Loshin, D. 2001. "The Cost of Poor Data Quality." *DM Review* (June 29) available at **www.information-management.com/infodirect/20010629/3605-1.html**.

Loshin, D. 2006. "Monitoring Data Quality Performance Using Data Quality Metrics." A white paper from Informatica (November).

Loshin, D. 2009. "The Data Quality Business Case: Projecting Return on Investment." available at **http://knowledge-integrity.com/Assets/data_quality_business_case.pdf**.

Moriarty, T. 1996. "Better Business Practices." *Database Programming & Design* 9,7 (September): 59–61.

Redman, T. 2004. "Data: An Unfolding Quality Disaster." *DM Review* 14,8 (August): 21–23, 57.

Russom, P. 2006. "Taking Data Quality to the Enterprise through Data Governance." *TDWI Report Series* (March).

Seiner, R. 2005. "Data Steward Roles & Responsibilities," available at **www.tdan.com**, July 2005.

Variar, G. 2002. "The Origin of Data." *Intelligent Enterprise* 5,2 (February 1): 37–41.

Westerman, P. 2001. *Data Warehousing: Using the Wal-Mart Model*. San Francisco: Morgan Kaufmann.

White, C. 2000. "First Analysis." *Intelligent Enterprise* 3,9 (June): 50–55.

Yugay, I., and V. Klimchenko. 2004. "SOX Mandates Focus on Data Quality & Integration." *DM Review* 14,2 (February): 38–42.

Further Reading

Eckerson, W. 2002. "Data Quality and the Bottom Line: Achieving Business Success Through a Commitment to Data Quality." **www.tdwi.org**.

Weill, P., and J. Ross. 2004. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Boston: Harvard Business School Press.

Web Resources

www.knowledge-integrity.com Web site of David Loshin, a leading consultant in the data quality and business intelligence fields.

http://mitiq.mit.edu Web site for data quality research done at Massachusetts Institute of Technology.

www.tdwi.org Web site of The Data Warehousing Institute, which produces a variety of white papers, research reports, and Webinars that are available to the general public, as well as a wider array that are available only to members.

www.teradatauniversitynetwork.com The Teradata University Network, a free portal service to a wide variety of journal articles, training materials, Webinars, and other special reports on data quality, data integration, and related topics.

CHAPTER 11

Big Data and Analytics

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Concisely define each of the following key terms: **big data, analytics, data lake, NoSQL, MapReduce, Hadoop, HDFS, Pig, Hive, business intelligence, descriptive analytics, predictive analytics, prescriptive analytics, online analytical processing, relational OLAP (ROLAP), multidimensional OLAP (MOLAP), data mining, and text mining.**
- Describe the reasons why data management technologies and approaches have expanded beyond relational databases and data warehousing technologies.
- List the main categories of NoSQL database management systems.
- Choose between relational databases and various types of NoSQL databases depending on the organization's data management needs.
- Describe the meaning of big data and the demands big data will place on data management technology.
- List the key technology components of a typical Hadoop environment and describe their uses.
- Articulate the differences between descriptive, predictive, and prescriptive analytics.
- Describe the impact of advances in analytics on data management technologies and practices.

INTRODUCTION

There are few terms in the context of data management that have seen such an explosive growth in interest and commercial hype as "big data," a term that is still elusive and ill-defined but at the same time widely used and applied in practice by businesses, scientists, government agencies, and not-for-profit organizations. **Big data** are data that exist in very large volumes and many different varieties (data types) and that need to be processed at a very high velocity (speed). Not surprisingly, big data analytics refers to analytics that deals with big data. We will discuss the big data concept at a more detailed level later in this chapter (including the introduction of more terms starting with a "*v*," in addition to volume, variety, and velocity), pointing out that the concept of big data is constantly changing depending on the state of the art in technology. Big data is not a single, separate phenomenon but an umbrella term for a subset of advances in a field that emerged much earlier—analytics (also called data analytics or, in business contexts, business analytics). At its most fundamental level, **analytics** refers to systematic analysis and interpretation of data—typically using mathematical, statistical, and computational tools—to improve our understanding of a real-world domain.

Big data
Data that exist in very large volumes and many different varieties (data types) and that need to be processed at a very high velocity (speed).

Analytics
Systematic analysis and interpretation of data—typically using mathematical, statistical, and computational tools—to improve our understanding of a real-world domain.

What makes big data and analytics so important that an entire chapter is justified? Consider the following story (adapted from Laskowski, 2014; this source describes Gartner's Doug Laney's 55 big data success stories):

> One of the top customers of Morton's Steakhouse was on Twitter lamenting a late flight that prevented him from dining at Morton's. The company used the opportunity to create a publicity stunt and surprised the customer with a meal delivered to him prepared exactly the way he typically wanted to have it. This was possible only with sophisticated social media monitoring, detailed customer data, and the ability to bring all of this together and act on it in real time.

The technologies discussed in this chapter help organizations implement solutions that are based on real-time analysis of very large and heterogeneous data sets from a variety of sources. For example, Telefónica UK O2—the number 2 wireless communications provider in the UK—brings together network performance data and customer survey data in order to understand better and faster how to allocate its network upgrade resources in a way that provides the highest value for the company and its customers (TCSET, 2014). None of this would have been possible without big data and analytics.

For a long period of time, the most critical issue of data management was to ensure that an organization's transaction processing systems worked reliably at a reasonable cost. As discussed earlier in this book, well-designed and carefully implemented relational databases allow us to achieve those goals even in very high-volume environments (such as Web-based e-commerce systems). Chapter 9 discussed the second major step in data management—the use of data warehouses that are separate from the transactional databases for two purposes: first, to enable analytics to describe how the organization has performed in the past and second, to make possible modeling of the future based on what we have learned about the history. Data are structured in a different way for data warehousing. Particularly for large companies, the warehouses are implemented with different technical solutions (often using appliances specifically designed to work as data warehouses).

Technologies related to big data have brought us to the third era of data management. These technologies have stringent requirements: They have to (1) process much larger quantities of data than either operational databases or data warehouses do (thus requiring, for example, a high level of scalability using affordable hardware), (2) deal effectively with a broad variety of different data types (and not only textual or numeric data), and (3) adapt much better to changes in the structure of data, and thus not require a strictly predefined schema (data model) as relational databases do. These requirements are addressed with two broad families of technologies: a core big data technology called Hadoop (and its alternatives/competitors) and database management technologies under the umbrella NoSQL (these days typically interpreted as "Not only SQL" instead of "No SQL"). We will discuss both at a more detailed level later in this chapter.

This chapter starts with a brief general overview of big data as a combination of technologies and processes that make it possible for organizations to convert very large amounts of raw data into information and insights for use in business, science, health care, law, and dozens of other fields of practice. We also place the concept of big data in the broader context of analytics. The discussion continues with a central element of this chapter: a section on the data management infrastructure that is required for modern analytics in addition to the technologies that we have covered in earlier chapters. We pay particular attention to alternatives to traditional relational DBMS technologies grouped under the title NoSQL and the technologies that are currently used to implement big data solutions, such as Hadoop. We next discuss typical uses of descriptive, predictive, and prescriptive analytics and provide an in-depth review of data infrastructure technologies for analytics. The chapter ends with a section on the uses and implications of big data analytics.

BIG DATA

Big data has been one of the most frequently covered concepts in the popular business press during the last few years. Even though it is clear that some of the enthusiasm related to big data is overheated, it is equally evident that big data represents something new, interesting, and quite promising. The most common ways to explain what big data is have approached the question from three perspectives labeled with names starting with a "v," including volume, variety, and velocity; these dimensions were originally presented in Laney (2001). As previously described, the concept of big data refers to a lot of data (high volume) that exists in many different forms (high variety) and arrives/is collected fast (at a high velocity or speed). This gives us a vague definition that is changing all the time: When technology develops, today's high volume, variety, and velocity will be perfectly ordinary tomorrow, and we will probably have new capabilities for processing data that will lead to new types of highly beneficial outcomes. Thus, it is likely to be futile to seek out a specific and detailed definition of big data.

It is still worth our time to discuss briefly the original three Vs and two others that some observers later added. These are summarized in Table 11-1.

- At a time when terabyte-size databases are relatively typical even in small and middle-sized organizations, the *Volume* dimension of big data refers to collections of data that are hundreds of terabytes or more (petabytes) in size. Large data centers can currently store exabytes of data.
- *Variety* refers to the explosion in the types of data that are collected, stored, and analyzed. As we will discuss in the context of the three eras of business intelligence and analytics later in this chapter, the traditional numeric administrative data are now only a small subset of the data that organizations want to maintain. For example, Hortonworks (2014) identifies the following types of data as typical in big data systems: sensor, server logs, text, social, geographic, machine, and clickstream. Missing from this list are still audio and video data.
- *Velocity* refers to the speed at which the data arrives—big data analytics deals with not only large total amounts of data but also data arriving in streams that are very fast, such as sensor data from large numbers of mobile devices, and clickstream data.
- *Veracity* is a dimension of big data that is both a desired characteristic and a challenge that has to be dealt with. Because of the richness of data types and sources of data, traditional mechanisms for ensuring data quality (discussed in detail in Chapter 10) do not necessarily apply; there are sources of data quality problems that simply do not exist with traditional structured data. At the same time, there is nothing inherent in big data that would make it easier to deal with data quality problems; therefore, it is essential that these issues are addressed carefully.
- *Value* is an essential dimension of big data applications and the use of big data to support organizational actions and decisions. Large quantities, high arrival speeds, and a wide variety of types of data together do not guarantee that data genuinely provides value for the enterprise. A large number of contemporary business books have made the case for the potential value that big data technologies bring to a modern enterprise. These include *Analytics at Work* (Davenport, Harris, and Morison, 2010), *Big Data at Work* (Davenport, 2014), and *Taming the Big Data Tidal Wave* (Franks, 2012).

TABLE 11-1 Five Vs of Big Data

Volume	In a big data environment, the amounts of data collected and processed are much larger than those stored in typical relational databases.
Variety	Big data consists of a rich variety of data types.
Velocity	Big data arrives to the organization at high speeds and from multiple sources simultaneously.
Veracity	Data quality issues are particularly challenging in a big data context.
Value	Ultimately, big data is meaningless if it does not provide value toward some meaningful goal.

Another important difference between traditional structured databases and data stored in big data systems is that—as we learned in Chapters 2 to 8—creating high-quality structured databases requires that these databases be based on carefully developed data models (both conceptual and logical) or schemas. This approach is often called *schema on write*—the data model is predefined and changing it later is difficult. The same approach is required for traditional data warehouses. The philosophy of big data systems is different and can be described as *schema on read*—the reporting and analysis organization of the data will be determined at the time of the use of the data. Instead of carefully planning in advance what data will be collected and how the collected data items are related to each other, the big data approach focuses on the collection and storage of data in large quantities even though there might not be a clear idea of how the collected data will be used in the future. The structure of the data might not be fully (or at all) specified, particularly in terms of the relationships between the data items. Technically, the "schema on read" approach is typically based on the use of either JavaScript Object Notation (JSON) or Extensible Markup Language (XML). Both of these specify the structure of each collection of attribute values at the record level (see Figure 11-1 for an example), and thus make it possible to analyze complex and varying record structures at the time of the use of the data. "Schema on read" refers to the fact that there is no predefined schema for the collected data but that the necessary models will be developed when the data are read for utilization.

Data lake
A large integrated repository for internal and external data that does not follow a predefined schema.

An integrated repository of data with various types of structural characteristics coming from internal and external sources (Gualtieri and Yuhanna, 2014, p. 3) is called a **data lake**. A white paper by The Data Warehousing Institute calls a data lake a "dumping ground for all kinds of data because it is inexpensive and does not require a schema on write" (Halper, 2014, p. 2). Hortonworks (2014, p. 13) specifies three characteristics of a data lake:

- *Collect everything*. A data lake includes all collected raw data over a long period of time and any results of processing of data.
- *Dive in anywhere*. Only limited by constraints related to confidentiality and security, data in a data lake can be accessed by a wide variety of organizational actors for a rich set of perspectives.
- *Flexible access*. "Schema on read" allows an adaptive and agile creation of connections among data items.

There are, however, many reasons why the big data approach is not suitable for all data management purposes and why it is not likely to replace the "schema on write" approach universally. As you will learn soon, the most common big data technologies

FIGURE 11-1 Examples of JSON and XML

JSON Example

```
{"products": [
        {"number": 1, "name": "Zoom X", "Price": 10.00},
        {"number": 2, "name": "Wheel Z", "Price": 7.50},
        {"number": 3, "name": "Spring 10", "Price": 12.75}
]}
```

XML Example

```
<products>
        <product>
                <number>1</number> <name>Zoom X</name> <price>10.00</price>
        </product>
        <product>
                <number>2</number> <name>Wheel Z</name> <price>7.50</price>
        </product>
        <product>
                <number>3</number> <name>Spring 10</name> <price>12.75</price>
        </product>
</products>
```


(those based on Hadoop) are based on batch processing—designing an analytical task and the approach for solving it, submitting the job to execute the task to the system, and waiting for the results while the system is processing it. With very large amounts of data the execution of the tasks may last quite a long time (potentially hours). The big data approach is not intended for exploring individual cases or their dependencies when addressing an individual business problem; instead, it is targeted to situations with very large amounts of data, with a variety of data, and very fast streams of data. For other types of data and information needs, relational databases and traditional data warehouses offer well-tested capabilities.

Next, we will discuss two specific categories of technologies that have become known as core infrastructure elements of big data solutions: NoSQL and Hadoop. The first is NoSQL (abbreviated from "Not only SQL"), a category of data storage and retrieval technologies that are not based on the relational model. The second is Hadoop, an open source technology specifically designed for managing large quantities, varieties, and fast streams of data.

NoSQL

NoSQL
A category of recently introduced data storage and retrieval technologies that are not based on the relational model.

NoSQL (abbreviated from "Not only SQL") is a category of recently introduced data storage and retrieval technologies that are not based on the relational model. We will first discuss the general characteristics of these technologies and then analyze them at a more detailed level using a widely used categorization into key-value stores, document stores, wide-column stores, and graph databases.

The need to minimize storage space used to be one of the key reasons underlying the strong focus on avoidance of replication in relational database design. Economics of storage have, however, changed because of a rapid reduction in storage costs, thus minimizing storage space is no longer a key design consideration. Instead, the focus has moved to scalability, flexibility, agility, and versatility. For many purposes, particularly in transaction processing and management reporting, the predictability and stability of databases based on the relational model continue to be highly favorable characteristics. For other purposes, such as complex analytics, other design dimensions are more important. This has led to the emergence of database models that provide an alternative to the relational model. These models, often discussed under the umbrella term of NoSQL, are particularly interesting in contexts that require versatile processing of a rich variety of data types and structures.

