db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

Temporal Tables – Useful feature?

Konfiguration / Aufbau und Verwendung von Temporal Tables

Demos und Präsentation:
http://bit.ly/Session_Temporal_Tables
Demodatenbank(en)
https://goo.gl/WNepvY

Autor: Uwe Ricken
©: db Berater GmbH (2017)

Referentenvorstellung

http://www.db-berater.de
info@db-berater.de

- Uwe Ricken
  - Microsoft Certified Master – SQL Server 2008
  - Microsoft Certified Solution Master- Charter: Data Platform
  - Most Valuable Professional (MVP)

- Kommunikation

  Email: uwe.ricken@db-berater.de
  Twitter: @dbberater
  WWW: http://www.db-berater.de
  Blog: http://www.sqlmaster.de

Microsoft CERTIFIED Master
SQL Server® 2008

Microsoft Most Valuable Professional

©: db Berater GmbH (2017)

Agenda

- Definition einer “System Versioned” Temporal Table

- System Versioned Temporal Tables in SQL Server 2016
  - Umbenennen von Objekten / Attributen, wenn Temporal Tables im Spiel sind
  - Temporal Tables und die Verwendung von NULL Werten?
  - Temporal Tables im Einklang mit Triggern?
  - Temporal Tables in Verbindung mit InMemory-Tabellen
  - Temporal Tables und berechnete Spalten
  - Temporal Tables und Security

©: db Berater GmbH (2017)

Was ist eine System Versioned Temporal Table?

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Eine System-Versioned Temporal Table ist eine Benutzertabelle für die Speicherung aller Datenänderungen
- Jede System Versioned Temporal Table hat zwei explizit definierte Periodenattribute für die Bestimmung des zeitlichen Gültigkeitsbereichs.
- Zusätzlich wird eine Referenz zu einer weiteren Tabelle mit identischem Schema benötigt. Diese History Table wird verwendet, um alle vorherigen Versionen eines Datensatzes mit zeitlichem Gültigkeitsbereich zu speichern.

©: db Berater GmbH (2017)

Einsatzgebiet einer System Versioned Temporal Tables?

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Datenanalyse mit Bezug zu historischen Daten wird immer mehr vom Business gefordert
- Auditing aus forensischen Gründen, sofern notwendig
- Rekonstruktion von Daten zu jedem vorherigen Zeitpunkt
- Kalkulation von Trends über eine zeitliche Periode

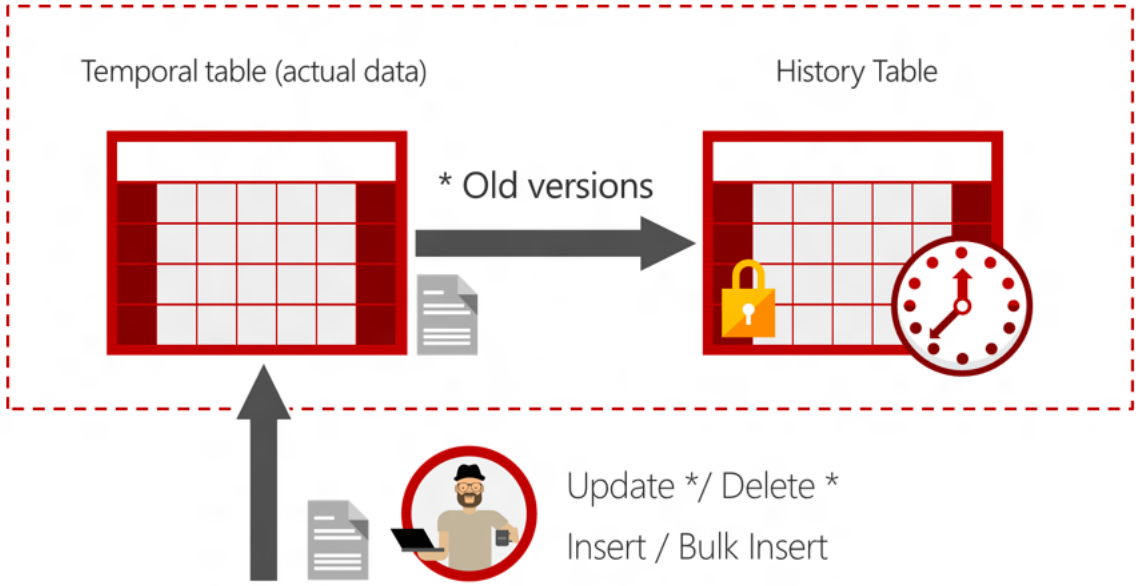
©: db Berater GmbH (2017)

