# Chapter 1

# What's in a Data Warehouse?

### In This Chapter

- Understanding what a data warehouse is and what it does
- Looking at the history of data warehousing
- Differentiating between bigger and better
- Grasping the historical perspective of a data warehouse
- Ensuring that your data warehouse isn't a data dump

If you gather 100 computer consultants experienced in data warehousing in a room and give them this single-question written quiz, "Define a data warehouse in 20 words or fewer," at least 95 of the consultants will turn in their paper with a one- or two-sentence definition that includes the terms *subject-oriented, time-variant,* and *read-only.* The other five consultants' replies will likely focus more on business than on technology and use a phrase such as "improve corporate decision-making through more timely access to information."

Forget all that. The following section gives you a no-nonsense definition guaranteed to be free of both technical and business-school jargon. Throughout the rest of the chapter, I assist you in better understanding data warehousing from its history and overall value to your business.



## The Data Warehouse: A Place for Your Data Assets

A *data warehouse* is a home for your high-value data, or *data assets,* that originates in other corporate applications, such as the one your company uses to fill customer orders for its products, or some data source external to your company, such as a public database that contains sales information gathered from all your competitors.

If your company's data warehouse were advertised as a product for sale, it might be described this way: "Contains high-quality, refined and purified information, all of which has undergone a 25-point quality check and is offered to you with a warranty to guarantee hassle-free ownership so that you can better monitor the performance of your business."

## Classifying data: What is a data asset?

Okay, I promised a definition free of technical and business-school jargon — but in the preceding section, I introduced a term (data asset) that might be considered jargon. So, I'll clarify what the term data asset means.

You can classify data that's managed within an enterprise in three groupings:

- **Run-the-business data:** Produced by corporate applications, such as the one your company uses to fill customer orders for its products or the one your company uses to manage financial transactions. The raw materials for a data warehouse.
- **Integrate-the-business data:** Built to improve the quality of and synchronize two or more corporate applications, such as a master list of customers. Data leveraged to integrate applications that weren't designed to work with each other.
- **Monitor-the-business data:** Presented to end users for reporting and decision support, such as your financial dashboard. The data is cleansed to enable users to better understand progress and evaluate cause-and-effect relationships in the data.

A *data asset* is the result of taking the raw material from the run-the-business data and producing higher-quality-data end products to integrate the business and monitor the business. Your data warehouse team should have the mission of providing high-quality data assets for enterprise use.

## Manufacturing data assets

Most organizations build a data warehouse for manufactured data assets in a relatively straightforward manner, following these steps:

1. The data warehousing team (usually computer analysts and programmers) selects a *focus area,* such as tracking and reporting the company's product sales activity against that of its competitors.

2. The team in charge of building the data warehouse assigns a group of business users and other key individuals within the company to play the role of subject-matter experts.

   Together, the data warehousing team and subject-matter experts compile a list of different types of information that can enable them to use the data warehouse to help track sales activity (or whatever the focus is for the project).

3. The group then goes through the list of information (data assets), item by item, and figures out where the data warehouse can obtain that particular piece of data (raw material).

   In most cases, the group can get the data from at least one internal (within the company) database or file, such as the one that the application uses to process orders over the Internet or the master database of all customers and their current addresses. In other cases, a piece of information isn't available from within the company's computer applications, but you could obtain it by purchasing it from some other company. Although a bank doesn't have the credit ratings and total outstanding debt for all its customers internally, for example, it can purchase that information from a third party — a credit bureau.

4. After completing the details of where the business can get each piece of information, the data warehousing team creates extraction programs.

   *Extraction programs* collect data from various internal databases and files, copy certain data to a *staging area* (a work area outside the data warehouse), cleanse the data to ensure that the data has no errors, and then copy the higher-quality data (data assets) into the data warehouse. Extraction programs are created either by hand (custom-coded) or by using specialized data warehousing products — ETL (extract, transform, and load) tools.

You can build a successful data warehouse by spending adequate time on the first two steps in the preceding list (analyzing the need for a data warehouse and how you should use it), which makes the next two steps (designing and implementing the data warehouse to make it ready to use) much easier to perform.

