Project Proposal
Transforming SQL Queries

Calvin Li, Ben Mishkanian

April 26, 2015

Problem Description and Motivation

Database engines, like any other software, must be tested for functional correctness. Random testing, which involves randomly generating input values, is easy to implement but on its own cannot test for correctness; it can only test for crashes or hangs. Instead, we can use differential testing, where we run the same input on two programs, in order to test correctness. This approach works well in testing compilers, but not as well for testing database engines, because SQL is not as standardized as other programming languages. Different engines have slightly different implementations of SQL, so pure differential testing cannot fully cover all the cases.

We will approach this problem from a different angle. We will generate a random query, then run it to get the correct output. Then, we will make some transformations to the query in a way where the new query is functionally equivalent to the old one, and run this new query. If the results are different, then the query is possibly error-revealing.

Technical Approach

We will try SQL transformation with SQLite, at least initially. It uses a relatively simple flavor of SQL, making it easier to work with. It also has not been tested as much as other engines, which means there are more bugs, so we should have more interesting results.

The idea is to run the query and look at the metadata of the output - the number of rows, min, max, and average values, etc. Then, we can alter the queries such that the output should be the same, even if we use transformations that don't guarantee equality for all queries and all databases.

Evaluation Methodology

We will evaluate our approach based on the accuracy of the transformation algorithm. For a given target query and its transformed query, we will run both and see if they have the same outputs. If they do, the transformation is more likely to be correct. If the two queries have different effects, the transformation was incorrect, and we will attempt to identify the reason for the discrepancy, and fix it if possible. If the transformation is correct, but the output is still different, then the query might reveal a bug in the engine. We will document these discrepancies for further investigation.