CHAPTER 5
Data and Knowledge Management

CHAPTER OUTLINE

LEARNING OBJECTIVES

5.1 Discuss ways that common challenges in managing data can be addressed using data governance.
5.2 Identify and assess the advantages and disadvantages of relational databases.
5.3 Define *Big Data* and explain its basic characteristics.
5.4 Explain the elements necessary to successfully implement and maintain data warehouses.
5.5 Describe the benefits and challenges of implementing knowledge management systems in organizations.
5.6 Understand the processes of querying a relational database, entity-relationship modeling, and normalization and joins.

Opening Case

MIS Our Genetic Data: A Double-Edged Sword

Personal data is an umbrella term referring to our names, addresses, phone numbers, Social Security numbers, and health and financial records, along with social media posts, location data, search-engine queries, and a myriad of other personal details. These data are highly sensitive because their misuse impacts all of us so severely: for example, identity theft, extortion, financial loss, loss of privacy, and many other negative consequences.

A major challenge associated with keeping our personal data safe is that a variety of entities collect them. In some instances, companies collect, analyze, and sell our personal data without our knowledge or consent. Recall our discussion of data aggregators in Chapter 3.

Sometimes, we provide our data willingly and knowingly, even though the specifics appear in lengthy, hard-to-read terms-of-service agreements. For example, with Google and Facebook, the consumer can use each platform's functions for free in return for allowing the companies to monetize our data by targeting us with advertisements.

In recent years, scientists have been able to discover genetic data, which is the most personal type of data for each of us. *Genetic data* refers to the inherited characteristics located in our chromosomal deoxyribonucleic acid (DNA) or ribonucleic acid (RNA). Each of us has a unique genome, which comprises our complete set of DNA, including all of our genes. As with all of our personal data, the problems with our genetic data are: Who has collected it? Who is collecting it now? Who has it? What are they doing with it?

As a result of patients taking a more proactive role in their health care, direct-to-consumer (DTC) laboratory testing is becoming increasingly popular. DTC genetic tests allow consumers to access information about their genetics without involving their health-care professionals.

As of July 2020 the International Society of Genetic Genealogy (**www.isogg.org**) listed 33 DTC companies that are performing genetic testing. More than 12 million people around the world have utilized the services of one of these companies, such as 23andMe (**www.23andme.com**). How does this process work? Basically, you send a vial of saliva along with a fee to 23andMe. In return, you receive a report that includes (a) the proportion of your DNA that comes from each of 45 global genetic populations, (b) the origins of your maternal and paternal ancestors, and (c) a feature that allows you to connect with DNA relatives, if you opt in. The report also contains health-care information such as DNA that might not affect your health but could affect the health of your children as well as how your DNA can influence your chances of contracting certain diseases.

In 2010, GEDmatch (**www.gedmatch.com**) was founded to help amateur and professional genealogists and adoptees searching for birth parents. Users could upload their DNA profiles from DTC companies to identify relatives who had also uploaded their profiles. Participants could hide their names by using aliases. Each account, however, had to include an associated email address. GEDmatch provided results such as the closest matches to a user's DNA and the estimated number of generations to a common ancestor. By December 2019 more than 1 million people had uploaded their profiles to the site.

Solving Crimes (The Good News)

The police began using GEDmatch to solve crimes, particularly cold cases. A cold case is an unsolved criminal investigation which remains active pending the discovery of new evidence. Until the authorities began using commercial genetic databases to assist with their detective work, the only DNA officially available to them were records of DNA samples provided by individuals involved with the justice system; that is, suspects and convicted criminals. This process of using commercial genetic databases led to an ongoing debate regarding public safety versus privacy concerns—in essence, the good news and the bad news about the use of our genetic data. Let's look closely at forensic genealogy.

Forensic genealogy is the practice of analyzing genetic data from DTC companies or from companies such as GEDmatch for the purpose of identifying suspects or victims in criminal cases. Forensic genealogy uses *genetic genealogy*, which is the use of DNA tests in combination with traditional genealogical methods to infer biological relationships between and among individuals. Traditional genealogical methods include the study of families and family history by conducting oral interviews and researching records such as birth and marriage certificates, census data, and newspaper obituaries. The process combines data from those techniques with data from modern sources such as social media platforms, particularly Facebook.

To use forensic genealogy, law enforcement agencies have uploaded crime-scene DNA data to genetic companies to locate possible relatives of potential suspects. Genealogy experts then assemble family trees and analyze demographic identifiers. One company, Parabon NanoLabs (**www.parabon-nanolabs.com**), is a leader in the use of forensic genealogy as an investigative tool.

After law enforcement agencies have identified potential suspects through forensic genealogy, they use conventional investigative methods such as comparing present physical features to past eyewitness statements and police sketches. This process can often narrow down the choices to a few candidates.

Solving Cold Cases - Example 1

In 1987 a young woman left her family home for what was supposed to be a quick trip with her boyfriend. They did not return the next day, as planned. After several days, her body was found. She had been shot. Two days later, his body was found 75 miles away. He had been strangled. Police found the van they were driving in a third location.

The police collected a semen sample from the woman's pants. By 1994, DNA analysis had advanced to the point that the semen could produce a genetic profile of the suspect. In 2003, investigators uploaded the profile to CODIS, the FBI's criminal DNA database. The investigators hoped for a match, but CODIS did not return one.

In 2017, a detective who was examining the department's cold cases heard that there was a way to obtain more information from DNA. He contacted Parabon. Genealogists there used the 30-year-old DNA sample and found two of the eventual suspect's second cousins: one on his father's side of the family, and one on his mother's side. The firm's lead genealogist used newspaper archives and

marriage records to build a family tree dating back to the two cousins' great-grandparents. She then found a particular couple who lived about 7 miles from where the boyfriend's body was found.

The genealogist concentrated on that couple's only son—who would have been 24 at the time of the murders—as a suspect. Next, an undercover officer who had begun watching the suspect picked up a cup that had fallen off his truck. The DNA from the cup and the DNA from a cheek swab provided by the suspect once in custody matched the DNA gathered 30 years before. He was convicted and sentenced to serve two life sentences concurrently. As of July 2020 he was appealing the verdict.

Solving Cold Cases - Example 2

The trail of the Golden State Killer had gone cold decades ago. The police linked him to more than 50 rapes and at least 12 murders from 1976 to 1986 but he eluded all attempts to find him. However, the police had retrieved a DNA sample from one of the crime scenes and had carefully preserved it over the years.

When the police heard about GEDmatch, they uploaded the suspect's DNA sample. After four months of close examination, GEDmatch provided matches to relatives of the suspect, although not the suspect himself. Because the site did provide family trees, genealogists were able to find third cousins of the suspect. After examining the family tree to find a common ancestor of the third cousins, they proceeded back down the tree to identify a likely suspect. In addition to the DNA evidence, some of his victims had described him as a 5'9", 165-pound white male, characteristics that matched his features.

In April 2018 the police arrested the suspect, who was then 72 years old and a former police officer. His DNA matched the sample taken in an earlier crime. In June 2020 he pled guilty to several counts of first-degree murder and in August the court sentenced him to 11 consecutive life sentences without the possibility of parole.

Privacy Concerns (The Bad News)

The use of open-source genetic databases has ignited a debate regarding the Fourth Amendment. This amendment states that a warrant is required in situations that violate an individual's reasonable expectations of privacy. Given the sensitivity of information surrounding commercial genetic databases, especially regarding familial associations, courts have asserted that individuals are subject to protection under the Fourth Amendment.

Privacy advocates protested that law enforcement agencies were accessing the entire GEDmatch database without the informed consent of the users. As a result, GEDmatch began to require its customers to specifically opt in to allow law enforcement agencies to access their genetic data. The agencies objected, claiming that GEDmatch's policy change would make it much more difficult to identify suspects and solve cold cases using genetic genealogy.

In September 2019 the U.S. Department of Justice (**www.justice.gov**) released interim guidelines stating that federal investigators could use forensic genealogy to discover suspects only in serious crimes such as murder and rape. The guidelines also stated that federal investigators must have a search warrant to collect DNA samples from a suspect's relatives, who must have previously opted in to allow law enforcement agencies to access their genetic data. Because the DOJ is a federal agency, these guidelines did not apply to state or local law enforcement agencies.

In December 2019 forensic for-profit DNA analysis company Verogen (**www.verogen.com**) bought GEDmatch. Verogen's CEO asserted that the site would focus on solving crimes, not just connecting family members through their DNA. He further stated that current and future users would have the ability to opt out of criminal DNA searches and that Verogen would fight "future attempts to access the data of those who have not opted in."

Forensic genealogists then began using Family Tree DNA (**www.familytreedna.com**) due to the increased difficulty involved in obtaining genetic profiles from Verogen. Family Tree's policy dictates that customers are automatically opted in unless they choose to opt out.

Sources: Compiled from S. Bradbury, "Killer Sentenced to Life in Prison in 1980 Colorado Cold Case Solved with DNA," *The Denver Post*, July 1, 2020; J. Chamary, "How Genetic Genealogy Helped Catch the Golden State Killer," *Forbes*, June 30, 2020; H. Murphy and T. Arango, "Joseph DeAngelo Pleads Guilty in Golden State Killer Cases," *New York Times*, June 29, 2020; S. Gilgore, "The Reston Company Cracking Cold Police Cases Is Coming to Your TV," *Washington Business Journal*, May 6, 2020; E. Ruiz, "40-Year-Old Cold Case Solved with New Genetic Genealogy Technology," *The Denver Channel*, March 6, 2020; D. Geiger, "Trucker Pleads Guilty to 21-Year-Old Woman's 40-Year-Old Cold Case Murder," **Oxygen.com**, February 25, 2020; "GEDmatch Sold, Will Serve as 'Molecular Witness' for Police," *The Crime Report*, December 10, 2019; H. Murphy, "Genealogy Sites Have Helped Identify Suspects. Now They've Helped Convict One." *New York Times*, July 1, 2019; "GEDmatch Puts DNA Database Off-Limits to Police: Will Cold Cases Get Colder?" *The Crime Report*, May 22, 2019; P. Aldhous, "This Genealogy Database Helped Solve Dozens of Crimes. But Its New Privacy Rules Will Restrict Access by Cops," *BuzzFeed News*, May 19, 2019; P. Aldhous, "The Arrest of a Teen on an Assault Charge Has Sparked New Privacy Fears about DNA Sleuthing," *BuzzFeed*, May 14, 2019; J. Lepola, "A Closer Look at Solving Crimes with the Help of Genetic Genealogy," *Sinclair Broadcast Group*, May 14, 2019; L. Matsakis, "The Wired Guide to Personal Data (and Who Is Using It)," *Wired*, February 15, 2019; "The Genomic Data Challenges of the Future," *The Medical Futurist*, October 27, 2018; D. Barry, T. Arango, and R. Oppel, "The Golden State Killer Left a Trail of Horror with Taunts and Guile," *New York Times*, April 28, 2018; G. Kolata and H. Murphy, "The Golden State Killer Is Tracked through a Thicket of DNA, and Experts Shudder," *New York Times*, April 27, 2018.

Questions

1. Should we be willing to sacrifice our privacy to solve crimes? In other words, does the end justify the means?
2. You are a candidate for a position at a company. Discuss the positive and negative consequences of that company having access to your DNA results.
3. Would you be willing to have your DNA results made available to law enforcement agencies to help in solving a criminal case? Why or why not? Support your answer.

Introduction

Information technologies and systems support organizations in managing—that is, acquiring, organizing, storing, accessing, analyzing, and interpreting—data. As you noted in Chapter 1, when these data are managed properly, they become *information* and then *knowledge.* Information and knowledge are invaluable organizational resources that can provide any organization with a competitive advantage.

So, just how important are data and data management to organizations? From confidential customer information (see this chapter's opening case) to intellectual property to financial transactions to social media posts, organizations possess massive amounts of data that are critical to their success. Of course, to benefit from these data, they need to manage it effectively. This type of management, however, comes at a huge cost. According to Symantec's (**www.symantec.com**) State of Information survey, digital information costs organizations worldwide more than $1 trillion annually. In fact, it makes up roughly *half* of an organization's total value. The survey found that large organizations spend an average of $40 million annually to maintain and use data, and small-to-medium-sized businesses spend almost $350,000.

This chapter examines the processes whereby data are transformed first into information and then into knowledge. Managing data is critical to all organizations. Few business professionals are comfortable making or justifying business decisions that are not based on solid information. This is especially true today, when modern information systems make access to that information quick and easy. For example, there are information systems that format data in a way that managers and analysts can easily understand. Consequently, these professionals can access these data themselves and then analyze the data according to their needs. The result is useful *information.* Managers can then apply their experience to use this information to address a business problem, thereby producing *knowledge.* Knowledge management, enabled by information technology, captures and stores knowledge in forms that all organizational employees can access and apply, thereby creating the flexible, powerful "learning organization."

Organizations store data in databases. Recall from Chapter 1 that a *database* is a collection of related data files or tables that contain data. We discuss databases in Section 5.2, focusing on the relational database model. In Section 5.6, we take a look at the fundamentals of relational database operations.

Clearly, data and knowledge management are vital to modern organizations. But, why should *you* learn about them? The reason is that you will play an important role in the development of database applications. The structure and content of your organization's database depend on how users (meaning you) define your business activities. For example, when database developers in the firm's MIS group build a database, they use a tool called *entity-relationship (ER) modeling*. This tool creates a model of how users view a business activity. When you understand how to create and interpret an ER model, then you can evaluate whether the developers have captured your business activities correctly.

Keep in mind that decisions about data last longer, and have a broader impact, than decisions about hardware or software. If decisions concerning hardware are wrong, then the equipment can be replaced relatively easily. If software decisions turn out to be incorrect, they can be modified, though not always painlessly or inexpensively. Database decisions, in contrast, are much harder to undo. Database design constrains what the organization can do with its data for a long time. Remember that business users will be stuck with a bad database design, while the programmers who created the database will quickly move on to their next projects.

Furthermore, consider that databases typically underlie the enterprise applications that users access. If there are problems with organizational databases, then it is unlikely that any applications will be able to provide the necessary functionality for users. Databases are difficult to set up properly and to maintain. They are also the component of an information system that is most likely to receive the blame when the system performs poorly and the least likely to be recognized when the system performs well. This is why it is so important to get database designs right the first time—and you will play a key role in these designs.

You might also want to create a small personal database using a software product such as Microsoft Access. If so, you will need to be familiar with at least the basics of the product.

After the data are stored in your organization's databases, they must be accessible in a form that helps users make decisions. Organizations accomplish this objective by developing *data warehouses.* You should become familiar with data warehouses because they are invaluable decision-making tools. We discuss data warehouses in Section 5.4.

You will also make extensive use of your organization's knowledge base to perform your job. For example, when you are assigned a new project, you will likely research your firm's knowledge base to identify factors that contributed to the success (or failure) of previous, similar projects. We discuss knowledge management in Section 5.5.

You begin this chapter by examining the multiple challenges involved in managing data. You then study the database approach that organizations use to help address these challenges. You turn your attention to Big Data, which organizations must manage in today's business environment. Next, you study data warehouses and data marts, and you learn how to use them for decision making. You conclude the chapter by examining knowledge management.

5.1 Managing Data

All IT applications require data. These data should be of high quality, meaning that they should be accurate, complete, timely, consistent, accessible, relevant, and concise. Unfortunately, the process of acquiring, keeping, and managing data is becoming increasingly difficult.

Author Lecture Videos are available exclusively in *WileyPLUS*.

Apply the Concept activities are available in the Appendix and in *WileyPLUS*.

The Difficulties of Managing Data

Because data are processed in several stages and often in multiple locations, they are frequently subject to problems and difficulties. Managing data in organizations is difficult for many reasons.

1. The amount of data is increasing exponentially with time. Much historical data must be kept for a long time, and new data are added rapidly. For example, to support millions of customers, large retailers such as Walmart must manage many petabytes of data. (A

petabyte is approximately 1,000 terabytes, or trillions of bytes; see Technology Guide 1.)

2. Data are also scattered throughout organizations, and they are collected by many individuals using various methods and devices. These data are frequently stored in numerous servers and locations and in different computing systems, databases, formats, and human and computer languages.

   MIS Organizations have developed information systems for specific business processes, such as transaction processing, supply chain management, and customer relationship management. The ISs that specifically support these processes impose unique requirements on data, which leads to repetition and conflicts across the organization. For example, the marketing function might maintain information on customers, sales territories, and markets. These data might be duplicated within the billing or customer service functions. This arrangement can produce inconsistent data within the enterprise. Inconsistent data prevent a company from developing a unified view of core business information—data concerning customers, products, finances, and so on—across the organization and its information systems. This situation refers to data silos.

   A **data silo** is a collection of data held by one group that is not easily accessible by other groups. Data silos hinder the process of gaining actionable insights from organizational data, create barriers to an overall view of the enterprise and its data, and delay digital transformation efforts (see Chapter 1). One major method to remove data silos is through cloud data management (see Technology Guide 3 for a complete discussion of cloud computing).

3. Another problem is that data are generated from multiple sources: internal sources (for example, corporate databases and company documents); personal sources (for example, personal thoughts, opinions, and experiences); and external sources (for example, commercial databases, government reports, and corporate websites).

   Some of these data sources are in the form of *data streams*, which are data that are continuously generated by point-of-sale systems, clickstream data, social media, and sensors. We take a brief look at these data streams here.

   - POM *Point-of-sale data*. Organizations capture data from each customer purchase with their POS systems. Clerks (or customers themselves using self-checkout) use bar code scanners to scan each item purchased. POS systems collect data in real time, such as the name, product identification number, and unit price of each item; the total amount of all items purchased; the sales tax on that amount; the payment method used; a time stamp of the purchase; and many other data points.
   - MKT *Clickstream data.* Clickstream data are those data that visitors and customers produce when they visit a website and click on hyperlinks (described in Chapter 6). Clickstream data include the terms that the visitor to the website entered into a search engine to reach that website, all links that users click, how long they spend on each page, if they click the "back" button, if they add or remove items from a shopping cart, and many other data points.
   - MKT *Social media data.* Social media data (also called social data) are the data collected from individuals' activity on social media websites, including Facebook, YouTube, LinkedIn, Twitter, and many others. These data include shares, likes and dislikes, ratings, reviews, recommendations, comments, and many other examples.
   - *Sensor data*. The Internet of Things (IoT; see Chapter 8) is a system in which any object, natural or manmade, contains internal or external wireless sensor(s) that communicate with each other without human interaction. Each sensor monitors and reports data on physical and environmental conditions around it, such as temperature, sound, pressure, vibration, and movement. Sensors can also control physical systems, such as opening and closing a valve and adjusting the fuel mixture in your car. (See our discussion of supervisory control and data acquisition (SCADA) systems in Chapter 4.)

   As with all technology, being able to collect massive amounts of data from many different sources is a double-edged sword. IT's About Business 5.1 shows how a startup company helps organizations make sense of the vast amounts of data available to them.

4. Adding to these problems is the fact that new sources of data such as blogs, podcasts, tweets, Facebook posts, YouTube videos, texts, and RFID tags and other wireless sensors are constantly being developed, and the data these technologies generate must be managed. Also, the data become less current over time. For example, customers move to new addresses or they change their names, companies go out of business or are bought, new products are developed, employees are hired or fired, and companies expand into new countries.

