Data Base Systems: A Practical Reference

7965455

Ian Palmer

华南工学院
图书馆藏书

TP31
P4

7965455

Data Base Systems:

A Practical Reference

Written By:

Ian R. Palmer

Foreword By:

Robert M. Curtice

E7965455

Editors:

Robert M. Curtice
Lawrence K. Grodman
Edwin F. Kerr

Q.E.D. Information Sciences, Inc. Wellesley, Massachusetts

Published in the United States of America by
Q.E.D. Information Sciences, Inc.
141 Linden Street
Wellesley, Massachusetts 02181

June, 1975 First U.S. Edition

December, 1975 Revised and Updated Edition

April, 1977 Second Printing

September, 1978 Third Printing

©1975 by Q.E.D. Information Sciences, Inc. and Ian R. Palmer
All Rights Reserved

No part of this book may be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of the publisher.

Previous editions published by SCICON Limited, London, England

Foreword

The increasing rate of technological development always carries with it new and broad demands for disseminating state-of-the-art advances. Technical people must have a sound understanding of underlying concepts and keep abreast of the details of new developments; management must be aware of the trends in making previously impractical capabilities economically viable; researchers must maintain awareness and close contact with other efforts and theoretical developments; and of course students must have a sound introduction both to the conceptual and practical aspects of the development. None of the conventional media is capable of serving as a vehicle for meeting these diverse needs perfectly in a time of constant change. A textbook requires stability; a journal or monograph prohibits the necessary comprehensive treatment; and a research report generally fails in breadth and broadness of view.

Yet this publication achieves all of the objectives for technical and management communications for data base systems. In a single volume, it thoroughly introduces the novice to the state-of-the-art. It embraces basic principles, existing implementation practices, and current research in the field. It is timely, reporting on present activities of equipment manufacturers, software houses and standards groups. It is interpretive, providing significant perspective on the issues to be faced. And above all it is practical, including guidelines for system selection and implementation, and reporting on the extensive experience of many user organizations. A textbook and a state-of-the-art report at the same time, this publication clearly achieves the most comprehensive treatment of data base systems to date.

It is clear from the first chapter that data bases represent a major investment for most organizations. An important concern is what the future holds in store for the technology and its impact on information processing. One of the very significant portions of this book is the chapter on approaches to data base software. Here for the first time is a complete picture of the efforts of the various standards and user groups which are affecting the future development of data base systems. Organizations do not want to invest in systems that will be incompatible with the "standard". Yet there continues to be controversy over standardization itself. The description of the standards efforts, the various directions taken by several data base systems, and the discussion of the issues involved in the standardization controversy all help to put the situation clearly in focus. From this chapter it becomes clear that standardization in any traditional sense is many years away, for the contestants are still at the starting line vying for position. Major differences of opinion still exist over what functions a data base system should perform and how it should look to those using it. The various approaches reviewed in this chapter address these issues somewhat differently, with diverse conclusions.

The possibilities for the future will clearly have great impact not only on the organizations using data base systems, but on the systems development process itself. Programming in a relational data base system, for example, is likely to change our view of the programming process much more than structured programming concepts. The prospect for considerable increase in data independence also has major implications. In the current state of development what is needed is not standardization, but an increased emphasis on theoretical and applied research.

Therefore, this valuable book by Ian R. Palmer, an experienced and respected consultant on data base technology, can play several roles. It is both a clear introduction to the field and a technical tutorial; an expository on user experience and an educated forecast of trends; a practical summary of several data base system specifications and a review of standards and techniques. The book is useful for industry programmers and computer science students alike; for line managers and project managers; for systems analysts and informed users. It is organized as follows:

In Chapter 1, "Data Base Concepts", the fundamental concepts of data base are presented. A history of the development of data base principles, definition of terms, and the place of data base in the spectrum of information processing are included. This chapter is a solid foundation for the material that follows.

Chapter 2, "Functions and Facilities of Data Base Systems," discusses in detail the functional requirements of a data base system, the various methods used to implement these functions, and how several existing systems fall short. The requirements for support of complex data structures and the need for data independence are particularly significant as these form the basis for important controversies about how data base systems should operate.

In Chapter 3, "Approaches to Data Base Software", the major conceptual approaches are discussed, including those of CODASYL, IBM, Guide-Share, and ANSI. The presentation of E.F. Codd's Relational Approach is an excellent introduction to a radically different treatment of the subject. As this chapter contains current status of each of these groups' efforts and how they conflict or complement each other, the reader is brought up to date on the very latest developments and issues in the data base field.

