

eledSQL – A New Web-Based Learning Environment for Teaching Databases and SQL at Secondary School Level

Andreas Grillenberger
Didactics of Informatics
University of Erlangen-Nuremberg
Martensstr. 3
91058 Erlangen, Germany
mail@andreas-grillenberger.de

Torsten Brinda
Didactics of Informatics
University of Duisburg-Essen
Schützenbahn 70
45127 Essen, Germany
torsten.brinda@uni-due.de

ABSTRACT

Data modeling using databases and SQL is a fundamental part of the curriculum of secondary computing education in Germany. Professional database tools like *HeidiSQL*, *phpMyAdmin* or *Microsoft Access* are often used in class as a “learning software”, although these tools have been developed for managing complex databases, often for companies, and not for educational purposes. Such tools offer a wide range of functions of which only a small part is required by secondary computing education. At the beginning of such instruction, students can hardly work independently with these programs, as they do not know the database language SQL by then. This often leads to theory-loaded introductory phases of such classes or alternatively to a usage of such tools like spreadsheet programs. To address this problem, a new web-based learning environment for databases and SQL (named *eledSQL*), also suitable for mobile devices and only with the functionality needed for secondary computing education, was developed. The basic idea was to initially allow students to make database queries using natural language and then gradually introduce them to the use of SQL. Starting with a problem analysis and a discussion of related work in the field of teaching databases and SQL, in this paper the conception of *eledSQL*, its implementation and first experiences with its practical use are described.

Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education—*Computer science education*

General Terms

Design, Human Factors, Languages

Keywords

Databases, SQL, Secondary Education, Learning Environment, Web-based Learning, Mobile Learning, Tools

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WiPSCE '12, November 8–9, 2012, Hamburg, Germany
Copyright 2012 ACM 978-1-4503-1787-0/11/12 ...\$15.00.

1. MOTIVATION

In secondary computing education in Germany data modeling using databases and SQL is a fundamental part of the curriculum [1]. As “learning software” professional database tools such as *HeidiSQL*¹, *phpMyAdmin*² or *Microsoft Access* are used in class – tools, which were not designed for educational purposes but for the professional management of databases. In supplementary material to the Bavarian computer science high school curriculum [16], for example, either the use of the *XAMPP* package³ (with web server, *MySQL* database server, *phpMyAdmin*) or of *Microsoft Access* is recommended by the authors. These professional tools have in common that on the one hand they offer a wide range of functions, of which only a small part is needed for secondary computing education, and on the other they require previous knowledge about databases and the SQL language for its competent use. The use of such programs in introductory lessons on databases is therefore hardly possible. As a consequence, often either theory-loaded introduction phases can be observed in class or the database software is used like a spreadsheet program there. The second option entails the risk that the students – because of the similarities of the user interfaces of for example *Microsoft Excel* and the table view of *Microsoft Access* – which supports their orientation in the beginning – often have difficulties in recognizing the differences between the two system categories and do not or not fully understand the difference between a spreadsheet and a database. Other problems arise, when the learners find out how to use the QBE function (query by example) within *Microsoft Access*, where the underlying SQL code is generated automatically [5].

The use of *XAMPP* has the advantage that with the appropriate user rights configuration the learners can experience the advantages of multi-user access to a database. With *Microsoft Access* this is potentially also possible but considerably more complicated. Another advantage of the client-server architecture is that the learners can access the database from outside the school (e.g. to do homework), if the database server is accessible on the Internet at all. However, secondary school servers are often configured very restrictively and often the installation of a database server is not allowed there for security reasons. In this case, client and server are sometimes installed in the computer room(s) of the school on every single workstation, which is very la-

¹www.heidisql.com (Jun 20th, 2012)

²www.phpmyadmin.net (Jun 20th, 2012)

³www.apachefriends.org/en/xampp.html (Jun 20th, 2012)

borious, contrary to the client-server idea and excludes the experience of multi-user access.

