

Temporal Database Support

Focus on retention and performance

paSQuaLe ceglie

@pceglie

<http://www.ugiss.org/author/pceglie/>

<http://pasqualeceglie.wordpress.com>



#566 | PARMA 2016

Sponsors



November 26°, 2016



Organizers

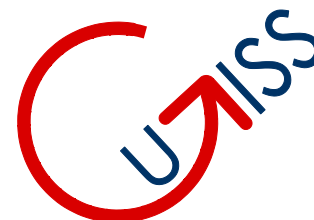


UNIVERSITÀ DEGLI STUDI DI PARMA

ENGAGE
IT SERVICES



getlatestversion.it



dan|ela
ma|visi
COMMUNICATION

November 26°, 2016



Pasquale Ceglie | @pceglie

- Microsoft MCSE – Data Management and Analytics
- Microsoft MCSE – Data Platform
- Attratto da
 - Modello Relazionale
 - Architettura DBMS
 - Tuning
 - Azure
 - Database Corruption
- NoSQL (DocumentDB, MongoDB) ... è bene sapere che esistono ☺
- Mail: pasqualeceglie@gmail.com

November 26°, 2016



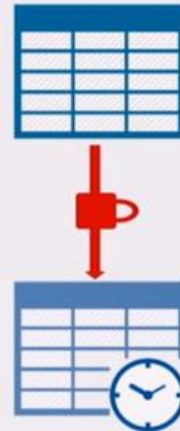
Database temporali, perché?

Le operazioni complesse diventano semplici

Analyze data history yourself

- 1 SCHEMA MAINTENANCE**
Double effort to maintain current and history.
- 2 HISTORY TRACKING**
User code is required (triggers, SPs, app).
Hard to maintain and achieve good performance.
- 3 DATA ANALYSIS**
Complex queries are required.
- 4 DATA PROTECTION**
Immutability of history data cannot be guaranteed.

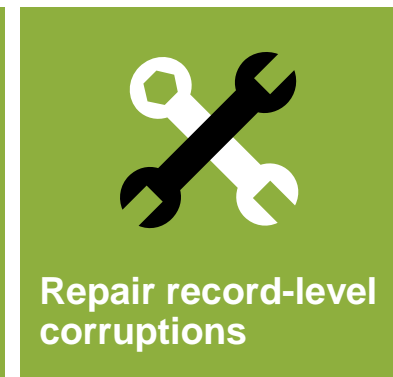
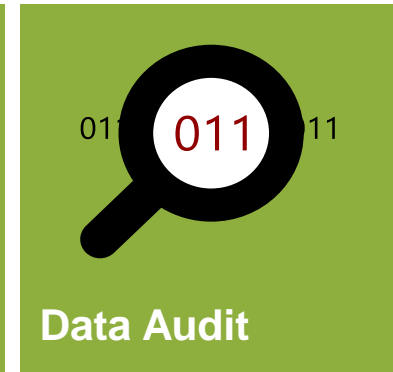
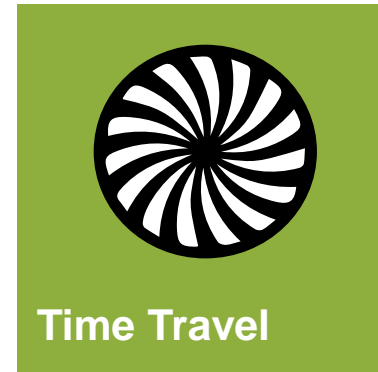
Let the temporal tables do it



- 1 SCHEMA MAINTENANCE**
Automatic and online.
- 2 HISTORY TRACKING**
Automatic and optimal.
- 3 DATA ANALYSIS**
Simple and efficient with
FOR SYSTEM_TIME clause.
- 4 DATA PROTECTION**
Out-of-box immutability of history data.

Database temporali, perché?

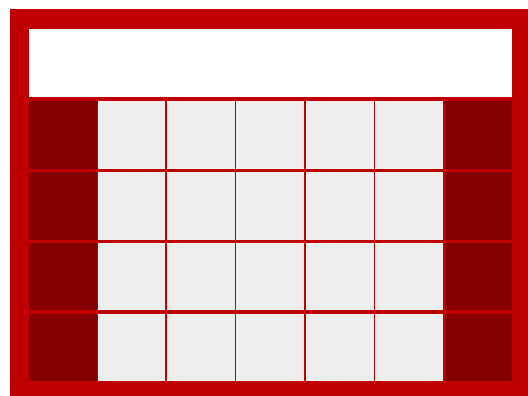
- «Workaround»
 - Complessi
 - Limitati
 - Inefficienti
- Funzionalità
 - Estensioni DML/DDL
 - ANSI SQL 2011



SQL Server 2016 rende tutto più facile
Tracking delle modifiche senza scrivere codice
Analisi del dato in qualsiasi momento nel tempo

Come funziona?

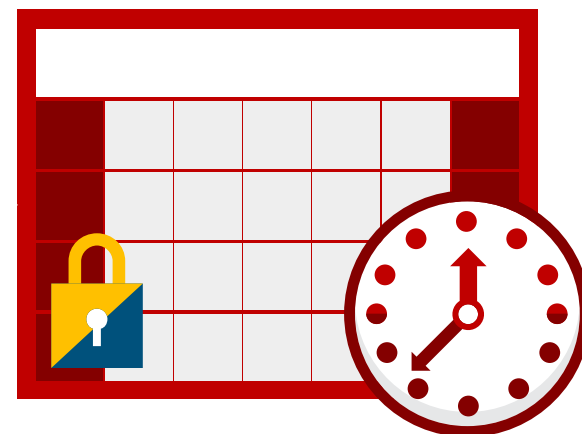
Temporal table (dati correnti)



