

Relational Algebra Teaching Support Tool

Jonathas Jivago de Almeida Cruz^{1*}, Kleber Kroll de Azevedo Silva²

¹ Federal Institute of Education, Science and Technology of Piauí, Campus Corrente, Piauí, BRAZIL

² Federal Institute of Education, Science and Technology of Rio Grande do Norte, Campus Ipanguaçu, Rio Grande do Norte, BRAZIL

*Corresponding Author: jonathas_jivago@hotmail.com

Citation: de Almeida Cruz, J.J. and de Azevedo Silva, K.K. (2017). Relational Algebra Teaching Support Tool. *Journal of Information Systems Engineering & Management*, 2(2), 8. doi: [10.20897/jisem.201708](https://doi.org/10.20897/jisem.201708)

Published: March 30, 2017

ABSTRACT

In recent years, there has been an increasing supply of digital, pedagogical tools, known as Digital Learning Objects (DLO) – digital resources (image, film, animation, etc.) and software developed specifically for educational purposes. In the area of Computer Science, teaching Databases present a particular challenge because of a lack of quality tools to work with Relational Algebra. The present study proposes a web-based tool to support teaching and learning Relational Algebra – an important subject that is particularly difficult for students to understand. The purpose of the proposed tool is to provide an alternative method for teaching Relational Algebra operations, such as: selection, projection, union, set difference, rename, intersection, Cartesian product, natural join, division and some aggregate functions. In addition, we propose a graphic definition of a database schema (using features such as drag and drop, column highlights, lines, fields, etc.), so students can use the tool easily, and in conjunction with the theory taught regarding the definition languages (DDL) and data manipulation (DML). We intend for this tool to serve as an appropriate means for teaching and learning Relational Algebra, contributing to the development of new teaching skills, as well motivating the students in the process of learning.

Keywords: relational algebra, digital learning object

INTRODUCTION

Relational Algebra is a fundamental part of the curriculum in the course on Technologies and Database Languages in Computer Science at the Federal Institute of Education, Science and Technology of Piauí, This language is essential for understanding database queries based on the relational data model for Structured Query Language (SQL).

Due to the lack of quality tools for working with Relational Algebra, teaching this language is often solely theoretical, which often makes learning difficult due to the high degree of abstraction required. Students' primary complaint is that current methods in teaching Relational Algebra do not give them the opportunity to visualize and verify the results of queries.

Thus, this work aims to develop a tool to support teaching Relational Algebra in a way that allows students to both visualize and verify the results of a query. The proposed tool intends to facilitate the understanding, and mainly the use, of concepts related to Relational Algebra, more specifically, providing:

- a) an intuitive graphical interface to create relationships to manipulate queries in Relational Algebra;
- b) strategies of how to deal with possible errors in algebraic expressions;
- c) visualization of the results of the queries performed.

The remainder of this article is organized as follows: Section 2 presents the theoretical foundation that addresses the main concepts related to this work; Section 3 presents the proposed tool; Section 4 a brief review of the literature in this field; finally, Section 5 reports the current state of development of the tool and the proposal of how the evaluation of the tool will be conducted.

THEORETICAL FRAMEWORK

In this digital age, with increasingly accelerated innovations in the Computer Sciences, teaching database languages presents a challenge and pedagogical practices must be as dynamic as the field itself. In this section, we present the theoretical framework on which our proposal for new pedagogical practices regarding the teaching/learning of query languages, specifically, Relational Algebra, is based.

Relational Algebra

A query language is a language that allows the user to request information from the database. Relational Algebra is a query language that has a set of operations that use one or two relations (tables) as input and produce a new relation as a result (Silberschatz, Korth and Sudarshan, 2006). Relational Algebra operations are divided into two groups: unary operations and binary operations. Selection, projection, and renaming are classified as unary operations because they use only one relation as input. Binary operations use two relations as inputs. Binary operations are union, difference, intersection, Cartesian product, natural junction and division (Elmasri and Navathe, 2011).

The operations selection, projection, union, difference, Cartesian product and renaming are considered fundamental. Intersection, natural join, division, and division are considered derivative operations because they can be obtained by combining fundamental operations.

The following subsections briefly describe the fundamental operations of Relational Algebra.

Selection

The selection operation, denoted by the Greek letter sigma (σ), returns as a result a new relation composed of the subset of tuples that satisfy a given predicate. A predicate may contain relational operators and can be combined with other predicates by forming a larger predicate through the use of logical connectives *and* (\wedge), *or* (\vee) and *not* (\neg).

Projection

The projection operation, denoted by the Greek letter pi (π), returns as a result a new relation containing a subset of the attributes of the relation passed as argument.

