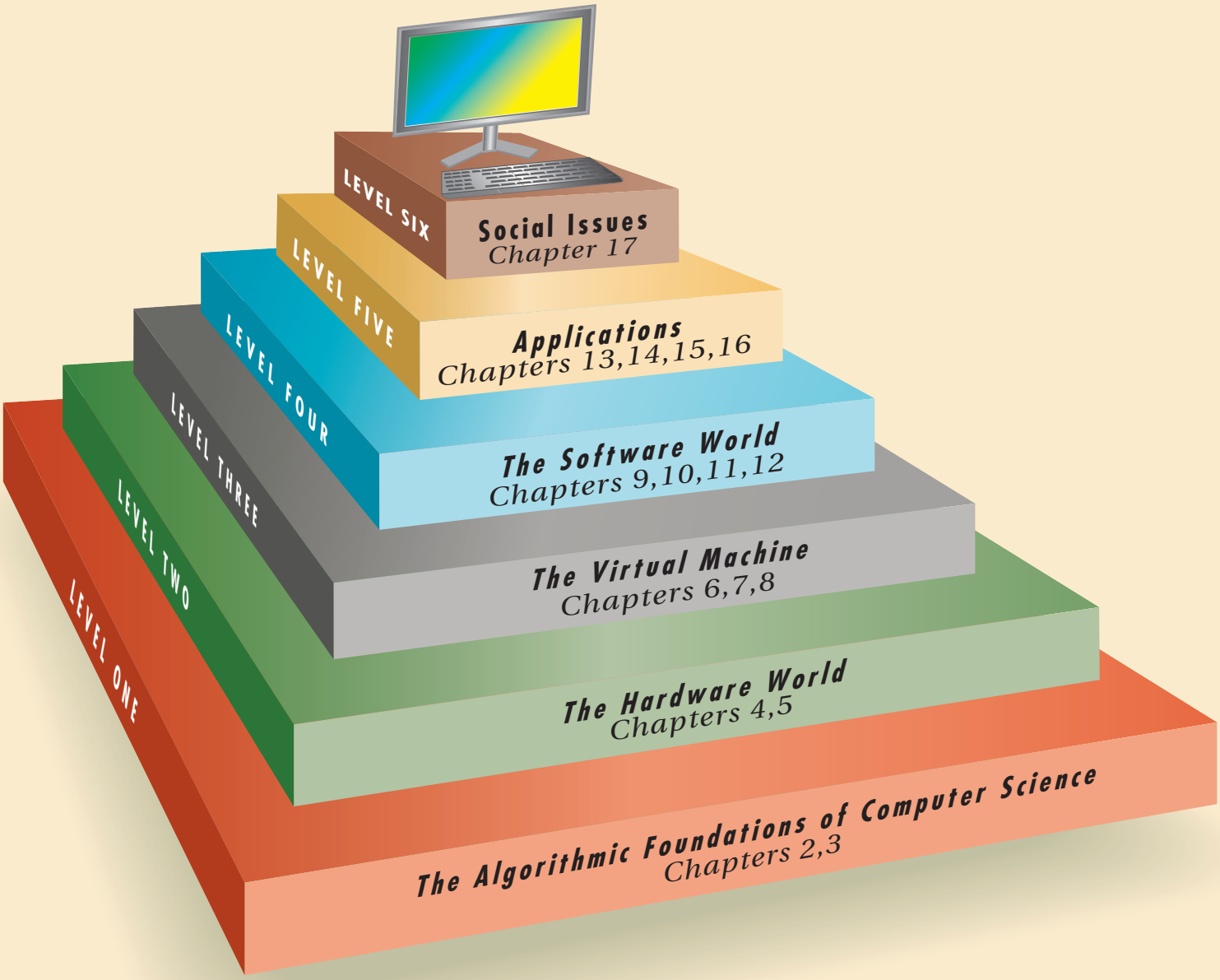

FIFTH EDITION
INVITATION TO
COMPUTER SCIENCE
G. Michael Schneider • Judith L. Gersting

5TH EDITION

# Invitation to Computer Science


LEVEL SIX
Social Issues
Chapter 17
LEVEL FIVE
Applications
Chapters 13,14,15,16
LEVEL FOUR
The Software World
Chapters 9,10,11,12
LEVEL THREE
The Virtual Machine
Chapters 6,7,8
LEVEL TWO
The Hardware World
Chapters 4,5
LEVEL ONE
The Algorithmic Foundations of Computer Science
Chapters 2,3

5TH EDITION

# Invitation to Computer Science

G. Michael Schneider
Macalester College

Judith L. Gersting
University of Hawaii, Hilo

Contributing author:
Keith Miller
University of Illinois, Springfield

COURSE TECHNOLOGY
CENGAGE Learning

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

**Invitation to Computer Science, Fifth Edition**
**G. Michael Schneider and Judith L. Gersting**

Executive Editor: Marie Lee

Acquisitions Editor: Amy Jollymore

Senior Product Manager: Alyssa Pratt

Development Editor: Deb Kaufmann

Editorial Assistant: Julia Leroux-Lindsey

Marketing Manager: Bryant Chrzan

Content Project Manager: Jennifer K. Feltri

Art Director: Faith Brosnan

Cover Designer: RHDG/Tim Heraldo

Cover Artwork: Fotolia.com (Royalty Free), Image # 375162

Compositor: Integra



ISBN-13: 978-0-324-78859-4
ISBN-10: 0-324-78859-2

**Course Technology**
20 Channel Center Street
Boston, MA 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at:
**international.cengage.com/region**

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

For your lifelong learning solutions, visit **course.cengage.com**
Visit our corporate website at **cengage.com**.



Printed in Canada
1 2 3 4 5 6 7 13 12 11 10 09

# BRIEF CONTENTS


**Chapter 1** An Introduction to Computer Science 1

**LEVEL 1** The Algorithmic Foundations of Computer Science 36

**Chapter 2** Algorithm Discovery and Design 39
**Chapter 3** The Efficiency of Algorithms 79

**LEVEL 2** The Hardware World 126

**Chapter 4** The Building Blocks: Binary Numbers, Boolean Logic, and Gates 129
**Chapter 5** Computer Systems Organization 187

**LEVEL 3** The Virtual Machine 236

**Chapter 6** An Introduction to System Software and Virtual Machines 239
**Chapter 7** Computer Networks, the Internet, and the World Wide Web 287
**Chapter 8** Information Security 333

**LEVEL 4** The Software World 356

**Chapter 9** Introduction to High-Level Language Programming 359
**Chapter 10** The Tower of Babel 397
**Chapter 11** Compilers and Language Translation 445
**Chapter 12** Models of Computation 491

**LEVEL 5** Applications 532

**Chapter 13** Simulation and Modeling 535
**Chapter 14** Electronic Commerce and Databases 561
**Chapter 15** Artificial Intelligence 585
**Chapter 16** Computer Graphics and Entertainment: Movies, Games, and Virtual Communities 617

**LEVEL 6** Social Issues in Computing 642

**Chapter 17** Making Decisions about Computers, Information, and Society 645

*Answers to Practice Problems* 673
*Index* 699

# CHAPTER 14

# Electronic Commerce and Databases


**14.1** Introduction
**14.2** E-commerce
- **14.2.1** The Vision Thing
- **14.2.2** Decisions, Decisions
- **14.2.3** Anatomy of a Transaction
- **14.2.4** Designing Your Web Site
- **14.2.5** Behind the Scenes

**14.3** Databases
- **14.3.1** Data Organization
- **14.3.2** Database Management Systems
- **14.3.3** Other Considerations
- LABORATORY EXPERIENCE 19

**14.4** Conclusion
EXERCISES
CHALLENGE WORK
FOR FURTHER READING

## 14.1 Introduction

As mentioned in Chapter 7, the Internet has been around for quite a while (since 1969), but it did not have a great impact on our everyday lives until the appearance of the World Wide Web in the early 1990s. Increasingly, the Web is our primary source of information about a variety of topics as well as a purveyor of goods and services from businesses on the other end of the wire.

