Database Design

Second Edition

Gio Wiederhold

Database Design

Second Edition

Gio Wiederhold
Assistant Professor of Computer Science
Stanford University

McGraw-Hill Book Company

New York St. Louis San Francisco Auckland Bogotá Hamburg
Johannesburg London Madrid Mexico Montreal New Delhi
Panama Paris São Paulo Singapore Sydney Tokyo Toronto

This book was set in Computer Modern Roman.
The editors were Eric M. Munson and Joseph F. Murphy;
the production supervisor was Joe Campanella.
New drawings were done by J & R Services, Inc.
The cover was designed by Jane Moorman.
Halliday Lithograph Corporation was printer and binder.

DATABASE DESIGN

1 2 3 4 5 6 7 8 9 0 HALHAL 8 9 8 7 6 5 4 3

ISBN 0-07-070132-6

Library of Congress Cataloging in Publication Data

Wiederhold, Gio.
Database design

(McGraw-Hill computer science series)
Bibliography: p.
Includes index.
1. Data base management. 2. File organization (Computer science) 3. Data structures (Computer science)
I. Title. II. Series.
QA76.9.D3W53 1983 001.64'25 82-20319
ISBN 0-07-070132-6

Preface

Origin

The material in this book is the result of courses given at Stanford University on Database and File Structures since 1971. Initially little coherent published material was available, even though a large number of references could be cited. In particular, no clear definition of the concept of a *schema* was available. Now many practical and scholarly sources for material exist but problems of emphasis and analysis remain.

This book brings together knowledge in the area of database management in a structured form suitable for teaching and reference. The first edition has found broad acceptance in course sequences where quantitative approaches are stressed. Unintended, but gratifying, is the place this book has found as a programmer's reference and analysis guide.

Analyses to predict logical correctness and adequate performance of systems prior to investment in an implementation effort should be expected from professional system designers. An analysis of system development methods assigns a cost ratio of 100 to 1 when errors are found in testing rather than caught in the design stage. Many more cross-references have been added and the extensive index has been further expanded to help the professional reader. The background sections have also been completely redone to reflect the best recent references. Other major changes made are discussed in the objective section.

Much has happened in the database field since the first edition was published. The area has become an accepted academic discipline. Conferences and journals covering the area abound. Selection of material is now more difficult than finding material. At the same time commercial interest in database management has exploded creating a serious shortage of qualified people. Already now the number of professionals working in the database area exceeds the number employed in traditional computing fields as compilers and operating systems. Not everyone working today with databases has the background to deal well with the complexities that can arise, so that complaints of system inadequacies are common.

Much of the literature remains descriptive or presents one-sided experiences, inadequate to provide a basis for the transfer of concepts to diverse applications. An effort has been made here to develop concepts from the wealth of material and to present the subject in such a way that the concepts which evolve can be applied in practice. An engineering attitude to the problems of database organization has been used in order to combine formality with applicability.

I hope that this greatly revised text will continue to fill the needs and that it will help extend and improve the teaching of the data-processing aspects of computer science.

Objective

This book is intended to present the methods, the criteria for choices between alternatives, and the principles and concepts, that are relevant to the practice of database design. No actual systems are described completely, nor are systems surveyed and compared, although specific and realistic examples are used throughout to illustrate points being made. The material provides the basis to allow the reader to understand, recognize the implications, and evaluate database approaches. Databases in this sense are a broader concept than database management systems. The design of a database involves understanding the meaning of the data, the systems chosen, be they database management systems or traditional file systems, just help in the implementation.

This book includes two major parts:

1 The description and analysis of file systems (Chaps. 2 to 6)

2 The description and analysis of database systems (Chaps. 7 to 10)

The first part is intended to provide a solid foundation for the latter part, since the issues arising in database design are difficult to discuss if file-design concepts are not available to draw upon. A number of subjects which pertain to both files and databases, namely reliability, protection of privacy, integrity, and coding are presented in the third part, consisting of Chaps. 11 to 14. If the material is taught in two distinct courses, the third part should not be ignored in either course.

The audience for this book ranges from students of computing who have finished a reasonably complete course in programming to applications and systems programmers who wish to synthesize their experiences into a more formal structure. The material covered should be known by systems designers or systems analysts faced with implementation choices. It probably presents too much detail to be of interest to management outside the database management area itself.

