

Automatic Merge (MS-SQL)

Donnerstag, 29. August 2019 10:00



Robert Aufreiter

Dipl. Ing. Informatik
Senior Consultant BI
Product Strategy

Abstract:

Mit dem Befehl MERGE stellt der MS SQL-SERVER eine überaus performante T-SQL-Funktion zur Verfügung um INSERT, UPDATE, DELETE in einer einzigen Transaktion ausführen zu können. Da dieser Befehl jedoch nicht ganz einfach hinzuschreiben ist, wird er nur selten verwendet.

Deshalb soll eine T-SQL StoredProcedure vorgestellt werden, welche die AUTOMATISCHE Erstellung des passenden MERGE-Befehls an Hand einer bereits vorhandenen Tabelle ermöglicht.

Zusätzlich wird gezeigt, wie diese StoredProcedure auch dazu verwendet werden kann, um automatisch eine "Historisierung" zu erzeugen (z.B. als Ausgangsbasis für "Slowly Changing Dimensions")

Beispiel:

- Material-Stammdaten: SAP-Tabelle MARA

Teilnehmer:

- SQL-Spezialisten, die große Tabellen in ein Datawarehouse laden müssen (z.B. große Dimensionstabellen)

Agenda:

1. Warum MERGE, warum "Automatic Merge"

I love MERGE because I think it is the greatest thing to happen to SQL querying.

Aus <http://www.sqlservercentral.com/articles/MERGE/103127/?utm_source=SSC&utm_medium=pubemail>

As far as I'm concerned, it is the neatest thing to come along in the Data Manipulation Language (DML) world since sliced bread.

Aus <<http://www.sqlservercentral.com/articles/MERGE/97867/>>

Merge statements can be confusing at first so there's a bit of a learning curve. However, once you get the hang of it, I think you'll agree that Merge is a great way to go.

Aus <<http://www.made2mentor.com/2013/08/how-to-load-slowly-changing-dimensions-using-t-sql-merge/>>

o Vorteile:

- EIN einziges Statement zum gleichzeitigen INSERT, UPDATE, DELETE.
- sehr flexibel

o Nachteile:

- komplex, schwierig zu verstehen
- viel Schreibaufwand

o Performance

- Keine generelle Aussage möglich (hängt von SQL-Server Version und auch sehr vom "Logging"-Aufwand ab)
- MERGE ist in etwa gleich gut wie Insert/Update/Delete - siehe [Performance of the SQL MERGE vs. INSERT/UPDATE](#)

o Entstehungsgeschichte "AUTOMATIC Merge":

➢ Ziele:

- ✓ Mit der "AUTOMATIC MERGE-StoredProcedure" wird der **Schreibaufwand** zur Erstellung des MERGE-Statements auf 0 reduziert!
- ✓ Vermeidung von **Fallstricken**, wie
 - [Writing T-SQL Merge Statements the Right Way](#)
 - [Use Caution with SQL Server's MERGE Statement](#)
 - Ever use "**WITH(HOLDLOCK)**"-hint (=SERIALIZABLE [Transaction Isolation Level](#)) on **TARGET**-table to prevent from "SQL Server Merge Concurrency Issues". This means that you could end up with race conditions or **primary key conflicts** when multiple users are attempting to fire the same MERGE statement concurrently.
 - [Minimal logging and MERGE statement](#)
 - [Using the Output Clause with T-SQL Merge](#)
- ✓ Die "**Komplexität**" in der korrekten Schreibweise wird auf ein Minimum **reduziert** - statt ihn richtig hinzuschreiben, muss man den erzeugten SQL-Code "nur mehr" verstehen.

- "Original"-Artikel von Glen Schwickerath(2009-04-07): [Dynamic SQL Merge](#)

- Dieser wurde von Robert Aufreiter und Michael Schwingenschlögl als Basis verwendet und deutlich erweitert, und noch um (viele) Parameter zur Erhöhung der Flexibilität ergänzt.

- Erweiterung zur automatischen Erzeugung von "**Slowly Changing Dimensions Typ2**"-Datensätzen.

- > Type 0 – Ignore updates
- > Type 1 – Only keep latest version
- > Type 2 – Track history by creating a new row

Umsetzungs-Konzept siehe

- [Using T-SQL Merge to load Data Warehouse dimensions](#)
- [Using the SQL Server MERGE Statement to Process Type 2 Slowly Changing Dimension](#)
- [How to Properly Load Slowly Changing Dimensions \(SCD\) using T-SQL Merge](#)

2. Verwendungs-Konzept:

- o Aufruf einer allgemeinen StoredProcedure **[dbo].[mrgsp_merge]** mit entsprechenden Parametern (siehe unten), welche ihrerseits eine StoredProcedure mit dem entsprechenden MERGE-Statement generiert (Name wird automatisch generiert, kann aber auch manuell gewählt werden).
- o Diese automatisch erstellte "MERGE-StoredProcedure" kann dann ausgeführt oder verändert werden.
(Es ist somit vollkommen transparent, was getan wird!)

