Main content

Chapter

14

Ecommerce, Databases, and Data Science

Change font sizeMain content

Chapter Contents

Chapter Introduction

r studying this chapter, you will be able to:

Explain the issues, pros, and cons of whether to create an ecommerce website
List the components of online transactions, and explain what is involved in each step
List one other ecommerce model besides the online retailer selling to the general public and explain how it works
List one other electronic payment system besides traditional credit cards and explain how it works
Describe the relational database model
Frame simple queries in SQL to retrieve information from one or more tables in a relational database
Define data mining
List three sources data brokers use to collect data on individuals

Give an example of data science used "for the greater good"

Main content

Chapter Contents

14.1Introduction

As mentioned in Chapter 7, the Internet has been around for quite a while (since 1969), but it did not have much of an 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 "in the cloud."

These days, if you own just about any type of business, you need to have a web presence. For example,

- Your business provides a service, such as landscaping, that does not sell products directly to retail customers. However, you use the web for advertising.
- Your business provides a service for which follow-up information is important. For example, you are a shipping company and you use your website to allow customers to track their shipments.
- Your business provides a service that enables customers to engage in online transactions, such as banking, that are not retail sales.
- 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 company is a retail business, and you maintain a *B2C* (*business-to-consumer*) website. You do this to advertise your products and to allow the general public to shop and 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.

Databases are an important component of any ecommerce business, in fact of any business. We'll discuss databases in more detail later in this chapter.

Then, we'll look at the relatively new world of *data science*: some of the tools it uses, privacy considerations that arise because information about us is likely to be stored in many databases, and finally some examples of how the use of data science can improve, or even save, lives.

Main content

Chapter Contents

14.2 Ecommerce

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 salespeople. In addition to your traditional store, you have decided to establish a web presence for your business where customers can visit, view area rugs for sale, ask questions, make a selection, purchase a rug, and arrange to have it delivered to them, all in a quick, easy, and secure electronic environment. In other words, you have decided to expand your retail business into the **ecommerce** world. In this 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.

Shopping on the Web

The Census Bureau of the U.S. Department of Commerce estimated ecommerce retail sales in the United States for the first quarter of 2017 to be $105.7 billion, an increase of 14% from the first quarter of 2016. In the first quarter of 2017, ecommerce sales accounted for 8.5% of total U.S. retail sales as opposed to just 1% in 2001. The growth in ecommerce sales, and its increasing percentage of total retail sales, continues unabated. This trend is having an effect on traditional brick-and-mortar stores. In just the first quarter of 2017, well-known retail brands such as Radio Shack, JCPenney, Macy's, Sears, and many others announced store closings as a result of reduced store traffic and profitability.

Fun Facts: During the 2016 holiday shopping season, November 1–December 19, Amazon shipped over a billion items. It sold a watch every 1.5 seconds, enough running shoes to run 18,603 times around the globe, and enough KitchenAid mixers to make 7.5 million cookies all at the same time.

Main content

Chapter Contents

14.2.1 Decisions, Decisions

In Chapter 10, we talked about HTML, the language used to build webpages. As a small business owner, you might not know much about HTML, to say nothing of the many other technologies used in creating webpages, such as XML (see the Special Interest Box "Beyond HTML" in Chapter 10). However, you can hire someone who knows these technologies, ask that person to put together some webpages for your store, and presto, you're in ebusiness! 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.

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 objective you are trying to achieve? 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 or functions within your existing business, so that the shipping department and the accounting department, for example, work off the same order form?

Any or all of these might be legitimate reasons for moving into ecommerce, 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 website? 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 website? If not, what will it cost to acquire new personnel or retrain existing personnel?
- Do you have the legal expertise on your staff 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?
- Will you have adequate security to protect sensitive online data from hackers who will attempt to steal information such as credit card numbers? (In 2013, Target suffered one of the biggest retail hacks to date, with the theft of personal data from 70 million customer accounts plus 40 million customers’ credit and debit card data. If it can happen to the likes of Target, it can happen to Rugs-For-You.)

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?

Once you decide to move into the ecommerce arena, there are still many questions to be answered and decisions to be made. The first major decision is choosing between *in-house development* (doing the work within your own company), *outsourcing* (hiring an outsider to do the work), or, for small retail businesses, using an off-the-shelf software package designed to host an online retail enterprise right out of the box. In fact, this is not a single decision but a whole host of decisions.

- *Personnel*—Are you going to use your existing staff to develop this ebusiness, either because they already have the necessary skills or because they will be retrained? Will you hire new personnel with the needed skills? Or will you turn the entire job over to an **ASP (application service provider)** who, for a fee, will design your website and manage it or host it on an ongoing basis?
- *Hardware*—You will need at least one web server machine to host your website. You may need additional computers to store your customer database information, to support program development, to provide backup capabilities in case of hardware malfunctions, 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? Or will you use a cloud computing service, which can supply computer assets that expand or shrink according to your needs?
- *Software*—You will also need a substantial amount 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 commercial software or develop your 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. Similarly, if you decide to use off-the-shelf software, you have no further software decisions beyond choices the software package may allow.

Change font sizeMain content

Chapter Contents

14.2.2 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, 365 days a year. 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 website to another. But this also means that your competition is just a mouse click or a finger tap 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 just mentioned.

Figure 14.1A typical online transaction in nine steps

Step 1

Getting There

How can you get customers to your website? Technically, once the customer enters the web address (the URL - Uniform Resource Locator) into his or her browser's address bar, the process to reach your website works exactly as described in Section 7.3.5. But how does your potential customer learn your URL in the first place? There are many possibilities:

Conventional advertising—You post your homepage 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 homepage 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, called *registrars*, 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 browsers can find your IP address and get to your site. A list of these accredited registrars can be obtained from www.internic.net/regist.html. A number of websites 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 website.