The revision of this book has been major. Nearly every paragraph has been rewritten, and some sections have been completely replaced. The number of tables and examples is also increased. The general outline, however, and the underlying principles could remain the same. Distribution of databases over multiple sites is now considered throughout the book. The equipment table in Chap. 2 deals with the much wider range of storage devices now available. The prevalence of B-trees for indexes has permitted a rewrite of Sec. 3-4 which is more modern and simpler. A new method for dealing with growth of direct-access areas is part of Sec. 3-5. New results dealing with device interference are presented in Chap. 5.

Chapter 7 includes the modern concepts of semantic modeling. A new section, 7-2 deals with the formal semantics now available for database design. Section 7-3 defines the conceptual tools for establishing the structural relationships between files. The entire design process is described step by step in Sec. 7-5. The introduction of commercial relational database implementations permits now a consistent description of these systems in Sec. 9-2. The performance of databases using relational operations can be predicted using the information introduced in Sec. 9-3.

The concept of using transactions to access the database is used throughout but has a major impact on the handling of reliability issues in Chap. 11 and integrity maintenance in Chap. 13. New sections have been added there.

Design Methodology

This book presents a comprehensive collection of database design tools. In order to employ them, a strategy of problem decomposition, followed by a structured design process is advised. A top-down design requires the underlying primitives to be understood. Since this book starts with the basics, the design process is initiated with concepts from Chap. 10.

The categorization of database approaches given in Chap. 10 helps to set the initial objective for a database. Chapter 7 provides the means to construct a model which integrates the requirements of multiple applications which share the database. The schemas in Chap. 8 provide methods to describe the model in machine-readable form. Existing database systems, described in Chap. 9 and referenced in Appendix B, suggest available implementation alternatives.

If the database is to be directly supported by a file system, the basic file choices in Chap. 3 and their combinations shown in Chap. 4 provide the alternatives. The data representation can be chosen using the material of Chap. 14.

The performance of the chosen approach can be predicted following the outline shown in Chap. 5. Factors relevant to specific database systems or file systems appear where they are discussed, but the terminology and variable definitions are consistent throughout, so that cross-referencing is simple. The structural model defined in Chap. 7 provides the framework for the translation of application loads to the load to be handled by the database. Transaction performance in database systems is estimated using the performance analyses for the prevalent approaches from Chap. 9. An optimal file design may be selected after application of the load parameters to the performance formulas from Chaps. 3 and 4. The formulas also require the hardware description parameters introduced in Chap. 2.

Problems of reliability, protection, and integrity (Chaps. 11, 12, and 13) require a close scrutiny of the available operating system. The long-term maintenance of a database is guided by considerations presented in Chap. 15.

Curricula

Modern curricula give increased emphasis to files and databases. This book matches the suggested courses well. The material of Chaps. 1 to 6 covers course CS-5 (Introduction to File Processing) and Chaps. 7 to 14 cover course CS-11 (Database Management System Design) as specified in the report by the ACM Curriculum on Computer Science [Austing et al[78]]. The quantitative approach taken in this book causes that algorithmic and analytic material assigned to courses CS-6 and CS-7 is included as well, albeit limited by relevance to databases. I do not agree with the recommendation that students write a database management system during CS-11; this is apt to be an exercise in trivializing real issues. Design and performance prediction of a nontrivial database application is part of the course taught at Stanford and enables a broad exposure to important concepts beyond programming. I agree with Ralston and Shaw[80] that mathematical competence is necessary for a computer science student.

The text provides all the material for the file and database subjects of Courses C1, C2, C3, C4, and D2 specified in the Curriculum Recommendations for Graduate Professional Programs in Information Systems [Ashenhurst[72]]. The author feels, however, that these courses are easier to teach using a depth-first approach to the subjects versus the breadth-first approach advocated in the proposal. I have been impressed by constructive comments received from readers who used the book for self-study.

In many schools files and databases are still neglected, possibly because of a shortage of teachers. Students who enter industry or commerce with a bachelor's-level education in computing or data-processing feel this void sharply. It is now reasonable to expect that students majoring in computing and computer applications should be familiar with this subject area [Teichrow[71], Sprowls[75]]. Projections regarding the future use of computers give a considerable weight to the importance of the database area [Dolotta et al[76]], so that we can expect an increasing demand for educational services in this area.

Terminology

We are grateful that the terminology in the area of database and file management is becoming more consistent. Within this book a major effort has been made to define all terms and to use only one term per concept. Some terms were changed since the first edition because usage developed differently than I had foreseen. All terms are listed in the index. In order to aid both experienced readers and users of the references, Appendix A cites common alternate terminology and compares it with the terminology used in this text. The introductory chapter is mainly devoted to definitions. It is assumed that subjects such as programming and basic functions of operating systems are familiar to the reader.

