Part

I

Introduction and Basic Concepts

In This Part

CHAPTER

1

Overview and Road Map

You, like many others, either realize or are considering the probability that using a database will alleviate a lot of the challenges that you are struggling with to manage information. But you are not familiar with database technology, and you don't know the best place to start or what to expect. Maybe you've tried some templates, or looked at some packaged solutions, and determined that they were not the best fit for your needs. That's where this book comes in.

This is arguably the most comprehensive book about Access database designs that you will find. The companion CD provides 31 database models, guides, and bonus material–a total of 76 files! We have included numerous tips and techniques from database developers, based on real-life examples, and have provided ready-to-use models. As we went through the process of creating the tables, we provided practical advice, included alternatives, and incorporated some sidebars to help you deal with more complex scenarios. We've managed to consolidate this into a compact book that is easy to understand and implement—written without a lot of database jargon, and in a style that we hope is comfortable for you, the businessperson.

We wanted to cover the most common business scenarios and provide you with advice and guidance from business experts who are also recognized for their database expertise. We did that by forming a team of ten Access developers who have earned worldwide recognition as Microsoft Access Most Valuable Persons (MVPs). Moreover, to ensure that the material presented in any chapter isn't just based on one person's perspective or experiences, we assigned each chapter to a team of MVPs, based on their subject matter expertise. Then each chapter was reviewed by at least two other Access MVPs.

Each of us invested hundreds of hours to make this the single best resource available! Rather than wade through countless other books, with all the different concepts, terms, and opinions that entails, this book provides a resource that will help you design solid database solutions—solutions that will support your current needs and be flexible enough to expand and change in response to the dynamics of your business environment.

Collectively, the contributors to this book and databases have well over 100 years of combined experience with Microsoft Access! And we are especially proud to provide you with the Bonus Resource database in the files for Appendix E on the book's CD. It contains many of the tools, shortcuts, and resources that we use to create and maintain our own Access applications. The ADD_BonusResource.mdb can serve as the start of your personal database development toolkit.

One of the benefits of having such a large team was drawing on their diverse background and experiences so that we could expose you to a variety of styles and techniques. There are typically several ways to create a solution for most database needs, and by demonstrating different approaches to the design process we are giving you multiple ways to think about and solve your own business challenges. In going through the examples, you will find a style and model that works best for your particular scenario, which of course will vary according to what you are facing at any particular moment.

We based the models on our experiences and have incorporated aspects from numerous projects; and we encourage you to take a similar approach when building solutions for your projects. The naming conventions we used had to evolve as we progressed, so ensuring consistency in the final files required a considerable amount of effort and renaming. Consequently, you are likely to find some legacy terms, particularly in the descriptions of fields. Although this does not diminish the functionality in any way, we encourage you to update the field names and descriptions as you incorporate the tables and fields into your projects.

For each line of business, we provide you with a ''base model'' that demonstrates how we met a set of business needs or solved a common problem. Then we expand on that base model to give you the benefit of our experience and the different ways you may approach similar situations. That way, you can adapt our models to your own circumstances. By explaining some of the alternatives, including examples and consequences, we hope to impart a wide range of database knowledge, making it easier for you to apply the experience to other areas and to respond to unique situations.

We take advantage of this approach extensively in Chapters 6 through 10. If you look at Chapters 6, 7, and 10 together, you will not only learn about database models that support sales and marketing, you will also create some core modules that can be used in countless other areas. For example, you will learn how to design a database to support a questionnaire. Those models and techniques can be used for a multitude of purposes outside of marketing. We'll expand on that momentarily.

As you create the database models for a manufacturing and inventory process in Chapters 8 and 9, you will also gain a stronger understanding of what ''normalization'' really means with respect to database design. We have provided a structured approach for listing and analyzing each detail that is considered to be pertinent to your database.

As we do not expect you to be a database expert, we focus on demonstrating methods that are effective, yet easy to learn and relatively simple to implement. Even so, we've included some divergent approaches with arguably more advanced material because we think that it is important for you to try different techniques and to learn something new about database design in every chapter. The best approach, of course, is the one that works for you and has the best results.

Throughout the book we have been careful to emphasize four stages that are essential to good database development. You will investigate needs, evaluate the details, convert the details into tables and fields, and then review the relationships and model as a whole. The first two stages are critical to the planning process. During the third stage, you will experience several approaches to creating the tables and fields. The fourth stage ensures that the design is complete and meets your needs.

For many database projects, it is appropriate to allocate one-third of the entire time to planning. No, that is not a typo. The planning process can easily take one-third of the overall time involved from conceptualization to deployment. The time spent on investigation, interviews, development of the database requirements and specifications, and diagramming the data flow provides the foundation for the database project. With the solid foundation, you can proceed with confidence to create the tables, design the forms and reports, and finally, to test and deploy the solution.

