The purpose of a data model is to express the semantics of the real world that it models. One common situation that should be modeled is that of many-to-many relationships between objects (entities). In this paper we distinguish between three different types of m:n relationships: non-constrained, constrained and mandatory. Their meaning and consequences are discussed along with examples from entity-relationship (ER), binary-relationship (BR) and relational models. These refined m:n relationship types bring to light additional semantics, thereby facilitate analysis and representation of the data structure.

A relationship type R between n objects O₁, O₂, .., Oₙ is a set of associations between these types of objects. R is a set of relationship instances rᵢ, where each rᵢ is an n-tuple of objects (o₁ᵢ, o₂ᵢ, .., oₙᵢ) and each object oᵢ in rᵢ is a member of object type Oᵢ (Elmasri and Navathe, 1989). The degree of a relationship type is defined by the number of participating object types. The cardinality ratio constraint on a relationship, also termed the relationship type, specifies the number of instances in which an object may participate. Usually, data models distinguish between one-to-one (1:1), one-to-many (1:n, or n:1), and many-to-many (m:n) cardinalities. An m:n relationship between objects means that an object of a certain type can be associated with many objects of another type, and vice versa.

In a semantic data model, e.g. ER, an m:n relationship is represented diagrammatically as a diamond-shape which is linked to the respective objects (termed entities), each represented by a rectangle; the m:n cardinality is marked inside the diamond or on the links (depending on the specific variant of this model). In the relational model the m:n relationship appears in the form of a multi-key field of a table (relation), where the m:n related objects (fields, attributes) constitute its key. In the CODASYL (network) model, the m:n relationship is represented as a “linking” record-type, which is a member of the “owner” record-types representing the associated objects. m:n relationships exist between more than two objects, and within m:n relationships of a higher degree; e.g., a multi-valued dependency (MVD) between subsets of a key consisting of three or more fields. Within any of the specific data models, all m:n relationships are represented in the same way, and no distinction is made between different sub-types.

In an earlier study (Shoval, 1991), I examined the 1:1 relationship type, specifically, how it is mapped from a semantic model (e.g. BR) to the relational one. Normally, either of the objects in the 1:1 relationship can be viewed as the determinant of the other. Therefore one object may be mapped to a primary-key of a table, whereby the other object becomes a non-key field. On the basis of the normalization theory, either solution is correct. However, I showed that this is not necessarily so: if other factors are considered (e.g., the number of tables in the schema or the enforcement of integrity constraints), the decision as to which object becomes the determinant (key) and which the dependent (non-key) is no longer arbitrary, but depends on the other relationship types.

In this paper I analyze the m:n relationship type and show that there are different types of m:n relationships, which have to be identified in order to increase the semantics of the target database schema. Before doing so, I review the current treatment of m:n relationships. As already mentioned, in the relational model, m:n relationships appear as multi-key fields. The design of a relational schema, including its tables and their key and non-key fields, is based on the normalization theory, which relies only on two types of dependencies be-
tween entities: a) functional dependencies, which support the design of tables up to the level of 3NF and BCNF, and b) multivalued dependencies (MVDs) between subsets of key fields consisting of three or more fields, which support the design of tables in 4NF and 5NF. For more detail on normalization see Date (1986); Elmasri and Navathe (1989); and Kent (1983).

The normalization theory, however, does not deal explicitly with m:n relationships (Ling, 1985). It turns out only indirectly that m:n relationships are represented in the relational schema as multi-key fields of tables, because functional dependencies and MVDs (which become the key of a table) may involve determinants consisting of more than one field. Beyond that, neither m:n relationships nor different types of such relationships are treated in the relational model. Ling (1985) demonstrated that there is a difficulty in identifying and using MVDs in the database design process, a difficulty which can be overcome by using of the entity-relationship (ER) model, which treats m:n (and other) relationships. Once reality is analyzed and represented in an ER schema, including m:n relationships, the schema can be transformed to a normalized relational schema, circumventing the issue of MVDs.

Semantic models, such as ER, handle m:n relationships (as well as 1:1 and 1:n, or n:1 relationships) but they too do not make a distinction between different sub-types of these relationships. Sometimes relationships in the semantic models are represented by upper (max.) and lower (min.) bounds; i.e., there may be a constraint on the number of occurrences. For example, assume that in an m:n relationship between employees and projects, an employee may work on a restricted number of projects. Such a constraint can be expressed in the ER and BR models (but not in the relational one) by specifying the minimum and maximum number of occurrences of each type of object. Minimum bounds are usually 1 or 0; the maximum is usually not bound (denoted by just n), but sometimes it is bound to a specific number. This type of constraint is not dealt with in this paper, as it applies to all types of relationships.

My objective is to show that in reality there are different types of m:n relationships, with different meanings (semantics). These have to be identified and expressed properly in the database schema, at the conceptual or logical level, in order to increase its semantics. I enhance the expressive power of the data models by making a distinction between non-constrained (simple), constrained, and mandatory m:n relationships, and discuss the meaning, consequences and representation of each type. Examples from the relational, entity-relationship (ER) and binary-relationship (BR) models are provided.

**Non-Constrained M:N Relationships**

In a simple, non-constrained m:n relationship, an object of a certain type may be associated with any object of another type, and vice versa. More formally, we define a non-

---

**Figure 1: Non-Constrained m:n Relationships**
Related Content

Rough Sets
www.igi-global.com/chapter/rough-sets/11207?camid=4v1a

Optimization of Multidimensional Aggregates in Data Warehouses
www.igi-global.com/article/optimization-multidimensional-aggregates-data-warehouses/3367?camid=4v1a

Ensuring Correctness, Completeness, and Freshness for Outsourced Tree-Indexed Data
www.igi-global.com/chapter/ensuring-correctness-completeness-freshness-outsourced/8031?camid=4v1a

Similarity Learning for Motion Estimation
Shaohua Kevin Zhou, Jie Shao, Bogdan Georgescu and Dorin Comaniciu (2009). Semantic Mining Technologies for Multimedia Databases (pp. 130-151).
www.igi-global.com/chapter/similarity-learning-motion-estimation/28831?camid=4v1a