

FUTURE DECODED

6-7 OTT '16 / MILANO

IN PARTNERSHIP WITH:



CommunityDays.it

www.futuredecoded.it



#FutureDecoded

DATA 01

SQL Server 2016 e Azure SQL Database

Gianluca Hotz

Presidente UGISS.org | Mentor SolidQ

@glhotz - ghotz@ugiss.org

www.futuredecoded.it

 #FutureDecoded



Speaker info



- Fondatore e Mentor SolidQ
 - 20 anni con SQL Server (dalla 4.21 nel 1996)
 - Modellazione basi di dati, dimensionamento e amministrazione, sviluppo, ottimizzazione
- Interessi
 - Modello relazionale, architettura DBMS, alta disponibilità e Disaster Recovery
- Microsoft MVP SQL Server dal 1998
- Fondatore e presidente UGISS
 - User Group Italiano SQL Server (PASS Chapter)

EDIZIONI SQL Server 2016

- Principali: Enterprise, Standard e Express
- Web
 - Disponibile solo per «hosting service providers»
- Developer
 - Rimane funzionalmente equivalente a Enterprise
 - **Gratuita!**
- Business Intelligence
 - Introdotta con versione 2012, non più disponibile con versione 2016
- Datacenter e Workgroup
 - Non più disponibili da versione 2008R2

PIANIFICAZIONE

- Supporto hardware e software
 - CPU: novità 4 sockets/24 cores versione Standard
 - RAM: sempre 128GB per versione Standard
 - Componenti principali solo per x64
 - Windows Server 2012 e 2012R2
 - Alcune versioni supportate su Windows 8, 8.1, 10
 - Linux coming soon... 😊
- Lista completa
 - [https://msdn.microsoft.com/en-us/library/ms143506\(v=sql.130\).aspx](https://msdn.microsoft.com/en-us/library/ms143506(v=sql.130).aspx)

INSTALLAZIONE SQL SERVER 2016

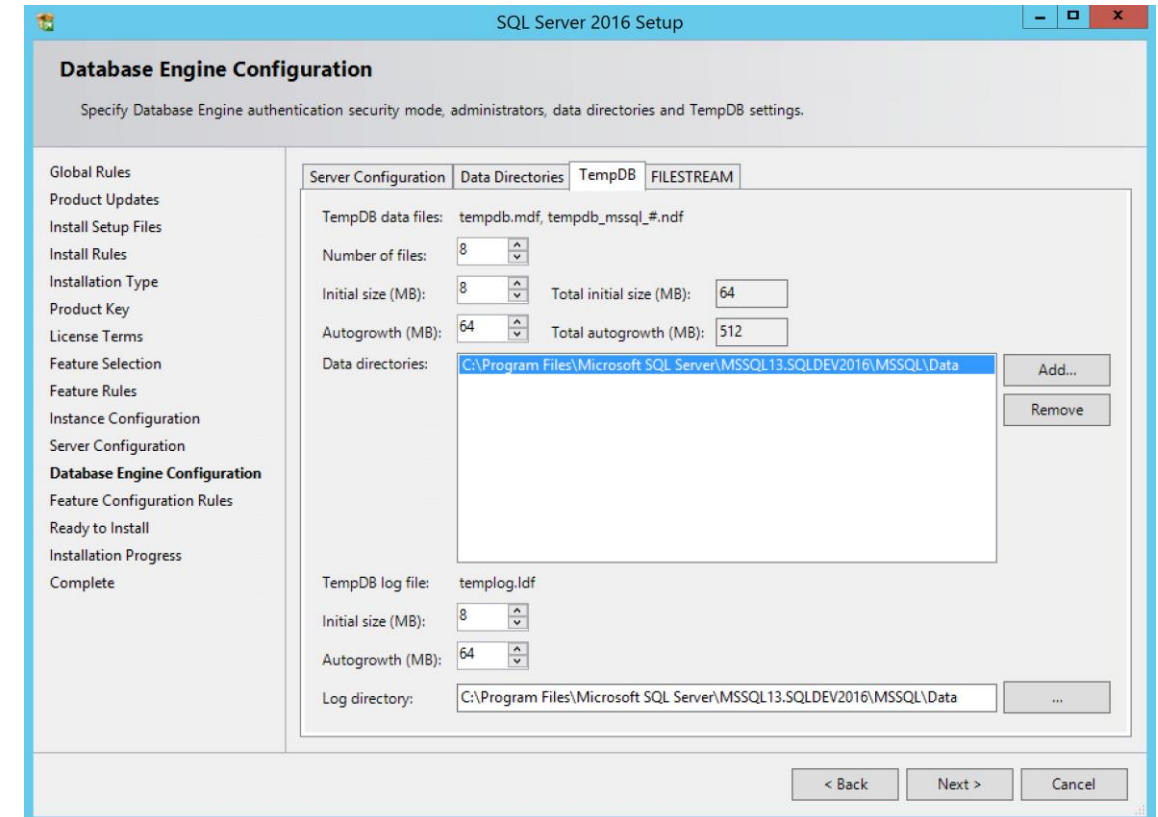
- Rimosso componente Management Studio
- Introdotti nuovi componenti
 - Integrazione con servizi R
 - PolyBase
 - R Server (standalone)
- Integrate alcune ottimizzazioni («It just runs faster»)
- «Installer Updates»
 - <https://support.microsoft.com/en-us/kb/3192738>

INSTALLAZIONE SQL SERVER 2016

- Critical Update
 - Problemi di stabilità, rilevato da Setup
 - Visual C Runtime 12.0.40649.5 o successiva
 - <https://support.microsoft.com/en-us/kb/3164398>
- Oppure «hotfix» recente: RTM Cumulative Update 2
 - <https://support.microsoft.com/en-us/kb/3182270>
 - <https://blogs.sqlsentry.com/team-posts/latest-builds-sql-server-2016>

Configurazione Tempdb

- Numero core fino a 8 file
 - Mitigare contesa PFS
 - Marginalmente anche GAM/SGAM
- «Auto growth» 64MB
 - Allineato indirizzamento PFS

