

Kennesaw State University DigitalCommons@Kennesaw State University

Faculty Publications

1-2008

Expanding the Database Curriculum


Meg Murray

Kennesaw State University, mcmurray@kennesaw.edu

Mario Guimaraes

Kennesaw State University, mguimara@kennesaw.edu

Follow this and additional works at: <http://digitalcommons.kennesaw.edu/facpubs>

 Part of the [Computer Sciences Commons](#), [Management Sciences and Quantitative Methods Commons](#), and the [Science and Mathematics Education Commons](#)

Recommended Citation

Murray, Meg and Mario Guimaraes. "Expanding the Database Curriculum." *Journal of Computing Sciences in Colleges* 23.3 (2008): 69-75.

This Article is brought to you for free and open access by DigitalCommons@Kennesaw State University. It has been accepted for inclusion in Faculty Publications by an authorized administrator of DigitalCommons@Kennesaw State University. For more information, please contact digitalcommons@kennesaw.edu.

EXPANDING THE DATABASE CURRICULUM*

Meg Murray and Mario Guimaraes
Department of Computer Science and Information Systems
Kennesaw State University
Kennesaw, GA 30144
mcmurray@kennesaw.edu; mguimara @ kennesaw.edu

ABSTRACT

As database concepts and technologies continue to evolve there exists a need to expand the topics included in database curricula. This is challenging given the restraints on the number of courses that can be included in a typical CS or IS program. While a set of commonly identified core concepts and principles exists, there is little consensus on what supplemental materials should be included in database courses. Through an NSF proof-of-concept grant, we designed and developed courseware incorporating the use of animations to deepen and enrich standard presentations of core database concepts and to complement database teachings as found in the most popular texts. This paper proposes extending that model to modules of relevant and advanced database topics. Modules covered include data warehousing, data exchange via XML, database security and database system architecture, performance and tuning. Students in the computing disciplines must develop a strong knowledge base and build a firm foundation in core database concepts and principles. They must also extend this to supplemental and advanced topics. The dilemma for CS educators is to find avenues through which these topics can be explored in an efficient and effective manner.

INTRODUCTION

Computing technologies, by their very nature are constantly evolving. While there are foundational concepts and constructs that remain somewhat constant, new understandings continually emerge. Consequently, computing curricula are always in a state of review and revision. This is true in the area of database curriculum as database instructors recognize that with the rapid rate of change in the field, they need to include a broad spectrum of fundamental concepts in the courses they teach. This trend was

* Copyright © 2007 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

discussed at the 2000, 2002 and 2004 annual conferences of the ACM Special Interest Group on Computer Software Education (SIGCSE) panels [1], [3], [9], [10] and Database Birds of Feather sessions. It was also observed that there is a need to extend the breadth and depth of database curricula and to find ways to incorporate newer technologies. Most universities teach one core database course and many also offer one or more electives, but there is very little consensus on exactly what should be taught in these courses.

Compounding the challenges of identifying appropriate course content is the limited time available in a typical curriculum to expand the current knowledge base. Consequently, it is difficult to provide opportunities for students to attain a high level of understanding of database concepts and technologies. There is always a tradeoff between breadth and depth. One solution to addressing this dilemma resides in the development of instructional support materials. Textbooks are the traditional mechanism for supplementing classroom instruction but by themselves are not sufficient. Instead, students need to be provided with a variety of learning activities. Based on this, a proof-of-concept grant was explored and subsequently funded by the NSF. The goal of this proof-of-concept was to design and develop courseware incorporating the use of animations to deepen and enrich standard presentations of database concepts and to complement database teachings as found in the most popular texts. These animated instructional materials can also be used to expand the topics covered in the database curriculum.

This paper proposes a database curriculum structure centered on core principles which are expanded through the development of teaching modules of relevant and advanced database topics. Modules covered include data warehousing, data exchange via XML, database security and database system architecture, performance and tuning. While implementation of this structure might be envisioned as multiple database courses - one core class and three electives - practicality dictates the need for more flexible options which allow modules to be incorporated into other courses.