If you own just about any type of business, you almost have to have a Web presence these days. For example,

- Your business provides a service, such as landscaping, that does not sell products directly to retail customers. But you use the Web for advertising—getting your name in front of the public, disseminating information on the services you provide, and convincing people to contact you or visit your place of business because you have superior services, knowledge, capabilities, and price.
- Your business provides a service for which follow-up information is important. For example, you are a shipping company and you use your Web site to allow customers to track their shipments.
- Your business provides a service that enables customers to engage in online transactions that are not retail sales. For example, you are a bank that allows customers to use the Web to view their current account balances and to transfer money between accounts.
- Your company sells products or materials to other companies rather than to the general public. You maintain a **B2B (business-to-business)** Web presence to streamline transactions between you as the seller and other businesses as buyers. Your goal is not only to advertise and attract new business customers but also to cut down transaction costs. (Note that the sales figures given in the box "Shopping on the Web" do not include these wholesale B2B transactions.)
- Your company is a retail business, and you maintain a **B2C (business-to-consumer)** Web site. You do this to advertise your products and to allow the general public to make online purchases.

In this chapter we'll talk mostly about the last scenario—selling retail products to the general public. This is how most consumers interact with and experience the Web's commercial capabilities.

Assume that you run a retail rug business—let's call it "Rugs-For-You"—out of a traditional store, that is, a store with a physical building, display windows,

aisles with merchandise, and salespersons. You have decided to expand your retail business into the **e-commerce world,** where financial transactions are conducted by electronic means. During the early stages of online commerce (the early and mid-1990s), this might have meant that a customer would fill out an order via the Web and submit it. The online order was printed out by the business at the other end, and this paper document was then processed much like any traditional purchase, including rekeying the order data for both the shipping and billing departments. The Web allowed the customer to initiate an order, but it had little or no role in filling the order, transferring funds, or restocking inventory.

This early approach to online commerce was cumbersome, inefficient, and error-prone. Today businesses have moved away from this restricted model of online commerce to a total **e-business** concept where orders are processed, credit is verified, transactions are completed, debits are issued, shipping is alerted, and inventory is reduced, all electronically—at least in theory. The business may operate completely online, or it may also have a physical retail site. In the latter case, it uses the Web to complement and enhance its traditional "brick-and-mortar" business. This describes the e-business model you want to implement for your Rugs-For-You business. In addition to your traditional store, you have decided to establish a Web presence for your business where customers can come, view area rugs for sale, ask questions, make a selection, purchase their rug, and have it delivered to them, all in a quick, easy, and secure electronic environment.

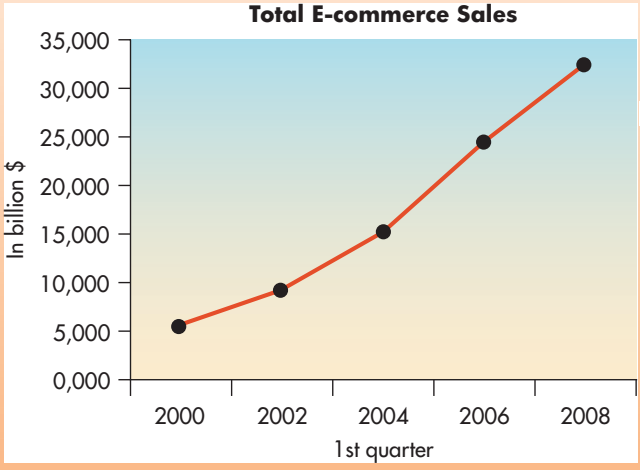
In the next section, we'll look at some of the many considerations involved in such a decision. Some of these are technical; some are purely business; many are a combination of the two. Then we'll look more closely at databases, one of the most important features of the e-commerce world.

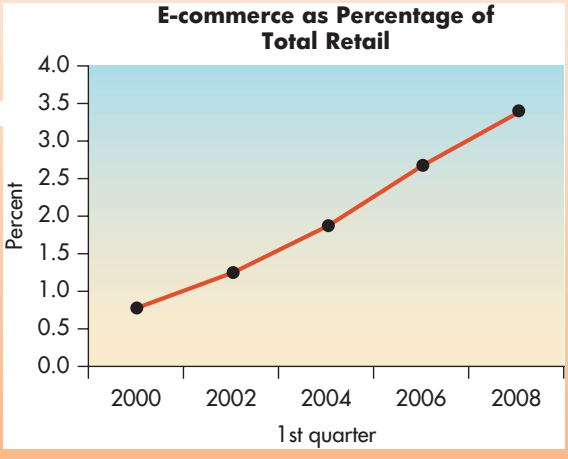
## Shopping on the Web

The Census Bureau of the U.S. Department of Commerce estimated e-commerce retail sales in the United States for the first quarter of 2008 to be $32.4 billion, an increase of 32% from the first quarter of 2006.

But to keep these numbers in perspective, in the first quarter of 2008, e-commerce sales accounted for only 3.4% of total U.S. retail sales, also an increase over the 2006 figure for the same period.

Clearly, there has been significant growth in the e-commerce retail sector, but, just as clearly, there is room for much more.



## 14.2 E-commerce

In Chapter 10 we talked about HTML, the language used to build Web pages. As a small business owner, you may not know much about HTML, to say nothing of the many other technologies used in creating Web pages, such as XML. (See the box "Beyond HTML" in Section 10.3.2.) However, you can hire someone who knows these technologies, ask that person to put together some Web pages for your store, and presto, you're in e-business! But maybe not for long.

Opening an online store requires at least as much planning as building another physical store location—in fact, probably more, because it is a different medium in which to do business. Although it's possible to build Web pages in a virtual environment more quickly and cheaply than constructing a building, that doesn't mean you should charge ahead without the proper level of planning and forethought.

### 14.2.1 *The Vision Thing*

The first question you need to answer is: What is your vision for this new part of your overall commercial enterprise? Put another way: What is the business problem you are trying to solve? Do you want to:

- Broaden your customer base?
- Recapture customers you are losing to competitors with online stores?
- Better serve your existing customer base?
- Better integrate departments/functions within your existing business, so that the Shipping Department and the Accounting Department, for example, work off the same order form?

Any of these might be legitimate reasons for moving into e-commerce, but have you considered the risks involved with this decision?

- Will you just move your in-store customers online and achieve no overall gain?
- When you expose yourself to online competition, will you have something unique to offer?
- Does your existing customer base need or want anything that you don't or can't provide in your traditional business environment? What part of your existing customer base will never shop online?
- Are the employees in your Shipping and Accounting departments in agreement with this idea, or do they feel threatened by change?

And we haven't even mentioned the costs involved with this decision:

- Do you have all the necessary hardware (computers), software, and infrastructure (network connectivity) to host a business Web site? If not, what will it cost you to acquire or lease them?
- Do you have the personnel and skills you need to build and maintain a Web site? If not, what will it cost to acquire new personnel or retrain existing personnel?

