Journal of Information Systems Education, Vol. 17(3)

An Alternative Approach to Teaching Database Normalization: A Simple Algorithm and an Interactive e-Learning Tool

Hsiang-Jui Kung
Department of Information Systems
Georgia Southern University
Statesboro, GA 30460
hjkung@georgiasouthern.edu

Hui-Lien Tung
Division of Business
Paine College
Augusta, GA 30901
tungh@mail.paine.edu

ABSTRACT

The relational data model is an important concept covered in the systems analysis and design course. It has been difficult to motivate students to learn database normalization because they find the subject dry and theoretical. An alternative approach has been developed to give students an easy-to-follow algorithm and an interactive, hands-on e-learning tool. The approach is suitable for database normalization in systems analysis and design and in database management courses. This paper describes the alternative approach and its effectiveness in teaching database normalization. The effectiveness of the approach has been evaluated in an exercise and a survey. The paper shows that the approach reduces error rate and increases students' perceived ease, confidence and performance of the normalization approach.

Keywords: Database Normalization, Relational Data Model, Functional Dependency, Third Normal Form

1. INTRODUCTION

Database design is important in business software development since virtually every business application uses a database management system. Databases have to be normalized to the third normal form (3NF) when relational database management systems are used. Databases not normalized to 3NF will stumble upon insertion, deletion, and updating anomalies. Database normalization has been a well-developed field since the introduction of Codd's seminal work on normal forms in 1970. Bernstein (1976), Diederich and Milton (1988), Concepcion and Villafuerte (1990), and Rosenthal and Reiner (1994) proposed algorithms and tools to synthesize a normalized database using functional dependencies. Maier (1988) indicated that relational data model theory (normalization) tends to be complex for the average designers. Jarvenpaa and Machesky (1989), Bock and Ryan (1993), and Batra and Antony (1994) showed that the relational data model leads to poor designer performance. The students' poor performance of normalization indicated that teaching normalization is a challenge to IS/IT educators.

The traditional database normalization technique has often relied on the definition of normal forms. Some database textbooks include normalization algorithms to find the canonical cover by removing extraneous attributes of functional dependencies (FDs) and then converting each FD in the canonical cover to a relation/table (Silberschatz, Korth, and Sudarshan, 2002). The normalization algorithms often require extensive programming/algorithm backgrounds that most Information Systems (IS)/Information Technology (IT) students lack. Most systems analysis and design (SA&D) textbooks rely on the definition of normal forms in their coverage of database normalization (Hoffer, George, and Valacich, 2005; Avison and Fitzerald, 2002). A table is in first normal form (1NF) if each domain contains simple values. The second normal form (2NF) tables are in 1NF and non-key attributes depend on the whole key (no partial dependency). A table is in third normal form (3NF) if that table is in 2NF and non-key attributes do not depend on other non-key attribute(s) (no transitive dependency). Applying the traditional normalization, students need to find out which normal form a relation is in. If a table is in the first normal form but not 2NF, students have to remove those attributes

(from the 1NF table) that cause partial dependency to create another table/relation. This step will ensure all the tables are in 2NF. If a table is in 2NF but not 3NF, students have to remove those attributes causing transitive dependency to create another table. To master the traditional normalization technique, students have to understand the concepts of partial and transitive dependencies clearly.

Teaching database normalization in IS/IT classes is challenging since neither curriculum includes relational algebra or algorithms. Moreover, normalization requires practice, and students, therefore, have to spend considerable time in order to master the concept and, even then, are often not successful. This paper explores an alternative approach that contains a simple normalization algorithm and an interactive e-learning tool to improve IS/IT students' learning of database normalization. Using this approach, the instructor is able to present and demonstrate to students the normalization steps interactively. The e-learning tool may be accessed at any time via the Internet.

The main objective of this paper is to describe the alternative normalization approach and its effectiveness in teaching and learning about normalization. The remainder of this paper is organized as follows: Section 2 describes the alternative approach, and Section 3 illustrates research design and data collection procedures. The effectiveness of the alternative approach is evaluated and interpreted in Section 4. Retention of the normalization skill is tested in Section 5, and Section 6 concludes the paper.

2. THE ALTERNATIVE NORMALIZATION APPROACH

Many IS/IT students are confused by the definitions of 1NF, 2NF, and 3NF. They have problems differentiating those three normal forms and are puzzled by association between FDs and normal forms. Most students request an easier way to learn database normalization. To address the issue, we developed an alternative normalization approach that consists of a simple normalization algorithm and an interactive e-learning tool to help students' learning of normalization.

It has been noticed that decompositions happened when (1) attributes on the right-hand side of functional dependencies have more than one copy, and (2) the number of the decomposed relations is exactly the same as the number of functional dependencies (FDs) when the FDs are in closure. To decompose a relation into the third normal form, one simply eliminates extraneous attributes on the right-hand side of the functional dependencies. The simple normalization algorithm is easy to follow without an extensive background in algorithms. The following steps describe the simple normalization algorithm by using the example of a universal table T and a set of FDs: FD_1, FD_2, and FD_3. Attributes in bold and underline font are the primary key(s) of a relation (foreign key in dashed underline). The simple normalization algorithm is sound and complete when the set of FDs is non-trivial and in closure (see Appendix). The universal relation T ($\underline{\boldsymbol{A}}$, B, C, D, $\underline{\boldsymbol{E}}$, F)

FDs: $FD_1 : A \rightarrow B, C, D$

$FD_2 : B \rightarrow C, D$

$FD_3 : A, E \rightarrow B, C, D, F$

2.1 The Simple Normalization Algorithm

1. In every functional dependency, keep attribute(s) on the left-hand side intact.
2. Extraneous attributes on the right-hand side should be eliminated. This step is to eliminate partial dependencies and transitive dependencies. Repeated right-hand side attributes should be identified in all the functional dependencies; one copy of the redundant attributes should be kept and the others should be deleted. The rules of thumb about which copy of attributes to keep are:
   a. the attributes of FDs that have fewer attributes on the left-hand side should be kept (this step will eliminate partial dependency).

