Database Query Analyzer Integration

Ryan Hardt
University of St. Thomas
St. Paul, MN, USA
ryan.hardt@stthomas.edu

Kamil Akhuseyinoglu
University of Pittsburgh
Pittsburgh, PA, USA
kaa108@pitt.edu

ABSTRACT

This paper describes updates to Database Query Analyzer (DBQA) that increase its interoperability with other learning tools using the Learning Tools Interoperability (LTI) protocol. As a result, DBQA has been integrated with Mastery Grids and allows for integration with learning management systems.

Author Keywords

LTI; database query processing; interactive learning environments; computer science education

INTRODUCTION

Database Query Analyzer (DBQA) [3] is a tool that illustrates the effects that clauses and conditions have on an SQL `SELECT` statement using a visualized, data-oriented approach. It does so in a step-by-step, procedural-like manner, illustrating the impacts that each clause and condition has on the intermediate datasets maintained by the DBMS. Mastery Grids [4], an open source learning platform, provides students with personalized content recommendations and exercises for a variety of computer science subjects, including SQL. One of these exercises, SQL query illustration, required updates, making integration with DBQA beneficial.

Since Mastery Grids already supports the LTI protocol [6] as a platform [5], this paper focuses on work that makes DBQA accessible for integration with other learning platforms through LTI. This aligns with the goals of the ACM ITiCSE working group [2] on integrating smart learning content using standardized protocols. This paper aims to identify current and future opportunities for integration and data sharing with DBQA.

DBQA OVERVIEW

Upon submission of an SQL `SELECT` statement, DBQA will: 1) highlight the currently evaluated clause or condition, 2) display the associated intermediate dataset, and 3) allow the user to move forward or backward in the query evaluation.

This step-by-step approach allows users to see how a dataset is initially obtained by the `FROM` clause and ultimately processed and filtered by the other clauses and conditions to obtain the final result set. As queries become more complex, particularly through the use of `JOIN`s and subqueries, DBQA frees users from the need to mentally manage intermediate datasets. This is important, as these types of queries have been shown to be among the most difficult types of SQL queries for students to construct [1].

The DBQA interface can be seen in Figure 1. If the user has access to multiple databases, one can be chosen using the provided list. Otherwise, DBQA will use its own example schema (as seen in Figure 1). Users can see all tables in the selected database along with their columns, primary keys, and foreign keys. Users also have the ability to view the table data within DBQA.

After a query is submitted, the currently evaluated clause or condition is highlighted in yellow, and previously evaluated clauses and conditions are highlighted in blue (if desired). At any given point, the displayed dataset is a result of executing the current clause or condition along with all previously executed clauses and conditions. This allows the user to easily see which clauses and conditions the current dataset is recognizing. DBQA generates each intermediate dataset by constructing a new `SELECT` query involving only the clauses and conditions evaluated to that point. If any subqueries are currently being analyzed, the corresponding datasets will be displayed in an additional table (as seen in Figure 1). The ability to step forward and backward gives users a simple way to see the effects that a clause or condition has on the query by observing the differences in the generated datasets.

DBQA is implemented as a web application written in Java using Spring, Maven, and Hibernate. It is currently configured to work with arbitrary MySQL, Oracle, and Derby databases. DBQA can be used to analyze `SELECT` statements involving any tables in the selected database schema.

LTI INTEGRATION

LTI [6] is a protocol developed by IMS Global Learning Consortium that provides secure communication between tools (e.g. DBQA) and platforms (e.g. Mastery Grids). DBQA was designed to require authentication, to allow users to reference web-accessible databases, and to allow users to submit arbitrary SQL `SELECT` statements. With the LTI support, the authentication requirement is fulfilled through the signed LTI launch request, and DBQA allows users to illustrate the execution of pre-defined queries on a default database. LTI users are not allowed to add new databases, and they cannot submit arbitrary queries. To ease the selection of predefined queries, DBQA allows platforms to select query exam-

Sixth SPLICE Workshop at L@S 2020 "Building an Infrastructure for Computer Science Education Research and Practice at Scale", SPLICE'20, August 12, 2020, Virtual Event

