3-1-2007

Comparing Two Bottom-up Database Design Methods

Hsiang-Jui Kung
hjkung@georgiasouthern.edu

Hui-Lien Tung

COMPARING TWO BOTTOM-UP DATABASE DESIGN METHODS

Hsiang-Jui Kung
Georgia Southern University
hjkung@georgiasouthern.edu

Hui-Lien Tung
Paine College
tungh@mail.paine.edu

Abstract

Bottom-up database design is a difficult task for novice database designers and is particularly challenging to teach. To address the problem, two methods—textbook and cookbook—have been suggested. The textbook method describes the Codd's normalization algorithm and data model visualization adopted by most textbooks. The cookbook method illustrates a simple and straightforward normalization algorithm and ER diagram mapping technique proposed by the authors. This paper describes the conceptual framework, experimental design, and results of a laboratory study that employed novice designers to compare the error rates of the two methods (between subjects) at two levels of task complexity. Results indicate that the cookbook method led to significant lower error rates.

Keywords: Data Model, Entity Relationship Diagram, Relational Model, Normalization

Introduction

Traditionally, database textbooks (Hoffer, Prescott & McFadden, 2005) have tackled conceptual data model problems using the top-down approach (e.g., Entity-Relationship (ER) modeling, Chen, 1976) or bottom-up approach (e.g., normalization) or both. A top-down approach assumes a higher order of processing, in which object instances possess a natural domain identity (e.g., are naturally recognized by end users as valid object types within that domain's ontology). A bottom-up approach to database design views the task of population identification as a process of generalizing object identity from examples of structural dependencies (bundling/categorizing attributes that appear to go co-occur). Although the top-down (ER) approach usually lead to better performance than the bottom-up approach, the ER approach is an error-prone task for novice designers (Batra, Hoffer, and Bostrom, 1990; Jarvenpaa and Machesky, 1989; Bartra and Wishart, 2004).

Most IS/IT textbooks describe the bottom-up database design as the integration of Bernstein's synthesis approach (1976) and Codd's (1970) seminal work on normal forms. With this approach, the database analyst initially focuses upon identifying functional dependencies within the domain. Functional dependencies are typically identified through close inspection of data structures within business documents, reports, and similar organizational artifacts (i.e., semantic artifacts). Once the (closure) set of functional dependencies has been identified, the database analyst then attempts to synthesize the data model based upon the formal rules of normalization. The textbook normalization algorithm has often relied on the definition of normal forms (Hoffer, George, and Valacich, 2005; Avison and Fitzerald, 2002). A table is in 1NF if each domain contains simple values. The 2NF tables are in 1NF and non-key attributes depend on the whole key (no partial dependency). A table is in 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 textbook normalization algorithm, students have to understand the concepts of partial and transitive dependencies clearly. If desired, the outcome of the normalization algorithm can then be represented in entity relationship diagram through foreign keys (e.g., ER model).

Teaching bottom-up database design in IS/IT classes is challenging since neither curriculum includes extensive computing theories. This paper explores an alternative approach that contains a simple and straightforward cookbook method to improve IS/IT students' learning of bottom-up database design. The main objective of this paper is to compare the textbook and cookbook methods and the effectiveness in teaching and learning about bottom-up database design. The remainder of this paper is organized as follows: Section 2 describes the textbook method, and Section 3 illustrates the cookbook method. Section 4 depicts research design and data collection procedures. The results is presented in Section 5 and Section 6 concludes the paper.

The Textbook Method

The textbook method contains two parts: normalization and data model visualization. The normalization steps will decompose relations to 3NF. The data model visualization will convert the normalized relations/tables to ER model. The textbook method is sound and complete based on these assumptions: (1) the universal relation covers a single business process domain, (2) the universal relation is in the first normal form (1NF) and (3) the set of functional dependencies are given and in closure. The example contains a universal relation/table T and a set of functional dependencies FD as the following:

T (**A**, B, C, **D**, E, F, G)

FD:

A → B, C, E
A, D → B, C, E, F, G
B → C
D → F

Attributes in bold and underlined are primary keys.

Normalize a universal relation/table

A) For every relation/table, determine which normal form the relation/table is in.
   - ⇨ Relation T is in the 1NF but not 2NF since T has partial dependencies.

