# 数据库系统导论

## An Introduction to Database Systems (Seventh Edition)

（英文版 · 第7版）
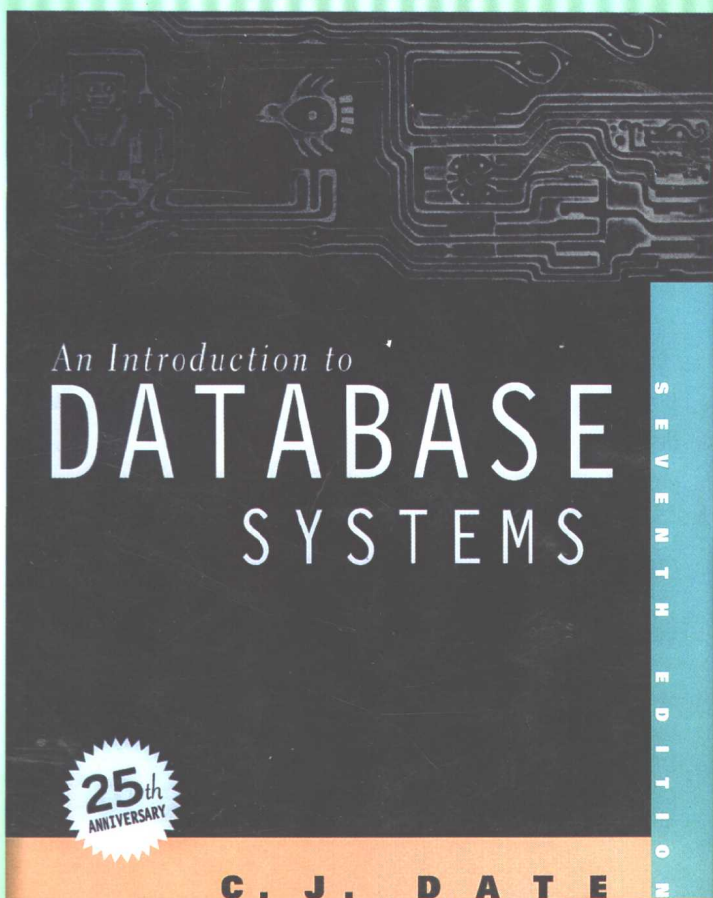
An Introduction to

# DATABASE
## SYSTEMS

SEVENTH EDITION

25th
ANNIVERSARY

C. J. DATE

〔美〕C. J. Date 著

（英文版·第7版）

# 数据库系统导论

# An Introduction to Database Systems

## (Seventh Edition)

（美）C. J. Date 著

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

| S | S# | SNAME | STATUS | CITY |
|---|---|---|---|---|
| | S1 | Smith | 20 | London |
| | S2 | Jones | 10 | Paris |
| | S3 | Blake | 30 | Paris |
| | S4 | Clark | 20 | London |
| | S5 | Adams | 30 | Athens |

| SP | S# | P# | QTY |
|---|---|---|---|
| | S1 | P1 | 300 |
| | S1 | P2 | 200 |
| | S1 | P3 | 400 |
| | S1 | P4 | 200 |
| | S1 | P5 | 100 |
| | S1 | P6 | 100 |
| | S2 | P1 | 300 |
| | S2 | P2 | 400 |
| | S3 | P2 | 200 |
| | S4 | P2 | 200 |
| | S4 | P4 | 300 |
| | S4 | P5 | 400 |

| P | P# | PNAME | COLOR | WEIGHT | CITY |
|---|---|---|---|---|---|
| | P1 | Nut | Red | 12 | London |
| | P2 | Bolt | Green | 17 | Paris |
| | P3 | Screw | Blue | 17 | Rome |
| | P4 | Screw | Red | 14 | London |
| | P5 | Cam | Blue | 12 | Paris |
| | P6 | Cog | Red | 19 | London |

