

PART ONE

*Principles of Good
Database Design*

Chapter 1	Introduction	3
Chapter 2	Introduction to Databases	13
Chapter 3	Entity-Relationship Modeling	33
Chapter 4	Relational Data Modeling and Normalization	83

CHAPTER

One Introduction

chapter OVERVIEW

- 1.1 Introduction
- 1.2 Defining a Decision Support System
- 1.3 Web-enabled Decision Support Systems
- 1.4 Decision Support System Applications
- 1.5 Textbook Overview
- 1.6 Summary
- 1.7 Exercises

1.1

Introduction

As a new graduate, equipped with the modeling and algorithmic skills taught in a standard operations management curriculum, Susan is ready to solve real-world problems. With a knowledge and understanding of theory and applications of mathematical programming, simulation techniques, supply-chain management and other IE/OR and business topics, she is ready to help her company solve distribution problems by linear programming, inventory problems by applying the EOQ model, and manpower planning problems by integer programming. However, as she works on projects with her experienced colleagues and presents results to management, she realizes that she needs more than her models and equations. The management needs Susan to help solve decision problems, but they only want to know the final analysis; they have no time or interest in understanding the mathematical model for these problems. They want this decision analysis tool to be available as a software system in which they can modify parameters and see different results for various scenarios. However, Susan is clueless about how to develop such a system. She knows the right model but she does not know how to package her model and how to present it with friendly graphical user interface. She feels that her education did not impart to her the skills she needs to meet her job requirements.

Susan is not the only one facing problems in her job as an operations research (OR) or business decision analyst. This is a widely prevalent problem that is not addressed in the current IE/OR or business curriculums. As OR practitioners and business analysts, students are support staff members and are required to build systems for non-OR users. They must know how to package OR/business models so that they can be comfortably used by top managers and other co-workers. Real-life decision making often requires building interactive systems, which students must know how to design and implement. To summarize, students must learn sufficient information technology skills so that they can build intelligent information systems, alternatively, called *decision support systems*, which can run sophisticated models at the back end, but are friendly enough at the front end to be used comfortably by any user.

A decision support system (DSS) gives its users access to a variety of data sources, modeling techniques, and stored domain knowledge via an easy to use *graphical user interface* (GUI). For example, a DSS can use the data residing in *databases*, prepare a mathematical model using this data, solve or analyze this model using problem-specific methodologies, and can assist the user in the decision-making process through a GUI. Students are frequently being employed in positions that require developing DSS, which are gaining widespread popularity. As more and more companies install enterprise resource planning (ERP) packages and invest in building data warehouses, those who are able to create decision technology-driven applications that interface with these systems and analyze the data they provide will become increasingly valuable. Indeed, imparting DSS development skills, which combine OR/business skills with *information technology* (IT) skills, will make students highly sought after in the modern workplace.

Developing courses that teach our students how to build decision support systems has been a challenging task so far since it requires the availability of platforms that allow the integration of various technologies (data, models, codes, etc.). However, in the past few years, several platforms have become available that allow such integration. One such platform is Microsoft's .NET framework. The *Visual Basic .NET* (VB .NET) package of the .NET platform can be used in conjunction with *Microsoft Access* databases to build Windows-based, single-user database applications. The Microsoft Access database allows data storage and VB .NET, an easy to learn and yet powerful object-oriented programming (OOP) language, allows us to design and build object models and develop professional quality GUI for applications. Further, Web programming has also become easier with the introduction of *Active Server Pages .NET* (ASP .NET). This .NET

package makes use of standard *HyperText Markup Language* or HTML (the front end) and VB .NET (the back-end) to deploy elegant and easy-to-build database connected Web-enabled applications. Another powerful .NET package is *Crystal Report .NET*. Crystal Report package provides an ideal environment for creating reports and charts that can effectively summarize data. Thus, Microsoft .NET and Microsoft Access provides a platform in which fairly sophisticated DSS applications can be built. This book imparts the skills needed to build such systems.

