Information Modeling and Relational Databases

Second Edition

The Morgan Kaufmann Series in Data Management Systems (Selected Titles)

Information Modeling and Relational Databases, 2nd Edition
Terry Halpin, Tony Morgan

Joe Celko's Thinking in Sets
Joe Celko

Business Metadata
Bill Inmon, Bonnie O'Neil, Lowell Fryman

Unleashing Web 2.0
Gottfried Vossen, Stephan Hagemann

Enterprise Knowledge Management
David Loshin

Business Process Change, 2nd Edition
Paul Harmon

IT Manager's Handbook, 2nd Edition
Bill Holtsnider & Brian Jaffe

Joe Celko's Puzzles and Answers, 2nd Edition
Joe Celko

Making Shoes for the Cobbler's Children
Charles Betz

Joe Celko's Analytics and OLAP in SQL
Joe Celko

Data Preparation for Data Mining Using SAS
Mamdouh Refaat

Querying XML: XQuery, XPath, and SQL/XML in Context
Jim Melton and Stephen Buxton

Data Mining: Concepts and Techniques, 2nd Edition
Jiawei Han and Micheline Kamber

Database Modeling and Design: Logical Design, 4th Edition
Toby J, Teorey, Sam S. Lightstone, Thomas P. Nadeau

Foundations of Multidimensional and Metric Data Structures
Hanan Samet

Joe Celko's SQL for Smarties: Advanced SQL Programming, 3rd Edition
Joe Celko

Moving Objects Databases
Ralf Hartmut Güting and Markus Schneider

Joe Celko's SQL Programming Style
Joe Celko

Data Mining, Second Edition: Concepts and Techniques
Ian Witten and Eibe Frank

Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration
Earl Cox

Data Modeling Essentials, 3rd Edition
Graeme C. Simsion and Graham C. Witt

Location-Based Services
Jochen Schiller and Agnès Voisard

Database Modeling with Microsoft® Visio for Enterprise Architects
Terry Halpin, Ken Evans, Patrick Hallock, Bill Maclean

Designing Data-Intensive Web Applications
Stephano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Sara Comai, Maristella Matera

Mining the Web: Discovering Knowledge from Hypertext Data
Soumen Chakrabarti

Advanced SQL: 1999—Understanding Object-Relational and Other Advanced Features
Jim Melton

Database Tuning: Principles, Experiments, and Troubleshooting Techniques
Dennis Shasha, Philippe Bonnet

SQL:1999—Understanding Relational Language Components
Jim Melton, Alan R. Simon

Information Visualization in Data Mining and Knowledge Discovery
Edited by Usama Fayyad, Georges G. Grinstein, Andreas Wierse

Transactional Information Systems
Gerhard Weikum and Gottfried Vossen

Spatial Databases
Philippe Rigaux, Michel Scholl, and Agnes Voisard

Managing Reference Data in Enterprise Database
Malcolm Chisholm

Understanding SQL and Java Together
Jim Melton and Andrew Eisenberg

Database: Principles, Programming, and Performance, 2nd Edition
Patrick and Elizabeth O'Neil

The Object Data Standar
Edited by R. G. G. Cattell, Douglas Barry

Data on the Web: From Relations to Semistructured Data and XML
Serge Abiteboul, Peter Buneman, Dan Suciu

Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations
Ian Witten, Eibe Frank

Joe Celko's Data and Databases: Concepts in Practice
Joe Celko

Developing Time-Oriented Database Applications in SQL
Richard T. Snodgrass

Web Farming for the Data Warehouse
Richard D. Hackathorn

Management of Heterogeneous and Autonomous Database Systems
Edited by Ahmed Elmagarmid, Marek Rusinkiewicz, Amit Sheth

Object-Relational DBMSs: 2nd Edition
Michael Stonebraker and Paul Brown,with Dorothy Moore

Universal Database Management: A Guide to Object/Relational Technology
Cynthia Maro Saracco

Readings in Database Systems, 3rd Edition
Edited by Michael Stonebraker, Joseph M. Hellerstein

Understanding SQL's Stored Procedures: A Complete Guide to SQL/PSM
Jim Melton

Principles of Multimedia Database Systems
V. S. Subrahmanian

Principles of Database Query Processing for Advanced Applications
Clement T. Yu, Weiyi Meng

