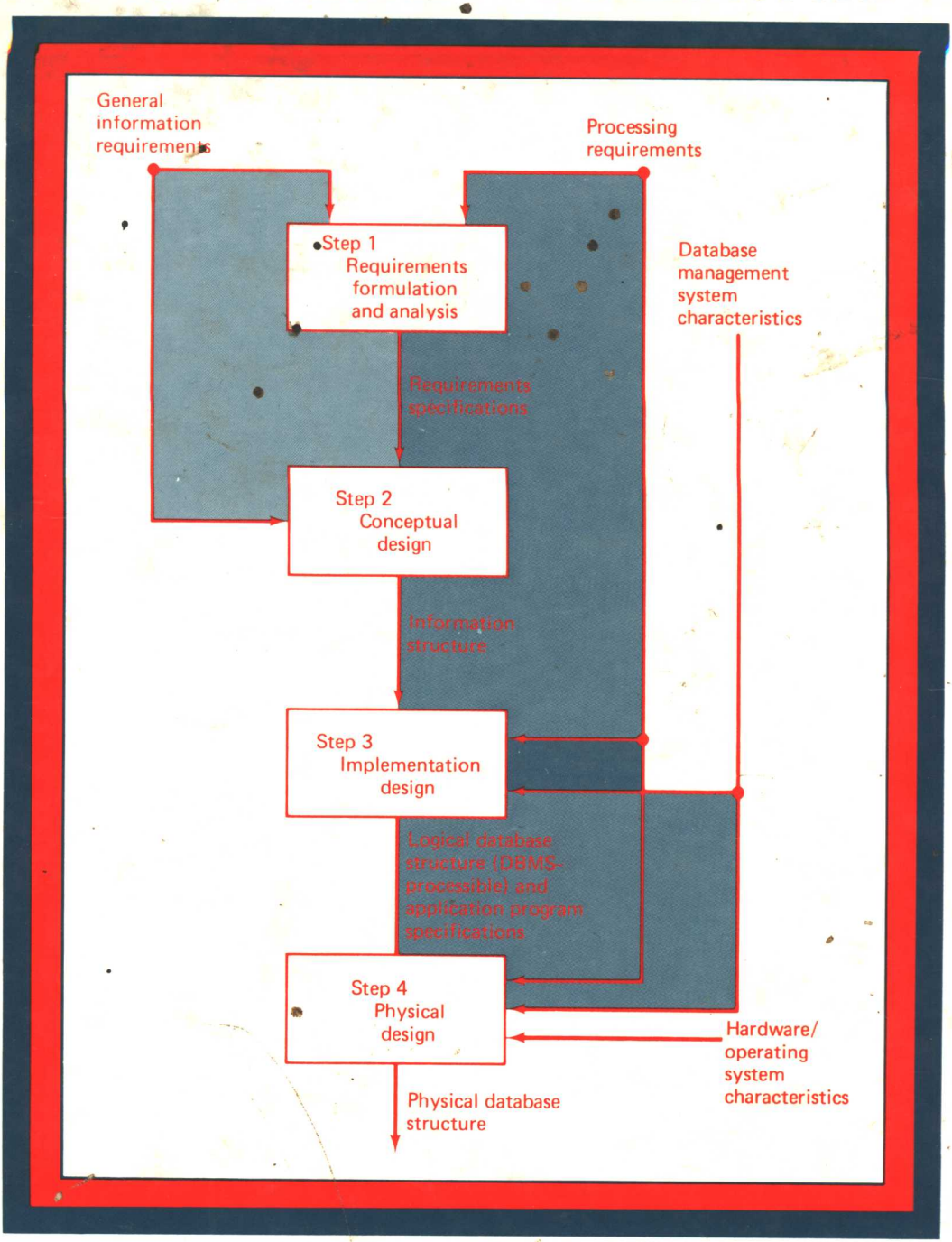
DESIGN OF DATABASE STRUCTURES

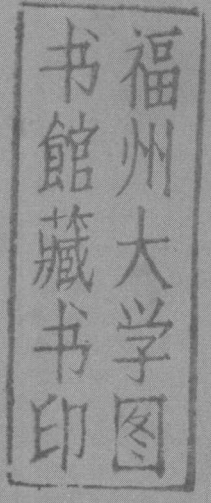
TOBY J. TEOREY ▪ JAMES P. FRY



DESIGN OF DATABASE STRUCTURES

TOBY J. TEOREY
JAMES P. FRY

The University of Michigan



PRENTICE-HALL, INC., Englewood Cliffs, N.J. 07632

Library of Congress Cataloging in Publication Data

Teorey, Toby J.
Design of database structures.