1.1.1 Topics

This chapter discusses the following topics:

- Definition and components of decision support system and Web-enabled decision support system.
- Decision support system applications.
- An overview of the textbook.

1.2 *Defining a Decision Support System*

A **decision support system** (DSS) is a model-based or knowledge-based system intended to support managerial decision making in semi-structured or unstructured situations. A DSS is not meant to replace a decision maker, but to extend his/her decision making capabilities. It uses data, provides a clear user interface, and can incorporate the decision maker's own insights. Some of the major DSS capabilities are the following:

1. A DSS brings together human judgment and computerized information for semi-structured decision situations. Such problems cannot be conveniently solved by standard quantitative techniques or computerized systems.
2. A DSS is designed to be easy to use. User friendliness, graphical capabilities, and an interactive human-machine interface greatly increase the effectiveness of a DSS.
3. A DSS usually uses models for analyzing decision-making situations and may also include a knowledge component.
4. A DSS attempts to improve the effectiveness of decision making rather than its efficiency.
5. A DSS provides support for various managerial levels from line managers to top executives. It provides support to individuals as well as groups. It can be PC-based or Web-based.

A DSS application contains five components: database, model base, knowledge base, GUI, and user (see Figure 1.1). The database stores the data, model and knowledge bases store the collections of models and knowledge, respectively, and the GUI allows the user to interact with the database, model base, and knowledge base. The database and knowledge base can be found in a basic information system. The knowledge base may contain simple search results for analyzing the data in the database. For example, the knowledge base may contain how many employees in a company database have worked at the company for over ten years. A decision support system is an *intelligent information system* because of the addition of the model base. The model base has the models used to perform optimization, simulation, or other algorithms for advanced calculations and analysis. These models allow the decision support system to not only supply information to the user but aid the user in making a decision. We now present a more detailed look at each of these components.

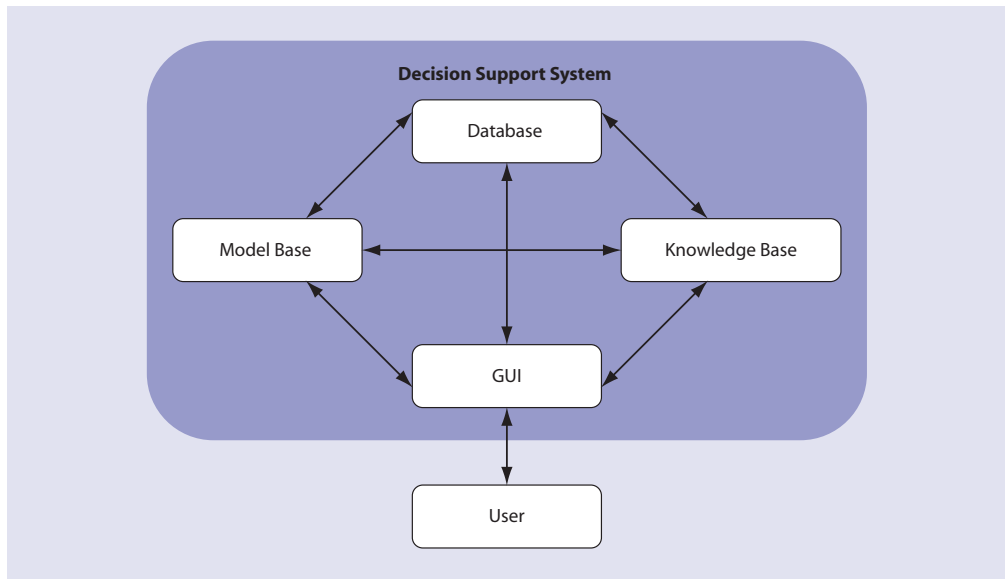


Figure 1.1 A schematic view of a decision support system.

