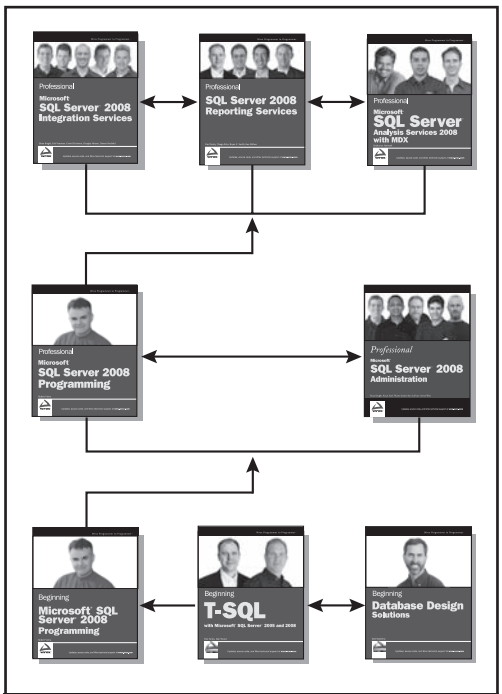
Wrox Programmer to Programmer™

Beginning Database Design Solutions

Rod Stephens

Updates, source code, and Wrox technical support at **www.wrox.com**

Beginning
Database Design Solutions



Enhance Your Knowledge
Advance Your Career

Professional Microsoft SQL Server 2008 Integration Services

978-0-470-24795-2

This book shows developers how to master the 2008 release of SSIS, covering topics including data warehousing with SSIS, new methods of managing the SSIS platform, and improved techniques for ETL operations.

Professional SQL Server 2008 Reporting Services

978-0-470-24201-8

This book teaches solutions architects, designers, and developers how to use Microsoft's reporting platform to create reporting and business intelligence solutions.

Professional Microsoft SQL Server 2008 Analysis Services

978-0-470-24798-3

Professional Microsoft SQL Server 2008 Analysis Services shows readers how to build data warehouses and multidimensional databases, query databases, and use Analysis Services and other components of SQL Server to provide end-to-end solutions.

Professional Microsoft SQL Server 2008 Programming

978-0-470-25702-9

This updated new edition of Wrox's best-selling SQL Server book has been expanded to include coverage of SQL Server 2008's new datatypes, new indexing structures, manageability features, and advanced time-zone handling.

Professional Microsoft SQL Server 2008 Administration

978-0-470-24796-9

A how-to guide for experienced database administrators, this book is loaded with unique tips, tricks, and workarounds for handling the most difficult SQL Server administration issues. The authors discuss data capture, performance studio, Query Governor, and new techniques for monitoring and policy management.

Beginning Microsoft SQL Server 2008 Programming

978-0-470-25701-2

This comprehensive introduction to SQL Server covers the fundamentals and moves on to discuss how to create and change tables, manage keys, write scripts, work with stored procedures, and much more.

Beginning T-SQL with Microsoft SQL Server 2005 and 2008

978-0-470-25703-6

Beginning T-SQL with Microsoft SQL Server 2005 and 2008 provides a comprehensive introduction to the T-SQL programming language, with concrete examples showing how T-SQL works with both SQL Server 2005 and SQL Server 2008.

Beginning Database Design Solutions

978-0-470-38549-4

Beginning Database Design Solutions introduces IT professionals—both DBAs and database developers—to database design. It explains what databases are, their goals, and why proper design is necessary to achieve those goals. It tells how to decide what should be in a database to meet the application's requirements. It tells how to structure the database so it gives good performance while minimizing the chance for error.

Beginning Database Design Solutions

Beginning
Database Design Solutions

Beginning
Database Design Solutions

Rod Stephens

Wiley Publishing, Inc.

Beginning Database Design Solutions

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN: 978-0-470-38549-4

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Stephens, Rod, 1961-
Beginning database design solutions / Rod Stephens.
p. cm.
Includes index.
ISBN 978-0-470-38549-4 (978-0-470-38549-4)
1. Database design. 2. Databases. I. Title.
QA76.9.D26S97 2008
005.74 — dc22

2008037282

About the Author

Rod Stephens started out as a mathematician but, while studying at MIT, discovered the joys of computer algorithms and programming and he's been programming professionally ever since. During his career, he has worked on a wide variety of applications in such diverse fields as telephone switching, billing, repair dispatching, tax processing, wastewater treatment, concert ticket sales, cartography, and training for professional football players.

Rod is a Microsoft Visual Basic Most Valuable Professional (MVP), consultant and author. He has written 18 books that have been translated into half a dozen different languages, and more than 250 magazine articles, mostly about Visual Basic. Currently he is a regular contributor of C# and Visual Basic articles at DevX.com (`www.devx.com`).

