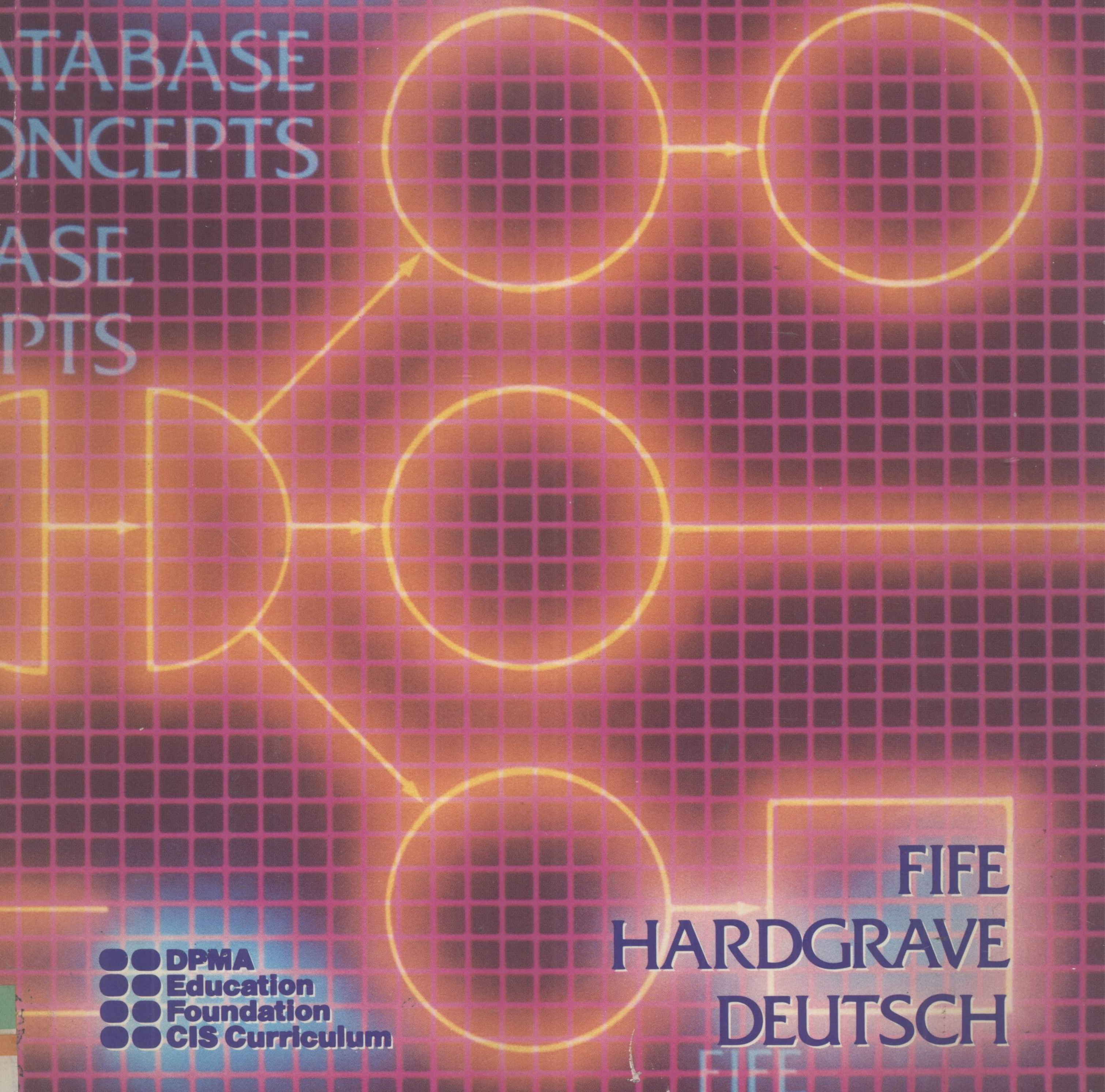
DATABASE CONCEPTS



ATABASE
ONCEPTS
ASE
PTS

DPMA
Education
Foundation
CIS Curriculum

FIFE
HARDGRAVE
DEUTSCH

FIFE
HARDGRAVE

DATABASE CONCEPTS

Dennis W. Fife
Associate Professor
Computer and Information Sciences
George Mason University
Fairfax, Virginia

W. Terry Hardgrave
Director
Product Development
Cucumber Information Systems
Rockville, Maryland

Donald R. Deutsch
Manager
Data Systems Technology
General Electric Information Services Company
Nashville, Tennessee

Published by
J96 **SOUTH-WESTERN PUBLISHING CO.**
CINCINNATI WEST CHICAGO, IL DALLAS PELHAM MANOR, NY LIVERMORE, CA

dBASE III is a registered trademark of Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230. All references to dBASE III in the text of this book are to the trademark of Ashton-Tate.

