2007

Mastery level learning and the art of database design

M. Pamela Neely

MASTERY LEVEL LEARNING AND THE ART OF DATABASE DESIGN

M. Pamela Neely
pneely@saunders.rit.edu
Rochester Institute of Technology

Abstract: Database design is as much an art as it is a science. In order for students to become skilled enough to produce an effective design, they will need to achieve the synthesis level of Bloom's taxonomy. Given the effort involved in reaching that level, a constructivist approach must be used to teach the course. However, even using a project based approach may not be sufficient. Based on learning styles research, it can be shown that learners with a mastery level coping style will be more likely to stick with the process long enough to achieve the synthesis level than will learners with a helpless learning style. The process described in this paper incorporates many standard database teaching techniques, along with an innovative example and various pedagogical techniques to help the students adopt a mastery level mindset and achieve the necessary level of cognitive skill to effectively design databases.

Keywords: database design, learning styles, Bloom's taxonomy, constructivist approach, SQL

Introduction

"Learn how to learn." "Don't be afraid of failure." "Database design is an art as much as it is a science." "You learn by doing." These are the things I tell my students on the first day of a Database Management Systems (DBMS) class. Like most database classes, the ultimate goal is for students to master the concepts of database design, including normalization, and SQL. However, the art of database design is difficult to learn. Students are frustrated by the fact that there is no "one right answer." Modeling a real world entity, even with concrete tools such as entity relationship diagrams, is a very abstract process. In addition, in order to make the course useful, we have learning objectives beyond the basics. Students should develop their skills in interpreting user requirements. They should be able to take a poorly designed database and convert it to one that is in third normal form (3NF). Ultimately they should be able to go beyond classroom exercises and comprehend the process of database design well enough so that they can go into a job with the necessary skills to turn those ugly databases we all see in practice into a model of efficiency. Coupling the difficulty of learning with these additional learning objectives and we realize that the "sage on the stage" method of teaching database design and concepts is very ineffective. We want students to be excited by databases. After all, most modern information systems have a relational design. Mastering database concepts is important regardless of the major- MIS, accounting, marketing, or new media development. The approach outlined in this paper appears to be helping to attract new students to the DBMS class from other majors within the college of business, as well as majors throughout the university. Based on anecdotal evidence, it has also helped students to reach the level of mastery that enables them to go into an internship or job, as well as a second course in database development, and get up and running quickly.

The next section of this paper outlines the motivation for developing this approach to teaching the database course. This is followed by a description of the methodology, and an example of the materials used. Finally, some anecdotal findings are discussed, suggesting that the method described in this course will lead to a better understanding of database design and its connection with SQL.

Motivation

As indicated earlier, relational databases are the foundation for most modern information systems, including web-based systems. And yet, the results of a study by Blaha (2004) indicate that the quality of databases is mediocre, at best. Certainly observational evidence supports the results of Blaha's research, a study which includes eleven years worth of data. As consultants in database design, colleagues have provided me with over a hundred examples of poor database designs implemented in Access or as the back-ends for web systems. Typically these designs involve one to three poorly constructed tables with numerous partial and transitive dependencies. The owners of these databases usually are at a point where they cannot accomplish some goal, such as a user friendly data entry screen, or a complex query, when they come to us with their problems. Given the perceived ease of use of a tool such as Access, many information workers will create databases for the analysis of data downloaded from a centralized system. Others will be asked to collect data for a specific project and will choose a database as their tool of choice. They have no real understanding of relational database concepts and will try to use a common sense approach to the development of their database. As Chilton (2006) describes however, many novice database developers confuse the "data at rest" with the actions performed on the data. In the process of designing a database, we need only be concerned with the facts that should be captured, not the processes that involve the data (Date, 2000). Novice designers have a tendency to create tables to capture the results of processes, and are confused when their designs are not able to be implemented. An approach to database design that tackles these problems, as well as an understanding of how the database will perform when implemented, will produce students who are capable of modeling and implementing the real-world systems that they will encounter.

