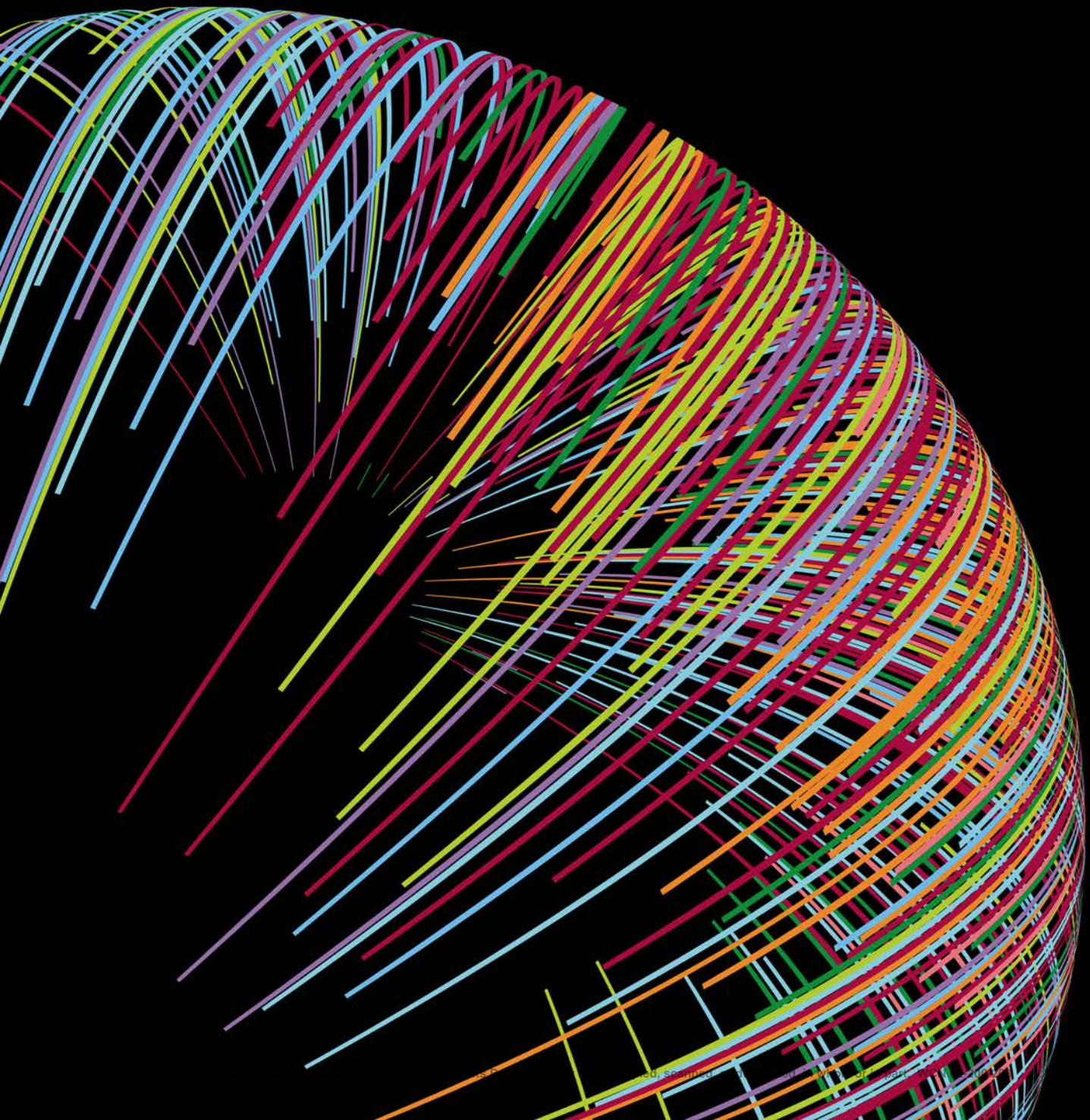


# DATABASE SYSTEMS

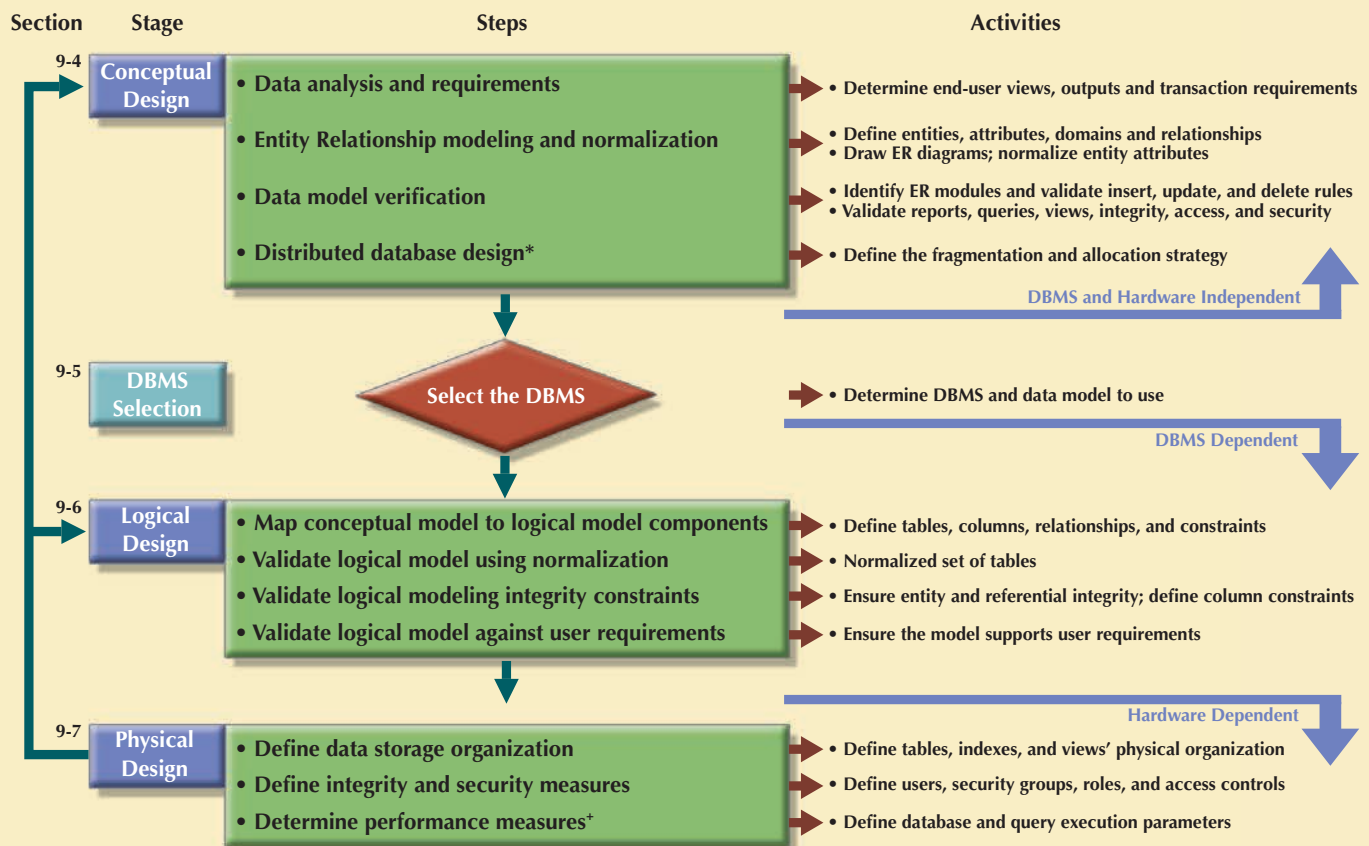
DESIGN, IMPLEMENTATION, & MANAGEMENT

CARLOS CORONEL • STEVEN MORRIS

13TH EDITION



# Database Design Process



\* See Chapter 12, Distributed Database Management Systems

<sup>+</sup> See Chapter 11, Database Performance Tuning and Query Optimization

# Data Modeling Checklist

## BUSINESS RULES

- ✓ Properly document and verify all business rules with the end users.
- ✓ Ensure that all business rules are written precisely, clearly, and simply. The business rules must help identify entities, attributes, relationships, and constraints.
- ✓ Identify the source of all business rules, and ensure that each business rule is justified, dated, and signed off by an approving authority.

## DATA MODELING

**Naming Conventions:** All names should be limited in length (database-dependent size).

### ENTITY NAMES:

- ✓ Should be nouns that are familiar to business and should be short and meaningful
- ✓ Should document abbreviations, synonyms, and aliases for each entity
- ✓ Should be unique within the model
- ✓ For composite entities, may include a combination of abbreviated names of the entities linked through the composite entity

### ATTRIBUTE NAMES:

- ✓ Should be unique within the entity
- ✓ Should use the entity abbreviation as a prefix
- ✓ Should be descriptive of the characteristic
- ✓ Should use suffixes such as \_ID, \_NUM, or \_CODE for the PK attribute
- ✓ Should not be a reserved word
- ✓ Should not contain spaces or special characters such as @, !, or &

### RELATIONSHIP NAMES:

- ✓ Should be active or passive verbs that clearly indicate the nature of the relationship

### Entities:

- ✓ Each entity should represent a single subject.
- ✓ Each entity should represent a set of distinguishable entity instances.
- ✓ All entities should be in 3NF or higher. Any entities below 3NF should be justified.
- ✓ The granularity of the entity instance should be clearly defined.
- ✓ The PK is clearly defined and supports the selected data granularity.

### Attributes:

- ✓ Should be simple and single-valued (atomic data)
- ✓ Should document default values, constraints, synonyms, and aliases
- ✓ Derived attributes should be clearly identified and include source(s)
- ✓ Should not be redundant unless they are justified for transaction accuracy, performance, or maintaining a history
- ✓ Nonkey attributes must be fully dependent on the PK attribute

### Relationships:

- ✓ Should clearly identify relationship participants
- ✓ Should clearly define participation, connectivity, and document cardinality

### ER Model:

- ✓ Should be validated against expected processes: inserts, updates, and deletes
- ✓ Should evaluate where, when, and how to maintain a history
- ✓ Should not contain redundant relationships except as required (see Attributes)
- ✓ Should minimize data redundancy to ensure single-place updates
- ✓ Should conform to the minimal data rule: "All that is needed is there and all that is there is needed."



## Fit your coursework into your hectic life.

Make the most of your time by learning your way. Access the resources you need to succeed wherever, whenever.

- Get more from your time online with an easy-to-follow five-step learning path.
- Stay focused with an all-in-one-place, integrated presentation of course content.
- Get the free MindTap Mobile App and learn wherever you are.

Break limitations. Create your own potential, and be unstoppable with MindTap.

## MINDTAP. POWERED BY YOU.

[cengage.com/mindtap](https://cengage.com/mindtap)





# **DATABASE SYSTEMS**

Design, Implementation,  
and Management

13e

Carlos Coronel | Steven Morris



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

This is an electronic version of the print textbook. Due to electronic rights restrictions, some third party content may be suppressed. Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. The publisher reserves the right to remove content from this title at any time if subsequent rights restrictions require it. For valuable information on pricing, previous editions, changes to current editions, and alternate formats, please visit [www.cengage.com/highered](http://www.cengage.com/highered) to search by ISBN#, author, title, or keyword for materials in your areas of interest.

Important Notice: Media content referenced within the product description or the product text may not be available in the eBook version.

***Database Systems: Design, Implementation,  
and Management, 13th Edition***

**Carlos Coronel and Steven Morris**

SVP, GM Skills: Jonathan Lau

Product Director: Lauren Murphy

Product Team Manager: Kirstin McNary

Associate Product Manager: Kate Mason

Executive Director of Development: Marah  
Bellegarde

Senior Content Development Manager: Leigh  
Hefferon

Content Developer: Maria Garguilo

Product Assistant: Jake Toth

VP, Marketing for Science, Technology, & Math:  
Jason Sakos

Marketing Director: Michele McTighe

Marketing Manager: Stephanie Albracht

Production Director: Patty Stephan

Content Project Manager: Michele Stulga

Art Director: Diana Graham

Cover Designer: Roycroft Design  
(roycroftdesign.com)

Cover Image: iStock.com/liuzishan

© 2019, 2015 Cengage Learning, Inc.

