Chapter 1

A brief history of data warehousing and first-generation data warehouses

In the beginning there were simple mechanisms for holding data. There were punched cards. There were paper tapes. There was core memory that was hand beaded. In the beginning storage was very expensive and very limited.

A new day dawned with the introduction and use of magnetic tape. With magnetic tape, it was possible to hold very large volumes of data cheaply. With magnetic tape, there were no major restrictions on the format of the record of data. With magnetic tape, data could be written and rewritten. Magnetic tape represented a great leap forward from early methods of storage.

But magnetic tape did not represent a perfect world. With magnetic tape, data could be accessed only sequentially. It was often said that to access 1% of the data, 100% of the data had to be physically accessed and read. In addition, magnetic tape was not the most stable medium on which to write data. The oxide could fall off or be scratched off of a tape, rendering the tape useless.

Disk storage represented another leap forward for data storage. With disk storage, data could be accessed directly. Data could be written and rewritten. And data could be accessed en masse. There were all sorts of virtues that came with disk storage.

DATA BASE MANAGEMENT SYSTEMS

Soon disk storage was accompanied by software called a "DBMS" or "data base management system." DBMS software existed for the

purpose of managing storage on the disk itself. Disk storage managed such activities as

- identifying the proper location of data;
- resolving conflicts when two or more units of data were mapped to the same physical location;
- allowing data to be deleted;
- spanning a physical location when a record of data would not fit in a limited physical space;
- and so forth.

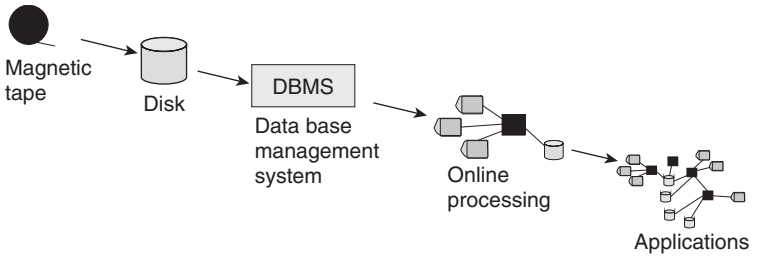
Among all the benefits of disk storage, by far and away the greatest benefit was the ability to locate data quickly. And it was the DBMS that accomplished this very important task.

ONLINE APPLICATIONS

Once data could be accessed directly, using disk storage and a DBMS, there soon grew what came to be known as online applications. Online applications were applications that depended on the computer to access data consistently and quickly. There were many commercial applications of online processing. These included ATMs (automated teller processing), bank teller processing, claims processing, airline reservations processing, manufacturing control processing, retail point of sale processing, and many, many more. In short, the advent of online systems allowed the organization to advance into the 20th century when it came to servicing the day-to-day needs of the customer. Online applications became so powerful and popular that they soon grew into many interwoven applications.

Figure 1.1 illustrates this early progression of information systems.

In fact, online applications were so popular and grew so rapidly that in short order there were lots of applications.



■ **FIGURE 1.1** The early progression of systems.

And with these applications came a cry from the end user—"I know the data I want is there somewhere, if I could only find it." It was true. The corporation had a whole roomful of data, but finding it was another story altogether. And even if you found it, there was no guarantee that the data you found was correct. Data was being proliferated around the corporation so that at any one point in time, people were never sure about the accuracy or completeness of the data that they had.

PERSONAL COMPUTERS AND 4GL TECHNOLOGY

To placate the end user's cry for accessing data, two technologies emerged—personal computer technology and 4GL technology.

Personal computer technology allowed anyone to bring his/her own computer into the corporation and to do his/her own processing at will. Personal computer software such as spreadsheet software appeared. In addition, the owner of the personal computer could store his/her own data on the computer. There was no longer a need for a centralized IT department. The attitude was—if the end users are so angry about us not letting them have their own data, just give them the data.

At about the same time, along came a technology called "4GL"—fourth-generation technology. The idea behind 4GL technology was to make programming and system development so straightforward that anyone could do it. As a result, the end user was freed from the shackles of having to depend on the IT department to feed him/her data from the corporation.

Between the personal computer and 4GL technology, the notion was to emancipate the end user so that the end user could take his/her own destiny into his/her own hands. The theory was that freeing the end user to access his/her own data was what was needed to satisfy the hunger of the end user for data.

And personal computers and 4GL technology soon found their way into the corporation.

