CHAPTER THREE OUTLINE

STUDENT LEARNING OUTCOMES

1. List and describe the key characteristics of a relational database.
2. Define the five software components of a database management system.
3. List and describe the key characteristics of a data warehouse.
4. Define the four major types of data-mining tools in a data warehouse environment.
5. Describe business intelligence and its role in an organization.
6. List key considerations in information ownership in an organization.

PERSPECTIVES

WEB SUPPORT

www.mhhe.com/haag

- Searching job databases
- Exploring Google Earth
- Financial aid resources
- Consumer information
- Demographics
- Bureau of Labor and Statistics
- Best of computer resources and statistics
- Global statistics and resources

SUPPORTING MODULES

XLM/C Designing Databases and Entity-Relationship Diagramming
Module C presents step-by-step instructions concerning how to design the optimal structure of a database including defining entity classes and primary keys, the relationships among the entity classes, and the information to be contained in each entity class.

XLM/J Implementing a Database with Microsoft Access
Module J presents hands-on instructions concerning how to implement a database using Microsoft Access including building tables, creating queries, building simple reports, creating customized reports, and creating input forms.

CHAPTER THREE

Databases and Data Warehouses

Building Business Intelligence

OPENING CASE STUDY: CAN COMPANIES KEEP YOUR PERSONAL INFORMATION PRIVATE AND SECURE?

Without a doubt, databases are one of the most important IT tools that organizations use today. Databases are large repositories of very detailed information. When a transaction occurs, a sale, for example, a database stores every detail of the transaction including your credit card number and other personal information, which products you purchased, what discount you received, the shipping schedule for your products, and so on.

Organizations must carefully manage their databases. This management function includes properly organizing the information in these information repositories in the most efficient way, ensuring that no erroneous information ever enters the databases, and—most important—protecting the information from theft and loss.

Information is a valuable commodity, and, sadly, there are bad people who want to steal valuable information. Much of that information is personal information. When someone steals your personal information (not necessarily by taking it from you but rather stealing it from a company you do business with), you can become a victim of identity theft. Identity theft is not some isolated event, something that happens to everyone else but you. Consider this short list of organizations that have lost information and the huge numbers of customers affected.

- CardSystems (40 million customers)
- Citigroup (3.9 million customers)
- DSW Shoe Warehouse (1.4 million customers)
- Bank of America (1.2 million customers)
- Wachovia (676,000 customers)

All those incidents of information loss occurred in 2005, and they represent only some of the worst losses in terms of numbers of customers. If each customer in the above list is unique, almost 47 million people had their personal information either stolen or lost. An even more staggering information loss was reported by TJX Companies—information on 45.6 million credit cards stolen between July 2005 and January 2007.

All organizations rely on a variety of technologies to analyze, manage, and securely store information. At the very heart of every organization are databases that store vast amounts of transaction information. This information is often very personal to large numbers of individuals, and all organizations have the responsibility to keep that information secure from predators. This chapter focuses on those large repositories of information (both databases and data warehouses) and the tools organizations use to manage and secure that information.[1,2]

Questions

1. Have you been a victim of identity theft? If so, what happened?
2. What can you do to protect yourself from identity theft?
3. How many organizations have your credit card number?

Introduction

As we've discussed in the first two chapters, you and your organization need more than just data and information. You need ***business intelligence (BI)***—collective information about your customers, your competitors, your business partners, your competitive environment, and your own internal operations—that gives you the ability to make effective, important, and often strategic business decisions. Business intelligence enables your organization to extract the true meaning of information so that you can take creative and powerful steps to ensure a competitive advantage. Many such actions by your organization support some or all the initiatives we discussed in Chapter 2—customer relationship management, supply chain management, and collaboration, to name just a few.

Of course, to create business intelligence you need both data and *information* (we'll commonly refer to both as *information* in this chapter). Business intelligence doesn't just magically appear. You must first gather and organize all your information. Then, you have to have the right IT tools to define and analyze various relationships within the information. In short, knowledge workers such as you use IT tools to create business intelligence from information. The technology, by itself, won't do it for you. However, technology such as databases, database management systems, data warehouses, and data-mining tools can definitely help you build and use business intelligence.

As you begin working with these IT tools (which we'll discuss in great detail throughout this chapter), you'll be performing the two types of information processing: online transaction processing and online analytical processing. ***Online transaction processing (OLTP)*** is the gathering of input information, processing that information, and updating existing information to reflect the gathered and processed information. Databases and DBMSs are the technology tools that directly support OLTP. Databases that support OLTP are most often referred to as ***operational databases.*** Inside these operational databases is valuable information that forms the basis for business intelligence.

As you can see in Figure 3.1, you can also query operational databases to gather basic forms of business intelligence, such as how many products individually sold over $10,000 last month and how much money was spent last month on radio advertising. While the results of these queries may be helpful, you really need to combine product and advertising information (with several other types of information including customer demographics) to perform online analytical processing.

Online analytical processing (OLAP) is the manipulation of information to support decision making. At Australian P&C Direct, OLAP within a data warehouse is a must. P&C has created a data warehouse that supports its customer relationship management activities, cross-selling strategies, and marketing campaigns. By creating a data warehouse with customer information (including census data and lifestyle codes), its wide array of insurance and financial products, and its marketing campaign information, P&C agents can view all the products a given customer has purchased and more accurately determine cross-selling opportunities and what marketing campaigns a given customer is likely to respond to.[3]

A data warehouse is, in fact, a special form of a database that contains information gathered from operational databases for the purpose of supporting decision-making tasks. When you build a data warehouse and use data-mining tools to manipulate the data warehouse's information, your single goal is to create business intelligence. So, data warehouses support only OLAP; they do not at all support OLTP. As you can see in Figure 3.1, you can perform more in-depth queries to gather business intelligence

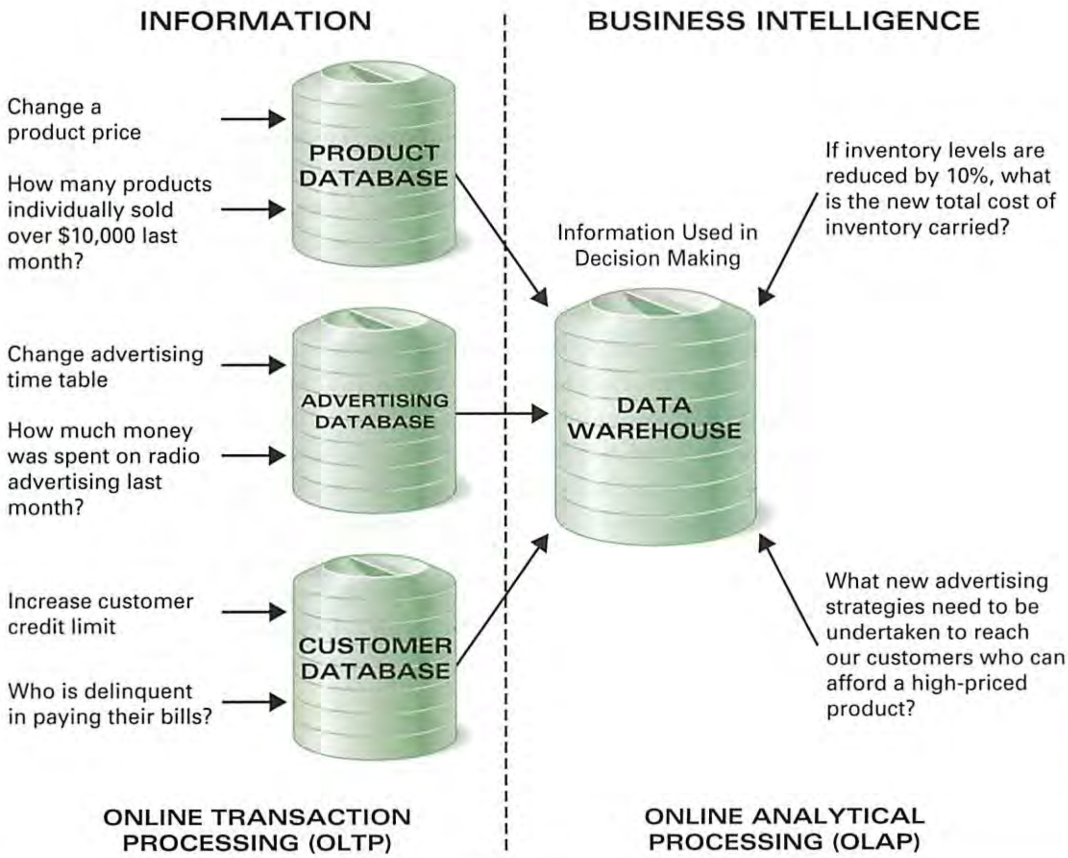


Figure 3.1

Building Business Intelligence

from a data warehouse than you can with a single database. For example, "What new advertising strategies need to be undertaken to reach our customers who can afford a high-priced product?" is a query that would require information from multiple databases. Data warehouses better support creating that type of business intelligence than do databases.