Chapter 4, "Data Base Management Systems," contains a review of many data base management systems representing a comprehensive picture of products both from American and European suppliers. It is perhaps the most complete and detailed review in the literature.

Chapter 5, "Installing a Data Base System," describes the major steps necessary for implementation, and provides suggestions and guidance for each. This material, aimed both at managers and technical people, describes the problems and considerations that must be recognized and dealt with in order to achieve success.

Chapter 6, "Likely Directions," contains a detailed review of the potential benefits and disadvantages of the data base approach. It contains a discussion of the possible impact of hardware and software advances on data base technology.

The Appendices contain a wealth of useful information, particularly Appendix A, "Case Studies." This contains an in-depth presentation of the experience of several major organizations who have implemented or are in the process of implementing data base systems. It represents the result of a considerable amount of experience and research by the author. Other appendices include a guide for comparative evaluation of data base management software, a complete glossary of terms and an extensive bibliography.

This book was originally written and recently updated under the auspices of Scientific Control Systems, Ltd. (SCICON) in London, for whom the author has been a Senior Consultant for several years. This new edition represents a considerable expansion and update, and is a clear, thorough and practical information source. It is a significant addition to the literature for professional data processors, managers and computer science students.

Robert M. Curtice
Arthur D. Little Inc.
Cambridge, Massachusetts

June, 1975

试读结束，需要全本PDF请购买 www.ertongbook.com

Preface

In 1972, I was undertaking a research project for Scientific Control Systems (SCICON) Limited into the relevance of database concepts in the future development of computing systems. Database was beginning to gain respectability as early users recovered from the traumatic experiences of pioneering a new technology and began to realize some of the anticipated benefits. IMS/2 became available to IBM users, Univac's DMS1100 gave credibility to the Codasyl database proposals, and Cincom's Total demonstrated that software houses would have a major influence on the database market. Yet, although the term "database" had been coined some five years earlier and the concepts themselves developed over at least ten years, there appeared to be no reference work treating the subject of database technology as a whole. Numerous articles and papers on such apparently esoteric subjects as the relational approach, the various detailed reports from Codasyl and a variety of superficial discussions in the trade press appeared, but no overview explaining what database was all about. I decided to attempt to fill that gap; the result was the first edition of this book.

In the ensuing three years the use of database systems has mushroomed, the technology has continued to develop, and new database software products abound. In order to keep pace with these developments, the book is already in its third edition. Earlier editions were named simply "Database Management" but as this title has since been used to label several different collections of articles and symposia proceedings, this edition has been renamed. Of greater significance are the many other revisions incorporated in this edition, including:

- Comments on the latest database work of Codasyl, Guide and ANSI.
- New ideas on access paths, system recovery, roll-back, reorganization, database statistics, and distributed database systems.
- Discussion on the ANSI SPARC proposals and on IBM's Data Independent Accessing Model.
- Current experience with software selection, data dictionaries, testing in a database environment and data administration.
- New case studies in the light of further experience of database users.
- Considerable extensions to the glossary, bibliography, and summary of data management systems.

The present role of database management is no less significant in the design and operation of commercial systems than it was three years ago. If anything, it is of even greater significance, as the control of data as a fundamental resource of an enterprise shared by all applications and departments becomes a practical reality with the advent of cheap mass storage and advanced data handling software. There are few computer users who have not or are not presently considering the possibility of the database approach. Its applicability is now as apparent to scientific systems and minicomputer users as to the large commercial users. This book is intended to contribute to their decision on whether to develop a database and if so, how to proceed.

Yet, given a field with such paucity of literature, it was difficult to know to which reader this book should be directed; to the manager needing an introduction to the subject, to the consultant advising on the feasibility of database, to the analyst designing a database system, or to the database administrator as a reference and guide. To resolve this dilemma, different chapters have been written with different readers in mind.

Another class of reader is the computer science student, as an increasing number of universities have recognized that a course in database management should be included as part of the syllabus for a degree in data processing. This book includes a great deal of valuable material for such a course.

In recent years I have perhaps been best known for my work on the Codasyl DDLC. Yet, as the book is designed to give an overall view of the current state of database technology, I have been at pains to present the various alternative approaches to that of Codasyl. Similarly, although the bulk of the book has been written in the light of my practical experience with databases, no one will expect me to have had hands—on experience with all forty database systems described (at least I hope they will not). When I was first involved with implementing databases, it was not even known that was in fact what we were doing, as the word was yet to be coined and the software was tailor-made. In more recent years, with responsibility for the database activities of a large international consulting company, I have been involved rather with feasibility studies, software selection and database education, which has provided a useful complementary vantage point to the earlier one of database implementation.