Rod's popular *VB Helper* Web site `www.vb-helper.com` receives several million hits per month and contains thousands of pages of tips, tricks, and example code for Visual Basic programmers, as well as example code for this book.

Credits

Executive Editor
Robert Elliott

Development Editor
Sydney Jones

Technical Editor
Steve Hoberman

Production Editor
Angela Smith

Copy Editor
Kim Cofer

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Project Coordinator, Cover
Lynsey Stanford

Proofreader
Publication Services, Inc.

Indexer
Jack Lewis

Acknowledgments

Thanks to Bob Elliott, Sydney Jones, Steve Hoberman, and all of the others whose hard work went into producing this book.

Special thanks to Sydney Jones for putting up with my sometimes overly generous interpretation of the Wrox guidelines.

Contents

Introduction

It has been estimated that more than 80 percent of all computer programming is database-related. This is certainly easy to believe. After all, a database can be a powerful tool for doing exactly what computer programs do best: store, manipulate, and display data.

Even many programs that seem at first glance to have little to do with traditional business-oriented data use databases to make processing easier. In fact, looking back on more than 20 years of software development experience, I'm hard pressed to think of a single non-trivial application that I've worked on that didn't use some kind of database.

Not only do databases play a role in many applications, but they also often play a critical role. If the data is not properly stored, it may become corrupted and the program will be unable to use it meaningfully. If the data is not properly organized, the program may be unable to find what it needs in a reasonable amount of time.

Unless the database stores its data safely and effectively, the application will be useless no matter how well-designed the rest of the system may be. The database is like the foundation of a building: without a strong foundation, even the best crafted building will fail, sometimes spectacularly (the Leaning Tower of Pisa notwithstanding).

With such a large majority of applications relying so heavily on databases, you would expect everyone involved with application development to have a solid, formal foundation in database design and construction. Everyone including database designers, application architects, programmers, database administrators, and project managers should ideally understand what makes a good database design. Even an application's key customers and users could benefit from understanding how databases work.

Sadly that is usually not the case. Many IT professionals have learned what they know about databases through rumor, trial-and-error, and painful experience. Over the years, some develop an intuitive feel for what makes a good database design but they may still not understand the reasons why a design is good or bad, and they may leave behind a trail of rickety, poorly constructed programs built on shaky database foundations.

This book provides the tools you need to design a database. It explains how to determine what should go in a database and how a database should be organized to ensure data integrity and a reasonable level of performance. It explains techniques for designing a database that is strong enough to store data safely and consistently, flexible enough to allow the application to retrieve the data it needs quickly and reliably, and adaptable enough to accommodate a realistic amount of change.

With the ideas and techniques described in this book, you will be able to build a strong foundation for database applications.

Who This Book Is For

This book is intended for IT professionals and students who want to learn how to design, analyze, and understand databases. The material will benefit those who want a better high-level understanding of databases such as proposal managers, architects, project managers, and even customers. The material will also benefit those who will actually design, build, and work with databases such as database designers, database administrators, and programmers. In many projects, these roles overlap so the same person may be responsible for working on the proposal, managing part of the project, and designing and creating the database.

This book is aimed at IT professionals and students of all experience levels. It does not assume that you have any previous experience with databases or programs that use them. It doesn't even assume that you have experience with computers. All you really need is a willingness and desire to learn.

What This Book Covers

This book explains database design. It tells how to plan a database's structure so the database will be robust, resistant to errors, and flexible enough to accommodate a reasonable amount of future change. It explains how to discover database requirements, build data models to study data needs, and refine those models to improve the database's effectiveness.

The book solidifies these concepts by working through a detailed example that designs a realistic database. Later chapters explain how to actually build databases using two common database products: Access 2007 and MySQL.

The book finishes by describing some of the topics you need to understand to keep a database running effectively such as database maintenance and security.

What You Need to Use This Book

This book explains database design. It tells how to determine what should go in a database and how the database should be structured to give the best results.

This book does not focus on actually *creating* the database. The details of database construction are different for different database tools so, to remain as generally useful as possible, this book doesn't concentrate on any particular database system. You can apply the techniques described here equally to whatever database tool you use, whether it's Access, SQL Server, Oracle, MySQL, or some other database product.

> *Most database products include free editions that you can use for smaller projects. For example, SQL Server Express Edition, Oracle Express Edition, and MySQL Community Server are all free.*

To remain database neutral, the book does not assume you are using a particular database so you don't need any particular software or hardware. To work through the Exercises, all you really need is a pencil and some paper. You are welcome to type solutions into your computer if you like but you may actually find working with pencil and paper easier than using a graphical design tool to draw pictures, at least until you are comfortable with database design and are ready to pick a computerized design tool.

