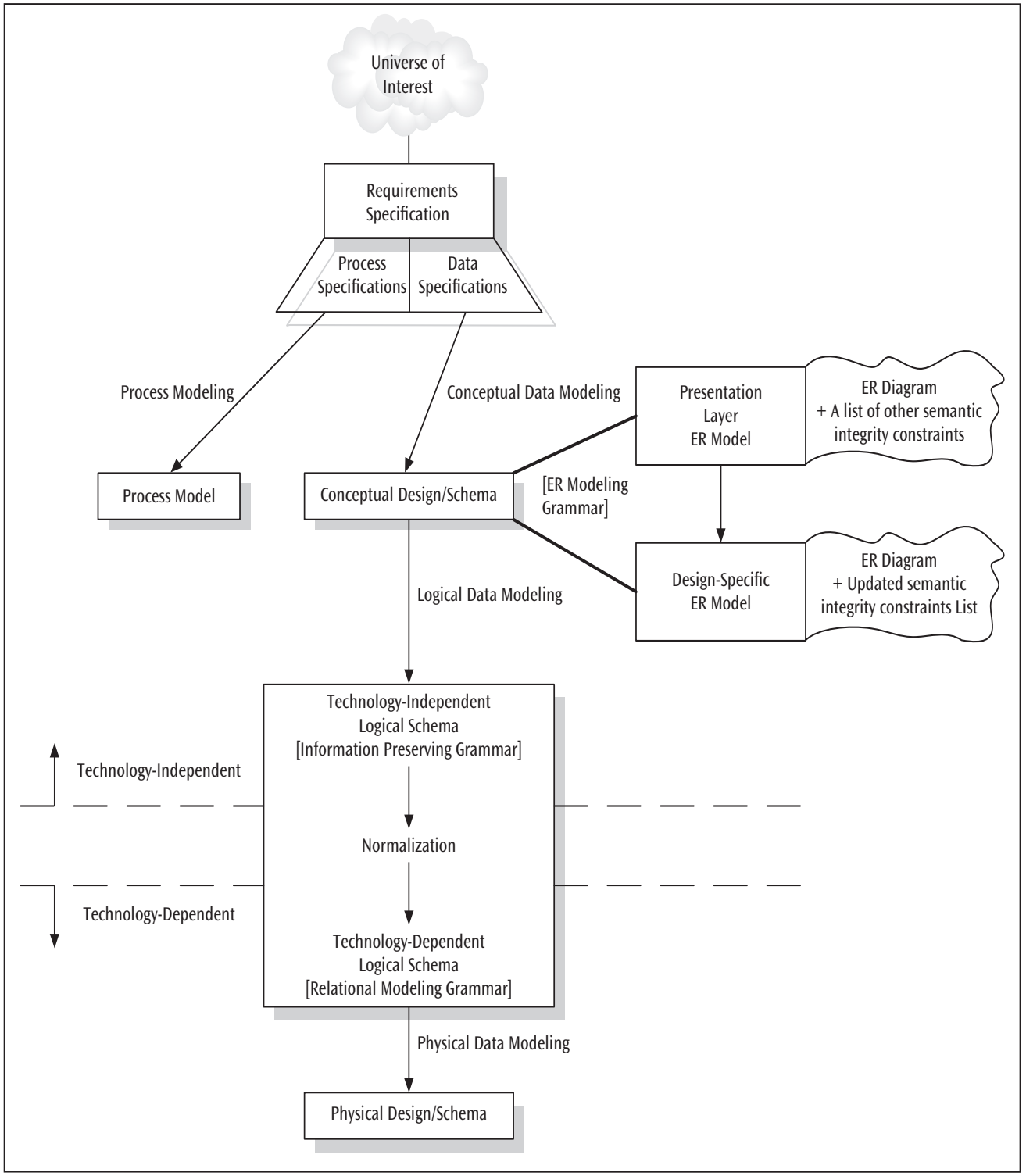
SECOND EDITION

DATA MODELING AND DATABASE DESIGN

UMANATH | SCAMELL

Data modeling/database design life cycle



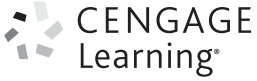
DATA MODELING AND DATABASE DESIGN

DATA MODELING AND DATABASE DESIGN

Second Edition

Narayan S. Umanath
University of Cincinnati

Richard W. Scamell
University of Houston



Australia • Brazil • Mexico • Singapore • United Kingdom • United States

CENGAGE Learning®

Data Modeling and Database Design, Second Edition
Narayan S. Umanath and Richard W. Scamell

