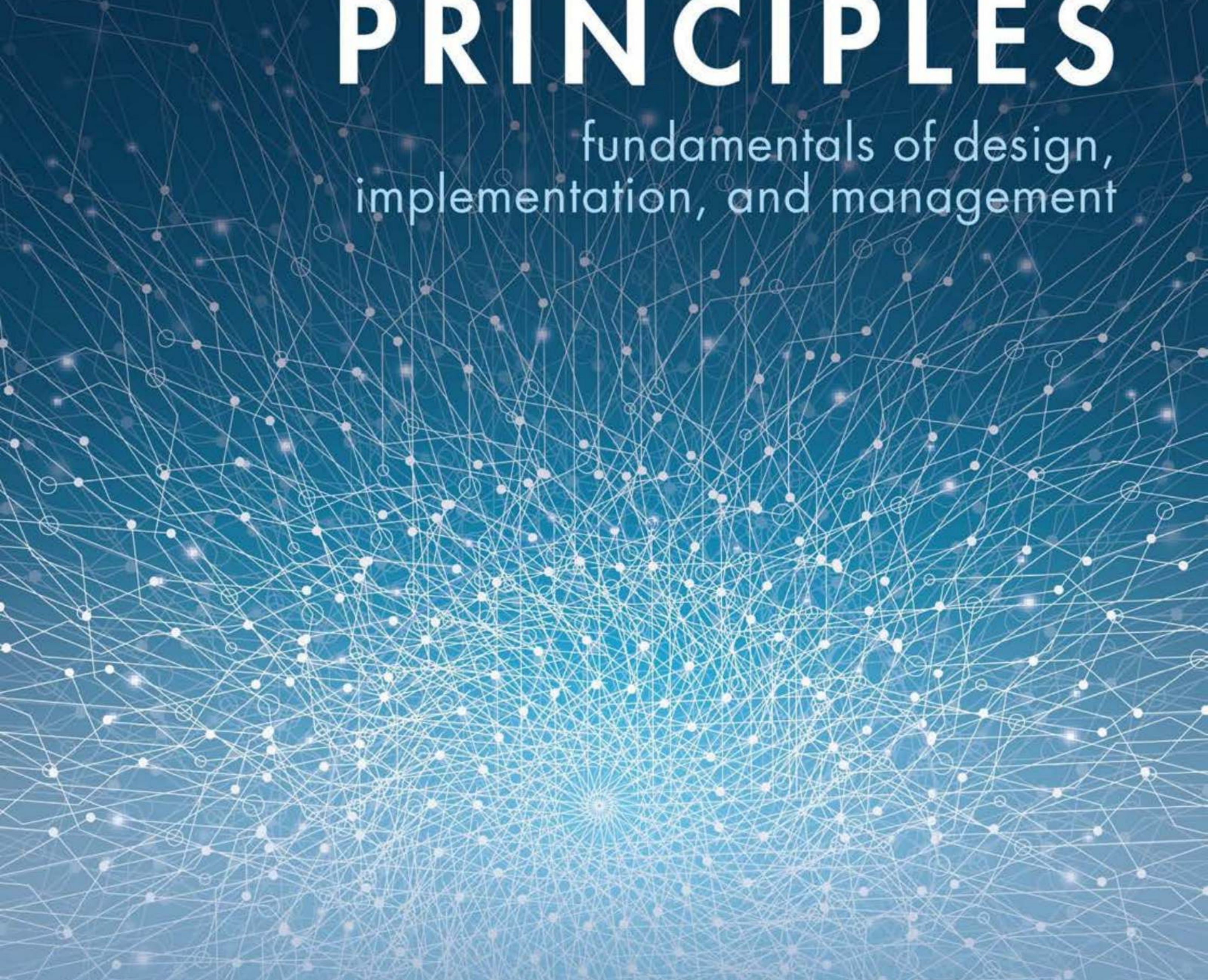
CENGAGE

THIRD EDITION

DATABASE PRINCIPLES

fundamentals of design, implementation, and management

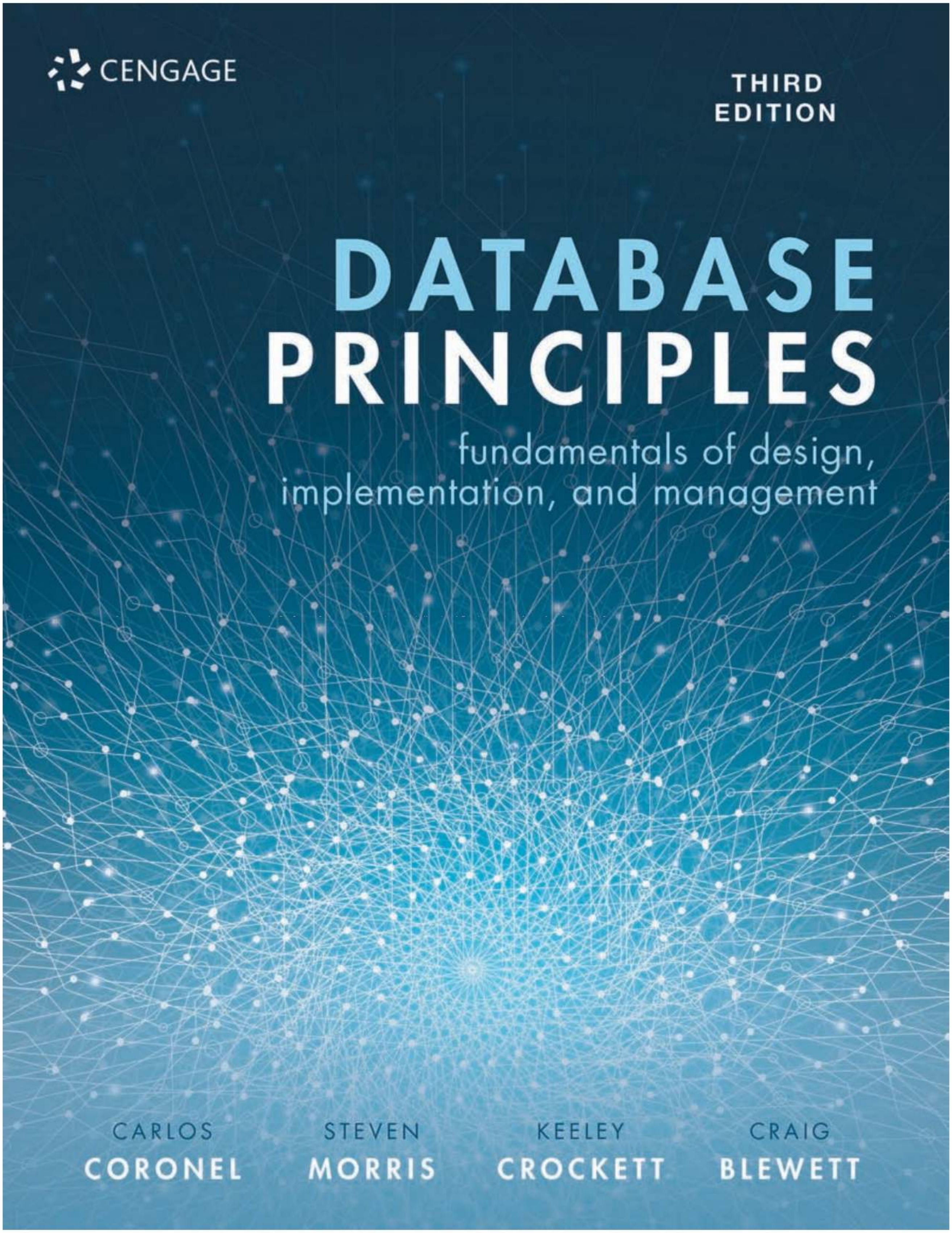


CARLOS CORONEL

STEVEN MORRIS

KEELEY CROCKETT

CRAIG BLEWETT



CENGAGE

THIRD EDITION

DATABASE PRINCIPLES

fundamentals of design, implementation, and management

CARLOS CORONEL STEVEN MORRIS KEELEY CROCKETT CRAIG BLEWETT

CENGAGE

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

CENGAGE

Database Principles: Fundamentals of Design, Implementation, and Management
Third Edition
US Authors: Carlos Coronel, Steven Morris
Adapters: Keeley Crockett, Craig Blewett

Publisher: Marinda Louw

Marketing Manager: Anna Reading

Senior Content Project Manager: Sue Povey

Manufacturing Manager: Eyvett Davis

Typesetter: SPi-Global

Cover Designer: Simon Levy Associates

Cover Image(s): ©Vijay Kumar/Getty Images

For product information and technology assistance, contact us at **emea.info@cengage.com**

For permission to use material from this text or product and for permission queries, email **emea.permissions@cengage.com**

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

ISBN: 978-1-4737-6804-8

Cengage Learning, EMEA
Cheriton House, North Way
Andover, Hampshire, SP10 5BE
United Kingdom

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.co.uk.**

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

To learn more about Cengage platforms and services, register or access your online learning solution, or purchase materials for your course, visit **www.cengage.com.**

Printed in China at RR Donnelley
Print Number: 01 Print Year: 2020

BRIEF CONTENTS

CONTENTS

3 Relational Model Characteristics 70

4 Relational Algebra and Calculus 119

Part II Design Concepts 162

5 Data Modelling with Entity Relationship Diagrams 165

6 Data Modelling Advanced Concepts 233

Appendices (Available online)

PREFACE

We are excited to introduce the third edition of *Database Principles*, which is designed 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 things, their successful creation depends on understanding the important concepts that define them.

This edition is suitable for a first course in databases at undergraduate level and will also provide essential material for conversion postgraduate courses. Providing comprehensive and practical coverage of core database concepts, it is an ideal text not only for those studying database management systems in the context of computer science, but also those on courses in the areas of business technology, introductory data science and data analytics.

The Approach: Continued Emphasis on the Stages of Design

As the title suggests, *Database Principles: 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 even database-inexperienced people to create databases and database applications. Unfortunately, the 'create without design' approach usually paves the way to any number of database disasters. In our experience, many, if not most, 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. Using an analogy, even the best bricklayers and carpenters can't create a good building from a bad blueprint.
- Most difficult problems associated with database system management seem to be triggered by poorly designed databases. It hardly seems worthwhile to use scarce resources to develop excellent and extensive database system management skills in order to exercise 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 organisations really function once a good database design is completed.
