Has there ever been a relational DBMS?

Rafael Camps i Paré

Report LSI-97-19-T
HAS THERE EVER BEEN A RELATIONAL DBMS?

R. Camps. Escola Universitària Politècnica. Vilanova i la Geltrú. 08800 - Spain

According to C. J. Date, the right way to go forward in the DB field in order to provide all of the functionality promised by object-oriented systems, and more, "is to do nothing" [15] to the relational model, RM, except implement it, which, according to him, doesn't seem to have been tried very much.

Which RM does he mean? Is there more than one single RM? Has there even been a DBMS "really" relational?

In The Third Manifesto [7] Darwen and Date declare, under the heading Back to the future, that the future should be built on the original RM as invented by Codd. Further on, they tell us that

"The version of the Relational Model that we espouse is, very specifically, that version first described in reference [13] (chapter 15) and further refined (slightly) in reference [12] (6th ed) part II). Note, however, that the definitions given herein for tuple and relation represent a small improvement over the definitions given in those earlier publications."

They find compelled to make this clarification owing to the significant extensions and modifications that the RM has undergone along the years, so that to simply say 'RM' is not specific enough.

In this paper I intend to show that the classical RM is not a unique rock solid model (even if we ignore some "deviations" like Nested-RM [11]).

Let us look back. The knowledge of the past can help us understand the present and build a better future.

CODD: THE CREATOR

In 1970, E. F. Codd put out the now so well-known paper [6], which gave birth to the whole relational world. In this first paper, though leaving many loose ends, he laid down the basic ideas of the RM, which were to have so much influence and significance. During the following years, until the late Seventies, Codd went on refining and completing his original RM idea, the one he later named Relational Model Version 1, RM/V1. This can be considered as the theoretical foundation on which RDBMSs are built.

In 1979, when the first commercial RDBMSs were entering into the market, Codd published, as the outcome of the evolution of his ideas, a paper [5] with a proposal for a new data model, RM/T. Although developed from the RM concepts, this new model was in the line of the semantic models then fashionable in the academic world. It was built on the then already well-known concepts of entity, associative entity, generalization/specialization, and so on. It also included "temporal" features, such as event precedence. The RM/T meant a huge change from the RM approach as it had been understood until then (RM/V1). The new model did not raise great enthusiasm among the software companies that were developing or starting to sell RDBMS products in accordance with the original RM line.

In 1981 and 1985, Codd wrote two papers [3,4] where, in a way, he implicitly redefined the RM. He did not regard them as new versions of the RM. M. Stonebraker (creator of Ingres) would say in 1988 that "It appears that Ted Codd has been designated the keeper of the faith and has the individual initiative to redefine the relational model whenever appropriate," and warned that "data models have much in common with religion... I people beliefs are not based on rational principles."

Later, in 1990, Codd [2] put forward a second version, RM/V2, in the initial RM line. In that proposal, 333 features are stated as constituting RM/V2. It appears that this text has had no real influence. In the foreword, it can be read that,

"It is also the first book on the relational model by the originator of that model. [...] Vendors of DBMS products have in many cases failed to understand the first version"
RM/V1, let alone RM/T." Therefore, the RM/T will be "gradually dropped into the sequence of versions RM/V2, RM/V3... the dropping will be gradual to allow DBMS vendors, and consumers time to understand them. [...] DBMS vendors, users, and inventors of new data models [...] seem to be unaware of the scope of the relational model and the scope of database management [...] the industry has not been able to maintain an adequate pace of product development and improvement."

Codd seems to take for granted that software developers shall follow the way that he, the Creator, dictates. But, mortals not being able to keep up with him, he must show them the way step by step. As far as I know, he has not published any other proposal for a new version; Perhaps he is waiting for DBMS developers and consumers to overcome their ignorance on database management and to finally understand the 333 features in RM/V2.

DATE: THE APOSTLE

From the mid Seventies to the late Eighties, C. J. Date wrote a large number of papers, with the purpose of spreading and clarifying the RM concepts as defined by Codd. Thereafter, in disagreement with Codd, he went on publishing to contribute his own vision (and version), subject to a slow, continuous evolution. But the main source for Date’s vision of relational databases is the book that made him famous, the best-seller “An Introduction to Database Systems” [12]. Published for the first time in 1976, its 6th edition was released in 1995. An examination of the changes introduced in the text by Date from edition to edition of his bible gives good hints of the evolution in the relational world. On this evolution, Date himself has said that

"Relational definitions have changed somewhat over the past ten years; criticism that relational system is a moving target are not unfounded." 1981 ([12] 3rd ed.)

"the relational model is not a static thing; it has been evolving over time [...] Whether our approach or Codd’s is adopted, however, one thing is certain: the model will continue to evolve." 1990 (5th ed., in a section headed “Whither the Relational Model?”)

Some areas of the RM where there have been more changes in official opinion, including major disagreements between Codd and Date, are those dealing with domains and value atomicity, nulls, primary key requirement, etc.

