ISSN: 2473-4901
v9 n5966

Teaching Database Normalization: Do Prerequisites Matter?

Kevin J. Slonka
kslonka@francis.edu
Computer Science & Cybersecurity Department
Saint Francis University
Loretto, PA 15940, USA

Neelima Bhatnagar
bhatnarg@pitt.edu
Information Sciences Department
University of Pittsburgh at Greensburg
Greensburg, PA 15601, USA

Abstract

Filling the large gap in literature, this research offers a comparison of student performance in the introductory database course at two different universities where the students entered the course having completed different prerequisites. Performance was compared between students who previously took their university's CS1 course, CS2 course, Discrete Math course, and those who did not. The hypotheses were tested using quantitative methods and produced significant, albeit perplexing, results. These results were not enough for the researchers to make any solid claims, instead laying the foundation for future research to investigate this particular issue with a much bigger population.

Keywords: database, normalization, programming, discrete math, pedagogy

1. INTRODUCTION

Designing relational databases is difficult for students, especially for those without any previous design experience (Thompson & Sward, 2005). This is especially the case when many textbooks disagree on how these concepts should be taught (Carpenter, 2008). Year after year, the researchers, as well as their colleagues, express their disdain for how they perceive the results of normalization lessons and assignments; usually these are the lowest student grades in the course. The database administration concepts (i.e., the use of a database), such as the various SQL statements that must be written to get data in and out of a database, are much more easily comprehended by students than the design aspects (i.e., properly organizing data according to the relational model).

Different universities and different degree programs have varying prerequisite requirements for the database course. Some universities do not require any programming courses as a prerequisite (Robert Morris University, 2023; University of Pittsburgh Greensburg, 2023). Some universities require at least one programming course that teaches procedural programming, such as Java, C++, etc. (Indiana University of Pennsylvania, 2023; Saint Francis University, 2023). Other universities take databases one step further and split the course into multiple courses (Lock Haven University, 2023), such as a lower-level course that focuses on database administration and an upper-level course that focuses on database design (Mount Aloysius College, 2023).

This research focuses on exploring an observation by Kung & Tung (2006) that hasn't fully been explored in the past two decades, whereby programming knowledge and abstract thinking are the keys to understanding database design. Two offerings of the database course will be compared: one without a programming prerequisite and one with a single procedural programming prerequisite. Such a comparison will allow the researchers to investigate the primary hypothesis:

H_1: Procedural programming experience does not affect student scores on database normalization assignments.

2. REVIEW OF THE LITERATURE

Many have found the design of relational databases to be a difficult concept for students to grasp (Thompson & Sward, 2005; Kung & Tung, 2006; Hingorani et al., 2017), especially when those students have no previous experience with database design or have experience with other types of design/modeling. These concepts are further exacerbated by the inconsistencies in various notation styles for the common diagramming technique, the Entity Relationship Diagram (ERD). The meaning of specific symbols is not intuitive, the size of the ERDs can grow quite large, and were an instructor to switch textbooks the notation style might change drastically. Thompson & Sward (2005) attempt to alleviate these concerns by inventing a new ERD notation style; however, this style has not been widely adopted. The process of designing a database ERD first is preferred by students (Hingorani et al., 2017).

Beyond the mere diagramming of a relational database, the process of normalization is also a challenge for educators to teach. A complete understanding of the process of normalization often requires more programming and algorithm design experience than most students have at the time of taking a database course or than they will ever have depending on their degree field (e.g., IT/IS students typically take less programming courses than CS students) (Kung & Tung, 2006).

An additional problem with teaching normalization is that there is disagreement among the authorities about which normal form should be achieved. While it is typically agreed that third normal form (3NF) is the minimum to ensure a properly functioning database, there are higher levels that can, and sometimes must, be achieved (e.g., 4NF, BCNF, 5NF, DKNF, etc.). The question comes down to a return on investment for the amount of time a designer wishes to invest in the normalization process (Carpenter, 2008).

