Journal of Information Systems Education
Volume 5, Number 2

THE DATABASE PROJECT: MAXIMIZING ITS VALUE

Dr. Bruce Rollier
Assistant Professor
Robert G. Merrick School of Business
Information & Quantitative Sciences
University of Baltimore
1420 North Charles Street
Baltimore, Maryland 21201-5779

ABSTRACT: The demand for employees with useful knowledge and skills for operating in a DBMS environment is increasing. Students must acquire technical database skills, but technical expertise is not sufficient; they must also be knowledgeable in other functional areas of the business, so that they will be competent to determine information needs of users. A carefully designed database project can facilitate deeper student understanding of important database concepts, while at the same time helping them to relate the textbook concepts to a simplified but realistic business environment. The goal is to provide them with a set of skills which will enable them to be productive but also adaptable to a wide range of business situations. This paper discusses the learning objectives of the project as well as other operational, pedagogical, and administrative issues.

KEYWORDS: Database, DBMS, Curriculum, Teaching Methods, Project

INTRODUCTION

Data base management, or some aspect of it, is at or near the top of most lists of important MIS issues. In a major Delphi study with senior information systems executives [1] it was found that "Developing an information architecture" ranked at the top of the list of key information systems management issues for the 1990s, and "Making effective use of the data resource" ranked second. These same issues ranked eighth and seventh respectively in a similar study conducted three years earlier [2], so they have moved up rapidly in importance. In the increasingly turbulent environments of recent years, organizations have found that rapid response to unpredictable events is critical to survival [3]; well-designed database systems can team with telecommunications to provide the organizational flexibility to change strategic directions as the need arises [4].

As more organizations implement database management systems, the demand for employees with useful knowledge and skills for operating in a DBMS environment is increasing [5, 6]. Such skills are of course essential for employees in database administration (DBA) departments, but also increasingly for systems analysts, application programmers, and others in the MIS function. A recent survey [7] indicates that familiarity with database principles is also required for end user liaison personnel, and that the requirements are not satisfied by the typical "Introduction to Information Systems" course.

To prepare students for such environments, most colleges and universities that provide a major in Computer Information Systems (CIS) or Management Information Systems (MIS) require a course in Database Management Systems for that major. In the IBM Management of Information Grant Program Syllabus Book [8], syllabi are included for introductory database courses at ten major universities; all of the courses require a term project that accounts for at least 20% of the course grade. It seems to be well-recognized that just teaching database theory is not adequate; the students need the hands-on experience of the project to make sense of the textbook and the lectures.

This paper explores some of the major issues of database project design. The emphasis is on the questions rather than the answers, since design decisions are contingent on the situation in the specific academic program. Limited availability of hardware, software, and telecommunications facilities may leave the course designer little choice. The answers may be different for graduate vs. Undergraduate programs, and for commuter schools vs. full-time programs. The most important questions, however, should be addressed for all database classes. Our own program will be described, and the design choices that we have made will be discussed, but this does not imply that our answers should be applied in other situations. The focus is on the choices that must be made in maximizing the project's value to students, not only in obtaining their first job but also in their subsequent careers.

At our university, the database course is required for undergraduate majors in CIS and for the Masters program in MIS, both within the School of Business. The graduate course is more rigorous, but the structure is the same at both levels, and most of the issues of project design are the same. Both courses require the implementation of a multi-table relational database, and extensive hands-on experience with SQL. The emphasis in both is at the conceptual and logical levels; there is some material concerning physical database, but much less than in a typical computer science course. All the projects are performed on the school's VAX using RDB, a relational DBMS from Digital Equipment Corporation. RDB has an excellent SQL interface, and provides support for domains and for referential integrity. For students who own computers and modems, it is feasible to perform much of the work from home.

All classes are given SQL training before commencing the project. Early in the semester, they are required to access a 6-table database on the instructor's account and perform about 20 queries. They are also exposed to normalization and database design prior to starting the project. About halfway through the semester, they are given a short case to read that describes the requirements for a database in a particular industry. One homework assignment and one class period are devoted to exercises for identifying entities, attributes, and relationships, and culminating in a set of relational tables. All students then implement the logical design we have jointly developed, typically in groups of three (undergraduate) or two (graduate).