However, as indicated earlier, the process of database development is as much an art as it is a science. In order for the students to effectively model the real world entities, they must progress to Bloom's fourth level of the taxonomy of the cognitive domain (Bloom, 1956). They must proceed from knowledge (the recall of specific information), through comprehension (the conversion of abstract content to concrete situation) and on to analysis (the comparison and contrast of the content to personal experiences). They must then reach the level of synthesis (the organization of thoughts, ideas, and information from the content) in order to effectively model a database and implement it. In addition, achievement of this level of learning will aid the student in moving beyond simple retrieval SQL queries and on to the more complex queries involving grouping, multi-table joins, and calculations.

How do we encourage students to put forth the effort necessary to move to this higher level of learning? Dweck and Sorich (1999) discuss the concept of mastery-oriented thinking versus helpless oriented thinking. Students with a helpless learning mindset think that intelligence is a fixed trait. They are more concerned with showing that they have a lot of intelligence (i.e. good grades on exams) than they are with learning. In addition, they will interpret their failures as a lack of intelligence. On the other hand, learners with a mastery learning mindset believe that intelligence is something that is cultivated over time; it is not a fixed trait. They focus on the goal of learning, not just on looking smart. More importantly, they interpret failures as meaning they need more effort or a new strategy. They do not interpret failure as a deficiency in intelligence. In addition to the two classes of learners, there are also two classes of goals. Performance goals, typically associated with the helpless mindset, have an aim of gaining favorable judgments of his or her competence (or avoid unfavorable judgments of it). Their goal is to "look smart." The goals of the mastery oriented mindset are called learning goals. Individuals with learning goals have an aim to increase their competence. They want to "get smarter."

Dweck and Sorich's research shows that there are ways to foster the mastery learning mindset. Even individuals with a predisposition towards the helpless coping style can move towards the mastery coping style. Both criticism and praise are key to this movement. As might be expected, judgmental criticism (attacking the person's traits or person as a whole) fosters a strong helpless reaction in response to later setbacks. However, feedback (even negative feedback) that focuses on the need for more effort or a new strategy can move an individual towards a mastery mindset in the face of later setbacks. Praise also can have both positive and negative consequences. Praise that reflects on the individual’s qualities (intelligence) can have a negative effect on coping- leading to a helpless mindset. In the face of later setbacks the student whose intelligence has been praised may turn the positive praise around and conclude that, because they failed, they must not really be all that smart. Praise that focuses on the effort of the individual has the effect of fostering the mastery mindset, such that later setbacks will be seen as needing more (or different) effort. These learners will then put forth the additional effort to tackle the task at hand with a different approach.

Connolly and Begg (2006) make a strong case for teaching database concepts using a constructivist approach. They argue that using a problem based learning approach will help students achieve the necessary knowledge and learning to acquire the skills for database design. Watson (2006) argues that the method of database modeling is much less critical than the ability to model a representation of reality. He also suggests that database modeling and SQL should be intertwined. The next section relates a method for teaching database concepts and SQL that will help the student to reach the level of synthesis, using an approach that encourages a mastery level coping style.

Teaching the Course

As indicated earlier, the goal of the class is to develop the necessary skills that students need in order to model the reality of an organization. This understanding is enhanced as the student learns SQL. Thus, integrating SQL with the design can help students to understand the difference between data at rest and the processes that are applied to the data. The next few sections of the paper will detail how the levels of Bloom's taxonomy are achieved, while at the same time focusing on helping students achieve a mastery level mindset.

Moving towards knowledge

Two things must be emphasized to students at the beginning of the course. First, it must be made clear that they will be expected to go far beyond the "memorize and regurgitate" level that is common in many courses. All exams are open book. The objective of testing this way is to encourage students to go beyond memorization and on to comprehension. However, it should be noted that at some level the students will be memorizing. This is the first step in learning terms and concepts. Many students will find that they must first memorize the definitions for terms such as third normal form or transitive dependency before they can grapple with the complexity of what these terms mean. Certainly, the terminology is used in classroom instruction. Communication between the student and the instructor is greatly enhanced if terms are understood. For example, in critiquing a entity relationship diagram, the instructor may suggest that the entity contains a partial dependency. If the student does not know what a partial dependency is, then they will not know what the problem is. However, if they have "memorized" that a partial dependency is one where a non-key attribute is dependent on part of the primary key, the process of removing that dependency becomes more straight-forward. The advantage of the open book test is that student will not feel the need to "memorize for the sake of memorization." They will be memorizing to further their understanding, rather than to regurgitate it back onto a test. Secondly, students are told that the course is cumulative. In an attempt to remove the fear of failure, if a student's score improves as the course progresses, the later grades will replace earlier grades. The exams become increasingly complex, so if learning is achieved by the end of the course then the goal is met and the student deserves the higher grade.