* Vecchie versioni



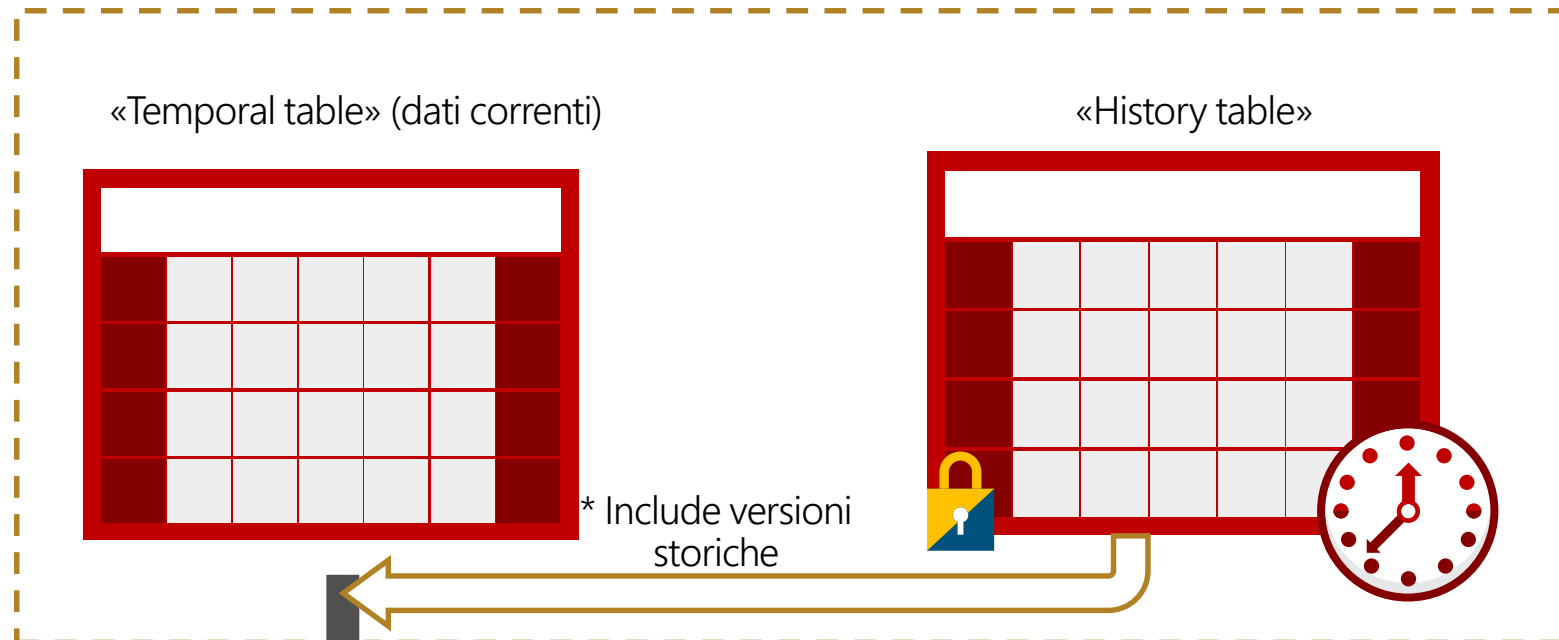
History Table



Insert / Bulk Insert

Update */ Delete */ Merge *

Query



Query normali
(dati correnti)



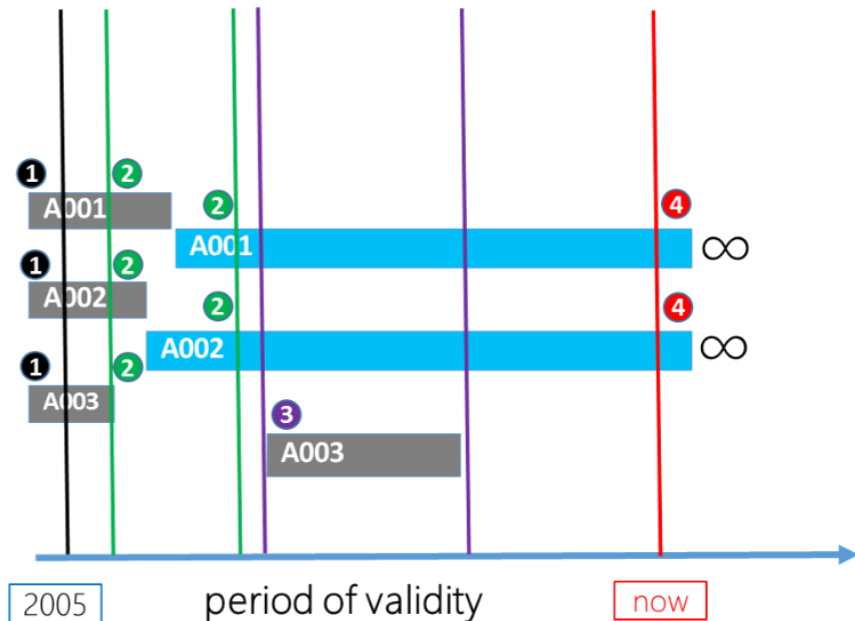
Query temporali *
FOR SYSTEM_TIME
ALL, AS OF,
BETWEEN ... AND ..., FROM ...
TO, CONTAINED IN

Predicati temporali

FOR SYSTEM_TIME

<u>EmployeeID</u>	Name	Salary	From	To
A001	John B.	62000	2005	2009
A001	Marketing	75000	2009	9999
A002	Michael P.	87500	2005	2008
A002	Michael P.	96300	2008	9999
A003	Tim W.	88400	2005	2007
A003	Tim W.	102000	2010	2012

AS OF (1) BETWEEN..AND (2) CONTAINED IN (3) current querying (4)
(without SYSTEM_TIME)