ISBN: 0-538-10960-2

Library of Congress Catalog Card Number: 84-50212

1 2 3 4 5 6 7 8 9 M 9 8 7 6

Printed in the United States of America

PREFACE

PERSPECTIVE

This book is about understanding and applying database concepts for practical use in information systems. Unlike predecessor texts, this book does not require the student to explore the myriad technical nuances required for the from-scratch design and development of databases.

The world of database implementation does not function that way any more. The 1980s have been marked by a parade of introductions of packaged database management systems (DBMS) of a seemingly infinite variety. In addition, the microcomputer phenomenon has been accompanied by a deluge of user-oriented software packages that have brought end users headlong into a first-hand acquaintanceship with database software. At the end of the 1970s, a major industry question centered on how to get managers and executives to start using keyboards. By the 1980s, questions centered on whether managers and executives were going too far in their hands-on involvement. At this writing, it is certainly safe to say that the advent of database and spreadsheet packages for microcomputers has helped to reverse the situation in which database capabilities and programs were distant mysteries to the management group who represented its most likely users.

There are DBMS packages that apply a number of different data models and use differing schema to define those models. There are also DBMS packages designed for a wide range of sizes and capacities of data storage. In short, database utilization currently and for the foreseeable future will involve a degree of sophistication needed to understand and evaluate data models and schema for their use. With the selection of DBMS packages currently available, it is no longer feasible, except in relatively few special situations, for a computer-using organization to develop its own database software from scratch. Therefore, it is no longer necessary for students who aspire to professionalism in information systems application to assume the burdens of expertise in from-scratch database development.

This text, accordingly, complies with the current realities of the database area of the computer information systems field. That is, the text is organized to impart the level of knowledge necessary for database software selection. Students also acquire the knowledge base they will need for on-the-job evaluation and implementation of DBMS packages. The content of the book is organized to provide a series of stepping stones leading toward this resulting knowledge base.

PEDAGOGICAL VALUE

This text has been developed specifically to meet the requirements of the CIS-6 course, entitled *Database Program Development,* within the *Data Processing Management Association (DPMA) Model Curriculum for Undergraduate Computer Information Systems Education.* The manuscript was developed under cognizance of the DPMA and has been endorsed by DPMA for compliance with the curriculum and for technical validity.

Within the context of the DPMA curriculum, CIS-6 is part of the series of core courses and is designed as an initial introduction to database principles. The course is intended for upper-division undergraduates who have had previous training in programming and systems development.

Though compliance with the DPMA Model Curriculum was a given target in the preparation of the manuscript for this book, the CIS-6 course outline has not been regarded as a constraint. The curriculum itself is sufficiently broad and general to have facilitated innovation that makes this work applicable to a broad segment of the computer-education marketplace.

As one example, the text is programming-language independent. By design, the content is applicable equally to students whose previous training has been in COBOL, Pascal, or in any other language. Similarly, the text is appropriate as a guideline for systems development projects under traditional life-cycle techniques and also for use of information center or packaged implementation methodologies. One of the authors has used the materials in this text with marked success in a course with mixed enrollment involving both CIS and computer science students.

In short, the concepts and principles of content organization for this book are universal. The text can be used as first exposure for advanced students and, with equal appropriateness, can be valuable for students whose previous computer-related education is limited to one semester of instruction in programming.

CONTENT REVIEW

Database Concepts is organized into four parts and 15 chapters that form a learning progression running from a basic indoctrination into needs and concepts through to a look at the future of databases. The content elements of the text are described, briefly and in sequence, in the presentations that follow.

