INDUSTRY PERSPECTIVE

SUPPORTING HEALTH CARE WITH ELECTRONIC MEDICAL RECORDS

In 2008, employers are estimated to pay 58 percent and employees 81 percent more for health care than they did in 2002. Those are percentage increases we would like to have in our investment portfolios, not in an expense category such as health care. In 2007, Intel spent approximately $1 billion on employee health care.

To be sure, health care costs are rising. In spite of low-cost generic prescription medications and the like, it's simply becoming more and more expensive to provide quality health care. But there is another factor weighing on rising health care costs and it's technology related. The costs associated with storing, organizing, and managing health care information are skyrocketing at an astronomical pace. Even more so, there are no well-defined standards for what an electronic medical record should look like. As a result, almost every health care provider and insurer stores medical records in different formats, making it almost impossible to share information from company to company.

This has everyone scared—so much so, that the federal government has created a panel, the Health Information Technology Standards Panel, and charged it with selecting a single standard for the health care industry to adopt. But therein lies another problem. To select the electronic standard being used by only one or a few health care companies would mean that all others have to change their systems, and that would be expensive.

Nonetheless, the standardized electronic medical record is an imperative. With it in place, health care costs can go down because of the ease, speed, and efficiency with which organizations will be able to organize and manage information in a database.[17]

In large organizations, both of these administrative functions are usually handled by steering committees rather than by a single individual. These steering committees are responsible for their respective functions and for reporting to the CIO.

SHARING INFORMATION WITH RESPONSIBILITY

Information sharing in your organization means that anyone—regardless of title or department—can access and use whatever information he or she needs. But information sharing brings to light an important question: Does anyone in your organization *own* the information? In other words, if everyone shares the information, who is ultimately responsible for providing the information and assuring the quality of the information? Information ownership is a key consideration in today's information-based business environment. Someone must accept full responsibility for providing specific pieces of information and ensuring the quality of that information. If you find the wrong information is stored in the organization's data warehouse, you must be able to determine the source of the problem and whose responsibility it is.

INFORMATION CLEANLINESS

Information "cleanliness" (an aspect of information ownership) is an important topic today and will be for many years. Have you ever received the same piece of advertising mail (snail mail, that is) multiple times from the same company on the same day? Many people have, and it's an example of "unclean" information. The reason may be your name may appear twice in a database, once with your middle initial and once without it. Or your name may appear twice in a database with two different spellings of your last name.

In all popular business-oriented DBMSs, such as Oracle, you can find utilities to help you "clean" your information. In the case of having your information twice in a database with two different spellings of your last name, the utility would probably determine that the two records actually belong to the same person (you) because of the identical nature of other associated information such as your address and phone number. Always remember GIGO—garbage in, garbage out (from Chapter 1). If bad information—such as duplicate records for the same customer—goes into the decision-making process, you can rest assured that the decision outcome will not be optimal.

Summary: Student Learning Outcomes Revisited

1. **List and describe the key characteristics of a relational database.** The ***relational database*** model uses a series of logically related two-dimensional tables or files to store information in the form of a database. Key characteristics include
   - A collection of information—Composed of many files or tables of information that are related to each other
   - Contain logical structures—You care only about the logical information and not about how it's physically stored or where it's physically located
   - Have logical ties among the information—All the files in a database are related in that some ***primary keys*** of certain files appear as ***foreign keys*** in others
   - Possess built-in ***integrity constraints***—When creating the data dictionary for a database, you can specify rules by which the information must be entered (e.g., not blank, etc.)
2. **Define the five software components of a database management system.** The five software components of a database management system include
   - ***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
   - ***Data definition subsystem***—Helps you create and maintain the data dictionary and define the structure of the files in a database
   - ***Data manipulation subsystem***—Helps you add, change, and delete information in a database and query it for valuable information
   - ***Application generation subsystem***—Contains facilities to help you develop transaction-intensive applications
   - ***Data administration subsystem***—Helps you manage the overall database environment by providing facilities for backup and recovery, security management, query optimization, concurrency control, and change management
3. **List and describe the key characteristics of a data warehouse.** The key characteristics of a data warehouse include
   - Multidimensional—While databases store information in two-dimensional tables, data warehouses include layers to represent information according to different dimensions
   - Support decision making—Data warehouses, because they contain summarized information, support business activities and decision-making tasks, not transaction processing
4. **Define the four major types of data-mining tools in a data warehouse environment.** The four major types of data-mining tools in a data warehouse environment include
   - ***Query-and-reporting tools***—Similar to QBE tools, SQL, and report generators in the typical database environment
   - ***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

- ***Multidimensional analysis (MDA) tools***—Slice-and-dice techniques that allow you to view multidimensional information from different perspectives
- **Statistical tools**—Help you apply various mathematical models to the information stored in a data warehouse to discover new information

5. **Describe business intelligence and its role in an organization.** ***Business intelligence*** is collective information—about your customers, your competitors, your 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 is much more than just a list of your products or to whom you've sold them. It might combine your product information with your advertising strategy information and customer demographics, for instance, to help you determine the effectiveness of various advertising media on demographic groups segmented by location.
6. **List key considerations in information ownership in an organization.** Key considerations in information ownership in an organization include:
   - Strategic management support
   - The sharing of information with responsibility
   - Information cleanliness

CLOSING CASE STUDY ONE

BEN & JERRY'S, BIGELOW TEAS, AND BUSINESS INTELLIGENCE

Organizations want information. Organizations need information. However, information must be in an organized format that supports the creation of business intelligence. Otherwise, according to Rebecca Wettemann, vice president of Research at Nucleus Research, "It's like having a bank account with millions of dollars in it but no ATM card. If you can't get it [business intelligence] and can't make it work for you, then it is not really useful."

In support of creating and using business intelligence, companies have focused much of their spending efforts on business intelligence software and data-mining tools. According to a Merrill Lynch survey in 2003, business intelligence software and data-mining tools were at the top of the technology spending list of CIOs. And according to A. G. Edwards, the market for that type of software is expected to grow from $4.7 billion in 2003 to $7.5 billion in 2006.

Consider two companies—Ben & Jerry's and Bigelow Teas—and their approach to creating and using business intelligence.

BEN & JERRY'S

Ben & Jerry's, located in Waterbury, Vermont, produces 190,000 pints of ice cream and frozen yogurt daily and ships to over 50,000 grocery stores in the United States and 12 other countries. Every single pint is meticulously tracked, first by being entered into an Oracle database. With that information carefully organized, Ben & Jerry's uses a sophisticated data-mining tool set from a company called Business Objects.

For example, the sales people can easily monitor sales to determine how much ground Cherry Garcia Frozen Yogurt is gaining on Cherry Garcia Ice Cream, its number one selling product. The consumer affairs staff can even correlate each of the several hundred calls and e-mails received each week to the exact pint of ice cream. If complaints are consistent concerning a specific batch, the consumer affairs staff can drill down to the supplier who provided the ingredients such as milk or eggs.

In one particular instance, Ben & Jerry's received a large number of complaints that its Cherry Garcia Ice Cream didn't have enough cherries. The complaints were coming in from all over the country, so it wasn't a regional problem. Employees continued drilling through business intelligence with Business Objects and determined that the manufacturing process (from the supplies of raw materials to the mixing) was satisfactory and had no anomalies. Eventually the problem was determined to be that the ice cream box for Cherry Garcia Ice Cream had on it a photo of frozen yogurt, a product with more cherries than the ice cream. Simply changing the photo on the box solved the problem.

BIGELOW TEAS

Bigelow Teas provides over 50 varieties of flavored, traditional, iced, decaffeinated, and herbal teas. Over the past 50 years, Bigelow Teas has relied on business intelligence to determine the success of each individual tea, and today is no different.

