3-1-2004

Online Data Modeling Tool to Improve Students' Learning of Conceptual Data Modeling

Hsiang-Jui Kung
hjkung@georgiasouthern.edu

Hui-Lien Tung

ONLINE DATA MODELING TOOL TO IMPROVE STUDNETS' LEARNING OF CONCEPTUAL DATA MODELING

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

Hui-Lien Tung
Troy State University
htung@troyst.edu

Abstract

Computer based instruction (CBI) has been used in science, mathematics, and engineering education, and showed positive outcomes. CBI is generally most effective when used as a supplement to traditional teaching. In this research-in-progress paper, the author presents an online data modeling tool as the supplement to data modeling teaching and proposes a research plan to measure students' perceived effectiveness of the online tool.

Introduction

Data-oriented design is one of the major software development approaches that Computer Science (CS), Information Systems (IS), and Information Technology (IT) students are required to master during the course of their studies. Some developers believe that the data model is the most important part of the statement of information system requirements. Today, virtually every business information system includes at least one database (Turban et al., 2002; Hoffer et al., 2002). Conceptual data modeling often refers to entity-relationship (E-R) diagram. E-R diagrams can be translated into a wide variety of technical architectures for data, such as relational, network, and hierarchical. Since relational database technology is the mainstream database today, this study focuses on relational data modeling (Mallach, 1996).

The primary deliverable from the conceptual data modeling is an E-R diagram that contains only entities and the business relationships between them. The next step is to complete the E-R diagram with attributes. For relational data model, this step is normalization that is considered to be essential in database design in order to (1) store information without unnecessary redundancy, and (2) ensure efficient database performance (Silberschatz et al., 2002).

Database normalization is the process of transforming data into well-formed or natural groupings such that one fact in one group is connected to other facts in other groups through relationships. It also involves simplifying the relations among groups and reducing anomalies that may otherwise occur during manipulation of the relations in a relational database (Codd, 1970). Normalization is essentially a mechanical process. This study proposes an online data modeling tool to automate the normalization process and draw normalized E-R diagram. The tool also provides step-by-step normalization process to help students' learning.

Online Data Modeling Tool

The first step in the online data modeling tool design is to define the learning objectives; these objectives should then drive the design of the final application. In this case, the goals for the tool are for students to learn the database normalization process by using functional dependencies, and then to draw E-R diagram based on the normalized relations. Fundamentally, these goals are that students learn, practice and retain data modeling skill. Kristof & Satran (1995) indicate that realization of these goals is best accomplished through simple, clear presentation of content that provides remediation when deemed necessary by the user.

The core of the online data modeling tool is the simple normalization technique that is easy to follow (Bernstein, 1976). The following steps describe the simple normalization process applied by the example relation $T(\underline{A}, B, C, D, \underline{E}, F)$ contains three functional dependencies:

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

$$FD_2 : B \rightarrow C, D$$

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

1. All functional dependencies on the left-hand side must keep their attributes intact.
   Note: Though attribute A appears in functional dependencies FD_1 and FD_3, attribute A cannot be eliminated. All attributes should be kept on the left-hand side of the functional dependencies as is.

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 are
   1) keep the attributes that have fewer numbers of attributes on the left-hand side (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 and FD_1 has only one attribute on the left-hand side. A new functional dependency $FD_3' : A, E \rightarrow F$ is formed.
   2) when two FDs have the same number of attributes on the left-hand side, keep the attributes that have fewer numbers of attributes on the right-hand side (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.
3. Construct relations. This step is to convert the functional dependencies without extraneous right-hand side attributes to relations. The new functional dependencies are as followings:

$$FD_1' : A \rightarrow B$$

$$FD_2 : B \rightarrow C, D$$

$$FD_3' : A, E \rightarrow F$$

The final normalized relations are exactly the same as the results of the traditional SA&D normalization techniques.

