OPTIONAL PROJECTS IN THE DATABASE COURSE: IMPLICATIONS FOR STUDENT PERFORMANCE

Dr. William I. Bullers, Jr., University of New Mexico, bullers@unm.edu

ABSTRACT

Projects are a common component in the database course in MIS curricula. A project normally requires students to define and manipulate a database in a commercially available DBMS such as Microsoft Access and/or Oracle. This paper examines the role of such projects in the database course, specifically when the project is optional and not counted in the student's course grade. The course is described and implications for student performance are examined.

Keywords: Pedagogy, database course, projects, student performance

INTRODUCTION

The database management systems (DBMS) course is a common element in Information Systems (IS) model curricula, including both undergraduate [8] and graduate [7] programs. Its importance for both entry level and career skills ranks among the highest of all the MIS courses [5]. Although the content of the course has evolved [3,17,20], the basic concepts remain stable with a heavy emphasis on relational concepts and a relational DBMS as profiled 15-20 years ago [15]. More broadly, these fundamental concepts include database system architecture, database design, and database implementation [17]. Beyond this core however, many database courses have diverged to include web interfaces, object-relational databases, and object-oriented programming, perhaps requiring a more advanced, 2nd semester database course [20,22].

Database Projects

Many DBMS instructors believe that database theory must be reinforced through practice. This typically requires implementation of a project [9,11,16,18,19,21], often as part of a project team. For a variety of reasons however, many instructors do not include a team project as part of the course. Previous surveys have indicated that "there is an equal split between database courses which always contain a team project, sometimes contain a team project, and never contain a team project (36%, 34%, 30%)" [16]. The difficulties cited for not including a team database project include lack of time, difficulties in evaluating team work, or the inability of students to meet to work as a team. Of those courses that do require a database team project, some require the entire class to work on the same case [18], while others recommend selecting a real-world project from industry [19]. Some utilize a PC DBMS such as Microsoft Access [20]; others utilize a client-server DBMS such as Oracle [13]; and still others allow the students to choose any DBMS [9]. Some projects are partitioned into multiple steps in order to facilitate feedback to the students and to promote learning during the implementation effort [16,19,21]. The grade weight associated with the database project may comprise up to 50% of the student's course grade [19]. The goal of most project implementations is not to provide vocational training in a particular vendor's DBMS, but to promote the learning of theoretical concepts and the development process associated with a database environment [16,21].

Some Problems with Database Projects

The implementation of a database project provides students with the opportunity to apply database theory to the solution of database problems. However, allocating adequate lecture time in class to cover a specific DBMS is often a problem given the expanding universe of database theory that also needs to be covered. This is particularly problematic if the instructor wants to use a complex DBMS such as Oracle, rather than say, Microsoft Access [13].

One approach to managing limited lecture time for DBMS software is to rely upon cooperative learning among the students [14,21]. By working in groups on the database implementation project, students help each other learn the intricacies of the DBMS system. Students have different learning styles and a variety of passive, active, and mixed teaching styles may provide an optimal learning environment [1]. Team-oriented database projects can provide some of this variety through collaborative team learning. However the assessment of individual student performance now becomes more difficult. How much of the course grade should be attributed to the team? How are individual contributions measured?

Other issues also arise. Should cooperative learning be strictly intra-team, or is inter-team collaboration permitted? If cooperation among teams is permitted, at what point does it become plagiarism? Students may have different perceptions from faculty regarding the acceptability of various collaborative practices. Many students admit to cheating and find a variety of cheating practices to be acceptable [6]. Yet some research has demonstrated that collaborative practices "despite falling within the definition of plagiarism, are a beneficial, and possibly useful, means of enhancing the learning process" [4]. Thus, cooperative learning may require both faculty and students to walk a thin ice between cooperative learning practices and plagiarism.

Faculty who assign database projects must recognize that students may cheat. Thus the instructor must spend time during grading to check for plagiarism. Humans pattern match very well. Still, checking each project for plagiarism is a burdensome task. This check is especially onerous at the end of the semester when most projects are graded, given deadlines for the submission of course grades.

Cheating also leads faculty to adopt the "proposition that assignments must be designed in such a way that re-use of the assignment from a previous running is not possible" [6]. I always construct a new project for each semester's assignment. A project may be similar to one used previously, but each is modified and disguised sufficiently to constitute a new and distinct project. I find the time spent in this type of activity to be non-productive, but necessary. It is not surprising that many instructors do not require a project in the database course.