   Example: Attributes B, C, and D depend on part of the whole key (attribute A). Attributes B, C, and D in functional dependency FD_3 will be deleted, since attributes B, C, and D appear in FD_1 also and FD_1 has only one attribute on the left-hand side. A new functional dependency $FD_3' : A, E \rightarrow F$ is formed.

   b. when two FDs have the same number of attributes on the left-hand side, the determinants that have fewer attributes on the right-hand side should be kept (this step will eliminate transitive dependency).

   Example: Attributes C and D appear in functional dependencies FD_1 and FD_2. Attributes C and D are transitively dependent on Attribute A. Attributes C and D will be deleted from functional dependency FD_1, since FD_1 has more right-hand side attributes than functional dependency FD_2. A new functional dependency $FD_1' : A \rightarrow B$ is formed.

   c. Construct relations. The purpose of this step is to convert the functional dependencies without extraneous right-hand side attributes into relations. The new functional dependencies are as follows:

   $FD_1' : A \rightarrow B$

   $FD_2 : B \rightarrow C, D$

   $FD_3' : A, E \rightarrow F$

   The final normalized relations are exactly the same as the results of the decomposition algorithm and SA&D normalization techniques:

   - T_1 ($\underline{\boldsymbol{A}}$, $\underline{\boldsymbol{B}}$)
   - T_2 ($\underline{\boldsymbol{B}}$, C, D)
   - T_3 ($\underline{\boldsymbol{A}}$, $\underline{\boldsymbol{E}}$, F)

2.2 The e-Learning Tool

The e-learning tool is developed using a Java applet with the simple normalization algorithm described above. The tool has the following features:

- Main window: The main window contains fields to key in and display functional dependencies. There are buttons to submit functional dependency one at a time; to reset/clear the memory for another exercise/practice; to normalize the database and display the normalized relations; to display the step-by-step normalization process; and to demonstrate the usage of the tool (Figure 1).
- Result window: After keying all the functional dependencies and then pressing the "Normalize" button, the user will see the normalized result in the normalized relations window (Figure 2). The tool will normalize the database based upon the functional dependencies the user submitted. The tool normalizes any set of tables to 3NF if the set of functional dependencies used are non-trivial and closed. A functional dependency $A \rightarrow U$ is non-trivial if $A \cap U = \varnothing$. In other words, the left- and right-hand sides of a non-trivial functional dependency have no attributes in common. A functional dependency $A \rightarrow U$ is closed under a set of functional dependencies FD if U is the set of all attributes that are functionally dependent on A in a given FD. The sets of functional dependencies used in this paper do meet these requirements.
- Step-by-step window: After normalizing the database and then pressing the "Step-by-Step" button, the user will see the step-by-step window showing the step-by-step normalization process (Figures 3-6). Figure 3