- $T_1^{''}$ ($\underline{A}$, B)

- $T_2^{''}$ ($\underline{B}$, C, D)

- $T_3^{''}$ ($\underline{A}$, $\underline{E}$, F)

Appendix A proves that any set of tables generated through the simple normalization technique is indeed fully normalized to 3NF. However, as Appendix A also shows, the result holds only if the set of functional dependencies used are non-trivial and closed (see Appendix A for definitions). The sets of functional dependencies used as examples in the remainder of this paper do indeed meet these requirements, as did all the exercises and exam questions given to students.This is the body of the paper. This is the body of the paper.

Research Plan

Research Design

Three sections of a junior level systems analysis and design course will participate in this study. Each section has approximately 25 students and all three sections use a common syllabus. The intent is to have all sections cover the same material at about the same pace. The 15-week course meets twice weekly. The Hoffer et al. (2002) textbook will be used to cover feasibility study, data modeling, process modeling, and physical design. Students will spend five 75-minute sessions on conceptual database design. In the first session, the instructor will explain conceptual data modeling (E-R diagram with entities only). In the second session, the instructor explains the importance of database normalization and the definitions of normal forms. In the third session, the instructor explains the simple normalization technique using the meaningful examples. The integration of normalization and E-R diagram drawing will be covered in the fourth session. In the last session, students will meet in computer lab and try the online data modeling tool.

The online data modeling tool is a Java applet that students need to key in functional dependencies (FDs) in order to normalize their database schema and create data model (E-R diagram). The steps are as follows:
1. Key in FD one at a time, then click "submit" button.
2. After keying all the FDs, students can click "normalize" button to normalize the database schema.
3. Click "Step by Step" button, students will see a window pop up to show the normalization steps.
4. Click "Draw E-R D" button, students will see a window pop up to show the normalized Entity-Relationship diagram.

Sample results of the online tool are in figures 1-7. Students can enter either hypothetical or meaningful attribute names when using the tool.

The study will use independent comparison design to measure the difference in effectiveness before and after introducing the online data modeling tool. This design will survey students' perception about conceptual data modeling before and after they use the online data modeling tool to solve data modeling problems.

The following exercise will be given to the students in the third and fourth session. Underlined attributes in the exercise are the candidate keys of the relations.

Sample Exercise:

Sales (<u>Invoice#</u>, Date, <u>Line#</u>, Quantity, S_Price, Coupon_Deduction, Savings, Sub_Total, Product#, P_Name, P_Price, Vendor#, V_Name, V_City)

FDs:

FDD_1: Invoice#, Line# $\rightarrow$ Date, Quantity, S_Price, Coupon_Deduction, Savings, Sub_Total, Product#, P_Name, P_Price, Vendor#, V_Name, V_City

FDD_2: Product# $\rightarrow$ P_Name, P_Price, Vendor#, V_Name, V_City

FDD_3: Vendor# $\rightarrow$ V_Name, V_City

FDD_4: Invoice# $\rightarrow$ Date

Procedure

The questionnaire, which will be administered at the end of the fourth and fifth session, consisted of 15 questions, with four asking for demographic information (status, gender, major, number of courses taken previously in discrete mathematics, data structure, algorithms, and databases), one open-ended question and ten questions that asked students about:

1. The difficulty to identify entities;

2. The difficulty to identify relationships between entities;

3. The difficulty to identify cardinalities for every relationship;

4. The difficulty to group attributes into entities;

5. The difficulty to draw Entity Relationship diagram;

6. The difficulty to normalize a database;

7. The difficulty to integrate normalization with E-R diagram drawing;

8. The perceived grade/score on ER diagram drawing;

9. The perceived grade/score on normalization technique;

10. The perceived grade/score on Conceptual Database design.

The questions will be answered on a five point Likert scales. An open-ended question enables the students to write down their comments about conceptual data modeling before and after using the online tool.
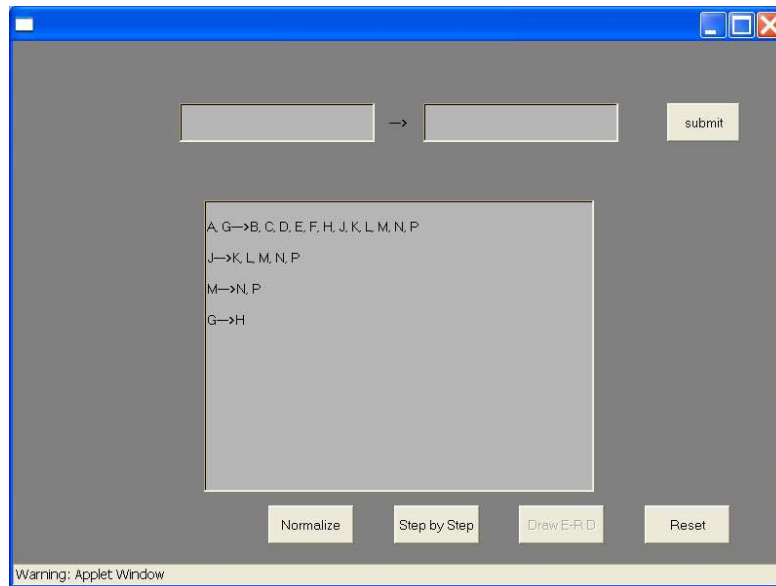
Figure 1. Main Window of the Online Data Modeling Tool
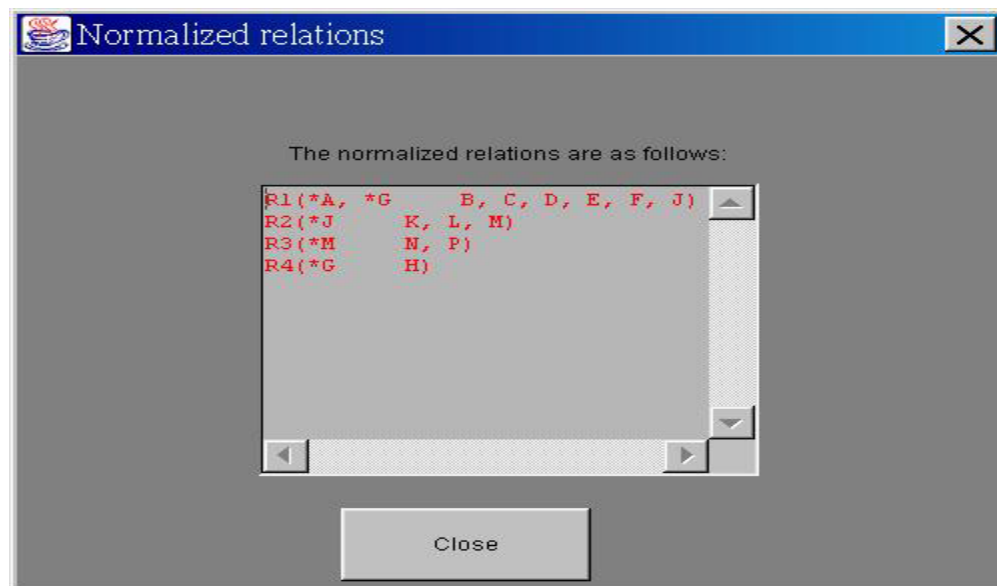