Most of the program examples throughout the text use a simple subset of PL/1. The variable names are chosen so that they will aid in the comprehension of the

programs; they are printed in lowercase. Keywords which are to be recognized by the translating programs appear in uppercase. The programs are designed to be obvious to readers familiar with any procedure-oriented programming language. A number of introductory PL/1 texts [Hume[75], Richardson[75], Mott[72]] can be used to explain features that are not recognized. Some PL/1 texts, unfortunately, omit the statements required for the manipulation of data files.

Most of the examples shown to illustrate features of design approaches or systems are based on actual systems but are of necessity incomplete. An effort has been made to note simplifying assumptions. The same should be done in students' design assignments, so that awareness of real-world complexities is fostered without overwhelming the design with trivia.

Exercises

The exercises listed in each chapter have been kept relatively simple. It is strongly suggested that an analysis of some of the systems described in the referenced literature be used for major assignments. Many of the problem statements in fact require such material. The analysis or comparison of actual systems may seem to be an excessively complex task, but has been shown to be manageable by students when the material of this book has been assimilated. Appendix B provides references to a large number of systems implementations. A knowledge of calculus will be helpful when doing some of the problems, but purely graphical methods will provide adequate results.

The primary exercise when this course is being taught at Stanford is a design project. Early in the course students prepare an objective statement for a database application of interest to them. Some individual research may be needed to obtain estimates of expected data quantities and transaction load frequencies. The students prepare a structural model describing their database while Chap. 7 is being covered. The project is fleshed out with a schema description and a performance prediction of selected important transactions for the application.

References

Source material for this book came from many places and experiences. References are not cited throughout the text since the intent is to produce primarily a text and reference book which integrates the many concepts and ideas available for database design. A background section at the end of every chapter cites the major sources used and indicates further study material. The references have been completely updated for the second edition and provide a generous foothold for students intending to pursue a specific topic in depth. Use of the references can direct research effort toward the many yet unsolved problems in the area.

The bibliography has been selected to include some important material for each of the subject areas introduced. The volume of database publications is now such that a comprehensive bibliography is beyond the scope of a textbook. Only sources that are easy to obtain, such as books and journals, have been chosen. Papers which appear in conference proceedings containing much relevant material are cited only by author and proceedings, and do not appear individually in the bibliography.

Typically only one or two early publications and some instances of recent work in an area are cited. To provide a complete view references in the recent material have to be traced. Trade publications, research reports, theses, and computer manuals are referenced only when used directly, although much relevant information can be found there. Up-to-date information on computer systems is best obtained from manufacturers.

I apologize to the authors of work I failed to reference, either due to application of these rules, or because of lack of awareness on my part. A large annotated bibliography is being maintained by me and has been made available. I look forward to distribution of the bibliography in computer-readable form since it is too large to be effectively scanned without computer assistance.

Acknowledgments

Parts of the first edition were carefully reviewed by John Bolstead, Frank Germano, Lance Hoffman, Eugene Lowenthal, Tim Merrett, Joaquin Miller, Richard Moore, Dick Karpinski, Bernard Pagurek, Donn Parker, Jean Porte, Gerry Purdy, Diane Ramsey-Klee, John Rhodes, Justine Roberts, Diane Rode, Hank Swan, and Steve Weyl. Thomas Martin and the members of the SHARE Database Committee provided many ideas during mutual discussions. The second edition has had comments by innumerable students, as well as by many colleagues. I thank especially Anne Beetem, Bob Blum, Ramez ElMasri, Sheldon Finkelstein, Jonathan King, Joaquin Miller, Toshi Minoura, Witold Litwin, Bob Paige, Domenico Saccà, and Kyu Young Whang for their reviews and corrections.

I received support from the National Library of Medicine during the preparation of the first edition [Wiederhold77]. Experience was obtained in part during evaluations funded by the National Center for Health Services Research. Subsequent support came from the Defense Advanced Research Projects Agency (contract N-39) for research in Knowledge Based Management Systems. This research provided the basis for many of the concepts presented here. I have also benefited from the computer services at Stanford University, some of which are supported by the NIH Division of Research Resources. The TEX program, developed by Donald Knuth79, was used to prepare the plates for printing. Arthur Keller provided the initial typesetting macros and advice. Peter Ullman, Chris Tucci, John and Randy Wiederhold managed the photo-typesetting. The ability to prepare beautiful copy under full control of the author is both an opportunity and a responsibility. I hope to have carried them adequately. Jayne Pickering read the proofs several times to insure that my editing matched my intentions. Any errors in content and format remain my responsibility, and I welcome all kinds of criticism.

