DBSnap 2: New Features to Construct Database Queries by Snapping Blocks

Yasin N. Silva
Loyola University Chicago
ysilva1@luc.edu

Alexis Loza
Arizona State University
aloza9@asu.edu

Humberto Razente
Universidade Federal de Uberlândia
humberto.razente@ufu.br

ABSTRACT

Block-based environments for creating computer programs have become very useful learning tools in computer science as they enable focusing on the logic of a program rather than on its syntactical details. While most block-based environments support conventional (imperative) instructions, a few tools have been proposed to create database queries. One of these tools is DBSnap, a highly dynamic and open-source tool to create database query trees by dragging and connecting visual blocks representing datasets and database operators. In this paper, we introduce DBSnap 2, an extension of DBSnap that provides a set of improvements to facilitate the creation of simple and complex queries. The improvements include the support of database views (a key database concept), saving and importing queries, inserting, updating, and deleting data, the creation of charts, and various visual improvements. The demonstration of DBSnap 2 will show how the new features simplify the creation of queries and enable the graphical visualization of query results.

1 Introduction

Block-based tools for creating computer programs, e.g., Scratch [1], Blockly [2], and Snap! [3], have become very useful and popular learning tools in computer science. Their visual and dynamic features have facilitated, in fact, the exposure of programming concepts to younger learners. While most of these tools support conventional (imperative) languages, a few tools developed in recent years support database query languages, e.g., SQLSnap [4], DataSnap [5], Bags [6], and DBSnap [7, 8]. DBSnap builds on the open-source and web-based framework proposed in Snap! [3] and supports a new set of dataset and data-operator blocks to construct relational algebra query trees. In this paper, we present DBSnap 2, an extension of DBSnap that provides multiple new features to facilitate the construction of queries and the visualization of query results. The extensions include the support of (1) database views (which enables building more complex queries), (2) saving and importing queries (allowing learners to complete the specification of a query in multiple sessions), (3) inserting, updating, and deleting data (to test how the query result changes when the data changes), (4) the creation of charts, and (5) visual improvements such as a new toolbar, new result grids, and block color selections. The approaches to integrate DBSnap into database courses [7] are also applicable to DBSnap 2.

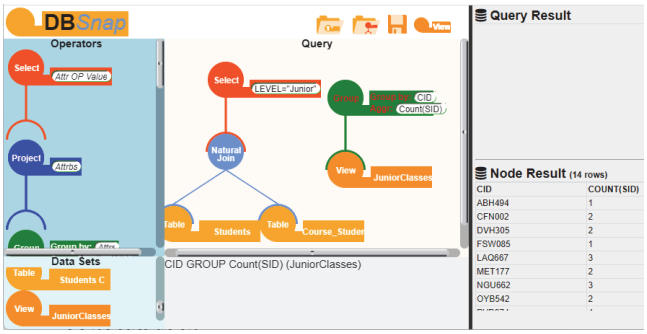

Figure 1: Creation of a View in DBSnap 2

2 New Features in DBSnap 2

Database views. Views are key database objects that represent a virtual dataset based on the result of a query. They have multiple applications in real scenarios, such as hiding certain parts of the data to certain users, simplifying the specification of complex queries, and capturing common intermediate results that are used to specify other queries. In DBSnap 2, the user can convert any query into a view by selecting the *Create View* option in the main application toolbar or in the context menu. As shown in Fig. 1, the created view is then added to the dataset panel as a new data block (views show the keyword *View* instead of *Table*)

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; Information systems education.

KEYWORDS

Relational algebra, SQL, block-based system, database queries

ACM Reference format:

Y. N. Silva, A. Loza, H. Razente. 2022. DBSnap 2: New Features to Construct Database Queries by Snapping Blocks. In *Proceedings of ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'22). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/3502717.3532156

ITiCSE 2022, July 8–13, 2022, Dublin, Ireland

ACM ISBN 978-1-4503-9200-6/22/07.
https://doi.org/10.1145/3502717.3532156

and can be used to construct multiple other queries. After the creation of a view block, the block behaves like any other dataset block and can be combined with any other dataset or operator blocks. The definition (query) of a view can be accessed in the bottom relational algebra panel by clicking on the view block. The left query in Fig 1 represents the query converted into a view (list of the courses taken by Junior students) while the right one uses the view (*JuniorClasses*) to identify the number of Junior students registered in every class.

Saving and importing queries. This feature of DBSnap 2 allows users to save a (complete or incomplete) query as a file and open it in another web session. This feature allows learners to complete a query after class or submit their queries to the instructor. These features can be accessed using the *Save Query* and *Import Query* in the main toolbar.

Inserting, updating, and deleting data. DBSnap 2 not only allows importing datasets but also modifying any (imported or built-in) dataset inside the tool. The available data manipulation operations can be performed directly in the node-result grid (bottom grid) after selecting a given dataset block. Users can add (clicking on the + symbol), update (directly editing the values in the grid) and remove (clicking on the trashcan icon) records. Fig. 2 shows the addition of a new Student record.

