University of Zurich[UZH]

Department of Informatics

University of Zürich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zürich

Lukas Yu

Prof. Dr. Michael Böhlen
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, 1 March 2017

Vertiefung
Topic: Integration of Ongoing Time Points into PostgreSQL

Data that is associated with time intervals to denote its validity is present in many real-world applications. Examples include employment contracts, insurance policies, and telecommunication contracts. A contract often has a fixed start date and an end date that keeps increasing as time passes by until the contract is modified or terminated. The end date is *ongoing*. A typical example of an ongoing end date is now.

Current database systems like PostgreSQL provide the keywords CURRENT_TIME, CURRENT_DATE, and CURRENT_TIMESTAMP to incorporate the ongoing time point now for different time granularities. These keywords are always instantiated at a query's compile time, i.e., they are replaced with the time the query is evaluated. Thus, none of the current database systems allows storing ongoing time points uninstantiated in a relation in order to, e.g., preserve the information that a contract has an ongoing end date.

In this project, the student should carefully study the concept of ongoing time points and implement a set of ongoing time points into the kernel of PostgreSQL to provide native support for ongoing time points without the need of instantiating them. Additionally, the student should understand the impact of supporting ongoing time points in a database system on the functions and predicates for the corresponding data types.

Tasks

1. Literature study on the basic ongoing time point now [1], possible implementations of now [3, 4], and why the ongoing time point now is not sufficient to achieve query results that remain valid as time passes by [2].
2. Implement time domain $\Omega^V = \{a, c+, +c, a+c\}$ (definition in [2]) in PostgreSQL by extending the *date* data type. The ongoing time points must not be instantiated in your

implementation.

3. Discuss the challenges of a database-internal implementation of ongoing time points (extension of data types) and its benefits in comparison with representing ongoing time points with existing, unmodified data types.
4. Discuss the impact of your extension of the *date* data type on the corresponding time interval data type *daterange* and the functions and predicates that PostgreSQL provides for the two data types.
5. Write a report.

References

[1] Clifford, James and Dyreson, Curtis and Isakowitz, Tomás and Jensen, Christian S. and Snodgrass, Richard Thomas. On the Semantics of Now in Databases. *ACM Transactions on Database Systems*, 1997.

[2] Mülle, Yvonne and Böhlen, Michael H. Generally Valid Queries in Databases with Ongoing Time Points. to be published.

[3] Stantic, B. and Thornton, J. and Sattar, A. A novel approach to model NOW in temporal databases. In *Proceedings. 10th International Symposium on Temporal Representation and Reasoning and Fourth International Conference on Temporal Logic.*, 2003.

[4] K. Torp, C. S. Jensen, and R. T. Snodgrass. Modification Semantics in Now-Relative Databases. *Information Systems*, 29(8):653–683, Dec. 2004.

Supervisor: Yvonne Mülle (muelle@ifi.uzh.ch)

Start date: 1 March 2017

End date/Oral Exam: 4 April 2017, 15:45

University of Zurich
Department of Informatics

Prof. Dr. Michael Böhlen
Professor