# The Evolution of an Advanced Database Course in an Information Systems Curriculum

Behrooz Seyed-Abbassi
Department of Computer and Information Sciences, University of North Florida
Jacksonville, Florida 32224, USA

## Abstract

With more areas in the field of database achieving capabilities for practical utilization in the business environment, it is becoming more and more imperative for academic database courses to re-evaluate and consider changes in the course content as well as teaching methodologies. The significant increase in the number of topics that could be covered in a database course requires considerations for shifting topics in the introductory database course and for adding an advanced database course to the curriculum. In response to the need to provide undergraduate students with an opportunity to learn more about current technology areas, such as data warehousing, data mining, and web databases, a second database course was recently added to the Information Systems curriculum at the University of North Florida. However, teaching advanced topics in a way that incorporates practical experiences with theoretical concepts presents a unique challenge for database instructors. This paper describes the experiences encountered with the addition of a second database course and the course content presented to undergraduate Information Systems students.

**Keywords**: IS education, advance database course, emerging technology, topic ranking, assignments

## 1. INTRODUCTION

Over the last decade, the scope of database technology has dramatically expanded with the migration of many legacy systems to relational systems which has subsequently generated an explosion of innovative developments in information technology. Leading-edge technology areas, such as data warehousing, web-based applications, object oriented databases, distributed databases, and front end tools are being increasingly utilized by organizations to manage the large volume of generated data and information (Robbert 2000). As a result, database systems are rapidly becoming the underpinning that supports many areas in computing technology necessitating that many jobs in today's businesses require more than a basic background in relational database (Springsteel 2000) (Urban 2001). The need to adapt the database curriculum to provide valuable educational experiences in these areas is increasingly crucial for Information Systems (IS) graduates to be both successful and productive in the work environment.

Until recently, the IS Program in the Department of Computer and Information Sciences (CIS) at the University of North Florida offered one required database course which followed completion of prerequisite courses in file structures and data structures. The content of the course, Database Systems, has been similar to courses offered at other educational institutions (Harder 2001). After a brief review of database history, access methods and data structures, the course covered basic database foundation topics, such as database architecture, entity-relationship, relational model, normalization, relational algebra, SQL, design methodology, application interfaces with an emphasis on relational database management systems. Advanced database topics were rarely or minimally covered due to time constraints. However, as these advanced areas become more prevalent, it is essential for students to be exposed to concepts beyond the ones typically taught in a traditional, one-semester database course (Springsteel 2000).

In an endeavor to fulfill the needs of local businesses for workers with current technological skills and requests from students for further database instruction, Database Systems II (DBII) was developed in 1999. It is geared for IS students desiring to extend the basic knowledge learned in the introductory course into more advanced database areas. Since the addition of DBII, it has become a popular major elective providing students with a broader foundation of database knowledge to utilize in their current and future computing careers. This paper describes the experiences and challenges encountered in the development of the advanced database course and the gradual integration of emerging technology since its inception.

## 2. DEVELOPMENT OF DATABASE SYSTEMS II

In the Database Systems II course, the instructional goals for the students are to learn about current areas in traditional and emerging database technologies by studying the methodologies and issues in the design and implementation of such areas as relational, data warehouse, web databases, and object oriented databases. An essential part of the course is to balance the concepts with practical experiences in the utilization of software applications for database design, development, implementation, and interfaces for optimized query processing and retrieval of information. The process to achieve this balance has presented a variety of challenges. For this instructor, the primary challenge has been learning and keeping up with current and emerging technologies in a constantly changing field as evidenced by frequent textbook and software revisions. This coupled with time constraints and academic responsibilities have made the preparation of pertinent lectures, software demonstrations, and valuable assignments to support the theoretical concepts with hands-on experience a daunting task.

Initially, the content of the DBII course centered on more advanced database concepts and practical experiences to fine-tune database designs and normalization techniques in relational database with considerations for object oriented methodologies, optimization, design tools, and issues involving naming conventions, documentation, and standards. These areas were more natural extensions from the basic database course and required more in-depth utilization of database design tools (ERwin and System Architect) and database systems (Oracle and MS Access) as well as development of more complex assignments (Computer Associates 2002) (Popkin 2002). From this point, topics have gradually been added each semester. With each addition, shifts in the content have been required to present the material in an appropriate sequence for assignments and to prevent the course from becoming a software training class.