- Familiarity with database design techniques promotes one's 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 are sufficiently challenging for students to 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 are capable of meeting end-user requirements while conforming to high design standards.

This edition retains the use of UML (Unified Modelling Language) notation for data modelling. Continual development by the Object Management Group has led to UML becoming an International Standard (UML 2.5.1 is available as the 2017 edition standard: ISO/IEC 19505-1 and 19505-2), which is continually reviewed. In keeping with the second edition, UML has continued to be used to produce entity relationship models within this third edition. However, as organisations still use both Chen and Crow's Foot notation approaches to data modelling in order to maintain legacy systems, it is important that familiarity is maintained. Appendix E, Comparison of ER Modelling Notations, contains coverage of both these notations.

CHANGES TO THE THIRD EDITION

In this third edition, we have added some new features and continued to strengthen the already strong database design coverage. Here are just a few of the highlights:

- To support the growth of Big Data and NoSQL technology, we have added a new Chapter 16: Big Data and NoSQL. The chapter focuses in greater depth on the characteristics of Big Data and the technologies that have been developed to support its use, including Hadoop and MongoDB.
- New and expanded coverage of data visualisation tools and techniques in Chapter 15, Databases for Business Intelligence.
- New and updated Business Vignettes to provide topical discussion points in the classroom.
- Coverage of MongoDB with hands-on exercises for querying MongoDB databases (Appendix Q).
- An additional appendix containing coverage of Neo4j with hands-on exercises for querying graph databases (Appendix R).

ACKNOWLEDGEMENTS

The publisher acknowledges the contribution of the following lecturers, who provided invaluable feedback on the second and third editions:

- Emilia Mwim, UNISA
- Patricia Alexander, University of Pretoria
- Judy van Biljon, UNISA
- Casper Wessels, Central University of Technology
- Theo Macdonald, University of the Free State
- Ismael Essop, University of Greenwich
- Chris Jakeman, Peterborough Regional College
- Andy Davies, Blackburn College
- Mick Ridley, University of Bradford
- Ray Turner, University of Essex
- Mark Green, Oxford Brookes University
- Duncan McPhee, University of Glamorgan

For this edition, I would like to say a special thanks to Pamela Quick, who previously worked as a Senior Lecturer in the School of Computing, Maths and Digital Technology at Manchester Metropolitan University. Her years of experience within the database field have been very valuable, specifically the coverage of relational algebra.

On this third edition, I have been lucky to work with a very patient, supportive and professional Publisher, Marinda Louw. Marinda provided fantastic support in answering all my emails. It has been a pleasure working with you.

Last, and certainly not least, thank you to my family (my ohana) for your patience and support.

Keeley Crockett
January 2020

ABOUT THE AUTHORS

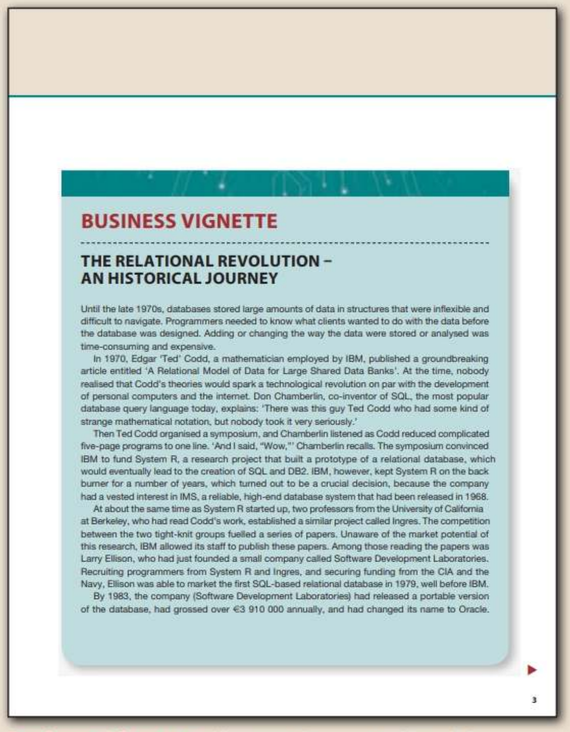
Carlos Coronel is currently the Lab Director for the College of Business Computer Labs at Middle Tennessee State University. He has over 25 years of experience in various fields as a Database Administrator, Network Administrator, Web Manager and Technology Specialist, and has taught courses in Web development, database design and development, and data communications at the undergraduate and graduate levels.