Production Director: Patty Stephan
Product Manager: Clara Goosman
Managing Developer: Jeremy Judson
Content Developer: Wendy Langeurd
Product Assistant: Brad Sullender
Senior Marketing Manager: Eric La Scola
IP Analyst: Sara Crane
Senior IP Project Manager: Kathryn Kucharek
Manufacturing Planner: Ron Montgomery
Art and Design Direction, Production Management, and Composition: PreMediaGlobal
Cover Image: © VikaSuh/www.Shutterstock.com

Library of Congress Control Number: 2014934580

ISBN-13: 978-1-285-08525-8

ISBN-10: 1-285-08525-6

Cengage Learning
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at **www.cengage.com/global**

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

To learn more about Cengage Learning Solutions, visit **www.cengage.com**

Purchase any of our products at your local college store or at our preferred online store **www.cengagebrain.com**

Printed in the United States of America
1 2 3 4 5 6 7 18 17 16 15 14

To Beloved Bhagwan Sri Sathya Sai Baba, the very source of my thoughts, words, and deeds
To my Graduate Teaching Assistants and students, the very source of my inspiration
To my dear children, Sharda and Kausik, always concerned about their dad overworking
To my dear wife Lalitha, a pillar of courage I always lean on

Uma

There is a verse that says
Focus on what I'm doing right now
And tell me that you appreciate me
So that I learn to feel worthy
And motivated to do more
Led by my family, I have always been surrounded by people (friends, teachers, and students) who
With their kind thoughts, words, and deeds treat me in this way.
This book is dedicated to these people.

Richard

BRIEF CONTENTS

Part IV: Database Implementation Using the Relational Data Model

TABLE OF CONTENTS

Part II: Logical Data Modeling

Part III: Normalization

Part IV: Database Implementation Using the Relational Data Model

PREFACE

QUOTE

*Everything should be made as simple as possible—**but no simpler**.*
—Albert Einstein

Popular business database books typically provide broad coverage of a wide variety of topics, including data modeling, database design and implementation, database administration, the client/server database environment, the Internet database environment, distributed databases, and object-oriented database development. This is invariably at the expense of deeper treatment of critical topics, such as principles of data modeling and database design. Using current business database books in our courses, we found that in order to properly cover data modeling and database design, we had to augment the texts with significant supplemental material (1) to achieve precision and detail and (2) to impart the depth necessary for the students to gain a robust understanding of data modeling and database design. In addition, we ended up skipping several chapters as topics to be covered in a different course. We also know other instructors who share this experience. Broad coverage of many database topics in a single book is appropriate for some audiences, but that is not the aim of this book.

The goal of *Data Modeling and Database Design, Second Edition* is to provide core competency in the areas that every Information Systems (IS), Computer Science (CS), and Computer Information Systems (CIS) student and professional should acquire: **data modeling and database design**. It is our experience that this set of topics is the most essential for database professionals, and that, covered in sufficient depth, these topics alone require a full semester of study. It is our intention to address these topics at a level of technical depth achieved in CS textbooks, yet make palatable to the business student/IS professional with little sacrifice in precision. We deliberately refrain from the mathematics and algorithmic solutions usually found in CS textbooks, yet we attempt to capture the precision therein via heuristic expressions.

Data Modeling and Database Design, Second Edition provides not just hands-on instruction in current data modeling and database design practices, it gives readers a thorough conceptual background for these practices. We do not subscribe to the idea that a textbook should limit itself to describing what is actually being practiced. Teaching only what is being practiced is bound to lead to knowledge stagnation. Where do practitioners learn what they know? Did they invent the relational data model? Did they invent the ER model? We believe that it is our responsibility to present not only industry "best practices" but also to provide students (future practitioners) with concepts and techniques that are not necessarily used in industry today

but can enliven their practice and help it evolve without knowledge stagnation. One of the coauthors of this book has worked in the software development industry for over 15 years, with a significant focus on database development. His experience indicates that having a richness of advanced data modeling constructs available enhances the robustness of database design and that practitioners readily adopt these techniques in their design practices.