### Utilization of Questionnaires

At the end of each semester, a questionnaire is given to the students to provide feedback about the course. The information gained from the questionnaires has been valuable in the continual process to reassess and improve the content of the course. The first two questions are 1. Did the content of the course have too much overlapping with the introductory Database Systems course? and 2. If yes, please list the topics that you would like to reduce or remove. Since the first semester (fall of 1999) when 66% of the students responded that there was too much overlap, the percentage has gradually decreased to 15% in the past spring semester. The overlap figures have been affected by the gradual addition of new technology topics, as well as students entering DBII after taking the introductory database course with different instructors. To help synchronize the review component, another questionnaire has been initiated at the beginning of the semester requesting students to rank their level of knowledge (not at all, some, average, very good, excellent) in several database topics (such as SQL, entity relationship modeling, normalization...). This information is used to minimize overlapping materials and at the same time, include the review material needed to ensure that the students are at a similar level of knowledge.

Another question requests the students to consider a list of topics related to database and provide a ranking of importance for being included in the Database Systems II course. The ranking is done on a scale from 1-5 with 5 being the most beneficial. The topics include Interactive SQL, PL/SQL, embedded SQL (with COBOL, C, and Java), MS Access, basic database administration responsibilities, design and modeling, object oriented design, web database design, software (ColdFusion, WebDB, HTML, PHP) for web design, and data warehouse. Since many of the students at the University of North Florida currently work either full time or part time in technology related areas, they have a good understanding of the skills that are currently needed by local businesses. The results of the rankings since 1999 has helped to provide an overview of how various areas are perceived by graduating seniors and what areas to concentrate on in upcoming semesters.

### Results of Ranking

Figure 1 shows the collected average from student responses for 5 semesters over 3 years. The average size of the classes was 20 undergraduate, senior students. The topics that have increasingly demonstrated a very strong student interest include web database design (90.83%), data warehouse (90.4%), PL/SQL (87.7%) and database administration (81.03%). The interest in Interactive SQL using Oracle (or another database system) has also been consistently high at 80.05%. One interesting area of change has been the increasing interest in embedded Java (79.43%) and the decreasing interest in embedded COBOL (48.1%) since the addition of object oriented (OO) programming to the IS curriculum 2 years ago.

### Course Content

Over the past three years, DBII has gradually evolved from a course teaching advanced database design and implementation techniques to a course with significant components of current technology and advanced techniques. In the first course offering of DBII, object oriented database design and embedded SQL with Java were introduced. Utilization of embedded SQL with Java teaches the students about interface designs using database

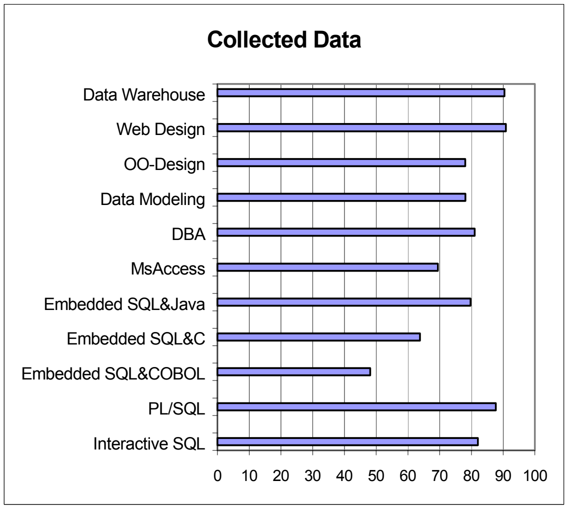

**Figure 1.** Collected average from student responses for 5 semesters over 3 years.

access by manipulation of data through an object oriented programming language. With the recent addition of object oriented programming into the curriculum, object oriented design concepts and embedded Java are currently in the process of being integrated into the introductory database course.

During the second semester of teaching DBII, PL/SQL was added to lectures. Students learned how to use SQL and database programming language directly at the server level to optimize their query processing and to write procedures, triggers, and packages for different cases and conditions during query processing in their projects.

In the third semester, data warehousing concepts including design methodologies, implementation, optimization and data mining were introduced into the course (Teorey 1999) (Seyed-Abbassi 2001). To allow adequate time to cover these concepts, the time for review material was significantly reduced. Based on information from the lectures, the students completed two data warehouse team assignments. In addition, basic database administration responsibilities and skills were also integrated into the lectures and demonstrations.

A section on database and the Internet (E-commerce) was initiated with lectures and software demonstrations during the fourth semester (Rob 2002). Students used ColdFusion to practice accessing a database on the server. Increased involvement of web utilization is anticipated as students begin to incorporate web-oriented query processing in their projects.

The continual changes and evolution of topics, such as data warehousing and E-commerce in the DBII course, consider innovations in database technology required by businesses and students' needs for current skills as well as guidelines under development as part of the IS 2002 curriculum model (Davis 2001).