Advanced Database Systems
Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T. Snodgrass, V. S. Subrahmanian, Roberto Zicari

Principles of Transaction Processing
Philip A. Bernstein, Eric Newcomer

Using the New DB2: IBMs Object-Relational Database System
Don Chamberlin

Distributed Algorithms
Nancy A. Lynch

Active Database Systems: Triggers and Rules For Advanced Database Processing
Edited by Jennifer Widom, Stefano Ceri

Migrating Legacy Systems: Gateways, Interfaces, & the Incremental Approach
Michael L. Brodie, Michael Stonebraker

Atomic Transactions
Nancy Lynch, Michael Merritt, William Weihl, Alan Fekete

Query Processing for Advanced Database Systems
Edited by Johann Christoph Freytag, David Maier, Gottfried Vossen

Transaction Processing
Jim Gray, Andreas Reuter

Database Transaction Models for Advanced Applications
Edited by Ahmed K. Elmagarmid

A Guide to Developing Client/Server SQL Applications
Setrag Khoshafian, Arvola Chan, Anna Wong, Harry K. T. Wong

Information Modeling and Relational Databases

Second Edition

Terry Halpin
Neumont University

Tony Morgan
Neumont University

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

Publishing Director	Chris Williams
Acquisitions Editor	Denise E. M. Penrose
Publishing Services Manager	George Morrison
Project Manager	Andre A. Cuello
Assistant Editor	Mary James
Composition	Terry Halpin and Tony Morgan
Interior printer	Sheridan Books, Inc.
Cover printer	Phoenix Color
Cover Design	Alisa Andreola

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
(Application Submitted)

ISBN: 978-0-12-373568-3

For information on all Morgan Kaufmann publications,
visit our Web site at www.mkp.com or www.books.elsevier.com

Printed in the United States of America
08 09 10 11 12 10 9 8 7 6 5 4 3 2 1

To Norma and Gwen, our wonderful wives.

Terry and Tony

Contents

Foreword

by John Zachman, Founder and President
Zachman International

It gives me great personal pleasure to write this foreword. I wrote the foreword to the first edition of *Information Modeling and Relational Databases* and to be brutally honest, I liked my first foreword and I haven't at all changed my mind, with the exception that I like the second edition even more than the first edition, if that is even possible. Anyone familiar with my work would know that I have been arguing for many years that an enterprise ontology must include more structural components than those typically related to information. Terry Halpin and Tony Morgan have incorporated some additional structural variables in this new edition.

I suppose you would have expected this, but the second edition even surpasses the first, not only in terms of the updated and expanded modeling coverage, now including XML, business processes, and even the Semantic Web, and the plethora of exercises, but in terms of the significance of seven more years of experience and wisdom that can only be accumulated through the concentrated and intense investment of one's life.

Because I liked my first Foreword, it is hard for me to materially improve on it, so I will borrow heavily from its basic content, making adjustments as appropriate.

I have known Terry Halpin for many years. I have known *about* Terry Halpin for many more years than I have actually known him personally. His reputation precedes him, and—take it from me—he is one of those people who is bigger than his reputation and far more humble than his contribution warrants. I have not known Tony Morgan for nearly as long but I know many people who have worked with Tony and have the highest regard for his work.

Both of these men have invested a lifetime in developing these enterprise modeling concepts, not because of the enormous market demand, but because of their intense belief that these concepts are vital for the advancement of our capability as humans to accommodate the extreme complexity and rates of change that characterize Twenty First Century life.

In fact, those of us who have devoted our lives to some of these apparently esoteric modeling pursuits are not doing it to make money, because the general market is hardly even aware of the issues, much less willing to pay for them. We are doing it because we know it is right and we are certain that survival of life as we know it is not dependent on writing more code. It is dependent upon being able to describe the complexities of enterprises (any or all human endeavor) so they can actually be designed, implemented as conceived, and dynamically changed as the environment changes around them.

We all owe Terry and Tony a debt of gratitude for persevering to produce this comprehensive modeling work.

When Terry asked me to write the industrial foreword to the first edition of this book, my first reaction was, "Good Night! Am I qualified to write a foreword for a Terry Halpin book"? I suggested that he send it to me and I would decide whether I could write it for him or not. After he sent me the book, my next problem was, I couldn't put the book down! Can you imagine that? A technical book that keeps you wanting to read the next page?

