Using Analogical Problem Construction As An Advance Organizer To Teach Advanced Database (SQL) Nomenclature

Robert J. Mills, Ph.D., Utah State University, USA

ABSTRACT

Although business faculty have an important teaching responsibility to prepare students for professional positions in industry, very few have any formal training in instructional design. Analogical problem construction and advance organizers are powerful design techniques used to link prior knowledge to new material. Unfortunately, the use of analogies as a formal teaching strategy is disappointingly low. This study examines the use of analogical problem constructions as an advance organizer strategy to teach advanced database (SQL) concepts.

Keywords: Information Systems (IS) Education; Structured Query Language (SQL); Analogy Learning; Analogical Problem Construction

INTRODUCTION

In Robert Fulghum's book, Uh Oh – Some Observations from Both Sides of the Refrigerator Door, he presents a story titled, Yes I Can. Fulghum begins his story talking about asking a kindergarten class if they can write, and of course all hands go up. Regrettably, when the story gets to college students, eventually no hands go up, not one. Fulghum then retorts what happened from kindergarten to college (Fulghum, 1991)?

IS managers and business professors have a responsibility to teach current and future IS professionals discipline-specific specializations. Professional opportunities in structured query language (SQL), big data, and analytics is particularly in demand with serious hiring shortages expected in these high-demand areas (Columbus, 2014; Gordon, 2013; Soat, 2014). Unfortunately, most IS managers and business professors have no formal academic background in educational methods or instructional design. Instructional design considerations address critical components of instruction such as motivation, sequencing, scaffolding, assessment, and using prior knowledge to improve retention.

Ausubel (1960) introduced advance organizer (see table 1) as a formal concept to address bridging new knowledge to an individual's prior knowledge. Prior researches on advance organizers suggest analogies to prior knowledge help "foster insight into new material" (Gentzer et al., 1997, p. 4). The goal of an advance organizer is for the learner to activate prior knowledge to better understand the analogy being studied. (Zheng, Yang, Garcia, & McCadden, 2008).

Table 1. Advance Organizer Considerations (Cellucci, Layman, Campbell, & Zeng, 2011)

- A short, abstract textual statement
- A bridge that connects new knowledge with prior knowledge
- Used to introduce a new module
- Outlines new knowledge
- Helps students structure new information
- Encourgages transfer
- Provides opportunities for critical thinking

Anagogical problem construction is another technique that addresses linking new knowledge to existing knowledge. Analogical problem construction provides the learner an opportunity to construct his or her own analogous problems to better understand the underlying knowledge and add their own experiences to the solutions (Bernardo, 2001).

Prior research on analogical learning supports this activity as a way to improve transfer of knowledge (Catrambone, 1989; Seyihoglu & Ozgurbuz, 2015). A study by Mills, Dupin-Byrant, and Johnson (Mills, Dupan-Bryant, & Johnson, In Press) examined analogical problem construction on students in an introductory SQL class and found perceived improvements on knowledge acquisition, understanding, and recall.

Unfortunately, the use of analogies as a formal teaching strategy is disappointingly low (Haskell, 1989). Developing instructional strategies, particularly in areas that prepare students to make data-driven decisions is critically important to help produce qualified future data scientists ((Olsen & Dupin-Bryant, 2012). The purpose of this paper is to illustrate the potential of taping into learner's prior knowledge by incorporating analogical problem construction as an advance organizer into an advanced database course. Advance organizers, and analogical problem construction both leverage prior knowledge for learning new material (Bassok, 1990). The hope is those teaching challenging concepts will carefully consider the value of tapping into a learner's prior knowledge to enhance the learning experience. Although the analogical problem constructions in this paper are centered on database (SQL) concepts, the instructional strategy is appropriate across disciplines.

ANALOGICAL PROBLEM CONSTRUCTION ACTIVITY

The initial idea for using an analogy problem construction in a database management class was primarily a consideration of an IS student's typical interview for a professional position. Graduates are typically required to describe database concepts topics (i.e., common table expression) as well as write structured query code (SQL) on a white board during the interview process. The analogical problem construction activity as an advance organizer was created to address the first interview challenge of describing complex database concepts. Hands-on labs or practicums provide effective practice for the second interview requirement (Mason, 2013).