The suppliers and parts database (sample values)

| S | S# | SNAME | STATUS | CITY |
|---|----|-------|--------|------|
| | S1 | Smith | 20 | London |
| | S2 | Jones | 10 | Paris |
| | S3 | Blake | 30 | Paris |
| | S4 | Clark | 20 | London |
| | S5 | Adams | 30 | Athens |

| P | P# | PNAME | COLOR | WEIGHT | CITY |
|---|----|-------|-------|--------|------|
| | P1 | Nut | Red | 12.0 | London |
| | P2 | Bolt | Green | 17.0 | Paris |
| | P3 | Screw | Blue | 17.0 | Rome |
| | P4 | Screw | Red | 14.0 | London |
| | P5 | Cam | Blue | 12.0 | Paris |
| | P6 | Cog | Red | 19.0 | London |

| J | J# | JNAME | CITY |
|---|----|-------|------|
| | J1 | Sorter | Paris |
| | J2 | Display | Rome |
| | J3 | OCR | Athens |
| | J4 | Console | Athens |
| | J5 | RAID | London |
| | J6 | EDS | Oslo |
| | J7 | Tape | London |

| SPJ | S# | P# | J# | QTY |
|-----|----|----|----|-----|
| | S1 | P1 | J1 | 200 |
| | S1 | P1 | J4 | 700 |
| | S2 | P3 | J1 | 400 |
| | S2 | P3 | J2 | 200 |
| | S2 | P3 | J3 | 200 |
| | S2 | P3 | J4 | 500 |
| | S2 | P3 | J5 | 600 |
| | S2 | P3 | J6 | 400 |
| | S2 | P3 | J7 | 800 |
| | S2 | P5 | J2 | 100 |
| | S3 | P3 | J1 | 200 |
| | S3 | P4 | J2 | 500 |
| | S4 | P6 | J3 | 300 |
| | S4 | P6 | J7 | 300 |
| | S5 | P2 | J2 | 200 |
| | S5 | P2 | J4 | 100 |
| | S5 | P5 | J5 | 500 |
| | S5 | P5 | J7 | 100 |
| | S5 | P6 | J2 | 200 |
| | S5 | P1 | J4 | 100 |
| | S5 | P3 | J4 | 200 |
| | S5 | P4 | J4 | 800 |
| | S5 | P5 | J4 | 400 |
| | S5 | P6 | J4 | 500 |

The suppliers-parts-projects database (sample values)

```
<table expression>
   ::=   <join table expression>
       | <nonjoin table expression>

<join table expression>
   ::=   <table reference>
             [ NATURAL ] JOIN <table reference>
             [ ON <conditional expression>
             | USING ( <column name commalist> ) ]
       | <table reference> CROSS JOIN <table reference>
       | ( <join table expression> )

<table reference>
   ::=   <table name> [ [ AS ] <range variable name>
                       [ ( <column name commalist> ) ] ]
       | ( <table expression> ) [ AS ] <range variable name>
                       [ ( <column name commalist> ) ]
       | <join table expression>

<nonjoin table expression>
   ::=   <nonjoin table term>
       | <table expression> UNION [ ALL ] [ CORRESPONDING
                       [ BY ( <column name commalist> ) ] ]
                       <table term>
       | <table expression> EXCEPT [ ALL ] [ CORRESPONDING
                       [ BY ( <column name commalist> ) ] ]
                       <table term>

<nonjoin table term>
   ::=   <nonjoin table primary>
       | <table term> INTERSECT [ ALL ] [ CORRESPONDING
                       [ BY ( <column name commalist> ) ] ]
                       <table primary>

<table term>
   ::=   <nonjoin table term>
       | <join table expression>

<table primary>
   ::=   <nonjoin table primary>
       | <join table expression>

<nonjoin table primary>
   ::=   TABLE <table name>
       | <table constructor>
       | <select expression>
       | ( <nonjoin table expression> )

<table constructor>
   ::=   VALUES <row constructor commalist>

<row constructor>
   ::=   <scalar expression>
       | ( <scalar expression commalist> )
       | ( <table expression> )

<select expression>
   ::=   SELECT [ ALL | DISTINCT ] <select item commalist>
             FROM <table reference commalist>
                 [ WHERE <conditional expression> ]
                   [ GROUP BY <column name commalist> ]
                       [ HAVING <conditional expression> ]

<select item>
   ::=   <scalar expression> [ [ AS ] <column name> ]
       | [ <range variable name>. ] *
```

