



# Temporal Data in Azure PostgreSQL

Danilo Dominici

[ddominici@sqlserverinfo.it](mailto:ddominici@sqlserverinfo.it)

# Sponsors & Organizers



UNIVERSITÀ  
POLITECNICA  
DELLE MARCHE

# Chi è Danilo Dominici

- Consulente Senior : database & BI
  - SQL Server (DWH & Power BI)
  - PostgreSQL
  - Redis
- Creatore di SQL Start! ([www.sqlstart.it](http://www.sqlstart.it))
- Community speaker, MS Certified Trainer dal 2000, 6x Data Platform MVP



2000+



2014-2020



# Tabelle temporali



# Tabelle temporali

- Strutture-dati che **conservano in automatico lo stato passato di ogni riga** e permettono query “time-travel” senza dover gestire manualmente tabelle di audit o trigger
- Lo standard ISO/IEC 9075 definisce due assi temporali:

Asse	Significato	Colonne tipiche	Sinonimi comuni
<b>Valid-time</b>	Quando il dato è (o era) “vero” nel dominio applicativo	VALID_FROM, VALID_TO	“application time”
<b>Transaction-time</b>	Quando il dato è stato fisicamente scritto nel DB	SYS_START, SYS_END	“system time”

- Una tabella può implementare uno dei due assi (mono-temporale) o entrambi (bi-temporale).

# Tabelle temporali

- **Coerenza ACID:** il salvataggio della riga attuale e della versione storica avviene nella stessa transazione.
- **Integrità temporale:** il motore impedisce sovrapposizioni nei periodi.
- **Indicizzabilità:** è possibile indicizzare colonne normali e quelle di periodo.
- **Gestione del retention:** policy per troncare automaticamente la history
- **Compatibilità con replica e CDC:** le versioni storiche viaggiano con la tabella principale.
- **Limiti:** – niente TRUNCATE; – alcune opzioni di ALTER TABLE richiedono la disattivazione temporanea del versioning; – attenzione allo spazio su disco (la history può crescere rapidamente).



# Scenari di utilizzo tipici

Scenario	Perché le tabelle temporali sono utili
<b>Audit e tracciabilità SOX/GDPR</b>	Log automatico di chi ha cambiato cosa e quando, senza codice extra.
<b>Ricostruzione di report retrospettivi</b>	Ricreare il bilancio “come appariva il 31 dicembre” anche dopo correzioni successive.
<b>Slowly Changing Dimension (SCD Tipo 2)</b>	Nei data-warehouse basta una sola tabella per tenere versioni storiche dei master data.
<b>Debug di incidenti</b>	“Il valore era giusto ieri?” – basta una query AS OF.
<b>Analisi bitemporale</b>	Studiare discrepanze fra “quando il fatto è avvenuto” e “quando è stato registrato” (es. ritardi di back-office).

# Scenari di utilizzo tipici

Scenario	Perché le tabelle temporali sono utili
<b>Ricalcoli fiscali o assicurativi</b>	Norme o tariffe retroattive: applico la versione del dato valida in quel periodo.
<b>Machine Learning basato su serie storiche</b>	Estraggo dataset al volo per qualsiasi finestra temporale senza snapshot manuali.
<b>Simulazioni “what-if”</b>	Copio una versione storica in ambiente di test per stress-test o migrazioni.
<b>Event sourcing light</b>	In sistemi OLTP tradizionali posso ri-creare l'evento (insert/update/delete) dal diff fra versioni.