Interestingly, the analysis steps (determining the focus of the data warehouse and working closely with business users to figure out what information is important) are nearly identical to the steps for any other type of computer application. Most computer applications create data as a result of a transaction or set of transactions while a particular application is being used to run the business, such as filling a customer's order. The primary difference between run-the-business applications and a data warehouse is that a data warehouse relies exclusively on data obtained from other applications and sources. Figure 1-1 shows the difference between these two types of environments.

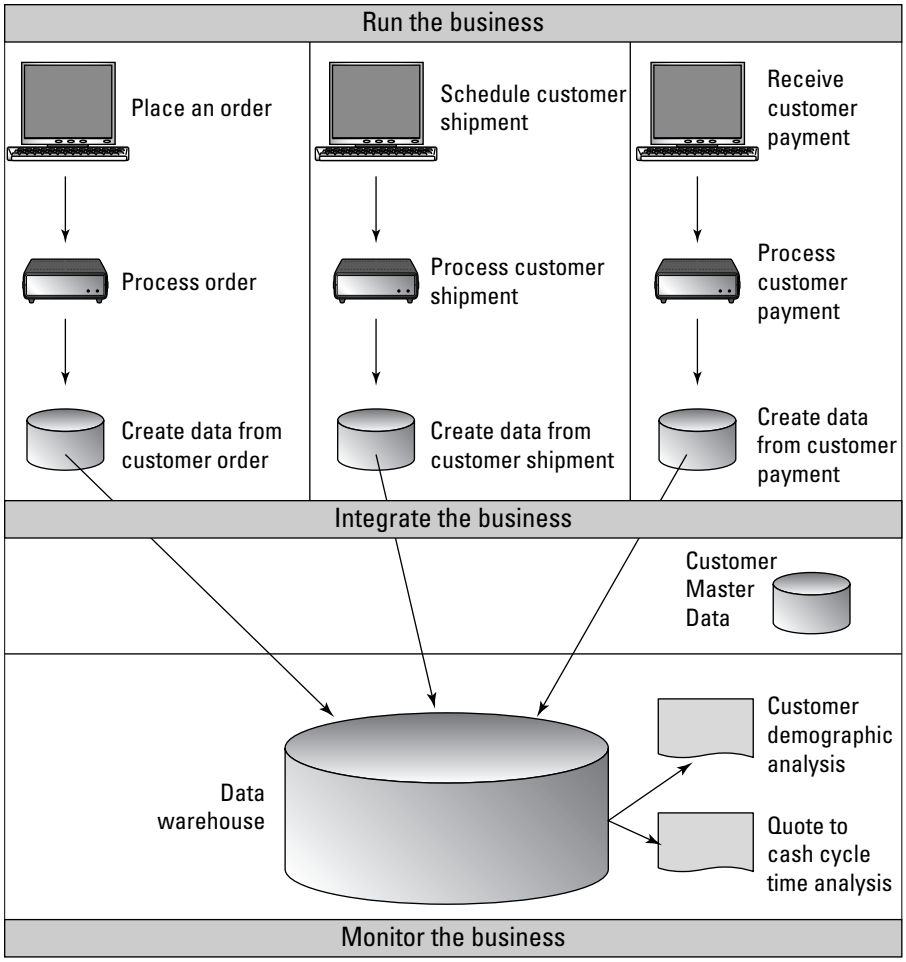


**Figure 1-1:** Most computer applications create data as a result of an activity or transaction; a data warehouse instead swipes data created elsewhere and converts it into information.

# Data Warehousing: A Working Definition

If you cringe at the thought of defining the concept of a data warehouse and the associated project to your executive sponsors, the following sections provide a more detailed and hype-free definition and explanation that you can use to wow them.

So, what's a data warehouse? In a literal sense, it is properly described through the specific definitions of the two words that make up the term:

- **Data:** Facts and information about something
- **Warehouse:** A location or facility for storing goods and merchandise

## Today's data warehousing defined

*Data warehousing* is the coordinated, architected, and periodic copying of data from various sources, both inside and outside the enterprise, into an environment optimized for analytical and informational processing.

The keys to this definition for computer professionals are that the data is copied *(duplicated)* in a controlled manner, and data that is copied periodically *(batch-oriented processing)*.

## A broader, forward looking definition

A data warehouse system has the following characteristics:

- It provides centralization of corporate data assets.
- It's contained in a well-managed environment.
- It has consistent and repeatable processes defined for loading data from corporate applications.
- It's built on an open and scalable architecture that can handle future expansion of data.
- It provides tools that allow its users to effectively process the data into information without a high degree of technical support.

