

Teaching Database in Two Courses: Reconciling Theoretical Framework with Practical Considerations

Li Yang, Mustafa Sanver

Department of Computer Science, Western Michigan University

Abstract

Database management systems represent a large field of study in computer science. Many computer science departments offer two or more database courses. A usual practice is to use the first course for general introduction and database application programming and to dedicate the second course to database management system implementation. Although this framework works well, there are practical issues that need to be resolved to meet the demands of students with different backgrounds. This paper discusses the contents of these database courses and the practical issues that we have experienced in teaching these courses.

1. Introduction

Database management systems (DBMSs) have been a fast-growing area during the past decades. DBMSs are now ubiquitous as fundamental tools for managing data and for supporting advanced applications. As a result, DBMS courses have been offered from different perspectives in many disciplines. This paper discusses DBMS education from a computer science perspective, that is, we focus on the fundamental principles and techniques inside DBMSs, especially relational DBMSs. These principles and techniques of DBMSs are now integral parts of computer science curricula.

DBMSs and information management represent a large part of computer science education. The following is a list of units of the Information Management area in ACM Curricula 2001¹ that are recommended to computer science undergraduate students:

- IM1. Information models and systems
- IM2. Database systems
- IM3. Data modeling
- IM4. Relational databases
- IM5. Database query languages
- IM6. Relational database design
- IM7. Transaction processing
- IM8. Distributed databases

- IM9. Physical database design
- IM10. Data mining
- IM11. Information storage and retrieval
- IM12. Hypertext and hypermedia
- IM13. Multimedia information and systems
- IM14. Digital libraries

Many Computer Science departments adopt the above recommendation by offering two or more regular courses on DBMSs to cover topics in units [IM1-IM7, IM9] and leave the rest units to advanced research seminar courses. It is commonly accepted that the typical DBMS topics in a computer science curriculum^{2,3,4} are organized into the following seven categories:

1. **Theory and concepts:** Introduction, ER model, conceptual design, relational model, relational algebra, relational calculus.
2. **Applications:** SQL, embedded SQL, database APIs, application development, Internet applications, XML.
3. **Storage and organization:** Data storage, tree index, hash table, buffer management.
4. **Query processing:** External sorting, nested-loop execution, merge-sort execution, index only execution, optimization of individual operators.
5. **Transaction management:** ACID properties, concurrency control, lock-based and timestamp-based, crash recovery, 2-phase commit.
6. **Design and tuning:** Schema refinement and normalization, physical design and performance tuning, authorization and database security.
7. **Advanced topics:** Distributed and parallel databases, object-oriented databases, deductive databases, data warehousing, data mining, information retrieval, geographical information systems, multimedia databases, etc.

Many computer science departments offer two regular courses and a list of seminar courses on DBMSs. The following is a typical partition of the above topics among these courses:

1. The first course is an introductory course which aims to present the principles of database systems, particularly relational database systems, with a practical focus on database design theory, database programming, and application development. The course teaches students how to use a database management system. It does not cover any topic on how a DBMS works inside. This first course usually covers topics in Categories 1 and 2.
2. The second course has a strong system emphasis and discusses implementation techniques of DBMSs. It assumes that students have good programming skills and are curious to know the various techniques used in a relational DBMS. Students can be asked to implement various parts of an experimental relational DBMS. Students will also learn physical design and DBMS performance tuning. The second database course covers topics in Categories 3, 4, 5, and 6.
3. Topics in Category 7 are usually left to advanced seminar courses.

There are also schools that combine the above two regular courses into one course. Such a comprehensive introductory course usually emphasizes the contents of the above first course and briefly introduces the contents of the above second course.

2. Database Courses at WMU

At WMU, CS500-level courses are taken both by undergraduates and by graduate students. CS600-level courses are reserved for graduate students. We have two regular courses on DBMS, CS543 and CS643. The prerequisite for CS543 is CS331 --- Data Structures and Algorithms. Students who wish to go further into DBMS take CS643, and then CS603 where advanced topics on data warehousing, data mining, web database are covered.

Our students choose the database courses for some of the following reasons. These reasons have helped us to understand students' expectations from CS543 and CS643.

1. The simplest reason is that database is a popular course.
2. Databases have created one of the largest job markets. There are enormous employment opportunities for database experts.
3. Problems and technical challenges in database systems are well defined. DBMSs provide an ideal platform of software development for students to apply the key concepts, algorithms and techniques that they have learned in other courses, for example, data structures, file structures and algorithms, programming languages, software engineering, parallel computation, etc.

Historically at WMU, CS543 was a comprehensive course that combined database introduction and system implementation. CS643 was a course on advanced topics emphasizing distributed databases. Clearly, this was a typical partition of database courses about 10-15 years ago when the field was much smaller than it is nowadays. CS543 emphasized database design and relational theory without discussing modern database applications and without discussing anything on modern database implementations. If an instructor taught the course nowadays, he/she would have actually no time in a single semester to cover DBMS implementation techniques, for example, query optimization and concurrency control. Consequently, there would be a big gap between CS543 and CS643, which discussed distributed database systems.

We have redesigned CS543 and CS643 to reflect modern contents of database systems. CS543 has been designed as an application-oriented course where students learn how to use a database. CS643 has been designed as a system-oriented course which covers DBMS implementation techniques. Student surveys showed that these changes are welcome by students. Many students just need to know how to use database systems rather than how to implement a DBMS while other students are interested in going further into database implementation and performance tuning.

3. Issues and Concerns

The redesigned CS543 covers topics in Categories 1 and 2. The redesigned CS643 covers topics in Categories 3, 4, 5 and 6. In general, CS543 is application-oriented and teaches students to be database users and application developers. CS643 is system-oriented and teaches students to be DBMS developers and/or system administrators. Specifically, the two courses and the topics in each of them are listed in the following:

Topics of CS543:

Theory and concepts: Introduction, ER model, conceptual design, relational model, relational algebra and relational calculus.

