Data Warehousing

Learning Objectives

- Understand the basic definitions and concepts of data warehouses
- Understand data warehousing architectures
- Describe the processes used in developing and managing data warehouses
- Explain data warehousing operations
- Explain the role of data warehouses in decision support
- Explain data integration and the extraction, transformation, and load (ETL) processes
- Describe real-time (active) data warehousing
- Understand data warehouse administration and security issues

The concept of data warehousing has been around since the late 1980s. This chapter provides the foundation for an important type of database, called a data warehouse, that is used increasingly for decision support and provides improved analytical capabilities. We discuss data warehousing in the following sections:

5.1 OPENING VIGNETTE: CONTINENTAL AIRLINES FLIES HIGH WITH ITS REAL-TIME DATA WAREHOUSE

As business intelligence (BI) becomes a critical component of daily operations, real-time data warehouses that provide end users with rapid updates and alerts generated from transactional systems are increasingly being deployed. Real-time data warehousing and BI, supporting its aggressive Go Forward business plan, have helped Continental

Airlines alter its industry status from "worst to first" and then from "first to favorite."

Continental Airlines is a leader in real-time BI. In 2004, Continental won the Data Warehousing Institute's Best Practices and Leadership Award.

BIG PROBLEMS

Continental Airlines was founded in 1934, with a single-engine Lockheed aircraft in the Southwestern U.S. As of 2006, Continental (Houston) is the fifth largest airline in the United States and the seventh largest in the world. Continental has the broadest global route network of any U.S. airline, with more than 2,300 daily departures to more than 227 destinations.

Back in 1994, Continental was in deep financial trouble. It had filed for Chapter 11 bankruptcy protection twice and was heading for its third, and probably final, bankruptcy. Ticket sales were hurting because performance on factors that are important to customers was dismal, including a low percentage of on-time departures, frequent baggage arrival problems, and too many customers turned away due to overbooking.

SOLUTION

The revival of Continental began in 1994, when Gordon Bethune became CEO and initiated the Go Forward plan, which consisted of four interrelated parts to be implemented simultaneously. Bethune targeted the need to improve customer-valued performance measures by better understanding customer needs as well as customer perceptions of the value of services that were and could be offered. Financial management practices were also targeted for a significant overhaul. As early as 1998, the airline had separate databases for marketing and operations, all hosted and managed by outside vendors. Processing queries and instigating marketing programs to its high-value customers were time-consuming and ineffective. In additional, information that the workforce needed to make quick decisions was simply not available. In 1999, Continental chose to integrate its marketing, IT, revenue, and operational data sources into a single, in-house, enterprise data warehouse (EDW). The data warehouse provided a variety of early, major benefits.

As soon as Continental returned to profitability and ranked first in the airline industry in many performance metrics, Bethune and his management team raised the bar by escalating the vision. Instead of just performing best, they wanted Continental to be their customers' favorite airline. The Go Forward plan established more actionable ways to move from first to favorite among customers. Technology became increasingly critical for supporting these new initiatives. In the early days, having access to historical, integrated information was sufficient. This produced substantial strategic value. But it became increasingly imperative for the data warehouse to provide real-time, actionable information to support enterprise-wide tactical decision making and business processes.

Luckily, the warehouse team had expected and arranged for the real-time shift. From the very beginning, the team had created an architecture to handle real-time data feeds into the warehouse, extracts of data from legacy systems into the warehouse, and tactical queries to the warehouse that required almost immediate response times. In 2001, real-time data became available from the warehouse, and the amount stored grew rapidly. Continental moves real-time data (ranging from to-the-minute to hourly) about customers, reservations, check-ins, operations, and flights from its main operational systems to the warehouse. Continental's real-time applications include the following:

- Revenue management and accounting
- Customer relationship management (CRM)

Copyrighted ma

- Crew operations and payroll
- Security and fraud
- Flight operations

BENEFITS

In the first year alone, after the data warehouse project was deployed, Continental identified and eliminated over $7 million in fraud and reduced costs by $41 million. With a $30 million investment in hardware and software over six years, Continental has reached over $500 million in increased revenues and cost savings in marketing, fraud detection, demand forecasting and tracking, and improved data center management. The single, integrated, trusted view of the business (i.e., the single version of the truth) has led to better, faster decision making.

Continental is now identified as a leader in real-time BI, based on its scalable and extensible architecture, practical decisions on what data are captured in real-time, strong relationships with end users, a small and highly competent data warehouse staff, sensible weighing of strategic and tactical decision support requirements, understanding of the synergies between decision support and operations, and changed business processes that use real-time data. (For a sample output screen from the Continental system, see **teradata.com/t/page/139245/**.)

Sources: Adapted from H. Wixom, J. Hoffer, R. Anderson-Lehman, and A. Reynolds, "Real-Time Business Intelligence: Best Practices at Continental Airlines," *Information Systems Management Journal*, Winter 2006, pp. 7–18; R. Anderson-Lehman, H. Watson, B. Wixom, and J. Hoffer, "Continental Airlines Flies High with Real-Time Business Intelligence," *MIS Quarterly Executive*, Vol. 3, No. 4, December 2004, pp. 163–176 (available at **teradatauniversitynetwork.com**); H. Watson, "Real Time: The Next Generation of Decision-Support Data Management," *Business Intelligence Journal*, Vol. 10, No. 3, 2005, pp. 4–6; M. Edwards, "2003 Best Practices Awards Winners: Innovators in Business Intelligence and Data Warehousing," *Business Intelligence Journal*, Fall 2003, pp. 57–64; R. Westervelt, *Continental Airlines Builds Real-Time Data Warehouse*, August 20, 2003, **searchoracle.techtarget.com**; R. Clayton, "Enterprise Business Performance Management: Business Intelligence + Data Warehouse = Optimal Business Performance," *Teradata Magazine*, September 2005, **teradata.com/t/page/139245/**; and The Data Warehousing Institute, *2003 Best Practices Summaries: Enterprise Data Warehouse*, 2003, **tdwi.org/display.aspx?ID=6749**.

Questions for the Opening Vignette

1. Describe the benefits of implementing the Continental Go Forward strategy.
2. Explain why it is important for an airline to use a real-time data warehouse.
3. Examine the sample system output screen at **teradata.com/t/page/139245/**. Describe how it can assist the user in identifying problems and opportunities.
4. Identify the major differences between the traditional data warehouse and a real-time data warehouse, as was implemented at Continental.
5. What strategic advantage can Continental derive from the real-time system as opposed to a traditional information system?

WHAT WE CAN LEARN FROM THIS VIGNETTE

The opening vignette illustrates the strategic value of implementing a data warehouse, along with its supporting BI methods. Continental was able to move from being one of the worst-ranked airlines, in bankruptcy, to a top-notch carrier in a short time, generating significant increases in revenue and reductions in costs. The cost reductions in the first full year of deployment more than covered the six-year investment in the system. The data warehouse integrated various databases throughout the organization into a

single, in-house enterprise unit to generate a single version of the truth for the airline, putting all employees on the same page. Furthermore, the data were made available in real-time to the decision makers who needed them, so they could use them in their decision making, ultimately leading to a strategic competitive advantage in the industry. The key lesson here is that a real-time, enterprise-level data warehouse combined with a strategy for its use in decision support can leverage data to provide massive financial benefits for an organization.

Using real-time data warehousing in conjunction with decision support systems (DSS) and BI tools is an important way to conduct business processes. The opening vignette demonstrates a scenario in which a real-time data warehouse supported decision making, through analyzing large amounts of data from various sources to provide rapid results to support critical processes. With real-time data flows, Continental can view the current state of its business and identify problems, which is the first step toward solving problems analytically. In addition, customers can obtain real-time status on flights and other account information, so the system also provides a significant competitive advantage over competitors.

Decision makers require concise, dependable information about current operations, trends, and changes. Data are often fragmented in distinct operational systems, so managers often make decisions with partial information, at best. Data warehousing cuts through this obstacle by accessing, integrating, and organizing key operational data in a form that is consistent, reliable, timely, and readily available, where needed.

