

Towards Online Relational Schema Transformations

Lesley Wevers, M. Hofstra, M. Tammens, M. Huisman, M. van Keulen
l.wevers@utwente.nl

Formal Methods and Tools Group & Databases Group
University of Twente

Current relational database systems are ill-equipped for changing the structure of data while the database is in use. This is a real problem for systems for which we expect 24/7 availability, such as telecommunication, payment, and control systems. As a result, developers tend to avoid making changes because of the downtime consequences. The urgency to solve this problem is evident by a multitude of tools developed in industry, such as `pt-online-schema-change`¹ and `oak-online-alter-table`². Also, MySQL recently added limited support for online schema changes³.

Contributions We want to draw the attention of the database community to *the problem of online schema changes*. We have defined *requirements for online schema change mechanisms*, and we have *experimentally investigated existing solutions*. Our results show that current solutions are unsatisfactory for complex schema changes. We propose *lazy schema changes* as a solution.

Requirements As a basis for our experiments, we have identified *general requirements* for online schema transformations mechanisms, and *specific requirements* for the relational data model. We assert that, ideally, schema transformations and data conversion can be performed *transactionally*, i.e., satisfying the ACID properties, and *online*, i.e., without blocking other transactions. Moreover, we have identified a basic set of relational schema transformations that, if these transformations can be performed online, and if they can be composed using transactions, they should allow for most practical schema transformations to be performed online.

Experimental Setup To assess the performance and behaviour of existing mechanisms for online schema changes, *we have developed an experiment based on the standard TPC-C benchmark*. For each of the relational schema transformation classes that we have identified, we chose a representative transformation for the TPC-C schema. We perform the schema change online while the TPC-C benchmark is running, and measure the impact on the TPC-C transaction throughput. We have performed our experiment on PostgreSQL, which does not support online schema changes, MySQL, which supports basic online schema changes, and using `pt-online-schema-change` on MySQL, as a representative for tools that use triggers to allow online schema changes.

Results We found that *existing solutions are inadequate except for the simplest of schema changes*. Some single-relation transformations can be performed transactionally and online. However, existing solutions do not allow schema transformations to be composed using transactions. As a result, in complex transformations, intermediate states can be exposed to database programs, which are non-trivial to handle correctly. A secondary problem is that these solutions are much slower than offline transformations, which may not be acceptable for certain applications.

Proposal We propose a more fundamental solution based on *lazy schema transformations*. The main idea is that schema changes can be described as a view on the existing schema, which can be materialized lazily to perform the schema transformation. The data in the new schema is immediately accessible by computing parts of the view on demand. For a large number of cases we expect that this approach allows schema transformations without any downtime, and with minimal impact on running transactions, while the ACID properties are maintained. Moreover, lazy transformations can naturally be composed as transactions, allowing complex online schema transformations. We are developing an implementation of these ideas based on a persistent functional language.

Relevant Publications

1. Lesley Wevers. Persistent Functional Languages: Toward Functional Relational Databases. June 2014. SIGMOD14 PhD Symposium, Snowbird, Utah, USA.
2. Lesley Wevers. A Persistent Functional Language for Concurrent Transaction Processing. Masters thesis. August 2012, University of Twente, the Netherlands.

¹<http://www.percona.com/doc/percona-toolkit/2.1/pt-online-schema-change.html>

²<http://openarkkit.googlecode.com/svn/trunk/openarkkit/doc/html/oak-online-alter-table.html>

³<http://dev.mysql.com/doc/refman/5.6/en/innoDB-online-ddl.html>