THEORY AND PRACTICE IN THE DATABASE COURSE

The main topics covered in the DBMS course include data modeling and logical database design; database architectures with an emphasis on the relational data model; physical database design; data and database administration; and data manipulation. Oracle's Database Server and Microsoft Access are used to illustrate these concepts. Database design is illustrated through GD-Pro for UML class diagrams and Oracle Designer for entity-relationship diagrams. The current database textbook is Kroenke's *Database Processing* [12].

Database development progresses from conceptual data modeling to logical database design to physical database implementation, as illustrated in Figure 1. That is also the sequence of typical coverage in most database texts, e.g., [12]. However, students often struggle with the abstract

modeling ideas embodied in conceptual data modeling, particularly without prior hands-on DBMS experience. Thus, the sequence of database topics covered in class is that illustrated by Figure 2, progressing in a top-down, left-to-right sequence.

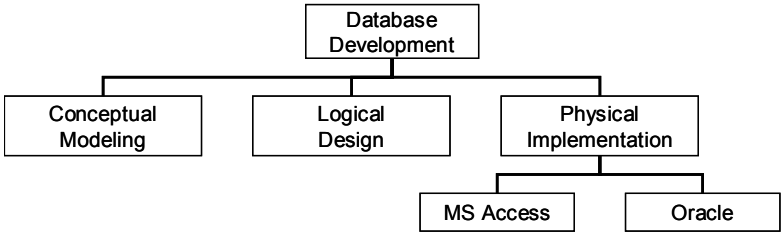

Figure 1. Database Development Process.

As illustrated in Figure 2, the sequence of topics differs from that of the database development process (and that found in most database textbooks). Rather than proceeding from abstract data modeling to concrete DBMS specification as actually occurs in database implementation, MS Access is introduced very early in the course. Its ease of use and seeming "simplicity" allow students to more readily grasp relational data model concepts by way of concrete examples. Having established some familiarity with database models through the use of Access, the relational data model is then discussed more formally (e.g., tables, relationships, and normal forms). Eventually the relational database that the students implemented in Access is reverse engineered into higher-level conceptual models, including UML Class Diagrams and Entity-Relationship Diagrams (ERD's). These abstract models are reinforced through the use of the Oracle Designer CASE tool (for ERD's). The database theory and modeling concepts are then re-applied to the same project database as was used in Access. However, it is now implemented in Oracle, using SQL for data definition and manipulation.

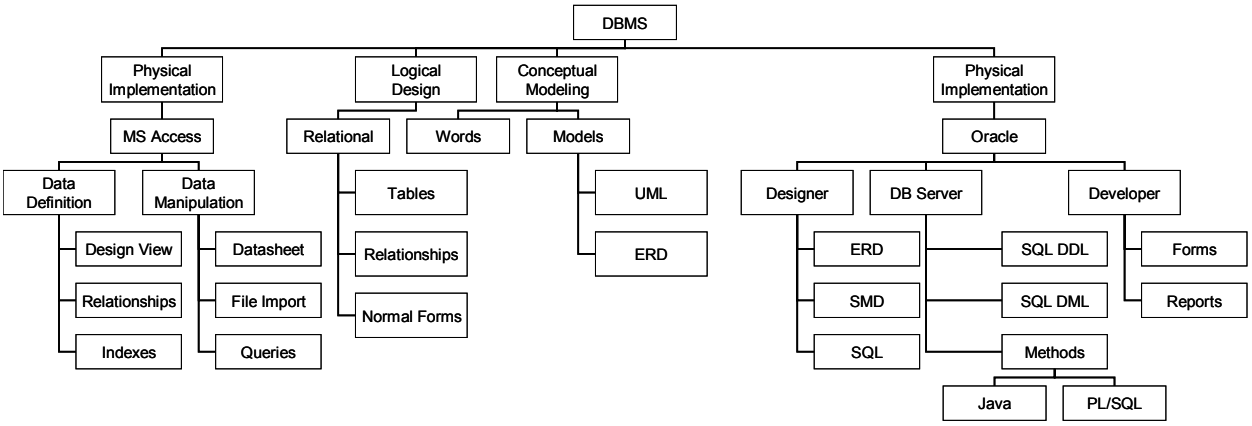

Figure 2. Database Course Sequence.