Although it may not seem like it, bringing a new tea to the market is a risky endeavor. It could fail in every way or it could simply cannibalize the sales of an existing tea, neither of which makes business sense. Employees at Bigelow Teas pore over consumer, sales, marketing, and finance business intelligence to ensure that they are making the right decisions in all aspects of the business.

To help facilitate the creation and use of business intelligence, Bigelow Teas turned to the Andrews Consulting Group and BusinessObjects. Prior to using BusinessObjects, Bigelow employees had a difficult time finding and using the right information. As Melanie Dower, project leader at Bigelow Teas, describes it, "Our existing end-user reporting tool wasn't user-friendly, so users simply weren't using it. Most users were unable to create their own reports, so we looked for a solution that offered self-serve business intelligence (BI) to free up IT resources." BusinessObjects is both easy to learn and easy to use because it looks like Microsoft Excel. Explains Dower, "Enterprise 6 [of BusinessObjects] looks and feels like Microsoft Excel, which speeds up the learning curve for our end users."

With BusinessObjects, Bigelow employees can access and view business intelligence in real time, more accurately predict sales forecasts based on shipment levels, identify where to increase sales efforts before it's too late, and even compare current consumer, sales, and marketing information with similar types of information up to five years old.

Gourmet coffees exploded onto the consumer market about seven years ago and gourmet teas quickly followed. Bigelow Teas is riding this wave of success with success of its own because of its use of business intelligence.[18,19,20]

Questions

1. Ben & Jerry's tracks a wealth of information on each pint of ice cream and frozen yogurt. If you were to design Ben & Jerry's data warehouse, what dimensions of information would you include? As you develop your list of dimensions, consider every facet of Ben & Jerry's business operations, from supply chain management to retail store monitoring.
2. Databases are the underlying technology that allows Ben & Jerry's to track ice cream and frozen yogurt information. Based on your knowledge of databases, what sort of tables or files of information would Ben & Jerry's need in its database? What would be the primary keys for each of those? What would be the foreign keys among those to create the necessary relationships?
3. According to the discussion of Bigelow Teas, part of the success of BusinessObjects comes from its look and feel being similar to Microsoft Excel. Why do you believe this is true? When introducing employees to enterprisewide BI tools such as BusinessObjects, why is it an advantage to have the BI tool look like and work like personal productivity software tools? Why was a similar look and feel to spreadsheet software more important than word processing or presentation software?
4. How could Bigelow Teas open up its business intelligence information to its suppliers and resellers? What benefits would Bigelow Teas gain by keeping its suppliers and resellers more informed with business intelligence? What types of business intelligence would Bigelow Teas want to exclude its suppliers and resellers from seeing? Why?
5. Neil Hastie, CIO at TruServe Corporation, once described most decision making in all types of businesses as "a lot of by-guess and by-golly, a lot of by-gut, and a whole lot of paper reports." That statement is not kind to managers in general or to IT specialists charged with providing the right people with the right technology to make the right decisions. What's the key to turning Neil's statement into a positive one? Is it training? It is providing timely information access? Is it providing everyone with a wide assortment of data-mining tools? Other solutions? Perhaps it's a combination of several answers?

CLOSING CASE STUDY TWO

MINING DINING DATA

Restaurants and fast-food chains rely heavily on business intelligence to make important decisions. Casinos do as well. Some of the leading restaurants, fast-food chains, and casinos using data warehouses include AFC Enterprises (operator and franchiser of more than 3,300 Church's Chicken, Popeye's Chicken and Biscuits, Seattle Coffee Company, Cinnabon, and Torrefazione outlets worldwide); Red Robin International (a 170-unit casual-dining chain); Harrah's Entertainment (owner of 26 U.U. casinos); Pizzeria Uno; and Einstein/Noah Bagel (operator of 428 Einstein's and 111 Noah's New York Bagel stores).

AFC ENTERPRISES

AFC Enterprises cultivates a loyal clientele by slicing and dicing its data warehouse to strategically configure promotions and tailor menus to suit local preferences. AFC's data warehouse helps it better understand its core customers and maximize its overall profitability. AFC tracks customer-specific information from name and address to order history and frequency of visits. This enables AFC to determine exactly which customers are likely to respond to a given promotion on a given day of the week.

AFC also uses its data warehouse to anticipate and manipulate customer behavior. For example, AFC can use its data warehouse to determine that coffee is added to the tab 65 percent of the time when a particular dessert is ordered and 85 percent of the time when that dessert is offered as a promotional item. Knowing that, AFC can run more promotions for certain desserts figuring that customers will respond by ordering more desserts and especially more coffee (coffee is a high-margin item in the restaurant business).

RED ROBIN INTERNATIONAL

Red Robin's terabyte-size data warehouse tracks hundreds of thousands of point-of-sale (POS) transactions, involving millions of menu items and more than 1.5 million invoices. As Howard Jenkins, Red Robin's vice president of Information Systems, explains it, "With data mining in place, we can ask ourselves, 'If we put the items with high margins in the middle of the menu, do we sell more versus putting it at the top or bottom, [and if so], to whom and where?' We can also tell if something cannibalizes the sale of other items and can give the marketing department an almost instant picture of how promotions are being sold and used."

The placement of items on a menu is strategic business, just as the placement of promotional items in a grocery store can mean increased sales for one item and reduced sales for another. The job of finding the right mix is definitely suited to mining a data warehouse.

Using Cognos Business Intelligence, Red Robin now has measurable results of promotion and menu changes, makes better and more timely decisions, and has realized seven-figure savings in operational costs.

HARRAH'S ENTERTAINMENT

Harrah's Entertainment uses its data warehouse to make decisions for its highly successful Total Gold customer recognition program. Depending on their spending records, Total Gold members can receive free vouchers for dining, entertainment, and sleeping accommodations. Knowing which rewards to give to which customers is key.

John Boushy, senior vice president of Entertainment and Technology for Harrah's, says, "We can determine what adds value to each customer and provide that value at the right time." Dining vouchers or free tickets for shows are awarded to day visitors, not sleeping accommodations. Customers who consistently visit a particular restaurant and order higher-end foods receive free dinners and cocktails, not vouchers for free (and cheaper) breakfasts.

PIZZERIA UNO

Pizzeria Uno uses its data warehouse to apply the 80/20 rule. That is, it can determine which 20 percent of its customers contribute to 80 percent of its sales and adjust menus and promotions to suit top patron preferences. These changes can often lead to converting some of the other 80 percent of Pizzeria Uno's customers to the more profitable 20 percent.

EINSTEIN/NOAH BAGEL

Einstein/Noah Bagel uses its data warehouse in real time to maximize cross-selling opportunities. For example, if data warehouse information reveals that a manager in a given store might be missing a cross-selling opportunity on a particular day, an e-mail is automatically sent out to alert managers to the opportunity. Salespeople can then respond by offering the cross- selling opportunity ("How about a cup of hot chocolate with that bagel since it's so cold outside?") to the next customer.[21,22,23,24]

Questions

1. Consider the issue of timely information with respect to the businesses discussed in the case. Which of the businesses must have the most up-to-date information in its data warehouse? Which business can have the most out-of-date information in its data warehouse and still be effective? Rank the five businesses discussed with a 1 for the one that needs the most up-to-date information and a 5 for the one that is least sensitive to timeliness of information. Be prepared to justify your rankings.
2. Harrah's Entertainment tracks a wealth of information concerning customer spending habits. If you were to design Harrah's Entertainment's data warehouse, what dimensions of information would you include? As you develop your list of dimensions, consider every facet of Harrah's business operations, including hotels, restaurants, and gaming casinos.
3. AFC Enterprises includes information in its data warehouse such as customer name and address. Where does it (or could it) gather such information? Think carefully about this, because customers seldom provide their names and addresses when ordering fast food at a Church's or Popeye's. Is AFC gathering information in an ethical fashion? Why or why not?
4. Visit a local grocery store and walk down the breakfast cereal aisle. You should notice something very specific about the positioning of the various breakfast cereals. What is it? On the basis of what information do you think grocery stores determine cereal placement? Could they have determined that information from a data warehouse or from some other source? If another source, what might that source be?
5. Suppose you're opening a pizza parlor in the town where you live. It will be a "take and bake" pizza parlor in which you make pizzas for customers but do not cook them. Customers buy the pizzas uncooked and take them home for baking. You will have no predefined pizza types but will make each pizza to the customer's specifications. What sort of data warehouse would you need to predict the use of toppings by time of day and by day of the week? What would your dimensions of information be? If you wanted to increase the requests for a new topping (such as mandarin oranges), what information would you hope to find in your data warehouse that would enable you to do so?

