zu übertragen, und dabei den Urzustand der Tabelle wieder herstellen zu können, ist keine leichte Aufgabe. Im Folgenden betrachten wir ein Beispiel davon, welche Strategie genutzt werden kann, um ohne Probleme eine temporale Tabelle zu Übertragen und den allgemeinen Umgang mit ihnen zu vereinfachen.

Kompatibilität der Datenbank bestimmen

Um eine temporale Tabelle in eine Datenbank übertragen zu können, muss gewährleistet sein, dass die verwendete SQL Server Version System-Versionierung unterstützt. Generell gilt: temporale Tabellen sind ab SQL Server Version 2016 verfügbar.

Um zu überprüfen, ob ihre SQL Server Version temporale Tabellen unterstützt, kann die Abfrage

```
SELECT SERVERPROPERTY('ProductVersion') AS [ProductVersion];
```

genutzt werden. Die als Rückgabe erhaltene Versionierungsummer muss mit einer Zahl größer 12 beginnen.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Temporale Tabellen identifizieren

Um zu identifizieren, dass es sich bei einer Tabelle um eine temporale handelt, kann folgende Abfrage verwendet werden:

```
SELECT temporal_type FROM sys.tables WHERE name = ‘MyTable’ and
schema_id = SCHEMA_ID('dbo');
```

Hierbei stehen

- **MyTable** für den Namen der Tabelle für die die Abfrage ausgeführt werden soll und
- **dbo** für das Schema der zuvor spezifizierten Tabelle

und müssen individuell angepasst werden. Ist das Ergebnis der Abfrage die Zahl **2**, handelt es sich bei der Tabelle die für die Abfrage genutzt wurde um eine temporale Tabelle.

Übertragen der temporalen Tabelle

Wenn alle Voraussetzungen erfüllt sind, kann mit dem Übertragen der temporalen Tabelle begonnen werden. Der einfachste Ansatz ist, die Tabelle in ihrer rohen Form (lediglich die Daten ohne die dahinter stehenden Funktionalität der System-Versionierung) zu übertragen und die erstellte Tabelle anschließend so zu modifizieren, dass die Abhängigkeiten mit den Spalten welche die Periode bestimmen wieder hergestellt sind.

In den folgenden Abschnitten wird davon ausgegangen, dass diese neu erstellte Tabelle bereits vorhanden und mit den Daten aus der original Tabelle gefüllt ist. D.h., es wurde bereits eine Kopie der temporalen Tabelle erstellt, die Kopie enthält jedoch lediglich die Daten, wird aber nicht automatisch versioniert.

Identifikation der Start- und End-Spalte der Periode der temporalen Tabelle

Um aus der neu erstellten Tabelle eine temporale zu erzeugen, muss die Periode der originalen Tabelle identifiziert werden. Diese Periode wird durch zwei Spalten in der originalen Tabelle beschrieben. Mit Hilfe der Abfragen

```
SELECT name FROM sys.columns WHERE OBJECT_ID = (’MyTable') AND column_id IN
(SELECT start_column_id FROM sys.periods WHERE object_id = OBJECT_ID('MyTable');
```

kann der Name der Spalte bestimmt werden, welche den Anfang (den Startpunkt) der Periode in der temporalen Tabelle bestimmt. Hierbei steht **MyTable** für den Namen der Tabelle für die die Abfrage ausgeführt werden soll. Analog kann die Spalte welche das Ende der Periode beschreibt mit der folgenden Abfrage bestimmt werden:

```
SELECT name FROM sys.columns WHERE OBJECT_ID = (’MyTable') AND column_id IN
(SELECT end_column_id FROM sys.periods WHERE object_id = OBJECT_ID('MyTable');
```

Definition der Periode (I.)