Thus, the actual course sequence follows a cyclic, iterative approach to the presentation of material. Database concepts are introduced through concrete, hands-on examples, generalized to more abstract logical and conceptual models, and then re-applied through more concrete examples. However, the achievement of the course learning objectives requires that the student actively participate in the learning opportunities presented through class lectures, reading assignments, exercises, and especially database development projects. Active learning is required in order for the student to learn the theory and practice of database development.

Grading Criteria

Previous class grades typically included weights as follows: data modeling exercises 5%; MS Access project 10%; database design exam 35%; Oracle project 15%; and relational, SQL and Oracle exam 35%. In total, database projects and exercises (all group or individual) constituted 30% of the course grade, but probably account for 50% or more of the effort.

Problems with the Course

Several problems exist with this course format. Plagiarism is a continual problem, particularly in the Oracle database project. Students cannot reuse project solution keys from previous semester, since a new project is assigned. However, that does not prevent inter-team collaboration which may border on plagiarism. Nor did it prevent one project group from hiring an experienced student from a prior semester's class to complete their project.

For grading purposes students in each group are credited with the same grade, regardless of each individual's participation in the database development. Students must still take an individual exam, based largely on the project material (particularly in the case of the Oracle exam). However, with group project scores accounting for 30% of the course grade, students may be credited with a higher course grade than they necessarily deserve.

The preparation of a new, structured database project each semester is time consuming. The database specification is invariably found to contain errors or lack desired characteristics. The database structure and test data values often fail to exhibit the richness desired in order to emphasize pertinent characteristics of the relational data model. Likewise, data complexity is often insufficient to support levels of query complexity desired. While changes in specification and the incidence of errors may be typical in real-world applications as well, they cause problems in an academic environment that are difficult to manage. The effort of developing a new project from scratch each semester also prevents extensions from being readily incorporated in the project as new topics emerge in database theory and practice.

Finally, the grading task associated with two database projects per semester can be quite burdensome given the enrollment explosion experienced by our (and most other) IS programs. The undergraduate database course has 70-80 students enrolled in two sections during the fall semester, while the graduate course averages 45 students in a single section in the spring. Thus grading is a chore, even though the utilization of project groups mitigates the volume of grading to some extent. A forthcoming scheduling change in which both the BBA and MBA classes are offered in the same semester means that projects for 120 students will have to be graded.

As a result of the various problems with required database projects, I decided to make the project optional. No grade credit is awarded for completion of the project. Students are free to complete projects individually, in groups, and/or in collaboration with any other group of students or work colleagues. Projects submitted are graded according to the same criteria by which required projects were previously subjected. I provide detailed feedback identifying errors, omissions, and crediting insights that extend beyond the project's nominal objectives.

OBSERVATIONS AND CONCLUSIONS WITH OPTIONAL PROJECTS

Experience in Semester 1

Optional projects were first assigned in the MBA course in Spring 2001, the last semester before the scheduling change to offer all database classes in the same semester. The enrollment in the

single graduate section was 54 students. Course grading consisted solely of two exams, a mid-term data modeling and relational theory exam, and a final exam requiring SQL data definition and manipulation. Two projects were assigned: a Microsoft Access database project to be completed prior to the mid-term exam, and an Oracle project due prior to the final. Projects submitted by students were "graded" and returned approximately a week prior to each exam.

This class of MBA students was exhorted to complete the projects. However, they had only the instructor's pleas to rely upon in judging the worth of the projects in terms of learning database theory and practice, or perhaps more pertinently to some, the worth in terms of impact on their exams, and hence course grade. Of the 48 students who completed the class, 33 (69%) did the Access project, but only 17 (35%) completed the Oracle project. The project "grades" ranged from 5-89 (out of 100) on Access with a mean score of 54.8, and ranged from 39-100 on Oracle with a mean of 81.3. At the completion of the course, a ranking of students based upon their exam scores revealed that 10 of the top 11 students (91%) and 14 of the top 19 students (74%) had completed the Oracle project. In contrast only 6 of the lowest ranking 29 students (21%) had completed the Oracle project.

Experience in Semester 2

In the Fall 2001 semester, 60 BBA and 29 MBA students completed the database classes. For the two BBA classes, 26 of 60 students (43%) completed the Access project, while 13 of 60 (22%) completed the Oracle project. The Access project grade range was 24-92 with a mean of 52.1 and for Oracle the range was 24-90 with a mean of 59.7. For the MBA class, 20 of 29 (69%) completed the Access project and 11 of 29 (38%) completed the Oracle project. The Access project range was 16-93 with a mean of 53.7; for Oracle the range was 46-92 and mean of 77.5.

