PREFACE

> The advantage of doing one's praising for oneself is that one can lay it on so thick and exactly in the right places.
>
> —Samuel Butler

Database management systems have become ubiquitous as a fundamental tool for managing information, and a course on the principles and practice of database systems is now an integral part of computer science curricula. This book covers the fundamentals of modern database management systems, in particular relational database systems. It is intended as a text for an introductory database course for undergraduates, and we have attempted to present the material in a clear, simple style.

A quantitative approach is used throughout and detailed examples abound. An extensive set of exercises (for which solutions are available online to instructors) accompanies each chapter and reinforces students' ability to apply the concepts to real problems. The book contains enough material to support a second course, ideally supplemented by selected research papers. It can be used, with the accompanying software and SQL programming assignments, in two distinct kinds of introductory courses:

1. A course that aims to present the principles of database systems, with a practical focus but without any implementation assignments. The SQL programming assignments are a useful supplement for such a course. The supplementary Minibase software can be used to create exercises and experiments with no programming.

2. A course that has a strong systems emphasis and assumes that students have good programming skills in C and C++. In this case the software can be used as the basis for projects in which students are asked to implement various parts of a relational DBMS. Several central modules in the project software (e.g., heap files, buffer manager, B+ trees, hash indexes, various join methods, concurrency control, and recovery algorithms) are described in sufficient detail in the text to enable students to implement them, given the (C++) class interfaces.

Many instructors will no doubt teach a course that falls between these two extremes.

Choice of Topics

The choice of material has been influenced by these considerations:

- To concentrate on issues central to the *design, tuning, and implementation of relational database applications*. However, many of the issues discussed (e.g., buffering and access methods) are not specific to relational systems, and additional topics such as decision support and object-database systems are covered in later chapters.
- To provide adequate coverage of implementation topics to support a concurrent laboratory section or course project. For example, implementation of relational operations has been covered in more detail than is necessary in a first course. However, the variety of alternative implementation techniques permits a wide choice of project assignments. An instructor who wishes to assign implementation of sort-merge join might cover that topic in depth, whereas another might choose to emphasize index nested loops join.
- To provide in-depth coverage of the state of the art in currently available commercial systems, rather than a broad coverage of several alternatives. For example, we discuss the relational data model, B+ trees, SQL, System R style query optimization, lock-based concurrency control, the ARIES recovery algorithm, the two-phase commit protocol, asynchronous replication in distributed databases, and object-relational DBMSs in detail, with numerous illustrative examples. This is made possible by omitting or briefly covering some related topics such as the hierarchical and network models, B tree variants, Quel, semantic query optimization, view serializability, the shadow-page recovery algorithm, and the three-phase commit protocol.
- The same preference for in-depth coverage of selected topics governed our choice of topics for chapters on advanced material. Instead of covering a broad range of topics briefly, we have chosen topics that we believe to be practically important and at the cutting edge of current thinking in database systems, and we have covered them in depth.

New in the Second Edition

Based on extensive user surveys and feedback, we have refined the book's organization. The major change is the early introduction of the ER model, together with a discussion of conceptual database design. As in the first edition, we introduce SQL-92's data definition features together with the relational model (in Chapter 3), and whenever appropriate, relational model concepts (e.g., definition of a relation, updates, views, ER to relational mapping) are illustrated and discussed in the context of SQL. Of course, we maintain a careful separation between the concepts and their SQL realization. The material on data storage, file organization, and indexes has been moved back, and the

material on relational queries has been moved forward. Nonetheless, the two parts (storage and organization vs. queries) can still be taught in either order based on the instructor's preferences.

In order to facilitate brief coverage in a first course, the second edition contains overview chapters on transaction processing and query optimization. Most chapters have been revised extensively, and additional explanations and figures have been added in many places. For example, the chapters on query languages now contain a uniform numbering of all queries to facilitate comparisons of the same query (in algebra, calculus, and SQL), and the results of several queries are shown in figures. JDBC and ODBC coverage has been added to the SQL query chapter and SQL:1999 features are discussed both in this chapter and the chapter on object-relational databases. A discussion of RAID has been added to Chapter 7. We have added a new database design case study, illustrating the entire design cycle, as an appendix.

Two new pedagogical features have been introduced. First, 'floating boxes' provide additional perspective and relate the concepts to real systems, while keeping the main discussion free of product-specific details. Second, each chapter concludes with a 'Points to Review' section that summarizes the main ideas introduced in the chapter and includes pointers to the sections where they are discussed.

For use in a second course, many advanced chapters from the first edition have been extended or split into multiple chapters to provide thorough coverage of current topics. In particular, new material has been added to the chapters on decision support, deductive databases, and object databases. New chapters on Internet databases, data mining, and spatial databases have been added, greatly expanding the coverage of these topics.