WHAT IS A DATA WAREHOUSE?

In simple terms, a **data warehouse (DW)** is a pool of data produced to support decision making; it is also a repository of current and historical data of potential interest to managers throughout the organization. Data are usually structured to be available in a form ready for analytical processing activities (e.g., online analytical processing [OLAP], data mining, querying, reporting, other decision support applications). A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process.

CHARACTERISTICS OF DATA WAREHOUSING

A common way of introducing data warehousing is to refer to its fundamental characteristics (see Inmon, 2005):

- *Subject oriented.* Data are organized by detailed subject, such as sales, products, or customers, containing only information relevant for decision support. Subject orientation enables users to determine not only how their business is performing but why. A data warehouse differs from an operational database in that most operational databases have a product orientation and are tuned to handle transactions that update the database. Subject orientation provides a more comprehensive view of the organization.
- *Integrated.* Integration is closely related to subject orientation. Data warehouses must place data from different sources into a consistent format. To do so, they must deal with naming conflicts and discrepancies among units of measure. A data warehouse is presumed to be totally integrated.

- *Time variant (time series).* A warehouse maintains historical data. The data do not necessarily provide current status (except in real-time systems). They detect trends, deviations, long-term relationships for forecasting and comparisons, leading to decision making. There is a *temporal* quality to every data warehouse. Time is the one important dimension that all data warehouses must support. Data for analysis from multiple sources contain multiple time points (e.g., daily, weekly, monthly views).
- *Nonvolatile.* After data are entered into a data warehouse, users cannot change or update the data. Obsolete data are discarded, and changes are recorded as new data. This enables the data warehouse to be tuned almost exclusively for data access.

Some additional characteristics may include the following:

- *Web based.* Data warehouses are typically designed to provide an efficient computing environment for Web-based applications.
- *Relational/multidimensional.* A data warehouse uses either a relational structure or a multidimensional structure.
- *Client/server.* A data warehouse uses the client/server architecture to provide easy access for end users.
- *Real-time.* Newer data warehouses provide real-time, or active, data access and analysis capabilities (see Basu, 2003; and Bonde and Kuckuk, 2004).
- *Include metadata.* A data warehouse contains metadata (data about data) about how the data are organized and how to effectively use them.

Whereas a data warehouse is a repository of data, *data warehousing* is literally the entire process (see Watson, 2002). Data warehousing is a *discipline* that results in applications that provide decision support capability, allows ready access to business information, and creates business insight. There are three main types of data warehouses: data marts, operational data stores (ODS), and EDW. In addition to discussing these next, we also discuss metadata.

DATA MARTS

Whereas a data warehouse combines databases across an entire enterprise, a data mart is usually smaller and focuses on a particular subject or department. A **data mart** is a subset of a data warehouse, typically consisting of a single subject area (e.g., marketing, operations). A data mart can be either *dependent* or *independent*. A **dependent data mart** is a subset that is created directly from the data warehouse. It has the advantages of using a consistent data model and providing quality data. Dependent data marts support the concept of a single enterprise-wide data model, but the data warehouse must be constructed first. A dependent data mart ensures that the end user is viewing the same version of the data that are accessed by all other data warehouse users. The high cost of data warehouses limits their use to large companies. As an alternative, many firms use a lower-cost, scaled-down version of a data warehouse referred to as an *independent data mart*. An **independent data mart** is a small warehouse designed for a strategic business unit (SBU) or a department, but its source is not an EDW.

OPERATIONAL DATA STORES

An **operational data store (ODS)** provides a fairly recent form of customer information file (CIF). This type of database is often used as an interim staging area for a data warehouse. Unlike the static contents of a data warehouse, the contents of an ODS are updated

OPERATIONAL DATA STORES

An **operational data store (ODS)** provides a fairly recent form of customer information file (CIF). This type of database is often used as an interim staging area for a data warehouse. Unlike the static contents of a data warehouse, the contents of an ODS are updated

through the course of business operations. An ODS is used for short-term decisions involving mission-critical applications rather than for the medium- and long-term decisions associated with an EDW. An ODS is similar to short-term memory in that it stores only very recent information. In comparison, a data warehouse is like long-term memory because it stores permanent information. An ODS consolidates data from multiple source systems and provides a near-real-time, integrated view of volatile, current data. The ETL processes (discussed later in this chapter) for an ODS are identical to those for a data warehouse. Finally, **oper marts** (see Imhoff, 2001) are created when operational data need to be analyzed multidimensionally. The data for an oper mart come from an ODS.

ENTERPRISE DATA WAREHOUSES (EDW)

An **enterprise data warehouse (EDW)** is a large-scale data warehouse that is used across the enterprise for decision support. It is the type of data warehouse that Continental developed, as described in the opening vignette. The large-scale nature provides integration of data from many sources into a standard format for effective BI and decision support applications. EDW are used to provide data for many types of DSS, including CRM, supply-chain management (SCM), business performance management (BPM), business activity monitoring (BAM), product lifecycle management (PLM), revenue management, and sometimes even knowledge management systems (KMS). For an example in practice, see MindTree Consulting's case study "Building an Enterprise Data Warehousing for a Major Pharmaceutical Company," available at **mindtree.com/clt/cs_dw_pharma.html**.

METADATA

Metadata are data about data (e.g., see Sen, 2004; and Zhao, 2005). Metadata describe the structure of and some meaning about data, thereby contributing to their effective or ineffective use. Mehra (2005) indicated that few organizations really understand metadata, and fewer understand how to design and implement a metadata strategy. Metadata are generally defined in terms of usage as *technical* or *business* metadata. *Pattern* is another way to view metadata. According to the pattern view, we can differentiate between *syntactic metadata* (i.e., data describing the syntax of data), *structural metadata* (i.e., data describing the structure of the data), and *semantic metadata* (i.e., data describing the meaning of the data in a specific domain).

We next explain traditional metadata patterns and insights into how to implement an effective metadata strategy via a holistic approach to enterprise metadata integration. The approach includes ontology and metadata registries; enterprise information integration (EII); extraction, transformation, and load (ETL); and service-oriented architectures (SOA). Effectiveness, extensibility, reusability, interoperability, efficiency and performance, evolution, entitlement, flexibility, segregation, user interface, versioning, versatility, and low maintenance cost are some of the key requirements for building a successful metadata-driven enterprise.

According to Kassam (2002), *business metadata* comprise information that increases our understanding of traditional (i.e., structured) data. The primary purpose of metadata should be to provide context to the reported data; that is, it provides enriching information that leads to the creation of knowledge. Business metadata, though difficult to provide efficiently, release more of the potential of structured data. The context need not be the same for all users. In many ways, metadata assist in the conversion of data and information into knowledge. Metadata form a foundation for a *metabusiness* architecture (see Bell, 2001). Tannenbaum (2002) described how to identify metadata requirements.

Vaduva and Vetterli (2001) provided an overview of metadata management for data warehousing. Zhao (2005) described five levels of metadata management maturity: (1) ad hoc, (2) discovered, (3) managed, (4) optimized, and (5) automated. These levels help in understanding where an organization is in terms of how and how well it uses its metadata.

The design, creation, and use of metadata—descriptive or summary data about data—and its accompanying standards may involve ethical issues. There are ethical considerations involved in the collection and ownership of the information contained in metadata, including privacy and intellectual property issues that arise in the design, collection, and dissemination stages (for more, see Brody, 2003).

Section 5.2 Review Questions

1. What is a data warehouse?
2. How is a data warehouse different from a database?
3. What is an ODS?
4. Differentiate among a data mart, an ODS, and an EDW.
5. Explain the importance of metadata.

5.3 DATA WAREHOUSING PROCESS OVERVIEW

Organizations, private and public, continuously collect data, information, and knowledge at an increasingly accelerated rate and store them in computerized systems. Maintaining and using these data and information becomes extremely complex, especially as scalability issues arise. In addition, the number of users needing to access the information continues to increase as a result of improved reliability and availability of network access, especially the Internet. Working with multiple databases, either integrated in a data warehouse or not, has become an extremely difficult task requiring considerable expertise, but it can provide immense benefits far exceeding its cost (see the opening vignette and Application Case 5.1).

