Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Christian Mancas[1]

[1] Bucharest Polytechnic University

Received: 11 December 2015 Accepted: 4 January 2016 Published: 15 January 2016

Abstract

I.Unfortunately, the widespread used one-to-many, many-to-one, one-to-one, and many-tomany database relationships lack precision and are very often leading to confusions that affect the quality of conceptual data modeling and database design. This paper advocates replacing them with the rigorous math notions of relations and (one-to-one) functions.

Index terms—

1 Christian Mancas

Unfortunately, the widespread used one-to-many, many-to-one, one-to-one, and many-to-many database relationships lack precision and are very often leading to confusions that affect the quality of conceptual data modeling and database design. This paper advocates replacing them with the rigorous math notions of relations and (one-to-one) functions.

he widely used Entity-Relationship (E-R) Data Model (E-RDM, e.g. [Chen, 1976], [Thalheim, 2000], [Mancas, 2015]) is and will continue to be successful in database (db) design mainly due to the graphical nature of its E-R Diagrams (E-RDs) and simplicity.

2 a) E-RDs

In its original version [Chen, 1976], atomic (entity-type) object sets are represented in E-RDs by rectangles, compound (relationship-type) ones by diamonds, and the Relational Data Model (RDM, e.g. [Codd, 1970], [Abiteboul et al., 1995], [Mancas, 2015]) attributes (object set properties) by ellipsis (attached to the corresponding rectangles and diamonds).

Structural E-RDs only contain rectangles and diamonds (which connect rectangles), without any ellipsis. As such, they are non-directed graphs whose nodes are rectangles and diamonds and whose edges are so-called "roles" (of the connected entity-type object sets in the corresponding relationship-type ones). Figure 1 shows an example of a Chen-style E-RD, while Figure 2 Roles have associated cardinalities. For example, read from left to right, CITIES_COUNTRIES is said to be a many-to-one relationship (as there generally are many cities in a country) and this is why belongs to has cardinality n, while hashas 1. Obviously, read from right to left, it is a one-to-many relationship (as generally a country has many cities). Similarly, COUNTRIES_CAPITALS is said to be a one-to-one relationship (as countries may have only one capital and any city may be the capital of only one country) and this is why both is capital and has capital have cardinality 1.

We are using a slightly different notation [Mancas, 2015]: just like in its original version, atomic (entity-type) object sets are represented by rectangles, mathematic non-functional relation type ones (i.e. subsets of Cartesian products) are represented by diamonds, but functional ones are represented as arrows, just like in math. Hence, in our version structural E-RD (from now on abbreviated as E-RD) are oriented graphs whose nodes are only object sets and whose edges are structuralfunctions (i.e. functions defined on and taking values from object sets 1 1 as compared to attribute-type ones, also defined on object sets, but taking values into (subsets of) data types (e.g. Population : CITIES? [0, 3*10 6])

) . For example, as, in fact, both CITIES_COUNTRIES and COUNTRIES_CAPITALS are functional, Figure 4 shows the equivalent of the Chenstyle E-RD from Figure 2.

3 The math-style E-RD equivalent to the one in

Figure 2 As MARRIAGES is not functional, our math-type notation is identical to the Chen-type one from Figure 3.

4 b) Corresponding mathematical relations

Recall that, algebraically, a relation is a nonempty subset of a Cartesian product. First (minor) difference of db relationships as compared to math relations is that they may be empty (at least immediately after they are declared and up to the moment when a first element is inserted into their instances, but possibly also afterwards, whenever their instances are emptied by deleting all of their elements and up to the moment when new elements are again inserted into them).

Second (major) difference between them is that the math ones are positional (as Cartesian products are non-commutative), whereas db ones are not: they only require that all roles of any relationship be pairwise distinct.

For example, mathematically, CITIES×COUNTRIES ?COUNTRIES ×CITIES, which means that when both relationships from Figures 2 and 5 are read either from left to right or from right to left they are distinct, whereas from the db perspective they are strictly equivalent, no matter how are they read (which would correspond to the equivalence classes of Cartesian products immune to the permutations of their member sets).

5 Country COUNTRIES CITIES Capital

Also recall that there is a very important particular case of math relations, namely the functions (mappings); a function is a binary relation satisfying two additional constraints: it is totally defined and it is functional. Read from left to right, the first set is called the domain, while the second is called co-domain. For example, the function Country :CITIES?COUNTRIES has domain CITIES and co-domain COUNTRIES and it is a function because is totally defined (that is any city belongs to a country) and functional (i.e. any city belongs to only one country).

Database functions (which in relational ones are implemented as table columns) differ slightly from math ones only because totality is not compulsory: for example, as capitals might not be temporarily known or of interest for any country, the function Capital :COUNTRIES?CITIESmay not be totally defined.

Totality is considered in dbs as a constraint that has to be explicitly asserted whenever desired. For example, in the (Elementary) Mathematical Data Model ((E)MDM, e.g. [Mancas, 1990], **??**Mancas, 2016]), the complete declaration of Country is Country : CITIES?COUNTRIES, total. In RDM, this is called a notnull constraint, meaning that the corresponding column does not accept null values (i.e. distinguished values represented either as null strings or with the keyword <NULL>). Considering a countable distinguished set NULLS, a possible dual (E)MDMnotation for the above two functions is Country : CITIES?COUNTRIES and Capital : COUNTRIES?CITIES? NULLS, respectively, in which case total definition is always satisfied, just like math.

