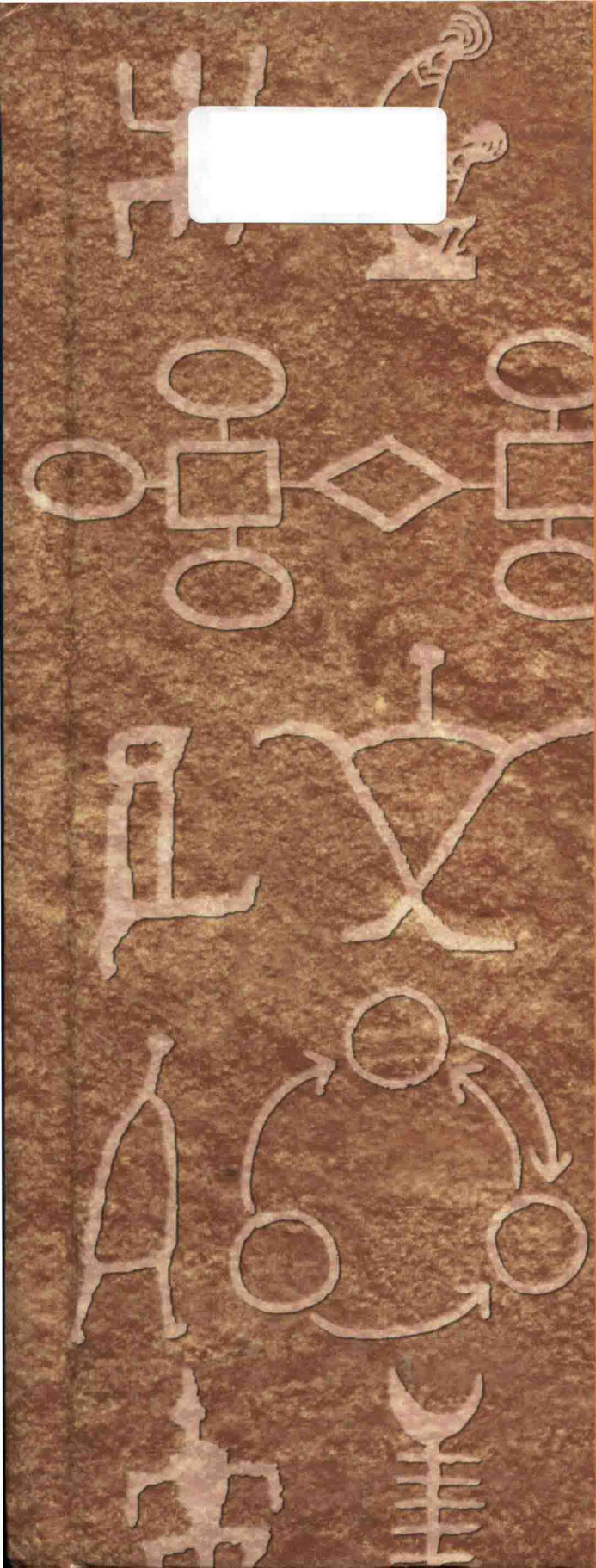




DATABASE SYSTEMS

THE COMPLETE BOOK

SECOND EDITION

Hector Garcia-Molina
Jeffrey D. Ullman
Jennifer Widom



Upper Saddle River, NJ 07458
www.prenhall.com

ISBN-13: 978-0-13-187325-4
ISBN-10: 0-13-187325-3

EAN



Garcia-Molina
Ullman
Widom

DATA BASE SYSTEMS

THE COMPLETE BOOK

SECOND
EDITION



DATABASE SYSTEMS

The Complete Book

Second Edition

Hector Garcia-Molina

Jeffrey D. Ullman

Jennifer Widom

*Department of Computer Science
Stanford University*



Upper Saddle River, New Jersey 07458

Library of Congress Cataloging-in-Publication Data

CIP data on file

Editorial Director, Computer Science and Engineering: *Marcia J. Horton*
Executive Editor: *Tracy Dunkelberger*
Editorial Assistant: *Melinda Haggerty*
Director of Marketing: *Margaret Waples*
Marketing Manager: *Christopher Kelly*
Senior Managing Editor: *Scott Disanno*
Production Editor: *Irwin Zucker*
Art Director: *Jayne Conte*
Cover Designer: *Margaret Kenselaar*
Cover Art: *Tamara L. Newman*
Manufacturing Buyer: *Lisa McDowell*
Manufacturing Manager: *Alan Fischer*



© 2009, 2002 by Pearson Education Inc.
Pearson Prentice Hall
Pearson Education, Inc.
Upper Saddle River, NJ 07458

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Pearson Prentice Hall™ is a trademark of Pearson Education, Inc.