I. Introduction

There are two introductory chapters. The first traces the evolution and roles of database concepts and techniques from the early days of the computer era to the present. Emphasis is given to the ability of databases, because they organize and deal with data at primitive levels, to provide support for multiple applications. The increased productivity of systems analysis and programmers in developing new applications within a database environment is covered. Other benefits of a database approach to an organization also are stressed. These include data integrity, security, relationships, sharing, and decision making support.

II. Data Models

A series of six chapters is devoted to concepts of design and data modeling. The sequence begins with the setting up of a case scenario that is traced through the remainder of the book to demonstrate the evaluation and selection of DBMS tools. In Chapter 3, the case of Composite National Bank is established. The bank, it is explained, plans to incorporate all of its customer information in an integrated database, doing away with a set of separate files. Data structures for the bank's system are established.

From this starting point, the succeeding chapters review the principles of data models and the three major types of models: relational, hierarchical, and network. These discussions set the stage for a chapter covering the principles of data model selection.

III. Physical Aspects of Databases

At the outset of this series of chapters, it is explained that the data model, though it is critical, represents only one factor in the selection of a DBMS package. This group of chapters deals with the other important aspects of DBMS selection.

As a basis for ongoing exploration of database features, Chapter 9 covers the physical design of databases and the characteristics of housing a database on a secondary storage device.

Chapter 10 follows by guiding students into the principles of DBMS functional design through a series of discussions about file organization and search techniques.

Chapter 11 then reviews the operational features that provide a basis for evaluating how well a DBMS meets the needs of a potential using organization. Topics include security, concurrency control, recovery techniques, and auditability.

This sequence concludes with a chapter that reviews the selection and implementation of a DBMS. Included is a review that relates selection of the data model to decisions involving the physical aspects of DBMS implementation.

IV. The Future

The final group of chapters begins by recognizing the impact of microcomputers on the future of database utilization. An initial chapter overviews the impact of microcomputers, then walks students through an example of how database packages are implemented on microcomputers. The presentation uses a popular database package (dBASE III™) as its example, providing a step-by-step discussion of design and development of a system that uses this software.

Remaining chapters overview advanced technologies that are on the horizon, including distributed databases, database machines, and intelligent database systems (expert systems).

The book concludes with a discussion of database-related careers, stressing particularly the role and responsibilities of a database administrator.

ACKNOWLEDGMENTS

In addition to the benefits derived from the cognizance of DPMA throughout the development of this work, a group of content reviewers scrutinized the manuscript to assure its completeness and validity. Particularly valuable contributions from the following are acknowledged:

William E. Leigh, University of Southern Mississippi, Hattiesburg
Frederick Gallegos, Los Angeles Region, U.S. General Accounting Office
Mike Michaelson, Palomar College, San Marcos, CA
Charles M. Williams, Georgia State University, Atlanta, GA

CONTENTS

THE EVOLUTION OF DATABASE MANAGEMENT 1

PERSPECTIVE

This chapter establishes a perspective and a uniform frame of reference to guide further study of database management systems (DBMS). The following principles are established:

- Database concepts evolved from a continuing series of hardware and software enhancements, such as disk files and operating systems.
- Within the perspective of this text, a database is a structured, usually large collection of data.
- A DBMS is a general purpose software and hardware facility to store, retrieve, protect, and share content of a database.
- There are no existing DBMS standards. At present, each DBMS product is unique.
- A DBMS implements a data model, which consists of a structure and a series of operations performed upon the structure.
- Use of a DBMS enhances user, as well as programmer, productivity.

DATA AS AN ASSET

Database management is an innovation in the creation, use, and preservation of computerized data. The basic premise that underlies the database management approach is that data are an asset. Data have value to an organization in achieving its goals and continuing its operation in a smooth and reliable way. The cost of capturing and maintaining computer data dictates that data must be preserved carefully over time and made readily accessible for many purposes.

This perception did not always exist. In the early days of the so-called computer revolution, emphasis in computer use was placed mostly upon sequential job processing. In effect, the computer was treated, systematically, as a higher-speed replacement for the punched card systems that preceded it. In a punched card system, processing involved passing cards through a sequence of machines, each of which performed a single sorting, collating, arithmetic, printing, or summarizing operation.

To complete an application, cards might have to be passed through a dozen or more separate machines. A computer could do all of this in one or a few processing passes. In effect, computers were seen as doing the same thing as had always been done, only more efficiently. It was some time before managers and data processing users saw that the files of data created as by-products of processing-oriented jobs had continuing value of their own to answer questions and provide results for other activities.

