Advanced Database Concepts for Undergraduates: Experience with Teaching a Second Course

Susan D. Urban and Suzanne W. Dietrich
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-5406
{s.urban | dietrich}@asu.edu

Abstract

This paper describes the development of a second database course for undergraduates, preparing students for the advanced database concepts they will experience in industry. Assuming an introductory course on relational database systems as a prerequisite, the topics addressed in the course include object-oriented data modeling, object-oriented database systems, object-relational database systems, Web access to databases, and professionalism and ethics. We present our experience with teaching the course, elaborating on the topics and assignments. We also present feedback from students and industry partners as well as our own assessment of future course refinements.

1 Introduction

Database management concepts have always been an important part of computer science education. Topics for teaching relational database management systems range from query languages such as relational algebra, relational calculus, and SQL, to an abundance of results on functional dependencies, constraints, normal forms, view updates, query processing, transaction processing, logging and recovery, concurrency control, and distributed processing. Database concepts have continued to expand over the years, creating many different types of specialized database systems. Recent database developments include issues involving information access on the World Wide Web.

The explosive growth of database concepts has created a dilemma for database education, especially at the undergraduate level. The demand for database skills in computer science majors with a Bachelor of Science degree continues to increase in industry. However, the typical three-credit hour course on introductory database concepts found at most universities is incapable of adequately covering traditional relational database concepts *and* advanced database concepts that have evolved over the last decade. Students with excellent academic records typically gain exposure to advanced database topics by taking graduate-level courses. This approach does not provide an equitable solution for the majority of undergraduate students. Enrolling undergraduates into graduate level courses also makes it difficult for an undergraduate computer science curriculum to address Engineering Curriculum 2000 (http://www.abet.org) and Computer Sciences Accreditation Board (http://www.csab.org) requirements, such as development of oral and written communication skills and coverage of professionalism and ethics. Undergraduates preparing for the job market are better served by approaching advanced database topics from a practical point of view in an environment that is specifically suited to the educational needs of undergraduates.

This paper describes our experience with the development of an advanced undergraduate database course (http://cse.asu.edu/~cse494db). The course is being taught within the Department of Computer Science and Engineering (CSE) at Arizona State University (ASU). Through a previous NSF Instrumentation and Laboratory Improvement grant (DUE-9451489), we have already enhanced our introductory relational database management course with active learning concepts and a semester-long group project that integrates database theory with practical design and implementation issues (http://cse.asu.edu/~cse412). The advanced course described in this paper provides a second, project-oriented database course at the undergraduate level with an industry view-point of topics that are generally taught only from a research perspective at the graduate level.

The specific topics addressed in the course include object-oriented data modeling, object-oriented database systems, object-relational database systems, and Web interaction with database systems. The course also emphasizes group activity and oral communication skills as well as ethical issues for data access, security, and privacy. The curriculum for the course is being evaluated and refined with the help of input from industry. Most computer science departments can accommodate such a course through the use of electives for students that want to

SIGCSE 2001 2/01 Charlotte, NC USA

specialize in database systems. Our goal is to produce a collection of advanced database modules that will provide educators with the flexibility to tailor the use of such modules to their own specific program.

In the following sections of this paper, we first provide a summary in Section 2 of undergraduate database education in the U.S. Section 3 then elaborates on the course contents, describing the topics and assignments of the course. Section 4 presents lessons learned from our first offering of the course, covering our evaluation of the course as well as student evaluations and feedback from industry. The paper concludes in Section 5 with a summary of our work and a discussion of our future plans for the course.

2 Undergraduate Database Education in the U.S.

To develop our plan for an advanced database course for undergraduates, we investigated the courses that are currently offered at several major universities across the U.S. with recognized computer science departments in database research and education. Figure 1 provides a summary of our findings, which were gathered through examination of information available within department Web sites at each university. Figure 1(a) provides a summary of the topics typically found in graduate-level database courses while Figure 1(b) indicates the topics that are typically offered at the undergraduate level. Graduate courses generally address data models beyond the relational model, such as deductive, object-oriented, and multimedia databases. Furthermore, graduate courses focus on an in-depth study of database language and database system implementation issues. Most departments offer such topics over two or three courses at the graduate level, while others provide special topics courses where the focus varies with each offering of the course.

