

Marcelo Fernandes



Temporal table e Query Store - Extraindo o máximo de vantagem do SQL Server

Begin transaction SP_WHO

- ▶ SQL Server Specialist
- ▶ Microsoft Most Valuable Professional (MVP)
- ▶ Frequent speaker (on-line and presential events)
- ▶ Books Co-author:
 - ▶ **SQL Server 2014: Alta Disponibilidade na Prática com AlwaysOn Failover Cluster Instances**
 - ▶ **SQL Server: Além do conceito SQL Server Blog Post Collection**
 - ▶ **SQL Server: Além do conceito SQL Server Blog Post Collection Vol. 2**
- ▶ Contacts
 - ▶ <http://marcelodba.wordpress.com>
 - ▶ fernandes_dba@hotmail.com
 - ▶ @marcelodba



Patrocinadores

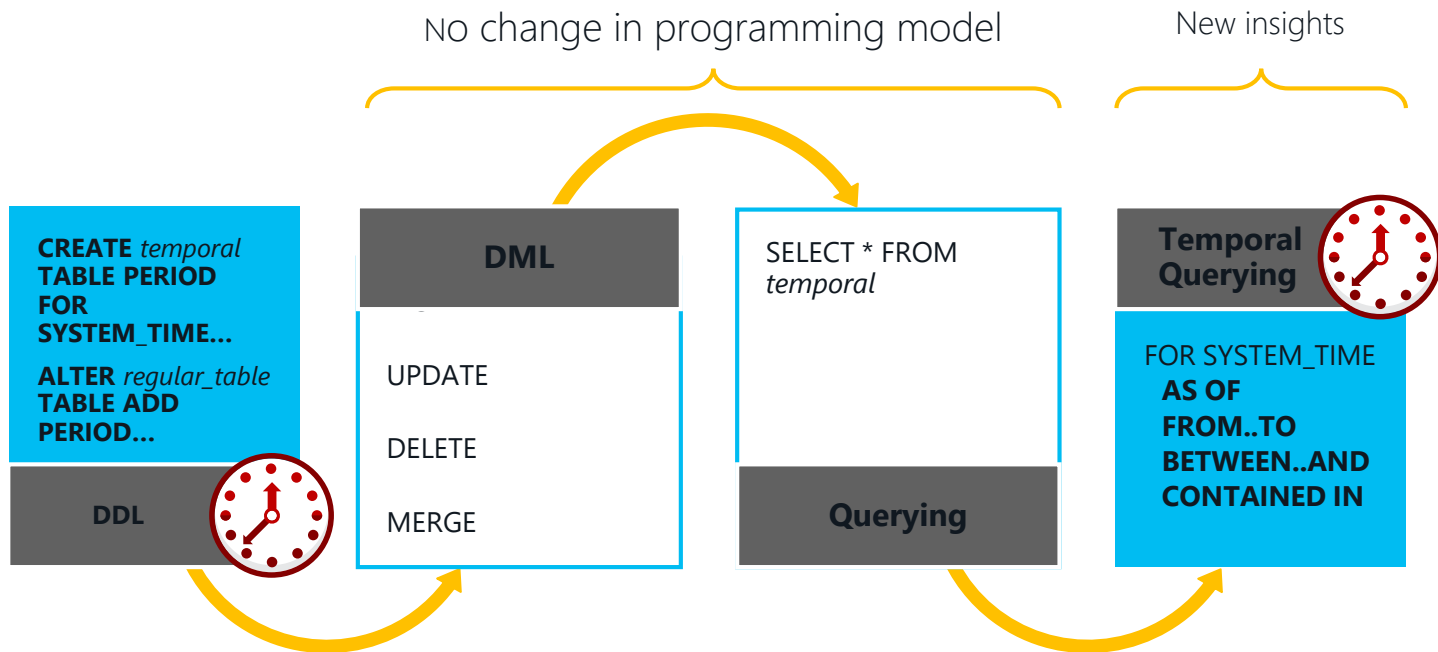


Agenda

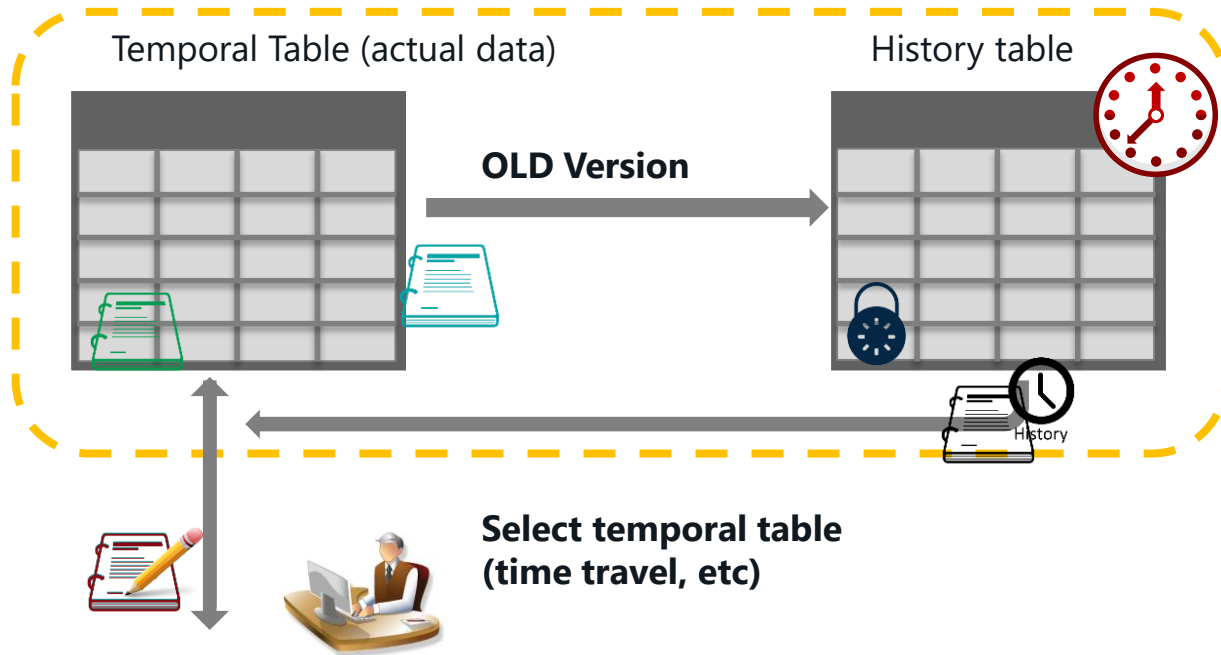
- Temporal Table – how it works?
- Temporal Table – Why?
- Temporal Table - Limitations
- Query Store – How it works
- Query Store – Why?
- Query Store - Limitations