The information that you use to formulate decisions typically is based on data gathered from previous experiences — what works and what doesn't. Data warehouses capture similar data, allowing business leaders to make informed decisions based on previous business data — what's working in the business and what's doesn't work in the business. Executives are realizing that the only way to sustain and gain an advantage in today's economy is to better leverage information. The data warehouse provides the platform to implement, manage, and deliver these key data assets.

*Data warehousing* is therefore the process of creating an architected information-management solution to enable analytical and informational processing despite platform, application, organizational, and other barriers.

The key concept in this definition is that a data warehouse breaks down the barriers created by non-enterprise, process-focused applications and consolidates information into a single view for users to access.

# A Brief History of Data Warehousing

Many people, when they first hear the basic principles of data warehousing — particularly copying data from one place to another — think (or even say), "That doesn't make any sense! Why waste time copying and moving data, and storing it in a different database? Why not just get it directly from its original location when someone needs it?"

To better understand the "why we do what we do" aspect of data warehousing, I outline its historical roots — how data warehousing became what it is today — in the following sections.

## Before our time — the foundation

The evolution of data warehousing can trace its roots to work done prior to computers being widely available, including

- **The continuous marketing research conducted by Charles Coolidge Parlin (1872–1942).** Parlin is now recognized as the Father of Marketing Research. He did marketing research for the Curtis Publishing Company to gather information about customers and markets to help Curtis sell more advertising in their magazine, *The Saturday Evening Post.*
- **In 1923, Arthur C. Nielsen, Sr., established ACNielsen in the United States.** Arthur C. Nielsen was one of the founders of the modern marketing research industry. Among many innovations in consumer-focused marketing and media research, Mr. Nielsen created a unique retail-measurement technique that gave clients the first reliable, objective information about competitive performance and the impact of their marketing and sales programs on revenues and profits. Nielsen information gave practical meaning to the concept of market share and made it one of the critical measures of corporate performance.

These two events in history led to what we now know as data warehousing because each of them required high-quality data to formulate trends and enable business users to make decisions.

## The 1970s — the preparation

The 1970s: Disco and leisure suits were in. And the computing world was dominated by the mainframe. Real data-processing applications, the ones run on the corporate mainframe, almost always had a complicated set of files or early-generation databases (not the table-oriented relational databases most applications use today) in which they stored data.

Although the applications did a fairly good job of performing routine data-processing functions, data created as a result of these functions (such as information about customers, the products they ordered, and how much money they spent) was locked away in the depths of the files and databases. It was almost impossible, for example, to see how retail stores in the eastern region were doing against stores in the western region, against their competitors, or even against their own performance in some earlier period. At best, you could have written up a report request and sent it to the data-processing department, where it was put on a waiting list with a couple thousand other report requests, and you might have had an answer in a few months — or not.

Some enterprising, forward-thinking people decided to take another approach to the data access problem. During the 1970s, while minicomputers were becoming popular, the thinking went like this: Rather than make requests to the data-processing department every time you need data from an application's files or databases, why not identify a few key data elements (for example, a customer's ID number, total units purchased in the most recent month, and total dollars spent) and have the data-processing folks copy this data to a tape each month during a slow period, such as over a weekend or during the midnight shift? You could then load the data from the tape into another file on the minicomputer, and the business users could use decision-support tools and *report writers* (products that allowed access to data without having to write separate programs) to get answers to their business questions and avoid continually bothering the data-processing department.

Although this approach worked (sort of) in helping to reduce the backlog of requests that the data-processing department had to deal with, the usefulness of the extracted and copied data usually didn't live up to the vision of the people who put the systems in place. Suppose that a company had three separate systems to handle customer sales: one for the eastern U.S. region, one for the western U.S. region, and one for all stores in Europe. Also, each of these three systems was independent from the others. Although data copied from the system that processed sales for the western U.S. region was helpful in analyzing western region activity for each month and maybe on a historical basis (if you retained previous batches of data), you couldn't easily answer questions about trends across the entire United States or the world without copying more data from each of the systems. People typically gave up because answering their questions just took too much time.