Database: The database provides the data with which decisions are made. The data may reside in databases or even in a data warehouse, a repository for decision-making data relevant to a corporation. The database allows a user to access, manipulate, and query data. Some examples of databases include an MS Access file containing personal banking account information and a data warehouse containing shipment records of various products.

Model Base: A model base contains statistical, financial, optimization, or simulation models that provide the analysis capabilities in a DSS. Some popular optimization models include linear programming, integer programming, and nonlinear programming. The DSS allows the ability to invoke, run, and change any model or combine multiple models. An example of a model base would be an integer programming model used to solve a capital budgeting problem.

Knowledge Base: Many managerial decision-making problems are so complex that they require special expertise for their solution. The knowledge base part of a DSS allows this expertise to be stored and accessed to enhance the operation of other DSS components. For example, credit card companies use a DSS to identify credit card thefts. They store in their knowledge base the spending patterns that usually follow credit card thefts; any abnormal activity in an account would trigger checking for the presence of those patterns and a possible suspension of the account.

GUI: The graphical user interface (GUI) covers all aspects of communication between a user and a DSS application. The user interface interacts with the database, model base, and knowledge base. It allows the user to enter data or update data, run the chosen model, view the results of the model, and possibly rerun the application with different data and/or model combinations. The user interface is perhaps one of the most important components of a DSS because much of the flexibility and ease of use of a DSS are derived from this component.

User: The person who uses the DSS to support the decision-making process is called the user, or decision maker. A DSS has two broad classes of users: managers and staff specialists, or engineers. When designing a DSS, it is important to know for which class of users the DSS is being designed. In general, managers expect a DSS to be more user-friendly than do staff specialists.

A DSS should be distinguished from more common *management information systems* (MIS). An MIS can be viewed as an information system that can generate standard and exception reports and summaries for managers, provide answers to queries, and help in monitoring the performance of a system using simple data processing. A DSS can be viewed as a more sophisticated MIS in which we allow the use of models and knowledge bases to process the data and perform analysis.

1.3 *Web-Enabled Decision Support Systems*

Traditionally, MIS have been used for relatively simple data processing and data presentation. However, as these systems have become more popular and organizations have become larger and more complex, an enormous amount of data is generated routinely. That is when a need for well-developed DSS is felt that can assimilate this data and derive meaningful decisions of managerial and economic significance. Today, the era of the *Internet* has taken information sharing to new heights, allowing billions of users to share information on the *World Wide Web* (WWW) simultaneously. DSS, which are now critical to the efficient functioning of any organization, need to be upgraded and must be Web-enabled. In this book, we will discuss the development of such *Web-enabled decision support systems*.

A **Web-enabled decision support system** is a DSS that can be accessed on the World Wide Web via Internet. A typical Web-enabled decision support system requires data, a *database management system* (DBMS), a programming language, and a mechanism for Web-enabling. A DBMS is used to store, manage, and process the data, while a programming language is used to build graphical user interfaces (GUI), to do complex data processing and presentation, and to incorporate external optimization engines. Several different software packages can be used to build such a DSS. We believe that the combination of Microsoft Access (a DBMS), VB .NET (a programming language), and ASP .NET (a scripting language for Web-enabling) provides the best way to teach these concepts to students in a single course.

1.4 *Decision Support System Applications*

Given the above definition of a Web-enabled decision support system, we have developed several complete, Web-enabled DSS applications using MS Access, VB .NET, and ASP .NET. As IE/OR or business graduates, students will discover opportunities in their careers to develop and use DSS applications. Many students who have taken DSS courses have reported that their skills of combining modeling with an information technology (IT) packaging have truly helped them be outstanding at their jobs. Let us consider three examples of DSS applications that may be found in industry.