Union

The union operation, denoted by the binary operator \cup , returns as a result a new relation that contains all the tuples present in the first and second relation passed as argument. The tuples that have the same information are not repeated in the new relation. For the union to be realized, the relations passed as an argument must be compatible – they must have the same amount of attributes, and the attributes must have the same domain.

Set Difference

The difference operation is denoted by the binary operator $-$ and returns as a result a new relation that contains all the tuples present in the first relation, but that are not present in the second relation. To carry out the difference operation the same criteria of the union operation must be met.

Cartesian Product

The Cartesian product, denoted by the binary operator \times , is an operation that returns a new relation containing the combination of each tuple of the first relation with all the tuples of the second relation.

Rename

The renaming operation, denoted by the Greek letter rho (ρ), as its name suggests, facilitates the renaming of attributes and relations.

Digital Learning Object

According to Wiley (2000), a Digital Learning Object (DLO) is any digital resource that can be reused to assist the teaching/learning process. It is understood as a digital resource of any type of digital file (image, film, animation etc.) or software developed specifically for educational purposes.

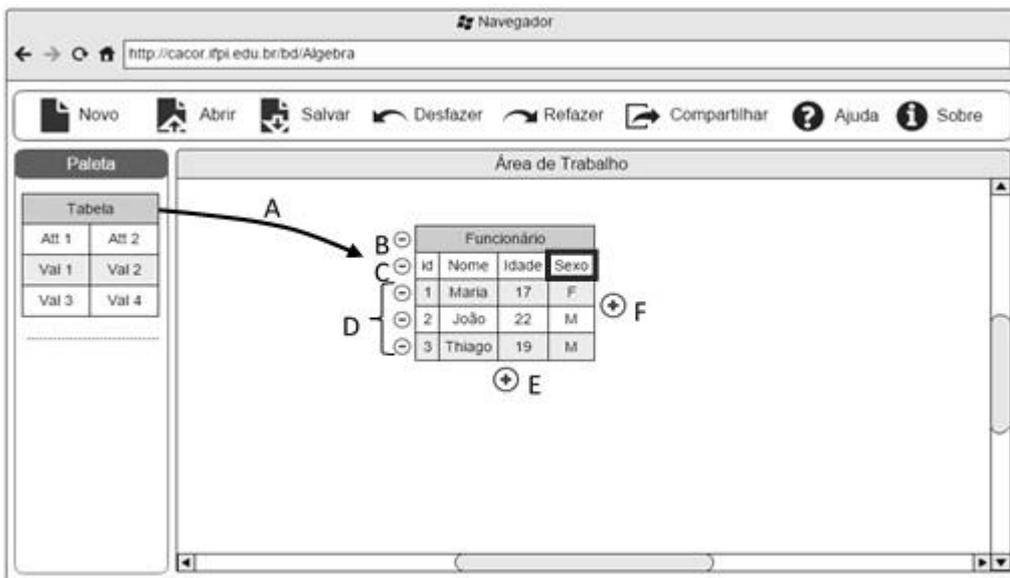


Figure 1. Module responsible for the management of the tables.

On the one hand, DLO helps build knowledge through the interaction of students with digital objects. In this way, students become active agents in the construction of their own knowledge. On the other hand, it is up to the teacher to create environments, situations and simulations aimed at teaching specific content, in addition to enabling the students to use digital objects. Usability in DLO is made possible by sharing digital objects, which can be stored in repositories.

In this work, we propose this tool as a digital learning object, which is readily available on the internet, so that it can help in the teaching/learning process of students and teachers at any educational institution, regarding contents related to Relational Algebra.

PROPOSED TOOL

This work proposes a user friendly tool that aims to facilitate the teaching/learning process of Relational Algebra. The main functionalities of this tool include: creating, changing and deleting tables, performing queries using the Relational Algebra operators and the ability to share projects between users.

The proposed tool will be executed in a web environment, and available as a DLO, thereby, making it available to anyone, at anytime, and anywhere there is access to the internet. This lets the user work with the tool without having to install and configure it.

Technologies used with this tool are HTML5 in the front end of the tool, which will be extensively used to provide the best user experience, as well as features such as drag and drop.

The tool has two modules: one responsible for managing the tables and the other responsible for the management of queries in Relational Algebra.

The module responsible for managing the tables is illustrated in [Figure 1](#). As the illustration shows, the user can either create a new project, open an existing project, or save the open project.

In this module the user can create tables by clicking and dragging the corresponding component to a table that is available in the component palette ([Figure 1-A](#)). To delete a table the user can click the symbol \ominus next to the table name ([Figure 1-B](#)). To delete an attribute you must click the attribute to be deleted and then click the symbol \ominus next to the attribute names ([Figure 1-C](#)). The deletion of a tuple is done by clicking on the symbol \ominus that is next to the tuple corresponding to the item to be excluded ([Figure 1-D](#)). To add a tuple, the user must click on the symbol \oplus , which is below the table ([Figure 1-E](#)). The addition of a new attribute is done by clicking on the symbol \oplus , which is next to the table ([Figure 1-F](#)).