The syntax of SQL <table expression>s

```
<conditional expression>
   ::=   <conditional term>
       | <conditional expression> OR <conditional term>

<conditional term>
   ::=   <conditional factor>
       | <conditional term> AND <conditional factor>

<conditional factor>
   ::=   [ NOT ] <conditional primary>

<conditional primary>
   ::=   <simple condition> | ( <conditional expression> )

<simple condition>
   ::=   <comparison condition>
       | <in condition>
       | <like condition>
       | <match condition>
       | <all or any condition>
       | <exists condition>
       | <unique condition>

<comparison condition>
   ::=   <row constructor>
                  <comparison operator> <row constructor>

<comparison operator>
   ::=   = | < | <= | > | >= | <>

<in condition>
   ::=   <row constructor> [ NOT ] IN ( <table expression> )
       | <scalar expression> [ NOT ] IN
                             ( <scalar expression commalist> )

<like condition>
   ::=   <character string expression>
                [ NOT ] LIKE <pattern> [ ESCAPE <escape> ]

<match condition>
   ::=   <row constructor>
                  MATCH UNIQUE ( <table expression> )

<all or any condition>
   ::=   <row constructor>
         <comparison operator> ALL ( <table expression> )
       | <row constructor>
         <comparison operator> ANY ( <table expression> )

<exists condition>
   ::=   EXISTS ( <table expression> )

<unique condition>
   ::=   UNIQUE ( <table expression> )
```

The syntax of SQL <conditional expression>s

# 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到"出版要为教育服务"。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以"计算机科学丛书"为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

"计算机科学丛书"的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，"计算机科学丛书"已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在"华章教育"的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成"国外经典教材"系列；对影印版的教材，则单独开辟出"经典原版书库"；定位在高级教程和专业参考的"计算机科学丛书"还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成"专家指导委员会"，为我们提供选题意见和出版监督。

"经典原版书库"是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。在广泛地征求并听取丛书的"专家指导委员会"的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：（010）68995265
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

# 专家指导委员会

（按姓氏笔画顺序）

# About the Author

C. J. Date is an independent author, lecturer, researcher, and consultant, specializing in relational database technology. He is based in Healdsburg, California.

In 1967, following several years as a mathematical programmer and programming instructor for Leo Computers Ltd. (London, England), Mr. Date moved to the IBM (UK) Development Laboratories, where he worked on the integration of database functionality into PL/I. In 1974 he transferred to the IBM Systems Development Center in California, where he was responsible for the design of a database language known as the Unified Database Language, UDL, and worked on technical planning and externals design for the IBM products SQL/DS and DB2. He left IBM in May 1983.

Mr. Date has been active in the database field for almost 30 years. He was one of the first people anywhere to recognize the significance of Codd's pioneering work on the relational model. He has lectured widely on technical subjects—principally on database topics, and especially on relational database—throughout North America and also in Europe, Australia, Latin America, and the Far East. In addition to the present book, he is author or coauthor of several other database books, including *Foundation for Object/Relational Databases: The Third Manifesto* (1998), a detailed proposal for the future direction of the field; *Database: A Primer* (1983), which treats database systems from the nonspecialist's point of view; a series of *Relational Database Writings* books (1986, 1990, 1992, 1995, 1998), which deal with various aspects of relational technology in depth; and another series of books on specific systems and languages—*A Guide to DB2* (4th edition, 1993), *A Guide to SYBASE and SQL Server* (1992), *A Guide to SQL/DS* (1988), *A Guide to INGRES* (1987), and *A Guide to the SQL Standard* (4th edition, 1997). His books have been translated into many languages, including Chinese, Dutch, French, German, Greek, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish, and Braille.

Mr. Date has also produced over 300 technical articles and research papers and has made a variety of original contributions to database theory. He is a regular columnist for the magazines *Database Programming & Design* and *Intelligent Enterprise*. His professional seminars on database technology (offered both in North America and overseas) are widely considered to be second to none for the quality of the subject matter and the clarity of the exposition.

