

The Pros and Cons of Using a Comprehensive Final Case Project in a Database Management Systems Course: Marvin's Magnificent Magazine Publishing House

Terri L. Lenox
Mathematics & Computer Science, Westminster College
New Wilmington, PA 16172 USA

Charles R. Woratschek
Computer Information Systems, Robert Morris University
Moon Township, PA 15108 USA

ABSTRACT

There are many challenges in providing a curriculum with a solid Information Systems foundation that meshes with the rapid changes in technology and its use within organizations. Educators must struggle to fit all the necessary information into a limited number of credits while continuing to add skills including soft-skills. One particular area of pressure is the need to expand topics in the typical database management course due to the increased importance of databases in organizations, the tremendous volume of data that must be handled, non-traditional types of data (e.g., multimedia, web-based) and the expanding array of database-related tools. This paper discusses the pros and cons of using a comprehensive database project as the culmination of an introductory course in database theory and design. An instructor-created, team database project is described. Marvin's Magnificent Magazine Publishing House database has provided students with a valuable experience on the four most popular database topics: the relational data model, SQL, the entity relational model, and normalization.

Keywords: database, curriculum, project, team-approach

1. INTRODUCTION

There are many challenges in providing a curriculum with a solid Information Systems foundation that meshes with the rapid changes in technology and its use within organizations. Educators must struggle to fit all the necessary information into a limited number of credits while continuing to add skills including soft-skills. One particular area of pressure is the apparent need to expand topics in the typical database management course. Data management is a \$10 billion dollar industry (Riccardi, 2003) and has rapidly changed from when ACM first published the curriculum for computer science in 1968. Today, educators are faced

with the need to handle terabytes of data that takes a range of forms from graphical to audio and from many sources including web-based data. In addition, data mining, data warehousing, and e-commerce are advanced topics of interest to both students and employers.

With the publishing of Curriculum '68 in March 1968 by the ACM, Computer Science became a distinct discipline from Mathematics (ACM, 1968). The curriculum was divided into information structure and processes, information processing systems and methodologies. One topic in methodologies included: "Data Processing and File Management: includes techniques application to

library, biomedical, and management information systems; file processing languages" (ACM, 1968). By the mid-1970s, the importance of databases in business was well established and documented. Therefore, it is of no surprise that some type of database course has been a long standing component in Information Systems (IS) and Computer Science (CS) undergraduate and graduate curricula. Recent surveys also indicate that database skills are considered very important by educators, students and employers (Janicki et al, 2004; Robbert and Ricardo (2003); Woratschek and Lenox, 2002; Abdullat, 2001).

The importance of including a database course in the curriculum is present even now with the emergence of Information Technology as a separate discipline. Ekstrom and Lunt (2003, pg. 198) state "...there is an emerging consensus that the core technical areas (pillars) of Information Technology are: 1) Programming; 2) Database design, deployment and management; 3) Web systems, design, deployment and management; 4) Network design, deployment and management; and 5) Human Computer Interfacing, including user advocacy."

2. CONTENTS OF THE DATABASE COURSE

Although there seems to be agreement that a database course is essential in Computer Science (CS), Information Systems (IS), or Information Technology (IT) programs, there is generally no consensus as to the name of the course or its content. Course content varies due to the specific curriculum (CS, IS, or IT) and other factors including: "program philosophy, the program emphasis, students' preparation and background, faculty background, textbook availability, availability of technical resources, tools, and the needs of the local community" (Abdullat, 2001).

The IS model 2002 curriculum for undergraduate degree programs in Information Systems sponsored by ACM, AIS, and AITP (Gorgone et al, 2002) details the following capabilities and knowledge expected for database design and administration as: 1) "Modeling and design, construction, schema tools, and DB Systems; 2) Triggers, stored procedures, design and development of audit controls; and 3) Administration: security,

safety, backup, repairs, and replicating" (pg. 14).