As this chapter unfolds, we'll look specifically at (1) databases and database management systems, (2) data warehouses and data-mining tools, and (3) the whole notion of business intelligence. Databases today are the foundation for organizing and managing information, and database management systems provide the tools you use to work with a database. To say the least, databases are the "heart and soul" of any organization because they organize and manage all of the organization's information resources. Data warehouses are relatively new technologies that help you organize and manage business intelligence, and data-mining tools help you extract that vitally important business intelligence.

As we first look at databases and database management systems in this chapter, we'll be exploring their use by Solomon Enterprises in support of customer relationship management and order processing. Solomon Enterprises specializes in providing concrete to commercial builders and individual homeowners in the greater Chicago area. Solomon tracks detailed information on its concrete types, customers, raw materials, raw materials' suppliers, trucks, and employees. It uses a database to organize and manage all this information. As we discuss Solomon Enterprises and its use of a database, we'll focus mostly on CRM and ordering processing. In *Extended Learning Module C,* we'll look at how to design the supply chain management side of Solomon's database.

The Relational Database Model

LEARNING OUTCOME 1

For organizing and storing basic and transaction-oriented information (that is eventually used to create business intelligence), businesses today use databases. There are actually four primary models for creating a database. The object-oriented database model is the newest and holds great promise; we'll talk more about the entire object-oriented genre in *Extended Learning Module G.* Right now, let's focus on the most popular database model: the relational database model.

As a generic definition, we would say that any ***database*** is a collection of information that you organize and access according to the logical structure of that information. In reference to a ***relational database,*** we say that it uses a series of logically related two-dimensional tables or files to store information in the form of a database. The term ***relation*** often describes each two-dimensional table or file in the relational model (hence its name *relational* database model). A relational database is actually composed of two distinct parts: (1) the information itself, stored in a series of two-dimensional tables, files, or relations (people use these three terms interchangeably) and (2) the logical structure of that information. Let's look at a portion of Solomon's database to further explore the characteristics of the relational database model.

COLLECTIONS OF INFORMATION

In Figure 3.2, we've created a view of a portion of Solomon's database. Notice that it contains five files (also, again, called tables or relations): *Order, Customer, Concrete Type, Employee,* and *Truck.* (It will contain many more as we develop it completely in *Extended Learning Module C.*) These files are all related for numerous reasons—customers make orders, employees drive trucks, an order has a concrete type, and so on. And you need all these files to manage your customer relationships and process orders.

Within each file, you can see specific pieces of information (or *attributes*). For example, the *Order* file contains *Order Number, Order Date, Customer Number, Delivery Address, Concrete Type, Amount* (this is given in cubic yards), *Truck Number,* and *Driver ID.* In the *Customer* file, you can see specific information including *Customer Number, Customer Name, Customer Phone,* and *Customer Primary Contact.* These are all important pieces of information that Solomon's database should contain. Moreover, Solomon needs all this information (and probably much more) to effectively process orders and manage customer relationships.

CREATED WITH LOGICAL STRUCTURES

Using the relational database model, you organize and access information according to its logical structure, not its physical position. So, you don't really care in which row of the *Employee* file Allison Smithson appears. You really need to know only that Allison's *Employee ID* is 984568756 or, for that matter, that her name is Allison Smithson. In the relational database model, a ***data dictionary*** contains the logical structure for the information in a database. When you create a database, you first create its data dictionary. The data dictionary contains important information (or logical properties) about your information. For example, the data dictionary for *Customer Phone* in the *Customer* file would require 10 digits. The data dictionary for *Date of Hire* in the *Employee* file would require a month, day, and year, as well.

This is quite different from other ways of organizing information. For example, if you want to access information in a certain cell in most spreadsheet applications, you must

ORDER FILE

Order Number	Order Date	Customer Number	Delivery Address	Concrete Type	Amount	Truck Number	Driver ID
100000	9/1/2004	1234	55 Smith Lane	1	8	111	123456789
100001	9/1/2004	3456	2122 E. Biscayne	1	3	222	785934444
100002	9/2/2004	1234	55 Smith Lane	5	6	222	435296657
100003	9/3/2004	4567	1333 Burr Ridge	2	4	333	435296657
100004	9/4/2004	4567	1333 Burr Ridge	2	8	222	785934444
100005	9/4/2004	5678	1222 Westminster	1	4	222	785934444
100006	9/5/2004	1234	222 East Hampton	1	4	111	123456789
100007	9/6/2004	2345	9 W. Palm Beach	2	5	333	785934444
100008	9/6/2004	6789	4532 Lane Circle	1	8	222	785934444
100009	9/7/2004	1234	987 Furlong	3	8	111	123456789
100010	9/9/2004	6789	4532 Lance Circle	2	7	222	435296657
100011	9/9/2004	4567	3500 Tomahawk	5	6	222	785934444

CUSTOMER FILE

Customer Number	Customer Name	Customer Phone	Customer Primary Contact
1234	Smelding Homes	3333333333	Bill Johnson
2345	Home Builders Superior	3334444444	Marcus Connolly
3456	Mark Akey	3335555555	Mark Akey
4567	Triple A Homes	3336666666	Janielle Smith
5678	Sheryl Williamson	3337777777	Sheryl Williamson
6789	Home Makers	3338888888	John Yu

CONCRETE TYPE FILE

Concrete Type	Type Description
1	Home foundation and walkways
2	Commercial foundation and infrastructure
3	Premier speckled (concrete with pea-size smooth gravel aggregate)
4	Premier marble (concrete with crushed marble aggregate)
5	Premier shell (concrete with shell aggregate)

EMPLOYEE FILE

Employee ID	Employee Last Name	Employee First Name	Date of Hire
123456789	Johnson	Emilio	2/1/1985
435296657	Evaraz	Antonio	3/3/1992
785934444	Robertson	John	6/1/1999
984568756	Smithson	Allison	4/1/1997

TRUCK FILE

Truck Number	Truck Type	Date of Purchase
111	Ford	6/17/1999
222	Ford	12/24/2001
333	Chevy	1/1/2002

Figure 3.2

A Portion of Solomon Enterprises' Database for Customer Relationship Management and Ordering Processing know its physical location—row number and column character. With a relational database, however, you need only know the field name of the column of information (for example, *Amount*) and its logical row, not its physical row. As a result, in Solomon's database example, you could easily change the amount for an order, without having to know where that information is physically stored (by row or column).

And with spreadsheet software, you can immediately begin typing in information, creating column headings, and providing formatting. You can't do that with a database. Using a database, you must clearly define the characteristics of each field by creating a data dictionary. So, you must carefully plan the design of your database before you can start adding information.