Summing up, although these systems cover all concepts to be taught in class by corresponding functions, they appear, mainly because of the required prior knowledge and the variety of functions contained in them, not very recommendable for a use in introduction phases into databases and SQL in computer science classes in secondary schools. So the question is: how should a system better suitable for this purpose be designed?

2. TEACHWARE FOR DATABASES / SQL

As a next step published research on learning environments in the area of databases and SQL was analyzed. In the literature a number of such environments are described which, however, refer mostly to higher education. Most of these environments can be categorized as systems, which visualize interactive SQL examples using multimedia or those that provide SQL problems and evaluate student solutions [2].

The *first category* includes, e.g. the desktop application *eSQL* [10], which visualizes stepwise the selection of output data instead of the presentation of a result for given SQL queries, the *QueryViz* system [6] with a special focus on the visualization of nested queries, the web-based *SAVI* system (SQL Advanced Visualization [4]), which allows navigation of the animations in both directions, and *ADbC* (Animated Database Courseware) [13], which provides a large number of interactive components for the visualization of concepts of the fields of databases and database security. The *ADbC* collection is extensive, sees itself as a blended learning material and provides interactive visualizations of e.g. various notations in the area of entity relationship (ER) modeling or of prepared questions, e.g. the transfer of given ER diagrams into tables with a visualization of the consequences of incorrect answers. In addition, visualizations of the execution of predefined SQL queries are provided. Other systems focus on the visualization of specific aspects such as on the process of ER modeling [9] or the normalization of a database [11].

The *second category* includes systems such as *SQLTutor* [12] and *SQLator* [15]. The learners are confronted with problems of everyday life and must create respective SQL queries to solve these problems. The solutions are then analyzed by these systems using heuristics, correct solutions are presented and any errors explained as far as possible. Elements of both categories can be found in web-based systems such as *SQLzoo.net* and *Teradata SQL Assistant*, which provide a set of predefined and not modifiable tables and queries plus a detailed feedback upon execution [5].

The most far-reaching integrated approach finally comes from Brusilovsky et al. [2, 3], who combined interactive examples, an SQL knowledge tester, adaptivity and SQL problems in a self-developed LMS using an underlying SQL ontology.

All these systems are pure learning environments. Some include hard-wired examples that are not changeable, others focus only on parts of the school-relevant content but all emphasize the explanation of concepts and neglect productive work. A system suitable for secondary schools should combine learning and working. This also requires an adjustment to the special needs of schools with teachers, classes and students.

3. REQUIREMENTS ANALYSIS

Based on the results of the analysis of database tools and learning environments, we propose a new system better suited to the needs of introductory school classes. Such a system should be accessible with any web browser over the Internet and offer software-ergonomic user interfaces for both desktop pcs and mobile devices of all kinds in accordance with the currently changing media use in schools [14]. The system's user interface should be easily adaptable to the corporate design of an educational institution. Installation on a school server or an external web space (in the case of school safety concerns) should be as simple as possible in order to keep the resulting workload for teachers low. A common weakness of previously published learning environments (see sect. 2) is that they provide many different useful visualizations and problems but that their integration into basic learning management system functionality adapted to the school needs is missing. The roles and user rights concept to be implemented allows differentiated functions and rights for teachers, school classes, learners and groups of learners as well as an administrator. This provides the possibility of adapting such a system to meet the specific needs of different computer science courses of a school and its students and to enable their simultaneous work with the system. Concerning content, the students should be introduced to working with databases and SQL with the system. This results in the requirement that the system must be customizable to the learning progress of learners and groups of learners. This could be realized by three different modes to interact with databases. At the lowest level the formulation of queries is partly based on forms and partly on natural language. Prior SQL knowledge is not required there. On the next higher level queries can be entered as on the lowest level, in addition, however, the underlying SQL syntax appears automatically (see fig. 1). On the third level the interaction with the database should be only possible with SQL. The system should provide only the functionality needed in class and try not to compete with professional database tools. The implemented functionality should cover typical secondary schools' database curricula like the Bavarian one [16].