But something unexpected happened along the way. While the end users were now free to access data, they discovered that there was a lot more to making good decisions than merely accessing data. The end users found that, even after data had been accessed,

- if the data was not accurate, it was worse than nothing, because incorrect data can be very misleading;
- incomplete data is not very useful;

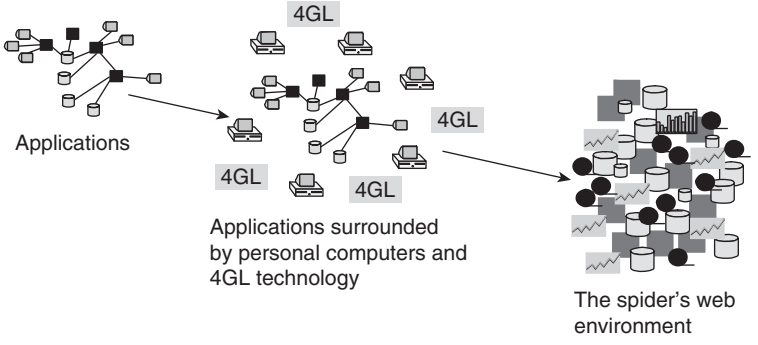
- data that is not timely is less than desirable;
- when there are multiple versions of the same data, relying on the wrong value of data can result in bad decisions;
- data without documentation is of questionable value.

It was only after the end users got access to data that they discovered all the underlying problems with the data.

THE SPIDER WEB ENVIRONMENT

The result was a big mess. This mess is sometimes affectionately called the "spider's web" environment. It is called the spider's web environment because there are many lines going to so many places that they are reminiscent of a spider's web.

Figure 1.2 illustrates the evolution of the spider's web environment in the typical corporate IT environment.

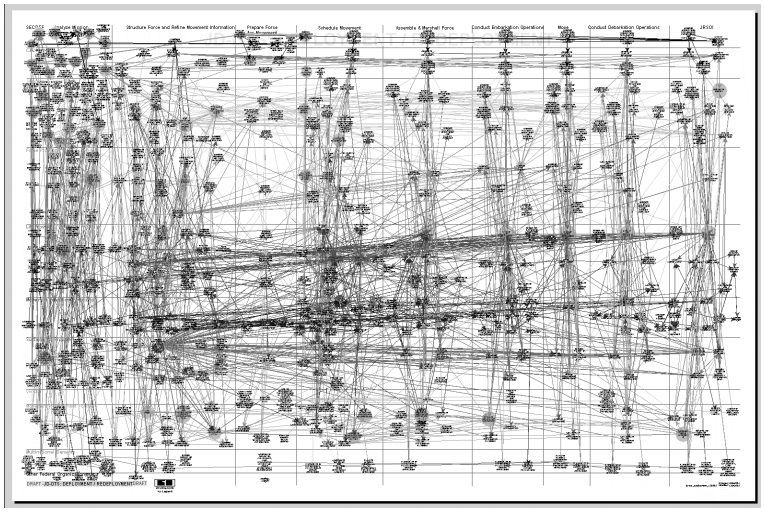


■ **FIGURE 1.2** The early progression led to the spider's web environment.

The spider's web environment grew to be unimaginably complex in many corporate environments. As testimony to its complexity, consider the real diagram of a corporation's spider's web of systems shown in Figure 1.3.

One looks at a spider's web with awe. Consider the poor people who have to cope with such an environment and try to use it for making good corporate decisions. It is a wonder that anyone could get anything done, much less make good and timely decisions.

The truth is that the spider's web environment for corporations was a dead end insofar as architecture was concerned. There was no future in trying to make the spider's web environment work.



■ **FIGURE 1.3** A real spider's web environment.

The frustration of the end user, the IT professional, and management resulted in a movement to a different information architecture. That information systems architecture was one that centered around a data warehouse.

EVOLUTION FROM THE BUSINESS PERSPECTIVE

The progression that has been described has been depicted from the standpoint of technology. But there is a different perspective—the perspective of the business. From the perspective of the business person, the progression of computers began with simple automation of repetitive activities. The computer could handle more data at a greater rate of speed with more accuracy than any human was capable of. Activities such as the generation of payroll, the creation of invoices, the payments being made, and so forth are all typical activities of the first entry of the computer into corporate life.

Soon it was discovered that computers could also keep track of large amounts of data. Thus were "master files" born. Master files held inventory, accounts payable, accounts receivable, shipping lists, and so forth. Soon there were online data bases, and with online data bases the computer started to make its way into the core of the business. With online data bases airline clerks were emancipated. With online processing, bank tellers could do a whole new range of functions. With online processing insurance claims processing was faster than ever.

It is in online processing that the computer was woven into the fabric of the corporation. Stated differently, once online processing began to be used by the business person, if the online system went down, the entire business suffered, and suffered immediately. Bank tellers could not do their job. ATMs went down. Airline reservations went into a manual mode of operation, and so forth.

