Chapter 2

Examining Big Data Types

In This Chapter

- Identifying structured and unstructured data
- Recognizing real-time and non-real-time requirements for data types
- Integrating data types into a big data environment

Variety is the spice of life, and variety is one of the principles of big data. In Chapter 1, we discuss the importance of being able to manage the variety of data types. Clearly, big data encompasses everything from dollar transactions to tweets to images to audio. Therefore, taking advantage of big data requires that all this information be integrated for analysis and data management. Doing this type of activity is harder than it sounds. In this chapter, we examine the two main types of data that make up big data — structured and unstructured — and provide you with definitions and examples of each.

Although data management has been around for a long time, two factors are new in the big data world:

- Some sources of big data are actually new like the data generated from sensors, smartphone, and tablets.
- Previously produced data hadn't been captured or stored and analyzed in a usable way. The main reason for this is that the technology wasn't there to do so. In other words, we didn't have a cost-effective way to deal with all that data.

You have many different ways to put big data to use to solve problems. For example, in some situations, you want to deal with data in real time, such as when you're monitoring traffic data. In other situations, real-time data management won't be necessary, such as when you're collecting massive amounts of data that you want to analyze in batch mode to determine an unsuspected pattern. Likewise, you sometimes need to integrate multiple sources of data as part of a big data solution, so we look at why you might want to integrate data sources. The bottom line is that what you want to do with your structured and unstructured data informs the technology purchases that you make.

Defining Structured Data

The term *structured data* generally refers to data that has a defined length and format. Examples of structured data include numbers, dates, and groups of words and numbers called *strings* (for example, a customer's name, address, and so on). Most experts agree that this kind of data accounts for about 20 percent of the data that is out there. Structured data is the data that you're probably used to dealing with. It's usually stored in a database. You can query it using a language like structured query language (SQL), which we discuss later in the "Defining Unstructured Data" section.

Your company may already be collecting structured data from "traditional" sources. These might include your customer relationship management (CRM) data, operational enterprise resource planning (ERP) data, and financial data. Often these data elements are integrated in a data warehouse for analysis.

Exploring sources of big structured data

Although this might seem like business as usual, in reality, structured data is taking on a new role in the world of big data. The evolution of technology provides newer sources of structured data being produced — often in real time and in large volumes. The sources of data are divided into two categories:

- **Computer- or machine-generated:** Machine-generated data generally refers to data that is created by a machine without human intervention.
- **Human-generated:** This is data that humans, in interaction with computers, supply.

Some experts argue that a third category exists that is a hybrid between machine and human. Here though, we're concerned with the first two categories.

Machine-generated structured data can include the following:

- **Sensor data:** Examples include radio frequency ID (RFID) tags, smart meters, medical devices, and Global Positioning System (GPS) data. For example, RFID is rapidly becoming a popular technology. It uses tiny computer chips to track items at a distance. An example of this is tracking containers of produce from one location to another. When information is transmitted from the receiver, it can go into a server and then be analyzed. Companies are interested in this for supply chain management and inventory control. Another example of sensor data is smartphones that contain sensors like GPS that can be used to understand customer behavior in new ways.

- **Web log data:** When servers, applications, networks, and so on operate, they capture all kinds of data about their activity. This can amount to huge volumes of data that can be useful, for example, to deal with service-level agreements or to predict security breaches.
- **Point-of-sale data:** When the cashier swipes the bar code of any product that you are purchasing, all that data associated with the product is generated. Just think of all the products across all the people who purchase them, and you can understand how big this data set can be.
- **Financial data:** Lots of financial systems are now programmatic; they are operated based on predefined rules that automate processes. Stock-trading data is a good example of this. It contains structured data such as the company symbol and dollar value. Some of this data is machine generated, and some is human generated.

Examples of structured human-generated data might include the following:

- **Input data:** This is any piece of data that a human might input into a computer, such as name, age, income, non-free-form survey responses, and so on. This data can be useful to understand basic customer behavior.
- **Click-stream data:** Data is generated every time you click a link on a website. This data can be analyzed to determine customer behavior and buying patterns.
- **Gaming-related data:** Every move you make in a game can be recorded. This can be useful in understanding how end users move through a gaming portfolio.