Figure 2. Results after Normalization
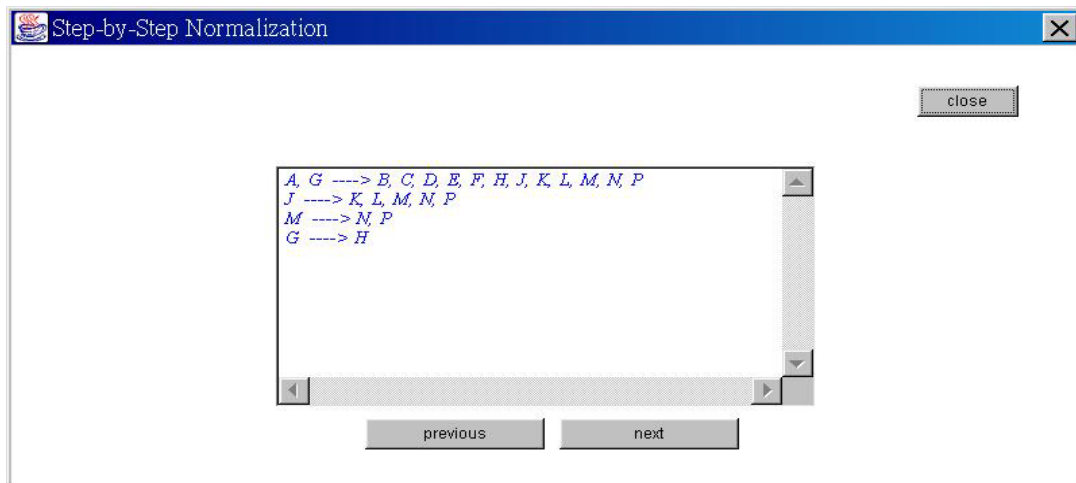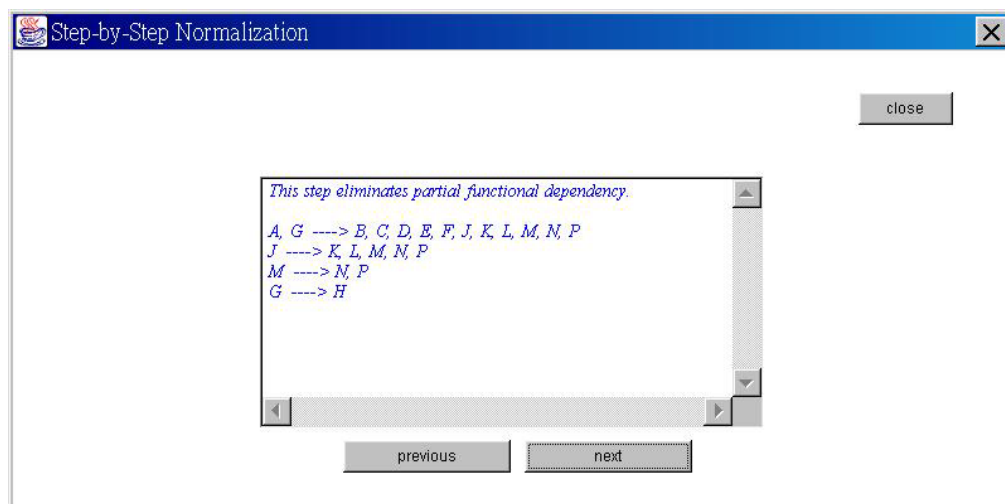
Figure 3. Step-by-Step Normalization Process (1)

Figure 4. Step-by-Step Normalization Process (2)
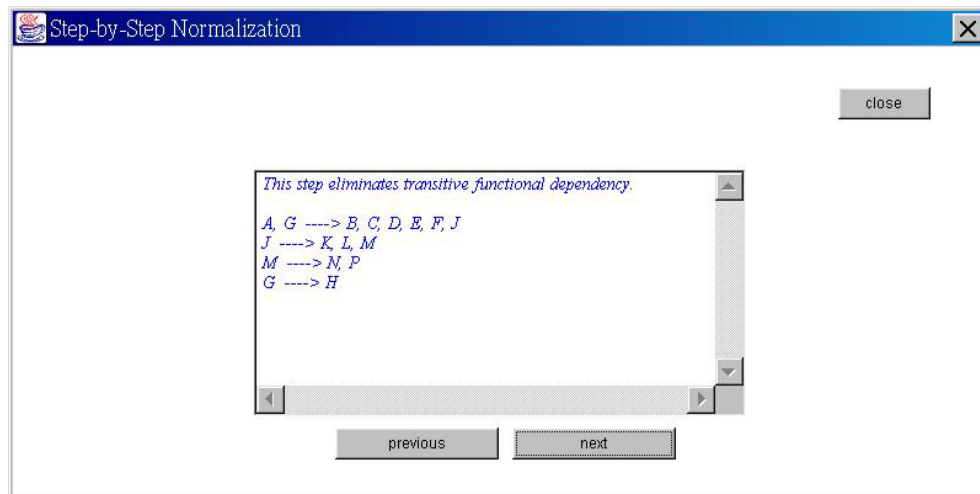
Figure 5. Step-by-Step Normalization Process (3)

This step eliminates transitive functional dependency.