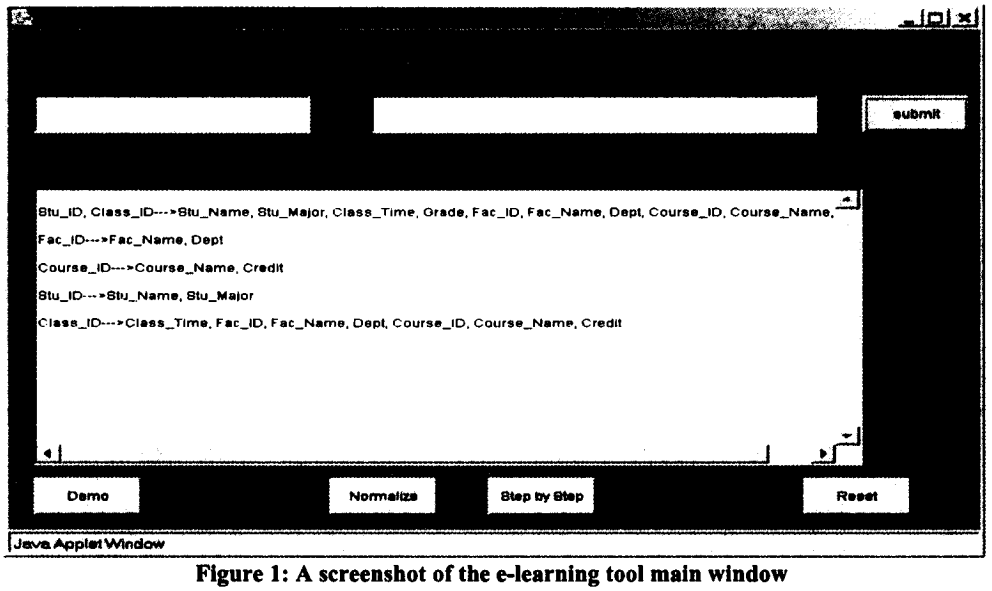

Figure 1: A screenshot of the e-learning tool main window

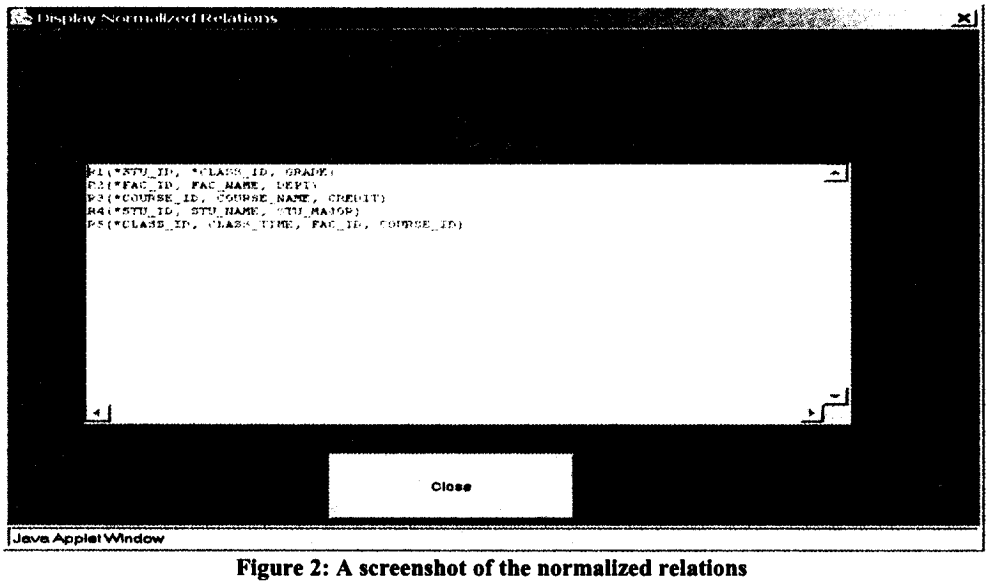

Figure 2: A screenshot of the normalized relations

shows the set of FDs that the user keys in. After pressing the "Next" button, the user will see the normalization to 2NF that eliminates partial dependencies (Figure 4). Going to the next step, the user will see the normalization to 3NF that eliminates transitive dependencies (Figure 5). Figure 6 shows the normalized result which should be the same as Figure 2. The user can navigate back and forth in the normalization process when he or she has a problem with it.

3. RESEARCH METHOD

The research framework is shown in Figure 7. Error rate and perception are the dependent variables. The model predicts that error rate will be affected by normalization approaches and designer experience. Our main interest is to identify the differences of error rate between normalization approaches and classes. As no prior empirical work has compared the two normalization approaches directly, it is difficult to predict which approach will result in lower error rate;

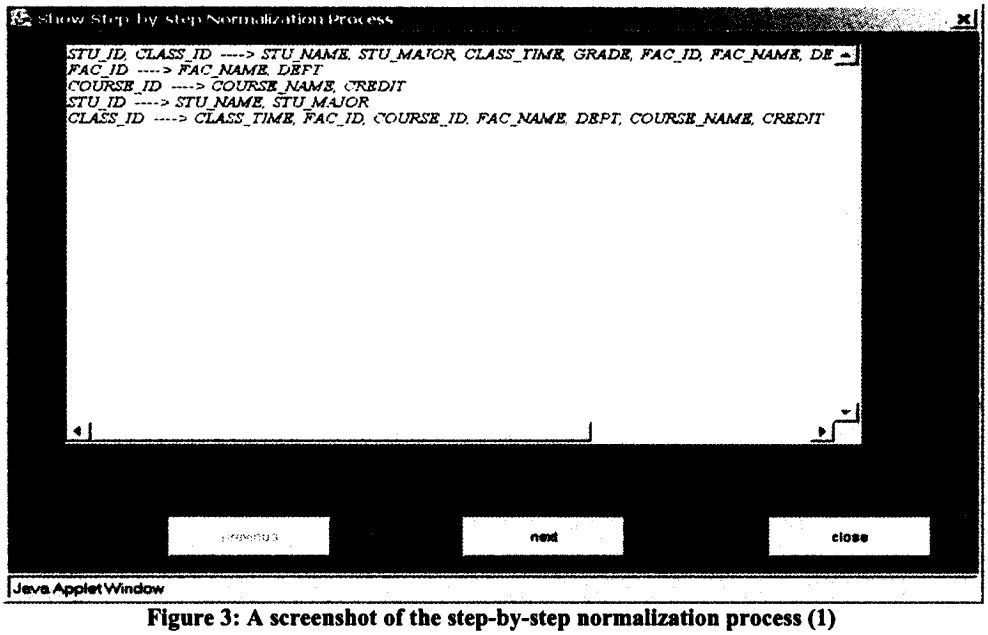

Figure 3: A screenshot of the step-by-step normalization process (1)

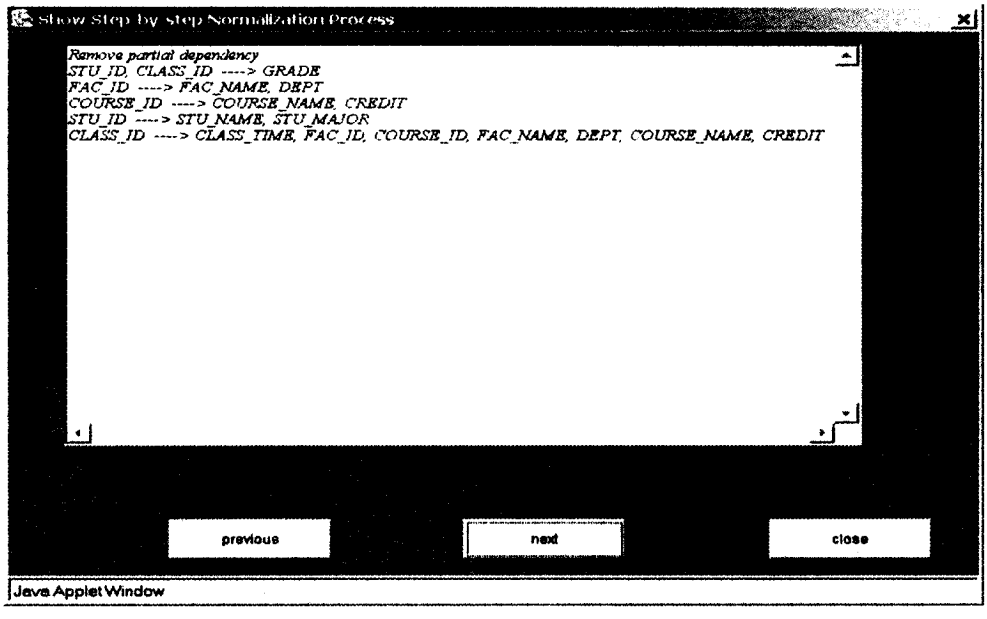

Figure 4: A screenshot of the step-by-step normalization process (2)