As has happened in many other aspects of the computer field, recognition of value and potential seems to have corresponded with product development. Files first became valuable as assets on their own in the late 1950s, following the introduction of the first random access disk file system—the IBM 305 RAMAC (for RAndoM ACcess). Almost immediately, computer systems took on a new dimension: Computers could store data for access at random, providing information or supporting business operations.

The predominant application for the early RAMAC systems was in a coordinated order processing, billing, and inventory management methodology. That is, data about customers and products for sale by a company were placed on disk files. This made it possible to input simple transaction data about customer orders, such as customer number, product number, and quantities sold. The computer then would retrieve the other information stored in its files to complete the order and print out the necessary order handling and billing documents. In addition, the computer would update its files to depict the current levels of inventory and customer credit. Order handling was simplified (from the human standpoint) and enhanced because recent customer activity data were available on disk. Ultimately, these systems were enhanced to put out exception reports on customers approaching or exceeding credit limits or inventory items reaching reorder points.

The introduction of random access capabilities led, in turn, to a new line of thought about what computers should be doing. Manufacturers began supplementing their hardware offerings with support software packages. In particular, programs like the batch monitor and input/output control system (IOCS) made it possible, in effect, for computers to tend their own operation, minimizing down time for setup and human support.

In proving themselves operationally, early utility programs such as IOCS established the feasibility of general purpose operating system software that could serve multiple users. No matter what processing had to be done, everybody had to get data into the system and out. This motivated the evolution of increasingly powerful utility programs for the creation and maintenance of files, as well as for other common job requirements.

The next step toward the evolution of database management systems, then, came with general-purpose file processors. These programs, which became widely available in the mid-1960s, were powerful tools that made it possible to store, retrieve, and update files for a variety of applications. However, it was still necessary, in using these tools, to define the specific, physical characteristics of the files and to be aware of the physical format of the records within them.

By contrast, database software treats data as an independent asset that has logical characteristics and value of its own. The fact that data represent a valuable asset has been borne out, through the years, because companies have actually gone out of business when their data assets have been destroyed. In this perspective, a computer with its DBMS capabilities takes on the role of guardian over the operating foundation and continuity of the organization itself. Under a database system, the computer is used as an asset manager, accruing to the collected data of an organization a value that is separate from and independent of the application programs that presently reference or handle the data.

DBMS—DEFINITIONS AND TERMS

A *database* is a structured, usually large, collection of data.

A *database management system (DBMS)* is a general purpose software and hardware facility to store, retrieve, protect, and share a database.

The terms within these definitions are used with specific meaning. Therefore, it is essential to highlight this terminology further.

Database size. The above definition specifies that a database is a large collection of data. This is a practical observation, rather than a physical necessity. In point of fact, database management software can be used for small collections of data. In some applications, databases start small and grow through use. In fact, database growth is a normal, realistic assumption. It would not be practical to invest in a database management capability to serve only a few dozen or even a few hundred records.

Structured. This term describes the ability to interrelate data stored within a database. The structures used, in turn, are among the dimensions for defining the data models upon which databases are built.

General purpose, application-independent. A database management system is a software capability or tool that can be used to support a wide range of applications and application-development efforts. That is, a database management system usually is not tailored to be usable for only a few individual applications. Rather, a DBMS is part of the processing environment for many applications in much the same way as the operating system software. Thus, it is entirely proper to consider the DBMS as a unit of systems software.

Hardware and software. Traditionally, database management systems have been software packages. However, there is an accelerating shift away from implementation in software alone. More recently, hardware has been playing an increasing role in DBMS implementation. Dedicated machines, as well as specialized processors and intelligent random access storage devices, have been introduced specifically for database management system implementation.

Database access. A database manager lives up to its name—quite literally. That is, an in-place database management system represents the only mechanism available for dealing with the data entrusted to it. The database software controls the insertion, change, retrieval, and deletion of all content of a database for which it is responsible.

Protect. The implication of protection of data assets by a DBMS is intended broadly. Given that data have value to their organization, they are entitled to the same caliber of protection as other assets—such as cash protected through placement in a vault. A DBMS protects data assets against unauthorized access, against software or hardware malfunctions, and against erroneous transactions.