The IS model 2002 curriculum also defines a database course: IS 2002.8 – Physical Design and Implementation with DBMS. "This course covers information systems design and implementation within a database management system environment. Students will demonstrate their mastery of the design process acquired in earlier courses by designing and constructing a physical system using database software to implement the logical design" (pg. 30). The topics include: "conceptual, logical, and physical data models, and modeling tools; structured and object design approaches; models for databases: relational and object oriented; design tools; data dictionaries, repositories, warehousing, and data mining; database implementation including user interface and reports; multi-tier planning and implementation; data conversion and post implementation review" (pg. 30).

Although the IS model 2002 curriculum provides the general guidelines, it does not address the issue of how to meet these guidelines. Issues arise concerning whether to include topics such as: object-oriented databases and the ODMG standard (Neubauer, 2001; Springsteel, 2000; Schahczenski, 2000), XML (Wagner and Moore, 2003; Neubauer, 2002), UML (Neubauer, 2002), data mining and data warehousing (Slazinski, 2003), and web-based databases (Mehta, 2002; Springsteel, 2000). For those programs lucky enough to be able to offer a second database course, some of these topics can be shifted to it. However, since many programs are limited to just one database course in the curriculum, the difficult task of topic elimination is a necessity.

Robbert and Ricardo (2003) reported three different surveys of educators regarding the most important topics taught in their database course. In 1999, 34 educators identified SQL (76%), database design (47%), and the entity-relational model (35%) as the three most important topics covered in their database courses. Object-relational databases (29%), normalization (26%), and relational algebra (21%) were next. In a second survey done in 2001, 106 educators were asked to specify the number of hours they devoted to each of a list of database

topics. The topic given the most time was SQL. Database design, the entity-relational model, the relational model, and the normalization of database tables were next. If the topics were counted, the topics named by more than 90% of the 106 respondents as being part of their database courses were relational database model, SQL, the entity-relational model, and the normalization of database tables. In a third survey done in 2002, 35 respondents stated that the most time was spent on relational data model, entity-relationship model, SQL, database design, and client tools. The most commonly covered topics were relational data models (97%), SQL (91%), and entity relational model and normalization of database tables (tied at 86%).

Abdullat (2001) identified eight topics to be covered in a one-semester database course. These topics include: database model and file systems, relational database model, data modeling using Entity Relational Model and the Semantic Model, structured Query Language, database design and normalization, transaction management and concurrency control, distributed database, and database and the Internet applications.

Other authors speak to the course content and the pedagogy used. Urban and Dietrich (1997) discuss the difficulties involved in including commercial database software in a course. Valuable time must be spent teaching students how to use the software without reducing the course to a software training experience. Springsteel (2000) reported that in DBMS "...the controversial areas besides DBMS platform, were Student Expectations: what do we expect the student to be able to do as a result of passing this course? Almost all courses assign some kind of Database Project; the devil is, as always, in the details: ten gave one project for all student or teams to do, while six gave project choices, and three used a mixed strategy!" (pg. 42).

3. DATABASE COURSE PROJECTS

A common pedagogy is the use of a course project to complement the database theory (Urban & Dietrich, 1997; Seyed-Abbassi, 1993; Abdullat, 2001; Saiedian & Farhat, 1991; Robbert, 2000; Baugh, 2004; Dietrich and Urban, 1996; and Tuttle, 2002). Rob-

bert (2000) stated that "Designing a project that meets the criteria is important and it is equally important to allow sufficient time for reflection" (pg. 36). However, not all database courses include this experience. Robbert (2000) discovered through SIGCSE surveys that 36% always contain a team project, 34% sometimes contain a team project and 30% never contain a team project.

In general, most courses employing a database project divide it into a series of phases, usually ranging from three to nine steps. Although the number of phases varies, the phases appear to be similar (e.g., requirements collection, design, transaction processing, testing and evaluation). Two general approaches to incorporating a project are reported: individual projects and team projects. From a review of the literature on these projects, it appears that the total amount of work is similar; even though in a team project, this work is divided across the members of the team. It is not clear whether the weight of individual and team database projects is equal in these courses or whether the instructors give more weight to individual projects than team ones. It is also not clear whether or not the expectations for these projects differ.