In a nutshell, our goal is to take an IS/CS/CIS student/professional through an intense educational experience, starting at conceptual modeling and culminating in a fully implemented database design—**nothing more and nothing less**. This educational journey is briefly articulated in the following paragraphs.

STRUCTURE

We have tried very hard to make the book "fluff-free." It is our hope that every sentence in the book, including this preface, adds value to a reader's learning (and *footnotes* are no exception to this statement).

The book begins with an introduction to rudimentary concepts of data, metadata, and information, followed by an overview of data management. Pointing out the limitations of file-processing systems, **Chapter 1** introduces database systems as a solution to overcome these limitations. The architecture and components of a database system that makes this possible are discussed. The chapter concludes with the presentation of a framework for the database system design life cycle. Following the introductory chapter on database systems architecture and components, the book contains four parts.

Part I: Conceptual Data Modeling

Part I addresses the topic of conceptual data modeling—that is, modeling at the highest level of abstraction, independent of the limitations of the technology employed to deploy the database system. Four chapters (Chapters 2–5) are used in order to provide an extensive discussion of conceptual data modeling. Chapter 2 lays the groundwork using the *Entity-Relationship (ER) modeling grammar* as the principal means to model a database application domain. Chapter 3 elaborates on the use of the ER modeling grammar in progressive layers and exemplifies the modeling technique with a comprehensive case called Bearcat Incorporated. This is followed by a presentation in Chapter 4 of richer data modeling constructs that overlap with object-oriented modeling constructs. The Bearcat Incorporated story is further enriched to demonstrate the value of Enhanced ER (EER) modeling constructs. Chapter 5 provides exclusive coverage of modeling complex relationships that have meaningful real-world significance. At the end of Part I, the reader ought to be able to fully appreciate the value of conceptual data modeling in the database system design life cycle.

This second edition of *Data Modeling and Database Design* includes the following major enhancements:

- The material in Chapters 2 and 3 has been reorganized and better streamlined so that the reader not only learns the ER modeling grammar but is able to develop very simple applications of ER modeling. In Chapter 3, the modeling method steps have been reconfigured across the Presentation Layer and

Design-Specific layer of the ER model. Also, the unique learning technique via error detection exclusively developed by us is presented at the end of Chapter 3.

- The intra-entity class relationships are introduced with a new simpler example at the beginning of Chapter 4.
- The already extensive coverage of complex relationships in Chapter 5 is augmented by a few newer modeling ideas. Additional examples clarifying decomposition of complex relationships in preparation for logical model mapping have also been added to this chapter.

Part II: Logical Data Modeling

Part II of the book is dedicated to the discussion of migration of a conceptual data model to its logical counterpart. Since the relational data model architecture forms the basis for the logical data modeling discussed in this textbook, Chapter 6 focuses on its characteristics. Other logical data modeling architectures prevalent in some legacy systems, the hierarchical data model, and the CODASYL data model appear in Appendix A. An introduction to object-oriented data modeling concepts is presented in Appendix B. The rest of Chapter 6 describes techniques to map a conceptual data model to its logical counterpart. An *information-preserving logical data modeling grammar* is introduced and contrasted with existing popular mapping techniques that are information reducing. A comprehensive set of examples is used to clarify the use and value of the information-preserving grammar.

An important addition to the current edition of the book is a section on mapping complex relationship types to the logical tier.

Part III: Normalization

Part III addresses the critical question of the "goodness" of a database design that results from a conceptual and logical data modeling processes. *Normalization* is introduced as the "scientific" way to verify and improve the quality of a logical schema that is available at this stage in the database design. Three chapters are employed to cover the topic of normalization. In Chapter 7, we take a look at data redundancy in a relation schema and see how it manifests as a problem. We then trace the problem to its source—namely, undesirable functional dependencies. To that end, we first learn about functional dependencies axiomatically and how inference rules (Armstrong's axioms) can be used to derive candidate keys of a relation schema. In Chapter 8, the solution offered by the normalization process to data redundancy problems triggered by undesirable functional dependencies is presented. After discussing first, second, third and Boyce-Codd normal forms individually, we examine the side effects of normalization—namely, dependency preservation and non-loss decomposition and their consequences. Next, we present real-world scenarios of deriving full-fledged relational schemas (sets of relation schemas), given sets of functional dependencies using several examples. The useful topic of denormalization is covered next. Reverse engineering a normalized relational schema to the conceptual tier often forges insightful understanding of the database design and enables a database designer to become a better data modeler. Despite its practical utility, this

topic is rarely covered in database textbooks. Chapter 9 completes the discussion of normalization by examining multi-valued dependency (MVD) and join-dependency (JD) and their impact on a relation schema in terms of fourth normal form (4NF) and Project/Join normal form, viz., PJNF (also known as fifth normal form—5NF) respectively.

