

Preface

Background

The database is now the underlying framework of the information system and has fundamentally changed the way many companies and individuals work. The developments in this technology over the last few years have produced database systems that are more powerful and more intuitive to use, and users are creating databases and applications without the necessary knowledge to produce an effective and efficient system. Looking at the literature, we found many excellent books that examine a part of the database system development lifecycle. However, we found very few that covered analysis, design, and implementation and described the development process in a simple-to-understand way that could be used by both business and business IT readers.

Our original concept therefore was to provide a book for the academic and business community that explained as clearly as possible how to analyze, design, and implement a database. This would cover both simple databases consisting of a few tables to large databases containing tens to hundreds of tables. During the initial reviews that we carried out, it became clear that the book would also be useful for the academic community and provide a very simple and clear presentation of a database design methodology that would complement a more extensive recommended textbook, such as our own book *Database System, Fourth Editions*. With the help of our reviewers and readers, we have extended our previous work, *Database Solutions, Second Edition*, to provide a more up-to-date integrated case study and added new chapters to cover advanced database systems and applications.

A Methodology for Understanding Database Design

The methodology we present in this book for relational Database Management Systems (DBMSs) – the predominant system for business applications at present – has been tried and tested over the years in both industrial and academic environments. The methodology is divided into three phases:

- a conceptual database design phase, in which we develop a model of the data used in an organization independent of *all* physical considerations;
- a logical database design phase, in which we develop a model of the data used in an organization based on a specific data model (in our case, the relational data model), but independent of a particular DBMS and other physical considerations;
- a physical database design phase, in which we decide how we are going to realize the implementation in the target DBMS, such as Microsoft Access, Microsoft SQL Server, Oracle, DB2, or Informix.

We present each phase as a series of simple-to-follow steps. For the inexperienced designer, we expect that the steps will be followed in the order described, and guidelines are provided throughout to help with this process. For the experienced designer, the methodology can be less prescriptive, acting more as a framework or checklist.

To help use the methodology and understand the important issues, we provide a comprehensive worked example that is integrated throughout the book based on an online DVD rental company called *StayHome Online Rentals*. To reinforce the methodology we also provide a second case study based on a veterinary clinic called *PerfectPets* that we use in the Exercises at the end of each chapter.

Common Data Models

As well as providing readers with additional experience of designing databases, Appendix I also provides many common data models that may be useful to you in your professional work. In fact, it has been estimated that one third of a data model consists of common constructs that are applicable to most companies and the remaining two thirds are either industry-specific or company-specific. Thus, most database design work consists of recreating constructs that have already been produced many times before in other companies. The models featured may not represent a company exactly, but they may provide a starting point from which a more suitable data model can be developed that matches the company's specific requirements. Some of the models we provide cover the following common business areas:

- Customer Order Entry
- Inventory Control
- Asset Management
- Project Management
- Course Management
- Human Resource Management
- Payroll Management.

UML (Unified Modeling Language)

Increasingly, companies are standardizing the way in which they model data by selecting a particular approach to data modeling and using it throughout their database development projects. A popular high-level data model used in logical database design, and the one we use in this book, is based on the concepts of the Entity-Relationship (ER) model. Currently there is no standard notation for an ER model. Most books that cover database design for relational DBMSs tend to use one of two conventional notations:

- Chen's notation, consisting of rectangles representing entities and diamonds representing relationships, with lines linking the rectangles and diamonds;

- Crow's Feet notation, again consisting of rectangles representing entities and lines between entities representing relationships, with a crow's foot at the end of a line representing a one-to-many relationship.

Both notations are well supported by current CASE tools. However, they can be quite cumbersome to use and a bit difficult to explain. Following an extensive questionnaire carried out on our behalf by Pearsxon Education, there was a general consensus that the favored notation for teaching database systems would become the object-oriented modeling language called **UML (Unified Modeling Language)**. UML is a notation that combines elements from the three major strands of object-oriented design: Rumbaugh's OMT modeling, Booch's Object-Oriented Analysis and Design, and Jacobson's Objectory.