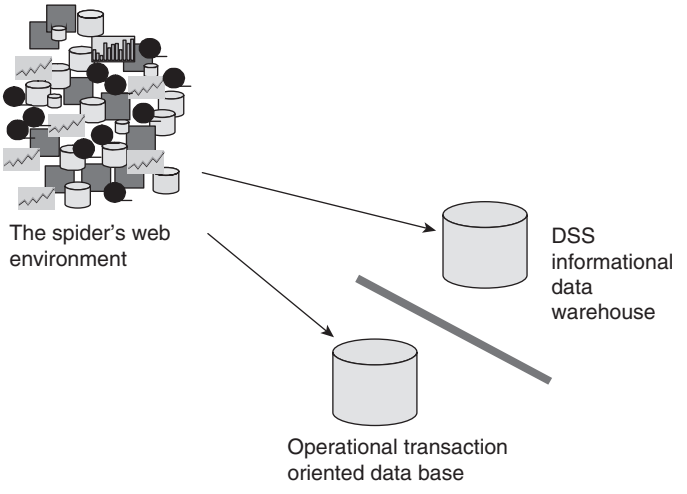
Today, there is yet another incursion by the computer into the fabric of the business, and that incursion is into the managerial, strategic decision-making aspects of the business. Today, corporate decisions are shaped by the data flowing through the veins and arteries of the corporation.

So the progression that is being described is hardly a technocentric process. There is an accompanying set of business incursions and implications, as well.

THE DATA WAREHOUSE ENVIRONMENT

Figure 1.4 shows the transition of the corporation from the spider's web environment to the data warehouse environment.

The data warehouse represented a major change in thinking for the IT professional. Prior to the advent of the data warehouse, it was thought that a data base should be something that served all purposes for data. But with the data warehouse it became apparent that there were many different kinds of data bases.



■ **FIGURE 1.4** A fundamental division of data into different types of data bases was recognized.

WHAT IS A DATA WAREHOUSE?

The data warehouse is a basis for informational processing. It is defined as being

- subject oriented;
- integrated;
- nonvolatile;
- time variant;
- a collection of data in support of management's decision.

This definition of a data warehouse has been accepted from the beginning.

A data warehouse contains integrated granular historical data. If there is any secret to a data warehouse it is that it contains data that is *both* integrated and granular. The integration of the data allows a corporation to have a true enterprise-wide view of the data. Instead of looking at data parochially, the data analyst can look at it collectively, as if it had come from a single well-defined source, which most data warehouse data assuredly does not. So the ability to use data warehouse data to look across the corporation is the first major advantage of the data warehouse. Additionally, the granularity—the fine level of detail—allows the data to be very flexible. Because the data is granular, it can be examined in one way by one group of people and in another way by another group of people. Granular data means that there is still only one set of data—one single version of the truth. Finance can look at the data one way, marketing can look at the same data in another way, and accounting can look at the same data in yet another way. If it turns out that there is a difference of opinion, there is a single version of truth that can be returned to resolve the difference.

Another major advantage of a data warehouse is that it is a historical store of data. A data warehouse is a good place to store several years' worth of data.

It is for these reasons and more that the concept of a data warehouse has gone from a theory derided by the data base theoreticians of the day to conventional wisdom in the corporation today.

But for all the advantages of a data warehouse, it does not come without some degree of pain.

INTEGRATING DATA—A PAINFUL EXPERIENCE

The first (and most pressing) pain felt by the corporation is that of the need to integrate data. If you are going to build a data warehouse,

you *must* integrate data. The problem is that many corporations have legacy systems that are—for all intents and purposes—intractable. People are really reluctant to make any changes in their old legacy systems, but building a data warehouse requires exactly that.

So the first obstacle to the building of a data warehouse is that it requires that you get your hands dirty by going back to the old legacy environment, figuring out what data you have, and then figuring out how to turn that application-oriented data into corporate data.

This transition is *never* easy, and in some cases it is almost impossible. But the value of integrated data is worth the pain of dealing with unintegrated, application-oriented data.

VOLUMES OF DATA

The second pain encountered with data warehouses is dealing with the volumes of data that are generated by data warehousing. Most IT professionals have never had to cope with the volumes of data that accompany a data warehouse. In the application system environment, it is good practice to jettison older data as soon as possible. Old data is not desirable in an operational application environment because it slows the system down. Old data clogs the arteries. Therefore any good systems programmer will tell you that to make the system efficient, old data must be dumped.

But there is great value in old data. For many analyses, old data is extremely valuable and sometimes indispensible. Therefore, having a convenient place, such as a data warehouse, in which to store old data is quite useful.

A DIFFERENT DEVELOPMENT APPROACH

A third aspect of data warehousing that does not come easily is the way data warehouses are constructed. Developers around the world are used to gathering requirements and then building a system. This time-honored approach is drummed into the heads of developers as they build operational systems. But a data warehouse is built quite differently. It is built iteratively, a step at a time. First one part is built, then another part is built, and so forth. In almost every case, it is a prescription for disaster to try to build a data warehouse all at once, in a "big bang" approach.

There are several reasons data warehouses are not built in a big bang approach. The first reason is that data warehouse projects tend to be

large. There is an old saying: "How do you eat an elephant? If you try to eat the elephant all at once, you choke. Instead the way to eat an elephant is a bite at a time." This logic is never more true than when it comes to building a data warehouse.