Search engine—Potential customers may use an Internet search engine (such as Google or Bing) to search for websites about products that you sell. For example, in response to a search using the words "rugs" and "retail sales", your company's website might turn up in the list of pages returned. You can also pay for a "sponsored link" so that a search on appropriate keywords will bring up links to your website 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 webpage with links to other webpages 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. You would certainly want to have a link to Rugs-For-You from this portal page, and you might even want to purchase a *banner ad* (a graphical ad, often with animation, placed in a prominent position on a webpage) so that anyone who goes to this portal sees the rugs-for-you.com link right away.

A Rose by Any Other Name…

Cybersquatting, also called *domain squatting*, is the practice of registering a domain name that uses the name or trademark of an existing business, with the intent to resell 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 in the United States. International disputes may be brought before the World Intellectual Property Organization (WIPO), a United Nations agency. In 2016, the WIPO heard over 3000 cases, a record number and a 10% increase over 2015. Since beginning this practice in 1999, the WIPO has received over 36,000 cases involving over 66,000 domain names. More than 80% of the cases settled in WIPO in the past five years were decided in favor of those who charged they were the victims of cybersquatting.

In June 2011, ICANN announced a new policy for "personalized" domain names. Top-level domain names, that is, the suffixes at the end of URLs, had been limited to two-character country codes plus the familiar .com, .org, .edu, and so forth. The new policy allowed for suffixes that are brand-specific or industry-specific names. This opened the door to over 1900 applications in 2012, and by 2016 there were over 1200 approved new top-level domain names, including .hotels, .wine, and .rugby. Disputes over these domain names have formed a growing percentage of the WIPO cases, 16% in 2016.

In 2013, future-president Donald Trump sued a Brooklyn man who had registered four domain names: trumpmumbai.com, trumpindia.com, trumpbeijing.com, and trumpbudhabi.com and had developed actual websites using those domain names. Trump sought $100,000 in damages for each of the four domain names and in 2014 was awarded total damages of $32,000, not by the WIPO but by the U. S. District Court in Brooklyn. (There actually is a Trump Tower in Mumbai, and if you enter trumpmumbai.com in your web browser, you will now reach the Trump Organization homepage showing a picture of this building.)

Step 2

Do I Know You?

Regular customers at your traditional store are treated with special care. You might 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 heart and soul 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 and what the customer bought. What the return customer sees is tailored to reflect this information.

Other sites that do not require a customer login might still greet the customer with "Welcome, John," for example, and arrange a webpage with items tied to John's apparent interests, based on his last purchase. This type of website 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 computer or handheld device. 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—"Hello John, we are having a sale on new area rugs." 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 cart and return at a later time to find them still there. 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). In addition, you should provide links to Facebook, Twitter, Pinterest, or other appropriate social media sites for further interaction with your customers. And certainly you should provide a toll-free number where your customers can speak with a real, live person.

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 or tap of a finger. Your website 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 webpage so much more than just an HTML programming exercise! Let’s assume that a customer has successfully navigated your website, 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 a secure environment 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. Although encryption provides for the secure transmission of data, this is of little use if the data is 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 you learned that the web server can pass to the browser a certificate of authentication issued by a trusted third party.

However, these behind-the-scenes security measures do nothing to reassure the customer. The website can display a visual seal assuring the customer that the site has been authenticated and meets high security standards. A secure webpage has the protocol heading *https* in the address bar, rather than *http*, with the *s* signifying a site under the protection of SSL. Customers may also see a little green padlock graphic on the webpage to indicate a secure site, and the browser address bar may turn green. Most customers won’t go any farther than this, but hovering the mouse over the lock or the green area may display a tag that reads, for example, “verified by: Verisign, Inc.”* Clicking or tapping the lock or the green area will display information similar to Figure 14.2; selecting More Information tells you whether you have visited this website previously, technical details on the encryption used, and so forth.

Figure 14.2 Secure site assurance

Steps 4 and 5

Payment Processing

Let’s assume that your customers will pay with credit cards, the most common online option. (Various other modes of online payment are discussed later in this chapter.) 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 email 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 email confirmation is sent to the customer. To use this option, you must maintain customer credit card information.

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).

ChangMain content

Chapter Contents

14.2.3 Designing Your Website

Your website 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. One of your earliest decisions is your website *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 rooms in the house.

Your customers should always know where they are on your website. As we mentioned in Chapter 7, hypertext allows a user to move easily from page to page by simply clicking or tapping 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 (that is, 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 website to any other page in four clicks or fewer. And your webpages should include the ability to search the site for a specific item, either by name or by product number.

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 press or click. It is also important to give customers the option to go back and change something or to clearly indicate to customers 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 email to confirm orders, and send follow-up emails or text messages when orders are shipped.

You may also want to offer extras to your customers. Put up a *FAQ* (*frequently asked questions*) page and links to contact customer service, review new products, or connect with other customers. Allow customers to track their shipment with 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 website 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 webpages, your site must adhere to good design principles. It must look professional and uncluttered. 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.

Your webpages need to be designed to be displayed on different machines (desktops, laptops, tablets, phones) with different operating systems and browsers (Internet Explorer, Safari, Chrome, Firefox, Edge). 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. 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. Adhere to ADA (Americans with Disabilities Act) requirements for web accessibility. One of the most common issues in webpage accessibility relates to images, charts, and photographs. Blind users or users with low vision have several assistive technologies available to them, but these technologies can only read text. Therefore a visual element on a webpage needs a corresponding text tag in the HTML code so that the browser will display text describing the image.