At the undergraduate level, the departments examined typically offered only one undergraduate database course. Although the courses vary from university to university, there is generally a core set of topics that are taught in such a course, which is reflected in the table of contents of several database textbooks [5, 9]. The core topics shown in Figure 1(b) are those that we offer in our own undergraduate database course at ASU. Some courses offer variations to these topics with coverage of legacy models such as the network and hierarchical models. A small number of courses also mentioned coverage of object-oriented models. Other courses provide instruction on database implementation issues, while others provide more emphasis on database design and analysis concepts.

What we have observed from this data is that undergraduates are not significantly exposed to advanced database topics. A single three or four hour course at the undergraduate level is packed full of important concepts on basic data modeling as well as relational database features, from query languages to system-related issues such as

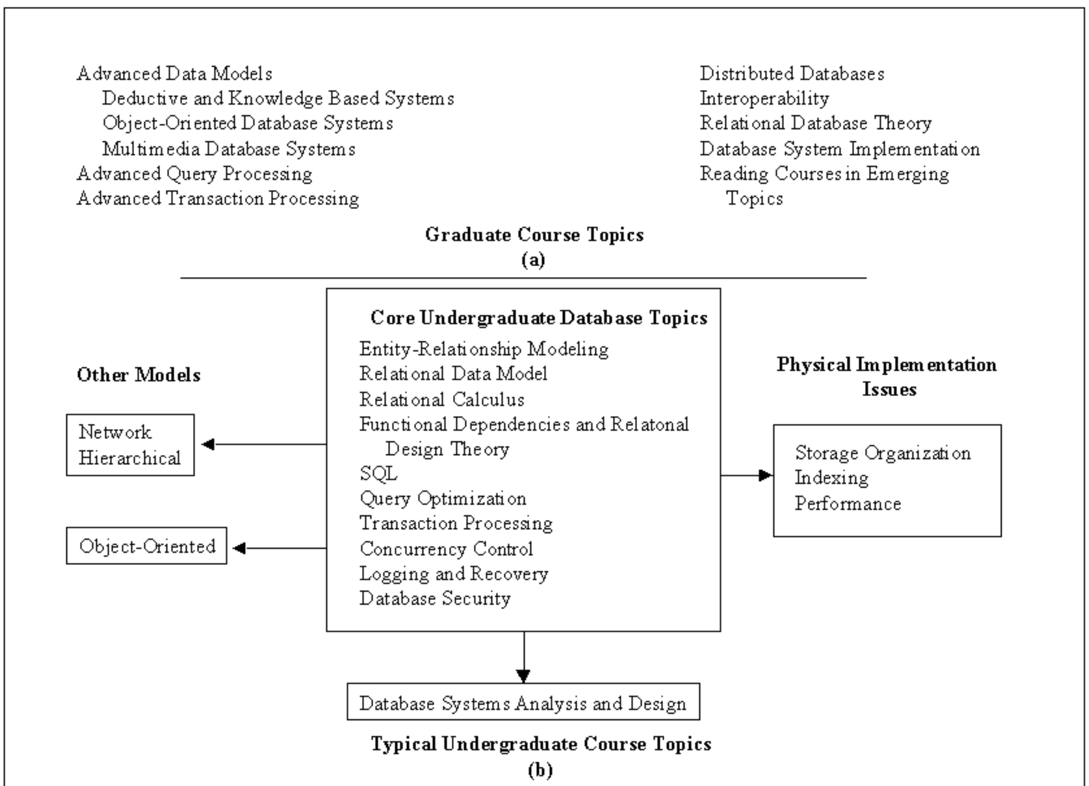

Figure 1: Overview of Database Education

transaction processing and concurrency control. As a result, providing in-depth coverage of more advanced database issues is difficult. Having a course that is already full with a wide range of topics also makes it difficult to provide students with experience in oral and written communication. Furthermore, ethical issues associated with database access are not thoroughly addressed. Although database security issues are included in most database textbooks, security issues are addressed in terms of the features provided by database management systems for controlling user access. Database education should play a stronger role in covering the professional and ethical issues associated with information access and dissemination, especially considering the recent growth in information access through the Internet. Undergraduate database education must be expanded to 1) provide undergraduates with a more well-rounded exposure to current database concepts, and 2) provide opportunities for the inclusion of ethical issues as well as oral and written communication exercises.