- Do you have the legal expertise onboard to manage issues such as (1) protecting your intellectual property; (2) navigating regulations, tariffs, and taxes in the many geographic regions where you will now be doing business (including perhaps overseas); and (3) legally handling customer data collected online? If not, what will it cost you to acquire this expertise?
- Do you know the potential costs of diverting resources away from your existing traditional business?

Let's assume that you and your company officers have assessed the objectives, the risks, and the costs, and you feel that overall your bottom line will improve by going online. What should happen next?

### 14.2.2 *Decisions, Decisions*

Once you decide to move into the e-commerce arena, there are still many questions to be answered and decisions to be made. The first major decision is choosing between **in-house development** and **outsourcing**; in fact, this is not a single decision but a whole host of decisions.

First, the personnel issues: Are you going to use your existing staff to develop this e-business, either because they already have the necessary skills or because they will be retrained? Will you hire new personnel with the needed skills? Will you hire consultants who bring a good deal of expertise and will work with your people to get things up and running quickly? Or, alternately, will you turn the entire job over to an **ASP (application service provider)** who, for a fee, will design your Web site and manage it on an ongoing basis? The answers to these questions depend, of course, on the skills of your existing staff, how quickly you want your site to be up and running, the costs involved, and how much control you are willing to relinquish. Whoever is chosen to develop your Web site, it is important that business information about your company is made available and is used to shape all decisions along the way. In the end, your Web site should capture the "image" and provide the customer services of *your* company, not the company someone else may have in mind.

Similar questions follow regarding the hardware and software. You will need at least one Web server machine to host your Web site. You may need additional computers to store your customer database information, to support program development, and to supply the appropriate network connections and security. Do you have these machines? Will you buy them? Will you lease space on someone else's commercial Web server? You will also need a good deal of new software, such as programs to process the customer orders that you hope will come pouring in, to interact with your accounting, shipping, and inventory control software, and to manage and store customer information. Will you use inexpensive, off-the-shelf software or more expensive packages that you can customize for your business needs? Will you use commercial software or **open source software**? (See "The Open Source Movement" box in Chapter 6.) Will your company develop its own proprietary software that can be modified whenever your business needs change? Of course, if you decide to turn everything over to an ASP, you will have little or no control over these hardware and software decisions.

### 14.2.3 *Anatomy of a Transaction*

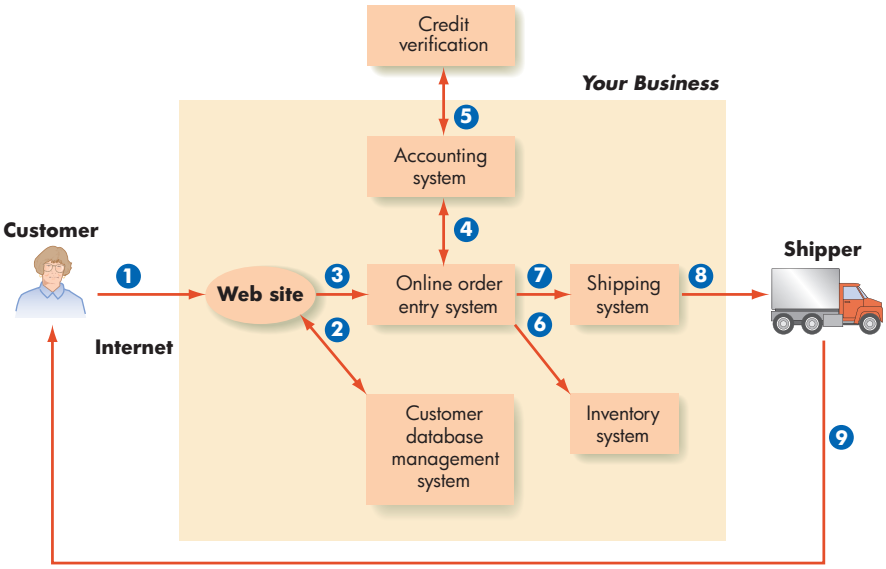
What draws a customer to online shopping? The number one attraction is probably convenience. Your online store is open 24 hours a day. People can shop from the comfort of home, save time, and avoid the hassles of traffic. It is also easy to comparison shop merely by hopping from one Web site to another. But this also means that your competition is just a click away. Your goals are to:

- Draw potential customers to your site
- Keep them there
- Set up optimum conditions for them to complete a purchase

Figure 14.1 illustrates the major components of an online purchase, which we have broken down into nine steps. Next, we'll elaborate on these steps, with an eye to the three goals mentioned above.

**STEP 1: GETTING THERE.** How can you get customers to your Web site? Technically, once the customer knows the URL (uniform resource locator), the process works exactly as described in Chapter 7. The customer hooks up to the Internet through his or her ISP (Internet service provider) and puts the URL into his or her Web browser. The browser works with the DNS (Domain Name System) to find the unique IP (Internet Protocol) address for this URL. Using this address, the TCP (Transport Control Protocol) routes a connection through the Internet from the customer's machine to the appropriate server. The browser uses this connection to send an HTTP GET message for the desired Web page, which is then transmitted from the Web server back to the browser and displayed on the user's screen, at which time the TCP/IP connection is broken.

**FIGURE 14.1**
*A Typical Online Transaction in Nine Steps*



But how does your potential customer learn your URL in the first place? There are many possibilities:

- *Conventional advertising*. You post your home page URL on flyers, in print and TV advertisements, on letterhead, and on any other traditional promotional materials you may produce.
- *Obvious domain name*. You want your **domain name** (your home-page URL) to relate so closely to your business name that potential customers can easily guess it if they don't have it in front of them. Who wouldn't try *www.mcdonalds.com* to reach this well-known fast-food giant? Of course, Rugs-For-You might not be quite that well known. Domain names are registered by companies that are accredited for this purpose by **ICANN (Internet Corporation for Assigned Names and Numbers)**, a nonprofit corporation that took over the task of domain name management from the U.S. government in 1998. When a domain name is registered, it becomes part of the DNS so that Web users can find your IP address and get to your site. A list of accredited registrars can be obtained from *http://www.icann.org/registrars/accredited-list.html*, an information Web site maintained by ICANN. A number of Web sites allow you to determine whether a particular domain name has already been registered. In addition to registering your "real" domain name (*rugs-for-you.com*), you would be wise to register obvious spelling variants (*rugs-for-u.com*, *rugs-4-u.com*, etc.) if they are available, so that all roads lead to your Web site.
- *Search engine*. Potential customers may use an Internet search engine to search for Web sites about products that you sell, and your company's Web site may turn up in the list returned as a result of this search. You can also pay for a "sponsored link" so that a search on appropriate keywords will bring up links to your Web site in a prominent spot on the search engine's page or near the top of the list of search results.
- *Portal*. A **portal** is an entry point Web page with links to other Web pages on some topic. It can be thought of as a starting point to learn about a particular subject, and it typically contains many helpful pointers to useful information on that subject. For example, *www.floorbiz.com* is a portal with links to retail stores selling rugs, carpet, tile, adhesives, padding, cleaning equipment, and so forth. This site also features links about flooring materials and manufacturers, links to news articles and press releases, upcoming conventions, and employment opportunities connected with the flooring industry, as well as forums for bulletin board postings, links to tips (e.g., how to maintain hardwood floors), leads for contractors to bid on floor installation jobs, and an opportunity to register to receive e-mail. You would certainly want to have a link to Rugs-For-You from this portal page, and you may even want to purchase a **banner ad** (a graphical ad, often with animation, placed in a prominent position on a Web page) so that anyone who goes to this portal sees the *rugs-for-you.com* link right away.