[Figure 2](#) shows the screen for the query management module in Relational Algebra. In this figure a projection operation is shown to exemplify the use of the proposed tool. To perform a query, the user must choose the operation and drag it to the query area ([Figure 2-A](#)). After choosing the desired operation, the user can choose the relation that will be used in the query ([Figure 2-B](#)). After choosing the relation the user can highlight the attributes to be projected.

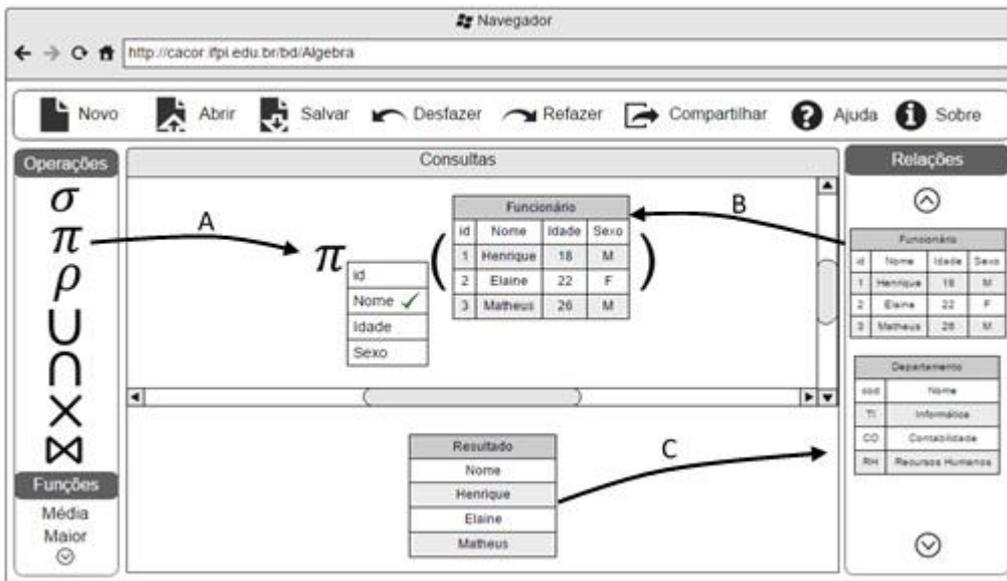


Figure 2. Module responsible for management of queries.

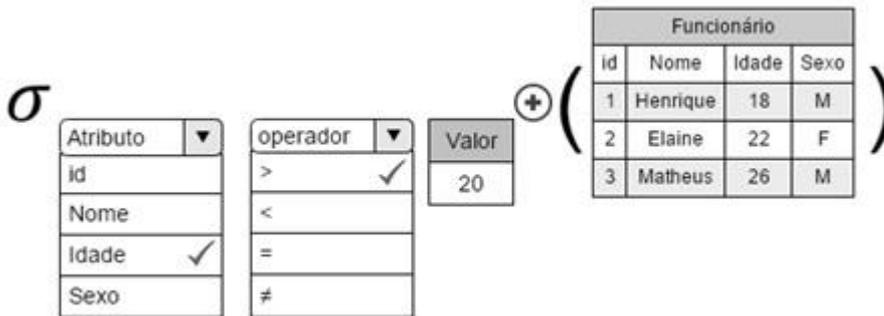


Figure 3. Query Example using the selection operator.



Figure 4. Example query with more than one condition in the predicate.

The result of the query performed is displayed at the bottom. This result can be dragged into the relations area to be used in another query later on. In this way, it is possible to carry out consultations with many step-by-step operations, facilitating the understanding of the students. After familiarizing themselves with Relational Algebra operations, students will then be able to perform nested queries directly in the area for queries.

In Figure 3 a query is shown using the selection operator. After choosing the selection operator (σ) and the choice of the Employee table, the user must specify the conditions for the query. This is done quite intuitively because the user chooses the attributes and operators with a few clicks. If you want to add more conditions, you can add them by clicking on the symbol \oplus next to the relationship name, as shown in Figure 4.

The presentation of the proposed tool clearly shows that it aims to be simple and intuitive, relieving users of the need to be concerned about syntax and notations.

RELATED WORKS

Several works were developed in order to make Relational Algebra teaching/learning more enjoyable, but these studies present several problems that hinder their adoption by the tech-education community.