Key Terms and Concepts

Application generation subsystem, 122
Backup, 123
Business intelligence (BI), 112
Competitive intelligence (CI), 131
Data administration, 132
Data administration subsystem, 123
Database, 114
Database administration, 132
Database management system (DBMS), 118
Data definition subsystem, 119
Data dictionary, 114
Data manipulation subsystem, 120
Data mart, 128
Data-mining tool, 127
Data warehouse, 125
DBMS engine, 118
Digital dashboard, 131
Foreign key, 116
Integrity constraint, 117
Logical view, 118
Multidimensional analysis (MDA) tool, 127
Online analytical processing (OLAP), 112
Online transaction processing (OLTP), 112
Operational database, 112
Physical view, 118
Primary key, 116
Query-and-reporting tool, 127
Query-by-example (QBE) tool, 122
Recovery, 124
Relation, 114
Relational database, 114
Report generator, 121
Structured query language (SQL), 122
View, 121

Short-Answer Questions

1. What is business intelligence? Why is it more than just information?
2. What is online transaction processing (OLTP)?
3. What is online analytical processing (OLAP)?
4. What is the most popular database model?
5. How are primary and foreign keys different?
6. What are the five important software components of a database management system?
7. How are QBE tools and SQL similar? How are they different?
8. What is a data warehouse? How does it differ from a database?
9. What are the four major types of data-mining tools?
10. What is a data mart? How is it similar to a data warehouse?

Assignments and Exercises

1. **FINDING "HACKED" DATABASES** *The Happy Hacker* (www.happyhacker.org/news/newsfeed.shtml) is a Web site devoted to "hacking"—breaking into computer systems. When people hack into a system, they often go after information in databases. There, they can find credit card information and other private and sensitive information. Sometimes, they can even find designs of yet-to-be-released products and other strategic information about a company. Connect to *The Happy Hacker* Web site and find an article that discusses a database that was hacked. Prepare a short report for your class detailing the incident.
2. **DEFINING QUERIES FOR A VIDEO RENTAL STORE** Consider your local video rental store. It certainly has an operational database to support its online transaction processing (OLTP). The operational database supports such things as adding new customers, renting videos (obviously), ordering videos, and a host of other activities. Now, assume that the video rental store also uses that same database for online analytical processing (OLAP) in the form of creating queries to extract meaningful information. If you were the manager of the video rental store, what kinds of queries would you build? What answers are you hoping to find?
3. **CREATING A QUERY** On the Web site that supports this text (www.mhhe.com/haag, choose Chapter 3 and then Solomon Enterprises), we've provided the database (in Microsoft Access) we illustrated in this chapter. Connect to the text's Web site and download that database. Now, create three queries using the QBE tool. The first one should extract information from only one file (your choice). The second one should extract information found in at least two files. The third should include some sort of selection criteria. How easy or difficult was it to perform these three queries? Would you say that a DBMS is just as easy to use as something like word processing or spreadsheet software? Why or why not? (By the way, *Extended Learning Module J* takes you through the step-by-step process of creating a query in Access.)
4. **CAREER OPPORTUNITIES IN YOUR MAJOR** Knowledge workers throughout the business world are building their own desktop databases (often called end-user databases or knowledge worker databases). To do so, they must understand both how to design a database and how to use a desktop DBMS such as Microsoft Access or FileMaker (made by FileMaker). The ability to design a database and use a desktop DBMS offers you a great career advantage. Research your chosen major by looking at job postings (the Web is the best place to start). How many of those jobs want you to have some database knowledge? Do they list a specific DBMS package? What's your take—should you expand your education and

learn more about databases and DBMSs? Why or why not?

5. **SALARIES FOR DATABASE ADMINISTRATORS** Database administrators (DBAs) are among the highest paid professionals in the information technology field. Many people work for 10 to 20 years to get a promotion to DBA. Connect to Monster.com (www.monster.com) or another job database of your choice and search for DBA job openings. As you do, select all locations and job categories and then use "dba" as the keyword search criteria. How many DBA job postings did you find? In what industries were some of the DBA job openings? Read through a couple of the job postings. What was the listed salary range (if any)? What sort of qualifications were listed?
6. **HOW UP-TO-DATE SHOULD DATA WAREHOUSE INFORMATION BE?** Information timeliness is a must in a data warehouse—old and obsolete information leads to poor decision making. Below is a list of decision-making processes that people go through for different business environments. For each, specify whether the information in the data warehouse should be updated monthly, weekly, daily, or by the minute. Be prepared to justify your decision.
   a. To adjust classes sizes in a university registration environment.
   b. To alert people to changes in weather conditions.
   c. To predict scores of professional football games.
   d. To adjust radio advertisements in light of demographic changes.
   e. To monitor the success of a new product line in the clothing retail industry.
   f. To adjust production levels of food in a cafeteria.
   g. To switch jobs to various printers in a network.
   h. To adjust CD rates in a bank.
   i. To adjust forecasted demands of tires in an auto parts store.

Discussion Questions

1. Databases and data warehouses clearly make it easier for people to access all kinds of information. This will lead to great debates in the area of privacy. Should organizations be left to police themselves with respect to providing access to information or should the government impose privacy legislation? Answer this question with respect to (1) customer information shared by organizations, (2) employee information shared within a specific organization, and (3) business information available to customers.
2. Business intelligence sounds like a fancy term with a lot of competitive advantage potentially rolled into it. What sort of business intelligence does your school need? Specifically, what business intelligence would it need to predict enrollments in the coming years? What business intelligence would it need to determine what curriculums to offer? Do you think your school gathers and uses this kind of business intelligence? Why or why not?
3. Consider your school's registration database that enforces the following integrity constraint: to enroll in a given class, the student must have completed or currently be enrolled in the listed prerequisite (if any). Your school, in fact, probably does have that integrity constraint in place. How can you get around that integrity constraint and enroll in a class for which you are not taking nor have completed the prerequisite? Is this an instance of when you should be able to override an integrity constraint? What are the downsides to being able to do so?
4. In this chapter, we listed the five important software components of a DBMS: the DBMS engine, the data definition, data manipulation, application generation, and data administration subsystems. Which of those are most and least important to users of a database? Which of those are most and least important to technology specialists who develop data applications? Which of those

are most and least important to the chief information officer (CIO)? For each of your responses, provide justification.