3 Advanced Database Course

Figure 2 provides an outline of the topics covered in the undergraduate course that we are developing for advanced database concepts. As indicated in Figure 2, the course is divided into five main modules addressing advanced data modeling, databases and the Web, object-oriented database systems, object-relational database systems, and professionalism and ethics. The three main challenges in teaching these topics are: 1) presenting the material in a manner that is appropriate at the undergraduate level, 2) providing students with experience in the use of advanced database tools without reducing the course to a technology training class, and 3) presenting the material in an appropriate sequence that provides balance to the due dates for assignments. We have addressed the first two challenges by building on our experience as described in [4, 11]. In particular, we make use of in-class, cooperative learning exercises to promote team-building skills as part of the learning experience. We also provide software and database examples of the tools used in class assignments, where students work together in groups to study the examples for the purpose of learning the syntactic details of the tools. As described in [11], this approach minimizes the amount of time that must be spent in class to teach the use of specific tools. Section 4 addresses our assessment of the topic sequence in more detail.

Advanced Data Modeling
- The EER Model
- UML Conceptual Class Diagrams
- EER vs. UML

Databases and the Web
- Architectural Issues
- JDBC
- Servlet, ASP and JSP Case Studies

Object-Oriented Database Systems
- Characteristics of OODBs
- Data Model - ODMG Standard
- Data Definition - ODL
- Data Manipulation - OQL
- Objectivity/DB Case Study

Object-Relational Database Systems
- Abstract Data Types
- Data model – Object extensions to relations
- Data Definition
- Data Manipulation – Object extensions of SQL
- Oracle-8 Case Study

Professionalism and Ethics
- Privacy & Confidentiality
- Unauthorized Access

Figure 2: Advanced Database Concepts for Undergraduates

The module on advanced data modeling provides the conceptual foundation for advanced implementation data models, such as object-oriented and object-relational data models. Since the prerequisite database course covers the basic Entity-Relationship (ER) model, we begin with coverage of the Enhanced Entity Relationship (EER) conceptual model, introducing advanced modeling concepts involving class hierarchies, constraints on class hierarchies, inheritance, and categories. We also present mapping techniques for generating relational schemas from EER diagrams. Since the Unified Modeling Language (UML) [10] is emerging as a modeling standard for software engineering in industry, we also cover UML conceptual class diagrams, which is a superset of the EER model. UML allows us to introduce the behavioral aspects of classes into the modeling process. We use cooperative learning exercises to reinforce the material, where students construct EER and UML diagrams, generate relational schemas, and compare/contrast the EER and UML techniques. Through the remainder of the course, we use UML diagrams in our examples because of the object-oriented nature of the database systems that we address.

The introduction to UML provides a way to transition into the topic of object-oriented database systems (OODBs). We begin the OODB section with an overview of fundamental object-oriented concepts, such as complex objects, encapsulation of behavior, inheritance, and polymorphism. Most of the students in class are already familiar with these basic concepts from previous courses on object-oriented programming, so our focus is on persistence as well as the difference between types and extents. We also focus on object identity and the way in which objects are related through object identifiers rather than through external keys as in the relational model. Topics such as the comparison of objects through identity, shallow equality, and deep equality are also addressed.

From an OODB modeling point of view, we cover the object data definition language (ODL) of the Object Data Management Group (ODMG) standard [1] and discuss mapping issues between EER, UML, and ODL schemas. Our coverage of the ODMG standard also includes basic features of the object query language (OQL). We use Objectivity/DB as a case study of OODB technology [7]. Students are given a software example that we have prepared using an Objectivity/DB database together with a graphical user interface (GUI) written in Java that allows a user to browse and update any object-oriented database [3]. The GUI assumes the implementation of a given interface for accessing the particular object-oriented database. Since past experience with the use of an OODB product illustrated that the students spent a significant amount of time coding an appropriate interface to demonstrate the correctness of their implementation, we developed a GUI that the students reuse as an interface in their assignment to develop their own Objectivity/DB implementation. Students work together in teams to study the Objectivity example, with the implementation exercise assigned as an individual project.