Although much progress has been made in the past three years, this book would not have been possible but for the support of SCICON in sponsoring its earlier editions. My thanks are also due to the company's staff members for their various contributions to the original research project, and to several dozen friends and colleagues in the computer industry for their invaluable comments and discussions which have led to the evolution of the various editions of the book. I am also deeply indebted to the several user organizations in the United States, Canada and Europe for their valuable input to the case studies. The rapid progress of database technology shows no signs of abating, so that it is unlikely that this edition will be the last. I shall be delighted to receive further constructive comments and ideas from readers, and to exchange views and experiences with computer users and researchers world wide.

Ian R. Palmer
CACI Inc. – International
London, England

June, 1975

Mr. Ian Palmer is the Manager of the Database and TP Group, CACI, Inc. – International, 30 St. George Street, London WI, England.

CACI, founded in 1962, is an international group of companies with substantial experience and expertise in Database Management Systems, Custom Systems Design and Programming Services. CACI's worldwide offices include Washington, D.C., Los Angeles, New York City, Pennsylvania, Ohio, Bermuda, the United Kingdom, the Netherlands and Italy.

Contents

CHAPTER 1 DATABASE CONCEPTS

1.1 Why Databases?

Any enterprise, whatever its ultimate purpose, is concerned with the manipulation of data. A hospital keeps patient records, a railway has time-tables, a hotel prepares reservations, a warehouse keeps stock records, a factory has production plans; all these enterprises also have accounting records and personnel files. Without the means of accessing and changing data, the enterprises rapidly would grind to a halt. Without communicating with existing data, no new stock could be ordered, the staff could not be paid, invoices could not be sent out, no tickets could be issued.

Communication is simply the exchange of data. It can be by word of mouth, by internal memo, by filling in a form, by punching a card for subsequent printing, or by instructing the computer to process the data and to issue the required results. The movement and volume of data in an enterprise can be enormous, with consequent problems of control and the difficulty of ensuring that the right people have the right information at the right time.

The introduction of data processing was an attempt to alleviate such problems. With the increasing pace of modern life and size of modern enterprises, the time was approaching when enterprises would have collapsed under the weight of their data. The computer imposed some discipline upon the situation, took over some of the routine data handling tasks, and made data easier to store and retrieve at the time and in the form required.

However, the computer brought problems of its own. The discipline imposed upon the exchange of data often was not natural to the organization, due to the rigorous definition of files and processing runs. As a result the computer was by-passed at times, and increasingly the data it controlled failed to represent the real-life situation of the enterprise. Present development of corporate databases with on-line access is an attempt to allow the computer to follow the natural rhythm of the enterprise and to model the natural relationships of various types of data.

In an ideal database system, the user department is no longer constrained by batch processing requirements. No longer does the user feel that he is not fully in control of the situation because some data needed for planning or operations is known only to the computer and is available only after an unacceptable delay. No longer must data be associated in certain limited ways to meet the requirements of the computer. The ultimate aim of a database management system is to enable the computer to model completely the natural data relationships and requirements of the enterprise without imposing any unwanted restriction on the use and communication of data.

1.2 What is a Database?

1.2.1 Definitions

The first problem one faces in studying this subject is to determine precisely what is a database and what constitutes a database management system. The casual user of the term "database" tends to refer to any organized collection of data capable of being accessed by a computer. This could be (and often is) applied to a couple of reels of magnetic tape or a few boxes of punched cards. As such, it is not a useful concept.

More helpfully a database has been defined as:

1. "A common pool of shared data."
2. "The fundamental storage of relevant data for the operational and strategic planning and control of an enterprise." (Reference 81)
3. "All record occurrences, set occurrences and areas controlled by a specific schema." (Reference 28)
4. "A collection of data on a defined range of subjects together with all the information needed to access that data."
5. "A named collection of units of physical data which are related to each other in a specified manner." (Reference 81)
6. "A generalized, common, intergrated collection of company or installation owned data which fulfills the data requirements of all applications which access it, and which is structured to model the natural data relationships which exist in an enterprise." (Reference 140)

These definitions each regard the database from a different viewpoint; its access, purpose, description, contents and integration. Yet each refers to specific collections of data, rather than to any data on computer-readable media. Even the first definition, the simplest, implies that the data is available at any one time to several users for several different purposes, which is not the case with a magnetic tape. The last definition does consider three essential and practical characteristics of a database; that it is:

- An organized, integrated collection of data.
- A natural representation of the data, with few imposed restrictions or modifications to suit the computer.
- Capable of use by all relevant applications without duplication of data.