THE DOMAIN OF DOMAINS

A good example of radical changes over the years is the term domain, which is the basis of the intended OO revitalization/reincarnation of the RM supported in The Third Manifesto. Here are some examples from C.J.Date texts:

1975 [12] "Every value within a relation -- i.e., each domain in each tuple-- is an atomic (non-decomposable) data item (e.g., a number or a character string)." Nowhere in the book was any example with a domain more complex, not even any example using dates.


1990 [9] "In fact, it is not clear that there is any fundamental reason why data values need to be atomic"

1995 The Third Manifesto [7]. "it shall be possible to define a domain whose values are relations". Such prescription is coming from RM, according to the authors!

More details on domains, value atomicity, etc, can be found in [1].

THE THIRD MANIFESTO

In "The Third Manifesto", Darwen and Date propose the creation of a language, D, grounded on the RM. This language should be subject to:

- 33 Prescriptions (26 of them coming from the RM, plus 7 labeled as OO)
- 14 Proscriptions (10 of them belonging to the RM, and 4 being OO)
- 13 Very Strong Suggestions (9 arising from the RM, and 4 from the OO approach)

Darwen and Date take as RM the latest version of what is in Date’s published works. In case
there still were doubts, they insist that the RM version they adopt is not RM/V2, nor RM/V1. In fact, Date had already given in 1992 a "reconstituted" version of RM/V1, deeming that Codd's was wrong.

From the set of 60 features (prescriptions, proscriptions, and very strong suggestions) defined by Darwen and Date for the D language, I summarize below those that may be the most unexpected to the traditional user of the relational approach:

- Domains may have arbitrarily complex, encapsulated values. **Domain** and **data type** are interchangeable. There can be domains whose values are **tuples or relations**. There can be type constructors such as **TUPLE, RELATION, LIST, ARRAY, and SET.**

- Domains, and not relations, correspond with the OO concept of **object class**. This is the main idea in the Manifesto, and, according to the authors, it places them against the current trend in **Object-Relational** prototypes and products.

- There must be operators to "expose the actual representation" (in fact, one possible representation) of the domain values. These would be the only operators to violate encapsulation.

- Truth values domain. "**No more nulls and no more many-valued logic.**"

- For every base relation there must be, at least, one candidate key. For every derived relation, it must be possible to define candidate keys: there must be a restriction inference mechanism, able to infer candidate keys in derived relations. Definition of a primary key is not compulsory, neither for base nor for derived relations.

- D should include support for system generated keys. "Object-IDs" are rejected.

- Simple and multiple inheritance.

- Nested transactions.

- Single memory level and computationally complete DB language approach.

- Tuple by tuple operations must not be allowed (particularly, via **cursor updates** must not be allowed).

- D shall not be called SQL.

**SO, WHAT IS THE RM?**

From what is said above, I believe it clear enough that, to the question "**What is the RM?**", the answer may be, "**Which of them all?**" or "**From which author and date?**" Nevertheless, there is a set of basic fundamental ideas that have persisted. These may be summarized in the following terms:

- Attribute, Domain, Tuple, Relation
- Relational algebra and calculus
- Derived relations (or Views)
- Restrictions: identity, domain, referential integrity.

But the details and the coverage of these concepts have not persisted.

The huge research effort, not interrupted since 1970, into the fundamental ideas known as the RM has borne a large number of improvements in the database world, mainly owing to the formal tools, of proven utility, provided by its theoretical basis. A significant part of the current research in the database field still uses basic concepts introduced by the RM. All this, however, should not make us forget that there is no such thing as a single, clearly defined, RM.

**RELATIONABILITY: THE 1981 RULES**

In his lecture on being awarded the ACM Turing Prize in 1981, Codd said that

"the time has come to draw a very sharp line between relational and non-relational database systems, so that the label relational will not be used in misleading ways."

As the RM configuration was changing and diversifying, so were doing the criteria to officially label as 'relational quality' a DBMS, although the changes were not parallel. Traditionally, these criteria are known as **relationability criteria.** But, who has the authority to establish such relationability criteria? At first, everybody thought it reasonable that the Creator himself be the one to establish the **Commandments of the Law.** So
it was until the mid Eighties. Then, the DB community, tired of requirements and prohibitions, began to abandon the respect for the Law of God (observance having never been attained). In fact, Stonebraker [14] remarks that System-R and Ingres, the two experimental systems from which all current commercial systems come, were not "particularly faithful to the relational model."