Application Case 5.1

Data Warehousing Supports First American Corporation's Corporate Strategy

First American Corporation changed its corporate strategy from a traditional banking approach to one that was centered on CRM. This enabled First American to transform itself from a company that lost $60 million in 1990 to an innovative financial services leader a decade later. The successful implementation of this strategy would not have been possible without its VISION data warehouse, which stores information about customer behavior, such as products used, buying preferences, and client value positions. VISION provides:

- Identification of the top 20 percent of profitable customers
- Identification of the 40 to 50 percent of unprofitable customers
- Retention strategies

- Lower-cost distribution channels
- Strategies to expand customer relationships
- Redesigned information flows

Access to information through a data warehouse can enable both evolutionary and revolutionary change. First American Corporation achieved revolutionary change, moving itself into the "sweet 16" of financial services corporations.

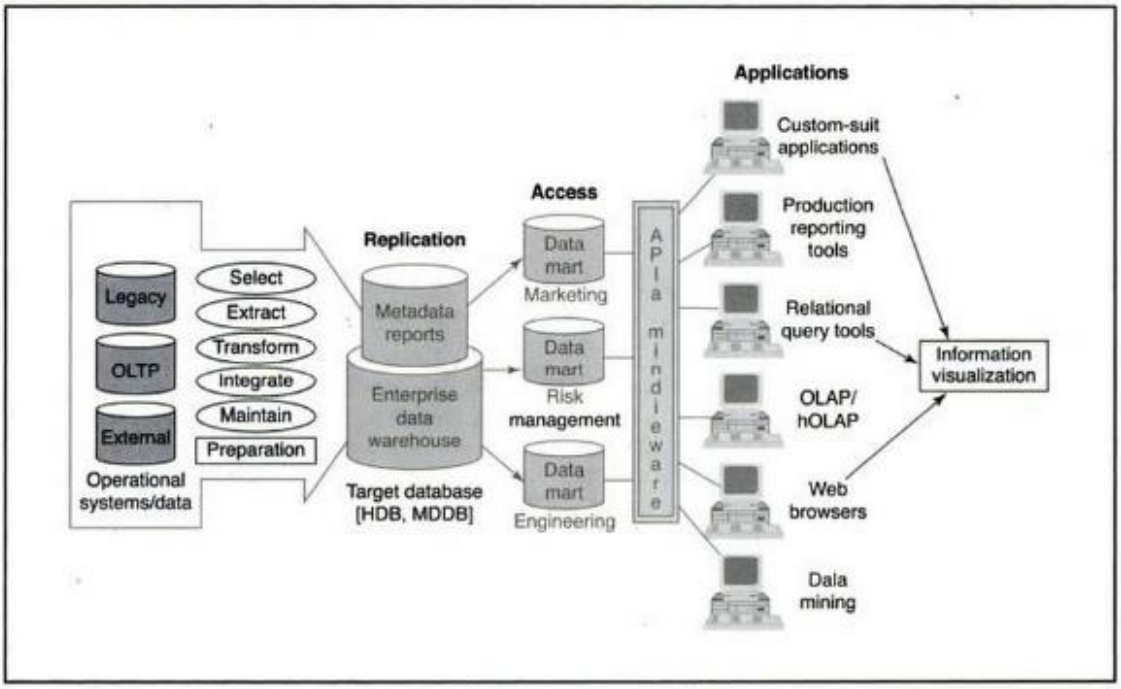
Sources: Adapted from B.L. Cooper, H.J. Watson, B.H. Wixom, and D.L. Goodhue, "Data Warehousing Supports Corporate Strategy at First American Corporation," *MIS Quarterly*, Vol. 24, No. 4, 2000, pp. 547–567; and B.L. Cooper, H.J. Watson, B.H. Wixom, and D.L. Goodhue, *Data Warehousing Supports Corporate Strategy at First American Corporation*, SIM International Conference, Atlanta, August, 15–19, 1999.

Many organizations need to create data warehouses—massive data stores of time-series data for decision support. Data are imported from various external and internal resources and are cleansed and organized in a manner consistent with the organization's needs. After the data are populated in the data warehouse, data marts can be loaded for a specific area or department. Alternatively, data marts can be created first, as needed, and then integrated into an EDW. Often, though, data marts are not developed, but data are simply loaded onto PCs or left in their original state for direct manipulation using BI tools.

In Figure 5.1, we show the data warehouse concept. These are the major components of a data warehousing process:

- *Data sources*. Data are sourced from multiple independent operational "legacy" systems and possibly from external data providers (such as the U.S. Census). Data may also come from an online transaction processing (OLTP) or enterprise

FIGURE 5.1 Data Warehouse Framework and Views



resource planning (ERP) system. Web data in the form of Web logs may also feed a data warehouse.

- *Data extraction.* Data are extracted using custom-written or commercial software called ETL.
- *Data loading.* Data are loaded into a staging area, where they are transformed and cleansed. The data are then ready to load into the data warehouse.
- *Comprehensive database.* Essentially, this is the EDW to support all decision analysis by providing relevant summarized and detailed information originating from many different sources.
- *Metadata.* Metadata are maintained so that they can be assessed by IT personnel and users. Metadata include software programs about data and rules for organizing data summaries that are easy to index and search, especially with Web tools.
- *Middleware tools.* Middleware tools enable access to the data warehouse. Power users such as analysts may write their own SQL queries. Others may employ a managed query environment, such as Business Objects, to access data. There are many front-end applications that business users can use to interact with data stored in the data repositories, including data mining, OLAP, reporting tools, and data visualization tools.

Section 5.3 Review Questions

1. Describe the data warehousing process.
2. Describe the major components of a data warehouse.
3. Identify the role of middleware tools.

5.4 DATA WAREHOUSING ARCHITECTURES

There are several basic architectures for data warehousing. Two-tier and three-tier architectures are common (see Figures 5.2 and 5.3), but sometimes there is simply one tier. Hoffer et al. (2007) distinguished among these by dividing the data warehouse into three parts:

1. The data warehouse itself, which contains the data and associated software.
2. Data acquisition (back-end) software, which extracts data from legacy systems and external sources, consolidates and summarizes them, and loads them into the data warehouse.
3. Client (front-end) software, which allows users to access and analyze data from the warehouse (a DSS/BI/business analytics [BA] engine)

In a three-tier architecture, operational systems contain the data and the software for data acquisition in one tier (i.e., the server), the data warehouse is another tier, and the third tier includes the DSS/BI/BA engine (i.e., the application server) and the client (see Figure 5.2). Data from the warehouse are processed twice and deposited in an additional multidimensional database, organized for easy multidimensional analysis and presentation, or replicated in data marts. The advantage of the three-tier architecture is its separation of the functions of the data warehouse, which eliminates resource constraints and makes it possible to easily create data marts.

In a two-tier architecture, the DSS engine physically runs on the same hardware platform as the data warehouse (see Figure 5.3). Therefore, it is more economical than the three-tier structure. The two-tier architecture can have performance problems for large data warehouses that work with data-intensive applications for decision support.