A database management system (DBMS) simply is the software that supports such a database. The system of itself does not produce output of direct value to user departments; rather, the DBMS is a tool by which output can be produced more readily. In this respect it differs from the concept of an integrated management information system (IMIS). The DBMS itself does not include application programs that translate stored data into information meaningful to the user; however, a DBMS is likely to form the basis of a successful management information system.

1.2.2 Requirements of a Database System

Definitions are not of great use in deciding whether a system is genuinely a database management system, or whether it would be better classified as a management information system, a file management system or an information retrieval system. It is preferrable instead to define a database management system in terms of the facilities it could be expected to provide:

试读结束，需要全本PDF请购买 www.ertongbook.com

1. The controlled integration of data, so as to avoid the inefficiency and inconsistency of duplicated data.

2. The separation of physical data storage from the logic of the applications using the data, to aid flexibility and ease of change in a dynamic environment.

3. A single control of all data, permitting controlled concurrent use by a number of independent on-line users.

4. Provision for complex file structures and access paths, such that relevant relationships between data units can be readily expressed and data can be retrieved most efficiently for a variety of applications.

5. Generalized facilities for the rapid storage, modification, reorganization, analysis and retrieval of data, so that the use of a database system imposes no restrictions upon the user.

6. Privacy controls to prevent unauthorized access to specific units of data, types of data or combinations of data.

7. Integrity controls to prevent misues or corruption of stored data, and facilities to provide complete reconstruction in the event of hardward or software failure.

8. Performance, both in a batch mode and on-line, that is consistent, measurable, and capable of being optimized.

9. Compatibility with major programming languages, existing source programs, a variety of hardware systems and operating systems, and data external to the database.

The most advanced of the database management systems currently available provide many but not all of these features. They do so in significantly different ways with resulting differences in features, performance and usage characteristics. In later chapters, the possible approaches to meeting user requirements are examined, together with the particular systems that use each approach.

1.2.3 The Database Approach

The database approach is more than merely a different computer technique involving the storage of data and the use of additional generalized software. Rather, it involves a new approach to designing and operating information systems throughout the enterprise and can have far-reaching effects well beyond the data processing department. It regards data as a resource to be managed along with the more generally recognized resources of an enterprise (staff, premises, machinery etc.) so as to be available to a variety of applications and users.

The integrated database is intended to provide a consistent view of the enterprise's data for all user departments. Although most departments have responsibility for specific data, several departments using the same type of data can operate on the same data values. A parochial view of data would be discouraged, if not impossible, and the uncoordinated exchange of information among departments would not be necessary. Instead, all basic data would be input to the database by each department responsible, and retrieved (possibly in a summarized form) by those departments needing access to it.

Thus a database is more than merely a base of data. It is the basis of a new approach to data processing. As systems becomed advanced a database will not be merely the base of data for an enterprise. It will contain and control all the data resources of the enterprise. Thus, the alternative spelling "data base" is misleading and has been avoided in this publication. The database approach is not dependent on any particular structure or base of date and indeed, as subsequent chapters show, many database concepts can be applied to conventional files and existing systems.

The database approach is virtually impractical unless a computer is used, and is most beneficial when on-line access is provided. Even for a small enterprise the volume of data involved and the difficulty of extracting information in an acceptable time period are such that a data processing system is necessary. With on-line access the collection of data and the extraction of timely reports is straightforward because the restrictions and delays of batch processing are not imposed upon the user.

The database approach cannot be adopted overnight even if restricted to only a few user departments. It requires not only the creation of a meaningful database containing all significant data, but also thorough preparation of user staff. In practice a phased implementation might involve using a batch system before progressing to an on-line system.

1.3 Terminology

1.3.1 Database Jargon

As is normally the case with data processing, the mystique of the subject of databases is increased by its jargon. This is not to suggest that new terms are not essential, as there are certainly many new concepts that must be named. However, research and development in database technology have been widespread, and each company involved has tended to employ its own terminology, frequently with several organizations using the same word in slightly different or even contradictory senses.

