A Team-Oriented, Project-Intensive Database Course

Hossein Saiedian & Hassan Farhat
University of Nebraksa
Omaha, NE 68106

Abstract

There are various approaches to teaching a database management systems (DBMS) course in colleges and universities. Some instructors only discuss the theory of the databases. Others may discuss the theory, design and implementation concepts but only in the traditional format of class-lectures. Teaching a DBMS course without actual database implementation, however, is as effective as teaching a guitar playing course by lecture methods only. This paper outlines a team-oriented, project-intensive approach that combines theory and practice. While the theoretical concepts of DBMS are presented in the classroom, the students are requested to form *database design teams* to apply the learned concepts during the design phases of the database project assigned to them. That is, students apply the database design concepts by actually implementing and maintaining a database for a small business enterprise. In this manner, students can encounter the idiosyncrasies of a practical database system development and reinforce many of the concepts studied in the classroom. The purpose of this paper is to review the approach we have employed in hopes that it might be useful for those who are currently teaching a DBMS course or those who are planning to teach such a course in future.

1 Introduction

Perhaps one of the most pressing problems in the effective use of computers today is organizing data to be responsive to the varying needs of users. Database Management Systems (DBMS) offer an irresistible method in solving this problem. They have been successful in providing an environment for organizing data and controlling complexity of data storage and retrieval.

It has become quite clear that during the last two decades, and perhaps long into future, DBMS will both be the heart and the binding force of all large application developments in industry and business communities. In September of 1981, Datamation reported that, the DBMS market was approximately $500 million a year and that it should experience a 25% to 30% annual rate of growth. Based on this estimate, the current DBMS market should be well over four billion dollars.

The Computer and Information Systems departments in the universities and colleges not only should be sensitive to industry's data processing need, but also should keep abreast of the industry's manpower requirements. The above statements give a strong indication that from view point of business community and industry, DBMS background and course work is desirable and perhaps expected from the new graduates in the field of Computer and Information Systems. This strong indication as to industry's preference is also evident from a study done by Archer [1]. Archer's study shows that business and industry consider DBMS as one of the most important courses in a list of common Computer

Course Title	Percentage Responding Yes to the Offering
Systems Analysis	93.5
File Processing	88.1
Database Processing	**81.0**
Data Structures	81.0
MIS	80.4
Distributed Processing	48.2
Statistical Methods	44.6
Assembly Programming	30.4
Computer Simulation	25.0
Symbolic Logic	21.4
Numerical Analysis	16.7
Linear Algebra	9.5
Calculus	4.8
Discrete Mathematics	4.8

Table 1: Questionnaire Responses

and Information systems courses offered. Table 1 summarizes Archer's findings and suggests that a course in DBMS should be an important and integral part of Computer and Information Systems curriculum.

There are various approaches to teaching a DBMS course in colleges and universities. Some instructors discuss the theory of the databases and leave it to the students to apply the theoretical concepts when employed. Others may discuss the theory, design and implementation concepts but only in the traditional format of class-lectures. Both of these approaches have the advantage that there is easily enough time to present the major concepts of DBMS. The main disadvantage of these approaches is that teaching a DBMS course without actual database implementation is as effective as teaching a guitar playing course by lecture method only.

We have been teaching a project-oriented DBMS course which combines theory and practice. While the theoretical concepts of DBMS are presented in the classroom, the students are requested to form *database design teams* to apply the learned concepts during the design phases of the database case project that is given to them during the first week of the classes. That is, the students apply the learned concepts by actually implementing and maintaining a database for a small business enterprise. The case project is developed in detail throughout the semester to ensure that all aspects of the database development process are addressed. In this manner, students can encounter the idiosyncrasies of a practical database system development and reinforce many of the concepts studied in the classroom. The students' response to this approach, which has been very positive, is reported later in this paper.

This course, which is a one-semester course taught twice a year is intended for senior/first-year graduate students majoring in Computer and Information Systems and has demonstrated quite clearly that it is possible for one instructor to teach a project-intensive course in which students engage in all phases of a typical database development. The course is ordinarily taught to students who have learned to program and who have some knowledge of common data structures.

The purpose of this paper is to review the approach we have employed in hopes that it might be useful for those who are currently teaching a DBMS course or those who are planning to teach such a course in future.