At present, the following list of topics reflects the tentative schedule for the 16 week (one semester) Database Systems II course.

Major Topics Covered in the DBII Course (W=weeks)
1. Review of general database and design concepts (1 W)
2. Overview of system development and database life cycles (.5 W)
3. Advanced design diagrams (ER, EER, IE, IDEF1X, shading, UML) and design tools (1.5 W)
4. Documentation, data dictionary, and standards (.5 W)
5. Advanced SQL and PL/SQL (1.5 W)
6. Database normalization, denormalization and considerations in different designs (.5 W)
7. Data warehouse (4 W)
8. Database and the Internet (E-commerce) (2.5 W)
9. Basic DBA responsibilities (1 W)
10. Object oriented databases (1 W)
11. Team Presentations (1 W)
12. Exams (1 W)

### Technical Support

Another challenge encountered in offering advanced topics is the provision for reliable hardware and software systems to meet the course requirements for up-to-date technological experiences. The support of the lab personnel and the administration are critical for the success of an advanced database course. The classroom is equipped with a machine that is connected to the database servers in the computer labs. The lectures are presented with different presentation and demonstration tools using a projector. Lecture notes and practice problems are available on Blackboard for the students to study.

## 3. ASSIGNMENTS

The students integrate their knowledge through individual assignments and team assignments. Using a mixture of individual and team assignments provides students with the opportunity to develop personal skills in problem solving, and at the same time, provides experience in sharing responsibilities as a team member. The assignments vary somewhat semester to semester to present new challenges to each group of students and to add new components in different combinations. The following are brief descriptions of the individual and team assignments that are typically given in a semester. Due to the order of the lecture material presented and the time requirements for completion of the team projects, there is some overlap between the individual assignments and the projects.

### Individual Assignments

1. The first assignment that requires students to use the library and the web to research an area related to database

and then compile the information into a 4-5 page typewritten paper with references. The topics vary each semester. The most recent topic involved researching the history of an assigned major software company (Oracle, Informix, IBM, Sybase, Microsoft) including the direction of the company's database technology and the development of their software products. This assignment enhances research and writing skills while learning about an area in database.

Assignments #2 and #3 are geared to familiarize all the students with the available software tools for design and implementation. These assignments have a quick turn around time.

2. Assignments #2 and #3 involve the utilization of various design tools (ERwin, System Architect, and UML with Rational Rose) for the logical and physical design of a database with considerations for the more advanced design diagrams (ER, EER, IE, IDEF1X, shading, and UML). These assignments support the high interest (70.83%) expressed by students in the past 5 semesters on the questionnaire to learn and use different design tools. The first part is the development of a logical diagram using one of the tools to design a database (such as, suppliers-parts-projects, banking system, and department-faculty-course) with given requirements. The second part of assignment #2 is the conversion of the logical diagram to a physical level diagram with Oracle (or any other database) as the target database and real world values for the data type and size for each attribute name. Using the design tool, the students create the SQL code for their designs.

3. With the SQL created from assignment #2, students create an MS Access DB-space to create the tables in the database and then, insert tuples to each table for the first part. In the second part, Oracle is used to create the tables with the same attribute names and tuples with Oracle data types.

4. After lectures and demonstrations on PL/SQL, stored procedures, triggers, and packages, each student develops a PL/SQL program using cursor to insert records to a designed table, such as a checking account table. Then, the records are sorted in the table by transaction amount in descending sequence into a new table. The total transaction amounts are calculated by transaction type and stored in a report table. Students often chose to use PL/SQL in the data warehouse team assignment.

5. In this assignment, students use embedded SQL with Java to retrieve required information from database tables and to design an interface to the database. Building on OO experiences from two Java courses and the introductory database course, students learn to implement more complex queries with well-designed interfaces to the database.

6. In the database and Internet component, students learn about interface writing using Cold Fusion or Oracle WebDB for simple programs to access the database over the Internet. An assignment is given to access information in a database example (banking) by using a web browser.

### Team Assignments

During the semester, the students work in teams for two projects. The typical number of team members is 3-5 depending on the class size. Working together to analyze, develop, implement, document, and present a project provides valuable experiences in group interaction similar to what they will encounter in the work environment. The teams are determined by the students or selected by drawings depending on the decision of the class. The instructor is available for consultation through Blackboard, email, office hours, and special group meetings to help the teams in the development of the projects. After the first project, the students sometimes change teams for the final project. The following brief descriptions illustrate the typical type of team projects given during the semester.