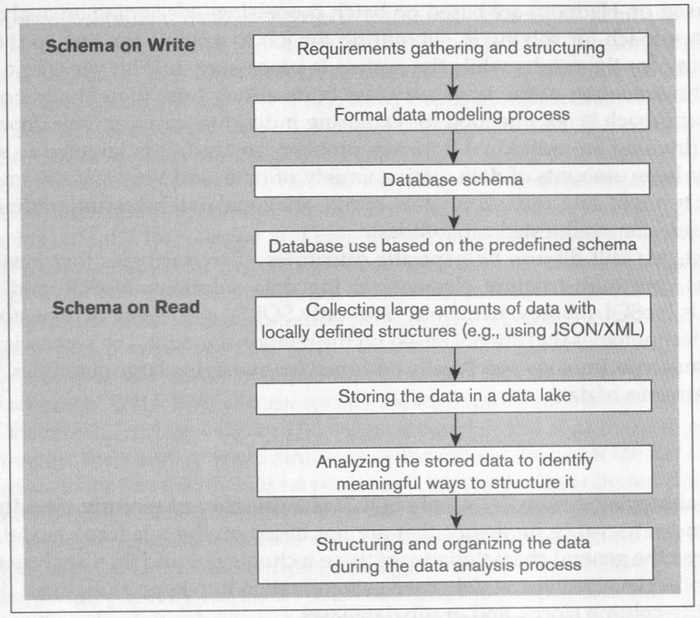
NoSQL database management systems allow "scaling out" through the use of a large number of commodity servers that can be easily added to the architectural solution instead of "scaling up," an older model used in the context of the relational model that in many cases required large stepwise investments in larger and larger hardware. The NoSQL systems are designed so that the failure of a single component will not lead to the failure of the entire system. This model can be easily implemented in a cloud environment in which the commodity servers (real or virtual) are located in a service provider's data center environment accessible through the public Internet. Many NoSQL systems enable automated sharding, that is, distributing the data among multiple nodes in a way that allows each server to operate independently on the data located on it. This makes it possible to implement a shared-nothing architecture, a replication architecture that does not have separate master/slave roles.

NoSQL systems also provide opportunities for the use of the "schema on read" model instead of the "schema on write" model that assumes and requires a predefined schema that is difficult to change. As previously discussed and illustrated in Figure 11-2, "schema on read" is built on the idea that every individual collection of individual data items (record) is specified separately using a language such as JSON or XML.

Interestingly, many NoSQL database management systems are based on technologies that have emerged from open source communities; for enterprise use, they are offered with commercial support.

It is important to understand that most NoSQL database management systems do not support ACID (atomicity, consistency, isolation, and durability) properties of transactions, typically considered essential for guaranteeing the consistency of administrative systems and discussed in Chapter 12 (available on the book's Web

FIGURE 11-2 Schema on write vs. schema on read



site). NoSQL database management systems are often used for purposes in which it is acceptable to sacrifice guaranteed consistency for ensure constant availability. Instead of the ACID properties, NoSQL systems are said to have BASE properties: basically available, soft state, and eventually consistent. Eric Brewer's (2000) CAP theorem states that no system can achieve consistency, high availability, and partition tolerance at the same time in case errors occur; in practice, this means that distributed systems cannot achieve high availability and guaranteed consistency at the same time. NoSQL database management systems are choosing high availability over guaranteed consistency whereas relational databases with ACID propertied are offering guaranteed consistency while sacrificing availability in certain situations (Voroshilin, 2012).

Classification of NoSQL Database Management Systems

There are four main types of NoSQL database data models (McKnight, 2014): key-value stores, document stores, wide-column stores, and graph databases.

KEY-VALUE STORES Key-value stores (illustrated in Figure 11-3a) consist of a simple pair of a key and an associated collection of values. A key-value store database maintains a structure that allows it to store and access "values" (number, name, and price in our example) based on a "key" (with value "Prod_1" in our example). The "key" is typically a string, with or without specific meaning, and in many ways it is similar to a primary key in a relational table. The database does not care or even know about the contents of the individual "value" collections; if some part of the "value" needs to be changed, the entire collection will need to be updated. For the database, the "value" is an arbitrary collection of bytes, and any processing of the contents of the "value" is left for the application. The only operations a typical key-value store offers are *put* (for storing the "value"), *get* (for retrieving the "value" based on the "key"), and *delete* (for deleting a specific key-value pair). As you see, no update operation exists.

DOCUMENT STORES Document stores (illustrated in Figure 11-3b) do not deal with "documents" in a typical sense of the word; they are not intended for storing, say, word-processing or spreadsheet documents. Instead, a document in this context is a structured set of data formatted using a standard such as JSON. The key difference between key-value stores and document stores is that a document store has the capability of accessing and modifying the contents of a specific document based on its structure; each

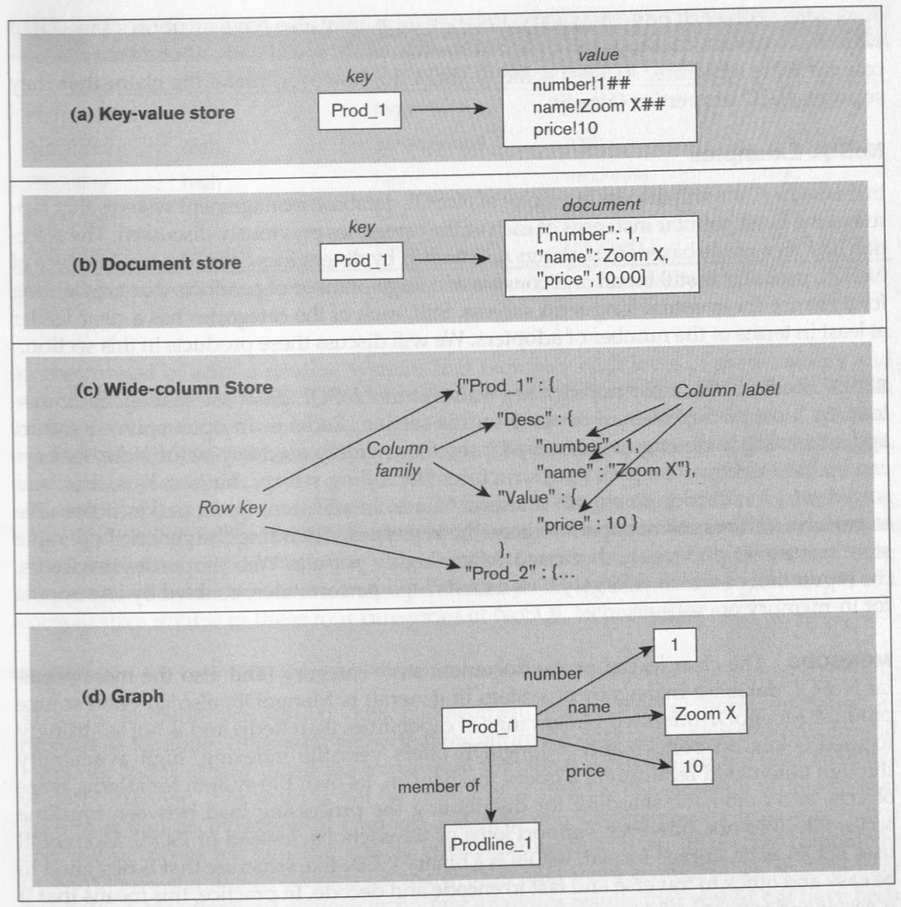


FIGURE 11-3 Four-part figure illustrating NoSQL databases

Some of the example structures have been adapted from Kauhanen (2010).

"document" is still accessed based on a "key." In addition to this, the internal structure of the "document" (specified within it using JSON) can be used to access and manipulate its contents. In our example, the key "Prod_1" is used to access the document consisting of components "number," "name," and "price." Each of these can be manipulated separately in a document store context. The "documents" may have a hierarchical structure, and they do not typically reference each other.

WIDE-COLUMN STORES Wide-column stores or extensible record stores (illustrated in Figure 11-3c) consist of rows and columns, and their characteristic feature is the distribution of data based on both key values (records) and columns, using "column groups" or "column families" to indicate which columns are best to be stored together. They allow each row to have a different column structure (there are no constraints defined by a shared schema), and the length of the rows varies. Edjladi and Beyer (2013) suggest that wide-column stores are particularly good for storing semi-structured data in a distributed environment.

GRAPH-ORIENTED DATABASES Graph-oriented databases (illustrated in Figure 11-3d) have been specifically designed for purposes in which it is critically important to be able to maintain information regarding the relationships between data items (which, in many cases, represent real-world instances of entities). Data in a graph-oriented database is stored in nodes with properties (named attribute values), and the connections between the nodes represent relationships between the real-world instances. As with other forms of NoSQL database management systems, the collections of attributes

associated with each node may vary. Relationships may also have attributes associated with them. Conceptually, graph-oriented databases are specifically not based on a row-column table structure. At least some of them do, however, make the claim that they support ACID properties (Neo4j).

NoSQL Examples

In this section, we will discuss examples of NoSQL database management systems that represent the most popular instances of each of the categories previously discussed. The selection has been made based on rankings maintained by db-engines.com. The marketplace of NoSQL products is still broad and consists of a large number of products that are fighting for a chance for eventual long-term success. Still, each of the categories has a clear leader at least in terms of the number of adopters. We will discuss these products in this section.

REDIS Redis is the most popular key-value store NoSQL database management system. As most of the others discussed in this section, Redis is an open source product and, according to db-engines.com, by far the most widely used key-value store. Its keys can include various complex data structures (including strings, hashes, lists, sets, and sorted sets) in addition to simple numeric values. In addition, Redis makes it possible to perform various atomic operations on the key types, extending the generic key-value store feature set previously discussed. Many highly popular Web properties use Redis, the reputation of which is largely based on its high performance, enabled by its support for in-memory operations.

MONGODB The clear leader in the document store category (and also the most popular NoSQL database management system in general) is MongoDB, also an open source product. MongoDB offers a broader range of capabilities than Redis and is not as strongly focused solely on performance. MongoDB offers versatile indexing, high availability through automated replication, a query mechanism, its own file system for storing large objects, and automatic sharding for distributing the processing load between multiple servers. It does not, however, support joins or transactions. Instead of JSON, MongoDB uses BSON as its storage format. BSON is a binary JSON-like structure that is designed to be easy and quick to traverse and fast to encode and decode. In practice, this means that it is easier and faster to find things within a BSON structure than within a JSON structure.

APACHE CASSANDRA The main player in the wide-column store category is Apache Cassandra, which also competes with MongoDB for the leading NoSQL DBMS position (although Cassandra still has a much smaller user base than MongoDB). Google's BigTable algorithm was a major inspiration underlying Cassandra, as was also Amazon's Dynamo; thus, some call Cassandra a marriage between BigTable and Dynamo. Cassandra uses a row/column structure, but as with other wide-column stores, rows are extensible (i.e., they do not necessarily follow the same structure), and it has multiple column grouping levels (columns, supercolumns, and column families).

NEO4J Finally, Neo4j is a graph database that was originated by Neo Technologies in 2003, before the NoSQL concept was coined. As previously mentioned, Neo4j supports ACID properties. It is highly scalable, enabling the storage of billions of nodes and relationships, and fast for the purposes for which it has been designed, that is, understanding complex relationships specified as graphs. It has its own declarative query language called Cypher; in addition to the queries, Cypher is used to create new nodes and relationships and manage indexes and constraints (in the same way SQL is used for inserting data and managing relational database indexes and constraints).

Impact of NoSQL on Database Professionals

From the perspective of a database professional, it is truly exciting that the introduction of NoSQL database management systems has made a rich variety of new tools available for the management of complex and variable data. Relational database management systems and SQL will continue to be very important for many purposes, particularly

TABLE 11-2 Comparison of NoSQL Database Characteristics (Based on Scofield, 2010)

	Key-Value Store	Document Store	Column Oriented	Graph
Performance	high	high	high	variable
Scalability	high	variable/high	high	variable
Flexibility	high	high	moderate	high
Complexity	none	low	low	high
Functionality	variable	variable (low)	minimal	graph theory

Source: http://www.slideshare.net/bscofield/nosql-codemash-2010. Courtesy of Ben Scofield.

in the context of administrative systems that require a high level of predictability and structure. In addition, SQL will continue to be an important foundation for new data manipulation and definition languages that are created for different types of contexts because of SQL's very large existing user base. The exciting new tools under the NoSQL umbrella add a significant set of capabilities to an expert data management professional's toolkit. For a long time, relational DBMSs were the primary option for managing organizational data; the NoSQL database management systems discussed in this section provide the alternatives that allow organizations to make informed decisions regarding the composition of their data management arsenal. Table 11-2 provides an example of a comparative review of these four categories of NoSQL technologies.

Hadoop

There is probably no current data management product or platform discussed as broadly as Hadoop. At times it seems that the entire big data discussion revolves around Hadoop, and it is easy to get the impression that there would be no big data analytics without Hadoop. The truth is not, of course, this simple. The purpose of this section is to give you an overview of Hadoop and help you understand its true importance and the purposes for which it can be effectively used. It is an important technology that provides significant benefits for many (big) data management tasks, and Hadoop has helped organizations achieve important analytics results that would not have been possible without it. However, it is also important to understand that Hadoop is not a solution for all data management problems; instead, it is a tool in the data management toolbox that needs to be used for the right purposes.

MapReduce
An algorithm for massive parallel processing of various types of computing tasks.

Hadoop
An open source implementation framework of MapReduce.

The foundation of Hadoop is **MapReduce**, an algorithm for massive parallel processing of various types of computing tasks originally published in a paper by two Google employees in the early 2000s (Dean and Ghemawat, 2004). The key purpose of MapReduce is to automate the parallelization of large-scale tasks so that they can be performed on a large number of low-cost commodity servers in a fault-tolerant way. **Hadoop**, in turn, is an open-source implementation framework of MapReduce that makes it easier (but not easy) to apply the algorithm to a number of real-world problems. As will be discussed below, Hadoop consists of a large number of components integrated with each other. It is also important to understand that Hadoop is fundamentally a batch-processing tool. That is, it has been designed for tasks that can be scheduled for execution without human intervention at a specific time or under specific conditions (e.g., low processing load or a specific time of the day).

Thus, Hadoop is not a tool that you would run on a local area network to address the administrative data processing needs of a small or middle-sized company. It is also not a tool that you can easily demonstrate on a single computer (however powerful its processing capabilities might be). Hadoop's essence is in processing very large amounts (terabytes or petabytes) of data by distributing the data (using Hadoop Distributed File System or HDFS) and processing task(s) among a large number of low-cost commodity servers.

A large number of projects powered by Hadoop are described in **http://wiki.apache.org/hadoop/PoweredBy**; the smallest of them have only a few nodes but most of them dozens and some hundreds or more (for example, Facebook describes an 1100-machine (8800 core) system with 12 petabytes of storage). Hadoop is also not a tool

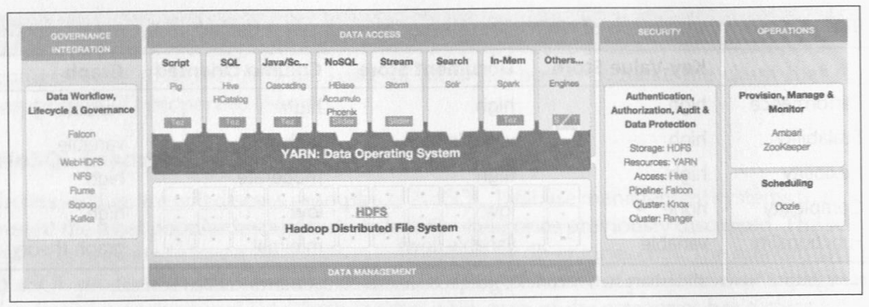


FIGURE 11-4 Hortonworks Enterprise Hadoop Data Platform

Adapted from http://hortonworks.com/hdp. Courtesy of HortonWorks, Inc.

that you manage and use with a high-level point-and-drag interface; submitting even a simple MapReduce job to Hadoop typically requires the use of the Java programming language and specific Hadoop libraries. Fortunately, many parties have built tools that make it easier to use the capabilities of Hadoop.