5. Data are also subject to *data rot.* Data rot refers primarily to problems with the media on which the data are stored. Over time, temperature, humidity, and exposure to light can cause physical problems with storage media and thus make it difficult to access data. The second aspect of data rot is that finding the machines needed to access the data can be difficult. For example, it is almost impossible today to find 8-track players to listen to music on. Consequently, a library of 8-track tapes has become relatively worthless, unless you have a functioning 8-track player or you convert the tapes to a more modern medium such as DVDs.

6. Data security, quality, and integrity are critical, yet they are easily jeopardized. Legal requirements relating to data also differ among countries as well as among industries, and they change frequently.

7. ACCT FIN Two other factors complicate data management. First, federal regulations—for example, the Sarbanes–Oxley Act of 2002—have made it a top priority for companies to better account for how they are managing information. Sarbanes–Oxley requires that (1) public companies evaluate and disclose the effectiveness of their internal financial controls, and (2) independent auditors for these companies agree to this disclosure. The law also holds CEOs and CFOs personally responsible for such disclosures. If their companies lack satisfactory data management policies and fraud or a security breach occurs, then the company officers could be held liable and face prosecution.

   Second, companies are drowning in data, much of which are unstructured. As you have seen, the amount of data is increasing exponentially. To be profitable, companies must develop a strategy for managing these data effectively. (See IT's About Business **5.1**.)

8. An additional problem with data management is Big Data. Big Data is so important that we devote Section 5.3 to this topic.

IT's About Business 5.1

MIS FIN What to Do with All that Data?

Today, capturing data is easy, and storing those data is relatively inexpensive. For this reason, an increasing number of companies are turning to Enigma to help them make sense of their data.

Founded in 2011, Enigma (**www.enigma.com**) is a data management and business intelligence company that specializes in data integration and analytics. The firm can rapidly make sense of multiple disconnected, disparate public and private data sources for a variety of uses. The company collects, cleans, organizes, integrates, and analyzes data from thousands of sources around the world with the use of machine learning algorithms (see Chapter 14).

Enigma differentiates itself from other platforms such as Amazon, Google, and Facebook. The company asserts that these firms apply machine learning to Big Data to "get people to click on things." In contrast, Enigma defines its mission as fundamentally changing how businesses function by collecting data from diverse sources and modeling how the world operates with machine learning algorithms.

The company began by collecting publicly available data, such as Federal Aviation Administration flight logs (**www.faa.gov**), university research publications, business filings, shipping manifests, the Census Bureau (**www.census.gov**), the Federal Communications Commission (**www.fcc.gov**), the Federal Election Commission (**www.fec.gov**), the Internet Revenue Service (**www.irs.gov**), the U.S. Customs & Border Protection agency (CBP; **www.cbp.gov**), and building permits.

The company then moved on to complex, hard-to-find data. For example, using Freedom of Information Act (FOIA) requests, they were able to access the CBP's Automated Manifest System to track every container ship arriving in the United States, including the importer and the ship's port of call. From the National Fire Incident Reporting System (**www.nfirs.fema.gov/NFIRSWeb/login**), they retrieved the cause and location of every fire in the country. To address energy, they collected oil well data from the Railroad Commission of Texas (**www.rrc.state.tx.us**), founded in 1891 to establish tariffs. From New York City's Metropolitan Transportation Authority (**www.new.mta.info**), they were able to access years of rail incident and injury data.

At the same time Enigma was accessing numerous disparate data sources, it was also developing machine learning algorithms to integrate data from these sources and analyze those data for insights. Let's look at several examples.

- FIN Consider the financial services industry. Banks integrate Enigma's data with customer data stored in their own systems to help them recognize fraud more quickly. For instance, Enigma helps American Express with its anti-money-laundering operations.

  As another example, to help banks accurately identify the best candidates for small business loans, Enigma integrates property tax filing data with state business filings and Uniform Commercial Code liens to produce credit ratings for each candidate.
- Pharmaceutical companies use Enigma's data and insights to improve drug safety. Specifically, Enigma has collected data on every molecule used by the U.S. pharmaceutical industry, as well as all drug trials, patent filings, and adverse events.
- FIN If a hedge fund wants to determine which restaurant chains have the potential to grow the fastest, Enigma can check FCC logs for radio licenses, which are required to open drive-through windows.
- Insurers ask Enigma to perform risk assessment. For example, if insurers want to avoid underwriting in risky fire zones, Enigma integrates and analyzes data sets on emergency call logs and building permits.

  MetLife (**www.metlife.com**) is one of the largest global providers of insurance, annuities, and employee benefit programs, with 90 million customers located in 60 countries. The company's insurance group integrates Enigma data gathered from public health systems and universities with its own data to improve its underwriting process. *Underwriting* is the process of accepting liability under an insurance policy, thus guaranteeing payment in case loss or damage occurs. MetLife's $588 billion investment management group is using Enigma data to quantify how the quality of restaurants, parks, and community event spaces affects real estate prices.
- Consider voter registration data in the United States, a public data set. These data are difficult to access and equally difficult to structure. Enigma works with this data set to help companies in the consumer packaged goods (CPG) industry place products such as drinks and soups. The startup bases their recommendations on where people live, their driving distance from businesses, and many other data points.

Not all of Enigma's machine learning algorithms target profits. For example, the firm has volunteered its applications toward studying the gender salary gap across 558 occupations. It has discovered that some of the most serious disparities occur in accounting, retail, and sales.

Enigma is also working with Polaris (**www.polarisproject.org**), a nonprofit, nongovernmental organization (NGO) that combats and prevents slavery and human trafficking. Interestingly, Enigma works with banks in this area, helping them catch people, because banks are required by regulations to do so, and banks incur liabilities when human traffickers conduct transactions in their networks.

Enigma notes that it has always been difficult for banks to share data, principally due to privacy concerns. Therefore, Enigma, along with Polaris, has deployed a crowdsourcing tool that many banks are using to share information on slavery and human trafficking. The tool is private to the banking industry because Enigma and Polaris do not want suspected traffickers to discover details about it.

As of July 2020 Enigma had integrated 100,000 data sets in more than 100 countries, organized data on 30 million small businesses, and accumulated 140 billion data points on the U.S. population. Analysts estimate the firm's valuation to be $750 million with annual revenue around $30 million.

Sources: Compiled from E. Williams, "Where Insights Meet Privacy: Privacy-Preserving Machine Learning," *Forbes*, July 2, 2020; "10 Emerging Tech Companies Showcase Inventive Products and Services," *Business Wire*, June 25, 2020; R. Sarfin, "Data Integration and Machine Learning: 3 Real-World Use Cases," *Syncsort*, August 6, 2019; D. Costa, "How Enigma Is Using Big Data

to Fight Human Trafficking," *PC Magazine*, June 10, 2019; A. Gara, "Data's Cartographers," *Forbes*, February 28, 2019; "Angel Nguyen Swift: It Is All about the Data," *ACAMS Today*, January 24, 2019; "Why We're Joining the Fight against Human Trafficking," *Medium.com*, December 4, 2018; "BB&T Announces Fintech Investment in Enigma," *PR Newswire*, September 18, 2018; A. Patnaik, "Data Integration and Machine Learning: A Natural Synergy," *tdwi.org*, August 18, 2017; M. Richardson, "Using Machine Learning Techniques to Automate Data Integration," *IT Toolbox*, May 31, 2017; **www.enigma.com**, accessed July 14, 2020.

Questions

(**look ahead to Section 5.2 for the definitions of structured and unstructured data**)

1. **Provide examples of structured data that Enigma collects and analyzes.**
2. **Provide examples of unstructured data that Enigma collects and analyzes.**

Data Governance

To address the numerous problems associated with managing data, organizations are turning to data governance. **Data governance** is an approach to managing information across an entire organization. It involves a formal set of business processes and policies that are designed to ensure that data are handled in a certain, well-defined fashion. That is, the organization follows unambiguous rules for creating, collecting, handling, and protecting its information. The objective is to make information available, transparent, and useful for the people who are authorized to access it, from the moment it enters an organization until it becomes outdated and is deleted.

One strategy for implementing data governance is master data management. **Master data management** is a process that spans all of an organization's business processes and applications. It provides companies with the ability to store, maintain, exchange, and synchronize a consistent, accurate, and timely "single version of the truth" for the company's master data.

Master data are a set of core data, such as customer, product, employee, vendor, geographic location, and so on, that span the enterprise's information systems. It is important to distinguish between master data and transactional data. **Transactional data**, which are generated and captured by operational systems, describe the business's activities, or *transactions*. In contrast, master data are applied to multiple transactions, and they are used to categorize, aggregate, and evaluate the transactional data.

Let's look at an example of a transaction. You (Mary Jones) purchase one Samsung 42-inch LCD television, part number 1234, from Bill Roberts at Best Buy, for $2,000, on April 20, 2017. In this example, the master data are "product sold," "vendor," "salesperson," "store," "part number," "purchase price," and "date." When specific values are applied to the master data, then a transaction is represented. Therefore, transactional data would be, respectively, "42-inch LCD television," "Samsung," "Bill Roberts," "Best Buy," "1234," "$2,000," and "April 20, 2017."

An example of master data management is Dallas, Texas, which implemented a plan for digitizing the city's public and private records, such as paper documents, images, drawings, and video and audio content. The master database can be used by any of the 38 government departments that have appropriate access. The city is also integrating its financial and billing processes with its customer relationship management program. (You will learn about customer relationship management in Chapter 11.)

How will Dallas use this system? Imagine that the city experiences a water-main break. Before it implemented the system, repair crews had to search City Hall for records that were filed haphazardly. Once the workers found the hard-copy blueprints, they would take them to the site and, after examining them manually, would decide on a plan of action. In contrast, the new system delivers the blueprints wirelessly to the laptops of crews in the field, who can magnify or highlight areas of concern to generate a rapid response. This process reduces the time it takes to respond to an emergency by several hours.

Along with data governance, organizations use the database approach to efficiently and effectively manage their data. We discuss the database approach in Section 5.2.

Before you go on…

1. What are some of the difficulties involved in managing data?

2. Define *data governance*, *master data*, and *transactional data*.

5.2 The Database Approach

From the mid-1950s, when businesses first adopted computer applications, until the early 1970s, organizations managed their data in a *file management environment*. This environment evolved because organizations typically automated their functions one application at a time. Therefore, the various automated systems developed independently from one another, without any overall planning. Each application required its own data, which were organized in a data file.

Author Lecture Videos are available exclusively in *WileyPLUS*.

Apply the Concept activities are available in the Appendix and in *WileyPLUS*.

A **data file** is a collection of logically related records. In a file management environment, each application has a specific data file related to it. This file contains all of the data records the application requires. Over time, organizations developed numerous applications, each with an associated application-specific data file.

For example, imagine that most of your information is stored in your university's central database. In addition, however, a club to which you belong maintains its own files, the athletics department has separate files for student athletes, and your instructors maintain grade data on their personal computers. It is easy for your name to be misspelled in one of these databases or files. Similarly, if you move, then your address might be updated correctly in one database or file but not in the others.

Using databases eliminates many problems that arose from previous methods of storing and accessing data, such as file management systems. Databases are arranged so that one set of software programs—the database management system—provides all users with access to all of the data. (You will study database management systems later in this chapter.) Database systems minimize the following problems:

- *Data redundancy*: The same data are stored in multiple locations.
- *Data isolation*: Applications cannot access data associated with other applications.
- *Data inconsistency*: Various copies of the data do not agree.

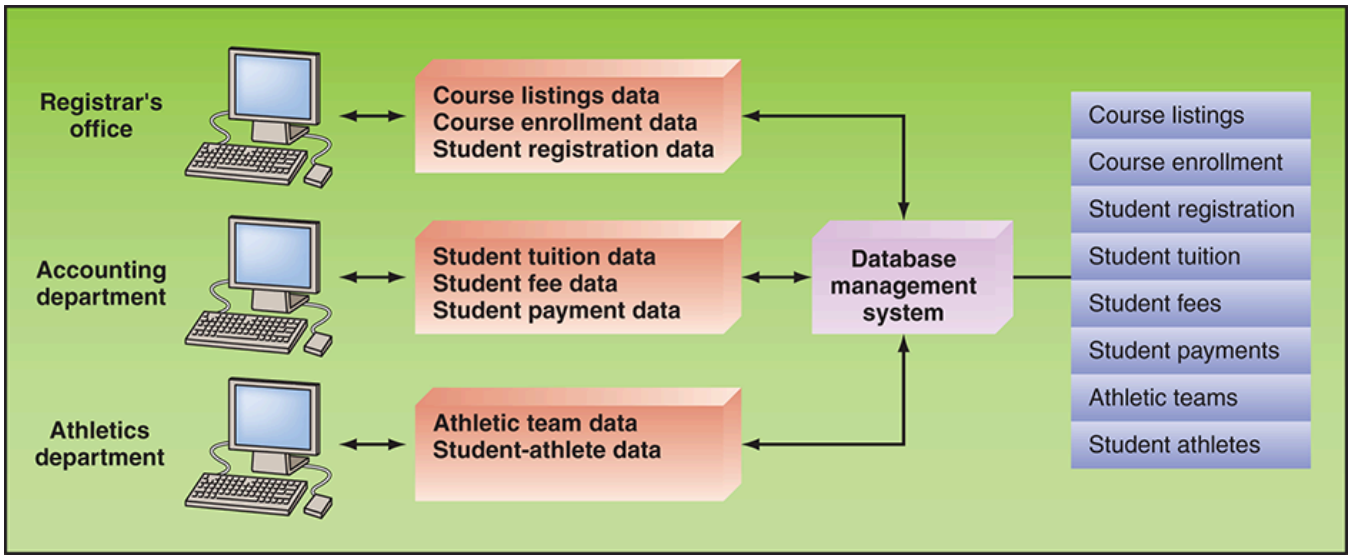


FIGURE 5.1 Database management system.

Database systems also maximize the following:

- *Data security*: Because data are "put in one place" in databases, there is a risk of losing a lot of data at one time. Therefore, databases must have extremely high security measures in place to minimize mistakes and deter attacks.
- *Data integrity*: Data meet certain constraints; for example, there are no alphabetic characters in a Social Security number field.
- *Data independence*: Applications and data are independent of one another; that is, applications and data are not linked to each other, so all applications are able to access the same data.

Figure 5.1 illustrates a university database. Note that university applications from the registrar's office, the accounting department, and the athletics department access data through the database management system.

A database can contain vast amounts of data. To make these data more understandable and useful, they are arranged in a hierarchy. We take a closer look at this hierarchy in the next section.

The Data Hierarchy

Data are organized in a hierarchy that begins with bits and proceeds all the way to databases (see Figure 5.2). A **bit** (*b*inary dig*it*) represents the smallest unit of data a computer can process. The term *binary* means that a bit can consist only of a 0 or a 1. A group of eight bits, called a **byte**, represents a single character. A byte can be a letter, a number, or a symbol. A logical grouping of characters into a word, a small group of words, or an identification number is called a **field**. For example, a student's name in a university's computer files would appear in the "name" field, and her or his Social Security number would appear in the "Social Security number" field. Fields can contain data other than text and numbers, such as an image, or any other type of multimedia. Examples are a motor vehicle department's licensing database that contains a driver's photograph, or a field that contains a voice sample to authorize access to a secure facility.

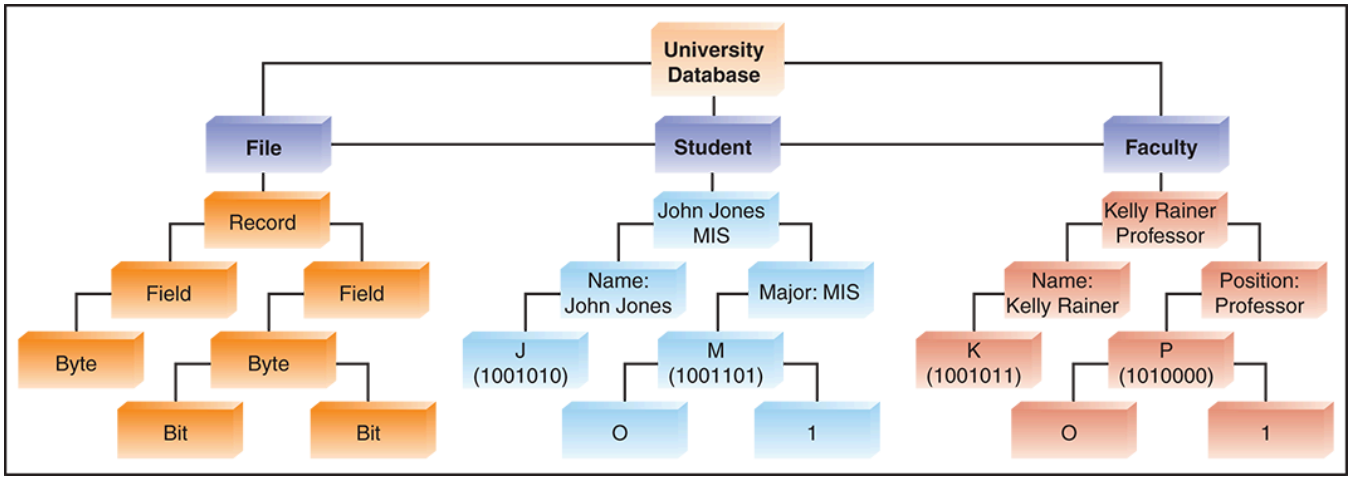


FIGURE 5.2 Hierarchy of data for a computer-based file.

A logical grouping of related fields, such as the student's name, the courses taken, the date, and the grade, comprises a **record**. In the Apple iTunes Store, a song is a field in a record, with other fields containing the song's title, its price, and the album on which it appears. A logical grouping of related records is called a data file or a **table**. For example, a grouping of the records from a particular course,

consisting of course number, professor, and students' grades, would constitute a data file for that course. Continuing up the hierarchy, a logical grouping of related files constitutes a *database.* Using the same example, the student course file could be grouped with files on students' personal histories and financial backgrounds to create a student database. In the next section, you will learn about relational database models.

The Relational Database Model

A **database management system (DBMS)** is a set of programs that provide users with tools to create and manage a database. Managing a database refers to the processes of adding, deleting, accessing, modifying, and analyzing data that are stored in a database. An organization can access these data by using query and reporting tools that are part of the DBMS or by utilizing application programs specifically written to perform this function. DBMSs also provide the mechanisms for maintaining the integrity of stored data, managing security and user access, and recovering information if the system fails. Because databases and DBMSs are essential to all areas of business, they must be carefully managed.

There are a number of different database architectures, but we focus on the relational database model because it is popular and easy to use. Other database models—for example, the hierarchical and network models —are the responsibility of the MIS function and are not used by organizational employees. Popular examples of relational databases are Microsoft Access and Oracle.

Most business data—especially accounting and financial data—traditionally were organized into simple tables consisting of columns and rows. Tables enable people to compare information quickly by row or column. Users can also retrieve items rather easily by locating the point of intersection of a particular row and column.

The **relational database model** is based on the concept of two-dimensional tables. A relational database generally is not one big table —usually called a *flat file*—that contains all of the records and attributes. Such a design would entail far too much data redundancy. Instead, a relational database is usually designed with a number of related tables. Each of these tables contains records (listed in rows) and attributes (listed in columns).