DEMO

Temporal (1) with online app

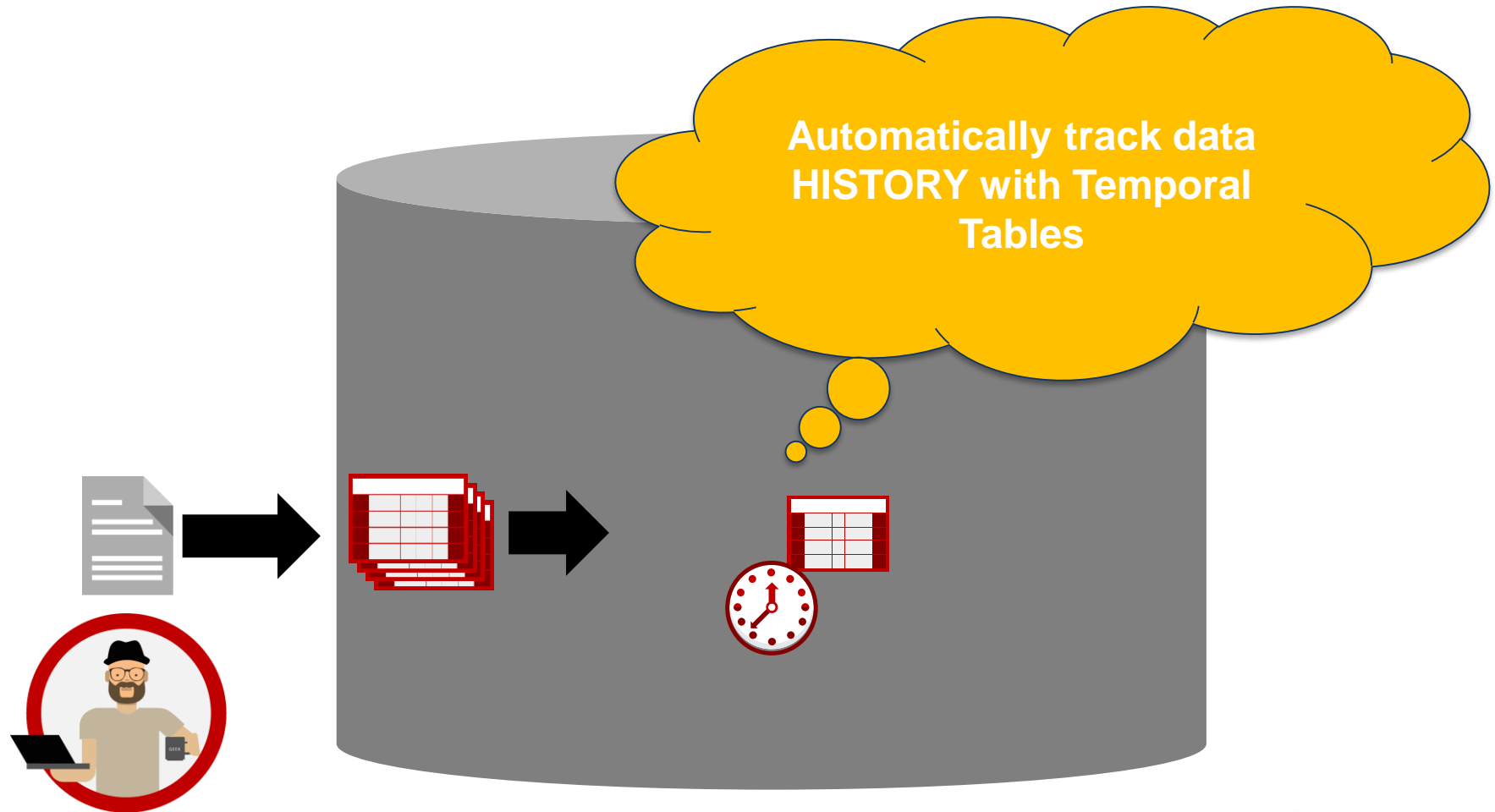
Temporal (2) script

November 26°, 2016



#sqlsatParma
#sqlsat566

Cronologia



November 26°, 2016

DEMO

Temporal indexing

(30)

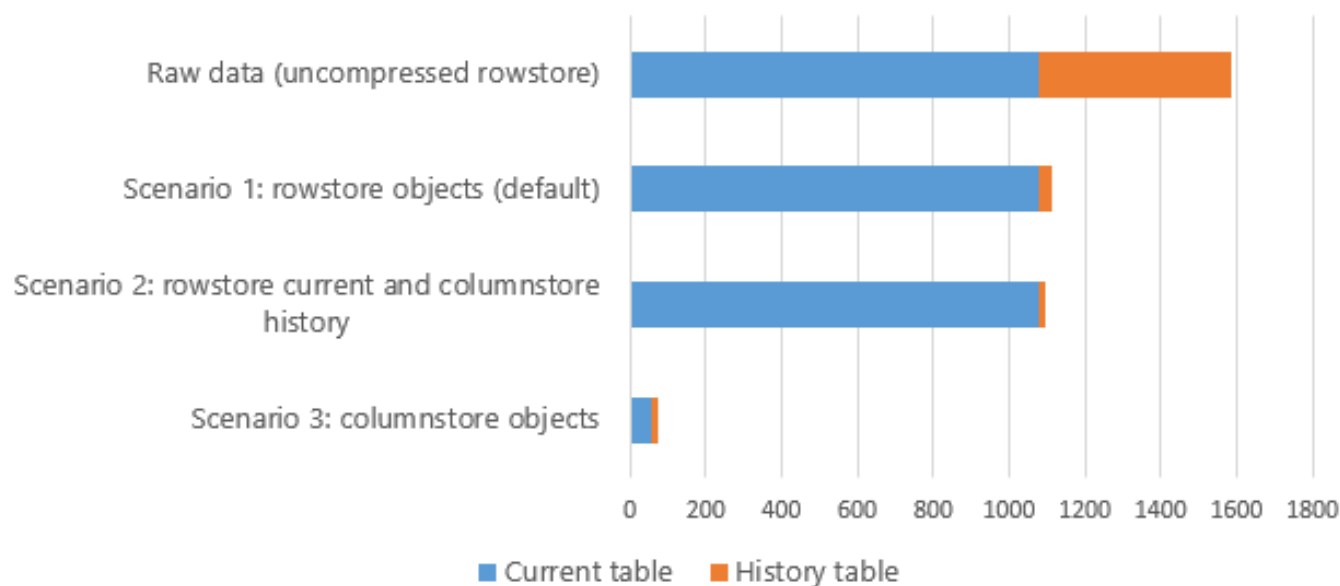
November 26°, 2016



Spazio su disco in MB

	Current table	History table	Total
Raw data (uncompressed rowstore)	1080	506	1586
Scenario 1: rowstore objects	1080	30	1110
Scenario 2: rowstore current and columnstore history	1080	17	1097
Scenario 3: columnstore objects	56	17	73