The take-home activity included three parts and students were provided a week to complete the assignment (See Appendix A.). Approximately 50 senior-level students taking an elective advanced database course participated. The activity was approved by the universities Institutional Review Board (IRB) and permissions were obtained to incorporate student examples into this paper. In addition, students' first names and last initials were included to provide appropriate recognition.

First, students were asked to define ten database terms (See Table 2). For instance, a case expression is a conditional logic expression. Second, students were asked to write a sample SQL code incorporating the database term and commenting on specific code elements using the comment feature of Microsoft Word. Finally, students were asked to create analogies based on their own prior experiences to describe the database concepts displayed in Table 2. An analogical problem construction example was provided based SQL CASE expression, which is "a scalar expression that returns a value based on conditional logic" (Ben-Gan, 2012, p. 53). A CASE expression is a bit like a vending machine. When a selection is made, the machine dispenses the first match. It is great for a fixed number of options and even provides an ELSE option when no match is found. In the case of a vending machine, your money would be refunded. Finally, the student is tasked with creating an analogy based on their own knowledge to bridge the new knowledge.

Grading for the activity was broken down into two parts. First, 40% of the grade was based on an accurate definition and coding sample of the SQL concept being addressed. The remainder of the grade (60%) was based on the overall quality, accuracy, and creativity of the analogy. Quality included an analogy that used correct grammar an sentence structure. Accuracy focused on whether the analogy actually correctly represented the SQL concept being addressed. Creativity points were assigned based on the unique approach to describing the SQL concept.

Table 2. Database (SQL) Concepts Included in Analogical Problem Activity

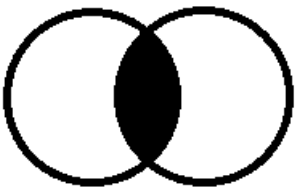
- Top & Offset Filter
- Correlated Subquery
- Common Table Expressions
- Views
- Derived Tables
- Stored Procedures
- Triggers
- UNION & UNION ALL
- INSERECT
- EXCEPT

RESULTS

Students had one week to complete the analogy problem constructions and the submissions were due on the day of a major SQL exam. The analogies provided interesting insight into the past knowledge structures that could not be replicated by the professor. Some analogies were also clear enough for someone without a background in SQL to understand. The following analogy is an example of the SQL INTERSECT concept, which according to set theory and given set A and set B, "is the set of all elements that belong to A and also belong to B (Ben-Gan, 2012, p. 194). Figure 1 illustrates the SQL INTERSECT using a Venn Diagram.

Figure 1. Venn Diagram of SQL INTERSECT

Imagine you are looking for your future wife and you have two qualities you are looking for. The first is that she's attractive and the second is that isn't crazy. So have a list of all those who are attractive but some of them are just too crazy. And vice versa you have a list of people who aren't crazy but some of them really aren't that attractive. What you want is both attractive and not crazy and intersect will get you there. Same with being a DBA. That will get you attractive non-crazy women. ***- T. Crandall***



Based on the student's analogical problem construction, the code that would illustrate the INTERSECT clause analogy code might look something like:

```
SELECT      FirstName, LastName
FROM        POTENTIAL_WIFE
WHERE       Appearance = 'attractive'

INTERSECT

SELECT      FirseName, LastName
FROM        POTENTIAL_WIFE
WHERE       Personality <> 'crazy';
```

SQL analogies for common table expressions, correlated subqueries, and stored procedures are included in Table 3.

Table 3. Additional (SQL) Analogy Problem Construction Examples: Creative

COMMON TABLE EXPRESSIONS • Common table expressions are like mowing your lawn with a pair of scissors your entire life, and then realizing that you have a lawn mower. Everyone starts off learning derived table expressions. Then you take Bob Mills's 4330 class and realize that CTE's do the same thing, but they are more organized, easier to write, and can be referred to multiple times! I LOVE CTE's now. - **C. Kendall**
CORRELATED SUBQUERY • A correlated subquery is similar to the movie, Inception, because the film is about dreams within dreams. Correlated subqueries are queries within queries. It can almost be paradoxical. If you have ever had a dream where you woke up and thought you were awake, but then woke up again, then you know what I am talking about. - **T. Crandall**
STORED PROCEDURE • Think of the Jetson's. There was a menial task that had to be performed daily by each individual (showering, getting dressed, getting ready for the day.) Instead of doing this manually every single day, they just invented a "stored procedure", or machine, that does it for them! They just have to stand on the right spot, and push the button (or execute the procedure) and BOOM! They are showered, dressed, and ready for the day. **- A. Olsen**

Additional SQL analogies for union/union all, triggers, and top/offset functions are included in Table 4.