Yes, it is a technical book, and the second edition is hardly any different. It is a *very* technical book that goes into detail on how to produce graphic models that exquisitely and rigorously capture the semantic details of an information system. But, it is also an easy-to-read book because it spells out clearly, concisely, and so simply the complexities of logic that provide any enterprise and any natural language with its capacity for nuance of expression and richness of description. For every step in the logic, there is provision of illustration and a test for your comprehension. There are hosts of models and exercises of real cases, none so contrived or so convoluted that it takes more energy to understand the case than to get the point.

Yes, Object Role Modeling 2 (ORM 2) is the notation for most of the illustrations, not simply because Terry actually "wrote the book" on ORM, but because of its incomparable ability to capture semantic intent and unambiguously express it graphically. And, yes, there is a discussion of ORM 2 modeling in sufficient detail for modelers to acquire the ORM 2 language capability. But the cases and illustrations are rich with analysis that can help even modelers unfamiliar with ORM to accurately extract the precise semantics of a "universe of discourse."

But to me, all of this is not the strength of this book. The enduring strength of the book is two-fold. First, this is a very clear and vivid demonstration of the incredible complexities of accurately discerning and capturing the intentions of the enterprise and transforming them into the realities of an implementation. There is little wonder why the systems we have been implementing for the last 50 years (total history of "Data Processing") are so inflexible, unadaptable, misaligned, unintegrated, unresponsive, expensive, unmaintainable, and so frustrating to management. We never bothered to produce an accurate description of the concepts of the enterprise in the first place!

If you don't rigorously describe the enterprise to begin with, why would anybody expect to be able to produce a relevant design and implementation that reflected enterprise management's reality or intent, or that could be adapted over time to accommodate their changes?

Tragically, few general managers are likely to read so technical a book as the second edition of *Information Modeling and Relational Databases*. But *all* general managers ought to read this book to get some perspective on the semantic complexity of their own enterprises, of the challenges of accurately capturing that complexity, of the necessity of their own participation and decisions in conceptual modeling, of the sophistication of the engineering that is required to produce quality and flexible implementations, of the fact that systems (automated or not automated) are not magic, they are logic and good judgment and engineering rigor and a lot of hard work.

In fact, every data modeler regardless of his or her syntactic specialty—whether it be Chen, Barker, Finkelstein, IDEF1X, IDEF1x(Object), UML 2, XML, or XYZ—ought to read the book for the same reasons. In fact, modelers of all kinds ought to read the book. In fact, every *programmer* ought to read the book. In fact, anyone who has anything to do with information or information systems ought to read the book!

The second strength of this book lies in the derivations from the high standard of semantic expression established by employing the second version of Object Role Modeling. Having demonstrated that it is possible to be rigorous and graphic in capturing precise semantic intent, the book straight-forwardly evaluates all the other popular graphic modeling notations in terms of their ability to duplicate that expression. There is a comparison with every other modeling notation that I have ever heard of, including the ones I mentioned above like Chen, IDEF1X, UML 2 etc. This is the most objective and precise comparison I have ever seen. The authors are very apologetic about appearing to be critical of other languages, but my observation is that this comparison was the most dispassionate and objective discussion I have ever seen. They even point out the strengths of these other notations and how and where in the overall process they can be used effectively. How's that for objectivity?!

There is one more interesting dimension of these rigorous, precise semantic models —they have to be transformed into databases for implementation. The authors describe in detail and by illustration the transformation to logical models, to physical database design, and to implementation. In this context, it is easy to evaluate and compare the various database implementation possibilities including relational databases, object-oriented databases, object-relational databases, and declarative databases; and they throw in star schemas and temporal databases for good measure! Once again, I cannot remember seeing so dispassionate and objective an evaluation and comparison of the various database structures. Within this context, it is straight-forward to make a considered and realistic projection of database technology trends into the foreseeable future.

This is a book that is going to sit on my bookshelf forever. I would consider it a candidate to be one of the few classics in our rather young, 50-year old discipline of information management. I hope I have the personal discipline to pick it up about once a year and refresh my understanding of the challenges and the possibilities of the information profession. I am confident you will find the second edition of *Information Modeling and Relational Databases* as useful and enlightening as I have, and I hope that there will be many more editions to come!