Components of Hadoop

The Hadoop framework consists of a large number of components that together form an implementation environment that enables the use of the MapReduce algorithm to solve practical large-scale analytical problems. These components will be the main focus of this section. Figure 11-4 includes a graphical representation of a Hadoop component architecture for an implementation by Hortonworks.

HDFS

HDFS or Hadoop Distributed File System is a file system designed for managing a large number of potentially very large files in a highly distributed environment.

THE HADOOP DISTRIBUTED FILE SYSTEM (HDFS) **HDFS** is the foundation of the data management infrastructure of Hadoop. It is not a relational database management system or any type of DBMS; instead, it is a file system designed for managing a large number of potentially very large files in a highly distributed environment (up to thousands of servers). HDFS breaks data into small chunks called blocks and distributes them on various computers (nodes) throughout the Hadoop cluster. This distribution of data forms the foundation for Hadoop's processing and storage model: Because data are divided between various nodes in the cluster, it can be processed by all those nodes at the same time.

Data in HDFS files cannot be updated; instead, it can only be added at the end of the file. HDFS does not provide indexing; thus HDFS is not usable in applications that require real-time sequential or random access to the data (White, 2012). HDFS assumes that hardware failure is a norm in a massively distributed environment; with thousands of servers, some hardware elements are always in a state of failure and thus HDFS has been designed to quickly discover the component failures and recover from them (HDFSDesign, 2014). Another important principle underlying HDFS is that it is cheaper to move the execution of computation to the data than to move the data to computation.

A typical HDFS cluster consists of a single master server (NameNode) and a large number of slaves (DataNodes). The NameNode is responsible for the management of the file system name space and regulating the access to files by clients (HDFSDesign, 2014). Replication of data is an important characteristic of HDFS. By default, HDFS maintains three copies of data (both the number of copies and the size of data blocks can be configured). An interesting special characteristic of HDFS is that it is aware of the positioning of nodes in racks and can take this information into account when designing its replication policy. Since Hadoop 2.0, it has been possible to maintain two redundant NameNodes in the same cluster to avoid the NameNode becoming a single point of failure. See Figure 11-5 for an illustration of a HDFS Cluster associated with MapReduce.

A highly distributed system requires a traffic cop that controls the allocation of various resources available in the system. In the current version of Hadoop (Hadoop 2), this component is called YARN (Yet Another Resource Allocator, also called MapReduce 2.0). YARN consists of a global ResourceManager and a per-application ApplicationMaster, and its fundamental role is to provide access to the files stored on HDFS and to organize the processes that utilize this data (see also Figure 11-4).

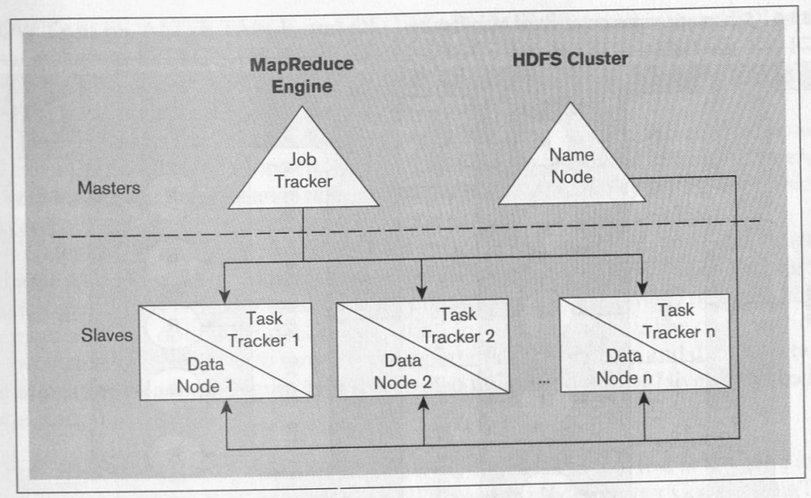


FIGURE 11-5 MapReduce and HDFS

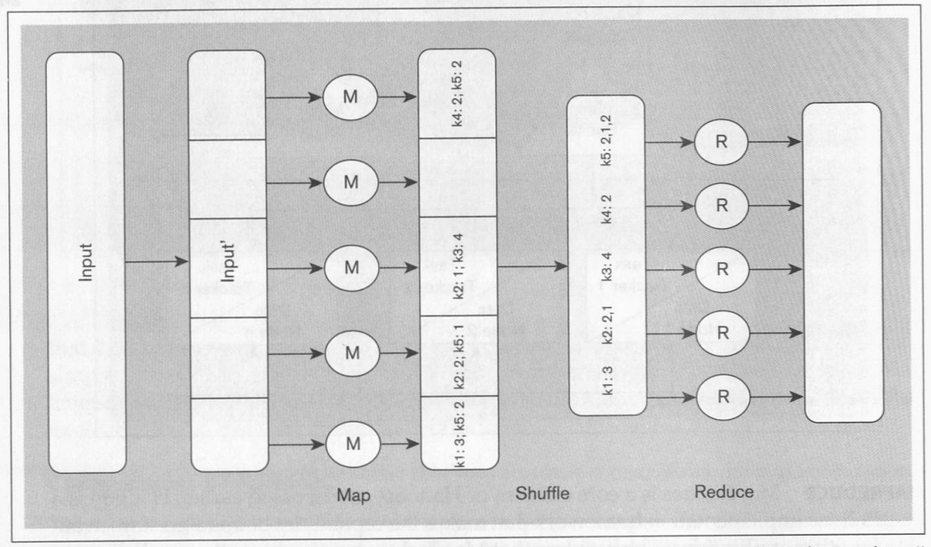
MAPREDUCE MapReduce is a core element of Hadoop; as discussed earlier, Hadoop is a MapReduce implementation framework that makes the capabilities of this algorithm available for other applications. The problem that MapReduce helps solve is the parallelization of data storage and computational problem solving in an environment that consists of a large number of commodity servers. MapReduce has been designed so that it can provide its capabilities in a fault-tolerant way. The authors of the original MapReduce article (Dean and Ghemawat, 2004) specifically state that MapReduce is intended to allow "programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system" (Dean and Ghemawat, 2004, p. 1). MapReduce intends to make the power of parallel processing available to a large number of users so that (programmer) users can focus on solving the domain problem instead of having to worry about complex details related to the management of parallel systems. In the component architecture represented in Figure 11-4, MapReduce is integrated with YARN.

The core idea underlying the MapReduce algorithm is dividing the computing task so that multiple nodes of a computing cluster can work on the same problem at the same time. Equally important is that each node is working on local data and only the results of processing are moved across the network, saving both time and network resources. The name of MapReduce comes from the names of the components of this distribution process. The first part, *map*, performs a computing task in parallel on multiple subsets of the entire data, returning a result for each subset separately. The second part, *reduce*, integrates the results of each of the *map* processes, creating the final result. It is up to the developer to define the mapper and the reducer so that they together get the work done. See Figure 11-6 for a schematic representation.

Let's look at an example. Imagine that you have a very large number of orders and associated orderline data (with attributes productID, price, and quantity), and your goal is to count the number of orders in which each productID exists and the average price for each productID. Let's assume that the volumes are so high that using a traditional RDBMS to perform the task is too slow. If you used the MapReduce algorithm to perform this task, you would define the mapper so that it would produce the following (key → value) pairs: (productID → [1, price]) where productID is the key and the [1, price] pair is the value. The mapper on each of the nodes could independently produce these pairs that, in turn, would be used as input by the reducer. The reducer would create a set of different types of (key → value) pairs: For each productID, it would produce a count of orders and the average of all the prices in the form of (productID → [countOrders, avgPrice].

In this case, the mapper and reducer algorithms are very simple. Sometimes this is the case with real-world applications; sometimes those algorithms are quite complex. For example, **http://highlyscalable.wordpress.com/2012/02/01/mapreduce-patterns/** presents a number of interesting and relevant uses for MapReduce. It is important to note that in many cases these types of tasks can be performed easily and without any extra effort with

FIGURE 11-6 Schematic representation of MapReduce



MapReduce: Simplified Data Processing on Large Clusters, Jeff Dean, Sanjay Ghemawat, Google, Inc., http://research.google.com/archive/mapreduce-osdi04-slides/index-auto-0007.html. Courtesy of the authors.

a RDBMS—only when the amounts of data are very large, data types are highly varied, and/or the speeds of arrival of data are very high (i.e., we are dealing with real big data) do massively distributed approaches, such as Hadoop, produce real advantages that justify the additional cost in complexity and the need for an additional technology platform.

In addition to HDFS, MapReduce, and YARN, other components of the Hadoop framework have been developed to automate the computing tasks and raise the abstraction level so that Hadoop users can focus on organizational problem solving. These tools also have unusual names, such as Pig, Hive, and Zookeeper. The rest of the section provides an overview of these remaining components.

Pig
A tool that integrates a scripting language and an execution environment intended to simplify the use of MapReduce.

PIG MapReduce programming is difficult, and multiple tools have been developed to address the challenges associated with it. One of the most important of them is called **Pig**. This platform integrates a scripting language (appropriately called PigLatin) and an execution environment. Its key purpose is to translate execution sequences expressed in PigLatin into multiple sequenced MapReduce programs. The syntax of Pig is familiar to those who know some of the well-known scripting languages. In some contexts it is also called SQL-like (**http://hortonworks.com/hadoop-tutorial/how-to-use-basic-pig-commands/**), although it is not a declarative language.

Pig can automate important data preparation tasks (such as filter rows that do not include useful data), transform data for processing (e.g., convert text data into all lower case or extract only needed data elements), execute analytic functions, store results, define processing sequences, etc. All this is done at a much higher level of abstraction than would be possible with Java and direct use of MapReduce libraries. Not surprisingly, Pig is quite useful for ETL (extract–transform–load) processes (discussed in Chapter 10), but it can also be used for studying the characteristics of raw data and for iterative processing of data (**http://hortonworks.com/hadoop/pig/**). Pig can be extended with custom functions (UDFs or user defined functions). See Figure 11-4 for an illustration of how Pig fits the overall Hadoop architecture.

Hive
An Apache project that supports the management and querying of large data sets using HiveQL, an SQL-like language that provides a declarative interface for managing data stored in Hadoop.

HIVE I am sure you are happy to hear that the SQL skills you've learned earlier are also applicable in the big data context. Another Apache project called **Hive** (which Apache calls "data warehouse software") supports the management of large data sets and querying them. HiveQL is an SQL-like language that provides a declarative interface for managing data stored in Hadoop. HiveQL includes DDL operations (CREATE TABLE, SHOW TABLES, ALTER TABLE, and DROP TABLE), DML operations, and SQL operations, including SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY, JOIN, UNION, and subqueries. HiveQL also supports limiting the answer to top [x] rows with LIMIT [x] and using regular expressions for column selection.

At runtime, Hive creates MapReduce jobs based on the HiveQL statements and executes them on a Hadoop cluster. As with Hadoop in general, HiveQL is intended for very large scale data retrieval tasks primarily for data analysis purposes; it is specifically not a language for transaction processing or fast retrieval of single values.

Gates (2010) discusses the differences between Pig and Hive (and, consequently, PigLatin and HiveQL) at Yahoo!, and illustrates well the different uses for the two technologies. Pig is typically used for data preparation (or *data factory*) whereas Hive is the more suitable option for data presentation (or *data warehouse*). The combination of the two has allowed Yahoo! to move a major part of its data factory and data warehouse operations into Hadoop. Figure 11-4 shows also the positioning of Hive in the context of the Hadoop architecture.

HBASE The final Hadoop component that we will discuss in this text is HBase, a wide-column store database that runs on top of HDFS and is modeled after Google's BigTable (Chang et al., 2008). As discussed earlier in this chapter, another Apache project called Cassandra is more popular in this context. A detailed comparison between HBase and Cassandra is beyond the scope of this text; both products are used to support projects with massive data storage needs. It is, however, important to understand that HBase does not use MapReduce; instead, it can serve as a source of data for MapReduce jobs.

Integrated Analytics and Data Science Platforms

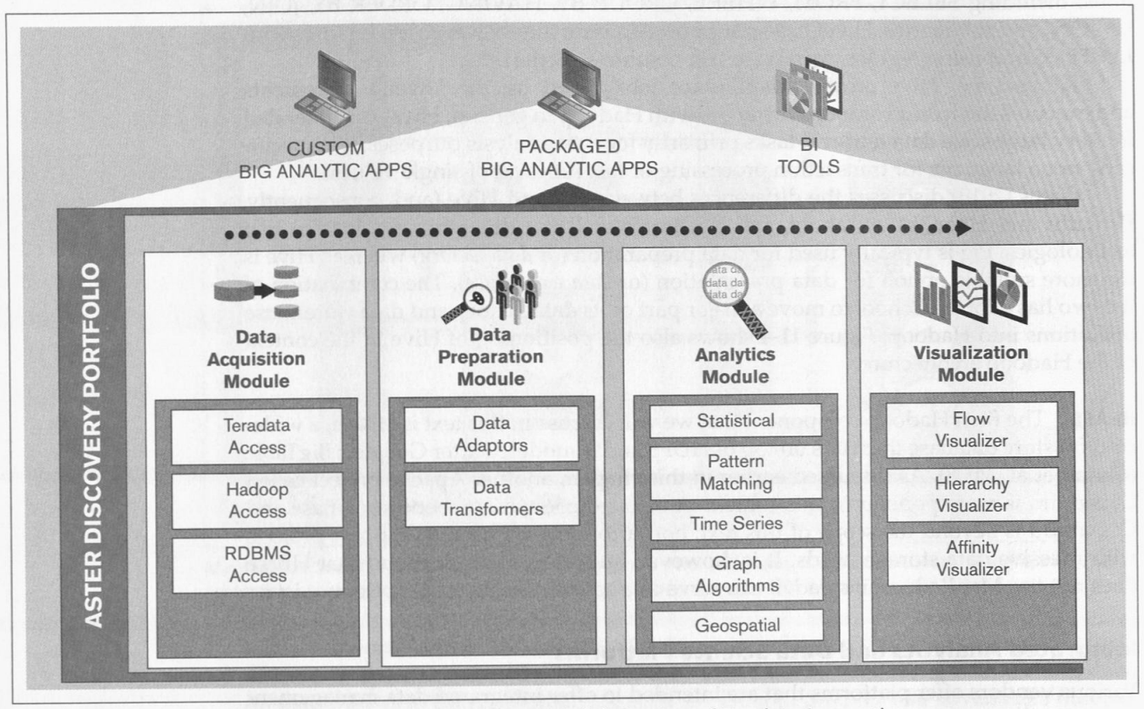
Various vendors offer platforms that are intended to offer integrated data management capabilities for analytics and data science. They bring together traditional data warehousing (discussed in Chapter 9) and the big data–related capabilities discussed earlier. In this section, we will briefly discuss the key characteristics of a few of them in order to demonstrate the environments that organizations are using to make big data work in practice. They include HP's HAVEn, Teradata's Aster, and IBM's Big Data Platform.

