DATABASE LIBRARY

E. F. Codd challenges the industry with RM/V2

BY DAVID McGOVERAN

A Long Time Coming

DR. E. F. CODD'S LONG-awaited book is now available at last. For anyone who's ever wanted to have all of Codd's thoughts about the relational model in one place, it's all contained in *The Relational Model for Database Management: Version 2.* (Reading, Mass.: Addison-Wesley, 1990. ISBN 0-201-14192-2, 538 pages, $37.75.) Although it's not as technical as some of the papers in which Codd's ideas originally appeared, the key concepts are now gathered under a single cover for the first time. Database users, evaluators, consultants, administrators, vendors, designers, and applications developers will all find something valuable in this book. If you're at all technically inclined and interested in the relational model, you should buy this book.

I wish to emphasize that, in my general view, this is an extremely important work that should not be overlooked. Nevertheless, I don't agree with everything Codd has to say—in fact, I have several criticisms, which I intend to discuss to warn the reader in advance of potential problems, in the hope that the reader will gain more from an informed reading of the book.

Codd's first generally available paper was an IBM Research Report entitled "Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks," and published in 1969. In the interim, he's published over 30 technical papers that elucidate and extend his early concepts. Perhaps most key among those papers was "Extending the Database Relational Model to Capture More Meaning" (*ACM Transactions on Database Systems*, 4(4), 1979). This paper introduced significant new ideas beyond the relational model as it had appeared in earlier papers.

This new book represents the inventor of the relational model's point of view and large collection of the author's contributions. Its tone and wording are prescriptive, which will make it difficult for the typical reader to distinguish between features that are Codd's strongly felt implementation suggestions and those that are absolutely essential to the consistency of the relational model. Readers need to remind themselves that the prescriptive tone shouldn't be taken literally (even if it was intended that way); users and vendors shouldn't naively interpret or implement these recommendations. All readers, and especially vendors, should exercise due caution.

A great deal of the book is likely to engender controversy. Codd's position with regard to missing information and the use of three-valued and four-valued logic (3VL and 4VL), for example, isn't accepted by many experts in the field; the (unproven) potential doesn't appear to outweigh the costs. Codd even states that he doesn't feel that "this part of the relational model rests on [as] solid a theoretical foundation as the other parts."

Similarly, his interpretations and objections to the object-oriented and entity-relationship approaches don't agree with those of many users. For example, it seems unfair to criticize the entity-relationship data model because it doesn't meet one's own definition of a data model, without first determining what the originator meant by *data model*. Still, if Codd's position serves to encourage creation of a forum for further discussion of these points, the entire industry will benefit greatly.

These topics are given considerable attention in the book and numerous (sometimes subtle) points are equally controversial. Unfortunately, not all the questionable material is simply controversial. Some material is erroneous and may detract from, rather than further, Codd's great achievements to date. Codd invented the relational model for database management in the late 1960s. The effect on the computer industry (both software and hardware, user and vendor alike) is now unimpeachable. Nevertheless, the full impact of his invention has yet to evolve; much theoretical and technological work remains.

Since it was first introduced, the relational model has frequently been misunderstood. Some PC DBMS vendors have attempted to redefine the relational database as a flat- or multi-file system, or as any DBMS supporting SQL manipulation. Vendors with only partial support for RM/V1 have claimed to have fully relational products (for marketing reasons or merely out of ignorance) or have introduced apparently harmless features with unexpected results. Even the adoption of SQL as a standard relational language has impaired the progress of the relational model. The results of these unfortunate events won't be corrected without considerable effort.

Codd has begun his personal attempt at this corrective effort with the publication of his book. By carefully (and at times pedantically) clarifying the features that he considers essential to the relational model, he attempts to explain to users and vendors alike what is required for version 2 (RM/V2). At the same time, he tries not to introduce more than what is useful at this stage of the industry's maturity. Additional features will be introduced in versions RM/V3, RM/V4, RM/V5, and so on, and perhaps through additional books or publications.

Most readers familiar with RM/V1 are likely to associate it with the 12 rules that are often referred to as Codd's 12 Rules. First published in a two-part article entitled, "How Relational Is Your Database Management System?" (*Computerworld*, Oct. 14 and 21, 1985), these *rules* (they should be called objectives) describe the relational model at a fairly high level of abstraction and aren't independent of each other. They're deceptively simple and have been incorrectly applied, engendering many false claims of full compliance with the relational model.

Where RM/V1 focused on logical features from the users' point of view and didn't address implementation specifics, RM/V2 deals with issues such as administration, authorization, naming, DBMS design, protection, data types, and operators for data manipulation. Many of these details aren't usually considered a part of the logical model. Apparently, Codd feels this departure from logical to implementation details is justified in order to preclude implementations that "block extensions needed to advance the DBMS." Unfortunately, the distinction between prescriptions of this nature and essential, logical features of RM/V2 aren't made in the text.