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

Printed in the United States of America

6 7 8 9 10 V092 16 15 14 13

ISBN 0-13-187325-3

978-0-13-187325-4

Pearson Education Ltd., *London*
Pearson Education Australia Pty. Ltd., *Sydney*
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd., *Hong Kong*
Pearson Education Canada, Inc., *Toronto*
Pearson Educación de México, S.A. de C.V.
Pearson Education—Japan, *Tokyo*
Pearson Education Malaysia, Pte. Ltd.
Pearson Education, Inc., *Upper Saddle River, New Jersey*

DATABASE SYSTEMS

The Complete Book

Preface

This book covers the core of the material taught in the database sequence at Stanford. The introductory course, CS145, uses the first twelve chapters, and is designed for all students — those who want to use database systems as well as those who want to get involved in database implementation. The second course, CS245 on database implementation, covers most of the rest of the book. However, some material is covered in more detail in special topics courses. These include CS346 (implementation project), which concentrates on query optimization as in Chapters 15 and 16. Also, CS345A, on data mining and Web mining, covers the material in the last two chapters.

What's New in the Second Edition

After a brief introduction in Chapter 1, we cover relational modeling in Chapters 2–4. Chapter 4 is devoted to high-level modeling. There, in addition to the E/R model, we now cover UML (Unified Modeling Language). We also have moved to Chapter 4 a shorter version of the material on ODL, treating it as a design language for relational database schemas.

The material on functional and multivalued dependencies has been modified and remains in Chapter 3. We have changed our viewpoint, so that a functional dependency is assumed to have a set of attributes on the right. We have also given explicitly certain algorithms, including the “chase,” that allow us to manipulate dependencies. We have augmented our discussion of third normal form to include the 3NF synthesis algorithm and to make clear what the tradeoff between 3NF and BCNF is.

Chapter 5 contains the coverage of relational algebra from the previous edition, and is joined by (part of) the treatment of Datalog from the old Chapter 10. The discussion of recursion in Datalog is either moved to the book's Web site or combined with the treatment of recursive SQL in Chapter 10 of this edition.

Chapters 6–10 are devoted to aspects of SQL programming, and they represent a reorganization and augmentation of the earlier book's Chapters 6, 7, 8, and parts of 10. The material on views and indexes has been moved to its own chapter, number 8, and this material has been augmented with a discussion of

important new topics, including materialized views, and automatic selection of indexes.

The new Chapter 9 is based on the old Chapter 8 (embedded SQL). It is introduced by a new section on 3-tier architecture. It also includes an expanded discussion of JDBC and new coverage of PHP.

Chapter 10 collects a number of advanced SQL topics. The discussion of authorization from the old Chapter 8 has been moved here, as has the discussion of recursive SQL from the old Chapter 10. Data cubes, from the old Chapter 20, are now covered here. The rest of the chapter is devoted to the nested-relation model (from the old Chapter 4) and object-relational features of SQL (from the old Chapter 9).

Then, Chapters 11 and 12 cover XML and systems based on XML. Except for material at the end of the old Chapter 4, which has been moved to Chapter 11, this material is all new. Chapter 11 covers modeling; it includes expanded coverage of DTD's, along with new material on XML Schema. Chapter 12 is devoted to programming, and it includes sections on XPath, XQuery, and XSLT.

Chapter 13 begins the study of database implementation. It covers disk storage and the file structures that are built on disks. This chapter is a condensation of material that, in the first edition, occupied Chapters 11 and 12.

Chapter 14 covers index structures, including B-trees, hashing, and structures for multidimensional indexes. This material also condenses two chapters, 13 and 14, from the first edition.