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced or distributed in any form or by any means, except as permitted by U.S. copyright law, without the prior written permission of the copyright owner.

For product information and technology assistance, contact us at  
**Cengage Learning Customer & Sales Support, 1-800-354-9706**

For permission to use material from this text or product, submit all  
requests online at **www.cengage.com/permissions**  
Further permissions questions can be emailed to  
**permissionrequest@cengage.com**

Screenshots for this book were created using Microsoft Access®, Excel®, and Visio® and were used with permission from Microsoft. Microsoft and the Office logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle is a registered trademark, and Oracle12c and MySQL are trademarks of Oracle Corporation.

iPhone, iPad, iTunes, and iPod are registered trademarks of Apple Inc.

Library of Congress Control Number: 2015955694

Student Edition ISBN: 978-1-337-62790-0

Loose Leaf Edition ISBN: 978-1-337-68882-6

**Cengage**

20 Channel Center Street  
Boston, MA 02210  
USA

Cengage Learning is a leading provider of customized learning solutions with employees residing in nearly 40 different countries and sales in more than 125 countries around the world. Find your local representative at **www.cengage.com**.

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

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

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

# Dedication

To the treasures in my life: To Victoria, for 28 wonderful years. Thank you for your unending support and for being my angel, my sweetie, and, most importantly, my best friend. To Carlos Anthony, who has become a remarkable man, pride of his father, and husband to our beautiful, sweet, and smart daughter-in-law, Jered. Thank you for your words of wisdom, hard-working attitude, and for giving us reasons to be happy. You are still young; your best times are still to come. To Gabriela Victoria, who is the image of brilliance, beauty, and faithfulness. The way you give your time and talents in the service of others is an inspiration to all of us. Thank you for being my sunshine on cloudy days. Your future is bright and endless. To Christian Javier, who is smarter than all of us. Thank you for being the youthful reminder of life's simple beauties. Keep challenging yourself to new highs and keep working hard to achieve your dreams. To my parents, Sarah and Carlos, thank you for your sacrifice and example. To all of you, you are all my inspiration. "TQTATA."

## **Carlos Coronel**

To Pamela, from high school sweetheart through nearly 30 years of marriage, you are the beautiful love of my life who has supported, encouraged, and inspired me. More than anyone else, you are responsible for whatever successes I have achieved. To my son, Alexander Logan, your depth of character is without measure. You are my pride and joy. To my daughter, Lauren Elizabeth, your beauty and intensity take my breath away. You are my heart and soul. Thank you all for the sacrifices you have made that enabled me to pursue this dream. I love you so much more than I can express. To my mother, Florence Maryann, and to the memory of my father, Alton Lamar, together they instilled in me the desire to learn and the passion to achieve. To my mother-in-law, Connie Duke, and to the memory of my father-in-law, Wayne Duke, they taught me to find joy in all things. To all of you, with all my love, I dedicate this book.

## **Steven Morris**

### **For Peter**

To longtime colleague and friend, Peter Rob: Your drive and dedication to your students started this book. Your depth of knowledge, attention to detail, and pursuit of excellence made it succeed. Your patience and guidance continue to light our path. It is our sincere hope that, as we move forward, we can continue to live up to your standard. Enjoy your retirement, my friend; you have surely earned it.

## **Carlos Coronel and Steven Morris**



# Brief Contents

Preface, xv	
Text Features, xx	
Additional Features, xxii	
Acknowledgments, xxiv	
<b>Part 1: Database Concepts</b>	<b>1</b>
1. Database Systems, 2	
2. Data Models, 34	
<b>Part 2: Design Concepts</b>	<b>67</b>
3. The Relational Database Model, 68	
4. Entity Relationship (ER) Modeling, 113	
5. Advanced Data Modeling, 167	
6. Normalization of Database Tables, 199	
<b>Part 3: Advanced Design and Implementation</b>	<b>243</b>
7. Introduction to Structured Query Language (SQL), 244	
8. Advanced SQL, 359	
9. Database Design, 439	
<b>Part 4: Advanced Database Concepts</b>	<b>481</b>
10. Transaction Management and Concurrency Control, 482	
11. Database Performance Tuning and Query Optimization, 515	
12. Distributed Database Management Systems, 553	
13. Business Intelligence and Data Warehouses, 589	
14. Big Data and NoSQL, 657	
<b>Part 5: Databases and the Internet</b>	<b>691</b>
15. Database Connectivity and Web Technologies, 692	
<b>Part 6: Database Administration</b>	<b>733</b>
16. Database Administration and Security, 734	
Glossary, 782	
Index, 793	

The following appendices are included on the Instructor and Student Companion Sites at [www.cengagebrain.com](http://www.cengagebrain.com).

- Appendix A1:** Designing Databases with Visio Professional 2010: A Tutorial
- Appendix A2:** Designing Databases with Visio 2013: A Tutorial
- Appendix B:** The University Lab: Conceptual Design
- Appendix C:** The University Lab: Conceptual Design Verification, Logical Design, and Implementation
- Appendix D:** Converting an ER Model into a Database Structure
- Appendix E:** Comparison of ER Modeling Notations
- Appendix F:** Client/Server Systems
- Appendix G:** Object-Oriented Databases
- Appendix H:** Unified Modeling Language (UML)
- Appendix I:** Databases in Electronic Commerce
- Appendix J:** Web Database Development with ColdFusion
- Appendix K:** The Hierarchical Database Model
- Appendix L:** The Network Database Model
- Appendix M:** MS Access Tutorial
- Appendix N:** Creating a New Database Using Oracle 12c
- Appendix O:** Data Warehouse Implementation Factors
- Appendix P:** Working with MongoDB
- Appendix Q:** Working with Neo4j

# Contents

Preface, xv

Text Features, xx

Additional Features, xxii

Acknowledgments, xxiv

## Part 1: Database Concepts 1

### Chapter 1: Database Systems 2

- 1-1** Why Databases? 3
- 1-2** Data versus Information 4
- 1-3** Introducing the Database 6
  - 1-3a Role and Advantages of the DBMS 7
  - 1-3b Types of Databases 9
- 1-4** Why Database Design Is Important 12
- 1-5** Evolution of File System Data Processing 15
  - 1-5a Manual File Systems 15
  - 1-5b Computerized File Systems 15
  - 1-5c File System Redux: Modern End-User Productivity Tools 18
- 1-6** Problems with File System Data Processing 18
  - 1-6a Structural and Data Dependence 19
  - 1-6b Data Redundancy 20
  - 1-6c Data Anomalies 21
- 1-7** Database Systems 21
  - 1-7a The Database System Environment 22
  - 1-7b DBMS Functions 24
  - 1-7c Managing the Database System: A Shift in Focus 27
- 1-8** Preparing for Your Database Professional Career 28
  - Summary 30 • Key Terms 31 • Review Questions 31 • Problems 32

### Chapter 2: Data Models 34

- 2-1** Data Modeling and Data Models 35
- 2-2** The Importance of Data Models 36
- 2-3** Data Model Basic Building Blocks 36
- 2-4** Business Rules 38
  - 2-4a Discovering Business Rules 38
  - 2-4b Translating Business Rules into Data Model Components 39
  - 2-4c Naming Conventions 40
- 2-5** The Evolution of Data Models 40
  - 2-5a Hierarchical and Network Models 40
  - 2-5b The Relational Model 42
  - 2-5c The Entity Relationship Model 44
  - 2-5d The Object-Oriented Model 47
  - 2-5e Object/Relational and XML 48
  - 2-5f Emerging Data Models: Big Data and NoSQL 49
  - 2-5g Data Models: A Summary 53
- 2-6** Degrees of Data Abstraction 54
  - 2-6a The External Model 57
  - 2-6b The Conceptual Model 58
  - 2-6c The Internal Model 59
  - 2-6d The Physical Model 60
  - Summary 61 • Key Terms 62 • Review Questions 62 • Problems 63

## Part 2: Design Concepts 67

### Chapter 3: The Relational Database Model 68

- 3-1 A Logical View of Data 69**
  - 3-1a Tables and Their Characteristics 69
- 3-2 Keys 72**
  - 3-2a Dependencies 72
  - 3-2b Types of Keys 73
- 3-3 Integrity Rules 76**
- 3-4 Relational Algebra 78**
  - 3-4a Formal Definitions and Terminology 78
  - 3-4b Relational Set Operators 79
- 3-5 The Data Dictionary and the System Catalog 87**
- 3-6 Relationships within the Relational Database 89**
  - 3-6a The 1:M Relationship 89
  - 3-6b The 1:1 Relationship 91
  - 3-6c The M:N Relationship 93
- 3-7 Data Redundancy Revisited 97**
- 3-8 Indexes 99**
- 3-9 Codd's Relational Database Rules 100**
  - Summary 102 • Key Terms 103 • Review Questions 103 • Problems 106

### Chapter 4: Entity Relationship (ER) Modeling 113

- 4-1 The Entity Relationship Model 114**
  - 4-1a Entities 114
  - 4-1b Attributes 114
  - 4-1c Relationships 120
  - 4-1d Connectivity and Cardinality 121
  - 4-1e Existence Dependence 122
  - 4-1f Relationship Strength 123
  - 4-1g Weak Entities 125
  - 4-1h Relationship Participation 127
  - 4-1i Relationship Degree 131
  - 4-1j Recursive Relationships 133
  - 4-1k Associative (Composite) Entities 136
- 4-2 Developing an ER Diagram 138**
- 4-3 Database Design Challenges: Conflicting Goals 146**
  - Summary 150 • Key Terms 151 • Review Questions 151 • Problems 154 • Cases 159

### Chapter 5: Advanced Data Modeling 167

- 5-1 The Extended Entity Relationship Model 168**
  - 5-1a Entity Supertypes and Subtypes 168
  - 5-1b Specialization Hierarchy 169
  - 5-1c Inheritance 170
  - 5-1d Subtype Discriminator 172
  - 5-1e Disjoint and Overlapping Constraints 172
  - 5-1f Completeness Constraint 174
  - 5-1g Specialization and Generalization 175
- 5-2 Entity Clustering 175**
- 5-3 Entity Integrity: Selecting Primary Keys 176**
  - 5-3a Natural Keys and Primary Keys 177
  - 5-3b Primary Key Guidelines 177
  - 5-3c When to Use Composite Primary Keys 177
  - 5-3d When to Use Surrogate Primary Keys 179
- 5-4 Design Cases: Learning Flexible Database Design 180**
  - 5-4a Design Case 1: Implementing 1:1 Relationships 181
  - 5-4b Design Case 2: Maintaining History of Time-Variant Data 182
  - 5-4c Design Case 3: Fan Traps 185
  - 5-4d Design Case 4: Redundant Relationships 186
  - Summary 187 • Key Terms 187 • Review Questions 188 • Problems 189 • Cases 190

## Chapter 6: Normalization of Database Tables 199

- 6-1 Database Tables and Normalization 200**
- 6-2 The Need for Normalization 200**
- 6-3 The Normalization Process 203**
  - 6-3a Conversion to First Normal Form (1NF) 205**
  - 6-3b Conversion to Second Normal Form (2NF) 209**
  - 6-3c Conversion to Third Normal Form (3NF) 211**
- 6-4 Improving the Design 213**
- 6-5 Surrogate Key Considerations 217**
- 6-6 Higher-Level Normal Forms 218**
  - 6-6a The Boyce-Codd Normal Form 219**
  - 6-6b Fourth Normal Form (4NF) 222**
- 6-7 Normalization and Database Design 224**
- 6-8 Denormalization 227**
- 6-9 Data-Modeling Checklist 230**
  - Summary 232 • Key Terms 233 • Review Questions 233 • Problems 235