3. Voraussetzungen:

- o MS SQL Server 2008R2
- o Eine QUELL-Tabelle oder -View mit den zu ändernden Daten
- o Eine ZIEL-Tabelle mit allen bisherigen Daten und zusätzlichen Spalten zur Historisierung (wenn erwünscht)
- o ZIEL-Tabelle muss einen Primary Key (zumindest einen Unique Index) enthalten, der eindeutig die Datensätze identifiziert (Update/Delete vs. Insert).
Aus Performance-Gründen besitzt auch die Quelltable idealerweise einen entsprechenden Primary Key (oder Unique Index).

4. Installation:

- Install-SQL-Script ausführen
 - von https://svn.cubeware.de/svn/BistroApps/Automatic_Merge aktuelles ZIP herunterladen
 - Zum "Installieren" genügt es das SQL-Script "Install Merge (V###).sql" einmalig in JEDER gewünschten Datenbank auszuführen.
- Folgende Datenbank-Objekte werden erzeugt:
 - ✓ Datenbank-Schema: [ddic]
 - ✓ StoredProcedure: [dbo].[uspPrintError]
 - ✓ Table: [ddic].[ErrorLog]
 - ✓ StoredProcedure: [dbo].[uspLogError]
 - ✓ StoredProcedure: [dbo].[mrgsp_merge_base]
 - StoredProcedure: [dbo].[mrgsp_merge]
 - (diese dient als das eigentliche User-Interface und wird aufgerufen)
 - ✓ UserDefined Table-valued Function: [ddic].[tf_get_SQLObjectAttributes]
 - ✓ View: [ddic].[V_INFORMATION_SCHEMA_COLUMNS]
 - ✓ View: [ddic].[V_sys_Index_Columns]
 - ✓ StoredProcedure: [ddic].[mrgsp_MergeTable_mrg_Columns]
 - ✓ StoredProcedure: [ddic].[mrgsp_init_merge]
 - ✓ Table: [ddic].[mrg_Columns]
 - ("zentrale" Verwaltungstabelle mit sämtlichen benötigten Metadaten und Parameter-Einstellungen)
- Weiters enthält dieses ZIP auch Beispiele, Erklärungen und dieses Dokument.
 - Beispiele-Aufrufe siehe "Code_Samples.sql"

5. Beispiele (CodeSamples)

- a. "normales" Insert/Update
 - ✓ [sap].[MARA_TEMP] --> [sap].[MARA] (Material-Stammdaten aus SAP)
 - Code Sample-1
- b. "normales" Insert/Update/Delete(Marker)
 - ✓ [temp].[Benutzer] --> [dbo].[Benutzer]
 - Code Sample-5
- c. Historisierung
 - ✓ Erweiterung Zieltabelle [Benutzer_Hist] um "Historisierungs-Informationen"
 - Muster SQL-Script "Add_HistorisationFields (Sample).sql"
 - ✓ Tabelle [Benutzer] --> [Benutzer_Hist]
 - Code Sample-6
- d. "Slowly Changing Dimensions Typ-2" --> Parameter "SCD2" mit Versions-relevanten Spalten
 - > Type 2 – Track history by creating a new row
 - ✓ [sap].[T000_temp] --> [sap].[T000] (Mandanten-Stammdaten aus SAP)
 - Code Sample-2
- e. Merge-StoredProcedure mit Aufrufparametern und View als Quelle
 - ✓ [ddic].[V_INFORMATION_SCHEMA_COLUMNS] --> [ddic].[mrg_Columns]
 - Code Sample-3
- f. Verwendung für "Incremental Load" in Cubeware-Importer (extra Webinar am 10.10.2019)
 - Tabelle [MARA] - Material-Stammdaten aus SAP
 - (1) Ermitteln Spalte, die Informationen für "Incremental Load" enthält:
 - [MARA~LAEDA]
 - (2) Ermitteln **MAXIMUM** aller bereits übertragener Daten:
 - SELECT convert(varchar(8), isnull(max([MARA~LAEDA]), '19000101'), 112) as gmax_incrfield
 - FROM [sap].[MARA] MARA
 - WHERE [MARA~MANDT] = '800' AND MARA_DWH_VERSION = '201508'
 - (3) Nur jene Datensätze in die **Zwischentabelle [extr_sap].[MARA]** transferieren, die größer gleich diesem Maximum sind:
 - SELECT MARA~**
 - FROM MARA
 - WHERE (MARA~LAEDA >= '<@&@gmax_incrfield>') OR (MARA~ERSDA >= '<@&@gmax_incrfield>')
 - (4) **"Automatic Merge"** von [extr_sap].[MARA] in [sap].[MARA] - Insert/Update
 - /* Create "Automatic Merge" for table [sap].[MARA] */
 - exec dbo.mrgsp_merge
 - @SrcSchema='extr_sap' ,
 - @SrcTable='MARA' ,
 - @TgtSchema='sap' ,
 - @UpdateStoredProcedure='Y' /* insert 'N', if you don't want automatic update of storedprocedure */
 - /* ,@Debug='Y' */