Chapters 15 and 16 cover query execution and query optimization, respectively. They are similar to the old chapters of the same numbers. Chapter 17 covers logging, and Chapter 18 covers concurrency control; these chapters are also similar to the old chapters with the same numbers. Chapter 19 contains additional topics on concurrency: recovery, deadlocks, and long transactions. This material is a subset of the old Chapter 19.

Chapter 20 is on parallel and distributed databases. In addition to material on parallel query execution from the old Chapter 15 and material on distributed locking and commitment from the old Chapter 19, there are several new sections on distributed query execution: the map-reduce framework for parallel computation, peer-to-peer databases and their implementation of distributed hash tables.

Chapter 21 covers information integration. In addition to material on this subject from the old Chapter 20, we have added a section on local-as-view mediators and a section on entity resolution (finding records from several databases that refer to the same entity, e.g., a person).

Chapter 22 is on data mining. Although there was some material on the subject in the old Chapter 20, almost all of this chapter is new. It covers association rules and frequent itemset mining, including both the famous A-Priori Algorithm and certain efficiency improvements. Chapter 22 includes the key techniques of shingling, minhashing, and locality-sensitive hashing for finding similar items in massive databases, e.g., Web pages that quote substantially

from other Web pages. The chapter concludes with a study of clustering, especially for massive datasets.

Chapter 23, all new, addresses two important ways in which the Internet has impacted database technology. First is search engines, where we discuss algorithms for crawling the Web, the well-known PageRank algorithm for evaluating the importance of Web pages, and its extensions. This chapter also covers data-stream-management systems. We discuss the stream data model and SQL language extensions, and conclude with several interesting algorithms for executing queries on streams.

Prerequisites

We have used the book at the “mezzanine” level, in a sequence of courses taken both by undergraduates and by beginning graduate students. The formal prerequisites for the course are Sophomore-level treatments of:

1. Data structures, algorithms, and discrete math, and
2. Software systems, software engineering, and programming languages.

Of this material, it is important that students have at least a rudimentary understanding of such topics as: algebraic expressions and laws, logic, basic data structures, object-oriented programming concepts, and programming environments. However, we believe that adequate background is acquired by the Junior year of a typical computer science program.

Exercises

The book contains extensive exercises, with some for almost every section. We indicate harder exercises or parts of exercises with an exclamation point. The hardest exercises have a double exclamation point.

Support on the World Wide Web

The book’s home page is

<http://infolab.stanford.edu/~ullman/dscb.html>

You will find errata as we learn of them, and backup materials, including homeworks, projects, and exams. We shall also make available there the sections from the first edition that have been removed from the second.

In addition, there is an accompanying set of on-line homeworks and programming labs using a technology developed by Gradiance Corp. See the section following the Preface for details about the GOAL system. GOAL service

can be purchased at <http://www.prenhall.com/goal>. Instructors who want to use the system in their classes should contact their Prentice-Hall representative or request instructor authorization through the above Web site.

There is a solutions manual for instructors available at

<http://www.prenhall.com/ullman>

This page also gives you access to GOAL and all book materials.

Acknowledgements

We would like to thank Donald Kossmann for helpful discussions, especially concerning XML and its associated programming systems. Also, Bobbie Cochrane assisted us in understanding trigger semantics for a earlier edition.

A large number of people have helped us, either with the development of this book or its predecessors, or by contacting us with errata in the books and/or other Web-based materials. It is our pleasure to acknowledge them all here.

Marc Abromowitz, Joseph H. Adamski, Brad Adelberg, Gleb Ashimov, Donald Aingworth, Teresa Almeida, Brian Babcock, Bruce Baker, Yunfan Bao, Jonathan Becker, Margaret Benitez, Eberhard Bertsch, Larry Bonham, Phillip Bonnet, David Brokaw, Ed Burns, Alex Butler, Karen Butler, Mike Carey, Christopher Chan, Sudarshan Chawathe.

Also Per Christensen, Ed Chang, Surajit Chaudhuri, Ken Chen, Rada Chirkova, Nitin Chopra, Lewis Church, Jr., Bobbie Cochrane, Michael Cole, Alissa Cooper, Arturo Crespo, Linda DeMichiel, Matthew F. Dennis, Tom Dienstbier, Pearl D'Souza, Oliver Duschka, Xavier Faz, Greg Fichtenholtz, Bart Fisher, Simon Frettloeh, Jarl Friis.