Obviously, Capital is a one-to-one function, i.e. one for which to any pair of distinct domain elements corresponds a pair of distinct function values. This is why, in our notation (e.g. Figures 4 and 6) its arrow is a double one, and its complete (E)MD Mdefinition is Capital : COUNTRIESâ??”CITIES.

Note that roles of non-functional relationships (e.g. Husband and Wife from Figure 3 above) are also structural functions, namely canonical Cartesian ? Their names are confusing: obviously, for example, both Country and Capital are much clearer than CITIES_COUNTRIES and COUNTRIES_CAPITALS; a clear sign that they are unnatural objects is that they lack natural names, which only exist for nonfunctional relationships (e.g. STOCKS instead of WAREHOUSES_PRODUCTS). ? The need for three distinct names (for the relationship and its two roles) instead of only one (the function) is also unnatural. Again, as compared to non-functional relationship role names, which are natural (e.g. Husband, Wife, Product, Warehouse, Home Team, Visiting Team, etc.), they generally have an Artificial Intelligence flavor (e.g. is, has, belongs, etc.), not a db or math one. ? The redundancy of one-to-many relationships: as we read math from left to right, functions are manyto-one relationships; one-to-many ones are the same corresponding functions, but read from right to left (i.e. from the co-domain to the domain).

6 b) Confusion between one-to-oneness and bijectivity

Not only beginners, but also, for example, MS Access designers are confusing one-to-oneness with bijectivity. For example, if you first declare Capital as a (unique) key (i.e. as being one-to-one) and then try to enforce its referential integrity, depending on the instances of the two tables it relates, you might not succeed in either enforcing it (when there are more cities than countries, whichis the norm) or inserting data in any There is only one advantage in using E-RD relationships, especially when using our simpler and math-type notation: the fact that they are graphic (and a good picture is worth thousand words). Unfortunately, there are much more important disadvantages as well.

7 a) Unnaturalness of Chen-type functional relationships

Representing functional relationships as diamonds has several pitfalls: ? It is true that, being particular cases of binary relations, they can be thought of as object sets as well (in particular, sets of elements of the type <x, f(x)>), but, in fact, both mathematically (which considers them functions, not sets) and from the db point of view (which, by applying the Key Propagation Principle [Mancas, 2015], implements them as table columns, in particular foreign keys) they are not dealt with as such, just like the nonfunctional ones (which are implemented as tables, just like for the entity-type ones).

of the two involved tables (when both instances are empty, enforcing referential integrity succeeds, but then you may not enter either cities, as there are no corresponding countries, or countries, as there are no corresponding cities). This is clearly due to the confusion done between one-to-oneness and bijectivity (i.e. one-tooneness and ontoness). 2

8 c) The many-to-many relationships trap

The worst issue with db relationships is that they may not even correspond to object sets.

For example, if you enforce uniqueness of elements in the above MARRIAGES (i.e. uniqueness of the product Husband?Wife), then you may not store remarriages (e.g. Elisabeth Taylor and Richard Burton married and divorced each other several times). If you do not enforce it, then it is not even a set, as it accepts duplicates.

Generally, you have to validate data modeling correctness for each relationship, by checking the oneto-oneness of the product of all of its roles: if it is not (like for MARRIAGES, where Husband?Wifeis not one-toone), then the corresponding relationship is ill-defined (and either it lacks at least another role or it is, in fact an entity-type object set).

Consequently, the correctmodel in all contexts in which divorce (hence, remarrying) is possible is the one in Figure **??**: Correct data model of MARRIAGES (as an entity, not relationship-type object set, like it is incorrectly modeled in Figure 3)

To conclude with, during conceptual data modeling and db design it is always much, much better to think in terms of math relations and functions, rather than in those of one-to-many, many-to-one, one-to-one, and many-to-many ones. 3 [1] [2]

1

Figure 1: Figure 1 :

2

Figure 2: Figure 2 :

Year
2016
13
(C)

Introduction

Name Population Name Code Population TelPrefix

ZipCode

belongs to has

n CITIES_COUNTRIES 1

CITIES COUNTRIES

1 COUNTRIES_CAPITALS 1

is capital has capital

belongs to has

n CITIES_COUNTRIES 1

CITIES COUNTRIES

1 COUNTRIES_CAPITALS 1

is capital has capital

Figure 3:

[2] Fortunately, there is a workaround for it in MS Access too: if you first enforce referential integrity and only then uniqueness, no issue arises.

.1 References Références Referencias

[Mancas ()] *A deeper insight into the Mathematical Data Model*, C Mancas . 1990. ISDBMS, Mamaia, Romania. p. . (Mancas, 1990. 13th Intl. Seminar on DBMS)

[Codd ()] 'A relational model for large shared data banks'. E F Codd . *CACM* 1970. 13 (6) p. . (Codd, 1970)

[Abiteboul et al. ()] *Addison-Wesley: Reading, MA. Otherwise, you risk confusions between one-tomany and many-to-one ones, one-to-oneness and bijectivity, and even between relationship and entity-type object sets. Moreover, our E-RD notations*, S Abiteboul , R Hull , V Vianu . 1995. 1995. (Foundations of Databases. Mancas, 2015] are much simpler, natural, and close to math than the original ones)

[Mancas ()] 'Conceptual Data Modeling and DB Design'. C Mancas . *The Shortest Available Path*, (NJ) 2015. Apple Academic Press. I.

[Thalheim ()] *Fundamentals of Entity-Relationship Modeling*, B Thalheim . 2000. 2000. Berlin: Springer-Verlag.

[Chen ()] 'The entityrelationship model: Toward a unified view of data'. P P Chen . *ACM TODS* 1976. 1976. 1 (1) p. .