Individual Projects and Evaluation Methods

The use of individual database projects has several strengths. Students, particularly strong students, often prefer to work on their own without the communication and coordination issues that a team project entails. In addition, it may be easier for the instructor to assign an individual project score rather than determining who did what on a team project. Finally, the student must perform all database design and implementation tasks on their individual project.

Tuttle (2002) effectively used the idea of a project notebook for her student projects. The student kept a notebook with the following sections: proposal section, data model section, business rules section, database schema section, database population section, example queries section, example reports section, discussion section, and miscellaneous section. These sections acted as milestones for due dates, project evolution, and final deliverable. She also reported on

the value of scheduled, well-defined milestones.

Baugh (2004) also used an individual project approach in her database management course. The student must first complete a database project proposal describing: 1) the central focus of the database; 2) why this topic was selected; 3) the types of information the database will provide for the user; and 4) if the project being created for the student's own use or another's use. Baugh used a two-phase approach. In the first phase, Requirements, the tables, keys, constraints, forms, buttons, lists, and relationships were defined and implemented by the student. In second phase, the student produced reports, queries, updates, macros, and pop-ups. Students were also required to produce additional documents: ER design, Microsoft Access relationships, report and query explanations, data dictionary, user specifications and table documentation. Baugh reported a high level of enthusiasm from her students concerning the database projects and a desire to have an advanced database course in the curriculum.

Both Tuttle and Baugh reported great success with their approaches. It is important to note that they also spoke of the tremendous time allotment needed to provide feedback and evaluation on the individual projects.

Team Projects and Evaluation Methods

By far the more common approach is using a team project in the course. Again, there is a great deal of variety as to the approach, pedagogy, and evaluation of the team project. Team projects have several advantages to individual projects, particularly with their emphasis on cooperative learning, communication, and group dynamics (Dietrich & Urban, 1996). They can be less time-consuming for the instructor to evaluate the overall project soundness, but more difficult to determine how to assess individual performance on the tasks.

Saiedian and Farhat (1991) gave their student teams two options for projects. Either the team was given a case project description by the instructor or they had to find an appropriate database project from the local business community. Regardless of the

choice, the teams built a database in six phases. For each phase the team was required to produce an output document. This document was turned in twice; once when it was initially done and once at the end of the semester. Additionally each team was required to give a team-organized formal presentation on their data to the entire class and they had to demonstrate the implemented database. Saiedian and Farhat believe that a team-oriented approach provides students with a "real world" experience that demonstrated the importance of coordination, communication, and cooperation. They also point out that teams were better at finding the "hidden" project requirements and can benefit from their teammates' strengths.

Saiedian and Farhat (1991) admitted that it is difficult to grade a team project. To evaluate the performance and contribution of individual team members they asked each student to: 1) write a description of what s/he has done for the project; 2) evaluate the project by pointing out its strengths and weaknesses, and 3) briefly explain how differently the student would have approached the project if s/he had to do it again.

Leeper (1990) used a team project in a second database course. While the team concept was used for the course project, each member of a team was assigned a specific part of the overall project to complete. Each member's part was integrated with other parts, so coordination of effort was needed to produce a workable system. No two teams did exactly the same project. A student's grade was determined in part by his/her individual work, and in part, by the success of the entire team project.

Sayed-Abbassi (1993) used an instructor-provided case for the database project. The project was divided into four parts and each team was allotted 15-20 minutes to present their design to the entire class. The team's presentation included a description of the created tables with their reasoning for the relationships created between the tables in their database. The other members of the class evaluated the team's presentation using an instructor-created template. An additional evaluation tool was a written report submitted by the team which included: 1) a report on design of the tables; 2) referential

integrity considerations; 3) reasoning on semantic dependency (add/delete); 4) a description of changes(s) to the existing tables; 5) query processing; 6) an outline of the presentation; and 7) an evaluation of each group member and their participation level.