The material can be divided into roughly seven parts, as indicated in Figure 0.1, which also shows the dependencies between chapters. An arrow from Chapter I to Chapter J means that I depends on material in J. The broken arrows indicate a weak dependency, which can be ignored at the instructor's discretion. It is recommended that Part I be covered first, followed by Part II and Part III (in either order). Other than these three parts, dependencies across parts are minimal.

Order of Presentation

The book's modular organization offers instructors a variety of choices. For example, some instructors will want to cover SQL and get students to use a relational database, before discussing file organizations or indexing; they should cover Part II before Part III. In fact, in a course that emphasizes concepts and SQL, many of the implementation-oriented chapters might be skipped. On the other hand, instructors assigning implementation projects based on file organizations may want to cover Part

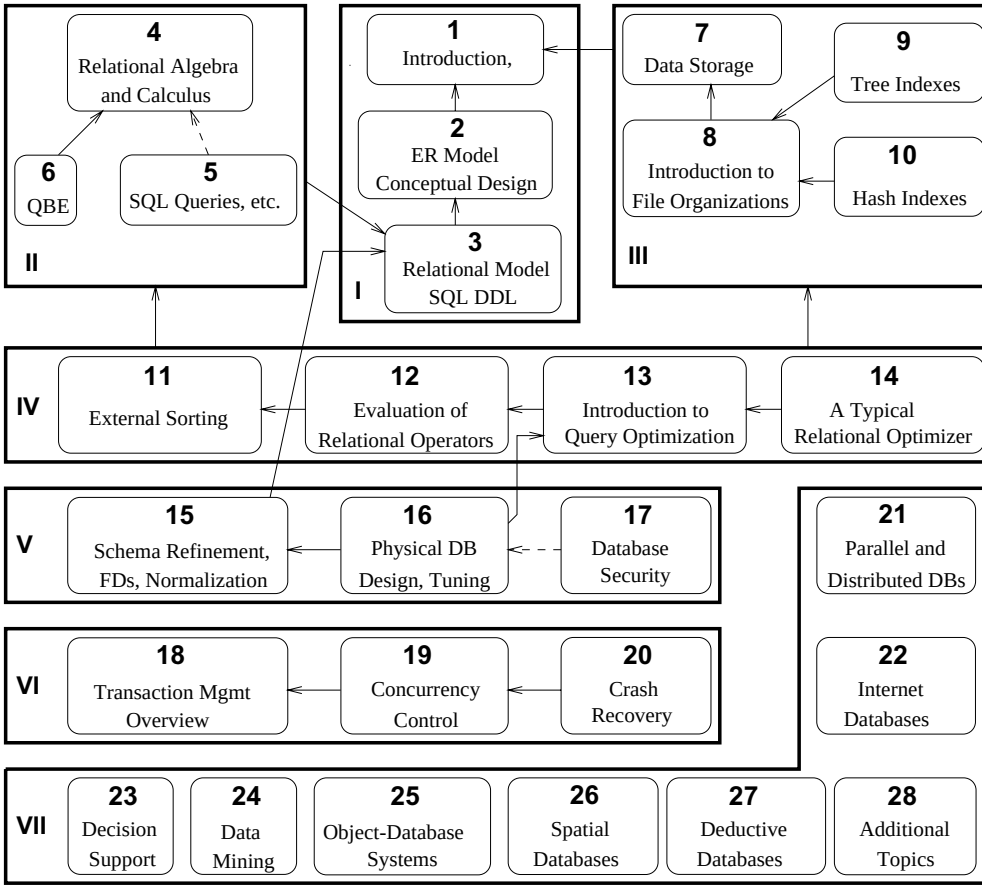

Figure 0.1 Chapter Organization and Dependencies

III early to space assignments. As another example, it is not necessary to cover all the alternatives for a given operator (e.g., various techniques for joins) in Chapter 12 in order to cover later related material (e.g., on optimization or tuning) adequately. The database design case study in the appendix can be discussed concurrently with the appropriate design chapters, or it can be discussed after all design topics have been covered, as a review.

Several section headings contain an asterisk. This symbol does not necessarily indicate a higher level of difficulty. Rather, omitting all asterisked sections leaves about the right amount of material in Chapters 1–18, possibly omitting Chapters 6, 10, and 14, for a broad introductory one-quarter or one-semester course (depending on the depth at which the remaining material is discussed and the nature of the course assignments).

The book can be used in several kinds of introductory or second courses by choosing topics appropriately, or in a two-course sequence by supplementing the material with some advanced readings in the second course. Examples of appropriate introductory courses include courses on file organizations and introduction to database management systems, especially if the course focuses on relational database design or implementation. Advanced courses can be built around the later chapters, which contain detailed bibliographies with ample pointers for further study.