Steven Morris completed his Bachelor of Science and PhD from Auburn University. He has taught Database Design and Development, Database Programming with Advanced SQL and PL/SQL, Systems Analysis and Design, and Principles of MIS at Middle Tennessee State University. Steven has published many articles, and currently serves on the review boards of several journals.

Dr Keeley Crockett is a Reader in Computational Intelligence in the School of Computing, Mathematics and Digital Technology at Manchester Metropolitan University. She gained a BSc Degree (Hons) in Computation from UMIST in 1993, and a PhD in the field of machine learning in 1998 entitled 'Fuzzy Rule Induction from Data Domains'. She has been teaching within the field of database systems and data engineering for 20 years to both undergraduate and postgraduate students. She leads the Computational Intelligence Research Lab, which has established a strong international presence for its research into Adaptive Psychological Profiling using artificial intelligence, fuzzy systems, and natural language dialogue systems. She has published over 125 refereed conference papers and journal articles in major international conferences and journals. She is an active volunteer in the IEEE undertaking many roles such as being a member of the IEEE Women in Engineering Leadership committee, and IEEE Women in Computational intelligence subcommittee among many other roles. Keeley is also proud to be a STEM Ambassador with a passion for outreach in computer science in rural schools.

Dr Craig Blewett has been researching and teaching in the area of Information Systems and Technology in South Africa for over 25 years. His Masters explored the application of Artificial Intelligence to database transaction management. His PhD, in education technology, resulted in the development of the Activated Classroom Teaching (ACT) model, a unique approach to teaching with technology. Craig is the founder of multiple technology companies and is the author of numerous books covering topics such as computer literacy, database systems, teaching with technology, running, and active living. He is also an internationally acclaimed speaker who is using his innovative approaches to help change education in our rapidly changing digital world.

WALK-THROUGH TOUR



BUSINESS VIGNETTE

THE RELATIONAL REVOLUTION – AN HISTORICAL JOURNEY

Until the late 1970s, databases stored large amounts of data in structures that were inflexible and difficult to navigate. Programmers needed to know what clients wanted to do with the data before the database was designed. Adding or changing the way the data were stored or analysed was time-consuming and expensive.

In 1970, Edgar 'Ted' Codd, a mathematician employed by IBM, published a groundbreaking article entitled 'A Relational Model of Data for Large Shared Data Banks'. At the time, nobody realised that Codd's theories would spark a technological revolution on par with the development of personal computers and the internet. Don Chamberlin, co-inventor of SQL, the most popular database query language today, explains: 'There was this guy Ted Codd who had some kind of strange mathematical notation, but nobody took it very seriously.'

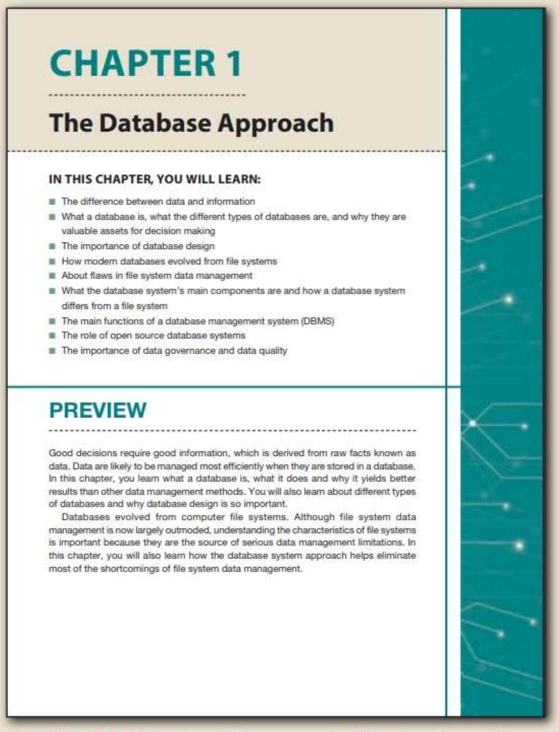
Then Ted Codd organised a symposium, and Chamberlin listened as Codd reduced complicated five-page programs to one line. 'And I said, "Wow,"' Chamberlin recalls. The symposium convinced IBM to fund System R, a research project that built a prototype of a relational database, which would eventually lead to the creation of SQL and DB2. IBM, however, kept System R on the back burner for a number of years, which turned out to be a crucial decision, because the company had a vested interest in IMS, a reliable, high-end database system that had been released in 1968.

At about the same time as System R started up, two professors from the University of California at Berkeley, who had read Codd's work, established a similar project called Ingres. The competition between the two tight-knit groups fuelled a series of papers. Unaware of the market potential of this research, IBM allowed its staff to publish these papers. Among those reading the papers was Larry Ellison, who had just founded a small company called Software Development Laboratories. Recruiting programmers from System R and Ingres, and securing funding from the CIA and the Navy, Ellison was able to market the first SQL-based relational database in 1979, well before IBM.

By 1983, the company (Software Development Laboratories) had released a portable version of the database, had grossed over €3 910 000 annually, and had changed its name to Oracle.

3

Business Vignettes illustrate the part topics with a genuine scenario and show how the subject integrates with the real world.



CHAPTER 1

The Database Approach

IN THIS CHAPTER, YOU WILL LEARN:

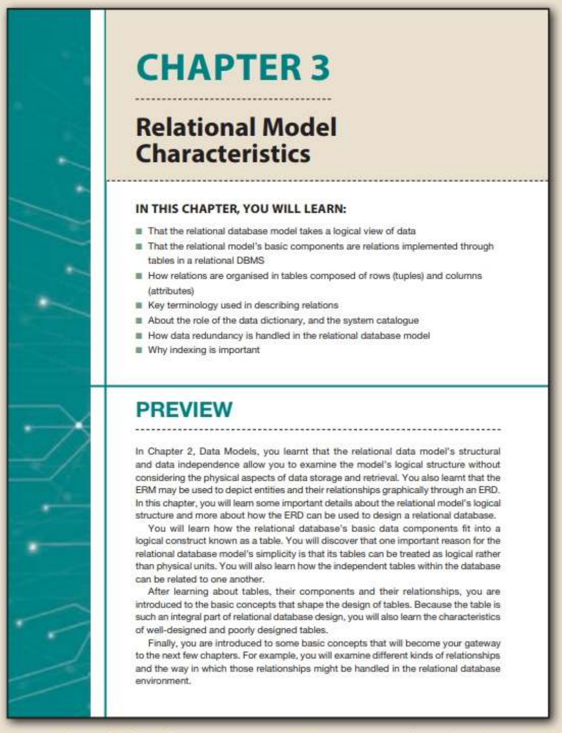
- The difference between data and information
- What a database is, what the different types of databases are, and why they are valuable assets for decision making
- The importance of database design
- How modern databases evolved from file systems
- About flaws in file system data management
- What the database system's main components are and how a database system differs from a file system
- The main functions of a database management system (DBMS)
- The role of open source database systems
- The importance of data governance and data quality

PREVIEW

Good decisions require good information, which is derived from raw facts known as data. Data are likely to be managed most efficiently when they are stored in a database. In this chapter, you learn what a database is, what it does and why it yields better results than other data management methods. You will also learn about different types of databases and why database design is so important.