Abdullat (2001) allowed teams of students to choose their own 'real world' scenario for their database project. The team produced a final project document with three phases (definition, design and implementation).

In another approach, Urban and Dietrich (1996) had students identify the specific application to be developed for the course project including an expert in the application area. They also used the concept of cooperative learning. The project was divided into three phases with specific assessment and technical deliverables. The assessment included group status reports, team assessments and confidential evaluations. In this way, students evaluated their own performance as well as the performance of others. Additionally, in each phase students were assigned different roles (phase leader, phase recorder, phase checker or phase technical advisor) to experience different responsibilities. Urban and Dietrich found that intermediate deliverables provided an important method of feedback to the students before they complete the project and improved the final deliverables. The authors also found that group status reports at the end of each phase provided timely and sufficient feedback.

With a different methodology, Robbert (2000) used both an individual and team approach in her database management project incorporating three components and an additional reflection element.

4. MARVIN'S MAGNIFICENT MAGAZINE PUBLISHING HOUSE PROJECT

To give database students practice and experience in the four most popular database topics: the relational data model, SQL, the entity relational model, and normalization, a comprehensive project was developed by one of the authors some ten years ago entitled "Marvin's Magnificent Magazine Publishing House." This project was created from the authors' personal consulting experiences

in the business community. At Robert Morris University, this project has been used in the undergraduate senior level introductory database course and just recently in an eight-week graduate database course. In both cases the students were requested to work in groups and the project was used in lieu of a final examination. At Westminster College, this project has been used several times in a 16-week undergraduate course. Students may or may not work in groups depending on the size of the class.

Students are presented with a scenario about Marvin's Magnificent Magazine Publishing House and a database that needs to be designed and implemented. Students are provided with incomplete data for four different entities: magazine, customer, advertising costs, and advertisers. Magazine data consists of the magazine name, interval, price and newsstand price. Subscriber data includes the customer name, address, magazine and the month the subscription ends. Advertising costs includes the page size and cost of an advertisement in each of the magazines. Advertiser data consists of the name, address, telephone numbers and e-mail addresses of the advertisers.

Students are also told that no one at Marvin's has any database experience. In the course(s) at Robert Morris University, students were required to implement their database using Oracle. These students have little or no exposure to Oracle before starting this project, but they have used Microsoft Access for several course assignments. At Westminster College, students were given the choice of Microsoft Access or Postgresql to implement their projects. They have used both of these software packages in the course prior to the final project.

Finally, students are given nine transactions that their database design must minimally satisfy:

- How many people subscribe to HOUSES?
- How many subscriptions have expired?
- How much is any given subscription?
- How much money did xxxxx pay in total subscription prices?
- How much money did Whirlpool spend in advertising in HOUSES this month? This year?

- xxxxx wishes to drop his/her subscription immediately, how much money is s/he entitled to as a refund?
- How many people subscribe to all three magazines that Marvin publishes?
- I know that one or more of our advertisers have the word LIGHTING in their name, I just can't remember the exact name of the company. Can you supply the exact name?
- Of the people that subscribe to xxx how many live in zip code xxxxx?

The data provided to the students includes some ambiguities as would readily be found in a "real world" project. For example, page size data is provided as words (whole, 1/2 page, etc.). However, the optimal implementation would be to use numeric values instead. Another ambiguity is that no actual advertisement data is given and the fact that multiple advertisements from the same company can appear in any magazine for any month(s) is not stated.

Project Goals

In designing this database project, the four most popular database topics covered in the majority of database courses as discussed in the literature review were used as a starting point. Because most 'real world' business database design projects do not include every entity and/or transaction needed to successfully design a given database, the same was done in the Marvin project. The ambiguities were chosen for Marvin to stress the importance of including appropriate primary keys, dates and timestamps in the database design, and creating entities and appropriate test data that are not obvious from the problem description. The team design approach is favored for the project because it mimics the way the majority of actual databases are designed and more importantly, it allows for creative problem solving and group discussion about the missing pieces of the problem.

