

# A Teaching Note on Demonstrating the Ambiguity of ERDs with Respect to Referential Integrity

*Norman Pendegrift*  
*College of Business and Economics,*  
*University of Idaho, Moscow, ID, USA*

[norman@uidaho.edu](mailto:norman@uidaho.edu)

## Abstract

A limitation of entity relationship diagram (ERD) notation in representing referential integrity (RI) is discussed. Several examples are used to illustrate both the limitations of the ERD and the variety of RI issues in SQL. This paper suggests that confusion is created by conflating the ERD model with the relational model. A modest suggestion for clarification is proposed for inclusion in a database class.

**Keywords:** referential integrity, ERD, weak entity

## Introduction

The author recently encountered a computer aided software engineering (CASE) tool that creates a referential integrity constraint contrary to his expectations. This occurred while he was preparing for a database design course required of senior information systems (IS) majors at a western state university college of business and economics. Since identifying and implementing referential integrity constraints is an essential part of designing a database, it is an important topic in the course. In order to clarify the issue, several prominent text books were reviewed and several experiments were conducted. In general the texts were found to be correct, but they displayed various levels of detail in how RI should be implemented in SQL. The paper is intended to be of interest to those who teach similar or related courses.

The remainder of the paper is organized as follows: A brief background example is given to illustrate the issue. The next section briefly summarizes the coverage of referential integrity and weak entities in several texts and offers a short discussion. The paper then offers a proposal for use in the classroom and concludes with a summary of the lessons learned in this effort.

## ERDs and Referential Integrity

The problem came to the attention of the author after the recent adoption of a new CASE tool in

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

the course (Oracle SQL Developer Data Modeler (DM)). (Oracle 2011b) An ERD with 1:M relationship was created with both entities optional. The resultant definition language (DDL) code included a REFERENCES phrase in the child table. REFERENCES is part of a FOREIGN KEY constraint in SQL which is used to impose referential integrity constraints (Loney, 2011; Oracle 2011a). Since the parent entity was op-

tional and the references phrase makes a tuple in the parent table necessary, the relational implementation seems to be inconsistent with the intent of the ERD.

To clarify the issue, an ERD was created in which four 1:M relationships were created spanning the four possible combinations of minimum cardinality. The ERD is shown in Figure 1. Two relationships (Entity1:Entity2 and Entity1:Entity3) have the parent (Entity2 and Entity3) optional. In both cases the DDL generated included a REFERENCES clause with ON DELETE SET NULL. Since the REFERENCES phrase makes the parent tuple mandatory before a child tuple can be inserted, these amount to mandatory parents, at least for inserts. Both allow deletions of parents so the relationship is optional for delete, but not for insert. The other two relationships are both mandatory for insert and restrict deletions in different ways, one by CASCADE and one by RESTRICT. Thus, the simple idea of optionality in the ERD turns out in the DDL to be other than simple.

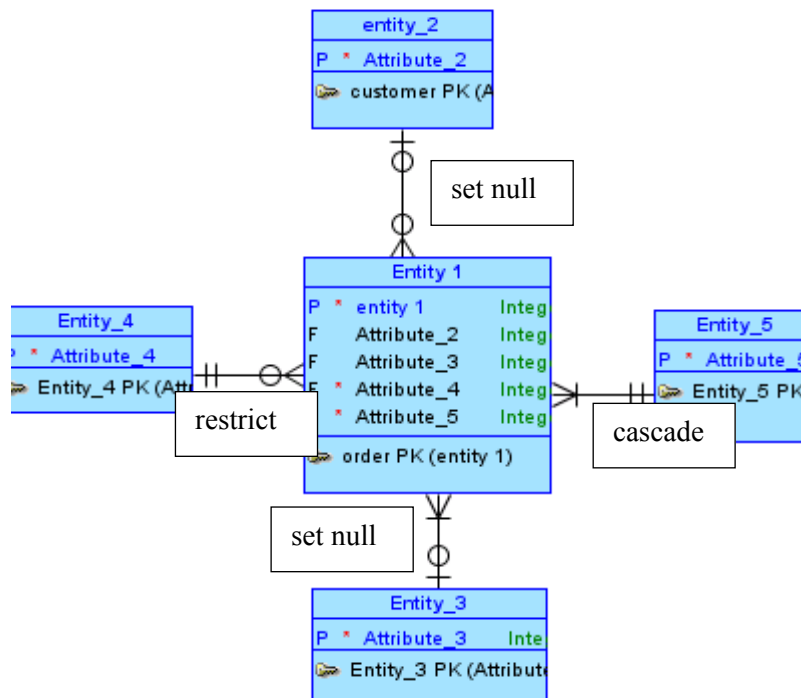


Figure 1. Four Relationships

## Weak Entities and Referential Integrity

To better understand what was going on, several current database texts (listed in Table 1) were consulted. The list included those on the author’s shelf which were recent and in at least a third edition. This was taken as evidence that they had found a sufficient audience to be considered in some sense authoritative. However, the list is in no way comprehensive. The results are summarized in Table 1 in the appendix.