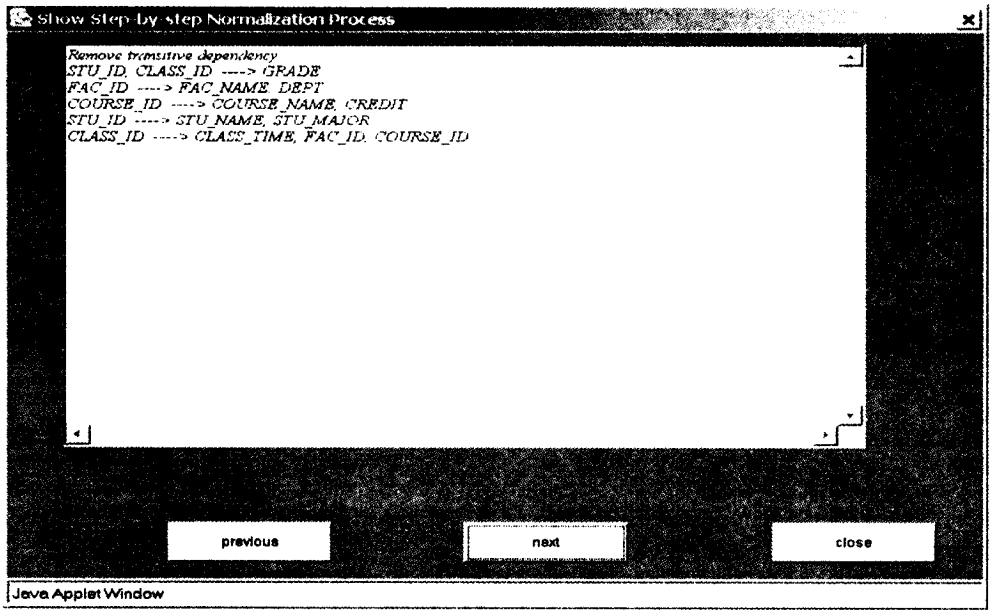

Figure 5: A screenshot of the step-by-step normalization process (3)

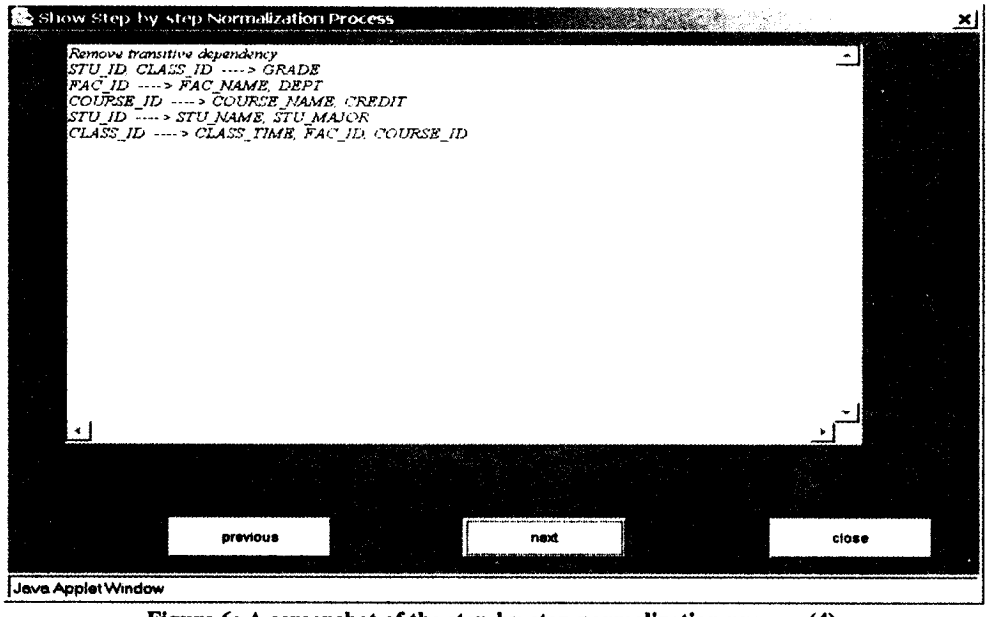

Figure 6: A screenshot of the step-by-step normalization process (4)

however, given that computer-aided teaching demonstrates. better learning results in many studies, it is plausible to support the notion that novice database designers practicing with the alternative approach will perform better than those using the traditional approach

The hypotheses (presented in null form) addressed in this study are as follows:

H_1 : No difference in subjects' error rate based on the different approaches will exist.

H_2 : No difference in subjects' error rate based on the different classes will exist.

H_3 : No difference in subjects' perception of the different approaches will exist.

H_4 : No difference in subjects' perception of the different classes will exist.