Wie funktionieren System Versioned Temporal Tables?

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- System Versioning für eine Tabelle wird immer als “Tabellenpaar” implementiert.
- Zwei Attribute mit dem Datentypen `datetime2` werden zur Bestimmung der zeitlichen Gültigkeit eines Datensatzes verwendet
- Optional können diese Attribute als HIDDEN gekennzeichnet werden!



https://msdn.microsoft.com/en-us/library/dn935015.aspx

©: db Berater GmbH (2017)

Voraussetzungen/Einschränkungen für eine System Versioned Temporal Table

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Eine PRIMARY KEY-Einschränkung muss vorhanden sein!
- Die Periodenspalten müssen den Datentypen [datetime2] verwenden!
- Bei Verwendung einer selbstdefinierten History Table müssen Schema und Tabellenname angegeben werden!
- FILETABLE kann in System Versioned Temporal Tables nicht eingesetzt werden!
- System Versioned Temporal Tables müssen in der gleichen Datenbank vorhanden sein!
- Die History Table darf keine Constraints (PK, FK, DEFAULTS, …)!

- System Versioned Temporal Tables verwenden **PAGE-Compression!**

©: db Berater GmbH (2017)

Voraussetzungen/Einschränkungen für eine System Versioned Temporal Table

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Indizierte Views können keine Syntax von Temporal Tables (FOR SYTEM TIME OF…) verwenden!
- Die Periodenspalten können nicht unmittelbar bearbeitet werden!
- TRUNCATE TABLE ist für Temporal Tables nicht erlaubt!
- Direkte Bearbeitung von Daten in der History Table ist nicht möglich!
- DELETE und UPDATE CASCADE ist nicht erlaubt, wenn die Tabelle die Detailtabelle in einer Fremdschlüsselbeziehung ist!
- INSTEAD_OF Trigger sind nicht erlaubt!

©: db Berater GmbH (2017)

Erstellen einer System Versioned Temporal Table

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

```
CREATE TABLE
(
   PERIOD FOR
   SYSTEM_TIME
);

ALTER TABLE
ADD PERIOD
FOR SYSTEM_TIME;
```

DDL

- INSERT
- UPDATE
- DELETE
- MERGE

DML

```
SELECT *
FROM TABLE
WHERE ...
ORDER BY;
```

QUERYING

```
SELECT *
FROM TABLE
FOR SYSTEM_TIME
...
WHERE
ORDER BY;
```

TEMPORAL QUERYING

©: db Berater GmbH (2017)

Erstellen einer System Versioned Temporal Table

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Temporal Tables können auf drei verschiedene Arten erzeugt werden:
  - System Versioned Temporal Table mit anonymisierter History Table
    Nur das Schema der Benutzertabelle wird definiert. Das System erstellt eine korrespondierende History-Tabelle mit einem automatisch generierten Namen.
  - System Versioned Temporal Table Unter Angabe einer explizit benutzerdefinierten History Table.
  - System Versioned Temporal Table mit VORHER definierter und implementierter History Table

©: db Berater GmbH (2017)

