

REINFORCING DATABASE CONCEPTS BY USING ENTITY RELATIONSHIPS DIAGRAMS (ERD) AND NORMALIZATION TOGETHER FOR DESIGNING ROBUST DATABASES

*Kamal Hingorani, Alabama State University, khingorani@alasu.edu
Dexter Gittens, Alabama State University, dgittens@alasu.edu
Nicholas Edwards, Alabama State University, nedwards1303@myasu.alasu.edu*

ABSTRACT

Most books on Accounting Information Systems include chapters on Entity relationship diagrams (ERD) and on normalization of databases. These concepts have to be taught within a framework on topics that include Enterprise Systems, E-business, and Internal Controls. This paper discusses how the concepts of ERD and normalization were introduced and used together to design a database in a graduate-level accounting information class at a southeastern university. The course made use of Oracle's Data Modeler for ER diagrams and Microsoft Access for normalization to streamline and reduce the drudgery involved in the process. The learning was significantly better with this approach than the approach in which these concepts were taught in isolation.

Keywords: Information Technology (IT), Database Management, Entity-relationship model, Normalization, Logical Model, Relational Model, ERD-relational schema mapping.

INTRODUCTION

Data modeling is an important component of the curriculum of both Computer Science and Information Systems programs (Ahrens & Song, 1991; Batra & Anthony, 1994; Yang & Li, 2016). Entity relationship diagram (ERD) is the most widely used technique for data modeling. An ERD starts with the designer capturing the entities, their attributes, and the binary relationships between entities. The logical ERD is then converted to a relational model using a set of scientific rules. The relational model is then converted into a physical database based on the software platform for implementation (Chen, 1976).

A number of tools have been developed to automate the ERD process beyond the depiction of the logical model. One such tool, the Oracle SQL Developer Data Modeler is a data modeling and database design tool that provides an environment for capturing, modeling, managing, and exploiting metadata. It automates the entire process beyond the logical model. The conversion of the logical model into a relational model using this tool is done through a click of a button. The tool has the capabilities to write the SQL statements for the various types of enterprise class databases. Using Oracle SQL Developer, Data Modeler users can create, browse and edit, logical, relational, physical, multi-dimensional, and data type models. The Data Modeler provides forward and reverse engineering capabilities and supports collaborative development through integrated source code control. The Data Modeler can be used in both traditional and in Cloud environments (Kreie & Ernst, 2013; Oracle, 2011; "SQL Developer Data Modeler4.2", n.d.).

The Version 4.2 of the Oracle SQL Developer Data Modeler is available for free download on the web. The installation is very simple; it requires unzipping the downloaded file and copying it to a folder on the hard-drive.

Another important concept in database, normalization is a systematic approach of analyzing database tables to eliminate data redundancy and undesirable characteristics that would create Insertion, Update and Deletion anomalies. It is usually a multi-step process that creates additional tables by removing duplicated data from tables (Kolp & Zimanyi, 2000).

A database designed using the ERD process generally results in tables that are in their 3NF. Databases designed on the fly or converted from spreadsheets have to be checked for normalization. Although there are 6 normal forms, most practitioners struggle to decide between normalization and denormalization. Normalized databases fare very well under conditions where the applications are write-intensive and the write-load is more than the read-load. Denormalized databases fare well under heavy read-load and when the application is read intensive ("Databases:

Normalization or Denormalization. Which is the better technique?”, 2011). A database in its 3NF is considered to strike a good balance between normalization and denormalization.

Teaching the concepts of ERD and normalization is a challenge in standalone courses on Database Management. This challenge is even greater when these concepts have to be taught in a course on Accounting Information Systems (AIS). This paper describes how concepts in database were taught by using the technique of ERD and the rules of normalization together to analyze the same set of users’ requirements in a graduate-level AIS course at southeastern university in the United States. The learning was significantly better with this approach than the approach in which these concepts were taught in isolation.

CREATING DATABASE WITH NORMALIZED TABLES USING THE ER MODEL AND THE PROCESS OF NORMALIZATION.