# PostgreSQL e i dati temporali



# PostgreSQL standard: cosa c'è

- PostgreSQL non integra ancora il supporto nativo allo standard: è oggetto di discussione ma non è ancora stato implementato nella v17
- Si possono però ottenere funzioni equivalenti tramite:
  - **Estensioni C o PL/pgSQL**
  - **Trigger + History table**: al verificarsi di AFTER INSERT/UPDATE/DELETE si sposta la riga manualmente
  - **Range type + exclusion constraint**: Colonna di tipo tstzrange per il periodo e un indice GiST con esclusione per evitare sovrapposizioni
  - **MVCC system columns** (xmin e xmax): utile per il debug, ma non per conservazione a lungo termine (VACUUM li reinizializza)

# PostgreSQL standard: cosa manca

Limite	Commento
<b>Mancanza di sintassi FOR SYSTEM_TIME</b>	Le query richiedono filtri espliciti (WHERE TIMESTAMP '2025-05-01' <@ period).
<b>Nessun DDL automatico</b> per ALTER TABLE con history	Se aggiungi una colonna, va replicata manualmente nella _hist.
<b>Chiavi esterne temporali</b>	Non c'è enforcement nativo (temporal FK) – occorre logica custom. ( <a href="#">The World of Data</a> )
<b>Riutilizzo di xmin</b>	Non affidabile per archiviazioni oltre il ciclo di VACUUM.
<b>Estensioni non disponibili</b> su alcuni managed services	Su RDS/Aurora o Cloud SQL servono versioni in puro PL/pgSQL. ( <a href="#">GitHub</a> )

Demo





Le estensioni per i dati  
temporali



# Le principali estensioni per i dati temporali

Extension	Temporal Dimensions	Impl. & Cloud-Friendliness	Last Release (Stars)	Pros	Cons
<b>temporal_tables</b>	System-time only	C extension (needs superuser)	v1.2.2 · 960★	Fast, mature, simple versioning() trigger	Not valid-time; blocks on RDS/Azure
<b>periods</b>	Uni-, bi-, tri-temporal (SQL 2016 syntax)	C extension	v1.2 · 290★	Near-standard SQL syntax; supports WITHOUT OVERLAPS constraints	Early-stage; limited docs; no managed-DB support
<b>temporal_tables_plpgsql</b> (NearForm rewrite)	System-time	Pure PL/pgSQL (works on RDS/Azure)	2024-02 tag	Cloud-friendly; no compile step	~2× slower than C; fewer guard rails

# Le principali estensioni per i dati temporali

Extension	Temporal Dimensions	Impl. & Cloud-Friendliness	Last Release (Stars)	Pros	Cons
<b>table_version</b>	Snapshot-by-revision (manual)	PL/pgSQL + triggers	v1.4 · 180★	Fine-grained revision API, diff functions	Requires explicit revision calls; no automatic period logic
<b>pgmemento</b>	System-time + DDL history	PL/pgSQL (cloud OK)	v0.7 · 390★	Tracks DML <b>and</b> DDL; JSONB delta log; restore helpers	Heavier schema; no valid-time; learning curve
<b>pg_bitemporal</b>	Full bi-temporal	PL/pgSQL, relies on GiST/exclusion	2023-11 · 150★	True application + system time; constraint-based integrity	Less active; no automated history partitioning

Demo





# Considerazioni finali

- Utilizzare indici BRIN per le tabelle storiche
- Usare le esclusioni negli indici GiST per gli *overlaps*
- Partizionare i dati storici per anno o anno/mese
- La tabella storico cresce rapidamente: regolare la soglia di AUTOVACUUM
- Utilizzare DEFERRABLE INITIALLY IMMEDIATE per garantire la coerenza anche con transazioni lunghe

# Resources

- <https://sigmodrecord.org/publications/sigmodRecord/1209/pdfs/07.industry.kulkarni.pdf>
- <https://www.wiscorp.com/SQLStandards.html>
- <https://www.red-gate.com/simple-talk/author/borisnovikov/>

# Resources

- PERIODS  
<https://github.com/xocolatl/periods>
- TEMPORAL\_TABLES  
[https://github.com/nearform/temporal\\_tables](https://github.com/nearform/temporal_tables)
- BI-TEMPORAL  
[https://github.com/hettie-d/pg\\_bitemporal/blob/master/README.md](https://github.com/hettie-d/pg_bitemporal/blob/master/README.md)



Domande ?





Grazie 😊