An interesting enhancement in Chapter 8 is the introduction of a fast-track algorithm to achieve a non-loss, dependency-preserving 3NF design. Two distinct examples demonstrating the use of the algorithm are presented. The discussion of MVD and 4NF, of JD and 5NF, and their respective expressiveness of ternary and n-ray relationships is presented in Chapter 9. Additional examples offer unique insights into apparently conflicting alternative solutions.

Part IV: Database Implementation Using the Relational Database Model

Part IV pertains to database implementation using the relational data model. Spread over four chapters, this part of the book covers relational algebra and the ANSI/ISO standard Structured Query Language (SQL). Chapter 10 focuses on the data definition language (DDL) aspect of SQL. Included in the discussion are the SQL schema evolution statements for adding, altering, or dropping table structures, attributes, constraints, and supporting structures. This is followed by the development of SQL/DDL script for a comprehensive case about a college registration system. The chapter also includes the use of INSERT, UPDATE, and DELETE statements in populating a database and performing database maintenance.

Chapters 11, 12, and 13 focus on relational algebra and the use of SQL for data manipulation. **Chapter 11** concentrates on E. F. Codd's eight original relational algebra operations as a means to specify the logic for data retrieval from a relational database. SQL, the most common way that relational algebra is implemented for data retrieval operations, is the subject of Chapter 12. Chapter 13 covers a number of built-in functions used by SQL to work with strings, dates, and times, and it illustrates how SQL can be used to do retrievals against hierarchically structured data. This chapter also provides an introduction to some of the features of SQL that facilitate the summarization and analysis of data. The chapter ends with an SQL database project that provides students with a real-life scenario to test and apply the skills and concepts presented in Part IV.

FEATURES OF EACH CHAPTER

Since our objective is a crisp and clear presentation of rather intricate subject matter, each chapter begins with a simple introduction, followed by the treatment of the subject matter, and concludes with a chapter summary and a set of exercises based on the subject matter.

WHAT MAKES THIS BOOK DIFFERENT?

Every book has strengths and weaknesses. If lack of breadth in the coverage of database topics is considered a weakness, we have deliberately chosen to be weak in that dimension. We have not planned this book to be another general book on

database systems. We have chosen to limit the scope of this book exclusively to data modeling and database design since we firmly believe that this set of topics is the core of database systems and must be learned in depth by every IS/CS/CIS student and practitioner. Any system designed robustly has the potential to best serve the needs of the users. More importantly, a poor design is a virus that can ruin an enterprise.

In this light, we believe these are the unique strengths of this book:

- It presents conceptual modeling using the entity-relationship modeling grammar including extensive discussion of the enhanced entity-relationship (ER) model.
  We believe that a conceptual model should capture all possible constraints conveyed by the business rules implicit in users' requirement specifications. To that end, we posit that an ER diagram is not an ER model unless accompanied by a comprehensive specification of characteristics of and constraints pertaining to attributes. We accomplish this via a list of semantic integrity constraints (sort of a conceptual data dictionary) that will accompany an ER diagram, a unique feature that we have not seen in other database textbooks. We also seek to demonstrate the systematic development of a multi-layer conceptual data model via a comprehensive illustration at the beginning of each Part. We consider the multi-layer modeling strategy and the heuristics for systematic development as unique features of this book.
- It includes substantial coverage of higher-degree relationships and other complex relationships in the entity-relationship diagram.
  Most business database books seem to provide only a cursory treatment of complex relationships in an ER model. We not only cover relationships beyond binary relationships (e.g., ternary and higher-degree relationships), we also clarify the nuances pertaining to the necessity and efficacy of higher-degree relationships and the various conditions under which even recursive and binary relationships are aggregated in interesting ways to form cluster entity types.
- It discusses the information-preserving issue in data model mapping and introduces a new information-preserving grammar for logical data modeling.
  Many computer scientists have noted that the major difficulty of logical database design (i.e., transforming an ER schema into a schema in the language of some logical model) is the information preservation issue*. Indeed, assuring a complete mapping of all modeling constructs and constraints that are inherent, implicit or explicit, in the source schema (e.g., ER/EER model) is problematic since constraints of the source model often cannot be represented directly in terms of structures and constraints of the target model (e.g., relational schema). In such a case, they must be realized through application programs; alternatively, an* information-reducing transformation *must be accepted* (Fahrner and Vossen, 1995)*. In their research, initially presented at the Workshop on Information Technologies (WITS) in the ICIS (International Conference on Information Systems) in Brisbane,*