WITH LOGICAL TIES WITHIN THE INFORMATION

In a relational database, you must create ties or relationships in the information that show how the files relate to each other. Before you can create these relationships among files, you must first specify the primary key of each file. A ***primary key*** is a field (or group of fields in some cases) that uniquely describes each record. In Solomon's database, *Order Number* is the primary key for the *Order* file and *Customer Number* is the primary key for the *Customer* file. That is to say, every order in the *Order* file must have a unique *Order Number* and every customer in the *Customer* file must have a unique *Customer Number.*

Keys

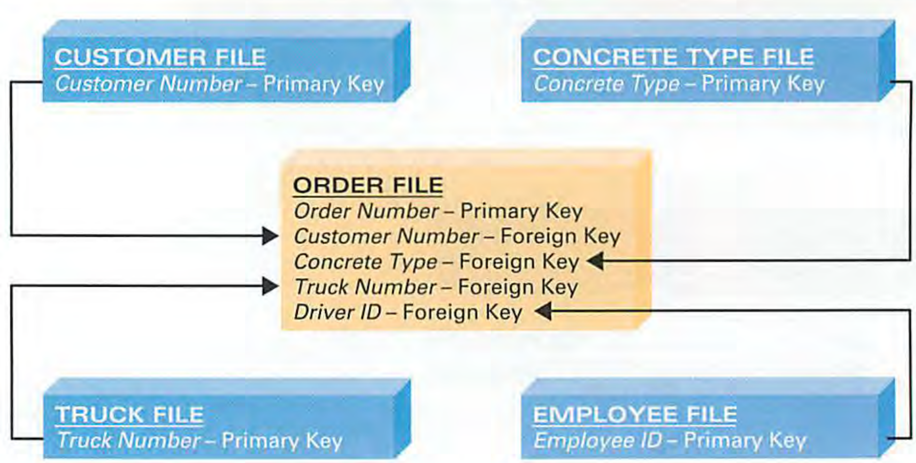
When you define that a specific field in a file is the primary key, you're also stating as well that the field cannot be blank. That is, you cannot enter the information for a new employee in the *Employee* file and leave the *Employee ID* field blank. If that were possible, you could potentially have two employees with identical primary keys (blank), which is not possible in a database environment.

Again, this is quite different from working with spreadsheets. Using a spreadsheet, it would be almost impossible to ensure that each field in a given column is unique. This reinforces the notion that, while spreadsheets work with information according to physical location, databases work with information logically.

If you look back at Figure 3.2, you can see that *Customer Number* appears in both the *Customer* and *Order* files. This creates a logical relationship between the two files and is an example of a foreign key. A ***foreign key*** is a primary key of one file that appears in another file. Now look at Figure 3.3. In it, we've provided the logical relationships among all five files. Notice, for example, that *Truck Number* is the primary key for the *Truck* file. It also appears in the *Order* file. This enables Solomon to track which trucks were used

Figure 3.3

Creating Logical Ties with Primary and Foreign Keys



INDUSTRY PERSPECTIVE

BRITANNIA AIRWAYS FLIES HIGH WITH A CONTENT MANAGEMENT DATABASE

Airlines fly millions of passengers around the globe each year. That creates a lot of paperwork, too much, in fact. According to John Gough, Britannia Airways' E-Business Program Manager, "Britannia had over 120 forms related to the flight. We urgently needed to rationalize the amount of paperwork we were generating and the efficiency of our databases processes around the paperwork."

To create a competitive advantage and reduce the paperwork involved, Britannia implemented a content management database (a special kind of database) solution provided by Open Text Corporation (www.opentext.com). The content management database tracks and organizes information such as:

- Departure and destination times and locations of flight crews and passengers
- Capacity utilization information for flight crews and baggage handlers
- Customer relationship management information to flight attendants

The new system goes far beyond just an operational database that manages and organizes transaction information. It provides each crew member with a single log-in and point of access through a Web portal. The new system provides automated workflow capabilities to ensure that the right employees receive and process the necessary information in the least amount of time. It even provides forums in which all employees can record and discuss ideas for making flights better and more comfortable for passengers.

So, the new system supports both bottom-line efforts (i.e., optimizing information processing to reduce costs) and top-line initiatives (such as customer relationship management), giving Britannia the ability to grow and nurture relationships with its most frequent flyers.[4]

to deliver the various orders. So, *Truck Number* is the primary key in the *Truck* file and is also a foreign key that appears in the *Order* file. There are other examples of foreign keys as well in Figure 3.3.

Foreign keys are essential in the relational database model. Without them, you have no way of creating logical ties among the various files. As you might guess, we use these relationships extensively to create business intelligence because they enable us to track the logical relationships within many types of information.

WITH BUILT-IN INTEGRITY CONSTRAINTS

By defining the logical structure of information in a relational database, you're also developing ***integrity constraints***—rules that help ensure the quality of the information. For example, by stating that *Customer Number* is the primary key of the *Customer* file and a foreign key in the *Order* file, you're saying (1) that no two customers can have the same *Customer Number* and (2) that a *Customer Number* that is entered into the *Order* file must have a matching *Customer Number* in the *Customer* file. So, as Solomon creates a new order and enters a *Customer Number* in the *Order* file, the database management system must find a corresponding and identical *Customer Number* in the *Customer* file. This makes perfect sense. You cannot create an order for a customer who does not exist.

Consumer Reports magazine has rated the Ritz-Carlton first among luxury hotels.[5] Why? It's simple: Ritz-Carlton has created a powerful guest preference database to provide customized, personal, and high-level service to guests of any of its hotels. For example, if you leave a message at a Ritz-Carlton front desk that you want the bed turned down at 9 P.M., prefer no chocolate mints on your pillow, and want to participate in the 7 A.M. aerobics class, that information is passed along to the floor maid (and others) and is also stored in the guest preference database. By assigning to you a unique customer ID that creates logical ties to your various preferences, the Ritz-Carlton transfers your information to all of its other hotels. The next time you stay in a Ritz-Carlton hotel, in Palm Beach for example, your information is already there, and the hotel staff immediately knows of your preferences.

For the management at Ritz-Carlton, achieving customer loyalty starts first with knowing each customer individually (the concept of customer relationship management). That includes your exercise habits, what you most commonly consume from the snack bar in your room, how many towels you use daily, and whether you like a chocolate on your pillow. To store and organize all this information, Ritz-Carlton uses a relational database, and employees use it to meet your needs (or whims).

Database Management System Tools

LEARNING OUTCOME 2

When working with word processing software, you create and edit a document. When working with spreadsheet software, you create and edit a workbook. The same is true in a database environment. A database is equivalent to a document or a workbook because they all contain information. And while word processing and spreadsheet are the software tools you use to work with documents and workbooks, you use database management system software to work with databases. A ***database management system (DBMS)*** helps you specify the logical organization for a database and access and use the information within a database. A DBMS contains five important software components (see Figure 3.4):

1. DBMS engine
2. Data definition subsystem
3. Data manipulation subsystem
4. Application generation subsystem
5. Data administration subsystem

The DBMS engine is perhaps the most important, yet seldom recognized, component of a DBMS. The ***DBMS engine*** accepts logical requests from the various other DBMS subsystems, converts them into their physical equivalent, and actually accesses the database and data dictionary as they exist on a storage device. Again, the distinction between logical and physical is important in a database environment. The ***physical view*** of information deals with how information is physically arranged, stored, and accessed on some type of storage device such as a hard disk. The ***logical view*** of information, on the other hand, focuses on how you as a knowledge worker need to arrange and access information to meet your particular business needs.

Databases and DBMSs provide two really great advantages in separating the logical from the physical view of information. First, the DBMS engine handles the physical tasks. So you, as a database user, can concentrate solely on your logical information needs. Second, although there is only one physical view of information, there may be numerous knowledge workers who have different logical views of the information in a database. That is, according to what business tasks they need to perform,