DATABASE CURRICULUM CONTENT

Studies which have looked at content included in database curriculum have identified a wide array of topics. Over time new topics have been added and emphasis on other topics has changed. For instance, topics frequently found in database textbooks published before 1990 include the hierarchical model, the network model, relational algebra, relational calculus and relational data structures. Topics frequently found in database textbooks published after 1990 include XML and data warehouse/data mining, mobile databases, middleware, object-oriented DBMS, SQL from within applications, stored procedures/functions, web and DBMS. However, common to both legacy database textbooks and modern database textbooks are topics including database design (creating E-R diagrams, converting E-R diagrams to tables and normalization), relational model (primary keys, foreign keys, domain integrity, referential Integrity), SQL, views, triggers, concurrency, security, recovery, tuning and performance. Given that such a diverse range of topics is representative of the breadth of database concepts and technologies, it is challenging not only to identify what should be included within a database curriculum but what specific topics should be covered in individual database courses.

Core Principles

While database curriculum includes a breadth of topics, there are foundational and core principles that should be included in an introductory database course. The concepts which appear most frequently in studies and in introductory database texts include database design, entity-relational model, relational data model and SQL and normalization [2],[8]. These topics require students to develop conceptual understanding, and areas such as design and SQL also demand that students acquire a basic set of skills. This type of learning is not efficiently achieved solely through the use of lectures and textbooks. We developed a series of animated tutorials and exercises, referred to as Animated Database Courseware (ADbC), that emphasize and reinforce these core concepts and skills. The ADbC currently consists of over 70 animations or tutorials categorized into three main modules (Database Design, SQL Queries and Transactional Processing) divided into 13 sub-modules. [Courseware prototypes for this work have been made freely available and may be found at <http://coffee.kennesaw.edu>]

The first module in the ADbC, Database Design, covers converting scenarios to E-R diagrams, converting E-R diagrams to tables and converting functional dependencies into normalization. In addition, it presents the visualization of E-R diagrams in several different notations to complement the notation set preferred by the instructor or used in the accompany textbook.

The SQL module covers interactive SQL, SQL from within applications, and stored procedures/functions. This module has been expanded to include support for more advanced SQL topics such as table management, including SQL in procedural code and the relationship of SQL with relational algebra. Sub-sections within this module include an Introduction to DDL and DML, Constructing SQL Queries, Animating SQL through Procedural Code, Animating SQL through Relational Algebra and SQL Misconceptions animated through relational algebra. We have also included a section on stored procedures and functions.

The module entitled, Transactions, covers the concepts of concurrency, triggers and recovery. Concurrency animations depict deadlocking, serial locking, lost update and record locking. Recovery examples illustrate physical update versus the logical update and the Triggers sub-module shows what happens to the triggering event when a trigger error occurs.

The ADbC is fairly intuitive to use so it has no learning curve. In addition, the software is not tailored to any specific product or textbook nor is it intended to substitute for a textbook or a product. Instead, it provides a means to facilitate student learning resulting in an opportunity to include more depth or breadth to the concepts covered in a database course. ADbC components may be used in the classroom, the lab or as out-of-class assignments. For example, in the classroom, the number of snapshots that the instructor can draw on the board is limited whereas the animations can be easily displayed. Furthermore, the program contains multiple views / solutions for the same problem. An example animation, “Animating SQL through Relational Algebra,” is presented in Figure 1.