As you can see from our brief discussion, designing webpages, or at least a successful set of commercial webpages, is a difficult and complex task. It involves not only computer science skills (HTML, XML, HTTP, TCP/IP, networking, databases), but also 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 webpage, but much more difficult to create a really good one.

Less Is More

Good webpage design does not necessarily involve complexity. The goal should be a page that is uncluttered, clean, attractive, informative, and easy to use. The Google home page, one of the best-known—and most effective—webpages, is a model of striking simplicity:

Finally, in addition to your own business website, you need a presence on social networking sites. Create a Facebook page, post on Twitter and YouTube, and put links on your own website so people can easily find, follow, friend, like, watch, and tweet you. People now expect to be able to post user reviews, provide product feedback, and share opinions and experiences. Companies gain valuable information through this social media process, which used to be obtained through more costly invitation-only “focus groups.”

Main content

Chapter Contents

14.2.4Behind the Scenes

Your business maintains a number of other computer applications in addition to your online order-entry system. In Figure 14.1, you 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 website), whereas others may be legacy code that has been around for dozens of years.

Practice Problems

Locate a portal page for at least one of the following topics: health care, environmental issues, fantasy sports, higher education, or the alternative energy industry.

Answer

For example:

www.webmd.com
https://en.wikipedia.org/wiki/Portal:Environment
https://www.fantasysp.com
www.petersons.com
https://www.dawnbreaker.com/portals/altenergy

Take a look at the website of a major online retailer such as Amazon.com, Apple.com, or Walmart.com and identify some characteristics of its site that you find helpful and some things that you find annoying or troublesome.

Answer

As an alternative, students could be asked to compare two such sites.

Obviously, these systems are not all independent of one another, and some must collaborate quite closely. 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 website. Because of this, once the website is up and running, you may need to invest in **middleware**—software that allows separate, existing programs to communicate and work together seamlessly. Middleware does such things as translate between incompatible data representations, file formats, and network protocols to allow otherwise disparate systems to exchange information. Think of middleware as a “translator” that allows for meaningful communications between, say, one businessperson who speaks just Chinese and another fluent only in Turkish.

Finally, as soon as you have your enterprise humming along smoothly as an ecommerce 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? Will you be able to survive a massive natural disaster? What will you do if a hacker breaks into your website 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.

Change font sizeMain content

Chapter Contents

14.2.5 Other Ecommerce Models

The ecommerce model we have been discussing is that of an online retailer selling products or services over the web to the general public, but there are other models.

eBay. One of the most successful alternative ecommerce models is *eBay*, founded in 1995 and now a huge international business. As of April 2017, eBay had almost 169 million active registered users, and at any given time there are over 800 million items on sale. The most expensive item ever purchased was a 405-ft luxury yacht, which was purchased in 2006 by a Russian billionaire for $168 million. One of the weirdest was a cornflake in the shape of the state of Illinois, which sold in 2008 for $1,350.

Unlike the traditional retailer/customer relationship, eBay facilitates peer-to-peer relationships in an “auction” mode. Anyone can post an item to be sold at auction on eBay. The item must conform to eBay’s policies (for example, it can’t be a prescription drug, stolen property, or used cosmetics). Restrictions for this multinational company are aligned with regional or country laws. The seller selects the appropriate category for the item, writes a description, includes shipping details and costs, and sets the opening bid price and duration of the bidding period. eBay collects a fee to list an item and a second fee if the item sells.

As a buyer on eBay, you see the current maximum bid for an item and the time left for bidding. You enter the maximum amount you are willing to pay, which is not revealed to the seller or to other bidders, and the system raises your bid just enough to make you the highest bidder, up to your maximum price. If the bidding exceeds your maximum price, you are notified that you have been outbid and you have an opportunity to enter a new maximum price. If the current maximum bid is yours and there is no further bidding, then you have purchased the item, possibly at a lower price than your maximum price. Safeguards are in place to protect both the buyer and the seller, for example, seller ratings, privacy policies, and standardized electronic payment mechanisms. A “fixed-price” selling option is also available.

eBay was a precursor to the growing **peer-to-peer (collaborative consumption) economy** that now includes companies such as Uber, Lyft, Airbnb, and others.

Craigslist. Craigslist is an online classified ad site. Actually, it is a network of local sites for various cities or areas. Each such site gives postings for local items for sale, job opportunities, housing options, personal ads, discussion forums, and so on. Begun in 1995 as a modest list of San Francisco events circulated to friends of the founder, Craig Newmark, Craigslist now has over 700 local sites in 70 countries, from Ahmedabad, India, to Zamboanga in the Philippines. Craigslist users post about 80 million new classified ads each month, and the sites receive about 50 billion page views per month. Craigslist supports 13 languages: Catalan, Danish, Dutch, English, Filipino, French, German, Italian, Norwegian, Portuguese, Spanish, Swedish, and Turkish. Unfortunately, there have been several cases of crimes committed and scams perpetrated based on contacts established through Craigslist (the Craigslist online community site includes information about how to detect and avoid scams, www.craigslist.org/about/scams), but the overall effect is a sense of local community and people-to-people trust.

Groupon. Groupon has some similarities to Craigslist, but is less peer-to-peer in nature. The name is a shortened form of “group coupon.” Groupon’s first site, in Chicago, was launched in November 2008. As of March 2017, it served over 500 local markets in 15 countries and had sales in 2016 of $3.1 billion. Here's how it works: A local business offers a coupon through Groupon for a great deal—on museum admission, a spa session, a restaurant meal, or whatever. The coupon offer is featured on the local Groupon site for a single day. The business specifies a minimum number of customers who must purchase the coupon. If that number is not met, the deal is off; no one gets a coupon and no one gets charged. If that minimum is met, the deal is on and additional customers can purchase the coupon. Coupons are emailed or texted the next day to customers who purchased them. Groupon splits the coupon charge with the business, so the business spends no out-of-pocket money to advertise unless a minimum number of customers is already guaranteed.

