Temporal Tables
The New Hotness in Data Auditing

John Morehouse
Consultant
Denny Cherry & Associates Consulting


john@dcac.com


http://linkedin.com/in/sqlrus


@SqlRUs


http://www.sqlrus.com

Who Am I?

- Leader of the Louisville SQL Server/Power BI User Group
- Organizer/Speaker of SQL Saturday's & other conferences
- Heavily involved with SQL PASS
- Microsoft Data Platform MVP
- Friend of Redgate 2015 - 2018
- Idera ACE 2016
- SentryOne Product Advisory Council

Where Am I?

The vetted and certified experts at Denny Cherry and Associates Consulting assist companies with attaining IT goals such as HA, scalability, SQL Server virtualization, migration, and acceleration reliably while finding ways to save on costs. With clients ranging from Fortune 50 corporations to small businesses, their commitment to each is the same: to provide a deft, high-speed IT environment that leverages every aspect of their platform: from architecture, infrastructure, to network.

Visit DCAC at http://www.dcac.com

Quick Check

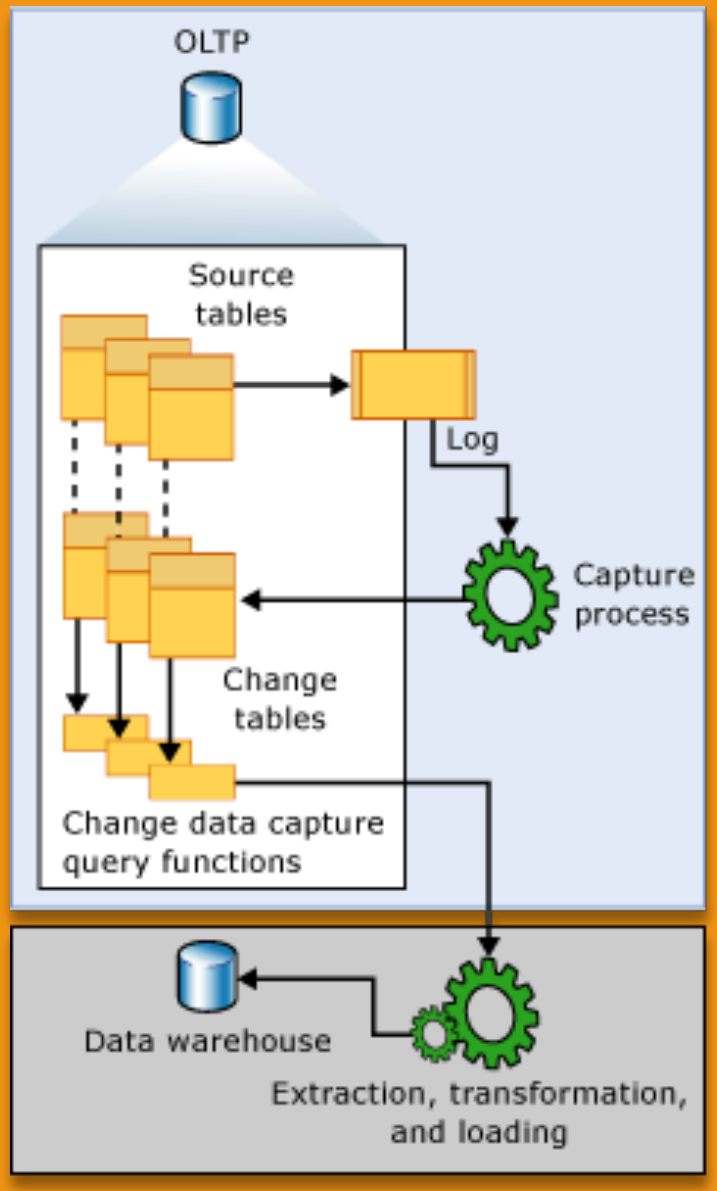
- How many:
  - DBA's
  - Developers
  - BI/DWH
  - Other
- Please make sure to fill out evaluations
- Ask questions!!!

Stop me if you've heard….

I want to be able to save everything for all time, including all data changes.

