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

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

Zürich, March, 2018

**BSc Thesis (18 KP)**
**Datenbanktechnologie**

**Topic: Optimization of mixed queries in MonetDB system**

The *relational matrix algebra* (RMA) is an extension of the relational algebra with matrix operations. It is a closed algebra that allows to apply matrix operations directly to relations and keeps contextual information of relations during operations. The approach opens up opportunities for developing new optimization techniques that consider both relational and matrix algebra operations.

The goal of this BSc thesis is to develop and implement optimization rules for mixed queries, i.e., queries that require to perform relational and matrix operations, and to empirically evaluate the efficiency of the implemented optimizations. The implementation should be integrated into the MonetDB system.

MonetDB[1] is a column-oriented database system that stores each attribute as a *binary association table* (BAT). A BAT is a table with two columns: the OIDs and the values. MonetDB constructs three main structures during query processing: the relation , the statement tree, and the MAL plan. In the *relation tree* each internal node is a relational algebra operation. In a *statement tree* each internal node is a BAT algebra expression or a list of BAT algebra expressions. The *MAL plan* is a sequence of BAT operations.

The work includes the following tasks:

1. Implement the matrix addition between two tables:
   (a) Write an SQL parser extension that recognizes the following new SQL command, where R and S are table names and A, B are lists of attributes of R and S,

[1] `https://www.monetdb.org/Documentation`

respectively (about 1 week).

```
SELECT * FROM ADD (R ON A, S ON B)
```

   (b) Extend the relation tree and the statement tree of MonetDB with the addition functionality, i.e., implement the addition between unordered tables R and S (about 2 weeks).

   (c) Extend the relation tree and the statement tree with ordering functionality, i.e., implement the sorting of R and S before computing the addition (about 1 week).

2. Elaborate and implement an optimization rule that pushes down selection on the example of addition operation:

   (a) Determine all required constraints that must be fulfilled in order to apply the rule (about 1 week).

   (b) Implement the optimization on the relation tree (about 2 weeks).

   (c) Implement the optimization on the statement tree (about 3 weeks).

3. Evaluate the efficiency of the implemented optimization, i.e., evaluate the performance of the optimized queries and compare it to the performance of non-optimized queries:

   - Run an experimental analysis with different table sizes, varying the most important parameters, such as number of attributes in application and descriptive parts, selectivity of predicates, and sorting of input relations (about 3 weeks).

4. Write a thesis (approximately 50 pages).

5. Present your thesis in a DBTG meeting (18.09.2018, 30 minutes).

**References**

**Supervisor:** Oksana Dolmatova
**Start date:** 13.03.2018
**End date:** 13.09.2018

University of Zürich
Department of Informatics

Prof. Dr. Michael Böhlen