BACK TO THE FUTURE WITH TEMPORAL TABLES

Randolph West
Data Platform MVP

BORNSQL

System-Versioned Temporal Tables

Keeps a full history of data changes

System-Versioned Temporal Tables

Allows easy point-in-time analysis

System-Versioned Temporal Tables

Period of validity for each row is managed by the database engine

System-Versioned Temporal Tables

- two **`PERIOD`** columns
- **`datetime2`** data type
- records validity period per row
- whenever a row is modified

System-Versioned Temporal Tables

- references a *history* table
- with a mirrored schema
- stores previous version of the row
- whenever a row is modified

System-Versioned Temporal Tables

History tables can be created manually, or by the database engine

Why Temporal?

Audit all data changes and perform data forensics when necessary

This is a marketing slide

Why Temporal?

Audit all data changes and perform data forensics when necessary

Why Temporal?

Reconstruct the state of the data at any time in the past

Why Temporal?

Calculate trends over time

Why Temporal?

Maintain slowly changing dimensions for decision-support applications

Why Temporal?

Recover from accidental data changes and application errors

Why Temporal?

Backward compatibility with `HIDDEN` period columns

Limits and Considerations

Primary Key in the current table,
no primary key in the history table
(or any type of constraints)

Limits and Considerations

The history table must be stored in the same database as the current table

Limits and Considerations

The history table is **PAGE** compressed by default

Limits and Considerations

Partitioned tables will store the history table in the default file group

Limits and Considerations

`(n)varchar(max)`, **`varbinary(max)`**, **`(n)text`**, and **`image`** incur significant storage and performance costs

Limits and Considerations

`TRUNCATE TABLE` is not supported while **`SYSTEM_VERSIONING`** is **`ON`**

Limits and Considerations

Direct modification of history data is not supported with system versioning

Limits and Considerations

Read them all:

https://docs.microsoft.com/en-us/sql/relational-databases/tables/temporal-table-considerations-and-limitations

Managing historical data retention

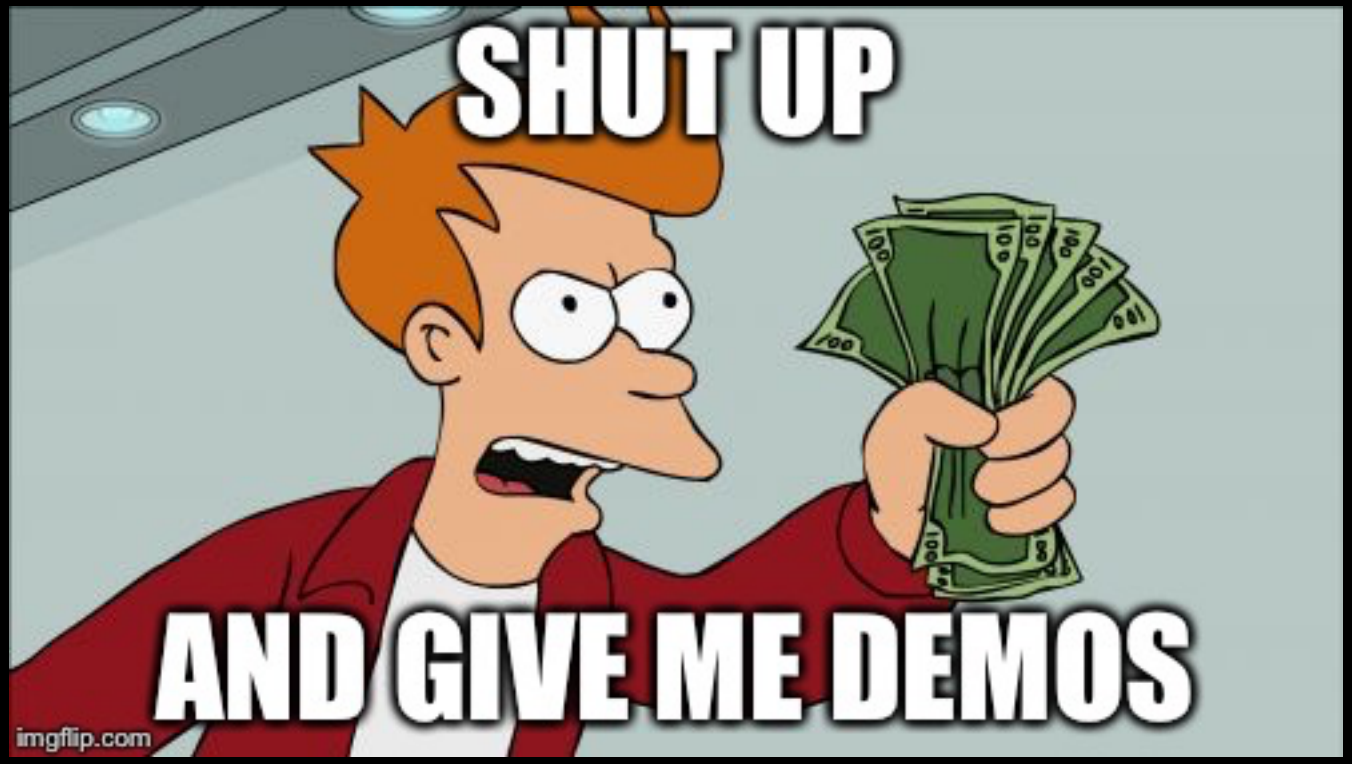
- Stretch database
- Table partitioning
- Custom cleanup
- Retention Policy

https://docs.microsoft.com/en-us/sql/relational-databases/tables/manage-retention-of-historical-data-in-system-versioned-temporal-tables

Memory-Optimized Temporal Tables

- Current table in-memory
- History table on disk
- Internal in-memory staging table
- Works on Standard Edition

Show Me The Money



imgflip.com