Some researchers moved beyond the topics themselves and focused on the teaching method for database courses. Many studies suggested that traditional methods of teaching database courses lead to information overload, leading students to only comprehend the most basic of principles that are not developed enough to fully encompass the way they are implemented in industry. The concepts and the programming (SQL) are too abstract to be easily understood. Teaching that provided visualizations and allowed students to interact proved to be the most effective (Hamah et al., 2019; Folorunso & Akinwale, 2010; Jaimez-Gonzalez & Martinez-Samora, 2020). Also of use was the Problem Based Learning (PBL) approach, but this approach required the most work on the part of the instructor to create differing assignments for each student for each lesson (Sastry, 2015).

3. METHODOLOGY

Two sections of a database management class were studied. One section was taught at a regional campus of an R1 university in Western Pennsylvania that does not require prerequisite programming knowledge (Group 1) and the other section was taught at a Western Pennsylvania Catholic university that requires one prerequisite programming course (Group 2). Students in Group 1, although not required to take a programming course before enrolling in the database course, were checked to see if they had taken their university's programming courses. Students in Group 2, due to the prerequisite, were ensured to have taken at least their university's CS1 course. Students were also checked to denote whether they had taken their university's CS2 course or Discrete Math course. Specific breakdowns are given in the Results section. This specific gathering of demographic data allowed the researchers to compare not based on the aforementioned Groups but based on the collection of students having completed the various courses. The sum of all participants led to this study having an n=42.

The researchers coordinated their content and grading. Both sections taught the same concepts in the first five weeks of the course culminating with normalization being taught during weeks four and five. Student grades for Exam 1 (the normalization exam) were compared between the groups of students who had and had not completed the CS1 course in order to determine the effect of prior procedural programming experience, accepting or rejecting this study's primary hypothesis:

H_1: Procedural programming experience does not affect student scores on database normalization assignments.

The additional demographic data also allowed the researchers to amend their hypotheses by adding the following two:

H_2: Object-oriented programming experience does not affect student scores on database normalization assignments.

H_3: Discrete math experience does not affect student scores on database normalization assignments.

4. RESULTS

The demographics of the population, shown in Table 1 show that the groupings of those taking each of the three prerequisite courses is fairly evenly distributed.

	n	%
Total	42	100.0
CS1		
Yes	29	69.0
No	13	31.0
CS2		
Yes	18	42.9
No	24	57.1
Discrete Math		
Yes	10	23.8
No	32	76.2

Table 1: Demographics

Although the initial impetus for this research was to examine the performance difference between students with and without procedural programming experience (that which is gained from the CS1 course), the researchers realized from the literature, as well as anecdotal evidence, that other courses (particularly CS2 and Discrete Math) teach skills that are useful to the database programmer. Thus, data pertaining to the students' completion of all three courses were gathered. This lead to the opportunity of expanding a single hypothesis study into a three-hypothesis study.

Analyzing H_1: Procedural programming experience does not affect student scores on database normalization assignments

Due to the data for the CS1 variable being nominal with two groups and the data for the Exam 1 score being scale, the appropriate statistical test was the independent samples T-test (Pallant, 2010). Shown in Table 2, Levene's Test for Equality of Variances produced a significant result. With the variances for the two groups being different, the results from the "Equal variances not assumed" portion of the T-test were used. In normal circumstances, where the population size is greater than 100, the typical alpha level of 0.05 is used to determine statistical significance. In the case of smaller populations, Stevens (1996) suggested using a more relaxed alpha level to achieve the necessary power of the statistical test. The alpha level used for this test was 0.10.

Levene	
F	6.814
Sig.	0.013
T-Test	
t	1.810
Two-tail Sig.	0.078
Effect Size	
Eta squared	0.076
Cohen's d	0.433

Table 2: H_1 Analysis

Given the two-tailed significance value of 0.078, the null hypothesis is rejected and the alternative hypothesis (procedural programming experience has an effect) is accepted.