Project Requirements

Each group is required to produce a deliverable at their final oral examination:

- A complete conceptual model of the database, an Entity-Relationship Diagram, and a BACHMAN or ERD diagram

- A description of the normal form of each entity in the database and an explanation as to why that entity is in that normal form
- All actual database files and logs of the creation of the database
- All actual database files and logs of the creation of primary keys and constraints
- All SQL queries and results that satisfy the nine transactions above
- Additional transactions created by the group and the SQL and results to answer these queries,
- Four ORACLE reports which use formatting functions based on transactions created by the group.

During the required oral demonstration of the database project, students are asked both group and individual questions:

- What normal form is your database in and why?
- Describe the groups' design technique (include how you picked candidate keys, PKEY, foreign keys, etc.)
- What did the group decide to index and why?
- How many design iterations has your group done?
- What have **you** changed in your design? What led to these changes?
- Describe the groups' test plan and give some examples of transactions that were used to test.
- What would **you** change (if anything) about your current design? Why?
- Did designing in a group help or hinder the process? Are there ideas/problems that were discovered that you would have not found, had you been working individually?

These questions required all members of the group to know the database design, its normal form, the problems encountered by the group in the design, the actual test plan used to achieve the final design, and the transactions used to guide and test the design.

In addition to group questioning, each individual is required to demonstrate SQL queries using the groups' database design:

- What percentage of all your customers subscribe to HOUSES?

- How much money did xxxxx spend on xxx page advertisements this year?
- Has Kelly Finnegaan renewed her subscription to RD magazine?
- Of all the people that are our customers, what is the most popular zip code?
- How much money does Tracey Skiman save a year by having a subscription with us for HS versus what she would pay for the magazine from the newsstand?
- How many advertisers took out xxxxx page ads in any of the magazines and what was the total amount they spent?

Instructor Observations about the Group Portion of the Project

An important component of any group project is the group's ability to coordinate, communicate and cooperate. Obviously, this communication is important among the members of the group, but it is equally important that the students communicate with the end user of the database project (e.g., employees of Marvin's). The authors have observed that after the students had been given the project and the groups formed, some groups were willing to query the instructor about the Marvin's Magazine Publishing House. In general, these groups were more successful designing and implementing the project. Other groups did not ask questions concerning the magazine domain.

During the project demonstration, it was found that students were very honest in their self-evaluations. They spoke about group discussions, how they actually went about designing and implementing the project, and the problems that they encountered. The students also were very verbal about their individual and group strengths and weaknesses. To accomplish the project work, many Robert Morris University groups used e-mail. E-mail also acted as a problem-solving environment for some groups who could not meet face-to-face due to commuters, business travelers or members who were otherwise unavailable. The majority of groups reported that e-mail was not sufficient; they needed to meet face-to-face to complete the project. Westminster students tended to communicate face-to-face when necessary.

The majority of the groups commented that the work of the project was divided amongst the team members and this made the project less monotonous and easier. Communication and collaboration was definitely the key to successfully completing the project. The time allotment and the complexity of the project allowed for no slackers. However, many groups realized at the completion of the project that all members needed to have a complete understanding of the entire project. Several of the groups realized too late that more communication among members about specific tasks was necessary for this understanding.

In general, it was also observed that students do not think through the problem carefully. They assumed that what details were given about the project were the only details necessary to successfully complete the project – they did not see the bigger picture. Some specific troublesome areas were test plans, dates, expired subscriptions, and multiple advertisements.

Surprisingly, when questioned about their particular test plan, some groups reported that they forgot all about developing a test plan. They simply used the nine given transactions, and never thought about trying extra queries to test their design (even though that was a specific requirement). The majority claim they were so wrapped up in the details of getting the nine transactions right that they focused on nothing else.