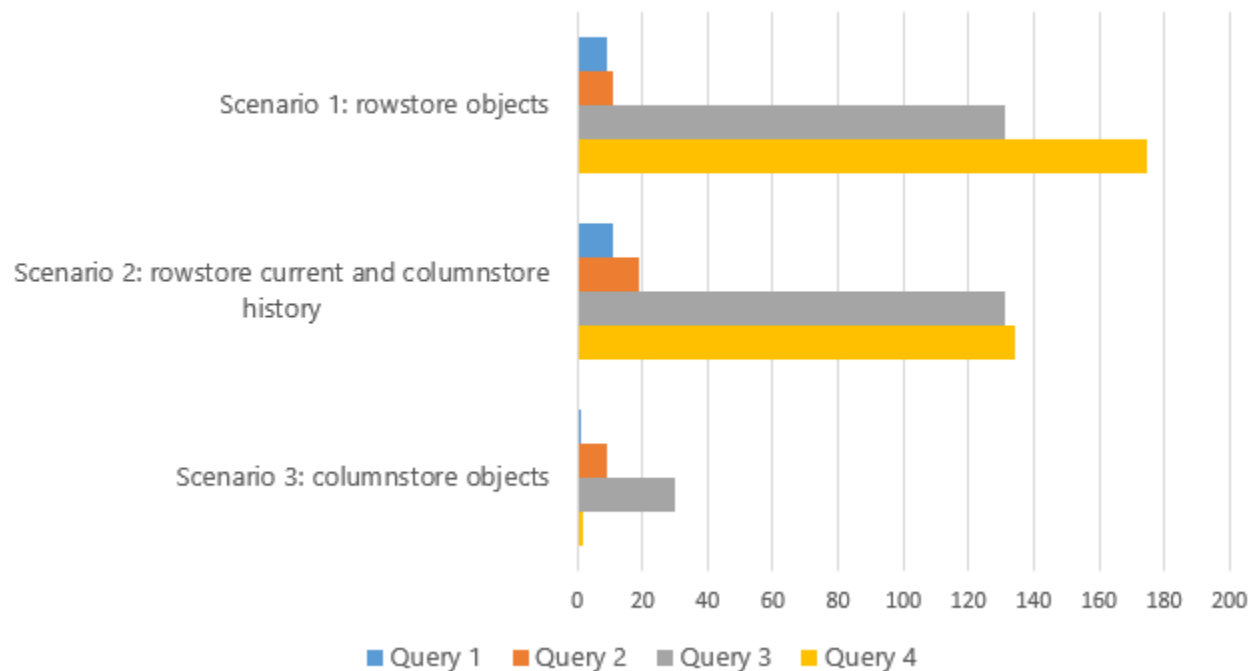
Chart 1. Disk space usage per scenario, MB



Select performance (per le 4 query)

	#1	#2	#3	#4
Scenario 1: rowstore objects	9	11	131	175
Scenario 2: rowstore current and columnstore history	11	19	131	134
Scenario 3: columnstore objects	1	9	30	2

Chart 3. SELECTs performance, milliseconds



^|^ Ma allora perchè non COLUMNSTORE di default?

Feature	RTM				SP1			
	Standard	Web	Express	Local DB	Standard	Web	Express	Local DB
Row-level security	Yes	No	No	No	Yes	Yes	Yes	Yes
Dynamic Data Masking	Yes	No	No	No	Yes	Yes	Yes	Yes
Change data capture*	No	No	No	No	Yes	Yes	No*	No*
Database snapshot	No	No	No	No	Yes	Yes	Yes	Yes
Columnstore	No	No	No	No	Yes	Yes	Yes	Yes
Partitioning	No	No	No	No	Yes	Yes	Yes	Yes
Compression	No	No	No	No	Yes	Yes	Yes	Yes
In Memory OLTP	No	No	No	No	Yes	Yes	Yes	No**
Always Encrypted	No	No	No	No	Yes	Yes	Yes	Yes
PolyBase	No	No	No	No	Yes	Yes	Yes	No
Fine grained auditing	No	No	No	No	Yes	Yes	Yes	Yes
Multiple filestream containers	No	No	No	No	Yes	Yes	Yes	Yes

SQL Server 2016 Service Pack 1 (SP1) released !!!

* Requires SQL Server agent which is not part of SQL Server Express Editions

** Requires creating filestream file groups which is not possible in Local DB due to insufficient permissions.

November 26°, 2016



**Keep, keep,
keep!**



Business owners,
end-users, ...

???
!#\$^*



DBA

**Cut, cut,
cut!**



Storage admins,
budget owners, ...

November 26°, 2016



Amministrare la crescita della Cronologia (dati storici)

Stretch Database

November 26°, 2016



Perché Stretch Database?

Database in continua crescita

- Tabelle enormi (milioni di righe, diversi GB/TB di dimensioni)
- Gli utenti vogliono conservare tutti i loro dati e usarli
- Vecchi dati (cold) acceduti raramente ma devono essere online
- Manutenzione onerosa (ad es. re-index)
- A rischio SLA (ad es. restore time)

Soluzioni possibili

- Aumentare lo storage disponibile
 - Non risolve problemi di manutenzione e SLA (a meno che non mettiate i dati su costosi dischi \$\$\$)
- Spostare i dati (di una tabella) verso un altro supporto, flat-file, backup, nastro
 - I dati però sono near-line oppure offline
- Cancellare vecchi dati
 - Ma sono veramente inutili?
 - Non applicabile in alcune realtà

Stretch SQL Server su Azure

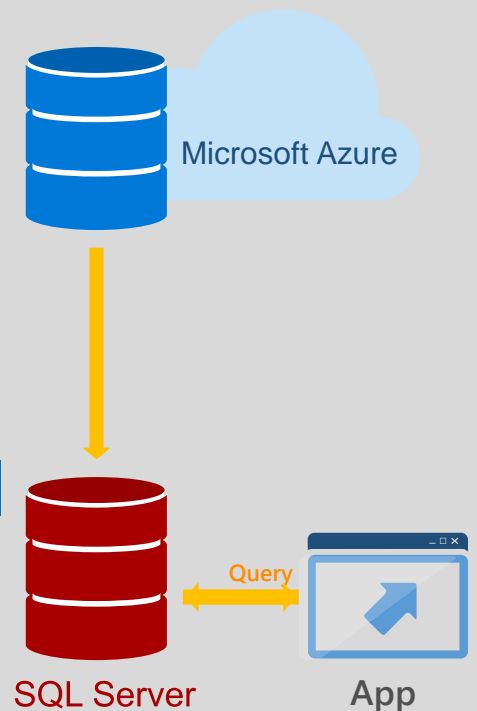
Stretch Database

Mike Wan ox7ff654ae6d 3/18/2005

Order history

Stretch to cloud

Name	SS	
Steve Karimi	cr	<input type="checkbox"/> Customer data
Mike Wan	ox	<input type="checkbox"/> Product data
Sapna Matuszyk	i2	<input checked="" type="checkbox"/> Order History
SQL Mojoe	nx	
Sapna Matuszyk	i2y36cg776rg	4/10/2005
SQL Mojoe	nx290pldo90l	4/27/2005
Blazej Zwilling	ypo85ba616rj	5/12/2005
Cheng Jeswani	bns51ra806fd	5/22/2005
Nikhat Lindell	mci12hh906fj	6/07/2005
Whitney Lang	utb76b916gi	6/18/2014
Lorenzo Olds	px61hi9306fj	7/1/2014
Sophie Cook	ol43bi506gd	7/12/2014
Aida Durham	tx83hal916fi	7/29/2014
Jaclyn Wade	nb95re926gi	8/10/2014