There is another good reason for building a data warehouse one bite at a time. That reason is that the requirements for a data warehouse are often not known when it is first built. And the reason for this is that the end users of the data warehouse do not know exactly what they want. The end users operate in a mode of discovery. They have the attitude—"When I see what the possibilities are, then I will be able to tell you what I really want." It is the act of building the first iteration of the data warehouse that opens up the mind of the end user to what the possibilities really are. It is only after seeing the data warehouse that the user requirements for it become clear.

The problem is that the classical systems developer has never built such a system in such a manner before. The biggest failures in the building of a data warehouse occur when developers treat it as if it were just another operational application system to be developed.

EVOLUTION TO THE DW 2.0 ENVIRONMENT

This chapter has described an evolution from very early systems to the DW 2.0 environment. From the standpoint of evolution of architecture it is interesting to look backward and to examine the forces that shaped the evolution. In fact there have been many forces that have shaped the evolution of information architecture to its highest point—DW 2.0.

Some of the forces of evolution have been:

- The demand for more and different uses of technology: When one compares the very first systems to those of DW 2.0 one can see that there has been a remarkable upgrade of systems and their ability to communicate information to the end user. It seems almost inconceivable that not so long ago output from computer systems was in the form of holes punched in cards. And end user output was buried as a speck of information in a hexadecimal dump. The truth is that the computer was not very useful to the end user as long as output was in the very crude form in which it originally appeared.
- Online processing: As long as access to data was restricted to very short amounts of time, there was only so much the business

person could do with the computer. But the instant that online processing became a possibility, the business opened up to the possibilities of the interactive use of information intertwined in the day-to-day life of the business. With online processing, reservations systems, bank teller processing, ATM processing, online inventory management, and a whole host of other important uses of the computer became a reality.

- The hunger for integrated, corporate data: As long as there were many applications, the thirst of the office community was slaked. But after a while it was recognized that something important was missing. What was missing was corporate information. Corporate information could not be obtained by adding together many tiny little applications. Instead data had to be recast into the integrated corporate understanding of information. But once corporate data became a reality, whole new vistas of processing opened up.

- The need to include unstructured, textual data in the mix: For many years decisions were made exclusively on the basis of structured transaction data. While structured transaction data is certainly important, there are other vistas of information in the corporate environment. There is a wealth of information tied up in textual, unstructured format. Unfortunately unlocking the textual information is not easy. Fortunately, textual ETL (extract/transform/load) emerged and gave organizations the key to unlocking text as a basis for making decisions.

- Capacity: If the world of technology had stopped making innovations, a sophisticated world such as DW 2.0 simply would not have been possible. But the capacity of technology, the speed with which technology works, and the ability to interrelate different forms of technology all conspire to create a technological atmosphere in which capacity is an infrequently encountered constraint. Imagine a world in which storage was held entirely on magnetic tape (as the world was not so long ago.) Most of the types of processing that are taken for granted today simply would not have been possible.

- Economics: In addition to the growth of capacity, the economics of technology have been very favorable to the consumer. If the consumer had to pay the prices for technology that were used a decade ago, the data warehouses of DW 2.0 would simply be out of orbit from a financial perspective. Thanks to Moore's law,

the unit cost of technology has been shrinking for many years now. The result is affordability at the consumer level.

These then are some of the evolutionary factors that have shaped the world of technology for the past several decades and have fostered the architectural evolution, the epitome of which is DW 2.0.

THE BUSINESS IMPACT OF THE DATA WAREHOUSE

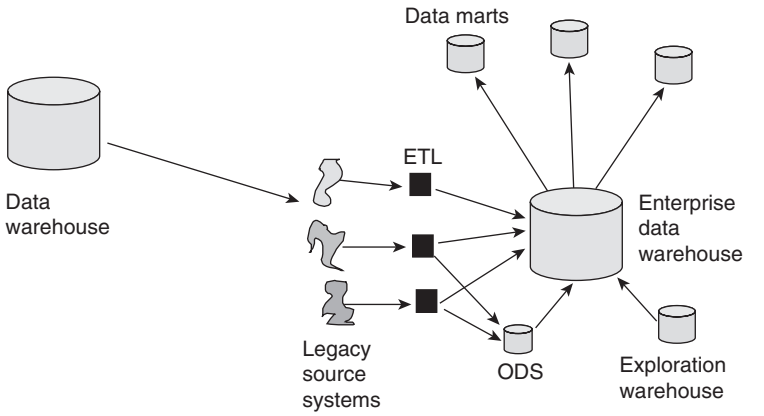
The impact of the data warehouse on business has been considerable. Some of the areas of business that are directly impacted by the advent of the data warehouse are:

- The frequent flyer programs from the airline environment: The single most valuable piece of technology that frequent flyer programs have is their central data warehouse.
- Credit card fraud analysis: Spending profiles are created for each customer based on his/her past activities. These profiles are created from a data warehouse. When a customer attempts to make a purchase that is out of profile, the credit card company checks to see if a fraudulent use of the credit card is occurring.
- Inventory management: Data warehouses keep detailed track of inventory, noting trends and opportunities. By understanding—at a detailed level—the consumption patterns for the goods managed by an organization, a company can keep track of both oversupply and undersupply.
- Customer profiles: Organizations wishing to "get to know their customer better" keep track of spending and attention patterns as exhibited by their customers and prospects. This detailed information is stored in a data warehouse.