5. Some people used to believe that data warehouses would quickly replace databases for both online transaction processing (OLTP) and online analytical processing (OLAP). Of course, they were wrong. Why can data warehouses not replace databases and become "operational data warehouses"? How radically would data warehouses (and their data-mining tools) have to change to become a viable replacement for databases? Would they then essentially become databases that simply supported OLAP? Why or why not?
6. Consider that you work in the human resources management department of a local business and that many of your friends work there. Although you don't personally generate payroll checks, you still have the ability to look up anyone's pay. Would you check on your friends to see if they're earning more money than you? For that matter, would you look up their pay just out of simple curiosity, knowing that you would never do anything with the information or share it with anyone else? Why or why not? People working at the Internal Revenue Service (IRS) were caught just curiously looking up the reported incomes of movie stars and other high-profile public figures. Is this acceptable? Why or why not?
7. In spite of the need for "clean" information, many organizations have databases with duplicate records for you. You've probably experienced the consequences of this by receiving two identical pieces of junk mail from the same company. One record in the database may have your middle initial while the other doesn't, or there is some other type of minor discrepancy. Why would some organizations intentionally *not* go through a process of cleaning their database information?

CHAPTER PROJECTS

Group Projects

- Building Value Chains: Helping Customers Define Value (p. 467)
- Using Relational Technology to Track Projects: Foothills Construction (p. 469)
- Building a Web Database System: Web-Based Classified System (p. 475)
- Creating a Database Management System: Mountain Bike Rentals (p. 484)

e-Commerce Projects

- Searching Job Databases (p. 499)
- Exploring Google Earth (p. 492)
- Financial Aid Resources (p. 493)
- Consumer Information (p. 488)
- Demographics (p. 490)
- Bureau of Labor and Statistics (p. 490)
- Best of Computer Resources and Statistics (p. 488)
- Global Statistics and Resources (p. 494)

EXTENDED LEARNING MODULE C

DESIGNING DATABASES AND ENTITY-RELATIONSHIP DIAGRAMMING

Student Learning Outcomes

1. Identify how databases and spreadsheets are both similar and different.
2. List and describe the four steps in designing and building a relational database.
3. Define the concepts of entity class, instance, primary key, and foreign key.
4. Given a small operating environment, build an entity-relationship (E-R) diagram.
5. List and describe the steps in normalization.
6. Describe the process of creating an intersection relation to remove a many-to-many relationship.

Introduction

LEARNING OUTCOME 1

As you learned in Chapter 3, databases are quite powerful and can aid your organization in both transaction and analytical processing. But you must carefully design and build a database for it to be effective. Relational databases are similar to spreadsheets in that you maintain information in two-dimensional files. In a spreadsheet, you place information in a cell (the intersection of a row and column). To use the information in a cell, you must know its row number and column character. For example, cell C4 is in column C and row 4.

Databases are similar and different. You still create rows and columns of information. However, you don't need to know the physical location of the information you want to see or use. For example, if cell C4 in your spreadsheet contained sales for Able Electronics (one of your customers), to use that information in a formula or function, you would reference its physical location (C4). In a database, you simply need to know you want *sales* for *Able Electronics.* Its physical location is irrelevant. That's why we say that a ***database*** is a collection of information that you organize and access according to the logical structure of that information.

So, you do need to design your databases carefully for effective use. In this module, we'll take you through the process of designing and building a relational database, the most popular of all database types. A ***relational database*** uses a series of logically related two-dimensional tables or files to store information in the form of a database. There are well-defined rules to follow, and you need to be aware of them.

As far as implementation is concerned, you then just choose the DBMS package of your choice, define the tables or files, determine the relationships among them, and start entering information. We won't deal with the actual implementation in this module. However, we do show you how to implement a database using Microsoft Access in *Extended Learning Module J*.

Once you've implemented your database, you can then change the information as you wish, add rows of information (and delete others), add new tables, and use simple but powerful reporting and querying tools to extract the exact information you need.

Designing and Building a Relational Database

LEARNING OUTCOME 2

Using a database amounts to more than just using various DBMS tools. You must also know *how* to actually design and build a database. So, let's take a look at how you would go about designing a database. The four primary steps include

1. Define entity classes and primary keys.
2. Define relationships among entity classes.
3. Define information (fields) for each relation (the term *relation* is often used to refer to a file while designing a database).
4. Use a data definition language to create your database.

Let's continue with the example database we introduced you to in Chapter 3, that of Solomon Enterprises. Solomon Enterprises specializes in providing concrete to commercial builders and individual home owners in the greater Chicago area. On page 116 (Figure 3.3) in Chapter 3, we provided a graphical depiction of some of the tables in Solomon's database, including *Customer, Concrete Type, Order, Truck,* and *Employee.* As you recall, an order is created when a customer calls in for the delivery of a certain concrete type. Once the concrete is mixed, Solomon has an employee drive the truck to the customer's location. That illustrates how you can use a database in support of your customer relationship management initiative and order-processing function.

In this module, we want to design and model the supply chain management side for Solomon Enterprises. Figure C.1 contains a supply chain management report that Solomon frequently generates. Let's make some observations.

Figure C.1

A Supply Chain Management Report for Solomon Enterprises

SOLOMON ENTERPRISES Supply Report Ending October 14, 2005							
CONCRETE		RAW MATERIAL				SUPPLIER	
Type	Name	ID	Name	Unit	QOH	ID	Name
1	Home	B	Cement paste	1	400	412	Wesley Enterprises
		C	Sand	2	1200	444	Juniper Sand & Gravel
		A	Water	1.5	9999	999	N/A
			TOTAL:	4.5			
2	Comm	B	Cement paste	1	400	412	Wesley Enterprises
		C	Sand	2	1200	444	Juniper Sand & Gravel
		A	Water	1	9999	999	N/A
			TOTAL:	4			
3	Speckled	B	Cement paste	1	400	412	Wesley Enterprises
		C	Sand	2	1200	444	Juniper Sand & Gravel
		A	Water	1.5	9999	999	N/A
		D	Gravel	3	200	444	Juniper Sand & Gravel
			TOTAL:	7.5			
4	Marble	B	Cement paste	1	400	412	Wesley Enterprises
		C	Sand	2	1200	444	Juniper Sand & Gravel
		A	Water	1.5	9999	999	N/A
		E	Marble	2	100	499	A&J Brothers
			TOTAL:	6.5			
5	Shell	B	Cement paste	1	400	412	Wesley Enterprises
		C	Sand	2	1200	444	Juniper Sand & Gravel
		A	Water	1.5	9999	999	N/A
		F	Shell	2.5	25	499	A&J Brothers
			TOTAL:	7			

- Solomon provides five concrete types: 1—home foundation and walkways; 2—commercial foundation and infrastructure; 3—premier speckled (with gravel); 4—premier marble; 5—premier shell.
- Solomon uses six raw materials: A—water; B—cement paste; C—sand; D—gravel; E—marble; F—shell.
- Mixing instructions are for a cubic yard. For example, one cubic yard of commercial concrete requires 1 part cement paste, 2 parts sand, and 1 part water. The terms "part" and "unit" are synonymous.
- Some raw materials are used in several concrete types. Any given concrete type requires several raw materials.
- QOH (quantity on hand) denotes the amount of inventory for a given raw material.
- Suppliers provide raw materials. For a given raw material, Solomon uses only one supplier. A given supplier can provide many different raw materials.
- QOH and supplier information are not tracked for water (for obvious reasons). However, Solomon places the value 9999 in the QOH for water and uses 999 for the ID of the supplier.

When you begin to think about designing a database application, you first need to capture your business rules. Business rules are statements concerning the information you need to work with and the relationships within the information. These business rules will help you define the correct structure of your database. From the report in Figure C.1 and the observations above, we derived the following business rules.

1. A given concrete type will have many raw materials in it.
2. A given raw material may appear in many types of concrete.
3. Each raw material has one and only one supplier.
4. A supplier may provide many raw materials. Although not displayed in Figure C.1, Solomon may have a supplier in its database that doesn't currently provide any raw materials.