## A Rose by Any Other Name . . .

*Cybersquatting* is the practice of registering a domain name that uses the name or trademark of an existing business, with the intent to sell the name to that business at a profit or to capitalize on that name for some other purpose. A 1999 federal law called the Anti-Cybersquatting Consumer Protection Act (ACPA) makes cybersquatting illegal. A trademark owner claiming to be a victim of cybersquatting can file a suit under the ACPA. To have its claim upheld, a trademark owner must prove that it was the first to use the name or trademark for commercial purposes, that the name or trademark was distinctive at the time the domain name was first registered, that the domain name is the same as or sufficiently similar to the trademark as to cause confusion, and that the domain name registrant had a bad faith intent to profit from the trademark. A trademark owner who wins a suit can obtain the rights to the domain name and perhaps be awarded monetary damages up to $100,000.

ICANN also arbitrates cybersquatting disputes, with essentially the same criteria, but does not award any monetary damages. International disputes may be brought before the World Intellectual Property Organization, a United Nations agency. In 2000, AT&T won its case before WIPO against WorldclassMedia.com of Austria over the domain name attmexico.com, and Microsoft won its case against Global Net 2000, Inc., of Tehran, Iran, over the domain name microsoftnetwork.com.

A variant of cybersquatting, called **typosquatting**, takes advantage of typographical errors a user might make when typing a URL directly into the browser, as opposed to following a link, estimated to be about 15% of all Web traffic. For example, *www.rugs-for-you.com* might be entered as *www.rugs-for-you.org* (wrong top-level domain name), *www.rug-for-you.com* (spelling error), or *www.rusg-for-you.com* (transposition error). A company may have registered a few of these types of variations so that they link to the company's legitimate Web site, but there are many possibilities, some of which may be registered to typosquatters. If the user reaches a typosquatting site, the resulting Web page may contain anything from shocking or pornographic material to pay-per-click advertising links targeted to the user's interests, based on the likely site the user was trying to reach. Each click on such an ad link generates a small amount of revenue from the advertiser for the typosquatter, as well as for the ad network that brokered placement of the ad. Google—legitimately—makes its money by collecting a small sum for each click on an advertising link on its own Web pages. But in May 2008, a federal judge ruled that Google might be subject to suit for violation of the ACPA because it might post links to typosquatting pages that contain ads from which Google stands to profit, in its ad network capacity. This is an interesting case because Google is not the owner of the typosquatting domain names.

Microsoft estimates that on any given day there are 2000 registered domain names that contain Microsoft trademark terms (e.g., *downloadvistaforwindows.com*) operated by cybersquatters seeking to profit from Microsoft intellectual property via pay-per-click online ads.

**STEP 2: DO I KNOW YOU?** Regular customers at your traditional store are treated with special care. You may mail them promotional offers that you think will be of interest to them, and the salespeople know them when they walk into the store and greet them by name. You pay particular attention to their needs because, after all, return customers are the basis of your business. How will your online store provide this type of personalized attention?

Some sites ask users to register and then log in when they revisit the site. These sites consult the database of registered customers and recall pertinent information—for example, how the customer browsed the site previously, what pages the customer visited, where the customer lingered, what the customer bought, as well as more mundane information such as name and address. What the return customer sees is tailored to reflect this information.

Other sites that do not require a customer login still greet the customer with "Welcome, John," for example, and arrange a Web page with items tied to John's apparent interests, based on his last purchase. This type of Web site personalization can be accomplished by means of *cookies*. A **cookie** is a small text file that the Web server sends to the user's browser and that gets stored

on the user's hard drive. It contains personal information about the user, such as name, address, time of visit, and what was looked at or bought. On the customer's next visit to that same site, the browser sends the cookie back to the server (along with the page request) so the server can create a customized page just for this shopper. This does more than merely create a friendly, personalized atmosphere. It also allows the server to record information for later use. For example, cookies enable a customer to put items into his or her online shopping basket and return at a later time to find them still there.

Transmission of Web pages between a client and server is **stateless**; that is, no information about this exchange is permanently retained by the server. Indeed, recall that the TCP/IP connection between the browser and Web server is (usually) broken once a Web page has been sent back to the browser. A totally new connection has to be established to access a different page or to return later to that same page. Without cookies, there is no association between the customer visiting one page and the same customer visiting another page, or between the same customer visiting the same page at different times. It's possible to configure a Web browser to not accept cookies, but cookies cannot execute on the client machine and are harmless. They just take up a little space.

You can provide incentives and benefits for return customers—product support for items already purchased, special promotions ("John, would you like some stain guard for that new rug you just bought? Click here for our special offer!"), free shipping, a clearly stated return policy (including the ability to return items to your traditional brick-and-mortar store if more convenient), and a chance to register complaints or ask questions online (to which you should pay attention and respond). And certainly you should provide a toll-free number where your customers can speak with a real, live person, although you don't want to make the number too prominent on your site, because you are looking for your online business to free up staff, not burden them.

Online customers, both new and returning, can leave your site in the blink of an eye or, more precisely, the click of a mouse button. Your Web site must invite them in, entice them to stay, and make their path toward purchase so convenient that there is no reason not to buy from you. This is what makes designing a Web page so much more than just an HTML programming assignment! We'll talk more about Web page features in Section 14.2.4, but for now let's assume that a customer has successfully navigated your Web site, selected an item to purchase, and is ready for Step 3.

**STEP 3: COMMITTING TO AN ONLINE PURCHASE.** Customers are understandably hesitant to transmit sensitive information such as their credit card number, or even their name and address, over the Web. Your site must provide security for transmitting this information, and that security comes in two pieces: encryption and authentication. **Encryption** encodes the data to be transmitted into a scrambled form, using a scheme agreed on between the sender and the receiver. While encryption provides for the secure transmission of data, this is of little use if the data are not being sent to the correct party. **Authentication** is the process of verifying the identity of the receiver of the data. In Step 3 of our online transaction process, the sender is the customer (actually the customer's Web browser) placing an order and sending confidential personal and financial information, and the receiver is the retailer's Web server. In Chapter 8 we discussed how the SSL (Secure Sockets Layer) and TLS (Transport Layer Security) protocols provide encryption and authentication for Web transactions. There we learned that the Web server can pass to the browser a certificate of authentication issued by a trusted third party such as VeriSign (*http://www.verisign.com*).

Because you decided to use VeriSign SSL software on your Web site, you can post a "VeriSign Secured" seal on your pages that users can click to verify that they are at the correct site. After all, they didn't walk into your physical place of business, so how do they know where they really are? The URL is *www.rugs-for-you.com*, and there are many pictures of rugs, but maybe it is simply a scam where the customer will send money but receive nothing in return. **Spoofing** is the practice of impersonating a legitimate site for the purposes of stealing money or stealing identity by collecting confidential information such as credit card numbers, names, and addresses. Clicking on your VeriSign seal might bring up a window with the information shown in part in Figure 14.2.

Customers about to transmit sensitive information to your Web site are alerted by a message saying they are being transferred to a secure site. The corresponding Web page has the protocol heading **https**, rather than http, with the *s* signifying a site under the protection of SSL. Customers may also see a little lock graphic on the Web page to indicate a secure site, and the browser address bar may turn green. When they leave the site, they receive a message saying they are leaving a secure site.