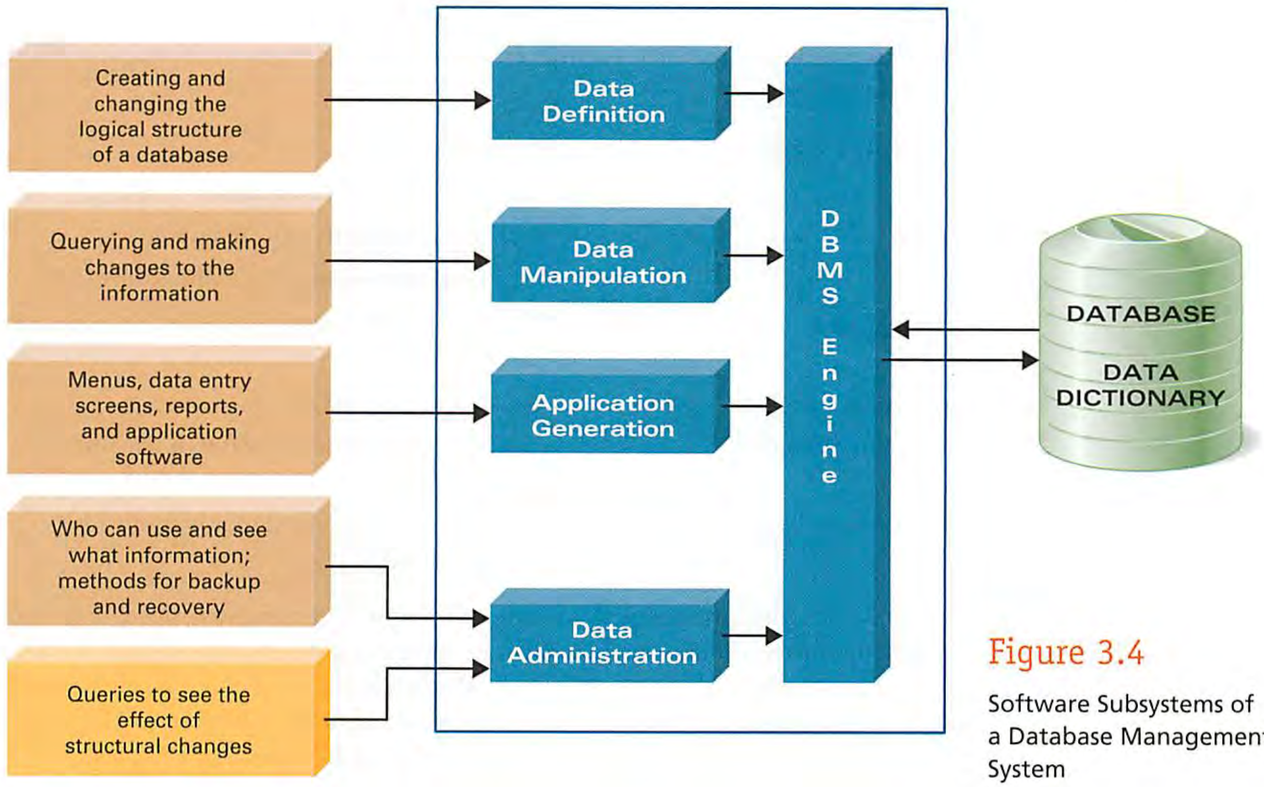


Figure 3.4

Software Subsystems of a Database Management System

different knowledge workers logically view information in different ways. The DBMS engine can process virtually any logical information view or request into its physical equivalent.

DATA DEFINITION SUBSYSTEM

The ***data definition subsystem*** of a DBMS helps you create and maintain the data dictionary and define the structure of the files in a database.

When you create a database, you must first use the data definition subsystem to create the data dictionary and define the structure of the files. This is very different from using something like spreadsheet software. When you create a workbook, you can immediately begin typing in information and creating formulas and functions. You can't do that with a database. You must define its logical structure before you can begin typing in any information. Typing in the information is the easy part: Defining the logical structure is more difficult. In *Extended Learning Module C* that follows this chapter, we take you through the process of defining the logical structure for the supply chain management (SCM) side of Solomon Enterprises' database. We definitely recommend that you read that module—knowing how to define the correct structure of a database can be a substantial career opportunity for you.

If you ever find that a certain file needs another piece of information, you have to use the data definition subsystem to add a new field in the data dictionary. Likewise, if you want to delete a given field for all the records in a file, you must use the data definition subsystem to do so.

As you create the data dictionary, you're essentially defining the logical properties of the information that the database will contain. Logical structures of information include the following:

Logical Properties	**Examples**
Field name	*Customer Number, Order Date*
Type	Alphabetic, numeric, date, time, etc.
Form	Is an area code required for a phone number?
Default value	If no *Order Date* is entered, the default is today's date.
Validation rule	Can *Amount* exceed 8?
Is an entry required?	Must you enter *Delivery Address* for an order or can it be blank?
Can there be duplicates?	Primary keys cannot be duplicates; but what about amounts?

These are all important logical properties to a lesser or greater extent depending on the type of information you're describing. For example, a typical concrete delivery truck can hold at most eight cubic yards of concrete. Further, Solomon may not accept orders for less than four cubic yards of concrete. Therefore, an important validation rule for *Amount* in the *Order* file is "must be greater than or equal to 4 and cannot be greater than 8."

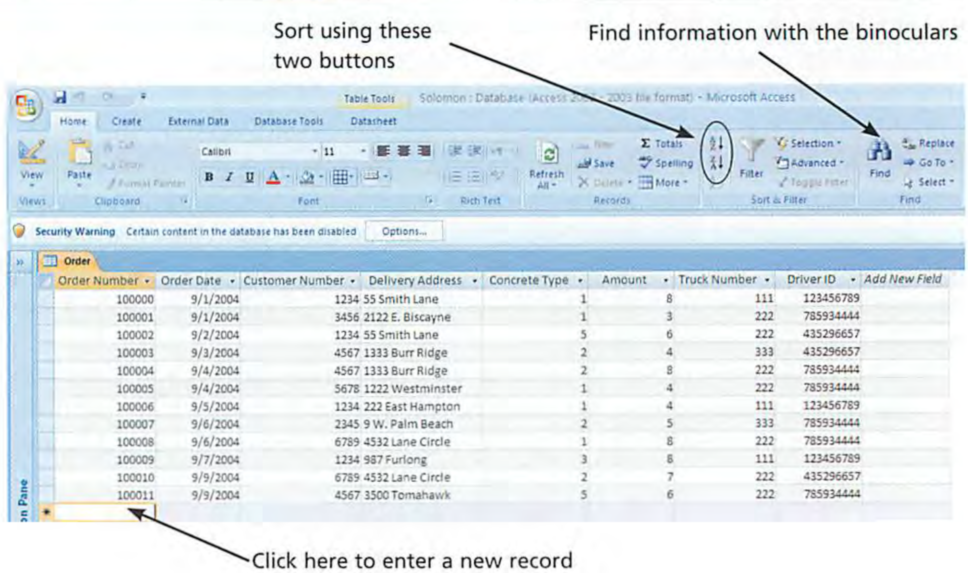
DATA MANIPULATION SUBSYSTEM

The ***data manipulation subsystem*** of a DBMS helps you add, change, and delete information in a database and query it for valuable information. Software tools within the data manipulation subsystem are most often the primary interface between you as a user and the information contained in a database. So, while the DBMS engine handles your information requests from a physical point of view, it is the data manipulation tools within a DBMS that allow you to specify your logical information requirements. Those logical information requirements are then used by the DBMS engine to access the information you need from a physical point of view.

In most DBMSs, you'll find a variety of data manipulation tools, including views, report generators, query-by-example tools, and structured query language.

Figure 3.5

A View in Microsoft Access



CUSTOMER AND AMOUNT REPORT

Customer Number	Order Number	Order Date	Amount
1234	100000	9/1/2004	8
1234	100002	9/2/2004	6
1234	100006	9/5/2004	4
1234	100009	9/7/2004	8
1234	100015	9/12/2004	8
2345	100007	9/6/2004	5
2345	100012	9/9/2004	8
3456	100001	9/1/2004	3
4567	100003	9/3/2004	4
4567	100004	9/4/2004	8
4567	100011	9/9/2004	6
4567	100013	9/10/2004	4
5678	100005	9/4/2004	4
6789	100008	9/6/2004	8
6789	100010	9/9/2004	7
6789	100014	9/10/2004	6

Figure 3.6

Using a Report Generator

VIEWS A ***view*** allows you to see the contents of a database file, make whatever changes you want, perform simple sorting, and query to find the location of specific information. Views essentially provide each file in the form of a spreadsheet workbook. The screen in Figure 3.5 shows a view in Microsoft Access for the *Order* file in Solomon's database. At this point, you can click on any specific field and change its contents. You could also point at an entire record and click on the Cut icon (the scissors) to remove a record. If you want to add a record, simply click in the *Order Number* field of the first blank record and begin typing. You can also perform a variety of other functions such as sorting, searching, spell checking, and hiding columns.

REPORT GENERATORS ***Report generators*** help you quickly define formats of reports and what information you want to see in a report. Once you define a report, you can view it on the screen or print it. Figure 3.6 shows two intermediate screens in Microsoft Access. The first allows you to specify which fields of information are to appear in a report. We have chosen to include *Customer Number, Order Number, Order Date,* and *Amount* from the *Order* file. The second allows you to choose from a set of predefined report formats. Following a simple and easy-to-use set of screens (including the two in Figure 3.6), we went on to specify that sorting should take place by *Customer Number* and that the name of the report should be "Customer and Amount Report." The completed report is also shown in Figure 3.6. Notice that it displays only those fields we requested, that it's sorted by *Customer Number*, and that the title is "Customer and Amount Report."