## Part 3: Advanced Design and Implementation 243

### Chapter 7: Introduction to Structured Query Language (SQL) 244

- 7-1 Introduction to SQL 245**
  - 7-1a Data Types 245**
  - 7-1b SQL Queries 247**
  - 7-1c The Database Model 248**
- 7-2 Basic SELECT Queries 249**
- 7-3 SELECT Statement Options 250**
  - 7-3a Using Column Aliases 251**
  - 7-3b Using Computed Columns 253**
  - 7-3c Arithmetic Operators: The Rule of Precedence 254**
  - 7-3d Date Arithmetic 255**
  - 7-3e Listing Unique Values 255**
- 7-4 FROM Clause Options 256**
  - 7-4a Natural Join 257**
  - 7-4b JOIN USING Syntax 259**
  - 7-4c JOIN ON Syntax 260**
  - 7-4d Common Attribute Names 261**
  - 7-4e Outer Joins 261**
  - 7-4f Cross Join 264**
  - 7-4g Joining Tables with an Alias 264**
  - 7-4h Recursive Joins 265**
- 7-5 ORDER BY Clause Options 266**
- 7-6 WHERE Clause Options 269**
  - 7-6a Selecting Rows with Conditional Restrictions 269**
  - 7-6b Using Comparison Operators on Character Attributes 271**
  - 7-6c Using Comparison Operators on Dates 272**
  - 7-6d Logical Operators: AND, OR, and NOT 273**
  - 7-6e Old-Style Joins 275**
  - 7-6f Special Operators 276**
- 7-7 Aggregate Processing 281**
  - 7-7a Aggregate Functions 281**
  - 7-7b Grouping Data 285**
  - 7-7c HAVING Clause 288**
- 7-8 Subqueries 290**
  - 7-8a WHERE Subqueries 292**
  - 7-8b IN Subqueries 293**
  - 7-8c HAVING Subqueries 294**
  - 7-8d Multirow Subquery Operators: ALL and ANY 294**
  - 7-8e FROM Subqueries 295**
  - 7-8f Attribute List Subqueries 296**
  - 7-8g Correlated Subqueries 298**
- 7-9 SQL Functions 302**
  - 7-9a Date and Time Functions 302**
  - 7-9b Numeric Functions 306**

- 7-9c String Functions 307
  - 7-9d Conversion Functions 309
  - 7-10 Relational Set Operators 311**
    - 7-10a UNION 311
    - 7-10b UNION ALL 313
    - 7-10c INTERSECT 314
    - 7-10d EXCEPT (MINUS) 315
    - 7-10e Syntax Alternatives 316
  - 7-11 Crafting SELECT Queries 317**
    - 7-11a Know Your Data 317
    - 7-11b Know the Problem 317
    - 7-11c Build One Clause at a Time 318
- Summary 319 • Key Terms 321 • Review Questions 321 • Problems 323

## Chapter 8: Advanced SQL 359

- 8-1 Data Definition Commands 360**
  - 8-1a Starting Database Model 360
  - 8-1b Creating the Database 361
  - 8-1c The Database Schema 362
  - 8-1d Data Types 362
- 8-2 Creating Table Structures 366**
  - 8-2a CREATE TABLE command 366
  - 8-2b SQL Constraints 370
  - 8-2c Create a Table with a SELECT Statement 373
  - 8-2d SQL Indexes 374
- 8-3 Altering Table Structures 375**
  - 8-3a Changing a Column's Data Type 376
  - 8-3b Changing a Column's Data Characteristics 376
  - 8-3c Adding a Column 377
  - 8-3d Adding Primary Key, Foreign Key, and Check Constraints 377
  - 8-3e Dropping a Column 378
  - 8-3f Deleting a Table from the Database 378
- 8-4 Data Manipulation Commands 379**
  - 8-4a Adding Table Rows 379
  - 8-4b Inserting Table Rows with a SELECT Subquery 381
  - 8-4c Saving Table Changes 382
  - 8-4d Updating Table Rows 383
  - 8-4e Deleting Table Rows 385
  - 8-4f Restoring Table Contents 386
- 8-5 Virtual Tables: Creating a View 387**
  - 8-5a Updatable Views 388
- 8-6 Sequences 391**
- 8-7 Procedural SQL 396**
  - 8-7a Triggers 401
  - 8-7b Stored Procedures 411
  - 8-7c PL/SQL Processing with Cursors 416
  - 8-7d PL/SQL Stored Functions 418
- 8-8 Embedded SQL 419**
  - Summary 423 • Key Terms 425 • Review Questions 425 • Problems 426 • Cases 433

## Chapter 9: Database Design 439

- 9-1 The Information System 440**
- 9-2 The Systems Development Life Cycle 442**
  - 9-2a Planning 442
  - 9-2b Analysis 443
  - 9-2c Detailed Systems Design 444
  - 9-2d Implementation 444
  - 9-2e Maintenance 445
- 9-3 The Database Life Cycle 445**
  - 9-3a The Database Initial Study 445
  - 9-3b Database Design 450
  - 9-3c Implementation and Loading 451
  - 9-3d Testing and Evaluation 454

- 9-3e Operation 456
- 9-3f Maintenance and Evolution 457
- 9-4 Conceptual Design 457**
  - 9-4a Data Analysis and Requirements 459
  - 9-4b Entity Relationship Modeling and Normalization 461
  - 9-4c Data Model Verification 464
  - 9-4d Distributed Database Design 467
- 9-5 DBMS Software Selection 467**
- 9-6 Logical Design 468**
  - 9-6a Map the Conceptual Model to the Logical Model 468
  - 9-6b Validate the Logical Model Using Normalization 470
  - 9-6c Validate Logical Model Integrity Constraints 470
  - 9-6d Validate the Logical Model against User Requirements 471
- 9-7 Physical Design 471**
  - 9-7a Define Data Storage Organization 472
  - 9-7b Define Integrity and Security Measures 472
  - 9-7c Determine Performance Measures 473
- 9-8 Database Design Strategies 473**
- 9-9 Centralized versus Decentralized Design 474**
  - Summary 477 • Key Terms 477 • Review Questions 477 • Problems 478

## Part 4: Advanced Database Concepts 481

### Chapter 10: Transaction Management and Concurrency Control 482

- 10-1 What Is a Transaction? 483**
  - 10-1a Evaluating Transaction Results 484
  - 10-1b Transaction Properties 487
  - 10-1c Transaction Management with SQL 488
  - 10-1d The Transaction Log 489
- 10-2 Concurrency Control 490**
  - 10-2a Lost Updates 490
  - 10-2b Uncommitted Data 491
  - 10-2c Inconsistent Retrievals 492
  - 10-2d The Scheduler 493
- 10-3 Concurrency Control with Locking Methods 495**
  - 10-3a Lock Granularity 496
  - 10-3b Lock Types 498
  - 10-3c Two-Phase Locking to Ensure Serializability 500
  - 10-3d Deadlocks 500
- 10-4 Concurrency Control with Time Stamping Methods 502**
  - 10-4a Wait/Die and Wound/Wait Schemes 502
- 10-5 Concurrency Control with Optimistic Methods 503**
- 10-6 ANSI Levels of Transaction Isolation 504**
- 10-7 Database Recovery Management 506**
  - 10-7a Transaction Recovery 506
- Summary 510 • Key Terms 511 • Review Questions 511 • Problems 512

### Chapter 11: Database Performance Tuning and Query Optimization 515

- 11-1 Database Performance-Tuning Concepts 516**
  - 11-1a Performance Tuning: Client and Server 517
  - 11-1b DBMS Architecture 518
  - 11-1c Database Query Optimization Modes 520
  - 11-1d Database Statistics 521
- 11-2 Query Processing 522**
  - 11-2a SQL Parsing Phase 523
  - 11-2b SQL Execution Phase 524
  - 11-2c SQL Fetching Phase 525
  - 11-2d Query Processing Bottlenecks 525
- 11-3 Indexes and Query Optimization 526**
- 11-4 Optimizer Choices 528**
  - 11-4a Using Hints to Affect Optimizer Choices 530

- 11-5 SQL Performance Tuning 531**
  - 11-5a Index Selectivity 531
  - 11-5b Conditional Expressions 533
- 11-6 Query Formulation 534**
- 11-7 DBMS Performance Tuning 536**
- 11-8 Query Optimization Example 538**
  - Summary 546 • Key Terms 547 • Review Questions 547 • Problems 548

## Chapter 12: Distributed Database Management Systems 553

- 12-1 The Evolution of Distributed Database Management Systems 554**
- 12-2 DDBMS Advantages and Disadvantages 556**
- 12-3 Distributed Processing and Distributed Databases 556**
- 12-4 Characteristics of Distributed Database Management Systems 559**
- 12-5 DDBMS Components 560**
- 12-6 Levels of Data and Process Distribution 561**
  - 12-6a Single-Site Processing, Single-Site Data 561
  - 12-6b Multiple-Site Processing, Single-Site Data 562
  - 12-6c Multiple-Site Processing, Multiple-Site Data 563
- 12-7 Distributed Database Transparency Features 564**
- 12-8 Distribution Transparency 565**
- 12-9 Transaction Transparency 568**
  - 12-9a Distributed Requests and Distributed Transactions 568
  - 12-9b Distributed Concurrency Control 571
  - 12-9c Two-Phase Commit Protocol 571
- 12-10 Performance and Failure Transparency 573**
- 12-11 Distributed Database Design 575**
  - 12-11a Data Fragmentation 575
  - 12-11b Data Replication 578
  - 12-11c Data Allocation 580
- 12-12 The CAP Theorem 581**
- 12-13 C. J. Date's 12 Commandments for Distributed Databases 583**
  - Summary 584 • Key Terms 585 • Review Questions 585 • Problems 586

## Chapter 13: Business Intelligence and Data Warehouses 589

- 13-1 The Need for Data Analysis 590**
- 13-2 Business Intelligence 590**
  - 13-2a Business Intelligence Architecture 592
  - 13-2b Business Intelligence Benefits 596
  - 13-2c Business Intelligence Evolution 597
  - 13-2d Business Intelligence Technology Trends 600
- 13-3 Decision Support Data 601**
  - 13-3a Operational Data versus Decision Support Data 601
  - 13-3b Decision Support Database Requirements 604
- 13-4 The Data Warehouse 606**
  - 13-4a Data Marts 609
  - 13-4b Twelve Rules That Define a Data Warehouse 609
- 13-5 Star Schemas 609**
  - 13-5a Facts 610
  - 13-5b Dimensions 610
  - 13-5c Attributes 611
  - 13-5d Attribute Hierarchies 613
  - 13-5e Star Schema Representation 615
  - 13-5f Performance-Improving Techniques for the Star Schema 616
- 13-6 Online Analytical Processing 620**
  - 13-6a Multidimensional Data Analysis Techniques 620
  - 13-6b Advanced Database Support 622
  - 13-6c Easy-to-Use End-User Interfaces 622
  - 13-6d OLAP Architecture 622
  - 13-6e Relational OLAP 625
  - 13-6f Multidimensional OLAP 627
  - 13-6g Relational versus Multidimensional OLAP 627



- 13-7 Data Analytics 628**
    - 13-7a Data Mining 629
    - 13-7b Predictive Analytics 631
  - 13-8 SQL Analytic Functions 632**
    - 13-8a The ROLLUP Extension 633
    - 13-8b The CUBE Extension 634
    - 13-8c Materialized Views 636
  - 13-9 Data Visualization 639**
    - 13-9a The Need for Data Visualization 640
    - 13-9b The Science of Data Visualization 642
    - 13-9c Understanding the Data 644
- Summary 645 • Key Terms 646 • Review Questions 647 • Problems 648

## Chapter 14: Big Data and NoSQL 657

- 14-1 Big Data 658**
    - 14-1a Volume 660
    - 14-1b Velocity 661
    - 14-1c Variety 662
    - 14-1d Other Characteristics 663
  - 14-2 Hadoop 664**
    - 14-2a HDFS 665
    - 14-2b MapReduce 667
    - 14-2c Hadoop Ecosystem 669
  - 14-3 NoSQL 672**
    - 14-3a Key-Value Databases 673
    - 14-3b Document Databases 674
    - 14-3c Column-Oriented Databases 675
    - 14-3d Graph Databases 677
    - 14-3e Aggregate Awareness 679
  - 14-4 NewSQL Databases 680**
  - 14-5 Working with Document Databases Using MongoDB 680**
    - 14-5a Importing Documents in MongoDB 682
    - 14-5b Example of a MongoDB Query Using find() 683
  - 14-6 Working with Graph Databases Using Neo4j 684**
    - 14-6a Creating Nodes in Neo4j 685
    - 14-6b Retrieving Node Data with MATCH and WHERE 686
    - 14-6c Retrieving Relationship Data with MATCH and WHERE 686
- Summary 688 • Key Terms 689 • Review Questions 690