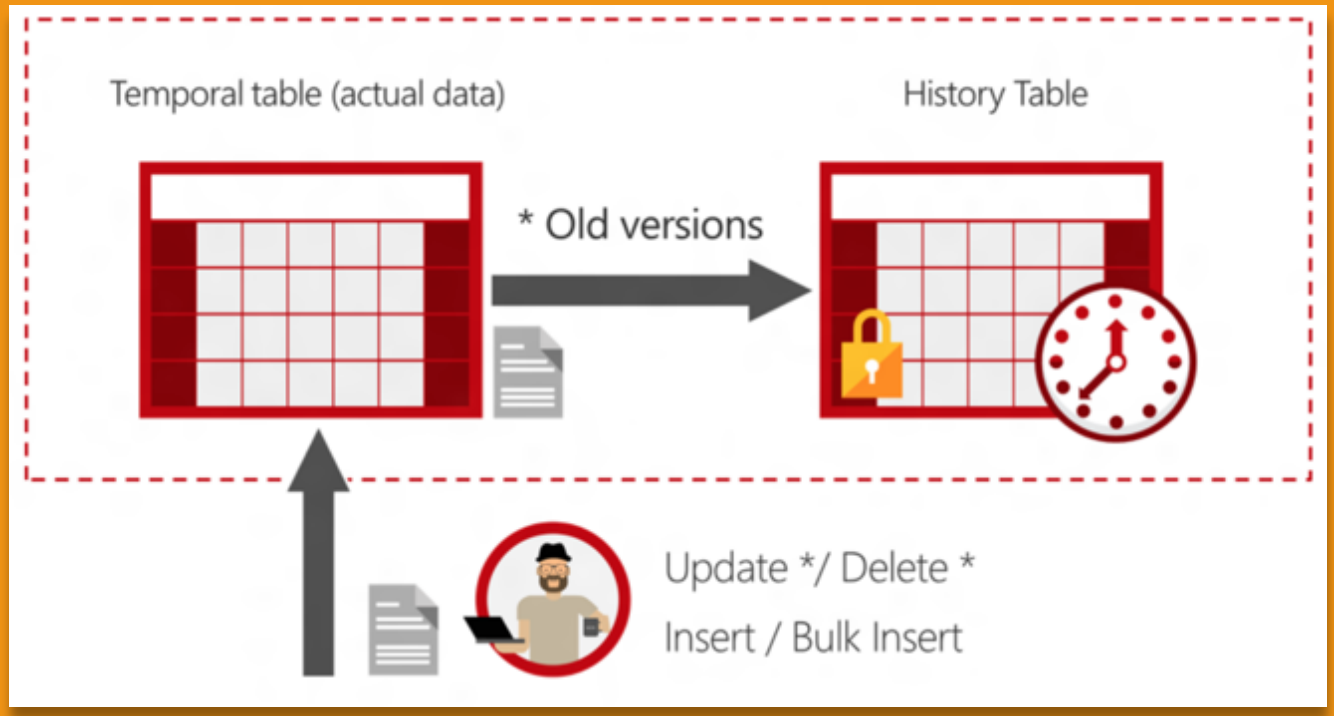
Old Hotness

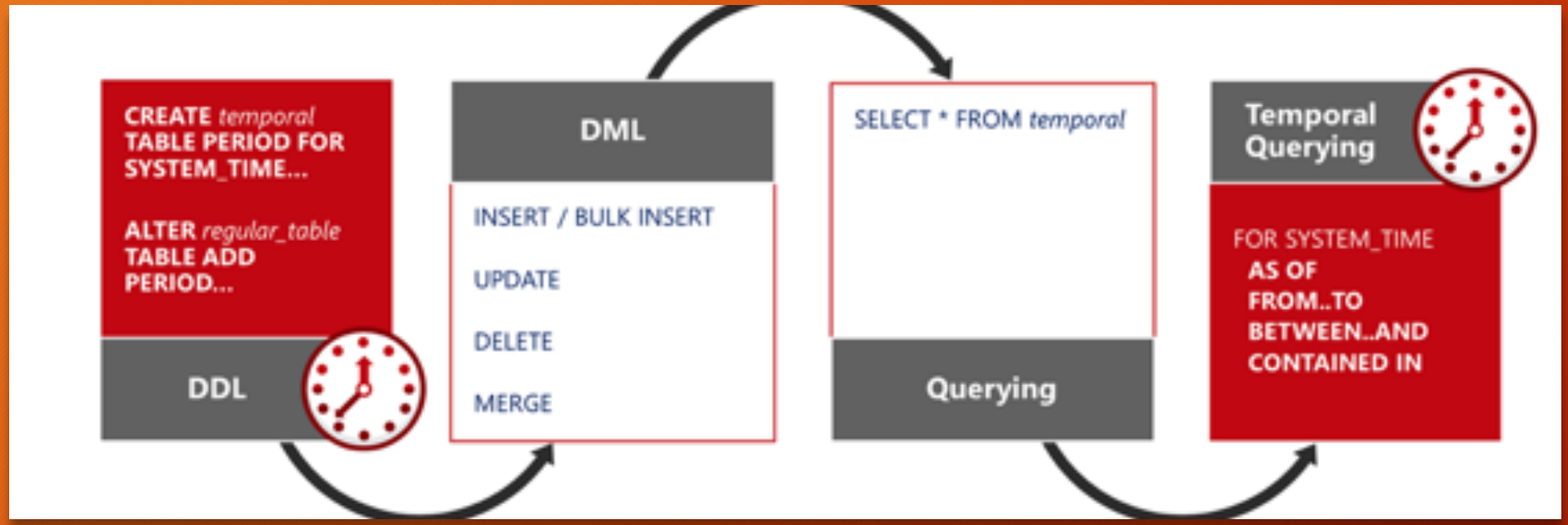
- Triggers
- Change Data Capture
- Database architecture