While building dozens of carefully created database models, this book emphasizes sound database design in every chapter. We explain the concepts that you need to follow, and occasionally some pitfalls to avoid. As you will learn, many designs that initially seem well constructed cannot withstand the test of time. A poorly designed database can easily be locked into a format that limits modifications, leads to poor performance and frustrates both users and developers. By working through the examples in this book, you will learn how to create a solid database that will stand up to years of use, modifications, and expansion. You will understand the benefits of ''normalization.''

The first part of this book explains the underlying concepts of and fundamental principles for creating sustainable database designs. The subsequent parts discuss and demonstrate how to create database models for the main business functions. The chapter examples map well to the mission of a business or department. The end of the book includes 10 appendixes that provide a wealth of resources and reference material that complement and expand on the concepts discussed in the chapters. We wanted the book to be as self-contained as reasonably achievable, so that it literally provides everything you need, right at your fingertips.

Although each chapter is designed to stand alone, we recommend that you read through the entire book. Each chapter contains examples that reflect different concepts and features that can be incorporated into other scenarios. By working progressively through the examples, you will develop the experience and confidence to create and modify solid database solutions for a multitude of purposes.

The files are a great starting point for building your own solutions. You can use the files as they are, or select various tables and features from several files, to lay the groundwork for a new solution. This will save you a significant amount of time and provide a solid foundation for your applications.

Part I: Introduction and Fundamentals

The first three chapters provide some background information to help you understand the flow of the book, an explanation of the processes that we'll use, and the terminology and fundamental principles that form the cornerstone of relational database design.

Chapter 1: Overview and Road Map

This first chapter serves as a road map for the rest of the book. It describes the contents of each chapter, including highlights, techniques, and bonus materials that you will find. By reading Chapter 1, you will learn how the business functions and data management techniques from the various chapters relate to and complement each other.

Chapter 2: Elements of a Microsoft Access Database

After reading Chapter 2 you will be familiar with the different parts of a database, including the names and nomenclature you need to get started. The information you learn from this chapter, coupled with the information from Chapter 3, starts you off on a path to building efficient, reliable, and scalable databases to satisfy your business needs.

In Chapter 2, you will learn about the main database objects: the schema, tables, relationships, and queries. In a nutshell, the schema is the overall structure or plan of the database, normally expressed in a diagram; the tables hold the data; the relationships combine the tables; and queries are used to enter, view, or modify information from the tables. Each of these database elements is expanded upon with examples.

The database schema refers to the configurations, definitions, and rules that control how the data is entered, stored, and retrieved. There are additional controls built into the user interface, but our focus is on what you can do at the table level to define how your data will be stored and organized.

Each major subject is recorded in a separate table. The tables are comprised of fields and indexes. Each table contains any number of data records; and each record can contain numerous fields. The fields are used to store the individual details that make up a record. Indexes on the fields are used to organize the records within a table and help the database function more efficiently.

Relationships define the link that joins two tables together based on common key fields between them. You will learn each type of relationship that can be created: one-to-one, one-to-many, and many-to-many. You will also learn about referential integrity and how it can help you maintain clean data within your database.

Queries enable you to request data or perform an action on your data or table structure. They are predominately used to retrieve your data in a specific order based on values in the fields. We define different types of queries that you will use, and explain how to use joins to associate data from multiple tables.

With a basic understanding of how the tables and queries work with Access, you can then see how they are used to support a relational data model.

Chapter 3: Relational Data Model

The discussion of the relational data model (RDM) and normalization builds on the database fundamentals that were covered in Chapter 2. Together, these two chapters give you the terminology and the basics for understanding the RDM and how Access implements it. Although this chapter may seem somewhat academic, it provides an important foundation for understanding why and how you should organize data into appropriate tables and fields to support an efficient database.

This chapter covers some of the history of the RDM and its inventor. This is by far the most popular database model in use today. The same basic model used by Access is used in all other popular database programs, even the ‘‘heavy hitters’’ such as Oracle and Microsoft SQL Server. The model is based on mathematical set theory. What the inventor called a ‘‘relation’’ we now call a table. The purpose of the RDM is to store the data in a logical, consistent manner so you can later retrieve exactly the data you need to process and display it, or to create reports from it.

In addition to learning about the relational data model, you will learn a set of criteria known as the *normal forms*. The process of applying the normal forms, called *normalization*, helps build databases that follow the relational data model. While explaining the normal forms and normalization, we have tried to use terms that are as nontechnical as possible to reduce the possibility of confusion.

The discussions are accompanied by visual examples that demonstrate both what your data should look like when it complies with a normal form, and what the data might look like if it does not.

With a basic understanding of the fundamentals for database tables and designs, you are ready to start applying the concepts. You begin by creating a database for managing data about people, a subject that everyone is familiar with.

Part II: Dealing with Customers and Customer Data

People and organizations are topics you will find in many serious business database applications, regardless of the business function that the application addresses. The chapters in this section guide you through creating database models to manage information about people and organizations of all types.

In looking at customer relationship management, we will consider ways to leverage the data that you might have so that you can better serve your customers or clients, whether they are businesses, individuals, or any other entity.

