An Exercise in New Course Assessment

Kurt Jordan
Assistant Professor
Information Systems and Computer Programming
Purdue University Calumet
Hammond, Indiana 46323, USA
jordank@calumet.purdue.edu
http://www.calumet.purdue.edu

Abstract

Whenever a new course is offered for the first time, it can become a learning experience not only for the students, but also for the instructor. This paper describes the experiences encountered in teaching a new upper-level database administration course. The course particulars, the problems and experiences encountered, the assessments, and the changes put in place based on the assessments will be described.

Keywords: Assessment, Course Evaluation, course feedback, Course Development, Database Administration

1. INTRODUCTION

Because Information Technology changes so quickly, IT faculty are challenged with developing new courses more frequently than they would probably like. With many new courses, what the course developer intends for the course and what actually happens as the course is delivered for the first time diverges widely. Among the causes are lack of experience in course development, improper or inadequate prerequisites and unreasonable student and faculty expectations for the course.

The assessment and subsequent changes that occur after the first semester for a new course are critical to making necessary improvements quickly so the course can settle into a solid, consistent educational experience for the students.

The first section of the database administration course described in this paper was held in the Spring 2002 semester. The difficulties with teaching the initial section of the course were most pronounced in the hands-on lab exercises. The assessment and corresponding course alterations described here are limited to that area.

2. PURPOSE OF COURSE

The idea for the database administration course arose out of a new Internet/Web degree option that was recently developed. Three areas of specialty were identified: Web designer, Web programmer, and Web administrator. The Web administrator would be responsible for installing and administering the software that composed a typical Web site: operating systems, Web server software, and relational database management systems, among others. The purpose of the database administration course described here is to prepare those students who aspire to be Web administrators with RDBMS installation and administration skills.

Three prerequisites were selected for the course that would, it was hoped, provide adequate background for the student to be successful. The first is an introductory computer operating system course. Students should have basic knowledge of modern operating systems concepts such as memory management, I/O control systems and process management, as this knowledge is essential to proper RDBMS setup and tuning.

The second is an introduction to networks course. Students should have basic knowledge of how networks and the Internet work. Most databases in use today are available over a network. Gartner Group indicates that in 2002, over 70 percent of new application development projects use the browser as the primary interface (Gartner, 2002).

The third course is an introductory applied database course, which introduced students to SQL using report writer software and an already-existing Oracle database. Knowledge of SQL syntax is necessary for many DBA tasks.

The database administration course is a senior level course. By the time most students have the three prerequisites listed above, they also have taken an introductory C++ programming course, and a structured systems analysis and design course. So, before enrolling in the database administration course, most students have taken at least five computer-related courses.

3. COURSE DESCRIPTION

The course covers database administration tasks and techniques. Students install and implement relational database management systems. The topics covered include RDBMS architecture, installation, configuration, creating databases, migrating data, database object management, user account management, tuning and backup and recovery.

4. COURSE OBJECTIVES

After completing the course, it is hoped the student will be able to perform the following list of objectives.

- Describe the physical architecture of a relational database management system
- Discuss and make appropriate physical layout decisions for RDBMS physical files
- Create and manage RDBMS objects
- Create and manage database user accounts
- Discuss performance tuning issues and tradeoffs
- Install and configure RDBMS software
- Create, manage and tune databases
- Design and implement backup and recovery procedures
- Discuss and implement appropriate database security measures

5. CLASS PROCEDURE

Surveys indicated the top three RDBMS in use throughout the industry are Oracle, DB2 and SQLServer (Gartner, 2000). However, with sixteen weeks available, it became clear there would not be enough time to adequately cover all three. It was decided to cover the top two: Oracle and DB2.

The course was divided into halves. Oracle would be covered in the first half. DB2 would be covered in the second half. For each half, the student will be required to complete a presentation and a lab report. The student will install the RDBMS software, create accounts and customize the system for typical use. Throughout the term, relevant tasks are assigned that correspond to the topic covered in the lecture portion of the course. I tried to assign one new task per week. The lab report should include a description of each task including its purpose, documentation on how the task was accomplished, a description of any problems that were encountered, along with the solution to the problem, and a summary of thoughts concerning the task.

One presentation is made on each database system. The presentation is a short five to ten minute explanation of some command, process, procedure or technique used in the RDBMS. The instructor assigns the topics. The student must research a topic relevant to the RDBMS being studied and present it to the class.

6. SEMESTER SCHEDULE

The semester schedule shown below was used for the first offering of the course.