1. For the initial team project, the students are required to design a logical and physical diagram based on the requirements for a given data warehouse. Then, they create a star schema; list and define necessary attribute hierarchies; convert to a snowflake schema using normalization and denormalization; and implement the data warehouse design using MS Access. The students are required to turn in the entity relationship diagrams, the information about the attribute hierarchies, and a diskette with the old/new tables and queries. The assignment is completed in 3 weeks.

2. Different types of final team projects have been utilized over the past five semesters. The final projects are completed in 5 weeks. For the final project, the team members submit a confidential team evaluation form on which a team member evaluates his/her performance as a team member and the performance of the other team members during the project development. The evaluation form directly affects each student's final grade.

2.a Initially, the projects involved the design and implementation of a database for a particular organization selected by the team. The teams developed the data dictionary, naming convention, valid records, and inquiry statements. The turn-ins included entity relationship diagrams; documentation for naming convention and normalization; data dictionary reports; DDL documents; create, insert and DML commands; database tables with FD diagrams; and queries with results. There was flexibility for the teams to integrate their experience with object oriented concepts, embedded languages, interfaces, PL/SQL, and web access depending on the decisions of the team. Each team presented the final outcome for their organization to the class.

2.b Another team project involved the design, implementation, mining, and documentation of several data warehouse projects. The students selected the areas for the data warehouses. The guidelines for development were based on a list of considerations that the data warehouse needed to support, such as flexibility for future expansion, capability to store detailed and summarized information, star schema with at least 6 tables, and dimensional tables of 150 tuples. The students determined the utilization of interactive SQL, PL/SQL, and embedded SQL using Java and/or other programming languages. The final submission included description of the data warehouse with its business purpose/requirements; reasoning with assumptions for tables and attributes; attribute hierarchy; entity relationship diagrams; all commands and tables; reasoning for financial, aggregated, and forecast criteria; mining queries with results; and suggestions for future improvements. The teams presented their projects to the class.

2.c Using the team projects described in 2.b, an alternative project considered modifications and enhancements to the previously designed data warehouses. The teams needed to analyze the existing data warehouse documentation and coding. Then, they applied new requirements to the design, implementation, data mining, and documentation. The turn-in materials were similar to 2.b. The teams presented their projects to the class.

## 4. CONCLUSION

The recognition of the need for an advanced database course and its addition to the curriculum proved to be only the initiation of a course that has been a rewarding challenge to teach. The gradual addition of topics to the course has enabled the instructor to prepare appropriate lectures, software demonstrations, and assignments in the areas of advanced design and database utilization, PL/SQL, data warehousing, web database design, and database administration responsibilities. The course and topics have been very well received by IS undergraduate students who are interested in continuing their study of database. Other areas that will be considered as the course develops include client-server, distributed, temporal, multimedia, and deductive databases. As technology changes, continual re-evaluation of traditional topics and considerations to shifting topics in both introductory and advanced database courses is essential to expose students to current and emerging database areas.

## 5. REFERENCES

Computer Associates: http://www3.ca.com/Solutions/Product.asp?ID=260

Davis, Gordon et al., 2001, "IS 2002: An Update of the Information Systems Model Curriculum," *International Academy for Information Management Proceedings*, December 14-16, pp. 389-397.

Harder, Joseph, 2001, "A Survey of Database Curricula Using Web-Posted Syllabi," *International Academy for Information Management Proceedings*, December 14-16, pp. 135-140.

Popkin: http://www.popkin.com/products/sa2001/systemarchitect.htm

Rob, Peter and C. Coronel, 2002, *Database Systems Design, Implementation, and Management*, Course Technology.

Robbert, Mary Ann, Ming Wang, Mario Guimaraes, and Martha Myers, 2000, "The Database Course: What Must Be Taught," . *SIGCSE Bulletin, Proceedings of 31st SIGCSE Technical Symposium on Computer Science Education*, March 8-12, pp. 403-404.

Seyed-Abbassi, Behrooz, 2001, "Teaching Effective Methodologies for Designing a Data Warehouse," *18th Annual Information Systems Education Conference Proceedings*, November 1-4, CD#35C.

Springsteel, Frederick, Mary Ann Robbert, and Catherine Ricardo, 2000, "The Next Decade of The Database Course: Three Decades Speak to the Next," *SIGCSE Bulletin, Proceedings of 31st SIGCSE Technical Symposium on Computer Science Education*, March 8-12, pp. 41-45.

Teorey, Toby, 1999, *Database Modeling & Design*, Morgan Kaufmann.

Urban, Susan and Suzanne Dietrich, 2001, "Advanced Database Concepts for Undergraduates: Experience with Teaching a Second Course," *SIGCSE Bulletin, Proceedings of 32nd SIGCSE Technical Symposium on Computer Science Education*, February 21-25, pp. 357-361.