

# Remember Back When?

## Temporal Tables in SQL Server 2016+

Brian Hansen

brian@tf3604.com

@tf3604



\*PASS

**SQLSATURDAY**

SALT LAKE CITY | AUG 29 2020



Thank you to  
our Global  
Sponsors and  
Supporters





## The First PASS Virtual Summit

You don't want to miss this.

[Register Now](#)

We are thrilled to say the first ever PASS Virtual Summit will be taking place this November!

We will be launching PASS Summit into a whole new world, and with that comes exciting opportunities with what we can offer you. We will go beyond simply moving sessions online to fully embracing everything a virtual environment can offer.

**Find out more at [PASS.org/summit](https://PASS.org/summit) - Utah County SQL Group discount code - LGDIS6I0T (\$50 off registration)**





# Session Evaluations

<https://www.sqlsaturday.com/997/Sessions/Session-Evaluation>

Alternatively go to the Schedule and click on the session you would like to give feedback on and click ***Session Evaluation***





## ***Utah Geek Events slack channels for attendees to chat***

To get an invite to the Slack Space

<http://bit.ly/2bm3vwW>

### ***Channels***

#sqlsaturday997\_room1

#sqlsaturday997\_room2

#sqlsaturday997\_room3

#sqlsaturday997\_room4





Brian Hansen



brian@tf3604.com



@tf3604.com

- 20+ Years working with SQL Server
  - Development work since 7.0
  - Administration going back to 6.5
  - Fascinated with SQL internals

[www.tf3604.com/temporal](http://www.tf3604.com/temporal)



# Temporal Tables

- Temporal = time-based = system versioned
- Main purposes
  - Logging / Reversal of changes / Anomaly detection
  - Point-in-time business analytics / trends
- Other purposes – but with complexity / caveats
  - Auditing
  - Change detection / Temporal data capture
  - Slowly-changing dimensions



# The Scenario

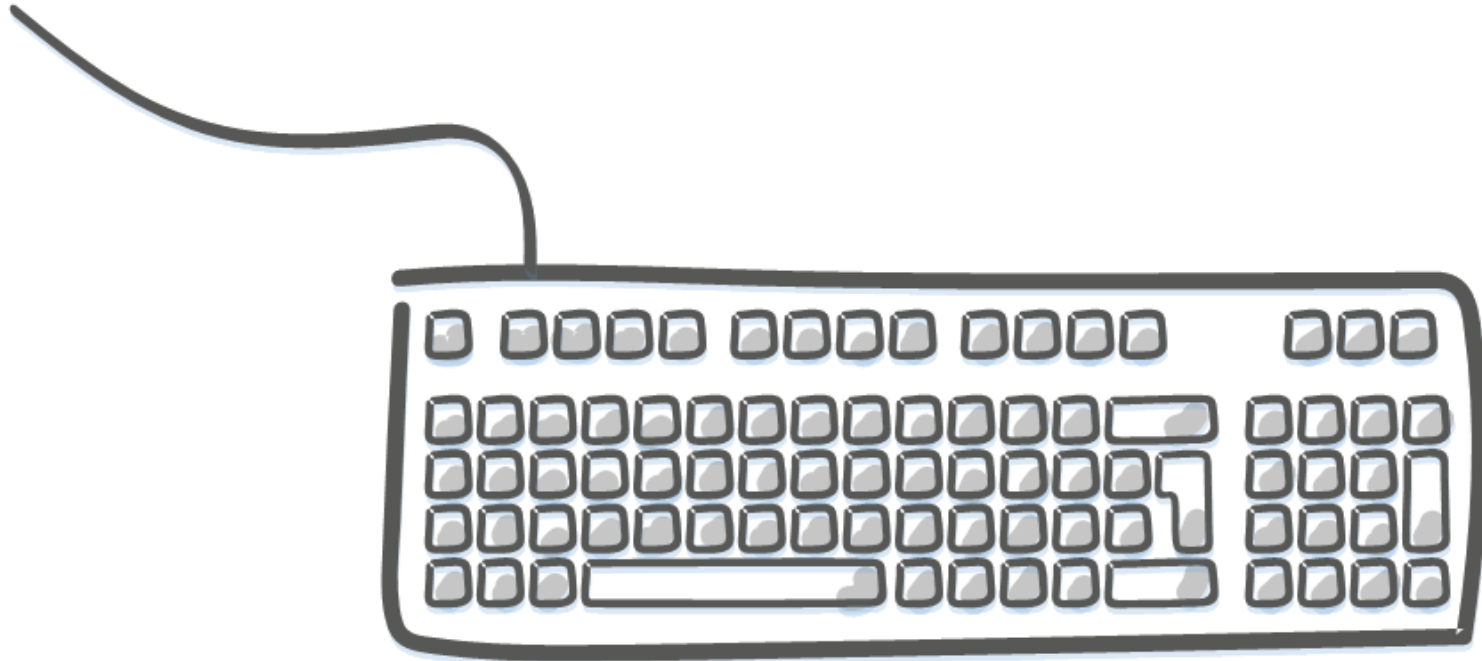
Congratulations! You have just been awarded a lucrative contract with international automobile manufacturer FordoyotaBenz.