Mr. Date holds an Honours Degree in Mathematics from Cambridge University, England (BA 1962, MA 1966) and the honorary degree of Doctor of Technology from De Montfort University, England (1994).

# Preface to the Seventh Edition

This book is a comprehensive introduction to the now very large field of database systems. It provides a solid grounding in the foundations of database technology and gives some idea of how the field is likely to develop in the future. The book is meant primarily as a textbook, not a work of reference (though I believe it can be useful as a reference also, to some extent); the emphasis throughout is on **insight** and **understanding,** not just on formalisms.

## PREREQUISITES

The book as a whole is meant for anyone professionally interested in computing in some way who wants to gain an understanding of what database systems are all about. I assume you have at least a basic knowledge of both:

- The storage and file management capabilities (indexing, etc.) of a modern computer system;
- The features of one or more high-level programming languages (e.g.. C, Java, Pascal, PL/I, etc.).

## STRUCTURE

The book is divided into six major parts:

I. Basic Concepts

II. The Relational Model

III. Database Design

IV. Transaction Management

V. Further Topics

VI. Object and Object/Relational Databases

Each part in turn is divided into several chapters:

- Part I (four chapters) provides a broad introduction to the concepts of database systems in general and relational systems in particular. It also introduces the standard database language **SQL.**

- Part II (five chapters) consists of a detailed and very careful description of **the relational model,** which is not only the theoretical foundation underlying relational systems, but is in fact the theoretical foundation for the database field as a whole.

- Part III (four chapters) discusses the general question of **database design;** three chapters are devoted to design theory, the fourth considers semantic modeling and the entity/relationship model.

- Part IV (two chapters) is concerned with **transaction management** (i.e., recovery and concurrency controls).

- Part V (eight chapters) is a little bit of a *potpourri.* In general, however, it shows how relational concepts are relevant to a variety of further aspects of database technology— **security, distributed databases, temporal data, decision support,** and so on.

- Finally, Part VI (two chapters) describes the impact of **object technology** on database systems. Chapter 25 in particular, the last in the book, considers the possibility of a *rapprochement* between object and relational technologies and discusses **object/relational** systems.

There are also three appendixes—one giving further details of SQL, one on "SQL3" (a new version of SQL that is likely to be ratified as a standard round about the time this book appears in print), and one that lists some important abbreviations and acronyms.

*Note:* An online *Instructor's Manual* is also available, giving guidance on how to use the book as a basis for teaching a database course. It consists of a series of notes, hints, and suggestions on each part, each chapter, and each appendix, as well as answers to exercises not answered in the book itself and other supporting material. For instructions on how to access the *Manual,* please contact your local Addison-Wesley sales representative. To locate your local representative, please visit us on the web at *http://hepg.awl.com/rep-locator.*

## HOW TO READ THIS BOOK

The book overall is meant to be read in sequence more or less as written, but you can skip later chapters, and later sections within chapters, if you choose. A suggested plan for a first reading would be:

- Read Chapters 1 and 2 "once over lightly";

- Read Chapters 3 and 4 very carefully;

- Read Chapters 5, 6, 8, and 9 carefully, but skip Chapter 7 (except perhaps for Section 7.7);

- Read Chapter 10 "once over lightly";

- Read Chapters 11 and 13 carefully, but skip Chapter 12;

- Read Chapters 14 and 15 carefully;

- Read subsequent chapters selectively, according to taste and interest.

Each chapter opens with an introduction and closes with a summary; in addition, most chapters include a set of exercises, usually with answers (often the answers give additional information about the subject of the exercise). Most chapters also include an extensive list of references, many of them annotated. This structure allows the subject matter to be treated in a multi-level fashion, with the most important concepts and results being presented "in line" in the main body of the text and various subsidiary issues and more complex aspects being deferred to the Exercises or Answers or References section, as appropriate. *Note:* References are identified in the text by two-part numbers in square brackets. For example, the reference "[3.1]" refers to the first item in the list of references at the end of Chapter 3: namely, a paper by E. F. Codd published in *CACM*, Vol. 25, No. 2, in February, 1982. (For an explanation of abbreviations used in references—e.g., "CACM"—see Appendix C.)