A, G ----> B, C, D, E, F, J
J ----> K, L, M
M ----> N, P
G ----> H

Figure 6. Step-by-Step Normalization Process (4)

*R1(*A, *G, B, C, D, E, F, J)*
*R2(*J, K, L, M)*
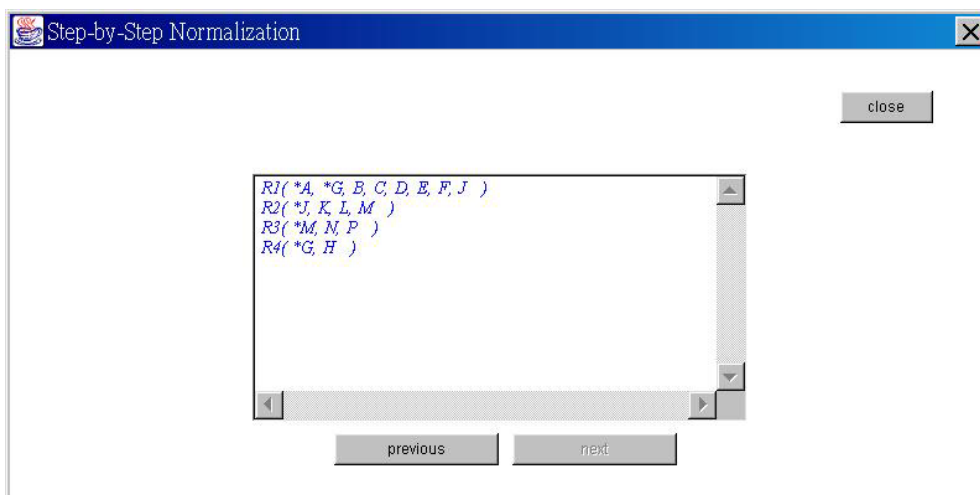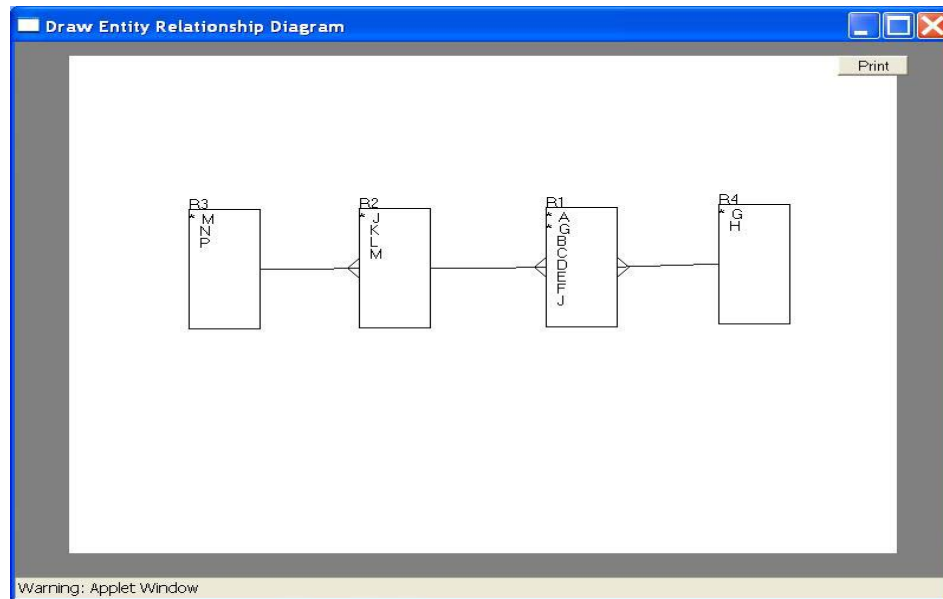*R3(*M, N, P)*
*R4(*G, H)*

Figure 7. E-R Diagram

Conclusion

This paper describes a research plan to investigate the effectiveness of an online data modeling tool in helping students' learning. Such research holds potential for improving the teaching of conceptual database design. The design of the online tool is not to replace traditional teaching but to provide a supplement to enhance students' learning.