Da es sich bei den Start- und End-Spalten um automatisch generierte Spalten handelt, müssen zusätzliche Spalten eingefügt werden, um den Umgang zu erleichtern. Mit der Abfrage:

```
ALTER TABLE ‘dbo.MyTable’ ADD myStartColumn_temp datetime2 GENERATED ALWAYS AS
ROW START NOT NULL
CONSTRAINT DF_MyTable_myStartColumn DEFAULT (GETUTCDATE()),
myEndColumn_temp datetime2 GENERATE ALWAYS AS ROW END NOT NULL
CONSTRAINT DF_MyTable_myEndColumn DEFAULT (’999-12-31 23:59:59.9999999'),
PERIOD FOR SYSTEM_TIME (myStartColumn_temp, myEndColumn_temp);
```

wird für die Tabelle **dbo.MyTable** zwei neue automatisch generierte Spalten eingefügt (**myStartColumn_temp** und **myEndColumn_temp**) und basierend darauf eine PERIOD definiert.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Hinweis: In der oben beschriebenen Abfrage müssen die Werte individuell angepasst werden, wobei

- **dbo.MyTable** den Name der neu erstellten Tabelle beschreibt,
- **myStartColumnTemp** und **myEndColumnTemp** die Namen der Start- und End-Spalte der Periode beschreiben und
- **DF_MyTable_myStartColumn** und **DF_MyTable_myEndColumn** die Namen der zugehörigen Constraints beschreiben.

Die Definition der Periode ist die Voraussetzung für das erstellen der **GENERATE ALWAYS** Spalten. Diese muss jedoch wieder entfernt werden, um das Übertragen der Gewünschten Daten in die neu erstellten Spalten zu ermöglichen. Folgende Abfrage kann für das entfernen der Periode genutzt werde:

```
ALTER TABLE ‘dbo.MyTable' DROP PERIOD FOR SYSTEM_TIME;
```

Übertragen der Daten

Mit den neu definierten Spalten und der entfernten Periode, kann nun begonnen werden die gewünschten Daten in die neuen Spalten zu übertragen. Dies kann beispielsweise mit folgender Abfrage geschehen:

```
UPDATE dbo.MyTable SET myStartColumn_temp = myStartColumn;
```

Innerhalb der Tabelle werden nun also die Werte die aus der originalen temporalen Tabelle übertragen wurden in die Spalte eingefügt, die später die Periode bestimmen soll. In diesem Beispiel ist also

- **dbo.MyTable** der Name der Tabelle für welche die Daten übertragen werden sollen,
- **myStartColumn_temp** der Name der zuvor in Abschnitt "**Definition der Periode (I.)**" erstellten Spalte und
- **myStartColumn** der Name der Kopie der Spalte aus der originalen temporalen Tabelle

und alle Werte müssen individuell angepasst werden.
Analog müssen die Daten der End-Spalte übertragen werden.

Entfernen der alten Spalten

Nachdem die Daten übertragen wurden, können die jetzt überflüssig gewordenen Spalten entfern werden. Hierfür kann die folgende Abfrage verwendet werden:

```
ALTER TABLE dbo.MyTable DROP COLUMN myStartColumn, myEndColumn;
```

Wobei hier

- **myStartColumn** für den Namen der Spalte steht, welche in der originalen Tabelle die Start-Spalte der Periode beschreibt und
- **myEndColumn** für den Namen der Spalte steht, welche in der originalen Tabelle die End-Spalte der Periode beschreibt

ACHTUNG: Bei den Namen der Spalten die entfernt werden sollen, handelt es sich um die Spalten deren Name mit "_temp" enden.

Definition der Periode (II.)

Da die gewünschten Daten sich nun in den neu erstellen und automatisch generierten Spalten befinden, kann die Period wieder hergestellt werden. Hierfür kann die Abfrage

```
ALTER TABLE 'dbo.MyTable' ADD PERIOD FOR SYSTEM_TIME(myStartColumn_temp,
myEndColumn_temp);
```

genutzt werden.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Umbenennen der Start- und End-Spalte

Um den Zustand der originalen temporalen Tabelle bezüglich der Spalten-Namen wiederherzustellen, kann die **gespeicherte Prozedur** "sp_rename" verwendet werden. Folgender Syntax kann genutzt werden, um den Namen der Spalten anzupassen:

```
ALTER TABLE dbo.MyTable DROP COLUMN myStartColumn, myEndColumn;
```

Um also die Start- und End-Spalten zu entfernen, muss folgende Abfrage ausgeführt werden:

```
EXEC sp_rename('dbo.MyTable.myStartColumn_temp', 'myStartColumn',
'COLUMN');
```

Analog wird die End-Spalte umbenannt.

Abschließend

Nachdem alle oben genannten Schritte durchgeführt wurden, sollte die temporale Tabelle erfolgreich mit allen bestehenden Daten übertragen worden sein.

java, jdbc, migration, sql server, sqlsyncer, system versioning, t-sql, temporal table, transfer table

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612