Users of the role of *teachers* should be able to create student accounts or import accounts from a file and manage their user rights. Students should be assignable to a class or a course by them. Furthermore, groups of students for group work should be definable. Another point of criticism of existing learning environments was a lack of adaptability to the specific target group (see sect. 2). The teacher should be able to set the query mode (see above) for entire classes. Many learning environments provide predefined tables and queries, which can not be changed and adapted to a specific course. Therefore, a teacher should be able to create own table templates, import such templates from files in CSV format and share them with classes, groups or learners as a copy. By this, tables can easily be recovered when needed again. To focus the educational processes on the essential, SQL functions, such as **select**, **update**, **insert**, **delete**, **show** (or their natural language representation), should be activatable and also deactivatable for specific tables. At various locations in the system, the teacher should be able to adjust teaching texts and provide stepwise assistance for the SQL queries. A teacher should finally be able to switch to the perspective of any learner and use all his or her features. A user in the *administrator* should additionally be able to

create and manage teacher accounts.

Users in the role of *students* should be able to perform all activated operations on the tables provided by the teacher. As the students work only on table copies all activated functions can be executed without any risk of destruction. This also supports exploratory learning. There is also the possibility that students, groups or classes can create, edit and delete tables together, if this feature has been enabled by the teacher. The aim is to make the learners experience the multi-user mode of a database. The tables provided by the teacher can only be deleted though, if this has explicitly been approved by him or her. Finally, the learners should be able to see only those materials provided to them, their group or class and not those provided to any others.

4. DESIGN AND IMPLEMENTATION

The development process of the specified system, the *Erlangen learning environment for databases and SQL – eledSQL*, with requirements analysis, use cases, database design, class diagrams and implementation of the software is described in detail in [7]. In the following, we sketch only some important aspects. Due to the requirements that the system should be web-based and that the installation efforts should be kept low for teachers, the decision for a server-based system with browser-based access was made. To meet school conditions, it was taken into account that on the one hand, the components needed on an internal school server to install *eledSQL* are standard components free of charge and that the software can also be run via an external web hoster, if the school safety conception is contrary to the installation on site. External web hosters usually offer pre-structured packages where, for example, usable programming languages and databases are predefined. Because the language PHP and MySQL databases are considerably more widespread than their alternatives, both have been chosen for the development of *eledSQL*. PHP is very well suited for the development of web applications and also supports object-oriented software development, so that the system was developed in an object-oriented way because of the resulting structuring benefits, where possible and appropriate. Conceptually, the software was structured in three parts following the model-view-controller pattern. The view contains templates that correspond to the user interface (see fig. 1) and can be

easily adapted to the corporate design of a school without deep PHP knowledge. In order to support different language versions of the software, all texts of the user interface have been moved to a language file. The model contains functional classes that generate content based on user input and database access. The controller mediates between these levels (see fig. 2). In order to use potentially other