Table 4. Additioual (SQL) Analogy Problem Construction Examples: Humorous

UNION & UNION ALL • Two people get married. They decide that they both love their stuff. They decide to keep it all! The wife (named UNION) decides that they can throw away the husbands toaster, tv, couch, and lamp because she already has those same things. The husband (named UNION ALL) decides that he doesn't care if they have 2 microwaves. That just means twice the hot-pocket preparation preparedness. He keeps everything. **- A. Olsen**
TRIGGERS • A trigger is like when you set a trap alarm for someone in your house. For example, let's say that you place a wire by your room that when someone crosses it a hammer will swing from the ceiling and make forceful contact with the perpetrators head as a reminder to that person that they should not be sneaking around your room. The hammer swinging was automatically triggered by their actions without you needed to do anything. **- T. Crandall**
TOP/OFFSET FUNCTION • If you had a cookie jar and used the top function you could select the top (in this case 5) cookies and take them to eat. The Offset function allows you to skip a number of the cookies on the top and take the next few (which can be specified) to be eaten. (Let's be honest. If raisin cookies are on the Top then the offset fetch feature is your new best friend). - **T. Wood**

DISCUSSION AND TEACHING IMPLICATIONS

One potential argument about analogical problem constructions being used as an advance organizer is that advance organizers are typically a pre-instruction technique. However, given the close alignment with many of an advance organizer's characteristics described in Table 1 such as a bridge that connects new knowledge with prior knowledge, outlines new knowledge, helps students structure new information, encourages transfer, and provides opportunities for critical thinking, it seems reasonable to consider analogical problem construction a type of advance organizer.

In order to link prior knowledge to new knowledge with analogical problem construction, learners must have at least a precursory understanding of the concepts being learned. From this experience, taking advantage of a student's prior knowledge to learn SQL is beneficial. Further, what better way to create this link than to have the student, not the instructor, come up with the analogies?

Students are bright, creative, and bring a variety of prior knowledge and experiences to the classroom. Analogy problem construction was a formal instructional strategy directly targeted at linking this prior knowledge with new database concepts and terminology.

To summarize the analogy problem construction SQL activity, student Andrew Olsen provides excellent insight:

> Creating my own analogies helped me to better understand new advanced database concepts in several different ways. Not only did I have to really think about the concept and strive to understand it in order to come up with an analogy, but analogies have a way of helping you remember concepts for longer periods of time, especially if you're the one that came up with the analogy. I feel pretty confidently that coming up with my own analogies for database concepts helped me to learn and understand them better than any other assignment we had. The only thing that proved as useful for me was running queries on my own and watching how they worked while I was running them.
>
> There are few things in the world as frustrating as looking at a query full of advanced database concepts, and having no clue how to interpret it. Something that helped me tremendously was learning the individual functions in a non-technical way. By analyzing each function individually and creating analogies to explain them, I was able to gain a deeper understanding of SQL as a whole.
>
> One of the most difficult things of any technical task, is trying to interpret what the computer is actually doing. No amount of 1's and 0's will ever make sense in my mind. However, when I try to explain what the functionality of a line of code does, by creating an analogy, I find that I don't have to memorize anything technical at all. By remembering something silly about the Jetson's, I'll never forget what a stored procedure is.
>
> I've found this method of learning not only valuable in classroom Database concepts, but in an actual professional environment as well. I will continue to call upon this strategy as long as I continue to learn.

One potential modification to the analogy assignment not implemented during this activity related to addressing analogies that do not accurately reflect the SQL concept being addressed. In the future, identifying the logic errors in the analogies could be used as a teaching tool. The class could help identify the logic errors and correct the analogy syntax to correctly reflect the SQL Concept.

Perhaps with business professors using instructional techniques that tap into students existing knowledge structures to learn new concepts such as SQL, university students will be more likely to announce, "Yes I can!"

AUTHOR BIOGRAPHY

Robert J. Mills is an Associate Professor of Management Information Systems in the Jon M. Huntsman School of Business at Utah State University. His research interests include computer-based learning environments, knowledge transfer, and MIS education. Bob Mills has consulted on technology-based training projects for a variety of organizations including Silicon Graphics International (SGI), EnergySolutions Arena / Utah Jazz, International Center for Captive Insurance Education (ICCIE), and IBM.