**STEPS 4 AND 5: PAYMENT PROCESSING.** Let's assume that your customers will pay with credit cards, the most common online option. The online order form communicates with your accounting system (Step 4), which might verify the customer's credit and process this transaction with the credit company (Step 5) on the fly, that is, while the customer waits. This way, the customer can be alerted and given another chance to enter information if there is an error. In addition, you do not have to store the customer credit card number in your database, which reduces your security risk.

Another option is to collect information on the customer's order, including an e-mail address (Step 4), close the order process, and then evaluate the customer's credit and complete the transaction offline (Step 5). Once the transaction is completed, an e-mail confirmation is sent to the customer. To use this option you must maintain customer credit card information.

**FIGURE 14.2**
*Secure Site Assurance*

WWW.RUGS-FOR-YOU.COM is a VeriSign Secure Site

Security remains the primary concern of online consumers. The VeriSign Secure Site Program allows you to learn more about Web sites you visit before you submit any confidential information. Please verify that the information below is consistent with the site you are visiting.

Name: WWW.RUGS-FOR-YOU.COM Status: Valid
Validity Period: 14-May-09–25-May-12
Server ID Information:
Country = U.S.
State = Ohio
Locality = Cleveland
Organization = Rugs-For-You, Inc.
Common Name = www.rugs-for-you.com

If the information is correct, you may submit sensitive data (e.g., credit card numbers) to this site with the assurance that:

This site has a VeriSign Secure Server ID.
All information sent to this site, if in an SSL session, is encrypted, protecting against disclosure to third parties.

**STEPS 6–9: ORDER FULFILLMENT.** Once your customer's credit is approved, your order entry system must alert your inventory system to decrement the number of items in stock by whatever quantity the user has purchased (Step 6) and must also contact your shipping system to arrange for shipping (Step 7). The shipping system works with the shipping company you use (Step 8) to pick up and deliver the purchase to the customer (Step 9).

### 14.2.4 *Designing Your Web Site*

Your Web site must be designed with your customers in mind. It has to be fresh and up to date, ever changing, and always displaying the latest product information. Department stores don't keep the same displays in their windows for months or years on end, and neither should you. One of your earliest decisions is your Web site **taxonomy**—how information is classified and organized so customers can easily find what they want. At *rugs-for-you.com*, you could organize your site by rug manufacturer, color, size, material, or by rooms in the house. There are many options, and you must consider how your customers usually shop for their rugs.

Your customers should always know where they are on your Web site. As we mentioned in Chapter 7, hypertext allows a user to move easily from page to page by simply clicking a link. However, after a few clicks, it is easy to become totally lost and not know where you are or how to get back. A **site map** or a **navigation bar** can provide a high-level overview of your site architecture, plus make it easy to navigate (i.e., move from page to page) through the site. A good rule of thumb is that the customer should be able to get from any page in your Web site to any other page in four clicks or fewer. And although you want to encourage browsing, just as you do in your physical store, you also want customers to be able to find what they are looking for quickly, so your Web pages should include the ability to search the site.

You need electronic "shopping carts" and order checkout forms. Keep in mind that customers want to feel in control (especially of their money!). Be sure that as customers step through the ordering process, they are always informed about the current order—items being ordered, quantity, price, and so on—and about what will happen with the next button click. It is also important to give customers the option to go back and change something or to clearly indicate that, following the next click, the order will be final and no further changes will be possible.

Give customers shipping options so that they can make the best trade-off between cost and speed of delivery. Send e-mail to confirm orders, and send follow-up e-mails when orders are shipped.

Display your privacy policy on the Web page. Tell your customers what personal information you collect, why this information is needed, how you will use it, whether you will share it and with whom you might share it, and how you will store and safeguard it. Also, understand what information you can legally collect, based on the regulations of the state or country of your target users.

You may also want to offer extras to your customers. Put up a **FAQ (frequently asked questions)** page or a bulletin board for discussion groups. You can ask customers if they wish to subscribe to an e-mail newsletter to alert them to the latest products (no spam, please), with the option to unsubscribe at any time. Give your customers a "suggestion box." Allow them to track their shipment through an order number. Post news and press releases

about your business or products. And again, configure your site in a personalized way for return customers. All of these measures can help improve customer satisfaction, build customer relationships, and bring people back to your Web site time and time again. The suggestions and ideas listed above are part of your online **CRM (customer relationship management)** strategy.

At the same time that you want to cram all this content into your Web pages, your site must adhere to good design principles. It must look professional and uncluttered. Avoid glaring colors, flashing images, and annoying pop-up windows, although there could be a *judicious* use of animation or changing, tasteful images. Make good use of white space—it can draw attention to the items you want emphasized. All of your pages should have a consistent look and feel and a consistent set of navigation tools; this can be accomplished by designing a master template page from which all pages are derived. Be sure your company logo and/or slogan are part of this master template.

On the technical side, your Web pages should be designed to be displayed on many different machines with different operating systems and browsers (e.g., Internet Explorer, Safari, or Firefox). Not all browsers render every HTML element in exactly the same way. Users may run monitors at different screen resolutions and have widely varying communication speeds, from tens of thousands to tens of millions of bits per second. (See Section 7.2.1, "Communication Links.") Your Web design should use only those features that you know will work satisfactorily on virtually every machine and browser that your customers are likely to use. Offer features such as text-only options for users with slow connections. Adhere to ADA (Americans with Disabilities Act) requirements for Web accessibility (see box).

As you can see from our brief discussion, designing Web pages, or at least a successful set of commercial Web pages, is a difficult and complex task. It

## Accessible Web Pages

The Americans with Disabilities Act, signed into law in 1990, ensures equal opportunity for persons with disabilities in employment, government services, public facilities, and transportation. ADA has been interpreted to apply to Web page accessibility. In addition, Section 508 of the federal Rehabilitation Act mandated in 1998 that all U.S. federal agencies must make their Web pages accessible to people with disabilities.

One of the most common issues in Web page accessibility relates to images, charts, or photographs. Blind users or users with low vision have several assistive technologies available to them, such as speech synthesizers that speak text that appears on the screen or devices that translate text on the screen into touch-readable Braille. These technologies can only read text, so a visual element on a Web page needs a corresponding text tag in the HTML code for the image. The text tag should be as descriptive as possible. Here's an image that might appear on one of the *rugs-for-you.com* Web pages with its corresponding text tag as displayed by the browser.

ORIENTAL THROW RUGS

Afghanistan, wool, 2 ft x 3 ft

**Rectangular rug, intricate red and gold geometric designs on black background, wide border with repeating geometric pattern**

Price: $475

involves not only computer science skills (e.g., HTML, XML, HTTP, TCP/IP), but a knowledge of such fields as art, graphics design, business, management, and consumer psychology, to name but a few. It is easy to create just any Web page, but much more difficult to create a really good one.

### 14.2.5 *Behind the Scenes*

Your business maintains a number of other computer applications in addition to your online order entry system. In Figure 14.1, we saw that there are accounting, inventory control, and shipping systems as well as a customer database, and that's just to deal with customers. You also have systems that deal with your suppliers to manage orders, shipments, billing, and payments. Finally, you have personnel systems to deal with your employees—payroll, insurance, Social Security. Some of these systems may be brand-new and just installed (like your new Web site), whereas others may be "legacy" code that has been around for dozens of years.