# Demo

## Automobile Tracking



# Appendix

## Usage summary



# Summary

Required, indicates the columns that will contain the times where the row is valid; must be datetime2 and not null

Required, indicates which columns make update system time validity period

```
create table dbo.AutoModel
```

```
(
```

```
  ModelId int not null identity(1,1),  
  Description nvarchar(100) null,
```

```
  ValidFrom datetime2 generated always as row start hidden not null,  
  ValidTo datetime2 generated always as row end hidden not null,
```

```
  period for system_time (ValidFrom, ValidTo)
```

```
)
```

```
with (system_versioning = on (history_table = history.AutoModelHistory));
```

Optional; hides column from select \* and insert

“system\_versioning = on” required to make this a temporal table; “history\_table” is optional

Must specify schema name (even if “dbo”)



# SQL 2017 Enhancements: Retention

-- Note: This is usually ON by default  
alter database AutoTracker set  
temporal\_history\_retention on;

```
select  
is_temporal_history_retention_enabled  
from sys.databases  
where name = 'AutoTracker';
```



# SQL 2017 Enhancements: Retention

```
create table dbo.Customer  
(  
...  
)  
with (system_versioning = on  
(history_table = history.CustomerHistory,  
history_retention_period = 3 months));
```



# SQL 2017 Enhancements: Retention

```
alter table dbo.Customer  
set (system_versioning = on  
(history_retention_period = 7 weeks));
```



# SQL 2017 Enhancements: Retention

```
select history_retention_period,  
history_retention_period_unit,  
history_retention_period_unit_desc  
from sys.tables  
where name = 'Customer';
```

| history_retention_period | history_retention_period_unit | history_retention_period_unit_desc |
|--------------------------|-------------------------------|------------------------------------|
| 7                        | 4                             | WEEK                               |



# Summary

**Temporal querying:** `FROM TableName FOR SYSTEM_TIME _____`

Point in time

`AS OF '2019-10-11 08:30:00'`

Full history

`ALL`

Between (ValidFrom <= EndTime AND ValidTo > StartTime)

`BETWEEN '2019-10-11 08:30:00' AND '2019-10-12 17:00:00'`

From (ValidFrom < EndTime AND ValidTo > StartTime)

`FROM '2019-10-11 08:30:00' TO '2019-10-12 17:00:00'`

Contained in (ValidFrom >= StartTime AND ValidTo <= EndTime)

`CONTAINED IN ('2019-10-11 08:30:00', '2019-10-12 17:00:00')`



# Summary

```
select
```

```
    cast('2017-06-10 10:10:00' as datetime2)  
    at time zone 'Central Standard Time'  
    at time zone 'UTC';
```

```
select * from sys.time_zone_info;
```



# Summary

- New catalog objects
  - `sys.periods` (view)
  - `sys.tables.temporal_type` (column)
  - `sys.tables.temporal_type_desc` (column)
  - `sys.tables.history_table_id` (column)
  - `sys.columns.generated_always_type` (column)
  - `sys.columns.generated_always_type_desc` (column)



# Thank You

This presentation and supporting materials can be found at [www.tf3604.com/temporal](http://www.tf3604.com/temporal).

Slide deck

Scripts

Sample databases

brian@tf3604.com • @tf3604

