

AN EXPERIMENT IN TEACHING DATABASE CONCEPTS INDEPENDENT OF SOFTWARE PLATFORM

Dr. Matthew A. North, Washington & Jefferson College, mnorth@washjeff.edu

ABSTRACT

This paper examines Information Systems teaching, as opposed to training, in the context of a Database Concepts course. The course was developed using the IS 2002 model curriculum as a guide, with specific effort made to focus on teaching students fundamental concepts and techniques while using a branded RDBMS software package as a tool for instruction. The course was taught by the same instructor in three consecutive semesters using three different database software packages (Oracle, MS SQL Server, and MySQL). Student scores from these three offerings are compared statistically using ANOVA, revealing that the IS 2002 objectives for the database course can be achieved independent of the software package selected for use in instruction. Students taught in one environment were then tested on database skills in a different environment in order to determine whether or not their comprehension of database concepts and techniques transcended the software environment in which they were taught. Students achieved comparable, passing scores using different software packages, lending support to the position that if instruction is focused on teaching database concepts and techniques rather than training in one specific tool, students will be able to work effectively in various database software environments upon graduation.

Keywords: IS Education, IS Training, Databases, Model Curriculum.

INTRODUCTION

In an information-driven economy, the demand for skills in information systems and technologies (IS&T) creates many opportunities for organizations to offer instruction in these fields [5]. This instruction can take many forms, often dictated by needs of the organizations that look to hire skilled individuals to perform IS&T work for them [1, 9]. The spectrum of instruction subsequently ranges from software-specific training to broader based fundamental and conceptual education programs [4, 10]. The format of instruction will also then vary, from boot camps and week-long intensive training sessions, to semester-long courses and even multi-year college programs of study [1, 4]. Indeed, the

type of institution offering the instruction will likely impact the format and content of their offerings [12]. For example, the purpose and mission of most community and technical colleges would be most compatible with skills and software training, while colleges and universities would be more likely to focus on broader education, including bachelor's and graduate degree programs.

Within the spectrum of IS&T training and education, database skills are increasingly in demand [7]. Much of this demand for database knowledge and skill is fueled by the fact that databases form the underlying foundation of many modern information systems utilized throughout the world today [9]. Due to this ubiquity, and the relatively unknown nature of database computing platforms a student may encounter as they enter the workforce, educational programs which focus on developing strong conceptual understanding in students serve as a versatile preparation for a diverse IS&T landscape [2].

The purpose of this paper is not to propose one instructional form (i.e. training vs. education) as superior to another when approaching database education, but rather to investigate how the use of specific software tools in an education-focused instructional offering can be neutralized in order to avoid slipping into a mentality of training on a chosen software when the intent of a given course is to focus on teaching fundamental database concepts and techniques [6]. When the goal is education, and when that education is accomplished using software tools in an instructional setting, instructors must be conscious of their delivery and tool use in order to avoid inadvertently creating a training environment in the classroom [11]. This study was thus undertaken to make a calculated effort to adhere to the education-based mission of a Database Concepts course, while testing the ability to use different software solutions as instructional tools to teach fundamental database concepts including entity relationships, attribute and data type definition, keys, referential integrity, data modeling, database programming, structured query language, administration and performance tuning. The objective is to determine the success with which these concepts and skills can be taught using a

specific database software, but without confining the learning of the concepts and skills only to that particular database software.

Statement of the Problem

The ANSI standard followed by most Relational Database Management System (RDBMS) producers today creates an environment where skills learned in one database software can generally be applied to other software packages on currently available to individuals and organizations. Where the goal of a given course of instruction in databases is to educate students in foundational concepts and techniques, there is a risk of slipping into a mentality of training on a software tool which was chosen for instruction, rather than focusing on the more elemental, and software independent, concepts of database design and development. For example, if an instructor chooses Oracle as the instructional software for their course, they may fall into a pattern of “teaching Oracle”, as opposed to teaching the aforementioned database skills and concepts (e.g. the relational model, SQL, etc.) [8].

Research Question

The objective of this paper is to answer one fundamental question: Can a single selected database software tool be used as an instructional mechanism for database education, preparing students with the fundamental database knowledge necessary to work in any major RDBMS software environment, without causing a skills training mentality to enter into a classroom intended to provide foundational education? Stated as a null hypothesis, we would say that there is no significant difference in student learning in a database course between three groups of students taught the same material but using different software tools.