2 Course Organization

Very broadly speaking, this course is organized as follows:

1. Introductory Materials: An introduction to basic terminology and concepts, file processing *vs* database processing, intended uses of DBMS, advantages of databases, importance of *data independence*, typical architecture of most database management systems, and so forth.

2. Database Design Life Cycle: A database design life cycle is provided which serves as the basis and guideline for the students' database project. This life cycle shows the required steps for building a database.

3. Conceptual Modeling: The importance of using a high-level data model is discussed. The basic concepts of the Entity-Relationship model [3] (E-R) as well as E-R diagramming techniques are presented.

4. Logical Data Modeling: A comparison of the three major implementation models (i.e., the relational, network and the hierarchical models) is provided. Advantages and disadvantages of these important models are discussed. It is shown how each logical concept from the E-R model can be represented in each of these three models. (The network and hierarchical models are covered more extensively later in the semester.)

5. Relational Data Model: We emphasize the use of the relational model and languages, and thus this model is discussed in greater detail. The main issues facing database designers are then presented. An in depth discussion of the theory of functional dependencies, and the normalization techniques are included. (It is our experience that many of the important database concepts are best explained via the relational model.)

6. Other Database Concepts: The hierarchical and network models are revisited as well as other database concepts and issues such as backup and recovery, concurrent access to the database, security, advances in database modeling (e.g., object-oriented databases), distributed database systems, and commercial DBMS are discussed later in the semester subject to the amount of time left.

3 Why Team-Oriented?

The purpose of this course is to give students a broad background in DBMS issues as well as provide them with a "real world" experience in the design and development of a database. With these goals in mind, we feel that a team project is necessary.

During the first week of class, the course philosophy is explained to the students and they are asked to form 4-member database design teams. Each team elects a team leader who is responsible for coordinating the activities of the other team members as well as communicating with the instructor.

There are several advantages to team projects:

- Due to time constraints, most systems developed by industry are not designed and implemented by a single person. Students need to develop the co-ordinating and communication skills required in real-world environments. Thus, it is felt that students should have the experience of developing a system with team approach.
- The students must do their part of the project using another student's work. This generates a substantive amount of discussion between students.
- Because of the nature of developing database in a team environment, other issues emerge. Within this framework, students come to realize that effective communication is a must if groups and project teams are to be successful.
- Students learn that they "can't do it all" and that they have to depend on each other. They learn to recognize and use their individual strengths and expertise. They also learn to recognize each team member's strengths and weaknesses.
- Case projects tend to have "hidden" requirements which only emerge under diligent probing by team discussions.
- Students are exposed to the challenges of integrating other individuals' work into their own.

All of these benefits definitively outweigh problems involved in administrating the project.[1]

[1] One of the common problems in administrating a team project is to determine who did what, and whether all team members contributed equally. This issue is discussed later in this paper.

4 Types of Project

The students are given two project choices as explained below:[2]

1. A 'Case Project Description' describing the operations of a small business enterprise is given to the students by the instructor. The project description requires students to consider designing a database that would address the information needs (e.g., information retrieval, storage, update; report-generation, etc.) of that enterprise.

2. Design teams are responsible for selection of a database project of an appropriate size from the local business community. To ensure that the database design process does not get out of control, the students are warned not to choose a large business.

Both of the above two choices have their own advantages and disadvantages as explained below:

1. Choice 1 has the unique advantage that students have the project description in their hands and therefore do not have to make any trips off campus.

   It has the disadvantage that team members very often need to make assumptions about the operations, problems and requirements of the enterprise that are not specified in the project description. Depending on the assumptions made, the design process may be oversimplified, unnecessarily too complicated, or inconsistent.

2. Choice 2 has the advantage that the students will have to deal with a real world situation and will realize how important it is to have good communication skills to interact with users who are often ambiguous and unclear as to what they need. Thus the students will gain important real-world experience while interacting with their client and will realize that no amount of lecturing is as effective as real life episodes.

   The obvious disadvantage of this choice is that the team members have to make off campus trips during the design process.

Nevertheless, choices are made by the teams. The team members who have chosen the instructor's project will have to play the roles of both the client the designer. Usually, teams that design a database for an actual business end up more satisfied with their end-product because of its realistic nature.