References

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

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

Hoffer, J. A., George, J. F., and Valacich, J. S., (2002). *Modern Systems Analysis & Design*, 3rd Ed., Prentice Hall.

Kristof, R. & Satran, A. (1995). *Interactivity by Design: Creating and ommunicating with New Media*. Mountain View, CA: Adobe Press.

Mallach, E. G., (1996). Education and Training Aren't the Same, *Computerworld*, Oct.

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

Turban, E., McLean, E., and Wetherbe, J. (2001). *Information Technology for Management: Making Connections for Strategic Advantage*, 3rd Ed., New York, NY: Wiley.

Appendix A: Proof of the Simple Normalization Technique

Notation: In what follows, capital letters 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 method is fully normalized up to 3NF.

Proof: The simple method reduces FD to a set FD', from which the tables are generated. Assume that the set of tables generated from FD' through the simple method 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 method are fully normalized up to 3NF.

Appendix B: The Data Modeling Effectiveness Survey

_____

The purpose of this survey is to measure how you feel about the effectiveness of Data Modeling before and after using the online data modeling tool.

On the following pages you will find different factors, each related to some aspect of Data Modeling. You are to rate each factor on the descriptive scales that follow it, based on your evaluation of the factor.

The scale positions are defined as follows:

adjective Y   :_____:_____:_____:_____:_____: adjective Z

(1)   (2)   (3)   (4)   (5)

(1) Y   (4) slightly Z

(2) slightly Y   (5) Z

(3) equally Y or Z

The following example illustrates the scale positions and their meanings:

My vacation in the Bahamas was:

restful/healthy   :_____:_____:_____:_____:__X__:hectic/unhealthy

According to the responses, the person's vacation was hectic and unhealthy.

INSTRUCTIONS

1. Check each scale in the position that describes your evaluation of the factor being judged.

2. Check every scale; do not omit any.

3. Check only one position for each scale.

4. Check in the space, not between spaces.

THIS,   NOT THIS

:__X__: :_____X_____:

5. There are no right or wrong answers. Work rapidly. Rely on your first impressions.

Thank you very much for your cooperation.

Would you like a copy of the study's results?

Yes [   ]   No [   ]

_____

I. Demographic Information

1. Please indicate your academic status.

   Freshman [ ]   Sophomore [ ]   Junior [ ]   Senior [ ]   Graduate [ ]

2. Please indicate your gender.

   Male [ ]   Female [ ]

3. Please indicate your major.

   IS [ ]   IT [ ]   CS [ ]   Double Major [ ]   Others [ ]

4. How many Computer Technology related courses have you taken?

   Discrete Mathematics/Mathematical Structures [ ] Data Structure [ ] Algorithms [ ] Database [ ]

II. Effectiveness of Database Normalization Technique

1. How difficult was it to identify entities in Database design?

difficult   :_____:_____:_____:_____:_____: easy

2. How difficult was it to identify relationships between entities in Database design?

difficult   :_____:_____:_____:_____:_____: easy

3. How difficult was it to identify cardinalities for every relationship in Database design?

difficult   :_____:_____:_____:_____:_____: easy

4. How difficult was it to group attributes into entities?

difficult   :_____:_____:_____:_____:_____: easy

5. How difficult was it to draw Entity Relationship diagram?

difficult   :_____:_____:_____:_____:_____: easy

6. How difficult was it to normalize a database?

difficult   :_____:_____:_____:_____:_____: easy

7. How difficult was it to integrate normalization with E-R diagram drawing?

difficult   :_____:_____:_____:_____:_____: easy

8. How well do you know ER diagram drawing?

( ) Excellent ( ) Very Good ( ) Satisfactory ( ) Unsatisfactory ( ) Fail

9. How well do know normalization technique?

( ) Excellent ( ) Very Good ( ) Satisfactory ( ) Unsatisfactory ( ) Fail

10. How well do you know Conceptual Database design?

( ) Excellent ( ) Very Good ( ) Satisfactory ( ) Unsatisfactory ( ) Fail

11. Your comments about Database design. Which parts of database design are difficult to you?

_____

_____