## Part 5: Databases and the Internet 691

### Chapter 15: Database Connectivity and Web Technologies 692

- 15-1 Database Connectivity 693**
  - 15-1a Native SQL Connectivity 694
  - 15-1b ODBC, DAO, and RDO 695
  - 15-1c OLE-DB 697
  - 15-1d ADO.NET 699
  - 15-1e Java Database Connectivity (JDBC) 703
- 15-2 Database Internet Connectivity 704**
  - 15-2a Web-to-Database Middleware: Server-Side Extensions 705
  - 15-2b Web Server Interfaces 707
  - 15-2c The Web Browser 708
  - 15-2d Client-Side Extensions 709
  - 15-2e Web Application Servers 710
  - 15-2f Web Database Development 711
- 15-3 Extensible Markup Language (XML) 715**
  - 15-3a Document Type Definitions (DTD) and XML Schemas 717
  - 15-3b XML Presentation 719
  - 15-3c XML Applications 721

- 15-4 Cloud Computing Services 722**
  - 15-4a Cloud Implementation Types 725
  - 15-4b Characteristics of Cloud Services 725
  - 15-4c Types of Cloud Services 726
  - 15-4d Cloud Services: Advantages and Disadvantages 727
  - 15-4e SQL Data Services 729
- Summary 730 • Key Terms 731 • Review Questions 731 • Problems 732

## Part 6: Database Administration 733

### Chapter 16: Database Administration and Security 734

- 16-1 Data as a Corporate Asset 735**
- 16-2 The Need for a Database and Its Role in an Organization 736**
- 16-3 Introduction of a Database: Special Considerations 738**
- 16-4 The Evolution of Database Administration 739**
- 16-5 The Database Environment's Human Component 743**
  - 16-5a The DBA's Managerial Role 745
  - 16-5b The DBA's Technical Role 750
- 16-6 Security 757**
  - 16-6a Security Policies 758
  - 16-6b Security Vulnerabilities 758
  - 16-6c Database Security 760
- 16-7 Database Administration Tools 761**
  - 16-7a The Data Dictionary 762
  - 16-7b Case Tools 764
- 16-8 Developing a Data Administration Strategy 767**
- 16-9 The DBA's Role in the Cloud 768**
- 16-10 The DBA at Work: Using Oracle for Database Administration 769**
  - 16-10a Oracle Database Administration Tools 770
  - 16-10b Ensuring That the RDBMS Starts Automatically 770
  - 16-10c Creating Tablespaces and Datafiles 772
  - 16-10d Managing Users and Establishing Security 774
  - 16-10e Customizing the Database Initialization Parameters 776
- Summary 777 • Key Terms 779 • Review Questions 779
- Glossary 782
- Index 793

The following appendices are included on the Instructor and Student Companion Sites at [www.cengagebrain.com](http://www.cengagebrain.com).

**Appendix A1:** Designing Databases with Visio Professional 2010: A Tutorial

**Appendix A2:** Designing Databases with Visio 2013: A Tutorial

**Appendix B:** The University Lab: Conceptual Design

**Appendix C:** The University Lab: Conceptual Design Verification, Logical Design, and Implementation

**Appendix D:** Converting an ER Model into a Database Structure

**Appendix E:** Comparison of ER Modeling Notations

**Appendix F:** Client/Server Systems

**Appendix G:** Object-Oriented Databases

**Appendix H:** Unified Modeling Language (UML)

**Appendix I:** Databases in Electronic Commerce

**Appendix J:** Web Database Development with ColdFusion

**Appendix K:** The Hierarchical Database Model

**Appendix L:** The Network Database Model

**Appendix M:** MS Access Tutorial

**Appendix N:** Creating a New Database Using Oracle 12c

**Appendix O:** Data Warehouse Implementation Factors

**Appendix P:** Working with MongoDB

**Appendix Q:** Working with Neo4j

# Preface

It is our great pleasure to present the thirteenth edition of *Database Systems*. We are grateful and humbled that so many of our colleagues around the world have chosen this text to support their classes. We wrote the first edition of this book because we wanted to explain the complexity of database systems in a language that was easy for students to understand. Over the years, we have maintained this emphasis on reaching out to students to explain complex concepts in a practical, approachable manner. This book has been successful through twelve editions because the authors, editors, and the publisher paid attention to the impact of technology and to adopters' questions and suggestions. We believe that this thirteenth edition successfully reflects the same attention to such factors.

In many respects, rewriting a book is more difficult than writing it the first time. If the book is successful, as this one is, a major concern is that the updates, inserts, and deletions will adversely affect writing style and continuity of coverage. The combination of superb reviewers and editors, plus a wealth of feedback from adopters and students of the previous editions, helped make this new edition the best yet.

## Changes to the Thirteenth Edition

In this thirteenth edition, we have responded to the requests and suggestions of numerous adopters. We have substantially reorganized the SQL coverage to make the presentation easier to follow and easier to reference. We start with simple SQL statements to familiarize students with the basic SQL syntax and environment. This provides students the confidence to transition to the more advanced SQL features and commands. These changes provide a better flow of material. Additionally, more SQL examples and figures have been added to help students better visualize and understand the code that is presented.

Aside from enhancing the already strong coverage of database design, we made other improvements in the topical coverage. In particular, the continued growth of Big Data and NoSQL technologies continue to challenge the status quo in the database industry. Therefore, we have added two new online appendices on MongoDB and Neo4j, two of the most important of the NoSQL offerings. This new material provides coding examples and data files to allow students to gain hands-on experience using two of the most popular NoSQL databases. The thirteenth edition also presents a major step forward in the integration of digital content with the text through online, automatically graded coding labs that allow students to write SQL code in an interactive environment that can grade and provide feedback on problems. Here are a few of the highlights of changes in the thirteenth edition:

- Streamlined and reorganized coverage of normalization for enhanced clarity
- Complete reorganization of SQL and Advanced SQL chapters to improve flow and make references to keywords and techniques easier to access
- Expanded coverage of MongoDB with hands-on exercises for querying MongoDB databases (Appendix P)
- Expanded coverage of Neo4j with hands-on exercises for querying graph databases using Cypher (Appendix Q)
- New and expanded coverage of data visualization tools and techniques

This thirteenth edition continues to provide a solid and practical foundation for the design, implementation, and management of database systems. This foundation is built on the notion that, while databases are very practical, their successful creation depends on understanding the important concepts that define them. It's not easy to come up with the proper mix of theory and practice, but the previously mentioned feedback suggests that we largely succeeded in our quest to maintain the proper balance.

# The Approach: A Continued Emphasis on Design

As the title suggests, *Database Systems: Design, Implementation, and Management* covers three broad aspects of database systems. However, for several important reasons, special attention is given to database design.

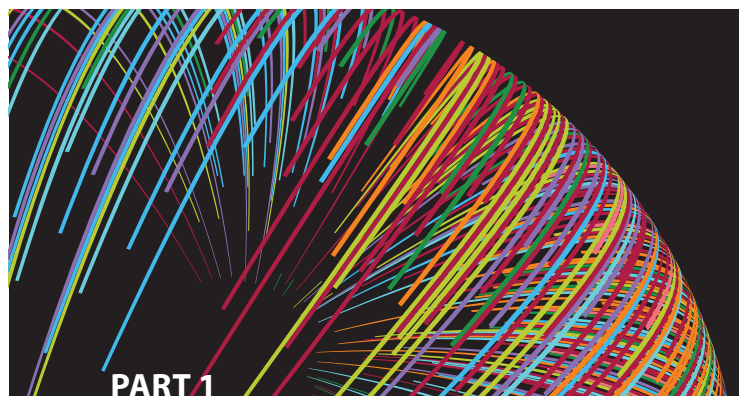
- The availability of excellent database software enables people with little experience to create databases and database applications. Unfortunately, the “create without design” approach usually paves the road to a number of database disasters. In our experience, many database system failures are traceable to poor design and cannot be solved with the help of even the best programmers and managers. Nor is better DBMS software likely to overcome problems created or magnified by poor design. Even the best bricklayers and carpenters can’t create a good building from a bad blueprint.
- Most vexing problems of database system management seem to be triggered by poorly designed databases. It hardly seems worthwhile to use scarce resources to develop excellent database management skills merely to use them on crises induced by poorly designed databases.
- Design provides an excellent means of communication. Clients are more likely to get what they need when database system design is approached carefully and thoughtfully. In fact, clients may discover how their organizations really function once a good database design is completed.
- Familiarity with database design techniques promotes understanding of current database technologies. For example, because data warehouses derive much of their data from operational databases, data warehouse concepts, structures, and procedures make more sense when the operational database’s structure and implementation are understood.

Because the practical aspects of database design are stressed, we have covered design concepts and procedures in detail, making sure that the numerous end-of-chapter problems and cases are sufficiently challenging, so students can develop real and useful design skills. We also make sure that students understand the potential and actual conflicts between database design elegance, information requirements, and transaction processing speed. For example, it makes little sense to design databases that meet design elegance standards while they fail to meet end-user information requirements. Therefore, we explore the use of carefully defined trade-offs to ensure that the databases meet end-user requirements while conforming to high design standards.

## Topical Coverage

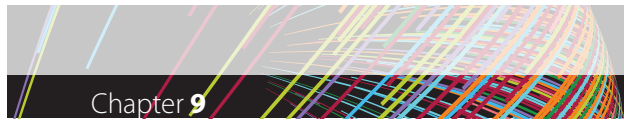
### The Systems View

The book’s title begins with *Database Systems*. Therefore, we examine the database and design concepts covered in Chapters 1–6 as part of a larger whole by placing them within the systems analysis framework of Chapter 9. Database designers who fail to understand that the database is part of a larger system are likely to overlook important design requirements. In fact, Chapter 9, Database Design, provides the map for the advanced database design developed in Appendices B and C. Within the larger systems framework, we can also explore issues such as transaction management and concurrency control (Chapter 10), distributed database management systems (Chapter 12), business intelligence and data warehouses (Chapter 13), new technologies for Big Data (Chapter 14), database connectivity and web technologies (Chapter 15), and database administration and security (Chapter 16).



### PART 1 Database Concepts

- 1 Database Systems
- 2 Data Models



## Chapter 9

### Database Design

#### After completing this chapter, you will be able to:

- Describe the role of database design as the foundation of a successful information system
- Describe the five phases in the Systems Development Life Cycle (SDLC)
- Design databases using the six phases in the Database Life Cycle (DBLC) framework
- Conduct evaluation and revision within the SDLC and DBLC frameworks
- Distinguish between top-down and bottom-up approaches in database design
- Distinguish between centralized and decentralized conceptual database design

#### Preview

Databases are a part of a larger picture called an information system. Database designs that fail to recognize this fact are not likely to be successful. Database designers must recognize that the database is a critical means to an end rather than an end in itself. Managers want the database to serve their management needs, but too many databases seem to force managers to alter their routines to fit the database requirements.

Information systems don't just happen; they are the product of a carefully staged development process. Systems analysis is used to determine the need for an information system and to establish its limits. Within systems analysis, the actual information system is created through a process known as systems development.

The creation and evolution of information systems follows an iterative pattern called the Systems Development Life Cycle (SDLC), which is a continuous process of creation, maintenance, enhancement, and replacement of the information system. A similar cycle applies to databases: the database is created, maintained, enhanced, and eventually replaced. The Database Life Cycle (DBLC) is carefully traced in this chapter, and is shown in the context of the larger Systems Development Life Cycle.

At the end of the chapter, you will be introduced to some classical approaches to database design: top-down versus bottom-up and centralized versus decentralized.

Data Files Available on [cengagebrain.com](http://cengagebrain.com)



#### Note

Because it is purely conceptual, this chapter does not reference any data files.