After implementing the database in RDB, all students are required to prepare a comprehensive report of the experience, which counts as 40% of the project grade. Since there is little difference between the completed databases, the report is the main source for differentiating between students and groups. The major purpose of the report is to encourage the students to think of the organization for which they are implementing the database as a real-life company, and to use their creativity to think of additional attributes and of enhancements to data entry procedures, integrity, security, and performance. There is an executive summary addressed to the chief executive; a history of the project describing division of responsibilities, a time summary, analysis of major problems encountered, and an evaluation of the DBMS. The last section of the report asks each student in the group to write a brief summary of the value derived from the project and for recommended changes, providing feedback for course revisions in the succeeding classes.

> ***As more organizations implement database management systems, the demand for employees with useful knowledge and skills for operating in a DBMS environment is increasing [5, 6].***

DATABASE PROJECT DESIGN ISSUES

Careful planning of the project is a wise investment that will pay dividends in time saved and in satisfaction gained for the instructor as well as the students. The project should always be viewed as an integral part of the course, not just something added on, and the planning for both should be performed before the semester starts. There should also be some flexibility, however; what works well for one class may have to be modified for another. Design issues include learning objectives, operational decisions such as hardware and software, pedagogical decisions such as choice of database model, specifics of the tasks to be performed, and dynamics such as how the groups are assigned, organized, and monitored.

Learning Objectives

As in any design process, the first step should be to establish objectives, in this case learning objectives. What do we want the students to gain from participation in the project? It will take a substantial amount of their time; how can we ensure that it will be a meaningful experience and not just "busy work"?

It is our view that the project objectives should be the same as the overall course learning objectives. The project should not be designed to teach new and different things that are not covered in the classroom. It should reinforce and enhance the learning that has been initiated by lectures, textbook, and exercises. A complex concept such as referential integrity or rollback is not "learned" just by memorizing the definition. Learning is a process, not a product. A well-designed project can provide a level of comprehension of the more difficult database concepts that would not be achieved in a lecture alone. The understanding will be retained in long term memory, rather than quickly forgotten after the examination. This does not mean, however, that it is "learned" in the project. It is the combination of lecture and hands-on experience that produces real learning.

Our principal objectives are that students have:

- Awareness of the problems encountered and techniques used in identifying information to be collected, and in actually collecting the necessary data. This includes discussion of how to determine what data are needed, interviewing techniques, internal and external sources of data, data collection devices and procedures, editing techniques, and the importance of data accuracy and integrity. Students should be made aware that an effective DBA must understand the content of the database, and therefore must have interdisciplinary knowledge. This objective can be reinforced in the project's final report, by asking students to envision themselves in the role of DBA for the company in the project case.

- Ability to organize the collected data and design appropriate structures for storage and retrieval. This includes diagramming techniques, normalization, data structures, assignment of primary and foreign keys, secondary keys and indexes, domains, referential integrity, and retrieval methods. These topics comprise much of the content of the lectures, but the project can reinforce their understanding significantly. For example, the class can be given assignments with relations which are not correctly normalized; they can attempt to enter data that violates a foreign key constraint.

- Implementation experiences. These are, of course, the primary purpose of the database project, but the experiences can be much more meaningful and valuable if they are designed to reinforce lecture concepts. As will be elaborated in a later section of the paper, the project can provide a much stronger understanding of such concepts as referential integrity, commit and rollback, or domains than can be achieved with lectures alone. In implementing a database, students will naturally make errors; the instructor should stress that these errors can be quite valuable. Analysis of what caused the error message often requires intense critical thinking. Some of the most harmful errors do not produce an error message; recognition of this is startling to some students and helps them to see the importance of editing. Students are conditioned to think of errors as something to be ashamed of; they need to be shown the learning value of mistakes.

- Skill in using SQL. This language has long been the standard in mainframe relational DBMS, and is becoming increasingly popular for microcomputer databases. It can be a valuable addition to a student resume. It can be used throughout the course to illustrate important concepts. We stress the similarities between SQL, entity-relationship diagrams, and relational notation. Query-by-Example (QBE) is also an excellent instructional language, and is similar enough in structure to SQL that they can be taught in parallel if desired. We expose students to SQL query statements as early as possible, usually by the third week of the semester. By the time the project begins, about halfway through the course, they are experienced at composing and executing complex multi-table queries.

Students are conditioned to think of errors as something to be ashamed of; they need to be shown the learning value of mistakes.

These are the primary learning objectives; objectives for individual class lectures are generally subsets of these. It is likely that object-oriented concepts will be added to the list in the future. We also try to include some material on ethics (e.g., privacy) and international issues. Since it is difficult to find any slack time in the course to fit this material in, the final report in the project can be designed to expose students to thinking about such issues. For example, they can be asked to imagine that they are implementing the project in a foreign city, and to include a page or so in the report to discuss the special problems they might have, the additional data that might be required, and so forth.