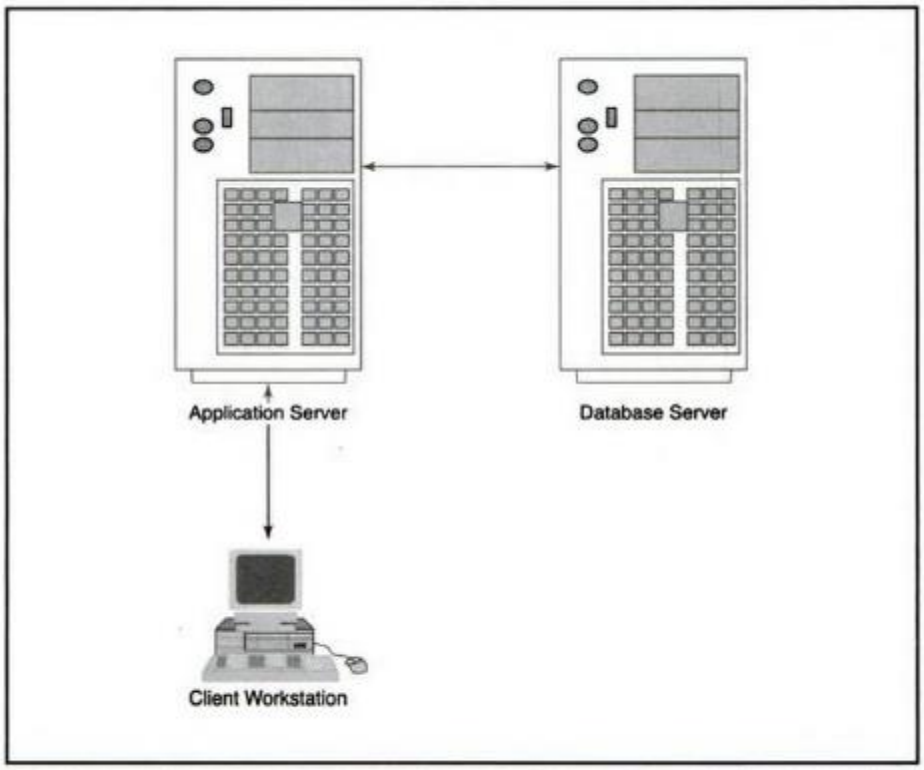


FIGURE 5.2 Architecture of a Three-Tier Data Warehouse

Much of the common wisdom assumes an absolutist approach, maintaining that one solution is better than the other, despite the organization's circumstances and unique needs. To further complicate these architectural decisions, many consultants and software vendors focus on one portion of the architecture, therefore limiting their capacity and motivation to assist an organization through the options based on its needs. But these aspects are being questioned and analyzed. For example, Ball (2005) provided decision criteria for organizations that plan to implement a BI application and have already determined their need for multidimensional data marts but need help deciding about the appropriately tiered architecture. His criteria revolve around forecasting needs for space and speed of access (see Ball, 2005, for details).

Data warehousing and the Internet are two key technologies that offer important solutions for managing corporate data. The integration of these two technologies produces Web-based data warehousing. In Figure 5.4 we show the architecture of Web-based data warehousing. The architecture is three tiered and includes the PC client, Web server, and application server. On the client side, the user needs an Internet connection and a Web browser (preferably Java enabled) through the familiar graphical user interface (GUI). The Internet/intranet/extranet is the communication medium between client and servers. On the server side, a Web server is used to manage the inflow and outflow of information between client and server. It is backed by both a data warehouse and an application server. Web-based data warehousing offers several compelling advantages, including ease of access, platform independence, and lower cost.

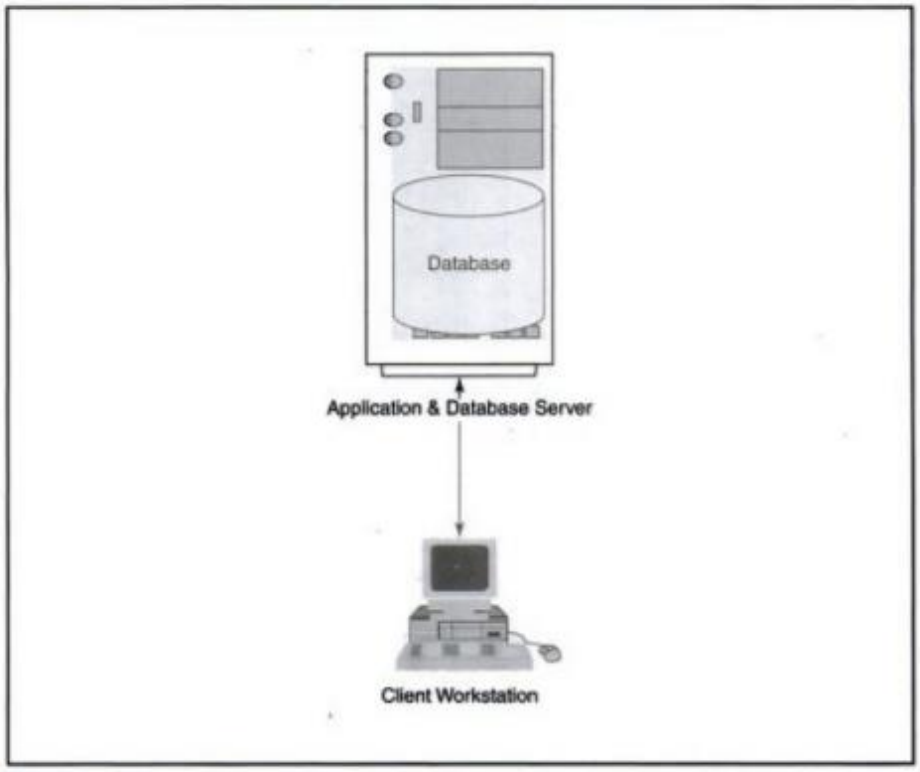
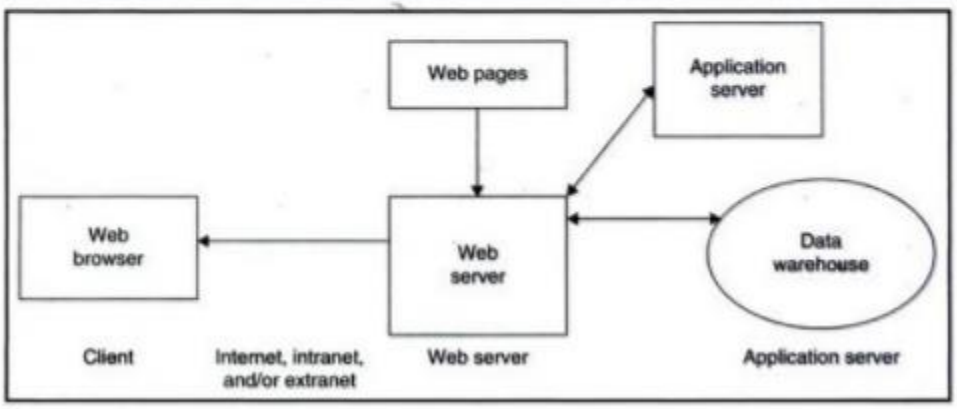


FIGURE 5.3 Architecture of a Two-Tier Data Warehouse

The Vanguard Group moved to a Web-based, three-tier architecture for its enterprise architecture to integrate all its data and provide customers with the same views of data as internal users (Dragoon, 2003). Likewise, Hilton migrated all its independent client/server systems to a three-tier data warehouse, using a Web design enterprise system. This change involved an investment of $3.8 million (excluding labor) and affected

FIGURE 5.4 Architecture of Web-Based Data Warehousing



1,500 users. It increased processing efficiency (speed) by a factor of 6. When it was deployed, Hilton expected to save between $4.5 to $5 million annually. Finally, Hilton experimented with Dell's clustering (i.e., parallel computing) technology to enhance scalability and speed (see Anthes, 2003).

Web architectures for data warehousing are similar in structure to other data warehousing architectures, requiring a design choice for housing the Web data warehouse with the transaction server or as a separate server(s). Page loading speed is an important consideration in designing Web-based applications; therefore, server capacity must be carefully planned.