Foreword

by Professor Dr. Sjir Nijssen, CTO
PNA Group, The Netherlands

It gives me great personal pleasure to write this foreword. I have known Terry Halpin since 1986. As John Zachman has said about Terry, he is one of those people who is bigger than his reputation and far more humble than his contribution warrants. Terry is one of the most effective and dedicated authors of a new wave in knowledge engineering and requirements specification. I would like to classify ORM (Terry's focus, called Object-Role Modeling) as a fact orientation approach. This by itself is already much broader than data modeling. It is my professional opinion, based on extensive experience during more than 40 years in business and academia, that fact orientation is the most productive data modeling approach, by far. This approach could be considered as a best business practice for SBVR (*Semantics of Business Vocabulary and Business Rules*), the standard adopted by the Object Management Group (OMG) on December 11, 2007.

With fact orientation, it is useful to distinguish between structure and structuring. Both are important in practice and theory. With respect to structuring, one of the subprocesses is verbalization. Verbalization is a major and unique part of the CogNIAM and ORM methodology. This entered the fact orientation community in 1959, when I was training young people to plot the movements of airplanes in an area where radar could not yet see. In 1967 I got the chance of a professional lifetime. My manager said "*I want to hire you, and you have only one mission: nearly every software professional in our company* (Control Data Corporation, one of the most powerful computer companies in that period) *is preoccupied with programming. I want you to ignore procedural programming, and concentrate on the data underlying all programming".*

During the seventies, conceptual modeling—of which ORM is an instance—was developed primarily in Europe by a group of people from various companies and universities. However, two excellent American researchers also contributed substantially: Dr. Michael Senko and Dr. Bill Kent, both of IBM. Anyone reading this book will also enjoy reading the classics of Mike Senko published in the *IBM Systems Journal* and the pearl *Data and Reality*, the book written by Dr. Kent. In that period, NIAM was conceived. In the late seventies a group of international conceptual modelers undertook in ISO the task of writing the report *Concepts and Terminology for the Conceptual Schema and the Information Base*. It was a report about conceptual modeling, natural language, and logic.

In 1986, when I was professor of computer science at the University of Queensland, a relatively young Terry became my colleague as lecturer in the Computer Science Department. What a fantastic colleague! He very quickly caught on to my lecturing on conceptual modeling. It quickly became apparent to me that he was full of ideas, resulting in many collaborative sessions and the further development of the NIAM method at that time. His own lecture preparation was excellent, generating many exercises, and it was a real pleasure to work with him. We jointly published a book in 1989 which was largely written by Terry. In 1989, Terry completed a doctoral thesis that formalized and enhanced NIAM, and he and I went our own separate ways as I decided to return to The Netherlands.

In the following period, many excellent extensions to the NIAM 1989 version were added by Terry and his team on the one hand, and myself and associates on the other. In retrospect, I consider it a very good approach to work independently for some time and subsequently come to the conclusion that beautiful improvements have been developed. Two years ago, we decided to establish the best combination of the improvements developed independently. One of the strong points of our methodology is its incomparable ability to capture precise semantic intent and unambiguously express it graphically.

I strongly recommend every serious data modeler, business process modeler, and programmer to study this excellent book very carefully. I perused the chapters with pleasure and found it very useful and clearly presented. It is an excellent textbook for universities that intend to provide a first-class conceptual modeling course. UML has received much greater attention than ORM up till now, and I personally find that a shame, because in my opinion there are many areas in which ORM outperforms UML. I expect that this point will become clear to all with UML experience, who have their first encounter with ORM by reading this book. I therefore hope this book will result in further attention to ORM, which would be much deserved.

The relationship between the new OMG standard SBVR and ORM is not explicitly mentioned in the text, but be assured that there is a clear philosophical link between them. People familiar with both standards will recognize this easily.

Terry's coauthor, Tony, has added a very interesting chapter about processes, and has incorporated the task of modeling these processes as part of conceptual modeling. Tony is an excellent teacher, as I have recently had the pleasure of witnessing at the ORM2007 Conference in Portugal: British humor thrown in for free!