Also John Fry, Chiping Fu, Tracy Fujieda, Prasanna Ganesan, Suzanne Garcia, Mark Gjøl, Manish Godara, Seth Goldberg, Jeff Goldblat, Meredith Goldsmith, Luis Gravano, Gerard Guillemette, Himanshu Gupta, Petri Gynther, Zoltan Gyongyi, Jon Heggland, Rafael Hernandez, Masanori Higashihara, Antti Hjelt, Ben Holtzman, Steve Huntsberry.

Also Sajid Hussain, Leonard Jacobson, Thulasiraman Jeyaraman, Dwight Joe, Brian Jorgensen, Mathew P. Johnson, Sameh Kamel, Jawed Karim, Seth Katz, Pedram Keyani, Victor Kimeli, Ed Knorr, Yeong-Ping Koh, David Koller, Gyorgy Kovacs, Phillip Koza, Brian Kulman, Bill Labiosa, Sang Ho Lee, Young-han Lee, Miguel Licon.

Also Olivier Lobry, Chao-Jun Lu, Waynn Lue, John Manz, Arun Marathe, Philip Minami, Le-Wei Mo, Fabian Modoux, Peter Mork, Mark Mortensen, Ramprakash Narayanaswami, Hankyung Na, Mor Naaman, Mayur Naik, Marie Nilsson, Torbjorn Norbye, Chang-Min Oh, Mehul Patel, Soren Peen, Jian Pei.

Also Xiaobo Peng, Bert Porter, Limbek Reka, Prahash Ramanan, Nisheeth Ranjan, Suzanne Rivoire, Ken Ross, Tim Roughgarden, Mema Roussopoulos, Richard Scherl, Loren Shevitz, Shrikrishna Shrin, June Yoshiko Sison,

Man Cho A. So, Elizabeth Stinson, Qi Su, Ed Swierk, Catherine Tornabene, Anders Uhl, Jonathan Ullman, Mayank Upadhyay.

Also Anatoly Varakin, Vassilis Vassalos, Krishna Venuturimilli, Vikram Vijayaraghavan, Terje Viken, Qiang Wang, Steven Whang, Mike Wiacek, Kristian Widjaja, Janet Wu, Sundar Yamunachari, Takeshi Yokukawa, Bing Yu, Min-Sig Yun, Torben Zahle, Sandy Zhang.

The remaining errors are ours, of course.

H. G.-M.
J. D. U.
J. W.
Stanford, CA
March, 2008

GOAL

Gradiane Online Accelerated Learning (GOAL) is Pearson's premier online homework and assessment system. GOAL is designed to minimize student frustration while providing an interactive teaching experience outside the classroom. (Visit www.prenhall.com/goal for a demonstration and additional information.)

With GOAL's immediate feedback and book-specific hints and pointers, students will have a more efficient and effective learning experience. GOAL delivers immediate assessment and feedback via two kinds of assignments: multiple choice homework exercises and interactive lab projects.

The homework consists of a set of multiple choice questions designed to test student knowledge of a solved problem. When answers are graded as incorrect, students are given a hint and directed back to a specific section in the course textbook for helpful information. Note: Students that are not enrolled in a class may want to enroll in a "Self-Study Course" that allows them to complete the homework exercises on their own.

Unlike syntax checkers and compilers, GOAL's lab projects check for both syntactic and semantic errors. GOAL determines if the student's program runs but more importantly, when checked against a hidden data set, verifies that it returns the correct result. By testing the code and providing immediate feedback, GOAL lets you know exactly which concepts the students have grasped and which ones need to be revisited.

In addition, the GOAL package specific to this book includes programming exercises in SQL and XQuery. Submitted queries are tested for correctness and incorrect results lead to examples of where the query goes wrong. Students can try as many times as they like but writing queries that respond correctly to the examples is not sufficient to get credit for the problem.

Instructors should contact their local Pearson Sales Representative for sales and ordering information for the GOAL Student Access Code and textbook value package.

About the Authors

HECTOR GARCIA-MOLINA is the L. Bosack and S. Lerner Professor of Computer Science and Electrical Engineering at Stanford University. His research interests include digital libraries, information integration, and database application on the Internet. He was a recipient of the SIGMOD Innovations Award and a member of PITAC (President's Information-Technology Advisory Council). He currently serves on the Board of Directors of Oracle Corp.