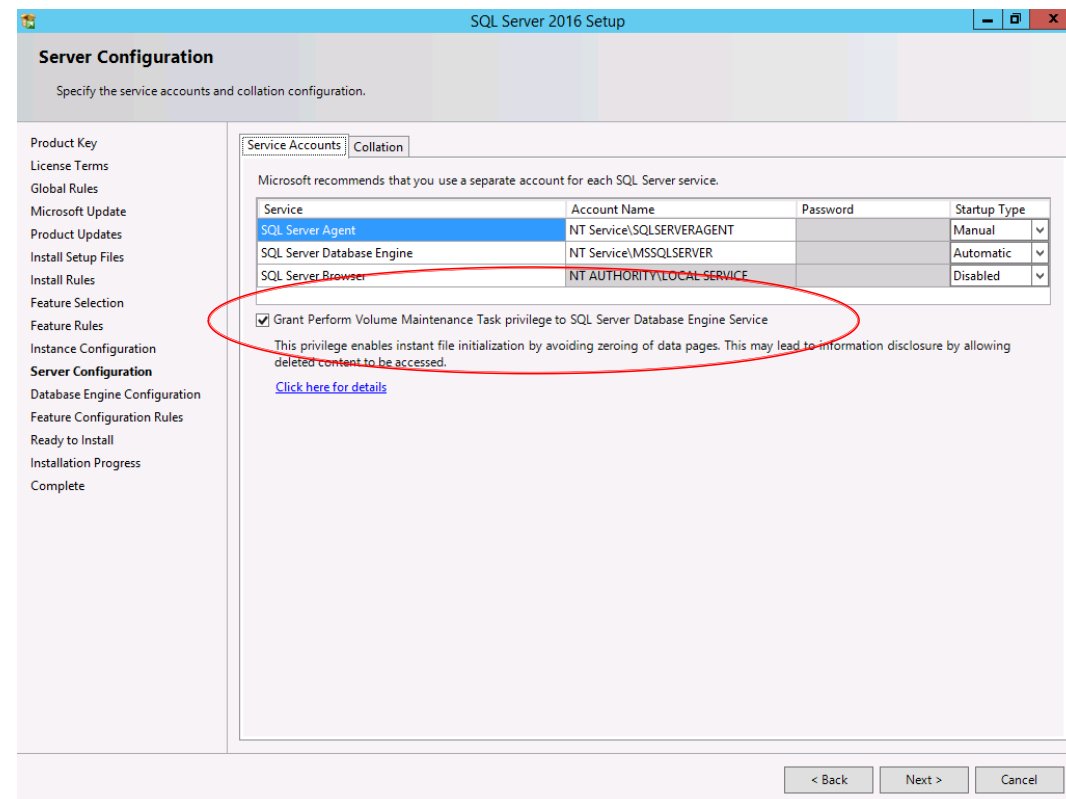


Trace flags Tempdb

- Best practice attivazione trace flag
 - T1117: crescita omogenea file database
 - T1118: utilizzo solo di «uniform extent»
 - Impatto solitamente per tempdb (ma attivi per tutti)
- Non più necessario a partire da 2016
 - Sempre attivi per tempdb
 - Estensione ALTER DATABASE per altri database

«Instant File Initialization»

- Non inizializza i file riempiendoli di zeri
 - Creazione
 - Crescita
 - Solo file dati (no transaction log!)
- Policy locale «Performance Volume Maintenance Tasks»
- In SQL Server 2016 integrato con Setup
- Problemi di sicurezza? Marginali...



DATABASE SCOPED CONFIGURATION

- Pulizia cache piani di esecuzione per database
- Parametri di configurazione a livello di database
 - Impostazione MAXDOP
 - «Parameter Sniffing»
 - «Cardinality Estimator»
 - indipendente da «Database Compatibility Level»
- Distinzione tra database primari/secondari in AlwaysOn AG

«Automatic Soft NUMA»

- NUMA: partizionamento CPU per accesso locale RAM
- SQL Server 2005 sviluppato con NUMA «built-in»
- Design originale fino a 8 CPU logiche per nodo...
- CPU moderne 24+ core... problemi simili a sistemi SMP
- In 2016 suddivisione in più nodi quando > 8 processori fisici
 - Default attivo (ALTER SERVER CONFIGURATION)
 - Tracciato in ERRORLOG
 - 1 IOCP Worker per nodo

«Dynamic Memory Objects»

- CMEMTHREAD wait
 - Sincronizzazione tra thread per allocazione memoria generica (no BP)
 - Collo di bottiglia con molti core
 - Situazioni problematiche gestite tramite hotfix
- In SQL Server 2016: partizionamento dinamico
 - Per nodo NUMA e poi per CPU (solo CPU se singolo nodo)
 - Automatico, in base a contesa
 - Trace Flag 8048 non più necessario

«Parallel INSERT»

- Già disponibile per SELECT/INTO in SQL Server 2014
 - Solo per Heaps e Clustered Columnstore
 - TABLOCK hint (non richiesto per tabelle temporanee #)
 - Altre restrizioni (es. UDF nella SELECT, trigger on viste indicizzate, ...)
- Operazione «minimally logged»
- Può causare contesa PFS in tempdb...

«Parallel REDO»

- Fasi «recovery» di un database: analisi log, REDO e UNDO
 - Fase di REDO storicamente «I/O bound»
 - Nuovo storage più veloce, serve utilizzare più CPU
- Lista pagine da riscrivere distribuite a più task
- Riduce tempi per riportare il database in linea
 - singola istanza dopo ripartenza
 - Availability Groups

DBCC CHECK*

- Molte ottimizzazioni in SQL Server 2008
- In SQL Server 2016 target principale PHYSICAL_ONLY
 - Ridotto contesa latch MULTI_OBJECT_SCANNER
 - Approccio algoritmico «no-lock» tipo «In-Memory OLTP»
 - Miglioramenti da 2x a 7x
- ... WITH MAXDOP = X

QUERY HINT

- DBCC CHECK* ... WITH MAXDOP
- MIN_GRANT_PERCENT/MAX_GRANT_PERCENT
- NO_PERFORMANCE_SPOOL

«Log I/O at the speed of RAM»

- Log collo di bottiglia principale anche per «In-memory OLTP»
 - Latenza passata da ms a μ s con SSD PCI NVMe
 - ns con NVDIMM in Windows Server 2016 supporta «block storage»
- Funzionalità ancora in «preview»
 - Formattare volume con /dax
 - Attivare –T9921
 - Creare secondo log su nuovo volume
 - SQL Server 2016 riconosce lo scenario
 - Copia della sola coda del log simile a «memcpy»

OPERAZIONI MASSIVE

- TRUNCATE TABLE ... WITH PARTITIONS
- ALTER TABLE ... ALTER COLUMN
 - Più operazioni non bloccanti

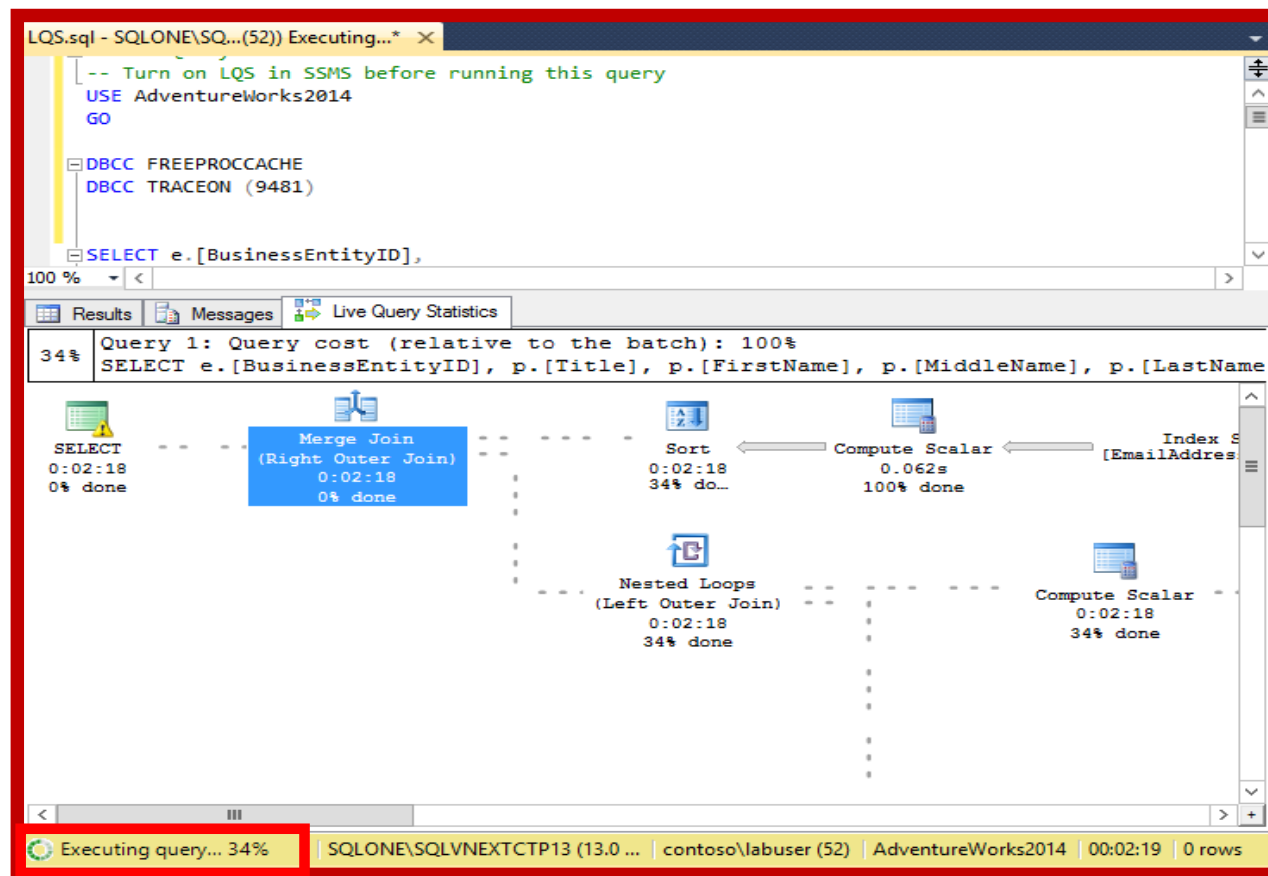
«Spatial Flies»