HP HAVEn HP HAVEn is a platform that integrates some core HP technologies with open source big data technologies, promising an ability to derive insights fast from very large amounts of data stored on Hadoop/HDFS and HP's Vertica column-oriented data store. Vertica is based on an academic prototype called C-Store (Lamb et al., 2012) and was acquired by HP in 2011. In addition to Hadoop and Vertica, HAVEn includes an Autonomy analytics engine with a particular focus on unstructured textual information.

TERADATA ASTER Teradata, one of the long-term leaders in data warehousing, has recently extended its product offerings to cover both big data analytics and marketing applications as two additional pillars of its strategy. In big data, the core of its offering is based on a 2011 acquisition called Aster. One of the core ideas of Aster is to integrate a number of familiar analytics tools (such as SQL, extensions of SQL for graph analysis and access to MapReduce data stores, and the statistical language R) with several different analytical data store options. These, in turn, are connected to a variety of external sources of data. Figure 11-7 shows a schematic representation of the Aster platform.

IBM BIG DATA PLATFORM IBM brings together in its Big Data Platform a number of components that offer similar capabilities to those previously described in the context of HP and Teradata. IBM's commercial distribution of Hadoop is called InfoSphere BigInsights. In addition to standard Hadoop capabilities IBM offers JSON Query Language (JAQL), a high-level functional, declarative query language for analyzing large-scale semi-structured data that IBM describes as a "blend of Pig and Hive." Moreover, BigInsights offers connectors to IBM's DB2 relational database and analytics data sources such as Netezza, IBM's 2010 acquisition. Netezza is a data warehousing appliance that allows fast parallel processing of query and analytics tasks against

FIGURE 11-7 Teradata Aster Discovery Portfolio



Source: http://www.teradata.com/Teradata-Aster-Discovery-Portfolio/. Courtesy of Teradata Corporation.

large amounts of data. DB2, BigInsights, Netezza, and IBM's enterprise data warehouse Smart Analytics System all are feeding into analytics tools such as Cognos and SPSS.

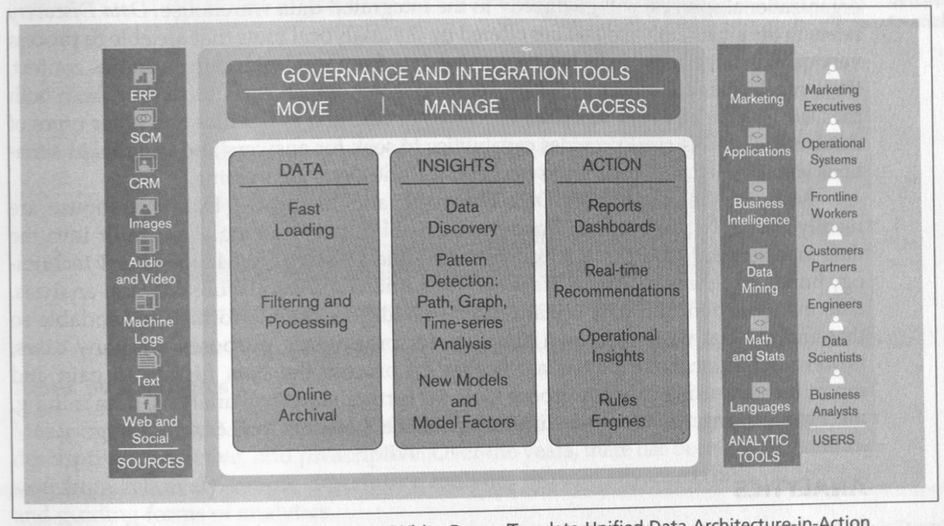
Putting it All Together: Integrated Data Architecture

To help you understand all of this together, we will be using a framework description from one of the vendors discussed earlier. Teradata has developed a model illustrating how various elements of a modern data management environment belong together. It is called Unified Data Architecture and presented in Figure 11-8.

In this model, the various *Sources* of data are included on the left side. These are the generators of data that the data management environment will collect and store for processing and analysis. They include various enterprise systems (ERP, SCM, CRM) and other similar structured data sources, data collected from the Web and various social media sources, internal and external text documents, and multimedia sources (still images, audio, and video). This version of the model includes machine logs (capturing what takes place in various devices that together comprise the organizational systems). These could be extended with sensor data and other Internet of Things sources (data generated with devices serving various practical purposes both in households, corporations, and public organizations and spaces).

In the middle are the three core activities that are required for transforming the raw data from the sources to actionable insights for the *Users* on the right. They include preparing the *Data*, enabling *Insights*, and driving *Action*. The *Data* category refers to the actions that bring the data into the system from the sources, process it to analyze and ensure its quality, and archive it. *Insights* refer to the activities that are needed for making sense of the data through data discovery, pattern recognition, and development of new models. Finally, the *Action* category produces results that can be put to action as direct recommendations, insights, or rules. Alternatively, action support can be generated through reports and dashboards that users will use to support their decision making. *Insights* and *Action* are achieved through various *Analytic tools* used either by professional analysts and data scientists or directly by managers.

FIGURE 11-8 Teradata Unified Data Architecture – logical view



Source: http://www.teradata.com/Resources/White-Papers/Teradata-Unified-Data-Architecture-in-Action. Courtesy of Teradata Corporation.

Figure 11-9 presents an implementation perspective of the same model using Teradata's technologies. For our purposes the most interesting element of this version is the division of labor between the three components. *Data Platform* refers to the capabilities that are required to capture or retrieve the data from the *Sources*, store it for analytical purposes, and prepare it for statistical analysis (by, for example, ensuring the quality of the data to the extent it is possible). The capabilities of Hadoop would be used in this context to manage, distribute, and process in parallel the large amounts of data generated by the sources. *Integrated Data Warehouse* is the primary context for analytics that supports directly ongoing strategic and operational analytics, activities that are often planned and designed to support ongoing business. This element of the model is familiar to you from

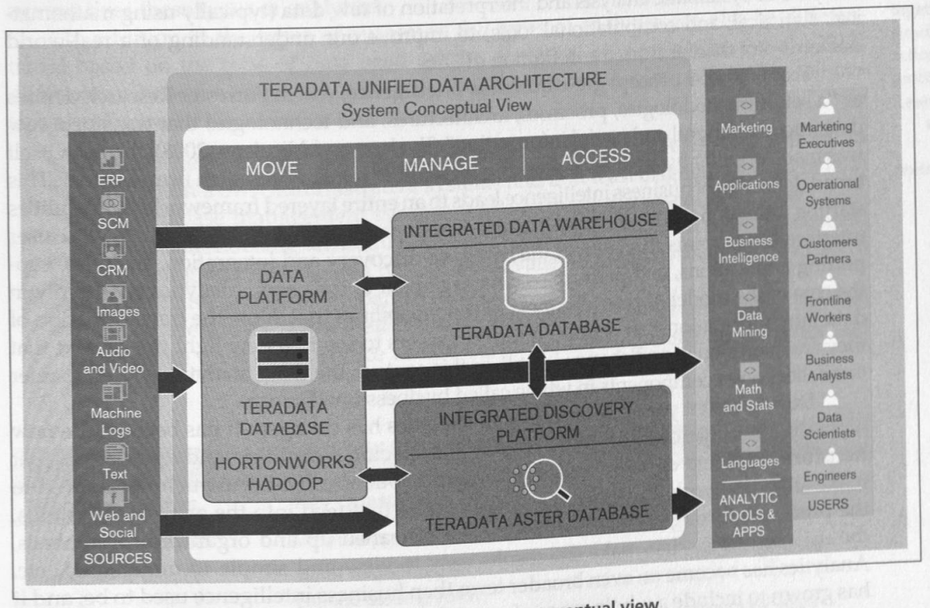


FIGURE 11-9 Teradata Unified Data Architecture – system conceptual view

Source: http://www.teradata.com/Resources/White-Papers/Teradata-Unified-Data-Architecture-in-Action. Courtesy of Teradata Corporation.

Chapter 9. Please note that some of the data (particularly structured data from traditional organizational sources) will go directly to the integrated data warehouse. *Data Discovery* refers to the exploratory capabilities offered by the analytical tools that are able to process very quickly large amounts of heterogeneous data from multiple sources. In this context, these capabilities are implemented by Teradata Aster, which can utilize data from both the *Data Platform* and *Integrated Data Warehouse* in addition to flat files and other types of databases. *Data Discovery* provides capabilities to seek for answers and insights in situations when sometimes both the answers and the questions are missing.

Many of the results from Data Discovery and Integrated Data Warehouse are readily usable by the analysts. Analytical capabilities increasingly are built into the data management products. For example, Teradata's Aster SQL-MapReduce® technology builds into the familiar SQL framework additional functions for statistical analysis, data manipulation, and data visualization. In addition, the platform is expandable so that analysts can write their own functions for proprietary purposes. In many cases, however, additional capabilities are needed to process the data further to gain and report insights, using special-purpose tools for further statistical analysis, data mining, machine learning, and data visualization, all in the *Analytics Tools & Apps* category.

ANALYTICS

During your earlier studies of Information Systems related topics, you might have encountered several concepts that are related to analytics. One of the earliest is decision support systems (DSS), which was one of the early information system types in a commonly used typology, together with transaction processing systems (TPS), management information systems (MIS), and executive information systems (EIS). Sprague (1980) characterized decision support systems as systems that support less structured and underspecified problems, use models and analytic techniques together with data access, have features that make them accessible by non-technical users, and are flexible and adaptable for different types of questions and problems. In this classification, one of the essential differences between structured and pre-defined MIS systems and DSS systems was that the former produced primarily pre-specified reports. The latter were designed to address many different types of situations and allowed the decision maker to change the nature of the support they received from the system depending on their needs. Earlier we defined analytics as systematic analysis and interpretation of raw data (typically using mathematical, statistical, and computational tools) to improve our understanding of a real-world domain—not that far from the definition of DSS.

Business intelligence
A set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information.

From the DSS concept grew **business intelligence**, which Forrester Research defines as "a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information" (Evelson and Nicolson, 2008); the term itself was made popular by an analyst working for another major IT research firm, Gartner. This broad definition of business intelligence leads to an entire layered framework of capabilities starting from foundational infrastructure components and data and ending with user interface components that deliver the results of discovery and integration, analytics, supporting applications, and performance management to the users. Analytics is certainly in the core of the model. It provides most of the capabilities that allow the transformation of data into information that enables decision makers to see in a new light the context that they are interested in and change it. Still, in this context, the word analytics is used to refer to a collection of components in whole called business intelligence.

During recent years the meaning of analytics has changed. It has become the new umbrella term that encompasses not only the specific techniques and approaches that transform collected data into useful information but also the infrastructure required to make analytics work, the various sources of data that feed into the analytical systems, the processes through which the raw data are cleaned up and organized for analysis, the user interfaces that make the results easy to view and simple to understand, etc. Analytics has become an even broader term than business intelligence used to be, and it has grown to include a whole range of capabilities that allow an organization to provide analytical insights. The transition from decision support systems to analytics through business intelligence is described in Figure 11-10.

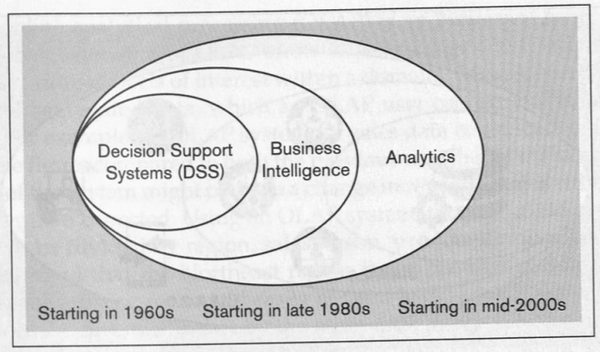


FIGURE 11-10 Moving from decision support systems to analytics

Types of Analytics

Many authors, including Watson (2014), divide analytics into three major categories: descriptive, predictive, and prescriptive. Over the years, there has been a clear progression from relatively simple descriptive analytics to more advanced, forward looking and guiding forms of analytics.

Descriptive analytics
Describes the past status of the domain of interest using a variety of tools through techniques such as reporting, data visualization, dashboards, and scorecards.

Predictive analytics
Applies statistical and computational methods and models to data regarding past and current events to predict what might happen in the future.

Prescriptive analytics
Uses results of predictive analytics together with optimization and simulation tools to recommend actions that will lead to a desired outcome.

Descriptive analytics is the oldest form of analytics. As the name suggests, it primarily focuses on describing the past status of the domain of interest using a variety of tools through techniques such as reporting, data visualization, dashboards, and scorecards. Online analytical processing (OLAP) is also part of descriptive analytics; it allows users to get a multidimensional view of data and drill down deeper to the details when appropriate and useful. The key emphasis of predictive analytics is on the future. **Predictive analytics** systems apply statistical and computational methods and models to data regarding past and current events to predict what might happen in the future (potentially depending on a number of assumptions regarding various parameters). Finally, **prescriptive analytics** focuses on the question "How can we make it happen?" or "What do we need to do to make it happen?" For prescriptive analysis we need optimization and simulation tools and advanced modeling to understand the dependencies between various actors within the domain of interest. Table 11-3 summarizes these types of analytics.

In addition to the types of outcomes, the types of analytics can also be differentiated based on the type of data used for the analytical processes. Chen et al. (2012) differentiate between three eras of Business Intelligence and Analytics (BI&A) as follows (see also Figure 11-11):

- BI&A 1.0 deals mostly with *structured quantitative data* that originates from an organization's own administrative systems and is at least originally stored in relational database management systems (such as those discussed in Chapters 4–7). The data warehousing techniques described in Chapter 9 are an essential element in preparing and making this type of data available for analysis. Both descriptive and predictive analytics are part of BI&A 1.0.
- BI&A 2.0 refers to the use of the *data that can be collected from Web-based sources*. The Web has become a very rich source of data for understanding customer behavior and

TABLE 11-3 Types of Analytics

Type of Analytics	Key Questions
Descriptive Analytics	What happened yesterday/last week/last year?
Predictive Analytics	What might happen in the future? How does this change if we change assumptions?
Prescriptive Analytics	How can we make it happen? What needs to change to make it happen?

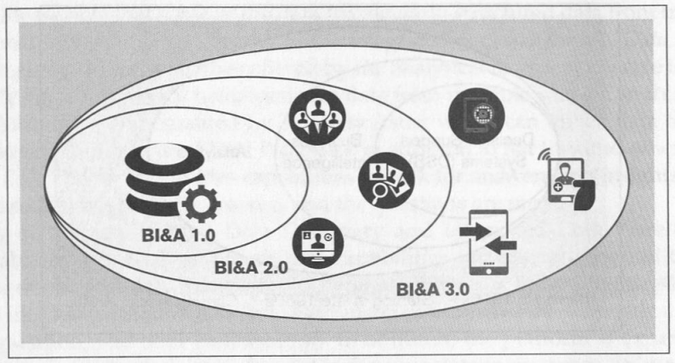


FIGURE 11-11 Generations of business intelligence and analytics

Adapted from Chen et al., 2012.

interaction both between organizations and their stakeholders and among various stakeholder groups at a much more detailed level than earlier. From an individual organization's perspective, this data includes data collected from various Web interaction logs, Web-based customer communication platforms, and social media sources. Much of this data is text-based in nature, thus the analytical techniques used to process it are different from those used for BI&A 1.0, including text mining, Web mining, and social network analysis. To achieve the most effective results, these techniques should be integrated with the more traditional approaches.