METHODOLOGY

This study was conducted at a four-year college, with the express mission of educating students in broad and foundational IS&T skills, as opposed to offering software-specific training. As previously mentioned then, the objective was not to “teach Oracle”, for example, but rather, to provide students with a solid understanding of fundamental database concepts which could be applied wherever databases are found, in whatever format. The course was intentionally designed to meet this overarching objective.

Section 8 of the IS 2002 model curriculum—“Physical Design and Implementation with DBMS”, was used as the guideline in the development of the course [3]. A partial list of specific course objectives for this section is given in Table 1:

Table 1. IS 2002.8 Partial List of Course Objectives

- Develop abilities in data modeling of databases
- Acquire skills in application and structuring of database management systems, including relational modeling, referential integrity, Structured Query Language (SQL), data validation, etc.
- Understand application and physical implementation of database systems, using a programming environment for stored procedures, user-defined functions and triggers, packages, etc.
- Develop and use project management, project standards, and a system implementation plan, and to implement a documentation plan
- Create and maintain systems which meet required levels of service and availability to meet client needs
- Articulate the syntactical and theoretical differences between database models
- Organize and create database conversion plans including mechanisms for database structure, data type, and data transfer conversions
- Create, test and implement relevant use cases and system test parameters

There are 16 objectives in all in the model curriculum, none of which suggest skill training on a specific platform is required, or desired, in database education. Although the same objectives may be used in developing a database training course, these objectives are expressly and intentionally created in a broad and generic sense, such that they do not favor any one specific brand of relational database management system over another.

With these objectives in mind, the course used in this study was developed and offered for the first time in the Fall semester of 2006. Oracle was chosen as the software tool for instruction in the first offering. The class consisted of 22 students. In the spring semester of 2007, the same course was offered to 24 students, this time using Microsoft SQL Server as the instructional software. Then in the fall of 2007, 18 students completed the same course using MySQL as the platform for instruction.

Data Collection

Throughout each of the three semesters, students completed graded assignments demonstrating their

ability to create entity relationship diagrams, data flow diagrams, SQL scripts (both data definition and data manipulation), stored procedures, custom functions, triggers, and referential integrity/data validation. The assignments remained the same across all three terms, regardless of software environment being used, as did the lectures, texts and exams. In addition, students completed an additional post-test at the end of each term, in order to assess their ability to apply what they had learned throughout the term in some other database environment. Scores from the post-test were not factored into students' final grades in the course. Those in the Oracle were post-tested using SQL Server; those taught in SQL Server used Oracle for the post-test, and those instructed using MySQL also we post-tested using SQL Server. At the time of the Oracle and SQL Server offerings, MySQL's production release of their community edition RDBMS did not yet contain all of the functionality required to be used in post-test. Specifically, the ability to create triggers and custom functions was not added to the MySQL RDBMS until version 5.0 was released. The functions for stored procedures were dramatically enhanced when the stable version of this release was made public as well [13].

RESULTS

Once the three semesters used in this experiment were complete, the final grades of all students were compared using a standard ANOVA test, in order to determine if any statistically significant variation in final grades existed between the students in the three course offerings. Table 2 lists descriptive statistics for each of the three course sections.

Table 2. Database Course Student Scores

	SQL Server	MySQL	Oracle
Grade Mean	89%	87%	86%
Grade Median	91%	90%	87%
Grade Range	41-100%	67-96%	69-94%

When ANOVA was performed on the final scores, the resultant P value was .90 ($\alpha=.05$), far too large for what would be necessary to find statistical significance.

Similar results were observed on the post-test administered in a different software environment. While the scores were lower across the board, they were not significantly so, and they were equally homogenous. Table 3 depicts the outcome of this analysis.

Table 3. Database Post-test Student Scores

	SQL Server	MySQL	Oracle
Grade Mean	84%	85%	82%
Grade Median	84%	84%	85%
Grade Range	63-92%	62-97%	59-91%

ANOVA results for these scores yield a P value of .51 ($\alpha=.05$), which is also not significant. In addition to being in a different database environment, the fact that the grade on this post-test was not factored into the course grade may be one of the factors explaining why the scores were lower than on the other work generated by students. In the interest of full disclosure, a brief discussion of slight syntax variations was provided to students before the post test. This was necessary due to Microsoft's use of T-SQL, which is considered to be ANSI compliant, but which also has some syntactical differences which are needed in order to write functional code in the SQL Server RDBMS.