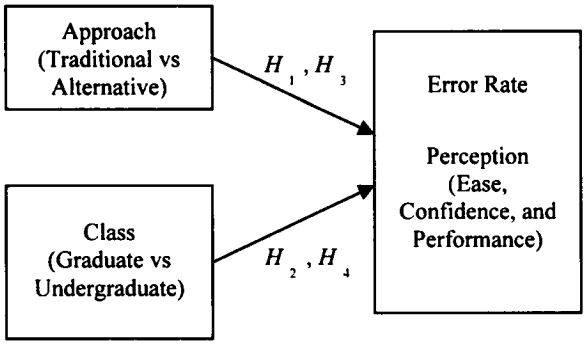

Figure 7: Research Framework

3.1 Dependent and Independent Variables

Dependent variables of the study are the subjects' error rate on an in-class exercise and their perceptions of ease, confidence and performance of the normalization approach. The error rate can be defined as the percentage of incorrect objects in a relation/table to the total objects of the relation/table. The objects of a relation are attributes, primary key(s), and foreign key(s). The error rate of each relation is denoted as equation (1). The overall normalization error rate is the average of the error rates of all relations, as shown in equation (2).

$$ErrorRate_i = \frac{Error_i}{Attribute_i + PrimaryKey_i + ForeignKey_i} \quad (1)$$

$$ErrorRate_{Overall} = \frac{\sum_{i=1}^{N} ErrorRate_i}{N} \quad (2)$$

The subjects' perceived ease, confidence, and performance are measured by using an instrument. The instrument, which was administered at the end of an in-class exercise, consisted of eight questions: one open-ended question for subjects' comments about the normalization approach at the end, four questions for demographic information (status, gender, undergraduate major, and number of database related courses taken) and three questions for their perceptions of ease, confidence, and performance as follows:

1. the ease of understanding the normalization approach;
   1 2 3 4 5
   Very Difficult Neutral Very Easy
2. the confidence of using the normalization approach;
   Very Little Neutral Very High
3. the performance on the in-class exercise using the normalization approach.
   Poor Unsatisfactory Satisfactory Good Excellent

One independent variable is the normalization approach. The traditional normalization approach refers to the approach used in a popular SA&D textbook (Hoffer et al, 2005). Subjects have to identify which normal form a relation is in. If a relation contains partial dependency (violating the definition of 2NF), subjects should remove those attributes causing partial dependency in order to form other relations that satisfy the 2NF definition. If a relation contains transitive dependency, subjects should remove those attributes causing transitive dependency to form other relations that satisfy the 3NF definition. The alternative approach contains the steps of the simple normalization algorithm as described in Section 2 and uses an e-learning tool. Considering the different levels of database design experience and learning motivation within the subject population, we added "class" as another independent variable in the research framework. Subjects in MBA business systems analysis (BSA) class have some data modeling experience, while the majority of subjects in undergraduate SA&D classes have no such experience.

3.2 Subjects

In a southeastern public university in the United States, SA&D is one of the core courses of IS/ IT program, and BSA is one of the elective courses of the MBA program. SA&D is offered every semester with multiple sections and BSA is offered every Spring semester with only one section. Undergraduate students can enroll in any section according to their schedule and/or preference. In Spring semester 2006, subjects enrolled in two sections of a junior level SA&D class, and an MBA BSA class participated in the experiment. Section A of the SA&D class with 17 subjects applied the alternative approach. The traditional approach was applied to Section B of SA&D with 7 subjects and to the BSA class of 9 subjects. The 15-week class met twice weekly for SA&D and weekly for BSA. The Hoffer et al. (2005) textbook was used to cover the feasibility study, data modeling, process modeling, and physical design. Subjects spent two weeks on the database normalization processes (four 75-minute sessions for SA&D and two 150-minute sessions for BSA). The instructor spent one week explaining the importance of database normalization and demonstrating the normalization approach with examples in all three classes. In the first half of the following week, subjects worked on two practice exercises using the normalization approach learned in the previous week. Subjects were aware of the in-class exercise when they participated in the experiment and were encouraged to practice the learned normalization approach after class. During the second half of the week, subjects applied the normalization approach to solve an in-class exercise.

3.3 In-Class Exercise

A universal relation *Publishing* and a set of FDs as follows were given to subjects. ISNN and AuthorID (bold and underlined) are the primary keys (composite key) of the universal relation. The subjects' task was to normalize *Publishing* to 3NF. Subjects had to identify all the primary keys and foreign keys in all the normalized relations.

Publishing (PublisherID, PublisherName, Address, **ISBN**, BookTitle, Category, Loyalty, **AuthorID**, AuthorName, AuthorPhone)

FDs:

PublisherID → PublisherName, Address

ISBN → PublisherID, PublisherName, Address, BookTitle, Category

AuthorID → AuthorName, AuthorPhone
AuthorID, ISBN → PublisherID, PublisherName, Address, BookTitle, Category, AuthorName, AuthorPhone, Loyalty

3.4 Experiment Procedure

Prior to the in-class exercise, the subjects completed a research participation consent form and an anonymity agreement. Next, subjects read the exercise scenario and applied the approach they learned to work on the exercise. A summary of the normalization approach was provided to the subjects for quick reference. Subjects using the alternative approach were not allowed to use the interactive e-learning tool for the exercise. Once the subjects announced that they had finished the exercise, they were provided with the survey instrument to measure their perceptions of the applied normalization approach.

The two raters graded the exercise of each subject by comparing subjects' answer with the correct solution as follows:

Publisher (**PublisherID**, PublisherName, Address)
Book (**ISBN**, PublisherID, BookTitle, Category)
Author (**AuthorID**, AuthorName, AuthorPhone)
Write (**ISBN**, **AuthorID**, Loyalty)

Each relation has its objects of attributes, primary keys and foreign key. Primary keys are shown in bold and solid underline and the foreign key is in dashed underline. The total number of objects of relation *Publisher* is 4 (3 attributes and 1 primary key), relation *Book* 6 (4 attributes, 1 primary key, and 1 foreign key), relation *Author* 4 (3 attributes and 1 primary key), and relation *Write* 5 (3 attributes and 2 primary keys). Each relation was graded separately. A missing attribute, an extra attribute, or a failure to identify a primary or foreign key was counted as one error. The error rate for each individual relation was calculated as the ratio of the total number of errors to the total number of objects in the relation. The overall design quality was computed as the average of all the individual relation error rates. The raters graded independently. The final overall error rate of each subject was the average of the two raters' scores.