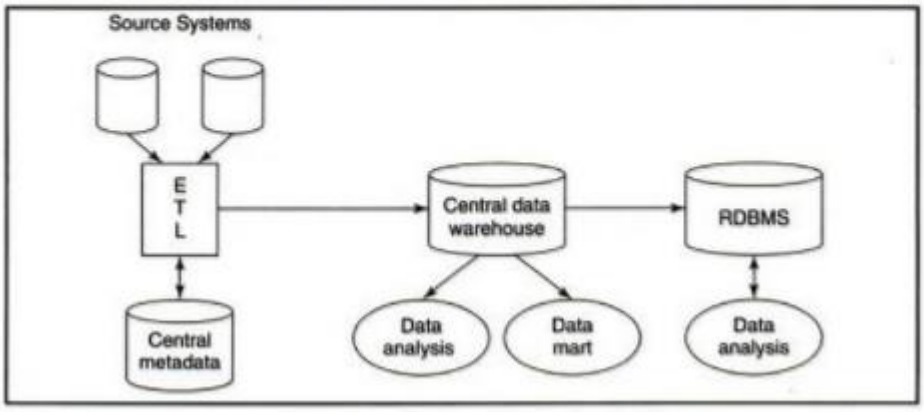
There are several issues to consider when deciding which architecture to use. Among them are the following:

- *Which database management system (DBMS) should be used?* Most data warehouses are built using relational database management systems (RDBMS). Oracle (Oracle Corporation, **www.oracle.com**), SQL Server (Microsoft Corporation, **microsoft.com/sql/**), and DB2 (IBM Corporation, **www.306.ibm.com/software/data/db2/**) are most commonly used. Each of these products supports both client/server and Web-based architectures.
- *Will parallel processing and/or partitioning be used?* Parallel processing enables multiple CPUs to process data warehouse query requests simultaneously and provides scalability. Data warehouse designers need to decide whether the database tables will be partitioned (i.e., split into smaller tables) for access efficiency and what the criteria will be. This is an important consideration that is necessitated by the large amounts of data contained in a typical data warehouse. Teradata (**teradata.com**) has successfully adopted this approach.
- *Will data migration tools be used to load the data warehouse?*
- *What tools will be used to support data retrieval and analysis?*

ALTERNATIVE ARCHITECTURES

The data warehouse architecture design viewpoints can be generally categorized into enterprise-wide data warehouse design and data mart design. In Figure 5.5 (a-e), we show some alternatives to the basic architectural design types, including a hub-and-spoke

FIGURE 5.5 Alternative Data Warehouse Architectures



5.5a Enterprise Data Warehousing Architecture

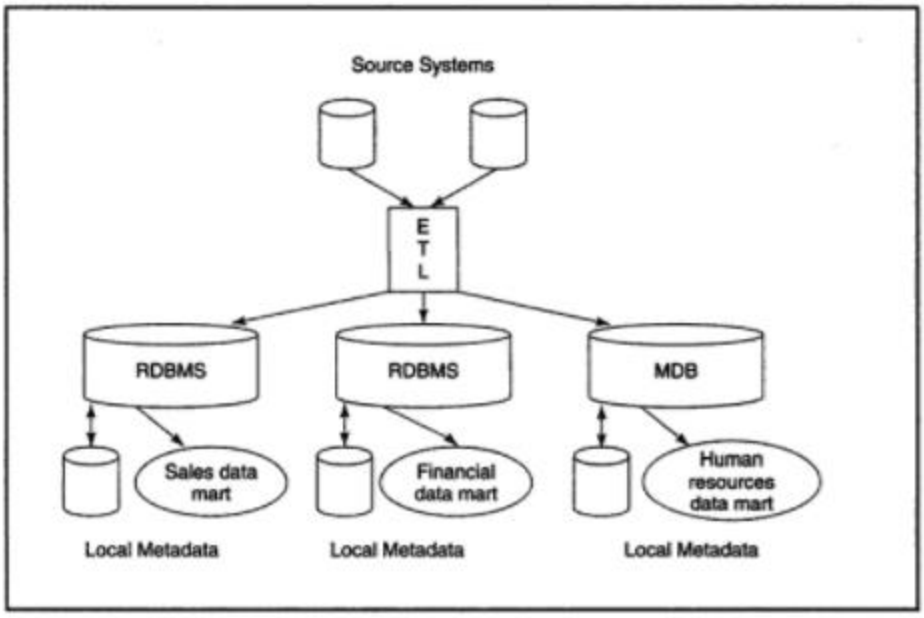


FIGURE 5.5b Data Mart Architecture

architecture, an enterprise warehouse with ODS (i.e., real-time access support), and a distributed EDW architecture. Sen and Sinha (2005) analyzed 15 different data warehousing methodologies. The sources of these methodologies are classified into three broad categories: core-technology vendors, infrastructure vendors, and information-modeling companies. See Sen and Sinha (2005) for further details.

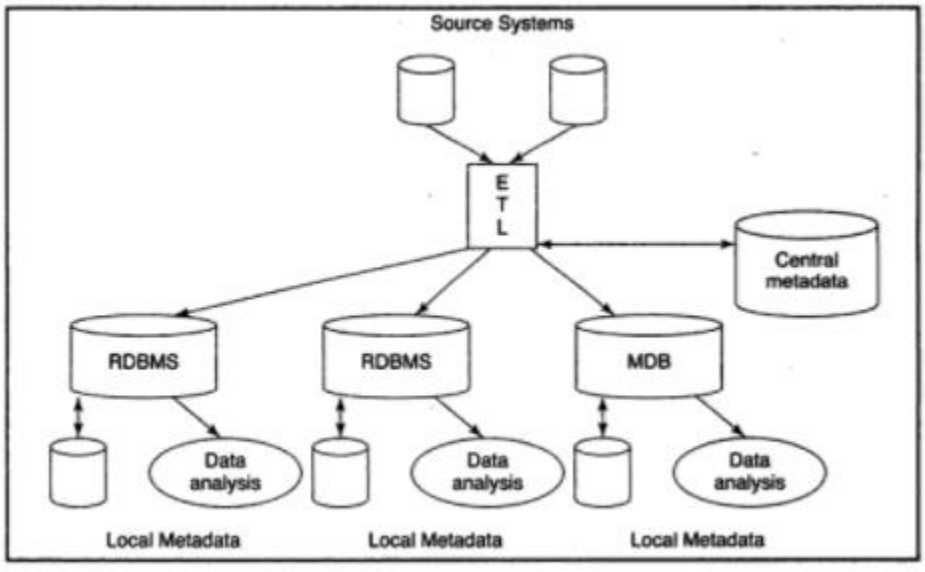


FIGURE 5.5c Hub-and-Spoke Data Mart Architecture

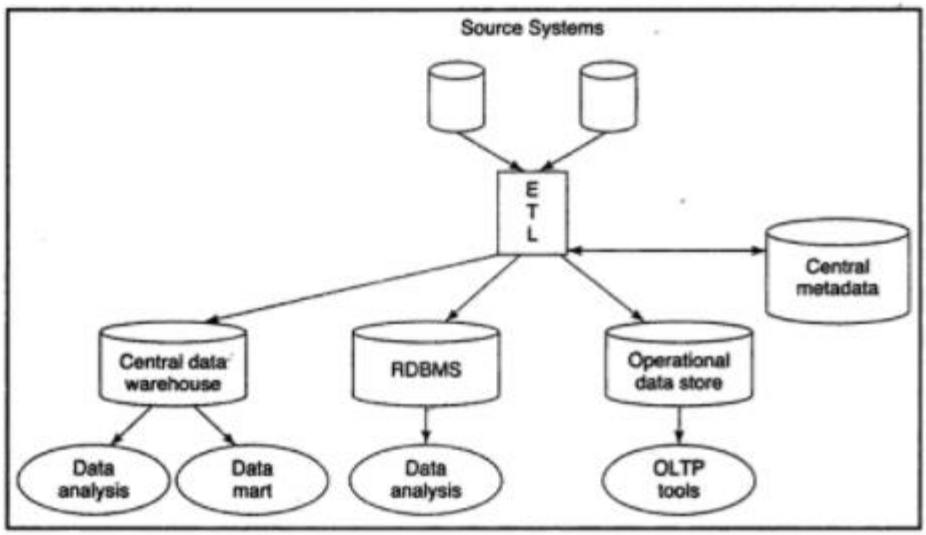


FIGURE 5.5d Enterprise Warehouse and Operational Data Store

The data warehousing literature provides additional discussions about a variety of architectures, such as independent data marts, data mart bus architecture with linked dimensional data marts, and federated data marts (see Ariyachandra and Watson, 2005, 2006a, 2006b); see Figure 5.6. In independent data marts, the marts are developed to