And there are many other ways in which the data warehouse has impacted business. In short the data warehouse becomes the corporate memory. Without the data warehouse there is—at best—a short corporate memory. But with a data warehouse there is a long and detailed corporate memory. And that corporate memory can be exploited in many ways.

VARIOUS COMPONENTS OF THE DATA WAREHOUSE ENVIRONMENT

There are various components of a data warehouse environment. In the beginning those components were not widely recognized.



■ **FIGURE 1.5** Soon the data warehouse evolved into a full-blown architecture sometimes called the corporate information factory.

But soon the basic components of the data warehouse environment became known.

Figure 1.5 illustrates the progression from an early stand-alone data warehouse to a full-fledged data warehouse architecture.

The full-fledged architecture shown in Figure 1.5 includes some notable components, which are discussed in the following sections.

ETL—extract/transform/load

ETL technology allows data to be pulled from the legacy system environment and transformed into corporate data. The ETL component performs many functions, such as

- logical conversion of data;
- domain verification;
- conversion from one DBMS to another;
- creation of default values when needed;
- summarization of data;
- addition of time values to the data key;
- restructuring of the data key;
- merging of records;
- deletion of extraneous or redundant data.

The essence of ETL is that data enters the ETL labyrinth as application data and exits the ETL labyrinth as corporate data.

ODS—operational data store

The ODS is the place where online update of integrated data is done with online transaction processing (OLTP) response time. The ODS is a hybrid environment in which application data is transformed (usually by ETL) into an integrated format. Once placed in the ODS, the data is then available for high-performance processing, including update processing. In a way the ODS shields the classical data warehouse from application data and the overhead of transaction integrity and data integrity processing that comes with doing update processing in a real-time mode.

Data mart

The data mart is the place where the end user has direct access and control of his/her analytical data. The data mart is shaped around one set of users' general expectations for the way data should look, often a grouping of users by department. Finance has its own data mart. Marketing has a data mart. Sales has a data mart, and so forth. The source of data for each data mart is the data warehouse. The data mart is usually implemented using a different technology than the data warehouse. Each data mart usually holds considerably less data than the data warehouse. Data marts also generally contain a significant amount of summarized data and aggregated data.

Exploration warehouse

The exploration warehouse is the facility where end users who want to do discovery processing go. Much statistical analysis is done in the exploration warehouse. Much of the processing in the exploration warehouse is of the heuristic variety. Most exploration warehouses hold data on a project basis. Once the project is complete, the exploration warehouse goes away. It absorbs the heavy duty processing requirements of the statistical analyst, thus shielding the data warehouse from the performance drain that can occur when heavy statistical use is made of the exploration warehouse.

The architecture that has been described is commonly called the corporate information factory.

The simple data warehouse, then, has progressed from the notion of a place where integrated, granular, historical data is placed to a full-blown architecture.

THE EVOLUTION OF DATA WAREHOUSING FROM THE BUSINESS PERSPECTIVE

In the earliest days of computing end users got output from computers in a very crude manner. In those days end users read holes punched in cards and read hexadecimal dumps in which one tiny speck of information hid in a thousand pages of cryptic codes. Soon reports became the norm because the earliest forms of interface to the end users were indeed very crude.

Soon end users became more sophisticated. The more power the end user got, the more power the end user could imagine. After reports came, online information was available instantaneously (more or less).

And after online transaction processing, end users wanted integrated corporate data, by which large amounts of data were integrated into a cohesive whole. And then end users wanted historical data.

Throughout this progression came a simultaneous progression in architecture and technology. It is through first-generation data warehouses that the end user had the ultimate in analytical capabilities.

Stated differently, without first-generation data warehouses, the end user was left with only a fraction of the need for information being satisfied. The end user's hunger for corporate information was the single most important driving force behind the evolution to first-generation data warehousing.