4. RESULTS AND ANALYSES

The experiment was an unbalanced factorial design. Subjects in the two SA&D classes applied one approach each, and the BSA students applied only the traditional approach (because only one BSA section was offered in Spring 2006). Thirty-three subjects completed the in-class exercise and the survey. Calculated means and standard deviations of the overall error rates are shown in Table 1. The standard deviations were quite uniform, varying only between 0.157 and 0.1872. Eight subjects (47%) in SA&D Section A delivered zero error results, none in SA&D Section B, and four (44%) in BSA.

A two-way between-groups ANOVA was performed (see Table 2). The main effect of approach was significant at p=0.050. The main effect of class was not significant (p=0.115). Thus null hypothesis H_1 was rejected, but hypothesis H_2 could not be rejected. The normalization approach had a statistically significant effect on overall error rate. Subjects using the alternative approach produced a lower overall error rate than did the subjects using the traditional approach. The interaction effect between approach and class could not be tested since it was an unbalanced factorial design.

Approach	Class	Mean Error Rate	Std. Deviation	N
Traditional	SA&D B	28.21%	.1800	7
	BSA	14.31%	.1872	9
Alternative	SA&D A	12.57%	.1570	17

Table 1: Means and standard deviations of approach and class (in-class exercise)

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Approach	.121	1	.121	4.187	**.050**
Class	.076	1	.076	2.629	.115
Error	.869	30	.029		

Table 2: Test of between-subjects effects with dependent variable (Error Rate)

The next set of analyses was performed to provide a general overview of student perceptions of the traditional and the alternative approaches. The means and standard deviations for the three perception items of the survey instrument were calculated for approach and class (see Table 3). With three being the mid-point on the scale, Table 3 illustrates that students generally viewed the alternative approach more positively than the traditional approach.

Perception	Approach	Class	Mean	Std. Dev.	N
Ease	Traditional	SA&D B	1.57	1.134	7
		BSA	3.11	1.054	9
	Alternative	SA&D A	3.06	1.298	17
Confidence	Traditional	SA&D B	2.00	1.528	7
		BSA	3.22	1.093	9
	Alternative	SA&D A	3.47	.943	17
Performance	Traditional	SA&D B	2.57	.787	7
		BSA	3.78	.667	9
	Alternative	SA&D A	3.71	1.105	17

Table 3: Means and standard deviations of perceptions

One-way multivariate analyses of variance (MANOVA) was used to compare the impact of the independent variables (approach and class) on the pattern of subject responses. The

one-way MANOVA revealed that the pattern of means for the alternative approach observed in Table 3 is statistically significant than that of the traditional approach. The multivariate *F* value of Approach (p=0.037) observed in the MANOVA indicates that this is indeed the case. Although the subjects in the graduate class generally had a higher perception than undergraduates, the difference was not statistically significant (p=0.075, see Table 4). Thus null hypothesis H_3 was rejected, but hypothesis H_4 could not be rejected. The normalization approach has a statistically significant effect on subjects' perception of the ease, confidence and performance; and subjects using the alternative approach are more confident and believe they do better than those using the traditional approach.

Effect	F	Hypothesis df	Error df	Sig.
Approach	3.237	3.000	28.000	**.037**
Class	2.560	3.000	28.000	.075

Table 4: MAVOVA test

Approach	Subjects' Comments
Traditional	1. Normalize a relation to 3NF is difficult to me, since it's difficult to know which item depends on the other. 2. Normalizing into 2NF is the most difficult because it doesn't make complete sense to me yet. I think I need more time to understand and study it. I can't get 2NF. It is easier to go to 3NF because I have an idea of how the entities should be grouped. However, I still have problems even then. 3. Putting relations in 2NF is somewhat difficult because I have to avoid going straight to 3NF. Once I overcome that, I should be fine. 4. I find myself trying to condense straight from given information to 3NF, conversion into 2NF then into 3NF. This is probably because I am more confident with 3NF, but also 2NF seems odd to me; almost like I have not analyzed the situation enough. 5. The hardest part of bottom up design is getting to 2NF. Everything else is easy.
Alternative	1. The tool is helpful in that it helps you get rid of any redundancy that is in your functional dependencies. 2. It eliminates the guess work. 3. The step-by-step window is very helpful to show you how a particular relation is normalized.

Table 5: Subjects' comments on normalization approaches

Preference for the alternative vs. traditional normalization approaches was also observed in responses to the open-ended question on the survey instrument. Table 5 shows subjects' comments about the normalization approach. Most comments about the traditional approach were negatively toned. However, most comments about the alternative approach were positively toned. These comments suggest that the alternative approach helped subjects visualize the normalization process. Subjects using the traditional approach indicated that 2NF is the most difficult concept. Subjects using the alternative approach indicated that the e-learning tool's step-by-step feature helped them to learn normalization.

Table 6 summarized the means and standard deviations of the error rates of individual relations/tables. Subjects had higher error rates in relations *Book* and *Author* (partial dependencies). The results are consistent with subjects' comments.

Relation		Approach	
		Traditional	**Alternative**
Publisher	Mean	7.81%	8.82%
	Std. Dev.	0.2536	0.1755
Book	Mean	30.00%	16.47%
	Std. Dev.	0.3011	0.2029
Author	Mean	31.25%	17.65%
	Std. Dev.	0.3575	0.2990
Write	Mean	12.50%	7.35%
	Std. Dev.	0.2582	0.1470

Table 6: Means and standard deviations of individual relation (Error Rate)

The overall objective of the study was to find a better normalization approach. Smith (2002) reported that the past performance of undergraduate IT students has a positive impact on their current/future academic performance. Usually, graduate students have higher learning motivation and skills since they are in the top half of their undergraduate classes and have more experiences than undergraduates. We expected that graduate subjects will deliver lower error rate in the exercise, but the results do not support this assumption. Class had no significant impact on overall error rate. The undergraduate subjects using the alternative approach performed better (scored lower in the overall error rate) even when compared against more experienced graduate subjects. The most difficult concept in normalization found in the study was 2NF (partial dependency).