Operational Issues

Should the project be designed for a mainframe environment, or for microcomputers? Most database textbooks provide at least some coverage of representative database systems for micros, but the emphasis is on mainframe systems [9, 10, 11,12]. As microprocessors become more powerful, and as microcomputer DBMS's become increasingly sophisticated, the distinctions between them will be less meaningful. Date [12] orients his text toward multi-user systems "for reasons of generality" but points out that "the distinction is largely irrelevant as far as most users are concerned" (p. 6). Perhaps a more important consideration than mainframe vs. micro will be standalone vs. network. We chose the VAX for our project because we do not have a suitable local area network environment and we want students to experience a multi-user system with data sharing. We want them to work with software that has concurrency and security facilities as comparable as possible to what they might encounter in a large organization. If they learn to cope with the problems of a multi-user environment, the transition to a standalone microcomputer database would seem to be relatively easy, whereas to move from standalone to multi-user would be difficult.

What data model should be stressed: relational, network, or hierarchical? There are still many hierarchical and network systems in operation, and that is likely to be true for many years. Many organizations, perhaps even most, have not yet made the move to relational. In curricular programs which can offer two semesters of database, it would make sense to teach all three of the major models. We do not have that luxury, and we chose to emphasize relational for the following reasons:

- Relational provides students with a much clearer idea of the logical design process than do the other models. Mapping from external views to the conceptual model is straightforward. Relational makes it easier for them to understand the information needs of a given situation and to see how that translates into relational tables.

- With relational, the students can more easily comprehend that relational tables, entity-relationship diagrams, and SQL formulations are all essentially the same thing, just different ways of illustrating the structure of the database. They can see, for example, that a one-to-many relationship on a diagram represents a foreign key in the table on the "many" side, and the foreign key makes possible a join of the two tables: one occurrence of the primary key to many matching occurrences of the foreign key.

- Even though there may be many databases using the older models today, our curricula should be designed for the future. The typical student will not graduate for another year or so after they've taken the database course, and it may be several years before they are admitted to a database administration group, and perhaps still longer before they are entrusted with database design tasks. By that time, the number of network and hierarchical databases may have dwindled considerably.
- Our students get some exposure to the fact that these older models are out there, and with interpreting a Parent-Child diagram. However, we do not teach the details of CODASYL or IMS. We could not spend enough time on either of these to do them justice, and to devote, say, a week to each would just confuse the students; they would not learn anything meaningful, in our opinion. It is certainly a valid argument that teaching the network model would give students a much better understanding of the physical structure of the database, and of the idea of navigating through the data. However, this is MIS, not Computer Science; it is more important, in our view, for them to understand the conceptual level than the physical.

Our DBMS software is RDB, a product of Digital Equipment Corporation. It was not "chosen"; it was the only SQL package available to us. We provide reference material with instructions for creating tables, inserting data, and for carrying out the other project tasks. RDB has proven to be an excellent educational product; it is easy to learn and provides support for domains, foreign keys, and good security features. It lacks a report generator; hopefully future versions will add this. Oracle, DB2, or any other product with a good SQL interface would be suitable.

...teaching the network model would give students a much better understanding of the physical structure of the database, and of the idea of navigating through the data. However, this is MIS, not Computer Science; it is more important, in our view, for them to understand the conceptual level than the physical.

Project Administration Decisions

Should students work on the project in groups or individually? No doubt some would get more value from working alone, but we believe that for most students a well-matched small group is preferable. There should be no more than three in the group, and it is best if they are relatively equal in ability. It is also advisable to instruct them to divide the work so that each one does some table creation, some data input, and some of each of the other major tasks, rather than one person doing the entire task. We have found that self-selection of groups works better than for the instructor to choose them, especially if they are advised to select others of comparable ability; they learn a great deal from each other. The instructor should have a written policy for handling the inevitable conflicts. Our policy is to encourage them to exchange written memos which explain the position of each side as a first step; often the disagreement can be resolved satisfactorily with minimal involvement from the instructor.

Our preference is to have the entire class work on the same case, rather than each choosing their own project. This ensures that the project will be comprehensive enough to reinforce most of the major learning objectives of the course. Selecting a project from industry would work well also [13], but our students are required to take a "Projects" course later when they will have this experience. The instructor materials for most of the database texts include at least one suitable case; it is fairly easy to adapt one of these or to create one. It is important to encourage the students to "play the role" of the DBA: to try to imagine that this is a real company and that they are the DBA for the company. In the final report, they are asked to carry this role further and provide suggestions for improved data accuracy, security, additional data, and other aspects to bring the experience as close as possible to an actual organization.