Before you begin the process of designing a database, it's important that you first capture and understand the business rules. These business rules will help you define the correct structure of your database.

STEP 1: DEFINE ENTITY CLASSES AND PRIMARY KEYS

LEARNING OUTCOME 3

The first step in designing a relational database is to define the various entity classes and the primary keys that uniquely define each record or instance within each entity class. An ***entity class*** is a concept—typically people, places, or things—about which you wish to store information and that you can identify with a unique key (called a primary key). A ***primary key*** is a field (or group of fields in some cases) that uniquely describes each record. Within the context of database design, we often refer to a record as an instance. An ***instance*** is an occurrence of an entity class that can be uniquely described with a primary key.

From the supply chain management report in Figure C.1, you can easily identify the entity classes of *Concrete Type, Raw Material,* and *Supplier.* Now, you have to identify their primary keys. For most entity classes, you cannot use names as primary keys because duplicate names can exist. For example, your school provides you with a unique student ID and uses that ID as your primary key instead of your name (because other students may have the same name as you).

From the report on page 144, you can see that the entity class *Concrete Type* includes two pieces of information—*Concrete Type* and a name or *Type Description*. Although *Type Description* is unique, the logical choice for the primary key is *Concrete Type* (e.g., 1 for home, 2 for commercial, and so on). Notice that the primary key name is the same as the entity class name. This is perfectly acceptable; if it is a potential point of confusion for you, change the primary key name to something like *Concrete Type ID* or *Concrete Type Identifier*. For our purposes, we'll use *Concrete Type* as the primary key name.

If you consider *Raw Material* as an entity class, you'll find several pieces of information including *Raw Material ID, Raw Material Name,* and *QOH*. The logical choice for a primary key here is *Raw Material ID* (e.g., A for water, B for cement paste, and so on). Although *Raw Material Name* is unique, we still suggest that you not use names.

Likewise, if you consider *Supplier* as an entity class, you'll find two pieces of information: *Supplier ID* and *Supplier Name*. Again, we recommend that you use *Supplier ID* as the primary key.

To summarize, the goal in this first step is to define and identify entity classes and their primary keys for your database. An entity class is something like a student, a supplier, a book, and so on. The primary key for each entity class uniquely defines each record or instance within the entity class. For our example of Solomon Enterprises, the entity classes and their primary keys are:

Entity Class	Primary Key
Concrete Type	*Concrete Type*
Raw Material	*Raw Material ID*
Supplier	*Supplier ID*

Your success in completing this first step will, in part, determine your success in completing the remaining steps correctly. If you incorrectly define entity classes and primary keys in this first step, you are assured of being unable to successfully complete the remaining steps. Take all the time you need to complete this vitally important first step correctly.

STEP 2: DEFINE RELATIONSHIPS AMONG THE ENTITY CLASSES

LEARNING OUTCOME 4

The next step in designing a relational database is to define the relationships among the entity classes. To help you do this, we'll use an entity-relationship diagram. An ***entity-relationship (E-R) diagram*** is a graphic method of representing entity classes and their relationships. An E-R diagram includes five basic symbols:

1. A rectangle to denote an entity class
2. A dotted line connecting entity classes to denote a relationship
3. A | to denote a single relationship
4. A O to denote a zero or optional relationship
5. A crow's foot (shown as <) to denote a multiple relationship

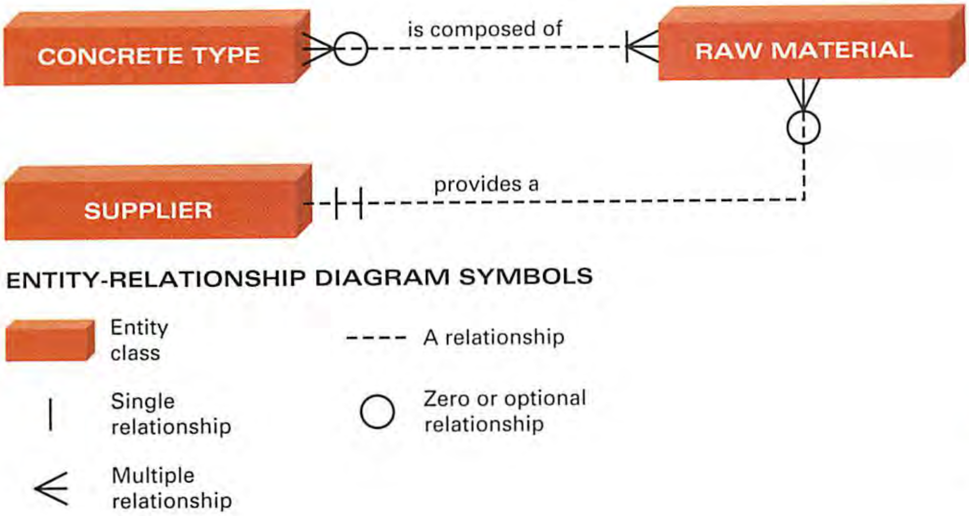

Figure C.2
An Entity-Relationship (E-R) Diagram

To use these symbols, you must first decide among which entity classes relationships exist. If you determine that two particular entity classes have a relationship, you simply draw a dotted line to connect them and then write some sort of verb that describes the relationship.

In Figure C.2, you can see the E-R diagram for the supply chain management side of Solomon's database. To determine where the relationships exist, simply ask some questions and review your business rules. For example, is there a relationship between concrete type and raw material? The answer is yes because raw materials are used in mixing the various concrete types. Likewise, the raw materials are provided by suppliers (another relationship). However, there is no logical relationship between concrete type and supplier. So, we drew dotted lines between *Concrete Type* and *Raw Material* and between *Raw Material* and *Supplier*. We then added some verbs to describe the relationships. For example, a *Concrete Type* is composed of *Raw Material,* and a *Supplier* provides a *Raw Material.*

It should also make sense (both business and logical) when you read the relationships in reverse. To do this, simply flip the location of the nouns in the sentence and change the verb accordingly. For example,

- *Concrete Type–Raw Material:* A *Concrete Type* is composed of *Raw Material.*
- *Raw Material–Concrete Type:* A *Raw Material* is used to create a *Concrete Type.*
- *Supplier–Raw Material:* A *Supplier* provides a *Raw Material.*
- *Raw Material–Supplier:* A *Raw Material* is provided by a *Supplier.*

Each of the preceding statements makes logical sense, follows the relationships we identify in Figure C.2, and reflects the business rules listed on page 145. Again, we stress the importance of using business rules. Technology (databases, in this instance) is a set of tools that you use to process information. So, your implementations of technology should match the way your business works. If you always start by defining business rules and using those rules as guides, your technology implementations will most likely mirror how your business works. And that's the way it should be.

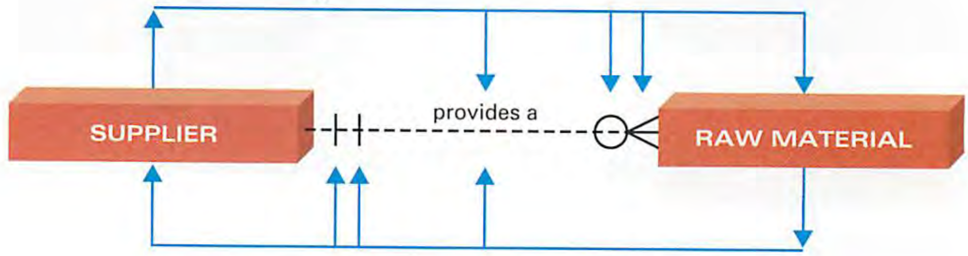

Figure C.3
Reading an Entity-Relationship (E-R) Diagram

Cardinality

Once you determine that a relationship does exist, you must then determine the numerical nature of the relationship, what we refer to as "minimum and maximum cardinality." To describe this, you use a | to denote a single relationship, a O to denote a zero or optional relationship, and/or a crow's foot (<) to denote a multiple relationship. By way of illustration, let's consider the portion of your E-R diagram in Figure C.3 above. To help you read the symbols and diagram, we've added blue lines and arrows. Following the line marked A, you would read the E-R diagram as:

"A *Supplier* may not provide any *Raw Material* (denoted with the O) but may provide more than one *Raw Material* (denoted with the crow's foot)."