History Table

```
CREATE TABLE [dbo].[Department]
(
   [DeptID]          int          NOT NULL     PRIMARY KEY CLUSTERED,
   [DeptName]        varchar(50)  NOT NULL,
   [ManagerID]       INT          NULL,
   [ParentDeptID]    int          NULL,
   [SysStartTime]    datetime2    GENERATED ALWAYS AS ROW START   NOT NULL,
   [SysEndTime]      datetime2    GENERATED ALWAYS AS ROW END     NOT NULL,
   PERIOD FOR SYSTEM_TIME ([SysStartTime], [SysEndTime])
)
WITH (SYSTEM_VERSIONING = ON [(HISTORY_TABLE = [history].[Department])]);
```


©: db Berater GmbH (2017)

Aktualisierung aus History Table

- UPDATE kann dazu verwendet werden, um Daten in der Tabelle zu einem bestimmten Zeitpunkt wiederherzustellen!

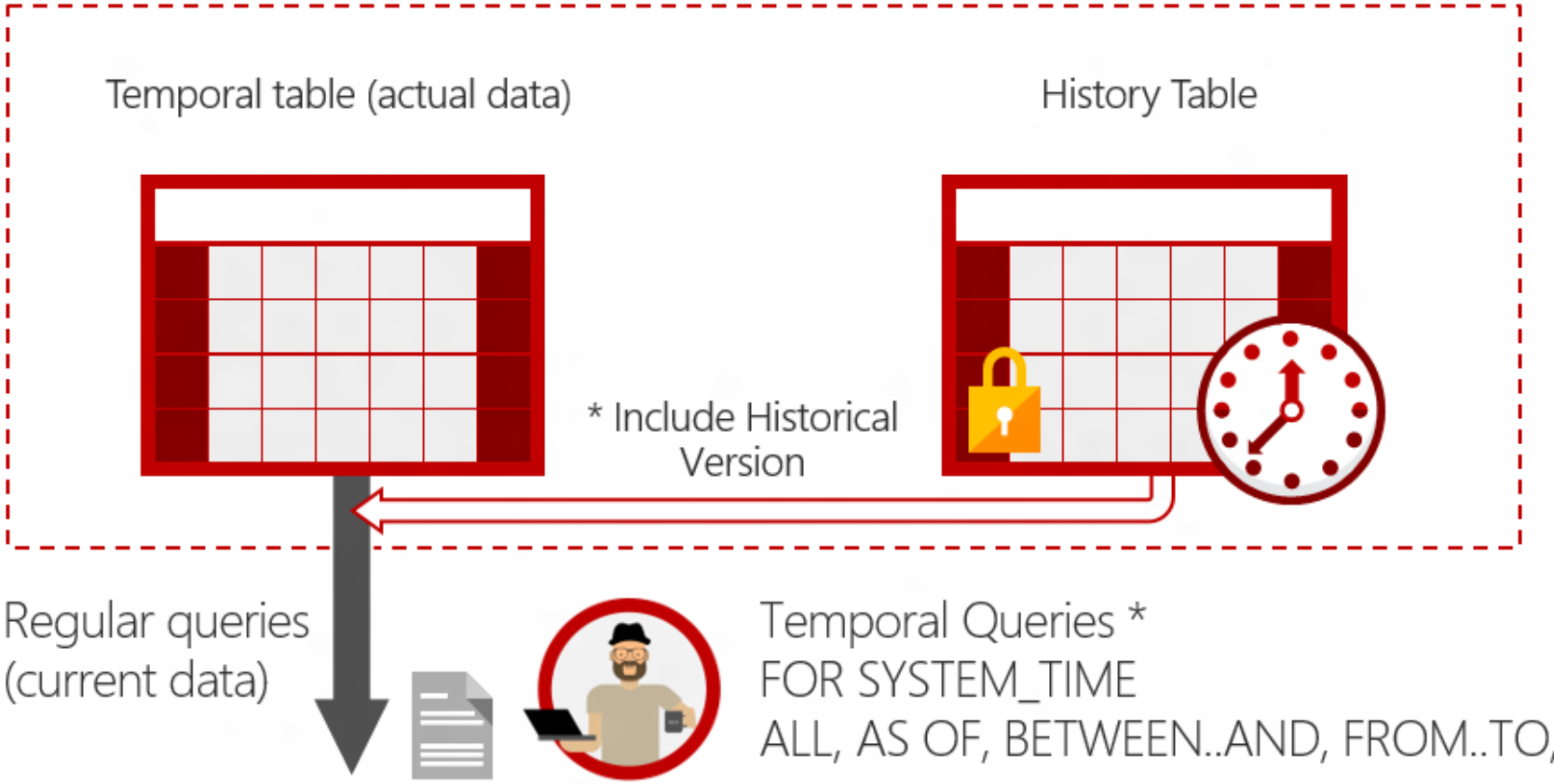
```
UPDATE      [dbo].[Department]
SET         [D].[DeptName] = [history].[DeptName]
FROM        [dbo].[Department]
FOR SYSTEM_TIME AS OF '2016-11-22T09:00:00' AS [history]
WHERE       [history].[DeptID] = 10
            AND [dbo].[Department].[DeptID] = 10;
```