Change font sizeMain content

Chapter Contents

14.2.6 Electronic Payment Systems

In addition to new models of ecommerce, there are electronic payment systems that are alternatives to a customer paying an online merchant by a traditional credit card.

PayPal. To use this online payment service, the customer must have a PayPal account that is tied to a credit card, debit card, or bank account; the PayPal account can be funded by a bank transfer. If a site accepts payment via PayPal, then the customer can choose to pay from his or her PayPal account balance, or from the associated bank account or credit card. PayPal provides other services as well, such as the ability to securely send money to someone else who has a PayPal account.

As of the first quarter of 2017, PayPal had 203 million active registered accounts and was available in more than 200 markets around the world. It supports transactions in multiple currencies, such as the U.S. dollar, the Polish zloty, and the Thai baht. In 2016, PayPay processed 6.1 billion transactions, averaging out to about 193 transactions per second.

Apple Pay. Apple Pay, and similar systems such as Android Pay and Samsung Pay, are mobile payment systems. First announced by Apple in 2014, Apple Pay is a *digital wallet*. To get started, on newer versions of the Apple iPhone or on the Apple Watch, you open the Wallet and add your credit card information. Apple contacts your bank to see that this is a valid credit card issued to you, and that completes the setup process. To use Apple Pay at a merchant with a contactless card reader, just hold your phone or watch near the reader and when an image of your card appears on the screen, use Touch ID or your passcode to authenticate your transaction. An increasing number of banks/credit cards and merchants accept Apple Pay.

Apple Pay, and similar systems, put a big focus on security. The credit card information you entered in the setup process does not stay on your phone; instead Apple Pay uses that information to create a device-specific account number that is stored on your phone. Then when you use Apple Pay for a purchase, Apple creates a one-time payment number and a dynamic security code that are both encrypted and sent for processing to the bank issuing your credit card. The merchant never sees information about your credit card, you don't have to sign anything, and Apple keeps no record of what you bought, from whom you bought it, or what you paid for it.

Bitcoin. Bitcoin is more than a payment system. Rather it is a form of money—sort of. Consider the U.S. dollar. It was originally based on the gold standard, meaning that a certain amount of gold could be redeemed for your dollar bill, the idea being that the gold was what made your dollar bill valuable, and therefore useful for the payment of debt or the purchase of goods. In 1971, the United States dropped the gold standard, and today your dollar bill is valuable only because the U.S. government says it is, and that it must be accepted as "legal tender" within the United States. Only the U.S. government can print U.S. dollar bills or any other U.S. currency.

In contrast, bitcoin is never printed, it isn't highly regulated by any country or other entity such as the Federal Reserve Board or the European Central Bank, and no entity guarantees its value. Yet you can buy and sell bitcoin at an online bitcoin exchange service and you can use your bitcoin to make purchases from the increasing number of online businesses that accept bitcoins as a medium of payment. How does that work? Bitcoin is accepted as legitimate currency in large part because of the security obtained from two mechanisms we discussed in Chapter 8, namely, *public-key encryption* and *hashing*.

The idea for bitcoin originated in 2008 in a report by an anonymous individual(s) using the pseudonym Satoshi Nakamoto. Bitcoin is managed by a decentralized network of computers. All bitcoin transactions are recorded in a public ledger called the *blockchain*, which frequently gets updated and distributed to the bitcoin network. Each user has one or more anonymous bitcoin addresses that are actually public keys, with associated private keys. Obtaining a private key from its corresponding public key is not computationally feasible.

Suppose that *X* has some amount of bitcoin available and wants to transmit that to *Y* (perhaps to a merchant as payment for a purchase). *X* constructs a transaction that includes the value *v* in bitcoin of the transaction, the public key for *Y*, information from the blockchain about the previous transaction (or transactions) that transferred *v* bitcoins to *X*, and a digital signature that bitcoin software creates using *X*'s private key. *X* then sends the transaction to *Y*'s public key (bitcoin address). *Y* uses another piece of bitcoin software that reads the signature to verify that this transaction comes from the owner of *X*'s public key. This validates the transaction, which then can be broadcast to the entire network.

Now here is where it gets interesting. Individuals (or conglomerates) called *bitcoin miners* pick up new transactions on the network, validate them, and eventually include them in a new block to be added to the blockchain. For this work, the miner is rewarded with additional, newly created bitcoin. This is how bitcoin comes into existence. The new block includes more than just these new transactions; it also includes a small random number and the *block header* of the previous block in the chain, which is the result of applying a standard hash function to that block. Then the same standard hash function is applied to this entire new block to create its block header. Powerful software allows this to be accomplished quickly and easily.

So, what's the work, you ask? The rules are that the miner can only add the new block if the value of its hash is smaller than the value of the block header (the hash) for the previous block; if that's not the case, the miner has to choose a new random number and repeat the hash. The first one to succeed gets to broadcast the new block to the network and, once the network agrees that the block is correct, it gets added to the blockchain and the miner collects the reward. On the average, the blockchain is updated every 10 minutes.

Finally, note that each blockchain header is derived in part from the previous blockchain header. If anyone attempts to tamper with a transaction or add a new transaction to a block, that block's hash value (header) will no longer be correct, and neither will the header of any subsequent block. Anyone can check a block by recomputing its hash value; if the result does not agree with that block's header, then something in that block or an earlier block has been tampered with. In fact, a newly created block is checked by everyone on the network and if it fails that test, it is ignored, not added to the blockchain, and the miner receives no reward.