Most books on Accounting Information Systems include chapters on Entity relationship diagrams (ERD) and on normalization of databases. These concepts have to be taught within a framework on topics that include Enterprise Systems, E-business, and Internal Controls (Gelinias et al., 2015). Covering these concepts within a 2-week period is very challenging and past experiences have shown that the students do not completely grasp the concepts. The authors decided to use a database consisting of one table converted from a spreadsheet (see Table 1) as a users’ requirements for designing a normalized database using both the ER diagrams and the rules of normalization. Although the students (working in teams) were successful in designing the correct database using both the methods, the task was more challenging for the approach that involved normalization of database tables.

Table 1. Users' Requirement through an Un-Normalized Table

SO_Number	Item_Number	Item_Name	Qty_Ordered	CUST_CODE	CUST_Name
1010	2010-0050	Formed Handlebar	2	WHEEL	Wheelaway Cycle Center
	1000-1	20 in. Bicycle	5	WHEEL	Wheelaway Cycle Center
1011	1002-1	24 in. Bicycle	5	ETC	Bikes Et Cetera
	1001-1	26 in. Bicycle	10	ETC	Bikes Et Cetera
1012	1003-1	20 in. Bicycle	5	WHEEL	Wheelaway Cycle Center
	1001-1	26 in. Bicycle	10	WHEEL	Wheelaway Cycle Center
1013	1001-1	26 in. Bicycle	50	IBS	Inter. Bicycle Sales
1014	1003-1	20 in. Bicycle	25	ETC	Bikes Et Cetera
1015	1003-1	20 in. Bicycle	25	WHEEL	Wheelaway Cycle Center
1016	1000-1	20 in. Bicycle	4	ETC	Bikes Et Cetera
	1003-1	20 in. Bicycle	5	ETC	Bikes Et Cetera
	3961-1041	Tire Tube, 26 in.	5	ETC	Bikes Et Cetera
	3965-1050	Spoke Reflector	50	ETC	Bikes Et Cetera

DESIGNING THE DATABASE USING THE NORMALIZATION PROCESS.

Normalization is a systematic approach of analyzing database tables to eliminate data redundancy and undesirable characteristics that would create Insertion, Update and Deletion anomalies. It is usually a multi-step process that creates additional tables by removing duplicated data from tables. The design approach for the set of users’ requirements depicted in Table 1 proceeded as follows.

1. The identification of the primary key. This was a relatively easy task and all students correctly identified that the SO_Number and the Item_Number together constituted the primary key.
2. The conversion of the table into its 1NF did not pose a great challenge to the students. A table is said to be in First Normal Form if (1) there is no row or column order (2) each row (record) is unique (3) each row by column value (field) contains exactly one value (4) there are no repeating columns. The conversion of table to the 1NF is shown below:

Table 2. Database Table in 1NF (1st Normal Form)

Table 1.1 (1NF)					
SO_Number	Item_Number	Item_Name	Qty_Ordered	CUST_CODE	CUST_Name
1010	2010-0050	Formed Handlebar	2	WHEEL	Wheelaway Cycle Center
1010	1000-1	20 in. Bicycle	5	WHEEL	Wheelaway Cycle Center
1016	1000-1	20 in. Bicycle	4	ETC	Bikes Et Cetera
1011	1002-1	24 in. Bicycle	5	ETC	Bikes Et Cetera
1012	1003-1	20 in. Bicycle	5	WHEEL	Wheelaway Cycle Center
1014	1003-1	20 in. Bicycle	25	ETC	Bikes Et Cetera
1015	1003-1	20 in. Bicycle	25	WHEEL	Wheelaway Cycle Center
1016	1003-1	20 in. Bicycle	5	ETC	Bikes Et Cetera
1016	3961-1041	Tire Tube, 26 in.	5	ETC	Bikes Et Cetera
1016	3965-1050	Spoke Reflector	50	ETC	Bikes Et Cetera
1011	1001-1	26 in. Bicycle	10	ETC	Bikes Et Cetera
1012	1001-1	26 in. Bicycle	10	WHEEL	Wheelaway Cycle Center
1013	1001-1	26 in. Bicycle	50	IBS	Inter. Bicycle Sales

3. The conversion of the table/database into the 2NF was much more difficult for the students. The rule for conversion is however very simple: remove any non-key attributes that only depend on part of the table key to a new table. Most students had issues with splitting the table in its 1NF into the required three tables for them to be in their 2NF. For example, Item_Name is functionally dependent on only Item_Number.