©: db Berater GmbH (2017)

Abfragen von System Versioned Temporal Tables?

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de



https://msdn.microsoft.com/en-us/library/dn935015.aspx

©: db Berater GmbH (2017)

Abfragen von System Versioned Temporal Tables?

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

Ausdruck	Qualifying Rows	Beschreibung
AS OF <date_time>	SysStartTime <= date_time AND SysEndTime > date_time	Gibt eine Tabelle mit Zeilen zurück, die die Werte enthalten, die zum angegebenen Zeitpunkt in der Vergangenheit real (aktuell) waren.
FROM … TO …	SysStartTime < end_date_time AND SysEndTime > start_date_time	Gibt eine Tabelle mit den Werten für alle Zeilenversionen zurück, die innerhalb des angegebenen Zeitbereichs aktiv waren.
BETWEEN … TO …	SysStartTime <= end_date_time AND SysEndTime => start_date_time	Wie <FROM…TO…>! Die zurückgegebene Tabelle enthält jedoch Zeilen, die an dem durch den <end_date_time>-Endpunkt definierten oberen Grenzwert aktiv wurden.
CONTAINED IN (…)	SysStartTime >= start_date_time AND SysEndTime <= end_date_time	Gibt eine Tabelle mit den Werten für alle Zeilenversionen zurück, die innerhalb des von den zwei Datums-/Uhrzeitwerten für das Argument CONTAINED IN definierten Zeitbereichs geöffnet und geschlossen wurden.
ALL	All rows	Gibt die Menge aller Zeilen zurück, die der aktuellen und der Verlaufstabelle angehören.

Autor: Uwe Ricken

©: db Berater GmbH (2017)

Metadatenänderungen in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Änderung der Einschränkung NULL zu NOT NULL
  - NULL wird zu NOT NULL in einer leeren Tabelle
  - NULL wird zu NOT NULL in einer gefüllten Tabelle
  - NOT NULL wird zu NULL in einer gefüllten Tabelle

http://www.db-berater.de/2016/06/temporal-tables-behandlung-von-null-einschrnkungen/

©: db Berater GmbH (2017)

Metadatenänderungen in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Hinzufügen von neuen Spalten
  - Neue Spalten mit `NULL`able Eigenschaften
  - Neue Spalten mit `NOT NULL`able Eigenschaften
  - Speicherung von DEFAULT-Einschränkungen in System Versioned Temporal Table und History Table

- Löschen von bestehenden Spalten

©: db Berater GmbH (2017)

Metadatenänderungen in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Änderung von Datentypen
  - Verändern von Datentypen mit fester Länge
  - Verändern von Datentypen mit variabler Länge

©: db Berater GmbH (2017)

Umbenennung von Objekten in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Umbenennen von Tabellennamen
- Umbenennen von Attribut aus “System Versioned Temporal Table”
- Umbenennen von Attribut aus “History Table”
- Umbenennen von Attributen bei deaktivierter “System Versioned Temporal Table”

http://www.db-berater.de/2016/06/temporal-tables-umbenennung-von-metadaten/

©: db Berater GmbH (2017)

Trigger in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Trigger erzeugen einen weiteren Datensatz in der History Table!