The last two chapters in this part demonstrate how to create databases that support marketing processes and sales. When discussing marketing, we also provide some examples that demonstrate how to design questionnaires. You can easily apply the techniques to inspection reports, surveys, checklists, voting forms, and a multitude of other purposes.

Chapter 4: People, Organizations, Addresses

For those of you who cannot wait to dig into real-life examples, this chapter provides a wealth of them. People, organizations, and addresses are a good starting point because nearly everyone is familiar with, or needs to organize, data about people and organizations. We all deal with data about individuals, and most of us are members of some type of organization or association, whether business, noncommercial, or social. Many of you have likely used some form of computerized data, and you are probably quite familiar with the business functions described in the chapter. This type of data will be part of most of the business functions that you will be modeling in your new databases.

You'll take a look at two models for storing information about people and organizations: a simpler one that is appropriate for use with business functions that are not centered on people and organizations; and a more complex design suited for larger business functions that are more people- and organization-centric. The data you need in both is very similar, but they are modeled differently. In either case, it is quite likely that you'll want to modify them by adding fields for data that is specific to your database application. Adding fields is easy; structuring your table layout is not as easy.

In addition to explaining the models and how they correspond to various business functions, this chapter also includes some bonus tables, including some that are populated with reference data. This chapter provides several tables that are used in subsequent chapters.

Chapter 5: Customer Relationship Management

Feeling intimidated by the prices of market-leading customer relationship management (CRM) packages, or even by the lower-end packages and the costs of setting them up and running them? Here's something that will give you a lift: Enterprise organizations who buy those packages use them to manage aspects of their business that affect their relationships with their customers. You can do the same for your small to medium-size business with database applications that you create yourself, and for a fraction of the cost of the top-rated CRM packages. Moreover, because you develop them, you can easily tailor your applications to the exact way you do business. Even if you hire a database developer to help with the design and modifications, the overall costs can be significantly lower and the benefits markedly higher.

You will begin with a review of business functions, identifying particular activities in your organization that can affect customer relations, and considering the subjects and details that lead to the data you will need to gather, collect, and store. First, you consider people and organization information, a familiar subject if you just read the previous chapter. Then you proceed to business functions and communication with your customers; tracking your sales and service functions; marketing, advertising, and awareness campaigns; and sharing information within your organization. These are factors important to your relationship with your customers, and any improvements you can make in the way you handle them will help improve these relationships.

Note an additional ''bonus'' in creating your own applications—you can prioritize the business functions in whatever order you determine will be most beneficial to your organization. You can implement them one at a time on a schedule that fits your business workload and budget. This is a far better alternative to purchasing a package in which much of the expense is front-loaded and the generic functionality does not necessarily target the challenges you face or in the same order of priority.

Finally, if you want to examine some functions in more depth, you will see that later chapters use a different approach to address some of these same issues. You'll find separate chapters on marketing, sales, production, warehousing, inventory, and services. If you implement databases in those areas, you may be able to give yourself an added benefit. The files may provide a source for CRM data that you can use directly by linking to the tables, or you may be able to import selected data into your customer relationship management database.

Chapter 6: Marketing

Although no database can do the marketing for you, this chapter will show you how to create one that makes your marketing efforts more effective and efficient. It covers the important functionality that is included in almost every marketing solution, plus a few extra options. Of course, this chapter expands on what you have already learned about customers, addresses, and organizations. Among the highlights are three models for one of the most useful and diverse tools that you can design: the survey or questionnaire.

The first part of the chapter is focused on marketing campaigns. We start by addressing the data requirements, so you'll explore the needs, activities, and data associated with managing a marketing campaign. An important source of data for this is the existing customer base. Gathering and storing marketing information about your clients can easily be used to increase sales, improve product offerings, and realize many other similar opportunities. We'll explain how the tables support the collection and reporting of data as a marketing campaign progresses through the various stages, such as contacting prospective customers and following up on responses with targeted mailings. As we go through the process, we will create a database named c06Campaign.mdb.

The second section of the chapter explains how to create a model to support a questionnaire, which can be an effective tool for initiating contact, determining interest or satisfaction, and tailoring offerings to a customer's specific interests. We start by creating a basic survey-type questionnaire. This model is easy to create and maintain, and it should be sufficient for many of your needs. We'll also explain how to incorporate a few extra features while maintaining the basic design.

Then, you'll add to the file as we explain the concepts for a more complex design that allows a lot of flexibility for adding questions and functionality. We also demonstrate how to create an interactive questionnaire that uses the existing responses to determine subsequent questions—very much like the online surveys. The concepts used to support a questionnaire are applicable to a multitude of purposes, from marketing campaigns to inspection reports, and from satisfaction surveys to providing book reviews. We even included a bonus query, qryInterviews, that demonstrates how to bring the data together. You'll definitely want to add some more sample data and experiment with qryInterviews!

There are countless ways to leverage these techniques, enabling you to incorporate the tables and functionality into an unlimited number of projects.