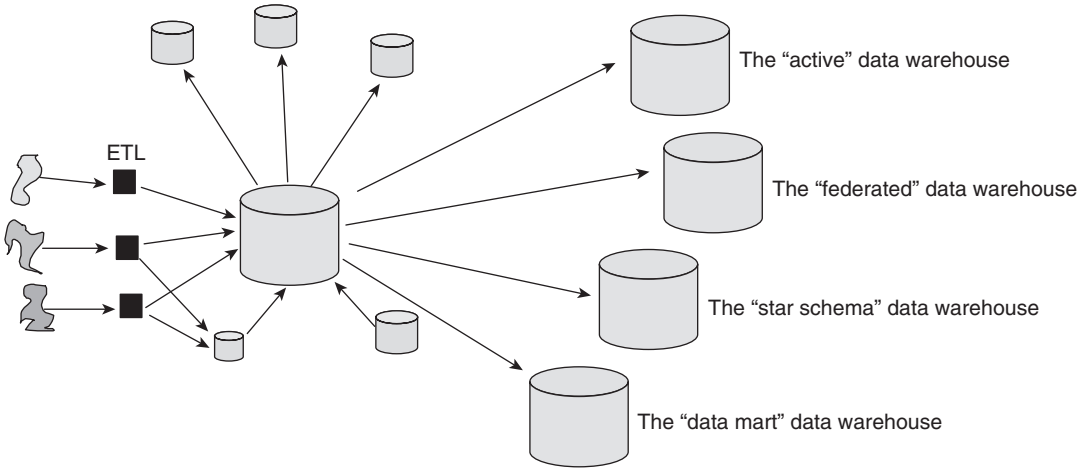
OTHER NOTIONS ABOUT A DATA WAREHOUSE

But there are forces at work in the marketplace that alter the notion of what a data warehouse is. Some vendors have recognized that a data warehouse is a very attractive thing, so they have "mutated" the concept of a data warehouse to fit their corporate needs, even though the data warehouse was never intended to do what the vendors advertised.

Some of the ways that vendors and consultants have mutated the notion of a data warehouse are shown in Figure 1.6.

Figure 1.6 shows that there are now a variety of mutant data warehouses, in particular,

- the "active" data warehouse;
- the "federated" data warehouse;



■ **FIGURE 1.6** Soon variations of the data warehouse began to emerge.

- the "star schema" data warehouse (with conformed dimensions);
- the "data mart" data warehouse.

While each of these mutant data warehouses has some similarity to what a data warehouse really is, there are some major differences between a data warehouse and these variants. And with each mutant variation come some major drawbacks.

THE ACTIVE DATA WAREHOUSE

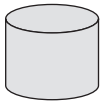
The active data warehouse is one in which online processing and update can be done. High-performance transaction processing is a feature of the active data warehouse.

Some of the drawbacks of the active data warehouse include:

- Difficulty maintaining data and transaction integrity: When a transaction does not execute properly and needs to be backed out, there is the problem of finding the data that needs to be plucked out and either destroyed or corrected. While such activity can be done, it is usually complex and requires considerable resources.
- Capacity: To ensure good online response time, there must be enough capacity to ensure that system resources will be available during peak period processing time. While this certainly can be done, the result is usually massive amounts of capacity sitting unused for long periods of time. This results in a very high cost of operation.

- Statistical processing: There is always a problem with heavy statistical processing conflicting with systems resource utilization for standard data warehouse processing. Unfortunately, the active data warehouse vendors claim that using their technology eliminates this problem.
- Cost: The active data warehouse environment is expensive for a myriad of reasons—from unused capacity waiting for peak period processing to the notion that all detail should be stored in a data warehouse, even when the probability of access of that data has long since diminished.

Figure 1.7 outlines some of the shortcomings of the active data warehouse approach.



■ **FIGURE 1.7** The active data warehouse.

THE FEDERATED DATA WAREHOUSE APPROACH

In the federated data warehouse approach, there is no data warehouse at all, usually because the corporation fears the work required to integrate data. The idea is to magically create a data warehouse by gluing together older legacy data bases in such a way that those data bases can be accessed simultaneously.

The federated approach is very appealing because it appears to give the corporation the option of avoiding the integration of data. The integration of old legacy systems is a big and complex task wherever and whenever it is done. There simply is no way around having to do integration of older data unless you use the federated approach.

But, unfortunately, the federated approach is more of an illusion than a solution. There are *many* fundamental problems with the federated approach, including but not limited to:

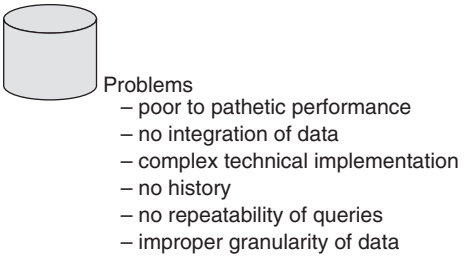
- Poor to pathetic performance: There are many reasons federating data can lead to poor performance. What if a data base that

needs to be joined in the federation is down or is currently being reorganized? What if the data base needed for federation is participating in OLTP? Are there enough machine cycles to service both the OLTP needs and the federated needs of the data base? What if the same query is executed twice, or the same data from different queries is needed? It has to be accessed and federated each time it is needed, and this can be a wasteful approach to the usage of resources. This is just the short list of why performance is a problem in the federated environment.

