

MODELING THE MANY-TO-MANY RELATIONSHIP USING MULTI-VALUED FOREIGN KEYS

Dr. Durward Jackson, California State University, LA, djackson@calstatela.edu
Dr. Ming Wang, California State University, LA, ming.wang@calstatela.edu

ABSTRACT

Modeling the many-to-many relationship in database design has been an important topic in traditional database textbooks for many years. Using these textbooks, database, instructors are able to teach students the correct approach, but it is difficult to explain the reasoning behind the algorithms and rules. This paper introduces a new way of teaching M:N relationships and their mapping from the entity-relationship model to the relational data model using multi-valued foreign keys. The aim is to provide a complementary approach to traditional ways of teaching the M:N relationship. We first describe a case scenario that we will use to describe our new approach. Secondly, we introduce a new way of mapping the M:N case to the relational model using multi-valued foreign key attributes. Finally we present some teaching techniques for enhancing students' understanding of the M:N relationship.

Keywords: Database design, Relational data model, Logical design, Database curriculum

INTRODUCTION

The many-to-many (M:N) relationship in database design process is not as intuitive to students as many database educators think. Students often have difficulty with the M:N relationship and its mapping from an entity-relationship model to a relational model. They may be confused when confronted with the M:N relationship even when they understand the relational model and 1:1 and 1:N relationships. This problem becomes more acute when UML is used in conceptual and logical database design, because the bridge table must be illustrated as a class in the UML class diagram with multiplicity notation.

The M:N relationship in the entity-relationship model can be a challenge without the explanatory power of actual tables loaded with sample data, because the entity-relationship modeling is highly abstract. The problem arises for students when the instructor fails to use tables and actual data when discussing the application of the algorithm. Database textbook authors have tried to make it understandable to students in two ways. The first way is to illustrate the M:N relationship with object occurrences in participation constraints [2, 5]. The second way is to present the mapping algorithm for mapping M:N relationships to the relational database and make a mapping rule that for each binary M:N relationship type R, create a new relation S to represent R, including as foreign key attributes in S the primary keys of the relations that represent the participating entity types so that their combination will form the primary key of S [2, 1, 4, 6].

This paper introduces a new way of teaching the mapping of M:N relationships. The aim is to provide a complementary approach to traditional ways of teaching the M:N case for the database courses. We first describe a case scenario and then present a way of mapping the M:N case to the

relational model using multi-valued foreign key attributes. Finally we present techniques for enhancing students' understanding of the M:N category.

CASE SCENARIO

In presenting the mapping process in database design, we use the entities EMPLOYEE and PROJECT and their many-to-many relationship. This scenario applies to consulting companies that are organized on a contract basis where employees are assigned to work on projects. For example, an aerospace consulting company might have projects such as Interstellar Explorer, Jupiter Probe and Mars Lander that have been assigned to engineers from its various engineering departments. Sometimes it is convenient, or even essential to associate attributes with a relationship [3]. Thus, the *Hours* attribute is associate with the Is-assign-to relationship. The entity-relationship diagram in database conceptual design derives from a set of business rules that govern structure and operations of an organization. In our example, the process of discovery of the M:N relationship could have resulted in the following business rule and diagrams:

One EMPLOYEE must be assigned to many PROJECTs.
 One PROJECT must have many EMPLOYEEs assigned.

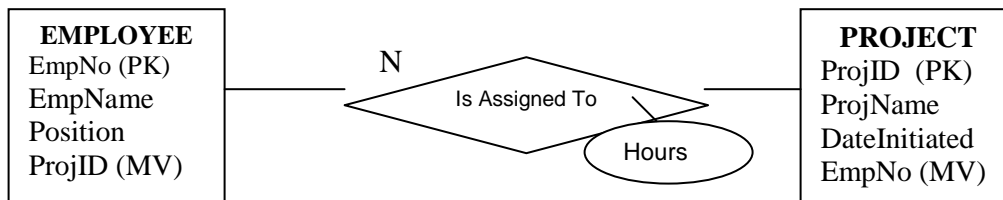


Figure 1. Entity-Relationship Diagram

Alternatively, an M:N relationship (is-assigned-to) can be replaced by a composite entity type (Assignment) and two 1:M relationships (Figure 2). The relationship name (is-assigned-to) has been changed to a noun (Assignment) to follow the convention of nouns for entity type names. The M:N relationship shown in Figure 1 is decomposed into the two 1:n relationships as follows:

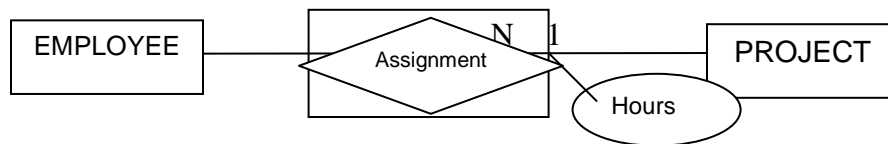


Figure 2. Entity-Relationship Diagram with a Composite Entity

Figure 2 can be interpreted as the following business rules:

1. One EMPLOYEE must have one or many ASSIGNMENTs and one ASSIGNMENT must have one and only one EMPLOYEE.
2. One PROJECT must have one or many ASSIGNMENTs and one ASSIGNMENT must involve one and only one PROJECT.

