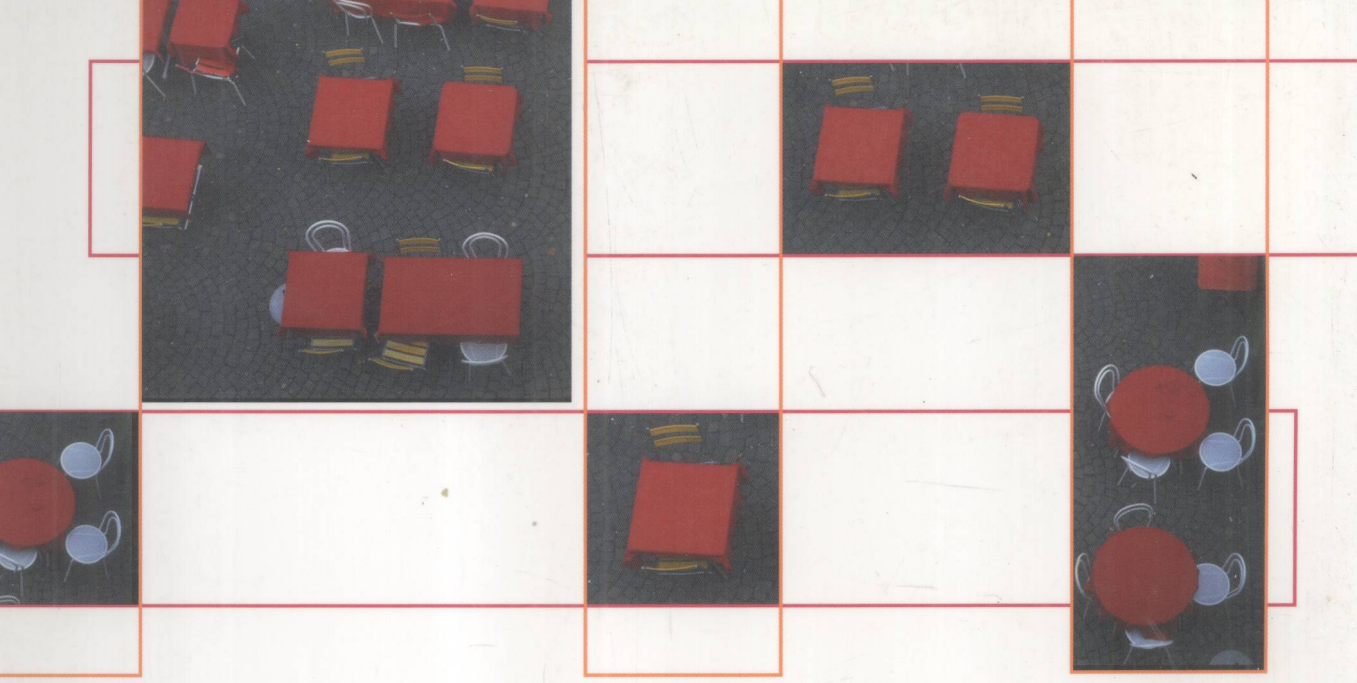

relational database design and implementation

THIRD EDITION

JAN L. HARRINGTON

MORGAN KAUFMANN

Relational Database Design and Implementation: Clearly Explained

Third Edition

Jan L. Harrington

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Morgan Kaufmann Publishers is an imprint of Elsevier

MORGAN KAUFMANN PUBLISHERS

AN IMPRINT OF ELSEVIER SCIENCE

Morgan Kaufmann Publishers is an imprint of Elsevier.
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA

This book is printed on acid-free paper.

Library of Congress Cataloging-in-Publication Data

Harrington, Jan L.
Relational database design and implementation : clearly explained / Jan L. Harrington.—3rd ed.
p. cm.
Rev. ed of: Relational database design clearly explained, 1998.
Includes bibliographical references and index.
ISBN 978-0-12-374730-3
1. Relational databases. 2. Database design. I. Harrington, Jan L. Relational database design clearly explained. II. Title.
QA76.9.D26H38 2009
005.75'6—dc22 2009022380

ISBN: 978-0-12-374730-3

For information on all Morgan Kaufmann publications,

visit our Web site at www.mkp.com or www.elsevierdirect.com

Printed in the United States of America

09 10 11 12 13 5 4 3 2 1

Relational Database Design and Implementation

Third Edition

Preface to the Third Edition

My favorite opening line for the database courses I teach is "Probably the most misunderstood term in all of business computing is *database*, followed closely by the word *relational*." At that point, some students look a bit smug because they are absolutely, positively sure that they know what a database is and that they also know what is means for a database to be "relational." Unfortunately, the popular press, with the help of some PC software developers, long ago distorted the meaning of both terms, which led many businesses to think that designing a database is a task that could be left to any clerical worker who had taken a one-week course on using database software. As you will see throughout this book, however, nothing could be further from the truth.