To sum up, here are the advantages of bitcoin:

- Transactions are anonymous but completely transparent (verifiable).
- There is no "middle man," such as a credit card or a bank, with associated fees.
- The results are immutable (tamper-proof).

Blockchain: A New Revolution?

Blockchain is the basis of bitcoin cybercurrency, which clearly has the potential to revolutionize the banking industry—or perhaps even eliminate banking as we know it. But this relatively new technology is starting to be explored for other uses. Basically, any transfer of assets can benefit from the transparency and immutability of blockchain technology. Think of selling/buying stocks and bonds, valuable art, or real estate without the expense of brokerage firms, auction houses, or real estate companies. Think of transferring commodities, important documents, digital media, or intellectual property. Many people feel that digital access to these distributed networks will open doors to much of the world's population that currently has no access to financial markets of any kind. On the other hand, such developments will mean the loss of many jobs we have today.

Main content

Chapter Contents

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 ecommerce, 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.

Main content

Chapter Contents

14.3.1 Data Organization

As you 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 as an employee ID. 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, 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.")

Figure 14.3 **Data organization hierarchy**

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.

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.

Figure 14.4 **Records and fields in a single file**

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; it could be numeric data, but because it is unlikely to be involved in computations, it could also be character string data. The data type must be specified when the file is created.

Figure 14.5

One record in the Rugs-For-You Employees file

ID	LastName	FirstName	Birthdate	PayRate	HoursWorked
149	Takasano	Frederick	5/23/1990	$12.35	250

Main content

Chapter Contents

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.

Figure 14.6

Employees table for Rugs-For-You

Employees					
ID	**LastName**	**FirstName**	**Birthdate**	**PayRate**	**HoursWorked**
116	Kay	Janet	3/29/1980	$16.60	94
123	Perreira	Francine	8/15/1993	$ 8.50	185
149	Takasano	Frederick	5/23/1990	$12.35	250
171	Kay	John	11/17/1978	$17.80	245
165	Honou	Morris	6/9/1997	$ 6.70	53

With the change from records in a file to a conceptual table representing data comes 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, in relational database terms, 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 order of the attributes (columns) is also not important, but of course must be the same for each tuple in the table. 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. Social Security numbers were previously used as primary keys to uniquely identify tuples that involve people, but because of privacy issues, most employers construct a unique internal identification number for each employee. 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. For example, a simple file system could not tell you specific information about the employee with ID number 123. However, given the *Employees* table of Figure 14.6, a database management system could be given the instruction shown here:

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,

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 tuples will be returned.

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

produces just the last name and pay rate for the employee(s) with the given last 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)**, a special-purpose language used for posing questions to a database management system. We briefly discussed SQL in Chapter 10.

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

This query says to retrieve *all* of the attribute values (the asterisk is shorthand notation 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 by the sorting algorithms in Chapter 3. (Of course, what happens internally is that SQL invokes its own sorting algorithm, perhaps even one of those described in Chapter 3. However, the user is shielded from the details of this algorithm and is allowed to work at a more abstract level.) The query

gets all the tuples for employees above a certain pay rate. Here we've searched all the tuples on a particular attribute without having to specify the details of the search, as we had to do when coding the sequential search or binary search algorithms of Chapter 3. Again, underneath this SQL command the system has invoked a sequential, binary, or other type of search algorithm, but we are insulated from this level of detail and allowed to think at a higher (and more productive) level of abstraction.

To manage a relational table over time, it must be possible 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.

Figure 14.7

InsurancePolicies table for Rugs-For-You

InsurancePolicies		
EmployeeID	**PlanType**	**DateIssued**
171	B2	10/18/1998
171	C1	6/21/2006
149	B2	8/16/2012
149	A1	5/23/2010
149	C2	12/18/2015

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 (for example, both health and disability insurance plans), and a given insurance plan may be held by more than one employee. Both attributes are underlined in the column headings in the figure, showing that they form a composite primary key. It is clear from Figure 14.7 that this composite primary key is needed for the current employees of Rugs-For-You, but even if that were not the case, the design of the database should use this composite primary key because it is a reasonable assumption that it might be needed in the future. 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:

This query uses the Boolean AND operation we encountered in Chapter 4 in our discussion on Boolean logic. The query is an instruction to the database management system 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. It is the last term in the WHERE clause of the query (the last line) that causes the two tables to be joined together by the match between primary key and foreign key.

The result of executing the entire query is

Now let's see how this works. The FROM clause, line 2 in the query, picks out the tables to be used in order to answer the query. (These are the only two tables in the database at the moment, but there could be more.) The rest of the query illustrates the use of three relational database operations:

- **Join**—Match tuples from two different relational tables using the specified attributes. The

part of the query is doing a *join* operation. The result of the join would be the temporary table

ID	LastName	FirstName	Birthdate	PayRate	HoursWorked	EmployeeID	PlanType	DateIssued
149	Takasano	Frederick	5/23/1990	$12.35	250	149	B2	8/16/2012
149	Takasano	Frederick	5/23/1990	$12.35	250	149	A1	5/23/2010
149	Takasano	Frederick	5/23/1990	$12.35	250	149	C2	12/16/2015
171	Kay	John	11/17/1978	$17.80	245	171	B2	10/18/1998
171	Kay	John	11/17/1978	$17.80	245	171	C1	6/21/2006

- **Restrict**—Pick out tuples that meet a certain condition. The

part of the query is doing a *restrict* operation, which has the following effect on the previous table (note that two tuples have been eliminated):

- **Project**—Pick out certain attributes (columns) from a set of tuples. The

part of the query is doing a *project* operation, which has the following effect on the previous table (note that most attribute columns have been eliminated):

leaving

as before.