It is my distinct pleasure to highly recommend this book to anybody seriously interested in acquiring competence in conceptual analysis or modeling, with the aim of making modeling an understandable form of engineering instead of considering it as just an art.

Foreword

by Dr. Gordon C. Everest, Professor Emeritus and Adjunct, MIS and DBMS
Carlson School of Management, University of Minnesota, USA.

I am delighted and honored to write a foreword to this second edition. It gives me another opportunity to convince those in the world of data modeling that there is a better way. I am absolutely convinced that Object Role Modeling (ORM) is a better way to do data modeling. My underlying motive in this foreword is to sufficiently perk your interest to seriously study ORM, and this book is the best resource available to you.

Data modeling is the foundation of information systems development—if you don't get the database design "right" then the systems you build will be like a house of cards, collapsing under the weight of inevitable future forces for revision, enhancement, integration, and quality improvement. Thus, we need a scheme to guide our data modeling efforts to produce data models that clearly and accurately represent the users' domain of discourse and facilitate human communication, understanding, and validation.

This book is a must for anyone who is serious about data modeling, but with a caution: you must devote enough time and effort to really understand ORM. Fortunately, I have my students as a captive audience for a whole semester—long enough for them to learn and practice ORM and become convinced that it is a better way. With ORM you can raise your data modeling skills to a whole new level.

This book also examines record-based modeling schemes: UML, SQL based on "Ted" Codd's relational model, and Peter Chen's Entity Relationship (ER) diagrams with many variations—Barker as in Oracle, Finkelstein's Information Engineering (IE), and IDEF1X (as in ERwin). Viewing these familiar modeling approaches from an ORM perspective provides an enriched understanding of their underlying nature.

Record-based modeling schemes use three constructs: Entity, Attribute, and Relationship. It is the clustering of attributes into entity records that is the root of many of our problems in data modeling. Normalization is the test to see if we clustered too much, and record decomposition is commonly used as a remedy to correct a violation of the normal forms.

Normalization is the Achilles heel of data modeling. Oh, to be able to avoid normalization altogether? The mere suggestion is intriguing to students and practitioners of data modeling. Well, with ORM you can. The problem stems from the lack of clear definition of relationships when we throw stuff into a record, so that the intra-record structure is implicitly defined or assumed. ORM forces you to separately consider and define all relevant relationships and constraints among the object domains in your universe.

ORM is actually based on only two constructs: objects and relationships (which correspond to the concepts of nouns as subject or object, and verbs as predicates in sentences). Both entities and attributes are treated as objects in ORM (not to be confused with objects in object-oriented technology). Objects play roles in relationships with other objects. Objects have attributes or descriptors by virtue of the roles they play in relationships with other objects. In record-based modeling, there are two kinds of relationships: inter-record, and intra-record among attributes. In ORM all relationships are represented the same way with a single construct. When the ORM model is a valid representation of the world being modeled, the functional and multivalued dependencies are explicitly defined, and hence, the generation of “records” (in a relational table) can be automated and can guarantee that the result will be fully normalized (to 5NF). That’s good news for data modelers.

ORM does not supplant ER diagrams or relational database designs, rather it is a stage before. It can enable, enlighten, and inform our development and understanding of ER/relational data models. We build records more for system efficiency, than for human convenience or comprehension. The premature notion of a record (a cluster of attribute domains along with an identifier to represent an entity) actually gets in the way of good data modeling. ORM does not involve records, tables, or attributes. As a consequence, we don't get bogged down in “table think”—there is no need for an explicit normalization process.

The second edition is even more focused on the centrality of data in information systems, and on the importance of semantics. Starting with the realization that users (collectively) know more than we could ever capture in a data model, we must use a data modeling scheme that captures the widest possible range of semantics, and express this meaning graphically. Semantics is paramount, and ORM goes way beyond any record-based modeling scheme in graphically depicting semantics. With this second edition, Terry and Tony have expanded the scope of ORM to include temporality, dynamics, state modeling, and business processes.