To be valuable, a relational database must be organized so that users can retrieve, analyze, and understand the data they need. A key to designing an effective database is the data model. A **data model** is a diagram that represents entities in the database and their relationships. An **entity** is a person, a place, a thing, or an event—such as a customer, an employee, or a product—about which an organization maintains information. Entities can typically be identified in the user's work environment. A record generally describes an entity. An **instance** of an entity refers to each row in a relational table, which is a specific, unique representation of the entity. For example, your university's student database contains an entity called "student." An instance of the student entity would be a particular student. Thus, you are an instance of the student entity in your university's student database.

Each characteristic or quality of a particular entity is called an **attribute**. For example, if our entities were a customer, an employee, and a product, entity attributes would include customer name, employee number, and product color.

Consider the relational database example about students diagrammed in **Figure 5.3**. The table contains data about the entity called students. As you can see, each row of the table corresponds to a single student record. (You have your own row in your university's student database.) Attributes of the entity are student name, undergraduate major, grade point average, and graduation date. The rows are the records on Sally Adams, John Jones, Jane Lee, Kevin Durham, Juan Rodriguez, Stella Zubnicki, and Ben Jones. Of course, your university keeps much more data on you than our example shows. In fact, your university's student database probably keeps hundreds of attributes on each student.

Every record in the database must contain at least one field that uniquely identifies that record so that it can be retrieved, updated, and sorted. This identifier field (or attribute) is called the **primary key**. For example, a student record in a U.S. university would use a unique student number as its primary key. (*Note:* In the past, your Social Security number served as the primary key for your student record. However, for security reasons, this practice has been discontinued.) In Figure 5.3, Sally Adams is uniquely identified by her student ID of 012345.

In some cases, locating a particular record requires the use of secondary keys. A **secondary key** is another field that has some identifying information but typically does not identify the record with complete accuracy. For example, the student's major might be a secondary key if a user wanted to identify all of the students majoring in a particular field of study. It should not be the primary key, however, because many students can have the same major. Therefore, it cannot uniquely identify an individual student.

A **foreign key** is a field (or group of fields) in one table that uniquely identifies a row of another table. A foreign key is used to establish and enforce a link between two tables. We discuss foreign keys in Section 5.6.

Organizations implement databases to efficiently and effectively manage their data. There are a variety of operations that can be performed on databases. We look at three of these operations in detail in Section 5.6: query languages, normalization, and joins.

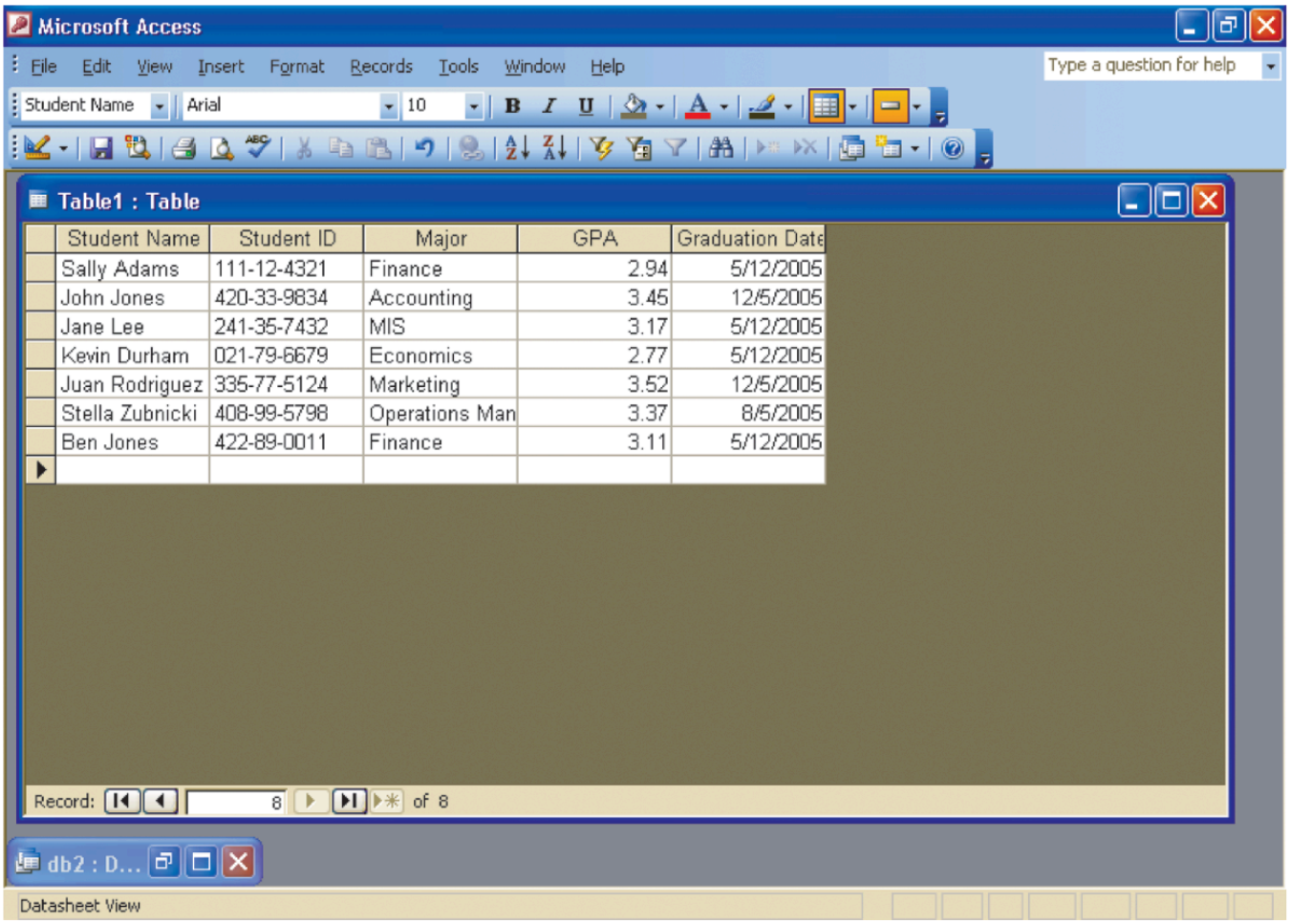


FIGURE 5.3 Student database example.

As we noted earlier in this chapter, organizations must manage huge quantities of data. Such data consist of structured and unstructured data and are called Big Data (discussed in Section 5.3). **Structured data** is highly organized in fixed fields in a data repository such as a relational database. Structured data must be defined in terms of field name and type (e.g., alphanumeric, numeric, and currency). **Unstructured data** refers to data that do not reside in a traditional relational database. Examples of unstructured data are e-mail messages, word processing documents, videos, images, audio files, PowerPoint presentations, Facebook posts, Tweets, Snaps, ratings and recommendations, and web pages. Industry analysts estimate that 80 to 90 percent of the data in an organization are unstructured. To manage Big Data, many organizations are using special types of databases, which we also discuss in Section 5.3.

Because databases typically process data in real time (or near real time), it is not practical to allow users access to the databases. After all, the data will change while the user is looking at them! As a result, data warehouses have been developed to allow users to access data for decision making. You will learn about data warehouses in Section 5.4.

Before you go on...

1. What is a data model?
2. What is a primary key? A secondary key?
3. What is an entity? An attribute? An instance?
4. What are the advantages and disadvantages of relational databases?

5.3 Big Data

We are accumulating data and information at an increasingly rapid pace from many diverse sources. In fact, organizations are capturing data about almost all events, including events that, in the past, firms never used to think of as data at all—for example, a person's location, the vibrations and temperature of an engine, and the stress at numerous points on a bridge—and then analyzing those data.

Author Lecture Videos are available exclusively in *WileyPLUS*.

Apply the Concept activities are available in the Appendix and in *WileyPLUS*.

Organizations and individuals must process a vast amount of data that continues to increase dramatically. According to IDC (a technology research firm; **www.idc.com**), the world generates over one zettabyte (10^{21} bytes) of data each year. Furthermore, the amount of data produced worldwide is increasing by 50 percent each year.

As recently as the year 2000, only 25 percent of the stored information in the world was digital. The other 75 percent was analog; that is, it was stored on paper, film, vinyl records, and the like. By 2020, the amount of stored information in the world was more than 98 percent digital and less than 2 percent nondigital.

As we discussed at the beginning of this chapter, we refer to the superabundance of data available today as Big Data. **Big Data** is a collection of data that is so large and complex that it is difficult to manage using traditional database management systems. (We capitalize *Big Data* to distinguish the term from large amounts of traditional data.)

Essentially, Big Data is about predictions (see Predictive Analytics in Chapter 12). Predictions do not come from "teaching" computers to "think" like humans. Instead, predictions come from applying mathematics to huge quantities of data to infer probabilities. Consider these examples:

- The likelihood that an e-mail message is spam
- The likelihood that the typed letters "teh" are supposed to be "the"
- The likelihood that the direction and speed of a person jaywalking indicates that he will make it across the street in time, meaning that a self-driving car need only slow down slightly

Big Data systems perform well because they contain huge amounts of data on which to base their predictions. Moreover, these systems are configured to improve themselves over time by searching for the most valuable signals and patterns as more data are input.

Defining Big Data

It is difficult to define Big Data. Here we present two descriptions of the phenomenon. First, the technology research firm Gartner (**www.gartner.com**) defines Big Data as diverse, high-volume, high-velocity information assets that require new forms of processing in order to enhance decision making, lead to insights, and optimize business processes. Second, the Big Data Institute (TBDI; **https://thebigdatainstitute.wordpress.com**) defines Big Data as vast datasets that:

- Exhibit variety;
- Include structured, unstructured, and semistructured data;
- Are generated at high velocity with an uncertain pattern;
- Do not fit neatly into traditional, structured, relational databases; and
- Can be captured, processed, transformed, and analyzed in a reasonable amount of time only by sophisticated information systems.

Big Data generally consists of the following:

- Traditional enterprise data—for example, customer information from customer relationship management systems, transactional enterprise resource planning data, Web store transactions, operations data, and general ledger data.
- Machine-generated/sensor data—for example, smart meters; manufacturing sensors; sensors integrated into smartphones, automobiles, airplane engines, and industrial machines; equipment logs; and trading systems data.
- Social data—for example, customer feedback comments; microblogging sites such as Twitter; and social media sites such as Facebook, YouTube, and LinkedIn.
- Images captured by billions of devices located throughout the world, from digital cameras and camera phones to medical scanners and security cameras.

Let's take a look at a few specific examples of Big Data:

- Facebook's 2.45 billion users upload more than 350 million new photos every day. They also click a "like" button or leave a comment more than 5 billion times every day. Facebook's data warehouse stores more than 300 petabytes of data, and the platform receives 600 terabytes of incoming data per day.
- The 2 billion users of Google's YouTube service upload more than 300 hours of video per minute. Google itself processes on average more than 70,000 search queries per second.
- In July 2020 industry analysts estimated that Twitter users sent some 550 million tweets per day.
- Autonomous cars generate up to 20 terabytes of data per car per day.

Characteristics of Big Data

Big Data has three distinct characteristics: volume, velocity, and variety. These characteristics distinguish Big Data from traditional data:

1. *Volume*: We have noted the huge volume of Big Data. Consider machine-generated data, which are generated in much larger quantities than nontraditional data. For example, sensors in a single jet engine can generate 10 terabytes of data in 30 minutes. (See our discussion of the Internet of Things in Chapter 8.) With more than 25,000 airline flights per day, the daily volume of data from just this single source is incredible. Smart electrical meters, sensors in heavy industrial equipment, and telemetry from automobiles compound the volume problem.
2. *Velocity*: The rate at which data flow into an organization is rapidly increasing. Velocity is critical because it increases the speed of the feedback loop between a company, its customers, its suppliers, and its business partners. For example, the Internet and mobile technology enable online retailers to compile histories not only on final sales, but on their customers' every click and interaction. Companies that can quickly use that information—for example, by recommending additional purchases—gain competitive advantage.
3. *Variety*: Traditional data formats tend to be structured and relatively well described, and they change slowly. Traditional data include financial market data, point-of-sale transactions, and much more. In contrast, Big Data formats change rapidly. They include satellite imagery, broadcast audio streams, digital music files, web page content, scans of government documents, and comments posted on social networks.

Irrespective of their source, structure, format, and frequency, Big Data are valuable. If certain types of data appear to have no value today, it is because we have not yet been able to analyze them effectively. For example, several years ago when Google began harnessing satellite imagery, capturing street views, and then sharing these geographical data for free, few people understood its value. Today, we recognize that such data are incredibly valuable because analyses of Big Data yield deep insights. We discuss analytics in detail in Chapter 12.

Issues with Big Data

Despite its extreme value, Big Data does have issues. In this section, we take a look at data integrity, data quality, and the nuances of analysis that are worth noting.

Big Data Can Come from Untrusted Sources.

As we discussed earlier, one of the characteristics of Big Data is variety, meaning that Big Data can come from numerous, widely varied sources. These sources may be internal or external to the organization. For example, a company might want to integrate data from unstructured sources such as e-mails, call center notes, and social media posts with structured data about its customers from its data warehouse. The question is, how trustworthy are those external sources of data? For example, how trustworthy is a Tweet? The data may come from an unverified source. Furthermore, the data itself, reported by the source, may be false or misleading.

Big Data Is Dirty.

Dirty data refers to inaccurate, incomplete, incorrect, duplicate, or erroneous data. Examples of such problems are misspelling of words, and duplicate data such as retweets or company press releases that appear multiple times in social media.

Suppose a company is interested in performing a competitive analysis using social media data. The company wants to see how often a competitor's product appears in social media outlets as well as the sentiments associated with those posts. The company notices that the number of positive posts about the competitor is twice as great as the number of positive posts about itself. This finding could simply be a case of the competitor pushing out its press releases to multiple sources; in essence, blowing its own horn. Alternatively, the competitor could be getting many people to retweet an announcement.

Big Data Changes, Especially in Data Streams.

Organizations must be aware that data quality in an analysis can change, or the data themselves can change, because the conditions under which the data are captured can change. For example, imagine a utility company that analyzes weather data and smart-meter data to predict customer power usage. What happens when the utility is analyzing these data in real time and it discovers that data are missing from some of its smart meters?

Managing Big Data

Big Data makes it possible to do many things that were previously much more difficult; for example, to spot business trends more rapidly and accurately, to prevent disease, to track crime, and so on. When Big Data is properly analyzed, it can reveal valuable patterns and information that were previously hidden because of the amount of work required to discover them. Leading corporations, such as Walmart and Google, have been able to process Big Data for years, but only at great expense. Today's hardware, cloud computing (see Technology Guide 3), and open-source software make processing Big Data affordable for most organizations.

For many organizations the first step toward managing data was to integrate information silos into a database environment and then to develop data warehouses for decision making. An *information silo* is an information system that does not communicate with other related information systems in an organization. After they completed this step, many organizations turned their attention to the business of information management—making sense of their rapidly expanding data. In recent years, Oracle, IBM, Microsoft, and SAP have spent billions of dollars purchasing software firms that specialize in data management and business analytics. (You will learn about business analytics in Chapter 12.)

In addition to existing data management systems, today many organizations employ NoSQL databases to process Big Data. Think of them as "not only SQL" (structured query language) databases. (We discuss SQL in section 5.6.)

As you have seen in this chapter, traditional relational databases such as Oracle and MySQL store data in tables organized into rows and columns. Recall that each row is associated with a unique record, and each column is associated with a field that defines an attribute of that account.

In contrast, NoSQL databases can manipulate structured as well as unstructured data as well as inconsistent or missing data. For this reason, NoSQL databases are particularly useful when working with Big Data. Hadoop and MapReduce are particularly useful when analyzing massive databases.

Hadoop (**http://hadoop.apache.org**) is not a type of database. Rather, it is a collection of programs that allow people to store, retrieve, and analyze very large data sets using massively parallel processing. *Massively parallel processing* is the coordinated processing of an application by multiple processors that work on different parts of the application, with each processor utilizing its own operating system and memory. As such, Hadoop enables users to access NoSQL databases, which can be spread across thousands of servers, without a reduction in performance. For example, a large database application that could take 20 hours of processing time on a centralized relational database system might take only a few minutes when using Hadoop's parallel processing.

MapReduce refers to the software procedure of dividing an analysis into pieces that can be distributed across different servers in multiple locations. MapReduce first distributes the analysis (map) and then collects and integrates the results back into a single report (reduce).

Many products use NoSQL databases, including Cassandra (**http://cassandra.apache.org**), CouchDB (**http://couchdb.apache.org**), and MongoDB (**www.mongodb.org**). Let's take a look at how eHarmony uses Redis's (**www.redis.io**) in-memory NoSQL database. An in-memory database is a DBMS that primarily relies on main memory (see Technology Guide 1) for data storage, in contrast to DBMSs that use hard-drive storage.

eHarmony (**www.eharmony.com**) uses Oracle's DBMS for cold data and Redis for hot data. *Cold data* refers to the storage of relatively inactive data that does not have to be accessed frequently or rapidly. *Hot data* refers to data that must be accessed frequently and rapidly. The eHarmony matching system applies analytics in near real time to quickly pair a candidate with a best-case potential match. Quickly serving up compatible matches requires rapid searches of personality trait data (i.e., Redis used with hot data). eHarmony's back-end business operations do not require high-speed access to data and therefore use Oracle with cold data.

Putting Big Data to Use

Modern organizations must manage Big Data and gain value from it. They can employ several strategies to achieve this objective.

Making Big Data Available.

Making Big Data available for relevant stakeholders can help organizations gain value. For example, consider open data in the public sector. Open data are accessible public data that individuals and organizations can use to create new businesses and solve complex problems. In particular, government agencies gather vast amounts of data, some of which are Big Data. Making those data available can provide economic benefits. In fact, an Open Data 500 study at the GovLab at New York University discovered 500 examples of U.S.-based companies whose business models depend on analyzing open government data.

Enabling Organizations to Conduct Experiments.

Big Data allows organizations to improve performance by conducting controlled experiments. For example, Amazon (and many other companies such as Google and LinkedIn) constantly experiments by offering slightly different looks on its website. These experiments are called A/B experiments, because each experiment has only two possible outcomes. Here is an example of an A/B experiment at Etsy (**www.etsy.com**), an online marketplace for vintage and handmade products.

MKT When Etsy analysts noticed that one of its web pages attracted customer attention but failed to maintain it, they looked more closely at the page and discovered that it had few "calls to action." (A call to action is an item, such as a button, on a web page that enables a customer to do something.) On this particular Etsy page, customers could leave, buy, search, or click on two additional product images. The analysts decided to show more product images on the page.

Consequently, one group of visitors to the page saw a strip across the top of the page that displayed additional product images. Another group saw only the two original product images. On the page with additional images, customers viewed more products and, significantly, bought more products. The results of this experiment revealed valuable information to Etsy.

Microsegmentation of Customers.

Segmentation of a company's customers means dividing them into groups that share one or more characteristics. Microsegmentation simply means dividing customers up into very small groups, or even down to the individual customer.