In terms of subjects' perception: the results show no significant differences of perceptions between different classes. From another perspective, subjects using the alternative approach thought that it is easy to learn normalization using such approach; they were confident about using that approach, and performed better. All in all, the alternative approach was found to lead to low error rate and was perceived as significantly an easy-to-learn, confidence-building, and well- performed approach.

5. RETENTION AND NORMALIZATION SKILL

While the experiment clearly shows the value of the alternative approach in learning normalization, the question remains whether this has a longer lasting effect. In particular,

how well do subjects do when they have to normalize a set of relations/tables without the benefit of access to their notes? Apparently, subjects in the traditional approach group are handicapped in performing normalization tasks based on the in-class exercise results. During the week following the in-class exercise, we showed subjects the other normalization approach. Subjects in the control group learned the alternative approach, and subjects in the treatment group learned the traditional approach. Subjects were allowed to choose the approach, based on their preference, in their exams. Subjects' normalization skills were tested in a closed-book (no usage of any electronic device) mid-term examination (two weeks after the open-note exercise) on the following normalization problem.

A universal relation *Spa*:

Spa (CustomerID, CustomerName, **ReservationID**, ReservationDate, Payment, **ServiceID**, ServiceTime, Preference, EmployeeID, ServiceName, Price, EmpName, EmpPhone, EmpAddress)

With the following FDs:

CustomerID → CustomerName

ReservationID → CustomerID, CustomerName, ReservationDate, Payment

ReservationID, ServiceID → ServiceTime, Preference, EmployeeID, ServiceName, Price, EmpName, EmpPhone, EmpAddress, CustomerID, CustomerName, ReservationDate, Payment

ServiceID → ServiceName, Price

EmployeeID → EmpName, EmpPhone, EmpAddress

The subjects' task was to normalize the universal relation *Spa* to 3NF. Subjects had to identify all the primary keys and foreign keys in all the normalized relations.

The normalized relations were as follows:

Customer (**CustomerID**, CustomerName)

Reservation (**ReservationID**, CustomerID, ReservationDate, Payment)

Reservation Item (**ServationID**, **ServiceID**, EmployeeID, ServiceTime, Preference)

Service (**ServiceID**, ServiceName, Price)

Employee (**EmployeeID**, EmpName, EmpPhone, EmpAddress)

Eighteen subjects in SA&D Section A completed the mid-term, five in SA&D Section B, and ten in BSA. The number of subjects who completed the mid-term does not match the number of subjects who completed the open-note exercise because some subjects missed the exercise session and some dropped out before the mid-term. All thirty-three subjects used the alternative normalization approach to solve the mid-term normalization problem. Table 7 summarized the means and standard deviations of the mid-term normalization problem results. Comparing Tables 1 with 7, we found that the error rate dropped from 16% to 4% in total, which is a 75% improvement. SA&D Section A's error rate dropped from 13% to 6%, a 50% improvement. SA&D Section B's error rate dropped from 28% to 0%, an 100% improvement. BSA's error rate dropped from 14% to 3%, an approximately 80% improvement. Nine subjects (50%) in SA&D Section A delivered zero error results, five (100%) in SA&D Section B, and seven (70%) in BSA. One reason for the improvements may be that subjects have been doing many normalization practices. Another reason for the improvements is the demonstration of the superiority of the alternative approach. Subjects in both SA&D Section B and BSA showed enormous improvement using the alternative approach.

Approach	Class	Mean Error Rate	Std. Dev.	N
Alternative	SA&D A	6.00%	.1052	18
	SA&D B	0%	.0000	5
	BSA	3.05%	.0749	10

Table 7: Means and standard deviations of the mid-term

6. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

An alternative normalization approach has been developed to enhance the teaching and learning of database normalization. The approach contains the easy-to-follow simple algorithm and the interactive e-learning tool available on the Web (http://www.georgiasouthern.edu/~hjkung/3NF). It was evaluated by students and found to be robust. Students' responses to the approach were mostly favorable. The students indicated that they had found the e-learning tool easy to use and noted that the step-by-step feature helped them gain an understanding of database normalization process. The approach had a positive impact on students' perceptions of the ease, confidence and performance.

These findings have many implications. Students can use the e-learning tool as a practice and drill tool to help them in learning database normalization. It is important to point out that we do not propose using the approach as the only coverage of normalization. Students can work on the normalization problems by themselves and use the e-learning tool to validate their answers. We believe that students need to understand the concept of normalization from the traditional approach and can practice the mechanical normalization steps using the simple algorithm and e-learning tool. Students can also use their normalization skills to validate their Entity-Relationship diagrams (ERD) that are error prone to novice designers (Batra and Wishart, 2004). The study suggests that the most common errors pertain to the partial dependency. Thus, IS/IT educators should spend a little more time explaining partial dependency.

We suggest several extensions to our research. Currently, the e-learning tool can handle small-size problems, e.g., a set of ten functional dependencies, which is adequate for teaching purposes. More features, e.g., 'Load,' 'Save,', and 'Print', are still under development. One extension of the future enhancement is to integrate normalization with ERD to show the connection between relational and ER models. Another extension of this study would be a longitudinal assessment of the normalization learning process from multiple institutions, so that we would have statistical tests for between- and within-subjects effect. The next extension would be to develop a perception instrument using multi-item scales to measure subjects' perceptions. Finally, we could refine the

error rate computation technique to weigh objects by the level of difficulties.