Table 3. Database with tables in 2NF (2nd Normal Form)

Table 2.1 (2NF)			Table 2.2 (2NF)		
SO_Number	Item_Number	Qty_Ordered	Item_Number	Item_Name	
1010	2010-0050	2	1000-1	20 in. Bicycle	
1010	1000-1	5	1001-1	26 in. Bicycle	
1016	1000-1	4	1002-1	24 in. Bicycle	
1011	1002-1	5	1003-1	20 in. Bicycle	
1012	1003-1	5	2010-0050	Formed Handlebar	
1014	1003-1	25	3961-1041	Tire Tube, 26 in.	
1015	1003-1	25	3965-1050	Spoke Reflector	
1016	1003-1	5			
1016	3961-1041	5	Table 2.3 (2NF)		
1016	3965-1050	50	SO_Number	CUST_CODE	CUST_Name
1011	1001-1	10	1010	WHEEL	Wheelaway Cycle Center
1012	1001-1	10	1011	ETC	Bikes Et Cetera
1013	1001-1	50	1012	WHEEL	Wheelaway Cycle Center
			1013	IBS	Inter. Bicycle Sales
			1014	ETC	Bikes Et Cetera
			1015	WHEEL	Wheelaway Cycle Center
			1016	ETC	Bikes Et Cetera

4. The conversion of the database with all tables in their 3NF is to make sure that the tables in their 2NF have no transitive dependencies. The third table has this problem and needs to be converted into 2 tables. The resultant database with all tables in their 3NF is shown below.

Table 4. Database with Tables in 3NF (3rd Normal Form)

Table 3.1 (3NF) ItemsOrdered			Table 3.2 (3NF) Items	
SO_Number	Item_Number	Qty_Ordered	Item_Number	Item_Name
1010	2010-0050	2	1000-1	20 in. Bicycle
1010	1000-1	5	1001-1	26 in. Bicycle
1016	1000-1	4	1002-1	24 in. Bicycle
1011	1002-1	5	1003-1	20 in. Bicycle
1012	1003-1	5	2010-0050	Formed Handlebar
1014	1003-1	25	3961-1041	Tire Tube, 26 in.
1015	1003-1	25	3965-1050	Spoke Reflector
1016	1003-1	5		
1016	3961-1041	5	Table 3.3 (3NF) SalesOrder	
1016	3965-1050	50	SO_Number	CUST_CODE
1011	1001-1	10	1010	WHEEL
1012	1001-1	10	1011	ETC
1013	1001-1	50	1012	WHEEL
			1013	IBS
Table 3.4 (3NF) Customer			1014	ETC
CUST_CODE	CUST_Name		1015	WHEEL
ETC	Bikes Et Cetera			
IBS	Inter. Bicycle Sales			
WHEEL	Wheelaway Cycle Center			

DESIGNING THE DATABASE USING THE ERD PROCESS.

The un-normalized table in Table 1 was used as the basis for identifying the entities, their attributes, and the relationships. The students are easily able to identify the entities with attributes (key attribute have been underlined) as shown below:

- Customer (Cust_code, Cust_Name)
- Items (Item_Number, Item_Name)
- Sales_Order (SO_Number)

The ER diagram was drawn using SQL Data Modeler. The three entities and the binary relationships between them are shown below:

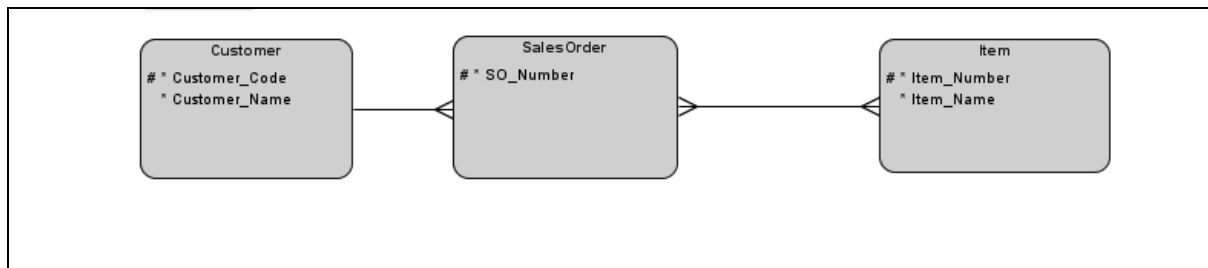


Figure 1. The ER diagram in Oracle Data Modeler

The relationship between customer and SalesOrder is 1:N while the relationship between SalesOrder and Item is an N:M relationship. The N:M relationship has an attribute called Qty_Ordered. Although the data modeler requires naming of every relationship, the names do not appear on the diagram. Also, the attribute of the relationship, if any, is not depicted in the diagram.

The conversion of the logical model into the relational model is a one-click process in Oracle Data Modeler. The relational model for the logical model displayed in Figure 1 is shown in Figure 2 below. The relational model has four tables, exactly like the one obtained through the process of normalization

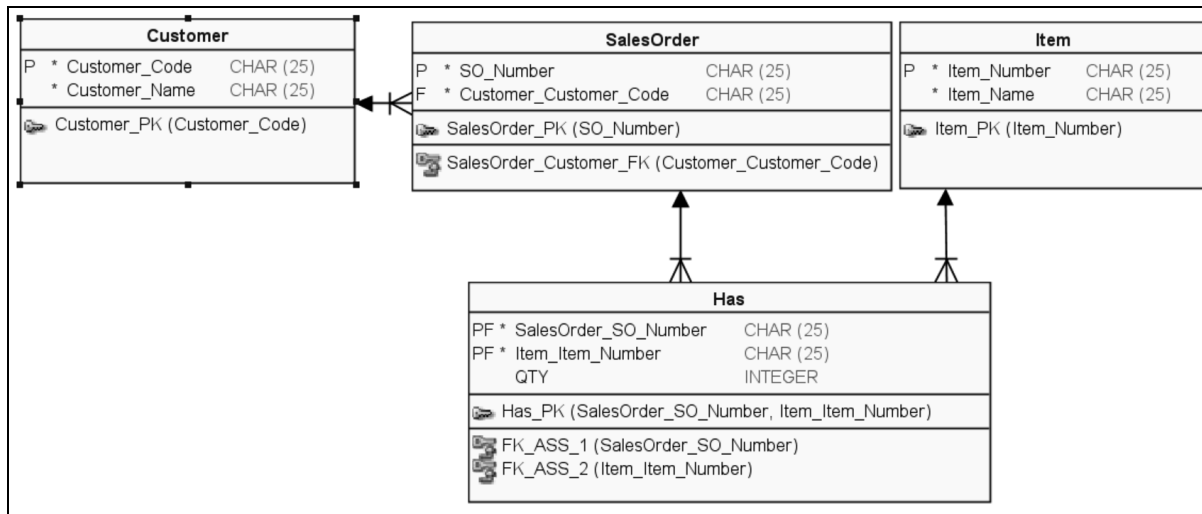


Figure 2. The Relational Model in Oracle Data Modeler

The tables in the relational model have their names, field names, their data types (as decided for the logical model), the primary keys, and the foreign keys. The intersection table gets its name from the name of the relationship and the key of the table is neatly depicted in the diagram.

ENHANCED LEARNING OF THE DATABASE CONCEPTS

The learning in Information Systems can be enhanced by using freely available tools such as Oracle Data Modeler. This particular tool automate routine tasks and neatly document the entire process of ERD from the logical to the physical model.

In most database courses, the concepts of ER diagrams and Normalization are taught separately without the students learning the fact that the ERD process, if properly implemented, usually results in database tables that are in their 3NF. The availability of the Data Modeler enabled the authors to combine both the topics in a single topic with the students working on an un-normalized table as the users' requirements for designing a database.

The ERD method was easier for the students to comprehend and grasp. The students were able to identify the entities, their attributes, and the relationships. The students did not have to apply the rules for converting the logical model to a relational model - that was done by the Data Modeler. The relational model gave them the solution in a neatly drawn diagram with all the tables, the fields, primary keys, foreign keys, and the relationship. They could compare the results that they had got through the process of normalization.