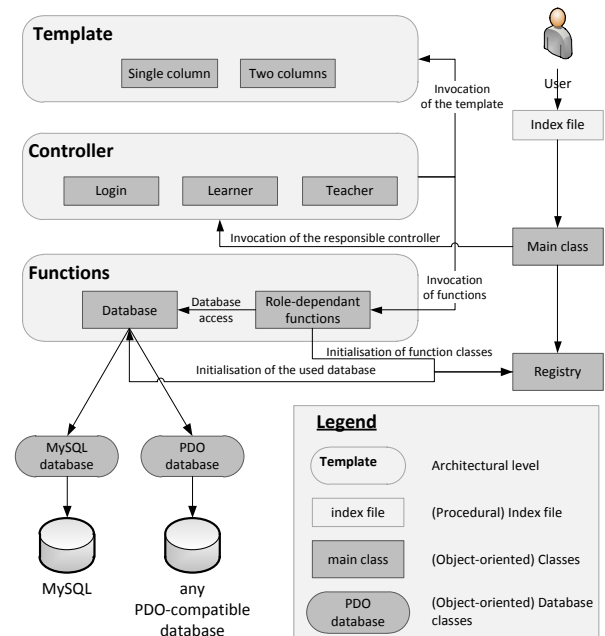


Figure 2: Architecture of the *eledSQL* system

databases like *PostgreSQL* an additional abstraction layer from the database queries using PHP Data Objects (PDO) was integrated into the system. This makes the system even more flexible. To implement the security requirements first separate databases for system data, classes, groups and individuals were considered. Since external web hosters typically limit the number of databases this option was dismissed. For this reason, the system uses a single database in which the table names reflect the intended addressees. System tables (marked with “sys”, such as user accounts) are processable only by means of the management function built-in into *eledSQL* or an external program. Template tables (marked with “tpl”) can be created, filled with data and then be made available to the learners as a copy by a teacher. Tables marked with an identifier of the respective class, group or learner are available only for the referred learners. Before executing an SQL statement it can be easily verified by checking the label of each involved table whether the permission to do so is given. Furthermore, it is easy to check whether a student is allowed to perform the respective functions on the tables involved. Also, SQL injections are therewith preventable.

The web-based user interface of the system of *eledSQL* was tested with major browsers of the operating systems Windows, Linux and MacOS. Furthermore, its usability was checked with some probands. Identified problems were eliminated.

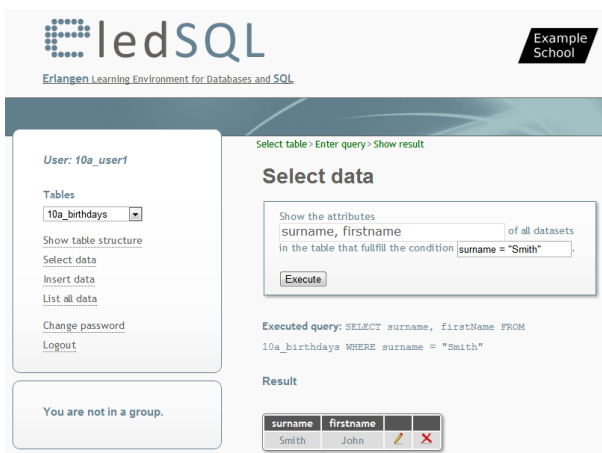


Figure 1: User interface of *eledSQL* system

5. FIRST CLASSROOM EXPERIENCES

An early version of *eledSQL* was used within a teaching internship of the first author at a secondary school in Erlangen (Germany), where a 9th grade class (learners 14 to 15 years old) was introduced into the field of databases. Before the teaching internship the class had learned to deal with spreadsheets. The experiences within this course were throughout positive – the learners could use the tool without a detailed introduction into the software itself. Thereby they were able to discover the differences between the well-known spreadsheets and the new databases by dealing with exercises prepared by the teacher. For example, they were to find the address of a person in a spreadsheet and a database with about 1.000 entries each or all persons with names including a defined string. While dealing with these exercises, they understood the different fields of application of databases and spreadsheets. This is a huge advantage over the conventional way of teaching – which is in many cases to tell the differences to the pupils, because they cannot discover them themselves due to missing SQL knowledge. This sometimes results in motivational problems, especially when the pupils cannot realize why “the same application with another name” should have these pros and cons, when only using the tabular view at the beginning.

6. SUMMARY AND OUTLOOK

In this paper we motivated the need for a new learning environment on databases and SQL for the computing education at secondary level. We analyzed database software and learning environments used in education and derived from the analysis results and curricular needs requirements for the new learning environment, whose conception, design, implementation and first experiences of the educational use we described afterwards. The current version of *eledSQL* is available for free (Open Source License) at Sourceforge⁴. The development of this software is part of a bigger student project that is continued as part of another bachelor thesis, where a detailed analysis of the classroom use in two parallel classes is made. A further development of the system is also planned. In a next release assessments and teaching material should be integrated.