B) Decompose every relation to 2NF if it's not in the 2NF. Cut and paste the non-key attribute(s) that cause partial dependencies to form other relation(s). Copy and paste the determinant(s) that cause partial dependencies to other relations as the primary key(s).
   - ⇨ The whole key in relation T is attributes A and D.
   - ⇨ Attributes B, C, and E depend on attribute A (part of key). So we cut attributes B, C, and E from relation T and paste these attributes to a new relation T_1. We also copy attribute A and paste to relation T_1. We bold and underline attribute A to make it the primary key of T_1. T_1 (**A**, B, C, E)
   - ⇨ Attribute F depends on attribute D (part of key). So we cut attribute F and paste to relation T_2. We also copy attribute D and paste it to relation T_2. We make attribute D the primary key of T_2. T_2 (**D**, F)
   - ⇨ The relation T becomes T (**A**, **D**, G) since we remove attributes B, C, E, and F. Attributes A and D remain in relation T since they are the primary keys.

C) Decompose every relation to 3NF if it's not in the 3NF. Cut and paste the non-key attributes cause transitive dependencies to form other relations. Copy and paste the primary key(s) to other relations.
   - ⇨ Relation T is in 3NF since it is in 2NF and has no transitive dependency.
   - ⇨ Relation T_1 is not in 3NF since attribute C depends on non-key attribute B (transitively dependent on attribute A). We cut attribute C and paste it to a new relation T_3. We also copy attribute B and paste it to relation T_3 and make it the primary key. T_3 (**B**, C) and T_1 (**A**, B)
   - ⇨ Relation T_2 is in 3NF since T_2 is in 2NF and has no transitive dependency.
   - ⇨ We have 4 relations: T, T_1, T_2, and T_3. T (**A**, **D**, G), T_1 (**A**, B, E), T_2 (**D**, F), and T_3 (**B**, C)

Draw ERD from normalized relations/tables

A) Every relation/table becomes an entity. Draw an entity for every 3NF relation, add attributes to the entity, and mark primary key(s). Identify foreign keys in every relation/table.

B) Connect two entities with a relationship (line) when they have common attribute.
C) Assign one-leg to the primary key and crow's foot to the foreign key in every relationship. Figure 1 shows the normalized ERD of the textbook method.

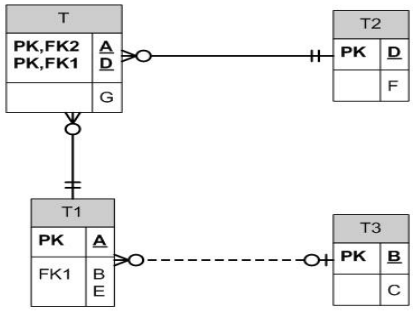

Figure 1: Normalized ERD of the textbook method

The Cookbook Method

The cookbook method is the extension of the alternative normalization technique (Kung and Tung, 2006). We use the same previous example to illustrate the cookbook method.

Normalize a universal relation/table

A) Count the number of attributes on left-hand-side (LHS) and right-hand-side (RHS) of every functional dependency. Don't change the numbers throughout the method.
   (1) A → B, C, E (3)
   (2) A, D → B, C, E, F, G (5)
   (1) B → C (1)
   (1) D → F (1)
B) Keep LHS attributes intact. All LHS attributes should be kept on the left-hand side, as is, even when the same LHS attributes appear on the LHS in another functional dependency.
   (1) A → B, C, E (3)
   (2) A, D → ~~B~~, C, E, F, G (5)
   (1) B → C (1)
   (1) D → F (1)
   No change should be made for the LHS attributes
C) Eliminate redundant attributes on RHS. Keep only one copy of all the RHS attributes and delete the additional ones.
   i) Keep the copy of attributes that has the smaller LHS number in that FD and delete the additional copies (this step will eliminate partial dependency).
      (1) A → B, C, E (3)
      (2) A, D → ~~B, C, E, F,~~ G (5)
      (1) B → C (1)
      (1) D → F (1)
   ii) When two FDs have the same LHS number, keep the copy that has the smaller RHS number and delete the additional copies (this step will eliminate transitive dependency).
      (1) A → B, ~~C,~~ E (3)
      (2) A, D → ~~B, C, E, F,~~ G (5)
      (1) B → C (1)
      (1) D → F (1)
   iii) Convert the FDs to relations. Make the LHS attribute(s) the primary key(s) of that relation.
      A → B, E
      A, D → G
      B → C
      D → F