Week 1 Introduction to course
Oracle objects

Week 2 Oracle Installation

Week 3 Startup/shutdown
Process monitoring

Week 4 Security

Week 5 Backup/Recovery

Week 6 Tuning

Week 7 Project reports due
Presentations

Week 8 test

Week 9 DB2 objects

Week 10 Installation

Week 11 Startup/Shutdown
Process monitoring

Week 12 Security

Week 13 Backup/Recovery

Week 14 Tuning

Week 15 Project reports due
Presentations

Week 16 Final examination

7. LABORATORY DESCRIPTION

In the lab, twenty PCs with removable hard drives were available for the course. Before the semester started, I installed RedHat linux on one drive and had our network support technician copy the disk image to the other removable hard drives assigned to my class. Each student received one of the drives. The textbook chosen for the course had a Oracle 8i CD bundled with it. For the DB2 half of the course, I downloaded a 90 day evaluation copy of DB2 from IBM's web site and copied the distribution file onto each student's removable hard drive.

Students came into the lab on the first lab day with the equipment and software ready to begin the installation. Students would get laboratory exercise specifications that describe the task to be performed, and the purpose of that task. Students have a two-hour block of time for lab work each week.

8. REALITY SETS IN

As the semester progressed, it became obvious that student work in the lab was not going quite like I had envisioned. I had initially started by simply supplying a high-level task description. For example: 'Install Oracle by following the steps in the installation chapter in your book.' The majority of the students seemed to lack the ability to take that task description and subdivide it into the atomic steps necessary to carry it out.

For example, the first set of instructions in the installation chapter in their textbook was to create user accounts on the linux system that would own the Oracle RDBMS software. Many did not have any experience with linux, had not even seen a command prompt and did not know how to go about getting one. The entire two-hour lab session would end with only a few managing to complete the assignment.

To address this, I tried giving students a set of abstract steps to perform instead of just a high-level description of the task. For example, instead of giving them the statement: 'turn on archive logging', I would give them this series of steps: 'Create a folder to hold archived logs, make necessary changes to the database startup file to enable archive logging, turn on archiving'. Still, a significant number of students (50% or more) continued to have problems figuring out exactly what they should be doing. Finally I started passing out a list of step-by-step instructions along with the lab explanation and purpose. Some steps were still abstract but simple, such as 'start the database.' Other steps presented the exact syntax necessary to carry out that step, such as 'type: alter database archivelog; press the ENTER key.'

These changes were sufficient to allow all students to successfully finish the lab exercise within the two hour allotted time. However, wide variations in the skill levels of the students were still noticeable. Even with the step-by-step instructions in hand, some students consumed the entire two-hour lab period for almost every lab exercise, while other students required only 20 minutes.

It became apparent from the problems many students were having that they did not have an adequate background in several areas to be successful in this course without significant coaching. Students lacked experience using a text-based command line interface, both at the operating systems and at the database level. Many tasks described in their textbook are performed using a command line. In addition many database administration tasks require using a command line-based SQL interpreter to get information from the database. Most students have used a graphical user interface (GUI) throughout their entire college career. Indeed, a GUI is the only interface they have ever seen. Significant time was spent during the labs bringing some students up to speed on the use of a command line interface.

Another problem was lack of adequate skills applying SQL to retrieve data from a table. For example, given a list of columns from a system table in the database, a significant number of students could not put together the correct SQL syntax for a simple SELECT statement to display data from that table.

For the second half of the semester, the topic shifted to IBM's DB2 RDBMS. A 90-day DB2 Universal Database for linux evaluation copy was available for download at IBM's Web site (IBM, 2002). The documentation and manuals were available on-line (IBM 2002). Reading assignments were made from those manuals. Students installed DB2 on the same removable hard drive on which the Oracle installation was performed.

The second half of the semester went more smoothly, I believe, for three reasons. First, the students had the benefit of half a semester of exposure to similar DBA concepts. For example, having created a database in Oracle, they already knew the basic steps, considerations, etc.

Second, DB2 has a better user interface than Oracle for database administration tasks. For example, when

creating an instance in DB2, the linux accounts are created automatically. Oracle assumes any linux accounts exist before hand. They must be created manually. Most students indicated they liked DB2 much better than Oracle because of the easier-to-use GUI.

Third, I continued to use the step-by-step format for lab exercises and significant coaching for SQL syntax.

9. TIME FOR ASSESSMENT

The assessment mechanism used in our department consists of an assessment document that is completed for each course at the end of each semester. The important section of the assessment document consists of four columns. The first contains a general description of the learning outcome the student is expected to acquire. The second column contains a specific description of what is done in this particular course to present the skills or topic addressed by the general learning outcome. The third column is the assessment mechanism used to determine whether or not the student has acquired the skills or knowledge of the topic. Examples are a test question, or lab exercises, or an in-class presentation. The fourth column is used to record the instructor's indication of whether or not the student was successful in acquiring the skill or knowledge of the topic, and suggested steps for improvement if a problem was indicated by the assessment mechanism. Figure1 shows sample columns from a Web programming course.