Finally, this EMPLOYEE - PROJECT M:N relationship can be converted to the UML Class Diagram (Figure 3). The M:N relationship name (is-assigned-to) is changed to the Assignment class.

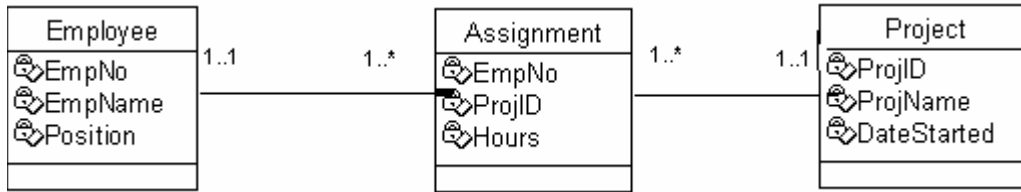


Figure 3. UML Class Diagram Drawn from Rational Rose 2000

The relational data model mapped from Figure 1, Figure 2 and Figure 3 database design diagrams would be identical even though the above three figures are drawn differently. The details will be presented in the following section.

MAPPING TO RELATIONAL DATA MODEL

Using the concept of multi-value foreign keys to model the M:N relationship will make students understand this mapping process better. It is important for instructors to illustrate relations with actual tables and data so that students are able to understand how and why the M:N ER model maps to the relational model. Through years of teaching experience, we have developed our own way to teach mapping the M:N relationship from conceptual to logical design, and have summarized the process in the following steps.

Step One: entity -----> relation

The first step to map an entity to a relation is to create a relation with the entity name that includes all the attributes of the entity and to choose one or more of the key attributes of the entity as the primary key for the relation. Again, it is important that both the EMPLOYEE and PROJECT tables be loaded with a minimal set of sample data so that students will have an intuitive, focused view of the database structure.

Employee			Project		
EmpID	EmpName	Position	ProjID	ProjName	DateInitiated
1	Tuan	Aerospace Engineer	E1	Interstellar	06/01/2004
2	George	Aerospace Engineer	JP12	Jupiter Probe	05/05/1999
3	Allison	Training Coordinator	ML5	Mars Lander	03/30/1995
4	Consuela	Mathematician			

Figure 4. Mapping From Entities to Relations Loaded with Data

Step Two: Analysis of the M:N Relationship with Multi-valued Foreign Keys

Consider the PROJECT entity. Its relationship with EMPLOYEE may be described in the ER model with the multi-valued (MV) EmpNo (each PROJECT has multiple EMPLOYEEs). When mapped to relations, the MV EmpNo must be promoted to the status of an entity (table) with

EmpNo, ProjID as the primary key of the new table. The reverse would be true if we chose to carry the MV foreign key ProjID in the EMPLOYEE table.

What if we chose to have multi-valued foreign keys in both the EMPLOYEE and PROJECT tables? The result would be the same, because the two tables that result from mapping both multi-valued attributes can be merged into one as shown below. The additional table in this case would have a primary key of ProjID, EmpNo with the same two foreign keys as before. Thus, we might derive the following mirror images in two new tables.

EmpNo	ProjID
1	JP12
1	E1
3	E1

ProjID	EmpNo
JP12	1
E1	1
E1	3

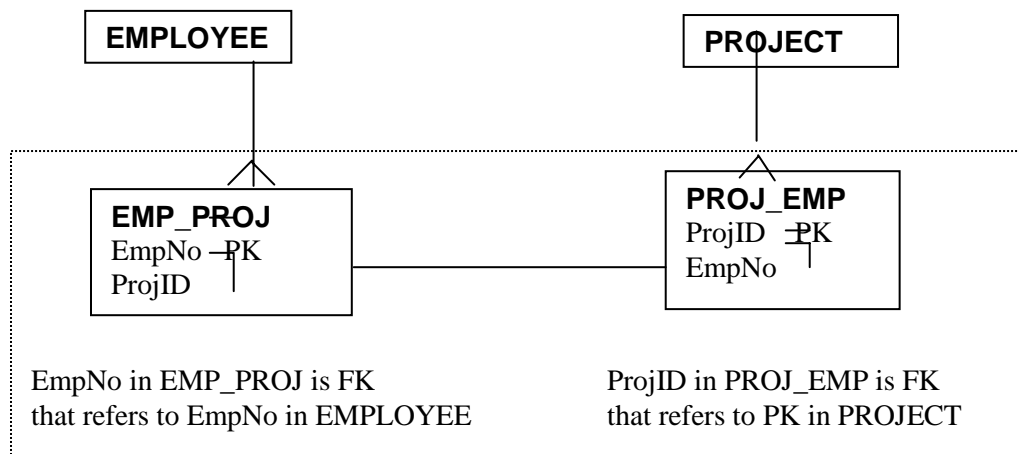
Figure 5. Multi-valued Attributes in Two Mirror Image Tables