5 Project Organization

Each team is responsible for designing and implementing a database for the project they have chosen. The databases are built in six phases. To provide feedback for the students, they are asked to produce an output document for each phase. Each document is turned in twice; once when it is initially created, and once at the end of the semester as parts of the completed project.[3] The instructor grades each project document and returns it to the teams along with an evaluation sheet. The evaluation sheet shows the grade for that project document and includes correction remarks, comments, and/or suggestions for improving the project. Students are told that they can recapture lost points by making the necessary revisions to their completed project. The following is a brief description of what each team turns in throughout the semester:

1. **Phase I: Requirements Analysis**

   During this phase, the teams produce a document which includes the following items:

   - A report describing the system requirements to be incorporated into

[2] We make simplifying assumptions to keep the project from getting too large. (Such assumptions must also be made in real world projects to conform to budget constrains.)

[3] It should be noted that we cover the material required for each phase of the project well ahead of time so the students have sufficient time to digest the material before they apply it in a practical setting. In addition, homework problems are assigned so students can gain experience, for example in developing E-R diagrams.

the database. In addition, this report will serve as a basis for mutual understanding between the designers (i.e., students) and their clients.

- A description of problem areas and recommendations for correcting these areas as well as information on performance requirements and preliminary design features for creating the database.
- A statement concerning the feasibility of implementing the proposed system.
- The costs involved in converting to a DBMS. The students are asked to make reasonable assumptions in stating the costs.
- A set of functional specifications describing the major functions performed in the business, including the input and output of these functions and the frequency of use, users, and other relevant information about each function.

2. **Phase II: Data Collection**

   During this phase, each team collects data into a data dictionary. The initial data dictionary for the database is turned in. Although in practice a comprehensive data dictionary contains data attributes, relations, schemas, subschemas, and reports, for students' projects we settle for the data attributes only. (Relational schemas are turned in as a separate document.) The data dictionary will of course grow in entries over the semester. Common attributes for each data dictionary item include: data name, aliases, data type, format, range, availability, security, dependencies, and comments. The data dictionary will serve as an integral tool throughout the design process.

3. **Phase III: Creating Conceptual Schemas**

   Based on the outputs of phases one and two, each team creates a conceptual schema using the E-R model. The E-R diagram most generally will change slightly as the students better understand the project. Students are encouraged to employ the enhanced features of E-R model when diagramming.[4]

4. **Phase IV: Discovering the FD's**

   During this phase, the students are required to discover and document all dependencies between data attributes. The dependencies are normally determined in conversations between students and their clients. Students are also asked to make reasonable assumptions about data dependencies and clearly document their assumptions.

5. **Phase V: Creating BCNF Relations**

   During this phase, the students create a relational database schema from the E-R diagram as well as using the data dependencies of the previous project. Once the relational schemas are determined, the relations are normalized to achieve the desired normal form, usually the Boyce-Code Normal Form (BCNF). Students are taught and are asked to use both the decomposition technique[5] as well as the synthesis algorithm [2] and compare the result of the two. Since the output of the synthesis algorithm may not be in BCNF, some teams use this algorithm to generate Third-Normal Form (3NF) schemas and then use the decomposition technique to achieve BCNF.

   For each relational schema, students define the attributes, functional dependencies (FD's), multivalued dependencies (MVD's) if any, integrity constraints, candidate key(s), and foreign key(s). Normally, each database has around 8–10 relation schemas.

6. **Phase VI: Physical Design**

   The students first create relational schemas in the chosen DBMS and then implement the database by creating empty tables, and then add data. Students test their database

[4] Enhanced features of E-R model are explained in [5].
[5] Decomposition technique is described in [11].

to make sure that they can update (i.e., insert, delete, and modify) tuples, and query the database for relevant information. Students must enforce key, entity, and referential integrity constraints. Key integrity constraint means that the primary key of each relation must always have unique value. Entity integrity constraint means that no attribute participating in a primary key should be allowed to have null values while referential integrity constraint implies that foreign key attributes should have values that match the values of the corresponding attributed in the base relation or else should have null values.

In addition to the above, each team is required to do the following:

1. Give a team-organized formal presentation to the entire class. Presentations consist of a description of the structure of the designed database and the team's approach in designing it. Instructor as well as other students are allowed to ask questions.

2. Give a team-organized demonstration of the implemented database. Since integrity constraints are of particular importance, the instructor always verifies that they have been implemented and are functioning correctly.