- «Spatial Data Types» in T-SQL
 - geography and geometry
 - implementati tramite SQL CLR/Assembly
 - chiamate da codice «managed» a codice «unmanaged»
 - collo di bottiglia con volumi medio/grandi
- In SQL Server 2016
 - Chiamate direttamente a codice «unmanaged»

LIVE QUERY STATISTICS IN SSMS

- Indicatore di progresso globale (stima in %)
- Piano di esecuzione «live» con progresso operatori!
- Attivazione anche da «Activity Monitor»
- Funziona anche con istanze SQL Server 2014
 - Utilizza **sys.dm_exec_query_profiles**
 - Richiede comunque SSMS 2016

LIVE QUERY STATISTICS IN SSMS



A nighttime cityscape featuring several tall skyscrapers with illuminated windows. In the foreground, a multi-lane highway shows long-exposure light trails from cars, with red trails from taillights and white/yellow trails from headlights. A pedestrian bridge with railings and some greenery is visible on the right side of the highway. The sky is dark blue.

DEMO

Live Query Statistics

www.futuredecoded.it

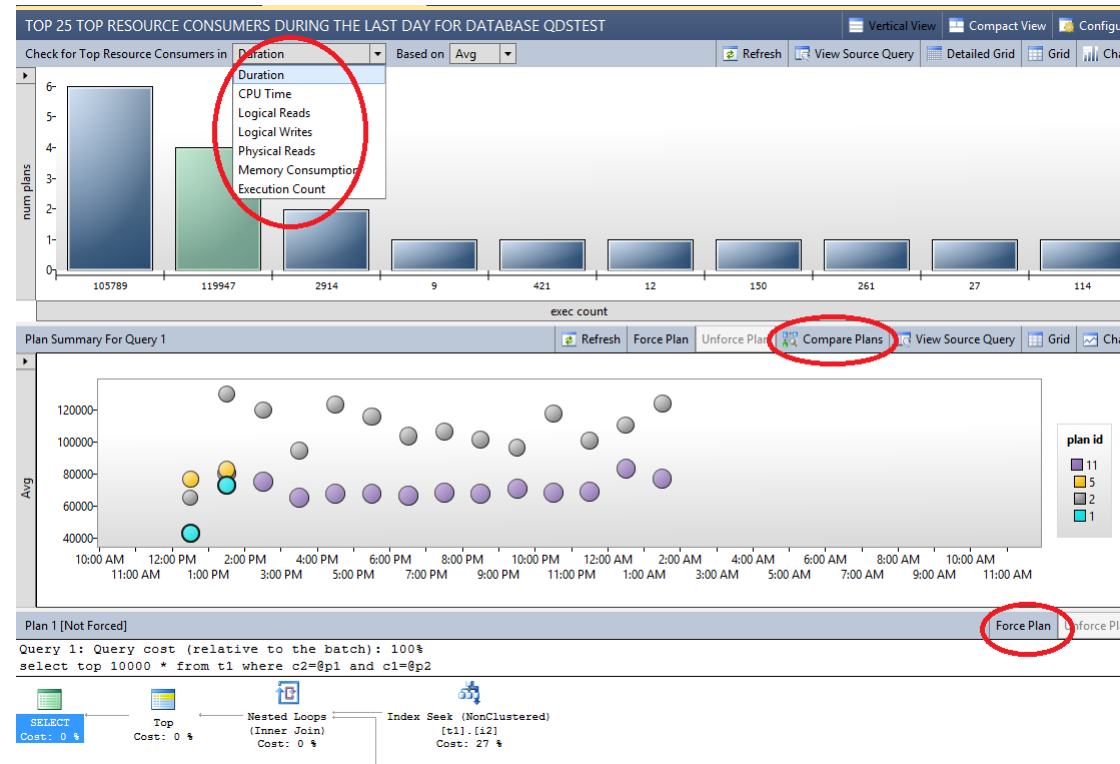


#FutureDecoded

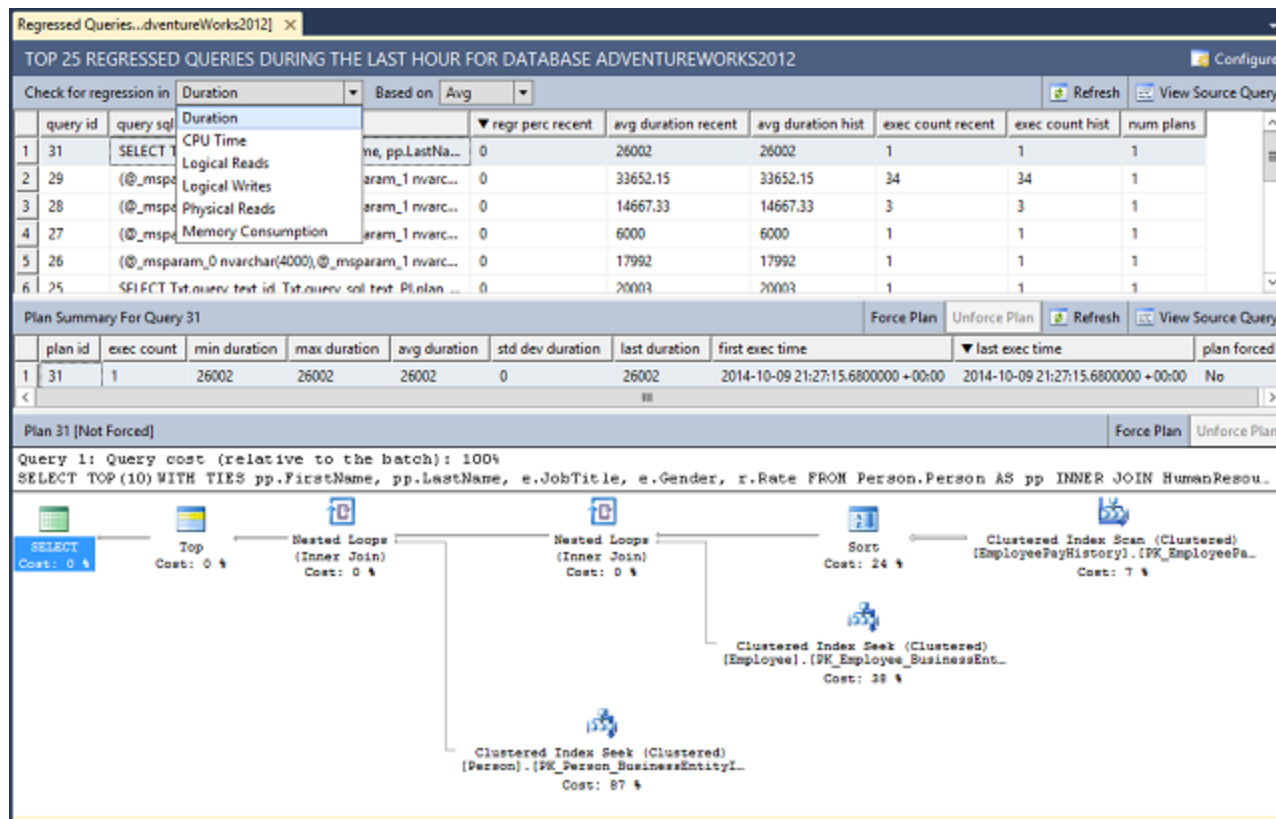
QUERY STORE

- Registra piani di esecuzione e metriche prestazioni
 - Per database
 - In memoria consolidando su disco in modalità asincrona
 - Integrato con SSMS e disponibile via DMV
- Riduce drasticamente complessità «troubleshooting»
 - Identificare «Top Query» per tempo/memoria/CPU/IO
 - Analizzare utilizzo risorse per un determinato database
 - Audit storia piani di esecuzione di una determinata query
 - Identificare regressione piani di esecuzione
 - Forzatura piano specifico da storico

QUERY STORE TOP CONSUMER



TOP «REGRESSED QUERIES»



A nighttime cityscape featuring several tall skyscrapers with illuminated windows. In the foreground, a multi-lane highway shows long-exposure light trails from vehicles, with red trails from taillights and white/yellow trails from headlights. A pedestrian bridge with railings and some greenery is visible on the right side of the highway. The sky is dark blue.