Databases evolved from computer file systems. Although file system data management is now largely outmoded, understanding the characteristics of file systems is important because they 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.

Chapter Previews set the scene for the chapter and provide an overview of the chapter's contents.



CHAPTER 3

Relational Model Characteristics

IN THIS CHAPTER, YOU WILL LEARN:

- That the relational database model takes a logical view of data
- That the relational model's basic components are relations implemented through tables in a relational DBMS
- How relations are organised in tables composed of rows (tuples) and columns (attributes)
- Key terminology used in describing relations
- About the role of the data dictionary, and the system catalogue
- How data redundancy is handled in the relational database model
- Why indexing is important

PREVIEW

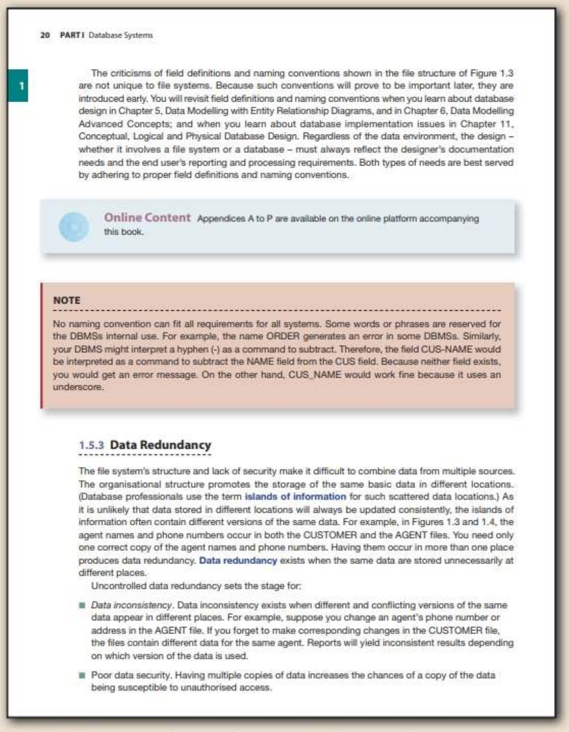
In Chapter 2, Data Models, you learnt that the relational data model's structural and data independence allow you to examine the model's logical structure without considering the physical aspects of data storage and retrieval. You also learnt that the ERM may be used to depict entities and their relationships graphically through an ERD. In this chapter, you will learn some important details about the relational model's logical structure and more about how the ERD can be used to design a relational database.

You will learn how the relational database's basic data components fit into a logical construct known as a table. You will discover that one important reason for the relational database model's simplicity is that its tables can be treated as logical rather than physical units. You will also learn how the independent tables within the database can be related to one another.

After learning about tables, their components and their relationships, you are introduced to the basic concepts that shape the design of tables. Because the table is such an integral part of relational database design, you will also learn the characteristics of well-designed and poorly designed tables.

Finally, you are introduced to some basic concepts that will become your gateway to the next few chapters. For example, you will examine different kinds of relationships and the way in which those relationships might be handled in the relational database environment.

Learning Objectives appear at the start of each chapter to help you monitor your understanding and progress through each chapter. Each chapter also ends with a summary section that recaps the key content for revision purposes.



20 PART I Database Systems

1

The criticisms of field definitions and naming conventions shown in the file structure of Figure 1.3 are not unique to file systems. Because such conventions will prove to be important later, they are introduced early. You will revisit field definitions and naming conventions when you learn about database design in Chapter 5, Data Modelling with Entity Relationship Diagrams, and in Chapter 6, Data Modelling Advanced Concepts; and when you learn about database implementation issues in Chapter 11, Conceptual, Logical and Physical Database Design. Regardless of the data environment, the design – whether it involves a file system or a database – must always reflect the designer's documentation needs and the end user's reporting and processing requirements. Both types of needs are best served by adhering to proper field definitions and naming conventions.

Online Content Appendices A to P are available on the online platform accompanying this book.

NOTE

No naming convention can fit all requirements for all systems. Some words or phrases are reserved for the DBMSs internal use. For example, the name ORDER generates an error in some DBMSs. Similarly, your DBMS might interpret a hyphen (-) as a command to subtract. Therefore, the field CUS-NAME would be interpreted as a command to subtract the NAME field from the CUS field. Because neither field exists, you would get an error message. On the other hand, CUS_NAME would work fine because it uses an underscore.

1.5.3 Data Redundancy

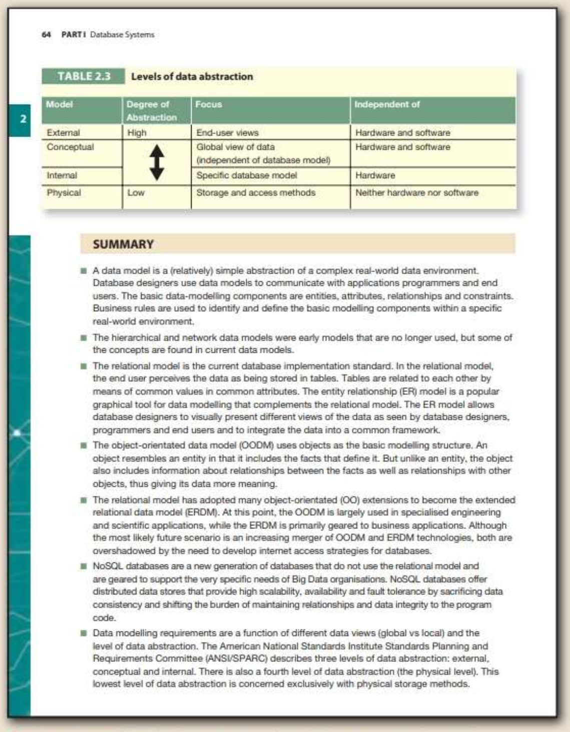
The file system's structure and lack of security make it difficult to combine data from multiple sources. The organisational structure promotes the storage of the same basic data in different locations. (Database professionals use the term **islands of information** for such scattered data locations.) As it is unlikely that data stored in different locations will always be updated consistently, the islands of information often contain different versions of the same data. For example, in Figures 1.3 and 1.4, the agent names and phone numbers occur in both the CUSTOMER and the AGENT files. You need only one correct copy of the agent names and phone numbers. Having them occur in more than one place produces data redundancy. **Data redundancy** exists when the same data are stored unnecessarily at different places.