In addition, many groups admitted that they did not consider dates in their database design. Although dates can be a difficult concept, this material had been covered in lectures (and some assignments) prior to the project. These groups were surprised and/or upset when questioned about the date when an advertiser ordered (a) particular advertisement(s) for a particular magazine or the date when the actual advertisement will appear in a particular magazine. Typical comments and questions such as "I didn't think we had to worry about that, " "Don't you place a request for an advertisement and pay for it the same month it appears in the magazine?" and "Why would you need to know that for?" were some typical responses made by the members of the groups.

Another issue that many groups failed to consider was individuals whose subscriptions had expired. Again, most groups were surprised and/or upset when they were told that they included these individual(s) in their transactions. We found little "reality checking" with respect to expired subscriptions which may be related back to the students' difficulties with dates.

The notion of multiple advertisements for the one advertiser seemed to be difficult for most groups. A query such as "How much money did Whirlpool Corporation spend on ¼ page and ½ page ads last year in Remodeling Digest Magazine?" was typically met with blank stares or comments such as "We can't answer that!" or "We didn't design for that!" Also, the majority of groups never thought about the fact that an advertiser, such as Sears, could advertise more than one item in the same magazine in the same month. It is not clear to the authors if this failing related to students' depth of understanding of the domain of the project or database design, but believe it is due to a little of both.

Overall, the authors believe that most undergraduate group projects were successful. The difficulties that the students encountered can be attributed to their inexperience with database design or with the software, and in some cases their lack of interest in a project that is not 'real.' Surprisingly a greater number of graduate projects failed. The authors believe that the graduate students had more time-management problems due to the eight-week course format and that many of them were enrolled in several courses while being employed full-time.

Instructor Observations About the Individual Portion of the Project

At the end of the group oral examination, each individual is given a set of SQL queries to demonstrate. The student is permitted to use any materials that s/he wished. However, s/he is not permitted to collaborate with anyone. The student is given the remaining class period time after their group oral evaluation to complete these queries.

Generally, students spent more than one hour on these queries, unless they had prior experience in SQL, databases or were ex-

perienced programmers. Generally, the individuals who were Computer Science majors or Information Systems individuals with extensive programming backgrounds performed better on this task. Some individuals actually turned in blank answer sheets in less than 20 minutes time, writing statements like "it took me two or three hours to figure out some of the queries for our database design - I can't do this in a short time," "I'm not a programmer," "I hate programming," or "I hate SQL, and I can't possibly do these by myself - I did all the others with help."

Another observation is that many students failed to check the underlying database or lacked an in-depth understanding of the data. At the mere appearance of output, they assumed that the output was correct (e.g., the refund question, bimonthly vs. monthly publications). Unfortunately, this is not an uncommon problem in many programming situations.

The authors believe that the queries requested during the individual evaluation of this project are not difficult. However, many students reported feeling great pressure and insecurity during this exercise. This was most evident in the students whose team had divided the work amongst the members. Students who were not responsible for their team's SQL statements regarding the nine transactions really struggled.

5. CONCLUSIONS: HOW DID MARVIN DO?

Although a single database project cannot hope to include all the possible topics in the database field, the authors believe that there is no substitute for a hands-on project. To fully understand the underlying theory, students must have the opportunity to design a project from start to finish. Whether or not this project is an individual project or a team project is debatable. The authors strongly believe that a team-approach has many advantages particularly to introduce students to the current industry practices. A team approach seems to be more practical to the authors. The amount of instructor effort required to support and evaluate individual database projects can often seem overwhelming, especially in high-enrollment settings.

The team project provided by Marvin's Magnificent Magazine Publishing House database allows our students to:

- Experience the joys and sorrows of working in teams.
- Deal with a greater level of ambiguity than found in smaller course assignments.
- Work with a database project from the almost its inception to its conclusion.
- Work with issues involved in multi-user access to a database (e.g., security, backup, recovery, and concurrency).