Australia, Umanath and Chiang (2000) describe a logical modeling grammar that generates an information preserving transformation. Umanath further revised this modeling grammar based on the feedback received at WITS. We have included this logical modeling grammar as a unique component of this textbook.

- It includes unique features under the topic of normalization rarely covered in business database books:
  - Inference rules for functional dependencies *(Armstrong's axioms) and derivations of candidate keys from a set of functional dependencies*
  - Derivation of canonical covers for a set of semantically obvious functional dependencies
  - Rich examples to clarify *the basic normal forms (first, second, third, and Boyce-Codd)*
  - Derivation of a complete logical schema *from a large set of functional dependencies considering lossless (non-additive) join properties and dependency preservation*
  - Reverse engineering a logical schema *to an entity-relationship diagram*
  - Advanced coverage of fourth and fifth normal form *(project-join normal form, abbreviated "PJNF") using a variety of examples*
- It supports in-depth coverage of relational algebra with a significant number of examples of their operationalization in ANSI/ISO SQL.

A NOTE TO THE INSTRUCTOR

The content of this book is designed for a rigorous one-semester course in database design and development and may be used at both undergraduate and graduate levels. Technical emphasis can be tempered by minimizing or eliminating the coverage of some of the following topics from the course syllabus: Enhanced Entity-Relationship (EER) Modeling (Chapter 4) and the related data model mapping topics in Chapter 6 (Section 6.8) on Mapping Enhanced ER Modeling Constructs to a Logical Schema; Modeling Complex Relationships (Chapter 5); and higher normal forms (Chapter 9). The suggested exclusions will not impair the continuity of the subject matter in the rest of the book.

SUPPORTING TECHNOLOGIES

Any business database book can be effective only when supporting technologies are made available for student use. Yet, we don't think that the type of book we are writing should be married to any commercial product. The specific technologies that will render this book highly effective include a drawing tool (such as Microsoft Visio), a software engineering tool (such as ERWIN, ORACLE/Designer, or Visible Analyst), and a relational database management system (RDBMS) product (such as ORACLE, SQL Server, or DB2).

SUPPLEMENTAL MATERIALS

The following supplemental materials are available to instructors when this book is used in a classroom setting. Some of these materials may also be found on the Cengage Learning Web site at *www.cengage.com.*

- ***Electronic Instructor's Manual***: The Instructor's Manual assists in class preparation by providing suggestions and strategies for teaching the text, and solutions to the end-of-chapter questions/problems.
- ***Sample Syllabi and Course Outline:*** The sample syllabi and course outlines are provided as a foundation to begin planning and organizing your course.
- ***Cognero Test Bank:*** Cognero allows instructors to create and administer printed, computer (LAN-based), and Internet exams. The Test Bank includes an array of questions that correspond to the topics covered in this text, enabling students to generate detailed study guides that include page references for further review. The computer-based and Internet testing components allow students to generate detailed study guides that include page references for further review. The computer-based and Internet testing components allow students to take exams at their computers, and also save the instructor time by automatically grading each exam. The Test Bank is also available in Blackboard and WebCT versions posted online at *www.course.com.*
- ***PowerPoint Presentations***: Microsoft PowerPoint slides for each chapter are included as a teaching aid for classroom presentation, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors can add their own slides for additional topics they introduce to the class.
- ***Figure Files***: Figure files from each chapter are provided for the instructor's use in the classroom.
- ***Data Files:*** Data files containing scripts to populate the database tables used as examples in Chapters 11 and 12 are provided on the Cengage Learning Web site at *www.cengage.com.*

ACKNOWLEDGMENTS

We have never written a textbook before. We have been using books written by our academic colleagues, always supplemented with handouts that we developed ourselves. Over the years, we accumulated a lot of supplemental material. In the beginning, we took the positive feedback from the students about the supplemental material rather lightly until we started to see comments like "I don't know why I bought the book; the instructor's handouts were so good and much clearer than the book" in the student evaluation forms. Our impetus to write a textbook thus originated from the consistent positive feedback from our students.