MKT For example, Paytronix Systems (**www.paytronix.com**) provides loyalty and rewards program software for thousands of different restaurants. Paytronix gathers restaurant guest data from a variety of sources beyond loyalty and gift programs, including social media. Paytronix analyzes this Big Data to help its restaurant clients microsegment their guests. Restaurant managers are now able to more precisely customize their loyalty and gift programs. Since they have taken these steps, they are noting improved profitability and customer satisfaction in their restaurants.

POM Creating New Business Models.

Companies are able to use Big Data to create new business models. For example, a commercial transportation company operated a substantial fleet of large long-haul trucks. The company recently placed sensors on all of its trucks. These sensors wirelessly communicate sizeable amounts of information to the company, a process called *telematics.* The sensors collect data on vehicle usage—including acceleration, braking, cornering, and so on—in addition to driver performance and vehicle maintenance.

By analyzing this Big Data, the company was able to improve the condition of its trucks through near-real-time analysis that proactively suggested preventive maintenance. The company was also able to improve the driving skills of its operators by analyzing their driving styles.

The transportation company then made its Big Data available to its insurance carrier. Using this data, the insurance carrier was able to perform a more precise risk analysis of driver behavior and the condition of the trucks. The carrier then offered the transportation company a new pricing model that lowered its premiums by 10 percent due to safety improvements enabled by analysis of the Big Data.

Organizations Can Analyze More Data.

In some cases, organizations can even process all of the data relating to a particular phenomenon, so they do not have to rely as much on sampling. Random sampling works well, but it is not as effective as analyzing an entire dataset. Random sampling also has some basic weaknesses. To begin with, its accuracy depends on ensuring randomness when collecting the sample data. However, achieving such randomness is problematic. Systematic biases in the process of data collection can cause results to be highly inaccurate. For example, consider political polling using landline phones. This sample tends to exclude people who use only cell phones. This bias can seriously skew the results because cell phone users are typically younger and more liberal than people who rely primarily on landline phones.

Big Data Used in the Functional Areas of the Organization

In this section, we provide examples of how Big Data is valuable to various functional areas in the firm.

HRM Human Resources.

Employee benefits, particularly health care, represent a major business expense. Consequently, some companies have turned to Big Data to better manage these benefits. Caesars Entertainment (**www.caesars.com**), for example, analyzes health-insurance claim data for its 65,000 employees and their covered family members. Managers can track thousands of variables that indicate how employees use medical services, such as the number of emergency room visits and whether employees choose a generic or brand name drug.

Consider the following scenario. Data revealed that too many employees with medical emergencies were being treated at hospital emergency rooms rather than at less expensive urgent-care facilities. The company launched a campaign to remind employees of the high cost of emergency room visits, and they provided a list of alternative facilities. Subsequently, 10,000 emergencies shifted to less expensive alternatives, for a total savings of $4.5 million.

Big Data is also having an impact on *hiring.* An example is Catalyst IT Services (**www.catalyte.io**), a technology outsourcing company that hires teams for programming jobs. Traditional recruiting is typically too slow, and hiring managers often subjectively choose candidates who are not the best fit for the job. Catalyst addresses this problem by requiring candidates to fill out an online assessment. It

then uses the assessment to collect thousands of data points about each candidate. In fact, the company collects more data based on *how* candidates answer than on *what* they answer.

For example, the assessment might give a problem requiring calculus to an applicant who is not expected to know the subject. How the candidate responds—laboring over an answer, answering quickly and then returning later, or skipping the problem entirely—provides insight into how that candidate might deal with challenges that he or she will encounter on the job. That is, someone who labors over a difficult question might be effective in an assignment that requires a methodical approach to problem solving, whereas an applicant who takes a more aggressive approach might perform better in a different job setting.

The benefit of this Big Data approach is that it recognizes that people bring different skills to the table and there is no one-size-fits-all person for any job. Analyzing millions of data points can reveal which attributes candidates bring to specific situations.

As one measure of success, employee turnover at Catalyst averages about 15 percent per year, compared with more than 30 percent for its U.S. competitors and more than 20 percent for similar companies overseas.

MKT Product Development.

Big Data can help capture customer preferences and put that information to work in designing new products. For example, Ford Motor Company (**www.ford.com**) was considering a "three blink" turn indicator that had been available on its European cars for years. Unlike the turn signals on its U.S. vehicles, this indicator flashes three times at the driver's touch and then automatically shuts off.

Ford decided that conducting a full-scale market research test on this blinker would be too costly and time consuming. Instead, it examined auto-enthusiast websites and owner forums to discover what drivers were saying about turn indicators. Using text-mining algorithms, researchers culled more than 10,000 mentions and then summarized the most relevant comments.

The results? Ford introduced the three-blink indicator on the new Ford Fiesta in 2010, and by 2013 it was available on most Ford products. Although some Ford owners complained online that they have had trouble getting used to the new turn indicator, many others defended it. Ford managers note that the use of text-mining algorithms was critical in this effort because they provided the company with a complete picture that would not have been available using traditional market research.

POM Operations.

For years, companies have been using information technology to make their operations more efficient. Consider United Parcel Service (UPS). The company has long relied on data to improve its operations. Specifically, it uses sensors in its delivery vehicles that can, among other things, capture the truck's speed and location, the number of times it is placed in Reverse, and whether the driver's seat belt is buckled. These data are uploaded at the end of each day to a UPS data center, where they are analyzed overnight. By combining GPS information and data from sensors installed on more than 46,000 vehicles, UPS reduced fuel consumption by 8.4 million gallons, and it cut 85 million miles off its routes.

MKT Marketing.

Marketing managers have long used data to better understand their customers and to target their marketing efforts more directly. Today, Big Data enables marketers to craft much more personalized messages.

The United Kingdom's InterContinental Hotels Group (IHG; **www.ihg.com**) gathered details about the members of its Priority Club rewards program, such as income levels and whether members prefer family-style or business-traveler accommodations. The company then consolidated all this information with information obtained from social media into a single data warehouse. Using its data warehouse and analytics software, the hotelier launched a new marketing campaign. Where previous marketing campaigns generated, on average, between 7 and 15 customized marketing messages, the new campaign generated more than 1,500. IHG rolled out these messages in stages to an initial core of 12 customer groups, each of which is defined by 4,000 attributes. One group, for example, tends to stay on weekends, redeem reward points for gift cards, and register through IHG marketing partners. Using this information, IHG sent these customers a marketing message that alerted them to local weekend events.

The campaign proved to be highly successful. It generated a 35 percent higher rate of customer conversions, or acceptances, than previous similar campaigns.

POM Government Operations.

Consider the United Kingdom. According to the INRIX Traffic Scorecard, although the United States has the worst congestion on average, London topped the world list for metropolitan areas. In London, drivers wasted an average of 101 hours per year in gridlock. Congestion is bad for business. The INRIX study estimated that the cost to the U.K. economy would be £307 billion between 2013 and 2030.

Congestion is also harmful to urban resilience, negatively affecting both environmental and social sustainability in terms of emissions, global warming, air quality, and public health. As for the livability of a modern city, congestion is an important component of the urban transport user experience (UX).

Calculating levels of UX satisfaction at any given time involves solving a complex equation with a range of key variables and factors: total number of transport assets (road and rail capacity, plus parking spaces), users (vehicles, pedestrians), incidents (roadwork, accidents, breakdowns), plus expectations (anticipated journey times and passenger comfort).

The growing availability of Big Data sources within London—for example, traffic cameras and sensors on cars and roadways—can help to create a new era of smart transport. Analyzing this Big Data offers new ways for traffic analysts in London to "sense the city" and enhance transport via real-time estimation of traffic patterns and rapid deployment of traffic management strategies.

Before you go on…

1. Define Big Data.

2. Describe the characteristics of Big Data.

3. Describe how companies can use Big Data to a gain competitive advantage.

5.4 Data Warehouses and Data Marts

Today, the most successful companies are those that can respond quickly and flexibly to market changes and opportunities. A key to this response is the effective and efficient use of data and information by analysts and managers. The challenge is to provide users with access to corporate data so they can analyze the data to make better decisions. Let's consider an example. If the manager of a local bookstore wanted to know the profit margin on used books at her store, then she could obtain that information from her database using SQL or query-by-example (QBE). QBE is a method of creating database queries that allows the user to search for documents based on an example in the form of a selected string of text or in the form of a document name or a list of documents. However, if she needed to know the trend in the profit margins on used books over the past 10 years, then she would have to construct a very complicated SQL or QBE query.

Author Lecture Videos are available exclusively in *WileyPLUS*.

Apply the Concept activities are available in the Appendix and in *WileyPLUS*.

This example illustrates several reasons why organizations are building data warehouses and data marts. First, the bookstore's databases contain the necessary information to answer the manager's query, but the information is not organized in a way that makes it easy for her to find what she needs. Therefore, complicated queries might take a long time to answer, and they also might degrade the performance of the databases. Second, transactional databases are designed to be updated. This update process requires extra processing. Data warehouses and data marts are read-only. Therefore, the extra processing is eliminated because data already contained in the data warehouse are not updated. Third, transactional databases are designed to access a single record at a time. In contrast, data warehouses are designed to access large groups of related records.

To solve these problems, companies are using a variety of tools with data warehouses and data marts to make it easier and faster for users to access, analyze, and query data. You will learn about these tools in Chapter 12 on business analytics.

Describing Data Warehouses and Data Marts

In general, data warehouses and data marts support business analytics applications. As you will see in Chapter 12, business analytics encompasses a broad category of applications, technologies, and processes for gathering, storing, accessing, and analyzing data to help business users make better decisions. A **data warehouse** is a repository of historical data that are organized by subject to support decision makers within the organization.

Because data warehouses are so expensive, they are used primarily by large companies. A **data mart** is a low-cost, scaled-down version of a data warehouse that is designed for the end-user needs in a strategic business unit (SBU) or an individual department. Data marts can be implemented more quickly than data warehouses, often in less than 90 days. Furthermore, they support local rather than central control by conferring power on the user group. Typically, groups that need a single or a few business analytics applications require only a data mart rather than a data warehouse.

The basic characteristics of data warehouses and data marts include the following:

- *Organized by business dimension or subject.* Data are organized by subject—for example, by customer, vendor, product, price level, and region. This arrangement differs from transactional systems, where data are organized by business process such as order entry, inventory control, and accounts receivable.
- *Use online analytical processing.* Typically, organizational databases are oriented toward handling transactions. That is, databases use *online transaction processing* (OLTP), where business transactions are processed online as soon as they occur. The objectives are speed and efficiency, which are critical to a successful Internet-based business operation. In contrast, data warehouses and data marts, which are designed to support decision makers but not OLTP, use online analytical processing (OLAP), which involves the analysis of accumulated data by end users. We consider OLAP in greater detail in Chapter 12.
- *Integrated.* Data are collected from multiple systems and are then integrated around subjects. For example, customer data may be extracted from internal (and external) systems and then integrated around a customer identifier, thereby creating a comprehensive view of the customer.
- *Time variant.* Data warehouses and data marts maintain historical data; that is, data that include time as a variable. Unlike transactional systems, which maintain only recent data (such as for the last day, week, or month), a warehouse or mart may store years of data. Organizations use historical data to detect deviations, trends, and long-term relationships.
- *Nonvolatile.* Data warehouses and data marts are nonvolatile—that is, users cannot change or update the data. Therefore, the warehouse or mart reflects history, which, as we just saw, is critical for identifying and analyzing trends. Warehouses and marts are updated, but through IT-controlled load processes rather than by users.
- *Multidimensional.* Typically, the data warehouse or mart uses a multidimensional data structure. Recall that relational databases store data in two-dimensional tables. In contrast, data warehouses and marts store data in more than two dimensions. For this reason, the data are said to be stored in a **multidimensional structure**. A common representation for this multidimensional structure is the *data cube*.

The data in data warehouses and marts are organized by *business dimensions*, which are subjects such as product, geographic area, and time period that represent the edges of the data cube. If you look ahead to **Figure 5.6** for an example of a data cube, you see that the product dimension is composed of nuts, screws, bolts, and washers; the geographic area dimension is composed of East, West, and Central; and the

time period dimension is composed of 2016, 2017, and 2018. Users can view and analyze data from the perspective of these business dimensions. This analysis is intuitive because the dimensions are presented in business terms that users can easily understand.

A Generic Data Warehouse Environment

The environment for data warehouses and marts includes the following:

- Source systems that provide data to the warehouse or mart
- Data-integration technology and processes that prepare the data for use
- Different architectures for storing data in an organization's data warehouse or data marts
- Different tools and applications for the variety of users. (You will learn about these tools and applications in Chapter 12.)
- **Metadata** (data about the data in a repository), data quality, and governance processes that ensure that the warehouse or mart meets its purposes

Figure 5.4 depicts a generic data warehouse or data mart environment. Let's drill down into the component parts.

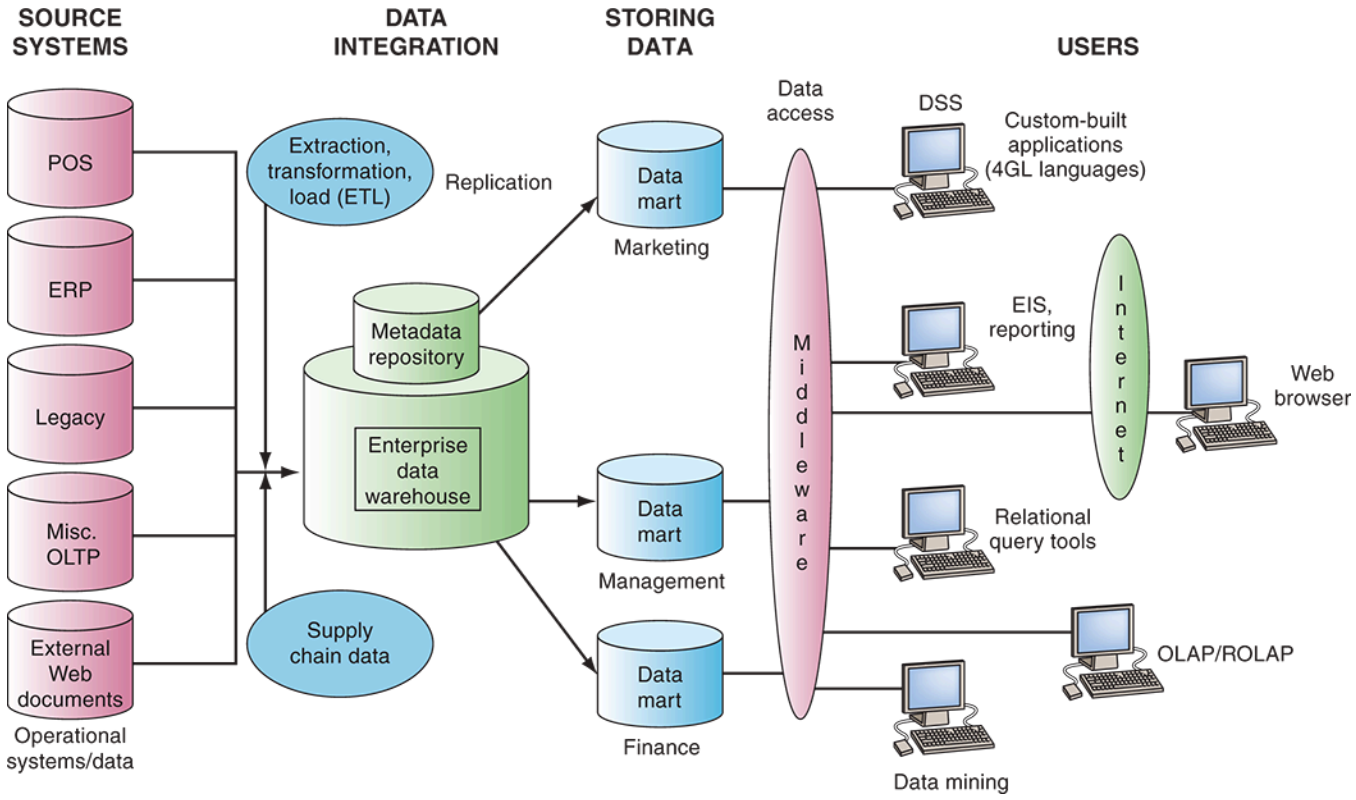


FIGURE 5.4 Data warehouse framework.

Source Systems.

There is typically some "organizational pain point"—that is, a business need—that motivates a firm to develop its BI capabilities. Working backward, this pain leads to information requirements, BI applications, and requirements for source system data. These data requirements can range from a single source system, as in the case of a data mart, to hundreds of source systems, as in the case of an enterprise-wide data warehouse.

Modern organizations can select from a variety of source systems, including operational/transactional systems, enterprise resource planning (ERP) systems, website data, third-party data (e.g., customer demographic data), and more. The trend is to include more types of data (e.g., sensing data from RFID tags). These source systems often use different software packages (e.g., IBM, Oracle), and they store data in different formats (e.g., relational, hierarchical).

A common source for the data in data warehouses is the company's operational databases, which can be relational databases. To differentiate between relational databases and multidimensional data warehouses and marts, imagine your company manufactures four products—nuts, screws, bolts, and washers—and has sold them in three territories—East, West, and Central—for the previous three years—2019, 2020, and 2021. In a relational database, these sales data would resemble **Figure 5.5**(a) through (c). In a multidimensional database, in contrast, these data would be represented by a three-dimensional matrix (or data cube), as depicted in **Figure 5.6**. This matrix represents sales *dimensioned by* products, regions, and year. Notice that Figure 5.5(a) presents only sales for 2016. Sales for 2017 and 2018 are presented in Figure 5.5(b) and (c), respectively. **Figure 5.7**(a) through (c) illustrates the equivalence between these relational and multidimensional databases.

Unfortunately, many source systems that have been in use for years contain "bad data"—for example, missing or incorrect data—and they are poorly documented. As a result, data-profiling software should be used at the beginning of a warehousing project to better understand the data. Among other things, this software can provide statistics on missing data, identify possible primary and foreign keys, and reveal how derived values—for example, column 3 = column 1 + column 2—are calculated. Subject area database specialists such as marketing and human resources personnel can also assist in understanding and accessing the data in source systems.

Organizations need to address other source systems issues as well. For example, many organizations maintain multiple systems that contain some of the same data. These enterprises need to select the best system as the source system. Organizations must also decide how granular,

or detailed, the data should be. For example, does the organization need daily sales figures or data for individual transactions? The conventional wisdom is that it is best to store data at a highly granular level because someone will likely request those data at some point.

(a) 2019

Product	Region	Sales
Nuts	East	50
Nuts	West	60
Nuts	Central	100
Screws	East	40
Screws	West	70
Screws	Central	80
Bolts	East	90
Bolts	West	120
Bolts	Central	140
Washers	East	20
Washers	West	10
Washers	Central	30

(b) 2020

Product	Region	Sales
Nuts	East	60
Nuts	West	70
Nuts	Central	110
Screws	East	50
Screws	West	80
Screws	Central	90
Bolts	East	100
Bolts	West	130
Bolts	Central	150
Washers	East	30
Washers	West	20
Washers	Central	40

(c) 2021

Product	Region	Sales
Nuts	East	70
Nuts	West	80
Nuts	Central	120
Screws	East	60
Screws	West	90
Screws	Central	100
Bolts	East	110
Bolts	West	140
Bolts	Central	160
Washers	East	40
Washers	West	30
Washers	Central	50

FIGURE 5.5 Relational databases.