- Instead of Trigger sind in System Versioned-Temporal Tables nicht erlaubt!

http://www.db-berater.de/2016/07/temporal-tables-verwendung-von-triggern/

©: db Berater GmbH (2017)

Variable Primärschlüssel in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

Autor: Uwe Ricken

©: db Berater GmbH (2017)

Sicherheitskonfiguration von System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Das Aktivieren und Deaktivieren von SYSTEM_VERSIONING erfordert die CONTROL-Berechtigung sowohl für die aktuelle Tabelle als auch für die Verlaufstabelle
- Wenn SYSTEM_VERSIONING ON ist, können Daten in der History Tabelle nicht geändert werden
- **Das Abfragen der History Tabelle erfordern SELECT-Berechtigung**

©: db Berater GmbH (2017)

Dynamic Data Masking in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Die dynamische Datenmaskierung (DDM) beschränkt die Offenlegung vertraulicher Daten, indem sie für nicht berechtigte Benutzer maskiert werden.
- Eine zentrale Datenmaskierungsrichtlinie wirkt sich direkt auf vertrauliche Felder in der Datenbank aus.
- Festlegen berechtigter Benutzer oder Rollen, die über Zugriff auf vertrauliche Daten verfügen.
- DDM umfasst Funktionen für die vollständige und teilweise Maskierung sowie eine willkürliche Maske für numerische Daten.
- Einfache Transact-SQL-Befehle dienen zum Definieren und Verw .
- Dynamic Data Masking verschlüsselt nicht. Die Daten können m Rückschluss- oder Brute-Force Verfahren ermittelt werden!

©: db Berater GmbH (2017)

Row Level Security in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Temporale Tabellen sind mit RLS kompatibel.
- Sicherheitsprädikate für die aktuelle Tabelle werden jedoch nicht automatisch in die Verlaufstabelle repliziert.
- Für jede Tabelle muss ein Sicherheitsprädikat einzeln hinzugefügt werden.

©: db Berater GmbH (2017)

Berechnete Spalten in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Solange eine Tabelle als System Versioned Temporal Table implementiert ist, ist das Hinzufügen von berechneten Spalten nicht möglich
- In die System Versioned Temporal Table können berechnete Spalten eingefügt werden.
- Die History Table muss eine entsprechende NICHT BERECHNETE Spalte enthalten!

©: db Berater GmbH (2017)

Historische Bestandsdaten mit System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

- Es ist möglich, bestehende historische Daten mit System Versioned Temporal Tables zu implementieren.
- Folgende Voraussetzungen sind dafür – zusätzlich – erforderlich:
  - Die Gültigkeitsdaten in den historischen Daten dürfen nicht in der Zukunft liegen
  - Die Gültigkeitsdaten in den historischen Daten dürfen sich nicht überschneiden

©: db Berater GmbH (2017)