New Hotness

- SQL Server 2016+
- On-prem/Azure SQL DB
- All editions
- Transparent
- Easier to implement





Temporal Tables

When to use Temporal Tables?

- Auditing all data changes/forensics
- Reconstructing data from the past
- Calculating trends over time
- Slowly changing dimensions
- Recovering from accidental data deletion

Creating a Temporal Table

```
CREATE TABLE Employee
(
    [EmployeeID] INT NOT NULL PRIMARY KEY CLUSTERED
    , [Name] NVARCHAR(100) NOT NULL
    , [Position] VARCHAR(100) NOT NULL
    , [Department] VARCHAR(100) NOT NULL
    , [Address] NVARCHAR(1024) NOT NULL
    , [AnnualSalary] DECIMAL(10,2) NOT NULL
    , [ValidFrom] DATETIME2 (2) GENERATED ALWAYS AS ROW START
    , [ValidTo] DATETIME2 (2) GENERATED ALWAYS AS ROW END
    , PERIOD FOR SYSTEM_TIME (ValidFrom, ValidTo)
)
WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = dbo.EmployeeHistory))
```

Temporal Tables require the use of a primary key

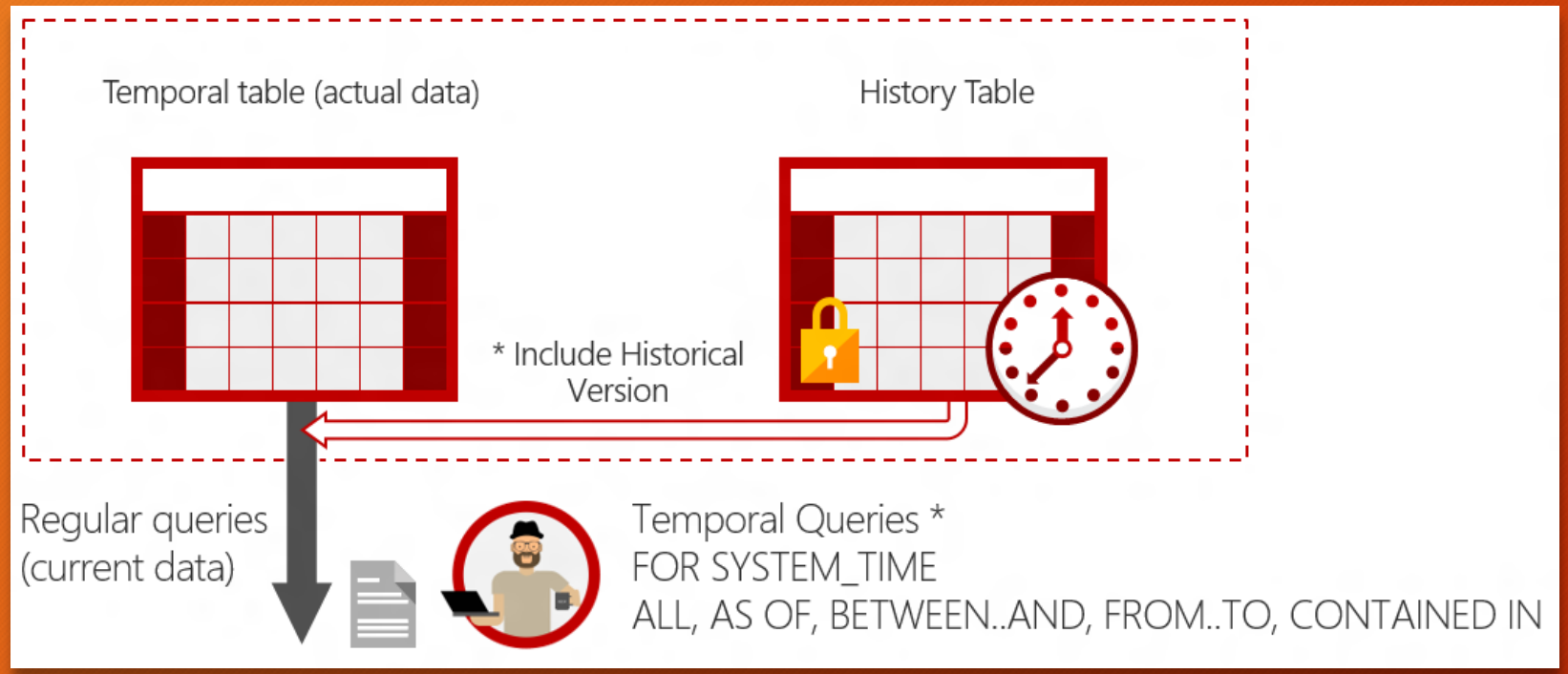
On insert SQL sets value for SysStartTime to begin time of the current transaction