This chapter expands on what you learned about organizations, people, and database tables, and provides a framework for creating a marketing database. Similar topics follow in subsequent chapters—for sales, services, and memberships.

Chapter 7: Sales

Sales is one of the most fundamental and comprehensive business areas to model; almost everyone is selling some sort of product. As with all of our examples, the first step is to gather information. You must understand the business rules concerning customers, orders, prices, taxes, quantities, deliveries, and so on. This leads to creating a list of business rules and database requirements. From this, you can determine the key subjects and details that need to be managed. The example we provide creates core tables for the subjects of customers, orders, products, and order details.

As we discuss the business scenarios and create the tables, we'll consider some alternative fields, along with their associated benefits and consequences. The first section of the chapter creates a basic model that will support a variety of sales operations. This is provided on the companion CD in the file c07Sales.mdb.

Next, the chapter covers a more complex model that includes additional requirements to support multiple price options and to identify the employees associated with the transactions. An additional table is added to support a new provision for notes, as shown in the file c07Sales_Complex.mdb.

The last segment of the chapter covers alternative design options and provides some powerful queries to demonstrate how to create customer invoices and calculate order totals. You will learn how to use a single table to store notes that relate to several subjects—somewhat analogous to using sticky notes. These features are provided in c07Sales_ComplexAlt.mdb.

After completing this chapter you should have a good idea of the design process and some alternative approaches. You will have several optional features that you can include in a variety of databases. With this, we are ready to transition our focus from people to processes. The next part of the book discusses areas related to process monitoring, including manufacturing, inventory management, and offering services. A central factor is that they associate a value with the time or process provided.

Chapter 8: Production and Manufacturing

If you are in a production and manufacturing environment, you know that a significant amount of information must be maintained. The database must be able to capture very detailed, process-specific information from the shop floor. It must also calculate and compile data to provide reports for all levels of staff and management. The example database in this chapter focuses on managing manufacturing processes. It will create alarms or notifications about processes that need attention, enable recording the deviations or changes that have been made to a process, and provide a common place to store and retrieve communications.

One special topic we will explore with you in this chapter is the self-referencing table, which is sometimes referred to as ''bill of material processing,'' especially when it refers to the products being manufactured. This type of table is used to organize hierarchical manufacturing processes. You'll find this approach useful for many purposes, such as organizational charts, file structures, and menus.

As an added bonus, in the chapter file, c08ProductionAndManufacturing.mdb, we have included the forms and queries for displaying the data using a TreeView control. The TreeView control is a powerful tool for efficiently organizing and working with data. You often see them used for menus, file lists, process controls, and organization charts

We guide you through each of the steps needed to create a database model, from concept to implementation. Following a structured approach to normalization, we'll create a database that is compact, thorough, and scalable. In walking through these steps, you will learn how to make design decisions such as creating lookup tables (or domain tables), and when it is effective to create child tables. You will also look at one-to-many, one-to-one, and many-to-many relationships between tables.

Finally, you will take a guided walk-through of the database application's operation by following the standard workflow of a typical production workday. From there, we move to the next chapter and create a database to manage inventory.

Chapter 9: Inventory Management

The goal of inventory management is to know what you have, how much of it is available, and where it is located. This chapter covers the four primary types of information needed to effectively monitor your inventory: products, vendors, storage locations, and inventory transactions—events that add or remove inventory. In addition to the necessary tables, the database for the chapter, c09InventoryManagement.mdb, also includes several queries that enable you to quickly retrieve data to support sound business decisions.

By recording transaction data, you can monitor for minimum stock conditions. Our sample database is developed in such a manner that by using the transactions table, you can calculate the quantity on hand of a product. You can also compare the in-stock levels with minimum stock conditions and determine how many replacement items to purchase to replenish your stock.

You will find that the Inventory Transaction table is a critical part of your database. It has a central role in the calculation of Quantity on Hand reports; for example, you will use Units Transacted to calculate Quantity on Hand. You also use this table to determine the rate at which products are used or sold. This will help you determine how much product you need to order and when to order it. As a bonus, we include a second database, c09PointOfSale.mdb, to emulate sales. It links to the inventory tables and automatically updates the quantity on hand.

Our database sample is designed to allow multiple vendors per product. Each vendor has its own price. In addition, pricing history is tracked, so you can see price trends by vendor and product. We also expand upon the vendors' contact information to provide the capability to store multiple addresses and contacts.

The locations in which you stock your products are managed through the use of a self-referencing table. As explained in Chapter 8, this is a special type of table structure that is highly effective for storing hierarchical information. The locations for storing products take on a hierarchical form that not only helps you group your data by varying levels of detail, but also enables you to easily manage your stock locations for each product.

Chapter 10: Services

A service can be thought of as payment for time, rather than a product. It is an agreement about the value of something performed in a period of time. As such, concepts for process management can be applied to tracking and recording services.