Uncontrolled data redundancy sets the stage for:

- *Data inconsistency*. Data inconsistency exists when different and conflicting versions of the same data appear in different places. For example, suppose you change an agent's phone number or address in the AGENT file. If you forget to make corresponding changes in the CUSTOMER file, the files contain different data for the same agent. Reports will yield inconsistent results depending on which version of the data is used.
- Poor data security. Having multiple copies of data increases the chances of a copy of the data being susceptible to unauthorised access.

Online Content boxes draw attention to relevant material on the online platform for this book.

Notes highlight important facts about the concepts introduced in the chapter.



64 PART I Database Systems

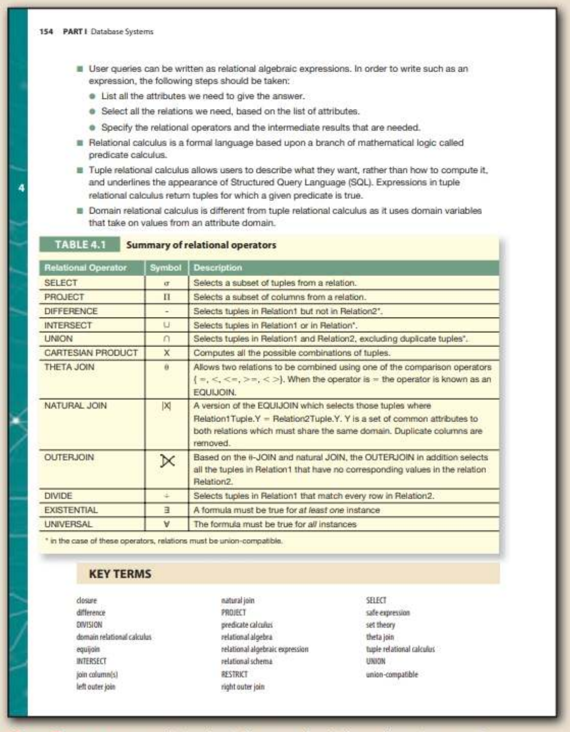
TABLE 2.3 Levels of data abstraction

Model	Degree of Abstraction	Focus	Independent of
External	High	End-user views	Hardware and software
Conceptual	↕	Global view of data (independent of database model)	Hardware and software
Internal		Specific database model	Hardware
Physical	Low	Storage and access methods	Neither hardware nor software

SUMMARY

- A data model is a (relatively) simple abstraction of a complex real-world data environment. Database designers use data models to communicate with applications programmers and end users. The basic data-modelling components are entities, attributes, relationships and constraints. Business rules are used to identify and define the basic modelling components within a specific real-world environment.
- The hierarchical and network data models were early models that are no longer used, but some of the concepts are found in current data models.
- The relational model is the current database implementation standard. In the relational model, the end user perceives the data as being stored in tables. Tables are related to each other by means of common values in common attributes. The entity relationship (ER) model is a popular graphical tool for data modelling that complements the relational model. The ER model allows database designers to visually present different views of the data as seen by database designers, programmers and end users and to integrate the data into a common framework.
- The object-orientated data model (OODM) uses objects as the basic modelling structure. An object resembles an entity in that it includes the facts that define it. But unlike an entity, the object also includes information about relationships between the facts as well as relationships with other objects, thus giving its data more meaning.
- The relational model has adopted many object-orientated (OO) extensions to become the extended relational data model (ERDM). At this point, the OODM is largely used in specialised engineering and scientific applications, while the ERDM is primarily geared to business applications. Although the most likely future scenario is an increasing merger of OODM and ERDM technologies, both are overshadowed by the need to develop internet access strategies for databases.
- NoSQL databases are a new generation of databases that do not use the relational model and are geared to support the very specific needs of Big Data organisations. NoSQL databases offer distributed data stores that provide high scalability, availability and fault tolerance by sacrificing data consistency and shifting the burden of maintaining relationships and data integrity to the program code.
- Data modelling requirements are a function of different data views (global vs local) and the level of data abstraction. The American National Standards Institute Standards Planning and Requirements Committee (ANSI/SPARC) describes three levels of data abstraction: external, conceptual and internal. There is also a fourth level of data abstraction (the physical level). This lowest level of data abstraction is concerned exclusively with physical storage methods.

Summary Each chapter ends with a comprehensive summary that provides a thorough recap of the issues in each chapter, helping you to assess your understanding and revise key content.



154 PART I Database Systems

- User queries can be written as relational algebraic expressions. In order to write such as an expression, the following steps should be taken:
  - List all the attributes we need to give the answer.
  - Select all the relations we need, based on the list of attributes.
  - Specify the relational operators and the intermediate results that are needed.
- Relational calculus is a formal language based upon a branch of mathematical logic called predicate calculus.
- Tuple relational calculus allows users to describe what they want, rather than how to compute it, and underlines the appearance of Structured Query Language (SQL). Expressions in tuple relational calculus return tuples for which a given predicate is true.
- Domain relational calculus is different from tuple relational calculus as it uses domain variables that take on values from an attribute domain.

TABLE 4.1 Summary of relational operators

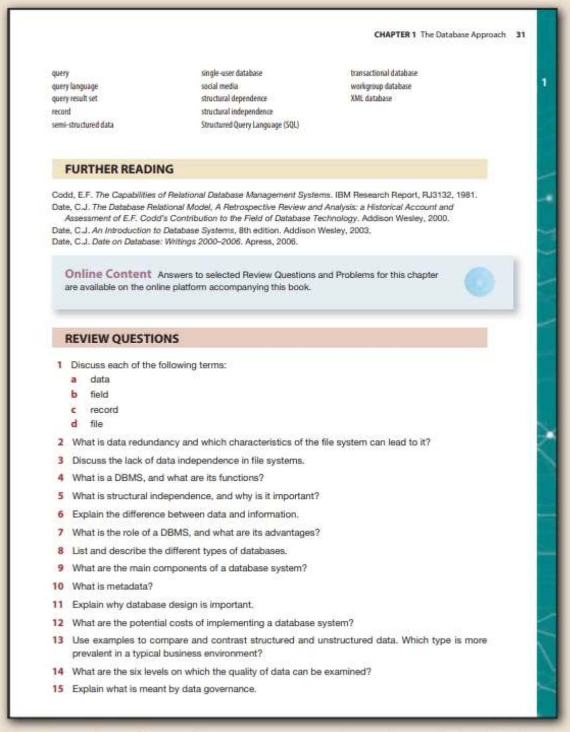
Relational Operator	Symbol	Description
SELECT	σ	Selects a subset of tuples from a relation.
PROJECT	Π	Selects a subset of columns from a relation.
DIFFERENCE	-	Selects tuples in Relation1 but not in Relation2*.
INTERSECT	∪	Selects tuples in Relation1 or in Relation*.
UNION	∩	Selects tuples in Relation1 and Relation2, excluding duplicate tuples*.
CARTESIAN PRODUCT	X	Computes all the possible combinations of tuples.
THETA JOIN	θ	Allows two relations to be combined using one of the comparison operators {=, <, <=, >=, < >}. When the operator is = the operator is known as an EQUIJOIN.
NATURAL JOIN	|X|	A version of the EQUIJOIN which selects those tuples where Relation1Tuple.Y = Relation2Tuple.Y. Y is a set of common attributes to both relations which must share the same domain. Duplicate columns are removed.
OUTERJOIN	⟗	Based on the θ-JOIN and natural JOIN, the OUTERJOIN in addition selects all the tuples in Relation1 that have no corresponding values in the relation Relation2.
DIVIDE	÷	Selects tuples in Relation1 that match every row in Relation2.
EXISTENTIAL	∃	A formula must be true for *at least one* instance
UNIVERSAL	∀	The formula must be true for *all* instances

* in the case of these operators, relations must be union-compatible.