REFERENCES

Ausubel, D. P. (1960). The use of advance organizers in learning and retention of meaningful material. *Journal of Educational Psychology, 51*(267-272).

Bassok, M. (1990). Transfer of domain-specific problem-solving procedures. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*, 522-533.

Ben-Gan, I. (2012). *Microsoft SQL Server 2012 T-SQL Fundamentals*. Sebastopol, CA: O'Reilly Media, Inc.

Catrambone, R., & Holyoak, K. J. . (1989). Overcoming contextual limitations on problem-solving transfer. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 1147-1156.

Cellucci, L. W., Layman, E. J., Campbell, R., & Zeng, X. (2011). Integrating Healthcare Ethical Issues Into IS Education. *Journal of Information Systems Education, 22*(3), 215-224.

Columbus, L. (2014). Where Big Data Jobs Will Be In 2015. *Forbes.* Retrieved from http://www.forbes.com/sites/louiscolumbus/2014/12/29/where-big-data-jobs-will-be-in-2015/

Fulghum, R. (1991). *Uh-oh - Some Observations From Both Sides Of The Refrigerator Door.* . New York, New York: Random House.

Gentzer, D., Brem, S., Fergeson, R., Markman, A., Levidow, B., Wolff, P., & Forbus, K. (1997). Analogical Reasoning and Conceptual Change: A Case Study of Johannes Kepler. *Journal of Learning Science, 6*(1), 3-40.

Gordon, J. (2013). Big Data, Analytics and the Future Of Marketing and Sales. *McKinsey: Digital Advantage*.

Haskell, R. E. (1989). A Cognitive Theory of the Origin and Development of Equivalence Transformations. *Metaphor and Symbolic Activity, 4*(4), 247-277.

Mason, R. T. (2013). A Database Practicum for Teaching Database Administration and Software Development at Regis University. *Journal of Information Technology Education: Innovations in Practice, 12*, 159-168.

Mills, R. J., Dupan-Bryant, P. A., & Johnson, J. D. (In Press). Examining the Impact of Analogical Problem Construction and Learning Styles on SQL Knowledge Acquisition. *Journal of Information Systems Education*.

Olsen, D. H., & Dupin-Bryant, P. (2012). Business Intelligence and Information Systems: Enhancing Student Knowledge In Database Courses. *Review of Business Information Systems, 16*(1), 1-14.

Seyihoglu, A., & Ozgurbuz, I. E. (2015). Analysis of Analogies in Geography Textbooks. *Education and Science, 40*, 163-179.

Soat, J. (2014). As Big Data Booms, SQL Makes a Comeback. *Forbes.* Retrieved from http://www.forbes.com/sites/oracle/2014/09/08/as-big-data-booms-sql-makes-a-comeback/

Zheng, R. Z., Yang, W., Garcia, D., & McCadden, E. P. (2008). Effects of multimedia and schema induced analogical reasoning on science learning. *Journal of Computer Assisted Learning, 24*, 474-482.

APPENDIX A. SQL Analogy Assignment Instructions

Using the following advanced SQL terms/concepts,

1. **Provide textbook definition of term.**

EXAMPLE: CASE EXPRESION - A CASE expression is a scalar expression that returns a value based on conditional logic….

2. **Show CODE example and using comments (Review Tab, New Comment) describe rules, syntax, and other potential unknowns about the term/concept.**

```
SELECT categoryid
    CASE categoryID
    WHEN 1 THEN 'Beverages'
    WHEN 2 THEN 'Food'
    ELSE 'Unknown Category'
END AS categoryname
FROM Production.Products;
```

Comment [M1]: Notice CategoryID, returns a scalar value.

Comment [M2]: For every WHEN, you need a THEN.

Comment [M3]: This is the alias created.

3. **Describe term to someone (using an analogy if possible) with NULL (just kidding, ZERO) SQL experience. For instance:**

A CASE expression is a bit like a vending machine. When a selection is made, the machine dispenses the first match. It is great for a fixed number of options and even provides an ELSE option when no match is found. In the case of a vending machine, your money would be refunded.

1. Top & Offset Filter
2. Correlated Subquery
3. Common Table Expressions
4. Views
5. Derived Tables
6. Stored Procedures
7. Triggers
8. UNION & UNION ALL
9. INTERSECT
10. EXCEPT

NOTES