Most addressed the limitations on the foreign key. There are two possible reactions on INPUT into the child table, and at least three in responses to UPDATE and DELETE (CASCADE, SET NULL, and RESTRICT). But, some include commentary which seems to imply that ON DE-

LETE CASCADE must be included. None suggested that an optional parent is included in a restriction. There was consistent, if not complete coverage of how to implement RI in SQL.

In general, an identifying relationship is one in which the key of the child includes the key of the parent. As can be seen from Table 1, there is not complete agreement on this issue. A weak entity is one for which an instance cannot exist without a matching instance in the parent entity. There is some disagreement on the relationship between weak entities and identifying relationships. In particular, some hold that weak entities must have an identifying relationship with the parent and other disagree. (See Table 1). The issue is complicated because there is not a standard way of representing weak entities on an ERD. Some CASE tools do not provide for the representation of weak entities. For example, DM allows the representation of identifying relationships, but no way of distinguishing weak entities. According to Kroenke and Auer (2012, C-7), the U.S. government standard for ERDs, IDEF1X, has no way of identifying a weak entity that is not identifying.

Date (2000) notes that weak entities clearly require RI on input and CASCADE on delete and update. On the other hand, mandatory strong entities clearly require RI on INPUT, but would seem to be ambiguous on delete and update. Since the weak notation implies CASCADE, it seems reasonable to assume otherwise with mandatory. Thus, the ERD cannot easily distinguish between the many possible combinations that can be implemented in the RDBMS. Further, in no sense does optional inclusion seem to require any sort of RI.

It seems to the author that Date (2000) offers the most cogent comment on the matter when he discusses the limitations of the ER model. In particular, he observes the inadequacy of the ER model in representing 1:M relationships (p.432). The ERD permits only two issues to be addressed. A mandatory inclusion in a relationship and a weak entity. The problem is compounded by the multitude of notations in the various ERD notations.

## A Modest Proposal

Referential integrity refers to a state in a relational database in which all Foreign Keys either match the value in some candidate key in the referenced table, or are null. This however, leaves several questions unanswered. Clearly this implies a restriction on input (input RI). However, it is not so clear what must happen when the “parent tuple” is deleted or when the candidate key in the “parent” table is changed. There are at least three possible reactions. Either the delete (update) can be forbidden (*restrict*), it could *cascade* to the child record, or the foreign key could be set to *NULL*.

It is clear, on reflection, that the idea of a weak entity captures the cascade idea, which would seem to leave the second for a mandatory strong entity. The problem is that these notations are not available in all CASE tools, nor is there clear agreement on these usages.

The issues noted above regarding the ERD is really about the inadequacy of the minimum cardinality notation. For in class discussion, the author has adopted the following notation. A third symbol may be added to the relationship indicating the impact of deletes. The author generally uses the crowfoot notation with both minimum and maximum cardinalities shown with “look here”. (Song et.al., 1995)

There are at least 10 possible states of referential integrity in SQL summarized in Table 2. Thus, it requires at least three bits of information to be able to distinguish between them. The ERD does not offer a sufficient notation to provide for this number of states.

The following simple additions are used to address the RI problems.

third symbol: Absent: RESTRICT, X on delete CASCADE, O on delete SET NULL

a fourth symbol for updates may be added if the response is different for updates.

In Figure 2, in the first example, the child is optional, with on delete cascade and on update set null. In the second, both deletes and updates are cascaded. To be sure this does not capture all of the possibilities allowed by the RDBMS, but it should capture most situations. This notation has the advantage of bringing the ERD graphics into closer accord with the capabilities of the programming language. In addition, it is a natural extension of existing notation.

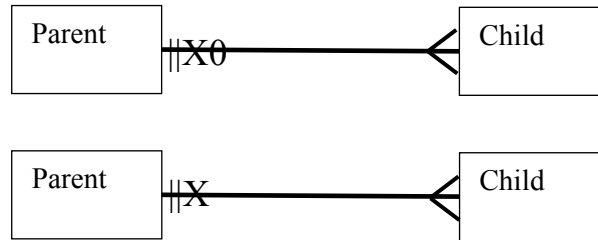


Figure 2. Example

While it would be nice to present a richer alternative notation, the simple fact is that until the various CASE tools recognize this issue, nothing useful is likely to come from it. Thus a major purpose of this paper is to call attention to the problem in hopes that the providers of CASE tools will provide a richer notation. The relational schema notation of DM uses a similar notation to capture the same ideas, and may provide a superior alternative for initial design. That remains to be seen in future offerings of the course.