Well, is that sufficient to pique your interest in learning more about ORM? If you are a would-be student of ORM and you take data modeling seriously, I encourage you to invest some time to read this book. You won't regret it. You will grow to appreciate ORM and will become a better data modeler for it. In order to develop effective and maintainable information systems we need good data models, and for that we need a good data modeling methodology. ORM allows us to develop database designs at the highest conceptual level, unencumbered by things that are not of primary concern to user domain specialists. My deep desire is to see more and more database designers using ORM. The systems we build and the world we live in will be better for it. Join me in this journey and enjoy the adventure.

Preface

This book is about information systems, focusing on information modeling and relational database systems. It is written primarily for data modelers and database practitioners as well as students of computer science or information management. It should also be useful to anyone wishing to formulate the information structure of business domains in a way that can be readily understood by humans yet easily implemented on computers. In addition, it provides a simple conceptual framework for understanding what database systems really are, and a thorough introduction to SQL and other key topics in data management.

A major part of this book deals with *fact-oriented modeling*, a conceptual modeling approach that views the world in terms of simple facts about objects and the roles they play. Originating in Europe, fact-orientation is today used worldwide and comes in many flavors, including the Semantics of Business Vocabulary and Business Rules (SBVR) approach adopted in 2007 by the Object Management Group. The version of fact-orientation described in this book is second generation *Object-Role Modeling* (ORM 2), and is based on extensions to NIAM (Natural-language Information Analysis Method).

Two other popular notations for information modeling are *Entity-Relationship* (ER) diagrams and *Unified Modeling Language* (UML) class diagrams. For conceptual information analysis, the ORM method has several advantages over the ER and UML approaches. For example, ORM models can be easily verbalized and populated for validation with domain experts, they are more stable under changes to the business domain, and they typically capture more business rules in diagram form.

However ER diagrams and UML class diagrams are good for compact summaries, and their structures are closer to the final implementation, so they also have value. Hence the coverage includes chapters on data modeling in ER and UML, and indicates how ER and UML data models can be easily generated from ORM models.

To make the text more approachable to the general reader with an interest in databases, the language is kept simple, and a formalized, mathematical treatment is deliberately avoided. Where necessary, relevant concepts from elementary logic and set theory are discussed prior to their application. Most of the material in this book has been class tested in courses to both industry and academia, and the basic ORM method has been taught successfully even at the high school level. The content is modularized, so that instructors wishing to omit some material may make an appropriate selection for their courses.

The first chapter motivates the study of conceptual modeling, and briefly compares the ORM, ER, and UML approaches. It also includes an historical and structural overview of information systems. Chapter 2 provides a structural background, explaining the conceptual architecture of, and development frameworks for, information systems. It introduces a number of key concepts that are dealt with more thoroughly in later chapters, and should be read in full by the reader with little or no database experience.

Chapter 3 is fundamental. Following an overview of conceptual modeling language criteria and the ORM Conceptual Schema Design Procedure (CSDP), this chapter covers the first three steps of the CSDP. The first step (verbalizing familiar examples in terms of elementary facts) may seem trivial, but it should not be rushed, as it provides the foundation for the model. The rest of this chapter covers the basic graphical notation for fact types, and then offers guidance on how to classify objects into types and identify information that can be arithmetically derived.

Chapter 4 begins the task of specifying constraints on the populations of fact types. The most important kind of constraint (the uniqueness constraint) is considered in detail. Then some checks on the elementarity of the fact types are discussed. This chapter also introduces the join and projection operations at the conceptual level—the relational version of these operations is important in the later work on relational databases.

Chapter 5 covers mandatory role constraints, including a check for detecting information that can be logically derived. Reference schemes are then examined in some depth. Some of the more complex reference schemes considered here could be skipped in a short course. The CSDP steps covered so far are then reviewed by applying them in a case study, and the logical derivation check is then considered.

Chapter 6 covers value, set comparison (subset, equality, and exclusion), and subtyping constraints. Section 6.6 deals with advanced aspects of subtyping—though important in practice, the material in this section could be skimmed over in a first reading.

Chapter 7 deals with the final step of the conceptual schema design procedure. Less common constraints are considered (e.g., occurrence frequencies and ring constraints), and final checks are made on the design. Sections 7.3–7.5 are somewhat advanced, and could be skipped in a short course.

Chapter 8 discusses the Entity Relationship (ER) approach, starting with Chen's original notation then moving on to the three most popular notations in current use: the

Barker ER notation, the Information Engineering notation, and the IDEF1X notation (actually a hybrid of ER and relational notations). Comparisons with ORM are included along the way.