KEY TERMS

closure
difference
DIVISION
domain relational calculus
equijoin
INTERSECT
join column(s)
left outer join
natural join
PROJECT
predicate calculus
relational algebra
relational algebraic expression
relational schema
RESTRICT
right outer join
SELECT
safe expression
set theory
theta join
tuple relational calculus
UNION
union-compatible

Key Terms are listed at the end of the chapter and explained in full in a Glossary at the end of the book, enabling you to find explanations of key terms quickly.



CHAPTER 1 The Database Approach 31

query
query language
query result set
record
semi-structured data
single-user database
social media
structural dependence
structural independence
Structured Query Language (SQL)
transactional database
workgroup database
XML database

FURTHER READING

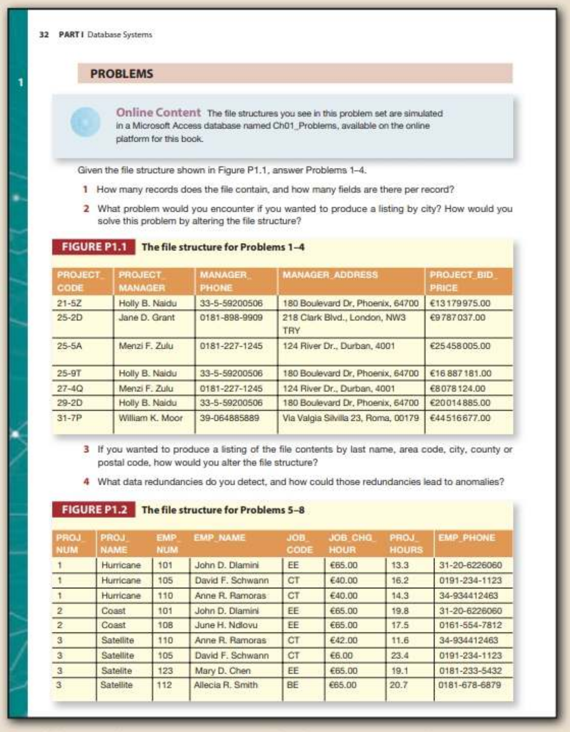
Codd, E.F. *The Capabilities of Relational Database Management Systems.* IBM Research Report, RJ3132, 1981.
Date, C.J. *The Database Relational Model, A Retrospective Review and Analysis: a Historical Account and Assessment of E.F. Codd's Contribution to the Field of Database Technology.* Addison Wesley, 2000.
Date, C.J. *An Introduction to Database Systems*, 8th edition. Addison Wesley, 2003.
Date, C.J. *Date on Database: Writings 2000–2006.* Apress, 2006.

Online Content Answers to selected Review Questions and Problems for this chapter are available on the online platform accompanying this book.

REVIEW QUESTIONS

1 Discuss each of the following terms:
   a data
   b field
   c record
   d file
2 What is data redundancy and which characteristics of the file system can lead to it?
3 Discuss the lack of data independence in file systems.
4 What is a DBMS, and what are its functions?
5 What is structural independence, and why is it important?
6 Explain the difference between data and information.
7 What is the role of a DBMS, and what are its advantages?
8 List and describe the different types of databases.
9 What are the main components of a database system?
10 What is metadata?
11 Explain why database design is important.
12 What are the potential costs of implementing a database system?
13 Use examples to compare and contrast structured and unstructured data. Which type is more prevalent in a typical business environment?
14 What are the six levels on which the quality of data can be examined?
15 Explain what is meant by data governance.

Further Reading allows you to explore the subject further, and acts as a starting point for projects and assignments. **Review Questions** help reinforce and test your knowledge and understanding, and provide a basis for group discussions and activities.



32 PART I Database Systems

PROBLEMS

Online Content The file structures you see in this problem set are simulated in a Microsoft Access database named Ch01_Problems, available on the online platform for this book.

Given the file structure shown in Figure P1.1, answer Problems 1–4.

1 How many records does the file contain, and how many fields are there per record?

2 What problem would you encounter if you wanted to produce a listing by city? How would you solve this problem by altering the file structure?

FIGURE P1.1 The file structure for Problems 1–4

PROJECT_CODE	PROJECT_MANAGER	MANAGER_PHONE	MANAGER_ADDRESS	PROJECT_BID_PRICE
21-5Z	Holly B. Naidu	33-5-59200506	180 Boulevard Dr, Phoenix, 64700	€13 179 975.00
25-2D	Jane D. Grant	0181-898-9909	218 Clark Blvd., London, NW3 TRY	€9 787 037.00
25-5A	Menzi F. Zulu	0181-227-1245	124 River Dr., Durban, 4001	€25 458 005.00
25-9T	Holly B. Naidu	33-5-59200506	180 Boulevard Dr, Phoenix, 64700	€16 887 181.00
27-4Q	Menzi F. Zulu	0181-227-1245	124 River Dr., Durban, 4001	€8 078 124.00
29-2D	Holly B. Naidu	33-5-59200506	180 Boulevard Dr, Phoenix, 64700	€20 014 885.00
31-7P	William K. Moor	39-064885889	Via Valgia Silvilla 23, Roma, 00179	€44 516 677.00

3 If you wanted to produce a listing of the file contents by last name, area code, city, county or postal code, how would you alter the file structure?

4 What data redundancies do you detect, and how could those redundancies lead to anomalies?

FIGURE P1.2 The file structure for Problems 5–8

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CODE	JOB_CHG_HOUR	PROJ_HOURS	EMP_PHONE
1	Hurricane	101	John D. Dlamini	EE	€65.00	13.3	31-20-6226060
1	Hurricane	105	David F. Schwann	CT	€40.00	16.2	0191-234-1123
1	Hurricane	110	Anne R. Ramoras	CT	€40.00	14.3	34-934412463
2	Coast	101	John D. Dlamini	EE	€65.00	19.8	31-20-6226060
2	Coast	108	June H. Ndlovu	EE	€65.00	17.5	0161-554-7812
3	Satellite	110	Anne R. Ramoras	CT	€42.00	11.6	34-934412463
3	Satellite	105	David F. Schwann	CT	€6.00	23.4	0191-234-1123
3	Satelite	123	Mary D. Chen	EE	€65.00	19.1	0181-233-5432
3	Satellite	112	Allecia R. Smith	BE	€65.00	20.7	0181-678-6879

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

DEDICATION

To my son, Kona, of whom I am so proud – keep following your dreams.