The cost of the service provided may be directly correlated with the time it takes, or it may depend upon how easily the provider can perform this task, especially if that provider can offer the service concurrently for many customers. In this chapter we will model the managing of appointments for a small service company, an events management example, and a process control example. We also create a process that will check for booking conflicts. The concepts can be applied to a multitude of scenarios where you need to check for or avoid overlapping tasks or subjects of any type. The pet grooming and boarding examples begin with the gathering of business rules. There are two different parts to one business: grooming and boarding, and appointments and events. Each has different requirements, so we will discuss them separately. The database for this example, c10Pets.mdb, includes the queries used to determine available openings. The chapter explains some important concepts and relatively advanced techniques related to managing schedules and booking appointments.

From appointments we move into planning complete events, and create a database to manage events, c10Events.mdb. The business scenario is for an events company that handles weddings and corporate events. The key subject areas include customers, events, venues, organizations, service types, and event details.

When you look at events from a process perspective, you can see that they can be broken down into a relatively pre-defined set of steps performed at various stages. For example, in a real estate title and sales business, the steps would likely include contact with clients, letters sent to inform parties of progress, completion and submission of documents, and inspections or searches performed on the property. We've included the file c10Events_Complex.mdb to provide some design alternatives that are not covered in the chapter, such as a centralized contact management layout that can be used to prevent duplication of phone numbers and addresses.

Having described the key business operations, the next part of the book shifts the focus to the administrative tasks related to financial data.

Part IV: Tracking and Analyzing Financial Data

In this part, we look at the financial side of a business. This follows a slightly different format; instead of each chapter being dedicated to a different business function, the tables are explained in Chapter 11, and the accounting functions are explained in Chapter 12.

The data stored in an accounting system can hold the key to a successful business, but only if it is used correctly. If the owners or management only use the data to meet the minimum required reports, important trends will be missed. Obviously, in order to take advantage of any strengths that an analysis may reveal, you must first perform the evaluation. Conversely, if you don't conduct the analysis, a weakness may be overlooked and earnings may suffer.

Chapter 11: Accounting Systems—Requirements and Design

An accounting database is not a solution that ensures a successful business, but rather an essential tool used to increase the chances of success. It is the business strategy systematically translated into financial numbers. The key to a successful accounting and budgeting system is the process used to prepare the projections. The data stored in an accounting system can hold the key to a successful business, but only if it is used correctly. In addition to supporting the reporting requirements, it is an invaluable tool for identifying trends and responding to novel activities.

This chapter covers the basic requirements and principles for an accounting system. We will discuss how to create and use a chart of accounts, the general ledger, subsidiary ledgers, and virtual subsidiary ledgers. You will learn how to create the necessary tables and how to establish the relationships between the tables.

In the discussion of subsidiary ledgers (subledgers), we demonstrate how to create a virtual subledger that uses the detail from the general ledger to create a view of the data, rather than use a separate table that is storing duplicate data. This technique ensures that the subledger always balances with the general ledger. Because the virtual subledger involves the use of queries, the chapter contains a detailed explanation of the queries, and we've included the actual queries in the chapter file, c11AccountingSystems.mdb.

The final section of the chapter describes the tables, queries, and steps used to support standard year-end accounting processes. After the accounting year has been closed, it is common to move data from active tables to *archive tables*. This also gives you the opportunity to create and use the queries needed to calculate and verify record counts and account balances.

After finishing the chapter, you should have all the necessary resources to create the basic model for an accounting system. The file that we create in this chapter is also used in Chapter 12.

Chapter 12: Accounting: Budgeting, Analysis, and Reporting

In this chapter, you will learn about recording financial transactions, working with budget data to perform analysis and create reports. Using the tables created in Chapter 11, the focus is now on understanding the accounting principles and creating the queries that support financial analysis and reporting. We'll illustrate the queries in Access design view as we explain the tables, fields, and criteria that you will need to use to retrieve data for your reports and analysis.

You will also learn how to create the managerial reports that are used to express formal financial goals and measure the financial results. And we will briefly discuss some of the basic concepts and explain the three major types of budgets: cash budgets, operating budgets, and capital budgets.

In addition to demonstrating how to create and use an operating budget, we discuss several helpful budgeting techniques. One example shows you how to allocate projections or expenses over a specified period of time. Because a lot of budget information and analysis is done in Excel, you will also look at different ways to import, export, and work with data in Excel.

After that, we demonstrate how to develop a base set of numbers that can be used to compare predicted and actual results. This enables you to identify any variances in budgeted vs. actual performance, and to perform ratio and financial analysis.

The final section explains how to use the data from the accounting system to create financial reports and analyze profitability. The two main reports are the balance sheet and the income statement, also known as the profit and loss statements. But before you can create the balance sheet, you need to have a trial balance, so we first explain how to create the queries that are used for the trial balance report. We'll then explain the process involved in mapping the data so that it will display on specified lines on the balance sheet and income statement reports. This is an impressive technique that you can apply to other areas outside of accounting.

Finally, we discuss several types of ratio analysis that are used to explain company performance. The main types of ratios are activity, profitability, and net operating. The query results are indicators that can help management make informed business decisions and take timely actions. Examining relationships between the statements and finding trends in those relationships is the definition of analyzing profitability.