## Database Design

The first item in the book's subtitle is *Design*, and our examination of database design is comprehensive. For example, Chapters 1 and 2 examine the development and future of databases and data models and illustrate the need for design. Chapter 3 examines the details of the relational database model; Chapter 4 provides extensive, in-depth, and practical database design coverage; and Chapter 5 explores advanced database design topics. Chapter 6 is devoted to critical normalization issues that affect database efficiency and effectiveness. Chapter 9 examines database design within the systems framework and maps the activities required to successfully design and implement the complex, real-world database developed in Appendices B and C. Appendices A1 and A2 are good introductory tutorials on designing databases with Visio Professional 2010 and Visio 2013, respectively.

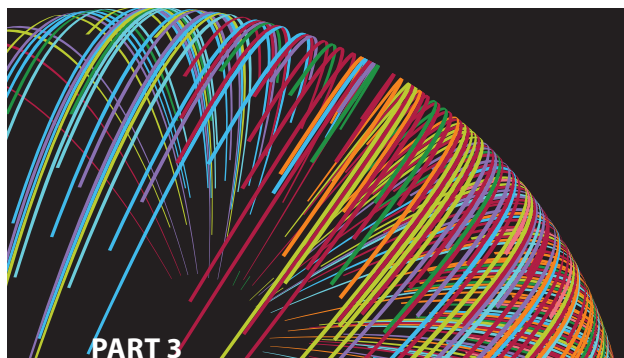
Because database design is affected by real-world transactions, the way data is distributed, and ever-increasing information requirements, we examine major database features that must be supported in current-generation databases and models. For example, Chapter 10, Transaction Management and Concurrency Control, focuses on the characteristics of database transactions and how they affect database integrity and consistency. Chapter 11, Database Performance Tuning and Query Optimization, illustrates the need for query efficiency in a world that routinely generates and uses terabyte-size databases and tables with millions of records. Chapter 12, Distributed Database Management Systems, focuses on data distribution, replication, and allocation. In Chapter 13, Business Intelligence and Data Warehouses, we explore the characteristics of databases that are used in decision support and online analytical processing, including coverage of data visualization and data analytics. Chapter 14, Big Data and NoSQL, explores the challenges of leveraging nonrelational databases to use vast global stores of unstructured data. Chapter 15, Database Connectivity and Web Technologies, covers the basic database connectivity issues in a web-based data world, development of web-based database front ends, and emerging cloud-based services.

Chapter 11, Database Performance Tuning and Query Optimization, illustrates the need for query efficiency in a world that routinely generates and uses terabyte-size databases and tables with millions of records. Chapter 12, Distributed Database Management Systems, focuses on data distribution, replication, and allocation. In Chapter 13, Business Intelligence and Data Warehouses, we explore the characteristics of databases that are used in decision support and online analytical processing, including coverage of data visualization and data analytics. Chapter 14, Big Data and NoSQL, explores the challenges of leveraging nonrelational databases to use vast global stores of unstructured data. Chapter 15, Database Connectivity and Web Technologies, covers the basic database connectivity issues in a web-based data world, development of web-based database front ends, and emerging cloud-based services.

## Implementation

The second portion of the subtitle is *Implementation*. We use Structured Query Language (SQL) in Chapters 7 and 8 to show how relational databases are implemented and managed. Appendix M, Microsoft Access Tutorial, provides a quick but comprehensive guide to implementing an MS Access database. Appendices B and C demonstrate the design of a database that was fully implemented; these appendices illustrate a wide range of implementation issues. We had to deal with conflicting design goals: design elegance, information requirements, and operational speed. Therefore, we carefully audited the initial design in Appendix B to check its ability to meet end-user needs and establish appropriate implementation protocols.

The result of this audit yielded the final design developed in Appendix C. While relational databases are still the appropriate database technology to use in the vast majority of situations, Big Data issues have created an environment in which special requirements can call for the use of new, nonrelational technologies. Chapter 14, Big Data and NoSQL, describes the types of data that are appropriate for these new technologies and the array of options available in these special cases. Appendix P, Working with MongoDB, and Appendix Q, Working with Neo4j, provide hands-on coverage of using MongoDB and Neo4j, some of the most popular NoSQL options. The



## PART 3

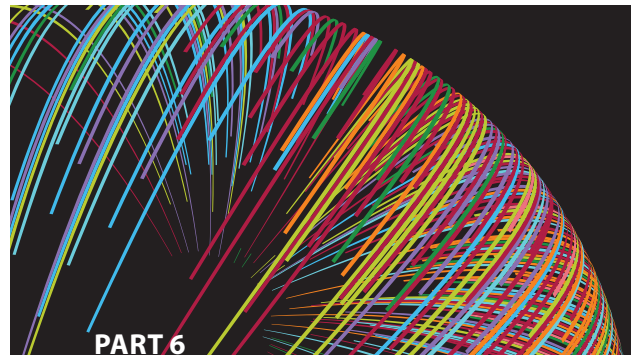
### Advanced Design and Implementation

- 7 Introduction to Structured Query Language (SQL)
- 8 Advanced SQL
- 9 Database Design

special issues encountered in an Internet database environment are addressed in Chapter 15, Database Connectivity and Web Technologies, and in Appendix J, Web Database Development with ColdFusion.

## Management

The final portion of the subtitle is *Management*. We deal with database management issues in Chapter 10, Transaction Management and Concurrency Control; Chapter 12, Distributed Database Management Systems; and Chapter 16, Database Administration and Security. Chapter 11, Database Performance Tuning and Query Optimization, is a valuable resource that illustrates how a DBMS manages data retrieval. In addition, Appendix N, Creating a New Database Using Oracle 12c, walks you through the process of setting up a new database.



**PART 6**

**Database Administration**

**16** Database Administration and Security

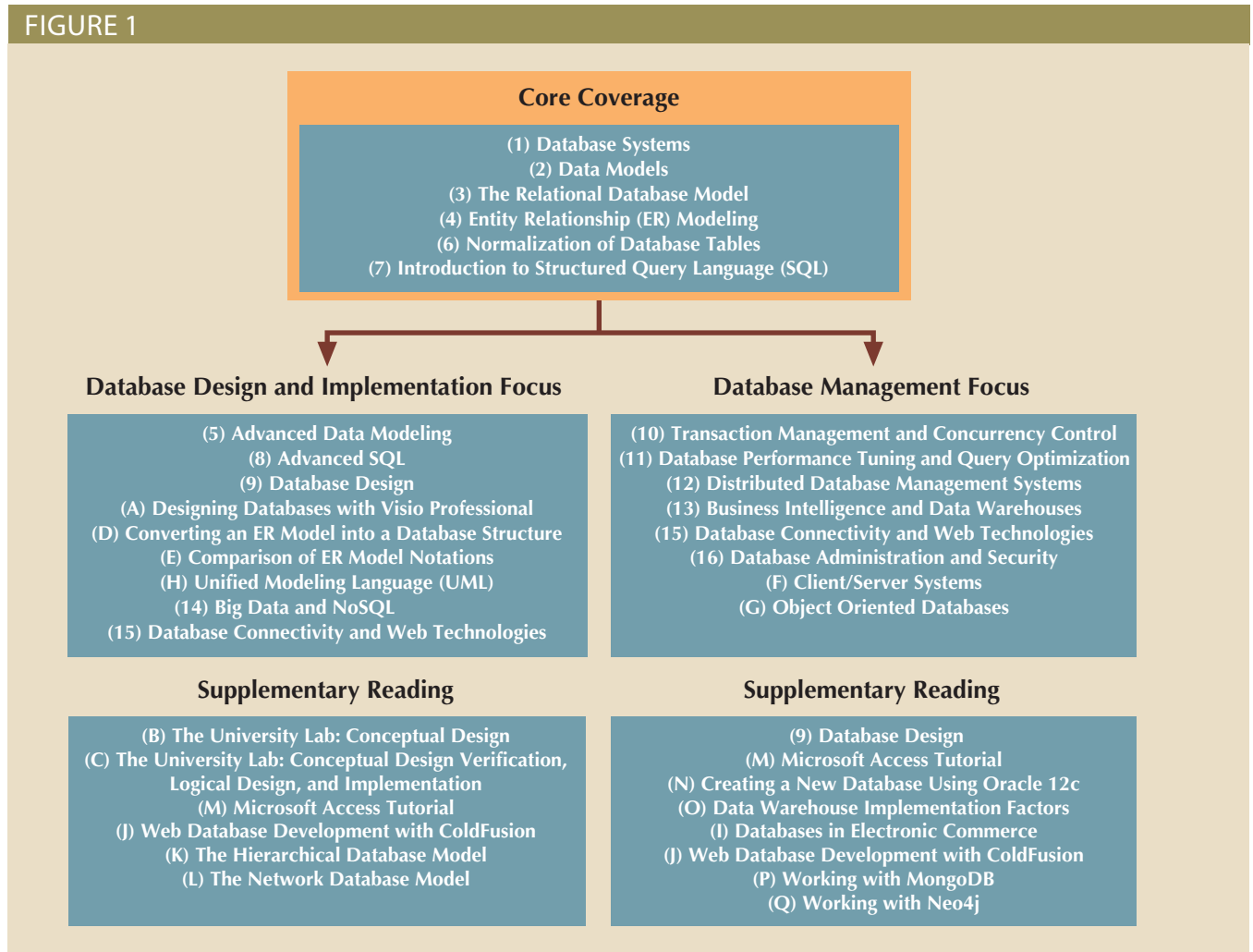
## Teaching Database: A Matter of Focus

Given the wealth of detailed coverage, instructors can “mix and match” chapters to produce the desired coverage. Depending on where database courses fit into the curriculum, instructors may choose to emphasize database design or database management. (See Figure 1.)

The hands-on nature of database design lends itself particularly well to class projects in which students use instructor-selected software to prototype a system that they design for the end user. Several end-of-chapter problems are sufficiently complex to serve as projects, or an instructor may work with local businesses to give students hands-on experience. Note that some elements of the database design track are also found in the database management track, because it is difficult to manage database technologies that are not well understood.

The options shown in Figure 1 serve only as a starting point. Naturally, instructors will tailor their coverage based on their specific course requirements. For example, an instructor may decide to make Appendix I an outside reading assignment and make Appendix A a self-taught tutorial, and then use that time to cover client/server systems or object-oriented databases. The latter choice would serve as a gateway to UML coverage.

FIGURE 1





# Text Features

**Online Content** boxes draw attention to material at [www.cengagebrain.com](http://www.cengagebrain.com) for this text and provide ideas for incorporating this content into the course.

## Online Content



All of the databases used to illustrate the material in this chapter (see the Data Files list at the beginning of the chapter) are available at [www.cengagebrain.com](http://www.cengagebrain.com). The database names match the database names shown in the figures.

**Notes** highlights important facts about the concepts introduced in the chapter.

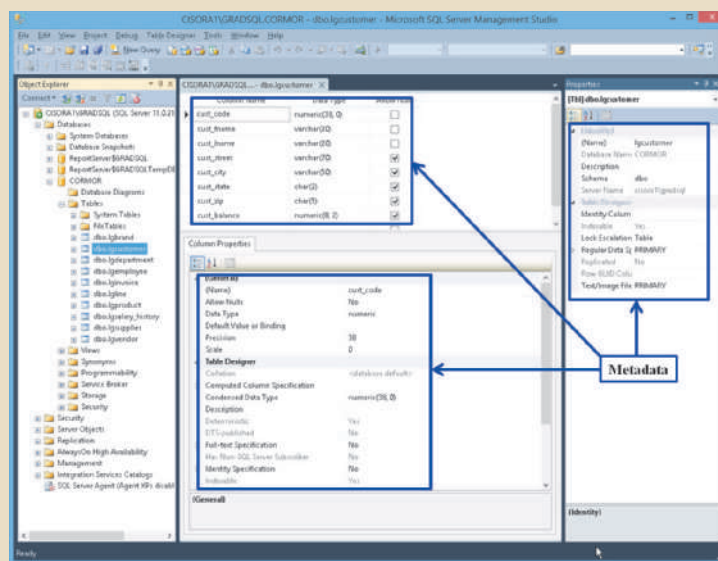


## Note

This chapter focuses on SELECT queries to retrieve data from tables. Chapter 8 will explain how those tables are actually created and how the data is loaded into them. This reflects the experience of most entry-level database positions. As a new hire working with databases, you will likely spend quite a bit of time retrieving data from tables that already exist before you begin creating new tables and modifying the data.