To Craig, my best friend and patient husband. Thank you for supporting my crazy busy life – without you nothing would be possible. In memory of my father, Frank Crockett, who inspired me to be the person I am today. To my mother, Norma Crockett, who is the angel in my life. Thank you for always being there for me.

To my mother- and father-in-law Jackie and Bill Smith who have provided me with much love and support.

In memory of Leslie Crockett, a true gentleman and much-loved uncle.

To my family and friends, all of whom have painted rainbows in my life.

Much love and aloha to you all.

Keeley Crockett

CENGAGE

Teaching & Learning Support Resources

Cengage's peer-reviewed content for higher and further education courses is accompanied by a range of digital teaching and learning support resources. The resources are carefully tailored to the specific needs of the instructor, student and the course. Examples of the kind of resources provided include:

- A password-protected area for instructors with, for example, a test bank, PowerPoint slides and an instructor's manual.
- An open-access area for students including, for example, online appendices, useful weblinks and glossary terms.

Lecturers: to discover the dedicated teaching digital support resources accompanying this textbook please register here for access: **cengage.com/dashboard/#login**

Students: to discover the dedicated learning digital support resources accompanying this textbook, please search for Database Principles: Fundamentals of Design, Implementation, and Management. Edition on: **cengage.com**

BE UNSTOPPABLE!

Learn more at **cengage.com**

DATABASE PRINCIPLES

Part I

DATABASE SYSTEMS

1 The Database Approach

2 Data Models

3 Relational Model Characteristics

4 Relational Algebra and Calculus

BUSINESS VIGNETTE

THE RELATIONAL REVOLUTION – AN HISTORICAL JOURNEY

Until the late 1970s, databases stored large amounts of data in structures that were inflexible and difficult to navigate. Programmers needed to know what clients wanted to do with the data before the database was designed. Adding or changing the way the data were stored or analysed was time-consuming and expensive.

In 1970, Edgar 'Ted' Codd, a mathematician employed by IBM, published a groundbreaking article entitled 'A Relational Model of Data for Large Shared Data Banks'. At the time, nobody realised that Codd's theories would spark a technological revolution on par with the development of personal computers and the internet. Don Chamberlin, co-inventor of SQL, the most popular database query language today, explains: 'There was this guy Ted Codd who had some kind of strange mathematical notation, but nobody took it very seriously.'

Then Ted Codd organised a symposium, and Chamberlin listened as Codd reduced complicated five-page programs to one line. 'And I said, "Wow,"' Chamberlin recalls. The symposium convinced IBM to fund System R, a research project that built a prototype of a relational database, which would eventually lead to the creation of SQL and DB2. IBM, however, kept System R on the back burner for a number of years, which turned out to be a crucial decision, because the company had a vested interest in IMS, a reliable, high-end database system that had been released in 1968.

At about the same time as System R started up, two professors from the University of California at Berkeley, who had read Codd's work, established a similar project called Ingres. The competition between the two tight-knit groups fuelled a series of papers. Unaware of the market potential of this research, IBM allowed its staff to publish these papers. Among those reading the papers was Larry Ellison, who had just founded a small company called Software Development Laboratories. Recruiting programmers from System R and Ingres, and securing funding from the CIA and the Navy, Ellison was able to market the first SQL-based relational database in 1979, well before IBM.

By 1983, the company (Software Development Laboratories) had released a portable version of the database, had grossed over €3 910 000 annually, and had changed its name to Oracle.

Spurred on by competition, IBM finally released SQL/DS, its first relational database, in 1980.[1] In 2008, a group of leading database researchers met in Berkeley and issued a report declaring that the industry had reached an exciting turning point and was on the verge of another database revolution.[2]

In 2010, Oracle acquired MySQL as part of its acquisition of Sun. It has since maintained the free open-source MySQL Community Edition while providing several versions (Standard Edition, Enterprise Edition and Cluster Edition) for commercial customers. In 2019, the release of MySQL Document Store brought together the SQL and the NoSQL languages, enabling developers to link SQL relational tables to schema-less NoSQL databases.[3] Oracle's latest offering is Oracle Database 19c, where the 'c' represents cloud; new versions now come out every year.

In our historical journey, we must also mention PostgreSQL, developed in1986 as part of the POSTGRES project at the University of California at Berkeley. PostgreSQL[4] is a free, open source, object-relational database that extends the traditional SQL language by allowing creation of new datatypes and functions, and the ability to write code in different programming languages. It is a strong competitor to MySQL, given that it has had over 33 years of active development.

Analysts, journalists and business leaders continually see new developments with data acquisition and its management, such as the explosion of unstructured data, the growing importance of business intelligence, and the emergence of cloud technologies, which may require the development of new database models. Although traditional relational databases meet rigorous standards for data integrity and consistency, they do not scale unstructured data as well as new database models such as NoSQL. NoSQL is also known as a non-relational database, which allows the storage and retrieval of unstructured data using a dynamic schema. A key question asked by database developers today is whether they need a NoSQL database or an SQL database for their application. For example, Twitter and Facebook, which do not require high levels of data consistency and integrity, have adopted NoSQL databases. In 2019, businesses are opting for SQL and NoSQL multiple database combinations, which suggests that one size does not fit all.

As of March 2019, the most popular database management systems worldwide were Oracle, MySQL, Microsoft SQL and PostgreSQL.[5] So, what is the future? Disruptive database technologies are required for business to remain competitive and the key is real-time data. Alternative database models such as cloud database platforms, which have the capability for real-time data analytics, are for certain. Big data has a role to play as additional data sources must be processed using data pipelines, all in accordance with the new General Data Protection Regulation (GDPR) data regulations. The relational model will survive, but it will also adapt at unprecedented speed.

[1] 'IBM and Oracle Trade Barbs over Databases', https://phys.org/news/2007-05-ibm-oracle-barbs-databases.html

[2] Rakesh Agrawal et al.,'The Claremont Report on Database Research', http://db.cs.berkeley.edu/claremont/claremontreport08.pdf.