The correspondence between primary keys and foreign keys is what establishes the relationships among various entities in a database and makes a *join* operation possible. 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).

Figure 14.8 **Three entities in the Rugs-For-You database**

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 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 describing *how* it is to be done.

Main content

Chapter Contents

14.3.3 Other Considerations

Existing tuples in a relational database table can be modified or deleted, and new tuples can be added to a table. These operations must be done with care to be sure the data remains correct and consistent throughout the database. In database terminology, the *integrity* of the data must be preserved. There are three integrity rules that, if enforced during additions, modifications, or deletions, will help in this goal. The **entity integrity** rule says that no primary key value, or no component of a composite primary key value, can be missing ("null") in a tuple. The reason is, if the primary key uniquely identifies a tuple, then a tuple with (part of) its primary key missing might not be uniquely identifiable. The **data integrity** rule specifies that values for a particular attribute must come from the appropriate category of information for that attribute. In the Rugs-For-You *InsurancePolicies* table, for example, any values for the *PlanType* attribute must be designations for valid plan types, and any values for the *DateIssued* must be valid dates. Finally, the **referential integrity** rule specifies that any value of a foreign key attribute in a given table must match a value in the corresponding primary keys of the related table. For example, we can't add a tuple to the *InsurancePolicies* table of the Rugs-For-You database with an *EmployeeID* value that does not exist in the Employees table. Most database systems enforce the integrity rules by default.

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 to the person waiting for the result. 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.

Creating small additional records to be stored along with the file, although consuming extra storage, can significantly reduce access time. 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.

SQL, NoSQL, NewSQL

Throughout this database section we have been talking about relational databases, where a file is thought of as a rather rigid two-dimensional table, there are various connections between "related" tables, and queries are processed using SQL. *NoSQL* databases are designed to give high performance on massive clusters of data and rely on various structures other than tables. Driven by Big Data (see Section 14.4) and real-time web applications, such systems are intended to be highly scalable, highly distributed, flexible, and error-resilient. NewSQL databases are sort of a combination of the two; they do rely on SQL and achieve the consistency required for traditional transaction-process activities, while featuring the scalability of NoSQL systems. Neither NoSQL nor NewSQL systems are overall replacements for standard relational databases, but they are alternative choices depending on the application/data of interest.

A **distributed database** allows 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 the data has to travel across a network.

Practice Problems

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

Answer

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 must include the table name as well.

Answer

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.

Answer

Assuming that no other changes are made to any of the three tables in the Rugs-For-You database, what integrity constraint is violated if the tuple with ID 171 is deleted from the *Employees* table?

Answer

Referential integrity. Removing this tuple leaves the InsurancePolicies table with tuples that have EmployeeID foreign key values no longer existing as a primary key value in the Employees table.

Change font sizeMain content

Chapter Contents

14.4 Data Science

We are surrounded by data, and the amount of data is growing exponentially. According to International Data Corporation (IDC), the “global datasphere,” that is, the amount of data that existed in the world in 2017, was 25 zettabytes, which measured in bytes is 25 followed by 21 zeroes, or 25 trillion gigabtyes. Wow. And IDC estimates that by the year 2025, this figure will be 160 zettabytes.

Much of this data is trivial, but a lot is critically important. How do we make sense of it all? How can we locate truly useful information from this vast ocean of data? How can we make use of it to improve living conditions, find solutions to major health problems, protect the environment, and an endless stream of other important questions?

Because of the massive amounts of data, we need skills to analyze important segments of data, extract information from them, and apply that information to solve problems from how to increase potato production to how to cure cancer.

First, we’ll look at some common terms. The first three are often used interchangeably, so the descriptions given here would not be universally agreed upon.

- *Big Data*—A term that expresses that we now have huge amounts of data available, as mentioned earlier.
- *Data analysis*—The process of finding the right data sets, putting the data into the right format, and writing queries to extract information from the data, much as we did with the Rugs-For-You database.
- *Data science*—**Data science** incorporates many of the tasks of data analysis, but also involves knowledge of the enterprise in order to formulate useful queries, along with the use of sophisticated statistics and visualization techniques. Data science also involves interpreting the results in terms of the enterprise and predicting future strategies likely to achieve a desired goal.

- *Data warehouse*—A data warehouse is a collection of databases that contain current and archived data used for research and analysis purposes rather than to manage day-to-day business transactions such as inventory control or payroll data.

Main content

Chapter Contents

14.4.1 Tools

One of the major objectives of data science is to analyze large amounts of data (often obtained from data warehouses) to extract and interpret previously hidden patterns contained therein. This process is called data mining. In other words, **data mining** is used to discover previously hidden patterns that a big data set might contain. This sounds rather magical.

Data mining is part of an overall process consisting of several steps:

1. Determine what problem you are trying to solve: What information do you hope to unearth from your large data set?
2. Review the condition of the data you have. Are there several data sets? If so, do they all have the same structure, for example, does each tuple have the same attributes? Are there any tuples with missing or obvious "outlier" attribute values that should be eliminated?
3. Determine a model to represent your data in some way that will help bring out patterns. You want to use these patterns to classify your existing data and help determine which attributes are the strongest predictors of a given outcome. It is this step that is the data mining part: creation of a model.
4. Evaluate your model. Are the results predicted borne out by further data? Would a different model give better results?
   Let's look at an extremely simple example. You are the loan manager at Gringotts Third Bank, and you must decide who gets approved for a loan, that is, who is a high risk (likely to default on some payments) and who is a low risk (very likely to repay the loan on time). Past experience at the bank has produced the data shown in Figure 14.9.