- BI&A 3.0 is based on an even richer and more individualized data based on the *ubiquitous use of mobile devices* that have the capability of producing literally millions of observations per second from various sensors, capturing measurements such as guaranteed identification, location, altitude, speed, acceleration, direction of movement, temperature, use of specific applications, etc. The number of smartphones is already counted in billions. The Internet of Things (Chui, Löffler, and Roberts, 2010) adds yet another dimension to this: An increasingly large number of technical devices and their components are capable of producing and communicating data regarding their status. The opportunities to improve the effectiveness and efficiency of the way in which we individually and collectively work to achieve our goals are very significant.

As discussed earlier in this section, analytics is often divided into three categories: descriptive analytics, predictive analytics, and prescriptive analysis. We will next discuss these categories at a more detailed level, illustrating how these technologies can be used for analytical purposes.

Use of Descriptive Analytics

Most of the user interface tools associated with traditional data warehouses will provide capabilities for descriptive analytics, which, as we discussed earlier in this section, primarily focuses on describing the status of the domain of interest from the historical perspective. This was also the original meaning of the widely used term *business intelligence*.

Descriptive analytics is the oldest form of analytics. As the name suggests, it primarily focuses on describing the past status of the domain of interest using a variety of tools. The simplest form of descriptive analytics is the *reporting* of aggregate quantitative facts regarding various objects of interest, such as quarterly sales per region, monthly payroll by division, or the average length of a hospital stay per department. Aggregated data can be reported either in a tabular form or using various tools and techniques of *data visualization*. When descriptive data are aggregated in to a few key indicators, each of which integrates and represents an important aspect of the domain of interest, descriptive analytics is said to use a *dashboard*. A *scorecard* might include a broader range of more detailed indicators, but still, a scorecard reports descriptive data regarding past behavior.

Finally, online analytical processing (OLAP) is an important form of descriptive analytics. Key characteristics of OLAP allow its users to get an in-depth multidimensional view of various aspects of interest within a domain. Typical OLAP processes start with high-level aggregated data, which an OLAP user can explore from a number of perspectives. For example, an OLAP system for sales data could start with last month's overall revenue figure compared to both the previous month and the same month a year ago. The user of the system might observe a change in revenue that is either significantly higher or lower than expected. Using an OLAP system, the user could easily ask for the total revenue to be divided by region, salesperson, product, or division. If a report by region demonstrated that the Northeast region is the primary reason underlying the decrease in revenue, the system could easily be used to drill down to the region in question and explore the revenue further by the other dimensions. This could further show that the primary reason for the decrease within the region is a specific product. Within the product, the problem could be narrowed down to a couple of salespeople. OLAP allows very flexible ad hoc queries and analytical approaches that allow quick adaptation of future questions to the findings made previously. Speed of execution is very important with OLAP databases.

Many of the data warehousing products discussed in Chapter 9 are used for various forms of descriptive analytics. According to Gartner (Edjlali and Beyer, 2013), the leaders of the underlying data warehousing products include Teradata (including Aster), Oracle (including Oracle Exadata), IBM (Netezza), SAP (Sybase IQ and Hana), Microsoft (SQL Server 2012 Parallel Data Warehouse), and EMC (Greenplum). Building on these foundational products, specific business intelligence and analytics platforms provide deeper analytical capabilities. In this category, Gartner (Sallam et al., 2014) identified Tableau, Qlik, Microsoft, IBM, SAS, SAP, Tibco, Oracle, MicroStrategy, and Information Builders as leading vendors. The descriptive capabilities that Gartner expected a product to have to do well in this category included reporting, dashboards, ad hoc reports/queries, integration with Microsoft Office, mobile business intelligence, interactive visualization, search-based data discovery, geospatial and location intelligence, and OLAP.

In this section, we will discuss a variety of tools for querying and analyzing data stored in data warehouses and data marts. These tools can be classified, for example, as follows:

- Traditional query and reporting tools
- OLAP, MOLAP, and ROLAP tools
- Data visualization tools
- Business performance management and dashboard tools

Traditional query and reporting tools include spreadsheets, personal computer databases, and report writers and generators. We do not describe these commonly known tools in this chapter. Instead, we assume that you have learned them somewhere else in your program of study.

SQL OLAP QUERYING The most common database query language, SQL (covered extensively in Chapters 6 and 7), has been extended to support some types of calculations and querying needed for a data warehousing environment. In general, however, SQL is not an analytical language (Mundy, 2001). At the heart of analytical queries is the ability to perform categorization (e.g., group data by dimension characteristics), aggregation (e.g., create averages per category), and ranking (e.g., find the customer in some category with the highest average monthly sales). Consider the following business question in the familiar Pine Valley Furniture Company context:

> Which customer has bought the most of each product we sell? Show the product ID and description, customer ID and name, and the total quantity sold of that product to that customer; show the results in sequence by product ID.

Even with the limitations of standard SQL, this analytical query can be written without the OLAP extensions to SQL. One way to write this query, using the large version of the Pine Valley Furniture database provided with this textbook, is as follows:

```
SELECT P1.ProductId, ProductDescription, C1.CustomerId,
  CustomerName, SUM(OL1.OrderedQuantity) AS TotOrdered
  FROM Customer_T AS C1, Product_T AS P1, OrderLine_T
    AS OL1, Order_T AS O1
  WHERE C1.CustomerId = O1.CustomerId
    AND O1.OrderId = OL1.OrderId
    AND OL1.ProductId = P1.ProductId
  GROUP BY P1.ProductId, ProductDescription,
    C1.CustomerId, CustomerName
  HAVING TotOrdered >= ALL
  (SELECT SUM(OL2.OrderedQuantity)
  FROM OrderLine_T AS OL2, Order_T AS O2
  WHERE OL2.ProductId = P1.ProductId
    AND OL2.OrderId = O2.OrderId
    AND O2.CustomerId <> C1.CustomerId
  GROUP BY O2.CustomerId)
ORDER BY P1.ProductId;
```

This approach uses a correlated subquery to find the set of total quantity ordered across all customers for each product, and then the outer query selects the customer whose total is greater than or equal to all of these (in other words, equal to the maximum of the set). Until you write many of these queries, this can be very challenging to develop and is often beyond the capabilities of even well-trained end users. Even this query is rather simple because it does not have multiple categories, does not ask for changes over time, or does not want to see the results graphically. Finding the second in rank is even more difficult.

Some versions of SQL support special clauses that make ranking questions easier to write. For example, Microsoft SQL Server and some other RDBMSs support clauses of FIRST n, TOP n, LAST n, and BOTTOM n rows. Thus, the query shown previously could be greatly simplified by adding TOP 1 in front of the SUM in the outer query and eliminating the HAVING and subquery. TOP 1 was illustrated in Chapter 7, in the section on "More Complicated SQL Queries."

Recent versions of SQL include some data warehousing and business intelligence extensions. Because many data warehousing operations deal with categories of objects, possibly ordered by date, the SQL standard includes a WINDOW clause to define dynamic sets of rows. (In many SQL systems, the word OVER is used instead of WINDOW, which is what we illustrate next.) For example, an OVER clause can be used to define three adjacent days as the basis for calculating moving averages. (Think of a window moving between the bottom and top of its window frame, giving you a sliding view of rows of data.) PARTITION BY within an OVER clause is similar to GROUP BY; PARTITION BY tells an OVER clause the basis for each set, an ORDER BY clause sequences the elements of a set, and the ROWS clause says how many rows in sequence to use in a calculation. For example, consider a SalesHistory table (columns TerritoryID, Quarter, and Sales) and the desire to show a three-quarter moving average of sales. The following SQL will produce the desired result using these OLAP clauses:

```
SELECT TerritoryID, Quarter, Sales,
  AVG(Sales) OVER (PARTITION BY TerritoryID
    ORDER BY Quarter ROWS 2 PRECEDING) AS 3QtrAverage
FROM SalesHistory;
```

The PARTITION BY clause groups the rows of the SalesHistory table by TerritoryID for the purpose of computing 3QtrAverage, and then the ORDER BY clause sorts by quarter within these groups. The ROWS clause indicates how many rows over which to calculate the AVG(Sales). The following is a sample of the results from this query:

TerritoryID	Quarter	Sales	3QtrAverage
Atlantic	1	20	20
Atlantic	2	10	15
Atlantic	3	6	12
Atlantic	4	29	15
East	1	5	5
East	2	7	6
East	3	12	8
East	4	11	10
...			

In addition, but not shown here, a QUALIFY clause can be used similarly to a HAVING clause to eliminate the rows of the result based on the aggregate referenced by the OVER clause.

The RANK windowing function calculates something that is very difficult to calculate in standard SQL, which is the row of a table in a specific relative position based on some criteria (e.g., the customer with the third-highest sales in a given period). In the case of ties, RANK will cause gaps (e.g., if there is a two-way tie for third, then there is no rank of 4, rather the next rank is 5). DENSE_RANK works the same as RANK but creates no gaps. The CUME_DIST function finds the relative position of a specified value in a group of values; this function can be used to find the break point for percentiles (e.g., what value is the break point for the top 10 percent of sales or which customers are in the top 10 percent of sales?).

Different DBMS vendors are implementing different subsets of the OLAP extension commands in the standards; some are adding capabilities specific to their products. For example, Teradata supports a SAMPLE clause, which allows samples of rows to be returned for the query. Samples can be random, with or without replacement, a percentage or count of rows can be specified for the answer set, and conditions can be placed to eliminate certain rows from the sample. SAMPLE is used to create subsets of a database that will be, for example, given different product discounts to see consumer behavior differences, or one sample will be used for a trial and another for a final promotion.

ONLINE ANALYTICAL PROCESSING (OLAP) TOOLS A specialized class of tools has been developed to provide users with multidimensional views of their data. Such tools also usually offer users a graphical interface so that they can easily analyze their data. In the simplest case, data are viewed as a three-dimensional cube.

Online analytical processing (OLAP) is the use of a set of query and reporting tools that provides users with multidimensional views of their data and allows them to analyze the data using simple windowing techniques. The term *online analytical processing* is intended to contrast with the more traditional term *online transaction processing (OLTP)*. The differences between these two types of processing were summarized in Table 9-1 in Chapter 9. The term *multidimensional analysis* is often used as a synonym for OLAP.

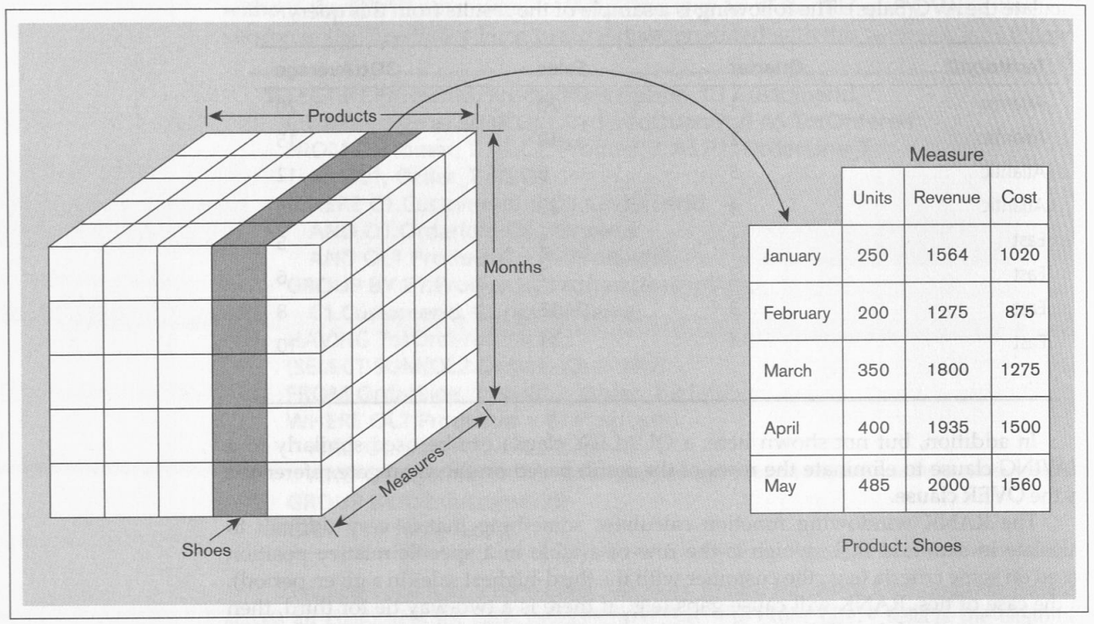
Online analytical processing (OLAP)
The use of a set of graphical tools that provides users with multidimensional views of their data and allows them to analyze the data using simple windowing techniques.

An example of a "data cube" (or multidimensional view) of data that is typical of OLAP is shown in Figure 11-12. This three-dimensional view corresponds quite closely to the star schema introduced in Chapter 9 in Figure 9-10. Two of the dimensions in Figure 11-12 correspond to the dimension tables (PRODUCT and PERIOD) in Figure 9-10, whereas the third dimension (named measures) corresponds to the data in the fact table (named SALES) in Figure 9-10.

OLAP is actually a general term for several categories of data warehouse and data mart access tools (Dyché, 2000). **Relational OLAP (ROLAP)** tools use variations of SQL and view the database as a traditional relational database, in either a star schema or

Relational OLAP (ROLAP)
OLAP tools that view the database as a traditional relational database in either a star schema or other normalized or denormalized set of tables.

FIGURE 11-12 Slicing a data cube



Measure

	Units	Revenue	Cost
January	250	1564	1020
February	200	1275	875
March	350	1800	1275
April	400	1935	1500
May	485	2000	1560

Product: Shoes

another normalized or denormalized set of tables. ROLAP tools access the data warehouse or data mart directly. **Multidimensional OLAP (MOLAP)** tools load data into an intermediate structure, usually a three or higher-dimensional array (hypercube). We illustrate MOLAP in the next few sections because of its popularity. It is important to note with MOLAP that the data are not simply viewed as a multidimensional hypercube, but rather a MOLAP data mart is created by extracting data from the data warehouse or data mart and then storing the data in a specialized separate data store through which data can be viewed only through a multidimensional structure. Other, less-common categories of OLAP tools are database OLAP (DOLAP), which includes OLAP functionality in the DBMS query language (there are proprietary, non-ANSI standard SQL systems that do this), and hybrid OLAP (HOLAP), which allows access via both multidimensional cubes or relational query languages.

Multidimensional OLAP (MOLAP)
OLAP tools that load data into an intermediate structure, usually a three- or higher-dimensional array.

Figure 11-12 shows a typical MOLAP operation: slicing the data cube to produce a simple two-dimensional table or view. In Figure 11-12, this slice is for the product named Shoes. The resulting table shows the three measures (units, revenues, and cost) for this product by period (or month). Other views can easily be developed by the user by means of simple "drag and drop" operations. This type of operation is often called *slicing and dicing* the cube. Another operation closely related to slicing and dicing is data pivoting (similar to the pivoting possible in Microsoft Excel). This term refers to rotating the view for a particular data point to obtain another perspective. For example, Figure 11-12 shows sales of 400 units of shoes for April. The analyst could pivot this view to obtain (for example) the sales of shoes by store for the same month.

