Session-based Browsing for More Effective Query Reuse*

Nodira Khoussainova, YongChul Kwon, Wei-Ting Liao,
Magdalena Balazinska, Wolfgang Gatterbauer, and Dan Suciu

Department of Computer Science and Engineering
University of Washington, Seattle, WA, USA
{nodira, yongchul, liaowt, magda, gatter, suciu}@cs.washington.edu

1 Introduction

Scientists today are able to generate and collect data at an unprecedented scale [1]. Afterwards, scientists analyze and explore these datasets. Composing SQL queries, however, is a significant challenge for scientists because most are not database experts.

In this work, we leverage the collaborative environment that many scientists work in, which often includes a shared database with many scientists asking queries over it. As such, we utilize the collective knowledge of all the users by providing new users with access to a log of past queries, which can be used as starting points for writing new queries. However, navigating a large log of queries can be difficult and overwhelming.

In this paper, we introduce the *Smart Query Browser* (SQB) system. SQB supports efficient retrieval of relevant queries using what we call *session-based browsing*. We also show results from a user study where we investigated whether SQB speeds up the query formulation by supporting better query reuse. [1]

2 SQB Overview

To start, SQB provides keyword search over a query log. Instead of simply listing all matching queries, it presents the results as a *set of query sessions*. A query session, as introduced in our previous work [3], is a set of queries written by a user to achieve a single task. For example, an astronomer who wants to find, in the Sloan Digital Sky Survey database [5], all the stars of a certain brightness in the r-band within 2 arc minutes of a known star, is likely to write multiple SQL queries before completing this task.

SQB thus allows users to view each result query in the context of the task that it aimed to complete. With this approach, SQB helps the user to more rapidly identify relevant queries because the user can decide on the relevance of entire sessions. It also helps users see how simple queries evolved into more complex ones. Finally, query sessions enable users to discriminate between high-quality and low-quality queries: queries that appear near the end of a session tend to be of higher quality because the author has spent time to edit and improve the query.

3 Evaluation

We performed a user study with 16 participants to investigate whether SQB speeds up the query formulation through query reuse. We find that, on average, SQB allows users to complete their tasks 2.3 times faster compared to having no access to a query browser.

* This work was partially supported by NSF CAREER award IIS-0845397 and IIS-0627585.

[1] We invite the reader to read our technical report for more details on SQB and the user study [4].

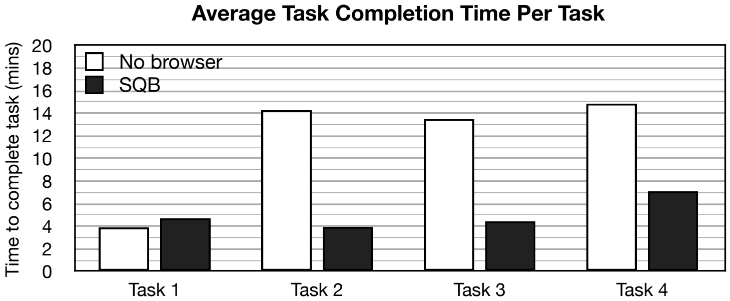

Fig. 1. Mean task completion time per interface, grouped by task.

The evaluation dataset consists of all SQL queries written by students in an undergraduate database class, offered at the University of Washington in 2008. These queries were logged as students worked on nine different problems for an assignment that used the IMDB database [2]. For the user study, we used a subset of this query log with 492 queries. Each participant in the user study was asked to translate four English sentence questions into four SQL queries.

Figure 1 presents the average completion time per task across the users. Note that a smaller completion time is better. We see that the SQB interface greatly outperforms the interface with no-browser for three of the tasks. Task 1 is a select-from-where query that can be written easily, and thus there is no benefit from SQB. In contrast, Task 4 is both difficult to write (i.e. requires a `GROUP BY` and either `TOP` or a `NOT EXISTS` subquery) but is not similar to any past query. The most similar query requires the user to make structural changes to the query before achieving the goal. Despite this heavy editing, SQB still helps users complete the task in less than half the time compared to no browser. The queries for Tasks 2 and 3 are also complex, requiring a `GROUP BY` and a self-join, respectively. However, there are similar queries in the query log. Therefore, we see a more than 3-fold improvement in speed with SQB.

4 Conclusion

We presented SQB, a tool for browsing through past SQL queries. The key insight behind SQB's design is the concept of query sessions. We showed that query sessions help speed up query composition by organizing queries in a large repository in a manner that facilitates the identification of relevant, high-quality queries to use as example.

References

1. The Fourth Paradigm: Data-Intensive Scientific Discovery, 2009.
2. IMDB course assignment. `http://www.cs.washington.edu/education/courses/cse444/08au/project/project1/project1.html`.
3. N. Khoussainova, Y. Kwon, M. Balazinska, and D. Suciu. SnipSuggest: Context-aware Autocompletion for SQL. *Proc. VLDB Endow.*, 4:22–33, October 2010.
4. N. Khoussainova, Y. Kwon, W.-T. Liao, M. Balazinska, W. Gatterbauer, and D. Suciu. SQB: Session-based Query Browsing for More Effective Query Reuse. Technical Report 2011-04-02, Department of Computer Science and Engineering, University of Washington, 2011.
5. Sloan Digital Sky Survey. `http://www.sdss.org/`.