Potenzialità

Effettuare "stretch" delle tabelle che contengono "cold data" con sicurezza da scenari on-prem verso Azure grazie al "remote query processing"

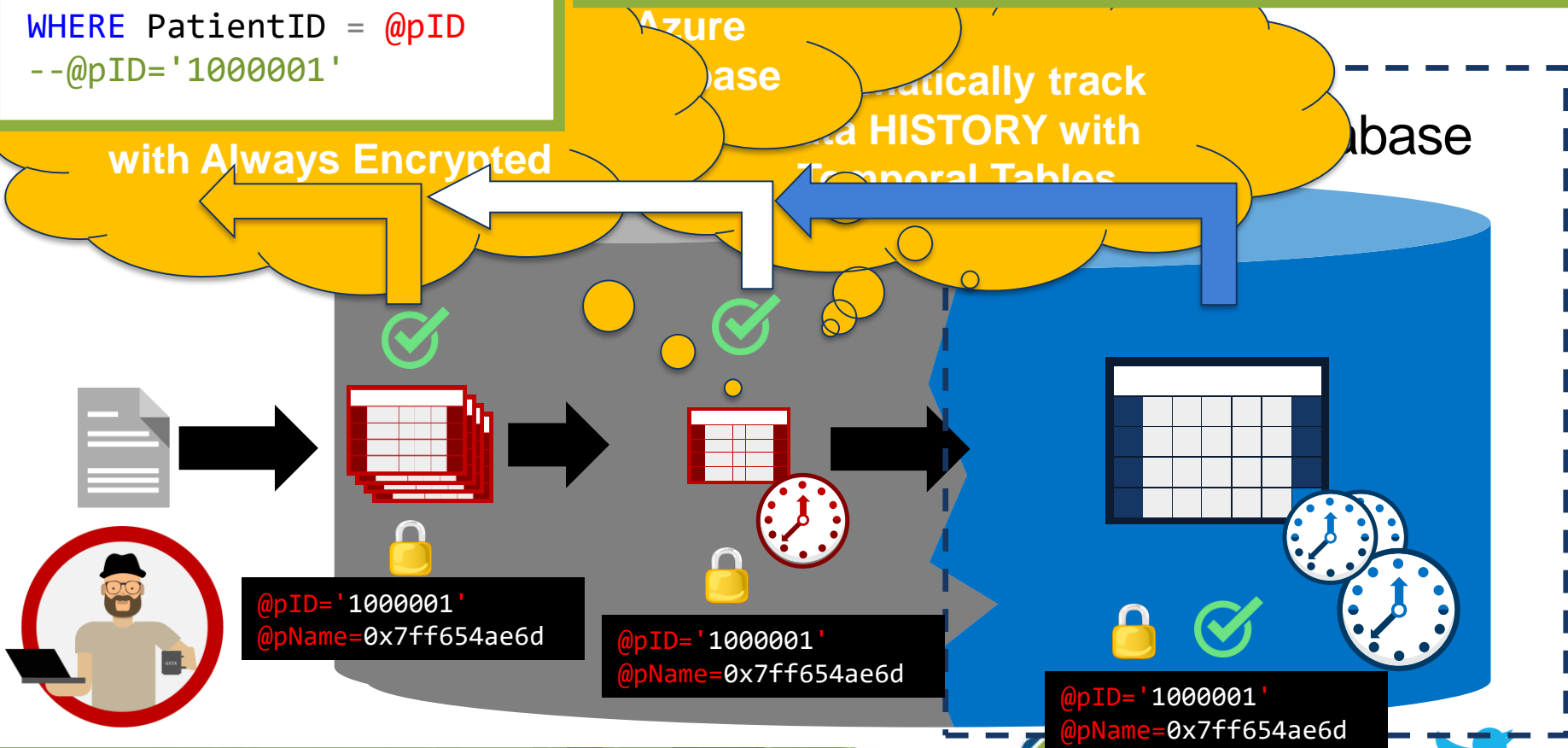
Vantaggi

- economico
- L'intera tabella resta online e rimane accessibile dalle applicazioni esistenti
- Nessuna modifica alle applicazioni esistenti
- Sicuro "by default" con la possibilità di aggiungere ulteriore sicurezza con Always Encrypted e Row Level Security

Securely keep data indefinitely

```
SELECT * FROM Patients
FOR SYSTEM_TIME
AS OF '2010.01.01'
WHERE PatientID = @pID
--@pID='1000001'
```

Keep data forever, offload to Azure, protect data everywhere, query across transparently!

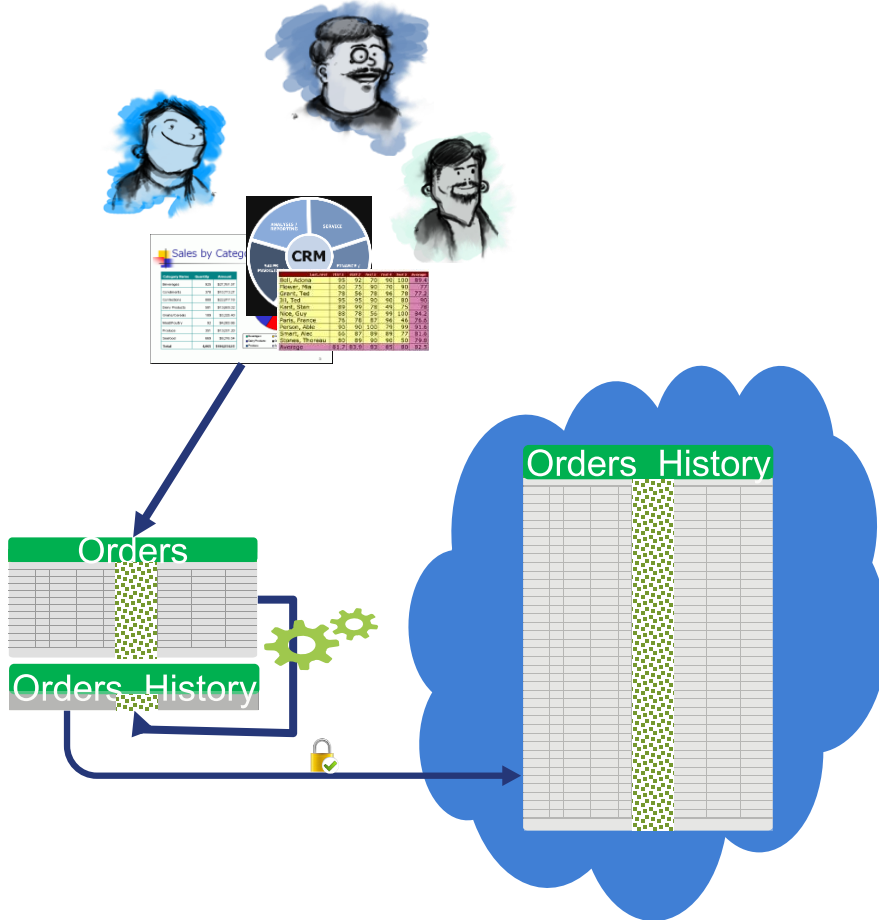


November 26°, 2016

PASS SQL
saturday

#sqlsatParma
#sqlsat566

Secure by default



- Data in motion always via secure channels (default TLS 1.2)
- Supports Always Encrypted
 - Encryption key remains on-premises with your application
- Row Level Security
- SQL Server and SQL Azure audit just works

