Educational Database Prototype: the Simplest of All

Takashi Matsuzawa
University of Wisconsin-Madison
Madison, Wisconsin
takashi@cs.wisc.edu

Yi Lyu
University of Wisconsin-Madison
Madison, Wisconsin
ylyu76@wisc.edu

Yiyin Shen
University of Wisconsin-Madison
Madison, Wisconsin
yshen82@wisc.edu

ABSTRACT

PVLDB Reference Format:
Takashi Matsuzawa, Yi Lyu, and Yiyin Shen. Educational Database Prototype: the Simplest of All.

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/yiyins2/Educational-Database-Prototype.

1 INTRODUCTION

1.1 Motivation

Database Management System (DBMS) is designed to help store and process large collections of data. Unlike many other Computer Science and Electrical Engineering related fields (e.g., Wireless Networking), DBMS permits researchers and developers to perform a wide range of optimizations while an appropriate high-level interface is presented. While most researches and projects have been working on optimizations for complex commercial databases, we focus on presenting a simple and lightweight database system structure for educational purposes instead.

1.1.1 Constructivist-based Learning. Traditional instructional methods for DBMS courses consist of didactic lectures and a fixed set of assignments. These passive learning methods discourage students to realize that usually there is no definite implementation of DBMS. Constructivism emphasizes on an active learning process in which students construct concepts based on existing foundation to gain new knowledge [2]. Many constructivist-based learning methodologies are proposed, such as problem-based learning and inquiry-based-learning. For this project, we want to focus on project-based learning (PBL), in which learners acquire in-depth knowledge through exploring a real-world problem or challenge, ideally out of personal interest. Several studies showed positive feedback of applying PBL in database courses from both the students and the faculties. Connolly's and Begg's study [2] showed that students reported that PBL is more motivating and engaging than traditional learning. Faculty felt students acquired more knowledge, especially skills not traditionally covered in database modules. This study also suggests that one integrative project covering several modules would be a powerful instructional approach. Domínguez and Jaime [3] conducted an experiment to test the effectiveness of PBL: students could choose voluntarily to either follow PBL or traditional learning, then the performance (measured by exam dropout rates, exam passing rates, exam marks, and class attendance) of these two student groups were compared. Students who followed PBL obtained better performance and performance was not mediated by excellent students' preference to follow PBL rather than traditional learning. Naik and Gajjar [7] reported that satisfaction level of both students and faculties are over 90% for a DBMS class utilizing PBL. Faculties indicated that except from learning the core concepts, students improved their problem-solving skills, confidence, team management, and communication skills. We would like to borrow this PBL ideology and implement it in our undergraduate DB course in University of Wisconsin Madison (CS 564).

1.1.2 Limitations of Current Curriculum. Besides adoption of PBL, improvements to current curriculum of DBMS courses are necessary. Saeed [9] showed that current curriculum in most universities fails to provide undergraduate students with the skill to solve poor DB performance in their future careers, and one of the primary reasons is the lack of lower-level practices. To counter this, current CS 564 involves implementing specific modules of DB architecture (i.e., B+ Tree), but many of such assignments turned into busy work for students to deal with numerous corner cases. Consequently, students cannot grasp a more comprehensive picture of DB internal design. Sciore [10] proposed that instructor could provide students with the source code of pedagogically-written DB system and have students modify specific modules. Students can then learn the overall DB structure by studying the source code. In addition, this pedagogically-written DB system should be simple enough to avoid a steep learning curve for students. Its simplicity also allows plenty of space for students to freely modify the code or add new implementations.

1.1.3 Goal. This project tries to implement an educational DB prototype which consists the interface that only allows the basic SQL transactions with the minimum amount of coding possible. The objectives of the prototype are the following:

(1) It can provide a clean and concise overview of the database system, and convey the fact that database system can be extremely simple;
(2) It provides a platform that students are able to perform any optimization they have learned throughout the semester (in class or out of class);

This project will also develop an experimental integrative final project, in which the educational DB prototype serves as a skeleton. Students will be asked to improve the efficiency of the prototype by implementing optimizations. They will also participate in a competition, where they are ranked based on the efficiency of the improved prototype against a benchmark.

ISSN 2150-8097.

