The Database Course: What Must be Taught

Mary Ann Robbert (Moderator)
Bentley College
CIS Department
175 Forest Street
Waltham, MA 02154
Mrobbert@bentley.edu

Ming Wang
Embry-Riddle
Aeronautical Univ.
Department of CS
Daytona Beach, FL
32114
wangm@db.erau.edu

Mario Guimaraes
Texas A&M Univ.
Department of
Computer Science
Corpus Christi, TX
78412
mariog@sci.tamucc.edu

Martha E. Myers
Kennesaw State Univ.
Computer Science and
Information Systems
Department
Kennesaw, GA 30144
Mmyers@kennesaw.edu

Keywords: database systems, pedagogy

Introduction

Last year, in New Orleans, database faculty assembled and discussed the content of database courses. This panel is a formalized follow up to that discussion, examining the content of current database courses and proposing other topics that should be included. An objective of the session is to produce a prioritized core content for a database course that encompasses both traditional and contemporary topics and is dynamic enough to reflect the state of database systems into the new millennium.

The courses content indicated by the forty-five SIGCSE members who responded to my survey indicates that the primary material covered in the course has varied little over the past twenty years. Two-thirds listed SQL as one of three primary topics covered. Over half had database modeling and design as the second topics but these subjects were restricted to the relational model. Relational algebra and normalization were the other leading topics. Again focusing on the relational model.

A database course should include three components: database system architecture, database design and database implementation. Single semester database courses tend to focus on the relational model thus restricting topics and forcing the inclusion of others. Once the relational model is selected, relational algebra, the E-R model, SQL, and normalization are included in the curriculum and a particular relational database management system is used for the implementation. Because of time constraints, there is a tendency to present all three components but only through the relational view of the world. There is more to the database world than the relational model and other topics deserve coverage.

The database course usually includes an implementation. If a specific database management system is used, teaching the tool can further restrict the time available for current database topics such as object-oriented databases, distributed databases, data warehousing, front end tools and web based applications.

This panel will present their methods for determining and weighing what must be included in a single semester database course. Panelists with different perspectives were selected to balance the examination of the curriculum issue. Discussion of the content of a second semester elective will be included as time permits.

Ming Wang

We need to teach the relational model, relational algebra, SQL, ER diagram, normalization, database design, database administration, OODBMs and non web and database web application programming. I teach other database models in one class period of time. My focus is RDBMs. I believe that RDBMs is going to be around for many years. However, we now need to talk about the Object-relational database (ORDBMs) also.

Using the ORDBMS we can resolve many of the weaknesses of relational databases and teach, reusing and sharing of data, understanding object-oriented concepts, maintenance of a consistent structure in the database, and processing complex data. Object concepts are easy to learn and implement in software such as Oracle8 and DB2. It takes about at most two classes assuming students already know SQL/PLSQL.

The term project is very important for this course. It summarizes all our students have learnt during the semester and it can be presented during the job interview.

Mario Guimaraes

An undergraduate database course should include: Relational Algebra, SQL, Database Design, Accessing the Database through a front-end language, and a project where the student will focus on these topics. Relational Algebra is the foundation of SQL and can be presented in a small amount of time. SQL is a worldwide standard and deserves a considerable amount of time (four or more classes). Database Design, including E-R diagrams, normalization, creating integrity constraints and views, and granting/revoking access to tables and views should also be the core of an introductory class. The design lays the groundwork for success in supporting current and future needs of any database application.

A real world project has multiple benefits: 1) students learn better through a particular domain of their interest, 2) students see the practical value of what they learned, 3) it re-enforces the material learned, 4) students will see a need to access the database through a front-end (Developer/2000, Visual Basic, Delphi, etc). The topics listed above are of important value to the student and are easily assimilated through a practical project. This contrasts with the difficulty of teaching Data Management issues such as backup and recovery and performance issues.

Martha E. Myers

The first course in database systems should include data modeling, database design, database processing, and exposure to a particular DBMS. More than one database model should be considered, the best models are the relational and, to a lesser extent, the object-oriented models. Assuming prerequisite knowledge of programming principles and some experience with elementary data modeling, other topics that can be considered are ER modeling, OO modeling, logical database design, physical database design, some aspects of implementing specific applications, and data administration. Logical database design should include transforming data models into relational designs, as well as OO designs. Logical database topics should include a discussion of normalization; physical database topics should include a discussion of de-normalization. Various methods of database access should be examined including QBE, SQL, and database programming. Some additional (optional) topics may include web access to DBs, client-server and distributed architectures, data warehousing, a day in the life of the database administrator, fine-tuning a database design, and advanced normalization techniques.

While a project is an important learning experience in this field, the scope of the database project may be almost trivial if the CS majors curriculum includes other classes (such as software engineering) with a heavy project component.