JEFFREY D. ULLMAN is the Stanford W. Ascherman Professor of Computer Science (emeritus) at Stanford University. He is the author or co-author of 16 books, including *Elements of ML Programming* (Prentice Hall 1998). His research interests include data mining, information integration, and electronic education. He is a member of the National Academy of Engineering, and recipient of a Guggenheim Fellowship, the Karl V. Karlstrom Outstanding Educator Award, the SIGMOD Contributions and Edgar F. Codd Innovations Awards, and the Knuth Prize.

JENNIFER WIDOM is Professor of Computer Science and Electrical Engineering at Stanford University. Her research interests span many aspects of nontraditional data management. She is an ACM Fellow and a member of the National Academy of Engineering, she received the ACM SIGMOD Edgar F. Codd Innovations Award in 2007 and was a Guggenheim Fellow in 2000, and she has served on a variety of program committees, advisory boards, and editorial boards.

DATABASE SYSTEMS

The Complete Book

Table of Contents

1	The Worlds of Database Systems	1
1.1	The Evolution of Database Systems	1
1.1.1	Early Database Management Systems	2
1.1.2	Relational Database Systems	3
1.1.3	Smaller and Smaller Systems	3
1.1.4	Bigger and Bigger Systems	4
1.1.5	Information Integration	4
1.2	Overview of a Database Management System	5
1.2.1	Data-Definition Language Commands	5
1.2.2	Overview of Query Processing	5
1.2.3	Storage and Buffer Management	7
1.2.4	Transaction Processing	8
1.2.5	The Query Processor	9
1.3	Outline of Database-System Studies	10
1.4	References for Chapter 1	12
I	Relational Database Modeling	15
2	The Relational Model of Data	17
2.1	An Overview of Data Models	17
2.1.1	What is a Data Model?	17
2.1.2	Important Data Models	18
2.1.3	The Relational Model in Brief	18
2.1.4	The Semistructured Model in Brief	19
2.1.5	Other Data Models	20
2.1.6	Comparison of Modeling Approaches	21
2.2	Basics of the Relational Model	21
2.2.1	Attributes	22
2.2.2	Schemas	22
2.2.3	Tuples	22
2.2.4	Domains	23
2.2.5	Equivalent Representations of a Relation	23

2.2.6	Relation Instances	24
2.2.7	Keys of Relations	25
2.2.8	An Example Database Schema	26
2.2.9	Exercises for Section 2.2	28
2.3	Defining a Relation Schema in SQL	29
2.3.1	Relations in SQL	29
2.3.2	Data Types	30
2.3.3	Simple Table Declarations	31
2.3.4	Modifying Relation Schemas	33
2.3.5	Default Values	34
2.3.6	Declaring Keys	34
2.3.7	Exercises for Section 2.3	36
2.4	An Algebraic Query Language	38
2.4.1	Why Do We Need a Special Query Language?	38
2.4.2	What is an Algebra?	38
2.4.3	Overview of Relational Algebra	39
2.4.4	Set Operations on Relations	39
2.4.5	Projection	41
2.4.6	Selection	42
2.4.7	Cartesian Product	43
2.4.8	Natural Joins	43
2.4.9	Theta-Joins	45
2.4.10	Combining Operations to Form Queries	47
2.4.11	Naming and Renaming	49
2.4.12	Relationships Among Operations	50
2.4.13	A Linear Notation for Algebraic Expressions	51
2.4.14	Exercises for Section 2.4	52
2.5	Constraints on Relations	58
2.5.1	Relational Algebra as a Constraint Language	59
2.5.2	Referential Integrity Constraints	59
2.5.3	Key Constraints	60
2.5.4	Additional Constraint Examples	61
2.5.5	Exercises for Section 2.5	62
2.6	Summary of Chapter 2	63
2.7	References for Chapter 2	65
3	Design Theory for Relational Databases	67
3.1	Functional Dependencies	67
3.1.1	Definition of Functional Dependency	68
3.1.2	Keys of Relations	70
3.1.3	Superkeys	71
3.1.4	Exercises for Section 3.1	71
3.2	Rules About Functional Dependencies	72
3.2.1	Reasoning About Functional Dependencies	72
3.2.2	The Splitting/Combining Rule	73