The database should have several tables and enough attributes to make the entities realistic. Our undergraduate databases typically have eight or nine tables; the graduate cases a few more than that. Students must create their own data, and it must be realistic. Each table has a minimum number of rows which must be filled. There are a number of queries which their database must correctly answer after it is implemented.

Our students are asked to keep a diary, and a time log for each participant. While there is a wide range of times, 25 to 30 hours seems to be typical of the time that each student spends on all aspects of the project. A few wait until near the end of the semester to get started, but the majority space the effort out over several weeks and it is not overly burdensome.

Electronic mail can be a particularly effective technique for communicating between instructor and students or between students who are members of a group. Many students quickly perceive that a home computer with modem can greatly facilitate the project tasks. It enables them to do most of their work at home rather than making trips to the school, and they can gain access to the system 24 hours a day, rather than cope with the limited hours that the laboratory is open. With the E-mail facility, they can send a message to the instructor if they have a major problem they have not been able to solve on their own. I check for messages two or three times a day, so I can examine their database, analyze the problem, and send them a hint to help them solve it. In two recent classes, I received about 150 messages from each class during the semester. E-mail also facilitates group work; it enables the group members to communicate with each other without an inordinate number of face-to-face meetings.

PROJECT-FACILITATED LEARNING

What if the database project comprised the entire course -- no textbook, no lectures except to explain how to implement the database and how the software works, no examinations? If everyone agrees that the project is valuable for the hands-on experience it provides, why wouldn't it be even more valuable with a greater amount of such experience? Probably most readers, and even most students would agree that this would not be a good idea; doubling a good thing doesn't necessarily make it twice as good. The project is an excellent reinforcer and illuminator, but it is not effective at teaching new material. In this section, we will explore some of the concepts for which the project can improve comprehension.

Referential Integrity

This is one of the most important concepts in the database course: that there is a "foreign key" in one table that represents the entity defined in another table. In a sense, this foreign key is a surrogate for the entity, and an example of encoded data. It is fascinating that so many students seem to understand this, and may give the correct examination answer, but then fail to see the connection to their own database. RDB supports foreign keys, so when referential integrity is violated, an error message occurs. If Table B has a foreign key that references the primary key of Table A, then (a) Table A must be created first, (b) data cannot be loaded into Table B until Table A has been loaded, and (c) every entry in the foreign key column must match a value in Table A. If any of these three rules is violated, the error message appears: VIOLATION OF CONSTRAINT B_FOREIGN1. In a recent class, at least ten students who got that message did not understand what it meant, and came for guidance to the instructor. This included several of the best students, since they were ahead of the others in carrying out the project; the slower students learned from these pioneers. Once they understood the error and how to correct it, the real meaning of referential integrity suddenly became comprehensible, and it is likely they will retain this knowledge for much longer than if they had merely "learned" the definition from the textbook.

Domains

This is another difficult concept to understand from the textbook presentation or to explain in a lecture. The idea that a column in a table can share characteristics with another column in the same or a different table, thus facilitating data independence, is not easy to grasp. The students are required to create appropriate domains for all the attributes, and their initial tendency is to create a domain for each attribute in each table. When it is pointed out to them that, for example, PILOT_NAME and PASSENGER_NAME have the same length and the same general format (last name first, comma and space after last name) they realize what sharing the domain means. The inclination then is to go too far in the other direction, lumping all the 4-character fields together, for instance, and reducing the database to a very few domains. This can lead to a discussion of integrity checks that can be applied to a column, and an understanding that the most desirable number of domains is between these extremes. It can also facilitate their understanding of the concept of data independence.

Commit and Rollback

Prior to commencement of the project, all students will have been exposed to one or two lectures on backup, recovery, concurrency, and the concept of a transaction. Some of them can laboriously follow the logic of forward recovery, backward recovery, commit, and rollback, but unless they have had experience with real world databases it seems doubtful that they have a deep understanding of these concepts. In the project, students may attempt to load some data from an input file and receive an error message indicating a syntax error. They may then correct the error and run the input file again; now they get an error message to the effect that they are violating the primary key constraint. What has happened is that some of the data from the first attempt got into the table, and now they are trying to re-enter those records; as far as the system is concerned, these are different records with the same primary key. The student had an incomplete transaction, and should have rolled it back before re-entering the data. Another common mistake is to log off the system without committing the changes to the database, and at the next log-in a substantial amount of work may have been lost. One or two incidents like these provides a much better understanding of commit and rollback.