(Prentice-Hall software series)
Bibliography: p. 470
Includes index.
1. Database management. 2. System design. I. Fry.
James P. II. Title. III. Series
QA76.9.D3T43 001.64 81-23375
ISBN 0-13-200097-0 AACR2

Editorial/production supervision
and interior design by *Linda Mihatov Paskiet*
Cover designer: *Frederick Charles, Ltd.*
Manufacturing buyer: *Gordon Osbourne*

Prentice-Hall Software Series
Brian W. Kernighan, advisor

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-200097-0

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall of Canada, Ltd., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

PREFACE

The purpose of this book is to establish a consistent framework for multilevel database design; to define a workable methodology; and to describe a set of general principles, tools, and techniques for database design at each level. The methodology uses the top-down (and iterative) design path, providing for evaluation at many points, allowing for redesign when necessary, and using the same basic terminology throughout. The goal of the design process itself is to formulate a database structure that accurately represents the real-world environment it serves, and one that can be efficiently implemented on an existing hardware/software system. System independence, however, is maintained as long as possible in the design process (i.e., the conceptual design step).

The methodology is illustrated throughout the text using detailed examples from statements of database system requirements through specification of logical and physical database structures that satisfy those requirements. The approach is applicable to network, relational, and hierarchical database systems.

Many steps in the database design process can be assisted with proper use of computer modeling techniques and other tools such as requirements analysis software. These tools range from documentation mechanisms to sophisticated design methodologies. Techniques such as simulation, graph theory, or mathematical optimization are commonly used, although simple expected-value estimates of performance have been found practical at various stages of database design. Categorized references are provided for the most appropriate tools at each design step.

Although emphasis is placed on the design of the database structure, its relationship to efficient program execution is also explored. Constraints for issues such as integrity, recovery, and security are discussed in terms of trade-offs with efficiency goals.

This book is for the professional database analyst, designer, database administrator, and application programmer. It could also be used for a one-semester senior or first-year graduate course in database design. The material is presented tutorially, and

is supplemented with exercises and a bibliography, glossary of terms, index, and summary of variables. The level of mathematics is college algebra with a knowledge of some basic calculus. The reader is assumed to be familiar with data structures, searching and sorting techniques, and characteristics of the major database management systems (DBMSs). A course based on this text would naturally follow an introductory course on database management systems. Both theory and practice are presented, and their interaction is strongly emphasized.

The order of the chapters follows the database system life cycle steps (Chapter 1) very closely and presents the material in the order that a normal design process would follow. Part I (Chapters 1 to 3) is common to an understanding of all chapters, but thereafter each of the design phases is a complete entity and may be studied in isolation. For instance, a short course in physical design would follow Chapters 1 to 3 and 9 to 16. Chapter 18 presents an overview of the database design issues in distributed systems for the purpose of defining problems for future research.

Requirements analysis and conceptual design are illustrated with a common example (Chapters 3 and 6). Implementation design has a separate complete example (Chapter 8), and the physical design discussion contains smaller individual examples to demonstrate specific points. The examples serve to show that database design can be performed manually, but they also point out where computer-based tools can be effectively implemented.

Ann Arbor

Toby J. Teorey

James P. Fry

ACKNOWLEDGMENTS

We would like to express our gratitude for the various examples and manuscript critiques made by Larry Brown, Janis Bubenko, Bob Curtice, Jeffrey Hoffer, Alan Merten, Shamkant Navathe, Donna Rund, Mario Schkolnick, Dennis Severance, Diane Smith, John Smith, Dick Volz, and the several reviewers. Much of the design framework evolved from a course taught at the University of Michigan Engineering Summer Conference; it was also heavily influenced by the work at the 1978 NYU Symposium on Database Design and the 1978 Database Design Workshop in New Orleans. We acknowledge the contribution to our approach from our colleagues Sakti Ghosh, David Jefferson, Paul Jones, Bob Taylor, Vincent Lum, and Bing Yao.

At the University of Michigan we were assisted with software development, design tool evaluation, and manuscript revisions by the many able staff members of the Information Systems Research Group, including Ed Birss, David Chen, Rick Cobb, K. Sundar Das, Mark Deppe, Don DeSmith, Alberto Garcia, Eric Kintzer, Chris Merrill, Don Novak, Lew Oberlander, Don Swartwout, and Mike Wilens.

We thank Connie Allen, Pam Downie, Carol Dunn, and Izena Goulding for their help in manuscript preparation.

We appreciate the use of the library and technical facilities provided by the Department of Management Information Systems at the University of Arizona under Jay Nunamaker.

Exercises B5, B6, and B7 were contributed by Dennis Severance of the University of Michigan. Exercise B8 was suggested by William Weiler of Blue Cross/Blue Shield of Massachusetts, and Design Problem 3 in Appendix A was designed by Paul Helman and Marilyn Mantei of the University of Michigan. Design Problem 2 was revised from Chen [1978].

Finally, we wish to acknowledge the support and encouragement of our families, including Eunice L. and Thomas F. Teorey, Bessie May and Robert W. Teorey, and Anne J. and Palmer E. Fry.

CONTENTS

I INTRODUCTION

1 DATABASE SYSTEMS

Designing an integrated database is a difficult, time-consuming, and often unstructured process. It is a complex problem that pervades not only the data-processing function but eventually the entire organization. The quality of the resulting database structure is dependent upon the design methodology, the design techniques used in the steps of the methodology, the validity of the information requirements, and the commitment of the organization's developmental and operational resources to the endeavor.