DEMO

Stretch Database (40)

November 26°, 2016



Considerazioni

- Monitoring
 - Extended Events, DMVs e system monitor
 - Portale Azure per costi e gestione della sottoscrizione
- Backup/Restore
 - Attuali procedure di backup/restore sono valide
 - **sp_reauthorize** dopo aver effettuato un restore di un database «stretched» (riconciliazione)
 - Ricordate i backup delle encryption keys (non c'è modo di recuperarle)
- Prestazioni (query)
 - Dati correnti e «hot-data» generalmente non interessati
 - Dati temporali/cronologici locali non interessati
 - Latenza su interrogazioni di «cold-data» dipende dalle performance della rete
 - ^!^ RBAR Moolto male , Set-based molto meglio!

Amministrare la crescita della Cronologia (dati storici)

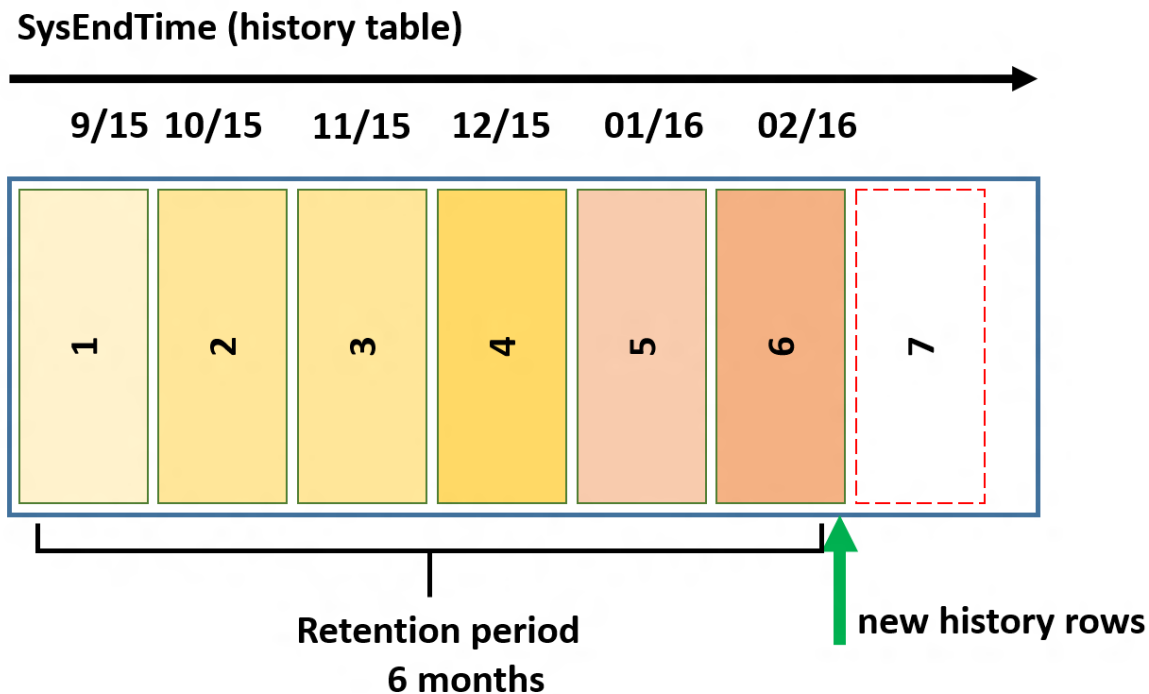
Partizionamento delle tabelle

1. Attività di configurazione del partizionamento
2. Attività di manutenzione ricorrenti della partizione

Amministrare la crescita della Cronologia (dati storici)

Partizionamento delle tabelle - esempio

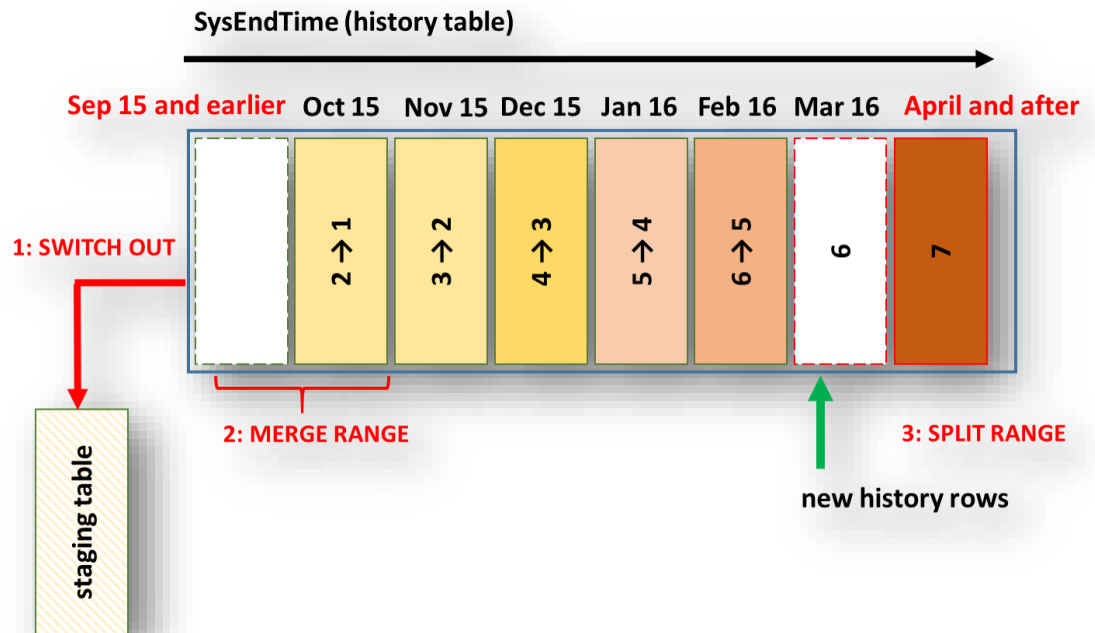
- Retention pari a 6 mesi (1 mese = 1 partizione)
- Inizio versioning Settembre 2015