A variety of **four-color figures**, including ER models and implementations, tables, and illustrations, clearly illustrate difficult concepts.

FIGURE 1.12 ILLUSTRATING METADATA WITH MICROSOFT SQL SERVER EXPRESS



## Summary

- An information system is designed to help transform data into information and to manage both data and information. Thus, the database is a very important part of the information system. Systems analysis is the process that establishes the need for an information system and its extent. Systems development is the process of creating an information system

A robust **Summary** at the end of each chapter ties together the major concepts and serves as a quick review for students.

## Key Terms

bottom-up design	Database Life Cycle (DBLC)	module coupling
boundaries	database role	physical design
centralized design	decentralized design	scope
clustered tables	description of operations	systems analysis
cohesivity	differential backup	systems development
computer-aided software engineering (CASE)	full backup	Systems Development Life Cycle (SDLC)
conceptual design	information system	top-down design
database development	logical design	transaction log backup
database fragment	minimal data rule	virtualization
	module	

An alphabetic list of **Key Terms** summarizes important terms.

## Review Questions

1. What is an information system? What is its purpose?
2. How do systems analysis and systems development fit into a discussion about information systems?
3. What does the acronym SDLC mean, and what does an SDLC portray?
4. What does the acronym DBLC mean, and what does a DBLC portray?
5. Discuss the distinction between centralized and decentralized conceptual database design.

**Review Questions** challenge students to apply the skills learned in each chapter.

## Problems

In the following exercises, you will set up database connectivity using MS Excel.

1. Use MS Excel to connect to the Ch02\_InsureCo MS Access database using ODBC, and retrieve all of the AGENTS.
2. Use MS Excel to connect to the Ch02\_InsureCo MS Access database using ODBC, and retrieve all of the CUSTOMERS.

**Problems** become progressively more complex as students draw on the lessons learned from the completion of preceding problems.

# Additional Features

## MindTap® for Database Systems 13e

MindTap® combines learning tools—such as readings, multimedia, activities, and assessments—into a singular learning path that guides students through the course. You’ll find a full ebook as well as a robust set of auto-gradable homework problems. Multiple-choice homework questions developed from the end-of-chapter review questions confirm students’ understanding of core concepts and key terms. Higher-level assignments enable students to practice database design concepts in an automated environment, and chapter quizzes help prepare students for exams. Students will also benefit from the chapter-opening videos created by the authors, as well as study tools such as crossword puzzles and key-term flashcards.

MindTap® is designed to be fully integrated with any Learning Management System and can be used as a stand-alone product or in conjunction with a print textbook.

## Appendices

Eighteen online appendices provide additional material on a variety of important areas, such as using Microsoft® Visio® and Microsoft® Access®, ER model notations, UML, object-oriented databases, databases and electronic commerce, Adobe® ColdFusion®, and working with newer NoSQL databases MongoDB and Neo4j.

## Database, SQL Script, JSON Documents, and ColdFusion Files

The online materials for this book include all of the database structures and table contents used in the text. For students using Oracle®, MySQL, and Microsoft SQL Server™, SQL scripts are included to help students create and load all tables used in the SQL chapters (7 and 8). Text documents for importing JSON-formatted documents into MongoDB and a script for creating a graph database in Neo4j (Appendices P and Q) are also included. In addition, all ColdFusion scripts used to develop the web interfaces in Appendix J are included.

## Instructor Resources

*Database Systems: Design, Implementation, and Management, Thirteenth Edition*, includes teaching tools to support instructors in the classroom. The ancillary material that accompanies the textbook is listed below. They are available on the web at [www.cengagebrain.com](http://www.cengagebrain.com).

### Instructor’s Manual

The authors have created this manual to help instructors make their classes informative and interesting. Because the authors tackle so many problems in depth, instructors will find the *Instructor’s Manual* especially useful. The details of the design solution process are shown in the *Instructor’s Manual*, as well as notes about alternative approaches that may be used to solve a particular problem.

### SQL Script Files for Instructors

The authors have provided teacher’s SQL script files to allow instructors to cut and paste the SQL code into the SQL windows. (Scripts are provided for Oracle, MySQL, and MS SQL Server.) The SQL scripts, which have all been tested by Cengage Learning, are a major convenience for instructors. You won’t have to type in the SQL commands, and the use of the scripts eliminates typographical errors that are sometimes difficult to trace.

## ColdFusion Files for Instructors

The ColdFusion web development solutions are provided. Instructors have access to a menu-driven system that allows teachers to show the code as well as its execution.

## Databases

For many chapters, Microsoft® Access® instructor databases are available that include features not found in the student databases. For example, the databases that accompany Chapters 7 and 8 include many of the queries that produce the problem solutions. Other Access databases, such as the ones that accompany Chapters 3, 4, 5, and 6, include implementations of the design problem solutions to allow instructors to illustrate the effect of design decisions. In addition, instructors have access to all the script files for Oracle, MySQL, and MS SQL Server so that all the databases and their tables can be converted easily and precisely.

## Cengage Learning Testing Powered by Cognero

**A flexible, online system that allows you to:**

- Author, edit, and manage test bank content from multiple Cengage Learning solutions
- Create multiple test versions in an instant
- Deliver tests from your LMS, your classroom, or wherever you want

**Start right away!**

Cengage Learning Testing Powered by Cognero works on any operating system or browser.

- No special installs or downloads needed
- Create tests from school, home, the coffee shop—anywhere with Internet access

**What will you find?**

- Simplicity at every step. A desktop-inspired interface features drop-down menus and familiar, intuitive tools that take you through content creation and management with ease.
- Full-featured test generator. Create ideal assessments with your choice of 15 question types (including true/false, multiple-choice, opinion scale/Likert, and essay). Multi-language support, an equation editor, and unlimited metadata help ensure your tests are complete and compliant.
- Cross-compatible capability. Import and export content into other systems.

## PowerPoint® Presentations

Microsoft PowerPoint slides are included for each chapter. Instructors can use the slides in a variety of ways—for example, as teaching aids during classroom presentations or as printed handouts for classroom distribution. Instructors can modify these slides or include slides of their own for additional topics introduced to the class.

## Figure Files

Figure files for solutions are presented in the *Instructor's Manual* to allow instructors to create their own presentations. Instructors can also manipulate these files to meet their particular needs.

# Acknowledgments

Regardless of how many editions of this book are published, they will always rest on the solid foundation created by the first edition. We remain convinced that our work has become successful because that first edition was guided by Frank Ruggirello, a former Wadsworth senior editor and publisher. Aside from guiding the book's development, Frank also managed to solicit the great Peter Keen's evaluation (thankfully favorable) and subsequently convinced Peter Keen to write the foreword for the first edition. Although we sometimes found Frank to be an especially demanding taskmaster, we also found him to be a superb professional and a fine friend. We suspect Frank will still see his fingerprints all over our current work. Many thanks.

A difficult task in rewriting a book is deciding what new approaches, topical coverage, and changes to depth of coverage are appropriate for a product that has successfully weathered the test of the marketplace. The comments and suggestions made by the book's adopters, students, and reviewers play a major role in deciding what coverage is desirable and how that coverage is to be treated.

Some adopters became extraordinary reviewers, providing incredibly detailed and well-reasoned critiques even as they praised the book's coverage and style. Dr. David Hatherly, a superb database professional who is a senior lecturer in the School of Information Technology, Charles Sturt University–Mitchell, Bathhurst, Australia, made sure that we knew precisely what issues led to his critiques. Even better for us, he provided the suggestions that made it much easier for us to improve the topical coverage in earlier editions. All of his help was given freely and without prompting on our part. His efforts are much appreciated, and our thanks are heartfelt.

We also owe a debt of gratitude to Professor Emil T. Cipolla, who teaches at St. Mary College. Professor Cipolla's wealth of IBM experience turned out to be a valuable resource when we tackled the embedded SQL coverage in Chapter 8.

Every technical book receives careful scrutiny by several groups of reviewers selected by the publisher. We were fortunate to face the scrutiny of reviewers who were superbly qualified to offer their critiques, comments, and suggestions—many of which strengthened this edition. While holding them blameless for any remaining shortcomings, we owe these reviewers many thanks for their contributions:

Laurie Crawford,  
Franklin University

Mava Wilson,  
Lee University

John E. MacDonald IV,  
Binghamton University

In some respects, writing books resembles building construction: When 90 percent of the work seems done, 90 percent of the work remains to be done. Fortunately for us, we had a great team on our side.

- We are deeply indebted to Deb Kaufmann for her help and guidance. Deb has been everything we could have hoped for in a development editor and more. Deb has been our editor for almost all the editions of this book, and the quality of her work shows in the attention to detail and the cohesiveness and writing style of the material in this book.
- After writing so many books and thirteen editions of *this* book, we know just how difficult it can be to transform the authors' work into an attractive product. The production team, both at Cengage (Michele Stulga) and Lumina Datamatics (Kiruthiga Sowndarajan), have done an excellent job.

- We also owe Maria Garguilo, our Content Developer, special thanks for her ability to guide this book to a successful conclusion, and John Freitas, our technical editor, deserves many thanks for making sure all code and technical references were accurate.

We also thank our students for their comments and suggestions. They are the reason for writing this book in the first place. One comment stands out in particular: “I majored in systems for four years, and I finally discovered why when I took your course.” And one of our favorite comments by a former student was triggered by a question about the challenges created by a real-world information systems job: “Doc, it’s just like class, only easier. You really prepared me well. Thanks!”

Special thanks go to a very unique and charismatic gentleman. For over 20 years, Peter Rob has been the driving force behind the creation and evolution of this book. This book originated as a product of his drive and dedication to excellence. For over 22 years, he was the voice of *Database Systems* and the driving force behind its advancement. We wish him peace in his retirement, time with his loved ones, and luck on his many projects.

Last, and certainly not least, we thank our families for their solid support at home. They graciously accepted the fact that during more than a year’s worth of rewriting, there would be no free weekends, rare free nights, and even rarer free days. We owe you much, and the dedications we wrote are but a small reflection of the important space you occupy in our hearts.

**Carlos Coronel and Steven Morris**





# PART 1

## Database Concepts

**1** Database Systems

**2** Data Models



# Chapter 1

## Database Systems

### After completing this chapter, you will be able to:

- Define the difference between data and information
- Describe what a database is, the various types of databases, and why they are valuable assets for decision making
- Explain the importance of database design
- See how modern databases evolved from file systems
- Understand flaws in file system data management
- Outline the main components of the database system
- Describe the main functions of a database management system (DBMS)

### Preview

Organizations use data to keep track of their day-to-day operations. Such data is used to generate information, which in turn is the basis for good decisions. Data is likely to be managed most efficiently when it is stored in a database. Databases are involved in almost all facets and activities of our daily lives: from school to work, medical care, government, nonprofit organizations, and houses of worship. In this chapter, you will learn what a database is, what it does, and why it yields better results than other data management methods. You will also learn about various types of databases and why database design is so important.

Databases evolved from the need to manage large amounts of data in an organized and efficient manner. In the early days, computer file systems were used to organize such data. Although file system data management is now largely outmoded, understanding the characteristics of file systems is important because file systems are the source of serious data management limitations. In this chapter, you will also learn how the database system approach helps eliminate most of the shortcomings of file system data management.

### Data Files and Available Formats

	MS Access	Oracle	MS SQL	My SQL		MS Access	Oracle	MS SQL	My SQL
CH01_Text	✓	✓	✓	✓	CH01_Problems	✓	✓	✓	✓

Data Files Available on [cengagebrain.com](http://cengagebrain.com)

## 1-1 Why Databases?

So, why do we need databases? In today's world, data is ubiquitous (abundant, global, everywhere) and pervasive (unescapable, prevalent, persistent). From birth to death, we generate and consume data. The trail of data starts with the birth certificate and continues all the way to a death certificate (and beyond!). In between, each individual produces and consumes enormous amounts of data. As you will see in this book, databases are the best way to store and manage data. Databases make data persistent and shareable in a secure way. As you look at Figure 1.1, can you identify some of the data generated by your own daily activities?

