32 A Relational Model of Large Shared Data Banks (1970)

Edgar F. Codd

It is common in academic circles to think that the science of computing is about algorithms. But in business, computing has always been about data. Of course the two perspectives are not opposed, but the world looks very different from a data-oriented view than from an algorithm-oriented view. A database scientist told me that data was the ocean, deep, eternal, and mysterious, and algorithms were just boats skimming its surface.

Until the 1960s, most practical applications of computers were for computing numbers—either values of mathematical functions (such as the tables generated by Aiken's Mark I) or parameters of physical phenomena (astronomical calculations, for example, or the predicted behavior of an atom bomb). Of course any calculation involving physical phenomena requires numerical data, but early computers did not have enough storage to process large amounts of it. The scientific interest in computing was driven by the need to perform numerical calculations, plus the extraordinary discovery by Church and Turing of the ontology of algorithms. Turing's remarkable codebreaking work at Bletchley Park was an exercise of logic and carefully controlled combinatorial search driven by very small amounts of data (intercepted encrypted messages).

Certainly Vannevar Bush had foreseen the importance of large-scale data manipulation ("Selection devices ... will soon be speeded up from their present rate of reviewing data at a few hundred a minute," page 115). Howard Aiken and Grace Hopper anticipated the importance of business applications, and the International Business Machines Corporation, which had grown out of companies supplying tabulating machines for the U.S. Census, came to dominate the business market for computers. Starting in the mid-1960s, IBM developed a database product to manage inventory for the Apollo space program. The data model of this system, known originally as IMS, drew ultimately on graph theory: mechanical parts had sub-parts, and the same sub-parts might be used in several different larger assemblies, so the connection between data entities resembled the links connecting nodes of a directed graph.

The quantity of data managed by such systems steadily grew, and it became clear that the way data were described and queried need not correspond to the memory structures used to store it. In much the same way as compilers had made it possible to abstract the statement of algorithms from the particulars of the machine code that executed them, there was a need to talk about data with some formal clarity, leaving to computer systems themselves the problem of optimally organizing the data in physical devices for most effective storage and retrieval.

Reprinted from Codd (1970), with permission from the Association for Computing Machinery.

For Edgar Frank Codd (1923–2003), the way to talk about data was to use a language derived from the predicate calculus. Data was to be organized as relations. A relation is a set of n-tuples, for example a set of ordered triples or ordered quadruples, where each position or "column" contains data of a particular type. The relation can be depicted as a table, with one n-tuple per row, but the order of the rows in the table is semantically irrelevant because the n-tuples are logically a set.

In this seminal paper Codd worked out the basics of the relational view of data, and most importantly, developed a relational algebra for combining relations. Codd spent much of his career with IBM, and this work emerged from his frustration that existing database systems entangled the database's logical structure (and thus the logic of programs accessing the database) with its internal "physical" representation. IBM welcomed Codd's innovations only tepidly, perhaps because a successful relational database system would compete with existing IBM products, so Codd left to start his own firm. Research implementations of the relational database model began to appear later in the 1970s, with IBM's System R and the Ingres system of Michael Stonebraker at Berkeley. Oracle Corporation, Tandem Computers, Stonebraker's Relational Technology Inc., and IBM all released commercial implementations between 1979 and 1981, establishing the model as a *de facto* standard. The model and the associated data management language SQL are now ubiquitous, and Codd was recognized with the Turing award for his contribution.

32.1 Relational Model and Normal Form

THIS paper is concerned with the application of elementary relation theory to systems which provide shared access to large banks of formatted data. Except for a paper by Childs (1968), the principal application of relations to data systems has been to deductive question-answering systems. Levien and Maron (1967) provide numerous references to work in this area.

In contrast, the problems treated here are those of data *independence*—the independence of application programs and terminal activities from growth in data types and changes in data representation—and certain kinds of data *inconsistency* which are expected to become troublesome even in nondeductive systems.

The relational view (or model) of data described in §32.1 appears to be superior in several respects to the graph or network model (Bachman, 1965; McGee, 1969) presently in vogue for noninferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in §32.2. The network model, on

Ideas That Created the Future

Classic Papers of Computer Science

edited by Harry R. Lewis

The MIT Press
Cambridge, Massachusetts
London, England

This book was set in New Times Roman by the editor using LaTeX. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Names: Lewis, Harry R., editor.
Title: Ideas that created the future : classic papers of computer science / edited by Harry R. Lewis.
Description: Cambridge, Massachusetts : The MIT Press, [2021] | Includes bibliographical references and index.
Identifiers: LCCN 2020018950 | ISBN 9780262045308 (paperback)
Subjects: LCSH: Computer science.
Classification: LCC QA76 .I34 2020 | DDC 004–dc23
LC record available at https://lccn.loc.gov/2020018950

10 9 8 7 6 5 4 3

To those c
Phil Bridg
Sheila Gr