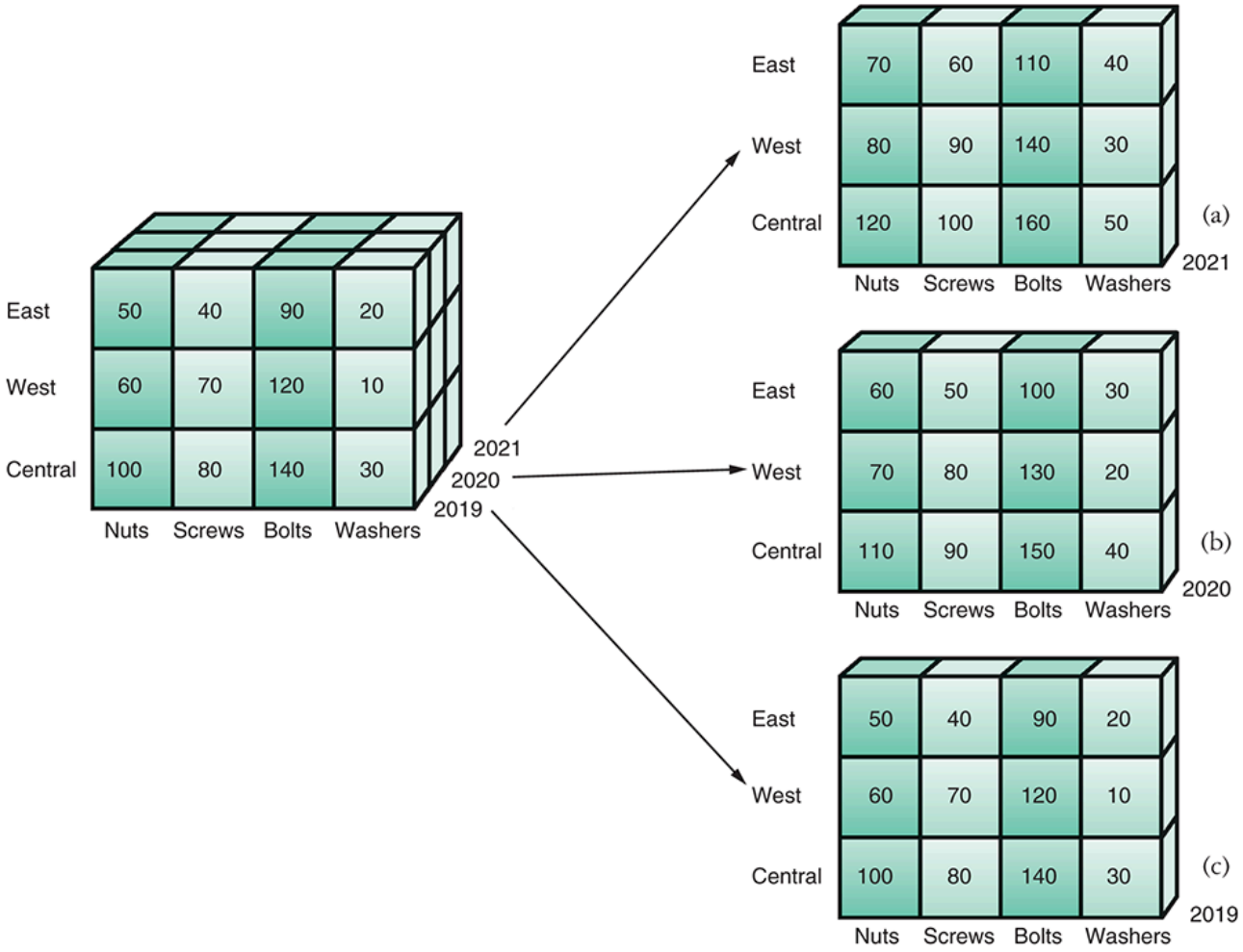


FIGURE 5.6 Data cube.

Data Integration.

In addition to storing data in their source systems, organizations need to *e*xtract the data, *t*ransform them, and then *l*oad them into a data mart or warehouse. This process is often called ETL, although the term *data integration* is increasingly being used to reflect the growing number of ways that source system data can be handled. For example, in some cases, data are extracted, loaded into a mart or warehouse, and then transformed (i.e., ELT rather than ETL).

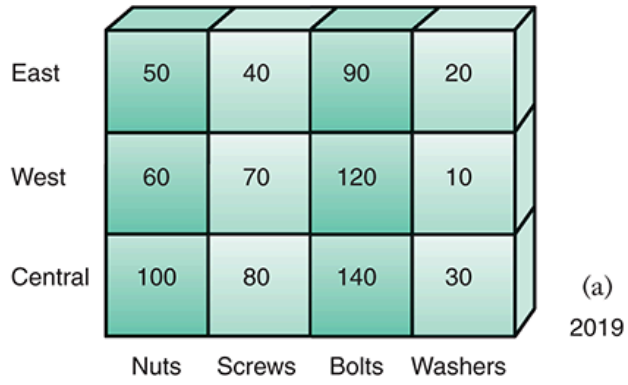
Data extraction can be performed either by handwritten code such as SQL queries or by commercial data-integration software. Most companies employ commercial software. This software makes it relatively easy to (1) specify the tables and attributes in the source systems

that are to be used; (2) map and schedule the movement of the data to the target, such as a data mart or warehouse; (3) make the required transformations; and, ultimately, (4) load the data.

After the data are extracted, they are transformed to make them more useful. For example, data from different systems may be integrated around a common key, such as a customer identification number. Organizations adopt this approach to create a 360-degree view of all of their interactions with their customers. As an example of this process, consider a bank. Customers can engage in a variety of interactions: visiting a branch, banking online, using an ATM, obtaining a car loan, and more. The systems for these touch points—defined as the numerous ways that organizations interact with customers, such as e-mail, the Web, direct contact, and the telephone—are typically independent of one another. To obtain a holistic picture of how customers are using the bank, the bank must integrate the data from the various source systems into a data mart or warehouse.

Product	Region	Sales
Nuts	East	50
Nuts	West	60
Nuts	Central	100
Screws	East	40
Screws	West	70
Screws	Central	80
Bolts	East	90
Bolts	West	120
Bolts	Central	140
Washers	East	20
Washers	West	10
Washers	Central	30

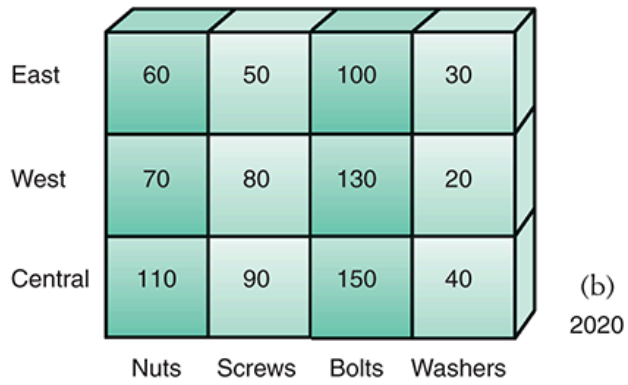
2019



(a) 2019

Product	Region	Sales
Nuts	East	60
Nuts	West	70
Nuts	Central	110
Screws	East	50
Screws	West	80
Screws	Central	90
Bolts	East	100
Bolts	West	130
Bolts	Central	150
Washers	East	30
Washers	West	20
Washers	Central	40

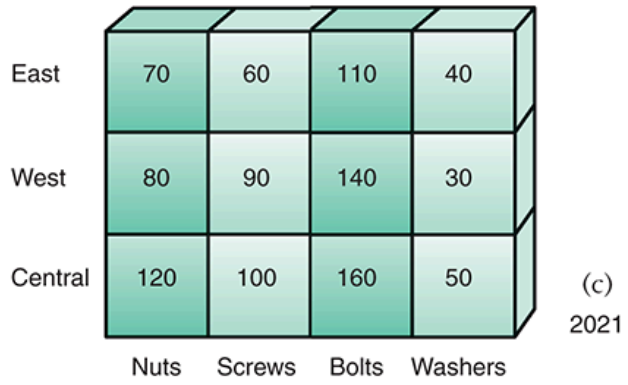
2020



(b) 2020

Product	Region	Sales
Nuts	East	70
Nuts	West	80
Nuts	Central	120
Screws	East	60
Screws	West	90
Screws	Central	100
Bolts	East	110
Bolts	West	140
Bolts	Central	160
Washers	East	40
Washers	West	30
Washers	Central	50

2021



(c) 2021

FIGURE 5.7 Equivalence between relational and multidimensional databases.

Other kinds of transformations also take place. For example, format changes to the data may be required, such as using *male* and *female* to

denote gender, as opposed to 0 and 1 or M and F. Aggregations may be performed, say on sales figures, so that queries can use the summaries rather than recalculating them each time. Data-cleansing software may be used to clean up the data; for example, eliminating duplicate records for the same customer.

Finally, data are loaded into the warehouse or mart during a specified period known as the "load window." This window is becoming smaller as companies seek to store ever-fresher data in their warehouses. For this reason, many companies have moved to real-time data warehousing, where data are moved using data-integration processes from source systems to the data warehouse or mart almost instantly. For example, within 15 minutes of a purchase at Walmart, the details of the sale have been loaded into a warehouse and are available for analysis.

Storing the Data.

Organizations can choose from a variety of architectures to store decision-support data. The most common architecture is *one central enterprise data warehouse*, without data marts. Most organizations use this approach because the data stored in the warehouse are accessed by all users, and they represent the single version of the truth.

Another architecture is *independent data marts.* These marts store data for a single application or a few applications, such as marketing and finance. Organizations that employ this architecture give only limited thought to how the data might be used for other applications or by other functional areas in the organization. Clearly this is a very application-centric approach to storing data.

The independent data mart architecture is not particularly effective. Although it may meet a specific organizational need, it does not reflect an enterprise-wide approach to data management. Instead, the various organizational units create independent data marts. Not only are these marts expensive to build and maintain, but they often contain inconsistent data. For example, they may have inconsistent data definitions such as: What is a customer? Is a particular individual a potential or a current customer? They might also use different source systems, which can have different data for the same item, such as a customer address (if the customer had moved). Although independent data marts are an organizational reality, larger companies have increasingly moved to data warehouses.

Still another data warehouse architecture is the *hub and spoke.* This architecture contains a central data warehouse that stores the data plus multiple dependent data marts that source their data from the central repository. Because the marts obtain their data from the central repository, the data in these marts still comprise the *single version of the truth* for decision-support purposes.

The dependent data marts store the data in a format that is appropriate for how the data will be used and for providing faster response times to queries and applications. As you have learned, users can view and analyze data from the perspective of business dimensions and measures. This analysis is intuitive because the dimensions are presented in business terms that users can easily understand.

Metadata.

It is important to maintain data about the data, known as *metadata*, in the data warehouse. Both the IT personnel who operate and manage the data warehouse and the users who access the data require metadata. IT personnel need information about data sources; database, table, and column names; refresh schedules; and data-usage measures. Users' needs include data definitions, report and query tools, report distribution information, and contact information for the help desk.

Data Quality.

The quality of the data in the warehouse must meet users' needs. If it does not, then users will not trust the data and ultimately will not use it. Most organizations find that the quality of the data in source systems is poor and must be improved before the data can be used in the data warehouse. Some of the data can be improved with data-cleansing software. The better, long-term solution, however, is to improve the quality at the source system level. This approach requires the business owners of the data to assume responsibility for making any necessary changes to implement this solution.

To illustrate this point, consider the case of a large hotel chain that wanted to conduct targeted marketing promotions using zip code data it collected from its guests when they checked in. When the company analyzed the zip code data, they discovered that many of the zip codes were 99999. How did this error occur? The answer is that the clerks were not asking customers for their zip codes, but they needed to enter something to complete the registration process. A short-term solution to this problem was to conduct the marketing campaign using city and state data instead of zip codes. The long-term solution was to make certain the clerks entered the actual zip codes. The latter solution required the hotel managers to assume responsibility for making certain their clerks entered the correct data.

Governance.

To ensure that BI is meeting their needs, organizations must implement *governance* to plan and control their BI activities. Governance requires that people, committees, and processes be in place. Companies that are effective in BI governance often create a senior-level committee composed of vice presidents and directors who (1) ensure that the business strategies and BI strategies are in alignment, (2) prioritize projects, and (3) allocate resources. These companies also establish a middle management–level committee that oversees the various projects in the BI portfolio to ensure that these projects are being completed in accordance with the company's objectives. Finally, lower-level operational committees perform tasks such as creating data definitions and identifying and solving data problems. All of these committees rely on the collaboration and contributions of business users and IT personnel.

Users.

Once the data are loaded in a data mart or warehouse, they can be accessed. At this point, the organization begins to obtain business value from BI; all of the prior stages constitute creating BI infrastructure.

There are many potential BI users, including IT developers; frontline workers; analysts; information workers; managers and executives; and suppliers, customers, and regulators. Some of these users are *information producers,* whose primary role is to create information for other users. IT developers and analysts typically fall into this category. Other users—including managers and executives—are *information consumers*, because they use information created by others.

Companies have reported hundreds of successful data-warehousing applications. You can read client success stories and case studies at the websites of vendors such as NCR Corp. (**www.ncr.com**) and Oracle (**www.oracle.com**). For a more detailed discussion, visit the Data Warehouse Institute (**http://tdwi.org**). The benefits of data warehousing include the following:

- End users can access needed data quickly and easily through Web browsers because these data are located in one place.
- End users can conduct extensive analysis with data in ways that were not previously possible.
- End users can obtain a consolidated view of organizational data.

These benefits can improve business knowledge, provide competitive advantage, enhance customer service and satisfaction, facilitate decision making, and streamline business processes.

Despite their many benefits, data warehouses have some limitations. IT's About Business 5.2 points out these limitations and considers an emerging solution; namely, data lakes.

IT's About Business 5.2

MIS Data Lakes

Most large organizations have an enterprise data warehouse (EDW), which contains data from other enterprise systems such as customer relationship management (CRM), inventory, and sales transaction systems. With EDWs, organizations maintain the data using traditional databases, meaning that the EDW is built upon labeled rows and columns of data. EDWs are the primary mechanism in many organizations for performing analytics, reporting, and operations.

Despite their benefits to organizations, EDWs do have problems. Specifically, they require organizations to design the data model—called the schema—before they load any data into the EDW. A *database schema* defines the structure of both the database and the data contained in that database. For example, in the case of relational databases, the schema specifies the tables and fields of the database. The schema also describes the content and structure of the physical data stored. These descriptions are called *metadata*.

As a result, EDWs are relatively inflexible and can answer only a limited number of questions. It is therefore difficult for business analysts and data scientists who rely on EDWs to ask ad hoc questions of the data.

It is also difficult for EDWs to manage new sources of data, such as streaming data from sensors (see the Internet of Things in Chapter 8), and social media data such as blog postings, ratings, recommendations, product reviews, Tweets, photographs, and video clips.

EDWs are also too rigid to be effective with Big Data, with its huge data volumes, broad variety of data, and high data velocity. As a result of these problems, organizations have begun to realize that EDWs cannot meet all of their business needs.

The emergence of systems such as Apache Hadoop (**http://hadoop.apache.org**) has enabled organizations to implement parallel searches on large data repositories to greatly speed up operations on the data. Hadoop provided the impetus for the creation of data lakes.

A **data lake** is a central repository that stores all of an organization's data, regardless of the source or format of those data. Data lakes receive data in any format, both structured and unstructured. Also, the data do not have to be consistent. For example, organizations might have the same type of information in different data formats, depending on where the data originate.

Organizations typically use Hadoop to build their data lakes. They can then employ a variety of storage and processing tools to extract value quickly from these data lakes and to inform key business decisions.

Organizations do not transform the data before entering them into the data lake as they would for an EDW. In fact, the structure of the data is not known when the data are fed into the data lake. Rather, it is discovered only when the data are read, meaning that users do not model the data until they actually use it. This process is more flexible, and it makes it easier for users to discover new data and to enter new data sources into the data lake.

Data lakes provide many benefits for organizations:

- Organizations can derive value from unlimited types of data.
- Organizations have no limits on how they can query the data.
- Organizations do not create data silos. Instead, data lakes provide a single, unified view of data across the organization.

To load data into a data lake, organizations should take these steps:

- Define the incoming data from a business perspective.
- Document the context, origin, and frequency of the incoming data.
- Classify the security level (public, internal, sensitive, restricted) of the incoming data.
- Document the creation, usage, privacy, regulatory, and encryption business rules that apply to the incoming data.
- Identify the owner (sponsor) of the incoming data.
- Identify the data steward(s) who monitor and maintain the datasets.

After organizations follow these steps, they load all the data into a large table. Each piece of data—whether a customer's name, a photograph, or a Facebook post—is placed in an individual cell. It does not matter where in the data lake that cell is located, where the data came from, or their format, because metadata tags connect all of the data. Organizations can add or change these tags as requirements evolve. Further, they can assign multiple tags to the same piece of data. Because the rules for storing the data do not need to be defined in advance, there is no need for expensive and time-consuming data modeling.

Organizations can also protect sensitive information by specifying who has access to the data in each cell, and under what circumstances, as the data are loaded. For example, a retail operation might make cells containing customers' names and contact data available to sales and customer service. At the same time, however, it might make the cells containing more sensitive, personally identifiable information or financial data available only to the finance department. In that way, when users run queries on the data, their access rights restrict which data they can view.

It is very important to note that organizations use both EDWs and data lakes. To understand this arrangement, let's distinguish between "small data" and "Big Data" questions. A small data question would be: What is the total revenue for the northeast region in 2020? This question is easily and quickly answered by an EDW because the data are well defined.

A Big Data question would be: Describe the detailed customer relationship over the past three years for a high-value customer who has moved her business to another firm. This question is a much better fit for a data lake because the variables are not clear from the outset and will probably include unstructured data such as email messages and audio clips. This query would be very difficult to answer with an EDW.

Many firms use data lakes as a holding area for data that it does not plan to use immediately but that may be valuable later. An example is archiving data for regulatory data-retention requirements. In another example, companies might use their data lakes to

test assumptions on massive volumes of data and then extract and load the most useful data into their EDWs for decision making.

There are many examples of data lakes in practice. Let's take a look at how L'Oréal (**www.lorealparisusa.com**) employs its data lake.

POM MKT L'Oréal, a 100-year-old cosmetics industry leader, owns more than 40 brands and must analyze a vast amount of data, including 7 billion products manufactured annually, 50 million data points created each day, and 500 patents filed each year. The firm relies on scientists and marketing professionals to work together to create several thousand new formulas every year. The company must also ensure that its products are safe for humans. This process requires analyzing data about product formulas and raw materials in addition to what consumers think of the new formulas.

To accomplish its goals, L'Oréal employed Talend (**www.talend.com**), a leading cloud-based data integration company, to create a data lake on Microsoft Azure. The platform integrates structured laboratory data with varying, often raw, unstructured data, such as images of models using L'Oréal cosmetics. The data are available in real time, and the data lake is refreshed several times every day.