RM/V2 introduces 333 features. These features are divided into 18 classes according to the issues they address (structural, operator, integrity, authorization, and so forth). Within these 333 features are embedded restatements of the 12 rules of RM/V1. Table 1 lists the correspondences between each of the 1985 rules and one or more RM/V2 features. RM/V1 is essentially contained within RM/V2. (Note: Although such a table appears in the book as Appendix A, section 7, there are two errors, and the page numbers are not given.)

Codd attempts to explain what is required for RM/V2

Like the 12 rules of RM/V1, neither the classes nor the features of RM/V2 constitute a minimal, non-redundant set. Thus, completely failing to support any one feature is likely to indicate only partial support for some other feature. Beyond restating the 12 rules of RM/V1, Codd uses the new features to:

□ Expound what is already implicit in the logical relational model

□ Prescribe a number of implementation details, some of which are "under the covers"

□ Introduce ideas not readily found elsewhere, even if published for the most part

□ Provide a forum for his concerns.

RM/V2 adds features that deal with missing data, integrity constraints, updating views, some principles of DBMS product and relational language design, the catalog, distributed database management, and some fundamental laws on which the relational model was implicitly (perhaps in retrospect) based. In addition to discussing these 333 features, the book also contains chapters concerning omissions in current products and their correction, flaws in SQL, extending the relational model, and a rebuttal of various proposed alternatives to the relational model. This last item is particularly interesting. Codd briefly examines and gives reasons for not supporting entity-relationship and object-oriented approaches as well as the binary relation and universal relation approach.

ALL IN ALL, A GREAT deal of this book is either already part of the industry's technical folklore or has been published previously by Codd. Even though Codd takes responsibility for the material in the book, this doesn't imply that topics themselves are treated here for the first time. Many of the issues addressed have been treated by others elsewhere in the literature. This is particularly true of naming (Chapter 6) and distributed database management (Chapters 24 and 25). Unfortunately, Codd doesn't provide references to these other treatments, nor does he contrast them with his own approaches.

As noted earlier, Codd has chosen to present the relational model in a series of versions, each

1985 Rule	RM/V2 Feature(s)	Name	Page
1	RS-1	Information rule	30
2	RM-1	Guaranteed access	229
3	RS-13, RM-11	Missing information	39, 236
4	RC-1	Active catalog	278
5	RM-3	Comprehensive data sublanguage	231
6	RV-6	View updatability	290
7	RM-4	High level language	231
8	RP-1	Physical data independence	345
9	RP-2	Logical data independence	346
10	RP-3	Integrity independence	347
11	RP-4	Distribution independence	347
12	RI-16	Non-subversion	252

TABLE 1. *Codd's 12 Rules in RM/V2.*

of which builds on the previous one. In retrospect, RM/V1 is considered to have been presented in papers published before 1979. RM/V2 builds on RM/V1 and is the first in the new series of versions. As he explains it, this series of versions is an attempt to make the features of the model more accessible to vendors and users alike.

I've heard vendors object that this process doesn't allow them to develop a fully relational product and forces them to address a moving target; this is an unfortunate perception. The relational model provides a foundation for dealing with logical and semantic concepts that Codd refers to as the *logical level*. Its implementation concerns both psychological concepts (the user's level), and storage-oriented and access-method concepts (the physical level). Implementation technology continues to be elucidated and, as with any technology, enhanced. Nevertheless, the more formal mathematical foundation described in the earlier papers provides a means to test implementation details and proposed enhancements for consistency. The failure of vendors to study this mathematical foundation and treat its impact seriously has led to the need for a detailed feature-by-feature exposition of the relational model.

UNFORTUNATELY, the book's presentation isn't easy to follow. For example, the "20 fundamental laws of database management" are essential to motivate a lot of the book's material, evidenced by a diagram of the laws on the inside front cover of the book. But the 20 laws aren't introduced until Chapter 29. As another example, the MAYBE qualifier appears in Chapter 5 in the context of outer join but isn't formally introduced until Chapter 10.

Similarly, terms and acronyms are often used before they're defined. This is especially hard on the reader when the term is associated with a technical concept. For example, when Codd discusses the timing assigned to integrity constraints, the types are used a full page before they're defined in passing.

The book isn't intended as a tutorial on the relational model

Of course, part of the problem is that the book isn't intended as a tutorial on the relational model. According to Codd, it's intended to "challenge vendors to get the job done." In some respects, it more closely resembles a narrative collection of technical notes for reference use. Another contributing factor to the problems outlined here is the degree to which the various features of the relational model (both versions 1 and 2) are intricately intertwined. As Codd points out on several occasions, these features are neither orthogonal nor a minimal set.

