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

Oliver Leumann

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

Zürich, March, 2013

BSc Thesis (17ECTS)

Topic: Query Compilation of Statement Modifiers.

Statement modifiers [1] have been designed to provide a systematic and comprehensive query language for the management of temporal data. The crucial concept of statement modifiers is that first, queries are written as traditional and well known SQL queries, and second statement modifiers are prepended to queries to modify their behavior. In particular, if the `SEQ VT` modifier is prepended the statement is evaluated on each snapshot of a temporal database [1].

Consider as an example the following relation **p**, that records projects assignments to departments, where **D** is the department in charge, **N** is the project number within a department, and [**Ts**, **Te**) is the project period. For instance, tuple r_1 records a project for the DB department with project number 1 from February 2013 to July 2013.

p

	D	N	Ts	Te
r_1	DB	1	Feb-2013	Aug-2013
r_2	DB	2	May-2013	Jan-2014
r_3	AI	1	Apr-2013	Sep-2013

Consider the following query **q**.

q: What is the number of projects per department?

Query **q** asks for a count of the number of projects per department and without time can be answered with the SQL-query:

```
SELECT D, Count(*)
FROM p
```

```
GROUP BY D
```

Since we are interested in the *sequenced* result, i.e., the count at each point in time, we prepend the statement modifier for sequenced queries `SEQ VT` as follows:

```
SEQ VT
SELECT D, Count(*)
FROM p
GROUP BY D
```

and get the result:

q

D	Count	Ts	Te
DB	1	Feb-2013	May-2013
DB	2	May-2013	Aug-2013
DB	1	Aug-2013	Jan-2014
AI	1	Apr-2013	Sep-2013

That corresponds to counting the number of projects per department at each point in time. For instance, the first result tuple records that for the DB department 1 project exists for February 2013 to April 2013. The second result tuple, records that for the DB department 2 projects exit from May 2013 to July 2013.

Within this project statement modifiers shall be integrated into PostgreSQL. First, the grammar of statement modifiers shall be integrated into the database parser. Second, queries with statement modifiers shall be translated to SQL queries with temporal primitives [3, 2].

Tasks:

- Definition of an SQL language with statement modifiers that allows to scale attribute values at query time.
- Definition of SQL mapping for the DBMS parser [4], `src/backend/parser/gram.y`[1]
- Development of translation algorithm for the query compilation
- Implementation and testing in PostgreSQL
- Writing thesis (approx. 50 pages)
- Presentation of the results at the dbtg group meeting (approx. 10 min.).

Setup:

Within this project the student will have the opportunity to take a closer look at the inside and implementation of the parser of the database system PostgreSQL, with regular weekly meetings and strong support with PostgreSQL.

[1] `http://www.ifi.uzh.ch/dbtg/research/align.html`

References

[1] M. H. Böhlen, C. S. Jensen, and R. T. Snodgrass. Temporal statement modifiers. *ACM Trans. Database Syst.*, 25(4):407–456, 2000.

[2] A. Dignös, M. Böhlen, and J. Gamper. Query time scaling of attribute values in interval timestamped databases. In *Proceedings of the 29th IEEE International Conference on Data Engineering*, ICDE '13, pages 1304–1307. IEEE, 2013.

[3] A. Dignös, M. H. Böhlen, and J. Gamper. Temporal alignment. In *Proceedings of the 2012 international conference on Management of Data*, SIGMOD '12, pages 433–444. ACM, 2012.

[4] H. Garcia-Molina, J. D. Ullman, and J. Widom. Parsing. In *Database systems - the complete book (international edition)*, chapter 16.1, pages 788–795. Pearson Education, 2002.

Supervisor: Anton Dignös

Start date: 14.05.2013

End date: 14.09.2013

University of Zürich
Department of Informatics

Prof. Dr. Michael Böhlen