Another choice to be made is whether to allow the students to select their own projects or to use an instructor-created project. The advantages to an instructor-created group project are less clear, but may include:

- Known risks, workload and time-to-complete.
- The ability of the instructor to select the appropriate difficulty and amount of ambiguity in the project.
- A greater amount of control over the project, its deliverables and the consequences of failure. In contrast, some outside client-based projects reported problems with expectations for the project, time constraints, work-load issues, and privacy/security concerns.

On the other hand, student-selected group projects can create a high level of enthusiasm from the students. If the database will be actually used, this is a powerful incentive for the students. On occasion, the authors have encountered some problems with the scope, work-load and difficulty level of student-selected projects which make them wary of these types of projects.

Marvin's Magnificent Magazine Publishing House database problem has provided students with a valuable experience on the four most popular database topics: the relational data model, SQL, the entity relational model, and normalization. Additionally, it stresses to students the importance of thoroughness in the logical design of a database. By using a team-approach, the authors believe their students also gain insight into industry practices.

6. REFERENCES

- Abdullat, A. A. (2001). Teaching a database systems design course: Is it theory or practice? *Proceedings of ISECON 2001*, Cincinnati, OH.
- Baugh, J. M. (2004). A first course in database management. *Proceedings of ISECON 2003*, San Diego, CA.
- Dietrich, S., W. and S. D. Urban (1996). Database theory in practice: learning from cooperative group projects. *SIG-CSE '96*. Philadelphia, PA. February 1996.
- Gorgone, J. T., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L. & Longnecker H. E., Jr. (2002). IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. *ACM, AIS and AITP*.
- Janicki, T. N., D. Kline, J. A. Gowan, & R. Konopaske (2004). Matching employer needs with IS curriculum: An exploratory study. *Information Systems Education Journal*, 2(21), April 7, 2004.
- Leeper, R. (1990). A project course in database. *ACM 1990*, 78-80.
- Mehta J. (2002). Should we try to squeeze database-packed websites into the curriculum – and if so, where? *Consortium for Computing Sciences in Colleges*, 77 – 84.
- Neubauer, B. J. (2001). A strategy for the integration of object-oriented data modeling into the undergraduate course. *Journal of Computing in Small Colleges*, 16 (3), 240-246.
- Neubauer, B. J. (2002). Data modeling in the undergraduate database course: adding UML and XML modeling to the traditional course content. *Consortium for Computing Sciences in Colleges 2002*, 147-153.
- Riccardi, G. (2003). Database Management with Web Site Development Applications, Addison Wesley.

- Robbert, M. A. (2000). Enhancing the value of a project in the database course. *SIGCSE 2000*, Austin, TX, 36-40.
- Robbert, M. A. & C. M. Ricardo (2003). Trends in the evolution of the database curriculum. *ITICSE '03*, June 30 – July 2, 2003, Thessaloniki, Greece, 139-143.
- Saiedian, H. and H. Farhat (1991). A team-oriented, project-intensive database course. *ACM 1991*, 192-198.
- Schahczenski, C. (2000). Object-oriented databases in our curricula. *Consortium for Computing Sciences in Colleges 2000*, 174-181.
- Seyed-Abbassi, B. (1993). A SQL project as a learning method in a database course. *ACM 1993*, 291-297.
- Slazinski, E. D. (2003). Teaching data warehousing to undergraduates – tales from the warehouse floor. *CITC4'03*, October 16-18, 2003, Lafayette, IN, 242 – 248.
- Tuttle, S. M. (2002). Practical lessons from experience with the database design course project. *Consortium for Computing Sciences in Colleges 2002*, 32-42.
- Urban, S. D. and S. W. Dietrich (1997). Integrating the practical use of a database product into a theoretical curriculum. *SIGCSE 97*, 121- 125.
- Wagner, P. J. & T. K. Moore (2003). Integrating XML into a database systems course. *SIGCSE'03*, February 19-23, 2003, Reno, NV, 26-30.
- Woratschek, C. R. & T. L. Lenox (2002). Information systems entry-level job skills: a survey of employers. *Information Systems Education (ISECON 2002)*, San Antonio, TX, October 31 – November 3, 2002.