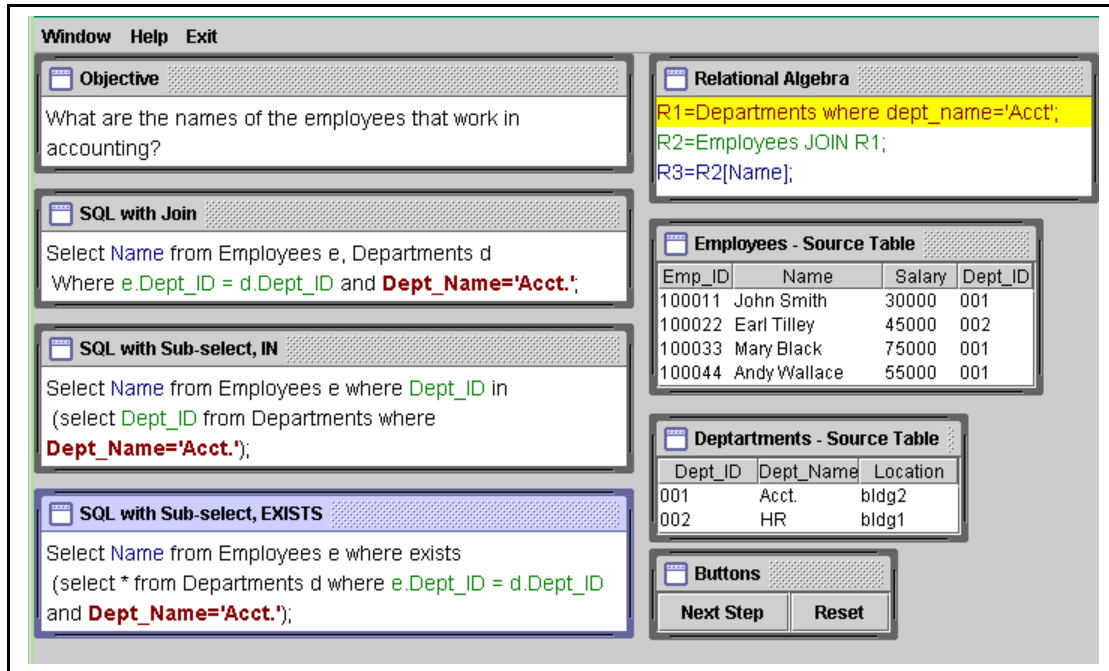


Figure 1. Example animation, “Animating SQL through Relational Algebra”

Additional Database Topics

Beyond the core, deciding what supplemental topics to include in an introductory database class is complicated, and yet, deciding what to cover in advanced classes is a subject of much greater difficulty. Topics that extend foundational knowledge in database principles often include such topics as security, XML data exchange, data warehousing, data mining, distributed database systems, middleware and web-based user interfacing, embedded use of SQL and development of stored procedures (which we discussed under core principles), mobile database performance, database system performance and tuning. Organizing and structuring this content into a CS or IS curriculum requires hard decisions to be made. Each of these topics could easily be designed as its own course, designed as a topic in a multi-topic advanced database course, introduced simply to provide exposure or not included at all. In any case, supplemental instructional materials will help to facilitate student learning. We are proposing that continued work in the development of animations to support database instruction be focused on data warehousing, XML data exchange, data security and database management system architecture, performance and tuning. Our experience has been that topics such as mobile databases or interaction with databases via Internet technologies are often covered in software architecture/application courses.

The use of data warehousing is experiencing explosive growth and student exposure to this technology can be achieved through software animations. A data warehousing component must begin by including an overview of basic data warehouse concepts and terminology as well as an introduction to multidimensional data models [7]. Additional considerations include metadata characterization, information retrieval, transactional processing and data mining. An important aspect of this course is to obtain large database

repositories in order to test performance. This course should also cover different tools to improve performance such as Quest's tool to re-write SQL queries (<http://www.quest.com/landing/leccotech.asp>). This course should also re-emphasize concepts learned in the first database class by including stored procedures, triggers, views and SQL from within programs. Functional aspects of data warehousing such as partitioning, roll-up, drill-down operation, slice and dice (projection operation on dimensions); and pivoting (cross tabulation) can be animated. In addition, strategies for extracting and loading, for cleaning and transforming, and different scenarios based on distribution of data (such as skewed data sets) can be supported through visualization.

XML is increasingly being incorporated with database systems and is becoming widely accepted as a means of data exchange between multiple database systems. Therefore, we propose a module focused on XML and its supporting technologies. Specifically, a first module in XML should introduce XML document creation, use of an XML schema and XML document conversion to and from a relational database. Finally, an animation will be developed that demonstrates abstracting data from one database, validating that data against a common schema, and passing the validated document to an external database system for insertion into the appropriate tables.