Much of the book is written in language accessible to the average technical reader of a magazine like *DATABASE PROGRAMMING & DESIGN*. However, numerous passages don't meet consistently with this analysis. Codd assumes that the reader has knowledge of the basic terminology of the relational model and of database management in general, and sometimes of specialized topics in the field. The reader is also expected to understand something about set theory and logic. Vendors new to or confused by the relational model can't be expected to have such expertise any more than the average user.

This brings me to my most critical comments regarding the book. Because the level of technical expertise expected of the reader isn't well-defined, it's likely that many readers won't recognize when an issue is controversial or an exposition is simply wrong. Even worse, they're not informed that they need additional background to make these judgments.

From time to time, Codd explains in painstaking detail the simplest relational concepts and give examples to further elucidate them. Similarly, nontechnical, unclear, and even arguably incorrect definitions of terms such as *transaction* and *dynamically* are sometimes given. At other times, the level of discussion is quite technical and requires effort to follow (especially when he discusses new relational operators).

Sometimes the reader is clearly expected to be a vendor developing a relational product. Codd discusses features that should never be exposed to the user, such as the semi-join. He gives numerous examples of features that are implementation details and should be "under the covers." It would've been helpful if these features had been clearly noted in the text to distinguish them from the "logical level" that is the relational model itself.

Once in a while, there's a contradiction. For example, on page 247, Codd states that an integrity constraint of type E (for entity) always requires timing of type TC. On page 248, however, he says timing of type TT for constraints of type E are "probably quite rare."

A number of disturbing definitions are included in the book, which I hope Codd will explain or clarify elsewhere. The notion of composite domains, for example, introduces a dependence on order of the component domains that heretofore has been deprecated in the relational model. Following this idea, composite columns are then defined. The meaning of LESS THAN between two composite columns C and D, consisting of C1, C2, and C3, and of D1, D2, and D3, respectively, is defined as a sequential comparison of C1 and D1, then C2 and D2, and finally C3 and D3. The first of these to fail supposedly causes the entire test to fail. The wording of this section presents the definition as a prescription, but it can't possibly be appropriate in the general case of comparing two ordered composite columns. This is obvious to anyone familiar with vectors, for example.

At times, similar phrases are interchanged; for example, *primary domain* is used on page 48 where *primary key* was probably intended or at least would have been a better example. Terminology is occasionally introduced without ever being used in context. New terminology is used where a more common industry term or phrase would have been clearer and made the text more approachable. For example, *comparing term* is used in place of comparison predicate, and the new term *mark* is introduced for null.

IN ALL FAIRNESS, some errors in the text have been detected by Codd, who recently sent me a copy of his April and May errata sheets, which he had already sent to his publisher. A reference to "roughly 40 features listed in Appendix A" required for a DBMS to be called relational in the early 1990s is unlocatable. According to the errata sheet, this statement should refer to "67 features" in Appendix A sections 4 and 5 that are required for a DBMS to be called "adequately" relational. The appendix section labeled "A.5 Investment Protection" has subheadings of "DBMS Design" and "Language Design." Section A.5 should be titled "Additional Features for 'adequately relational', 1990-1994" and should combine A.5 and A.6 with Functions, Investment Protection, DBMS Design, and Language Design as subheadings. This causes a reference on page 500, line 10 to be changed from Appendix A.7 to A.6. Codd is likely to make additional corrections and clarifications over time. Hopefully, these will appear in the next printing or next edition of the book. In the meantime, you should read the book with concern for such errors.

These errors should have been detected by a thorough technical editing. It's possible that all the problems I've discussed are a function of this more general problem—perhaps Codd can explain them to the satisfaction of most readers. When I encountered new, undefined, or poorly defined terms, I found I could comprehend Codd's overall thrust by simply plunging onward. This allowed me to forgive many editorial errors and omissions.

Without a willingness to read on or use supplementary references, it's likely that most readers will have difficulty. This book isn't for the faint of heart or those uninitiated to the relational model. The level of difficulty is uneven, so some readers new to the relational model will probably find sections that are easy to absorb, while some old-timers will find passages that are obscure.

The point of view Codd expresses in his book is worth the considerable effort to understand. His past contributions give him the benefit of the doubt with regard to any errors or lack of clarity found in the text. Because this book makes a great deal of Codd's contributions generally available, it's extremely important. But it remains to be seen whether the book's effect on the industry will be as significant as his earlier contributions.

David McGoveran is the president of Alternative Technologies, a Santa Cruz, Calif.-based consulting firm that provides relational DBMS applications design and development services.