That concludes the discussion of financial analysis. The next part of the book includes a chapter on intellectual property, followed by a chapter that covers a system for managing membership information.

Part V: Independent Areas

This part has three stand-alone chapters. Even if your projects don't fall directly into these areas, we recommend that you read the chapters. You will find many correlations to other business functions, so the concepts and techniques discussed in these chapters may benefit your other projects.

The chapters in this part are intended to help you combine concepts and models from multiple sources, and to work with data in other formats. One goal is to enable you to extend your solutions by leveraging available resources.

Chapter 13: Managing Memberships

Organizations with members abound in our society, and all but the very smallest of those will have to automate their activities in some way. This chapter covers several areas that are common to most membership organizations, whether they are formal or informal, profit or nonprofit, public or private. You'll learn how to manage several aspects of data about members, such as contact information, expiration dates, membership types and payment information, and the positions and officers.

Because organizations also hold events outside of the regular meetings and activities, we've included a section to incorporate event management. Rather than providing a full-fledged event-management solution, this material deals primarily with the planning, scheduling, and staffing of events; execution; and follow-up. Recruiting might be considered as a special activity, so we mention that here as well.

This database does not attempt to manage every aspect of a membership organization, but it does give you a good start by covering many common membership management needs. You could incorporate additional features based on functions covered in other chapters in this book or in existing programs that you or the organization might already use.

As you are preparing to create your membership database, you will also want to read the next chapter, ''Implementing the Models.'' It describes a scenario for a not-for-profit organization and demonstrates how to incorporate the examples and data from several chapters and business functions into one database.

Chapter 14: Implementing the Models

This chapter demonstrates how to leverage the features that Access offers as a rapid database development tool. Instead of focusing on creating tables, we show you how to quickly create a solution for a not-for-profit organization by using tables from three other chapters and adding a few basic forms and reports. The application is designed to track membership information, billing and dues, donations, and sales.

The goal is not to create a slick-looking application, but rather to introduce you to several methods for creating forms, reports, and other user interface (UI) objects. We also demonstrate how to create several types of queries to combine data from multiple tables and to filter or limit the results to the specific records that you need. Our examples include parameter queries, select queries, and action queries. These become the record sources for the forms and reports that we create.

This leads directly to creating the forms and walking through the process of adding specific types of controls. We create 16 forms and subforms to provide a menu-driven system to manage member information, collect dues, identify products for sale, create invoices, and send e-mail. You will also learn how to create the reports to list products for sale, create an invoice, list invoices, and create a dues notification letter.

We want to show you how to use the features in Access to quickly create a solution. We have intentionally used several different types of forms, explaining how to create the record source, and even including a little bit of code to control how the forms work. We left the forms and reports in a relatively raw state to make it easy for you to follow the process and compare your work to our examples. Our emphasis is on retrieving and presenting the specific information that you want, rather than adding the format and style that you would display to users.

Once you have created the basic forms and reports, you can easily save them with new names, and then experiment with designs and styles to get just the look and layout that you'd like. We suggest that you create some standard sections that include your group logo and similar items. You can quickly copy these into other forms and reports as desired.

As you develop forms and reports, it is highly recommended that you make frequent backups of your work. Of course, it is easy to create a copy of the complete file, but returning to an early version does not always solve the problem if you are trying to recover from a corrupted object. Sometimes you will need to rebuild forms or reports, which can be quite tedious if they are complicated. It is much easier if you have saved your code to text. As a bonus, we've created a form that contains the instructions and code to save your forms and reports to text.

This is the last chapter that walks you through creating tables and databases. The final chapter in the book talks about working with data that is stored in formats other than Access.

Chapter 15: SQL Server and Other External Data Sources

The principles covered in this book are not only applicable to databases created for Microsoft Access; they are basically the same for Microsoft SQL Server, Oracle, and many other database platforms. The way we have gathered business rules, identified subjects and attributes and created tables, established table relationships, created queries, and created reports is the same for these enterprise relational data management systems as for Access. However, the software platform you choose can affect your design.

In this chapter, we discuss different data types used in SQL Server, and compare them to what is used in Access. Server functionality is covered in detail, so you will learn about the additional features of the client-server platform in the areas of programming, robustness, and the different software editions. We also briefly discuss why server designs often exhibit a preference for narrow and long tables, lookup tables, denormalizing, optimization, problem tables and conversions, and indexing.

There are brief discussions of using Access to work with data on a server with linked tables, pass-through queries and local table storage, as well as using Access Data Project (ADP) format. In additional to SQL Server, we mention some considerations for using SharePoint and provide a short section on working with CSV and XML formats.

The Appendixes

This book contains 10 appendixes that either provide additional reference materials or elaborate on concepts presented within the chapters, supporting and complementing the material presented there. By dividing the topics into printed and bonus items, we have provided extra examples and details to make this book as self-contained as possible.

