

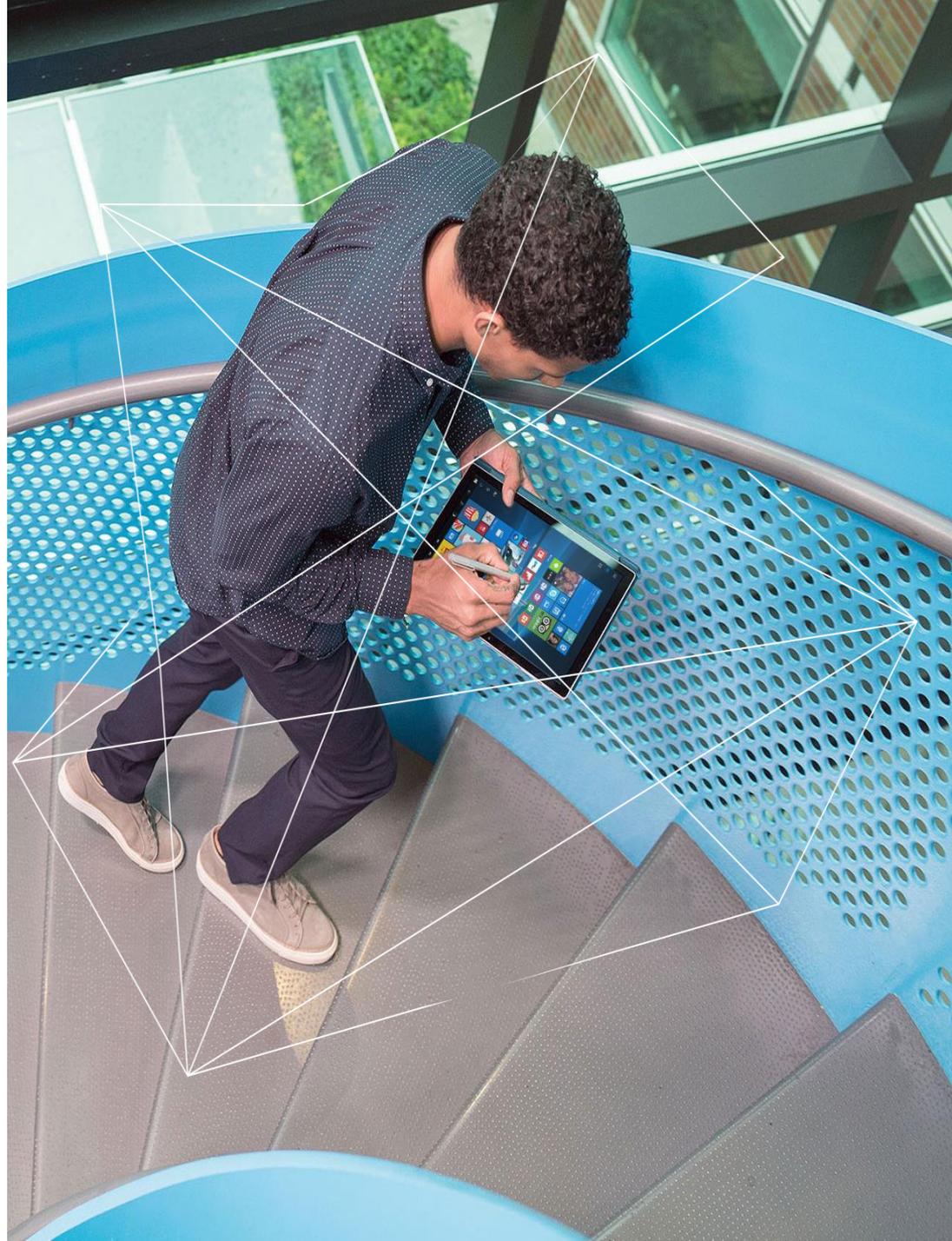


Fun with Legal Information in Microsoft SQL Server: Determining the Law to Be Used