[3] MySQL Editions, www.mysql.com/products/

[4] PostgreSQL, www.postgresql.org/about/

[5] Top 10 Databases for 2019, The Database Journal, www.databasejournal.com/features/oracle/slideshows/top-10-2019-databases.html

CHAPTER 1

The Database Approach

IN THIS CHAPTER, YOU WILL LEARN:

- The difference between data and information
- What a database is, what the different types of databases are, and why they are valuable assets for decision making
- The importance of database design
- How modern databases evolved from file systems
- About flaws in file system data management
- What the database system's main components are and how a database system differs from a file system
- The main functions of a database management system (DBMS)
- The role of open source database systems
- The importance of data governance and data quality

PREVIEW

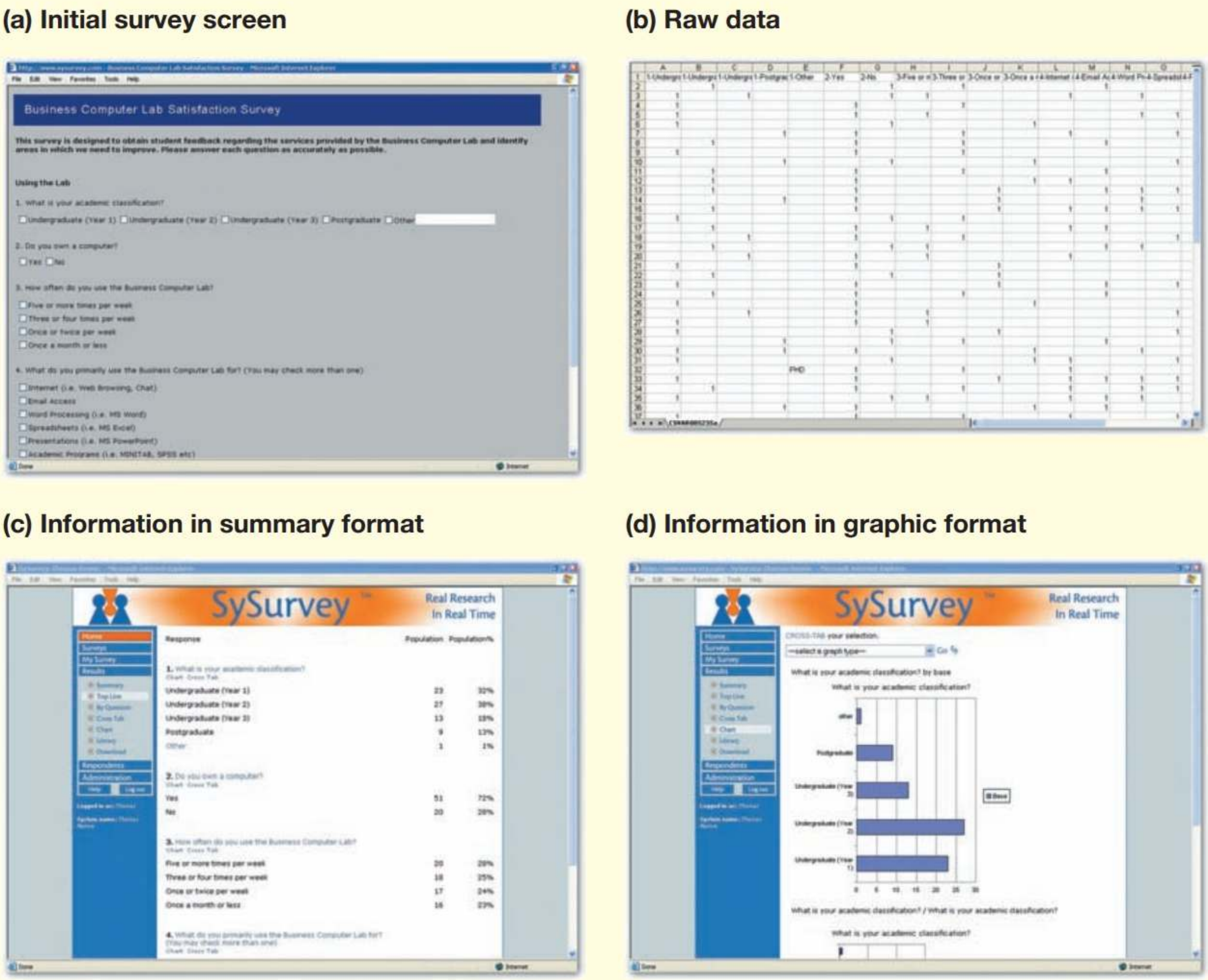
Good decisions require good information, which is derived from raw facts known as data. Data are likely to be managed most efficiently when they are stored in a database. In this chapter, you learn what a database is, what it does and why it yields better results than other data management methods. You will also learn about different types of databases and why database design is so important.

Databases evolved from computer file systems. Although file system data management is now largely outmoded, understanding the characteristics of file systems is important because they 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.

1.1 DATA VS INFORMATION

To understand what drives database design, you need to understand the difference between data and information. **Data** are raw facts. The word *raw* indicates that the facts have not yet been processed to reveal their meaning. For example, suppose that you want to know what the users of a computer lab think of its services. Typically, you would begin by surveying users to assess the computer lab's performance. Figure 1.1, Panel (a), shows the Web survey form that enables users to respond to your questions. When the survey form has been completed, the form's raw data are saved to a data repository, such as the one shown in Figure 1.1, Panel (b). Although you now have the facts in hand, they are not particularly useful in this format – reading page after page of zeros and ones is not likely to provide much insight. Therefore, you transform the raw data into a data summary like the one shown in Figure 1.1, Panel (c). It is now possible to get quick answers to questions such as, 'What is the composition of our lab's customer base?' In this case, you can quickly determine that most of your customers are second-year undergraduates (38 per cent) and first-year undergraduates (32 per cent). And, because graphics can enhance your ability to extract meaning from data quickly, you show the data summary bar graph in Figure 1.1, Panel (d).

FIGURE 1.1 Transforming raw data into information



(a) Initial survey screen

Business Computer Lab Satisfaction Survey

This survey is designed to obtain student feedback regarding the services provided by the Business Computer Lab and identify areas in which we need to improve. Please answer each question as accurately as possible.

Using the Lab

1. What is your academic classification?

□ Undergraduate (Year 1) □ Undergraduate (Year 2) □ Undergraduate (Year 3) □ Postgraduate □ Other

2. Do you own a computer?

□ Yes □ No

3. How often do you use the Business Computer Lab?

□ Five or more times per week
□ Three or four times per week
□ Once or twice per week
□ Once a month or less

4. What do you primarily use the Business Computer Lab for? (You may check more than one)

□ Internet (i.e. Web Browsing, Chat)
□ Email Access
□ Word Processing (i.e. MS Word)
□ Spreadsheets (i.e. MS Excel)
□ Presentations (i.e. MS PowerPoint)
□ Academic Programs (i.e. MINITAB, SPSS etc)

(b) Raw data

(c) Information in summary format

Response	Population	Population%
1. What is your academic classification?		
Undergraduate (Year 1)	23	32%
Undergraduate (Year 2)	27	38%
Undergraduate (Year 3)	13	18%
Postgraduate	9	13%
Other	1	1%
2. Do you own a computer?		
Yes	51	72%
No	20	28%
3. How often do you use the Business Computer Lab?		
Five or more times per week	20	28%
Three or four times per week	18	25%
Once or twice per week	17	24%
Once a month or less	16	23%
4. What do you primarily use the Business Computer Lab for? (You may check more than one)		

(d) Information in graphic format

SySurvey
Real Research In Real Time
What is your academic classification? by base
What is your academic classification?
other
Postgraduate
Undergraduate (Year 3)
Undergraduate (Year 2)
Undergraduate (Year 1)
Base
0
5
10
15
20
25
30
What is your academic classification? / What is your academic classification?