Online Shopping: Consider an example of online shopping Web sites like Amazon.com and Buy.com. Typically, for these kinds of industries, its “Web site” is the main or only way to interact with its customers. Customers visit the Web sites, browse ten hundreds of products or even search the product of interest in specific product category. They can read a detailed description of the product, view product images and reviews left by previous buyers before placing an order. Let us consider an interesting DSS application for online shopping. As customers buy products, the Web site can make some recommendations to the buyers. For example, if a customer buys a laptop of a particular brand, the Web site can suggest that people who bought

this laptop also shopped for these compatible printers and accessories. Giving such online recommendations requires sophisticated DSS support. The DSS keeps tracks of all past transactions in a database. A business model then dictates which products within a transaction are really related to each other (e.g., laptop and printer), how popular they are, discount for combined package, what is the inventory level of recommended products, and so forth. The output or decision is then displayed to the user in online manner. The system evolves as more and more transactions are made and stored in the database.

Car Production: Consider a factory that produces cars. The manager of the factory, probably a business or IE/OR graduate, may need to make some important decisions about ordering parts, hiring/firing employees, finding new suppliers, or making changes to the production process. Let us focus on a production process DSS application. The manager may be deciding where to place a new piece of equipment or how to add a new product part to the production sequence. With the use of some basic simulation and analysis tools, one could develop a DSS that allows the manager to enter the parameters to describe a possible scenario and see how it would affect production. The manager may not want to know the details of the models used, but rather what would be the effect on cost and production time and quantity if a specific change was made. One can imagine having this system accessible via internal Internet or Intranet through a Web browser within a factory to different decision makers. This is just one example of how a DSS could be used to aid in the car production industry.

Railroad Car Management: Consider a railroad company that owns several trains on which they place several thousand railroad cars that ship to several cities in the country. A distribution manager, again with an IE/OR or business background, may need to decide which cars should go on which trains to which cities. He would benefit from using an optimization model that allows him to modify certain constraints or focus on various objectives and compare the resulting distribution plans. He may want to display the car and train plans visually and may project them on a country map, to have a better understanding of the effects of one solution compared to another. A DSS would aid him in accomplishing this analysis and making a decision that considers all scenarios and possible outcomes.

The applications we develop are basic illustrations of decisions that are made in IE/OR and business industries. Two examples, selected from the DSS applications developed in Part V of this book, are described below.

Portfolio Management and Optimization: In this application, we allow users to create and/or edit their portfolios as well as to optimize their investments. To create a portfolio, users can choose from a list of stocks in an MS Access database and add or remove them to/from their portfolios. The stocks are listed by name and category. These stocks can also be compared using their historical annual returns and changes in market price. Once users have created their portfolios, they may also edit them at any time. Users may then optimize their investments by specifying a desired return on their portfolios and the amount available for investing. The application then solves an optimization problem that minimizes users' risk on their selected portfolios. The suggested investment strategy is then displayed to the users. This application allows users to interact with the database of stock information and the knowledge base of comparative statistics. The model base for this DSS uses a heuristic algorithm coded in VB .NET to perform an optimization that minimizes the user's risk on the portfolio investments.

Advertisement Allocation Problem: Television is the most popular media of entertainment. Considering its impact on us, it is natural to make optimal use of this media for advertisement and entertainment programs. Generally, a cable service provider can track information for each household; which channel and the period the viewer is tuned in on any particular channel. This information along with the data about customer, television programs, episodes, and

the show times, can be used to develop a decision support system. As one of the functionalities of the DSS, we allocate advertisement and episode based on some predetermined criteria. For example, given a set of advertisements, their target age group, race, and income group, an advertisement budget, and cost of advertising per program, we optimally allocate these advertisements to different TV programs to maximize the viewership per dollar spent. The model base for the advertisement allocation problem makes use of a greedy algorithm of knapsack problems programmed in VB .NET.