DEMO

Query Store

www.futuredecoded.it



#FutureDecoded

BACKUP IN SQL SERVER 2016

- Backups «Microsoft Azure Blob Storage»
 - Supporto per «block blobs»
 - Supporto per «striping» fino a un massimo di 12,8TB
- «Managed Backup»
 - Supporto per database di sistema
 - Supporto per database con «simple recovery model»
 - Supporto schedulazione personalizzata
 - Supporto per «block blobs» in Azure
- Backup «File-Snapshot»
 - Disponibile solo per database con file dati/log in Azure Storage
 - Backup FULL e T-LOG
 - «Snapshot» direttamente in Azure Storage
 - Utilizza una frazione dello spazio e non necessità di I/O tra server e Storage

ALWAYSON IN SQL SERVER 2016

- Group Managed Service Accounts (gMSA)
 - Rotazione password automatica
 - Gestiti direttamente da AD
 - Supporto anche per SQL Server 2014

ALWAYSON AG IN SQL SERVER 2016

- Fino a 3 repliche sincrone con «failover» automatico
- «Load balancing» repliche in sola lettura
- Migliori prestazioni trasporto log record
- MSDTC supportato con Windows Server 2016
- Database crittografati supportati (es. SSISDB)
- «Cross-domain» e «Domain Independent» Availability Groups
- «Seeding»: inizializzazione senza bisogno di backup/restore

ALWAYSON BASIC AVAILABILITY GROUPS

- Disponibile in edizione Standard
 - Soluzione per «Database Mirroring» deprecato (quasi...)
- Limitato a 2 sole repliche
 - Sia sincrone che asincrone
 - Replica può essere in Azure
- 1 solo database per AG
- Replica secondaria non accessibile (lettura/backup/...)

«AlwaysOn AG Turbocharged»

- Prestazioni non sufficienti con repliche sincrone
 - Non in tutti gli scenari...
 - Necessario migliorare scalabilità (con più risorse disponibili/più veloci)
 - Obiettivo: 95% delle prestazioni con una singola replica sincrona
- Svariate ottimizzazioni in SQL Server 2016
 - Es. riduzione «worker thread», maggiore parallelismo, riduzione contese
- Risultati
 - 95% con una replica sincrona, 90% con due repliche

ALWAYSON CAMBIO NEL «LICENSING»

- Software Assurance necessaria per i server passivi!
 - “Beginning with SQL Server 2014, each active server licensed with SA coverage allows the installation of a single passive server used for fail-over support.”
 - “The active server license (s) must be covered with SA, and allow for one passive secondary SQL Server, with up to the same amount of compute as the licensed active server, only.”
- Impatto su tutto: log shipping, mirroring, availability groups, failover clustering
- SQL Server 2014 Licensing Datasheet
 - http://download.microsoft.com/download/6/6/F/66FF3259-1466-4BBA-A505-2E3DA5B2B1FA/SQL_Server_2014_Licensing_Datasheet.pdf
- SQL Server 2016 Licensing Datasheet
 - http://download.microsoft.com/download/F/D/5/FD5E5C28-6973-4273-8737-D69AA3BEA243/SQL_Server_2016_Licensing_Datasheet_EN_US.pdf

IN-MEMORY OLTP IN SQL SERVER 2016

- Tabelle «memory-optimized»
 - Supporto per modifiche a «Schema» (anche indici)
 - Supporto vincoli UNIQUE, CHECK, FK
 - Supporto «Trigger» (solo AFTER e compilati)
 - Supporto per «LOB» e «Large Row»
 - Campionamento statistiche (prima solo «FULLSCAN»)
 - Statistiche aggiornate automaticamente
 - Supporto fino a 2TB di dati per database
 - Supporto «Transparent Data Encryption»

IN-MEMORY OLTP IN SQL SERVER 2016

- Moduli T-SQL compilati
 - Supporto per «UDF» scalari, «Inline TVF» e «Trigger»
 - Esteso supporto costrutti T-SQL
 - OUTPUT, UNION, UNION ALL, SELECT DISTINCT, OUTER JOIN e «subquery» in SELECT
 - Esteso supporto a funzioni native (sicurezza/matematica)
 - «EXECUTE AS CALLER» non più necessario
 - «LOB» utilizzabili come variabili/parametri

A nighttime cityscape featuring several tall skyscrapers with illuminated windows. In the foreground, a multi-lane highway shows long-exposure light trails from vehicles, with red trails from taillights and white/yellow trails from headlights. A pedestrian bridge with railings and some greenery is visible on the right side of the highway. The sky is dark blue.

DEMO

In-Memory OLTP

www.futuredecoded.it



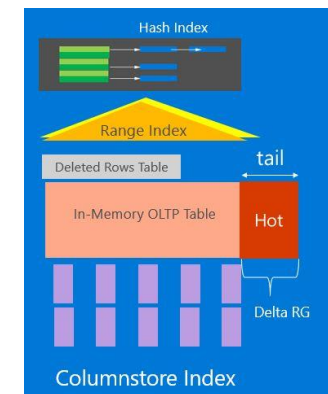
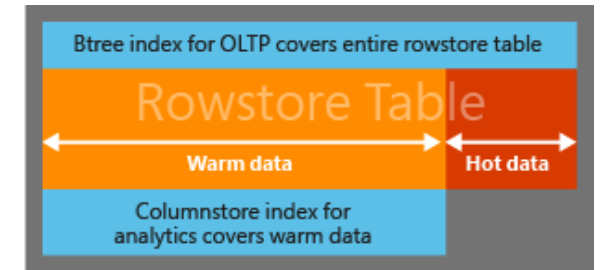
#FutureDecoded

IN-MEMORY DW IN SQL SERVER 2016

- Indici «Columnstore non-clustered» aggiornabili
- Indici «Columnstore non-clustered» filtrati
- Indici «B-Tree non-clustered» con «Columnstore clustered»
- Indici «Columnstore» con tabelle «memory-optimized»
- Supporto livelli di isolamento RCSI e Snapshot
- Modalità «batch» estesa a query «single-threaded»
- Possibilità di specificare indice in fase di creazione tabella
- Indici «Columnstore» aggiornabili supportati replica secondaria AG