Also calculated with the T-test were two different variations of effect size: eta squared and Cohen's d. The most common calculation for effect size with this test is eta squared. Using Cohen's (1988) guidelines (small=.01, medium=.06, large=.14) we can conclude that the effect size of H1 was medium, meaning that it would be noticeable to the naked eye. Cohen's d is another common calculation for effect size. This calculation was also run in order to corroborate the results from the eta squared calculation. Cohen (1988) also proposed guidelines for interpreting Cohen's d (small=.2, medium=.5, large=.8) and, using that scale, the effect size is also medium.

Analyzing H_2: Object-oriented programming experience does not affect student scores on database normalization assignments

Due to the data for the CS2 variable being nominal with two groups and the data for the Exam 1 score being scale, the appropriate statistical test was the independent samples T-test (Pallant, 2010). Shown in Table 3, Levene's Test for Equality of Variances did not produce a significant result. With the variances for the two groups being equal, the results from the “Equal variances assumed” portion of the T-test were used. As above, the alpha level used for this test was 0.10.

Levene	
F	3.227
Sig.	0.080
T-Test	
t	1.114
Two-tail Sig.	0.272

Table 3: H_2 Analysis

Given the two-tailed significance value of 0.272, the null hypothesis is accepted (object-oriented programming experience does not have an effect).

Analyzing H_3: Discrete math experience does not affect student scores on database normalization assignments

Due to the data for the Discrete Math variable being nominal with two groups and the data for the Exam 1 score being scale, the appropriate statistical test was the independent samples T-test (Pallant, 2010). Shown in Table 4, Levene’s Test for Equality of Variances did not produce a significant result. With the variances for the two groups being equal, the results from the “Equal variances assumed” portion of the T-test were used. As above, the alpha level used for this test was 0.10.

Levene	
F	3.703
Sig	0.061
T-Test	
t	1.956
Two-tail Sig.	0.058
Effect Size	
Eta squared	0.087
Cohen’s d	0.708

Table 4: H_3 Analysis

Given the two-tailed significance value of 0.058, the null hypothesis is rejected and the alternative hypothesis (discrete math experience has an effect) is accepted.

Both eta squared and Cohen’s d were calculated to determine the size of the effect. With the eta squared value being 0.087 and Cohen’s d being 0.708 we can conclude that the effect size lies between medium and large, meaning that it would be noticeable to the naked eye.

5. DISCUSSION

The results of this study are not immediately comprehensible without some further data. Table 5 attempts to summarize the results along with the group means and Table 6 shows the grade distribution for each institution.

	Mean	
	With Course	**Without Course**
H_1: Procedural Programming		
Reject	75.017	79.731
H_2: Object-Oriented Programming		
Accept	Not significant	
H_3: Discrete Math		
Reject	70.750	78.266

Table 5: Analysis Summary

	Univ. 1	**Univ. 2**
Grade		
A	1	2
B	7	8
C	4	14
D	2	0
F	4	0

Table 6: Grade Distribution

Although H_1 and H_3 had significant results, the data leads us to the intuitively opposite conclusion (once we account for the means): taking the corresponding course accounts for *lower* grades. Were it true that taking courses makes students *less knowledgeable*, the entire education system would be turned on its head. Instead, this allows us to synthesize the results with the limitations of the study to arrive at the essence of this study’s contribution.

As previously stated, the sample size in this study was low such that a modification to the alpha value was warranted. Were the standard alpha value of 0.05 used, none of the results would have been significant. A study with a population >100 would be necessary in order to use the globally accepted alpha value and achieve stronger, more generalizable results.

Additionally, as Table 6 depicts, the grade distribution at University 1 is more normal than at University 2, where no students received grades lower than C. Despite the coordination of the researchers, both grading all exams from both institutions and averaging them together, there appears to have been a difference that led to such skewed grades. The exam construction was left to each professor, ensuring that both exams met the same learning objectives although by different means. Having both professors grade all exams from all institutions and averaging the scores should have accounted for any differences. It is apparent that in order for a study such as this to be successful a standardized exam with a standardized rubric must be used by all researchers in order to ensure that every student is graded consistently no matter the institution or professor.

