

Preface

Preface to the Third Edition

As with the second edition, let me begin by thanking all of those readers who have helped to make this book so successful.

The third edition of the book includes two new chapters; the first of which is Chapter 18. With the sad and, in my opinion, highly unfortunate demise of DAO at Microsoft's hands, it seemed necessary to bring the book up to speed on that aspect of ADO that gives the programmer most of the functionality of the Data Definition Language (DDL) portion of DAO.

ADOX is an acronym for *ADO Extensions for Data Definition and Security*. When making comparisons between ADO and DAO, proponents of DAO will point out that ADO does not include features for data definition—that is, features that can be used to create and alter databases and their components (tables, columns, indexes, etc.). This is precisely the purpose of ADOX. (Our concern here is with ADOX as it relates to Jet.)

Unfortunately, ADOX is not a complete substitute for DAO's data-definition features. For example, query creation in ADOX has a serious wrinkle. Namely, a query created using ADOX will not appear in the Access user interface! I elaborate on this in Chapter 18.

The other new chapter for the third edition is Chapter 19. In this chapter, I present a number of problems that are commonly encountered when dealing with data, along with their solutions couched in terms of SQL. I hope that this chapter will provide some good food for thought, as well as useful examples for your own applications.

Preface to the Second Edition

Let me begin by thanking all of those readers who have helped to make the first edition of this book so very successful. Also, my sincere thanks go to the many readers who have written some very flattering reviews of the first edition on amazon.com and on O'Reilly's own web site. Keep them coming.

With the recent release of Office 2000, and in view of the many suggestions I have received concerning the first edition of the book, it seemed like an appropriate time to do a second edition. I hope that readers will find the second edition of the book to be even more useful than the first edition.

Actually, Access has undergone only relatively minor changes in its latest release, at least with respect to the subject matter of this book. Changes for the Second Edition are:

- A discussion (Chapter 8 and Chapter 9 of Access' new VBA Integrated Development Environment. At last Access shares the same IDE as Word, Excel, and PowerPoint!)
- In response to reader requests, I have significantly expanded the discussion of the VBA language itself, which now occupies Chapter 10, Chapter 11, Chapter 12, and Chapter 13.
- Chapter 17, which is new for this edition, provides a fairly complete discussion of ActiveX Data Objects (ADO). This is also accompanied by an appendix on Open Database Connectivity (ODBC), which is still intimately connected with ADO.

As you may know, ADO is a successor to DAO (Data Access Objects) and is intended to eventually replace DAO, although I suspect that this will take considerable time. While the DAO model is the programming interface for the Jet database engine, ADO has a much more ambitious goal—it is a programming model for a *universal* data access interface called OLE DB. Simply put, OLE DB is a technology to connect to *any* type of data—traditional database data, spreadsheet data, web-based data, text data, email, and so on.

Frankly, while the ADO object model is smaller than that of DAO, the documentation is much less complete. As a result, ADO seems far more confusing than DAO, especially when it comes to issues such as how to create the infamous *connection strings*. Accordingly, I have spent considerable time discussing this and other difficult issues, illustrating how to use ADO to connect to Jet databases, Excel spreadsheets, and text files.

I should also mention that while the Access object model has undergone significant changes, as you can see by looking at Figure 14-7, the DAO object model has changed only in one respect. In particular, DAO has been upgraded from Version 3.5 to Version 3.6. Here is what Microsoft itself says about this new release:

DAO 3.6 has been updated to use the Microsoft® Jet 4.0 database engine. This includes enabling all interfaces for Unicode. Data is now provided in unicode (internationally enabled) format rather than ANSI. No other new features were implemented.

Thus, DAO 3.6 does not include any new objects, properties, or methods.

This book appears to cover two separate topics—database design and database programming. It does. It would be misleading to claim that database design and database programming are intimately related. So why are they in the same book?

The answer is that while these two subjects are not related, in the sense that knowledge of one leads directly to knowledge of the other, they are definitely *linked*, by the simple fact that a power database user needs to know something about both of these subjects to effectively create, use, and maintain a database.