There are three primary reasons for adopting UML:

- (1) It is now an industry standard and has been adopted by the Object Management Group (OMG) as the standard notation for object methods.
- (2) It is arguably clearer and easier to use. We have used both Chen's and Crow's Feet notation but made the transition to UML in 1999 and have found it much easier to sue to teach modeling concepts. The feedback we received from the two editions of the previous version of this book, *Database Solutions*, was fully supportive of the UML notation.
- (3) UML is now being adopted within academia for teaching object-oriented analysis and design (OOAD), and using UML for teaching in database courses provides more synergy.

Therefore, in this book we have adopted a simplified version of the class diagram notation from UML. We believe you will find this notation easier to understand and use.

Showing How to Implement A Design

We believe it is important to show you how to convert a database design into a physical implementation. In this book, we show how to implement the first case study (the online DVD rental company called *StayHome Online Rentals*) in the Microsoft Office Access 2003 DBMS.

Who Should Read This Book?

We have tried to write this book in a self-contained way. The exception to this is physical database design, where you need to have a good understanding of how the target DBMS operates. Our intended audience is anyone who needs to develop a database, including but not limited to the following:

- information modelers and database designers;
- database application designers and implementers;
- database practitioners;
- data and database administrators;
- Information Systems, Business IT, and Computing Science professors specializing in database design;

- database students, both undergraduate, advanced undergraduate, and graduate;
- anyone wishing to design and develop a database system.

Structure Of This Book

We have divided the book into five parts and a set of four appendices:

- Part 1 – Background. We provide an introduction to DBMSs, the relational model, and a tutorial-style chapter on SQL and QBE in Chapters 1, 2, and 3. To avoid complexity in this part of the book, we only consider the SQL Data Manipulation Language (DML) statements in Chapter 3 and cover the main Data Definition Language (DDL) statements and the programming language aspects of SQL in Appendix E. We also provide an overview of the database system development lifecycle in Chapter 4.
- Part 2 – Database Analysis and Design Techniques. We discuss techniques for database analysis in Chapter 5 and show how to use some of these techniques to analyze the requirements for the online DVD rental company *StayHome Online Rentals*. We show how to draw Entity-Relationship (ER) diagrams using UML in Chapter 6, advanced ER techniques in Chapter 7, and how to apply the rules of normalization in Chapter 8. ER models and normalization are important techniques that are used in the database design methodology we describe in Part 3.
- Part 3 – Database Design Methodology. In this part of the book we describe and illustrate a step-by-step approach for database design. In Step 1 presented in Chapter 9, we discuss conceptual data modeling and create a conceptual model for the DVD rental company *StayHome*. In Step 2 presented in Chapter 10, we discuss logical data modeling and map the ER model to a set of database tables. In Chapter 11 we discuss physical database design, and show how to design a set of base tables for the target DBMS. In this chapter we also show how to choose file organizations and indexes, how to design the user views and security mechanisms that will protect the data from unauthorized access, when and how to introduce controlled redundancy to achieve improved performance, and finally to how monitor and tune the operational system.
- Part 4 – Current and Emerging Trends. In this part of the book, we examine a number of current and emerging trends in Database Management, covering: Database Administration and Security (Chapter 12), Professional, Legal, and Ethical Issues in Database Management (Chapter 13), Transaction Management (Chapter 14), eCommerce and Database Systems (Chapter 15), Distributed and Mobile DBMSs (Chapter 16), object-oriented DBMSs (OODBMSs) and object-relational DBMSs (ORDBMSs) in Chapter 17, and Data Warehousing, OnLine Analytical Processing (OLAP), and Data Mining (Chapter 18).
- Appendices. Appendix A provides a further user view for the *StayHome Online Rentals* case study to demonstrate an extension to the basic logical database design methodology for database systems with multiple user views that have requirements, which are managed using the view integration approach. Appendix B provides a second case study based on a veterinary clinic called *PerfectPets*, which we use in the Exercises at the end of many chapters. Appendix C examines the two main alternative ER notations: Chen's notation and the Crow's Feet notation. Appendix D provides a summary of the methodology as a