On update SQL moves previous record to history table and marks record as closed

System Versioning keyword sets temporal on

Deletes stores previous value in history table and marks record as closed

Querying Temporal Data



- ALL
- AS OF
- BETWEEN...AND
- FROM...TO
- CONTAINED IN

```
SELECT * FROM Employee
    FOR SYSTEM_TIME
        BETWEEN '2017-01-01 00:00:00.0000000' AND '2018-01-01 00:00:00.0000000'
            WHERE EmployeeID = 1000 ORDER BY ValidFrom;
```

Query Qualifiers

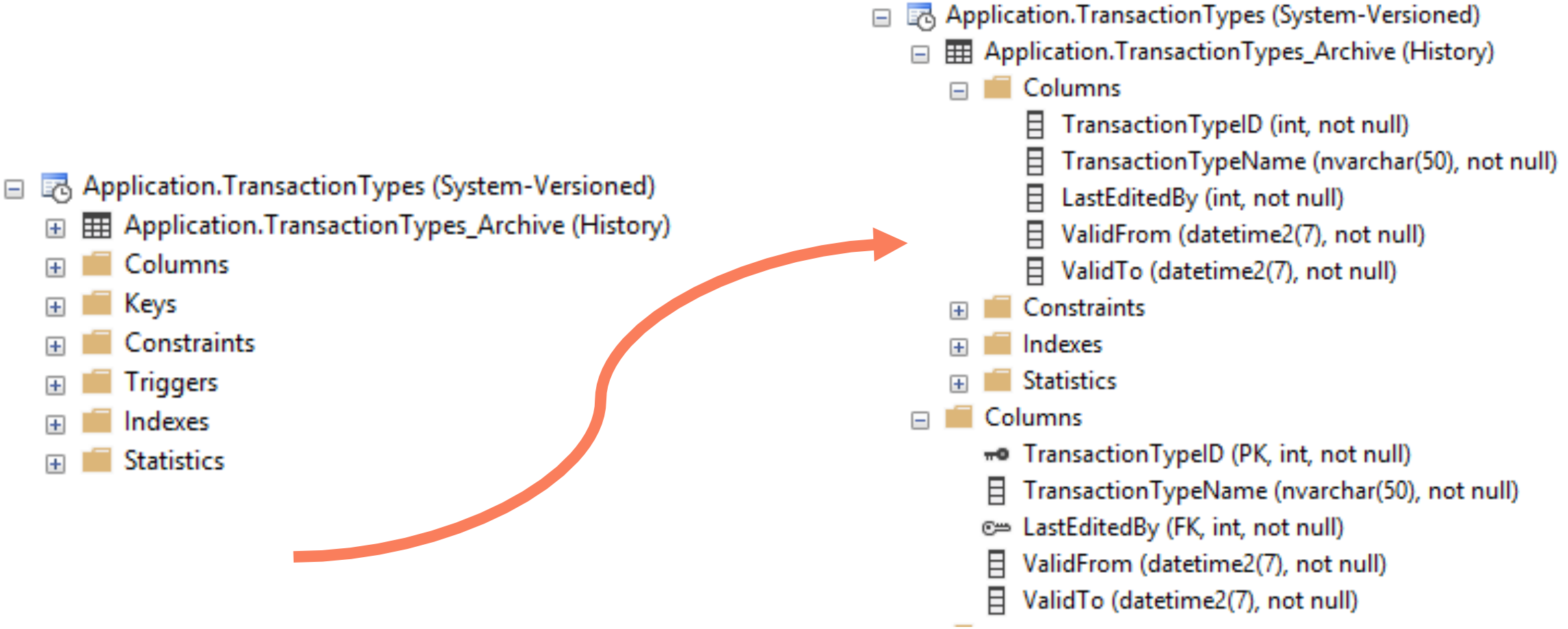
AS OF <date_time>

FROM <start_date_time> TO <end_date_time>

BETWEEN <start_date_time> AND <end_date_time>

CONTAINED IN (<start_date_time>,<end_date_time>)