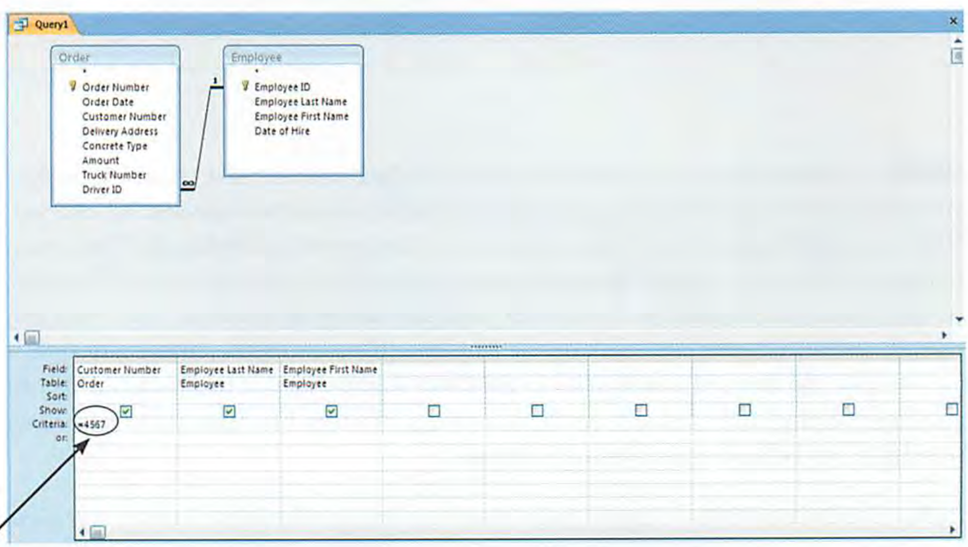
A nice feature about report generators is that you can save a report format that you use frequently. For example, if you think you'll use the report in Figure 3.6 often, you can save it by giving it a unique name. Later, you can request that report and your DBMS will generate it, using the most up-to-date information in the database. You can also choose from a variety of report formats (we chose a simple one for our illustration). And you can choose report formats that create intermediate subtotals and grand totals, which can include counts, sums, averages, and the like.

QUERY-BY-EXAMPLE TOOLS ***Query-by-example (QBE) tools*** help you graphically design the answer to a question. Suppose for example that Janielle Smith from Triple A Homes (*Customer Number* 4567) has called and ordered a delivery of concrete. Although she can't remember the name of the driver, she would like to have the driver that comes out the most often to deliver concrete to Triple A Homes. Solomon's task, from a customer relationship management point of view, is to go through all the orders and determine which employee most often delivers concrete to Triple A Homes. The task may seem simple considering that Solomon currently has very few orders in its database. However, can you imagine trying to answer that question if there were thousands of orders in Solomon's database? It would not be fun.

Fortunately, QBE tools can help you answer this question and perform many other queries in a matter of seconds. In Figure 3.7, you can see a QBE screen that formulates the answer to the question. When you perform a QBE, you (1) identify the files in which the needed information is located, (2) drag any necessary fields from the identified files to the QBE grid, and (3) specify selection criteria.

Figure 3.7

Using a Query-by-Example to Find Information



Our selection criteria

Customer Number	Employee Last Name	Employee First Name
4567	Evaraz	Antonio
4567	Robertson	John
4567	Robertson	John
4567	Robertson	John

For the names of employees who have delivered concrete to Triple A Homes, we identified the two files of *Order* and *Employee.* Second, we dragged *Customer Number* from the *Order* file to the QBE grid and dragged *Employee Last Name* and *Employee First Name* from the *Employee* file to the QBE grid. Finally, we specified in the Criteria box that we wanted to view only the orders for *Customer Number* 4567 (Triple A Homes). Access did the rest and provided the information in Figure 3.7.

QBEs rely heavily on the logical relationships within a database to find information. For example, *Order Number* 100004 has the *Customer Number* of 4567 (Triple A Homes). So, the QBE tool took the *Driver ID* from the *Order* file for that order and found a match in the *Employee* file. When it found a match, it presented the *Employee Last Name* and *Employee First Name* (John Robertson). Without the logical relationships being correctly defined, this QBE query would not have worked properly.

STRUCTURED QUERY LANGUAGE ***Structured query language (SQL)*** is a standardized fourth-generation query language found in most DBMSs. SQL performs the same function as QBE, except that you perform the query by creating a statement instead of pointing, clicking, and dragging. The basic form of an SQL statement is

SELECT . . . FROM . . . WHERE . . .

After the SELECT, you list the fields of information you want; after the FROM, you specify what logical relationships to use; and after the WHERE, you specify any selection criteria. Thoroughly introducing you to the syntax of building SQL statements is outside the scope of this text and would easily require almost 100 pages of material. But you should be aware that SQL does exist. If you're majoring in IT or MIS, you'll undoubtedly take a course in SQL.

APPLICATION GENERATION SUBSYSTEM

The ***application generation subsystem*** of a DBMS contains facilities to help you develop transaction-intensive applications. These types of applications usually require that you perform a detailed series of tasks to process a transaction. Application generation subsystem facilities include tools for creating visually appealing and easy-to-use data entry screens, programming languages specific to a particular DBMS, and interfaces to commonly used programming languages that are independent of any DBMS.

As with SQL, application generation facilities are most often used by IT specialists. As a knowledge worker, we recommend that you leave application generation to IT specialists as much as you can. You need to focus on views, report generators, and QBE tools. These will help you find information in a database and perform queries so you can start to build and use business intelligence.

DATA ADMINISTRATION SUBSYSTEM

The ***data administration subsystem*** of a DBMS helps you manage the overall database environment by providing facilities for backup and recovery, security management, query optimization, concurrency control, and change management. The data administration subsystem is most often used by a data administrator or database administrator—someone responsible for assuring that the database (and data warehouse) environment meets the entire information needs of an organization:

- *Backup and recovery facilities*—provide a way for you to (1) periodically back up information and (2) restart or recover a database and its information in case of a failure. A ***backup*** is simply a copy of the information stored on a computer.

INDUSTRY PERSPECTIVE

SERVING CITIZENS AND GOVERNMENT AGENCIES WITH A GIS DATABASE

Databases on the Web typically serve a variety of users. In the case of Pierce County, Washington, its Web-based geographic information system database serves both its citizenry and county agencies. In a geographic information system (GIS) format, users of all types can query information according to geographical location. The county expects to save almost $3 million in maintenance costs by implementing its new Web-based GIS database.

The county citizens will use the database, via a simple Web interface, to query such information as neighborhood crime statistics, property-tax information, property-survey reports, and voting information like polling information and locations. The GIS database serves the citizens of Pierce County by providing easy access to these types of vitally important information.

Government agencies such as law enforcement, natural resource management, land development, and utilities will also have access to the GIS information. They will use the database to access and analyze such information as crime patterns, maps of public facilities, and information on terrains and contours. Law-enforcement and emergency-services personnel can even use wireless handheld devices to access the GIS database while in the field.

All told, Pierce County's databases will serve its citizenry and government agencies with almost 10 terabytes of GIS information.[6]

Recovery is the process of reinstalling the backup information in the event the information was lost. In Chapter 7, we talk specifically about how to develop plans and strategies in the event of some sort of failure. We call this business continuity planning or disaster recovery planning.

- *Security management facilities*—allow you to control who has access to what information and what type of access those people have. Always remember **CRUD**—**C**reate, **R**ead, **U**pdate, and **D**elete. Identifying who can perform those functions on various database information is vitally important.
- *Query optimization facilities*—often take queries from users (in the form of SQL statements of QBEs) and restructure them to minimize response times. Basically, these facilities find the "shortest route" to the information you want so you don't have to.
- *Reorganization facilities*—continually maintain statistics concerning how the DBMS engine physically accesses information and reorganizes how information is physically stored. For example, if you frequently access a file by a specific order, the reorganization facilities may maintain the file in that presorted order by creating an index that maintains the sorted order in that file.
- *Concurrency control facilities*—ensure the validity of database updates when multiple users attempt to access and change the same information. Consider your school's online registration system. What if you and another student try to register for a class with only one seat remaining at exactly the same time? Who gets enrolled in the class? What happens to the person who does not get his or her desired class schedule?
- *Change management facilities*—allow you to assess the impact of proposed structural changes to a database. For example, if you decide to add a character identifier to a numeric truck number, you can use the change management facilities to see how many files would be affected.