In fact, it might be said that creating and maintaining a database application in Microsoft Access is done in three broad steps—designing the database, creating the basic graphical interface (i.e., setting up the tables, queries, forms, and reports), and then getting the application to perform in the desired way.

The second of these three steps is fairly straightforward, for it is mostly a matter of becoming familiar with the relatively easy-to-use Access graphical interface. Help is available for this through Access' online help system, as well as through the dozens of overblown 1,000-plus-page tomes devoted to Microsoft Access. Unfortunately, none of the books that I have seen does any real justice to the other two steps. Hence this book.

To be a bit more specific, the book has two goals:

- To discuss the basic concepts of relational database theory and design
- To discuss how to extract the full power of Microsoft Access, through programming in the Access Structured Query Language (SQL) and the Data Access Object (DAO) component of the Microsoft Jet database engine

To accomplish the first goal, I describe the how and why of creating an efficient database system, explaining such concepts as:

- Entities and entity classes
- Keys, superkeys, and primary keys
- One-to-one, one-to-many, and many-to-many relationships
- Referential integrity
- Joins of various types (inner joins, outer joins, equi-joins, semi-joins, θ -joins, and so on)
- Operations of the relational algebra (selection, projection, join, union, intersection, and so on)
- Normal forms and their importance

Of course, once you have a basic understanding of how to create an effective relational database, you will want to take full advantage of that database, which can only be done through programming. In addition, many of the programming techniques I discuss in this book can be used to create and maintain a database from within other applications, such as Microsoft Visual Basic, Microsoft Excel, and Microsoft Word.

I should hasten to add that this book is not a traditional cookbook for learning Microsoft Access. For instance, I do not discuss forms and reports, nor do I discuss such issues as database security, database replication, and multiuser issues. This is why I've been able to keep the book to a (hopefully) readable few hundred pages.

This book is for Access users at all levels. Most of it applies equally well to Access 2.0, Access 7.0, Access 8.0, Access 9.0 (which is a component of Microsoft Office 2000), and Access 2002 (which is included with Office XP). I will assume that you have a passing

acquaintance with the Access development environment, however. For instance, I assume that you already know how to create a table or a query.

Throughout the book, I will use a specific modest-sized example to illustrate the concepts discussed. The example consists of a database called LIBRARY that is designed to hold data about the books in a certain library. Of course, the amount of data used will be kept artificially small—just enough to illustrate the concepts.

The Book's Audience

Most books on Microsoft Access focus primarily on the Access interface and its components, giving little attention to the more important issue of database design. After all, once the database application is complete, the interface components play only a small role, whereas the design continues to affect the usefulness of the application.

In attempting to restore the focus on database design, this book aspires to be a kind of “second course” in Microsoft Access—a book for Access users who have mastered the basics of the interface, are familiar with such things as creating tables and designing queries, and now want to move beyond the interface to create programmable Access applications. This book provides a firm foundation on which you can begin to build your database-application development skills.

At the same time that this book is intended primarily as an introduction to Access for aspiring database-application developers, it also is of interest to more experienced Access programmers. For the most part, such topics as normal forms or the details of the relational algebra are almost exclusively the preserve of the academic world. By introducing these topics to the mainstream Access audience, *Access Database Design and Programming* offers a concise, succinct, readable guide that experienced Access developers can turn to whenever some of the details of database design or SQL statements escape them.

The Sample Code

To follow along with the sample code, you will need to set a reference in the Visual Basic Editor to the DAO object model and the ADO and ADOX object models. Once in the VB Editor, go to the *Tools* menu, choose *References*, and select the references entitled:

- Microsoft DAO 3.XX Object Model
- Microsoft ActiveX Data Objects 2.X Library
- Microsoft ADO Ext. 2.5 for DLL and Security

Organization of This Book

Access Database Design and Programming consists of 19 chapters that are divided into six parts. In addition, there are five appendixes.