6. CONCLUSION

This study, despite having two areas of the research design that should be altered before being replicated, uncovered multiple items that need to be investigated further in order to enhance the learning environment provided for our students. Although the results were not what the researchers expected, differences between students who took CS1 and Discrete Math before taking the database course were uncovered. With a stricter research design, these differences can be better analyzed, which would suggest the proper prerequisite courses for a university's database course, thus ensuring that students are set up for success instead of struggle due to the lack of vital knowledge.

7. REFERENCES

Carpenter, D. A. (2008). Clarifying normalization. *Journal of Information Systems Education, 19*(4), 379-382.

Cohen, J. W. (1988). *Statistical power analysis for the behavioral sciences, 2nd edition*. Lawrence Erlbaum Associates.

Folorunso, O. & Akinwale, A. (2010). Developing visualization support system for teaching/learning database normalization. *Campus-Wide Information Systems, 27*(1), 25-39.

Hamzah, M. L., Rukun, K., Fahmi, R., Purwati, A. A., Hamzah, H., & Zarnelly. (2019). A review of increasing teaching and learning database subjects in computer science. *Revista Espacios, 40*(26), 6-14.

Hingorani, K., Gittens, D., & Edwards, N. (2017). Reinforcing database concepts by using entity relationships diagrams (ERD) and normalization together for designing robust databases. *Issues in Information Systems, 18(*1), 148-155.

Indiana University of Pennsylvania. (2023, March 16). *Computer Science, BA*. https://www.iup.edu/academics/find-your-degree/programs/macs/ug/computer-science-ba.html

Jaimez-Gonzalez, C. R. & Martinez-Samora, J. (2020). DiagrammER: A web application to support the teaching-learning process of database courses through the creation of E-R diagrams. *International Journal of Emerging Technologies in Learning, 15*(19), 4-21.

Kung, H. & Tung, H. (2006). An alternative approach to teaching database normalization: A simple algorithm and an interactive e-learning tool. *Journal of Information Systems Education, 17*(3), 315-325.

Lock Haven University. (2023, March 16). *Applied Computer Science and Information Systems Major.* https://www.lockhaven.edu/checksheets/documents/bscomputerscience22017.pdf

Mount Aloysius College. (2023, March 16). *Course Descriptions: Computer Science*. http://catalog.mtaloy.edu/content.php?filter%5B27%5D=CSIT&filter%5B29%5D=&filter%5Bcourse_type%5D=-1&filter%5Bkeyword%5D=&filter%5B32%5D=1&filter%5Bcpage%5D=1&cur_cat_oid=14&expand=&navoid=1549&search_database=Filter

Pallant, J. (2010). *SPSS survival manual*. McGraw Hill.

Robert Morris University. (2023, March 16). *Course Catalog: INFS4240 – Database Management System*. https://sentry.rmu.edu/OnTheMove/wpCrsehist.get_results?itrm=202280&iCrse=INFS4240&iShowReg=Yes&iShowhdr=1&icalledby=WPCRSEHIST&it=&iattr=&ipage=701&redir=

Saint Francis University. (2023, March 16). *Courses of instruction: Computer Science.* https://catalog.francis.edu/content.php?filter%5B27%5D=CPSC&filter%5B29%5D=&filter%5Bcourse_type%5D=-1&filter%5Bkeyword%5D=&filter%5B32%5D=1&filter%5Bcpage%5D=1&cur_cat_oid=15&expand=&navoid=707&search_database=Filter&filter%5Bexact_match%5D=1

Sastry, K. S. (2015). An effective approach for teaching database course. *International Journal of Learning, Teaching, and Educational Research, 12*(1), 53-63.

Stevens, J. (1996). *Applied multivariate statistics for the social sciences, 3rd edition*. Lawrence Erlbaum.

Thompson, C. B. & Sward, K. (2005). Modeling and teaching techniques for conceptual and logical relational database design. *Journal of Medical Systems, 29*(5), 513-525.

University of Pittsburgh Greensburg. (2023, March 16). *INFSCI 1022 – Database Management Systems*. https://catalog.upg.pitt.edu/preview_course_nopop.php?catoid=216&coid=1170275