While it is not likely that such notation will be adopted in industry among experienced data base programmers, it may prove useful in classrooms. Here students do not have the depth of experience to recognize that the ERD is not unambiguous. While a professional designer probably already has a kit bag of design “tricks”, the beginner does not. The intention of this proposal is to alert those students to some issues that might otherwise go unnoticed.

## Conclusion

This work suggests a number of ideas for those who teach a database course. One of the lessons is that one may not safely assume that a CASE tool will generate code that will perform as intended. Therefore, one should carefully build several examples and examine the behavior of the resultant code. This lesson is relevant to the students as well as to faculty.

A second lesson is that simple, important terms (like minimum cardinality, referential integrity, weak entity, identifying relationships, etc.) may be related, and the relationships between them should be clearly stated if students are to avoid the confusion which motivated this manuscript. Further, the way they are implemented in SQL can vary depending on the precise business need.

There are significant limitations to the ERD that need to be clearly explained. In the experience of the author, students tend to assume that the ERD unambiguously maps into a schema design. These simple examples demonstrate that this is not so and that the relationships, in particular, need to be carefully analyzed.

The final principle finding of this work is that these ideas need to be emphasized in the database class. The examples developed here have helped to serve that purpose, and provide the gist for

excellent homework assignments. The author hopes that others teaching this sort of class will find this paper useful in preparing those discussions.

Table 1: Summary of Authorities	
Connolly & Berg, 2010	<p>“If a foreign key exists in a relation, either the FK value must match a candidate key value of some tuple in its home relation [sic] or the FK value must be null (164). If the child is mandatory, then nulls are not allowed, if optional, then nulls are allowed.” (453)</p> <p>PK of a weak entity is fully or partially derived from the parent entity (442)</p>
Date, 2000	<p>“The database must contain no unmatched foreign keys.” (263)</p> <p>Weak entity must have CASCADE (432)</p>
Elmasi & Navathe, 2011	<p>RI enforced via the FK clause (insert integrity). FK must either be found in parent tuple or be null. (73)</p> <p>Weak entities have identifying relationships with the parent (221)</p>
Hoffer et.al.2011	<p>FK must match a PK or the FK must be null.(570)</p> <p>Weak entity requires an identifying relationship (70)</p>
Kroenke & Auer, 2012	<p>RI constraint limits the values of the FK.(112)</p> <p>Weak entity may have a non-identifying relationship with its parent (167)</p>
Loney, 2011	<p>provided by REFERENCES which provides input integrity. “On delete cascade ...maintain[s] referential integrity ...” when the target row is deleted. (300-301)</p>
Oracle, 2011a	<p>A referential integrity rule is a rule defined on a key (a column or set of columns) in one table that guarantees that the values in that key match the values in a key in a related table (the referenced value).</p> <p>Referential integrity also includes the rules that dictate what types of data manipulation are allowed on referenced values and how these actions affect dependent values.</p>
Microsoft, 2010.	<p>CASCADE and other options are editable in the Relationships window.</p>

	ON INSERT	ON DELETE	ON UPDATE	
1	no			
2	yes	restrict	restrict	
3			cascade	X
4			set null	O
5		cascade	restrict	X
6			cascade	X
7			set null	XO
8		set null	restrict	O
9			cascade	OX
10			set null	OO

## Acknowledgements

The author thanks the editor and several anonymous referees for their very helpful comments.

## References

- Connolly, T. & Berg, C.,(2010). *Database Systems* 5th. Addison Wesley, Boston.
- Date, C.J.,2000. *An Introduction to Database Systems* 7ed. . Addison Wesley, Reading MA.
- Elmasi, R., & Navathe, S.B. (2011). *Fundamentals of database systems* (6<sup>th</sup> ed.). Boston: Addison Wesley.
- Hoffer, J. A, Ramesh, V., &Tope, H., (2011). *Modern database management* (10th ed.). Boston: Prentice Hall.
- Kroenke, D. M., & Auer, D. J. (2012). *Database processing: Fundamentals, design, and implementation* (12th ed.). Pearson,
- Loney, K. (2009). *Oracle database 11g: The complete reference*. New York: McGraw Hill.
- Microsoft. (2010). *Microsoft Office Professional Plus, 2010*. Access.
- Oracle. (2011a). Oracle Database SQL Language Reference 11g Release 2, [http://download.oracle.com/docs/cd/E11882\\_01/server.112/e26088.pdf](http://download.oracle.com/docs/cd/E11882_01/server.112/e26088.pdf), 21 Oct 2011.
- Oracle. (2011b.) Oracle SQL Developer Data Modeler 11g, <http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/index.html>
- Song, I, Evans, M., &Park, E.K., 1995. A comparative analysis of entity-relationship diagrams. *Journal of Computer and Software Engineering*, 3(4), 427-459.

## Biography

**Norman Pendegraft** is Professor of management Information Systems in the College of Business and Economics at the University of Idaho. He teaches database design and telecommunications management. His research interests include modeling and simulation of information security economics and information systems pedagogy.