Obviously, these systems are not all independent of one another, and some must collaborate quite closely. For example, your inventory control system must communicate with the supplier order system whenever you run low on an item and must restock it. Your accounting system needs to inform the customer database when a payment has been made. However, these systems may have been developed by different vendors (some functions may even be done by hand) and may run on different machines using totally different protocols and formats from those on your new Web site. Because of this, once the Web site is up and running, you may need to invest in **middleware**—software that allows separate, existing programs to communicate and work together seamlessly. These middleware packages do such things as translate between incompatible data representations, file formats, and network protocols to allow otherwise incompatible systems to exchange information. This allows your new e-commerce application to access and/or transmit important business data to all other parts of your company.

Finally, as soon as you have your enterprise humming along smoothly as an e-commerce site, you will need an effective **disaster recovery strategy**. What are your plans for backing up critical data? What is your plan to keep your online business open even when your server fails? What will you do if a hacker breaks into your Web site and steals customer information? Without a plan, you are never more than one electrical storm, one malicious user, or one disk failure away from catastrophe.

By now you may have surmised that you need a lot of help to put together your successful e-business. You need network help, programming help, graphics design help, and legal advice, as well as input from those who know your business well.

## PRACTICE PROBLEM

Try to locate a portal page for at least one of the following topics: health care, environmental issues, basketball, higher education, and/or the steel industry.

## The Price of Success

One happy thought is that your e-business might grow to be so successful that you have to scale up beyond your expectations. Amazon.com is one of the most successful e-businesses, and has expanded beyond its original book-selling role to include sales of toys, clothing, electronics, kitchen goods, housewares, and home and garden items.

July 21, 2007, marked the release of *Harry Potter and the Deathly Hallows*, the final book in the very popular Harry Potter series by author J. K. Rowling. By the day before it went on sale, Amazon.com had received orders for 2.3 million books that had to be shipped out worldwide. Imagine the demand this record sale created on the online servers, the back-office applications, and the shippers!

## 14.3 Databases

The management and organization of data have always been important problems. It is likely that a strong impetus for the development of written language was the need to record commercial transactions ("On this day Procrastinus traded Consensius 4 sheep for 7 barrels of olive oil"). From there it is only a short step to recording inventories ("Procrastinus has 27 sheep"), wages paid, profits gained, and so on. As the volume of data grows, it becomes more difficult to keep track of all the facts, harder to extract useful information from a large collection of facts, and more difficult to relate one fact to another. With the 1890 U.S. census (Chapter 1), Herman Hollerith demonstrated the advantages that can accrue from mechanizing the storage and processing of large amounts of data.

We talked about the online customer database as part of your expansion into e-commerce, but databases are probably a key part of your business whether you have an online presence or not. You have a set of data to maintain about your employees (names, addresses, pay rates, Social Security numbers, etc.), another set of data to maintain about your suppliers (names, addresses, products, orders, etc.), and yet another set of data to maintain about your business itself (sales, expenses, taxes, etc.). Previously, such items of data were recorded by hand, but they are now maintained in electronic databases. The important thing about an electronic database is that it is more than a storehouse of individual data items; these items can easily be extracted, sorted, and even manipulated to reveal new information. To see how this works, let's examine the structure of a file containing data.

### 14.3.1 *Data Organization*

As we learned in Chapters 4 and 5, the most basic unit of data is a single **bit**, a value of 0 or 1. A single bit rarely conveys any meaningful information. Bits are combined into groups of eight called **bytes**; each byte can store the binary representation of a single character or a small integer number. A byte is a single unit of addressable memory. A single byte is often too small to store meaningful information, so a group of bytes is used to represent a string of characters—say, the name of an employee in a company or a larger numerical value. Such a group of bytes is called a **field**. A collection of related fields—say, all the information

about a single employee—is called a **record**, a term inherited from the pencil and paper concept of "keeping records." Related records—say, the records of all the employees in a single company—are kept in a **data file**. (*File* is another term inherited from the familiar *filing cabinet*.) And finally, related files make up a **database**. Thus,

Bits combine to form bytes.
Bytes combine to form fields.
Fields combine to form records.
Records combine to form files.
Files combine to form databases.

Figure 14.3 shows this hierarchical organization of data elements. (This figure was drawn to look neat, but files in a database are almost never all the same size or "shape.")

Bits and bytes are too fine a level of detail for what we will discuss in this section. Also, for the moment, let's assume for simplicity that the database consists of only a single file. Figure 14.4 illustrates a single file made up of five records (the rows), each record composed of three fields (the columns). The various fields can hold different types of data. One field in each record might hold character strings; another field in each record might hold integer data.

Each record in a file contains information about an item in the "universe of discourse" that the file describes. In our example, we assume that the universe of discourse is the set of employees at Rugs-For-You and that each record corresponds to a single employee. An individual employee record, with six different fields, is shown in Figure 14.5. Here it is clear that the *LastName* and *FirstName* fields hold character strings. The type of data being stored in the *ID* field is not clear to us as human beings from looking at the record; they could be numeric data, but because they are unlikely to be involved in computations, they could also be character string data. The data type must be specified when the file is created.

**FIGURE 14.3**
*Data Organization Hierarchy*

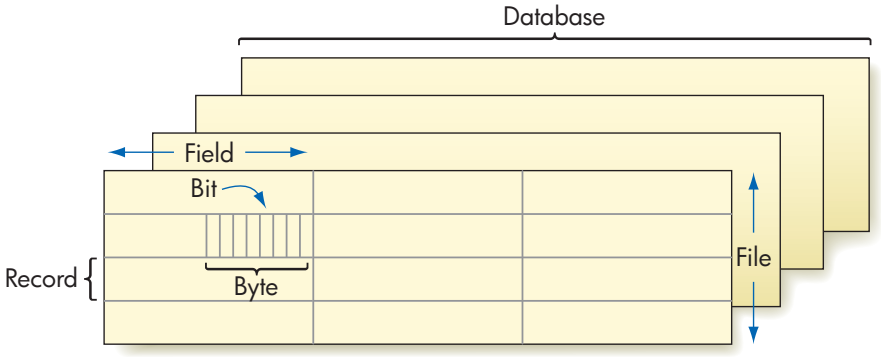



**FIGURE 14.4**
*Records and Fields in a Single File*

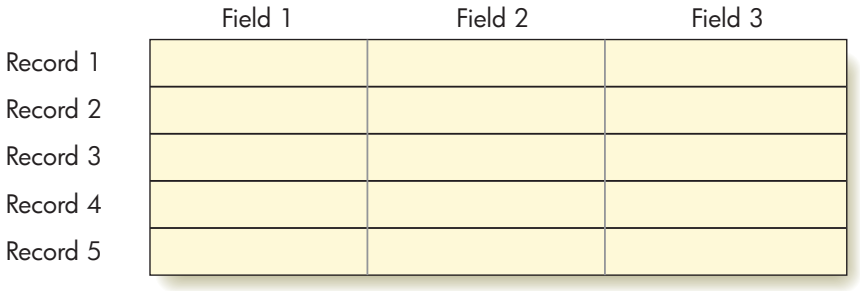


FIGURE 14.5

*One Record in the Rugs-For-You Employees File*

| ID | LastName | FirstName | Birthdate | PayRate | HoursWorked |
|---|---|---|---|---|---|
| 149 | Takasano | Frederick | 5/23/1966 | $12.35 | 250 |

## 14.3.2 Database Management Systems

A **database management system (DBMS)** manages the files in a database. We know that such files actually consist of collections of individual records. However, Edgar F. Codd (mentioned in Chapter 12 as a Turing Award winner for his work in database management systems) proposed the conceptual model of a file as simply a two-dimensional table. In this **relational database model**, the *Employees* file at Rugs-For-You would be represented by the *Employees* table of Figure 14.6.