This book would not have been written without the inspiration, support, and just plain hard work by my wife, Voy. The appreciation she has received from her students, and users of computer manuals she has written, has encouraged me to attempt to present this material in as straightforward a fashion as she has been able to present PL/1 programming.

Gio Wiederhold

Contents

PART I FILE STRUCTURES AND DESIGN

Chapter 1

Definitions and Introduction to the Subject

The order and connection of ideas is the same as the order and connection of things.

Baruch (Benedict de) Spinoza
Prop. VII from Ethics, Part Two.

When we talk informally about a *database*, we refer to a collection of mutually related data, to the computer hardware that is used to store it, and to the programs used to manipulate it.

By mutually related we mean that the data represents knowledge about a specific enterprise, say a company, a university, or a governmental institution. The data may also be related because it deals with a certain problem area, perhaps about a disease which is of concern to the staff of a number of hospitals. The data should be organized so that it can be processed to yield information.

The organization of the data in a database has to represent the underlying meaning or semantics of the data correctly and efficiently. In conventional programs the structure of data is arranged for the convenience of the program. A database contains data to be used by many and diverse programs. The organization of a database can hence not be solely determined by decisions made while programming specific functions.

This chapter begins by defining the concept of a file, since files are the major physical units into which a database is organized. The second section discusses

operations or tasks performed when using a database. In Sec. 1-3 we develop a classification of data management which will provide the structure for all subsequent material. We then take familiar concepts from programming and relate them to the presented classification. This sequence is intended to provide a link between our existing programming experience and the approach taken in this presentation of database-management methodology. In Sec. 1-8 a list of application areas is given as an aid for the selection of a topic for analysis to be followed throughout the text.

You may have learned nothing new after you have read this chapter, but the framework established will allow us to proceed through the book in an orderly fashion.

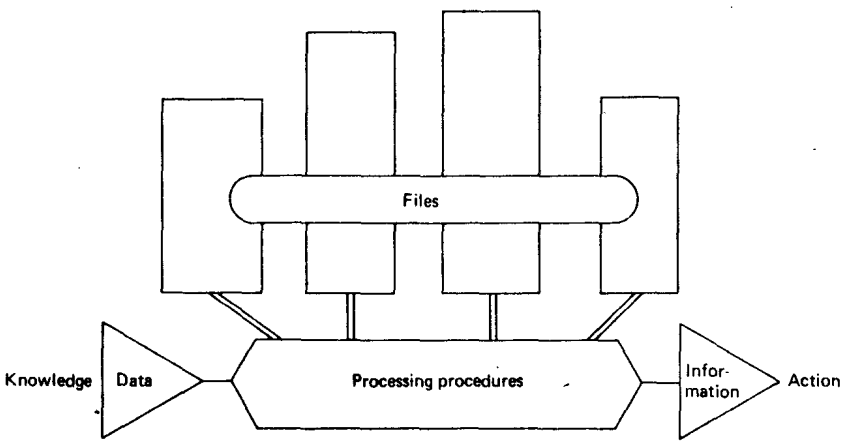


Figure 1-1 A database.

1-1 FILES

A database is a collection of related data. The data storage for a database is accomplished by the use of one or more files. All files of one database are accessible from one computer or from any interconnected computer if the database is distributed over several computers. Procedures in these computers are used to enter data, store data in files, and process the data in response to requests for information. A *file* is defined to be a collection of similar records kept on secondary computer storage devices.

Typical of *secondary storage* devices are disk drives with magnetic disks, but there are a number of alternatives. A *record* is defined at this point to be a collection of related *fields* containing elemental data items. A more formal and detailed definition will be developed in Chap. 3. A data item is typically represented by a value which is part of a description of an object or an event. Computational processes can manipulate such values.

1-1-1 Size

To warrant the attention and the approaches discussed in this book, the database should be reasonably *large*. We will concentrate on topics that are applicable to

large external files. This outlook limits the choice of algorithms that can be applied. Collections of data that can be processed in their entirety in the directly addressable memory of a computer, its primary storage, allow techniques that will not be covered here. The use of the term database also implies that a variety of people are involved. Data entry may be done by people far removed from the users and the data may contain information suitable for several purposes. The quantity of data to be handled may range from moderately large to very large. These measures of size depend on the hardware and on operational constraints which may apply in a given environment.