You get the idea. Some of this data may not be that big on its own, such as profile data. However, when taken together with millions of other users submitting the same information, the size is astronomical. Additionally, much of this data has a real-time component to it that can be useful for understanding patterns that have the potential of predicting outcomes. The bottom line is that this kind of information can be powerful and can be utilized for many purposes.

Understanding the role of relational databases in big data

Data persistence refers to how a database retains versions of itself when modified. The great granddaddy of persistent data stores is the *relational database management system (RDBMS).* In its infancy, the computing industry used what are now considered primitive techniques for data persistence.

You may recall "flat files" or "network" data stores that were prevalent before 1980 or so. Although these mechanisms were useful, they were very difficult to master and always required system programmers to write custom programs to manipulate the data.

The relational model was invented by Edgar Codd, an IBM scientist, in the 1970s and was used by IBM, Oracle, Microsoft, and others. It is still in wide usage today and plays an important role in the evolution of big data. Understanding the relational database is important because other types of databases are used with big data. We contrast various kinds of databases used for big data throughout this book.

In a relational model, the data is stored in a table. This database would contain a *schema* — that is, a structural representation of what is in the database. For example, in a relational database, the schema defines the tables, the fields in the tables, and the relationships between the two. The data is stored in columns, one each for each specific attribute. The data is also stored in the rows. For instance, the two tables shown in Figure 2-1 represent the schema for a simple database. The first table stores product information; the second stores demographic information. Each has various attributes (customer ID, order number, purchase code for a product, and so on). Each table can be updated with new data, and data can be deleted, read, and updated. This is often accomplished in a relational model using a structured query language (SQL).

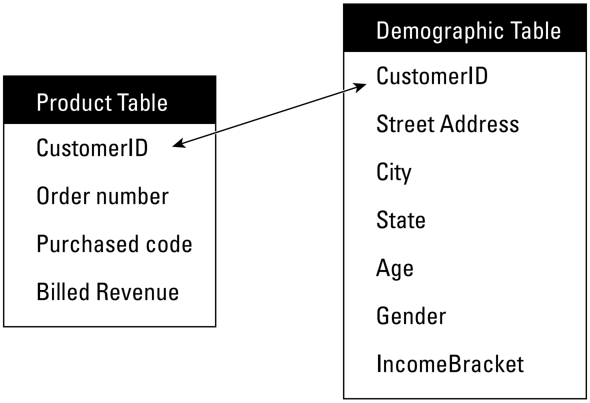

Figure 2-1: The relationships between tables.

Another aspect of the relational model using SQL is that tables can be queried using a common key (that is, the relationship). In Figure 2-1, the common key in the tables is CustomerID.

You can submit a query, for example, to determine the gender of customers who purchased a specific product. It might look something like this:

```
Select CustomerID, State, Gender, Product from
           "demographic table", "product table" where
           Product= XXYY
```

Although relational databases have ruled the roost for the last several decades, they can be difficult to use when you're dealing with huge streams of disparate data types. Relational database vendors are not standing still, however, and are starting to introduce relational databases designed for big data. In addition, new database models have evolved to help people manage big data. We talk a little bit about technologies like NoSQL, streaming databases, and others in Chapter 1. These data management systems are a subject unto themselves, so we devote all of Part III to them.

TIP

PostgresSQL (`www.postgressql.org`), a technology we talk about in Chapter 7, is the most widely used open source relational database available. Its extensibility and the fact that it is available on many varieties of mainframes make it a foundation technology for some relational big data databases.

Defining Unstructured Data

Unstructured data is data that does not follow a specified format. If 20 percent of the data available to enterprises is structured data, the other 80 percent is unstructured. Unstructured data is really most of the data that you will encounter. Until recently, however, the technology didn't really support doing much with it except storing it or analyzing it manually.

Exploring sources of unstructured data

Unstructured data is everywhere. In fact, most individuals and organizations conduct their lives around unstructured data. Just as with structured data, unstructured data is either machine generated or human generated.

Here are some examples of machine-generated unstructured data:

- **Satellite images:** This includes weather data or the data that the government captures in its satellite surveillance imagery. Just think about Google Earth, and you get the picture (pun intended).
- **Scientific data:** This includes seismic imagery, atmospheric data, and high energy physics.