So, that part of the E-R diagram states that the logical relationship between *Supplier* and *Raw Material* is that a *Supplier* may provide no *Raw Material* currently in inventory but may provide more than one *Raw Material* currently in inventory. This is exactly what business rule 4 (on page 145) states.

Following the blue line marked B, you would read the E-R diagram as:

"A *Raw Material* must be provided by one *Supplier* (denoted with the first |) and can only be provided by one *Supplier* (denoted with the second |)."

That statement again reinforces business rule 4.

Similarly, you can also develop statements that describe the numerical relationships between *Concrete Type* and *Raw Material* based on that part of the E-R diagram in Figure C.2. Those numerical relationships would be as follows:

- A *Concrete Type* is composed of more than one *Raw Material* and must be composed of at least one *Raw Material.*
- A *Raw Material* can be used to create more than one *Concrete Type* but is not required to be used to create any *Concrete Type.*

Again, these statements reinforce business rules 1 and 2 on page 145.

To properly develop the numerical relationships (cardinality) among entity classes, you must clearly understand the business situation at hand. That's why it's so important to write down all the business rules.

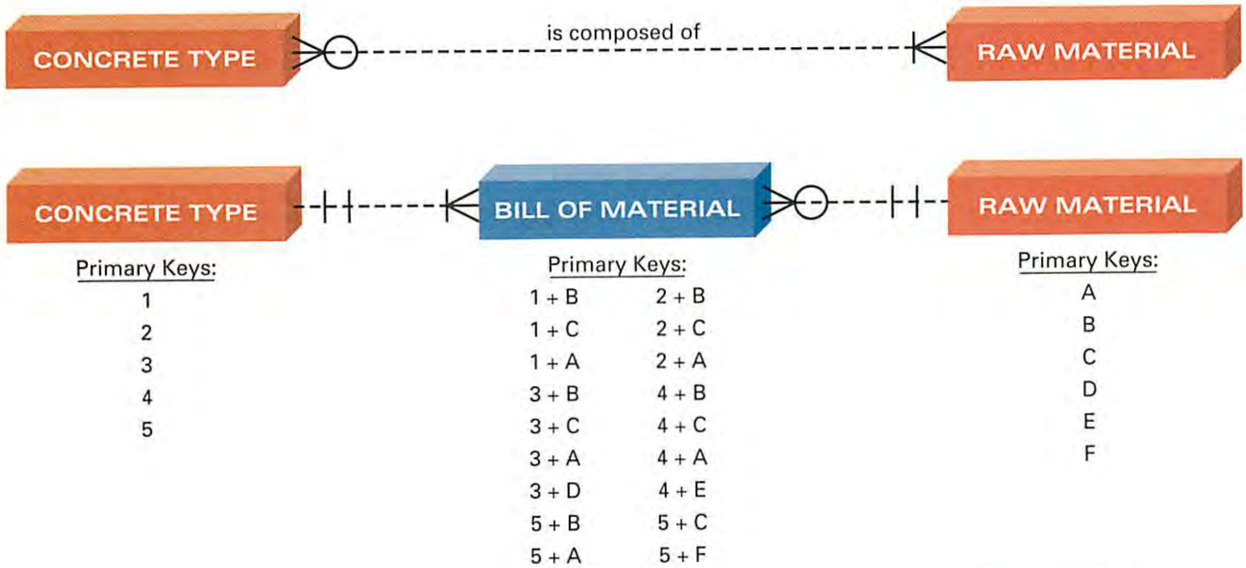

Figure C.4
Creating an Intersection Relation to Remove a Many-to-Many Relationship

LEARNING OUTCOME 5

After developing the initial E-R diagram, it's time to begin the process of normalization. ***Normalization*** is a process of assuring that a relational database structure can be implemented as a series of two-dimensional relations (remember: relations are the same as files or tables). The complete normalization process is extensive and quite necessary for developing organizationwide databases. For our purposes, we will focus on the following three rules of normalization:

1. Eliminate repeating groups or many-to-many relationships.
2. Assure that each field in a relation depends only on the primary key for that relation.
3. Remove all derived fields from the relations.

The first rule of normalization states that no repeating groups or many-to-many relationships can exist among the entity classes. You can find these many-to-many relationships by simply looking at your E-R diagram and note any relationships that have a crow's foot on each end. If you look back at Figure C.2 on page 147, you'll see that a crow's foot is on each end of the relationship between *Concrete Type* and *Raw Material.* Let's look at how to eliminate it.

In Figure C.4 above, we've developed the appropriate relationships between *Concrete Type* and *Raw Material* by removing the many-to-many relationship. Notice that we started with the original portion of the E-R diagram and created a new relation between *Concrete Type* and *Raw Material* called *Bill of Material,* which is an intersection relation. An ***intersection relation*** (sometimes called a ***composite relation***) is a relation you create to eliminate a many-to-many relationship. It's called an intersection relation because it represents an intersection of the primary keys between the first two relations. That is, an intersection relation will have a ***composite primary key*** that consists of the primary key fields from the two intersecting relations. The primary key fields from the two original relations now become foreign keys in the intersection relation. A ***foreign key*** is a primary key of one file (relation) that appears in another file (relation). When combined, these two foreign keys make up the composite primary key for the intersection relation.

For Solomon's supply chain management portion of its database, the intersection relation *Bill of Material* represents the combination of raw materials that go into each concrete type. Listed below is how you would read the relationships between *Concrete Type* and *Bill of Material* and *Raw Material* and *Bill of Material* (see Figure C.5).

- *Concrete Type–Bill of Material*
  From left to right: A *Concrete Type* can have multiple listings of *Raw Material* in *Bill of Material* and must have a listing of *Raw Material* in *Bill of Material.*
- From right to left: A *Concrete Type* found in *Bill of Material* must be found and can be found only one time in *Concrete Type.*
- *Raw Material–Bill of Material*
  From left to right: A *Raw Material* can be found in many *Bill of Material* listings but may not be found in any *Bill of Material* listing.
- From right to left: A *Raw Material* found in *Bill of Material* must be found and can be found only one time in *Raw Material.*

If you compare the E-R diagram in Figure C.5 to the E-R diagram in Figure C.2, you'll notice that they are very similar. The only difference is that the E-R diagram in Figure C.5 contains an intersection relation to eliminate the many-to-many relationship between *Concrete Type* and *Raw Material.*

LEARNING OUTCOME 6

Removing many-to-many relationships is the most difficult aspect when designing the appropriate structure of a relational database. If you do find a many-to-many relationship, here are some guidelines for creating an intersection relation:

Intersection Relations

1. Just as we did in Figure C.4, start by drawing the part of the E-R diagram that contains a many-to-many relationship at the top of a piece of paper.
2. Underneath each relation for which the many-to-many relationship exists, write down some of the primary keys.
3. Create a new E-R diagram (showing no cardinality) with the original two relations on each end and a new one (the intersection relation) in the middle.
4. Underneath the intersection relation, write down some of the composite primary keys (these will be composed of the primary keys from the other two relations).
5. Create a meaningful name (e.g., *Bill of Material*) for the intersection relation.
6. Move the minimum cardinality appearing next to the left relation just to the right of the intersection relation.
7. Move the minimum cardinality appearing next to the right relation just to the left of the intersection relation.