FIGURE 1.1 THE PERVERSIVE NATURE OF DATABASES



Data is not only ubiquitous and pervasive; it is also essential for organizations to survive and prosper. Imagine trying to operate a business without knowing who your customers are, what products you are selling, who is working for you, who owes you money, and to whom you owe money. All businesses have to keep this type of data and much more. Just as important, they must have that data available to decision makers when necessary. It can be argued that the ultimate purpose of all business information systems is to help businesses use information as an organizational resource. At the heart of all of these systems are the collection, storage, aggregation, manipulation, dissemination, and management of data.

Depending on the type of information system and the characteristics of the business, this data could vary from a few megabytes on just one or two topics to terabytes covering hundreds of topics within the business's internal and external environment.

Telecommunications companies, such as Sprint and AT&T, are known to have systems that keep data on trillions of phone calls, with new data being added to the system at speeds up to 70,000 calls per second! Not only do these companies have to store and manage immense collections of data but also they have to be able to find any given fact in that data quickly. Consider the case of Internet search staple Google. While Google is reluctant to disclose many details about its data storage specifications, it is estimated that the company responds to over 91 million searches per day across a collection of data that is several terabytes in size. Impressively, the results of these searches are available almost instantly.

How can these businesses process this much data? How can they store it all, and then quickly retrieve just the facts that decision makers want to know, just when they want to know it? The answer is that they use databases. Databases, as explained in detail throughout this book, are specialized structures that allow computer-based systems to store, manage, and retrieve data very quickly. Virtually all modern business systems rely on databases. Therefore, a good understanding of how these structures are created and their proper use is vital for any information systems professional. Even if your career does not take you down the amazing path of database design and development, databases will be a key component of the systems that you use. In any case, you will probably make decisions in your career based on information generated from data. Thus, it is important that you know the difference between data and information.

## 1-2 Data versus Information

To understand what drives database design, you must understand the difference between data and information. **Data** consists of raw facts. The word *raw* indicates that the facts have not yet been processed to reveal their meaning. For example, suppose that a university tracks data on faculty members for reporting to accrediting bodies. To get the data for each faculty member into the database, you would provide a screen to allow for convenient data entry, complete with drop-down lists, combo boxes, option buttons, and other data-entry validation controls. Figure 1.2(a) shows a simple data-entry form from a software package named Sedona. When the data is entered into the form and saved, it is placed in the underlying database as raw data, as shown in Figure 1.2(b). Although you now have the facts in hand, they are not particularly useful in this format. Reading through hundreds of rows of data for faculty members does not provide much insight into the overall makeup of the faculty. Therefore, you transform the raw data into a data summary like the one shown in Figure 1.2(c). Now you can get quick answers to questions such as “What percentage of the faculty in the Information Systems (INFS) department are adjuncts?” In this case, you can quickly determine that 20 percent of the INFS faculty members are adjunct faculty. Because graphics can enhance your ability to quickly extract meaning from data, you show the data summary pie chart in Figure 1.2(d).

**Information** is the result of processing raw data to reveal its meaning. Data processing can be as simple as organizing data to reveal patterns or as complex as making forecasts or drawing inferences using statistical modeling. To reveal meaning, information requires *context*. For example, an average temperature reading of 105 degrees does not mean much unless you also know its context: Is this reading in degrees Fahrenheit or Celsius? Is this a machine temperature, a body temperature, or an outside air temperature? Information can be used as the foundation for decision making. For example, the data summary for the faculty can provide accrediting bodies with insights that are useful in determining whether to renew accreditation for the university.

Keep in mind that raw data must be properly *formatted* for storage, processing, and presentation. For example, dates might be stored in Julian calendar formats within the database, but displayed in a variety of formats, such as day-month-year or month/day/year, for different purposes. Respondents’ yes/no responses might need to be converted

### data

Raw facts, or facts that have not yet been processed to reveal their meaning to the end user.

### information

The result of processing raw data to reveal its meaning. Information consists of transformed data and facilitates decision making.

FIGURE 1.2 TRANSFORMING RAW DATA INTO INFORMATION

a) Data entry screen

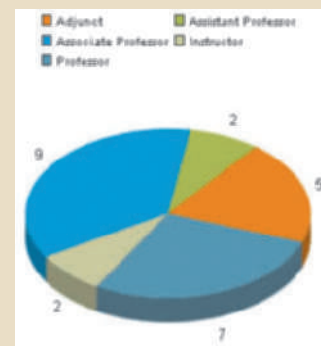
b) Raw data

ID	Last Name	Mid Name	First Name	Dept Code	Office	E-mail	Rank	Hire Year	Degree
1	Washington	A.	George	MGMT	1413	gwa@mtsu.edu	Professor	2011	Ph.D.
2	Adams	J.	John	FEA	1431	adams@mtsu.edu	Professor	1984	Ph.D.
3	Jefferson	L.	Thomas	ECOM	1431	jefferson@mtsu.edu	Instructor	2002	M.B.A.
4	Madison	D.	James	FIN	1438	madison@mtsu.edu	Associate Professor	1984	Ph.D.
5	Monroe	R.	James	ACCT	1441	monroe@mtsu.edu	Assistant Professor	1985	Ph.D.
6	Adams	O.	John	ACCT	1449	adams@mtsu.edu	Associate Professor	1979	Ph.D.
7	Johnson	C.	Andrew	ECOM	1439	johnson@mtsu.edu	Associate Professor	1993	Ph.D.
8	Van Buren	T.	Monte	FEA	1436	vanburen@mtsu.edu	Professor	1989	Ph.D.
9	Harrison	R.	William	MGMT	1418	wharrison@mtsu.edu	Professor	1984	Ph.D.
10	Tyler	M.	John	MGMT		tyler@mtsu.edu	Assistant Professor	2003	Ed.D.
11	Clark	Cheryl	MGMT	1414	clark@mtsu.edu	Associate Professor	2002	Ph.D.	
12	Taylor	G.	Zachary	ACCT	1445	taylor@mtsu.edu	Associate Professor	1986	Ph.D.
13	Fleming	M.	Mildred	JOB	1423	fleming@mtsu.edu	Professor	1992	Ph.D.
14	Price	A.	Franklin	MGMT	1429	price@mtsu.edu	Instructor	2005	M.B.A.
15	Reichman	T.	James	MGMT	1414	reichman@mtsu.edu	Associate Professor	1986	C.B.A.
17	Lynch	W.	Larry	MGMT	1410	lynch@mtsu.edu	Associate Professor	1986	Ph.D.
18	Johnson	Andrew	ECOM	1439	johnson@mtsu.edu	Professor	1992	Ph.D.	
19	Grant	Katie	MGMT	1412	grant@mtsu.edu	Assistant Professor	1989	C.B.A.	
20	Rutherford	J.	Heena	ACCT	1448	rutherford@mtsu.edu	Assistant Professor	1992	Ph.D.
21	Griffith	T.	Carlene	ACCT		griffith@mtsu.edu	Assistant Professor	2014	Ph.D.
22	Arfath	Emad	ACCT	1443	arfath@mtsu.edu	Associate Professor	2003	J.D.	
23	Deverett	O.	Robert	ACCT	1449	deverett@mtsu.edu	Associate Professor	1997	Ph.D.
24	Hernandez	X.	Patricia	B.L.A.	1448	hernandez@mtsu.edu	Associate Professor	2001	J.D.
25	McLindley	B.	Priscilla	SYG	1433	mcclindley@mtsu.edu	Adjunct	1984	M.B.
26	Rosenwell	F.	Hilary	MGMT	1419	rosenwell@mtsu.edu	Associate Professor	2002	Ph.D.
27	Wilson	Leann	BCEN	1446	wilson@mtsu.edu	Professor	1992	Ph.D.	
28	Harding	Wynne	MGMT	1414	harding@mtsu.edu	Professor	1984	Ed.D.	
29	Coastidge	Celan	ECOM	1439	coastidge@mtsu.edu	Professor	1985	Ph.D.	
30	Hopfer	Link	MGMT		hopfer@mtsu.edu	Adjunct	1983	M.B.A.	
31	Thaman	Beth	ACCT	1446	thaman@mtsu.edu	Professor	1971	Ed.D.	
32	Johnson	Robert	BCEN	1440	johnson@mtsu.edu	Professor	2001	Ph.D.	

c) Information in summary format

Rank	COUNT	%INFS	TOT/COL	%COL.TOT.	%COL.FAC.
Adjunct	5	20.00%	23	21.74%	3.27%
Assistant Professor	2	8.00%	28	7.14%	1.31%
Associate Professor	9	36.00%	37	24.32%	5.88%
Instructor	2	8.00%	18	11.11%	1.31%
Professor	7	28.00%	47	14.89%	4.58%

d) Information in graphical format



to a Y/N or 0/1 format for data storage. More complex formatting is required when working with complex data types, such as sounds, videos, or images.

In this “information age,” production of accurate, relevant, and timely information is the key to good decision making. In turn, good decision making is the key to business survival in a global market. We are now said to be entering the “knowledge age.”<sup>1</sup>

Data is the foundation of information, which is the bedrock of **knowledge**—that is, the body of information and facts about a specific subject. Knowledge implies familiarity, awareness, and understanding of information as it applies to an environment. A key characteristic of knowledge is that “new” knowledge can be derived from “old” knowledge.

Let’s summarize some key points:

- Data constitutes the building blocks of information.
- Information is produced by processing data.
- Information is used to reveal the meaning of data.
- Accurate, relevant, and timely information is the key to good decision making.
- Good decision making is the key to organizational survival in a global environment.

The previous paragraphs have explained the importance of data, and how the processing of data is used to reveal information that in turn generates “actionable” knowledge. Let’s explore a simple example of how this works in the real world.

In today’s information-centric society, we use smartphones on a daily basis. These devices have advanced GPS functionality that constantly tracks your whereabouts. This data is stored and shared with various applications. When you get a new smartphone,

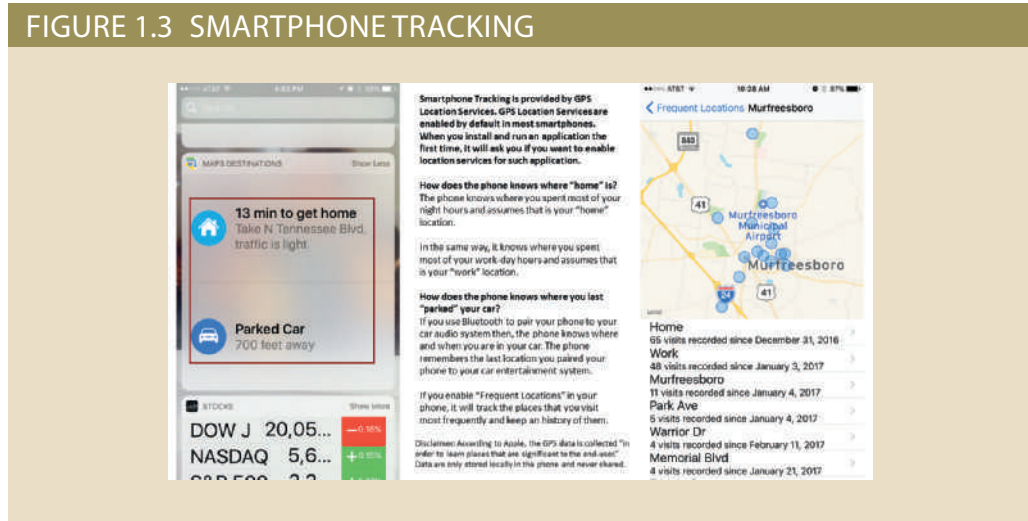
**knowledge**

The body of information and facts about a specific subject. Knowledge implies familiarity, awareness, and understanding of information as it applies to an environment. A key characteristic is that new knowledge can be derived from old knowledge.

<sup>1</sup>Peter Drucker coined the phrase “knowledge worker” in 1959 in his book *Landmarks of Tomorrow*. In 1994, Esther Dyson, George Keyworth, and Dr. Alvin Toffler introduced the concept of the “knowledge age.”

you can use the map application to go places and to set up your home address (now the phone knows where you live!). The GPS feature in your phone tracks your daily locations. In some cases, the information generated is very helpful: it can help you navigate to various locations and even to find where you parked your car. Figure 1.3 shows screenshots from one of the authors' smartphone. The phone “knows” that this is about the time he goes home and tells him how long it is going to take to get there. It also tells him where he parked his car; if he clicks the Parked Car icon, it will open a map so he can locate the car.