This problem might well be expected between different suppliers' products, and certainly makes their comparison more difficult. However such organizations as Codasyl and the Guide-Share Group differ widely in the terminology they use. Even Codasyl itself has difficulty in standardizing precise meanings among its several committees, so rapidly are new concepts developed.

Wherever possible, this publication used the terminology of the Codasyl Database Task Group April 1971 Report (Reference 28). In describing particular software products, the suppliers' terms have been used, except where these are clearly misleading. Appendix C provides definitions of the key terms used in this book or found in the literature.

1.3.2 Units of Data

The area of immediate difficulty with conflicting terminology is in defining the various levels of data breakdown. It seems desirable to limit the number of levels to those with a distinct logical meaning.

1. Database

In this book, the database is the highest level of data grouping. An enterprise has only one database, which contains all its data. Some writers term this the *databank,* consisting of several independent databases, but it is doubtful whether in a single enterprise any group of data can ultimately be completely independent.

A databank is more usefully viewed as a particular type of database, where data is stored because of its intrinsic reference value, and in which it is not necessary to know how individual users will make use of the data. Research reports, economic statistics, tax schedules and similar relatively static information can be thought of as data bank contents as opposed to more volatile data concerning the operations of an enterprise.

2. File

The concept of files has been useful in conventional data processing, but most database systems have intentionally abandoned it because of its previous connotations. Instead, the term *Area* is often used to indicate a subdivision of the database directly related to a unit of physical storage. The most useful definition of a file restricts its contents to records of a single type, or at least to closely related records such as order records, and their detail in terms of line-item records.

3. Set

This is a concept peculiar to the database approach. A set represents a relationship between associated records and provides a means of access from one related record to another. Thus, the order record and its line-items could constitute a set. It is a Codasyl term; other systems use the word chain.

4. Record

A record is the logical unit of input-output in an application program. With most database systems it is the basic data unit by which information is processed and stored. Hence, it consists of a collection of data-items that identify and describe a particular thing. The terms *entry, entity* and *tuple* are used synonomously with record; but, in fact, each has distinct connotations that are explored in later chapters.

Records are often the units related in a data structure and in particular in a hierarchic structure. When two records are related, the superior record in the hierarchy is variously known as the "owner," "master" or "parent record," and the lower as the "member," "detail," or "child record."

5. Group

A group is a unit of data within a record, again consisting of a collection of data-items, but grouped so that they can be referenced together. Thus, the three data-items day, month and year can together be referenced as the group date. Codasyl uses the term *data-aggregate;* other systems refer to them as *segments.*

6. Data-Item

An item is the smallest unit of definable data. It is also known as an *element, elementary* *item* and *field.* Items of the same type constitute a domain. The value of an item is its attribute.

The distinction among these six levels of data-aggregation is not always clear. In IBM's IMS, a file is known as a database; with Codasyl, a file can consist of a single set; in several other systems it is unclear whether it is the record or the group (segment) that is the basic unit of data manipulation.

1.3.3 Areas of Potential Confusion

In reading database literature there are several areas where one needs to be particularly wary, as similar terms are used in different or even conflicting ways. In particular, it is usually essential to distinguish between:

1. Data Type and Occurrence

The generic term for a unit of data (e.g. record or set) is often used in reference to a single instance or occurrence of the unit, collectively all units of that type, or even the program definition of the unit. In this publication the terms unit-occurrence, unit-type and unit-entry are used for these meanings respectively, wherever there is the possibility of ambiguity. This is the convention adopted by Codasyl. Generally, where the word unit is unqualified, it refers to a single occurrence of the unit.

Thus, for example, record or record-occurrence refers to a particular record, such as the payroll record for employee 1234; record type refers to all records of that type, such as all payroll records; record-entry refers to the coded definition of the payroll record.

2. Physical and Logical Data

Data can be viewed at various levels, from that of the storage media to that of the user department. Physical data can be regarded as existing at the level at which data is stored or handled by the operating system. Higher levels can be termed logical data. The terms *notional* and *virtual* data are also used. However, in a discussion on database management, one is normally concerned with only two levels:

1. The data as it is handled by the database management system and described in its schema.
2. The data as it is required by the application program and described in its sub-schema.

In this publication physical and logical have been used respectively in these two senses. But it ought to be kept in mind that the physical/logical nature of data is not cut and dry, but ranges from the pattern of magnetic dots on a disc, to the representation of an entity and its attributes in the database, to the figures printed in a report.