- Lack of data integration: There is no integration of data with the federated approach. If the data that is being federated has already been integrated, fine; but that is seldom the case. Federation knows little to nothing about the integration of data. If one file has dollars in U.S. dollars, another file has dollars in Canadian dollars, and a third file has dollars in Australian dollars, the dollars are all added up during federation. The fundamental aspects of data integration are a real and serious issue with the federated approach.
- Complex technical mechanics: However federation is done, it requires a complex technical mechanism. For example, federation needs to be done across the DBMS technology of multiple vendors. Given that different vendors do not like to cooperate with each other, it is no surprise that the technology that sits on top of them and merges them together is questionable technology.
- Limited historical data: The only history available to the federated data base is that which already exists in the federated data warehouse data bases. In almost every case, these data bases jettison history as fast as they can, in the quest for better performance. Therefore, usually only a small amount of historical data ever finds its way into the federated data warehouse.
- Nonreplicable data queries: There is no repeatability of queries in the federated environment. Suppose a query is submitted at 10:00 AM against data base ABC and returns a value of $156.09. Then, at 10:30 AM a customer comes in and adds a payroll deposit to his account, changing the value of the account to $2971.98. At 10:45 AM the federated query is repeated. A different value is returned for the same query in less than an hour.
- Inherited data granularity: Federated data warehouse users are stuck with whatever granularity is found in the applications

that support the federated query. As long as the granularity that already exists in the data base that will support federation is what users want, there is no problem. But as soon as a user wants a different level of data granularity—either higher or lower—a serious fundamental problem arises.

Figure 1.8 summarizes the fundamental problems with the federated approach to data warehousing.



■ **FIGURE 1.8** The federated data warehouse.

THE STAR SCHEMA APPROACH

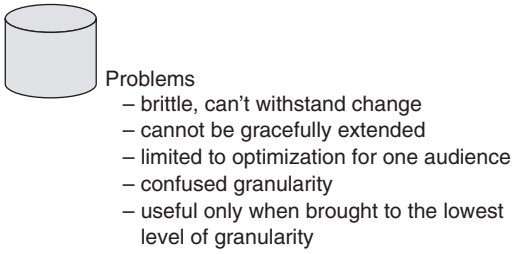
The star schema approach to data warehousing requires the creation of fact tables and dimension tables. With the star schema approach, many of the benefits of a real data warehouse can be achieved. There are, nevertheless, some fundamental problems with the star schema approach to data warehousing, including:

- Brittleness: Data warehouses that consist of a collection of star schemas tend to be "brittle." As long as requirements stay exactly the same over time, there is no problem. But as soon as the requirements change, as they all do over time, either massive changes to the existing star schema must be made or the star schema must be thrown away and replaced with a new one. The truth is that star schemas are designed for a given set of requirements and only that given set of requirements.
- Limited extensibility: Similarly, star schemas are not easy to extend. They are pretty much limited by the requirements upon which they were originally based.
- One audience: Star schemas are optimized for the use of one audience. Usually one set of people find a given star schema optimal, while everyone else finds it less than optimal. The essential mission of a data warehouse is to satisfy a large and diverse audience.

A data warehouse is suboptimal if there are people who are served in a less than satisfactory manner, and creating a single star schema to serve all audiences does just that.

- Star schema proliferation: Because a single star schema does not optimally satisfy the needs of a large community of users, it is common practice to create multiple star schemas. When multiple star schemas are created, inevitably there is a different level of granularity for each one and data integrity comes into question. This proliferation of star schemas makes looking for an enterprise view of data almost impossible.
- Devolution: To solve the problem of multiple granularity across different stars, the data in each star schema must be brought to the lowest level of granularity. This (1) defeats the theory of doing star schemas in the first place and (2) produces a classical relational design, something the star schema designer finds abhorrent.

Figure 1.9 shows the challenges of using a star schema for a data warehouse.



■ **FIGURE 1.9** The star schema data warehouse.

Star schemas are not very good in the long run for a data warehouse. When there are a lot of data and a lot of users, with a lot of diversity, the only way to make the star schema approach work is to make sure the data is nonredundant and model it at the lowest level of granularity. Even then, when the requirements change, the star schema may still have to be revised or replaced.

However, in an environment in which there is little diversity in the way users perceive data, if the requirements do not change over time, and if there aren't many users, then it is possible that a star schema might serve as a basis for a data warehouse.

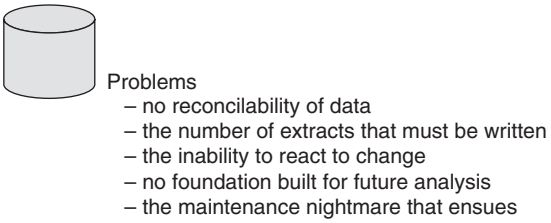
THE DATA MART DATA WAREHOUSE

Many vendors of online application processing (OLAP) technology are enamored of the data mart as a data warehouse approach. This gives the vendor the opportunity to sell his/her products first, without having to go through the building of a real data warehouse. The typical OLAP vendor's sales pitch goes—"Build a data mart first, and turn it into a data warehouse later."