Chapter 9 examines the use of UML class diagrams for data modeling, including a detailed comparison with ORM. Business rule constructs in ORM with no graphic counterpart in UML are identified and then captured in UML using user-defined constraints or notes.

Chapter 10 considers several advanced aspects of information modeling, such as join constraints, historical fact types, collection types, open/closed world semantics, and higher-order types. The discussion of deontic rules and nominalization is fundamental to understanding the SBVR flavor of fact-orientation. This chapter is technically the most challenging in the book, and could be skipped in an introductory course.

Chapter 11 describes how a conceptual model may be implemented in a relational database system. The first three sections are fundamental to understanding how a conceptual schema may be mapped to a relational schema. Section 11.4 considers advanced mapping aspects, and could be omitted in a short course.

Chapter 12 provides a foundational introduction to relational databases and SQL queries. Section 12.1 covers relational algebra—although not used as a practical query language, the algebra is important for understanding the basic relational operations supported by SQL. Section 12.2 provides an overview of how the relational model of data compares with data models adopted by some relational database management systems. Sections 12.3–12.14 cover the main features of SQL, with attention to the SQL-89, SQL-92, SQL:1999, SQL:2003, and SQL:2008 standards, and some popular dialects.

Chapter 13 discusses further aspects of SQL (e.g., data definition, triggers, and stored procedures), the use of other languages such as XML in conjunction with SQL, and introduces some practical issues such as security, metadata and concurrency.

Chapter 14 discusses how and when to transform one schema into another schema at the same level (conceptual or logical). Sections 14.1–14.4 examine the notion of conceptual schema equivalence, and ways in which conceptual schemas may be reshaped. As one application of this theory, section 14.5 specifies a procedure for optimizing a database design by performing conceptual transformations before mapping. Section 14.6 provides a concise coverage of normalization theory, including some new insights. Section 14.7 briefly considers denormalization and low-level optimization. Sections 14.8–14.9 illustrate the role of conceptual optimization in database reengineering, and conclude with a discussion of data migration and query transformation. Sections 14.4, 14.5, 14.7, 14.8, and 14.9 are of an advanced nature and may be skipped in a short course. In a very short course, the whole chapter could be skipped.

Chapter 15 broadens the treatment of information systems analysis by examining behavioral aspects of business using process and state models. The fundamental concepts underlying business processes and workflows are explained, including popular graphical notations, process patterns, and process standards. Some ways of integrating behavioral models with information models are also considered.

Chapter 16 examines other modeling aspects and trends. Topics covered include data warehousing, conceptual query languages, schema abstraction mechanisms, further design aspects, ontologies and the semantic web, post-relational databases (e.g. object databases and object-relational databases) and metamodeling. Though these topics are important and interesting, they could be omitted in a short course.

In line with the ORM method, this text adopts a "cookbook" approach, with plenty of diagrams and examples. Each chapter begins with a brief overview, and ends with a chapter summary of the major points covered, with chapter notes to provide fine points and further references. One of the major features of the book is its large number of exercises, which have been thoroughly class-tested. A bibliography of all cited references is included at the end of the book, where you will also find glossaries of technical symbols and terms for ORM, ER, and UML (class diagrams only). A comprehensive index provides easy access to explanations of technical topics.

For readers familiar with the previous edition of this book, the major differences are now summarized. The coverage of ORM and UML has substantially updated to cover their latest versions (ORM 2 and UML 2), which necessitated the redrawing of almost all diagrams in the earlier edition. Whole new chapters have been added (Advanced Modeling Issues, Using Other Database Objects, and Process and State Modeling), as well as new chapter sections (e.g. ontologies and the semantic web). All previous chapters have been substantially revised, with several topics covered in greater depth, and there are many new exercises. The new content has led to a much larger book, which now has two coauthors, with Terry Halpin responsible for Chapters 1–11, 14, 16, most of Chapter 12, and part of Chapter 13, and Tony Morgan responsible for Chapter 15, part of Chapter 12, and most of Chapter 13.

U.S. spelling is used throughout the book. U.S. punctuation rules have also been used, except for quoted items, where item separators and sentence terminators (e.g., commas and periods) appear after, rather than just before, closing quotes.