quick reference guide. Appendix E presents some advanced SQL material covering the SQL DDL statements and elements of the SQL programming language, such as cursors, stored procedures and functions, and database triggers. Appendices F and G provide guidelines for choosing index and when to consider denormalization to help with Step 4.3 and Step 7 of the physical database design methodology presented in Part 3. Appendix H provides an introduction to the main object-oriented concepts for those readers who are unfamiliar with them. Finally, Appendix I provides a set of 15 common data models.

The logical organization of the book and the suggested paths through it are illustrated in Figure P.1.

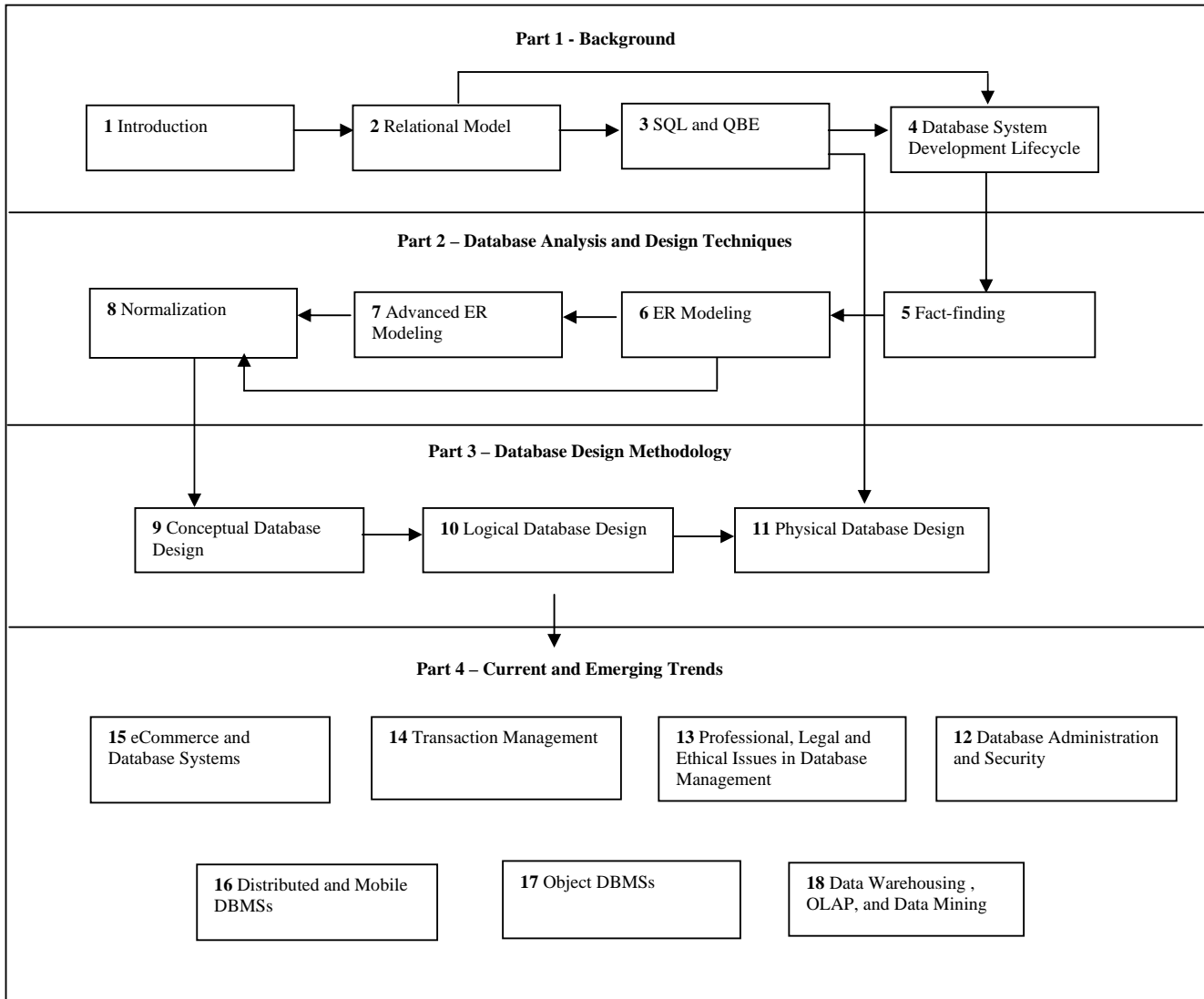


Figure P.1 Logical organization of the book and the suggested paths through it.

Pedagogy

To make the book as readable as possible, we have adopted the following style and structure:

- A set of objectives for each chapter, clearly highlighted at the start of the chapter.
- A summary at the end of each chapter covering the main points introduced.
- Review Questions and Exercises at the end of most chapters.
- Each important concept that is introduced is clearly defined and highlighted by placing the definition in a box.
- A series of notes and tips – you will see these throughout the book with an adjacent icon to highlight them.
- Diagrams are liberally used throughout to support and clarify concepts.
- A very practical orientation. Each chapter contains many worked examples to illustrate the points covered.
- A Glossary at the end of the book, which you may find useful as a quick reference guide.

Business schools that are (or aspire to be) accredited by the Association to Advance Collegiate Schools of Business (AACSB) can make immediate use of the chapter objectives to form clear and concise learning objectives mandated by the 2005 AACSB re-accreditation requirements.

Accompanying Instructor's Guide and Web Site

A comprehensive supplement containing numerous instructional resources is available for this textbook, upon request to Pearson Education. The accompanying Instructor's Guide includes:

- *Transparency masters* (created using PowerPoint) containing the main points from each chapter, enlarged illustrations, and tables from the text are provided to help the instructor associate lectures and class discussion to the material in the textbook.