- **Photographs and video:** This includes security, surveillance, and traffic video.
- **Radar or sonar data:** This includes vehicular, meteorological, and oceanographic seismic profiles.

The following list shows a few examples of human-generated unstructured data:

- **Text internal to your company:** Think of all the text within documents, logs, survey results, and e-mails. Enterprise information actually represents a large percent of the text information in the world today.
- **Social media data:** This data is generated from the social media platforms such as YouTube, Facebook, Twitter, LinkedIn, and Flickr.
- **Mobile data:** This includes data such as text messages and location information.
- **Website content:** This comes from any site delivering unstructured content, like YouTube, Flickr, or Instagram.

And the list goes on.

Some people believe that the term *unstructured data* is misleading because each document may contain its own specific structure or formatting based on the software that created it. However, what is internal to the document is truly unstructured.

By far, unstructured data is the largest piece of the data equation, and the use cases for unstructured data are rapidly expanding. On the text side alone, text analytics (a technology that we discuss in Chapter 13) can be used to analyze unstructured text and to extract relevant data and transform that data into structured information that can be used in various ways. For example, a popular big data use case is social media analytics for use with high-volume customer conversations. In addition, unstructured data from call center notes, e-mails, written comments in a survey, and other documents is analyzed to understand customer behavior. This can be combined with social media from tens of millions of sources to understand the customer experience.

Looking at semi-structured data

Semi-structured data is a kind of data that falls between structured and unstructured data. Semi-structured data does not necessarily conform to a fixed schema (that is, structure) but may be self-describing and may have simple label/value pairs. For example, label/value pairs might include: `<family>=Jones`, `<mother>=Jane`, and `<daughter>=Sarah`. Examples of semi-structured data include EDI, SWIFT, and XML. You can think of them as sort of payloads for processing complex events.

Understanding the role of a CMS in big data management

Organizations store some unstructured data in databases. However, they also utilize enterprise content management systems (CMSs) that can manage the complete life cycle of content. This can include web content, document content, and other forms media.

According to the Association for Information and Image Management (AIIM; www.aiim.org;), a nonprofit organization that provides education, research, and best practices, Enterprise Content Management (ECM) comprises the "strategies, methods, and tools used to capture, manage, store, preserve, and deliver content and documents related to organizational processes." The technologies included in ECM include document management, records management, imaging, workflow management, web content management, and collaboration.

A whole industry has grown up around managing content, and many content management vendors are scaling out their solutions to handle large volumes of unstructured data. However, new technologies are also evolving to help support unstructured data and the analysis of unstructured data. Some of these support both structured and unstructured data. Some support real-time streams. These include technologies like Hadoop, MapReduce, and streaming. These technologies each require chapters of their own, and we devote Chapters 8, 9, and 10 to them, respectively.

Systems that are designed to store content in the form of content management systems are no longer stand-alone solutions. Rather, they are likely to be part of an overall data management solution. For example, your organization may monitor Twitter feeds that can then programmatically trigger a CMS search. Now, the person who triggered the tweet (maybe looking for a solution to a problem) gets an answer back that offers a location where the individual can find the product that he or she might be looking for. The greatest benefit is when this type of interaction can happen in real time. It also illustrates the value of leveraging real-time unstructured, structured (customer data about the person who tweeted), and semi-structured (the actual content in the CMS) data.

TIP

The reality is that you will probably use a hybrid approach to solve your big data problems. For example, it doesn't make sense to move all your news content, for example, into Hadoop on your premises because it is supposed to help manage unstructured data.

Looking at Real-Time and Non-Real-Time Requirements

As we discuss in previous sections of this chapter, big data is often about doing things that weren't widely possible because the technology was not advanced enough or the cost of doing so was prohibitive. The big change that we are encountering with big data is the capability to leverage massive amounts of data without all the complex programming that was required in the past. Many organizations are at a tipping-point in terms of managing large volumes of complex data. Big data approaches will help keep things in balance so we don't go over the edge as the volume, variety, and velocity of data changes. Companies have had a difficult time managing increasing amounts of data that needs to be managed at high speeds. Organizations had to settle for analyzing small subsets of data which often lacked critical information to get a full picture that the data could reveal. As big data technologies evolve and get deployed, we will be able to more easily analyze the data and use it to make decisions or take actions.