As can be seen in Figure 1 below, one of the earliest communications to the student is that there are a lot of pieces to the puzzle. They should not feel overwhelmed after the first lecture. Giving them the "big picture" helps them to understand that there are many tools to be put in the toolbox before they can expect to really understand the art of database design. This diagram depicts the fact that the process is iterative, from identifying an information need, to meeting with the users, to writing up a narrative, on to the entity relationship diagram, normalization, and populating the tables with data and then back to the information need, which may need to be modified after this analysis.

The first level of Bloom's taxonomy, knowledge, is achieved through short exercises that emphasize the basic concepts of entities, attributes and relationships. Classes are structured primarily as lab sessions. The first lecture, introducing the basic concepts, lasts about an hour. This is the only time in the course where the lecture lasts this long. During the remainder of the course, explanations of key concepts rarely run longer than 15 minutes. Each new concept is immediately followed with multiple exercises, performed as small groups. Discussion is strongly encouraged and active participation in the "whole class debriefing" is a vital part of the learning experience. The assignments are of the type typically found in end-of-chapter exercises. Assessment of the knowledge level is accomplished via online quizzes. A test bank of 100 questions randomly generates 10-question quizzes. Students may take the quizzes as many times as they want, with the highest grade counting.

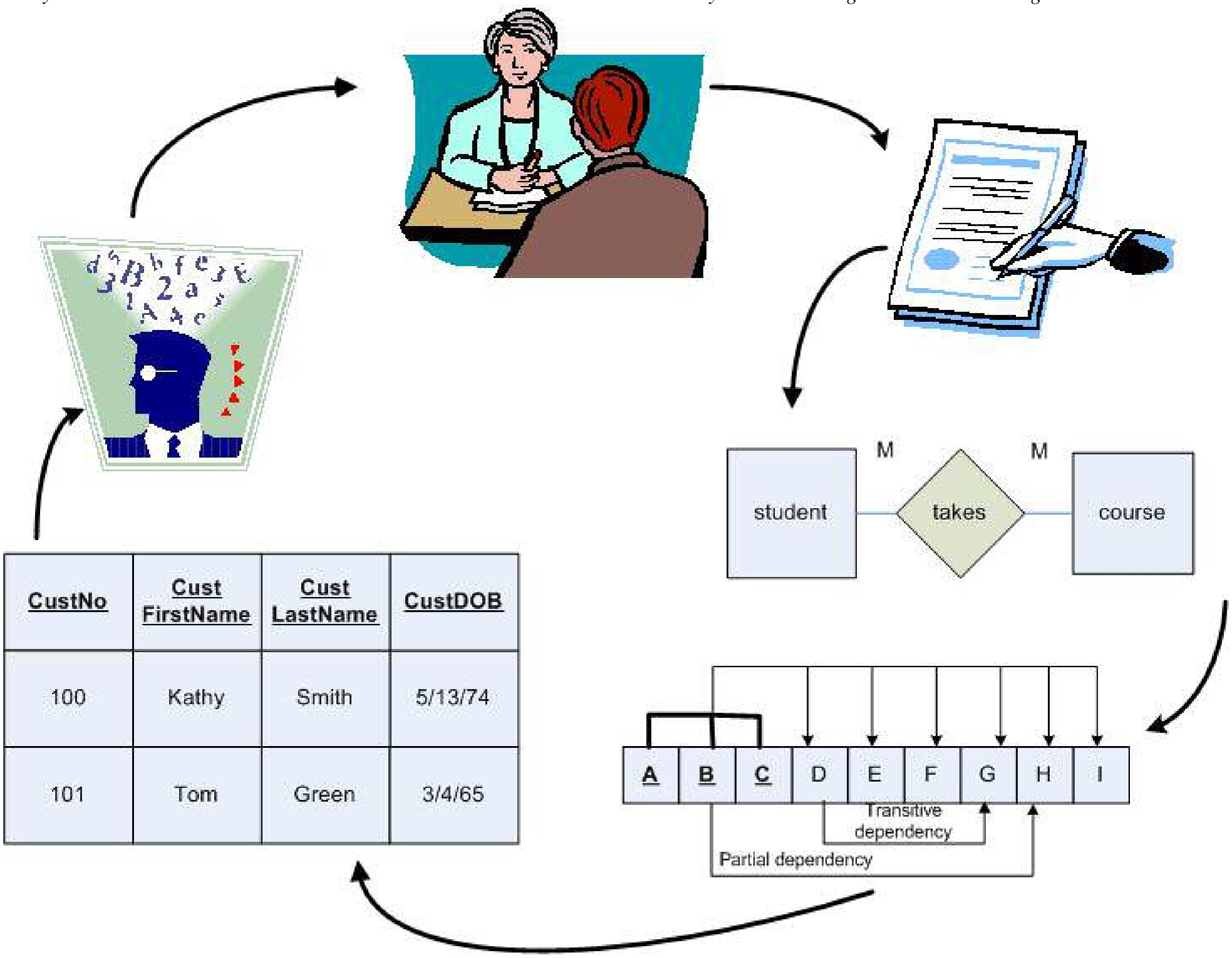

Figure 1- Design is an Iterative Process

From knowledge to comprehension

Moving towards comprehension involves providing students with exercises that encourage them to recognize the entities and attributes from short narratives. I have had considerable success with some of the exercises posted on the ISWorld database portal (http://www.magal.com/iswn/teaching/database/index.cfm). In particular, a set of exercises submitted by Monica Garfield (In class ERDs) have proven very effective. There are 12 exercises, each one different from the other. We usually work through the first 8, emphasizing concepts such as many-to-many relationships, recursive relationships, and aligning the attributes with the appropriate entities. Generally, once the students are comfortable with these exercises they can feel confident that they have begun to move from the concrete towards the abstract. Assessment at this level involves graded homework assignments, as well as class discussions.

From comprehension to analysis

In order to move towards the analysis level of learning, a group project is assigned. This project, which should be identified within the second week of class, helps the student to personalize the process of database design. It is strongly encouraged that the project be one of real meaning to at least one member of the group. Over the years I have seen several databases related to work projects (real estate holdings, tracking employee time off, tracking residence hall keys), as well as database applications related to club and athletic activities.