Appendix A: Field Properties

In addition to specifying the data type for a field, you can also set several field properties to control what data can be accepted, how it will be displayed, and how it can be used. This appendix expands on some of the concepts you've learned in this book as you were creating the tables and fields. In several examples, we set specific field properties to control the type of data that a field can contain or to support relationships between the tables.

Rather than leave you to learn from experience, we are providing a functional database to explore field properties and to demonstrate how to leverage the properties to make it easier to efficiently manage your data. In addition to looking at the data types and field properties as they are used in the files provided, we look at some alternatives. We'll also discuss some helpful combinations for single and multiple field indexes. We end with a detailed explanation and demonstration of how to use Yes/No data types.

Appendix B: Relationships, Joins, and Nulls

The database models that we've created have relied on you to identify and define relationships between tables; when we created queries, we used joins between the tables. In this appendix, we take a closer look at the tools Access provides to manage those database components. We also demonstrate what effect the type of relationship or join has on the records that are returned by a query.

This appendix also includes a detailed discussion about nulls, which have an unknown value, and zero-length strings, which are known to have no value. The examples provided will not only help you understand how they occur, but also how to avoid or to leverage them in your projects. We end by exploring ways to create and use domain or lookup tables.

Appendix C: Resolving Relationship Triangles

In several chapters, we presented models for managing customer data that requires a customer to be categorized as either an individual or an organization, but not both. This type of either-or scenario is quite common, and is by no means limited to dealing with customers, or even people. As is often the case, implementing a table design to support this business rule leads to hard choices between options because there is more than one viable way to do it—and they all require controls in the user interface, or forms.

In this appendix, we take a quick look at two approaches and provide the minimal tables, forms, and queries to demonstrate how they are implemented. We start with a simplified interface that uses less normalized table design, and then we explain how to implement a similar effect with a more normalized table design. The extra benefit for you is having forms and queries that you can incorporate and modify to work for your own scenarios.

Appendix D: Measures—Financial and Performance

Measures are used to evaluate the performance of a business, or segments of a business. This appendix explains how to apply financial and nonfinancial data from both internal and external sources. Financial measures, like the financial ratios and ratio analysis discussed in Chapter 12, are specific to financial results. Performance measures can be, and often are, applied to data that is not strictly financial in nature, such as customer satisfaction. This appendix and the companion files provide instructions and examples that will help you to compile and analyze the data you need to make sound business decisions.

One benefit of using nonfinancial data is creating measures that are more easily compared to other companies. Comparing the dollar amounts of a small business to the dollar amounts of a large corporation would not provide you with a valid benchmark, but if a nonfinancial measure were introduced, such as units produced, a direct comparison of unit costs could be made.

Measures must fit the needs and goals of both management and the people doing and supervising the work. We explain how to create baseline data and benchmarks so that analysis can be done over a given period of time on both quantitative and qualitative measures. The appendix and companion database provide valuable examples of how to create queries to summarize the data.

Appendix E: References and Resources

When you first start searching for help online, the volume of results returned can be overwhelming. Not only do you need to find information that relates to your specific issue, you also need to know that you are getting good advice. This appendix shares some of the sites that we like to frequent and use when we are looking for inspiration and/or help. These include newsgroups, blogs, personal or group sites, and forums. Since most of the resources listed here are online, we have provided the information in electronic format as well as in the book.

The real jewel of this appendix is the database, ADD_BonusResource.mdb. While we were pulling together the sample databases for this book, we realized that we could provide an invaluable bonus just by compiling several of the database tools that we use into one convenient file. Therefore, in addition to providing the database models and explaining the rationale for their design, the Bonus Resource database includes some tools to help you analyze, work with, and create your own database solutions.

The database includes selected forms, tables, queries, and modules from the chapter files. We've also included several bonus modules from the authors that you will find to be helpful as you design and work with other database files. They will help you to efficiently create, manage, and analyze tables. We've also provided several pre-populated reference tables that you can copy into other applications.

Appendix F: What's on the CD-ROM

The material on the CD is arguably the most valuable part of the book. Of course, reading the chapters will help you understand the database models and the real-world scenarios that they are designed to support. By exploring and using the database models you will gain an in-depth knowledge of Access and save countless hours when creating your own solutions.

This appendix provides a brief summary of the directory *AccessDatabaseDesigns* that is provided on the CD. This contains 18 folders with 31 database files and several supporting documents. If you have any hesitation about purchasing this book, we encourage you to scan the list of database models. You'll immediately realize that we've provided an unprecedented compilation of models, guidance, and explanations.

You'll find a more complete list of files and benefits in the *AccessDatabaseDesignsCDGuide* and in the *File Guide* that is in each folder. These are Word documents, so you can easily print them.

With any project of this size and scope, there is always material that cannot fit between the covers of the book; but rather than forego valuable content, we have included one chapter and four appendixes on the CD. You can open the files from the Bonus Content button on the CD Content tab. By saving the files to your computer, you can quickly search, copy and paste, and click to open references—and, of course, you may want to print and store the hardcopy with the book.