Another type of operation often used in multidimensional analysis is *drill-down*—that is, analyzing a given set of data at a finer level of detail. An example of drill-down is shown in Figure 11-13. Figure 11-13a shows a summary report for the total sales of three package sizes for a given brand of paper towels: 2-pack, 3-pack, and 6-pack. However, the towels come in different colors, and the analyst wants a further breakdown of sales by color within each of these package sizes. Using an OLAP tool, this breakdown can be easily obtained using a "point-and-click" approach with a pointing device. The result of the drill-down is shown in Figure 11-13b. Notice that a drill-down presentation is equivalent to adding another column to the original report. (In this case, a column was added for the attribute color.)

FIGURE 11-13 Exampie of drill-down
a) Summary report

Brand	Package size	Sales
SofTowel	2-pack	$75
SofTowel	3-pack	$100
SofTowel	6-pack	$50

(b) Drill-down with color attribute added

Brand	Package size	Color	Sales
SofTowel	2-pack	White	$30
SofTowel	2-pack	Yellow	$25
SofTowel	2-pack	Pink	$20
SofTowel	3-pack	White	$50
SofTowel	3-pack	Green	$25
SofTowel	3-pack	Yellow	$25
SofTowel	6-pack	White	$30
SofTowel	6-pack	Yellow	$20

Executing a drill-down (as in this example) may require that the OLAP tool "reach back" to the data warehouse to obtain the detail data necessary for the drill-down. This type of operation can be performed by an OLAP tool (without user participation) only if an integrated set of metadata is available to that tool. Some tools even permit the OLAP tool to reach back to the operational data if necessary for a given query.

It is straightforward to show a three-dimensional hypercube in a spreadsheet-type format using columns, rows, and sheets (pages) as the three dimensions. It is possible, however, to show data in more than three dimensions by cascading rows or columns and using drop-down selections to show different slices. Figure 11-14 shows a portion of a report from a Microsoft Excel pivot table with four dimensions, with travel method and number of days in cascading columns. OLAP query and reporting tools usually allow this way to handle sharing dimensions within the limits of two-dimension printing or display space. Data visualization tools, to be shown in the next section, allow using shapes, colors, and other properties of multiples of graphs to include more than three dimensions on the same display.

DATA VISUALIZATION Often the human eye can best discern patterns when data are represented graphically. Data visualization is the representation of data in graphical and multimedia formats for human analysis. Benefits of data visualization include the ability to better observe trends and patterns and to identify correlations and clusters. Data visualization is often used in conjunction with data mining and other analytical techniques.

In essence, data visualization is a way to show multidimensional data not as numbers and text but as graphs. Thus, precise values are often not shown, but rather the intent is

FIGURE 11-14 Sample pivot table with four dimensions: Country (pages), Resort Name (rows), Travel Method, and No. of Days (columns)

Country	(All)

Average of Price	Travel Method	No. of Days													
	Coach				Coach Total	Plane									Plane Total
Resort Name	4	5	7			6	7	8	10	14	16	21	32	60	
Aviemore			135		135										
Barcelona															
Black Forest	69				69										
Cork								269							269
Grand Canyon														1128	1128
Great Barrier Reef													750		750
Lake Geneva								699							699
London															
Los Angeles							295			375					335
Lyon										399					399
Malaga											234				234
Nerja						198				255					226.5
Nice							289								289
Paris–Euro Disney															
Prague		95			95										
Seville									199						199
Skiathos												429			429
Grand Total	69	95	135		99.66666667	198	292	484	199	343	234	429	750	1128	424.5384615

to more readily show relationships between the data. As with OLAP tools, the data for the graphs are computed often from SQL queries against a database (or possibly from data in a spreadsheet). The SQL queries are generated automatically by the OLAP or data visualization software simply from the user indicating what he or she wants to see.

Figure 11-15 shows a simple visualization of sales data using the data visualization tool Tableau. This visualization uses a common technique called small multiples, which places many graphs on one page to support comparison. Each small graph plots metrics of SUM(Total Sales) on the horizontal axis and SUM(Gross Profit) on the vertical axis. There is a separate graph for the dimensions region and year; different market segments are shown via different symbols for the plot points. The user simply drags and drops these metrics

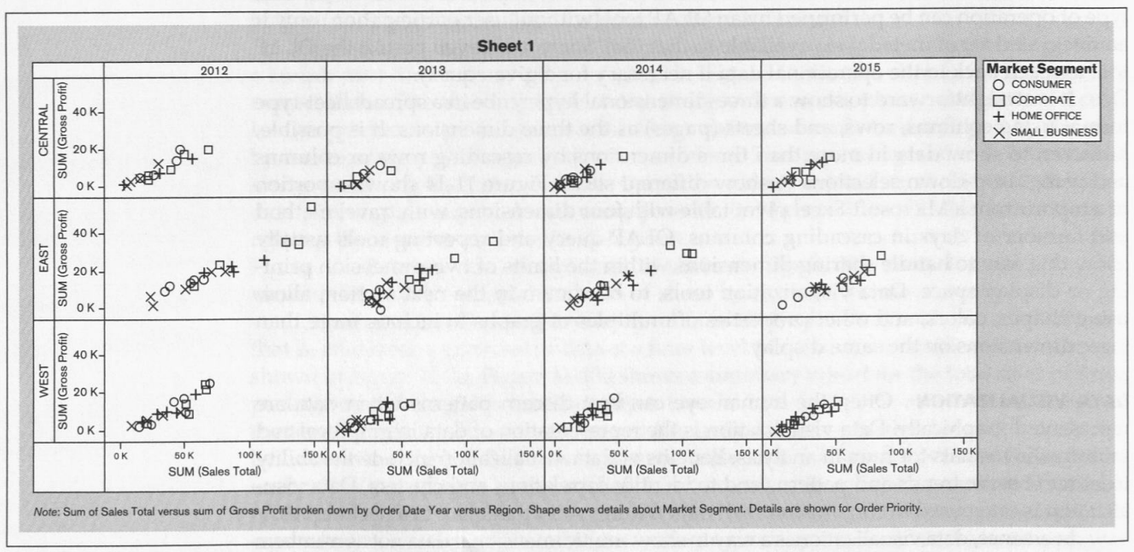


FIGURE 11-15 Sample data visualization with small multiples

and dimensions to a menu and then selects the style of visualization or lets the tool pick what it thinks would be the most illustrative type of graph. The user indicates what he or she wants to see and in what format instead of describing how to retrieve data.

BUSINESS PERFORMANCE MANAGEMENT AND DASHBOARDS A business performance management (BPM) system allows managers to measure, monitor, and manage key activities and processes to achieve organizational goals. Dashboards are often used to provide an information system in support of BPM. Dashboards, just as those in a car or airplane cockpit, include a variety of displays to show different aspects of the organization. Often the top dashboard, an executive dashboard, is based on a balanced scorecard, in which different measures show metrics from different processes and disciplines, such as operations efficiency, financial status, customer service, sales, and human resources. Each display of a dashboard will address different areas in different ways. For example, one display may have alerts about key customers and their purchases. Another display may show key performance indicators for manufacturing, with "stoplight" symbols of red, yellow, and green to indicate if the measures are inside or outside tolerance limits. Each area of the organization may have its own dashboard to determine health of that function. For example, Figure 11-16 is a simple dashboard for one financial measure, revenue. The left panel shows dials about revenue over the past three years, with needles indicating where these measures fall within a desirable range. Other panels show more details to help a manager find the source of out-of-tolerance measures.

Each of the panels is a result of complex queries to a data mart or data warehouse. As a user wants to see more details, there often is a way to click on a graph to get a menu of choices for exploring the details behind the icon or graphic. A panel may be the result of running some predictive model against data in the data warehouse to forecast future conditions (an example of predictive modeling).

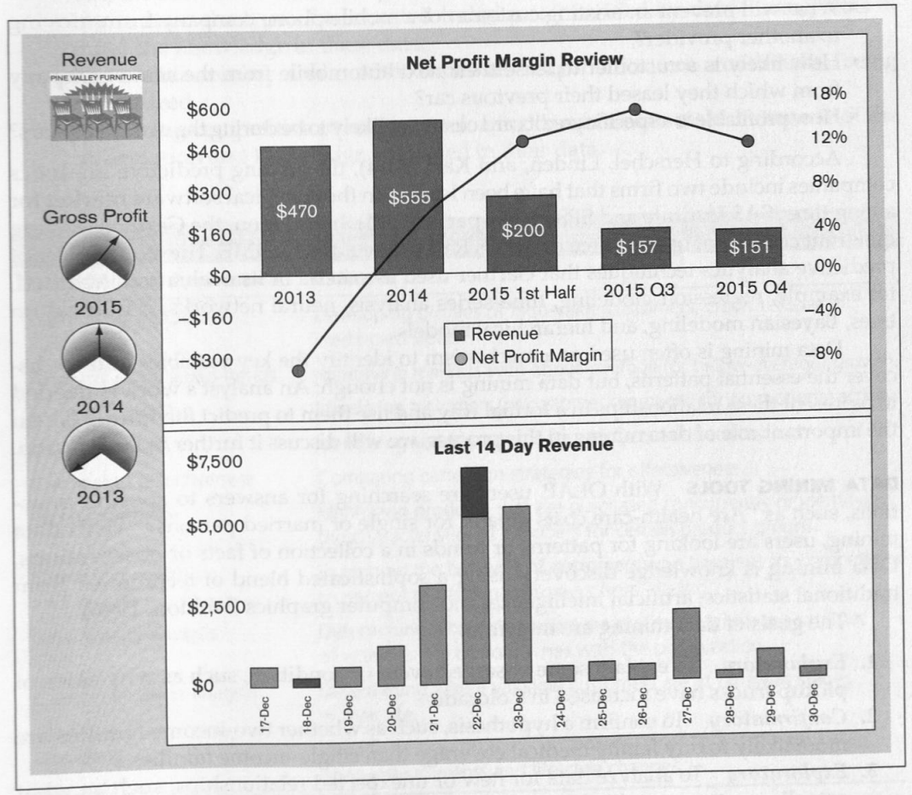


FIGURE 11-16 Sample dashboard

Integrative dashboard displays are possible only when data are consistent across each display, which requires a data warehouse and dependent data marts. Stand-alone dashboards for independent data marts can be developed, but then it is difficult to trace problems between areas (e.g., production bottlenecks due to higher sales than forecast).

Use of Predictive Analytics

If descriptive analytics focuses on the past, the key emphasis of predictive analytics is on the future. Predictive analytics systems use statistical and computational methods that use data regarding past and current events to form models regarding what might happen in the future (potentially depending on a number of assumptions regarding various parameters). The methods for predictive analytics are not new; for example, classification trees, linear and logistic regression analysis, machine learning, and neural networks have existed for quite a while. What has changed recently is the ease with which they can be applied to practical organizational questions and our understanding of the capabilities of various predictive analytics approaches. New approaches are, of course, continuously developed for predictive analytics, such as the golden path analysis for forecasting stakeholder actions based on past behavior (Watson, 2014). Please note that even though predictive analytics focuses on the future, it cannot operate without data regarding the past and the present—predictions have to be built on a firm foundation.

Predictive analytics can be used to improve an organization's understanding of fundamental business questions such as this (adapted from Parr-Rud, 2012):

- What type of an offer will a specific prospective customer need so that she/he will become a new customer?
- What solicitation approaches are most likely to lead to new donations from the patrons of a non-profit organization?
- What approach will increase the probability of a telecommunications company succeeding in making a household switch to their services?
- What will prevent an existing customer of a mobile phone company from moving to another provider?
- How likely is a customer to lease their next automobile from the same company from which they leased their previous car?
- How profitable is a specific credit card customer likely to be during the next five years?

According to Herschel, Linden, and Kart (2014), the leading predictive analytics companies include two firms that have been leaders in the statistical software market for a long time: SAS Institute and SPSS (now part of IBM). In addition, the Gartner leading quadrant consists of open source products RapidMiner and KNIME. The availability of predictive analytics techniques that Gartner used as criteria in its evaluation included, for example, regression modeling, time-series analysis, neural networks, classification trees, Bayesian modeling, and hierarchical models.

Data mining is often used as a mechanism to identify the key variables and to discover the essential patterns, but data mining is not enough: An analyst's work is needed to represent these relationships in a formal way and use them to predict the future. Given the important role of data mining in this process, we will discuss it further in this section.

DATA MINING TOOLS With OLAP, users are searching for answers to specific questions, such as "Are health-care costs greater for single or married persons?" With data mining, users are looking for patterns or trends in a collection of facts or observations. **Data mining** is knowledge discovery using a sophisticated blend of techniques from traditional statistics, artificial intelligence, and computer graphics (Weldon, 1996).

Data mining
Knowledge discovery using a sophisticated blend of techniques from traditional statistics, artificial intelligence, and computer graphics.

The goals of data mining are threefold:

1. ***Explanatory*** To explain some observed event or condition, such as why sales of pickup trucks have increased in Colorado
2. ***Confirmatory*** To confirm a hypothesis, such as whether two-income families are more likely to buy family medical coverage than single-income families
3. ***Exploratory*** To analyze data for new or unexpected relationships, such as what spending patterns are likely to accompany credit card fraud.

TABLE 11-4 Data-Mining Techniques

Technique	Function
Regression	Test or discover relationships from historical data
Decision tree induction	Test or discover if . . . then rules for decision propensity
Clustering and signal processing	Discover subgroups or segments
Affinity	Discover strong mutual relationships
Sequence association	Discover cycles of events and behaviors
Case-based reasoning	Derive rules from real-world case examples
Rule discovery	Search for patterns and correlations in large data sets
Fractals	Compress large databases without losing information
Neural nets	Develop predictive models based on principles modeled after the human brain

Several different techniques are commonly used for data mining. See Table 11-4 for a summary of the most common of these techniques. The choice of an appropriate technique depends on the nature of the data to be analyzed, as well as the size of the data set. Data mining can be performed against all types of data sources in the unified data architecture, including **text mining** of unstructured textual material.

Text mining
The process of discovering meaningful information algorithmically based on computational analysis of unstructured textual information.

Data-mining techniques have been successfully used for a wide range of real-world applications. A summary of some of the typical types of applications, with examples of each type, is presented in Table 11-5. Data-mining applications are growing rapidly, for the following reasons:

- The amount of data in the organizational data sources is growing exponentially. Users need the type of automated techniques provided by data-mining tools to mine the knowledge in these data.
- New data-mining tools with expanded capabilities are continually being introduced.
- Increasing competitive pressures are forcing companies to make better use of the information and knowledge contained in their data.

TABLE 11-5 Typical Data-Mining Applications

Data-Mining Application	Example
Profiling populations	Developing profiles of high-value customers, credit risks, and credit-card fraud.
Analysis of business trends	Identifying markets with above-average (or below-average) growth.
Target marketing	Identifying rustomers (or customer segments) for promotional activity.
Usage analysis	Identifying usage patterns for products and services.
Campaign effectiveness	Comparing campaign strategies for effectiveness.
Product affinity	Identifying products that are purchased concurrently or identifying the characteristics of shoppers for certain product groups.
Customer retention and churn	Examining the behavior of customers who have left for competitors to prevent remaining customers from leaving.
Profitability analysis	Determining which customers are profitable, given the total set of activities the customer has with the organization.
Customer value analysis	Determining where valuable customers are at different stages in their life.
Upselling	Identifying new products or services to sell to a customer based upon critical events and life-style changes.

Source: Based on Dyché (2000).