Share. The definition presented above assumes a large amount of data as a qualification for using a DBMS. Given a large amount of data, it is assumed also that there will be multiple users who will share the database. They access the system concurrently, with each user experiencing a level of service that makes it appear as though each transaction had exclusive use of the data.

DBMS ACCEPTANCE AND GROWTH

The discussion to this point has traced the first evolutionary steps leading to the introduction of database management systems. Also presented were the basic terms and explanation of databases and database management systems. The discussion that follows picks up the historic perspective once again, focusing more directly on DBMS evolution.

DBMS Roots and Sources

Through the years, literally hundreds of DBMS products have been introduced commercially. An important characteristic of the DBMS field is the fact that each of these products is unique and largely incompatible with all of the others. Thus, one of the challenges in implementing a DBMS within any enterprise lies in evaluating and choosing from scores of qualified alternatives at any point in time.

The relative disarray of the DBMS marketplace at the moment stems, at least in large part, from the fact that there are no national, federal, or international standards that direct the evolution of database management systems. Rather, database management systems have been influenced by a variety of organizations and special interests. For example:

- Hardware vendors have brought out DBMS packages largely to stimulate sales of their equipment.
- University and government laboratories have developed DBMS products largely as technical research advances carried out by interested staff.
- Proprietary software and remote computing service companies have developed and offered their own database products to improve their market positions.

Significantly, the general acceptance of database concepts coincided closely with the "unbundling" of software from hardware within the computer industry as a whole. During the late 1960s, partly under the watchful eye of government antitrust regulators, decisions were made to separate the sale of software products from equipment contracts. This development saw a sudden, major growth in the availability of independently developed application packages.

Also, a number of companies found it profitable, because of the broad applicability of a DBMS, to offer database products as independent commercial ventures. Software houses sell or lease rights to use of their database manager packages, including future, enhanced versions or updates. Remote computing services organizations maintain DBMS service on networks that users access through communications-connected terminals.

Customers of these companies pay for the use of the software, for the processing on remote computers, and also for the storage of databases. For some companies in this field, the majority of revenue realized today is database related. Because of the specialized nature of these organizations, they have played a major role in product introduction and acceptance within the database field.

As mentioned above, there are, at this time, no national, federal, or international database standards in existence. However, efforts have been made by the Conference On DAta SYstems Languages (CODASYL) to establish appropriate designs and specifications that are widely usable for database vendors.

CODASYL played a major role in the software evolution of the computer industry as a whole by developing, introducing, and issuing specifications for COBOL (COmmon Business Oriented Language), which is the most widely used language for the programming of business applications. Following CODASYL's pioneering work in COBOL, the American National Standards Institute (ANSI) issued a series of standards through the years, that have guided the properties of COBOL compilers.

Subsequently, CODASYL issued a number of specifications for network-structured database management systems. In turn, a number of vendors have developed their own versions of database management systems following these specifications according to their own interpretations. At this writing, ANSI is reviewing the CODASYL work, among other contributions, with an eye toward developing database standards. Although no definitive standards have yet been accepted and applied, the ANSI effort is expected to produce results.

DBMS Development and Growth

Database management systems form a logical extension following the development of operating systems. Operating systems give computer users control of their equipment. Database management systems do the same thing for data resources.

Operating systems evolved largely because early software approaches did not take full advantage of computer capabilities. Initially, application programs written in machine language by users were all the software available. Each execution of an input command caused the computer to go through an entire three-step cycle before the next command could be presented. That is, each execution was carried through the steps of input, processing, and output before the next execution could begin.

Very early in the computer era, it became apparent that processing equipment was operating at only one-third of its potential capacity. To overcome this inefficiency, techniques were developed that made it possible to process instructions in a continuous stream. That is, as one execution moved from the input cycle to the processing cycle, a second command was input. Job streams were formed in which all three processing functions could be carried out upon different data simultaneously. However, at this stage, processing still took place upon one application at a time. After the application was processed, the computer was taken down and new application media and programs were mounted.

A logical next step was to devise management or executive software that could tend job queues. Jobs were automatically loaded in sequence as they were presented. In turn, implementation of these software concepts led to recognition that the computer was still spending most of its time idling while file access or input/output commands were being executed. That is, the processor could cycle much faster than the peripheral functions needed to support program execution. This, then, led to the interleaving of programs and to the development