Part VI: Bonus Content

Chapter B1 (on the CD): Knowledge: Intellectual Property, Structural Capital, and Intellectual Capital

''Knowledge is power'' is an old axiom, but today a parallel statement could be that without knowledge, there can be no success. This chapter addresses three categories of knowledge that are important assets: intellectual property, structural capital, and intellectual capital.

Intellectual property is ‘‘knowledge that can be owned’’—so you may need to protect it, and you may have to pay for the use of it. As we create a database, b1IPTracking.mdb, to track intellectual property owned by or used by your company, we’ll discuss the issues involved with proving ownership, public domain and right to use, and with documenting associated costs and payments.

The second area, structural capital, refers to business assets that collect, store, and display information. This is the definition of your business processes, the procedures you follow, and the methods you employ to give your business an advantage. That advantage can quickly turn into a disadvantage if some of the information is lost. The model that we create, b1ProcessInfo.mdb, could be considered the modern replacement for the employee manual or handbook. It contains information about jobs (positions), business structure (departments), the work done in the departments, the business functions executed to perform that work, and any appropriate details.

The third area, intellectual capital, is knowledge about relationships with employees, suppliers, and customers. The discussion in this chapter focuses on creating database models to manage information relevant to relationships with employees and with suppliers. Customer relationship management is such a key area that we dedicated an entire chapter to it, Chapter 5.

The first model, b1EmployeeRel.mdb, manages information about employees. It provides tools to recognize their achievements and to record additional skills or expertise that they may have in areas not directly related to their job that might qualify them for a promotion or openings in different business areas. It can also be used as a job board, to post openings, and even to help evaluate applicants by comparing qualifications to requirements.

The second model, b1SupplierTrk.mdb, is for tracking your organization’s dealings with suppliers. You can evaluate the timeliness and accuracy of orders and deliveries, the quality of material received, and amounts spent with given suppliers. Having this information will enable you to analyze which suppliers are the best choices for a continuing or expanded business relationship.

Appendix BA: Database and Business Terms

This appendix defines many of the database and business terms as they are used in this book. Our goal is to help you understand the terminology in the context that we used it—so the definitions will not necessarily mirror those found in other reference materials.

Appendix BB: Gathering Requirements

Often overlooked or short-circuited, the initial planning phase of database development should include a significant focus on clearly defining both the overall purpose of the database and the specific needs and requirements of the users. This information will become the blueprint for building a solid foundation for the final product: an efficiently working and useful database.

This appendix looks at the process of gathering and documenting the requirements that the database being developed must meet. The documentation helps to ensure that the user and the developer share the same expectations, and it provides a checklist to confirm that the necessary functionality and features are addressed. The emphasis is on asking questions, learning about existing processes, collecting examples of current and desired forms and reports, and learning what users expect and need from the program.

Appendix BC: Data Warehousing Concepts

The concept of a data warehouse is explained as an approach to tables that are specifically designed to make it easier to create reports. The table designs typically do not conform to the rules of normalization required in a relational database. Many of the tables will contain data that has been summarized, include calculated fields, contain repeating groups, and even have the same values stored in multiple tables.

Instead of the normalized designs that you've learned about for transactional tables, the data warehouse model uses fact tables and dimension tables. Fact tables include summarized data with pre-calculated values to make reporting easier and more efficient for the user. Dimension, or dim, tables are much like lookup tables in a relational database; they provide details that describe the data in the fact tables.

A warehouse can also be imitated using temporary tables, extracting data to create them on an ad hoc basis. The dim tables that describe the facts in the temporary fact tables can be added to the permanent set of tables as needed. You also use *temporary tables* or *work tables* to create a quasi-data warehouse in Access; but instead of emulating the fact and dim tables, you might combine these into larger *flat files* specifically designed and populated to support specific types of reports.

There are two key reasons to separate the data that you are analyzing from the live process data. One, it enables you to work with a snapshot of the data. Two, the data is in a flat file and no longer normalized. The snapshot allows you to run a variety of reports against the same set of data, which ensures consistency and avoids the risk of affecting the production data. Having all the data in one table makes it much easier and faster to perform calculations and comparisons. You may even add additional fields.

After creating some basic fact and dim tables, we explain how to create a query to filter and group the data for additional calculations. This demonstrates how using a simple data warehouse design or a temporary table not only makes it easier to create complicated reports, but also improves the performance of slow queries and reports.

Appendix BD: Database Platforms

In addition to understanding the principles of a good database design, you need to consider which platform is the most appropriate for your application and deployment scenario. A platform, in this instance, refers to the software (such as Access) or the combination of hardware and software (such as when using SQL Server) that enables your database application to run efficiently.

Several platforms support the relational database model. To help you determine the appropriate platform for a particular scenario, this appendix discusses some common platforms, along with their key limitations and benefits. The main aspects to evaluate include volume of data, number of concurrent users, data security, whether the data is accessed locally or over a network, and cost.