6. Verfügbare Parameter StoredProcedure [dbo].[mrgsp_merge]:

```
-- *** Input-Parameters:
--@TgtDatabase           Target database. - NULL for @SrcDatabase or DB_NAME().
--@TgtSchema            Target schema. - NULL for @SrcSchema or SCHEMA_NAME().
--@TgtTable             Target table.
--@SrcServer            Linked server for iSeries or other SQL Server. - NULL for (local).
--@SrcDatabase         Source database. - NULL for @TgtDatabase or DB_NAME().
--@SrcSchema           Source schema. - NULL for @TgtSchema or SCHEMA_NAME().
--@SrcTable            Source table or view.
--@SrcType             Source server type: "LINK" (SQL Server Link), or "SQL" (default)
--@SrcWhereClause      WHERE clause to subset source-data merged (excluding "WHERE"). If left empty or NULL -> entire source-table is merged.
--@DefColumnFrom       Defines base for column- and key-definition: "Source" or "Target" (default)
--@IgnoreKeyColumns    List of key-column-names separated by ',' -like IN-list: NULL (default)
--@IndexName           Name of a unique index to use for MATCH-Clause instead of PrimaryKey, default '(=use PrimaryKey)
--@IgnoreColumns       List of column-names separated by ',' (like IN-list): NULL (default)
-- *** Executing-Options:
--@DeleteStmt          Create "DELETE"-segment of MERGE: "Y" or "N" (default)
--                     !Be very carefully using this parameter - ALL records in TARGET table not existing in SOURCE will be deleted!
--@UpdateStoredProcedure Update existing StoredProcedure: "Y" or "N" (default)
--@MergeOutput         Merge OUTPUT-ColumnList or "Y"(='INSERTED.*') or NULL (= no Output) (default)
--                     !Do not use this parameter, if created StoredProcedure is called from Cubeware Importer!
--@OutputColumns       Output ALL source-columns in Select: "Y" (default) or "N" (=select * from ...)
--@OutputDelColumns    Output deleted-columns: "Y" or "N" (default)
--@CheckHashbyteRow    Check row creted with HASHBYTE-function to identify changes: "Y" or "N" (default)
--@UseExceptChanges    Use SQL-EXCEPT to identify changes: "Y" (default) or "N"
-- *** Programming-Options:
--@MrgSp_Name          Name of "MERGE"-storedprocedure
--                     -- default "mrgsp_MergeTable_<Target_TableName>"
--@Parameter           Parameter-Definition for executing created stored procedure
--@ParseOnly           Generate MERGE statement but do not execute: "Y" or "N" (default)
--@Debug              Displays debugging information: "Y" or "N" (default)
-- *** Options for Historisation / Slowly Changing Dimensions Typ-2:
```

--@SysHistorisation	Use Historisation and necessary additional Fields: "Y", "SCD2" or "N" (default)
--@SCD2Columns	List of column-names separated by ',' (like IN-list): NULL (default) relevant for SCD2
--@_SysChangeTime_	Fieldname for HistChangeTime: -- default "" when @SysHistorisation='N' -- default "_SysChangeTime_" when @SysHistorisation='Y' or 'SCD2'
--@_SysChgDateValue	Value or fieldname(when available in SourceTable) used for ValidFrom- and ValidTo-Value
--@_SysStartTime_	Fieldname for HistStartTime: -- default "" when @SysHistorisation='N' -- default "_SysStartTime_" when @SysHistorisation='Y' or 'SCD2'
--@_SysEndTime_	Fieldname for HistEndTime: -- default "" when @SysHistorisation='N' -- default "_SysEndTime_" when @SysHistorisation='Y' or 'SCD2'
--@_SysEndTimeValue	Default value used for @_SysEndTime_: default '99991231 23:59:59.997'
--@_isActive_	Fieldname for HistIsActive: -- default "" when @SysHistorisation='N' -- default "_isActive_" when @SysHistorisation='Y' or 'SCD2'

★ Weitere Notizen siehe [TSQL: \(Automatic\) MERGE](#)