

WHY IS TEACHING DATA MODELING SO HARD?

Janet M. Drake
Department of Computer Science
University of Northern Iowa
Cedar Falls, Iowa 50614-0507
drake@cs.uni.edu

Abstract

In an introductory database course we teach SQL and Entity Relationship (ER) modeling. SQL is a 4th generation language and it is easy to teach and learn. Teaching and learning ER modeling is another story. Learning modeling is difficult and not all students successfully master data modeling.

Students easily see the relationship between the ER model and SQL data definition statements. But the ER model focuses on problem solving; specifically it focuses on mapping the customer problems into a model that can be used to implement a database application.

Learning ER modeling is hard – very hard. My question is “Why is ER modeling so difficult?” To try to answer this question I created a survey designed to identify something in my students’ background that related to their success in modeling. The survey results show no clear answers but examining the questions leads to an interesting discussion.

Introduction

Models are abstraction that let us represent the customers problem in a way that represents our understanding and allows the customer to validate our understanding. If we model the right problem we have a chance of solving the customers' problem.

There are many models that are used in analysis and design. Object models, use cases, state transition diagrams, data flow diagrams, and entity relationship models are examples. Each model addresses a different view of the application: data, process, and/or control aspect. Entity relationship models address data. Of all the models listed above, ER models are the easiest to understand and create. They are more concrete than the other models because the elements of the model relate directly to data structures – arrays of records or database tables.

Even though ER models are the easiest of the models they are still hard to create. I teach ER modeling in two courses: database systems and software engineering. In software engineering the students just get an introduction to data modeling. They can read an ER model but not create one. In the database systems course the students must master both reading and creating ER models. In order to make a database application that satisfies a customer's need, the developer must be able to create a model that the customer can verify to ensure a usable application.

In the database course students learn SQL and ER modeling. Teaching and learning SQL is straight forward. Given a description of the tables, attributes, data types, and keys, students have no problem using SQL to implement the database. The problem comes when the students have to define the tables, attributes, data types, and keys – the entity relationship model – for the application. This is a much more difficult task because modeling is not a concrete operation – it starts with a blank paper.

My Process for Teaching Data Modeling

In my course I teach SQL first and then ER modeling. The textbook [Rob & Cornel] used this order in their first three editions. They switched the order in editions 4 and 5 but I still prefer SQL first. Students build a database from a data dictionary. They see tables, primary keys, and foreign keys. They use SQL's data definition and data manipulation language. They are able to master SQL. They are able to use SQL in Oracle SQL Plus, Posgress, and Access and most students find and use capabilities beyond standard SQL.

The next challenge is to design the database using an ER model. In the SQL exercises students use the designs that were given – now they have to understand a problem and make a design to solve the problem. In the course they have multiple modeling assignments. As individuals they do an in-class exercise ER model, several small ER models from the textbook, and an ER model based off an order form. As members of a

team they do a large problem from the textbook and a real problem where their customer is a campus or community group.

Note on CASE tool: The students use [Visible Analyst Workbench] to create their ER models and data dictionary. The tool is straight forward and gives professional looking products – graphic models and data dictionary reports. The CASE tool also has the capability to generate schema but most teams elect not to use this feature.

Note on reverse engineering an existing form: I select an order form from a mail order catalogue. This year we used the order form in the Lands End catalogue. Reverse engineering a form into a database is a good way to start any real project. The designers see the existing business data and establish a realistic ER model as a starting point.

Note on large project from the textbook: The textbook has several nice larger problems. I selected a college motor pool problem. In this problem the students go from design through implementation. This ties the whole process together.

Note on teamwork: When students are trying to master large concepts sometimes they can master the concept with the help of their peers. The teams tend to spend considerable time discussing their models and the quality of the team models is an improvement over the individual models.

Note on real project: The projects are from the university or local community. This year the projects were the Math Education Library, SIGADA Conference Registration, and a library for Computer Science Projects and Papers. Two teams are assigned to each project and the finished product is usually an Access database where the database is in good shape but the interface still needs some work. They also deliver a specification document that contains the ER model and functional requirements. Students take pride in their projects and I believe that the project ties together all they have learned in the course.

The Modeling Problem

Even with all this work on ER modeling, many students still struggle. The high-level modeling concept seems very difficult. The first ER model quiz was quite a disappointment to me. I realized (again this year) that the students were not grasping the concepts. Most could find major entities but they had trouble with primary keys and foreign keys. Even though they had mastered the concept of uniqueness and joins in SQL, these concepts did not transfer over to design. I believe that the students were trying hard. They wanted to learn and were working hard on the homework problems. What is the problem? Why do they have such problems with high-level concepts?