FIGURE 1.3 SMARTPHONE TRACKING



Furthermore, and maybe even scarier in terms of privacy issues, your smartphone may know more about your activities than you imagine. For example, suppose that every Wednesday night you go to the gym and play indoor soccer with your friends. Next Wednesday night, 20 minutes before you leave home, your phone pops up a message saying “19 minutes to [gym address]. Traffic is light.” The phone has been storing GPS data on your movements to develop patterns based on days, times, and locations to generate this knowledge. It can then associate such knowledge as your daily activities provide more data points. Imagine that on Wednesday when you go to the Magic Box gym to play soccer, when you arrive you use Facebook on your phone to check in to the gym. Now, your phone also knows the name of the place where you go every Wednesday night.

As you can see from this example, knowledge and information require timely and accurate data. Such data must be properly generated and stored in a format that is easy to access and process. In addition, like any basic resource, the data environment must be managed carefully. **Data management** is a discipline that focuses on the proper generation, storage, and retrieval of data. Given the crucial role that data plays, it should not surprise you that data management is a core activity for any business, government agency, service organization, or charity.

## 1-3 Introducing the Database

Efficient data management typically requires the use of a computer database. A **database** is a shared, integrated computer structure that stores a collection of the following:

- End-user data—that is, raw facts of interest to the end user
- **Metadata**, or data about data, through which the end-user data is integrated and managed

### data management

A process that focuses on data collection, storage, and retrieval. Common data management functions include addition, deletion, modification, and listing.

### database

A shared, integrated computer structure that houses a collection of related data. A database contains two types of data: end-user data (raw facts) and metadata.

### metadata

Data about data; that is, data about data characteristics and relationships. See also *data dictionary*.

The metadata describes the data characteristics and the set of relationships that links the data found within the database. For example, the metadata component stores information such as the name of each data element, the type of values (numeric, dates, or text) stored on each data element, and whether the data element can be left empty. The metadata provides information that complements and expands the value and use of the data. In short, metadata presents a more complete picture of the data in the database. Given the characteristics of metadata, you might hear a database described as a “collection of *self-describing* data.”

A **database management system (DBMS)** is a collection of programs that manages the database structure and controls access to the data stored in the database. In a sense, a database resembles a very well-organized electronic filing cabinet in which powerful software (the DBMS) helps manage the cabinet’s contents.

### 1-3a Role and Advantages of the DBMS

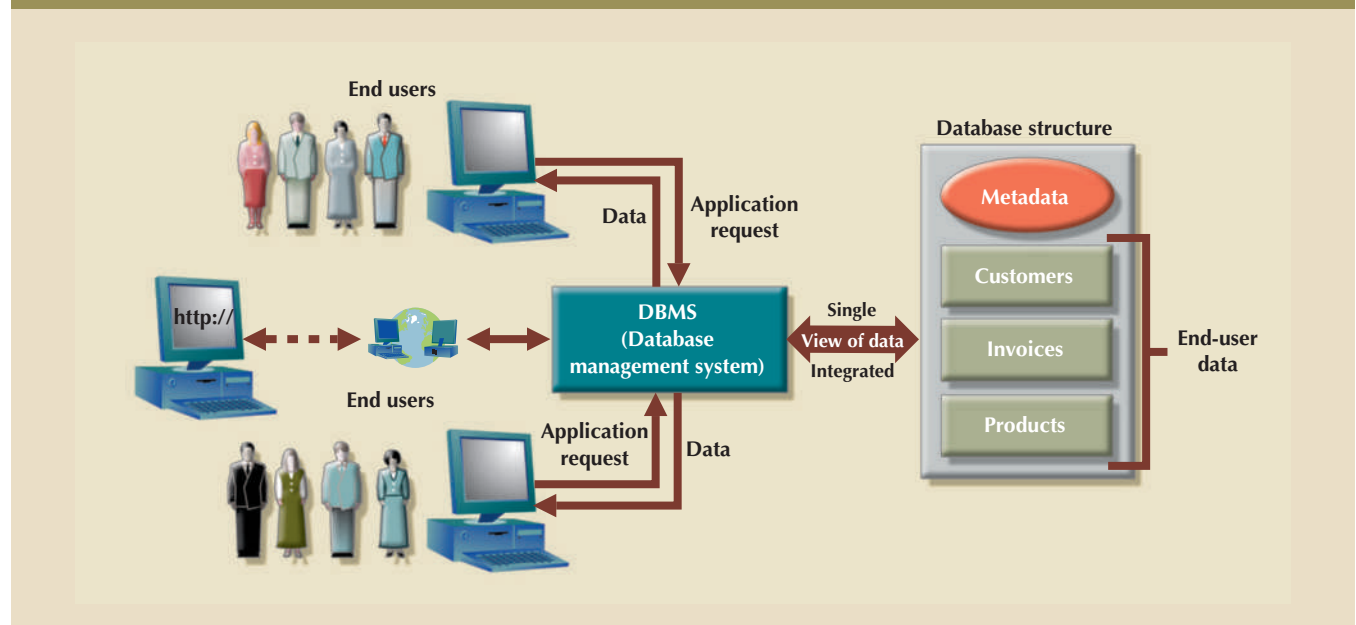
The DBMS serves as the intermediary between the user and the database. The database structure itself is stored as a collection of files, and the only way to access the data in those files is through the DBMS. Figure 1.4 emphasizes the point that the DBMS presents the end user (or application program) with a single, integrated view of the data in the database. The DBMS receives all application requests and translates them into the complex operations required to fulfill those requests. The DBMS hides much of the database’s internal complexity from the application programs and users. The application program might be written by a programmer using a programming language, such as Visual Basic.NET, Java, or C#, or it might be created through a DBMS utility program.

Having a DBMS between the end user’s applications and the database offers some important advantages. First, the DBMS enables the data in the database *to be shared* among multiple applications or users. Second, the DBMS *integrates* the many different users’ views of the data into a single all-encompassing data repository.

#### database management system (DBMS)

The collection of programs that manages the database structure and controls access to the data stored in the database.

FIGURE 1.4 THE DBMS MANAGES THE INTERACTION BETWEEN THE END USER AND THE DATABASE



Because data is the crucial raw material from which information is derived, you must have a good method to manage such data. As you will discover in this book, the DBMS helps make data management more efficient and effective. In particular, a DBMS provides these advantages:

- *Improved data sharing.* The DBMS helps create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.
- *Improved data security.* The more users access the data, the greater the risks of data security breaches. Corporations invest considerable amounts of time, effort, and money to ensure that corporate data is used properly. A DBMS provides a framework for better enforcement of data privacy and security policies.
- *Better data integration.* Wider access to well-managed data promotes an integrated view of the organization's operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the company affect other segments.
- *Minimized data inconsistency.* **Data inconsistency** exists when different versions of the same data appear in different places. For example, data inconsistency exists when a company's sales department stores a sales representative's name as Bill Brown and the company's personnel department stores that same person's name as William G. Brown, or when the company's regional sales office shows the price of a product as \$45.95 and its national sales office shows the same product's price as \$43.95. The probability of data inconsistency is greatly reduced in a properly designed database.
- *Improved data access.* The DBMS makes it possible to produce quick answers to ad hoc queries. From a database perspective, a **query** is a specific request issued to the DBMS for data manipulation—for example, to read or update the data. Simply put, a query is a question, and an **ad hoc query** is a spur-of-the-moment question. The DBMS sends back an answer (called the **query result set**) to the application. For example, when dealing with large amounts of sales data, end users might want quick answers to questions (ad hoc queries). Some examples are the following:
  - What was the dollar volume of sales by product during the past six months?
  - What is the sales bonus figure for each of our salespeople during the past three months?
  - How many of our customers have credit balances of \$3,000 or more?
- *Improved decision making.* Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based. The quality of the information generated depends on the quality of the underlying data. **Data quality** is a comprehensive approach to promoting the accuracy, validity, and timeliness of the data. While the DBMS does not guarantee data quality, it provides a framework to facilitate data quality initiatives. Data quality concepts will be covered in more detail in Chapter 16, Database Administration and Security.
- *Increased end-user productivity.* The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

#### data inconsistency

A condition in which different versions of the same data yield different (inconsistent) results.

#### query

A question or task asked by an end user of a database in the form of SQL code. A specific request for data manipulation issued by the end user or the application to the DBMS.

#### ad hoc query

A "spur-of-the-moment" question.

#### query result set

The collection of data rows returned by a query.

#### data quality

A comprehensive approach to ensuring the accuracy, validity, and timeliness of data.

The advantages of using a DBMS are not limited to the few just listed. In fact, you will discover many more advantages as you learn more about the technical details of databases and their proper design.

## 1-3b Types of Databases

A DBMS can be used to build many different types of databases. Each database stores a particular collection of data and is used for a specific purpose. Over the years, as technology and innovative uses of databases have evolved, different methods have been used to classify databases. For example, databases can be classified by the number of users supported, where the data is located, the type of data stored, the intended data usage, and the degree to which the data is structured.

The number of users determines whether the database is classified as single user or multiuser. A **single-user database** supports only one user at a time. In other words, if user A is using the database, users B and C must wait until user A is done. A single-user database that runs on a personal computer is called a **desktop database**. In contrast, a **multiuser database** supports multiple users at the same time. When the multiuser database supports a relatively small number of users (usually fewer than 50) or a specific department within an organization, it is called a **workgroup database**. When the database is used by the entire organization and supports many users (more than 50, usually hundreds) across many departments, the database is known as an **enterprise database**.

Location might also be used to classify the database. For example, a database that supports data located at a single site is called a **centralized database**. A database that supports data distributed across several different sites is called a **distributed database**. The extent to which a database can be distributed and the way in which such distribution is managed are addressed in detail in Chapter 12, Distributed Database Management Systems.

Both centralized and decentralized (distributed) databases require a well-defined infrastructure (hardware, operating systems, network technologies, etc.) to implement and operate the database. Typically, the infrastructure is owned and maintained by the organization that creates and operates the database. But in recent years, the use of cloud databases has been growing in popularity. A **cloud database** is a database that is created and maintained using cloud data services, such as Microsoft Azure or Amazon AWS. These services, provided by third-party vendors, provide defined performance measures (data storage capacity, required throughput, and availability) for the database, but do not necessarily specify the underlying infrastructure to implement it. The data owners do not have to know, or be concerned about, what hardware and software are being used to support their databases. The performance capabilities can be renegotiated with the cloud provider as the business demands on the database change. For example, 3M Health Information Systems, the world's largest provider of health care analytics software in hospitals, used Amazon's AWS cloud database services to consolidate its multiple IT centers. 3M did not have to buy, install, configure, or maintain any hardware, operating systems, or network devices. It simply purchased storage and processing capacity for its data and applications. As the demands on the databases increased, additional processing and storage capabilities could be purchased as needed. As a result, server provisioning processes that previously took 10 weeks to complete could be done in mere minutes. This allows the company to be more responsive to the needs of customers and innovate faster.

In some contexts, such as research environments, a popular way of classifying databases is according to the type of data stored in them. Using this criterion, databases are grouped into two categories: general-purpose and discipline-specific databases. **General-purpose databases** contain a wide variety of data used in multiple disciplines—for example, a census database that contains general demographic data and the LexisNexis and ProQuest databases that contain newspaper, magazine, and journal articles for a variety of topics. **Discipline-specific databases** contain data focused on specific subject areas. The data in this type of database is used mainly for academic or research purposes

### single-user database

A database that supports only one user at a time.

### desktop database

A single-user database that runs on a personal computer.

### multiuser database

A database that supports multiple concurrent users.

### workgroup database

A multiuser database that usually supports fewer than 50 users or is used for a specific department in an organization.

### enterprise database

The overall company data representation, which provides support for present and expected future needs.

### centralized database

A database located at a single site.

### distributed database

A logically related database that is stored in two or more physically independent sites.

### cloud database

A database that is created and maintained using cloud services, such as Microsoft Azure or Amazon AWS.

### general-purpose database

A database that contains a wide variety of data used in multiple disciplines.

### discipline-specific database

A database that contains data focused on specific subject areas.