7. REFERENCES

- [1] T. Brinda, H. Puhlmann, and C. Schulte. Bridging ICT and CS: educational standards for computer science in lower secondary education. In *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education*, ITiCSE '09, pages 288–292, New York, NY, 2009. ACM.
- [2] P. Brusilovsky, S. Sosnovsky, D. H. Lee, M. Yudelson, V. Zadorozhny, and X. Zhou. An open integrated exploratorium for database courses. In *Proceedings of the 13th annual conference on Innovation and technology in computer science education*, ITiCSE '08, pages 22–26, New York, NY, 2008. ACM.
- [3] P. Brusilovsky, S. Sosnovsky, M. V. Yudelson, D. H. Lee, V. Zadorozhny, and X. Zhou. Learning SQL programming with interactive tools: From integration to personalization. *Transactions on Computing Education*, 9(4):19:1–19:15, Jan. 2010.
- [4] M. Cembalo, A. De Santis, and U. Ferraro Petrillo. SAVI: a new system for advanced SQL visualization. In *Proceedings of the 2011 conference on Information technology education*, SIGITE '11, pages 165–170, New York, NY, 2011. ACM.
- [5] J. Cigas and B. Kushan. Experiences with online SQL environments. *Journal of Computing Sciences in Colleges*, 25(5):251–257, May 2010.
- [6] J. Danaparamita and W. Gatterbauer. Queryviz: helping users understand SQL queries and their patterns. In *Proceedings of the 14th International Conference on Extending Database Technology*, EDBT/ICDT '11, pages 558–561, New York, NY, 2011. ACM.
- [7] A. Grillenberger. Entwurf und Implementierung einer Lernumgebung für den Informatikunterricht zur Unterrichtssequenz Datenbanksysteme (in German). Bachelor's thesis, University of Erlangen-Nuremberg, Didactics of Informatics, Erlangen, Germany, 2012.
- [8] M. Guimaraes and M. Murray. Using animation courseware in the teaching of database security. In *Proceedings of the 8th ACM SIGITE conference on Information technology education*, SIGITE '07, pages 253–258, New York, NY, USA, 2007. ACM.
- [9] L. Hall and A. Gordon. A virtual learning environment for entity relationship modelling. In *Proceedings of the 29th ACM SIGCSE technical symposium on Computer science education*, SIGCSE '98, pages 345–349, New York, NY, 1998. ACM.
- [10] R. Kearns, S. Shead, and A. Fekete. A teaching system for SQL. In *Proceedings of the 2nd Australasian conference on Computer science education*, ACSE '97, pages 224–231, New York, NY, 1996. ACM.
- [11] H.-J. Kung and H.-L. Tung. A web-based tool for teaching data modeling. *Journal of Computing Sciences in Colleges*, 26(2):231–237, Dec. 2010.
- [12] A. Mitrovic. Learning sql with a computerized tutor. In *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, SIGCSE '98, pages 307–311, New York, NY, 1998. ACM.
- [13] M. Murray and M. Guimaraes. Animated database courseware: using animations to extend conceptual understanding of database concepts. *Journal of Computing Sciences in Colleges*, 24(2):144–150, Dec. 2008.
- [14] C. Piech and E. Roberts. Informatics education using nothing but a browser. *Proceedings of the IFIP Conference on ICT and Informatics in a Globalised World of Education, Mombasa, Kenya, August 2011*, 2011. <http://cs.stanford.edu/~eroberts/papers/NothingButABrowser.pdf>.
- [15] S. Sadiq, M. Orłowska, W. Sadiq, and J. Lin. SQLator: an online SQL learning workbench. In *Proceedings of the 9th annual ACM SIGCSE conference on Innovation and technology in computer science education*, ITiCSE '04, pages 223–227, New York, NY, 2004. ACM.
- [16] Staatsinstitut für Schulqualität und Bildungsforschung. Informatik am Naturwissenschaftlich-technologischen Gymnasium, Jahrgangsstufe 9 (Handreichung, in German), 2004.

⁴<http://eledsql.sourceforge.net/>