Since object-relational systems extend traditional relational systems with object-oriented concepts, we cover object-relational database (ORDB) systems after we cover OODB systems. After addressing some of the fundamental concepts of ORDBs, such as the integration of abstract data types with relational systems as well as the use of objects in relations, we use the Oracle 8 object-relational database system as a case study [8]. The specific features covered include embedded objects, object tables, nested tables, and VARRAYS (variable length arrays). Some of our own research has developed guidelines for mapping conceptual models such as EER and UML diagrams to object-relational schemas [12]. As a result, we revisit object-oriented modeling concepts and the manner in which object-oriented designs can be effectively mapped to object-relational features. These mapping issues include the study of advanced features of database technology, such as the use of triggers and stored procedures for constraint enforcement. We also study object extensions to SQL in Oracle 8. As in the OODB part of the course, students are given a database example that we have developed using Oracle 8 to study for the preparation of their own implementation assignment.

The module on databases and the web is addressed in a different manner than most of the other modules. The ability to access database systems from the Web is an important topic that students often face as they enter the job market. There are many different technologies that can be used, such as Java servlets, Active Server Pages (ASP), and Java Server Pages (JSP) [2, 6]. We focus on teaching techniques for the remote access of database systems, such as Java Database Connectivity (JDBC) [6]. We then provide an overview of Web technology that uses features such as JDBC and Open Database Connectivity (ODBC) to access database systems. We provide our own examples of JDBC, servlets, ASP, and JSP for database access. Students explore these examples and are assigned a semester-long Web/DB group project, where each group must design and implement a Web-accessible relational database. Most of the students have already been exposed to some aspects of Web technology in other courses. The group project gives the students the opportunity to fully explore the use of such technology for the development of database applications.

Throughout the course, we emphasize issues associated with professionalism and ethics (P&E). We are especially interested in concepts associated with the sensitive nature of data management as well as responsible behavior in the design and implementation of information systems. Our course Web page provides links to resources on P&E as well as specific case studies. Students are given an initial assignment to read the case studies and prepare for an in-class discussion of the issues. After discussion of the case studies, the students are required to find a recent news article involving P&E issues. As a group exercise, the students must prepare their own case study, develop a set of discussion questions, and lead an in-class discussion of their case study. We have found this to be a quite creative exercise for the students, providing opportunities for developing oral and written communication skills. Most of the students found this to be a useful and informative change of pace from traditional implementation exercises.

4 Lessons Learned

As mentioned in the previous section, developing a sequence for presentation of the course modules was a challenge. We presented the material in the following order: 1) advanced modeling, 2) initial discussion of P&E case studies, 3) OODBs (except for OQL), 4) databases and the Web, 5) ORDBs, 6) OQL, 7) student-led discussions of P&E case studies, 8) Web/DB group project presentations. In our next offering of the course, we plan to introduce Web/DB issues as well as advanced relational topics, such as triggers and stored procedures, earlier in the semester. Students will therefore be able to immediately begin the Web/DB group project. Covering advanced relational topics before the introduction of object extensions to relations is also important to ensure that students are not overloaded with advanced relational issues in the ORDB module. We feel that students can more easily grasp object extensions to relational systems after covering pure OODB concepts. But since the students enter the course with a background in relational systems, they can easily get started with the more procedural issues of relational applications early in the semester, while object extensions are addressed at a later time.

Other planned content changes include coverage of the Extensible Markup Language (XML) (http://xml.about.com/compute/xml/mbody.htm), coverage of SQL99 (http://www.sql-99.org), and more complete coverage of OQL with comparisons to SQL99. Students have also indicated an interest in covering XML. There should be time to add these topics to the course since we plan to eliminate the in-class group project presentations in favor of out-of-class project demonstrations. Students receive plenty of opportunity for oral communication through the in-class exercises and the P&E case study discussions. Students liked the P&E exercises and suggested that we enhance this module with industry speakers. Other feedback from students indicates that they appreciated the database examples that we provided, although some of the examples need to be scaled down and refined. Students also suggested the use of a specialized tool for drawing UML diagrams.