1.5 Textbook Overview

Developing Web-Enabled Decision Support Systems describes how to design a database, how to implement databases using Microsoft Access database management system, and how to add additional application functionalities using the VB .NET programming language. We also demonstrate how to Web-enable these decision support systems using ASP .NET scripting language. We illustrate this development process through five case studies based on practical situations taken from the domains of science, engineering, and management. The book contains the following five modules:

- Part I: Principles of Good Database Design
- Part II: Database Development with Microsoft Access
- Part III: Windows Application Development with VB .NET
- Part IV: Web Application Development with ASP .NET
- Part V: Case Studies

We now present an overview of these modules.

1.5.1 Overview of Principles of Good Database Design

The desire to collect and organize information for easy access predates computer technology. Before the advent of computers, or even writing, civilizations preserved important information in stories, myths, and proverbs. These civilizations honored storytellers, who maintained a verbal record of events ranging from battles to harvests. The advent of print technology gave rise to systematic data collection that promoted scribes over storytellers. From this perspective, we can view books and libraries as the initial models for the databases of today. In an era in which information technology and communication are of paramount importance, databases are pervasive. We access databases for a variety of everyday activities such as shopping online, getting driving directions, or registering for classes. This discussion emphasizes the need to master the concepts, design, and development of databases.

Part I of this book covers principles of good database design. The database design process primarily involves two kinds of data modeling: object-based modeling and record-based modeling. The object-based data modeling use entities, attributes, and relationships to present the information. The *entity-relationship* (E-R) *model* is an example of an object-based data model and has emerged as one of the most popular and widely used techniques. It describes the data in the form of an E-R diagram, which is the subject of Chapter 3. Record-based data modeling use records to present the data. The *relational data model* is an example of a record-based model. Relational data modeling is based on the mathematical concept of *relations* and uses relations to represent data and relationships. This is dealt with in detail in Chapter 4.

1.5.2 Overview of Database Development with MS Access

Having introduced the fundamentals of database design, we next delineate the process of database development based on the relational data model. There are several relational database management system (DBMS) software packages available. Some of the most popular DBMS packages are Microsoft Access, Oracle and SQL Server.

We have selected the Access DBMS to work with for several reasons. Access DBMS is immensely popular in small to medium size industries. It is easy to learn and can be used in conjunction with many other applications for Microsoft Windows. It is simple and compatible, yet powerful enough to compete with relational database management systems, such as Oracle and SQL Server, which are used by major corporations. Part II of this book is devoted to Access DBMS.

In Chapter 5, we provide a tour of Access's development environment. We introduce the development window, various menus and toolbars, and the architecture of Access and its main components. In Chapter 6, we give a detailed explanation of Access tables, the central element of relational databases. Access tables store data and provide an effective grid structure for data organization. We also discuss how to design and modify Access tables in this chapter. Chapter 7 is dedicated to relationships among Access tables. Once we have populated them with data, we can query tables to retrieve useful information. We do this through Access queries, which are the subject of Chapter 8. Chapter 9 introduces SQL, the standard language used to write database queries.

1.5.3 Overview of Windows Application Development with VB .NET

Knowledge of database design and development alone won't suffice to meet today's business needs. There is a need for an additional tool that can present processed data effectively and efficiently to the user. Presenting the data in a legitimate format is as important as processing it. A presentation tool not only broadens audience perception of our information, but also enhances the efficiency of data editing, processing, and printing operations. A *database application* allows us to achieve these goals.

With potential uses of a database application in mind, we present VB .NET as a database programming language in Part III of the book. We have organized this part as follows. In Chapter 10, we introduce the Visual Studio environment. Visual Studio is an *integrated development environment* (IDE) that provides all the tools we need to develop a database application. In Chapter 11, we introduce VB .NET along with its programming language constructs: variables, data types, control structures, operators, and arrays. Chapter 12 presents a discussion of the object-oriented support features of VB .NET, namely, classes, modules, procedures, methods, and properties. In Chapter 13, we formally introduce Windows forms and controls. Forms and controls constitute the building blocks of a GUI. In Chapter 14, we discuss database connectivity using ADO .NET. In Chapter 15, we illustrate advance database connectivity topics. Finally, in Chapter 16, we introduce Crystal Reports .NET, a powerful tool that can summarize data in the form of reports and graphs, and discuss its integration with Windows forms.