At the end of the semester I ranked students in both BBA and MBA groups by exam scores. For the BBA students, 12 of 15 students in the top quartile (Q1) did the Access project, but only 9 of 15 in Q2, 5 of 15 in Q3, and 0 of 15 in Q4. Likewise among the BBA students, 8 of 15 students in Q1 did the Oracle project, but only 3 of 15 in Q2, 1 of 15 in Q3, and 0 of 15 in Q4. In sum among the BBA students, 14 of 15 students in the top quartile, Q1, did one project or the other, but only 10 of 15 in Q2, 5 of 15 in Q3, and none of 15 in Q4 did either project. Results were similar for the MBA students although not quite as dramatic, with 7 of 7 students in Q1 completing one project or the other, 6 of 7 in Q2, 5 of 7 in Q3, and 3 of 8 in Q4.

Summary of Experience with both Semesters

To date 137 students in 4 classes (2 BBA and 2 MBA) have completed the database course under optional projects. A summary of those who completed Access or Oracle projects, based upon their quartile ranking, is shown in Figure 3.

Conclusions

The participation of students in the optional projects has been disappointing. Only 78 out of 137 students (57%) have completed the Access project, while even fewer, 43 out of 137 (31%), have completed the Oracle project. It is difficult to ascertain cause and effect in this limited data analysis. Clearly those students ranking in the top quartile are more likely to complete one project or the other. Those in the lowest quartile are least likely to do so. Further analysis is required to answer the question of whether such a phenomenon merely reflects good students doing the work or whether completion of the project is a causal factor in determining a student's performance.

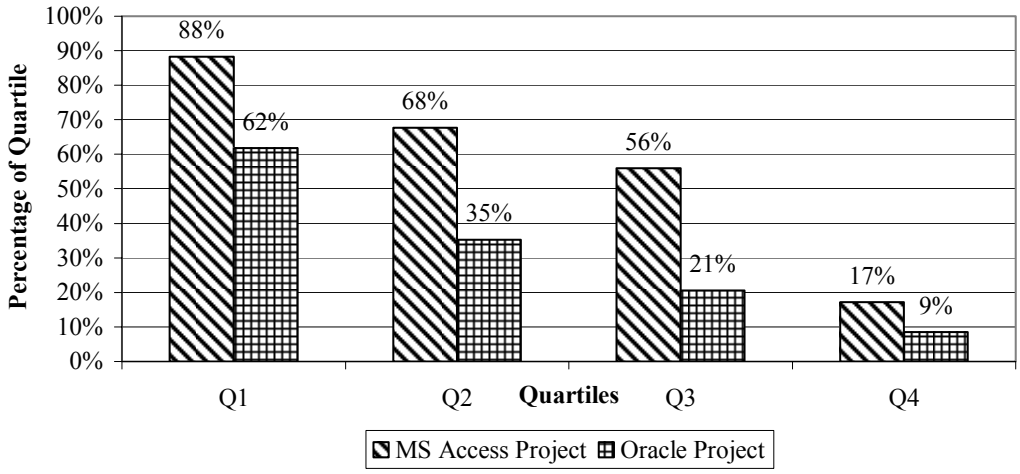

Figure 3. Quartile Ranking vs. Project Completion.

The "Big Bang" project due dates, with the entire Access or Oracle project due on a designated deadline, may give students too much opportunity to procrastinate. This procrastination, coupled with the optional grade associated with the project, may contribute to the failure of many students to complete the project. While studies have shown that people are willing to set self-imposed deadlines to help curb their tendency to procrastinate, these self-imposed deadlines are not as effective as externally imposed deadlines in improving task performance [2]. Instructor imposed deadlines for a required project, particularly involving interim due dates, may be a more effective means of insuring active student learning through project participation.

The reuse of the project's specification has permitted its extension in several, beneficial ways. The project has been recently integrated with the Personal Software Process (PSP) [10] to introduce the discipline of structured software development practices. This allows the students to better plan, track, and analyze their development activities and should lead to higher quality database projects. The database project is also being refined into a series of step-wise exercises, in order to promote cyclic development in conjunction with PSP software management techniques. The introduction of a series of interim due dates should also serve to more fully curb students' procrastination instincts. Finally, plans are underway to extend the project in the BBA classes to include Java programming and JDBC connectivity to the Oracle database.