The three concepts discussed in this section are not by any means the only new insights provided by the database project, but they are particularly striking ones, and they can lead to improved comprehension of other topics. The common thread through this discussion is that the greatest learning seems to occur when a mistake is made. In trying to figure out WHY it is a mistake, and then how to correct it, the student is forced to think back to the material in the textbook or the lectures and analyze how it relates to the error. This is a process of critical thinking [14]. It may not be correct to say that we only learn from our errors, but where does the learning take place? A plausible hypothesis is that the concept is not learned all at once. It is like a database transaction; it is partially learned from the lecture, but not "committed". Some knowledge of the concept is available in the student's mind, but is not solidly bound. When the error occurs and the error analysis performed, the transaction is completed, and commitment takes place. Unfortunately, until this point the temporary knowledge is subject to "rollback", especially after an examination.

SUMMARY AND CONCLUSIONS

This paper has addressed the topic of the database project. The basis for discussion was a series of classes taught by the author during a three-year period. Learning objectives and the learning value of the project were analyzed. It was stressed that the project requires careful planning and that it should be designed to reinforce the lecture material, rather than as an isolated task with distinct objectives. Some of the most important project design decisions were discussed. The opportunity for students to make errors and to increase the depth of learning through the analysis of these mistakes is seen as a major value of their participation in the project.

The great majority of students in these classes, and probably in most database classes, perceive that participation in the project was a valuable experience. This seems to be because they see it as practical and relevant. Their comments stress the practicality of the project, and the opportunity for hands-on experience. It makes the theory more meaningful and comprehensible. Many of the comments have related to the positive experience of working with a group; for some reason these groups seemed to work together much more cooperatively and amiably than did groups in other courses. Possibly this is because the task is relatively well structured.

REFERENCES

1. Niederman, F., Brancheau, J. C., and Wetherbe, J. C. "Information Systems Management Issues for the 1990s", MIS Quarterly, 15, 1991, 474-500.
2. Brancheau, J. C. and Wetherbe, J. C. "Key Issues in Information Systems Management", MIS Quarterly, 11, 1987, 23-45.
3. Allaire, Y. and Firsitoru, M. E. "Coping with Strategic Uncertainty", Sloan Management Review, 30(3), Spring, 1989, 7-16.
4. Dixon, P. J. and John, D. A. "Technology Issues Facing Corporate Management in the 1990s", MIS Quarterly, 13, 1989, 247-255.
5. Athey, S. and Plotnicki, J. "A Comparison of Information System Job Requirements in Major Metropolitan Areas", Interface: The Computer Education Quarterly, 13, Winter, 1991-92, 47-53.
6. Mackowiak, K. "Skills Required and Jobs Available for CIS Majors", Interface: The Computer Education Quarterly, 13, Winter, 1991-92, 9-14.
7. Beccue, B. and Chrisman, C. "What Should Universities be Teaching Future Enduser Liaisons about Database?", Journal of Information Systems Education, 3, Fall, 1991, 2-9
8. Frand, J.L., Ray, L.N., Roldan, M.L.H., and Sutcliffe, N.G. (eds.). IBM Management of Information Systems (MoIS) Syllabus Book, 1985-1990 Grant Program, February, 1990.
9. McFadden, F.R. and Hoffer, J.A. Database Management, 3rd ed. Redwood City, CA: Benjamin/ Cummings Publishing, 1991.
10. Kroenke, D. Database Processing: Fundamentals, Design, Implementation, 4th ed. New York: Macmillan, 1992.
11. Pratt, P.J. and Adamski, J.J. Database Systems: Management and Design, 2nd ed. Boston: Boyd & Fraser, 1991.
12. Date, C.J. An Introduction to Database Systems, Volume I, 5th ed. Reading MA: Addison-Wesley, 1990.
13. Saiedian, H. "Guidelines for a Practical Approach to the Database Management Systems Course", Journal of Information Systems Education, 4(1), 1992, 23-29.
14. Paul, R. Critical Thinking: What Every Person Needs to Survive in a Rapidly Changing World. Center for Critical Thinking and Moral

AUTHOR'S BIOGRAPHY

Bruce Rollier is Assistant Professor of Information & Quantitative Sciences at the University of Baltimore. He received his Ph.D. in Information Systems from New York University, and an MBA from Northwestern University. Prior to joining academia, he was a manager for the IBM Corporation for several years.

Information Systems & Computing
Academic Professionals

Serving Information Systems Educators

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096