Abstract

The goal of this paper is to describe the development of animations and other courseware to support database curricula, as well as to elicit ideas and suggestions from other educators in this area. The work is the result of an NSF funded project (#0089412) that focuses on database design, SQL and Relational Algebra, and concurrency concepts. This paper describes animations developed to explain the major topics, as well as a repository of related tests. Evaluation and dissemination methods are also explained. Educators are encouraged to review the project including working prototypes at http://cofffee.kennesaw.edu.

1 Introduction

Many undergraduate computer science programs include a single required course in database systems. Yet database instructors know that it is difficult to present the critical topics in a single course. This paper describes the results of an NSF-CCLI proof of concept grant to develop courseware to accompany this required database course. Use of animations deepens and enriches standard presentations of topics by emphasizing a variety of learning styles [1]. In addition, the animations are designed to complement the most popular college database textbooks in use today.

The courseware topics include Database Modeling, Relational Algebra and Structured Query Language (SQL) and Concurrency. Animations form a major component of the project, and several have been developed to support each major concept. A repository of database test questions comprises a second component of the project. The next sections describe then animations, the repository of database tests, as well as evaluation and dissemination methods used.

2 Database Modeling

Animations have been developed to support standard modeling techniques such as Entity-relationship (ER) and Class Diagrams from the Unified Modeling Language. Additional animations have been developed to illustrate the mapping of diagrams into relational tables, and the normalization process of database design.

In mapping diagrams to Tables, the student is presented with a scenario and the corresponding ER diagram is then drawn (see Figure 1). The student is presented with several options for converting the data model into a relational schema. When the student chooses a solution for the transformation of the ERD, s/he is shown an animation that will reinforce the correct solution, or provide information about the flaws in incorrect solutions. That is, the pro’s and con’s of any option selected is graphically illustrated. Animations focus on avoidance of data redundancy and correct associations between entities.

![Figure 1 - E-R Diagram to Tables](image)

. Figure 1 - E-R Diagram to Tables
The normalization component is divided into tutorials and exercises. First the tutorial presents the student with an unnormalized relation or table. The student then may choose from a menu of options that graphically display key concepts such as functional dependencies, primary key, first normal form (1NF), second normal form (2NF) and third normal form (3NF). Normalization is illustrated for 1NF, 2NF, and 3NF. In each case functional dependencies are highlighted. In order to decompose tables in 1NF or 2NF, the student may choose to create a new table, selecting attributes that s/he wants to put into the new table. When the student finishes creating a table, the system will check for normalization violations. Specific feedback related to the violation is displayed then. When the student is finished creating all tables, the system will check each table for 3NF violations. If all the attributes from the original table are present in the new tables, then appropriate feedback is given to the student.

3 Relational Algebra and SQL

The animations illustrate SQL implementations of each primitive relational algebra operation, set functions, GROUP BY and HAVING clauses, and a unary join. Figure 2 shows an example of a join being implemented as a nested loop. The SQL Animations of these exercises are organized as follows

1. A description window that describes the problem and the objectives of the problem
2. An SQL query window containing the SQL query that solves the problem
3. One or more data windows containing the base tables
4. A procedure window containing the simplest form of a comparable algorithm, and
5. An output window containing the resultant table.

Animations also contain at least two buttons: a “clear” button that returns the query to its initial stage, and a “step” button that executes the query sequentially. Each step is highlighted and color coordinated to emphasize concept of query formation.

Depending on the query, there will also be a window containing the corresponding relational algebra solution to the exercise. These animations allow students to associate procedural code with new concepts SQL and relational algebra. This also encourages students to consider and examine the execution of a query in a step-wise fashion.

Since knowledge of a procedural programming language is a pre-requisite for most required database systems classes, and since effective learning often involves knowledge by association, the correspondence between procedural code and the composition of an SQL query allows for the students to master the new topics faster and to retain this knowledge longer.
We have also created an environment where students can construct their own queries. This gives the student the additional choice of increased interaction during the learning process, in addition to the above described visualization exercise. Figure 4 shows an example of the construction exercise involving the GROUP BY and HAVING clauses. The construction module allows student to decompose SQL queries into parts and quickly develop a solution. Notice that the student can manipulate the interaction through drop down menus and immediately see the results.

4 Concurrency

Concurrency animations initially examine problems that arise when data is accessed by two or more processes, transactions or users. These examples illustrate how the rollback and the commit statements work as well as different types of record locking strategies. Examples include the lost update with no record locking, deadlock, and dirty reads. Each is illustrated through animations as well as narrative descriptions of events leading to the concurrency issues. Figure 5 shows an example of deadlock where two transactions are trying to access the same records simultaneously. Transaction 1 holds a lock on the record transaction 2 is trying to access and transaction 2 holds a lock on the record transaction 1 is trying to access. As with other animations, every time the student clicks the “next” button, one line of code is executed and highlighted.
5 Repository of Database Tests

An online, free repository of database systems test questions and answers is now available, and is being increased frequently. Some educators are reluctant to return original tests to students because of the effort involved in designing new tests for each class and each semester. This web-based repository will provide students with the many examples they need in order to learn and reinforce key concepts without compromising the educator’s personal supply of test questions. Students will receive appropriate feedback via the online questions and answers, and this will reduce the risk of students memorizing answers from old test copies. Another purpose of this repository is to allow an instructor to quickly build a thorough database systems test for placement purposes, especially for new graduate students. One unresolved problem with the repository is that often rich questions are built in sequence, where one question depends on an earlier question. In the future, these streams of test questions will be appropriately categorized and organized on line.

6 Evaluation and Feedback

All visitors to http://coffee.kennesaw.edu are encouraged to complete the automated course survey after reviewing the software. It is also possible to download a paper copy and send to the project investigators. To date, the investigators have received much helpful feedback from both students and instructors using this on-line evaluation.

Initial evaluation of the project products involved reviews by local instructors and a small group of selected students. Following this early feedback, prototypes were then provided to an entire database class. Instructors showed students one example in class and then asked students to review the remaining examples outside of class. Results from these assessments led to several improvements in the software. For example, SQL animations were improved to provide students with a clear context and problem narrative. The developers assumed some pre-requisite knowledge and explanation that was present only when the instructor first illustrated the animations in class. So that students working solo may benefit, the software was modified to include a description window, describing the problem and the objectives. Other improvements have focused on user interface issues such as color, varying fonts and highlighting. Helpful suggestions led to placing more emphasis on the SQL Window and the problem description window. Another enhancement allows the student to choose to watch the entire animation at once, or to step through it using a button. However, in order for this option to be truly useful, the student must be able to control the speed and be able to interrupt the execution. Future work will incorporate this capability.

The current step in assessment involves educators and students from several universities. Although early results indicated a generally positive response, these evaluations may have been biased because students knew their professors had developed the courseware. For example, students requested the development of more animation examples. Interestingly, two students wrote in their confidential end-of-course evaluation that “the high quality animations greatly enhanced my learning of the course topics” and “animations contributed to the class presentations.” Subsequently a question concerning the database courseware has been included in this end-of-course evaluation.

7 Presentation Focus

The researchers will briefly present background literature and demonstrate some key prototypes in order to elicit suggestions from the audience. Audience
members will also be encouraged to review and use the remaining prototypes and provide feedback via an online form at http://coffee.kennesaw.edu.

References