In this chapter we begin to build the foundations for a practical database design methodology by establishing a standard set of terminology to be used throughout the text. We also review some of the basic concepts of data as used in the organization, as specified in application programs and database management systems, and as stored in computer systems.

1-1 DATA AND DATABASE MANAGEMENT

Data, as defined by Webster's is "a fact; something upon which an inference or an intellectual system of any sort is based" [Webster's, 1974]. The primitive components of data are the characters and numbers upon which a natural language is based, or their coded representations in strings of binary bits. A *data item* is the smallest unit of data that has meaning in the real world; it is the smallest named unit of data. A group of related data items treated as a unit by an application program is known as a *record,* and the collection of records of a single type is referred to as a *file*. Files can be manually stored or mechanized (electronically stored) on a computer. The distinction between logical records, as viewed by the applications programmer, and stored records, as viewed by the storage devices and systems programmer/analyst, is emphasized throughout the text.

The remarkable continued growth in the use of computers for a wide variety of

industrial, administrative, and scientific applications has led to the mechanization of extremely large quantities of data. During the late 1950s and early 1960s, businesses, government, and other organizations began to collect and store data on computer-accessible files. As each new need for data or data processing arose, a new file was created to meet that need. Individual groups within organizations developed their own computer applications and collected and maintained any required data in private files. The data files and applications programs were designed for one another, with much of the information being stored implicitly in the relationship of the program to the file.

Organizations eventually developed an awareness of the need for centralized management of data and applications. This awareness developed in several ways. Primarily, higher-level managers discovered that the information required to support their decision making was not easily obtainable. To fulfill a request for information, an application program had to be written to access several private files, with each file using its own format. The manager often canceled the request, either because the information would no longer be useful by the time it could be obtained, or because the information was not worth the cost involved. Second, decision making was hampered by the lack of data integrity. Computer reports conflicted because logically identical data items had different values. This was caused by data duplication among the private files, combined with inconsistent updating policies. There was also duplication of effort in the collection of data and the production of reports. Computer resources, both storage and processing, were being wasted. Finally, technology improved to the point where it became feasible to design, build, and operate large-scale collections of data in a computer environment. In summary, organizations realized that data were a valuable resource and needed to be centrally managed.

The concept of a database has thus emerged fully only in recent years. A *database* can be defined as a computerized collection of stored operational data that serves the needs of multiple users within one or more organizations. A key point is that the database is an integrated resource to be used by all members of the organizations who need information contained in it. Information is no longer implicit in the file-program combination but is stored explicitly in the database, which may include many different record types. The database is not single-program oriented as were private data files, but has an integrated requirements orientation.

Databases support managerial decision making in the following ways:

1. *Speed.* The computer system allows on-line querying for information.
2. *Total availability.* All the information contained within a database is available for use.
3. *Flexibility.* Previously unanswerable questions become answerable, and changes become relatively easy to implement.
4. *Integrity.* Data duplication is reduced and updating policies can be standardized, resulting in consistent data.

Simply having a database does not entirely solve the organization's data-processing and decision-making needs, however. Since the database is an integrated

resource for multiple users within an organization, it should be managed for the organization's benefit and from its viewpoint, not by individual users. Users naturally tend to develop applications for their own purposes, unconcerned about the impact that the new applications will have on other users. Without centralized management, the usefulness of a database declines over time.

Two additional concepts have been developed to solve the problem of controlling and managing the organization's database resource. Initially, software was developed to provide a common interface between all users and the integrated database. A common interface promotes privacy and data integrity. Also, users cannot store information implicitly and must use and modify data in a manner consistent with the organization's viewpoint. The software known as a database management system allows computer control of the data resource. A *database management system* (DBMS) is a generalized tool for manipulating a database; it is made available through special software for the interrogation, maintenance, and analysis of data. Its interfaces generally provide a broad range of languages to aid a wide variety of users. It also provides a convenient framework in which databases can be designed and used. A DBMS can be obtained by an organization through either a software vendor or in-house development.

The second concept is that of the *database administrator* (DBA). The DBA can be thought of as one or more individuals, possibly aided by a staff, who manage the organization's database resource. The DBA should be a dynamic and capable individual, primarily management oriented but also with a technical background. This person must be able to communicate with both upper-level management and data-processing users as well as manage the staff of technical specialists. The staff should include individuals experienced in a variety of areas, such as DBMS software, operating systems, computer hardware, applications programming, and system design. It is important that the staff also include individuals with a knowledge of the organization and its information requirements. The DBA's staff must be able to relate well to groups outside the data-processing department.

The DBA position was created when organizations recognized the need for centralized control of the data resource, data processing, and all other aspects that concern a database. The user community and individual users must be served as fairly as possible while keeping in mind the objectives of the organization as a whole. Thus, the DBA is responsible for determining all users' needs, designing a database, implementing the database, modifying or converting the database as needed, and assisting users with documentation and through education [Canning, 1972; DeBlasis and Johnson, 1977; Kent, 1978; Lyon, 1976a].

1-2 LEVELS OF DATA REPRESENTATION

At least three levels of abstraction are now recognized to exist in the specification of a database structure: the conceptual or enterprise administrator view, the implementation view of the applications programmer or end user, and the physical view of the