1.5.4 Overview of Web Application Development with ASP .NET

Though Windows applications are highly effective for single-user personal computers, they do not completely meet today's networking needs. The Internet era has taken information sharing to new heights, allowing billions of users to share information on the World Wide Web (WWW). This development has created the need for Web-enabled applications that are acces-

sible to a large number of users over the Internet. Today's information systems involving databases are pervasive, and a growing number of them are Web-enabled.

In Part IV of this book, we illustrate the process of developing Web applications. With the introduction of Microsoft's new technology—ASP .NET—the Web application development process became very similar to Windows application development. The database connectivity and data processing needs of an application are handled using VB .NET as in Windows applications. Thus, VB .NET forms the back-end of Web applications. As we drag and drop Windows controls on Windows forms to create Windows application GUI, ASP .NET allows us to drag and drop Web controls on Web forms to create Web GUI. As we build Web GUI, ASP .NET automatically writes HTML code of our actions. HTML is a Web standard for displaying data on Web pages. Thus, HTML forms the front-end of Web applications. We briefly discuss HTML in Chapter 17 and introduce ASP .NET in Chapter 18. In Chapter 19, we illustrate development of complete, database connected Web applications.

1.5.5 Overview of Case Studies

Part V of the book illustrates the relevance and importance of decision support systems and Web-DSS in the fields of industrial and systems engineering, business, and general engineering. We demonstrate the significance of DSS in various disciplines and showcase the development of DSS applications that integrate databases, models, methodologies, and user interfaces.

We have developed and presented five case studies, which are described in detail in this part of the book. These case studies walk through the complete process of developing a Web-enabled decision support system and are based on important applications of IE/OR and business. Through these case studies, students will learn how IE/OR and business techniques apply to real-life decision problems and how we can effectively use those techniques to build DSS applications. Our case studies include: Online Book Shopping, Television Channel—A Data Mining Application, Portfolio Management and Optimization, Voyage Prophecy—A Map Point Based Application, Accessing MS Excel Features in .NET Application—Forecasting and Optimization. Each case study also lists extensions that reader can use as additional projects or practice assignments.

1.6 Summary

- *Decision support systems* (DSS) are model-based or knowledge-based systems that support managerial decision making; they are not meant to replace decision makers, but to extend their decision-making capabilities.
- There are five components to a DSS: database, model base, knowledge base, GUI, and user.
- A *Web-enabled decision support system* is a DSS that can be accessed on WWW via Internet.
- Part I of the book covers the principles of good database design. We discuss the entity-relationship model (an example of object-based data modeling) and the relational data model (an example of record-based data modeling) in this part.
- Part II of the book is dedicated to Microsoft Access, a relational database management system (DBMS). We discuss Access tables, relationships, queries, and SQL in this part.
- A tool that can present processed data effectively and efficiently is crucial. Presentation tools not only improve audience perception of our information, but also enhance the efficiency of data editing, processing, and printing operations. Database applications allow us to achieve these goals. With potential uses of a database application in mind, we present VB .NET as a database programming language in Part III of the book.

- Web applications can be shared globally by billions of users over the Internet. In Part IV of this book, we illustrate the Web application development process with ASP .NET. We discuss HTML and VB .NET database connectivity in Web applications in this part.
- The case studies are intended to illustrate to the reader how to develop DSS applications that integrate databases, models, methodologies, and user interfaces. Part V of the book features five case studies.

1.7 Exercises

1.7.1 Review Questions

1. What are the components of a decision support system?
2. What is the difference between an information system and a decision support system?
3. What are some industrial engineering or business problems that may use databases and database applications? Which applications are suitable for extension to the Web?
4. What user interface would be necessary to communicate with a user who does not have a background in databases?
5. What are some industrial engineering applications that could require a decision support system?