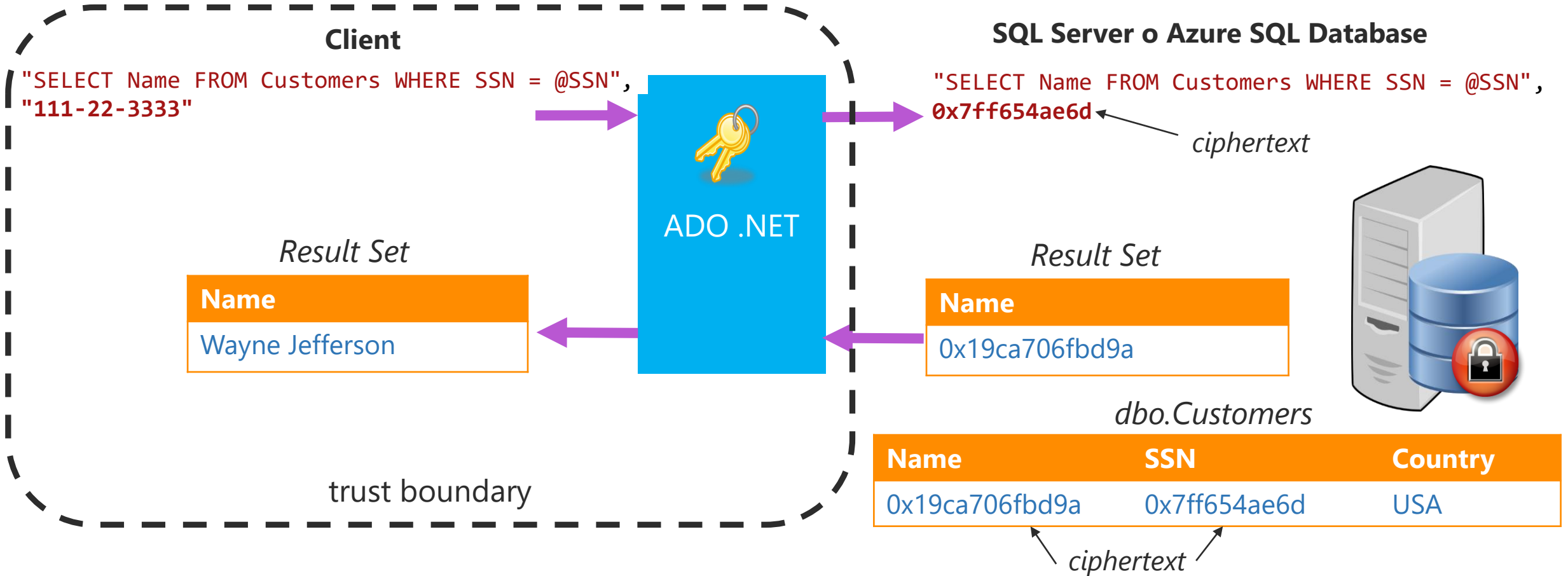
REAL-TIME OPERATIONAL ANALYTICS

- Carico di lavoro OLTP e OLAP con singolo database
 - Al posto di creare procedure ETL e DWH separato
 - Combinazione di tecnologie
- Scenario 1
 - Tabella «disk-based» con indice «Columnstore non-clustered» filtrato solo su dati storici
- Scenario 2
 - Tabella «memory-optimized» con indice «Columnstore clustered»



«ALWAYS ENCRYPTED»

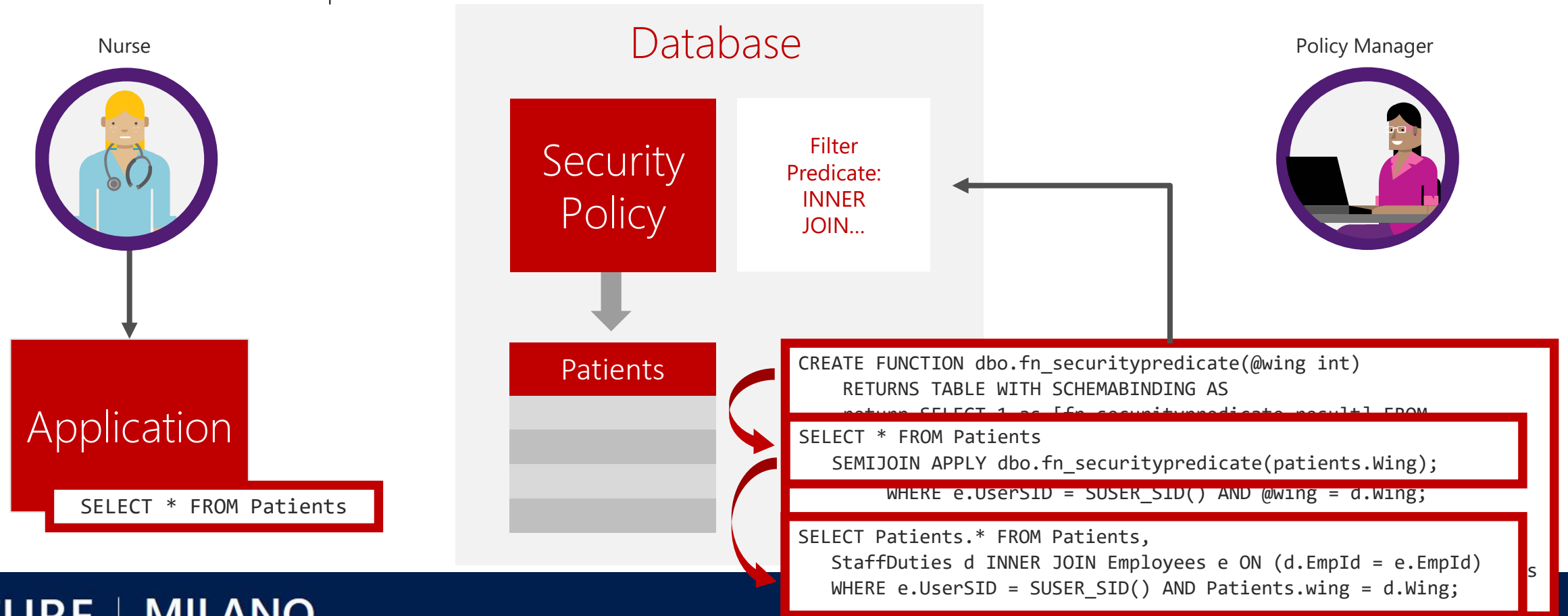
*Dati sensibili cifrati e corrispondenti chiavi crittografiche *mai* disponibili "in chiaro" a SQL Server*



«ROW LEVEL SECURITY»

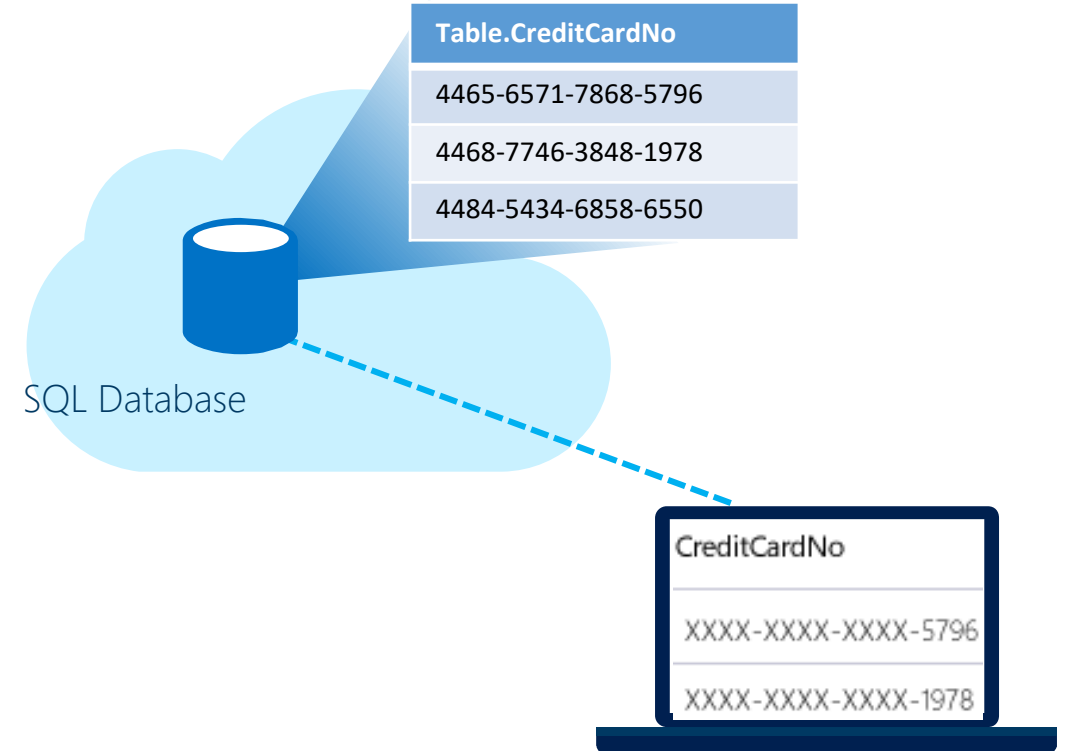
Die

Security Policy (Policy) è un tipo di politica di sicurezza che definisce i permessi di accesso ai dati. In T-SQL, vincolando il predicato alla tabella dei pazienti.