In addition, in our program at the University of Northern Iowa the students spend one year learning object oriented concepts and these high level concepts have great similarities to data modeling. In fact, [Coad and Yourdon] state that OO analysis is based

on the marriage on ER modeling and OO programming. From this background I expect some ability to grasp the high level concepts of data modeling.

This year I decided to try to identify the problem. While grading this miserable quiz, my teaching assistant, Pat Burke, and I started to form a survey. We tried to identify experiences in our students' background that impacted their ability to learn these concepts. I gave the survey on the same day I returned the not-so-great quiz. I explained my frustration and included my opinion that everyone was trying hard and that their inability to "get" the material was not because they were lazy or stupid but might be because something is lacking in their background. The survey was an attempt to identify the problem.

The Survey

In this section I will introduce the questions and describe their purpose. The questions are in Comic Sans, italic font, and the discussion in plain text.

1. *Name* _____

This was not a blind survey. The student names are needed because the answers are tracked to their grades. Primarily the grades are tracked back to the not-so-great quiz.

2. *Age* _____

Age is an interesting question. Learning theory states that the ability to abstract or use abstraction is one of the last mental skills we develop. It usually develops in the late teens to early twenties and some people unfortunately never develop the ability to abstract. [Brainerd]

3. *How many Computer Science courses have you taken?* _____

The number of computer science courses gives us some idea of the background of the student.

4. *Have you taken the Programming Languages course?* ___ Yes ___ No

My teaching assistant, Pat, suggested this question. He believes that his ability to abstract was positively affected by this course.

5. *Were you introduced to data modeling in another class (such as Software Engineering)?* ___ Yes ___ No

I also teach the Software Engineering course and introduce ER modeling as one of the 3 major modeling types in that course. Even though the students make several models in

Software Engineering, I expect acquaintance with the technique rather than mastery of the technique. It is possible that the previous introduction to modeling might help the students?

6. *Have you use other graphic models? ___ Yes ___ No*

If yes, do you believe you mastered the models?

___ Totally ___ Somewhat ___ No

What models?

Students could use other graphic modeling techniques in other courses. For example, in Software Engineering they use state models and data flow diagram. In object oriented courses they may have used class diagrams. I believe that mastering one high-level, abstract modeling technique should help in learning another.

7. *How many semesters of mathematics have you taken? ___*

Again Pat suggested this question. He noted that mathematics is a study of the abstract and that mastery of mathematics abstractions may carry over into data modeling.

8. *Do you or have you used databases at work? ___ Yes ___ No*

If yes, how much of your work is/was with databases?

___ Most ___ Some ___ Little

How much of you work is/was with developing databases?

___ Most ___ Some ___ Little

I believe that work is a great learning environment. Many of our students work and several use database systems in their work.

9. *Do you believe that the database course will give you knowledge that you will use in your future work? ___ Lots ___ Some ___ Little ___ None*

Need is a great motivator. This question tries to understand the students' motivation to master ER modeling.

10. *How confident are you in your data modeling skills?*

___ Very ___ Moderate ___ Low

The final question asks students to judge their own comfort and confidence in data modeling.