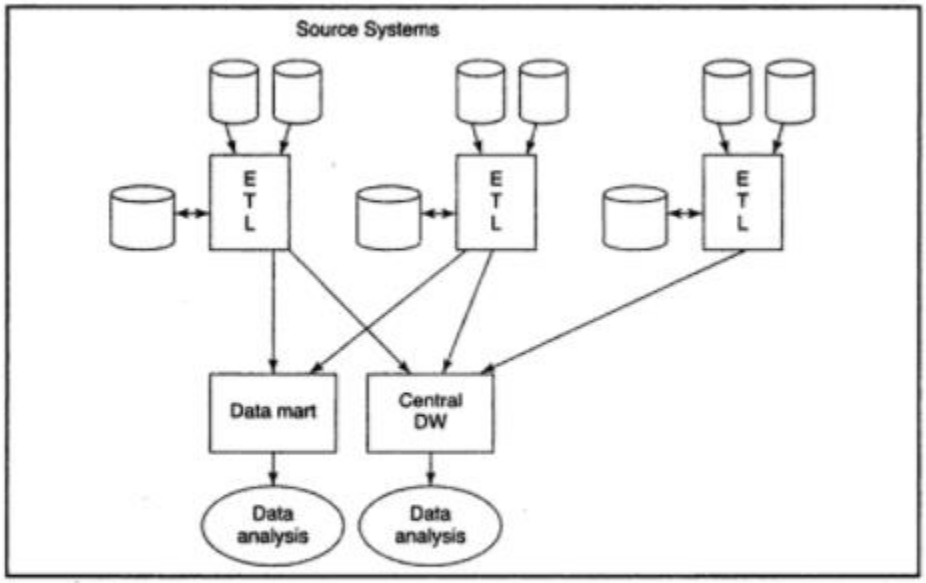
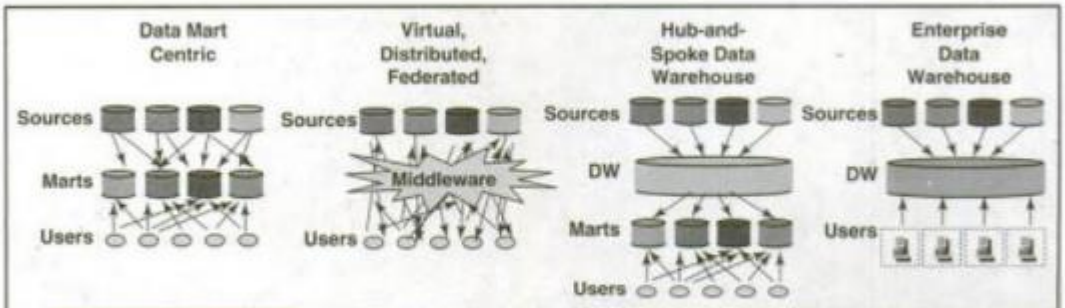


FIGURE 5.5e Distributed Data Warehouse Architecture

Source: Adapted from A. Sen and P. Sinha, "A Comparison of Data Warehousing Methodologies," *Communications of the ACM*, Vol. 48, No. 3, 2005, pp. 78–84; and T. Ariyachandra and H. Watson, "Which Data Warehouse Architecture Is Most Successful?" *Business Intelligence Journal*, Vol. 11, No. 1, First Quarter, 2006, pp. 4–6.



	Independent Data Marts	Leave Data Where it Lies	Dependent Data Marts	Centralized Integrated Data With Direct Access
Pros	• Easy to Build Organizationally • Easy to Build Technically	• No need for ETL • No need for separate platform	• Allows easier customization of user interfaces & reports	Business Enterprise view Design consistency & data quality Data reusability
Cons	• Business Enterprise view unavailable • Redundant data costs • High ETL costs • High App costs • High DBA and operational costs	• Only viable for low volume • Meta data issues • Network bandwidth and join complexity issues • Workload typically placed on workstation	• Business Enterprise view challenging • Redundant data costs • High DBA and operational costs • Data latency	Requires corporate leadership and vision

FIGURE 5.6 Alternative Architectures for Data Warehousing Efforts

Source: W. Eckerson, "Four Ways to Build a Data Warehouse," *What Works: Best Practices in Business Intelligence and Data Warehousing*, Vol. 15, The Data Warehousing Institute, Chatsworth, CA, June 2003, pp. 46–49. Used with permission.

operate independently of each another. Thus, they have inconsistent data definitions and different dimensions and measures, making it difficult to analyze data across the marts (i.e., it is difficult, if not impossible, to get to the "one version of the truth"). In a hub-and-spoke architecture, attention is focused on building a scalable and maintainable infrastructure; it is developed in an iterative way, subject area by subject area, and dependent data marts are developed. A centralized data warehouse is similar to the hub-and-spoke architecture except that there are no dependent data marts. The central data warehouses architecture, which is advocated mainly by Teradata Corp., advises using data warehouses without any data marts (see Figure 5.7). This centralized approach provides users with access to all data in the data warehouse instead of limiting them to data marts. In addition, it reduces the amount of data the technical team has to transfer or change, therefore simplifying data management and administration.

The *federated approach* is a concession to the natural forces that undermine the best plans for developing a perfect system. It uses all possible means to integrate analytical resources from multiple sources to meet changing needs or business conditions. Essentially, the federated approach involves integrating disparate systems. In a federated architecture, existing decision support structures are left in place, and data are accessed from those sources as needed. The federated approach is supported by middleware vendors that propose distributed query and join capabilities. These Extensible Markup Language (XML)–based tools offer users a global view of distributed data sources, including data warehouses, data marts, Web sites, documents, and operational

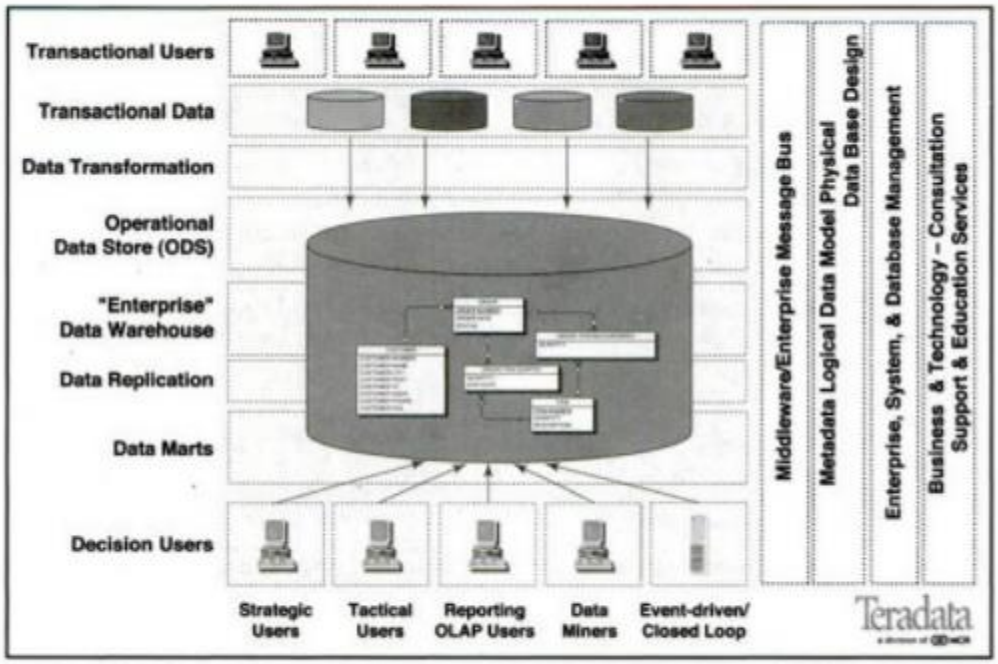


FIGURE 5.7 Teradata Corp.'s Enterprise Data Warehouse

Source: Teradata Corporation (**teradata.com**). Used with permission.

systems. When users choose query objects from this view and press the submit button, the tool automatically queries the distributed sources, joins the results, and presents them to the user. Because of performance and data quality issues, most experts agree that federated approaches work well to supplement data warehouses, not replace them (see Eckerson, 2005).