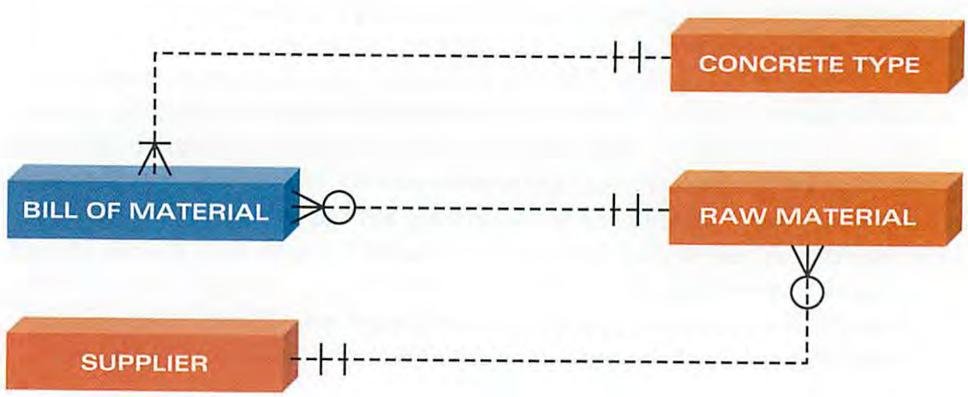

Figure C.5

The Completed E-R Diagram for the Supply Chain Management Side of Solomon's Database

8. The maximum cardinality on both sides of the intersection relations will always be "many" (the crow's foot).
9. As a general rule, the new minimum and maximum cardinalities for the two original relations will be one and one.

We would stress again that removing many-to-many relationships is the most difficult aspect when designing the appropriate structure of a relational database.

The business world is full of many-to-many relationships that must be eliminated before an organization can correctly implement a relational database. Let's talk through another example of a many-to-many relationship to help you better understand how to eliminate them. Consider that Solomon sometimes has to use more than one truck to make a delivery of concrete to a customer. That is, what if—given that a truck can carry at most 8 cubic yards of concrete—Triple A Homes asks for 12 cubic yards of premier marble for a given delivery. In that case, Solomon would have two choices for modeling and storing multiple trucks for the order. First, it could create two separate orders, one for 8 cubic yards of premier marble concrete and the other for 4 cubic yards of premier marble concrete. That option doesn't make business sense—when a customer places an order, then the entire order should be contained in only *one* order, not two.

The second choice—which is the correct one—is to have the ability to specify multiple trucks on a single order. In that case, Solomon would have a many-to-many relationship between *Order* and *Truck*. That is, the revised business rule would be that an *Order* can have multiple *Trucks* assigned to make the delivery and a *Truck* can be assigned to make a delivery on multiple *Orders*.

But it doesn't stop there. Solomon may wish to sometimes send two employees in one truck to deliver one order of concrete. Then, Solomon would need the ability to specify more than one employee per delivery truck.

As you can see, the business world is complex and full of many-to-many relationships. If you can master the art and science of eliminating many-to-many relationships in a database environment, you have created a substantial career opportunity for yourself.

STEP 3: DEFINE INFORMATION (FIELDS) FOR EACH RELATION

Once you've completed steps 1 and 2, you must define the various pieces of information that each relation will contain. Your goal in this step is to make sure that the information in each relation is indeed in the correct relation and that the information cannot be derived from other information—the second and third rules of normalization.

In Figure C.6 on the opposite page, we've developed a view of the relational database for Solomon based on the new E-R diagram with the intersection relation. To make sure that each piece of information is in the correct relation, look at each and ask, "Does this piece of information depend only on the primary key for this relation?" If the answer is yes, the information is in the correct relation. If the answer is no, the information is in the wrong relation.

Let's consider the *Raw Material* relation. The primary key is *Raw Material ID,* so each piece of information must depend only on *Raw Material ID.* Does *Raw Material Name* depend on *Raw Material ID?* Yes, because the name of a raw material depends on that particular raw material (as does *QOH* or quantity on hand). Does *Supplier ID* depend only on *Raw Material ID?* Yes, because the particular supplier providing a raw material depends on which raw material you're describing. In fact, *Supplier ID* in the *Raw Material* relation is a foreign key. That is, it is a primary key of one relation (*Supplier*) that appears in another relation (*Raw Material*).

What about *Supplier Name* in the *Raw Material* relation? Does it depend only on *Raw Material ID?* The answer here is no. *Supplier Name* depends only on *Supplier ID.* So, now the question becomes, "In which relation should *Supplier Name* appear?" The answer is in the *Supplier* relation, because *Supplier Name* depends on the primary key (*Supplier ID*) for that relation. Therefore, *Supplier Name* should appear in the *Supplier* relation (as it does) and not in the *Raw Material* relation.

Now, take a look at the intersection relation *Bill of Material.* Notice that it includes the field called *Unit. Unit* is located in this relation because it depends on two things: the concrete type you're describing and the raw material in it. So, *Unit* does depend completely on the composite primary key of *Concrete Type* + *Raw Material ID* in the *Bill of Material relation.*

If you follow this line of questioning for each relation, you'll find that all other fields are in their correct relations. Now you have to look at each field to see whether you can derive it from other information. If you can, the derived information should not be stored in your database. When we speak of "derived" in this instance, we're referring to information that you can mathematically derive: counts, totals, averages, and the like. Currently, you are storing the raw material total (*Raw Material Total*) in the *Concrete Type* relation. Can you derive that information from other information? The answer is yes—all you have to do is sum the *Units* in the *Bill of Material* relation for a given *Concrete Type.* So, you should not store *Raw Material Total* in your database (anywhere).

CONCRETE TYPE RELATION

Concrete Type	Type Description	Raw Material Total
1	Home foundation and walkways	5
2	Commercial foundation and infrastructure	4
3	Premier speckled (with smooth gravel aggregate)	8
4	Premier marble (with crushed marble aggregate)	7
5	Premier shell (with shell aggregate)	7

RAW MATERIAL RELATION

Raw Material ID	Raw Material Name	QOH	Supplier ID	Supplier Name
A	Water	9999	999	N/A
B	Cement paste	400	412	Wesley Enterprises
C	Sand	1200	499	A&J Brothers
D	Gravel	200	499	A&J Brothers
E	Marble	100	444	Juniper Sand & Gravel
F	Shell	25	444	Juniper Sand & Gravel

SUPPLIER RELATION

Supplier ID	Supplier Name
412	Wesley Enterprises
499	A&J Brothers
444	Juniper Sand & Gravel
999	N/A

BILL OF MATERIAL RELATION

Concrete Type	Raw Material ID	Unit
1	B	1
1	C	2
1	A	1.5
2	B	1
2	C	2
2	A	1
3	B	1
3	C	2
3	A	1.5
3	D	3
4	B	1
4	C	2
4	A	1.5
4	E	2
5	B	1
5	C	2
5	A	1.5
5	F	1.5

Unit belongs in this relation because it depends on a combination of how much of a given raw material *(Raw Material ID)* goes into each type of concrete type *(Concrete Type).*

Figure C.6

A First Look at the Relations for the Supply Chain Management Side of Solomon's Database

Once you've completed step 3, you've completely and correctly defined the structure of your database and identified the information each relation should contain. Figure C.7 shows your database and the information in each relation. Notice that we have removed *Supplier Name* from the *Raw Material* relation and that we have removed *Raw Material Total* from the *Concrete Type* relation.