While there was only Codd's RM/V1 and no commercial relational products, there was no real need for relationality criteria. But as early as 1979, Codd [5] defined what should be understood by a "fully relational" database. This paper, however, had little impact and no practical influence. In 1981, in the 3rd edition of his bible [12], Date took on the subject and wrote that a system can be called "fully relational" if it supports:

- Relational databases (domains, keys, the two integrity rules)
- A relationally complete language

In November 1981, in the Turing Prize lecture, Codd decided to establish the official relationality criteria, and divided the relational systems in three categories, "minimally relational," "relationally complete," and "fully relational." For a DBMS to qualify as relational, it had to support:

- "Tables without user-visible navigation links between them"
- "A data sublanguage with at least [...] the transformations specified by SELECT, PROJECT, and unrestricted JOIN operators"

Codd endorsed a DBMS as "fully relational" if "the full relational algebra is supported" (which implies three-valued logic), and enforcement of "the two integrity rules".

THE COMMANDMENTS OF THE LAW OF CODD

In 1985, when Codd was at the height of fashion, he published a new set of relational criteria in ComputerWorld [3], a popular industry journal not related to the academic and research world. The intent was, therefore, to maximize the influence on vendors and users. In this paper, Codd defined 12 rules that a DBMS must comply with in order to qualify as "fully relational." According to him, there were many products sold with this label that, in fact, were not even "minimally relational". In addition to these 12 basic rules, he stated 30 additional rules which defined the RM. These additional rules, divided in three groups, structural, integrity, and manipulation, do not match exactly neither RM/V1 nor RM/T. This is why these papers can be considered as defining a new RM version.

The 12 basic rules became utterly known, while the additional 30 were totally ignored, in spite of Codd's claim that a mid-eighties DBMS ought to comply with them all. A short time after, almost all DBMS providers, full of fear of Codd, claimed that their product complied with the 12 rules but a few, that would be enforced in its next version. More often than not, this was not true. Even today, only a few of the available commercial products comply with the 12 rules, and there is none that complies with the additional 30 (and will not be, as relationality does not seem to be much of a concern nowadays). As a matter of fact, Codd also stated a zero rule, on which the 12 rules were based, requiring not only that the user interface be relational, but that all database management be relational as well (e.g., a relational catalog).

Basing on all his rules, Codd put forward a scoring method to measure the degree of relationality of a system on a zero to 100 scale. He used his method to compare DB2 against two other DBMSs. However, he did not choose relational products such as ORACLE, INGRES or INFORMIX, then already in the market, but two pre-relational systems that had been extended towards the RM: DATACOM/DB and IDMS/R (a former CODASYL system). According to Codd, DB2 conformed to 7 out of the 12 rules, and the other two systems did not conform to any at all.

As an example, let us look at the first rule, the Information rule, considered by Codd as the main rule (and qualified by Date in 1990 as "the basic principle of the relational model"). This rule requires that "All information [...] is represented [...] in exactly one way—by values in tables." IDMS/R and DATACOM/DB did not conform to this rule, prohibitionist, because they allowed not only "values in tables," but also other means of representing information (such as Coddasyl sets, or element position in a repeating group). Taking all 42 rules into account, Codd gave to DB2 a score of 46% "relationability", and no more than 10% to the
other two systems. By then, IDMS/R was trying to "hide" some of its facilities, banned by the Law, to elude criticism from Codd's fundamentalism.

RELATIONABILITY AFTER 1985

Date said in 1990 [12] that as early as 1986 he had privately told Codd that, although broadly accepting the 12 rules, he could not but disagree with the additional rules defining the RM. Date's and Codd's views on the RM were slowly drifting apart.

In 1990, Codd defined the RM/V2 [2], where the former three parts of the model (structure, integrity and manipulation) became no less than 18. However, by then the subject of relationability was already declining, and the RM/V2 has not had any practical influence. In 1995, in Date's 6th edition of [12], no explicit mention is made of relationability criteria, not even as a historical reference.

In the late Seventies and the early Eighties, the users expected that the mathematical basis of the RM would lead to a single truth and rock-solid standards. The relationability subject is proper to the Eighties. Nowadays, the border lines of the DBMSs are fading and the users are no longer interested in the question "Is my DBMS relational?" (but some vendors will try to substitute this question by "Is my DBMS Object-Relational?"). Simply, it is commonly agreed in the industry that, for a DBMS to be called relational, it must have at least:

- Structures based on tuples and tables
- Views/Derived Tables
- Queries (loosely) based on relational algebra and calculus
- Physical world concealed from the programmer. Designing a DB must be possible with no regard to physical/internal considerations (leaving aside performance requirements)
- Schema/catalog to be handled by the same means as data.

A FINAL COMMENT

Some of the attitudes of the war between the CODASYL and Relational camps, are reappearing. Examples: The fight between ODMG-93 and SQL3, The Manifestos, Universal Servers against "pure-OR", Microsoft positions, etc. An objective view of the "first DB war" debate, free of the bias of commercial, personal, and academic interests, could be very useful today. It is always good to look backwards, as we can learn from past errors [1]. As reminded by Date in [8], "Those who don't know history are doomed to repeat it."

REFERENCES


3  Codd, E.F.: "Is your DBMS really relational?" (1st part), and "Does your DBMS run by the rules?" (2nd part), ComputerWorld 14th and 21st October 1985.