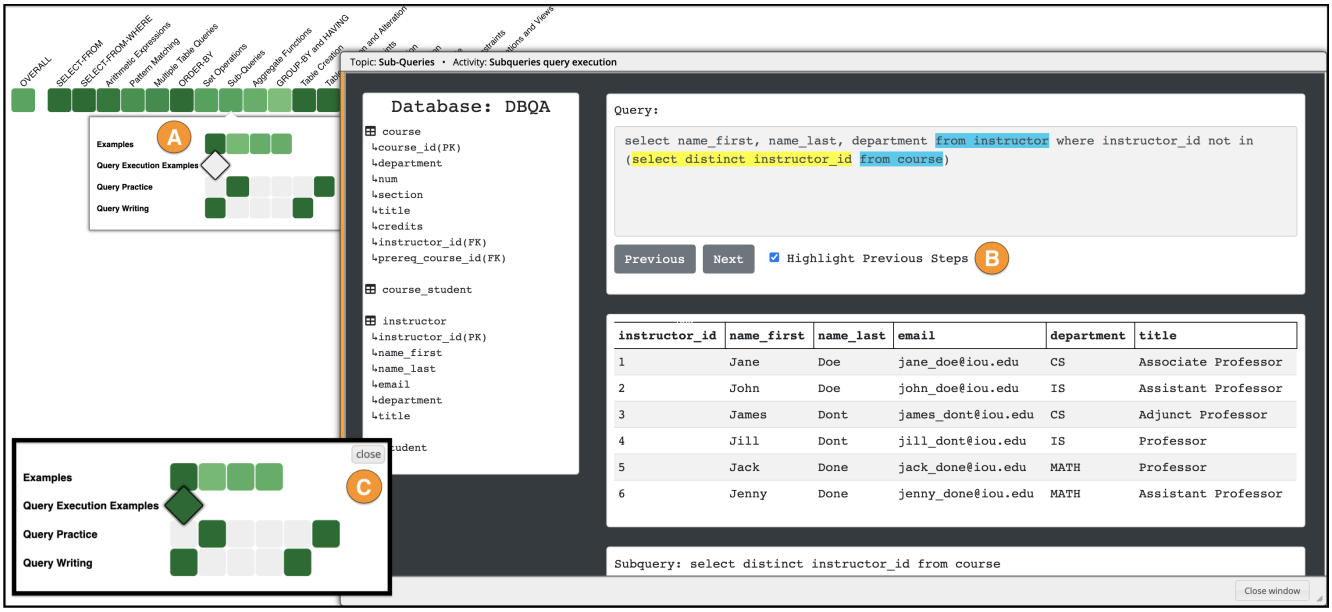

Figure 1. An instance of a DBQA query example in the "Sub-Queries" topic accessed from Mastery Grids using LTI protocol (A). The query steps are viewed using the navigational buttons (B). Mastery Grids updates the progress visualization after DBQA reports back the percentage of viewed steps using LTI outcome servive (C).

ples using Content-Item Message (renamed to Deep Linking). These query examples cover topics such as "SELECT-FROM", "Arithmetic Expressions", "SELECT-FROM-WHERE", "Pattern Matching", "Multiple Table Queries", "ORDER-BY", "Aggregate Functions", "GROUP-BY and HAVING", and "Sub-Queries". DBQA randomly selects a query to demonstrate from the requested topic. Then, DBQA reports back the percentage of viewed steps in the query to the platform as a grade using the LTI outcome service.

To demonstrate the interoperability features of DBQA, we integrated DBQA into Mastery Grids through LTI protocol. Using the LTI outcome service, Mastery Grids internally directs the reported grade result from DBQA to the student modeling service using a proprietary protocol. Following the redirection, Mastery Grids updates the progress visualization of the selected DBQA activity as shown in Figure 1. Currently, Mastery Grids only supports LTI launch requests as a consumer. Thus, LTI launch links to DBQA items are manually added to Mastery Grids without using the content discovery support of DBQA.

FUTURE WORK

DBQA is currently being updated to support the latest version of LTI (1.3). Future work will adopt new protocols to provide richer data-sharing to platforms. This means of interaction will allow platform users to submit arbitrary queries to DBQA. DBQA will report back to the platform with each query submitted by the user, as well as the user's steps through the visualization of the query execution.

REFERENCES

[1] Alireza Ahadi, Julia Prior, Vahid Behbood, and Raymond Lister. 2015. A Quantitative Study of the Relative Difficulty for Novices of Writing Seven Different Types of SQL Queries. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '15)*. ACM, New York, NY, USA, 201–206.

[2] Peter Brusilovsky, Stephen Edwards, Amruth Kumar, Lauri Malmi, Luciana Benotti, Duane Buck, Petri Ihantola, Rikki Prince, Teemu Sirkiä, Sergey Sosnovsky, Jaime Urquiza, Arto Vihavainen, and Michael Wollowski. 2014. Increasing Adoption of Smart Learning Content for Computer Science Education. In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference (ITiCSE-WGR '14)*. Association for Computing Machinery, New York, NY, USA, 31–57.

[3] Ryan Hardt and Esther Gutzmer. 2017. Database Query Analyzer (DBQA): A Data-Oriented SQL Clause Visualization Tool. In *Proceedings of the 18th Annual Conference on Information Technology Education (SIGITE '17)*. Association for Computing Machinery, New York, NY, USA, 147–152.

[4] Tomasz D Loboda, Julio Guerra, Roya Hosseini, and Peter Brusilovsky. 2014. Mastery Grids: An Open Source Social Educational Progress Visualization. In *European Conference on Technology Enhanced Learning*. Springer, 235–248.

[5] Hamza Manzoor, Kamil Akhuseyinoglu, Jackson Wonderly, Peter Brusilovsky, and Clifford A. Shaffer. 2019. Crossing the Borders: Re-Use of Smart Learning Objects in Advanced Content Access Systems. *Future Internet* 11, 7 (2019).

[6] Charles Severance, Ted Hanss, and Joseph Hardin. 2010. IMS Learning Tools Interoperability: Enabling a Mash-up Approach to Teaching and Learning Tools. *Technology, Instruction, Cognition & Learning* 7 (2010), 245–262.