All these—backup and recovery, security management, query optimization, reorganization, concurrency control, and change management—are vitally important facilities in any DBMS and thus any database environment. As a user and knowledge worker, you probably won't deal with these facilities specifically as far as setting them up and maintaining them is concerned. But how they're set up and maintained will affect what you can do. So knowing that they exist and understanding their purpose are important.

Data Warehouses and Data Mining

Suppose as a manager at Victoria's Secret, you wanted to know the total revenues generated from the sale of shoes last month. That's a simple query, which you could easily implement using either SQL or a QBE tool. But what if you wanted to know, "By actual versus budgeted, how many size 8 shoes in black did we sell last month in the southeast and southwest regions, compared with the same month over the last five years?" That task seems almost impossible, even with the aid of technology. If you were actually able to build a QBE query for it, you would probably bring the organization's operational database environment to its knees.

This example illustrates the two primary reasons so many organizations are opting to build data warehouses. First, while operational databases may have the needed information, the information is not organized in a way that lends itself to building business intelligence within the database or using various data manipulation tools. Second, if you could build such a query, your operational databases, which are probably already supporting the processing of hundreds of transactions per second, would seriously suffer in performance when you hit the Start button to perform the query.

To support such intriguing, necessary, and complex queries to create business intelligence, many organizations are building data warehouses and providing data-mining tools. A data warehouse is simply the next step (beyond databases) in the progression of building business intelligence. And data-mining tools are the tools you use to mine a data warehouse and extrapolate the business intelligence you need to make a decision, solve a problem, or capitalize on an opportunity to create a competitive advantage.

WHAT IS A DATA WAREHOUSE?

LEARNING OUTCOME 3

A ***data warehouse*** is a logical collection of information—gathered from many different operational databases—used to create business intelligence that supports business analysis activities and decision-making tasks. Sounds simple enough on the surface, but data warehouses represent a fundamentally different way of thinking about organizing and managing information in an organization. Consider these key features of a data warehouse, detailed in the sections that follow.

DATA WAREHOUSES ARE MULTIDIMENSIONAL In the relational database model, information is represented in a series of two-dimensional files or tables. Not so in a data warehouse—most data warehouses are multidimensional, meaning that they contain layers of columns and rows. For this reason, most data warehouses are really *multidimensional databases.* The layers in a data warehouse represent information according to different dimensions. This multidimensional representation of information is referred to as a *hypercube.*

In Figure 3.8 on the next page, you can see a hypercube that represents product information by product line and region (columns and rows), by year (the first layer), by customer segment (the second layer), and by the timing of advertising media (the third layer). Using this hypercube, you can easily ask, According to customer segment A, what

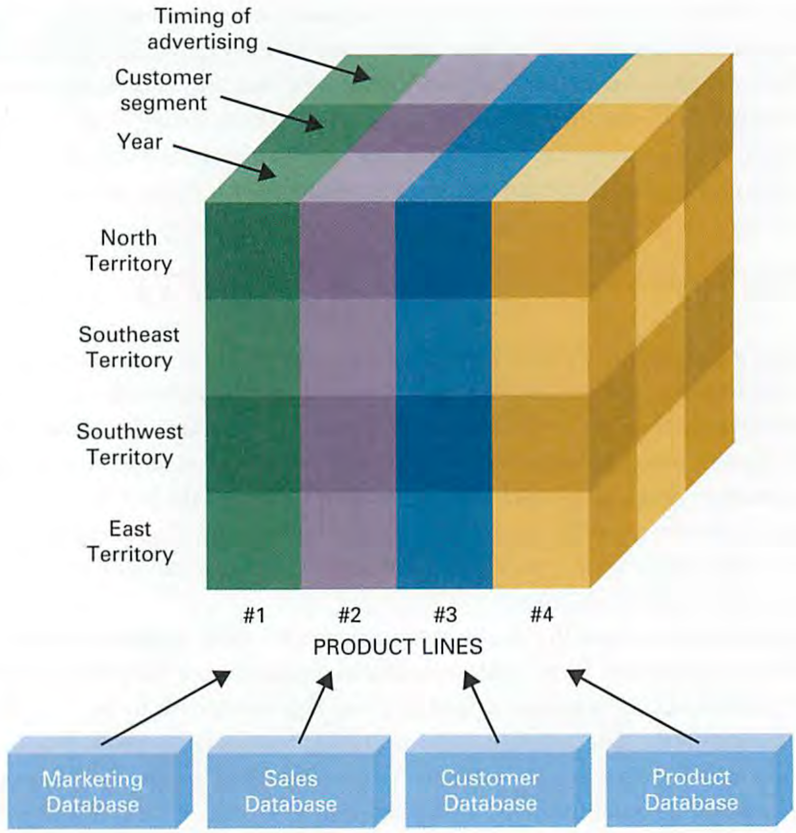


Figure 3.8

A Multidimensional Data Warehouse with Information from Multiple Operational Databases

percentage of total sales for product line 1 in the southwest territory occurred immediately after a radio advertising blitz? The information you would receive from that query constitutes business intelligence.

Any specific subcube within the larger hypercube can contain a variety of summarized information gathered from the various operational databases. For example, the forwardmost and top-left subcube contains information for the North territory, by year, for product line 1. So, it could contain totals, average, counts, and distributions summarizing in some way that information. Of course, what it contains is really up to you and your needs.

DATA WAREHOUSES SUPPORT DECISION MAKING, NOT TRANSACTION PROCESSING In an organization, most databases are transaction-oriented. That is, most databases support online transaction processing (OLTP) and, therefore, are operational databases. Data warehouses are not transaction-oriented: They exist to support decision-making tasks in your organization. Therefore, data warehouses support only online analytical processing (OLAP).

As we just stated, the subcubes within a data warehouse contain summarized information. So, while a data warehouse may contain the total sales for a year by product line, it does not contain a list of each individual sale to each individual customer for a given product line. Therefore, you simply cannot process transactions with a data warehouse. Instead, you process transactions with your operational databases and then use the information contained within the operational databases to build the summary information in a data warehouse.

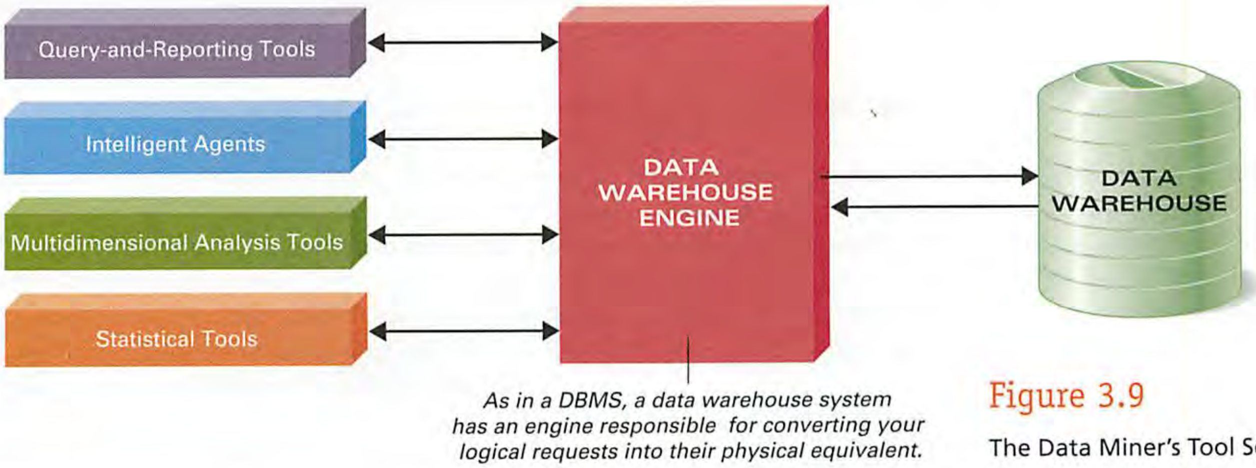


Figure 3.9
The Data Miner's Tool Set

WHAT ARE DATA-MINING TOOLS?

LEARNING OUTCOME 4

Data-mining tools are the software tools you use to query information in a data warehouse. These data-mining tools support the concept of OLAP—the manipulation of information to support decision-making tasks. Data-mining tools include query-and-reporting tools, intelligent agents, multidimensional analysis tools, and statistical tools (see Figure 3.9). Essentially, data-mining tools are to data warehouse users what data manipulation subsystem tools are to database users.