Additionally, commercial and hardware/software companies began to emerge with solutions to this problem. Between 1976 and 1979, the concept for a new company, Teradata, grew out of research at the California Institute of Technology (Caltech), driven from discussions with Citibank's advanced technology group. Founders worked to design a database management system for parallel processing with multiple microprocessors, specifically for decision support. Teradata was incorporated on July 13, 1979 and started in a garage in Brentwood, California. The name Teradata was chosen to symbolize the ability to manage *terabytes* (trillions of bytes) of data.

## The 1980s — the birth

The 1980s: the era of yuppies. PCs, PCs, and more PCs suddenly appeared everywhere you looked — as well as more and more minicomputers (and even a few Macintoshes). Before anyone knew it, "real computer applications" were no longer only on mainframes; they were all over the place — everywhere you looked in an organization. The problem called *islands of data* was beginning to look ominous: How could an organization hope to compete if its data was scattered all over the place on different computer systems that weren't even all under the control of the centralized data-processing department? (Never mind that even when the data was all stored on mainframes, it was still isolated in different files and databases, so it was just as inaccessible.)

A group of enterprising, forward-thinking people came up with a new idea: Because data is located all over the place, why not create special software to enable people to make a request at a PC or terminal, such as "Show per-store sales in all worldwide regions, ranked in descending order by improvement over sales in the same period a year earlier"? This new type of software, called a *distributed database management system* (distributed DBMS, or DDBMS), would magically pull the requested data from databases across the organization, bring all the data back to the same place, and then consolidate it, sort it, and do whatever else was necessary to answer the user's question. (This process was supposed to happen pretty darned quickly.)

To make a long story short, although the concept of DDBMSs was a good one and early results from research were promising, the results were plain and simple: They just didn't work in the real world. Also, the islands-of-data problem still existed.

Meanwhile, Teradata began shipping commercial products to solve this problem. Wells Fargo Bank received the first Teradata test system in 1983, a parallel RDBMS (relational database management system) for decision support — the world's first. By 1984, Teradata released a production version

of their product, and in 1986, *Fortune* magazine named Teradata Product of the Year. Teradata, still in existence today, built the first data warehousing appliance — a combination of hardware and software to solve the data warehousing needs of many. Other companies began to formulate their strategies, as well.

In 1988, Barry Devlin and Paul Murphy of IBM Ireland introduced the term *business data warehouse* as a key component of the EBIS (Europe/Middle East/Africa Business Information System). *EBIS* was defined as a comprehensive architecture aimed at providing a cross-functional business information system that's easy to use and has the flexibility to change while the business environment develops, even at a rapid rate. The flexibility and cross-functional support are a result of the relational database technology on which the EBIS system is based. When describing the business data warehouse, they articulated the need to "ease access to the data and to achieve a coherent framework for such access, it is vital that all the data reside in a single logical repository."

Additionally, Ralph Kimball founded Red Brick Systems in 1986. Red Brick began to emerge as a visionary software company by discussing how to improve data access. They were promoting a specialized relational database platform which enabled large performance gains for complex ad-hoc queries. Often, they could prove performance over ten times that of other vendor databases of the time. The key to Red Brick's technology was indexes — a software answer to Teradata's hardware-based solution. These indexes where technical solutions to the key manners in which users described the data within a data warehouse — customers, products, demographics, and so on.

In short, the 1980s were the birth place of data warehousing innovation.

## The 1990s — the adolescent

During the 1990s, disco made a comeback. At the beginning of the decade, some 20 years after computing went mainstream, business computer users were still no closer to being able to use the trillions of bytes of data locked away in databases all over the place to make better business decisions.

The original group of enterprising, forward-thinking people had retired (or perhaps switched to doing Web site development). Using the time-honored concept of "something old, something new" (the "something borrowed, something blue" part doesn't quite fit), a new approach to solving the islands-of-data problem surfaced. If the 1980s approach of reaching out and

accessing data directly from the files and databases didn't work, the 1990s philosophy involved going back to the 1970s method, in which data from those places was copied to another location — only doing it right this time.

And data warehousing was born.

In 1993, Bill Inmon wrote *Building the Data Warehouse* (Wiley). Many people recognize Bill as the Father of Data Warehousing. Additional publications emerged, including the 1996 book by Ralph Kimball, *The Data Warehouse Toolkit* (Wiley), which discussed general-purpose dimensional design techniques to improve the data architecture for query-centered decision support systems.