Temporal Table – How it works?



Temporal Table – How it works?



Temporal Table – Why?

DEMO



Temporal Table - Limitations

- Base table must have PK
- History table cannot have PK
- History table must be in same database as current table
- Querying temporal tables over Linked Server not allowed
- No data modification directly on History tables
- No INSERT/UPDATE directly on SYSTEM_TIME columns
- Cannot be FILETABLE, cannot have FILESTREAM
- AlwaysON: Fully supported.

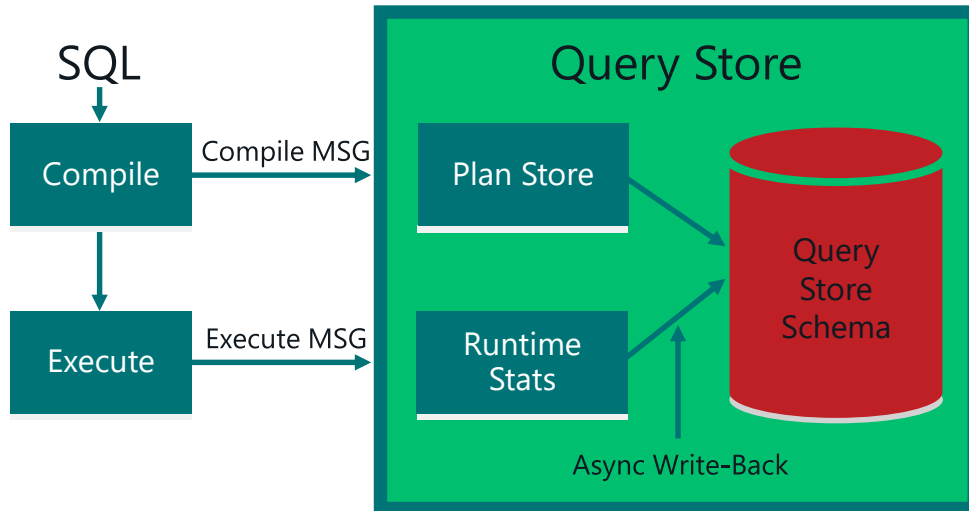


Temporal Table - Limitations

- No constraints on history table (except DEFAULT, which is ineffective)
- Supported only for DURABILITY = SCHEMA_AND_DATA
 - History tables must be disk based.
- Truncate table not allowed
- No INSTEAD OF triggers
- No triggers on history table
- No history table chaining
- Manual changes to system clock can cause failures
- History table is PAGE compressed by default.