We also realized that, contrary to popular belief, business students are certainly capable of assimilating intricate technical concepts; the trick is to frame the concepts in meaningful business scenarios. The unsolicited testimonials from our alumni about

the usefulness of the technical depth offered in our database course in solving real-world design problems reinforced our faith in developing a book focused exclusively on data modeling and database design that was technically rigorous but permeated with business relevance.

Since we both teach database courses regularly, we have had the opportunity to field-test the manuscript of this book for close to 10 years at both undergraduate-level and graduate-level information systems courses in the Carl Lindner College of Business at the University of Cincinnati and in the C. T. Bauer College of Business at the University of Houston. Hundreds of students—mostly business students—have used earlier drafts of this textbook so far. Interestingly, even the computer science and engineering students taking our courses have expressed their appreciation of the content. This is a long preamble to acknowledge one of the most important and formative elements in the creation of this book: our students.

The students' continued feedback (comments, complaints, suggestions, and criticisms) have significantly contributed to the improvement of the content. As we were cycling through revisions of the manuscript, the graduate teaching assistants of Dr. Umanath were a constant source of inspiration. Their meaningful questions and suggestions added significant value to the content of this book. Dr. Scamell was ably assisted by his graduate assistants as well.

We would also like to thank the following reviewers whose critiques, comments, and suggestions helped shape every chapter of this book's first edition:

Akhilesh Bajaj, *University of Tulsa*
Iris Junlgas, *Florida State University*
Margaret Porciello, *State University of New York/Farmingdale*
Sandeep Purao, *Pennsylvania State University*
Jaymeen Shah, *Texas State University*

Last, but by no means the least, we gratefully acknowledge the significant contribution of Deb Kaufmann and Kent Williams, the development editors of our first and second editions, respectively. We cannot thank them enough for their thorough and also prompt and supportive efforts.

Enjoy!

N. S. Umanath

R. W. Scamell

CHAPTER 1

DATABASE SYSTEMS: ARCHITECTURE AND COMPONENTS

Data modeling and database design involve elements of both art and engineering. Understanding user requirements and modeling them in the form of an effective logical database design is an artistic process. Transforming the design into a physical database with functionally complete and efficient applications is an engineering process.

To better comprehend what drives the design of databases, it is important to understand the distinction between data and information. Data consists of raw facts—that is, facts that have not yet been processed to reveal their meaning. Processing these facts provides information on which decisions can be based.

Timely and useful information requires that data be accurate and stored in a manner that is easy to access and process. And, like any basic resource, data must be managed carefully. Data management is a discipline that focuses on the proper acquisition, storage, maintenance, and retrieval of data. Typically, the use of a database enables efficient and effective management of data.

This chapter introduces the rudimentary concepts of data and how information emerges from data when viewed through the lens of metadata. Next, the discussion addresses data management, contrasting file-processing systems with database systems. This is followed by brief examples of desktop, workgroup, and enterprise databases. The chapter then presents a framework for database design that describes the multiple tiers of data modeling and how these tiers function in database design. This framework serves as a roadmap to guide the reader through the remainder of the book.

1.1 DATA, INFORMATION, AND METADATA

Although the terms are often used interchangeably, information is different from data. **Data** can be viewed as raw material consisting of unorganized facts about things, events, activities, and transactions. While data may have implicit meaning, the lack of organization renders it valueless. In other words, **information** is data in context—that is, data that has been organized into a specific context such that it has value to its recipient.