CONCLUSIONS AND AREAS FOR FUTURE RESEARCH

A course on Database Management is a core requirement in most MIS program across the nation. The semester-long courses have enough time to teach the concepts of ER Modeling and Normalization. Database

Management Systems also form an important component in courses on Accounting Information System (AIS). Some popular textbooks on AIS have separate chapters on Database Management Systems and on Relational Database. These chapters emphasize concepts such as ERD, normalization, and SQL among others. Teaching these concepts in an AIS course has always been a challenge because of the time constraints. The approach of analyzing a set of users' requirements through the approaches of ER diagram and normalization with the students seeing the same set of results helped in the learning process. This was possible because of the availability of Oracle SQL Data Modeler and Access that help automate many of the routine and rudimentary steps, including mapping of the logical model to the relational schema.

The authors plan to use this approach in the MIS course on Database Management and extend the scope to the enhanced entity-relationship (EER) model. A pre-test and post-test instrument will also be used to measure learning.

Database modeling and design is an important skill set for new hires. Universities are always looking for tools that can be used for imparting the requisite skills. The SQL Developer Data Modeler, a robust data modeling tool developed by Oracle, is a tool that can make teaching and learning of database design easier and students can understand difficult concepts such as ER diagrams and normalization and finish the course with far greater level of expertise.

REFERENCES

- Ahrens, J., & Song, I. (1991), EER data modeling aids for novice database designers. Proceedings of the 2nd Int. Conference Information Resources Association, 99-114, 1991.
- Batra, D., & Antony, S. R. (1994). Novice errors in conceptual database design. *European Journal of Information Systems*, 3(1), 57-69.
- Bock, D. B. and Yaeger, S. E. (2002). Improving Entity Relationship Modeling Accuracy with Novice Data Modelers. *Journal of Computer Information Systems* 42(2), 69-75.
- Chen, P. P. (1976). The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems*, 1(1), 9-36.
- Computer Science Curricula 2013 report, <http://ai.stanford.edu/users/sahami/CS2013/final-draft/CS2013-final-report.pdf>, retrieved March 28, 2016.
- Connolly, T., & Begg, M (2006). A constructivist-based approach to teaching database analysis and design, *Journal of Information Systems Education*, 17(1), 43-53.
- Databases: Normalization or Denormalization. Which is the better technique? (2011, July 04). Retrieved June 28, 2017, from <http://www.ovaistariq.net/199/databases-normalization-or-denormalization-which-is-the-better-technique/#.WVPs8uvyvc>
- Gelinas, U. J., Dull, R. B., & Wheeler, P. R. (2015). Accounting information systems. Stamford, CT: Cengage Learning.
- Hitchman, Steve (2002) "The Details of Conceptual Modeling Notations are Important - A Comparison of Relationship Normative Language," *Communications of the Association for Information Systems*, 9(10).
- Hoffer, J. (1982). An empirical investigation into individual differences in database model. Proceedings of 3rd Int. Conf. Information Systems, 153-167.

- IS-2010, IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, <http://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf>, retrieved March 28, 2016.
- Kolp, M., & Zimanyi, E. (2000). Enhanced ER to relational mapping and interrelational normalization. *Information and Software Technology*, 42, 1057–1073.
- Kreie, J., & Ernst, B. (2013). From database concepts to application: Use problem-based learning and Oracle development tools to facilitate learning. Proceedings of the Information Systems Education Conference, 2013.
- Li Yang, Li Cao, "The Effect of MySQL Workbench in Teaching Entity-Relationship Diagram (ERD) to Relational Schema Mapping", *International Journal of Modern Education and Computer Science(IJMECS)*, 8(7), pp.1-12, 2016.DOI: 10.5815/ijmecs.2016.07.01
- Oracle (2011) An Introduction to Oracle Developer Data Modeler, available at <http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>.
- SQL Developer Data Modeler4.2. (n.d.). Retrieved July 01, 2017, from <http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>
- Pons, A. P, Polak, P. & Stutz, J. (2006). Evaluating the Teaching Effectiveness of Various Data Modeling Notations. *Journal of Computer Information Systems*, 46(2), 78-84.