1.2 Related Work

DBMS related tools for education purposes have been introduced early since two decades ago, where Ramakrishnan et al. [8] proposed a web-based administration tool for projects that may be assigned to students. However, the system itself is mutable, hence students were only able to view and modify on a graphic user interface. Guting et al. [4] proposed a "generic" database system frame for teaching and researching, yet it has similar limitations.

More recently, many systems have been created for DBMS pedagogical purposes, including:

- SimpleDB [6] from MIT as course project during Fall 2010
- SimpleDB [10] by Sciore from Boston College, written in Java, first proposed in 2007 and latest updated on March 25, 2021
- BadgerDB [5] from UW-Madison as Database API

Apart from educational DB, tinyDB [1], an open source and lightweight database system written in Python, might also be helpful for retrieving a general idea during development processes.

2 METHODOLOGY

2.1 Task Plan

As mentioned in the Introduction section, we plan to implement an educational DB prototype that allows basic nontrivial SQL transactions and produce corresponding results in sequential orders, using the minimum amount of codes possible. We want to reflect the three-tier architecture of DB with a stress on the application and data tier. Specific domain of basic functionalities are

- Query Processor (query parser, query optimizer, plan executor, operator evaluator)
- Transactional Manager (transaction manager, lock manager, log manager)
- Storage Manager (access methods, buffer manager, file manager)
- Disk Storage (data files, index files)

The above functionalities are subject to changes, depending how much liberty we decide to give to students. Each functionalities will be split among each team member.

For the integrative final project, Takashi and Yiyin will design a set of grading rubrics to measure improvement of efficiency and implement such grading mechanism. Yi will develop a playground website and maintain a server for hosting students' competitions and leader board.

Documentation and bug hunting will be carried out throughout the project. The program will be written in C++.

2.2 Timeline

Date	Goal
Oct 25	Proposal due
Nov 13	Each team member finish their assigned functionalities
Nov 19	Combine all functionalities together
Nov 22	Prototype is able to process basic SQL transactions
Nov 24	Project meeting
Dec 4	Finish grading rubric and playground website for final project
Dec 6	Link grading rubric and playground website
Dec 11	Write sample final project to test on website
Dec 13	Presentation
Dec 18	Report due

REFERENCES

[1] 2021. tinyDB. https://tinydb.readthedocs.io/en/latest/

[2] Thomas M Connolly and Carolyn E Begg. 2006. A Constructivist-Based Approach to Teaching Database Analysis and Design. *Journal of Information Systems Education* 17, 1 (2006).

[3] César Domínguez and Arturo Jaime. 2010. Database design learning: A project-based approach organized through a course management system. *Computers & Education* 55, 3 (2010), 1312–1320.

[4] R.H. Guting, V. Almeida, D. Ansorge, T. Behr, Z. Ding, T. Hose, F. Hoffmann, M. Spiekermann, and U. Telle. 2005. SECONDO: an extensible DBMS platform for research prototyping and teaching. In *21st International Conference on Data Engineering (ICDE'05)*. 1115–1116. https://doi.org/10.1109/ICDE.2005.129

[5] Haiyun Jin. [n.d.]. BadgerDB. http://pages.cs.wisc.edu/~jignesh/cs564/projects/BadgerDB/BufMgr/docs/

[6] Samuel Madden, Robert Morris, Michael Stonebraker, and Carlo Curino. 2020. 6.830 Database Systems. Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare https://ocw.mit.edu. License: Creative Commons BY-NC-SA. (2020).

[7] Shefali Naik and Kunjal Gajjar. 2021. Applying and Evaluating Engagement and Application-Based Learning and Education (ENABLE): A Student-Centered Learning Pedagogy for the Course Database Management System. *Journal of Education* (2021), 00220574211032319.

[8] Sub Ramakrishnan and Emeka Nwosu. 2003. DBMS Course: Web Based Database Administration Tool and Class Projects. *SIGCSE Bull.* 35, 1 (Jan. 2003), 16–20. https://doi.org/10.1145/792548.611922

[9] Adham Mohsin Saeed. 2017. Role of database management systems (dbms) in supporting information technology in sector of education. *Int. J. Sci. Res* 6, 5 (2017), 1462–1466.

[10] Edward Sciore. 2007. SimpleDB: A Simple Java-Based Multiuser Syst for Teaching Database Internals. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (Covington, Kentucky, USA) *(SIGCSE '07)*. Association for Computing Machinery, New York, NY, USA, 561–565. https://doi.org/10.1145/1227310.1227498