## COMPARISON WITH EARLIER EDITIONS

The major differences between this edition and its immediate predecessor are summarized below.

- *Part I:* Chapters 1–3 cover roughly the same ground as Chapters 1–3 in the previous edition, but they have been rewritten, and the treatment of several topics has been improved and amplified. Chapter 4 is new (though partly based on the old Chapter 8); it provides an introduction to SQL, covering aspects that do not logically belong anywhere else in the book (in particular, host language bindings and embedded SQL).

- *Part II:* Chapters 5–9 (on the relational model) represent a completely rewritten, considerably expanded, and very much improved version of Chapters 4–7 and 17 from the previous edition. In particular, the sections on types (domains), relation values *vs.* relation variables, integrity, predicates, and views have all been drastically revised.

  *Note:* Some words of explanation are in order here. Earlier editions used SQL to illustrate relational ideas, in the belief that it is easier on the student to show the concrete before the abstract. Unfortunately, however, the gulf between SQL and the relational model has grown so wide that I now feel it would be actively misleading to use SQL for such a purpose. In fact, SQL in its present form is so far from being a true embodiment of relational principles—it suffers from so many sins of both omission and commission—that I would have preferred to relegate it to an appendix; but the language is so important from a commercial point of view (and every database professional needs to have some familiarity with it) that it would just not be appropriate to treat it in so dismissive a manner. I have therefore settled on a compromise—a chapter on SQL basics in Part I of the book, and individual sections in other chapters (where applicable) describing those aspects of SQL that are specific to the subject of the chapter in question.

- *Part III:* Chapters 10–13 are a significant revision of the old Chapters 9–12, with new material on relation-valued attributes, denormalization, orthogonal design, and alternative approaches to semantic modeling (including "business rules").

  *Note:* Again, some explanation is in order. Some reviewers of earlier editions complained that database design issues were being treated too late. But it is my feeling that

students are not ready to design databases properly or to appreciate design issues fully until they have some understanding of what databases are and how they are used; in other words, I believe it is important to spend some time on the relational model and related matters before exposing the student to design questions. Thus, I still believe Part III is in the right place in the book.

- *Part IV:* The two chapters of this part are slightly revised and extended versions of Chapters 13 and 14 from the previous edition.
- *Part V:* Chapters 19 (on type inheritance), 21 (on decision support), and 22 (on temporal databases) are all new. Chapters 16 (on security), 17 (on optimization), 18 (on missing information), and 20 (on distributed databases) are expanded and significantly revised versions of the old Chapters 15, 18, 20, and 21, respectively. Chapter 23 (on logic-based or deductive databases) is a revised version of the old Appendix C.
- *Part VI:* Chapter 24 is a completely rewritten and much improved version of the old Chapters 22–24. Chapter 25 is mostly new.

Finally, Appendix A is based on part of the old Chapter 8; Appendix B is new; and Appendix C is an updated version of the old Appendix D.

In addition to the changes sketched above, the following topics have been dropped in this edition:

- Storage structures and access methods (old Appendix A);
- Detailed discussion of DB2 (old Appendix B).

## WHAT MAKES THIS BOOK DIFFERENT?

Every database book on the market has its own individual strengths and weaknesses, and every writer has his or her own particular ax to grind. One concentrates on transaction management issues; another stresses entity/relationship modeling; yet another looks at everything through an SQL lens; still another takes a pure "object" point of view; still another views the field exclusively in terms of commercial products; and so on. And, of course, I am no exception to this rule—I too have an ax to grind: what might be called the **foundation** ax. I believe very firmly that we must get the foundation right, and understand it properly, before we try to build on that foundation in any way. This belief on my part explains the heavy emphasis in this book on the relational model; in particular, it explains the length of Part II—the most important part of the book—where I present my own understanding of the relational model as carefully as I can. I am interested in foundations, not fads and fashions.