⇨ R_1 (**A**, B, E), R_2 (**A, D**, G), R_3 (**B**, C), and R_4 (**D**, F)

Draw ERD from normalized relations/tables

A) Every relation becomes an entity. Draw an entity for every 3NF relation, add attributes to the entity, and mark primary key(s).
B) Connect entities using common attributes. Draw a relationship between two entities when the two entities have a common attribute.
C) Assign cardinalities to every relationship based on the common attribute.
   i) When the common attribute is the single primary key, assign a cardinality of 1 on that side of the relationship.
   ii) When the common attribute(s) of an entity is/are part of the primary keys, assign a cardinality of many toward that entity in the relationship.
   iii) When the common attribute of an entity is a non-key attribute, assign a cardinality of many toward that entity in the relationship. In this situation, the non-key attribute is a foreign key (FK). The normalized ERD is identical to Figure 1.

Research Method

The research framework is shown in Figure 2. Error rates of normalization and ERD are the dependent variables. The model predicts that error rates will be affected by design methods and levels of difficulty. Our main interest is to identify the differences of error rate between design methods (textbook versus cookbook) and levels (easy versus difficult). As no prior empirical work has compared the two design methods directly, it is difficult to predict which method will result in lower error rate.
The hypotheses (presented in null form) addressed in this study are as follows:

H_1: No difference in subjects' error rates based on the different methods will exist.

H_2: No difference in subjects' error rates based on the different levels will exist.

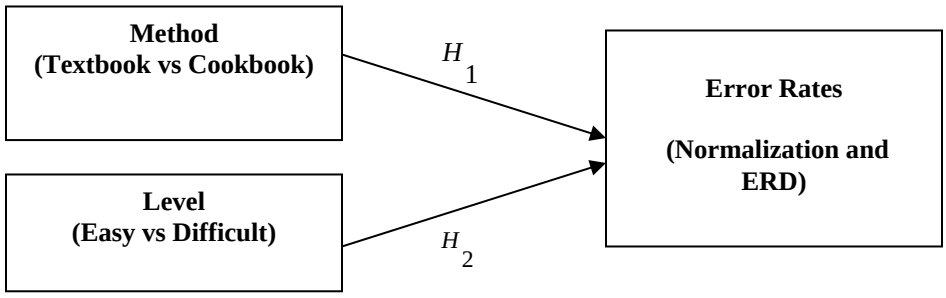

Figure 2: Research Framework

Dependent and Independent Variables

Dependent variable of the study is the subjects' error rates of two in-class exercises. We looked at the errors made in the two major "stages" of bottom-up database design: normalization and ERD modeling. The error rate is defined as the ratio of incorrect objects in normalized relations/ERD to the total objects of the relations/ERD. To examine the normalization error rate, we first calculate the error rate of each relation (Equation 1) and then find the average error rate of all relations (Equation 2). The objects of a relation used are attributes and primary key(s).
To examine the ERD error rate, we first calculate the error rate of each entity (Equation 3), and the error rate of relationships (Equation 4), and then calculate the average (Equation 5). The objects of an entity are attributes, primary key(s), and foreign key(s). The objects of a relationship are relationship and cardinalities.

$$ErrorRate_{Relation\_i} = \frac{Error_i}{Attribute_i + PrimaryKey_i} \quad (1)$$

$$ErrorRate_{Normalization} = \frac{\sum_{i=1}^{N} ErrorRate_{Relation\_i}}{N} \quad (2)$$

$$ErrorRate_{Entity\_i} = \frac{Error_i}{Attribute_i + PrimaryKey_i + ForeignKey_i} \quad (3)$$

$$ErrorRate_{Relationship\_i} = \frac{Error_i}{Relationship_i + Cardinality_i} \quad (4)$$

$$ErrorRate_{ERD} = \left(\frac{\sum_{i=1}^{N} ErrorRate_{Entity\_i}}{N} + \frac{\sum_{j=1}^{M} ErrorRate_{Relationship\_j}}{M}\right) / 2 \quad (5)$$

One independent variable is the design method. The textbook method (Section 2) refers to the approach used in most database and systems analysis and design (SA&D) textbooks. 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 difficulty of the in-class exercises, we added "level" as another independent variable in the research framework. The easy exercise has four entities and three one-to-many relationships and the difficult exercise has seven entities and six one-to-many relationships.