«DYNAMIC DATA MASKING»

- Mascheramento
 - Real-time
 - Parziale
- Basato su Policy
 - A livello di tabella/colonna
 - Permessi per vedere dati in chiaro



Azure Security Center

- Servizio in Azure per governare sicurezza risorse in Azure
- Raccomandazioni «actionable» anche per Azure SQL Database
 - «Encryption» (es. TDE)
 - «Audit & Monitor» (es. Server Auditing, Database Auditing)
 - «Dynamic Data Masking» (es. Data Masking per dati sensibili)
- <https://azure.microsoft.com/en-us/services/security-center>

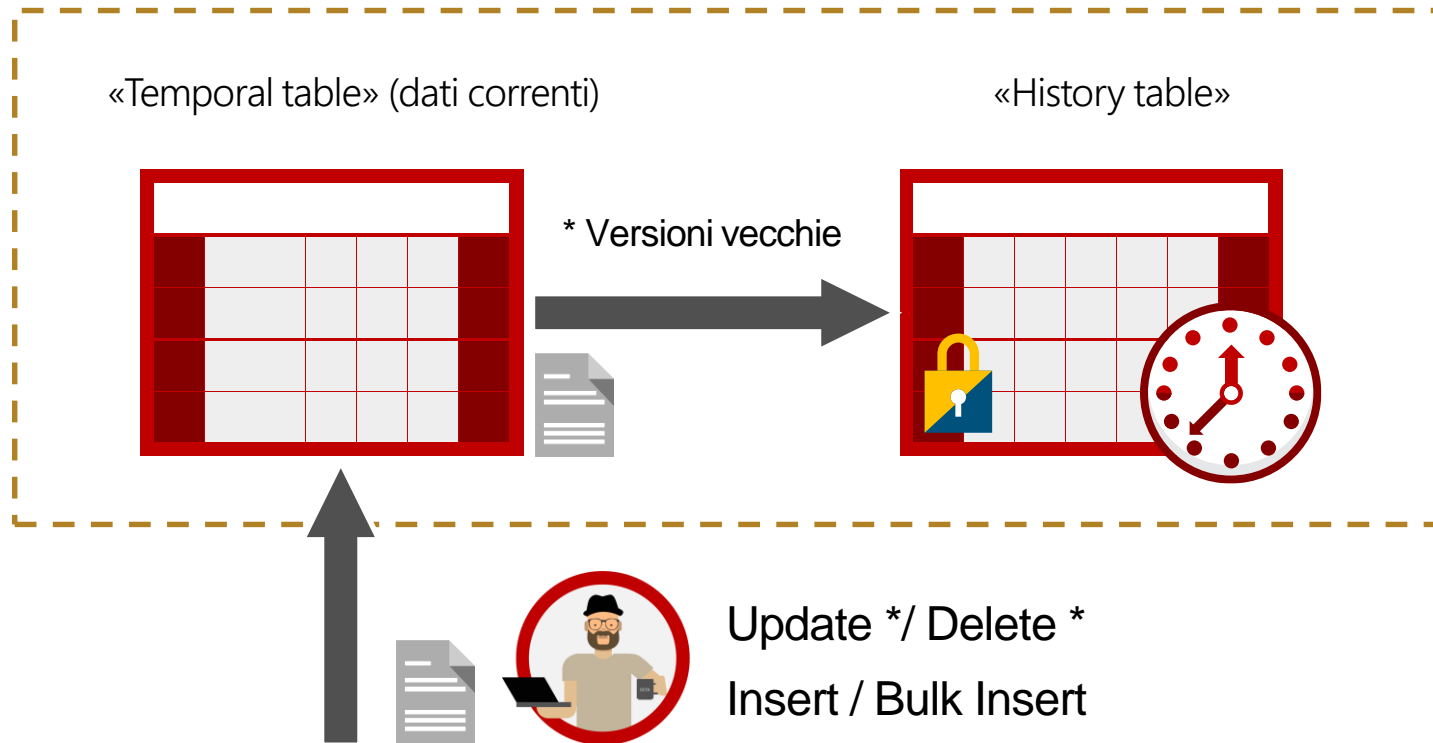
Azure SQL Database «Threat Detection»

- Semplice opzione da attivare per database
- Rileva potenziali minacce
 - Potenziali vulnerabilità
 - Attacchi di tipo «SQL Injection»
 - Accessi anomali (es. da luoghi anomali)
- Allarmi
 - In tempo reale
 - Raccomandazioni su cosa investigare e come mitigare/rimediare

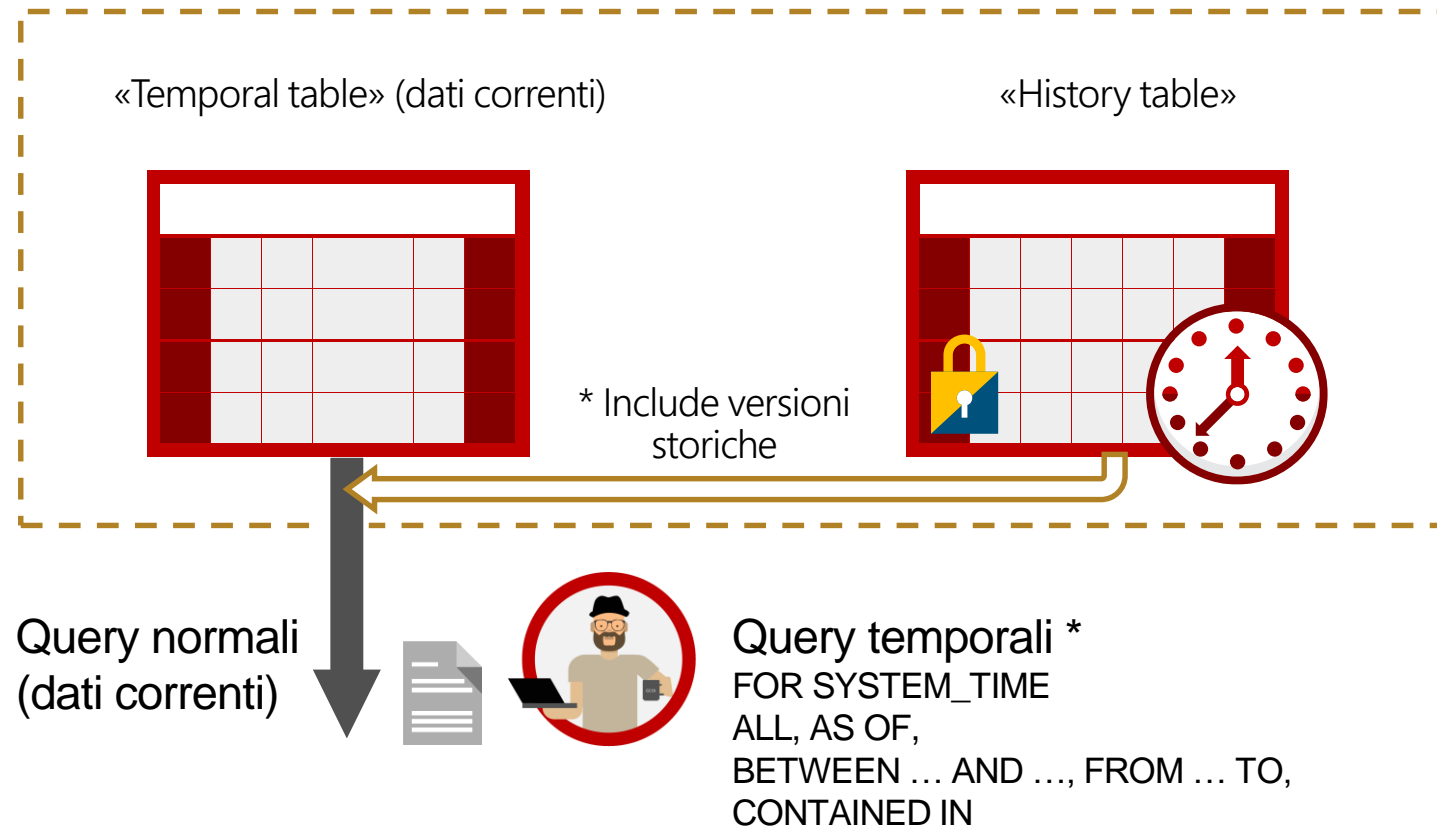
DATABASE TEMPORALI

- «System-versioned Temporal Table»
- Permette di gestire valori correnti e storici
- Periodo validità
 - definito da due colonne di tipo datetime2
 - aggiornato dal sistema a fronte di modifiche
- Implementata tramite due tabelle fisiche
 - «Current Table» per dati correnti
 - «History Table» per dati storici (modificati)
 - Schema speculare a «Current Table»