- *Teaching suggestions* These include lecture suggestions, teaching hints, and student project ideas that make use of the chapter content.
- *Solutions* Sample answers are provided for all review questions and exercises.
- *Examination questions* Examination questions (similar to the questions at the end of each chapter), with solutions.
- *Additional case studies and assignments* A large range of case studies and assignments that can be used to supplement the Exercises at the end of each chapter, with solutions.
- An implementation of the *StayHome Online Rentals* database system in Microsoft Access 2003.
- An SQL script to create an implementation of the *PerfectPets* database system. This script can be used to

create a database in many relational DBMSs, such as Oracle, Informix, and SQL Server.

- An SQL script for each common data model defined in Appendix I to create the corresponding set of base tables for the database system. Once again, these scripts can be used to create a database in many relational DBMSs.

Additional information about the Instructor's Guide and the book can be found on the Addison Wesley Longman web site at:

<http://www.booksites.net>

Corrections and Suggestions

As this type of textbook is so vulnerable to errors, disagreements, omissions, and confusion, your input is solicited for future reprints and editions. Comments, corrections, and constructive suggestions should be sent to Pearson Education, or by electronic mail to:

thomas.connolly@paisley.ac.uk

Acknowledgements

This book is the outcome of many years of work by the authors in industry, research, and academia. It is therefore difficult to name all the people who have directly or indirectly helped us in our efforts; an idea here and there may have appeared insignificant at the time but may have had a significant causal effect. For those people we are about to omit, we apologize now. However, special thanks and apologies must first go to our families, who over the years have been neglected, even ignored, while we have been writing our books.

We would first like to thank Simon Plumtree, our editor, and Mary Lince, our production editor. We should also like to thank the reviewers of this book, who contributed their comments, suggestions, and advice. In particular, we would like to mention those who took part in the Reviewer Panel listed below.XXX.

Thomas M. Connolly

Carolyn E. Begg

Richard. Holowczak

Fall 2006