The real-time aspects of big data can be revolutionary when companies need to solve significant problems. What is the impact when an organization can handle data that is streaming in real time? In general, this real-time approach is most relevant when the answer to a problem is time sensitive and business critical. This may be related to a threat to something important like detecting the performance of hospital equipment or anticipating a potential intrusion risk. The following list shows examples of when a company wants to leverage this real-time data to gain a quick advantage:

- Monitoring for an exception with a new piece of information, like fraud/intelligence
- Monitoring news feeds and social media to determine events that may impact financial markets, such as a customer reaction to a new product announcement
- Changing your ad placement during a big sporting event based on real-time Twitter streams
- Providing a coupon to a customer based on what he bought at the point of sale

Sometimes streaming data is coming in really fast and does not include a wide variety of sources, sometimes a wide variety exists, and sometimes it is a combination of the two. The question you need to ask yourself if you're moving to real time is this: Could this (problem) be solved with traditional information management capabilities or do we need newer capabilities? Is the sheer volume or velocity going to overwhelm our systems? Oftentimes it is a combination of the two.

So, if you need real-time capabilities, what are the requirements of the infrastructure to support this? We talk more about this in Chapter 3 when we discuss distributed computing. However, the following list highlights a few things you need to consider regarding a system's capability to ingest data, process it, and analyze it in real time:

- **Low latency:** Latency is the amount of time lag that enables a service to execute in an environment. Some applications require less latency, which means that they need to respond in real time. A real-time stream is going to require low latency. So you need to be thinking about compute power as well as network constraints.
- **Scalability:** Scalability is the capability to sustain a certain level of performance even under increasing loads.
- **Versatility:** The system must support both structured and unstructured data streams.
- **Native format:** Use the data in its native form. Transformation takes time and money. The capability to use the idea of processing complex interactions in the data that trigger events may be transformational.

The need to process continually increasing amounts of disparate data is one of the key factors driving the adoption of cloud services. The cloud model is large-scale and distributed. We talk more about the cloud in Chapter 6.

Putting Big Data Together

What you want to do with your structured and unstructured data indicates why you might choose one piece of technology over another one. It also determines the need to understand inbound data structures to put this data in the right place.

Managing different data types

Figure 2-2 shows a helpful table that outlines some of the characteristics of big data and the types of data management systems you might want to use to address each one. We don't expect you to know what these are yet; they are described in the chapters that follow.

Figure 2-2: The characteristics of different data types.

	Batch	Streaming	Complex Query
Structured	Hadoop	Key/Value	RDBMS
Unstructured	Document	Graph Spatial	Columnar
Both	Hybrid	Hybrid	Hybrid

Integrating data types into a big data environment

Another important aspect of big data is that you often don't need to own all the data that you will use. Many examples make the point. You may be leveraging social media data, data coming from third-party industry statistics, or even data coming from satellites. Just think about social media and you'll understand what we mean. Oftentimes, it becomes necessary to integrate different sources. This data may be coming from all internal systems, from both internal and external sources, or from entirely external sources. Much of this data may have been siloed before.

Data need not be coming to you in real time. You just may have a lot of it and it is disparate in nature. This could still qualify as a big data problem. Of course, you could also be faced with a scenario where you're seeing huge volumes of data, at high velocities, and it is disparate in nature. The point is that you won't get the business value if you deal with a variety of data sources as a set of disconnected silos of information.

Components you need include connectors and metadata, which we discuss next.

Connectors

You want to have some connectors that enable you to pull data in from various big data sources. Maybe you want a Twitter connector or a Facebook one. Maybe you need to integrate from your data warehouse with a big data source that's off your premises so that you can analyze both of these sources of data together. We discuss connectors in more detail in Chapter 15.

Metadata

A critical component to integrating all this data is the metadata. *Metadata* is the definitions, mappings, and other characteristics used to describe how to find, access, and use a company's data (and software) components. One example of metadata is data about an account number. This might include the number, description, data type, name, address, phone number, and privacy level.

Metadata can be used to help you organize your data stores and deal with new and changing sources of data. Although the idea of metadata is not new, it is changing and evolving in the context of big data. In the traditional metadata world, it is important to have a catalog that provides a single view of all data sources. But this catalog will have to be different when you don't control all these data sources. You may need an analytic tool that will help you understand the underlying metadata.