These projects require a great deal of analysis in order to be successful. Typically, because the student is learning the concepts at the same time that they are engaging in the project, the transition from knowledge to analysis can be rocky. I have instituted a policy of "consulting dollars" in order to encourage student groups to come and see me as their projects progress. Each student group gets a $1,000 line of credit (LOC) at the beginning of the course. This can go towards meetings with the consultant (the instructor) at an increasing cost as the course progresses. At the beginning of the term, the cost is $50 for a 15 minute session. By finals week, when the project is due, the cost is $500 per session. When students take advantage of the consulting early in the term, they rarely use up all of their money and inevitably end up with stronger projects. Because some students will want to save their consulting dollars for last-minute questions, I emphasize that no student group has ever run out of money. If they have been seeing me regularly I will usually increase the LOC if necessary. And, as I tell them on the first day of class, there's no sense in saving your money- you don't get to keep the leftovers! By encouraging the students to see me early and often they can learn different approaches and move towards the mastery oriented coping style.

A major deliverable is the preliminary entity relationship diagram associated with their project. Again, in order to encourage the mastery level mindset, the groups are asked to present their designs to the class as a whole. Grading for these presentations is based on the questions that students ask of the presenters, not the design that is being presented. Because the presenters know that their classmates will be grilling them, they put considerable effort into their initial designs. However, there is no expectation that the design will be perfect. Feedback from the class will always improve the design. Most student groups will have a sufficiently accurate design that the student questioners can highlight problem areas. Common problems are redundant data, foreign keys on the wrong side of a one-to-many relationship, and attributes that are in the wrong entity. By the time the students have been through 6 or 7 presentations, these errors become more evident and they will almost always recognize the errors without prompting from the instructor. Occasionally a design will be so poor, or the scenario so complex, that the student questioners are unable to follow the design. In cases like these, the presenters are usually encouraged to meet with the consultant to go over the diagram in private. Student feedback from these presentations has been overwhelmingly positive. They can begin to recognize the similarities and differences in their various projects. They can also appreciate the "common themes" such as composite or joining tables. For some, this is the first time they really understand the concept of a composite table with a composite primary key.