7. ACKNOWLEDGEMENTS

We are indebted to the reviewers for their detailed comments on an earlier version of this paper. Especially, their recommended title is appropriate for this paper.

8. REFERENCES

Avison, D. E. and Fitzgerald, G., (2002), *Information Systems Development: Methodologies, Techniques and Tools*, 3rd Ed., Lodon, UK: McGraw Hill.

Batra, D., Antony, S. R., (1994), Novice errors in database design, *European Journalof Information Systems*, 3 (1), 57–69.

Batra, D., and Wishart, N. A., (2004), Comparing a Rule-Based Approach with a Pattern-Based Approach at Different Levels of Complexity of Conceptual Data Modelling Tasks, *International Journal of Human-Computer Studies*, 61, 397-419.

Bernstein, P. A. (1976), Synthesizing Third Normal Form Relations from Functional Dependencies, *ACM Transactions on Database Systems*, Vol. 1. No. 4, pp 277-298.

Bock, D. B., Ryan, T., (1993), Accuracy in modeling with extended entity relationship and object oriented data models. *Journal of Database Management*, 4 (4), 30–39.

Codd, E. F., (1970), A Relational Model of Data for Large Relational Databases, *Communications of the ACM*, 13 (June), 377-387.

Concepcion, A. I., & Villafuerte, R. M., (1990), Expert DB: An assistant database design system, *Proceedings of the third international conference on Industrial and engineering applications of artificial intelligence and expert systems - Volume 1.*

Diederich, J. and Milton, J., (1988), "New Methods and Fast Algorithms for Database Normalization", *ACM Transactions on Database Systems*, 13 (3), 339-365

Hoffer, J. A., George, J. F., and Valacich, J. S., (2005), *Modern Systems Analysis & Design*, 4th Ed., Prentice Hall.

Jarvenpaa, S.L., Machesky, J.J., (1989), Data analysis and learning: an experimental study of data modelling tools. *International Journal of Man–Machine Studies*, 31, 367–391.

Maier, D., (1988), *The Theory of Relational Databases*. Computer Science Press: Rockville, MD.

Rosenthal, A., & Reiner, D., (1994), Tools and Transformations — Rigorous and Otherwise — for Practical Database Design, *ACM Transactions on Database Systems*, 19 (2), 167-211.

Silberschatz, A., Korth, H. F., and Sudarshan, S., (2002), *Database System Concepts*, 4th Edition, Boston, MA: McGraw Hill.

Smith, S. M., (2002), The Role of Social Cognitive Career Theory in Information Technology based Academic Performance, *Information Technology, Learning, and Performance Journal*, 20 (2), 1-10.

AUTHOR BIOGRAPHIES

Hsiang-Jui Kung is an assistant professor of information systems at Georgia Southern University. He received his PhD in Management from Rensselaer Polytechnic Institute in 1997. He joined Georgia Southern University full time in 2001. His research interests include systems analysis and design, database, e-business, and software evolution.

Hui-Lien Tung is an assistant professor of Management Information Systems in the Division of Business at Paine College. Her teaching and research interests include database management, system analysis and design, and web-based learning system.

APPENDIX: PROOF OF THE SIMPLE NORMALIZATION ALGORITHM

Notation: In what follows, capital letters (italic) represent non-empty sets of attributes.

Definition 1: A functional dependency $A \rightarrow U$ is non-trivial if $A \cap U = \varnothing$. In other words, the left-hand side and the right-hand side of a non-trivial functional dependency have no attributes in common.

Definition 2: A functional dependency $A \rightarrow U$ is closed under a set of functional dependencies FD if U is the set of all attributes that are functionally dependent on A given FD.

Theorem: If each functional dependency in a set of functional dependencies (FDs) is non-trivial and closed under FD, then the set of tables generated using the simple normalization algorithm is fully normalized up to 3NF.

Proof: The simple normalization algorithm reduces FD to a set FD', from which the tables are generated. Assume that the set of tables generated from FD' through the simple algorithm is not in 3NF. Then, at least one of the following must hold (definition of 3NF):

(i) There is a table T that violates 2NF (some non-key attributes depend on partial key(s));
(ii) There is a table T that violates 3NF (some non-key attributes depend on other non-key attributes).

Case (i)

The proof is by refutation. Suppose a table T violates 2NF, and that the primary key of T is a set of attributes A. FD' must contain functional dependencies

FD_i: $A \rightarrow U$ $(1 \leq i \leq n)$ and
FD_j: $D \rightarrow V$ $(1 \leq j \leq n)$

where $D \subset A$ and $V \subseteq U$ (definition of 2NF, and the fact that all functional dependencies are closed). Thus, D is a subset of A and U and V have attribute(s) in common.

By step 2(a), the common attributes V would have been eliminated from FD_i. Hence, we derive a contradiction and conclude that all tables are in 2NF.

Case (ii)

Suppose there is a functional dependency between non-key attributes in a table T. FD' must contain functional dependencies

FD_k: $B \rightarrow W$ and
FD_l: $C \rightarrow S$

where $C \subset W$ and $S \subseteq W$ (definition of 3NF). Thus, C and S are subsets of W.

Since FD_l is non-trivial, $C \cap S = \varnothing$. Since $C \subset W$, and there are attributes in C and hence in W, that are not also in S, $S \subset W$. Thus, S is a complete subset of W and S is the set of common attribute(s) in the right-hand side of FD_k and FD_l. Moreover, since S contains fewer attributes than W, by step 2(b), we would have eliminated S from FD_k. Hence, we derive a contradiction and conclude that no functional dependencies can be found between non-key attributes. Since we derive a contradiction for both cases, we conclude that the tables generated through the simple normalization algorithm are fully normalized up to 3NF.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096