Subjects

In a southeastern public university in the United States, SA&D is one of the core courses of undergraduate IS and IT programs. SA&D is offered every semester with multiple sections. Undergraduate students can enroll in any section according to their schedule and/or preference. In Fall semester 2006, subjects enrolled in two sections of a junior level SA&D class participated in the experiment. Section A of the SA&D class with 22 subjects applied the textbook method and Section B with 23 applied the cookbook method. The 15-week class met twice weekly. 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 bottom-up database design (four 75-minute sessions). The instructor spent one week explaining the bottom-up database design method with examples in both classes. In the first half of the following week, subjects worked on two practice exercises using the one of the methods 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 method after class. During the second half of the week, subjects applied the method to solve two in-class exercises.

In-Class Exercises

The two in-class exercises were to design two databases with different difficult levels: easy and difficult. The easy task had four entities with three relationships. The difficult task had seven entities with six relationships. The subjects' tasks were to normalize two universal relations to 3NF and draw ERDs.

Experiment Procedure

Prior to the in-class exercises, the subjects completed a research participation consent form and an anonymity agreement. Next, subjects read the exercise scenario and applied the method they learned to work on the exercises. A summary of the method was provided to the subjects for quick reference.
The error rates of normalization and ERD were assessed using grading equations (1) – (5) in the research method section. The grading equations are fairly algorithmic in nature since the errors are very predictable. All relation errors can be classified into attribute errors (missing and/or extra) and primary key errors (missing and/or extra). The ERD errors have two parts: entity and relationship. The entity errors are very similar to relation error. The relationship errors can be classified into connectivity errors (missing and/or extra) and cardinality errors.

Results

The experiment was a 2 × 2 factorial design. Subjects in the two SA&D sections applied one method to the two tasks with different difficulty levels. Forty-five subjects completed the in-class exercises. One-way multivariate analyses of variance (MANOVA) was used to compare the impact of the independent variables (method and level) on the error rates (normalization and ERD). The one-way MANOVA revealed that the pattern of means for the cookbook method observed in Table 1 is statistically significant than that for the textbook method. The multivariate *F* value (6.042) of method (p=0.004) observed in the MANOVA indicates that this is indeed the case. Although the

subjects generally had lower error rates in the easy task than the difficult, the difference was not statistically significant (p=0.194, see Table 1). Thus null hypothesis H_1 was rejected, but hypothesis H_2 could not be rejected.

Table 1: MAVOVA test

Effect	F	Hypothesis df	Error df	Sig.
Method	6.042	2	85	**.004**
Level	1.672	2	85	.194

A two-way between-groups ANOVA was performed (see Table 2). The main effect of method 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.

Table 2: Test of between-subjects effects with dependent variable

Source	Type III Sum of Squares	df	F	Sig.
Method				
Normalization	.170	1	6.433	**.013**
ERD	.729	1	11.938	**.001**

Conclusion

The cookbook method offers simple and straightforward steps for teaching bottom-up database design. We found that the cookbook method is a valuable addition and supplement to the database education. One would not use the cookbook method as the only coverage of database design. In our view, students also need to understand the rationale behind the concepts of normalization and ERD and to develop the skills needed to determine the functional dependencies in the first place. Our results suggest that the cookbook method may provide a better way to teach students how to normalize a database table(s) and convert the normalized tables to ERD than the traditional (textbook) method used to attempt to instill this skill in students.

References

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

Batra, D., Hoffer, J. A., and Bostrom, R. P. (1990). Comparing representations with relational and EER models, *Communications of the ACM*, 33 (2), 126-139.

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 modeling tasks, *International Journal of Human-Computer Studies*, 46, 397-419.

Bernstein, P.A. (1976). Synthesizing third normal form relations from functional dependencies. *ACM Transactions Database Systems*, 1(4), 277-298.

Chen, P. P. (1976) The entity-relationship model—Toward a unified view of data. *ACM Transactions on Database Systems*, 1 (1), 9-36.

Codd, E. F. (1970). A relational model of data for large relational databases. *Communications of the ACM*, 13 (June), 377-387.

Hoffer, J. A., Prescott, M. B., and McFadden, F. R. (2005). *Modern database management*, 7th Ed., Prentice-Hall, Upper Saddle River, NJ.

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

Kung, H. and 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), 315-324.