ALL



Object Explorer Changes

temporal_type	**tinyint**	**Applies to**: SQL Server 2016 (13.x) through SQL Server 2017 and Azure SQL Database. The numeric value representing the type of table: 0 = NON_TEMPORAL_TABLE 1 = HISTORY_TABLE 2 = SYSTEM_VERSIONED_TEMPORAL_TABLE
temporal_type_desc	**nvarchar(60)**	**Applies to**: SQL Server 2016 (13.x) through SQL Server 2017 and Azure SQL Database. The text description of the type of table: NON_TEMPORAL_TABLE HISTORY_TABLE SYSTEM_VERSIONED_TEMPORAL_TABLE
histo… table_id	**int**	**Ap… to:** S… Server 2016 (13.x) thro… SQL S… …and Azu… …ba…

Updates to sys.tables

```
3  SELECT [Name]
4       , SCHEMA_NAME(schema_id) AS [Schema Name]
5       , temporal_type
6       , temporal_type_desc
7  FROM WideWorldImporters.sys.tables
8  WHERE temporal_type IN (1,2)  -- 1 = History Table, 2 = parent table
9
```

100 %

Results | Messages

	Name	Schema Name	temporal_type	temporal_type_desc
1	Colors	Warehouse	2	SYSTEM_VERSIONED_TEMPORAL_TABLE
2	Colors_Archive	Warehouse	1	HISTORY_TABLE
3	PackageTypes	Warehouse	2	SYSTEM_VERSIONED_TEMPORAL_TABLE
4	PackageTypes_Archive	Warehouse	1	HISTORY_TABLE
5	StockGroups	Warehouse	2	SYSTEM_VERSIONED_TEMPORAL_TABLE
6	StockGroups_Archive	Warehouse	1	HISTORY_TABLE
7	StateProvinces	Application	2	SYSTEM_VERSIONED_TEMPORAL_TABLE
8	StateProvinces_Archive	Application	1	HISTORY_TABLE

Updates to sys.tables

Data Purging

- Partitioning
- Retention Policies
  - SQL Server 2017 CP1
- Custom Scripts
  - Must turn off system versioning
  - Lock the table!


Demo

By default history tables are page compressed

History table must be in same database is base table

Truncate table is not supported

ON DELETE CASCADE and ON UPDATE CASCADE are not supported

Always On Availability Groups are supported

Change Data Capture and Change Tracking Supported on the base table only

Columnstore indexes are supported for both tables

In-Memory is supported

Temporal Table Considerations

DBCC CHECKCONSTRAINTS

The names and number of columns is the same in both the current table and the history table.

The datatypes match for each column between the current table and the history table.

The period columns are set to **NOT NULL**.

The current table has a primary key constraint and the history table does not have a primary key constraint.

No **IDENTITY** columns are defined in the history table.

No triggers are defined in the history table.

No foreign keys are defined in the history table.

No table or column constraints are defined on the history table. However, default column values on the history table are permitted.

The history table is not placed in a read-only filegroup.

The history table is not configured for change tracking or change data capture.

Questions

Resources

- https://docs.microsoft.com/en-us/sql/relational-databases/tables/getting-started-with-system-versioned-temporal-tables?view=sql-server-2017
- https://docs.microsoft.com/en-us/sql/relational-databases/tables/changing-the-schema-of-a-system-versioned-temporal-table?view=sql-server-2017
- https://docs.microsoft.com/en-us/sql/relational-databases/tables/temporal-table-usage-scenarios?view=sql-server-2017
- https://docs.microsoft.com/en-us/sql/relational-databases/tables/temporal-table-security?view=sql-server-2017
- https://sqlperformance.com/2016/06/sql-server-2016/temporal-table-query-plan-behaviour

THANK YOU!!!!


john@dcac.com


http://linkedin.com/in/sqlrus


@SqlRUs


http://www.sqlrus.com