> *Large* implies a quantity of data which is greater than a single person can handle alone, even when aided by a computer system. The actual quantity will vary depending on the complexity of the data and applications. An example of a large database is the integrated personnel and product data system in a manufacturing company of about 6000 employees, with more than 300,000 records of 21 types [Emerson in Jardine[74]].*
>
> A *very large* database is an essential part of an enterprise and will be in continuous use by many people. At the same time it will require many storage devices. A very large database presents particular problems in its management, since it cannot be turned off without affecting the operation and well-being of the enterprise into which it is integrated. A formal definition of a very large database is one where the time required for making a copy is greater than the permissible interval between update transactions performed to keep the database up to date. An example of a very large database is found at a telephone company with 5 million subscribers [Karsner in Kerr[75]]. Much larger yet are databases at the social security administration and other national systems.

To have a copy of the contents of a file frozen at a certain instant in time is important for many purposes, including periodic analysis of data, backup for reliability purposes, and auditing. To avoid problems, it is best not to permit a file to be modified while it is being copied. In a very large database these two considerations conflict. This definition imposes certain design constraints on the systems which we will be discussing.

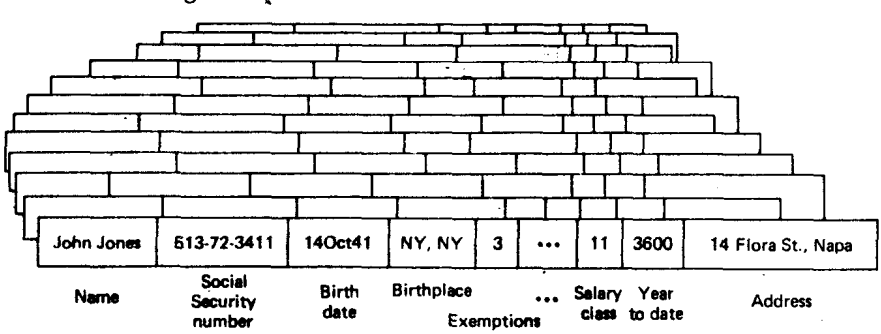


Figure 1-2 A payroll file.

*Superior number next to reference indicates the year of publication.

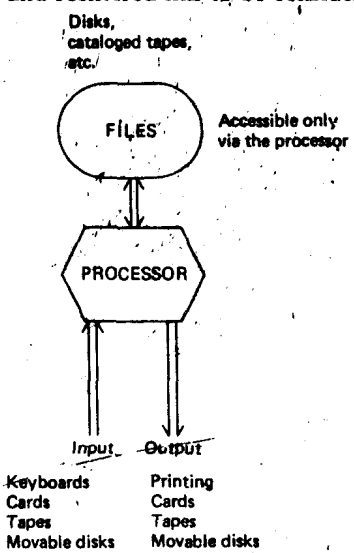
1-1-2 File Organization

Files not only are characterized by their size but are further distinguished by their organization. Differences in the organizations of files lead to great differences in performance when storing and retrieving records. The evaluation and comparison of file organizations is an important aspect of this book.

Six basic file organization types will be analyzed in detail in Chap. 3, and Chap. 4 will show some of the possible combinations and permutations of these basic file types. A database often requires more than one type of file.

1-1-3 Input-Output

When reading or writing files, data is transferred between storage and memory devices of the computer system. When reading input or writing output, data enters or leaves the computer system. A database is concerned with the data which remains within the scope of the system. Data which is written onto tapes, although the tapes are dismounted, securely stored, and later mounted and read again, can be part of the database. Data which is taken out, made accessible for modification, and reentered has to be considered new input, since its consistency is not assured.



Examples of devices used for files are nonremovable disks and drums, removable disks that are normally kept mounted, dismountable master tapes or disks kept at the computer site, archival tapes and disks kept in remote vaults for protection, and sometimes card decks containing master lists that are loaded into the system when needed.

Devices used for input and output are terminals and all those machines that read or write media as cards, paper tape, printed reports, microfilm output, and tapes or disks that are shipped between computer systems.

Figure 1-3 Files versus input-output.

In many computer systems the distinction of files versus input and output is not clearly delineated. The subject of input and output is at best as complex as the database area and will not be covered in this text. We will simply assume that an *operating system* is available which provides adequate input and output capabilities, including access via on-line terminals where appropriate, when we talk about database systems.

We will not discuss file organizations that are based on input and output facilities. These tend to regard data as a continuous stream of characters. *Stream*