Query Store – How it works?

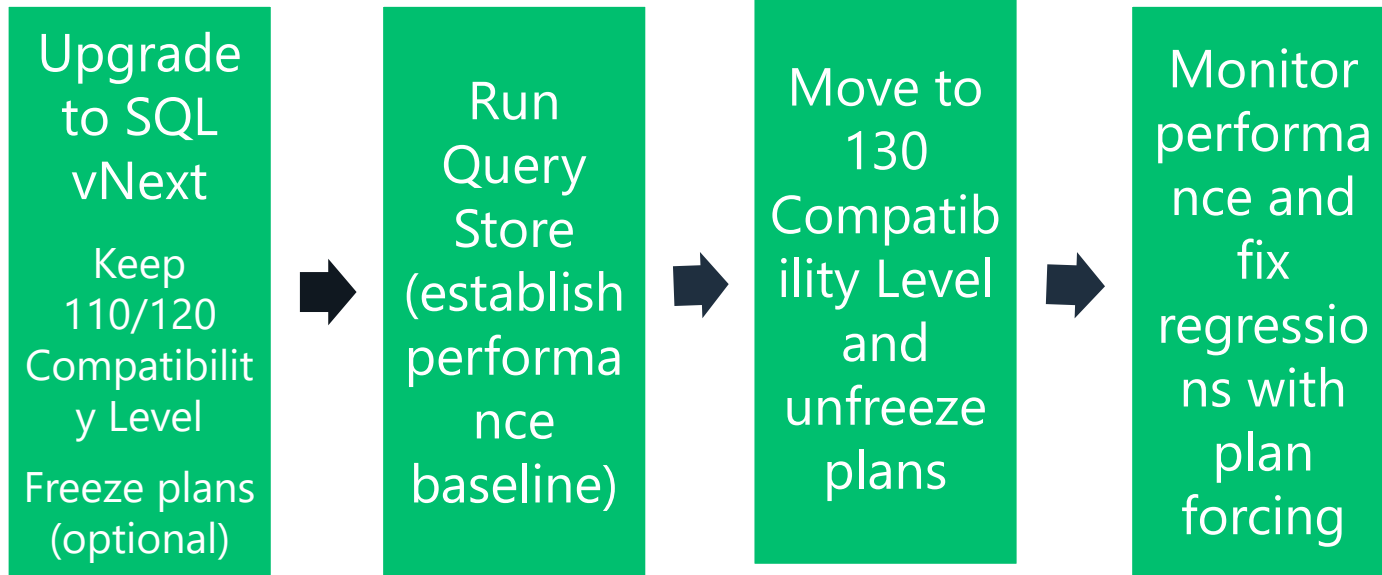


Durability latency controlled by DB option
DATA_FLUSH_INTERNAL_SECONDS

- Collects query texts (+ all relevant properties)
- Stores all plan choices and performance metrics
- Works across restarts / upgrades / recompiles
- Simplifies performance troubleshooting
- New Views
- Intuitive and easy plan forcing



Query Store – Why?



Query Store – Why?

- Explore the queries execution
- Resource consuming queries
- Figure out query plan
- Identify possible performance degradation queries
- Figure out why regressions happen
- Force the query processor to use a particular plan
- Query Store is accessible through Transact-SQL.
- Identify and improve ad-hoc workloads
- Pinpoint and fix queries
- Custom reporting and/or alerting through
- Dynamic Management Views (DMVs)



Query Store – DEMO



Query Store – Limitations

- Using the Query Store with In-Memory OLTP
 - There are some specific limitations:
<https://docs.microsoft.com/en-us/sql/relational-databases/performance/using-the-query-store-with-in-memory-oltp>
- The database should be Read Write.



References

Temporal in SQL Server 2016

<https://channel9.msdn.com/Shows/Data-Exposed/Temporal-in-SQL-Server-2016>

Monitoring Performance By Using the Query Store

<https://docs.microsoft.com/en-us/sql/relational-databases/performance/monitoring-performance-by-using-the-query-store>





<http://www.sqlsaturday.com/618>

19/08 – Brasilia

30/09 – São Paulo

21/10 – Rio de Janeiro

18/11 - Salvador