Figure 14.9

Existing data for bank loan risk

ID	EMPLOYED	GENDER	MARRIED	RISK
1	Y	M	Y	Low
2	Y	F	N	Low
3	N	M	N	High
4	Y	M	Y	Low
5	Y	F	Y	Low

ID	EMPLOYED	GENDER	MARRIED	RISK
6	N	F	N	High
7	Y	M	N	High
8	N	M	N	High
9	Y	F	N	Low
10	Y	M	Y	Low

What is the problem you want to solve? You would like to use the attributes of Employed (yes or no), Gender (male or female), and Married (yes or no) to predict Risk (high or low). What is the condition of the data (aside from being a ridiculously small data set)? All attributes have reasonable values.

What model should we use? This is the hard part because there are many ways to create a data mining model, some involving quite complex mathematics or statistics. We are going to use a rather simple model called a *decision tree*. (Recall that we used tree structures in Chapter 3 to represent the actions of the binary search algorithm and the possible paths in a graph to find Hamiltonian circuits.) A decision tree for data mining uses the input attributes (in our problem, Employed, Gender, and Married) as nodes in the tree; at each node, the branches below it represent the possible values. The leaves of the tree (the nodes at the bottom of the tree) represent the values for the target attribute, which here is Risk.

We'll make the root of the tree, that is, the top-most node, the attribute Employed. So the beginning of the tree is shown in Figure 14.10(a). Now as it happens, all tuples in Figure 14.9 who are not employed are at high risk, so we don't have to consider any further attributes for them and this terminates the N branch of Employed. For those that are employed, we next consider Gender (Figure 14.10(b)). The three employed females in Figure 14.9 are all at low risk, so that terminates another branch, but we have to consider the marital status of employed males (Figure 14.10(c), which is the complete decision tree).

Figure 14.10Decision tree for bank loan risk

According to this model, any new unemployed person looking for a loan, regardless of gender or marital state, will be turned down, whereas any employed female will get a loan, regardless of her marital state. But a male, in order to get loan approval, must be both employed and married.

It is easy to see how to turn this model into an algorithm that can be implemented in some programming language. Then, when a new customer applies for a loan at Gringotts

Third Bank, you can just collect the desired attribute values, plug the results into your computer program, and come out with the risk factor. In other words, you now have an algorithmic solution to your problem. But reliance on algorithmic predictions has risks:

1. Is the model the algorithm implements too simplistic? Of course, our bank loan model is an unrealistically simple model, based as it is on a very small data set. And there are many other attributes that might influence the applicant's creditworthiness, such as age, income, savings, amount of debt, amount of loan requested, term of loan requested, and so forth.

   Data modeling is never done with 10 items of data, but with millions or billions of data items. Often a large part of the available data, say 70–80%, is used as training data, that is, to build the model. Then the model is tested against the remaining existing data to see whether it will be a good predictor.

2. Is the data used to train the model biased, thereby leading to biased decision-making by the algorithm? (See the Special Interest box, "Algorithm Bias.")

Algorithm Bias

As data science becomes more pervasive, the consequences of biased decision algorithms, that is, decision algorithms based on biased input, become more apparent. For example,

"According to a 2013 article published by Sonja B. Starr, a professor of law at the University of Michigan Law School, nearly every state has adopted some type of risk-based assessment tools to aid in [criminal] sentencing. The primary concern related to these tools revolves around the use of computerized algorithms, which provide risk scores based on the result of questions that are either answered by defendants or pulled from criminal records, and whether such tools may ultimately penalize racial minorities by overpredicting the likelihood of recidivism in these groups."*

In January 2017, The ACM (Association for Computing Machinery) U.S. Public Policy Council issued a statement on Algorithmic Transparency and Accountability, noting that using algorithms for automated decision-making that affects individuals can result in harmful discrimination. The statement contains guidelines for institutions creating or using such algorithms.*

Other tools for data science incorporate statistics and visualization to observe patterns, trends, and relationships buried in a large data set. In Chapter 10, we briefly discussed the use of the programming language R (see R), which is specifically designed for statistical computing and graphics. There we analyzed a trivially small data set that looks like this, where surveys from six sites contributed information on the percentage of the survey population that exhibits medical condition X, medical condition Y, are smokers, are overweight, or have low income levels.

Practice Problems

Draw the resulting decision tree if the following tuple is added to Figure 14.9:

11	Y	F	N	High

2. Answer
3. The decision tree after adding the

new tuple would be

R provided us with an easy way to obtain basic statistical information such as the minimum, maximum, and average over the six sites for each of the attributes X, Y, Smoker, Overweight, and Low_Income. R also revealed that condition Y has a moderate correlation with being overweight, and both smoking and being overweight have a somewhat strong correlation with low income, where correlation means that these attributes increase or decrease more or less together. And R produced a graphical visualization of the data values for each of the six sites in the form of a bar chart. (Such information can also be obtained, although not quite as easily, using a spreadsheet.)

Main content

Chapter Contents

14.4.2 Personal Privacy

We learned that a database management system can easily make connections among different files, and even among data stored at different locations, so one might wonder, 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? 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. In general, issues of personal privacy and public safety are magnified enormously by the capabilities of networked databases, and privacy concerns arise because of the potential for information to be uncovered from massive databases using data mining techniques.

When does data mining become an actual privacy issue? Companies called **data brokers** collect data on virtually everyone and then in turn sell that data to clients, who use it primarily to target consumers with “personalized” advertising. Data brokers get their data from

- Public records (birth certificates, marriage certificates, death certificates, property records, bankruptcies, courthouse records, business ownership, professional listings, voter registrations, auto registrations);
- Publicly available data (telephone directories, business directories, newspapers, website tracking data, social networking sites, résumé sites, online forums); and
- Nonpublic data (consumer transaction data; cell phone records; information from mobile apps),

and they sell it to … almost anyone willing to pay for it.