From analysis to synthesis

The move from analysis to synthesis is usually accomplished through additional exercises and exams. The first exam is made up of several small exercises, similar to the exercises that are done in the first two phases of learning. The testing is at the comprehension level. After this exam, a series of exercises is provided. These exercises, an example of which is shown in the next section, all have the same basic structure. The student is provided with a database structure that is not in 3NF. This structure has been created by someone without database design knowledge (the client). As is seen in countless instances, the ubiquity of Access in business settings leads to numerous poorly designed databases. Each time the initial structure has some combination of multi-valued attributes, non-atomic attributes, partial and transitive dependencies. In addition to the initial database structure (which includes data so that students can recognize multi-valued attributes and identify the primary key), a narrative is included between the client and the student who has been called in to repair the damage. The narrative is a good place to infuse additional teaching tips and humor. Finally, each exercise includes one or more reports that the client would like to have, but cannot get because either the data is not currently being collected or the structure will not support the report.

Each of the exercises has the following deliverables:

- An initial dependency diagram (IDD). Generally, before the IDD can be drawn, the structure must be placed in first normal form (1NF), eliminating repeating groups. The primary key for the structure in 1NF should be identified, as well as all partial and transitive dependencies.
- An entity relationship diagram, generally using either the Chen method or the Crow's foot method. As indicated earlier, the actual representation is not as important as the relationships among the entities. Students must indicate the connectivity between the entities, although cardinality is usually optional.
- A set of tables in third normal form (3NF) with attributes. Again, representation is not important. They may list the tables, e.g. Table Name (Attribute 1, Attribute 2, Attribute 3…), or they may draw a diagram similar to a relational schema with entities indicated by rectangles and the attributes listed within them. Primary and foreign keys should be identified using some form of notation such as underlines and bold or a designating code such as PK or FK. Students are not allowed to turn in a relational schema from Access for this deliverable.

An Example

After searching diligently you have found the co-op of your dreams with Universal Novelty, a toy distribution center in Northglenn, CO. You are really excited because you have been asked to come in and help them with their current database system and it will give you a chance to put into practice what you learned in Professor Neely's database management systems class. After flying to Denver and getting settled into an apartment at the base of the spectacular Rocky Mountains, you are anxious to get started. You have scheduled an appointment with Bob Boyce, the manager of the department that you will be working in. Following is a transcript of your conversation with Bob:

Bob: Hi! Welcome to Denver where we have 300 days a year of sunshine. A little brighter than Rochester, I think.

You: Yes. I am really glad to be doing my co-op during the winter and spring as those months get really long in Rochester. I appreciate the opportunity to work with you and hopefully by the time I leave, you will have a database application that will be a little more user friendly than the one you are currently using.

Bob: Yeah, we had an intern come in from Podunk U and he created the database structure that we are currently using. However, I fear that the system is very sick and will ultimately die. Before that happens, I was hoping that you would be able to help us out. I understand that your professor really drilled you on the importance of good database design.

You: She sure did. And after looking at your current structure and the difficulties that you are having with it, I am really glad she did! Let's talk a little about the structure you have. I take it that this is supposed to be an application to keep track of your employees?

Bob: Yes. We need to know where people work and who they work for, as well as information about them personally. We keep track of their base salary, without overtime or commission. At this point we are keeping track of their payroll in another system, so the base salary is sufficient. Finally, our sales people go through a series of training classes. The sequence of classes that an employee takes is dependent on their prior experience and the type of sales job that they have been hired to do. There is some other basic information that we would like to know about employees as well, so I have given you a copy of our department listing report.

You: In database terms, the structure you have now is not in third normal form. There is a fair amount of redundant data, which can lead to problems when you update, delete, or insert data into the table. However, it isn't that difficult to fix. When I am finished you will have multiple tables and I will have added some new attributes. We will need to move the data from your current structure into your new structure. It is not a difficult process but it will be a little time consuming. Do you have a problem with that?

