Databases and Transaction Processing: An Application-Oriented Approach

by Philip M. Lewis, Arthur Bernstein, and Michael Kifer

Publisher: Addison Wesley, Copyright: 2002, 1014 pp, ISBN: 0-201-70872-8
Published 07/24/2001, Retail Price: $91.00 US

Reviewer: Alfons Kemper, University of Passau, 94030 Passau, Germany,
kemper@db.fmi.uni-passau.de

Summary of the Book

This book *Databases and Transaction Processing* constitutes a standard database textbook for advanced undergraduate and graduate courses—albeit with a somewhat different focus compared to the established books. As the subtitle *An Application-Oriented Approach* indicates, the authors put an emphasis on teaching the systematic usage of database systems—rather than concentrating on an in-depth coverage of implementation techniques for building database systems. The rationale for this focus is that many more students will be implementing applications than will actually be implementing database systems. Therefore, the authors set out to present the conceptual and engineering principles underlying the design and implementation of database and transaction processing applications. In order to provide a solid foundation for these principles, the book thoroughly presents the concepts and the theory underlying relational databases, conceptual/relational data modelling, and relational query languages.

This book covers the material traditionally taught in an introductory course in database systems: the relational model, conceptual design, query languages (including SQL, QBE, relational algebra, and relational calculus), normalization theory, and query processing.

In addition to these core database topics, the book also contains advanced database material often not covered in introductory texts and courses. Object databases, XML, and document processing on the web are given extensive coverage. Together with additional material on distributed databases, CORBA, OLAP, and data mining, the latter part of the book provides the basis for an advanced course on databases.

The text also covers transaction processing in depth such that an entire course can be devised based on this book—maybe with some supplementary material from dedicated textbooks on this subject (e.g., Weikum and Vossen's book which was reviewed by Marc Scholl in the preceding SIGMOD record). In addition to discussing issues related to building transactions and ACID properties, the book describes TP monitors, the architecture of centralized and distributed transaction processing systems, and the transactional issues related to building correct and efficient database applications.

A Walk Through the Contents

The contents of this ("heavy" 1000+ page) book is organized into four parts. The first part is used to motivate and overview database systems and to lay the foundations for the student registration system, that is used as a running example throughout the book. The second part constitutes the core of a traditional course on database management systems. It covers all the traditional topics ranging from data modelling, normalization theory, query languages to basic physical data organization techniques, query optimization and transaction control. Besides SQL, the authors cover the (currently?) most important programming interfaces to relational database systems, such as SQL:1999 triggers, Embedded SQL, Dynamic SQL, ODBC, JDBC, and SQLJ.

The third part is dedicated to advanced database topics like object-orientation, XML, distributed databases and OLAP. The object-relational concepts are presented in SQL:1999 syntax—rather than relying on a particular commercial DBMS's syntax.

Finally, the fourth part covers transaction processing in depth. Here, the book covers the basic implementation techniques as well as application-oriented topics like architectural and security issues.

The division of the material into those four parts indicates the intended usage of the book for several different courses. While part one and two are typically covered in an undergraduate introductory course, the third part could serve as a basis for a graduate course emphasizing the advanced use of database systems in various application domains (Internet, distributed systems, business administration). Part four is intended for a course on transaction processing systems. Again, the authors have included application-oriented topics to prepare students for advanced usage of database technology in distributed, high-performance and Internet/electronic commerce systems.

Recommendation

This book *Databases and Transaction Processing* is very well written and provides a complete and clear explanation of the conceptual and engineering principles underlying the design and implementation of database and transaction processing applications. Rather than focusing on how to implement the database management system itself, this text focuses on how to build database applications. To illustrate both database and transaction processing concepts from an application-oriented view point, a case study is carried throughout the book. The technical aspects of each chapter applied to the case study and the software engineering concepts required to implement the case study are discussed.

Almost ten years ago, in the SIGMOD 1993 conference in Washington, Ami Motro arranged a panel on how to teach databases. Already then (and more so in following years) the database researchers saw DBMSs as ubiquitous components of any large software system. This is all the more true for almost any kind of Web application. Therefore, it is essential that we adjust our approach in teaching database systems accordingly. Only a small percentage of our students will actually work for database vendors and build database systems. Most of our Bachelor and Master's students will work on software systems that utilize databases. Therefore, I do welcome this explicit application-oriented focus of the book. Certainly, almost all other database textbooks have gone through a transition and have incorporated more material on database system applications. However, only a few go as far as this book by developing an example application throughout most parts of the book.

The authors could have gone even further and incorporate more software engineering material into the text. In particular, I am surprised about the omission of UML (Unified Modelling Language). In most companies I work with, UML has become the de-facto standard tool for designing object-oriented software systems. Therefore, I would have liked to see UML covered as an alternative to the Entity-Relationship Model. There is even a book on how to use UML for database design (Database Design for Smarties: Using UML for Data Modelling, Robert J. Muller, Morgan Kaufmann, 1999).

Overall, I recommend this book as a basis for a very solid application-oriented database education. It covers several courses, an introductory one and two advanced courses which could be supplemented by additional, more specialized books.