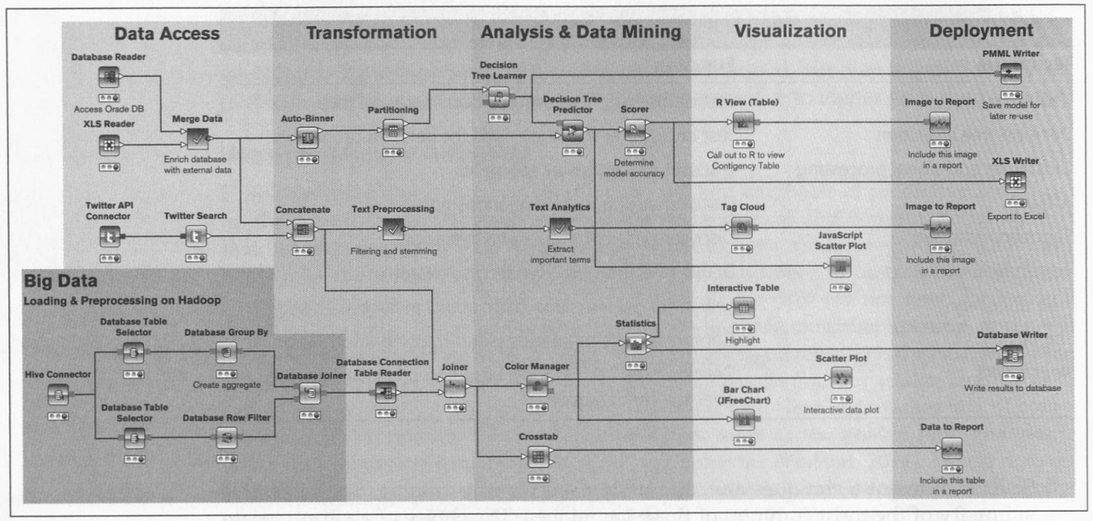


FIGURE 11-17 KNIME architecture

Source: https://www.knime.org/knime. Courtesy of KNIME.

For thorough coverage of data mining and all analytical aspects of business intelligence from a data warehousing perspective, see, for example, Sharda, Delen, and Turban (2013).

EXAMPLES OF PREDICTIVE ANALYTICS Predictive analytics can be used in a variety of ways to analyze past data in order to make predictions regarding the future state of affairs based on mathematical models without direct human role in the process. The underlying models are not new: The core ideas underlying regression analysis, neural networks, and machine learning were developed decades ago, but only the recent tools (such as SAS Enterprise Miner or KNIME) have made them easier to use. Earlier in this chapter we discussed in general terms some of the typical business applications of predictive analytics. In this section we will present some additional examples at a more detailed level.

KNIME's (see Figure 11-17) illustration of use cases includes a wide variety of examples from marketing to finance. In the latter area, credit scoring is a process that takes past financial data at the individual level and develops a model that gives every individual a score describing the probability of a default for that individual. In a KNIME example (**https://www.knime.org/knime-applications/credit-scoring**), the workflow includes three separate methods (decision tree, neural network, and machine learning algorithm called SVM) for developing the initial model. As the second step, the system selects the best model by accuracy and finally writes the best model out in the Predictive Model Markup Language (PMML). PMML is a de facto standard for representing a collection of modeling techniques that together form the foundation for predictive modeling. In addition to modeling, PMML can also be used to specify the transformations that the data has to go through before it is ready to be modeled, demonstrating again the strong linkage between data management and analytics.

In marketing, a frequently used example is the identification of those customers that are predicted to leave the company and go elsewhere (churn). The KNIME example (**https://www.knime.org/knime-applications/churn-analysis**) uses an algorithm called k-Means to divide the cases into clusters, in this case predicting whether or not a particular customer will be likely to leave the company. Finally, KNIME also includes a social media data analysis example (**https://www.knime.org/knime-applications/lastfm-recommodation**), demonstrating how association analysis can

be used to identify the performers to whom those listening to a specific artist are also likely to listen.

In addition to the business examples, the KNIME case descriptions illustrate how close the linkage between data management and analytics is in the context of an advanced analytics platform. For example, the churn analysis example includes the use of modules such as XLS Reader, Column Filter, XLS Writer, and Data to Report to get the job done. The social media example utilizes File Reader, Joiner, GroupBy, and Data to Report. Even if you do not know these modules, it is likely that they look familiar as operations, and the names directly refer to operations with data.

Use of Prescriptive Analytics

If the key question in descriptive analytics is "What happened?" and in predictive analytics is "What will happen?" then prescriptive analytics focuses on the question "How can we make it happen?" or "What do we need to do to make it happen?" For prescriptive analysis we need optimization and simulation tools and advanced modeling to understand the dependencies between various actors within the domain of interest. In many contexts, the results of prescriptive analytics are automatically moved to business decision making. For example:

- Automated algorithms make millions or billions of trading decisions daily, buying and selling securities in markets where human actions are far too slow.
- Airlines and hotels are pricing their products automatically using sophisticated algorithms to maximize revenue that can be extracted from these perishable resources.
- Companies like Amazon and Netflix are providing automated product recommendations based on a number of factors, including their customers' prior purchase history and the behavior of the people with whom they are connected.

The tools for prescriptive analytics are less structured and packaged than those for descriptive and predictive analytics. Many of the most sophisticated tools are internally developed. The leading vendors of predictive analytics products do, however, also include modules for enabling prescriptive analytics.

Wu (as cited in Bertolucci, 2013) describes prescriptive analytics as a type of predictive analytics, and this is, indeed, a helpful way to look at the relationship of the two. Without the modeling characteristic of predictive analytics, systems for prescriptive analytics could not perform their task of prescribing an action based on past data. Further, prescriptive analytics systems typically collect data regarding the impact of the action taken so that the models can be further improved in the future. As we discussed in the introduction, prescriptive analytics provides model-based views regarding the impact of various actions on business performance (Underwood, 2013) and often make automated decisions based on the predictive models.

Prescriptive analytics is not new, either, because various technologies have been used for a long time to make automated business decisions based on past data. What has changed recently, however, is the sophistication of the models that support these decisions and the level of granularity related to the decision processes. For example, service businesses can make decisions regarding price/product feature combinations not only at the level of large customer groups (such as business vs. leisure) but at the level of an individual traveler so that recommendation systems can configure individual service offerings addressing a traveler's key needs while keeping the price at a level that is still possible for the traveler (Braun, 2013).

Implementing prescriptive analytics solutions typically requires integration of analytics software from third parties and an organization's operational information systems solutions (whether ERPs, other packaged solutions, or systems specifically developed for the organization). Therefore, there are many fewer analytics packages labeled specifically as "prescriptive analytics" than there are those for descriptive or predictive analytics. Instead, the development of prescriptive analytics solutions requires more sophisticated integration skills, and these solutions often provide more distinctive business value because they are tailored to a specific organization's needs.

Hernandez and Morgan (2014) discuss the reasons underlying the complexity of the systems for prescriptive analytics. Not only do these systems require sophisticated predictive modeling of organizational and external data, they also require in-depth understanding of the processes required for optimal business decisions in a specific context. This is not a simple undertaking; it requires the identification of the potential decisions that need to be made, the interconnections and dependencies between these decisions, and the factors that affect the outcomes of these decisions. In addition to the statistical analysis methods common in predictive analytics, prescriptive analytics relies on advanced simulations, optimization processes, decision-analysis methods, and game theory. This all needs to take place real-time with feedback loops that will analyze the successfulness of each decision/recommendation the system has made and use this information to improve the decision algorithms.

Data Management Infrastructure for Analytics

In this section, we will review the technical infrastructure that is required for enabling the big data approach specified earlier and other forms of data sources for advanced analytics. We will not focus on the analytical processes themselves—other textbooks such as *Business Intelligence: A Managerial Perspective on Analytics* (Sharda, Delen, and Turban, 2013) and *Business Intelligence and Analytics: Systems for Decision Support* (Sharda, Delen, and Turban, 2014) are good sources for those readers interested in the approaches and techniques of analytics. In this text, we will emphasize the foundational technologies that are needed to enable big data and advanced analytics in general, that is, the infrastructure for big data and advanced analytics.

Schoenborn (2014) identified four specific infrastructure capabilities that are required for big data and advanced analytics. They are as follows:

- *Scalability*, which refers to the organization's planned ability to add capacity (processing resources, storage space, and connectivity) based on changes in demand. Highly scalable infrastructure allows an organization to respond to increasing demand quickly without long lead times or huge individual investments.
- *Parallelism*, which is a widely used design principle in modern computing systems. Parallel systems are capable of processing, transferring, and accessing data in multiple chunks at the same time. We will later discuss a particular implementation model of parallelism called massively parallel processing (MPP) systems, which is commonly used, among other contexts, in large data centers run by companies such as Google, Amazon, Facebook, and Yahoo!.
- *Low latency* of various technical components of the system. Low latency refers, in practice, to a high speed in various processing and data access and writing tasks. When designing high-capacity infrastructure systems, it is essential that the components of these systems add as little latency as possible to the system.
- *Data optimization*, which refers to the skills needed to design optimal storage and processing structures.

According to Schoenborn (2014), there are three major infrastructure characteristics that can be measured. All of these are enabled by the capabilities previously discussed: speed, availability, and access.

- *Speed* tells how many units of action (such as certain processing or data access task) the system is able to perform in a time unit (such as a second).
- *Availability* describes how well the system stays available in case of component failure(s). A highly available system can withstand failures of multiple components, such as processor cores or disk drives.
- *Access* illustrates who will have access to the capabilities offered by the system and how this access is implemented. A well-designed architecture provides easy access to all stakeholders based on their needs.

Four specific technology solutions are used in modern data storage systems that enable advanced analytics and allow systems to achieve the infrastructure capabilities previously described (see, e.g., Watson, 2014). These include massively parallel

TABLE 11-6 Technologies Enabling Infrastructure Advances in Data Management

Massively parallel processing (MPP)	Instead of relying on a single processor, MPP divides a computing task (such as query processing) between multiple processors, speeding it up significantly.
In-memory DBMSs	In-memory DBMSs keep the entire database in primary memory, thus enabling significantly faster processing.
In-database analytics	If analytical functions are integrated directly to the DBMS, there is no need to move large quantities of data to separate analytics tools for processing.
Columnar DBMSs	They reorient the data in the storage structures, leading to efficiencies in many data warehousing and other analytics applications.

processing (MPP), in-memory database management systems, in-database analytics, and columnar databases (see summary in Table 11-6).

Massively parallel processing (MPP) is one of the key advances not only in data storage but in computing technologies in general. The principle is simple: A complex and time-consuming computing task will be divided into multiple tasks that are executed simultaneously to increase the speed at which the system achieves the result. Instead of having one unit of computing power (such as a processor) to perform a specific task, the system will be able to use dozens or thousands of units at the same time. In practice, this is a very complex challenge and requires careful design of the computing tasks. Large Web-based service providers have made significant advances in understanding how massively parallel systems can be developed using very large numbers of commodity hardware, that is, standardized, inexpensive processing and storage units.

In-memory database management systems are also based on a simple concept: storing the database(s) in the database server's random access memory instead of storing them on a hard disk or another secondary storage device, such as flash memory. Modern server computers used as database servers can have several terabytes of RAM, an amount that just a few years ago was large even for a disk drive capacity. The primary benefit of storing the databases in-memory (and not just temporarily caching data in-memory) is improved performance specifically with random access: Jacobs (2009) demonstrates how random access reads with a solid-state disk (SSD) are about six times faster than with a mechanical disk drive but random in-memory access is 20,000 times faster than SSD access.

In-database analytics is an interesting architectural development, which is based on the idea of integrating the software that enables analytical work directly with the database management system software. This will make it possible to do the analytical work directly in the database instead of extracting the required data first onto a separate server for analytics. This reduces the number of stages in the process, thus improving overall performance, ensuring that all data available in the database will be available for analysis, and helping to avoid errors. This approach also makes it possible to integrate analytical tools with traditional data retrieval languages (such as SQL).

Columnar or column-oriented database management systems are using an approach to storing data that differs significantly from traditional row-oriented relational database management systems. Traditional RDBM technology is built around the standard relational data model of tables of rows and columns and physical structures that store data as files of records for rows, with columns as fields in each record. This approach has served the needs of RDBMs used for transaction processing and simple management reporting well. Complex analytics with very large and versatile data sets, however, can benefit from a different storage structure for data, one where data are stored on a column basis instead of on a row basis. That is, values are stored in sequence for one column, followed by the values for another column, and so on, thus virtually turning a table of data 90 degrees.

Vendors of column-based products claim to reduce storage space (because data compression techniques are used, for example, to store a value only once) and to speed query processing time because the data are physically organized to support analytical queries. Column database technologies trade off storage space savings (data compression of more

than 70 percent is common) for computing time. The conceptual and logical data models for the data warehouse do not change. SQL is still the query language, and you do not write queries any differently; the DBMS simply stores and accesses the data differently than in traditional row-oriented RDBMSs. Data compression and storage depend on the data and queries. For example, with Vertica (a division of HP), one of the leading column database management system providers, the logical relational database is defined in SQL as with any RDBMS. Next, a set of sample queries and data are presented to a database design tool. This tool analyzes the predicates (WHERE clauses) of the queries and the redundancy in the sample data to suggest a data compression scheme and storage of columnar data. Different data compression techniques are used depending on the type of predicate data (numeric, textual, limited versus a wide range of values, etc.).

We will not cover in this text the implementation details of these advances in data management technologies because they are primarily related to the internal technology design and not the design of the database. It is, however, essential that you understand the impact these technological innovations potentially have on the use of data through faster access, better integration of data management and analytics, improved balance between the use of in-memory and on-disk storage, and innovative storage structures that improve performance and reduce storage costs. These new structural and architectural innovations are significantly broadening the options companies have available for implementing technical solutions for making data available for analytics.

IMPACT OF BIG DATA AND ANALYTICS

In this final section of Chapter 11, we will discuss a number of important issues related to the impact of big data, primarily from two perspectives: applications and implications of big data analytics. In the applications section, we will focus on the areas of human activity most affected by the new opportunities created by big data and illustrate some of the ways in which big data analytics has transformed business, government, and not-for-profit organizations.

Applications of Big Data and Analytics

The following categorization of areas of human activity affected by big data analytics is adapted and extended from Chen et al. (2012):

1. Business (originally e-commerce and market intelligence),
2. E-government and politics,
3. Science and technology,
4. Smart health and well-being, and
5. Security and public safety.

The ways in which human activities are conducted and organized in all of these areas have already changed quite significantly because of analytics, and there is potential for much more significant transformation in the future. There are, of course, other areas of human activity that could also have been added to this list, including, for example, arts and entertainment.

From the perspective of the core focus area of this textbook, one of the key lessons to remember is that all of these exciting and very significant changes in important areas of life are only possible if the collection, organizing, quality control, and analysis of data are implemented systematically and with a strong focus on quality. Some of the discussions in the popular press and the marketing materials of the vendor may create the impression that truly insightful results emerge from various systems without human intervention. This is a dangerous fallacy, and it is essential that you as a professional with an in-depth understanding of data management are well-informed of what is needed to enable the truly amazing new applications based on big data analytics and decision making based on it. Sometimes you will need to do a lot of work to educate your colleagues and customers of what is needed to deliver the true benefits of analytics (Lohr, 2014).