Bob: No. Since I can't do everything I need to do now, it makes sense to put it into a form that I can use out into the future. Since you will only be here for 6 months I'd like to have a completed application before you leave.

You: Good. That's my thinking as well. As long as we are in the redesign process, what kind of information would you like to see that you are currently unable to get?

Bob: Well, I gave you a copy of the employee listing. I would also like to know how much our training classes are costing us. I can draw up a report that I think would be useful and give it to you.

You: Thanks. I find the more I know about the output of the system, the easier it is since I need to know what data needs to be stored and what data needs to be calculated to produce the output. Anything else?

Bob: Well, I am having trouble getting a count of employees who have specific degrees. For example, if I want to know how many of my employees have a Master's degree and I want to group them by type, such as MBA or MS in Accounting, I can't seem to do that. Do you have any suggestions for me?

You: Actually, when I finish the redesign process that won't be a problem any more. I'll just add a code to the new table that gives a degree type- either U for undergraduate or G for graduate.

Bob: Great! I am really looking forward to getting this going. Be sure to come and ask me if you have any questions that I haven't already answered. I want to be sure that the final product does everything I want it to do!

You: Will do. I'll talk with you later!

Emp Code	Last Name	Educa-tion	Dept Code	Dept	Dept Mgr	Job Class	Job Title	Dependents	DOB	Hire Date	Training	Base Salary
100	Gibson	BS/MIS, MS/ Acctg	Acct	Account-ing	Johns	25	AP	Sue (wife), Brandy (daughter)	5/15/57	8/4/87		$40,000
101	Johns	BS/Acct, MBA	Acct	Account-ing	Johns	37	Mgmt	Tom (husband)	9/11/45	7/15/94		$65,000
102	Dean	BS/ Mgmt	Sales	Sales	Curry	45	Rep	Mike (husband), John (son)	2/8/82	9/1/04	Basic Sales, Cold Calls	$27,000
103	James	BS/Mktg	Sales	Sales	Curry	45	Rep	Linda (wife)	4/6/80	6/1/02	Cold Calls, Closing the sale	$29,000
104	Curry	BS/IB	Sales	Sales	Curry	37	Mgmt		1/7/76	8/15/04	Basic Sales, Cold Calls, Closing the sale, Supervisor Skills	$35,000

Figure 2- Initial Database Structure for Employee Database

<u>Training Cost By Employee</u>

First Name	Last Name	Training Code	Training Description	Cost of Training
John	Dean	1	Basic Sales	100.00
		2	Cold Calls	250.00
			Total	$300.00
Betty	James	2	Cold Class	250.00
		3	Closing the Sale	1,000.00
			Total	$1,250.00
Lucy	Curry	1	Basic Sales	100.00
		2	Cold Calls	250.00
		3	Closing the Sale	1,000.00
		4	Supervisor Skills	1,250.00
				$1,500.00

Figure 3- Employee Training Report

Achieving synthesis

The final tool in the toolbox is SQL. The study of SQL is started immediately after the ERD presentation. Students are encouraged to contemplate how the data is transformed into information via calculations, grouping and criteria. They are also encouraged to think about what information they want to report from their own projects. For the final exam, the students are required to create the scripts that would produce the tables and insert the data into an Oracle database. ISQLPus is used as the engine for performing SQL queries. The same database that was designed in the second exam (similar to the example shown earlier) is used for the final exam. Students are given the tables and data definitions in a Word document. This ensures that all students have the correct design. Additionally, they receive the document in electronic form. This eliminates the need for a lot of typing. They simply convert table to text, separated by commas and add the necessary syntax to transform the raw data into INSERT commands. Thus, students develop the database design in the 2nd exam, reinforce the concepts of primary and foreign key relationships as well as data types in the creation of the scripts, and then are tested with queries on the database with which they have become intimately familiar. Although group work is encouraged throughout the course, a somewhat different model is suggested for SQL. Past experience shows that the most effective way for the students to learn SQL is clearly by doing SQL. This means that each student should complete their own work. However, as the best learning can occur when mistakes are made, students are given significant lab time to perform the assigned exercises, including the creation of the scripts for the final exam. It is also suggested that they complete the exercises that were not completed during class in a similar fashion- one where multiple people are working side-by-side completing the exercises. Thus, when an error message, such as "not a GROUP-BY function" is encountered, they have not only the textbook as a reference, but also each other. The errors that are encountered tend to be common to all, so the rationale is that once the issue has been addressed by one student, the remaining students can benefit.