Figure C.7

The Correct Structure of the Supply Chain Management Side of Solomon's Database

CONCRETE TYPE RELATION

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

RAW MATERIAL RELATION

Raw Material ID	Raw Material Name	QOH	Supplier ID
A	Water	9999	999
B	Cement paste	400	412
C	Sand	1200	444
D	Gravel	200	444
E	Marble	100	499
F	Shell	25	499

SUPPLIER RELATION

Supplier ID	Supplier Name
412	Wesley Enterprises
499	A&J Brothers
444	Juniper Sand & Gravel
999	N/A

BILL OF MATERIAL RELATION

Concrete Type	Raw Material ID	Unit
1	B	1
1	C	2
1	A	1.5
2	B	1
2	C	2
2	A	1
3	B	1
3	C	2
3	A	1.5
3	D	3
4	B	1
4	C	2
4	A	1.5
4	E	2
5	B	1
5	C	2
5	A	1.5
5	F	1.5

STEP 4: USE A DATA DEFINITION LANGUAGE TO CREATE YOUR DATABASE

The final step in developing a relational database is to take the structure you created in steps 1 through 3 and use a data definition language to actually create the relations. Data definition languages are found within a database management system. A ***database management system (DBMS)*** helps you specify the logical organization for a database and access and use the information within the database. To use a data definition language, you need the data dictionary for your complete database. Recall from Chapter 3 that the ***data dictionary*** contains the logical structure for the information in a database. Throughout this module and in the first part of Chapter 3, we provided you with the overall structure of Solomon's complete database including the relations of *Order, Truck, Customer, Employee, Concrete Type, Raw Material, Supplier,* and *Bill of Material.*

This is the point at which we'll end this extended learning module. But you shouldn't stop learning. We've written *Extended Learning Module J* to take you through the process of using a data definition language in Access to create the database for Solomon Enterprises.

Summary: Student Learning Outcomes Revisited

1. **Identify how databases and spreadsheets are both similar and different.** Databases and spreadsheets are similar in that they both store information in two-dimensional files. They are different in one key aspect: physical versus logical. Spreadsheets require that you know the physical location of information, by row number and column character. Databases, on the other hand, require that you know logically what information you want. For example, in a database environment you could easily request total sales for Able Electronics, and you would receive that information. In a spreadsheet, you would have to know the physical location—by row number and column character—of that information.
2. **List and describe the four steps in designing and building a relational database.** The four steps in designing and building a relational database include
    1. Define entity classes and primary keys
    2. Define relationships among entity classes
    3. Define information (fields) for each relation
    4. Use a data definition language to create your database
3. **Define the concepts of entity class, instance, primary key, and foreign key.** An ***entity class*** is a concept—typically people, places, or things—about which you wish to store information and that you can identify with a unique key (called a primary key). A ***primary key*** is a field (or group of fields in some cases) that uniquely describes each record. Within the context of database design, we often refer to a record as an instance. An ***instance*** is an occurrence of an entity class that can be uniquely described. To provide logical relationships among various entity classes, you use ***foreign keys***—primary keys of one file (relation) that also appear in another file (relation).
4. **Given a small operating environment, build an entity-relationship (E-R) diagram.** Building an entity-relationship (E-R) diagram starts with knowing and understanding the business rules that govern the situation. These rules will help you identify entity classes, primary keys, and relationships. You then follow the process of normalization, eliminating many-to-many relationships, assuring that each field is in the correct relation, and removing any derived fields.
5. **List and describe the steps in normalization.** ***Normalization*** is the process of assuring that a relational database structure can be implemented as a series of two-dimensional tables. The normalization steps include

1. Eliminate repeating groups or many-to-many relationships
2. Assure that each field in a relation depends only on the primary key for that relation
3. Remove all derived fields from the relations

6. **Describe the process of creating an intersection relation to remove a many-to-many relationship.** To create an intersection relation to remove a many-to-many relationship, follow these steps:

1. Draw the part of the E-R diagram that contains a many-to-many relationship
2. Create a new E-R diagram with the original two relations on each end and a new one (the intersection relation) in the middle
3. Create a meaningful name for the intersection relation
4. Move the minimum cardinality appearing next to the left relation just to the right of the intersection relation
5. Move the minimum cardinality appearing next to the right relation just to the left of the intersection relation
6. The maximum cardinality on both sides of the intersection relation will always be "many"
7. As a general rule, the new minimum and maximum cardinalities for the two original relations will be one and one

Key Terms and Concepts

Composite primary key, 149
Database, 143
Database management system (DBMS), 155
Data dictionary, 155
Entity class, 145
Entity-relationship (E-R) diagram, 146
Foreign key, 149
Instance, 145
Intersection relation (composite relation), 149
Normalization, 149
Primary key, 145
Relational database, 143

Short-Answer Questions

1. How are relational databases and spreadsheets both similar and different?
2. What is a database?
3. What are the four steps in designing and building a relational database?
4. What are some examples of entity classes at your school?
5. What is the role of a primary key?
6. What is an entity-relationship (E-R) diagram?
7. How do business rules help you define minimum and maximum cardinality?
8. What is normalization?
9. What are the three major rules of normalization?
10. What is an intersection relation? Why is it important in designing a relational database?
11. Why must you remove derived information from a database?
12. What is a database management system (DBMS)?

Assignments and Exercises

1. **DEFINING ENTITY CLASSES FOR THE MUSIC INDUSTRY** The music industry tracks and uses all sorts of information related to numerous entity classes. Find a music CD and carefully review the entire contents of the jacket. List as many entity classes as you can find (for just that CD). Now, go to a music store and pick out a CD for a completely different music genre and read its jacket. Did you find any new entity classes? If so, what are they?

2. **DEFINING BUSINESS RULES FOR A VIDEO RENTAL STORE** Think about how your local video rental store works. There are many customers, renting many videos, and many videos sit on the shelves unrented. Customers can rent many videos at one time. And some videos are so popular that the video rental store keeps many copies. Write down all the various business rules that define how a video rental store works with respect to entity classes and their relationships.
3. **CREATING AN E-R DIAGRAM FOR A VIDEO RENTAL STORE** After completing assignment 2, draw the initial E-R diagram based on the rules you defined. Don't worry about going through the process of normalization at this point. Simply identify the appropriate relationships among the entity classes and define the minimum and maximum cardinality of each relationship. By the way, how many many-to-many relationships did you define?
4. **ELIMINATING A MANY-TO-MANY RELATIONSHIP** Consider the following situation. At a small auto parts store, customers can buy many parts. And the same part can be bought by many different customers. That's an example of a many-to-many relationship. How would you eliminate it? What would you call the intersection relation? This one is particularly tough: You'll have to actually create two intersection relations to model this correctly.
5. **DEFINING THE CARDINALITY AMONG TWO ENTITY CLASSES** Consider the two entity classes of *Student* and *Advisor* at your school. How would you build an E-R diagram to show the relationship between these two entity classes? What is the minimum and maximum cardinality of the relationship?
6. **BUILDING A DATABASE OF STUDENTS, SEMINARS, AND TEACHERS** On the Web site for this text (www.mhhe.com/haag, select XLM/C), you'll find a robust running case study for this module. In Phase #1, you'll find a description of a school offering multiple sections of two different weekend seminars. You are charged with defining the entity classes and their primary keys for a supporting database. In Phase #2, you must define the relationships among the various entity classes. In Phase #3, you must define the cardinality for the relationships among the entity classes. In Phase #4, you'll have to create an intersection relation to take care of a many-to-many relationship. Finally, you must define all of the information (fields) for each relation. Tackle this case study to fully integrate everything you've learned in this module.