6 Students' Evaluation

Students' grade are determined by the following two factors:

- Classroom examinations (two intermediate tests and one comprehensive final) which count towards 50% of students' grade, and

- Database project (design, implementation, operation, and documentation of a database system) which counts towards the other 50%.

Each part of project is graded based on accuracy and completeness of its content as well as its organization (e.g., appropriate title, section and paragraph names) and appearance (e.g., consistent page numbers). Each graded part is given back to the students who then make the necessary corrections and modifications. At the end of the semester, all project parts are assembled and resubmitted as the final version of the project. The final project is once again graded for completeness and consistency.[6]

Administrating and grading a team project can be difficult. The instructor has to find ways to evaluate the performance and contribution of each individual student. A useful method that we have been using is to have each student write (1) a description of what he/she has done for the project, (2) evaluate the project by pointing out to its strengths and weaknesses and (3) briefly explain how differently the student would have approached the project if he/she had to do it again. Each student's response normally provides a good indication of how actively he/she has participated in the project.

7 Students' Responses

Students' response to this course has been very positive and most have rated the class among the most valuable course they completed. Although sometimes the students complain that there is too much work involved, they have consistently appreciated the application of the theory that they learn in the classroom, i.e., taking an entire system through design, implementation, operation, and documentation, and often have told us that this course is exactly the kind of training they should have.

8 Course Textbook

There are a number of good database textbooks. The following is a list of the books that we have directly or indirectly used for instruction:

1. Kroenke, D.M. and Dolan, K.A.: *Database Processing*, Third Edition, SRA, 1988.

[6] Students are encouraged to create professionally-looking documents, not only for their clients, but also for communication among themselves. Portfolios, labeled theme binders, etc., are recommended to the students and it is suggested that each document be proceeded by a narrative which explains the content and purpose of that document.

2. Korth, H. and Silberschatz, A.: *Database System Concepts*, McGraw-Hill, 1986.

3. Date, C. J. : *An Introduction to Database Systems*, Volume I, Forth Edition, Addison-Wesley, 1986.

4. Elmasri, R. and Navathe, S.B.: *Fundamentals of Database Systems*, Benjamin/Cummings, 1989.

Although the first three books are well-established database textbooks, we find the fourth book (by Elmasri and Navathe) well-organized and well-written. Elmasri and Navathe cover the material in the order and depth needed at this level as well as covering the state-of-the art topics in databases and provide a comprehensive reference on this field. We strongly recommends this book to other database instructors.[7]

9 Conclusion

The database course described in this paper represents an attempt to develop a meaningful and practical experience for the students in the database course. The database project is paced so that it parallels the classroom lectures and provides immediate feedback to the students. The development of this course has been a real learning experience for us and we hope that this paper will provide a few suggestions to those who are or will be teaching a DBMS course.

References

[1] Archer, C.B.: "*What Does Business and Industry Expects from Computer Science Graduates Today?*," *ACM SIGCSE Bulletin*, Vol. 15(1), (Feb. 1983), pp. 82-83.

[2] Bernstein, P.: "*Synthesizing Third Normal Form from Functional Dependencies*," *ACM Trans. on Database Systems*, Vol 1(4), (Dec. 1976).

[3] Chen, P.: "*The Entity-Relationship Model — Toward a Unified View of Database*," *ACM Trans. Database Systems*, Vol. 1(1), (March 1976), pp. 9-37.

[4] Date, C. J.: *An Introduction to Database Systems*, Volume I, Forth Edition, Addison-Wesley, 1986.

[5] Elmasri, R. and Navathe, S.B.: *Fundamentals of Database Systems* Benjamin-Cummings, 1989.

[6] Etlinger, H.A. (Review): "*Fundamentals of Database Systems*," *IEEE Computer*, Vol. 23(2), (Feb. 1990), pp. 118-119.

[7] Korth, H. and Silberschatz, A.: *Database System Concepts*, McGraw-Hill, 1986.

[8] Kroenke, D.M. and Dolan, K.A.: *Database Processing*, Third Edition, SRA, 1988.

[9] Orr, K.: *Structured Requirements Definition*, Ken Orr and Associates, 1981.

[10] Ross, D.: "*SADT*," *IEEE Trans. Software Engineering*, Vol. 3(1), (Jan. 1977).

[11] Ullman, J.: *Principles of Database Systems*, Second Edition, Computer Science Press, 1982.

[7] An excellent review of this book by Henry A. Etlinger appeared in [6].