Another major lesson to take away from this section is the breadth of current and potential applications of big data analytics. The applications of analytics are not limited

to business but extend to a wide range of essential human activities, all of which are going through major changes because of advanced analytics capabilities. These changes are not limited to specific geographic regions, either—applications of analytics will have an impact on countries and areas regardless of their location or economic development status. For example, the relative importance of mobile communication technologies is particularly high in developing countries, and many of the advanced applications of analytics are utilizing data collected by mobile systems.

We will next briefly discuss the areas that big data analytics is changing and the changes it is introducing.

BUSINESS In business, advanced uses of analytics have the potential to change the relationship between a business and its individual customers dramatically. As already discussed in the context of prescriptive analytics, analytics allows businesses to tailor both products and pricing to the needs of an individual customer, leading to something economists call *first degree* or *complete price discrimination*, that is, extracting from each customer the maximum price they are willing to pay. This is, of course, not beneficial from the customers' perspective. At the same time, customers do benefit from the ability to receive goods and services that are tailored to their specific needs.

Some of the best-known examples of the use of big data analytics in business are related to the targeting of marketing communication to specific customers. The often-told true story (Duhigg, 2012) about how a large U.S. retail chain started to send a female teenager pregnancy-related advertisements, leading to complaints by an irritated father, is a great example of the power and the dangers of the use of analytics. To make a long story short, the father, became even more irritated after finding out that the retail chain had learned about his daughter's pregnancy earlier than he had by using product and other Web search data. Big data analytics gives companies outstanding opportunities to learn a lot about their current and prospective customers. At the same time, it creates a major responsibility for them to understand the appropriate uses of these technologies.

Businesses are also learning a lot about and from their customers by analyzing data that they collect from Web- and mobile-based interactions between them and their customers and social media data. Customers leave a lot of clues about their characteristics and preferences through the actions that they take when navigating a company's Web site, performing searches with external search engines, or making comments regarding the company's products or services on various social media platforms. Many companies are particularly attentive to communication on social media because of the public nature of the communication. Sometimes a complaint on Twitter will lead to a faster response time than using e-mail for the same purpose.

E-GOVERNMENT AND POLITICS Analytics has also had a significant impact on politics, particularly in terms of how politicians interact with their constituents and how political campaigns are conducted. Again, social media platforms are important sources of data for understanding public opinion regarding general issues and specific positions taken by a politician, and social media forms important communication channels and platforms for interactions between politicians and their stakeholders (Wattal et al., 2010).

One important perspective on analytics in the context of government is that of the role of government as a data source. Governments all over the world both collect huge amounts of data through their own actions and fund the collection of research data. Providing access to the government-owned data through well-defined open interfaces has led to significant opportunities to provide useful new services or create insights regarding possible actions by local governments. Some of these success stories are told on the Web site of The Open Data Institute (**theodi.org/stories**), co-founded by World Wide Web inventor Sir Tim Berners-Lee and others at **opendatastories.org**.

In addition, it was at least originally hoped that data openness would be associated with gains in values associated with democracy (improved transparency of public actions, opportunities for more involved citizen participation, improved ability to evaluate elected officials, etc.); it is not clear whether or not these advances can truly materialize (Chignard, 2013).

SCIENCE AND TECHNOLOGY Big data analytics has already greatly benefited a wide variety of scientific disciplines, from astrophysics to genomics to many of the social sciences. As described in Chen et al. (2012, p. 1170), the U.S. National Science foundation described some of the potential scientific benefits of big data:

> "to accelerate the progress of scientific discovery and innovation; lead to new fields of inquiry that would not otherwise be possible; [encourage] the development of new data analytic tools and algorithms; facilitate scalable, accessible, and sustainable data infrastructure; increase understanding of human and social processes and interactions; and promote economic growth and improved health and quality of life."

We can expect significant advances in a number of scientific disciplines from big data analytics. One of the interesting issues that connects the role of government and the practice of science in the area of big data is the ownership and availability of research data to the broader scientific community. When research is funded by a government agency (such as the U.S. National Science Foundation or National Institutes of Health), should the raw data from that research be made available freely? What rules should govern the access to such data? How should it be organized and secured? These are significant questions that are not easy to answer, and any answers found need significant sophistication in data management before they can be implemented.

SMART HEALTH AND WELL-BEING The opportunities to collect personal health and well-being–related data and benefit from it are increasing dramatically. Not only are typical formal medical tests producing much more data than used to be the case, there are also new types of sources of large amounts of personal medical data (such as mapping of an individual's entire genome, which is soon going to cost a few hundred dollars or less). Furthermore, there are opportunities to integrate large amounts of individual data collected by, for example, insurance companies or national medical systems (where they exist). This data from a variety of sources can, in turn, be used for a variety of research purposes. Individuals are also using a variety of devices to collect and store personal health and well-being data using devices that they are wearing (such as Fitbit, Jawbone, or Nike FuelBand).

Chen et al. (2012) discuss the ways in which all this health and well-being–related data could be used to transform the entire concept of medicine from disease control to an evidence-based, individually focused preventive process with the main goal of maintaining health. Health and wellness is an area that will test the capabilities of the data collection and management infrastructure for analytics in a number of ways, given the continuous nature of the data collection and the highly private nature of the data.

SECURITY AND PUBLIC SAFETY Around the world, concerns regarding security, safety, and fraud have led to the interest in applying big data analytics to the processes of identifying potential security risks in advance and reacting to them before the risks materialize. The methods and capabilities related to the storage and processing of large amounts of data real-time are applicable to fraud detection, screening of individuals of interest from large groups, identifying potential cybersecurity attacks, understanding the behavior of criminal and terrorist networks, and many other similar purposes.

Concerns of security and privacy are particularly important in this area because of the high human cost of false identification of individuals as security risks and the fundamentally important need to maintain a proper balance between security and individual rights. The conversation regarding the appropriate role of government agencies started by the revelations made by Edward Snowden in 2013 and 2014 (Greenwald et al., 2013) has brought many essential questions regarding individual privacy to the forefront of public debate, at the minimum pointing out the importance of making informed decisions regarding the collection of private data by public entities.

Implications of Big Data Analytics and Decision Making

As already tentatively discussed, big data analytics raises a number of important questions regarding possible negative implications. As with any other new technology, it is important that decision makers and experts at various levels have a clear understanding of the possible implications of their choices and actions. Many of the opportunities created by big data analytics are genuinely transformative, but the benefits have to be evaluated in the context of the potentially harmful consequences.

In January 2014, a workshop funded by the U.S. National Science Foundation (Markus, 2014) brought together a large number of experts on big data analytics and decision making to identify the key implications of the technical developments related to big data. This section is built on the key themes that emerged from the workshop conversations.[1]

PERSONAL PRIVACY VS. COLLECTIVE BENEFITS Personal privacy is probably the most commonly cited concern in various conversations regarding the implications of big data analytics. What mechanisms should be in place to make sure that individual citizens can control the data that various third parties—including businesses, government agencies, and non-profit organizations—maintain about them? What control should an individual customer have over the profiles various companies build about them in order to target marketing communication better? What rights should individuals have to demand that data collected regarding them is protected, corrected, or deleted if they so desire? Should medical providers be allowed to collect detailed personal data in order to advance science and medial practice? How about the role of government agencies—how much should they be allowed to know about a random individual in order to protect national security? Many of these questions are, in practice, about the relationship between personal right to privacy vs. the collective benefits we can gain if detailed individual data are collected and maintained.

The legal and ethical issues that need to be considered in this context are complex and multiple, but no organization or individual manager or designer dealing with large amounts of individual data can ignore them. Legal codes and practices vary across the world, and it is essential that privacy and security questions are carefully built into any process of designing systems that utilize big data.

OWNERSHIP AND ACCESS Another complex set of questions is related to the ownership of the large collections of data that various organizations put together to gain the benefits previously discussed. What rights should individuals have to data that has been collected about them? Should they have the right to benefit financially about the data they are providing through their actions? Many of the free Web-based services are, in practice, not free: Individuals get access to the services by giving up some of their rights to privacy.

In this category is also the question about ownership of research data, particularly in the context of research projects funded by various government agencies. If research is taxpayer funded, should the data collected in that research be made available to all interested parties? If yes, how is individual privacy of research participants protected?

QUALITY AND REUSE OF DATA AND ALGORITHMS The fact that big data analytics is based on large amounts of data does not mean that data quality (discussed at a more detailed level in Chapter 10) is any less important. On the contrary, high volumes of poor quality data arriving at high speeds can lead to particularly bad analytical results. Some of the data quality questions related to big data are exactly the same that data management professionals have struggled with for a long time: missing data, incorrect coding, replicated data, missing specifications, etc. Others are specific to the new context particularly because particularly NoSQL-based systems often are not based on careful conceptual and logical modeling (in some contexts by definition).

In addition, often big data systems reuse data and algorithms for purposes for which they were not originally developed. In these situations, it is essential that reuse does not become misuse because of, for example, poor fit between the new purpose and

[1] Acknowledgement: The material in this section is partially based upon work supported by the National Science Foundation under Grant No. 1348929. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

the data that was originally collected for something else or algorithms that work well for one purpose but are not a good fit with a slightly different situation.

TRANSPARENCY AND VALIDATION One of the challenges with big data in both business and science is that in many cases it is impossible for anybody else but the party doing the analysis to verify if the analysis is correctly performed or if the data that was used is correct. For example, automated credit rating systems can have a major impact on an individual's ability to get a car loan or a mortgage and also on the interest cost of the borrowed funds. Even if the outcome is negative, it is very difficult for an individual to get access to the specific data that led to the decision and to verify its correctness.

The need for validation and transparency becomes particularly important in the case of entirely automated prescriptive analytics, for example, in the form of automated trading of securities or underwriting of insurance policies. If there is no human intervention, there should be at least processes in place that make a continuous review of the actions taken by the automated system.

CHANGING NATURE OF WORK Big data analytics will also have an impact on the nature of work. In the same way many jobs requiring manual labor have changed significantly because of robotics, many knowledge work opportunities will be transformed because sophisticated analytical systems will assume at least some of the responsibilities that earlier required expert training and long experience. For example, Frey and Osborne (2013) evaluated the future of employment based on computerization and based on a sophisticated mathematical model calculated probabilities for specific occupations to be significantly impacted by computer-based technologies (in the context of knowledge work, analytics). Their list of occupations suggests that many knowledge work professions will disappear in the relatively near future because of advances in computing.

DEMANDS FOR WORKFORCE CAPABILITIES AND EDUCATION Finally, big data analytics has already changed the requirements for the capabilities knowledge workers are expected to have and, consequently, for the education that a knowledge professional should have. It will not be long until any professional will be expected to use a wide variety of analytical tools to understand not only numeric data but also textual and multimedia data collected from a variety of sources. This will not be possible without a conscious effort to prepare analysts for these tasks.

For information systems professionals, the additional challenge is that the concept of data management has broadened significantly from the management of well-designed structured data in relational databases and building systems on the top of those. Data management brings together and is required to take care of a wide variety of data from a rich set of internal and external sources, ensuring the quality and security of those resources, and organizing the data so that they are available for the analysts to use.

Summary

The landscape of data management is changing rapidly because of the requirements and opportunities created by new analytics capabilities. In addition to its traditional responsibilities related to managing traditional organizational data resources primarily stored in relational databases, enterprise-wide data warehouses, and data marts, organizational data and information management function now has additional responsibilities. It is responsible for overseeing and partially implementing the processes related to bringing together semi- and unstructured data of various types from many external and internal sources, managing its quality and security, and making it available for a rich set of analytical tools.

Big data has created more excitement and sense of new opportunities than any other organizational data and information-related concept for a decade; therefore this umbrella concept referring to the collection, storage, management, and analysis of very large amounts of heterogeneous data that arrives at very high speeds is an essential area of study. For the purposes of organizational data and information management, the key questions are related to the specific requirements that big data sets for the tools and infrastructure. Two key new technology categories are data management environments under the title NoSQL (Not only SQL) and the massively parallel open source platform Hadoop. Both NoSQL technologies and Hadoop have given organizations tools for storing and analyzing very large amounts of data at unit costs that were not possible earlier. In many cases, NoSQL and Hadoop-based solutions do not have predefined schemas. Instead of the traditional *schema on write* approach, the structures are specified (or even discovered) at the time when the data are explored and analyzed (*schema on read*). Both NoSQL technologies and Hadoop provide important new capabilities for organizational data management.

Hadoop and NoSQL technologies are not, however, useful alone and, in practice, they are used as part of larger organization-wide platforms of data management technologies, which bring together tools for collecting, storing, managing, and analyzing massive amounts of data. Many vendors have introduced their own comprehensive conceptual architectures for bringing together the various tools, such as Teradata's Unified Data Architecture.

Given its recent surge in the ranks of organizational buzzwords, the world of analytics is currently in turmoil, suffering from inconsistent use of concepts. Dividing analytics into descriptive, predictive, and prescriptive variants provides useful structure to this evolving area. In addition, it also helps to categorize analytics based on the types of sources from which data are retrieved to the organizational analytics systems: traditional administrative systems, the Web, and ubiquitous mobile systems.

Big data and other advanced analytics approaches create truly exciting new opportunities for business, government, civic engagement, not-for-profit organizations, various scientific disciplines, engineering, personal health and well-being, and security. The advantages are not, however, without their potential downsides. Therefore, it is important that decision makers and professionals working with and depending on analytics solutions understand the implications of big data related to personal privacy, data ownership and access, quality and reuse of data and algorithms, openness of the solutions based on big data, changing nature of work, and the requirements for education and workforce capabilities.

Chapter Review

Key Terms

Analytics *445*
Big data *445*
Business intelligence *460*
Data lake *448*
Data mining *470*
Descriptive analytics *461*
Hadoop *453*
HDFS *454*
Hive *456*
MapReduce *453*
Multidimensional OLAP (MOLAP) *466*
NoSQL *449*
Online analytical processing (OLAP) *465*
Pig *456*
Predictive analytics *461*
Prescriptive analytics *461*
Relational OLAP (ROLAP) *465*
Text mining *471*

Review Questions

11-1. Define each of the following terms:
a. Hadoop
b. MapReduce
c. HDFS
d. NoSQL
e. Pig
f. data mining
g. online analytical processing
h. business intelligence

11-2. Match the following terms to the appropriate definitions:

_____ Hive	a. knowledge discovery using a variety of statistical and computational techniques
_____ text mining	b. analytics that suggests mechanisms for achieving desired outcomes
_____ data lake	c. tool that provides an SQL-like interface for managing data in Hadoop
_____ data mining	d. converting textual data into useful information
_____ descriptive analytics	e. form of analytics that forecasts future based on past and current events
_____ predictive analytics	f. a large, unstructured collection of data from both internal and external sources
_____ prescriptive analytics	g. systematic analysis and interpretation of data to improve our understanding of a real-world domain
	h. a form of analytics that provides reports regarding past events

11-3. Contrast the following terms:
a. Data mining; text mining
b. Pig; Hive
c. ROLAP; MOLAP
d. NoSQL; SQL
e. Data lake; data warehouse

11-4. Identify and briefly describe the five Vs that are often used to define big data.

11-5. What are the two challenges faced in visualizing big data?

11-6. List the differences between the two categories of technology, Hadoop and NoSQL , which have become core infrastructure elements of big data solutions.

11-7. What is the difference between explanatory and exploratory goals of data mining?

11-8. What is the trade-off one needs to consider in using a NoSQL database management system?