In Memory in System Versioned Temporal Tables

db Berater GmbH
Planung – Installation – Optimierung

http://www.db-berater.de
info@db-berater.de

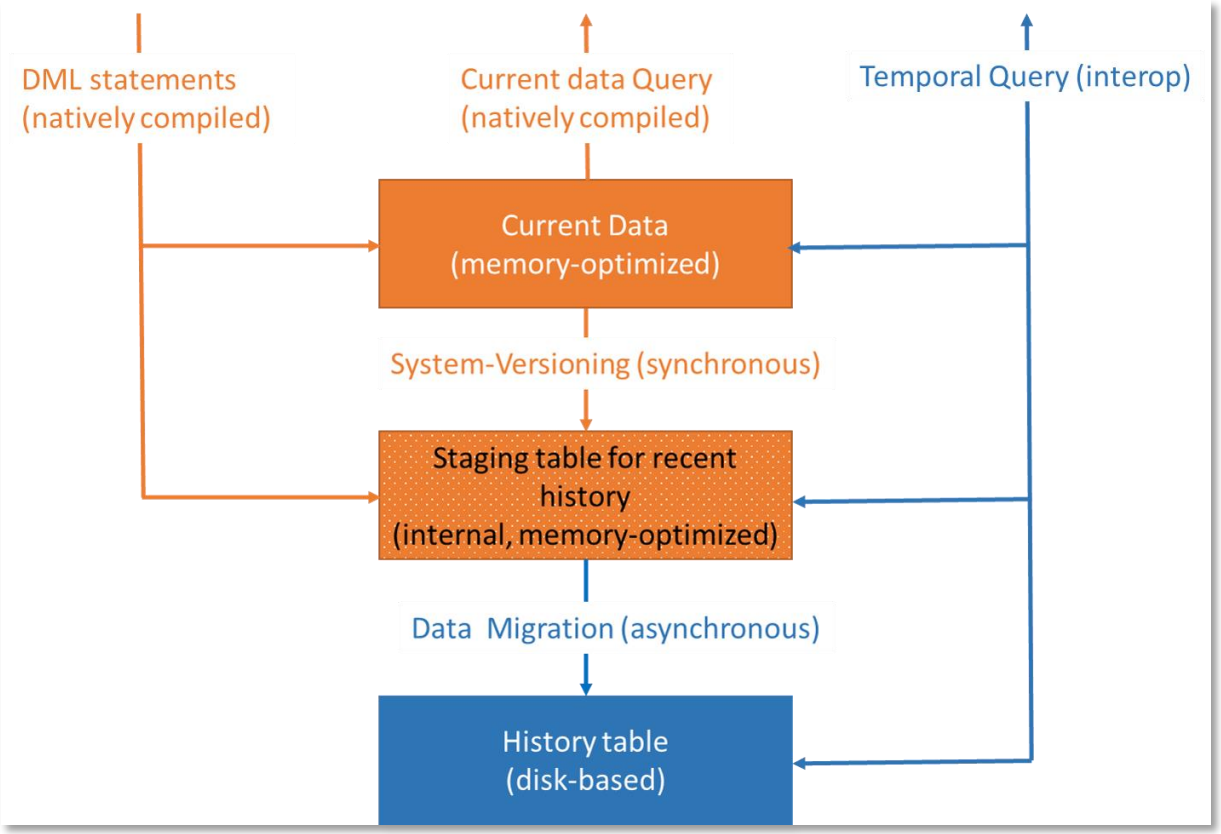


Abbildung: https://msdn.microsoft.com/de-de/library/mt590207.aspx

©: db Berater GmbH (2017)

- Nur dauerhafte speicheroptimierte Tabellen können der Systemversionsverwaltung unterliegen (**DURABILITY = SCHEMA_AND_DATA)**
- Die History Table für speicheroptimierte Tabellen muss datenträgerbasiert sein
- Wenn **SYSTEM_VERSIONING = ON**, wird automatisch eine interne speicheroptimierte Stagingtabelle erstellt
- Daten aus der internen **speicheroptimierten** Stagingtabelle werden vom asynchronen Datenleerungstask in die Verlaufstabelle verschoben
- Wenn **SYSTEM_VERSIONING = OFF** wird, wird der gesamte Inhalt des internen Stagingpuffers in die Verlaufstabelle verschoben

©: db Berater GmbH (2017)

Herzlichen Dank

uwe.ricken@db-berater.de
@dbberater
http://www.sqlmaster.de