In this connection, I should say too that I am well aware that the overall tone of this book has changed over the years. The first few editions were mostly descriptive in nature; they described the field as it actually was in practice, "warts and all." This edition, by contrast, is much more *prescriptive*; it talks about the way the field *ought* to be and the way it ought to develop in the future, if we do things right (in other words, it is a textbook with

an attitude!). And the first part of that "doing things right" is surely educating oneself as to what those right things actually are. I hope this edition can help in that educational endeavor.

And another (related) point: Some of you might know that, along with my colleague Hugh Darwen, I recently published another "prescriptive" book on database technology, whose title (abbreviated) is *The Third Manifesto* [3.3]. That book builds on the relational model to offer a detailed technical proposal for future database systems (it is the result of many years of teaching and thinking about such matters on the part of both Hugh and myself). And, not surprisingly, the ideas of the *Manifesto* inform the present book throughout. Which is not to say that the *Manifesto* is a prerequisite to the present book—it is not; but it *is* directly relevant to much that is in the present book, and further pertinent information is often to be found therein.

# A CLOSING REMARK

I would like to close these prefatory notes with the following edited extract from another preface—Bertrand Russell's own preface to *The Bertrand Russell Dictionary of Mind, Matter and Morals* (ed., Lester E. Denonn), Citadel Press, 1993, reprinted here by permission:

> *I have been accused of a habit of changing my opinions . . . I am not myself in any degree ashamed of [that habit]. What physicist who was already active in 1900 would dream of boasting that his opinions had not changed during the last half century? . . . [The] kind of philosophy that I value and have endeavoured to pursue is scientific, in the sense that there is some definite knowledge to be obtained and that new discoveries can make the admission of former error inevitable to any candid mind. For what I have said, whether early or late, I do not claim the kind of truth which theologians claim for their creeds. I claim only, at best, that the opinion expressed was a sensible one to hold at the time . . . I should be much surprised if subsequent research did not show that it needed to be modified. [Such opinions were not] intended as pontifical pronouncements, but only as the best I could do at the time towards the promotion of clear and accurate thinking. Clarity, above all, has been my aim.*

If you compare earlier editions of this book with this seventh edition, you will find that I too have changed my opinions on many matters (and no doubt I will continue to do so). I hope you will accept the remarks quoted above as adequate justification for this state of affairs. I share Bertrand Russell's perception of what the field of scientific inquiry is all about, but he expresses that perception far more eloquently than I could.

# ACKNOWLEDGMENTS

Once again it is a pleasure to acknowledge my debt to the many people involved, directly or indirectly, in the production of this book. First of all, I must thank my friends David Mc-Goveran and Hugh Darwen for their major involvement in this edition: David contributed the first draft of Chapter 21 on decision support, and Hugh contributed the first draft of

Chapter 22 on temporal databases. Hugh also did a very thorough reviewing job on large portions of the manuscript, including in particular all of the relational chapters and the appendix on SQL3. Second, the text has benefited from the comments of students on the seminars I have been teaching over the past several years. It has also benefited enormously from the comments of, and discussion with, numerous friends and reviewers, including Charley Bontempo, Declan Brady, Hugh Darwen (again), Tim Hartley, Adrian Larner, Chung Lee, David Livingstone, Nikos Lorentzos, Huizha Lu, Ramon Mata-Toledo, Nelson Mattos, David McGoveran (again), Fabian Pascal, Sudha Ram, Rick van der Lans, Yongdong Wang, Colin White, and Qiang Zhu. Each of these people reviewed at least some portion of the manuscript or made technical material available or otherwise helped me find answers to my many technical questions, and I am very grateful to all of them. I would also like to thank my wife Lindy for contributing the cover art once again. Finally, I am grateful (as always) to everyone at Addison-Wesley—especially Maite Suarez-Rivas and Katherine Harutunian—for all of their encouragement and support throughout this project, and to my editor Elydia Davis for her usual sterling job.

*Healdsburg, California*                                                                                      C. J. Date
*1999*