3. Data and Information

The values stored in a database are referred to as data. They become information only when associated with some definition of their meaning, purpose and relationship with other values. Normally it is the user who places an interpretation upon the data to derive information that is meaningful to him. Most current systems are concerned with data rather than information. Thus, the system may be aware that it has parts records with parts numbers, quantity and weight data-items, but not that these are the parts produced by a particular factory and that the weight is that of the standard pack, which contains one or more of the parts as given by the quantity field.

4. Programs and Run-units

   Strictly speaking, a program is a collection of object or source code; a run-unit is a particular execution of this code together with any associated code existing as part of the DBMS and operating system. Thus two run-units may be executing the same program. Except where the distinction is important, the term program is used in both senses in this publication.

1.4 Basic Concepts

1.4.1. Database Programming

Normally, a database is supported by DBMS software supplied by a manufacturer or software house. To control the operation of the database, the user interfaces with the software by means of several database programming languages, each with a specific function:

1. Data Description Language (DDL)

   The DDL defines the structure and format of data in the database, the relationships among units of data, and the methods of access to the data. This definition of the database is called its *schema.* The DDL can also define the data requirements of application programs. These definitions are known as *subschemas.* The DDL can be likened in function to the Data Division clauses of Cobol except that it can be used independently of any specific program.

2. Data Manipulation Language (DML)

   The DML specifies the operations of retrieval of data from the database, its modification, its storage and its deletion. Similarly, it establishes and removes relationships between data. The DML can be likened to the file access verbs in the Procedure Division of a Cobol program, and can exist only as part of an application program. If the DML is embedded in a conventional programming language, the latter is termed the *host language.*

3. Device Media Control Language (DMCL)

   The DMCL defines the mapping of the database to its storage media. It is likely to be employed only by systems programmers, as it is concerned with the optimal use of the total database rather than with individual applications.

The functions represented by these three languages can be recognized in any database system, although they may not exist as three independent entitles. For example, the DMCL may be provided as part of the DDL.

In general terms, a DDL need not be restricted to use with a database, but could describe any collection of data. Nor need a database be limited to the use of one DML; several DML's differing in procedurality and/or host language can be used to manipulate the same database just as, for example, Cobol and Fortran programs can process the same data.

The concept of independent DML's, DDL and DMCL is of great importance in permitting data and device independence (Section 2.4).

1.4.2 Database Operations

In its totality, a database management system can be regarded as a combination of staff, software and hardware functions responsible for maintaining and providing access to a database. The staff responsibilities are termed the *database administrator function.*

The DBA (Database Administrator) is a relatively new, yet key position within a DP department. The database imposes a degree of centralization upon the enterprise's data, and the control and coordination of this data as a vital resource of the enterprise is the task of the DBA. This is described more fully in Section 5.5.

A DBMS can be operated solely in batch mode, and economics may possibly dictate such an approach, but some advantages it provides are not utilized. Most database systems can be used in an on-line mode in conjunction with the operating system and a message control program. A few systems do contain their own data communications software to hande a variety of remote terminals and lines. This publication is concerned with on-line usage of databases, but not with such aspects as teleprocessing software or data communications.

1.4.3 Database Handler

Database management software can be implemented in a variety of ways. Typically in most modern implementations it interfaces with the operating system, the database, the applications programs, and generalized end-user facilities.

Figure 1 shows the conceptual operation of such a database system within a typical multiprocessing environment. Central to the system is the Database Handler, which is that part of DBMS providing access to the database for user programs. This software is also known as Data Management Routines (DMR), Database Control System (DBCS) and Database Manager (DBM). The diagram does not attempt to illustrate other aspects of the DBMS software, such as database utilities, query processors and schema and subschema compilers. The activities at each interface as indicated in the diagram are briefly as follows:

1. The run-unit issues a call to the database by means of a DML command, which is interpreted by the database handler.
2. The handler consults the previously compiled schema and the appropriate subschema to determine how the program's request can be satisfied.
3. Having determined the physical data involved, the database handler issues an appropriate command to the operating system, unless the data is already present in the system buffers.
4. The operating system interacts with the database in the same way as with files in a conventional system.
5. The necessary data is read from or written into the system buffers.
6. In the case of a DML GET command, the handler selects the appropriate data from the buffer, performs any transformations necessary to produce the logical data required by the user program, and stores the result in the user work area.

试读结束，需要全本PDF请购买 www.ertongbook.com