With the change from records in a file to a conceptual table representing data come some changes in terminology. The table represents information about an **entity**, a fundamental distinguishable component in the Rugs-For-You business—namely its employees. A row of the table contains data about one instance of this entity—that is, one employee—and the row is called a **tuple** (in Figure 14.6, each row is a 6-tuple, containing six pieces of information). How the tuples (rows) are ordered within the table is not important. Each category of information (*ID*, *FirstName*, and so on, in our example) is called an **attribute**. The heading above each column identifies an attribute. The table thus consists of tuples of attribute values. (In other words, in the relational model, files are thought of as tables, records as tuples, and fields as attributes.) A **primary key** is an attribute or combination of attributes that uniquely identifies a tuple. In our example, we are assuming that *ID* is a primary key; *ID* is underlined in the heading in Figure 14.6 to indicate that it is the primary key for this table. The Social Security number is often used as a primary key to uniquely identify tuples that involve people. Obviously, neither *LastName* nor *FirstName* can serve as a primary key—there are many people with the last name Smith and many people with a first name of Michael or Judith.

The computer's operating system functions as a basic file manager. As we learned in Chapter 6, the operating system contains commands to list all of the files on the hard drive, to copy or delete a file, to rename a file, and so forth. But a database management system, unlike a simple file manager, works at the level of individual fields in the individual records of the file; in more appropriate terminology, we should say that it works at the level of individual attribute values of individual tuples in the relational table. Given the *Employees* table of Figure 14.6, a database management system could be given the instruction shown on the page below.

FIGURE 14.6

*Employees Table for Rugs-For-You*

Employees

| ID | LastName | FirstName | Birthdate | PayRate | HoursWorked |
|---|---|---|---|---|---|
| 116 | Kay | Janet | 3/29/1956 | $16.60 | 94 |
| 123 | Perreira | Francine | 8/15/1987 | $ 8.50 | 185 |
| 149 | Takasano | Frederick | 5/23/1966 | $12.35 | 250 |
| 171 | Kay | John | 11/17/1954 | $17.80 | 245 |
| 165 | Honou | Morris | 6/9/1988 | $ 6.70 | 53 |

```
SELECT ID, LastName, FirstName, Birthdate, PayRate,
  HoursWorked
FROM Employees
WHERE ID = 123;
```

This command asks the system to retrieve all the information about the employee with ID 123. Because *ID* is the primary key, there can only be one such employee, and this is a relatively easy task. But the following request to locate all the information about an employee with a given last name,

```
SELECT ID, LastName, FirstName, Birthdate, PayRate,
  HoursWorked
FROM Employees
WHERE LastName = 'Perreira';
```

is done just as easily, even though the *LastName* attribute may not uniquely identify the tuple. If multiple employees in the table have the same name, all of the relevant entries will be returned.

If only some of the attributes are wanted, an instruction such as

```
SELECT LastName, PayRate
FROM Employees
WHERE LastName = 'Perreira';
```

produces just the last name and pay rate for the employee(s) with the given name.

Database management systems usually require specialized **query languages** to enable the user or another application program to **query** (ask questions of) the database, in order to retrieve information. The three preceding SELECT examples are written in a language called **SQL**, Structured Query Language. We briefly discussed SQL in Chapter 10.

To appreciate the power of SQL, consider the following simple SQL queries for more complicated tasks:

```
SELECT *
FROM Employees
ORDER BY ID;
```

This query says to retrieve all of the attribute values (the asterisk is shorthand for listing all attributes) for all the tuples (because there is no further qualification) in the *Employees* table sorted in order by *ID*. Thus, we have effectively sorted the tuples in the relational table using a single command. This is a significant gain in productivity over the step-by-step process of comparing items and moving them around used in the sorting algorithm in Chapter 3. (Of course, what has happened internally is that SQL has invoked its own sorting algorithm. However, the user is shielded from the details of this algorithm and is allowed to work at a more abstract level.) The query

```
SELECT *
FROM Employees
WHERE PayRate > 15.00;
```

gets all the tuples for employees above a certain pay rate. Here we've searched all the tuples on a particular attribute, again without having to specify all of the details, as we had to do when writing the sequential search or binary search algorithms of Chapter 3.

To manage a relational table, you must be able to add new tuples to the table (which is how the existing tuples got into the table in the first place), delete tuples from a table, and change information in an existing tuple. These tasks are easily handled by the SQL commands INSERT, DELETE, and UPDATE.

In order to explore further the power of a DBMS, let's expand our Rugs-For-You database to include a second relational table. The *InsurancePolicies* table shown in Figure 14.7 contains information on the insurance plan type and the date of issue of the policy for an employee with a given ID.

In the *InsurancePolicies* table, there is a **composite primary key** in that both *EmployeeID* and *PlanType* are needed to identify a tuple uniquely, because a given employee may have more than one insurance plan (e.g., both health and disability insurance plans). It is also true that an employee may have no plan; in Figure 14.7, there is no tuple with ID 116, although there is an employee with ID 116. Each value of *EmployeeID* in the *InsurancePolicies* table exists as an *ID* value in a tuple of the *Employees* table, where it is a primary key. Because of this, the *EmployeeID* attribute of the *InsurancePolicies* table is called a **foreign key** into the *Employees* table. This foreign key establishes the relationship that employees may have insurance plans.

The database management system can relate information between various tables through these key values—in our example, the linkage between the foreign key *EmployeeID* in the *InsurancePolicies* table and the primary key *ID* in the *Employees* table. Thus, the following query will give us information about Frederick Takasano's insurance plan, even though Frederick Takasano's name is not in the *InsurancePolicies* table:

```
SELECT LastName, FirstName, PlanType
FROM Employees, InsurancePolicies
WHERE LastName = 'Takasano'
AND FirstName = 'Frederick'
AND ID = EmployeeID;
```

The query is an instruction to retrieve the *LastName* and *FirstName* attributes from the *Employees* table and the *PlanType* attribute from the *InsurancePolicies* table by looking for the tuple with *LastName* attribute value "Takasano" and *FirstName* attribute value "Frederick" in the *Employees* table, and then finding the tuple(s) with the matching *EmployeeID* value in the *InsurancePolicies* table. (Here is the Boolean AND operation we encountered in Chapter 4 in our discussion on Boolean logic.) It is the last term in the WHERE clause of the query

**FIGURE 14.7**

*InsurancePolicies Table for Rugs-For-You*

INSURANCEPOLICIES

| EMPLOYEEID | PLANTYPE | DATEISSUED |
|---|---|---|
| 171 | B2 | 10/18/1974 |
| 171 | C1 | 6/21/1982 |
| 149 | B2 | 8/16/1990 |
| 149 | A1 | 5/23/1995 |
| 149 | C2 | 12/18/1999 |

(the last line) that causes the two tables to be joined together by the match between primary key and foreign key. The result of the query is

```
Takasano Frederick B2
Takasano Frederick A1
Takasano Frederick C2
```