As an example, consider the digits 2357111317. What does this string of digits represent? One response is that they are simply 10 meaningless digits. Another might be

the number 31 (obtained by summing the 10 digits). A mathematician may see a set of prime numbers, viz., 2, 3, 5, 7, 11, 13, 17. Another might see a person's phone number with the first three digits constituting the area code and the remaining seven digits the local phone number. On the other hand, if the first digit is used to represent a person's gender (1 for male and 2 for female) and the remaining nine digits the person's Social Security number, the 10 digits would mean something else. Numerous other interpretations are possible, but without a context it is impossible to say what the digits represent. However, when framed in a specific context (such as being told that the first digit represents a person's gender and the remaining digits the Social Security number), the data is transformed into information. It is important to note that "information" is not necessarily the "Truth" since the same data yields different information based on the context; information is an inference.

Metadata, in a database environment, is data that describes the properties of data. It contains a complete definition or description of database structure (i.e., the file structure, data type, and storage format of each data item), and other constraints on the stored data. For example, when the structure of the 10 digits 2357111317 is revealed, the 10 digits become information, such as a phone number. Metadata defines this structure. In other words, through the lens of metadata, data takes on specific meaning and yields information.[1] Metadata may be characterized as follows:

- The lens to view data and infer information
- A precise definition of the context for framing the data

Table 1.1 contains metadata for the data associated with a manufacturing plant. Later in this chapter, we will see that in a database environment, metadata is recorded in what is called a data dictionary.

Record Type	Data Element	Data Type	Size	Source	Role	Domain
PLANT	Pl_name	Alphabetic	30	Stored	Non-key	
PLANT	Pl_number	Numeric	2	Stored	Key	Integer values from 10 to 20
PLANT	Budget	Numeric	7	Stored	Non-key	
PLANT	Building	Alphabetic	20	Stored	Non-key	
PLANT	No_of_employees	Numeric	4	Derived	Non-key	

TABLE 1.1 Some metadata for a manufacturing plant

As reflected in Table 1.1, the smallest unit of data is called a **data element**. A group of related data elements treated as a unit (such as Pl_name, Pl_number, Budget, Building,

[1]With the advent of the data warehouse, the term "metadata" assumes a more comprehensive meaning to include business and technical metadata, which is outside the scope of the current discussion.

and No_of_employees) is called a **record type**. A set of values for the data elements constituting a record type is called a record instance or simply a **record**. A **file** is a collection of records. A file is sometimes referred to as a **data set**. A company with 10 plants would have a PLANT file or a PLANT data set that contains 10 records.

1.2 DATA MANAGEMENT

This book focuses strictly on management of data, as opposed to the management of human resources. Data management involves four actions: (a) data creation, (b) data retrieval, (c) data modification or updating, and (d) data deletion. Two data management functions support these four actions: Data must be accessed and, for ease of access, data must be organized.

Despite today's sophisticated information technologies, there are still only two primary approaches for accessing data. One is **sequential access**, where in order to get to the *nth* record in a data set it is necessary to pass through the previous *n–1* records in the data set. The second approach is **direct access**, where it is possible to get to the *nth* record without having to pass through the previous *n–1* records. While direct access is useful for *ad hoc* querying of information, sequential access remains essential for transaction processing applications such as generating payroll, grade reports, and utility bills.

In order to access data, the data must be organized. For sequential access, this means that all records in a file must be stored (organized) through some order using a unique identifier, such as employee number, inventory number, flight number, account number, or stock symbol. This is called sequential organization. A serial (unordered) collection of records, also known as a "heap file," cannot provide sequential access. For direct access, the records in a file can be stored serially and organized either randomly or by using an external index. A randomly organized file is one in which the value of a unique identifier is processed by some sort of transformation routine (often called a "hashing algorithm") that computes the location of records within the file (relative record numbers). An indexed file makes use of an index external to the data set similar in nature to the one found at the back of this book to identify the location where a record is physically stored.

As discussed in Section 1.5, a database takes advantage of software called a database management system (DBMS) that sits on top of a set of files physically organized as sequential files and/or as some form of direct access files. A DBMS facilitates data access in a database without burdening a user with the details of how the data is physically organized.

1.3 LIMITATIONS OF FILE-PROCESSING SYSTEMS

Computer applications in the 1960s and 1970s focused primarily on automating clerical tasks. These applications made use of records stored in separate files and thus were called file-processing systems. Although file-processing systems for information systems applications have been useful for many years, database technology has rendered them obsolete except for their use in a few legacy systems such as some payroll and customer