Many people are not even aware that data brokers exist. There is currently little regulation concerning what they cannot collect (except for medical records and data used to determine your credit rating or your eligibility for housing, employment, or insurance). Nor are they required to let you see what data they have collected about you.

The companies who purchase your data can use sophisticated data mining techniques to put together the separate little dots of information about you to form a startlingly complete picture, including your age, marital status, children’s ages, ethnicity, income level, hobbies, education level, occupation, buying habits, and vacation destinations.

They may even know your potential health conditions—do you do a lot of online searching for supersize clothes or for diet information?

What Your Smartphone Photo Knows

If you take a photo with your smartphone, the image file probably contains much more than just the image. This **metadata** (data about data, in this case, data about the image) may tell the local time when the photo was taken, the exact location where it was taken (GPS coordinates), information about the make and model of the camera, the size of the image file (in MB), and the dimensions of the image in pixels. All this metadata is transmitted as part of the image file when you email it or post it online. This makes it handy to catalog your photos or remind you where you were when a photo was taken, but if the photo is publicly viewed there may be privacy issues. For example, if a photo shows your lovely home (and gives its exact location), it may become a target for a break-in. (It is possible to disable the smartphone's location-information feature, but the details on how to do this vary by manufacturer. If this feature is totally disabled, then other smartphone apps such as driving directions will not work, either.)

According to its 2016 annual report, data broker Acxiom Corporation had by then collected data on over 700 million individuals, using that data to conduct over 1 trillion transactions per week with over 3000 clients. These clients come from all sectors of the economy, including financial services, retail, telecommunications, insurance, technology, healthcare, travel, entertainment, non-profit, and government. Acxiom's goal is to enable those clients "to reach audiences with highly relevant messages." Part of that goal is achieved by providing not just raw data, but categories of consumer "labels" such as "Expectant Parent," "Number of Bedrooms," "Delinquent Tax Flag," "Diabetic Focus," and many more. (Note that such data may be useful for political campaigns where the client is trying to influence not your purchase, but your vote.) Acxiom does allow consumers to see and correct information that Acxiom holds about them.

Advertising targeted to your individual profile can sometimes be helpful, such as information on sales of baby formula to expectant mothers. However, privacy experts fear that such classifications could lead to targeting vulnerable groups, for example, predatory loan offers to people with considerable debt or exclusions of high-risk patients from opportunities to purchase health services. In addition to this, you just might feel that some of your personal data should remain private.

Companies need not turn to a data broker to collect information. Retail companies such as Target, Amazon, grocery stores, and so forth can collect data about your shopping habits at the point of sale, then apply data mining to more effectively target advertising to your particular circumstances or interests.

Main content

Chapter Contents

14.4.3 For the Greater Good

We've perhaps painted a rather unsettling picture of data mining with respect to personal privacy. However, the main societal impact of data mining is the positive contributions it can make to science, medicine, ecology, and so many other fields. Without going into details, we'll give four diverse examples.

- The December 2016 issue of *The Journal of Infectious Diseases* was devoted entirely to the use of Big Data for infectious disease surveillance and modeling. One article discusses the possible use of global positioning data obtained from cell phone usage to reveal population movements that can help drive the spread of epidemic diseases.
- Data-mining techniques have been applied to data sets of housing prices. A research team in India in 2016 used data mining on previous market trends and home prices to develop a model that predicts future trends and prices. This has two-fold benefits. For consumers, given their housing priorities and budget constraints, this helps pinpoint the most suitable areas for house-hunting. For developers (home builders), this can guide development plans and help reduce financial risk.
- The aerospace industry is awash with data. The Boeing 787 Dreamliner is a long-haul, twin-engine jet plane that first entered commercial service in 2011. Due to the many sensors on the plane, a single flight can produce terabytes of data. But, quoting a Boeing engineer, “Big Data by itself doesn’t produce any value. The value really comes from being able to look at the data through algorithms, machine learning, and data mining that help you pull out the information and then analyze the results and transform that data into right decisions.” (This is pretty much our definition of data science!) For Boeing, data mining can help with design decisions, performance during flight, warnings of anomalous conditions, and many other aspects of safely and efficiently building, flying, and maintaining such complex machines.
- The “Ancientbiotics team” is a group of medievalists, microbiologists, medicinal chemists, parasitologists, pharmacists and data scientists from multiple universities and countries. Yes, medievalists—authorities on that period of time otherwise known as the European Middle Ages, from the 5th to the 15th centuries. This period is also known as the Dark Ages, implying ignorance or, at best, little social or scientific progress. Not so fast, says this team of researchers, who are using modern data science tools to extract information from medieval "recipes" for treating infections. This research can address a crucial issue, which is that current antibiotics are proving ineffective against microbes that have developed resistance to these drugs. If no solutions are found to this problem, it is estimated that by 2050, 10 million people will die annually from drug-resistant infections. The team translated a 1000-year-old recipe for treating eye infections, recreated the “potion,” and tested it. To their astonishment, early lab results show that the resulting "antibiotic" is extremely effective in combatting MSRA, a very potent drug-resistant bacterium that is a major problem for modern hospitals. Now the team is on the hunt for other potential medicines from the Middle Ages.

Main content

Chapter Contents

14.5 Conclusion

In this chapter, we’ve looked at ecommerce—an important application of computing. You’ve learned that there is much more involved in a retail web business than simply creating a webpage, and that technical areas of computer science such as information security, networking, and databases play a critical role. You’ve also seen new models of ecommerce and electronic payment that have become hugely popular. We went on to examine in more detail how databases work. Finally, we looked at some aspects of data science, including some of the tools it uses, its impact on personal privacy, and its potential for problem-solving in many important areas.

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.