MODIFICHE DATABASE TEMPORALI



INTERROGAZIONI DATABASE TEMPORALI



PREDICATI DATABASE 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)

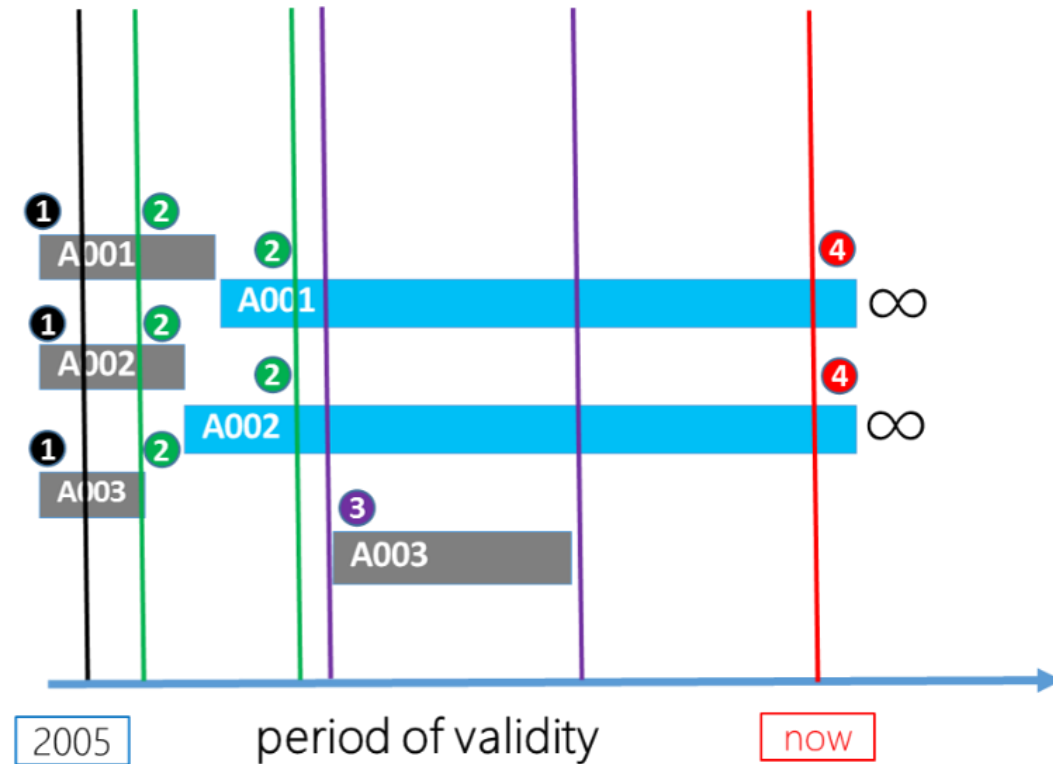
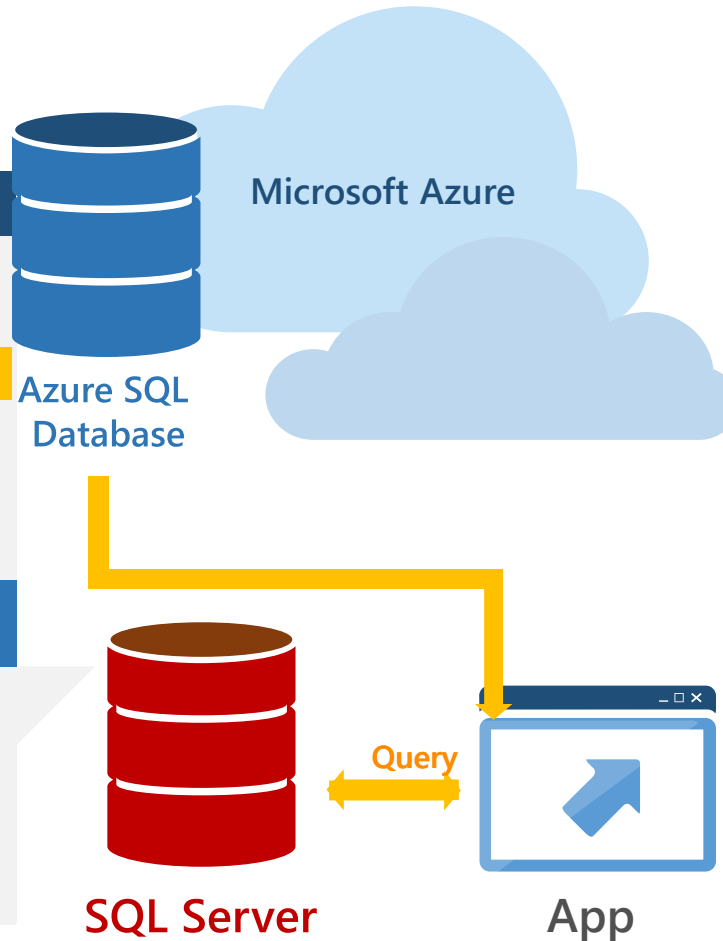


Image from <https://msdn.microsoft.com/library/mt631669.aspx>

STRETCH DATABASE

Always Encrypted

Order history		
Name	Date	Item
0x21ba906fdb52	1ba906fd	2ba906f
0x19ca706fbd9a	5re316rl	1da813t
1x59cm676rfd8b	1re306fd	3ha706f
2y36cg776rgd5b	3bg606fl	1ba906i
1t64ce87r6pd7d	5ba616rj	2ra933f
0y16cj676rfd3e	1ra806fd	3ra806t
Order history		
Stretch to cloud		
Name		
0x21ba906fdb52	<input type="checkbox"/> Customer data	
0x19ca706fbd9a	<input type="checkbox"/> Product data	
1x59cm676rfd8b	<input checked="" type="checkbox"/> Order History	
2y36cg776rgd5b		



- Estendere tabelle storiche/poco utilizzate in Azure
 - Vantaggio economico
 - Disponibilità on-premise
 - Trasparenza applicativa
- In maniera sicura
 - Always Encrypted
 - Row Level Security
- Backup
 - Include solo dati locali
 - Meccanismi per garanzia integrità