Learning Outcome assessment model	**Learning Outcome in course**	**Assessment**	**Results**
Describe the purpose and organization of common business functions and processes, and opportunities for automation and information and Web-based services.	1. Describe the client/server paradigm. 2. Describe multi-tiered architecture.	2. In-class discussions 3. Examinations	1. Students were easily able to accomplish this.
Describe characteristics of and apply information system solutions to personal, workgroup, enterprise, inter-enterprise and international problems and opportunities.	1. Explain the characteristics, advantages and disadvantages of thin and thick clients.	1. In-class discussions 2. Examinations	1. Students were easily able to accomplish this.

FIGURE 1

In the database administration course, most students were successful in acquiring the concepts and definitions presented in the course. The scores on the two tests were consistently high, with an average score of 82 and a median of 83. As noted earlier, the serious problems were encountered in the hands-on lab exercises.

The major problems encountered with the lab portion of the course were command line and SQL skills, and an inability to break down high-level tasks smaller, more manageable tasks.

For the SQL skills, I talked to the faculty responsible for the applied database techniques prerequisite course. This is the course where students received SQL skills. The faculty agreed to alter the course to focus more on SQL syntax and less on the report writer. However, it will take some time for students taking the altered applied database techniques course to percolate through the system and get to the database administration course.

To address the lack of command line skills problem, I will alter the database administration course so the first week in the lab focuses on basic command line skills, such as file and folder/directory management,

text file editing, basic system management tasks, and using an SQL interpreter.

The problem students seem to have with breaking down a high-level task into smaller steps is being addressed two ways. The first is a review of the prerequisites, both to ensure they are appropriate choices and to determine whether there is a problem within one or more courses with the students getting functional decomposition experience.

The second is a short-term fix. I'll guide the students through the breakdown process for the first two to four lab exercises until they understand and can carry out that process for themselves.

Table 1 summarizes the problems and associated resolutions.

TABLE 1

Deficiency	**Changes made**
Inability to break down high-level tasks into smaller, manageable steps	Review of current Prerequisites for appropriateness and for functional decomposition skills
Lack of text-based command line experience	Spend week in DBA course on command line skills
Lack of adequate SQL syntax knowledge	Alter pre-requisite course to focus more on SQL syntax

These changes to the course will consume time from an already tight semester schedule. There will not be enough time to provide adequate coverage for two RDBMS in the semester. Dropping one RDBMS from the course will allow adequate time for the changes described above and allow more time to be spent on each weekly topic. Eliminating the second presentation and project report will also free up some more time.

The RDBMS chosen for a semester could alternate between the top two or three popular systems in use in industry. In situations where multiple sections of the course are running simultaneously, a different RDBMS can be used in each section, providing students with a choice when they register.

10. CONCLUSION

When teaching a newly developed course, the assessment and subsequent changes that occur after the first semester are critical to quickly improving the course so it can settle into a consistent, stable offering for the students.

The problems students are having with the course should be correctly identified, so appropriate changes can be made. Depending on the problem, changes can take the form of new or changed prerequisites, altering the content of a prerequisite course, and re-structuring the course, perhaps in several ways, for example, spending time supplying the necessary skill sets and altering the semester schedule to cover fewer concepts or topics so more time can be spent on the remaining concepts or topics.

Of course, assessment results will vary, and the methods of implementing changes will also vary, depending on your own particular circumstances. However, the first assessment/implement change process for new courses can result in major improvements the second (and subsequent) time the course is offered.

11. REFERENCES

Database Management Systems Software Market Maintains Double-Digit Growth in 2000, Dataquest Alert SOFT-WW-DA-0002, 22 May 2001, Colleen Graham, Gartner Group

The Return of Client/Server — or, at Least, Rich Clients, Research Note SPA-15-4707, 8 March 2002, David Smith, Gartner Group

IBM Web page: DB2 Universal Database and DB2 Connect Online Support, May 2002, http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v7pubs.d2w/en_main

IBM Web page: DB2 Universal Database Downloads, May 2002, http://www14.software.ibm.com/webapp/download/product.jsp?s=c&id=TDUN-49EVGU&type=b&presb=&postsb=&cat=data&rs=&S_TACT=&S_CMP=&q=&k=any&pf=Linux&dt=TRIAL&v=&x=35&y=8