Figure 1 compares the mean course and post-test scores achieved by the student across the three sections of the course examined in this experiment.

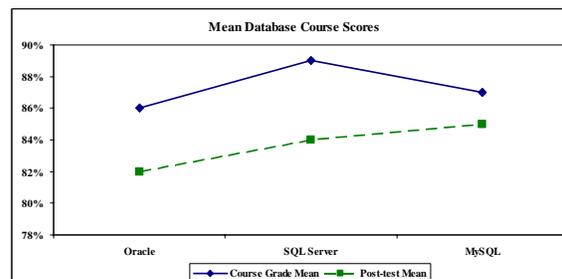


Figure 1. Mean Database Course Scores

Although the post-test scores are lower across the board for each of the three groups, they also clearly approximate one another and indicate satisfactory learning of the material taught in the course. The statistical tests revealed that the scores themselves are not significantly different from one semester offering to the next, nor are they different between the course grades and the post-tests given in the same semester. While working with a slightly foreign user interface on the post test, students were still able to successfully create and relate tables, define and assign data types to attributes, write queries, and create programs and functions which interacted with the entities in their databases.

DISCUSSION

The null hypothesis of this experiment states that there is no significant difference in student learning in a database course between three groups of students taught the same material but using different software tools. We have failed to reject that null hypothesis based on the outcome of the ANOVA statistical analysis of the grades earned by the students in the three courses used in this experiment.

Given the goal of this database course to focus on education, as opposed to specific software training, our failure to reject the null hypothesis in this experiment is a positive outcome. It is further positive to observe that students were able to achieve acceptable scores on their database work, which was designed to help them meet the stated IS 2002.8 model curriculum objectives. The mean grades in the mid- to high-80 percent range are indicative of effective learning of the database material taught throughout the semester. As shown in both Table 2 and Figure 1, students were successful in learning to design and develop databases according to the course objectives, while Table 3 provides support for the claim that they learned the necessary fundamentals well enough to perform database tasks in a variety of software environments.

The scores on each individual assignment between the three class offerings were not compared to one another, however a combined analysis of scores achieved by students who completed the course during the three semesters in question reveals some interesting results.

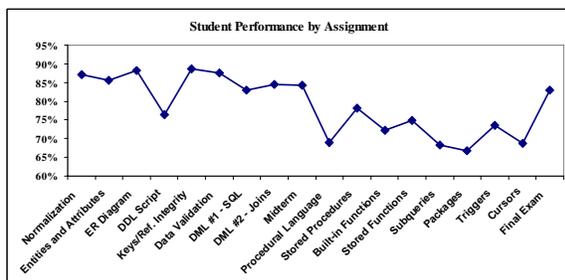


Figure 2. Student Scores by Assignment

There are two interesting phenomena illustrated in Figure 2. First, students appear to have grasped the earlier content in the course better, and as the term continued, with more advanced material being taught, scores decreased in the classes. This finding is not surprising, considering the fact that many of the students had only completed one programming class prior to entering Database Concepts, and the latter half of the class focused almost entirely on implementing procedural programs in databases. On

the final exam, which was comprehensive in nature, scores were up sharply as compared to the students' performance in the weeks immediately preceding the exam. This is likely due in large part to their grasp of the early topics in the class, which helped to improve their overall final grades.

The second interesting observation in Figure 2 is that collectively, student scores always remained within a range which indicates learning is taking place. Even when scores on assignments dropped into the lower 70% range on some of the later assignments, students were performing at a capable level, regardless of the software environment in which they were learning database skills and concepts. Students continued to demonstrate their ability to perform database tasks at similar levels in different relational database management systems when they completed the post-test following the final exam.

Future Study

This study creates some additional questions which could be answered by future study. One area to study might include a detailed comparison of the assignment scores across the three groups. For example, do students learn to create stored procedures equally well in Oracle, SQL Server and MySQL; or is this skill more difficult in a specific RDBMS? An examination of the students' performance at each stage of the experiment may provide additional insight into whether or not their performance is consistently equivalent across the three teaching platforms, or if some inconsistencies are masked by examining overall course averages or combined assignment scores.