In the Summer of 2000, a workshop was held at ASU to solicit feedback from industry on the contents of the course. Our industry partners confirmed that the current topics of the course are appropriate for advanced coverage of database issues. In addition, the industry participants strongly agreed with our assessment to include coverage of XML and SQL99. Although they indicated that there is already more than enough material for a second course, participants also suggested the addition of a course component to introduce advanced issues, such as interoperability, data warehousing, security, personalization and internationalization.

5 Summary

This paper has presented our efforts in the development of an advanced database course for undergraduates. Due to developments in information technology over the last decade, a basic background in relational database technology is often insufficient for the types of jobs that now exist in industry. A one-semester course on database systems, however, is incapable of adequately covering important relational topics as well as new developments in database technology. Our approach to this problem has been to define an advanced course that is specifically geared towards the needs of undergraduates. Student reaction to the first offering of the course has been overwhelmingly positive. We are currently developing a plan for following the industry experience of students who have taken the course to evaluate our success in preparing students for database-related jobs. We are refining the course with the feedback that we have received from industry and are also recruiting other universities to serve as test sites for teaching the course.

Acknowledgements

This work was supported by NSF's DUE CCLI-EMD program (DUE-9980417). We thank Microsoft Research and ASU CRESMET (Center for Research on Education in Science, Mathematics, Engineering, and Technology) for their initial funding of this project. We would like to thank the students of CSE 494DB for their constructive feedback. We are indebted to Marla Hart, Lakshmi Priya, Mathangi Ranganathan, Dan Suceava, Pablo Tapia, and Nilan Yang for their invaluable support in developing the first round of curriculum materials. We also want to thank the participants of the industry workshop: Lee Gowen (American Express), Leon Guzenda (Objectivity), Gary James (Homebid), Vishu Krishnamurthy (Oracle), Amy Sundermier (Homebid), Roger Tomas (AGCS), Steve Waters (Microsoft Research), Arleen Wiryo (Integrated Information Systems), Cindy Wu (Honeywell).

References

[1] R. G. G. Cattell, *The Object Database Standard: ODMG 3.0*, Morgan Kaufmann, 2000.

[2] Deitel, H. M., Deitel, P. J. and Nieto, T. R., *Internet and the World Wide Web: How to Program*, Prentice Hall, 2000.

[3] Dietrich, S. W., Suceava, D. Cherukuri, C., and Urban S. D., A Reusable Graphical User Interface for Manipulating Object-Oriented Databases using Java and XML. *Proceedings of the 32nd ACM SIGCSE Technical Symposium on Computer Science Education*, (Feb 2001).

[4] Dietrich, S. W. and Urban, S. D. A Cooperative Learning Approach to Database Group Projects: Integrating Theory and Practice. *IEEE Transactions on Education*, (Nov 1998), CD-ROM Directory 06, 11 p.

[5] Elmasri, R., and Navathe, S. *Fundamentals of Database Systems*, Addison Wesley, 2000.

[6] Hunter, J., Crawford, W., and Ferguson, P., *Java Servlet Programming*, O'Reilly Publishers, 1998.

[7] Objectivity/DB Version 5.1: *Database Development*, Objectivity Inc., 1999.

[8] Oracle Corporation, Oracle 8 Reference Manual, Release 8.0, 1999.

[9] Ramakrishnan, R., and Gehrke, J. *Database Management Systems*, 2nd edition, McGraw Hill, 2000.

[10] Rumbaugh, J., Jacobson, I. and Booch, G. *The Unified Modeling Language Reference Manual*, Addison Wesley Longman, 1999.

[11] Urban, S. D. and Dietrich, S. W. Integrating the Practical Use of a Database Product into a Theoretical Curriculum. *Proceedings of the 28th ACM SIGCSE Technical Symposium on Computer Science Education*, (Feb 1997), ACM Press, 121-125.

[12] Urban, S. D., Dietrich, S. W., and Tapia, P., Mapping UML Diagrams to Object-Relational Schemas in Oracle 8, *Succeeding with Object Databases: A Practical Look at Today's Implementations with Java and XML*, J. Wiley Publishers, A. Chaudri and R. Zicari (editors), 2000.