Supplementary Material

Each chapter contains several exercises designed to test and expand the reader's understanding of the material. Students can obtain solutions to odd-numbered chapter exercises and a set of lecture slides for each chapter through the Web in Postscript and Adobe PDF formats.

The following material is available online to instructors:

1. Lecture slides for all chapters in MS Powerpoint, Postscript, and PDF formats.
2. Solutions to all chapter exercises.
3. SQL queries and programming assignments with solutions. (This is new for the second edition.)
4. Supplementary project software (Minibase) with sample assignments and solutions, as described in Appendix B. The text itself does not refer to the project software, however, and can be used independently in a course that presents the principles of database management systems from a practical perspective, but without a project component.

The supplementary material on SQL is new for the second edition. The remaining material has been extensively revised from the first edition versions.

For More Information

The home page for this book is at URL:

`http://www.cs.wisc.edu/~dbbook`

This page is frequently updated *and contains a link to all known errors in the book, the accompanying slides, and the supplements.* Instructors should visit this site periodically or register at this site to be notified of important changes by email.

Acknowledgments

This book grew out of lecture notes for CS564, the introductory (senior/graduate level) database course at UW-Madison. David DeWitt developed this course and the Minirel project, in which students wrote several well-chosen parts of a relational DBMS. My thinking about this material was shaped by teaching CS564, and Minirel was the inspiration for Minibase, which is more comprehensive (e.g., it has a query optimizer and includes visualization software) but tries to retain the spirit of Minirel. Mike Carey and I jointly designed much of Minibase. My lecture notes (and in turn this book) were influenced by Mike's lecture notes and by Yannis Ioannidis's lecture slides.

Joe Hellerstein used the beta edition of the book at Berkeley and provided invaluable feedback, assistance on slides, and hilarious quotes. Writing the chapter on object-database systems with Joe was a lot of fun.

C. Mohan provided invaluable assistance, patiently answering a number of questions about implementation techniques used in various commercial systems, in particular indexing, concurrency control, and recovery algorithms. Moshe Zloof answered numerous questions about QBE semantics and commercial systems based on QBE. Ron Fagin, Krishna Kulkarni, Len Shapiro, Jim Melton, Dennis Shasha, and Dirk Van Gucht reviewed the book and provided detailed feedback, greatly improving the content and presentation. Michael Goldweber at Beloit College, Matthew Haines at Wyoming, Michael Kifer at SUNY StonyBrook, Jeff Naughton at Wisconsin, Praveen Seshadri at Cornell, and Stan Zdonik at Brown also used the beta edition in their database courses and offered feedback and bug reports. In particular, Michael Kifer pointed out an error in the (old) algorithm for computing a minimal cover and suggested covering some SQL features in Chapter 2 to improve modularity. Gio Wiederhold's bibliography, converted to Latex format by S. Sudarshan, and Michael Ley's online bibliography on databases and logic programming were a great help while compiling the chapter bibliographies. Shaun Flisakowski and Uri Shaft helped me frequently in my never-ending battles with Latex.

I owe a special thanks to the many, many students who have contributed to the Minibase software. Emmanuel Ackaouy, Jim Pruyne, Lee Schumacher, and Michael Lee worked with me when I developed the first version of Minibase (much of which was subsequently discarded, but which influenced the next version). Emmanuel Ackaouy and Bryan So were my TAs when I taught CS564 using this version and went well beyond the limits of a TAship in their efforts to refine the project. Paul Aoki struggled with a version of Minibase and offered lots of useful comments as a TA at Berkeley. An entire class of CS764 students (our graduate database course) developed much of the current version of Minibase in a large class project that was led and coordinated by Mike Carey and me. Amit Shukla and Michael Lee were my TAs when I first taught CS564 using this version of Minibase and developed the software further.

Several students worked with me on independent projects, over a long period of time, to develop Minibase components. These include visualization packages for the buffer manager and B+ trees (Huseyin Bektas, Harry Stavropoulos, and Weiqing Huang); a query optimizer and visualizer (Stephen Harris, Michael Lee, and Donko Donjerkovic); an ER diagram tool based on the Opossum schema editor (Eben Haber); and a GUI-based tool for normalization (Andrew Prock and Andy Therber). In addition, Bill Kimmel worked to integrate and fix a large body of code (storage manager, buffer manager, files and access methods, relational operators, and the query plan executor) produced by the CS764 class project. Ranjani Ramamurty considerably extended Bill's work on cleaning up and integrating the various modules. Luke Blanshard, Uri Shaft, and Shaun Flisakowski worked on putting together the release version of the code and developed test suites and exercises based on the Minibase software. Krishna Kunchithapadam tested the optimizer and developed part of the Minibase GUI.