Online Resources

To reduce the size and hence cost of the book, supplementary material has been made available online at the publisher's Web site (*www.mkp.com/imrd2/*) for downloading. There are at least three appendices. Appendix A provides an overview of the evolution of computer hardware and software. Appendix B discusses two kinds of subtype matrix that can be used to determine subtype graphs from significant populations. Appendix C discusses advanced aspects of SQL, focusing on set-comparison queries and group extrema queries. Appendices on other topics may be added as the need arises.

The answers to the exercises are contained in two files, one for the odd-numbered questions and one for the even-numbered questions. The answers to the odd-numbered questions are openly accessible, but the answers to the even-numbered questions are password protected, in order to provide classroom instructors with a range of exercises for classroom discussion. Additional material on ORM and other topics is available via the URLs listed in the Useful Web Sites section at the back of the book.

Electronic versions of the figures, as well as further exercises and related pedagogic material, are included in supplementary online resources that are available to instructors using this book as a course text. These resources and a password for exercise answers are available to classroom instructors at *http://textbooks.elsevier.com*.

ORM Software

ORM is supported by a variety of modeling tools. At the time of writing, the Neumont ORM Architect (NORMA) tool provides the most complete support for the ORM 2 notation discussed in this book. NORMA is an open source plug-in to Microsoft Visual Studio .NET and may be downloaded, along with supporting documentation, either from *www.ormfoundation.org* or from *http://sourceforge.net/projects/orm*.

The previous version of ORM (ORM 1) is supported as the ORM Source Model Solution in a high end version of Microsoft Visio (Halpin et al. 2003). A discontinued ORM tool known as VisioModeler is freely available as a download from Microsoft's MSDN Web site. Although VisioModeler does not run under Windows Vista, and the product is somewhat outdated in its database driver support, it does allow you to create ORM models under earlier versions of Windows and map them to a range of database management systems. To download the VisioModeler tool, point your browser at *http://msdn.microsoft.com/downloads*, and then search for "VisioModeler 3.1 (Unsupported Product Edition)".

Other tools supporting different flavors of fact orientation include Doctool, CaseTalk, and Infagon. Other fact-oriented tools under development at the time of writing include ActiveFacts and CogNIAM. Links to these tools may be found in the Useful Web Sites section at the back of this book.

Acknowledgments

We greatly appreciate the editorial assistance of Mary James at Morgan Kaufmann, the suggestions made by the anonymous reviewers, the technical editing by Dr. Andy Carver, and the copy editing by Melissa Revell. We also thank Technologies 'N Typography for permission to use their TNT fonts.

Some of this book's content is based on articles written for *The Business Rules Journal*, the *Journal of Conceptual Modeling*, and material from editions of the earlier book *Conceptual Schema and Relational Database Design*, previously published by Prentice Hall Australia and WytLytPub. The first edition of that book was coauthored by Dr. Sjir Nijssen, who wrote the final four chapters (since replaced).

The ORM approach discussed in this book is based on our revisions and extensions to the NIAM method, which was largely developed in its original form by Dr. Eckhard Falkenberg, Dr. Sjir Nijssen, and Prof. Robert Meersman, with other contributions from several researchers. Some of the work discussed in this book was developed jointly with current or former colleagues, including Dr. Anthony Bloesch, Dr. Linda Bird, Dr. Peter Ritson, Dr. Erik Proper, Dr. Andy Carver, and Dr. Herman Balsters. It

has been a pleasure working with these colleagues, as well as with the hundreds of students and practitioners with whom trial versions of the material in this book were tested. We also gratefully acknowledge permission by The University of Queensland and Neumont University to include a small selection of our past assessment questions within the exercises.

A modeling method as good as ORM deserves a good CASE tool. Over the last decade, talented staff at ServerWare, Asymetrix Corporation, InfoModelers Incorporated, Visio Corporation, Microsoft Corporation, and Neumont University have worked to develop state of the art CASE tools to support the specific ORM method discussed in this book. The following talented individuals currently working as lead software engineers on the NORMA tool deserve special mention: Matt Curland and Kevin Owen.

Finally we thank our wives, Norma and Gwen, for being so understanding and supportive while we were busily occupied in the writing task.