Applications: SQL, Embedded SQL, application development, database APIs, Internet applications, XML.

Design: Schema refinement and database normalization.

Topics of CS643:

Storage and index: Data storage, tree index, hash table, buffer management.

Query processing: external sorting, nested-loop execution, merge-sort execution, optimization of individual operators.

Transaction management: ACID properties, concurrency control, crash recovery.

Physical database design and tuning: Physical schema design, performance tuning, authorization and database security.

This arrangement works well for students. However, we have noticed some issues and concerns from students' feedback and our own teaching experience. The following are some of them and our ways in dealing with them:

1. *Undergraduates have no chance to learn database implementations.* There are always gifted undergraduate students who are curious to know how a DBMS works or who want to apply what they have learned in data structures, algorithms and operating systems to the management of data. Our CS600-level courses are dedicated to graduate students. Putting database implementation into CS643 means that our undergraduate students have no chance to learn it. As a tentative solution, these students are encouraged to take database implementation projects in CS597. CS597 is a course on research projects and is offered on an individual basis. It is open to undergraduate students.
2. *Many undergraduate students are interested in jobs as database administrators and therefore are interested to learn physical database design.* Unfortunately, physical database design is a topic in CS643 which requires an understanding of database implementation techniques. Again, we encourage these undergraduate students to take database projects under CS597.
3. *Do we need in CS543 an overview of CS643?* To encourage students to take the two courses continuously, we give a quick overview of problems and technical challenges in database implementation by the end of the course. We encourage students to think about these challenges and to give their own solutions by using what they learned in data structures and operating systems.
4. *Hands-on experience and projects.* We use Oracle 9i for lab assignments in CS543. Lab assignments include database creation, SQL query, embedded SQL (Pro*C in Oracle) and JDBC. To emphasis practical experience in CS543, we insist that students are organized into groups. Each group must select and complete a database application project. Each group is asked to choose a project from their personal or employment experience. Students are asked to go through all steps of requirement analysis, database

design, deployment, and application development. Teamwork is encouraged in course projects to prepare students for an industry-type working environment. Whenever possible, we encourage that each group consists of a mix of graduate and undergraduate students. This structure promotes the interaction between undergraduate and graduate students, develops the mentoring skills of graduate students, and exposes undergraduate students to the excitement and challenges of graduate education. All groups are asked to demonstrate their projects in the last week of the semester.

5. *Research exposure and technical survey in CS643.* How can we give students early exposure of research topics in databases? Do we need to introduce emerging techniques and new applications in regular courses? CS643 is a terminating regular course on DBMSs. We think it is necessary to give graduate students some research exposure. For this purpose, one part of CS643 is technical survey. Students are asked to form groups according to their own interests. Each group is asked to conduct a technical survey on a particular topic on DBMSs or a modern database application. Most of these topics are chosen from topics listed in Category 7. Gifted students who like technical challenges are invited to go further by choosing fundamental problems and do their own research after they finish CS643. Each group is given a list of representative research papers from leading technical journals or conferences. Each group is given a chance for class presentation.
6. *Sequence of teaching.* One important issue is in which order the topics of each course should be covered. In CS543, for example, a conventional way of instruction is to discuss topics in the following order: database design, relational model, relational theory, relational algebra and calculus, SQL and database application development. If the course has substantial database-application design and implementation projects, however, the instructor may find that SQL is taught too late that students do not have enough time to complete their projects. In this case, the instructor may have to reorder the material somewhat, so that SQL and application programming are taught earlier. On the other hand, the instructor may wish to defer the topics such as functional dependencies and normal forms, although students need knowledge about normalization for database design.

Finally, our CS543 and CS643 are designed to cover the following topics respectively. The topics of each course are listed in the order that they are taught in class.

CS543:

Introduction, ER model, conceptual design, relational model, relational algebra and relational calculus, SQL, embedded SQL, application development, database APIs, Internet applications, XML, FDs, schema refinement and normalization.

CS643:

Data storage, tree index, hash table, buffer management; external sorting, nested-loop execution, merge-sort execution, optimization of individual relational operators; ACID properties, concurrency control, crash recovery, physical design and performance tuning, authorization and database security.

4. Future Improvements

From the above, we can see that many problems were caused by the inaccessibility of the database implementation course to undergraduate students. In the near future, we plan to propose to push the contents of current CS643 to the undergraduate level and, consequently, reshape our database courses in the following way:

CS343 or CS443:	Introduction and database applications
CS543:	Database implementation
CS643:	Advanced topics in DBMS

Plus seminar courses in database applications such as data warehousing and data mining.

References:

1. Final Report of the Joint ACM/IEEE-CS Task Force on Computing Curricula 2001 for Computer Science, <http://www.computer.org/education/cc2001/final/index.htm>
2. C. J. Date. An Introduction to Database Systems, 8th Edition, Addison-Wesley, 2003.
3. R. Elmasri, S. Navathe. Fundamentals of Database Systems, 4th Edition, Addison-Wesley, 2003.
4. R. Ramakrishnan, J. Gehrke. Database Management Systems, 3rd Edition, McGraw-Hill, 2003.

LI YANG is an associate professor at the Department of Computer Science, Western Michigan University. His research interests include knowledge discovery, information visualization, pattern recognition and databases. He is a senior member of IEEE and a member of ACM.

MUSTAFA SANVER is a Ph.D. student in the Department of Computer Science, Western Michigan University. His interests include visualization of databases, distributed databases, and security for transactions over the Internet. He is a member of ACM and the Honor Society of Phi Kappa Phi.