Unfortunately there are many problems with building a bunch of data marts and calling them a data warehouse. Some of the problems are:

- No reconcilability of data: When management asks the question "What were last month's revenues?" accounting gives one answer, marketing gives another, and finance produces yet another number. Trying to reconcile why accounting, marketing, and sales do not agree is very difficult to do.
- Extract proliferation: When data marts are first built, the number of extracts against the legacy environment is acceptable, or at least reasonable. But as more data marts are added, more legacy data extracts have to be added. At some point the burden of extracting legacy data becomes unbearable.
- Change propagation: When there are multiple data marts and changes have to be made, those changes echo through all of the data marts. If a change has to be made in the finance data mart, it probably has to be made again in the sales data mart, then yet again in the marketing data mart, and so forth. When there are many data marts, changes have to be made in multiple places. Furthermore, those changes have to be made in the same way. It will not do for the finance data mart to make the change one way and the sales data mart to make the same change in another way. The chance of error quickly grows at an ever-increasing rate. The management, time, money, and discipline required to ensure that the propagating changes are accurately and thoroughly completed are beyond many organizations.
- Nonextensibility: When the time comes to build a new data mart, unfortunately most must be built from scratch. There is no realistic way for the new data mart to leverage much or even any of the work done by previously built data marts.

All of these factors add up to the fact that with the data mart as a data warehouse approach, the organization is in for a maintenance nightmare.



■ **FIGURE 1.10** The data mart data warehouse.

Figure 1.10 depicts the problems with the data mart as a data warehouse approach.

It is an interesting note that it is not possible to build a data mart and then grow it into a data warehouse. The DNA of a data mart is fundamentally different from the DNA of a data warehouse. The theory of building a data mart and having it grow up to be a data warehouse is akin to asserting that if you plant some tumbleweeds, when they grow up they will be oak trees. The DNA of a tumbleweed and the DNA of an oak tree are different. To get an oak tree, one must plant acorns. When planted, tumbleweed seeds yield tumbleweeds. The fact that in the springtime, when they first start to grow, oak trees and tumbleweeds both appear as small green shoots is merely a coincidence.

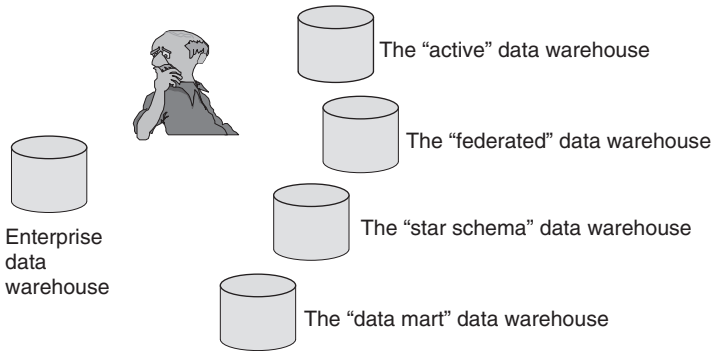
Likewise, while there are some similarities between a data mart and a data warehouse, thinking that one is the other, or that one can mutate into the other, is a mistake.

BUILDING A "REAL" DATA WAREHOUSE

The developer has some important choices to make at the architectural level and at the outset of the data warehouse process. The primary choice is, what kind of data warehouse needs to be built—a "real" data warehouse or one of the mutant varieties of a data warehouse? This choice is profound, because the financial and human resource cost of building any data warehouse is usually quite high. If the developer makes the wrong choice, it is a good bet that a lot of costly work is going to have to be redone at a later point in time. No one likes to waste a lot of resources, and few can afford to do so.

Figure 1.11 shows the dilemma facing the manager and/or architect.

One of the problems with making the choice is that the vendors who are selling data warehousing products are very persuasive. Their No. 1 goal is to convince prospective clients to build the type



■ **FIGURE 1.11** Both the long-term and the short-term ramifications are very important to the organization.

of data warehouse that requires their products and services, not necessarily the type of data warehouse that meets the business's needs. Unfortunately, falling for this type of sales pitch can cost a lot of money and waste a lot of time.

SUMMARY

Data warehousing has come a long way since the frustrating days when user data was limited to operational application data that was accessible only through an IT department intermediary. Data warehousing has evolved to meet the needs of end users who need integrated, historical, granular, flexible, accurate information.

The first-generation data warehouse evolved to include disciplined data ETL from legacy applications in a granular, historical, integrated data warehouse. With the growing popularity of data warehousing came numerous changes—volumes of data, a spiral development approach, heuristic processing, and more. As the evolution of data warehousing continued, some mutant forms emerged:

- Active data warehousing
- Federated data warehousing
- Star schema data warehouses
- Data mart data warehouses

While each of these mutant forms of data warehousing has some advantages, they also have introduced a host of new and significant disadvantages. The time for the next generation of data warehousing has come.