Clearly, the Minibase software would not exist without the contributions of a great many talented people. With this software available freely in the public domain, I hope that more instructors will be able to teach a systems-oriented database course with a blend of implementation and experimentation to complement the lecture material.

I'd like to thank the many students who helped in developing and checking the solutions to the exercises and provided useful feedback on draft versions of the book. In alphabetical order: X. Bao, S. Biao, M. Chakrabarti, C. Chan, W. Chen, N. Cheung, D. Colwell, C. Fritz, V. Ganti, J. Gehrke, G. Glass, V. Gopalakrishnan, M. Higgins, T. Jasmin, M. Krishnaprasad, Y. Lin, C. Liu, M. Lusignan, H. Modi, S. Narayanan, D. Randolph, A. Ranganathan, J. Reminga, A. Therber, M. Thomas, Q. Wang, R. Wang, Z. Wang, and J. Yuan. Arcady Grenader, James Harrington, and Martin Reames at Wisconsin and Nina Tang at Berkeley provided especially detailed feedback.

Charlie Fischer, Avi Silberschatz, and Jeff Ullman gave me invaluable advice on working with a publisher. My editors at McGraw-Hill, Betsy Jones and Eric Munson, obtained extensive reviews and guided this book in its early stages. Emily Gray and Brad Kosirog were there whenever problems cropped up. At Wisconsin, Ginny Werner really helped me to stay on top of things.

Finally, this book was a thief of time, and in many ways it was harder on my family than on me. My sons expressed themselves forthrightly. From my (then) five-year-old, Ketan: "Dad, stop working on that silly book. You don't have any time for *me*." Two-year-old Vivek: "You working *boook*? No no no come play basketball me!" All the seasons of their discontent were visited upon my wife, and Apu nonetheless cheerfully kept the family going in its usual chaotic, happy way all the many evenings and weekends I was wrapped up in this book. (Not to mention the days when I was wrapped up in being a faculty member!) As in all things, I can trace my parents' hand in much of this; my father, with his love of learning, and my mother, with her love of us, shaped me. My brother Kartik's contributions to this book consisted chiefly of

phone calls in which he kept me from working, but if I don't acknowledge him, he's liable to be annoyed. I'd like to thank my family for being there and giving meaning to everything I do. (There! I knew I'd find a legitimate reason to thank Kartik.)

Acknowledgments for the Second Edition

Emily Gray and Betsy Jones at McGraw-Hill obtained extensive reviews and provided guidance and support as we prepared the second edition. Jonathan Goldstein helped with the bibliography for spatial databases. The following reviewers provided valuable feedback on content and organization: Liming Cai at Ohio University, Costas Tsatsoulis at University of Kansas, Kwok-Bun Yue at University of Houston, Clear Lake, William Grosky at Wayne State University, Sang H. Son at University of Virginia, James M. Slack at Minnesota State University, Mankato, Herman Balsters at University of Twente, Netherlands, Karen C. Davis at University of Cincinnati, Joachim Hammer at University of Florida, Fred Petry at Tulane University, Gregory Speegle at Baylor University, Salih Yurttas at Texas A&M University, and David Chao at San Francisco State University.

A number of people reported bugs in the first edition. In particular, we wish to thank the following: Joseph Albert at Portland State University, Han-yin Chen at University of Wisconsin, Lois Delcambre at Oregon Graduate Institute, Maggie Eich at Southern Methodist University, Raj Gopalan at Curtin University of Technology, Davood Rafiei at University of Toronto, Michael Schrefl at University of South Australia, Alex Thomasian at University of Connecticut, and Scott Vandenberg at Siena College.

A special thanks to the many people who answered a detailed survey about how commercial systems support various features: At IBM, Mike Carey, Bruce Lindsay, C. Mohan, and James Teng; at Informix, M. Muralikrishna and Michael Ubell; at Microsoft, David Campbell, Goetz Graefe, and Peter Spiro; at Oracle, Hakan Jacobsson, Jonathan D. Klein, Muralidhar Krishnaprasad, and M. Ziauddin; and at Sybase, Marc Chanliau, Lucien Dimino, Sangeeta Doraiswamy, Hanuma Kodavalla, Roger MacNicol, and Tirumanjanam Rengarajan.

After reading about himself in the acknowledgment to the first edition, Ketan (now 8) had a simple question: "How come you didn't dedicate the book to us? Why mom?" Ketan, I took care of this inexplicable oversight. Vivek (now 5) was more concerned about the extent of his fame: "Daddy, is my name in *evvy* copy of your book? Do they have it in *evvy* compooter science department in the world?" Vivek, I hope so. Finally, this revision would not have made it without Apu's and Keiko's support.