Amministrare la crescita della Cronologia (dati storici)

Partizionamento delle tabelle

1. SWITCH OUT
2. MERGE RANGE
3. SPLIT RANGE



Amministrare la crescita della Cronologia (dati storici)

Custom delete script

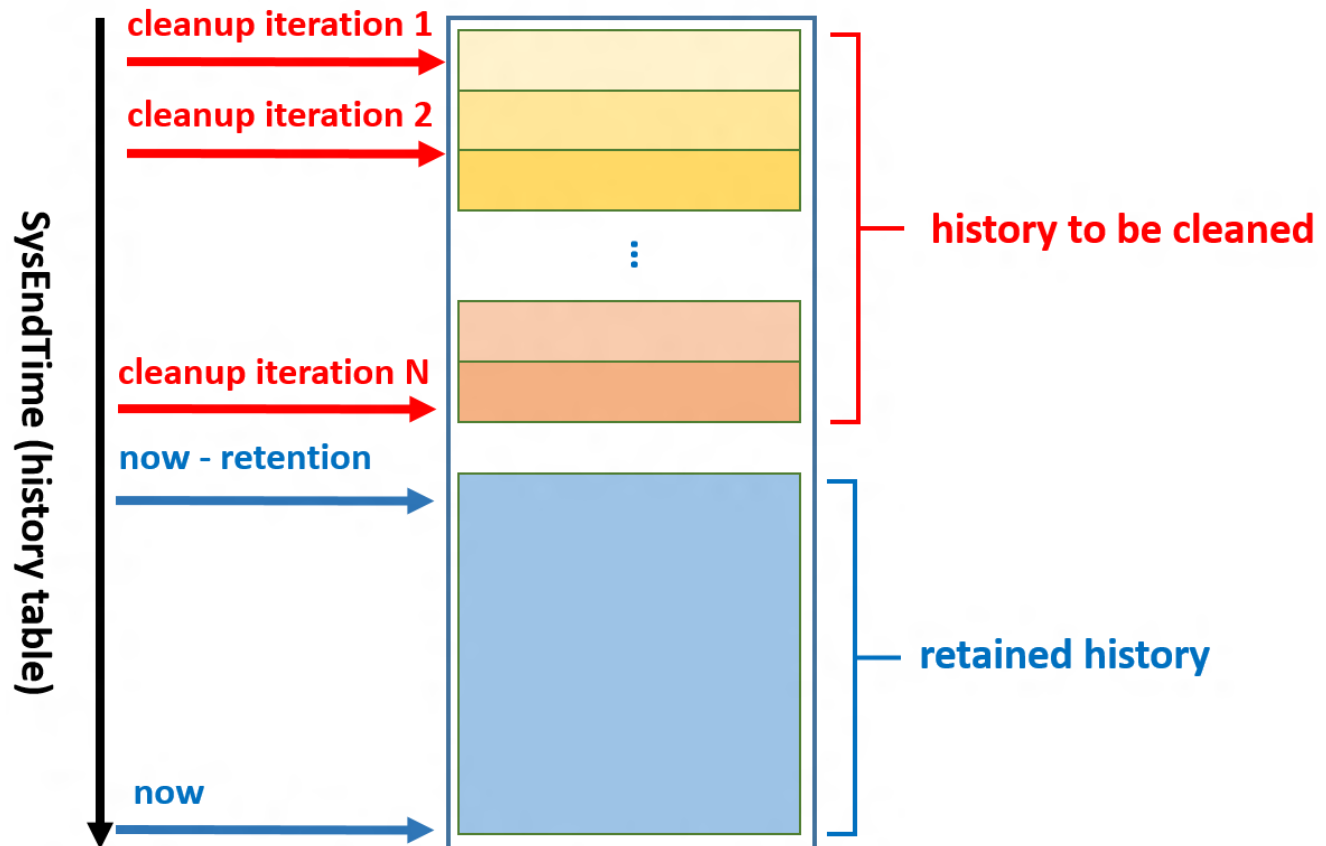
November 26°, 2016



Amministrare la crescita della Cronologia (dati storici)

Custom delete script

1. Turn off System-Versioning
2. Delete old rows
3. Turn on System-Versioning



Amministrare la crescita della Cronologia (dati storici)


Built-in retention policy

November 26°, 2016



Amministrare la crescita della Cronologia (dati storici)

Built-in retention policy



Only in
SQL Azure
Database

Amministrare la crescita della Cronologia (dati storici)

Built-in retention policy

- Definito a livello di database

```
SELECT  
  is_temporal_history_retention_enabled, name  
FROM sys.databases
```

- ON by default – modificabile dall'utente

```
ALTER DATABASE <myDB>  
SET TEMPORAL_HISTORY_RETENTION ON
```

- Auto OFF dopo un ripristino Point-in-time

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-temporal-tables-retention-policy>

Amministrare la crescita della Cronologia (dati storici)

Built-in retention policy

```
CREATE TABLE dbo.WebsiteUserInfo
(
    [UserID] int NOT NULL PRIMARY KEY CLUSTERED
    , [UserName] nvarchar(100) NOT NULL
    , [PagesVisited] int NOT NULL
    , [ValidFrom] datetime2 (0) GENERATED ALWAYS AS ROW START
    , [ValidTo] datetime2 (0) GENERATED ALWAYS AS ROW END
    , PERIOD FOR SYSTEM_TIME (ValidFrom, ValidTo)
)
WITH
(
    SYSTEM_VERSIONING = ON
    (
        HISTORY_TABLE = dbo.WebsiteUserInfoHistory,
        HISTORY_RETENTION_PERIOD = 6 MONTHS
    )
);
```


Amministrare la crescita della Cronologia (dati storici)

Built-in retention policy

Possibili valori per HISTORY_RETENTION_PERIOD:

- DAYS
- WEEKS
- MONTHS
- YEARS
- INFINITE (default se non specificato)

Impostando SYSTEM_VERSIONING = OFF viene perso il valore dell'impostazione HISTORY_RETENTION_PERIOD

```
ALTER TABLE dbo.WebsiteUserInfo  
SET (SYSTEM_VERSIONING = ON (HISTORY_RETENTION_PERIOD = 9 MONTHS));
```