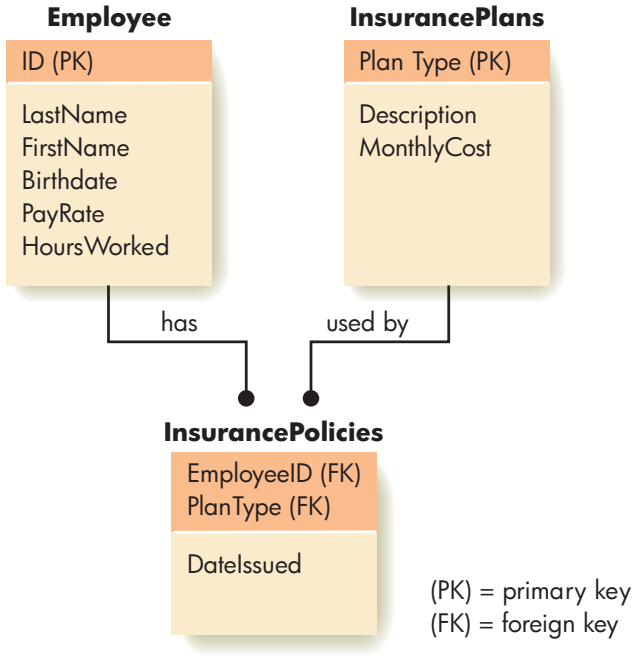
The correspondence between primary keys and foreign keys is what establishes the relationships among various entities in a database. The SQL command to create a table requires specification of the various attributes by name and data type, identification of the primary key, identification of any foreign keys, and identification of the tables into which these are foreign keys. This information is used to build the actual file that stores the data in the tuples.

We've now done a fairly complex query involving two different tables. It is easy to see how these ideas can be expanded to multiple tables, linked together by relationships represented by foreign keys and their corresponding primary keys. Figure 14.8 shows an expansion of the Rugs-For-You database to include a table called *InsurancePlans* that contains, for each type of insurance plan, a description of its coverage and its monthly cost. *PlanType* is the primary key for this table. This makes *PlanType* in the *InsurancePolicies* table a foreign key into the *InsurancePlans* table, as shown in Figure 14.8. This linkage would allow us to write a query to find, for example, the monthly cost of Mr. Takasano's insurance (see Practice Problem 2 at the end of this section).

Using multiple tables in a single database reduces the amount of redundant information that must be stored. For example, a stand-alone insurance file for Rugs-For-You employees would probably have to include employee names as well as IDs. It also minimizes the amount of work required to maintain consistency in the data (if Francine Perreira gets married and changes her name, the name change need only be entered in one place). But most important of all. the database gives the user, or the user's application software, the

**FIGURE 14.8**

*Three Entities in the Rugs-For-You Database*



ability to combine and manipulate data easily in ways that would be very difficult if the data were kept in separate and unrelated files.

As we have seen by looking at some queries, SQL is a very high-level language in which a single instruction is quite powerful. In terms of the language classifications of Chapter 10, it is also a nonprocedural language. A program written in SQL merely asks for something to be done (sort all tuples in some order, search all tuples to match some condition); it does not contain a specific sequence of instructions on *how* it is to be done.

### 14.3.3 *Other Considerations*

Performance issues definitely affect the user's satisfaction with a database management system; a slow response to a query is at best annoying and at worst unacceptable. Large files are maintained on disk in secondary storage rather than being brought in total into main memory. Accessing a record in the file involves at least one disk input/output (I/O) operation, which is a much slower process than accessing information stored in main memory, sometimes as much as three or four orders of magnitude slower.

In Chapter 5 we talked about the three components that contribute to reading an individual disk sector into memory or writing from memory to a disk sector: seek time (time to position the read/write head over the correct track on the disk), latency (time for the correct sector to rotate under the read/write head), and transfer time (time to read from or write to the entire sector). Organizing the way that records are stored on the disk can help to minimize the access time by reducing the number of disk I/O operations that must be done before finding the sector containing the desired record. For example, assume that we have a database that occupies 30 sectors on our disk, and there are 15 sectors per track. It would make the most sense to store the information on surface 0, track 0, sectors 0–14 and on surface 1, track 0, sectors 0–14. Using the same track on different surfaces means that the head does not have to move to a different track to obtain the data, and the seek time is always 0.

Also, creating additional records to be stored along with the file, although consuming extra storage, can significantly reduce access time. This works much like a library catalog system. To access a book, the user first consults a smaller structure that is organized in a useful way (alphabetically), and that directs the user to the desired book. The smaller structure stored with the file may even be organized in a treelike manner that is a generalization of the tree structure we used in Chapter 3 to visualize the binary search. Following the branches of the tree can quickly lead to information about the location in the file of the record with a particular primary key value. A good DBMS incorporates the services of a sophisticated file manager to organize the disk files in an optimal way, in order to minimize access time to the records.

**Distributed databases** allow the physical data to reside at separate and independent locations that are electronically networked together. The user at site A makes a database query that needs access to data physically stored at site B. The database management system and the underlying network make the necessary links and connections to get the data from where it is currently stored to the node where it is needed. To the user, it looks like a single database on his or her own machine, except perhaps for increased access time when data have to travel across a network.

## Think Big!

The world's largest database can be found at the World Data Center for Climate (*http://www.ngdc.noaa.gov/wdc/europe/climate.html*). One of 52 Centers in 12 countries making up the World Data Center System, the WDCC is located at the Max Planck Institute for Meteorology in Hamburg, Germany. Its mission is to collect, examine, and disseminate data related to climate change on all time scales, particularly data from scientific climate modeling (remember the discussion on climate modeling in Chapter 13).

In 2007 the WDCC database held about 340 terabytes of data! That's 340,000,000,000,000 bytes. A standard data DVD holds 4.7 GB, so this database represents the content of more than 72,000 DVDs. In addition to this huge amount of data available online to scientists from around the world, the WDCC also maintains 6 petabytes (6,000 terabytes) of additional data on magnetic tape.

If a database management system can easily make connections among different files, and even among data stored at different locations, how difficult is it to electronically link information in the IRS database with information in the FBI database, the Social Security database, credit card databases, banking databases, and so on? Obviously, it would not be difficult, using the technology that we have described in this chapter. Building these types of massive, integrated government databases raises fewer technical questions than legal, political, social, and ethical ones. Remember that even the online customers of Rugs-For-You want assurances as to how their personal information is used.

## PRACTICE PROBLEMS

1. Using the *Employees* table of Figure 14.6, what is the result of the following SQL query?

```
SELECT ID, PayRate
FROM Employees
WHERE LastName = 'Takasano';
```

2. Complete the following SQL query to find the monthly cost of Frederick Takasano's insurance; because *PlanType* is an attribute of both *InsurancePolicies* and *InsurancePlans*, we have to include the table name as well.

```
SELECT LastName, FirstName, _____
FROM Employees, InsurancePlans, InsurancePolicies
WHERE LastName = _____
AND ID = EmployeeID
AND InsurancePolicies.PlanType = _____;
```

3. Using the *InsurancePolicies* table of Figure 14.7, write an SQL query to find all the employee IDs for employees who have insurance plan type B2.

## LABORATORY EXPERIENCE 19

If you have a commercial database package available, you can work through the exercises in this laboratory experience using an expanded Rugs-For-You database. You will write SQL queries similar to the ones discussed in this section and will also learn how to use SQL to carry out some computations.

In general, issues of personal privacy and public safety are magnified enormously by the capabilities of networked databases. We'll discuss approaches to these and other ethical issues in Chapter 17.

## 14.4 Conclusion

In this chapter we've looked at e-commerce, a highly popular application of computing. We've learned that there is much more involved in a retail Web business than simply creating a Web page, and that technical areas of computer science such as information security and databases play a critical role.

In Chapter 15 we will look at another application of computer science, one that has long captured the public's attention through its depiction in science-fiction literature and movies—artificial intelligence.