Because this methodology of students helping each other fosters a mastery learning coping style (the criticism and praise students have for each other tends to be of the non-personal type), many students who adopt this model find that their command of SQL occurs quickly. As with the other exams, the final exam is open book. Students will have had examples of all of the queries required on the final. However, unless they have reached the level of analysis necessary to interpret the queries, they will not be able to perform on the exam, even with the examples right in front of them.

Conclusions

As many researchers argue (e.g. Chilton, 2006; Conolly & Begg, 2006; Dey, Storey, & Barron, 1999; Mohtashami & Scher, 2000), it is necessary to go far beyond the knowledge level of Bloom's taxonomy when educating students in the art and science of database design. The constructivist approach is a solid foundation for reaching the level necessary to effectively model databases and apply SQL in a problem solving manner. This paper has added to the problem based approach to teaching by marrying techniques that can foster a mastery learning mindset with the constructivist approach that is becoming common in database classes. Individually, the techniques that have been used in this course have been adopted in many classes. Collectively, the approach leads to a higher level of achievement on Bloom's taxonomy. To summarize, key factors in the ability to attain this higher level include:

- Encouraging mastery level thinking by removing some of the fear of failure via an alternative testing approach. Students may demonstrate mastery of the material at any time prior to the end of the course.
- Encouraging mastery level coping styles by placing more of the burden of praise and criticism in the hands of the students. Both the ERD presentation and the final project presentation are reviewed and critiqued by the students. Additionally, peer reviews of the group process are required as part of the final deliverables. Students are informed on day one that these peer evaluations can impact their grade (i.e. not all members of the group may receive the same grade). Also, students can be fired from groups for cause.
- Encouraging mastery level coping is also attained through the use of "consulting dollars" which allow the students to discover additional approaches to solving their problems.
- The progress of cognitive understanding progresses in a clear, well defined path. Students must attain knowledge (assessed using on-line multiple choice) before they move on to comprehension (assessed via in-class group exercises and debriefing). From there they move on to analysis (assessed via the project) and finally to synthesis, which is assessed with exams.

Student feedback using this approach has been positive. Increasingly, students are choosing to go into database work for internships and permanent positions. We are attracting students to the course from other business majors, as well as non-business majors. And, perhaps even more importantly, many students who have taken positions involving database development are reporting that they were able to be productive from the start, even reengineering existing databases. The confidence that they have gained, the mastery level coping skills, carry over to subsequent courses and job situations, and ultimately, that may be as important as the design skills that they have developed.

References

Blaha, M. (2004). A Copper Bullet for Software Quality Improvement. *Computer, 37*(2), 21-25.

Bloom, B. S. E. (1956). *Taxonomy of Educational Objectives: the Classification of Educational Goals:Handbook I, Cognitive Domain*. Toronto: Longmans.

Chilton, M. A. (2006). Data Modeling Using Entity Relationship Diagrams. *Journal of Information Systems Education, 17*(4), 385-394.

Conolly, T. M., & Begg, C. E. (2006). A Constructivist-Based Approach to Teaching Database Analysis and Design. *Journal of Information Systems Education, 17*(1), 43-51.

Date, C. J. (2000). *An Introduction to Database Systems* (7th ed.). Reading, MA: Addison-Wesley.

Dey, D., Storey, V. C., & Barron, T. M. (1999). Improving Database Design through the Analysis of Relationships. *ACM Transactions on Database Systems, 24*(4), 453-486.

Dweck, C. S., & Sorich, L. A. (1999). Mastery-Oriented Thinking. In C. Snyder (Ed.), *Coping: The Psychology of What Works* (pp. 232). Cary, NC: Oxford University Press.

Mohtashami, M., & Scher, J. M. (2000). *Application of Bloom's Cognitive Domain Taxonomy to Database Design.* Paper presented at the The Proceedings of the ISECON 2000 Conference.

Watson, R. T. (2006). The Essential Skills of Data Modeling. *Journal of Information Systems Education, 17*(1), 39 - 41.