Chapter 15, ''Microsoft Access,'' explains how to build databases using the Microsoft Access 2007 database product. If you want to follow along with the examples in that chapter and work through the Exercises, you need to have Microsoft Access 2007 installed (although other versions of Access will also work with a few differences). You can use any operating system that will run Microsoft Access 2007.

Similarly Chapter 16, ''MySQL,'' explains how to build databases using the MySQL Community Server database product. If you want to follow this chapter's examples and work through them, you will need to install MySQL Community Server. You can use any operating system that will run MySQL.

To experiment with the SQL database language described in Chapter 17, ''Introduction to SQL,'' and Chapter 18, ''Building Databases with SQL Scripts,'' you need any database product that supports SQL (that includes pretty much all relational databases) running on any operating system.

How This Book Is Structured

The chapters in this book are divided into five parts plus appendixes. The chapters in each part are described here. If you have previous experience with databases, you can use these descriptions to decide which chapters to skim and which to read in detail.

Part I: Introduction to Databases and Database Design

The chapters in this part of the book provide background that is necessary to understand the chapters that follow. You can skim some of this material if it is familiar to you but don't take it too lightly. If you understand the fundamental concepts underlying database design, it will be easier to understand the point behind important design concepts presented later.

Chapter 1, ''Goals of Effective Database Design,'' explains the reasons why people and organizations use databases. It explains a database's purpose and conditions that it must satisfy to be useful. This chapter also describes the basic ACID (Atomicity, Consistency, Isolation, Durability) and CRUD (Create, Read, Update, Delete) features that any good database should have. It explains in high-level general terms what makes a good database and what makes a bad database.

Chapter 2, ''Database Types,'' explains some of the different types of databases that you might decide to use. These include flat files, spreadsheets, hierarchical databases (XML), object databases, and relational databases. The relational database is one of the most powerful and most commonly used forms of database so it is the focus of this book, but it is important to realize that there are alternatives that may be more appropriate under certain circumstances. This chapter gives some tips on deciding which kind of database might be best for a particular project.

Chapter 3, ''Relational Database Fundamentals,'' explains basic relational database concepts such as tables, rows, and columns. It explains the common usage of relational database terms in addition to

the more technical terms that are sometimes used by database theorists. It describes different kinds of constraints that databases use to guarantee that the data is stored safely and consistently.

Part II: Database Design Process and Techniques

The chapters in this part of the book discuss the main pieces of database design. They explain how to understand what should be in the database, develop an initial design, separate important pieces of the database to improve flexibility, and refine and tune the design to provide the most stable and useful design possible.

Chapter 4, ''Understanding User Needs,'' explains how to learn about the users' needs and gather user requirements. It tells how to study the users' current operations, existing databases (if any), and desired improvements. It describes common questions that you can ask to learn about users' operations, desires, and needs, and how to build the results into requirements documents and specifications. This chapter explains what use cases are and tells how to use them and the requirements to guide database design and to measure success.

Chapter 5, ''Translating User Needs into Data Models,'' introduces data modeling. It explains how to translate the user's conceptual model and the requirements into other more precise models that define the database design rigorously. This chapter describes several database modeling techniques including user-interface models, semantic object models, entity-relationship diagrams, and relational models.

Chapter 6, ''Extracting Business Rules,'' explains how a database can handle business rules. It explains what business rules are, how they differ from database structure requirements, and how you can identify business rules. This chapter explains the benefits of separating business rules from the database structure and tells how to achieve that separation.

Chapter 7, ''Normalizing Data,'' explains one of the biggest tools in database design: normalization. Normalization techniques allow you to restructure a database to increase its flexibility and make it more robust. This chapter explains the various forms of normalization, emphasizing the stages that are most common and important: first, second, and third normal forms (1NF, 2NF, and 3NF). It explains how each of these kinds of normalization helps prevent errors and tells why it is sometimes better to leave a database slightly less normalized to improve performance.

Chapter 8, ''Designing Databases to Support Software Applications,'' explains how databases fit into the larger context of application design and lifecycle. This chapter explains how later development depends on the underlying database design. It discusses multi-tier architectures that can help decouple the application and database design so there can be at least some changes to either without requiring changes to the other.

Chapter 9, ''Common Design Patterns,'' explains some common patterns that are useful in many applications. Some of these techniques include implementing various kinds of relationships among objects, storing hierarchical and network data, recording temporal data, and logging and locking.