Matija Lah, SQL Server MVP



#ntk18



Speaker Information

- Matija Lah
 - Lawyer
 - SolidQ CEE, Mentor
 - Microsoft MVP (SQL Server), since 2007
 - 15+ years of SQL Server experience
 - Specializing in Legal Information Management and Natural Language Processing
 - Current focus areas
 - Database Engine
 - Integration Services
 - Analysis Services (incl. Data Mining)
 - Based in Ljubljana, Slovenia
 - mlah@solidq.com, @MatijaLah



Agenda

- Background
- Determining the law to be used



Background

- Legal information
 - A highly complex data management challenge
 - Document-oriented
 - Multiple versions (from few to hundreds)
 - Generally, versions do not overlap; very rarely they do
 - Entities inter-linked (explicitly, implicitly)
 - Entities associated into groups (explicitly, implicitly)



Document Types

- Laws
 - Define the rules imperatively,
erga omnes
- Contracts
 - Define the rules autonomously,
inter partes
- Decisions
 - Implement and enforce the rules
(correctly, but also incorrectly)
- Facts
 - Describe actual events
(truthfully, as well as untruthfully)



Describe potential events, in an abstract way



Example

- Law

- „Whoever takes the life of another human being shall be sentenced to imprisonment for not less than five years.“
(+ several associated rules)
- Multiple versions over time

- Fact

- „David shot John dead.“
- Initial event, followed by legal procedure
(a set of events in time)
- Multiple versions possible
(„Truth is in the eye of the beholder.“)



Determining the Law to be Used

- Material
 - Determining whether an event requires legal action
 - Time and place of the event, as well as other relevant circumstances
- Procedural
 - Determining what legal action is needed, how, by whom, against whom, etc.
 - Times and places of the legal actions, as well as other relevant circumstances



Determining the Law to be Used

- E.g. act of murder
 - David shot John in the evening, outside John's home.
 - But also:
 - David contemplated shooting John
 - David realized the consequences of his actions
 - Determining guilt, sanction
 - *Ignorantia iuris nocet.*
(Not knowing the law is harmful.)
 - *Nullum crimen, nulla poena sine lege praevia.*
(There is no crime, and no punishment, without a pre-existing law.)
 - *In dubio pro reo.*
(When in doubt, for the accused.)



Determining the Law to be Used

- E.g. in criminal proceedings in a murder case
 - Individual acts
 - The time (and place, etc.) when/where each act is (should be/must be) performed
 - Continuous acts
 - The time (and place, etc.) when/where each act commences
 - The time (and place, etc.) when/where each individual step of a multiple-step act commences
 - The time (and place, etc.) when/where a system of acts reaches conclusion