The Results and Discussion

Quiz Results

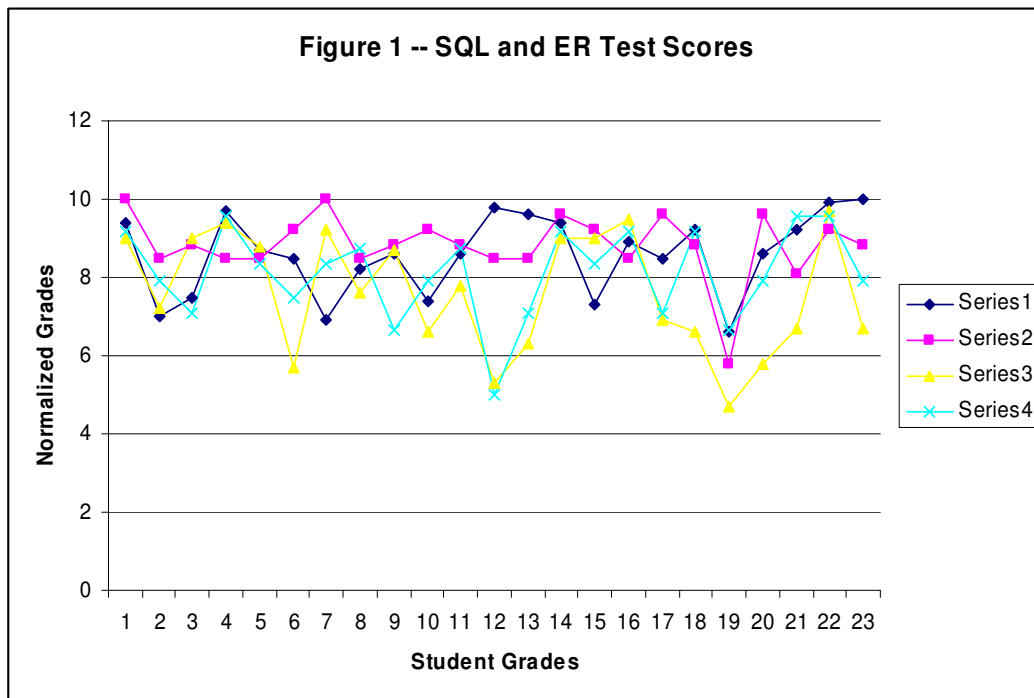


Figure 1: SQL and ER Test Scores

Figure 1 shows a comparison between test scores for SQL and ER modeling. Series 1 is the SQL 10 point quiz and Series 2 is the 26 point SQL portion of the final normalized to 10 points. Series 3 is the ER modeling quiz that I found so distressing. Series 4 is the 24 point ER modeling portion of the final normalized to 10 points. The averages of the SQL quiz and final were 8.7 and 8.8 respectively. The averages of the ER modeling quiz and final were 7.6 and 8.1 respectively.

Of course it is difficult to compare these test scores. The difficulty of the test questions and grading is not comparable. Grading of the first ER quiz was probably very generous. Grading SQL is straight forward. Syntax mistakes are mistakes. Grading ER models is more subjective. Most grading was based on proper use of components of the model but some grading was based on the representation of the problem.

The graph shows that, at least to my expectations, almost all the students mastered SQL but had more difficulty with ER modeling. It is also obvious that at least one student did not master anything and several students (1, 14, 16 & 22) mastered everything.

Questionnaire Results

The remainder of the graphs compares student's answers with their first quiz scores. I selected the first quiz scores because at the time of this quiz students had worked hard on modeling but many had not mastered nearly as much as expected. By final time, more students had improved their modeling skills.

The results show no clear correlations between the students' success and background. The sample size of 23 may be limiting. I will use the same questionnaire in my next class to increase the sample size.

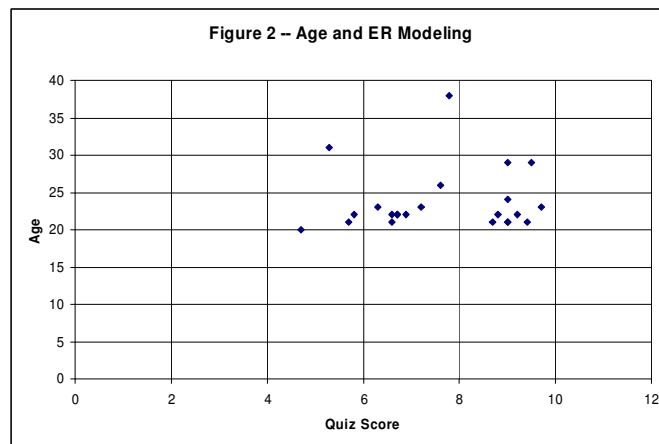


Figure 2: Age and Quiz Score

Figure 2 shows little correlation between age and difficulty modeling. I expected a better result here. No outliers were eliminated. If the two older students were eliminated, we might start to see a pattern.

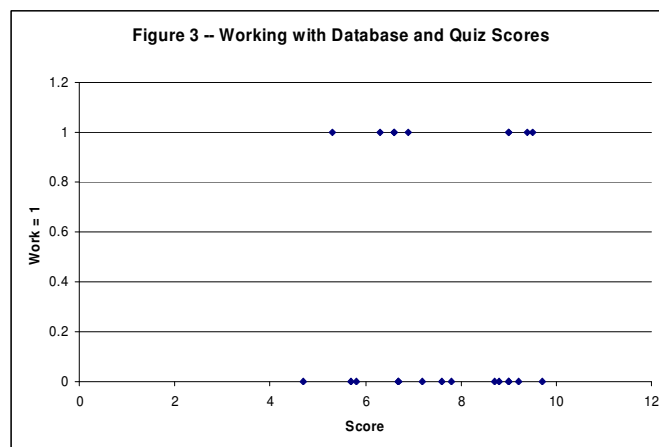


Figure 3: Working with Databases and Quiz Scores

Figure 3 shows little correlation between students work background and their results. Only 3 people reported working mainly with databases and 2 of them are the top 2.