Creating charts. DBSnap 2 supports the creation of several types of charts including Area, Bar, Bubble, Column, Histogram, Line, Pie, and Scatter charts. Charts can be created for the output of any query or node by clicking the chart icon in the query or node result grids and selecting the chart settings. Fig. 3 shows a bar chart generated with the result of a query that computes the number of students registered in each class.

Visual improvements. DBSnap 2 includes multiple visual improvements such as a new toolbar that provides quick access to common operations: importing a dataset, importing a query, saving a query, and creating a view. The tool also includes new query and node result grids that are more compact and flexible. These grids allow the user to modify datasets and sort the output based on any attribute. The tool also includes some improvements on the selection of colors for selected blocks.

Tool Availability and Documentation. DBSnap 2 is now publicly available online [9] and can be used without the need of creating any accounts. The project website also includes an updated documentation page with sample datasets and descriptions of how to use the new features of the tool. Further updates of DBSnap are expected to be released in the future.

3 Demonstration

We plan to describe each of the new features of DBSnap 2 with a live demonstration of the tool. To this end, we will show how to (1) import datasets, (2) create simple and complex queries (e.g., queries using multiple operators and views), (3) edit the content of build-in and imported datasets, and (4) create multiple types of charts using the results of different queries. We will also describe our experience integrating this tool in an introductory database course.

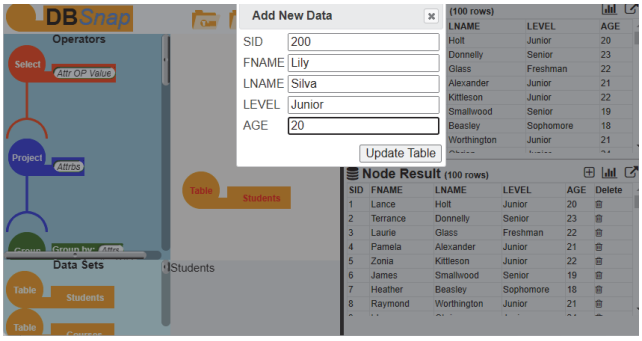

Figure 2: Adding a new dataset record in DBSnap 2

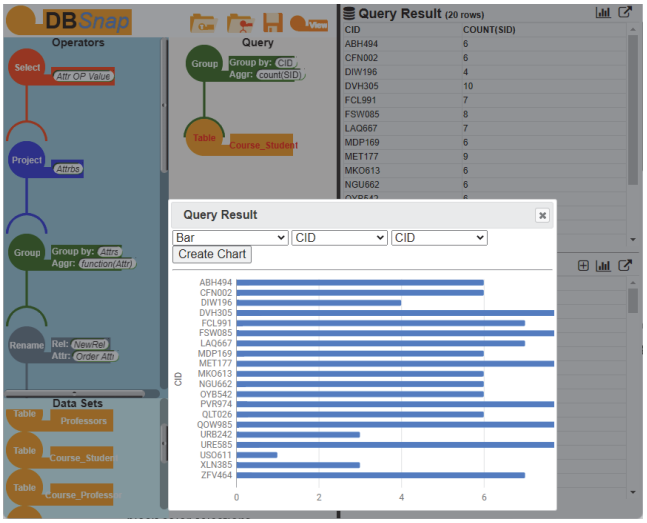

Figure 3: A chart created with the output of a query

4 Conclusion

In this paper, we present an extended version of the DBSnap tool. The main goal of the additions was to improve the process of creating queries and enable a better visualization of the results using more flexible grids and charts.

REFERENCES

[1] J. H. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk. Programming by choice: Urban youth learning programming with scratch. In ACM SIGCSE, 2008.
[2] A. Marron, G. Weiss, and G. Wiener. A decentralized approach for programming interactive applications with javascript and blockly. In AGERE!, 2012.
[3] C. North and B. Shneiderman. Snap-together visualization: Can users construct and operate coordinated visualizations? Int. J. Hum.-Comput. Stud., 53(5):715–739, 2000.
[4] Eckart Modrow. 2014. SQLsnap! http://snapextensions.uni-goettingen.de. (2014).
[5] Jonathon D. Hellmann. 2015. DataSnap: Enabling Domain Experts and Introductory Programmers to Process Big Data in a Block-Based Programming Language. Master's thesis. Virginia Tech, Virginia, USA.
[6] J. Gorman, S. Gsell, and C. Mayfield. Learning relational algebra by snapping blocks. In ACM SIGCSE, 2014.
[7] Y. N. Silva and J. Chon. 2015. DBSnap: Learning Database Queries by Snapping Blocks. In ACM SIGCSE.
[8] Y. N. Silva and J. Chon. 2015. Querying databases by snapping blocks. In IEEE ICDE.
[9] Y. N. Silva. 2022. DBSnap Project Website. Retrieved from https://ysilva.cs.luc.edu/dbsnap.