Ariyachandra and Watson (2005) identified 10 factors that potentially affect the architecture selection decision:

1. Information interdependence between organizational units
2. Upper management's information needs
3. Urgency of need for a data warehouse
4. Nature of end-user tasks
5. Constraints on resources
6. Strategic view of the data warehouse prior to implementation
7. Compatibility with existing systems
8. Perceived ability of the in-house IT staff
9. Technical issues
10. Social/political factors

These factors are similar to many success factors described in the literature for information systems projects and DSS and BI projects. Technical issues, beyond providing technology that is feasibly ready for use, is important but often not as important as behavioral issues, such as meeting upper management's information needs and user involvement in the development process (a social/political factor). Each data warehousing architecture has specific applications for which it is most (and least) effective and thus provides

maximal benefits to the organization. However, overall, the data mart structure seems to be the least effective in practice (see Ariyachandra and Watson, 2006b). See Ariyachandra and Watson (2006a) for some additional details.

Section 5.4 Review Questions

1. What are the key similarities and differences between a two-tiered architecture and a three-tiered architecture?
2. How has the Web influenced data warehouse design?
3. List the alternative data warehousing architectures discussed in this section.
4. What issues should be considered when deciding which architecture to use in developing a data warehouse? List the 10 most important factors.

5.5 DATA INTEGRATION AND THE EXTRACTION, TRANSFORMATION, AND LOAD (ETL) PROCESSES

Global competitive pressures, demand for ROI, management and investor inquiry, and government regulations are forcing business managers to rethink how they integrate and manage their businesses. A decision maker typically needs access to multiple sources of data that must be integrated. Before data warehouses, data marts, and BI software, providing access to data sources was a major, laborious process. Even with modern Web-based data management tools, recognizing what data to access and providing them to the decision maker is a nontrivial task that requires database specialists. As data warehouses grow in size, the issues of integrating data grow as well.

The needs of BA continue to evolve. Mergers and acquisitions, regulatory requirements, and the introduction of new channels can drive changes in BI requirements. In addition to historical, cleansed, consolidated, and point-in-time data, business users increasingly demand access to real-time, unstructured, and/or remote data. And everything must be integrated with the contents of an existing data warehouse (see Devlin, 2003). Moreover, access via PDAs and through speech recognition and synthesis is becoming more commonplace, further complicating integration issues (see Edwards, 2003). Many integration projects involve enterprise-wide systems. Orovic (2003) provided a checklist of what works and what does not work when attempting such a project. Properly integrating data from various databases and other disparate sources is difficult. But when it is not done properly, it can lead to disaster in enterprise-wide systems such as CRM, ERP, and supply-chain projects (see Nash, 2002). Also see Dasu and Johnson (2003).

DATA INTEGRATION

Data integration comprises three major processes that, when correctly implemented, permit data to be accessed and made accessible to an array of ETL and analysis tools and data warehousing environment: data access (i.e., the ability to access and extract data from any data source), data federation (i.e., the integration of business views across multiple data stores), and change capture (based on the identification, capture, and delivery of the changes made to enterprise data sources). See Sapir (2005) for details. See Application Case 5.2 for an example of how Bank of America benefits from implementing a data warehouse that integrates data from many sources. Some vendors, such as SAS Institute, Inc., have developed strong data integration tools. The SAS enterprise data integration server includes customer data integration tools that

improve data quality in the integration process. The Oracle Business Intelligence Suite assists in integrating data as well.

Application Case 5.2

Bank of America's Award-Winning Integrated Data Warehouse

In 2003, Bank of America won The Data Warehousing Institute's Best Practices and Leadership Award. Bank of America, one of the largest financial services networks in the United States, has realized significant operating savings by *integrating* its data warehouses. Its Teradata Warehouse is the platform for its integrated EDW. The data warehouse assists decision makers so they can:

- Maintain customer privacy
- Leverage customer information to develop products and identify trends
- Anticipate customer needs, leading to improved customer service and sales
- Lower costs, improve usage and performance, and respond quickly to changing business demands
- Make better and faster decisions

Sources: M. Edwards, "2003 Best Practices Awards Winners: Innovators in Business Intelligence and Data Warehousing," *Business Intelligence Journal*, Fall 2003, pp. 57–64; NCR, *Bank of America Expands Teradata Data Warehouse System*, October 6, 2005, **ncr.com/en/media_information/2005/oct/pr100605a.htm** (accessed April 2006); and Teradata, *Bank of America Expands Teradata Data Warehouse System*, October 6, 2005, **teradata.com/t/page/141826/index.html** (accessed April 2006).

A major purpose of a data warehouse is to integrate data from multiple systems. Various integration technologies enable data and metadata integration today:

A major purpose of a data warehouse is to integrate data from multiple systems. Various integration technologies enable data and metadata integration today:

- Enterprise application integration (EAI)
- Service-oriented architecture (SOA)
- Enterprise information integration (EII)
- Extraction, transformation, and load (ETL)

Enterprise application integration (EAI) provides a vehicle for pushing data from source systems into the data warehouse. It involves integrating application functionality and is focused on sharing functionality (rather than data) across systems, thereby enabling flexibility and reuse. Traditionally, EAI solutions have focused on enabling application reuse at the application programming interface (API) level. Recently, EAI is accomplished by using SOA coarse-grained services (a collection of business processes or functions) that are well defined and documented. Using Web services is a specialized way of implementing an SOA. EAI can be used to facilitate data acquisition directly into a near-real-time data warehouse or to deliver decisions to the OLTP systems. There are many different approaches to and tools for EAI implementation.

Enterprise information integration (EII) is an evolving tool space that promises real-time data integration from a variety of sources, such as relational databases, Web services, and multidimensional databases. It is a mechanism for pulling data from source systems to satisfy a request for information. EII tools use predefined metadata to populate views that make integrated data appear relational to end users. XML may be the most important aspect of EII because XML allows data to be tagged either at creation time or later. These tags can be extended and modified to accommodate almost any area of knowledge (see Kay, 2005).

Physical data integration has conventionally been the main mechanism for creating an integrated view with data warehouses and data marts. With the advent of EII tools (see Kay, 2005), new virtual data integration patterns are feasible. Manglik and Mehra (2005) discussed the benefits and constraints of new data integration patterns that can expand traditional physical methodologies to present a comprehensive view for the enterprise.

We next turn to the approach for loading data into the warehouse: ETL.

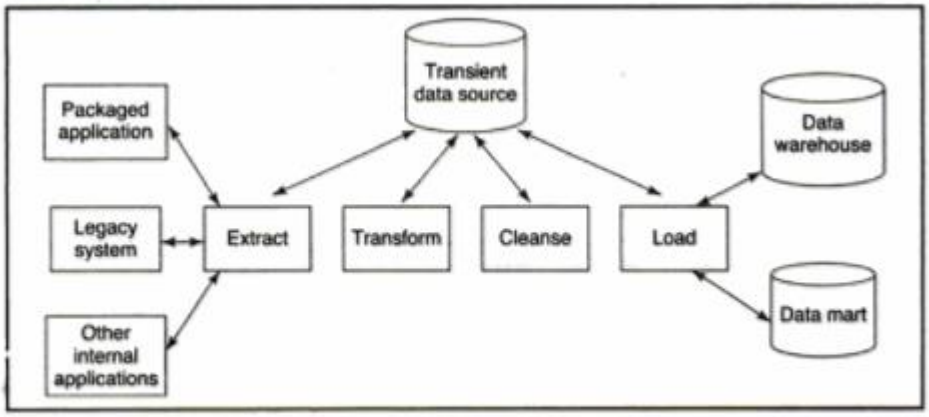
EXTRACTION, TRANSFORMATION, AND LOAD

At the heart of the technical side of the data warehousing process is **extraction, transformation, and load (ETL)**. ETL technologies, which have existed for some time, are instrumental in the process and use of data warehouses. The ETL process is an integral component in any data-centric project. IT managers are often faced with challenges because the ETL process typically consumes 70 percent of the time in a data-centric project.