ACCT L'Oréal developed its first application for the finance department to address the economic management of research. The application's dashboards displayed all of the key performance indicators for research-related activities and their associated costs, such as tests for product certification.

L'Oréal's next application addressed research into the impact of its products on the human microbiome, which consists of the genetic material of all of the microbes—bacteria, fungi, protozoa, and viruses—that live on and inside the human body. Another application involves products that can counter the effects of pollution on the skin.

Sources: Compiled from A. Thusoo, "Data Lakes and Data Warehouses: The Two Sides of a Modern Cloud Data Platform," *Forbes*, July 7, 2020; C. Foot, "Key Factors for Successful Data Lake Implementation," *TechTarget*, July 6, 2020; V. Combs, "L'Oréal's New Data Lake Holds 100 Years of Product Development Research," *TechRepublic*, October 30, 2019; "Essential Guide to Data Lakes," *Matillion*, 2019; S. Wooledge, "Data Lakes and Data Warehouses: Why You Need Both," *Arcadia Data*, October 11, 2018; T. King, "Three Key Data Lake Trends to Stay on Top of This Year," *Solutions Review*, May 11, 2018; T. Olavsrud, "6 Data Analytics Trends that Will Dominate 2018," *CIO*, March 15, 2018; P. Tyaqi and H. Demirkan, "Data Lakes: The Biggest Big Data Challenges," *Analytics Magazine*, September/October 2017; M. Hagstroem, M. Roggendorf, T. Saleh, and J. Sharma, "A Smarter Way to Jump into Data Lakes," *McKinsey and Company*, August 2017; P. Barth, "The New Paradigm for Big Data Governance," *CIO*, May 11, 2017; N. Mikhail, "Why Big Data Kills Businesses," *Fortune*, February 28, 2017; "Architecting Data Lakes," *Zaloni*, February 21, 2017; D. Kim, "Successful Data Lakes: A Growing Trend," *The Data Warehousing Institute*, February 16, 2017; L. Hester, "Maximizing Data Value with a Data Lake," *Data Science Central*, April 20, 2016.

Questions

1. **Discuss the advantages and disadvantages of enterprise data warehouses.**
2. **Describe the advantages and disadvantages of data lakes.**
3. **Why don't organizations use enterprise data warehouses for managing Big Data?**

Before you go on…

1. Differentiate between data warehouses and data marts.

2. Describe the characteristics of a data warehouse.

3. What are three possible architectures for data warehouses and data marts in an organization?

5.5 Knowledge Management

As we have noted throughout this text, data and information are vital organizational assets. Knowledge is a vital asset as well. Successful managers have always valued and used intellectual assets. These efforts may not have been systematic, however, and they may not have ensured that knowledge was shared and dispersed in a way that benefited the overall organization. Moreover, industry analysts estimate that most of a company's knowledge assets are not housed in relational databases. Instead, they are dispersed in e-mail, word processing documents, spreadsheets, presentations on individual computers, and in people's heads. This arrangement makes it extremely difficult for companies to access and integrate this knowledge. The result frequently is less effective decision making.

Author Lecture Videos are available exclusively in *WileyPLUS*.

Apply the Concept activities are available in the Appendix and in *WileyPLUS*.

Concepts and Definitions

Knowledge management (KM) is a process that helps organizations manipulate important knowledge that comprises part of the organization's memory, usually in an unstructured format. For an organization to be successful, knowledge, as a form of capital, must exist in a format that can be exchanged among persons. It must also be able to grow.

Knowledge.

In the information technology context, knowledge is distinct from data and information. As you learned in Chapter 1, data are a collection of facts, measurements, and statistics; information is organized or processed data that are timely and accurate. Knowledge is information that is *contextual*, *relevant*, and *useful*. Simply put, knowledge is information in action. **Intellectual capital** (or **intellectual assets**) is another term for knowledge.

To illustrate, a bulletin listing all of the courses offered by your university during one semester would be considered *data.* When you register, you process the data from the bulletin to create your schedule for the semester. Your schedule would be considered *information.* Awareness of your work schedule, your major, your desired social schedule, and characteristics of different faculty members could be construed as *knowledge*, because it can affect the way you build your schedule. You see that this awareness is contextual and relevant (to developing an optimal schedule of classes) as well as useful (it can lead to changes in your schedule). The implication is that knowledge has strong experiential and reflective elements that distinguish it from information in a given context. Unlike information, knowledge can be used to solve a problem.

Numerous theories and models classify different types of knowledge. In the next section, we will focus on the distinction between explicit knowledge and tacit knowledge.

Explicit and Tacit Knowledge.

Explicit knowledge deals with more objective, rational, and technical knowledge. In an organization, explicit knowledge consists of the policies, procedural guides, reports, products, strategies, goals, core competencies, and IT infrastructure of the enterprise. In other words, explicit knowledge is the knowledge that has been codified (documented) in a form that can be distributed to others or transformed into a process or a strategy. A description of how to process a job application that is documented in a firm's human resources policy manual is an example of explicit knowledge.

In contrast, **tacit knowledge** is the cumulative store of subjective or experiential learning. In an organization, tacit knowledge consists of an organization's experiences, insights, expertise, know-how, trade secrets, skill sets, understanding, and learning. It also includes the organizational culture, which reflects the past and present experiences of the organization's people and processes, as well as the organization's prevailing values. Tacit knowledge is generally imprecise and costly to transfer. It is also highly personal. Finally, because it is unstructured, it is difficult to formalize or codify, in contrast to explicit knowledge. A salesperson who has worked with particular customers over time and has come to know their needs quite well would possess extensive tacit knowledge. This knowledge is typically not recorded. In fact, it might be difficult for the salesperson to put into writing, even if he or she were willing to share it.

Knowledge Management Systems

The goal of knowledge management is to help an organization make the most productive use of the knowledge it has accumulated. Historically, management information systems have focused on capturing, storing, managing, and reporting explicit knowledge. Organizations now realize they need to integrate explicit and tacit knowledge into formal information systems. **Knowledge management systems (KMSs)** refer to the use of modern information technologies—the Internet, intranets, extranets, and databases—to systematize, enhance, and expedite knowledge management both within one firm and among multiple firms. KMSs are intended to help an organization cope with turnover, rapid change, and downsizing by making the expertise of the organization's human capital widely accessible.

Organizations can realize many benefits with KMSs. Most importantly, they make *best practices*—the most effective and efficient ways accomplishing business processes—readily available to a wide range of employees. Enhanced access to best-practice knowledge improves overall organizational performance. For example, account managers could make available their tacit knowledge about how best to manage large accounts. The organization could then use this knowledge when it trains new account managers. Other benefits include enhanced customer service, more efficient product development, and improved employee morale and retention.

At the same time, however, implementing effective KMSs presents several challenges. First, employees must be willing to share their personal tacit knowledge. To encourage this behavior, organizations must create a knowledge management culture that rewards employees who add their expertise to the knowledge base. Second, the organization must continually maintain and upgrade its knowledge base. Specifically, it must incorporate new knowledge and delete old, outdated knowledge. Finally, companies must be willing to invest in the resources needed to carry out these operations.

The KMS Cycle

A functioning KMS follows a cycle that consists of six steps (see **Figure 5.8**). The reason the system is cyclical is that knowledge is dynamically refined over time. The knowledge in an effective KMS is never finalized because the environment changes over time and knowledge must be updated to reflect these changes. The cycle works as follows:

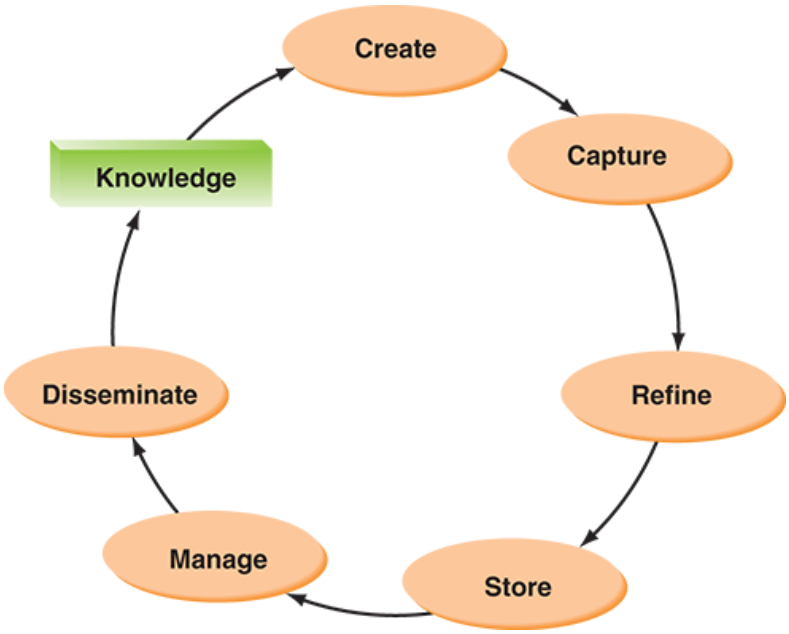


FIGURE 5.8 The knowledge management system cycle.

1. *Create knowledge.* Knowledge is created as people determine new ways of doing things or develop know-how. Sometimes external knowledge is brought in.
2. *Capture knowledge.* New knowledge must be identified as valuable and be presented in a reasonable way.
3. *Refine knowledge.* New knowledge must be placed in context so that it is actionable. This is where tacit qualities (human insights) must be captured along with explicit facts.
4. *Store knowledge.* Useful knowledge must then be stored in a reasonable format in a knowledge repository so that other people in the organization can access it.
5. *Manage knowledge.* Like a library, the knowledge must be kept current. Therefore, it must be reviewed regularly to verify that it is relevant and accurate.
6. *Disseminate knowledge.* Knowledge must be made available in a useful format to anyone in the organization who needs it, anywhere and any time.

Before you go on...

1. What is knowledge management?

2. What is the difference between tacit knowledge and explicit knowledge?

3. Describe the knowledge management system cycle.

5.6 Appendix: Fundamentals of Relational Database Operations

There are many operations possible with relational databases. In this section, we discuss three of these operations: query languages, normalization, and joins.

Author Lecture Videos are available exclusively in *WileyPLUS*.

Apply the Concept activities are available in the Appendix and in *WileyPLUS*.

As you have seen in this chapter, a relational database is a collection of interrelated two-dimensional tables consisting of rows and columns. Each row represents a record, and each column (or field) represents an attribute (or characteristic) of that record. Every record in the database must contain at least one field that uniquely identifies that record so that it can be retrieved, updated, and sorted. This identifier field, or group of fields, is called the primary key. In some cases, locating a particular record requires the use of secondary keys. A secondary key is another field that has some identifying information, but typically does not uniquely identify the record. A foreign key is a field (or group of fields) in one table that matches the primary key value in a row of another table. A foreign key is used to establish and enforce a link between two tables.

These related tables can be joined when they contain common columns. The uniqueness of the primary key tells the DBMS which records are joined with others in related tables. This feature allows users great flexibility in the variety of queries they can make. Despite these features, however, the relational database model has some disadvantages. Because large-scale databases can be composed of many interrelated tables, the overall design can be complex, leading to slow search and access times.

Query Languages

The most commonly performed database operation is searching for information. **Structured query language (SQL)** is the most popular query language used for interacting with a database. SQL allows people to perform complicated searches by using relatively simple statements or key words. Typical key words are SELECT (to choose a desired attribute), FROM (to specify the table or tables to be used), and WHERE (to specify conditions to apply in the query).

To understand how SQL works, imagine that a university wants to know the names of students who will graduate cum laude (but not magna or summa cum laude) in December 2005. (Refer to Figure 5.3 in this chapter.) The university IT staff would query the student relational database with an SQL statement such as:

SELECT Student_Name

FROM Student_Database

WHERE Grade_Point_Average > = 3.40 and Grade_Point_Average < 3.60.

The SQL query would return John Jones and Juan Rodriguez.

Another way to find information in a database is to use *query by example (QBE)*. In QBE, the user fills out a grid or template—also known as a *form*—to construct a sample or a description of the data desired. Users can construct a query quickly and easily by using drag-and-drop features in a DBMS such as Microsoft Access. Conducting queries in this manner is simpler than keying in SQL commands.

Entity–Relationship Modeling

Designers plan and create databases through the process of **entity–relationship modeling** using an **entity–relationship (ER) diagram**. There are many approaches to ER diagramming. You will see one particular approach here. The good news is that if you are familiar with one version of ER diagramming, then you will be able to easily adapt to any other version.

ER diagrams consist of entities, attributes, and relationships. To properly identify entities, attributes, and relationships, database designers first identify the business rules for the particular data model. **Business rules** are precise descriptions of policies, procedures, or principles in any organization that stores and uses data to generate information. Business rules are derived from a description of an organization's operations, and help to create and enforce business processes in that organization. Keep in mind that *you* determine these business rules, not the MIS department.

Entities are pictured in rectangles, and relationships are described on the line between two entities. The attributes for each entity are listed, and the primary key is underlined. The **data dictionary** provides information on each attribute, such as its name; if it is a key, part of a key, or a non-key attribute; the type of data expected (alphanumeric, numeric, dates, etc.); and valid values. Data dictionaries can also provide information on why the attribute is needed in the database; which business functions, applications, forms, and reports use the attribute; and how often the attribute should be updated.

ER modeling is valuable because it allows database designers to communicate with users throughout the organization to ensure that all entities and the relationships among the entities are represented. This process underscores the importance of taking all users into account when designing organizational databases. Notice that all entities and relationships in our example are labeled in terms that users can understand.

Relationships illustrate an association between entities. The *degree of a relationship* indicates the number of entities associated with a relationship. A *unary relationship* exists when an association is maintained within a single entity. A *binary relationship* exists when two entities are associated. A *ternary relationship* exists when three entities are associated. In this chapter, we discuss only binary relationships because they are the most common. Entity relationships may be classified as one-to-one, one-to-many, or many-to-many. The term *connectivity* describes the relationship classification.

Connectivity and cardinality are established by the business rules of a relationship. *Cardinality* refers to the maximum number of times an instance of one entity can be associated with an instance in the related entity. Cardinality can be mandatory single, optional single, mandatory many, or optional many. **Figure 5.9** displays the cardinality symbols. Note that there are four possible cardinality symbols: mandatory single, optional single, mandatory many, and optional many.

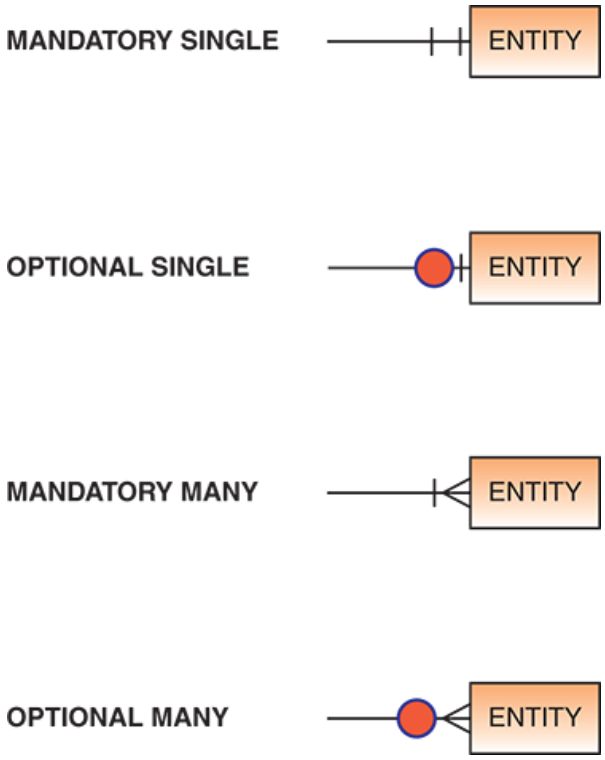


FIGURE 5.9 Cardinality symbols.

Let's look at an example from a university. An *entity* is a person, place, or thing that can be identified in the users' work environment. For example, consider student registration at a university. Students register for courses, and they also register their cars for parking permits. In this example, STUDENT, PARKING PERMIT, CLASS, and PROFESSOR are entities. Recall that an instance of an entity represents a particular student, parking permit, class, or professor. Therefore, a particular STUDENT (James Smythe, 8023445) is an instance of the STUDENT entity; a particular parking permit (91778) is an instance of the PARKING PERMIT entity; a particular class (76890) is an instance of the CLASS entity; and a particular professor (Margaret Wilson, 390567) is an instance of the PROFESSOR entity.

Entity instances have *identifiers*, or *primary keys*, which are attributes (attributes and identifiers are synonymous) that are unique to that entity instance. For example, STUDENT instances can be identified with Student Identification Number, PARKING PERMIT instances can be identified with Permit Number, CLASS instances can be identified with Class Number, and PROFESSOR instances can be identified with Professor Identification Number.

Entities have **attributes**, or properties, that describe the entity's characteristics. In our example, examples of attributes for STUDENT are Student Name and Student Address. Examples of attributes for PARKING PERMIT are Student Identification Number and Car Type. Examples of attributes for CLASS are Class Name, Class Time, and Class Place. Examples of attributes for PROFESSOR are Professor Name and Professor Department. (Note that each course at this university has one professor—no team teaching.)

Why is Student Identification Number an attribute of both the STUDENT and PARKING PERMIT entity classes? That is, why do we need the PARKING PERMIT entity class? If you consider all of the interlinked university systems, the PARKING PERMIT entity class is needed for other applications, such as fee payments, parking tickets, and external links to the state Department of Motor Vehicles.

Let's consider the three types of binary relationships in our example.

In a *one-to-one (1:1)* relationship, a single-entity instance of one type is related to a single-entity instance of another type. In our university example, STUDENT–PARKING PERMIT is a 1:1 relationship. The business rule at this university represented by this relationship is: students may register only one car at this university. Of course, students do not have to register a car at all. That is, a student can have only one parking permit but does not need to have one.

Note that the relationship line on the PARKING PERMIT side shows a cardinality of optional single. A student can have, but does not have to have, a parking permit. On the STUDENT side of the relationship, only one parking permit can be assigned to one student, resulting in a cardinality of mandatory single. See **Figure 5.10**.

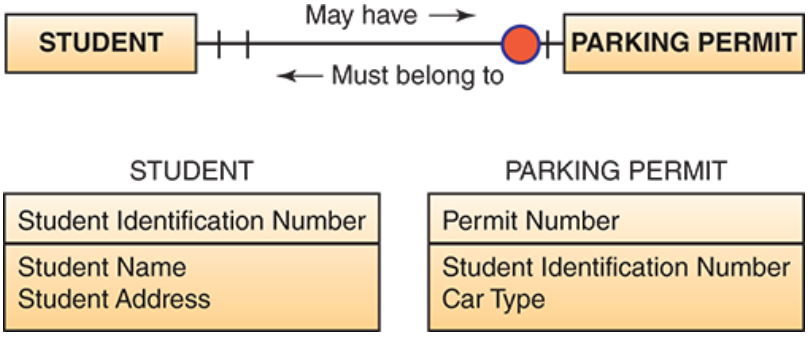


FIGURE 5.10 One-to-one relationship.