Note: The media has given us a number of nonsense computer terms such as *telephone modem* (we're modulating an analog signal, not a telephone), *software program* (the two words mean pretty much the same thing), and *cable modem* and *DSL modem* (they're not modems; they don't modulate and demodulate analog signals; they are more properly termed *codecs* that code and decode digital signals). It's all in an attempt to make computer jargon easier for people to understand, but it has generally had the effect of introducing misunderstandings.

This book is intended for anyone who has been given the responsibility for designing or maintaining a relational database. It will teach you how to look at the environment your database serves and to tailor the design of the database to the environment. It will also teach you how to design the database so it provides accurate and consistent data, avoiding the problems that are common to poorly designed databases. In addition, you will learn about design compromises that you might choose to make in the interest of database application performance and the consequences of making such choices.

If you are a college instructor, you may choose to use this book as a text in an undergraduate database management course. I've been doing that for a number of years (along with *SQL Clearly Explained*, this book's companion volume) and find that students learn from it quite well. They appreciate the straightforward language rather than a text that forces them to struggle with overly academic sentence structures. They also like the many real-world examples that appear throughout the book.

Changes in the Third Edition

The core of this book—Parts II and III, the bulk of the content of the previous editions—remains mostly unchanged from the second edition. Relational database theory has been relatively stable for more than 30 years (with the exception of the addition of sixth normal form) and requires very little updating from one edition to the next, although

it has been seven years since the second edition appeared. The major changes are the discussions of fifth and sixth normal forms. The first two case studies in Part III have been updated; the third case study is new.

The chapter on object-relational databases has been removed from this edition, as well as object-relational examples in the case studies. There are two reasons for this. First, support for objects within a relational environment has largely been provided as a part of the SQL standard rather than as changes to underlying relational database theory. Second, the direction that SQL's object-relational capabilities have taken since the second edition appeared involves a number of features that violate relational design theory, and presenting them in any depth in this book would be more confusing than helpful.

By far the biggest change, however, is the addition of the new Parts I and IV. Part I contains three chapters that provide a context for database design. Database requirements don't magically appear at the point an organization needs a database, although looking at the previous editions of this book, you might think they did. Chapter 1 presents several organizational aspects of database management, including the hardware architectures on which today's databases run, and a look at service-oriented architecture (SOA), an information systems technique in which databases, like other IT functions, become services provided throughout an organization.

Chapter 2 provides an overview of several systems analysis methods to show you how organizations arrive at database requirements. In Chapter 3 you'll discover why we care about good database design. (It really *does* matter!)

Part IV provides an overview of a variety of database implementation issues that you may need to consider as you design a relational database. The topics include concurrency control (keeping the database consistent while multiple users interact with it at the same time), data warehousing (understanding issues that may arise when your operational database data are destined for data mining), data quality (ensuring that data are as accurate and consistent as possible), and XML (understanding how today's databases support XML).

The addition of Parts I and IV also make this book better suited for use as a textbook in a college course. When I used the second edition as a text in my classes, I added supplementary readings to cover that material. It's nice to have it all in once place!

The material about older data models that was presented in Chapter 3 in the second edition has been moved into an appendix. None of the material in the body of the book depends on it any longer. You can read it if you are interested in knowing what preceded the relational data model, but you won't lose anything significant in terms of relational databases if you skip it.

What You Need to Know

When the first edition of this book appeared in 1999, you needed only basic computer literacy to understand just about everything the book discussed. The role of networking in database architectures has grown so much in the past decade that in addition to computer literacy, you now need to understand some basic network hardware and software concepts (e.g., the Internet, interconnection devices such as routers and switches, and servers).

Note: It has always been a challenge to decide whether to teach students about systems analysis and design before or after database management. Now we worry about where a networking course should come in the sequence. It's tough to understand databases without networking, but at the same time, some aspects of networking involve database issues.

Acknowledgments

As always, getting this book onto paper involved an entire cast of characters, all of whom deserve thanks for their efforts. First are the people at Morgan Kaufmann:

- Rick Adams, my editor of many years. (His official title is Senior Acquisitions Editor).
- Heather Scherer, Rick's capable assistant
- Marilyn Rash, the project manager. We've worked together on a number of books over many years and it's always a pleasure.
- Eric DeCicco, the designer of the wonderful cover.
- The folks who clean up after me: Debbie Prato, copyeditor, and Samantha Molineaux, proofreader.
- Ted Laux, the indexer.
- Greg deZam-O'Hare and Sarah Binns who pulled it all together at the end.

A special thanks goes out to my colleague, Dr. Craig Fisher, who is a well-known expert on data quality. He provided me with a wealth of resources on that topic, which he thinks should be a part of everyone's IT education.

JLH

Contents

PART II DATABASE DESIGN THEORY