With hardware and software for data warehousing becoming common place, writings began to emerge complementing those of Inmon and Kimball. Specifically, techniques appeared that enabled those employed by Information Systems departments to better understand the trend that involved not going after data from just one place, such as a single application, but rather going after all the data you need, regardless of how many different applications and computers are used in the organization. Client/server technology can be used to put the data on servers and give users new and improved analysis tools on their PCs.

## The 2000s — the adult

In the more modern era (the 2000s, the era of reality television shows and mobile communication devices), people are more connected than ever before. Information is everywhere. New languages are being created because of texting and instant messaging. Acronyms such as TTYL (talk to you later), LOL (laughing out loud), and BRB (be right back) are commonplace. And a huge number of people provide feedback to vote people off of competitions on shows such as *American Idol* — bringing new meaning to market research and understanding what will sell. For example, in 2006, viewers cast 63 million votes for the contestants in the *American Idol* finale — which exceeded the most votes obtained by a United States president (Ronald Reagan, with 54.5 million votes). So, the world is definitely now connected!

In the world of data warehousing, the amount of data continues to grow. But, while it does, the vendor community and options have begun to consolidate. The selection pool is rapidly diminishing. In 2006, Microsoft acquired ProClarity, jumping into the data warehousing market. In 2007, Oracle purchased Hyperion, SAP acquired Business Objects, and IBM merged with Cognos. The data warehousing leaders of the 1990s have been gobbled up by some of the largest providers of information system solutions in the world.

Although the vendor community has consolidated, innovation hasn't ceased. More cost-effective solutions have emerged, led by Microsoft enabling small and mid-sized businesses to implement data warehousing solutions. Additionally, less expensive alternatives are emerging from a new set of vendors, those within the open source community, including vendors such as Pentaho and Jaspersoft. Open source business intelligence tools enable corporate application vendors to embed data warehousing solutions into their software suites. And other innovations have emerged, including data warehouse appliances from vendors such as Netezza and DATAllegro (acquired by Microsoft), and performance management appliances that enable real-time performance monitoring. These innovative solutions can also provide cost savings because they're often plug-compatible to legacy data warehouse solutions.


While time ticks by, you need to have a plan in place before you begin your data warehousing process. Know the focus of what you're trying to do and the questions you're likely to be asking. Will you be asking mostly about sales activity? If so, put plans in place for regular monthly (or weekly or even daily) extractions of data about customers, the products they buy, and the amounts of money they spend. If you work at a bank and your business focus is managing the risk across loan portfolios, for example, get information from the bank's applications that handle loan payments, delinquencies, and other data you need; then, add in data from the credit bureau about your customers' respective overall financial profiles.

## Is a Bigger Data Warehouse a Better Data Warehouse?


A common misconception that many data warehouse aficionados hold is that the only good data warehouse is a big data warehouse — an enormously big data warehouse. Many people even take the stance that unless they have some astronomically large number of bytes stored, it isn't truly a data warehouse. "Five hundred gigabytes? Okay, that's a *real* data warehouse; it would be a better data warehouse, however, if it had at least a terabyte (1 trillion bytes) of data. Twenty-five gigabytes? Sorry, that's a data mart, not a data warehouse." (See Chapter 4 for a discussion of the differences between data marts and data warehouses.)

The size of a data warehouse is a characteristic — almost a by-product — of a data warehouse; it's not an objective. No one should ever set out with a mission to "build a 500-gigabyte data warehouse that contains (whatever)."

To determine the size you need for your data warehouse, follow these steps:

1. **Determine the mission, or the business objectives, of the data warehouse.**

   Ask the question, "Why bother creating this warehouse?"

2. **Determine the functionality that you want the data warehouse to have.**

   Figure out what types of questions users will ask.

3. **Determine what *contents* (types of data) the data warehouse needs to support its functionality.**

   Understand what types of answers your users will seek.

4. **Determine, based on the content volume (which is based on the functionality, which in turn is based on the mission), how big you need to make your data warehouse.**

# Realizing That a Data Warehouse (Usually) Has a Historical Perspective

In almost all situations, a data warehouse has a historical perspective. Some amount of time lag occurs between the time something happens in one of the data sources (a new record is added or an existing one is modified in a corporate application, for example) and the time that the event's results are available in the data warehouse.

The reason for the time lag is that you usually bulk-load data into a data warehouse in large batches. Figure 1-2 illustrates a model of bulk-loading data.