This study could also potentially be used as a model for testing differing teaching approaches in other IS&T subjects. For example, when attempting to teach programming fundamentals in a CS1 course, can students meet model course objectives equally well if they learn Visual Basic, C++ or Java; or will there be significant differences in students' abilities if one programming language (i.e. instructional tool) is given favor over another. The same questions can be posed in such a scenario. Are we training students to program in Java, or are we teaching students programming fundamentals which can be applied to any programming environment. This question may prove to be a bit more difficult to evaluate, because there is one fundamental difference between the database comparisons drawn in this paper and a similar study comparing programming languages. This difference is that while relational database management systems enjoy an ANSI standard, which

as a result causes a high degree of similarity among all database software, programming languages do not enjoy such a strong regulatory standard. Thus, syntax and approach (such as object-oriented approach), may make the teaching of programming fundamentals very different. Thus caution should be used when embarking upon an instruction experiment such as this one, in order to ensure that comparable results are generated between two or more groups.

CONCLUSION

When teaching database concepts, and particularly when database *education* is the goal, teachers can use a variety of software tools in their instructions. They must also be cautious however, to avoid slipping into software training mode when using a specific software solution in their instruction [12]. It is clear that software training is necessary in order for individuals to acquire specific skills required to perform needed work for their employers and clients [1]. Striking that balance is key in order to broadly prepare students to compete in an increasingly competitive IS&T marketplace, while ensuring they have the skills necessary to perform on whatever database platform they are required to when they enter the work force.

This paper has presented scores from three consecutive semesters where three different database software systems—Oracle, SQL Server, and MySQL—were used as instructional tools. Care was taken to ensure that aside from differing RDBMS platforms in the classroom, all learning activities remained the same across all three groups in the study. Statistical analysis using ANOVA found that no significant differences among the three groups of student scores were found, indicating that regardless of software platform, objectives for the model Database Concepts curriculum can be, and we met in this course.

REFERENCES

1. Ahrens, J.D. & Sankar, C.S. (1993). Tailoring Database Training for End Users. *MIS Quarterly*, 17(4), 419-439.
2. Blake, M.B. (2003). A Student-enacted Simulation Approach to Software Engineering Education. *IEEE Transactions*, 46(1), 124- 132.
3. Gorgone, J.T., Davis, G.B., Valacich, J.S., Topi, H., Feinstein, D. L., & Longenecker, H.E. (2002). Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. ACM Digital Library. [Electronic Version]. Retrieved on 3 March 2008 from: http://www.acm.org/education/curric_vols/is2002.pdf.
4. Heng-Li, Y. (2003). Comparing Relational Database Designing Approaches: Some managerial implications for database training. *Industrial Management & Data Systems*, 103(3), 150-166.
5. Nickerson, J. V. (2006). Teaching the Integration of Information Systems Technologies. *IEEE Transactions on Education*, 49(2), 271-277.
6. Robinson, J. (1980). Education and Training for Computer-Based Reference Services: A Case Study. *Journal of the American Society for Information Science*, 31(2), 97-104.
7. Springsteel, F., Robbert, M. A. & Ricardo C. M. (2000). The Next Decade of the Database Course: Three decades speak to the next. *ACM SIGCSE Bulletin*, 32(1), 41-45.
8. Tenopir, C. (1989). Educating Future Professional Searchers: The Role of Formal Education. *Library Journal*, 114(14), 164-165.
9. Trauth, E. M., Farwell, D. W. & Lee, D. (1993). The IS Expectation Gap: Industry Expectations Versus Academic Preparation. *MIS Quarterly*, 17(3), 293-307.
10. Urban, S. D. & Deitrich, S. W. (1997). Integrating the Practical Use of a Database Product into a Theoretical Curriculum. *Computer Science Education*, 29(1), 121-125.
11. Vician, C., Garfield, M., Hoffer, J. A., Prescott, M., Rollier, B., Strong, D. M. & Elder, K. L. (2004). The Chicken and the Egg Debate: Positioning Database Content in the Information Systems Curriculum. *Communications of AIS*, 4(14), 147-157.
12. Wu, M.M. & Ying-Hsang L. (2003). Intermediary's Information Seeking, Inquiring Minds, and Elicitation Styles. *Journal of the American Society for Information Science & Technology*, 54(12), 1117-1133.
13. Zaitsev, P. (2005). MySQL: New Features in Version 4.1, 5.0, 5.1. Open Source Forum, Moscow, Russia.