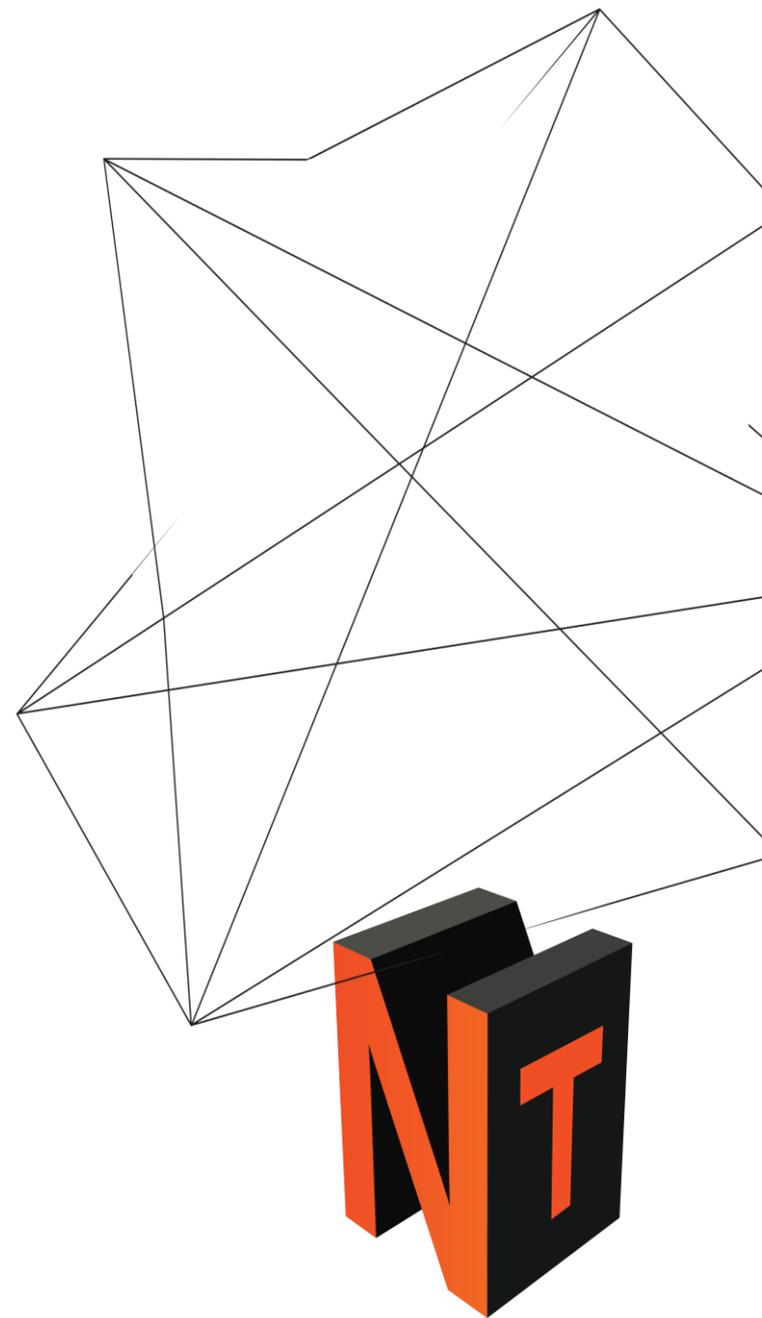
Unless ...

... the law says otherwise.



Determining the correct version of the Law

Demo



What is Needed

- Temporal database
 - Records changes
 - Keeps transactional history
(automated, built-in, DML-based row history management)
 - Is time-aware
 - Allows applications to control temporal boundaries
(controlled explicitly)
- Temporal constraints
 - Date/Time/Interval native constraints
- Temporal queries
 - Date/Time/Interval native predicates
(PERIOD FOR [SYSTEM TIME], CONTAINS, OVERLAPS, EQUALS, [IMMEDIATELY] PRECEDES, [IMMEDIATELY] SUCCEEDS, AS OF, BETWEEN)



What is Available in SQL Server 2016+

- Temporal tables
 - Transactional history only (CTP is not RTM!)
 - Not known, at this time, whether the complete ANSI SQL:2011 standard regarding temporal data will ever be implemented in SQL Server
- „Roll your own“
 - Multiple options at your disposal, but not available natively



SQL Server 2016+ System_Versioning

- Historical table maintained by the system automatically
 - Two tables, actually:
 - Principal table holds current rows
(existing table can be reused, but behaviour changes)
 - Historical table holds former versions of rows
(existing table can be re-used, or historical table is created by the system automatically)
 - Temporal boundaries are maintained automatically by the Database Engine
(once set cannot be changed)
 - Transactions always represent current, or past, but never future, events



Conclusion

- Transactional history tables are not suitable for legal information
 - Temporal boundaries governed by the Engine
 - Restricted to past and present
 - One-dimensional
(in reality: publication, validity, use, effect – multiple sets of temporal dimensions)
 - Insufficient temporal constraints
(Start Time \leq End Time)
 - Temporal predicate syntax requires literals or variables
(requires additional complexity to correlate queries)