Bulk-loading is giving way to *messaging,* the process of sending a small number of updates (perhaps only one at a time) much more frequently from the data source to a target — in this case, the data warehouse. With messaging, you have a much more up-to-date picture of your data warehouse's subject areas than you do with bulk-loading because you're putting information into an operational data store (as discussed in Chapter 20), rather than into a traditional data warehouse. Additionally, the world of service-oriented architectures (SOAs) and Web 2.0 are driving the messaging and presentation of data to near real-time in some industries. The combination of the data warehouse's historic perspective with this near-real-time sourcing of information enables business leaders to monitor the situation and make decisions at the speed of the business.

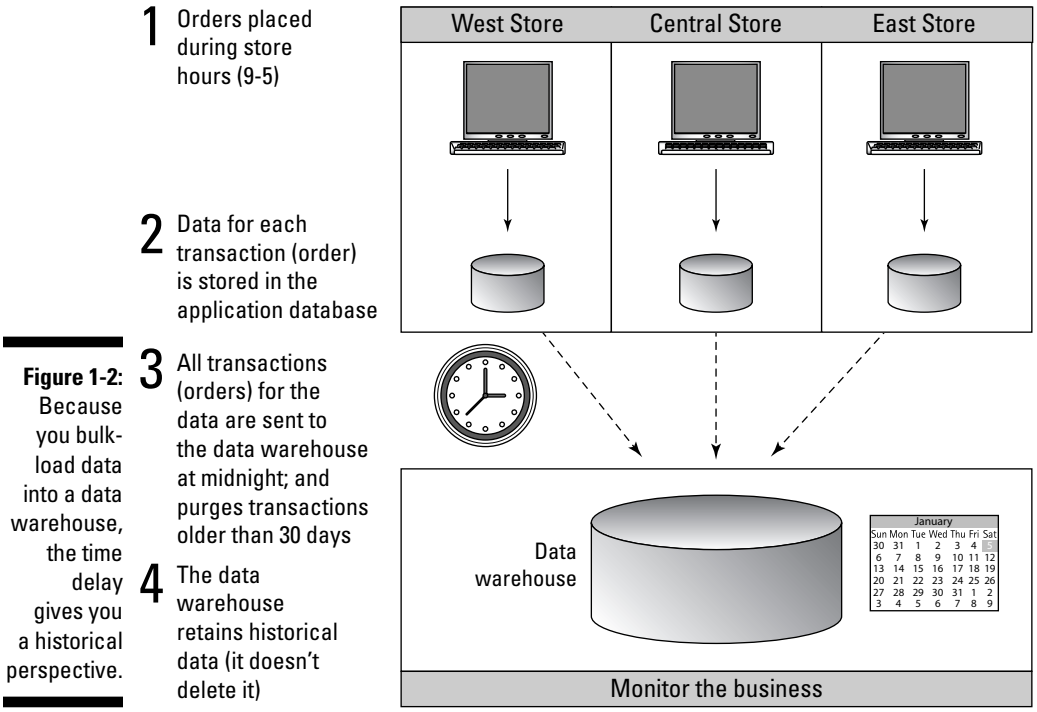


**Figure 1-2:** Because you bulk-load data into a data warehouse, the time delay gives you a historical perspective.

# It's Data Warehouse, Not Data Dump

An often-heard argument about what should be stored in a data warehouse goes something like this: "If I have to take the trouble to pull out data from all these different applications, why not just get as much as I possibly can? If I don't get everything, or as much as possible, I won't be able to ask all the business questions I might want to."

In a commonly related story about knowledge gained from a successful data warehouse implementation, a grocery-store chain discovered an unusually high correlation of disposable baby diapers and beer sales during a two- or three-hour period early every Friday evening and found out that a significant number of people on their way home from work were buying both these items. The store then began stocking display shelves with beer and disposable diapers next to one another, and sales increased significantly.

Although I don't know whether this story is true (it certainly has been told often enough), I believe that it confuses the issue when you have to figure out what should — and should not — be in your data warehouse. The moral of

this story is usually mistaken as, "Put as much data as possible in the warehouse." In reality, the data warehouse just described was probably one that focused exclusively on sales activity. Remember that although disposable diapers and beer are dramatically different products, they're both members of the same *type* of data (retail products).

The following example emphasizes why you should be selective about what goes in your data warehouse and not just assume that you have to get every possible type of data from all the sources, just in case you want to ask your data warehouse any question.