The second type of relationship, *one-to-many (1:M)*, is represented by the CLASS–PROFESSOR relationship in **Figure 5.11**. The business rule at this university represented by this relationship is: at this university, there is no team teaching. Therefore, each class must have only

one professor. On the other hand, professors may teach more than one class. Note that the relationship line on the PROFESSOR side shows a cardinality of mandatory single. In contrast, the relationship line on the CLASS side shows a cardinality of optional many.

The third type of relationship, *many-to-many (M:M)*, is represented by the STUDENT–CLASS relationship. Most database management systems do not support many-to-many relationships. Therefore, we use *junction* (or *bridge*) *tables*, so that we have two one-to-many relationships. The business rule at this university represented by this relationship is: students can register for one or more classes, and each class can have one or more students (see **Figure 5.12**). In this example, we create the REGISTRATION table as our junction table. Note that Student ID and Class ID are foreign keys in the REGISTRATION table.

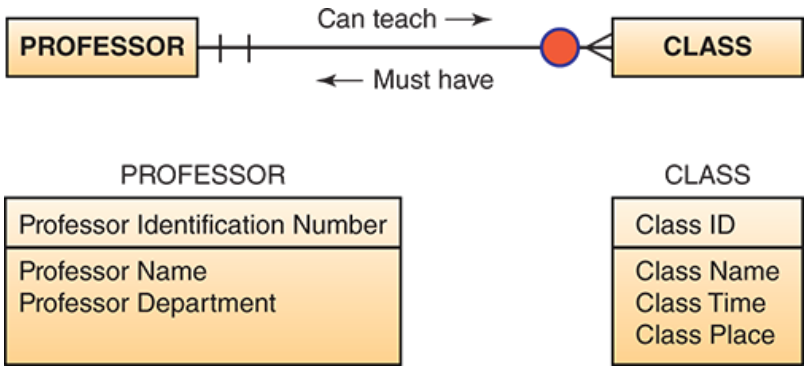


FIGURE 5.11 One-to-many relationship.

Let's examine the following relationships:

- The relationship line on the STUDENT side of the STUDENT–REGISTRATION relationship shows a cardinality of optional single.
- The relationship line on the REGISTRATION side of the STUDENT–REGISTRATION relationship shows a cardinality of optional many.
- The relationship line on the CLASS side of the CLASS–REGISTRATION relationship shows a cardinality of optional single.
- The relationship line on the REGISTRATION side of the CLASS–REGISTRATION relationship shows a cardinality of optional many.

Normalization and Joins

To use a relational database management system efficiently and effectively, the data must be analyzed to eliminate redundant data elements. **Normalization** is a method for analyzing and reducing a relational database to its most streamlined form to ensure minimum redundancy, maximum data integrity, and optimal processing performance. Data normalization is a methodology for organizing attributes into tables so that redundancy among the non-key attributes is eliminated. The result of the data normalization process is a properly structured relational database.

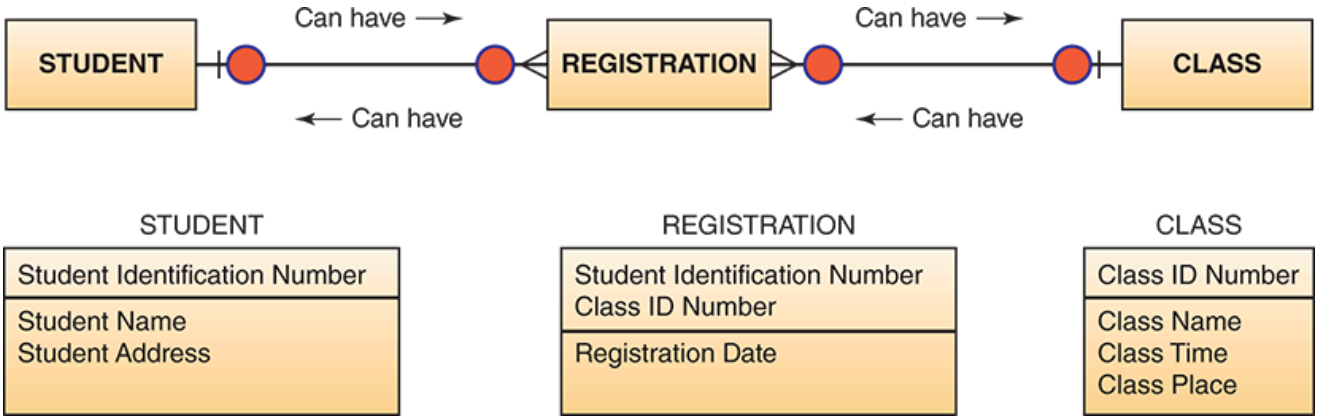


FIGURE 5.12 Many-to-many relationship.

Order Number	Order Date	Customer ID	Customer F Name	Customer L Name	Customer Address	Zip Code	Pizza Code	Pizza Name	Quantity	Price	Total Price
1116	9/1/14	16421	Rob	Penny	123 Main St.	37411	P	Pepperoni	1	$11.00	$41.00
							MF	Meat Feast	1	$12.00	
							V	Vegetarian	2	$9.00	
1117	9/2/14	17221	Beth	Jones	41 Oak St.	29416	HM	Ham and Mushroom	3	$10.00	$56.00
							MF	Meat Feast	1	$12.00	
							TH	The Hawaiian	1	$14.00	

FIGURE 5.13 Raw data gathered from orders at the pizza shop.

Data normalization requires a list of all the attributes that must be incorporated into the database and a list of all of the defining associations, or functional dependencies, among the attributes. **Functional dependencies** are a means of expressing that the value of one particular attribute is associated with a specific single value of another attribute. For example, for Student Number 05345 at a university, exactly one Student Name, John C. Jones, is associated with it. That is, Student Number is referred to as the determinant because its value *determines* the value of the other attribute. We can also say that Student Name is functionally dependent on Student Number.

As an example of normalization, consider a pizza shop. This shop takes orders from customers on a form. Figure 5.13 shows a table of nonnormalized data gathered by the pizza shop. This table has two records, one for each order being placed. Because there are several pizzas on each order, the order number and customer information appear in multiple rows. Several attributes of each record have null values. A null value is an attribute with no data in it. For example, Order Number has four null values. Therefore, this table is not in first normal form. The data drawn from that form is shown in **Figure 5.13**.

In our example, ORDER, CUSTOMER, and PIZZA are entities. The first step in normalization is to determine the functional dependencies among the attributes. The functional dependencies in our example are shown in **Figure 5.14**.

In the normalization process, we will proceed from nonnormalized data, to first normal form, to second normal form, and then to third normal form. (There are additional normal forms, but they are beyond the scope of this book.)

Figure 5.15 demonstrates the data in *first normal form.* The attributes under consideration are listed in one table and primary keys have been established. Our primary keys are Order Number, Customer ID, and Pizza Code. In first normal form, each ORDER has to repeat the order number, order date, customer first name, customer last name, customer address, and customer zip code. This data file contains repeating groups and describes multiple entities. That is, this relation has data redundancy, a lack of data integrity, and the flat file would be difficult to use in various applications that the pizza shop might need.

Consider the table in Figure 5.15 and notice the very first column (labeled Order Number). This column contains multiple entries for each order—three rows for Order Number 1116 and three rows for Order Number 1117. These multiple rows for an order are called *repeating groups.* The table in Figure 5.15 also contains multiple entities: ORDER, CUSTOMER, and PIZZA. Therefore, we move on to second normal form.

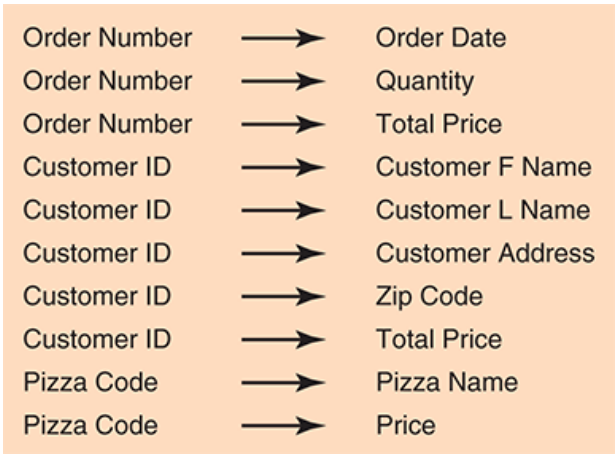


FIGURE 5.14 Functional dependencies in pizza shop example.

To produce second normal form, we break the table in **Figure 5.15** into smaller tables to eliminate some of its data redundancy. Second normal form does not allow partial functional dependencies. That is, in a table in second normal form, every non-key attribute must be functionally dependent on the entire primary key of that table. **Figure 5.16** shows the data from the pizza shop in second normal form.

Order Number	Order Date	Customer ID	Customer F Name	Customer L Name	Customer Address	Zip Code	Pizza Code	Pizza Name	Quantity	Price	Total Price
1116	9/1/14	16421	Rob	Penny	123 Main St.	37411	P	Pepperoni	1	$11.00	$41.00
1116	9/1/14	16421	Rob	Penny	123 Main St.	37411	MF	Meat Feast	1	$12.00	$41.00
1116	9/1/14	16421	Rob	Penny	123 Main St.	37411	V	Vegetarian	2	$9.00	$41.00
1117	9/2/14	17221	Beth	Jones	41 Oak St.	29416	HM	Ham and Mushroom	3	$10.00	$56.00
1117	9/2/14	17221	Beth	Jones	41 Oak St.	29416	MF	Meat Feast	1	$12.00	$56.00
1117	9/2/14	17221	Beth	Jones	41 Oak St.	29416	TH	The Hawaiian	1	$14.00	$56.00

FIGURE 5.15 First normal form for data from pizza shop.

If you examine Figure 5.16, you will see that second normal form has not eliminated all the data redundancy. For example, each Order Number is duplicated three times, as are all customer data. In *third normal form*, non-key attributes are not allowed to define other non-key attributes. That is, third normal form does not allow transitive dependencies in which one non-key attribute is functionally dependent on another. In our example, customer information depends both on Customer ID and Order Number. **Figure 5.17** shows the data from the pizza shop in third normal form. Third normal form structure has these important points:

Order Number	Order Date	Customer ID	Customer F Name	Customer L Name	Customer Address	Zip Code	Total Price
1116	9/1/14	16421	Rob	Penny	123 Main St.	37411	$41.00
1116	9/1/14	16421	Rob	Penny	123 Main St.	37411	$41.00
1116	9/1/14	16421	Rob	Penny	123 Main St.	37411	$41.00
1117	9/2/14	17221	Beth	Jones	41 Oak St.	29416	$56.00
1117	9/2/14	17221	Beth	Jones	41 Oak St.	29416	$56.00
1117	9/2/14	17221	Beth	Jones	41 Oak St.	29416	$56.00

Order Number	Pizza Code	Quantity
1116	P	1
1116	MF	1
1116	V	2
1117	HM	3
1117	MF	1
1117	TH	1

Pizza Code	Pizza Name	Price
P	Pepperoni	$11.00
MF	Meat Feast	$12.00
V	Vegetarian	$9.00
HM	Ham and Mushroom	$10.00
TH	The Hawaiian	$14.00

FIGURE 5.16 Second normal form for data from pizza shop.

ORDER

Order Number	Order Date	Customer ID	Total Price
1116	9/1/14	16421	$41.00
1117	9/2/14	17221	$56.00

CUSTOMER

Customer ID	Customer F Name	Customer L Name	Customer Address	Zip Code
16421	Rob	Penny	123 Main St.	37411
17221	Beth	Jones	41 Oak St.	29416

ORDER-PIZZA

Order Number	Pizza Code	Quantity
1116	P	1
1116	MF	1
1116	V	2
1117	HM	3
1117	MF	1
1117	TH	1

PIZZA

Pizza Code	Pizza Name	Price
P	Pepperoni	$11.00
MF	Meat Feast	$12.00
V	Vegetarian	$9.00
HM	Ham and Mushroom	$10.00
TH	The Hawaiian	$14.00

FIGURE 5.17 Third normal form for data from pizza shop.

- It is completely free of data redundancy.
- All foreign keys appear where needed to link related tables.

Let's look at the primary and foreign keys for the tables in third normal form:

- The *ORDER* relation: The primary key is Order Number and the foreign key is Customer ID.

- The *CUSTOMER* relation: The primary key is Customer ID.
- The *PIZZA* relation: The primary key is Pizza Code.
- The *ORDER–PIZZA* relation: The primary key is a composite key, consisting of two foreign keys, Order Number and Pizza Code.

Now consider an order at the pizza shop. The tables in third normal form can produce the order in the following manner by using the join operation (see **Figure 5.18**). The **join operation** combines records from two or more tables in a database to obtain information that is located in different tables. In our example, the join operation combines records from the four normalized tables to produce an ORDER. Here is how the join operation works:

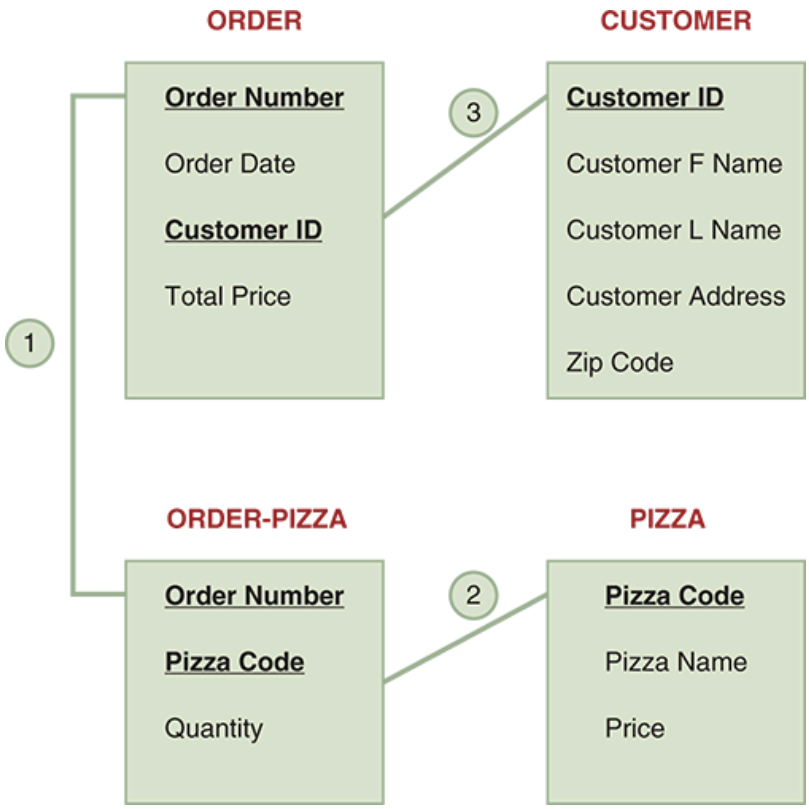


FIGURE 5.18 The join process with the tables of third normal form to produce an order.

- The ORDER relation provides the Order Number (the primary key), Order Date, and Total Price.
- The primary key of the ORDER relation (Order Number) provides a link to the ORDER–PIZZA relation (the link numbered 1 in **Figure 5.18**).
- The ORDER–PIZZA relation supplies the Quantity to ORDER.
- The primary key of the ORDER–PIZZA relation is a composite key that consists of Order Number and Pizza Code. Therefore, the Pizza Code component of the primary key provides a link to the PIZZA relation (the link numbered 2 in Figure 5.18).
- The PIZZA relation supplies the Pizza Name and Price to ORDER.
- The Customer ID in ORDER (a foreign key) provides a link to the CUSTOMER relation (the link numbered 3 in Figure 5.18).
- The CUSTOMER relation supplies the Customer FName, Customer LName, Customer Address, and Zip Code to ORDER.

At the end of this join process, we have a complete ORDER. Normalization is beneficial when maintaining databases over a period of time. One example is the likelihood of having to change the price of each pizza. If the pizza shop increases the price of the Meat Feast from $12.00 to $12.50, this process is one easy step in Figure 5.18. The price field is changed to $12.50 and the ORDER is automatically updated with the current value of the price.

Before you go on…

1. What is structured query language?

2. What is query by example?

3. What is an entity? An attribute? A relationship?

4. Describe one-to-one, one-to-many, and many-to-many relationships.

5. What is the purpose of normalization?

6. Why do we need the join operation?

What's in IT for me?

ACCT For the Accounting Major

The accounting function is intimately concerned with keeping track of an organization's transactions and internal controls. Modern databases enable accountants to perform these functions more effectively. Databases help accountants manage the flood of data in today's organizations so that they can keep their firms in compliance with the standards imposed by Sarbanes–Oxley.

Accountants also play a role in justifying the costs of creating a knowledge base and then auditing its cost-effectiveness. Also, if you work for a large CPA company that provides management services or sells knowledge, then you most likely will use some of your company's best practices, which are stored in a knowledge base.

FIN For the Finance Major

Financial managers make extensive use of computerized databases that are external to the organization, such as CompuStat and Dow Jones, to obtain financial data on organizations in their industry. They can use these data to determine if their organization meets industry benchmarks in return on investment, cash management, and other financial ratios.

Financial managers who produce the organization's financial status reports are also closely involved with Sarbanes–Oxley. Databases help these managers comply with the law's standards.

MKT For the Marketing Major

Databases help marketing managers access data from the organization's marketing transactions, such as customer purchases, to plan targeted marketing campaigns and to evaluate the success of previous campaigns. Knowledge about customers can make the difference between success and failure. In many databases and knowledge bases, the vast majority of information and knowledge concerns customers, products, sales, and marketing. Marketing managers regularly use an organization's knowledge base, and they often participate in creating that base.

POM For the Production/Operations Management Major

Production/operations personnel access organizational data to determine optimal inventory levels for parts in a production process. Past production data enable production/operations management (POM) personnel to determine the optimal configuration for assembly lines. Firms also collect quality data that inform them not only about the quality of finished products but also about quality issues with incoming raw materials, production irregularities, shipping and logistics, and after-sale use and maintenance of the products.

Knowledge management is extremely important for running complex operations. The accumulated knowledge regarding scheduling, logistics, maintenance, and other functions is very valuable. Innovative ideas are critical for improving operations, and they can be supported by knowledge management.

HRM For the Human Resources Management Major

Organizations maintain extensive data on employees including gender, age, race, current and past job descriptions, and performance evaluations. HR personnel access these data to provide reports to government agencies regarding compliance with federal equal opportunity guidelines. HR managers also use these data to evaluate hiring practices and salary structures and to manage any discrimination grievances or lawsuits brought against the firm.

Databases help HR managers provide assistance to all employees as companies turn over more and more decisions about health care and retirement planning to the employees themselves. The employees can use the databases for help in selecting the optimal mix among these critical choices.

HR managers also need to use a knowledge base frequently to find out how past cases were handled. Consistency in how employees are treated not only is important, but it also protects the company against legal actions. In addition, training for building, maintaining, and using the knowledge system is sometimes the responsibility of the HR department. Finally, the HR department might be responsible for compensating employees who contribute their knowledge to the knowledge base.

MIS For the MIS Major

The MIS function manages the organization's data as well as the databases. MIS database administrators standardize data names by using the data dictionary. This process ensures that all users understand which data are in the database. Database personnel also help users access needed data and generate reports with query tools.