QUERY-AND-REPORTING TOOLS ***Query-and-reporting tools*** are similar to QBE tools, SQL, and report generators in the typical database environment. In fact, most data warehousing environments support simple and easy-to-use data manipulation subsystem tools such as QBE, SQL, and report generators. Most often, data warehouse users use these types of tools to generate simple queries and reports.

INTELLIGENT AGENTS Intelligent agents utilize various artificial intelligence tools such as neural networks and fuzzy logic to form the basis of "information discovery" and building business intelligence in OLAP. For example, Wall Street analyst Murray Riggiero uses OLAP software called Data/Logic, which incorporates neural networks to generate rules for his highly successful stock and bond trading system.[7] Other OLAP tools, such as Data Engine, incorporate fuzzy logic to analyze real-time technical processes.

Intelligent agents represent the growing convergence of various IT tools for working with information. Previously, intelligent agents were considered only within the context of artificial intelligence and were seldom thought to be a part of the data organizing and managing functions in an organization. Today, you can find intelligent agents being used not only for OLAP in a data warehouse environment but also for searching for information on the Web. In Chapter 4, we'll explore artificial intelligence techniques such as intelligent agents.

MULTIDIMENSIONAL ANALYSIS TOOLS ***Multidimensional analysis (MDA) tools*** are slice-and-dice techniques that allow you to view multidimensional information from different perspectives. For example, if you completed any of the recommended group projects for Chapter 1, you were using spreadsheet software to literally slice and dice the

GLOBAL PERSPECTIVE

CREDIT SCORING WITH SAS AT CANADA'S LAURENTIAN BANK

Data mining has an untold number of applications that can help all types of businesses make better decisions. In the financial services industry, Canada's Laurentian Bank has turned to SAS (www.sas.com), the leading provider of statistical tools, to create a credit scoring model to approve loans for snowmobiles, ATVs, boats, RVs, and motorcycles.

According to Sylvain Fortier, senior manager for Retail Risk Management at Laurentian Bank, "Our objective was to improve our use of the technological tools available for data exploitation and analysis that we needed to develop a scorecard internally. We wanted to develop a flexible system that would help us learn about our customers and business through the loan process." The system, called SAS Credit Scoring, develops a credit risk scorecard for each loan applicant based on a variety of information including socioeconomic data and determines the risk level of the applicant in the form of a score.

In just a few shorts months, the bank realized a 33 percent return on investment and expects to reduce losses on automotive loans by 8 percent.[8]

provided information. Within the context of a data warehouse, we refer to this process as "turning the cube." That is, you're essentially turning the cube to view information from different perspectives.

This turning of the cube allows you to quickly see information in different subcubes. If you refer back to the data warehouse in Figure 3.8, you'll notice that information by customer segment and timing of advertising is actually hidden. Using MDA tools, you can easily bring this to the front of the data warehouse for viewing. What you've essentially done is to slice the cube vertically by layer and bring some of the background layers to the front. As you do this, the values of the information are not affected.

STATISTICAL TOOLS Statistical tools help you apply various mathematical models to the information stored in a data warehouse to discover new information. For example, you can perform a time-series analysis to project future trends. You can also perform a regression analysis to determine the effect of one variable on another.

Sega of America, one of the largest publishers of video games, uses a data warehouse and statistical tools to effectively distribute its advertising budget of more than $50 million a year.[9,10] With its data warehouse, product line specialists and marketing strategists "drill" into trends of each retail store chain. Their goal is to find buying trends that will help them better determine which advertising strategies are working best (and at what time of the year) and how to reallocate advertising resources by media, territory, and time. Sega definitely benefits from its data warehouse, and so do retailers such as Toys "Я" Us, Wal-Mart, and Sears—all good examples of customer relationship management through technology.

To learn more about today's best data warehousing and data-mining tools, visit the Web site that supports this text at www.mhhe.com/haag.

DATA MARTS: SMALLER DATA WAREHOUSES

Data warehouses are often perceived as organizationwide, containing summaries of all the information that an organization tracks. However, some people need access to only a portion of that data warehouse information as opposed to all of it. In this case, an organization can create one or more data marts. A ***data mart*** is a subset of a data

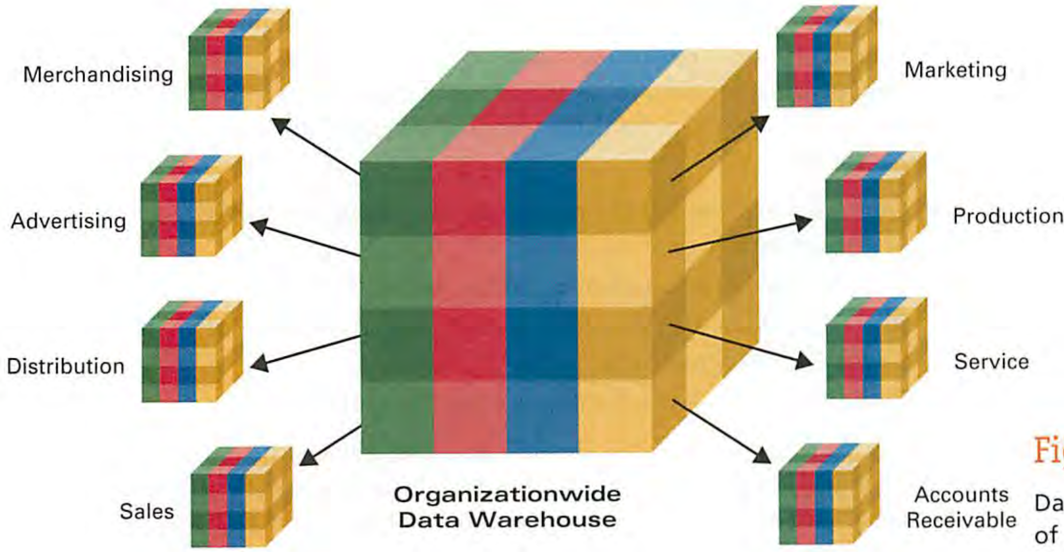


Figure 3.10

Data Marts Are Subsets of Data Warehouses

warehouse in which only a focused portion of the data warehouse information is kept (see Figure 3.10).

Lands' End first created an organizationwide data warehouse for everyone to use, but soon found out that there can be "too much of a good thing."[11] In fact, many Lands' End employees wouldn't use the data warehouse because it was simply too big, too complicated, and included information they didn't need access to. So, Lands' End created several smaller data marts. For example, Lands' End created a data mart just for the merchandising department. That data mart contains only merchandising-specific information and not any information, for instance, that would be unique to the finance department.

Because of the smaller, more manageable data marts, knowledge workers at Lands' End are making better use of information. If some of your employees don't need access to organizationwide data warehouse information, consider building a smaller data mart for their particular needs.

If you do choose to build smaller data marts for your employees, the data-mining tools are the same. That is, data marts support the use of query-and-reporting tools, intelligent agents, multidimensional analysis tools, and statistical tools. This yields efficiency in an organization with respect to training. Once you've trained your employees to use any or all data-mining tools, they can apply them to an organizationwide data warehouse or smaller data marts.

DATA MINING AS A CAREER OPPORTUNITY

Data mining represents a substantial career opportunity for you, no matter what your career choice. In the business world, you'll face numerous situations in which you need business intelligence to make the right and most effective decisions.

Fortunately, you don't have to be an IT expert to perform data mining. As you'll learn in *Extended Learning Module D (Decision Analysis with Spreadsheet Software)*, you can actually use a spreadsheet tool such as Microsoft Excel to build a three-dimensional cube similar to the one in Figure 3.8. You can then use Excel's other decision support features to build a graph, perform a regression analysis, and "turn the cube" by bringing new layers of information forward. You can do the same with Microsoft Access, by building a three-dimensional cube (i.e., data warehouse) of information stored in a database. We definitely recommend that you learn to use these tools and then note your proficiency in your e-portfolio under "Technology Skills."