Chapter 10, ''Common Design Pitfalls,'' explains some common design mistakes that occur in database development. It describes problems that can arise from insufficient planning, incorrect normalization, and obsession with ID fields and performance.

Part III: A Detailed Case Study

If you follow all of the examples and exercises in the earlier chapters, by this point you will have seen all of the major steps for producing a good database design. However, it's often useful to see all of the steps in a complicated process put together in a continuous sequence. The chapters in this part of the book walk through a detailed case study following all of the phases of database design for the fictitious Pampered Pet database.

Chapter 11, ''User Needs and Requirements,'' walks through the steps required to analyze the users' problem, define requirements, and create use cases. It describes interviews with fictitious customers that are used to identify the application's needs and translate them into database requirements.

Chapter 12, ''Building a Data Model,'' translates the requirements gathered in the previous chapter into a series of data models that precisely define the database's structure. This chapter builds user-interface models, entity-relationship diagrams, semantic object models, and relational models to refine the database's initial design. The final relational models match the structure of a relational database fairly closely so they are easy to implement.

Chapter 13, ''Extracting Business Rules,'' identifies the business rules embedded in the relational model constructed in the previous chapter. It shows how to extract those rules in order to separate them logically from the database's structure. This makes the database more robust in the face of future changes to the business rules.

Chapter 14, ''Normalization and Refinement,'' refines the relational model developed in the previous chapter by normalizing it. It walks through several versions of the database that are in different normal forms. It then selects the degree of normalization that provides a reasonable tradeoff between robust design and acceptable performance.

Part IV: Implementing Databases (with examples in Access and MySQL)

Though this book focuses on abstract database concepts that do not depend on a particular database product, it's also worth spending at least some time on more concrete implementation issues. The chapters in this part of the book describe some of those issues and explain how to build databases with two different database products: Access 2007 and MySQL.

Chapter 15, ''Microsoft Access,'' explains how to build a database with Microsoft Access 2007. This chapter doesn't cover everything there is to know about Access, it just explains enough to get started and to use Access to build non-trivial databases. You can use other versions of Access to work through this chapter, although the locations of menus, buttons, and other Access features are different in different versions.

Chapter 16, ''MySQL,'' explains how to build a database with MySQL. This chapter tells where to download a free version of MySQL. It explains how to use the MySQL Command Line Client as well as some useful graphical tools including MySQL Query Browser and MySQL Workbench.

Part V: Advanced Topics

Although this book does not assume you have previous database experience, that doesn't mean it cannot cover some more advanced subjects. The chapters in this part of the book explain some more sophisticated topics that are important but not central to database design.

Chapter 17, ''Introduction to SQL,'' provides an introduction to SQL (Structured Query Language). It explains how to use SQL commands to add, insert, update, and delete data. By using SQL, you can help insulate a program from the idiosyncrasies of the particular database product that it uses to store data.

Chapter 18, ''Building Databases with SQL Scripts,'' explains how to use SQL scripts to build a database. It explains the advantages of this technique, such as the ability to create scripts to initialize a database before performing tests. It also explains some of the restrictions on this method, such as the fact that the user must create and delete tables in specific orders to satisfy table relationships.

Chapter 19, ''Database Maintenance,'' describes some of the database maintenance issues that are part of any database application. Though performing and restoring backups, compressing tables, rebuilding indexes, and populating data warehouses are strictly not database design tasks, they are essential to any working application.

Chapter 20, ''Database Security,'' explains database security issues. It explains the kinds of security that some database products provide. It also explains some additional techniques that can enhance database security such as using database views to appropriately restrict the users' access to data.

Appendixes

The book's appendixes provide additional reference material to supplement the earlier chapters.

Appendix A, ''Exercise Solutions,'' gives solutions to Exercises so you can check your progress as you work through the book.

Appendix B, ''Sample Database Designs,'' includes the designs for a variety of common database situations. These designs store information about such topics as books, movies, documents, customer orders, employee timekeeping, rentals, students, teams, and vehicle fleets.

The Glossary provides definitions for useful database and software development terms. The Glossary includes terms defined and used in this book in addition to other useful terms that you may encounter while reading other database material. This appendix can be a useful reference when you encounter an unfamiliar term on the Web or in database articles.

How to Use This Book

Because this book is aimed at readers of all experience levels, you may find some of the material familiar if you have previous experience with databases. In that case, you may want to skim chapters covering material that you already thoroughly understand.

If you are familiar with relational databases, you may want to skim Chapter 1, ''Goals of Effective Database Design,'' Chapter 2, ''Database Types,'' and Chapter 3, ''Relational Database Fundamentals.''