Suppose that you're creating a data warehouse for a cruise ship company. As shown in Figure 1-3, the Tucson Desert Cruise Ship Company (its motto is "Who needs an ocean?") uses four applications that handle different tasks:

- Reservations and cancellations
- Food-and-beverage service for all cruises
- All trip itineraries and after-the-fact information about the weather, unusual events, and all onboard entertainment scheduling
- All crew assignments

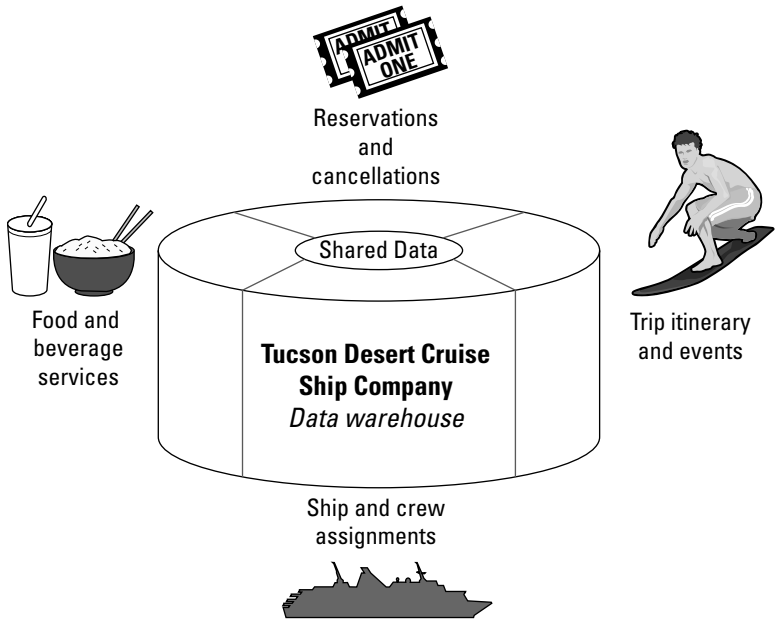


**Figure 1-3:** A fictional company's proposed data warehousing environment.

Figure 1-4 shows one possible environment for your data warehouse if you pursue the philosophy of "Go get everything you possibly can," or what I call the *data dump* approach.

By having the information shown in Figure 1-4 in your data warehouse, you — and every other person who uses the warehouse — can ask questions and make report requests, such as "What's the average number of room-service vegetarian meals ordered by passengers who were on their third cruise with Captain Grumby in command and in which a half-day stop was made in Grand Cayman when its temperature was between 75 and 80 degrees?"

Asking this type of question doesn't have any real business value, however. Assuming that you receive an answer to the question, what can you do with that information to have a positive business effect?

For some types of data, you can analyze, analyze, and analyze some more — and still find out little of value that could positively affect your business. Although you can put this data in your warehouse, you probably won't get much for your trouble. Other types of data, though, have significant value unavailable until placed in the data warehouse. Concentrate on the latter, and ignore the former!

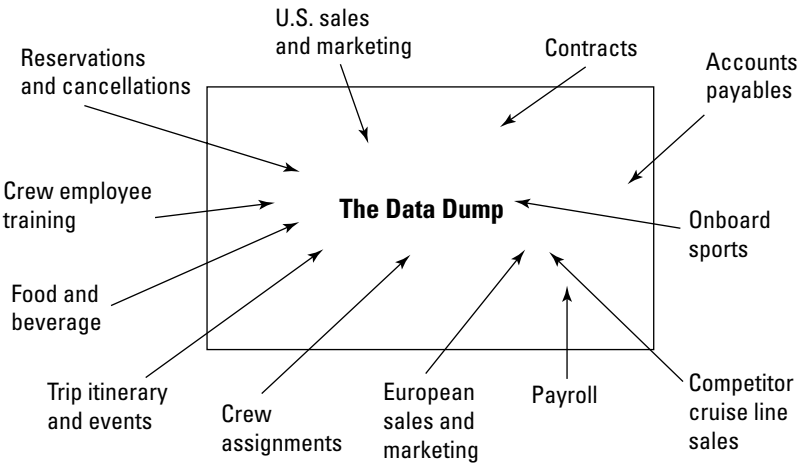


**Figure 1-4:** What your data dump can look like.