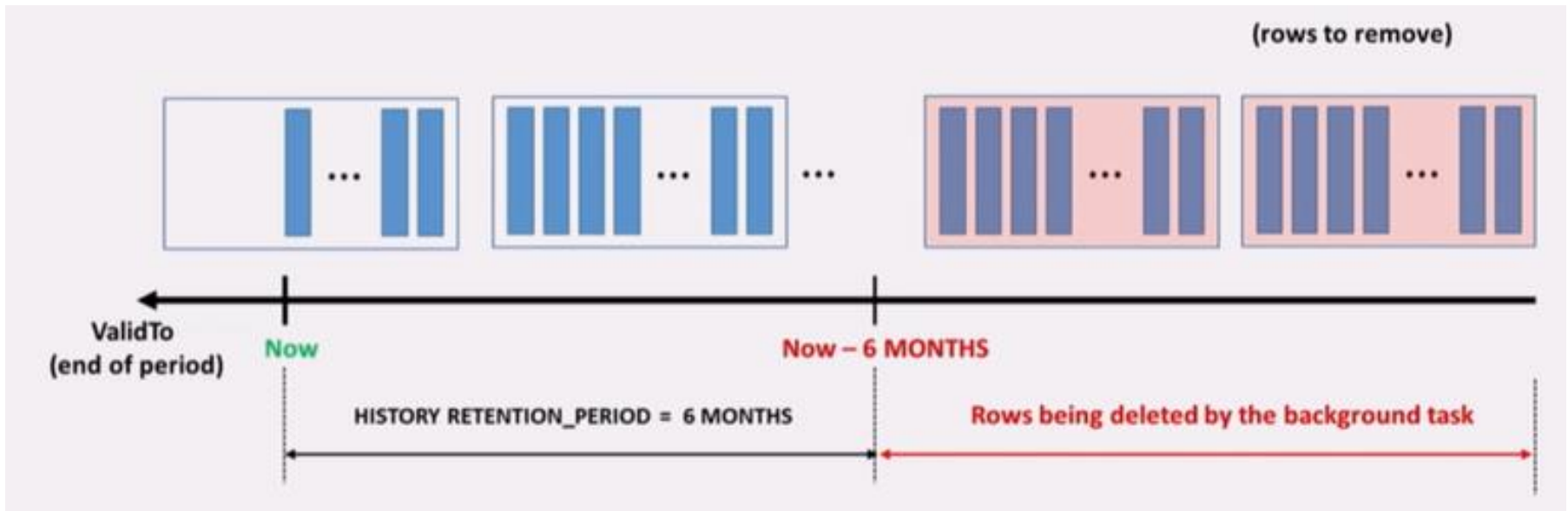
Amministrare la crescita della Cronologia (dati storici)

Built-in retention policy

- Solo le tabelle cronologiche con un indice cluster (B-Tree o Columnstore) può avere il parametro `HISTORY_RETENTION_PERIOD` impostato
- Processo background “netturbino”
 - Sugli indici B-Tree agisce su frammenti di 10K per minimizzare l’impatto sul database
 - Sugli indici COLUMNSTORE cancellazione per intero rowgroup (molto efficiente in ambienti in cui la cronologia è destinata a crescere in fretta)

Amministrare la crescita della Cronologia (dati storici)

Built-in retention policy



```
ALTER TABLE [dbo].[Inventory]
SET (SYSTEM_VERSIONING = ON (HISTORY_RETENTION_PERIOD = 6 MONTHS));
GO
```

DEMO

Built-in Retention (50)

November 26°, 2016



Amministrare la crescita della Cronologia (dati storici)

Built-in retention policy

Query

- Filtro automatico sulle query con predicati temporali se `is_temporal_history_retention_enabled = ON`

Restore point in time

`is_temporal_history_retention_enabled set to OFF`

Considerazioni e limitazioni delle tabelle temporali - I

- Una tabella temporale **deve** avere una chiave primaria e la tabella di cronologia **non può avere** una chiave primaria definita.
- Le colonne periodo SYSTEM_TIME usate per registrare i valori **SysStartTime** e **SysEndTime** devono essere datetime2 (elevata precisione)
- La tabella di cronologia deve essere creata nello stesso database della tabella corrente. L'esecuzione di query temporali su **Linked Server** non è supportata.
- **INSTEAD OF TRIGGER** – Né sulla tabella corrente che sulla cronologia
- **AFTER TRIGGER** – solo sulla tabella corrente
- L'istruzione **TRUNCATE TABLE** non è supportata quando l'opzione **SYSTEM_VERSIONING** è impostata su **ON**

<https://msdn.microsoft.com/it-it/library/mt604468.aspx>

Considerazioni e limitazioni delle tabelle temporali - II

- La modifica diretta dei dati in una tabella di cronologia non è consentita
- **ON DELETE CASCADE** e **ON UPDATE CASCADE** non sono consentiti nella tabella dei dati correnti. App-logic oppure trigger AFTER come workaround
- L'istruzione **TRUNCATE TABLE** non è supportata quando l'opzione **SYSTEM_VERSIONING** è impostata su **ON**
- **Replica limitata:**
 - **Always On:** completamente supportato
 - **Replica snapshot e transazionale:** supportata solo per un singolo server di pubblicazione senza attivazione di tabella temporale e per un sottoscrittore con attivazione di tabella temporale. In questo caso, il server di pubblicazione viene usato per un carico di lavoro OLTP, mentre il sottoscrittore viene usato per la ripartizione di report, inclusa l'esecuzione di query AS OF
 - **Replica di tipo merge:** non supportata per le tabelle temporali

Works great with other SQL features

In-Memory OLTP

Cost-effective auditing for intensive transactional workloads

Columnstore

Excellent compression and fast analysis for historical data

Stretch Database

Keep data history forever

Security features

Row-level security, Always Encrypted, Dynamic Data Masking

Tooling

First-class citizen in SSMS & SSDT

November 26°, 2016



Resource

- SQL Server 2016 Stretch Database
 - <http://go.microsoft.com/fwlink/?LinkId=627389>
- Azure SQL Stretch Database service
 - <https://azure.microsoft.com/en-us/services/sql-server-stretch-database/>
- Pricing
 - <http://go.microsoft.com/fwlink/?LinkID=724357>
- Q&A forums
 - Stackoverflow: <http://stackoverflow.com/questions/tagged/sql-server-2016>
 - MSDN: <http://go.microsoft.com/fwlink/?LinkId=627390>

Q&A

Questions?

November 26°, 2016





#sqlsatParma
#sqlsat566

THANKS!

November 26°, 2016