What's in IT for me? (Appendix: Section 5.6)

For all Business Majors

All business majors will have to manage data in their professional work. One way to manage data is through the use of databases and database management systems. It is likely that you will need to obtain information from your organization's databases. You will probably use structured query language to obtain this information. Further, as your organization plans and designs its databases, it will most likely use entity–relationship diagrams. You will provide much of the input to these diagrams. For example, you will describe the entities that you use in your work, the attributes of those entities, and the relationships among them. You will also help database designers as they normalize database tables by describing how the normalized tables relate to one another (e.g., through the use of primary and foreign keys). Finally, you will assist database designers as they plan their join operations to provide you with the information that you need when that information is stored in multiple tables.

Summary

5.1 Discuss ways that common challenges in managing data can be addressed using data governance.

The following are three common challenges in managing data:

- Data are scattered throughout organizations and are collected by many individuals using various methods and devices. These data are frequently stored in numerous servers and locations and in different computing systems, databases, formats, and human and computer languages.
- Data come from multiple sources.
- Information systems that support particular business processes impose unique requirements on data, which results in repetition and conflicts across an organization.

One strategy for implementing data governance is master data management. Master data management provides companies with the ability to store, maintain, exchange, and synchronize a consistent, accurate, and timely "single version of the truth" for the company's core master data. Master data management manages data gathered from across an organization, data from multiple sources, and data across business processes within an organization.

5.2 Discuss the advantages and disadvantages of relational databases.

Relational databases enable people to compare information quickly by row or column. Users also can easily retrieve items by finding the point of intersection of a particular row and column. However, large-scale relational databases can be composed of numerous interrelated tables, making the overall design complex, with slow search and access times.

5.3 Define Big Data and its basic characteristics.

Big Data is composed of high-volume, high-velocity, and high-variety information assets that require new forms of processing in order to enhance decision making, lead to insights, and optimize business processes. Big Data has three distinct characteristics that distinguish it from traditional data: volume, velocity, and variety.

- *Volume*: Big Data consists of vast quantities of data.
- *Velocity*: Big Data flows into an organization at incredible speeds.
- *Variety*: Big Data includes diverse data in differing formats.

5.4 Explain the elements necessary to successfully implement and maintain data warehouses.

To successfully implement and maintain a data warehouse, an organization must:

- Link source systems that provide data to the warehouse or mart.
- Prepare the necessary data for the data warehouse using data integration technology and processes.
- Decide on an appropriate architecture for storing data in the data warehouse or data mart.
- Select the tools and applications for the variety of organizational users.
- Establish appropriate metadata, data quality, and governance processes to ensure that the data warehouse or mart meets its purposes.

5.5 Describe the benefits and challenges of implementing knowledge management systems in organizations.

Organizations can realize many benefits with KMSs, including:

- Best practices are readily available to a wide range of employees
- Improved customer service
- More efficient product development
- Improved employee morale and retention

Challenges to implementing KMSs include:

- Employees must be willing to share their personal tacit knowledge.
- Organizations must create a knowledge management culture that rewards employees who add their expertise to the knowledge base.
- The knowledge base must be continually maintained and updated.
- Companies must be willing to invest in the resources needed to carry out these operations.

5.6 Understand the processes of querying a relational database, entity-relationship modeling, and normalization and joins.

The most commonly performed database operation is requesting information. *Structured query language* is the most popular query language used for this operation. SQL allows people to perform complicated searches by using relatively simple statements or key words. Typical key words are SELECT (to specify a desired attribute), FROM (to specify the table to be used), and WHERE (to specify conditions to apply in the query).

Another way to find information in a database is to use *query by example.* In QBE, the user fills out a grid or template—also known as a *form*—to construct a sample or a description of the data desired. Users can construct a query quickly and easily by using drag-and-drop features in a DBMS such as Microsoft Access. Conducting queries in this manner is simpler than keying in SQL commands.

Designers plan and create databases through the process of **entity–relationship modeling**, using an **entity–relationship diagram.** ER diagrams consist of entities, attributes, and relationships. Entities are pictured in boxes, and relationships are represented as lines. The attributes for each entity are listed, and the primary key is underlined.

ER modeling is valuable because it allows database designers to communicate with users throughout the organization to ensure that all entities and the relationships among the entities are represented. This process underscores the importance of taking all users into account when designing organizational databases. Notice that all entities and relationships in our example are labeled in terms that users can understand.

Normalization is a method for analyzing and reducing a relational database to its most streamlined form to ensure minimum redundancy, maximum data integrity, and optimal processing performance. When data are *normalized*, attributes in each table depend only on the primary key.

The *join operation* combines records from two or more tables in a database to produce information that is located in different tables.

Chapter Glossary

attribute Each characteristic or quality of a particular entity.

Big Data A collection of data so large and complex that it is difficult to manage using traditional database management systems.

bit A binary digit—that is, a 0 or a 1.

business rules Precise descriptions of policies, procedures, or principles in any organization that stores and uses data to generate information.

byte A group of eight bits that represents a single character.

database management system (DBMS) The software program (or group of programs) that provide access to a database.

data dictionary A collection of definitions of data elements; data characteristics that use the data elements; and the individuals, business functions, applications, and reports that use these data elements.

data file (also table) A collection of logically related records.

data governance An approach to managing information across an entire organization.

data lake A central repository that stores all of an organization's data, regardless of their source or format.

data mart A low-cost, scaled-down version of a data warehouse that is designed for the end-user needs in a strategic business unit (SBU) or a department.

data model A diagram that represents entities in the database and their relationships.

data silo A collection of data held by one group that is not easily accessible by other groups.

data warehouse A repository of historical data that are organized by subject to support decision makers in the organization.

entity Any person, place, thing, or event of interest to a user.

entity–relationship (ER) diagram Document that shows data entities and attributes and relationships among them.

entity–relationship (ER) modeling The process of designing a database by organizing data entities to be used and identifying the relationships among them.

explicit knowledge The more objective, rational, and technical types of knowledge.

field A characteristic of interest that describes an entity.

foreign key A field (or group of fields) in one table that uniquely identifies a row (or record) of another table.

functional dependency A means of expressing that the value of one particular attribute is associated with, or determines, a specific single value of another attribute.

instance Each row in a relational table, which is a specific, unique representation of the entity.

intellectual capital (or intellectual assets) Other terms for knowledge.

join operation A database operation that combines records from two or more tables in a database.

knowledge management (KM) A process that helps organizations identify, select, organize, disseminate, transfer, and apply information and expertise that are part of the organization's memory and that typically reside within the organization in an unstructured manner.

knowledge management systems (KMSs) Information technologies used to systematize, enhance, and expedite intra- and interfirm knowledge management.

master data A set of core data, such as customer, product, employee, vendor, geographic location, and so on, that spans an enterprise's information systems.

master data management A process that provides companies with the ability to store, maintain, exchange, and synchronize a consistent, accurate, and timely "single version of the truth" for the company's core master data.

multidimensional structure Storage of data in more than two dimensions; a common representation is the data cube.

normalization A method for analyzing and reducing a relational database to its most streamlined form to ensure minimum redundancy, maximum data integrity, and optimal processing performance.

primary key A field (or attribute) of a record that uniquely identifies that record so that it can be retrieved, updated, and sorted.

query by example To obtain information from a relational database, a user fills out a grid or template—also known as a *form*—to construct a sample or a description of the data desired.

record A grouping of logically related fields.

relational database model Data model based on the simple concept of tables in order to capitalize on characteristics of rows and columns of data.

relationships Operators that illustrate an association between two entities.

secondary key A field that has some identifying information, but typically does not uniquely identify a record with complete accuracy.

structured data Highly organized data in fixed fields in a data repository such as a relational database that must be defined in terms of field name and type (e.g., alphanumeric, numeric, and currency).

structured query language The most popular query language for requesting information from a relational database.

table A grouping of logically related records.

tacit knowledge The cumulative store of subjective or experiential learning, which is highly personal and hard to formalize.

transactional data Data generated and captured by operational systems that describe the business's activities, or transactions.

unstructured data Data that do not reside in a traditional relational database.

Discussion Questions

1. Is Big Data really a problem on its own, or are the use, control, and security of the data the true problems? Provide specific examples to support your answer.

2. What are the implications of having incorrect data points in your Big Data? What are the implications of incorrect or duplicated customer data? How valuable are decisions that are based on faulty information derived from incorrect data?

3. Explain the difficulties involved in managing data.

4. What are the problems associated with poor-quality data?

5. What is master data management? What does it have to do with high-quality data?

6. Explain why master data management is so important in companies that have multiple data sources.

7. Describe the advantages and disadvantages of relational databases.

8. Explain why it is important to capture and manage knowledge.

9. Compare and contrast tacit knowledge and explicit knowledge.

10. Draw the entity–relationship diagram for a company that has departments and employees. In this company, a department must have at least one employee, and company employees may work in only one department.

11. Draw the entity–relationship diagram for library patrons and the process of checking out books.

12. You are working at a doctor's office. You gather data on the following entities: PATIENT, PHYSICIAN, PATIENT DIAGNOSIS, and TREATMENT. Develop a table for the entity PATIENT VISIT. Decide on the primary keys and/or foreign keys that you want to use for each entity.

13. Read the article: S. Kliff and M. Sanger-Katz, "Bottleneck for U.S. Coronavirus Response: The Fax Machine," *New York Times*, July 13, 2020. Describe which of the problems in managing data (Section 5.1) are being emphasized by the COVID-19 pandemic.

Problem-Solving Activities

1. Access various employment websites (e.g., **www.monster.com** and **www.dice.com**) and find several job descriptions for a database administrator. Are the job descriptions similar? What are the salaries offered in these positions?

2. Access the websites of several real estate companies. Find the sites that take you through a step-by-step process for buying a home, that provide virtual reality tours of homes in your price range (say, $200,000 to $250,000) and location, that provide mortgage and interest rate calculators, and that offer financing for your home. Do the sites require that you register to access their services? Can you request that an e-mail be sent to you when properties you might be interested in become available? How does the process outlined influence your likelihood of selecting this company for your real estate purchase?

3. It is possible to find many websites that provide demographic information. Access several of these sites and see what they offer. Do the sites differ in the types of demographic information they offer? If so, how? Do the sites require a fee for the information they offer? Would demographic information be useful to you if you wanted to start a new business? If so, how and why?

4. Search the web for uses of Big Data in homeland security. Specifically, read about the spying by the U.S. National Security Agency (NSA). What role did technology and Big Data play in this questionable practice?

5. Search the Web for the article "Why Big Data and Privacy Are Often at Odds." What points does this article present concerning the delicate balance between shared data and customer privacy?

6. Access the websites of IBM (**www.ibm.com**), Microsoft (**www.microsoft.com**), and Oracle (**www.oracle.com**), and trace the capabilities of their latest data management products, including web connections.

7. Access the website for the Gartner Group (**www.gartner.com**). Examine the company's research studies pertaining to data management. Prepare a report on the state of the art.

8. Diagram a knowledge management system cycle for a fictional company that sells customized T-shirts to students.

Closing Case

MIS The Democratic Party Upgrades Its Data Repository and Its Data

Until 2011, the Democratic Party (**www.democrats.org**) stored data in multiple databases, making it difficult for campaigns to integrate the data to form a holistic profile of voters and issues important to those voters. In 2011, the party deployed Vertica (**www.vertica.com**), an analytics database, as its central data repository. Vertica enabled the party to store every state's voter file, every door knock and phone call that organizers made, and all commercially available data that campaigns collected. Using Vertica, the successful 2012 Obama reelection campaign was able to analyze the data to target potential voters with outreach and advertising at an individual level instead of placing them into broad categories such as urban voters or soccer moms.

After 2012, a lack of maintenance caused problems with Vertica. As a result, the party had to devote an increasing amount of resources to address these problems. A key problem with the system involved data. After 2012, the party began to collect rapidly increasing amounts of data, which Vertica was unable to manage. The data were poorly labeled, inconsistent, incorrect, and contained missing values such as voter phone numbers and addresses.

To compound this problem, Vertica's interface was not intuitive. It was difficult for Democratic Party personnel with limited experience in data analytics to use. As a result, state and local campaigns did not derive much value from the system. One analyst stated that you had to know or have participated in a prior campaign to understand where the really good data were and how to effectively access and use them. Such poor-quality data had consequences. For example, analysts noted that party operatives had a habit of knocking on dead people's doors.

Vertica also predated cloud-based systems. Therefore, the party had to deploy servers that could not manage the terabytes of data flowing into them. Further, the servers could not manage the thousands of data analysts trying to access data in the final days before an election.

As a result, in the months before the 2016 election, presidential candidate Hillary Clinton's team struggled with the system. It often crashed for 16 hours at a time during the campaign. In fact, the campaign had dozens of computer engineers on call 24 hours per day, ready to restart the system each time it went down.

Having witnessed the critical role that poor data and an unreliable database had played in their unsuccessful 2016 presidential election, the Democrats realized that they had to improve their data infrastructure. To accomplish this task, the party hired a new chief technology officer, who divided his 40-person IT staff into two teams. One team would maintain Vertica just long enough to get through Election Day 2018. The other team would build a new system to replace Vertica.

Up through the 2018 midterm elections, one team of party engineers continued to provide constant maintenance to keep Vertica operational. Even with their efforts, Vertica crashed for 10 hours one night just prior to voting.

A major goal for the other team was to develop a more stable platform that did not require the party to maintain its own servers. The party raised $5 million from donors explicitly for this project.

Accordingly, the team developed a new data repository called the Data Warehouse. The new system uses Google's analytics tool, BigQuery, a cloud-based platform capable of handling massive data sets at the scale and speed necessary for modern campaigns. Further, the Data Warehouse is more reliable and more intuitive for smaller campaigns, whose operatives generally do not have experience in data analytics.

Creating a data exchange. The Federal Election Commission (FEC) prohibits coordination between campaigns and outside groups. This ruling has traditionally prevented the candidate's campaign and its super PAC (political action committee) from comparing or intermingling the data collected by each entity. A *super PAC* is a type of independent committee that raises unlimited sums of money from corporations, unions, and/or individuals but is not permitted to contribute to or coordinate directly with parties or candidates.

In 2011, the Republicans found a way around that rule by creating a third-party organization called the Data Trust. This company is outside the Republican Party and acts as a data repository. Multiple Republican groups license their data to the trust, which allows other groups such as PACs to pay for access to it without violating FEC regulations. Democrats, quite correctly, viewed the Data Trust as a competitive advantage for the Republican Party.

The Democrat's Data Warehouse helped the party achieve one of its primary goals before the 2020 elections: the creation of its data exchange. The exchange allows the party and other political groups to share their data for the first time, without violating campaign finance laws. The Democrat's exchange is modeled on the Republican Party's GOP Data Trust.

Somewhat surprisingly, Democratic officials who manage their states' voter files were initially reluctant to give up control of their data. The party had to come up with a compromise. The Democratic National Committee (DNC) would house the data. The data exchange would track only who provides the data, the data they provide, who accesses the data, and the data they access.

Democratic operatives also had concerns about whether the Data Warehouse would be accessible enough to campaign staffers who do not have any SQL coding skills, because the system requires some programming skills. Therefore, it was critical for the Data Warehouse team to build tools that enabled the average field staffer to easily access and analyze the data stored in the Data Warehouse. In fact, the Democratic Party has developed a number of tools based on the Data Warehouse.

The Blueprint tool. One of these tools, called Blueprint, helps campaigns and state parties, particularly those with limited technical expertise, to access voter data to better target their campaign efforts. Blueprint increased the Democratic Party's digital capabilities at a time when campaigning in person is extremely difficult, if not impossible, due to the COVID-19 pandemic. It helps campaigns decide whom to call, text, e-mail, and target with digital advertisements.

Rather than having to search through the party's voter database themselves, Blueprint enables campaign workers at every level to access data such as voters' addresses, ethnicity, and voting history in a specific area. Blueprint is especially valuable for down-ballot candidates, who often do not have in-house technical teams focused on analyzing data to mobilize prospective voters and volunteers. The term *down-ballot* refers to a candidate who is relatively low-profile and local compared to a more prominent candidate whose name appears higher on the ballot.

The party piloted Blueprint in several states, including Texas. Using this tool, in March 2020 the Texas Democratic Party introduced a new model that scores every Texas voter from 1 to 100 according to how likely they are to vote for a Democratic candidate. This model helps the state party more efficiently identify and target undecided voters. Texas used Blueprint's data to make its scoring system more effective. The Data Warehouse and Blueprint allowed them to access demographic and consumer data that they did not already have. These data included voters' ethnicity, neighborhood, and income.

The voter registration tool. In June 2020 the party deployed another new tool to help Democratic campaign workers and state parties contact voters who, unknowingly, had been either purged from active voter rolls or designated as inactive voters. The party hopes that the tool will be valuable in states that removed voters who either had not voted recently or had not responded to mailings from the state. (States justify these policies by claiming that they protect against fraudulent voting.) The tool allows campaigns and state parties to recognize these voters and collect their names, phone numbers, and addresses. The campaigns can then target them with calls, text messages, and mail.

The new voter file. The COVID-19 pandemic severely limited traditional door-to-door canvassing. As a result, campaigns need to increase the efficiency of virtual canvassing. In July 2020 the Democratic Party deployed a new voter file model, designed to predict the likelihood that a person will (a) have a working cell phone number and (b) respond positively to a text message. The party wants to make its voter outreach efforts more productive by giving volunteers accurate cell phone numbers and preventing them from texting wrong or disconnected numbers.

Sources: Compiled from J. Turman, "DNC Hopes to Reach More Voters with New Voter File Model," *CBS News*, July 10, 2020; D. Merica, "Democrats Roll out New Tool to Combat Voter Purges," *CNN*, June 16, 2020; E. Birnbaum and I. Lapowsky, "New DNC Data Tool Aims to Give and Edge to Campaigns Light on Tech Expertise," *Protocol*, April 9, 2020; I. Lapowsky, "'We've Had People Panicking:' Tech Startups Scramble to Take the 2020 Race Digital," *Protocol*, April 1, 2020; M. Nickelsburg, "DNC's New Tech Leader Talks about What Went Wrong in 2016 and How Dems Are Preparing for 2020," *GeekWire*, July 29, 2019; B. Mitchell, "The Revolution Will Be Online: How Democrats Are Trying to Catch Up to Trump," *CNET*, June 3, 2019; J. Easley, "Inside the DNC's Plan to Defeat Trump," *The Hill*, May 31, 2019; R. Cramer, "Want the Voter File? Campaigns Will Have to Pay, Record Videos and Fundraise for the DNC to Get It," *BuzzFeed*, May 4, 2019; I. Lapowsky, "Inside the Democrats' Plan to Fix Their Crumbling Data Operation," *Wired*, April 2, 2019; "DNC Rolls out New Data Warehouse," **Democrats.org**, April 2, 2019; B. Barrow, "Howard Dean to Head New Democratic Voter Data Exchange," *Associated Press*, February 13, 2019; I. Lapowsky, "Democrats Uber-ized Activism. Can It Win Them the Midterms?" *Wired*, November 6, 2018; **www.democrats.org**, accessed July 13, 2020.

Questions

1. Are the data contained in the Data Warehouse Big Data? Provide specific examples to support your answer.
2. Are the data contained in the Data Warehouse structured? Provide specific examples to support your answer.
3. Describe another application that the Democratic Party could develop for the Data Warehouse.