«STRETCH» DATABASE TEMPORALI

```
SELECT * FROM Department  
FOR SYSTEM_TIME  
AS OF '2010.01'
```

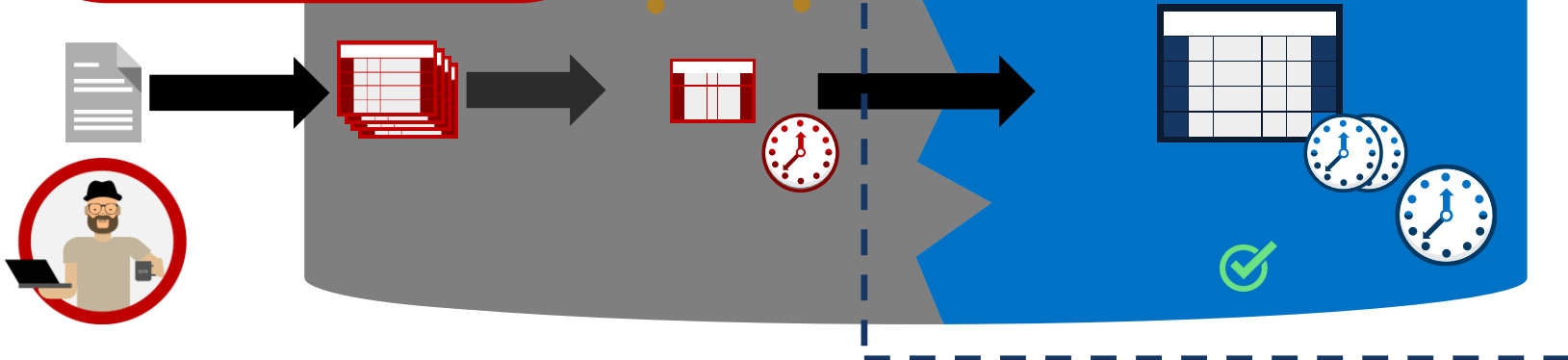
Soluzione:

Storico come «stretch table»:
PeriodEnd < "Now - 6 months"

Considerazioni:

Azure SQL Database





1. Storico **più voluminoso** dati correnti
2. Dati conservati da **3 a 10 anni**
3. «Caldi»: fino a **poche settimane/mesi**
4. «Freddi»: **interrogati raramente**



Azure SQL Database Advisor

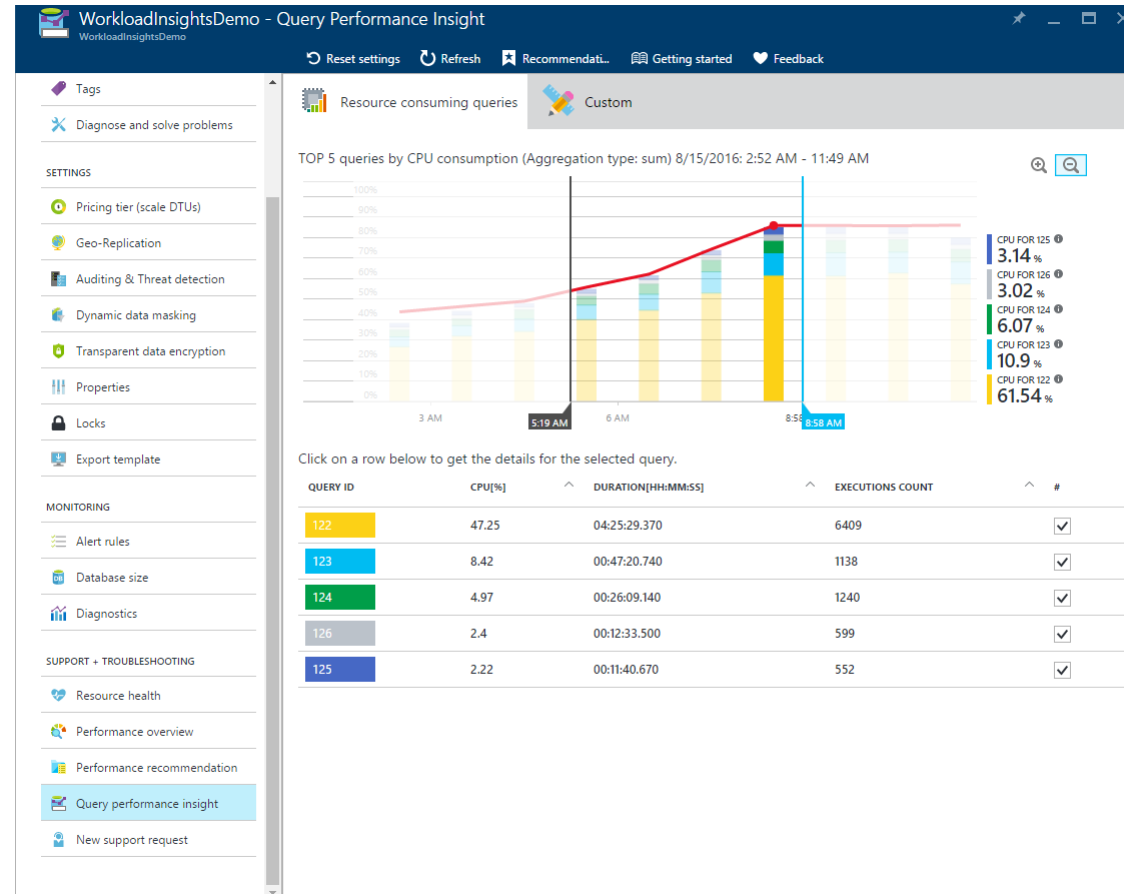
- CREATE/DROP indici
 - Automatizzabile Modello ML dietro le quinte...
- parametrizzazione query
 - Protezione contro regressioni!
- Problemi con schema

Tuning history

ACTION	RECOMMENDATION DESCRIPTION	STATUS	TIME
 DROP INDEX (PREVIEW) Initiated by: User	Index name: IR_[test_schema]_[test_table_0.182511]_1665B81581 Reason: Duplicate index	✓ Success	4/25/2016 4:28:05 PM
 CREATE INDEX Initiated by: User	Table: [test_table_0.138289] Indexed columns: [index_1],[index_2],[index_3]	✓ Success	4/25/2016 4:27:57 PM
 PARAMETERIZE QUERIES (PREVIEW) Initiated by: N/A	Scope: Entire database Reason: Non-parameterized queries are causing performanc	✓ Success	4/21/2016 4:40:30 PM
 DROP INDEX (PREVIEW) Initiated by: User	Index name: IR_[test_schema]_[test_table_0.574879]_C13F85C293 Reason: Duplicate index	✗ Error	4/25/2016 4:28:05 PM

Azure SQL Database Query Performance Insight

- TOP Query
 - Consumo CPU
 - Durata
 - Numero esecuzioni
- Basato su Query Store
- Annotazioni
 - Database Advisor



MOLTE ALTRE NOVITÀ

- Programmabilità Database Engine
 - Supporto a JSON
 - Nuove funzioni Es. STRING_SPLIT(), COMPRESS() ecc.
- Integrazione Database Engine
 - R, Polybase
- Tutti gli alti componenti (SSAS, SSRS, MDS, DQS)
- Elastic Database con Azure SQL Database
 - Pools, Sharding

Risorse

- E-Book «Introducing Microsoft SQL Server 2016»
 - <https://blogs.technet.microsoft.com/dataplatforminsider/2016/06/09/get-a-head-start-on-sql-server-2016-with-this-how-to-guide>
- Nuovi database di esempio
 - <https://github.com/microsoft/sql-server-samples>
 - [https://msdn.microsoft.com/library/mt734199\(v=sql.1\).aspx](https://msdn.microsoft.com/library/mt734199(v=sql.1).aspx)

Risorse

- "SQL Server 2016 Resource Kit" short video series
 - <https://channel9.msdn.com/Blogs/SQL-Server-2016-Training-Kit>
- «It just runs faster»
 - Blog: <https://blogs.msdn.microsoft.com/bobsq1/tag/it-just-runs-faster>
 - Ignite 2016: <https://myignite.microsoft.com/videos/2548>
- Approfondimento Database Temporal SQL Saturday #495
 - <http://www.sqlsaturday.com/495/Sessions/Details.aspx?sid=44544>

A night cityscape featuring several tall skyscrapers with illuminated windows. In the foreground, a multi-lane highway is visible with long-exposure light trails from cars, showing red and white streaks. A pedestrian bridge with a railing and some greenery is also visible. The overall scene is a vibrant urban environment at night.

Domande?

Materiale su

<http://www.communitydays.it/>

www.futuredecoded.it



#FutureDecoded