The LEAP tool, proposed by Leyton and Hill (1995), allows the management of relational databases. It allows querying in Relational Algebra, as well as creating, deleting and updating tables. One of the weaknesses of this tool

is the realization of queries in a totally textual way. Besides being textual, the user must learn the specific commands of the tool, which differ from those found in reference books of the area. The tool proposed here aims to create, delete, update and query Relational Algebra in a graphical and intuitive way, making the learning process more pleasant.

DBTools 2000 software (DBTools, 2000), developed at Georgia Tech, enables SQL queries, as well as queries in Relational Algebra. The software has an intuitive interface, however, to be executed in the desktop environment, the user must perform tasks related to the installation and configuration of the tool for it to execute. In our model, the proposed tool is executed in a web environment, and there is no need for the user to perform installations and configurations.

In the work of Appel and Traina Jr. (2004) a tool is proposed, called *IDFQL* (Interactive Data Flow Query Language), for queries in Relational Algebra in a constructivist mode. Constructivism proposes the construction of knowledge through the interaction of the student with the computer. In this context, the teacher becomes a mediator who participates in this learning process. *IDFQL* uses graphical elements to represent the queries in order to support the teaching-learning process of Relational Algebra. Another feature of this tool is the construction of algebraic expressions through trees. Once again, this study, like the others, does not propose the *IDFQL* tool as a learning object made available on the web, which is proposed in the present work.

Lautert (2010) proposes the *SimAlg* tool. This tool simulates queries in Relational Algebra. However, the main drawback of *SimAlg* is that it restricts the user when managing tables, limiting them to use only the tables already created and pre-established queries. In the tool proposed in this work, the user will be able to create, change and delete tables, besides giving them flexibility in relation to queries.

In the work of Prates et al. (2013), the *ProgramAR* tool is proposed. The *ProgramAR* aims to automate the process of visualizing the results of queries in Relational Algebra. The queries in this tool are carried out in a textual way, different from the tool proposed here, which allows the construction of queries in a totally graphical way.

In Bilecki and Kalempa (2015) a tool called *EasyRA* is proposed to convert queries into Relational Algebra in SQL queries, which are executed in DBMSs. In this way, the user can view the results of queries performed in Relational Algebra. The main divergence between *EasyRA* and the tool proposed here is that *EasyRA* is not available in the web environment, requiring installation and configuration.

CURRENT STATUS OF WORK AND PROPOSAL FOR THE EVALUATION OF RESULTS

The proposed tool has two main modules: a module for data definition and a query manager module in Relational Algebra. Clearly, the tool is in an early stage of development. Nonetheless, we have already developed an overview of the tool through the elaboration of screen prototypes and functionalities. In addition, we have also defined the technologies to be used throughout the development process.

We intend to evaluate the idea proposed in this work through the use of the tool in the Technologies and Languages for Database classes at the Federal Institute of Education, Science and Technology of Piauí - Campus Corrente. For this evaluation, questionnaires will be applied to student research participants regarding the experience of using the tool in these classes. Rates of their performance through practical tests will also be measured. This data will be correlated with a control group of students classes that did not use the tool.

REFERENCES

- Appel, A.P. and Traina Jr., C. (2004). *iDFQL - Uma Ferramenta de Apoio ao Processo de Ensino-Aprendizagem da Álgebra Relacional Baseado no Construcionismo*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação - USP. São Carlos, SP.
- Bilecki, L.F. and Kalempa, V.C. (2015). *EasyRA - Uma ferramenta para tradução de consultas em álgebra relacional para SQL*. Universidade do Estado de Santa Catarina – UDESC. São Bento do Sul, SC.
- DBTools 2000. Available at: <http://www.cc.gatech.edu/computing/Database/dbTools>. (Accessed 26 August 2015).
- Elmasri, R. and Navathe, S.B. (2011). *Sistemas de Banco de Dados*. São Paulo: Editora Campus.
- Lautert, L.A. (2010). *Implementação de um simulador de consultas em álgebra Relacional*. Trabalho de Graduação, Universidade Federal de Santa Maria. Santa Maria, RS.
- Leyton, R. and Hill, C. *Leap*. Available at: <http://leap.sourceforge.net>. (Accessed 26 August 2015).
- Prates, A., Claro, D.B., Proencia, E. and Queiroz, J. (2013). *ProgramAR - Ferramenta para auxiliar o ensino em Álgebra Relacional*. Instituto de Matemática (IM) – Universidade Federal da Bahia-UFBA. Salvador, BA.
- Silberschatz, A., Korth, H. and Sudarshan, S. (2006). *Sistema de Banco de Dados*. 5.ed. Rio de Janeiro: Editora Campus.

Wiley, D.A. *Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy*, In D.A. Wiley (Ed.), *The instructional use of learning objects* (2000). Available at: <http://www.reusability.org/read/chapters/wiley.doc> (Accessed 26 August 2015).