Analysis of this result would yield the insight that these two tables have a 1 to 1 relationship, mandatory on both sides. The fact that the two columns composing the primary keys are reversed in the two tables does not negate the fact that they represent the same reality and can therefore be merged into one table called Assignment (Emp_Proj or Proj_Emp). It would be appropriate to associate a new attribute hours with the Assignment table since hours is functionally dependent on the primary key (EmpNo, ProjID). The following diagram shows this result as a third table generated from the mapping of the Employee-Project relationship.

Assignment		
EmpNO	ProjID	Hours
1	E1	10
1	JP12	25
3	E1	30

Step Three: Process Summary

The new ASSIGNMENT table contains two foreign keys, EmpNo and ProjID. The process can be summarized in the following analytical mapping process diagram:



Step Four: Relational Schema

The process of creating the relational schema is rather straightforward. First, decompose the multi-valued attributes (see Figure 1) by promoting them to the status of entities. Second, merge the two new entities, choosing either entity's identifier as the identifier of the merged entity (shown as a dotted line in Figure 6). Third, map each of the three entities into relations as shown in Figure 7.

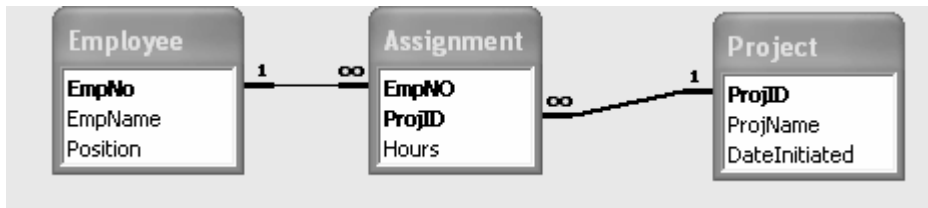


Figure 7. Relational Schema Mapped from the Employee-Project Relationship

CONCLUDING REMARKS

The combination of traditional methods and our innovation has proved to be successful. In conclusion, we have found that use of multi-valued foreign key attributes in mapping the M:N relationship enhances student learning. The learning outcome can be observed in student term projects and their final exams. Using the multi-valued foreign key attributes, we are able to provide students a thorough explanation of the M:N entity-relationship mapping process that complements other approaches.

We suggest teaching the multi-valued attribute mapping technique prior to teaching the mapping of the M:N relationship. It is easier for students to understand our approach to the M:N relationship mapping process with prior knowledge of multi-valued attribute mapping. Our course organization for modeling M:N relationship in our database course is as follows:

1. Introduce the M:N relationship with participation constraints [2].
2. Model the M:N relationship using an ER diagram.
3. Present mapping algorithms for the mapping of M:N relationships to the relational database.
4. Present mapping technique for handling MV attributes.
5. Show the mirror image tables with multi-valued foreign key attributes and actual data instances.
6. Present intermediate process for mapping from E-R model to relational model using the proposed mapping diagram with multi-valued foreign key attributes.
7. Illustrate the final result with the relational schema.

Of course, students need to be given the sufficient practice with business cases using this data modeling technique. The lecture needs to be followed with homework and/or classroom exercises. Additionally, a well-defined project needs to cover M:N relationship modeling practice.

REFERENCES

1. Connolly, T. & Begg, C. (2005). *Database systems: A practical approach to design, implementation, and management* (4th ed.). Addison Wesley.
2. Elmasri, R. & Navathe, S. (2004). *Fundamentals of database systems* (4th ed.). Addison Wesley.
3. Garcia-Molina, H., Ullman, J., & Widom, J. (2003). *Database Systems: The complete book*, Prentice Hall, Upper Saddle River, NJ.
4. Kroenke, D. (2001). *Database processing: Fundamentals, design and implementation*, (8th ed.). Prentice Hall, Upper Saddle River.
5. Ramakrishnan, R. & Gehrke, J. (2000). *Database management systems* (2nd ed.). McGraw Hill, Boston.
6. Recardo, C. (2004). *Databases Illustrated*, Jones and Bartlett, Subury, MT.