Database and database application security is of ultimate importance, but until recently, has been given little importance. According to Knox [6], "Securing the DB may be the single biggest action an organization can take to protect its assets." Traditionally, database security has focused on creating user accounts and managing user privileges to database objects. With the development of more distributed systems and expanded user access to data and database applications through networks and the Internet, an expanded focus on database security process and procedures is necessary. Components of a database security module might include access control, multilevel security, threats (SQL injection, buffer overflow), benefits and weakness of encryption within the database management system, database auditing and patching. Animations are proposed to be developed in database application security (SQL injection and buffer overflow) and database security (implementation of row and column level security using views and triggers as well as with virtual private databases, alternative methods of encryption, and database auditing procedures).

While topics related to database architecture and database startup are not commonly taught in a database course, students often have questions related to database administration. Therefore, this sub-module is proposed to meet student requests and to assist students if they should encounter the need for this information in advanced courses, in internships or on the job. Designing databases is a regular task of database administrators (DBA), and often, many hours are spent trying to locate missing parameters. Knowledge of database architecture improves the DBA performance in dealing with problems that may occur. Animation of the database startup process will include: 1) reading a parameter file to obtain sizing information and control file location, allocating System Global Area and starting the background process; 2) mounting the database (supplying instance with the location of the data and log files; 3) opening the database; and 4) examining background processes by checking header of each data file to make sure they have the most recent System Control Number. This module will also include discussions on ways to improve database system performance and tuning. For example, animations can be developed to illustrate the advantages and disadvantages of

the four popular indexing methods as they relate to database systems: balanced tree indexing, hashing, bitmap indexing and indexed-only tables.

CONCLUSION

As database concepts and technologies continue to evolve there exists a need to expand the topics included in database curricula. This is challenging given the restraints on the number of courses that can be included in a typical CS or IS program. Hard decisions must be made in regards to course content and topics to be covered. One way to approach this problem is through the development of thematic modules supplemented with instructional materials. We have developed a series of database courseware animations to address this need for commonly cited foundational knowledge covered in introductory database courses. These animations have been well received. Formal evaluation of this software through on-line evaluation forms, a comparison of test results from two introductory database classes and evaluations by faculty and students from external campuses reported positive results [5]. In this paper, we have proposed the development of additional modules which can be also be supported through software animation. Students in the computing disciplines must develop a strong knowledge base and build a firm foundation in core database concepts and principles. They must also extend this knowledge base to topics such as data warehousing, database security, data exchange and database system architecture and management. The dilemma for CS educators is to find avenues through which these topics can be explored in an efficient and effective manner.

REFERENCES

- [1] Adams, E. S., Granger, M., Goelman, D. and Ricardo, C. Managing the introductory database course: what goes in and what comes out? *Proceedings of the 35th SIGSCE Technical Symposium on Computer Science Education*, 497-498, 2004.
- [2] Conklin, M. and Heinrichs, L. In search of the right database text. *Journal of Computing in Small Colleges*, 21(2), 305-312, 2005.
- [3] Dietrich, S. and Urban, S. D. Database education: where are we now ? *Proceedings of the Thirty-fifth SIGCSE Technical Symposium on Computer Science Education*. 2004.
- [4] Guimaraes, M. New challenges in teaching database security. *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development*. InfoSecCD '06. ACM Press, New York, NY, 64-67. 2006.
- [5] Guimaraes, M. and Murray, M. A Review of the first phase of a project to develop and utilize animated database courseware. *Proceedings of the 2006 Annual Southern Association for Information Systems Conference*, 2006.
- [6] Knox, David C. *Effective Oracle Database 10g: Security by Design*. McGraw Hill, 2004.

- [7] McFadyen, R. Data warehousing in an undergraduate curriculum. *Journal of Computing in Small Colleges*. 22, 4 (Apr. 2007), 221-227, 2007.
- [8] Robbert, M. A. and Ricardo, C. M. Trends in the evolution of the database curriculum, *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education*, 139-143, 2003.
- [9] Springsteel, F., Robbert, M.A. and Ricardo, C. The next decade of the database course, *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, 41-45, 2000.
- [10] Wang, M., Guimaraes, M., and Myers, M. E. The database course (panel session): what must be taught. *Proceedings of the 31st SIGCSE Technical Symposium on Computer Science Education*, 403-404, 2000.