The brief hiatus offered by project reuse during the last couple of database course offerings has permitted the course to evolve with concomitant project support. This has certainly served to benefit the instructor. However, this benefit has seemingly come at the expense of some students' learning. The better students, particularly those willing and able to impose self-discipline in completing the optional project, appear to perform as well as in a class with a required project. Students with less self-discipline, however, do not appear to perform as well as with an externally imposed project requirement. As a result, I plan to drop the experiment with the optional project and re-introduce a graded, required database project.

REFERENCES

1. Allsopp, L. (2002). Teaching and Learning in First Year Seminars. Journal of Information Technology Education, 1(1).

2. Ariely, D., and Wertenbroch, K. (2002). Procrastination, Deadlines, and Performance. Psychological Science, 13(3), 219-224.
3. Cain, W.P. (1990). Changing Requirements for the Database Course Curriculum. Journal of Information Systems Education 2(2).
4. Carter, J. (1999). Collaboration or Plagiarism: What happens when students work together. SIGCSE Bulletin, 31(3), 52-55.
5. Chrysler, E., and Van Auken, S. (2002). Entry Level Value Versus Career Value of MIS Courses: Faculty Expectations Versus Alumni Perceptions. Journal of Computer Information Systems, 42(3), 38-43.
6. Dick, M., Sheard, J., and Markham, S. (2001). Is It Okay to Cheat? – The Views of Postgraduate Students. SIGCSE Bulletin, 33(3), 61-64.
7. Gorgone, J.T., Gray, P., Feinstein, D.L., Kasper, G.M., Luftman, J.N., Stohr, E.A., Valacich, J.S., and Wigand, R.T. (2000). MSIS 2000 Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems, DataBase 31(1).
8. Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D.L., and Longnecker, H.E., Jr. (2002). IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. DataBase 34(1).
9. Guimaraes, M. (1999). Experience Teaching an Introduction to DBMS. SIGCSE Bulletin, 31(4), 48-49.
10. Humphrey, W.S. (1997). Introduction to the Personal Software Process, Reading, MA. Addison-Wesley.
11. Keys, A.C. (2002-03). Using Group Projects in MIS: Strategies For Instruction and Management. Journal of Computer Information Systems, 43(2), 42-50.
12. Kroenke, D. (2002). Database Processing: Fundamentals, Design & Implementation, 8th Ed. Englewood Cliffs, NJ. Prentice-Hall.
13. Morrison, J., and Morrison, M. (2000). Using Oracle in the Information Systems Curriculum. Journal of Computer Information Systems, 40(3), 1-9.
14. Neufeld, D.J., and Haggerty, N. (2001). Collaborative Team Learning in Information Systems: A Pedagogy for Developing Team Skills and High Performance. Journal of Computer Information Systems, 42(1), 37-43.
15. Pierson, J.K. and Shorter, J.D. (1986/1987). Profile of Introductory Database Courses in AACSB-Accredited Institutions. Journal of Computer Information Systems, 27(2), 29-33.
16. Robbert, M.A. (2000). Enhancing the Value of a Project in the Database Course. SIGCSE Bulletin, 32(1), 36-40.
17. Robbert, M.A., Wang, M., Guimaraes, M., and Myers, M.E. (2000). The Database Course: What Must be Taught. SIGCSE Bulletin, 32(1), 403-404.
18. Rollier, B. (1993). The Database Project: Maximizing Its Value. Journal of Information Systems Education, 5(2).
19. Saiedian, H. (1992). Guidelines for a Practical Approach to the Database Management Systems Course. Journal of Information Systems Education 4(1).
20. Springsteel, F., Robbert, M.A., and Ricardo, C.M. (2000). The Next Decade of the Database Course: Three Decades Speak to the Next. SIGCSE Bulletin, 32(1), 41-45.
21. Urban, S.D., and Dietrich, S.W. (1997). Integrating the Practical Use of a Database Project Into a Theoretical Curriculum. SIGCSE Bulletin, 29(1), 121-125.
22. Urban, S.D., and Dietrich, S.W. (2001). Advanced Database Concepts for Undergraduates: Experience with Teaching a Second Course. SIGCSE Bulletin, 33(1), 357-361.