Beyond personal productivity tools, you should consider learning how to use some data-mining tools specific to the data warehouse environment. Some of the more popular ones include:

- Query and Analysis and Enterprise Analytic tools in Business Objects (www.businessobjects.com)
- Business Intelligence and Information Access tools in SAS (www.sas.com)
- ReportNet, PowerPlay, Visualizer, NoticeCast, and DecisionStream tools in Cognos (www.cognos.com)
- PowerAnalyzer tools in Informatica (www.informatica.com)

There are many, many others. You should have a look at your school's catalog of courses in data mining—you may find them offered in the technology department, statistics department, and other departments. We recommend that at the very least you become acquainted with the following: SAS (the leading vendor in statistical software), Cognos (the leading vendor in data warehousing and data-mining tools), and Informatica (the second-leading vendor in data warehousing and data-mining tools).

IMPORTANT CONSIDERATIONS IN USING A DATA WAREHOUSE

As is true with all types of technology, you can't simply implement a data warehouse and use data-mining tools just because they're a "hot" set of technologies and expect automatically to increase your efficiency and effectiveness. Always let your business needs drive your technology decisions. You have to need the technology and the technology has to fit your needs. With respect to data warehouse and data-mining tools, consider your answers to the following questions.

1. **Do you need a data warehouse?** Although great IT tools, they are not necessarily the best technologies for all businesses because (1) they are expensive, (2) they may not be necessary since some businesses can easily extract all the business intelligence they need from databases, and (3) they require extensive and often expensive support.
2. **Do all your employees need an entire data warehouse?** If not, consider building data marts.
3. **How up-to-date must the information be?** To create a data warehouse, you take "snapshots" of database information and load it into a data warehouse. If crucial information changes every second, this may not be possible.
4. **What data-mining tools do you need?** User needs should always drive the answer to this question. Whichever you choose, training will be key. If your users can fully exploit all the features of their chosen data-mining tools, your entire organization will reap the benefits.

Business Intelligence Revisited

LEARNING OUTCOME 5

Business intelligence is one of the hottest topics and markets today. The entire BI market (both hardware and software) is in the range of $50 billion annually with double-digit growth expected for the next several years. When *InformationWeek* asked 300 business technology managers about their immediate project plans, 44 percent identified data warehouses and 43 percent identified data-mining tools.[12]

"We were desperate to get good information quickly," explained Joel Taylor, director of IS for FiberMark North America, a manufacturer of specialty packaging and paper, which could not easily retrieve business intelligence from its expensive transaction processing systems. To address this problem, Taylor spent less than $75,000 on QlikView—BI software from QlikTech (www.qliktech.com). FiberMark's 29 salespeople, who previously printed 1,000-page monthly sales reports, now print on the average only four pages and it's exactly the four pages of business intelligence they need. The $75,000 investment paid for itself in nine months in saved paper and related costs alone. Of course, the real benefit is that salespeople and executives can get the specific, up-to-date BI they want anytime they want it.[13]

BI

FiberMark is one of thousands of success stories related to business intelligence. (Of course, an equal number have probably not met with as much success, sometimes because they did not adequately identify which technology they needed.) The objective of BI is to improve the timeliness and quality of the input for decision making by helping knowledge workers to understand the

- Capabilities available in the organization.
- State of the art, trends, and future directions in the markets.
- Technological, demographic, economic, political, social, and regulatory environments in which the organization competes (remember Porter's Five Forces Model).
- Actions of competitors and the implications of these actions.[14]

As illustrated in Figure 3.11, BI encompasses both internal and external information. Some business people treat *competitive intelligence* as a specialized branch of business intelligence. ***Competitive intelligence (CI)*** is business intelligence focused on the external competitive environment. There is even an organization for people who specialize in competitive intelligence called the Society for Competitive Intelligence Professionals (SCIP, at www.scip.org).

A survey of the strategic uses of business intelligence by the Gartner Group found that such uses were ranked by firms in the following order of importance:

1. Corporate performance management.
2. Optimizing customer relations, monitoring business activity, and traditional decision support.
3. Packaged stand-alone BI applications for specific operations or strategies.
4. Management reporting of business intelligence.[15]

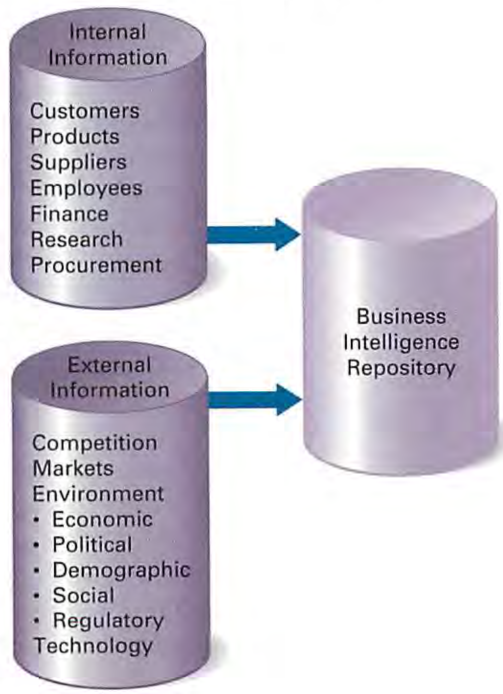
As we have stressed, one of the primary purposes of BI is to improve the timeliness and quality of input to the decision process. Companies with well-designed BI systems available to their managers find that their managers make better decisions on a variety of business issues. Higher quality managerial decision making lets companies gain an advantage over their competitors who operate without the benefit of BI systems for their managers to use. BI systems provide managers with actionable information and knowledge:

- at the right time
- in the right location
- in the right form[16]

One particularly interesting feature of many BI software packages is a *digital dashboard.* A ***digital dashboard*** displays key information

Figure 3.11

Building Business Intelligence



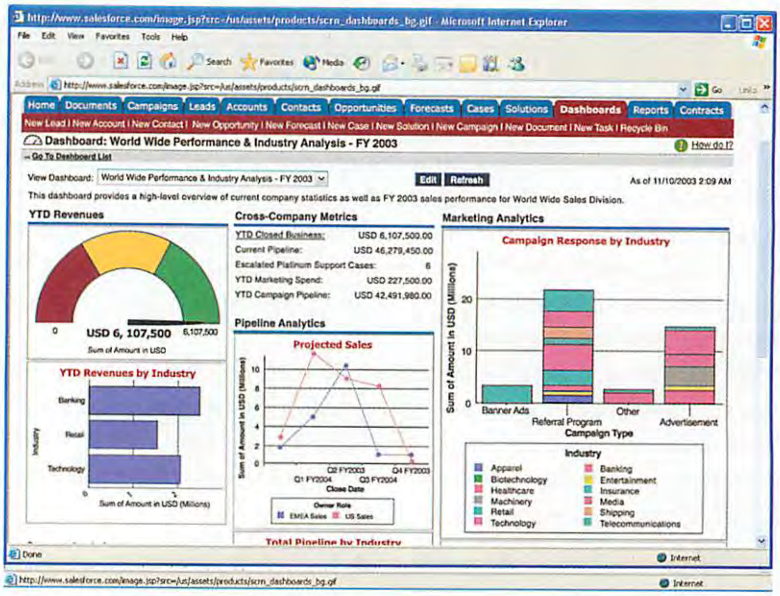


Figure 3.12

Sample Digital Dashboard

gathered from several sources on a computer screen in a format tailored to the needs and wants of an individual knowledge worker (see Figure 3.12). Digital dashboards can provide up-to-the-minute snapshots of any type of information and can often help you identify trends that may represent opportunities or that may be problems.

Information Ownership

LEARNING OUTCOME 6

Your organization will be successful, in part, because of your ability to organize and manage information in a way that best moves the organization toward its goals. As we close this chapter, let's look at the notion of *information ownership* and what it means.

STRATEGIC MANAGEMENT SUPPORT

As we discussed in Chapter 2, an organization's *IT culture* defines how the organization structures and views both the IT department and information responsibilities within it. Starting with vitally important roles such as CPO (chief privacy officer), CSO (chief security officer), CIO (chief information officer), and CTO (chief technology officer), organizations work down through the organization defining additional positions of responsibility with respect to the management of information and IT. Two of those positions are data administration and database administration.

Data administration is the function in an organization that plans for, oversees the development of, and monitors the information resource. This function must be completely in tune with the strategic direction of the organization to assure that all information requirements can be and are being met. ***Database administration*** is the function in an organization that is responsible for the more technical and operational aspects of managing the information contained in organizational information repositories (databases, data warehouses, and data marts). Database administration functions include defining and organizing database structures and contents, developing security procedures (in concert with the CSO), and approving and monitoring the development of database and database applications.