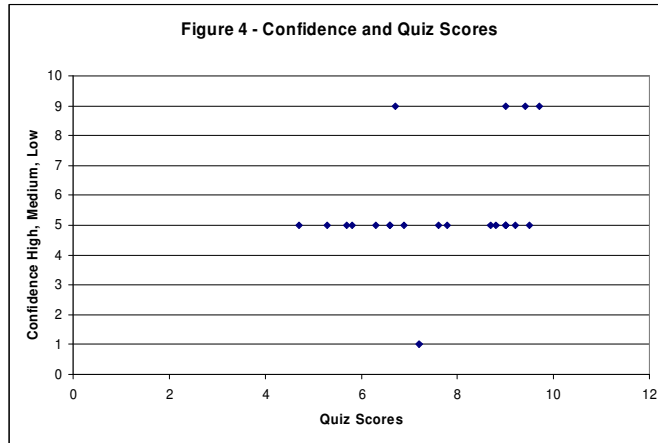


Figure 4: Confidence and Quiz Scores

Figure 4 shows the students' confidence. It suggests that most students understand how well they grasp the concepts.

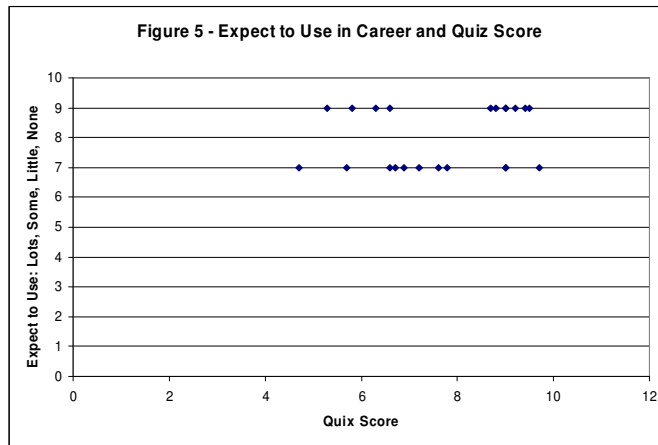


Figure 5: Expect to Use in Career and Quiz Scores

Figure 5 shows that the students all expect to use database systems in their work. Ten expect to use database a lot and the rest expect that they will use them somewhat in their work. Because of this result I believe that the students have an interest in the material and are trying to master modeling. No one in the course believes they will have little or no contact with database systems in their careers.

The following table, Table 1, shows the remainder of the survey question answers ordered on the quiz grade. There are no clues as to why some people master modeling and some do not from this data. The number of computer science and math courses made

no difference. Taking a programming language course did not relate to success. Previously seen models made no difference. In our curriculum students could have been introduced to models but were not expected to master any type of modeling.

ER Quiz	Number of Computer Science Courses	Programming Language Course	Have Seen ER Before	Have Seen Other Models	Number of Math Classes
4.7	5	no	no	no	3
5.3	55	yes	yes	yes	4
5.7	10	yes	yes	no	2
5.8	16	no	no	no	7
6.3	16	no	yes	no	
6.6	10	yes	yes	no	3
6.6	12	yes	yes	yes	7
6.7	12	no	yes	yes	8
6.9	10	yes	yes	no	6
7.2	15	no	no	no	3
7.6	6	no	no	no	4
7.8	5	yes	no	no	2
8.7	15	yes	yes	yes	4
8.8	12	no	yes	no	4
9	15	no	yes	yes	4
9	9	yes	no	yes	2
9	15	yes	no		
9	5	yes	yes	no	2
9.2	20	no	yes	no	8
9.4	10	no	yes	yes	6
9.5	20	no	yes	yes	
9.7	4	no	yes	no	4

Table 1: Results from the remainder of the survey questions.

Conclusion

Even though the survey showed no clear relationships between mastering ER modeling and students backgrounds, I believe that the questions are appropriate. The number of students, 22, is low. I will use the survey again the next time I teach the Database Systems course. Also, I am not a statistician. Eliminating outliers may improve my results but selecting and justifying the selection of outliers is out of my scope. I prefer to show you the whole story.

In addition, I believe that modeling is something that not everyone can do. Not all people are capable of the abstraction necessary to model. Not all people can leave behind the details of implementation and see the big picture. I believe that all my students understood the importance of modeling. When they did their final project – a real world

project – they saw the value of capturing the customer’s need and the clean relationship between their model and the database application that they created. Even when understanding the importance of modeling not all of the students will be able to become principal analyst/designer in their careers. On the other hand, I believe that all of them will be able to work successfully with a team and add to the analysis and design of large systems.

References

Brainerd, C., *Piaget's Theory of Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

Coad, P and Yourdon, E, *Object –Oriented Analysis*, Yourdon Press, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

Rob,P and Coronel, C, *Database Systems: Design, Implementation, and Management, 5th Edition*, Course Technology, Thompson Learning, 2002.

Visible Analyst Workbench, Visible Systems Corporation, 300 Bear Hill Road, Waltham, MA.