The ETL process consists of *extraction* (i.e., reading data from one or more databases), *transformation* (i.e., converting the extracted data from its previous form into the form in which it needs to be so that it can be placed into a data warehouse or simply another database), and *load* (i.e., putting the data into the data warehouse). Transformation occurs by using rules or lookup tables or by combining the data with other data. The three database functions are integrated into one tool to pull data out of one or more databases and place them into another, consolidated database or a data warehouse.

ETL tools also transport data between sources and targets, document how data elements (e.g., metadata) change as they move between source and target, exchange metadata with other applications as needed, and administer all runtime processes and operations (e.g., scheduling, error management, audit logs, statistics). ETL is extremely important for data integration as well as for data warehousing. The purpose of the ETL process is to load the warehouse with integrated and cleansed data. The data used in ETL processes can come from any source: a mainframe application, an ERP application, a CRM tool, a flat file, an Excel spreadsheet, or even a message queue. In Figure 5.8, we outline the ETL process.

FIGURE 5.8 The ETL Process



Source: Adapted from M.L. Songini, "ETL Quickstudy," *Computer World*, February 2, 2004; and T. Ariyachandra and H. Watson, "Which Data Warehouse Architecture Is Most Successful?" *Business Intelligence Journal*, Vol. 11, No. 1, First Quarter, 2006, pp. 4–6.

The process of migrating data to a data warehouse involves the extraction of data from *all* relevant sources. Data sources may consist of files extracted from OLTP databases, spreadsheets, personal databases (e.g., Microsoft Access), or external files. Typically, all the input files are written to a set of staging tables, which are designed to facilitate the load process. A data warehouse contains numerous business rules that define such things as how the data will be used, summarization rules, standardization of encoded attributes, and calculation rules. Any data quality issues pertaining to the source files need to be corrected before the data are loaded into the data warehouse. One of the benefits of a well-designed data warehouse is that these rules can be stored in a metadata repository and applied to the data warehouse centrally. This differs from an OLTP approach, which typically has data and business rules scattered throughout the system. The process of loading data into a data warehouse can be performed either through data transformation tools that provide a GUI to aid in the development and maintenance of business rules or through more traditional methods, such as developing programs or utilities to load the data warehouse, using programming languages such as PL/SQL, C++, or .NET Framework languages. This decision is not easy for organizations. Several issues affect whether an organization will purchase data transformation tools or build the transformation process itself:

- Data transformation tools are expensive.
- Data transformation tools may have a long learning curve.
- It is difficult to measure how the IT organization is doing until it has learned to use the data transformation tools.

In the long run, a transformation-tool approach should simplify the maintenance of an organization's data warehouse. Transformation tools can also be effective in detecting and scrubbing (i.e., removing any anomalies in the data). OLAP and data mining tools rely on how well the data are transformed.

As an example of effective ETL, Motorola, Inc., uses ETL to feed its data warehouses. Motorola collects information from 30 different procurement systems and sends them to its global SCM data warehouse for analysis of aggregate company spending (see Songini, 2004).

Solomon (2005) classified ETL technologies into four categories: sophisticated, enabler, simple, and rudimentary. It is generally acknowledged that tools in the sophisticated category will result in the ETL process being better documented and more accurately managed as the data warehouse project evolves.

Even though it is possible for programmers to develop software for ETL, it is simpler use an existing ETL tool. The following are some of the important criteria in selecting an ETL tool to include (see Brown, 2004):

- Ability to read from and write to an unlimited number of data source architectures
- Automatic capturing and delivery of metadata
- A history of conforming to open standards
- An easy-to-use interface for the developer and the functional user

Performing extensive ETL may be a sign of poorly managed data and a fundamental lack of a coherent data management strategy. Karacsony (2006) indicated that there is a direct correlation between the extent of redundant data and the number of ETL processes. When data are managed correctly as an enterprise asset, ETL efforts are significantly reduced, and redundant data are completely eliminated. This leads to huge savings in maintenance and greater efficiency in new development while also

improving data quality. Poorly designed ETL processes are costly to maintain, change, and update. Consequently, it is crucial to make the proper choices in terms of the technology and tools to use for developing and maintaining the ETL process.

Providers of packaged ETL systems are numerous. Database vendors currently offer ETL capabilities that both enhance and compete with independent ETL tools. SAS acknowledges the importance of data quality and offers the industry's first fully integrated solution that merges ETL and data quality to transform data into strategic valuable assets. Other ETL software providers include Microsoft, Oracle, IBM, Informatica, Embarcadero, and Tibco. For additional information on ETL, see Eckerson (2003), Karaksony (2006), and Songini (2004).

Section 5.5 Review Questions

1. Describe data integration.
2. Describe the three steps of the ETL process.
3. Why is the ETL process so important for data warehousing efforts?

5.6 DATA WAREHOUSE DEVELOPMENT

A data warehousing project is a major undertaking for any organization and is more complicated than a simple, mainframe selection and implementation project because it comprises and influences many departments and many input and output interfaces, and it can be part of a CRM business strategy. A data warehouse provides several benefits that can be classified as direct and indirect. Direct benefits include the following:

- Allowing end users to perform extensive analysis in numerous ways.
- A consolidated view of corporate data (i.e., a single version of the truth).
- Better and more timely information. A data warehouse permits information processing to be relieved from costly operational systems onto low-cost servers; therefore, many more end-user information requests can be processed more quickly.
- Enhanced system performance. A data warehouse frees production processing because some operational system reporting requirements are moved to DSS.
- Simplification of data access.

Indirect benefits result from end users using these direct benefits. On the whole, these benefits enhance business knowledge, present competitive advantage, enhance customer service and satisfaction, facilitate decision making, and help in reforming business processes, and therefore they are the strongest contributions to competitive advantage. (For a discussion of how to create a competitive advantage through data warehousing, see Parzinger and Frolick, 2001.) For a detailed discussion of how organizations can obtain exceptional levels of payoffs, see Watson et al. (2002). Given the potential benefits that a data warehouse can provide and the substantial investments in time and money that such a project requires, it is critical that an organization structure its data warehouse project to maximize the chances of success. In addition, the organization must, obviously, take costs into consideration. Kelly (2001) described a return on investment (ROI) approach that considers benefits in the categories of *keepers* (i.e., money saved by improving traditional decision support functions); *gatherers* (i.e., money saved due to automated collection and dissemination of information), and *users* (i.e., money saved or gained from decisions made using the data warehouse). Costs include those related to hardware, software, network bandwidth, internal development, internal support, training,

and external consulting. The net present value is calculated over the expected life of the data warehouse. Because the benefits are broken down approximately as 20 percent for keepers, 30 percent for gatherers, and 50 percent for users, Kelly indicated that users should be involved in the development process, a success factor typically mentioned as critical for systems that imply change in an organization.

Application Case 5.3 provides an example of a data warehouse that was developed and delivered intense competitive advantage for the Hokuriku (Japan) Coca-Cola Bottling Company. The system was so successful that plans are under way to expand it to encompass the more than one million Coca-Cola vending machines in Japan.

Application Case 5.3

Things Go Better with Coke's Data Warehouse

In the face of competitive pressures and consumer demand, how does a successful bottling company ensure that its vending machines are profitable? The answer for Hokuriku Coca-Cola Bottling Company (HCCBC) is a data warehouse and analytical software implemented by Teradata Corp. HCCBC built the system in response to a data warehousing system developed by its rival, Mikuni. The data warehouse collects not only historical data but also near-real-time data from each vending machine (viewed as a store) that could be transmitted via wireless connection to headquarters. The initial phase of the project was deployed in 2001. The data warehouse approach provides detailed product information, such as time and date of each sale, when a product sells out, whether someone was short-changed, and whether the machine is malfunctioning. In each case, an *alert* is triggered, and the vending machine immediately reports it to the data center over a wireless transmission system. (Note that Coca-Cola in the United States has used modems to link vending machines to distributors for over a decade.)

In 2002, HCCBC conducted a pilot test and put all its Nagano vending machines on a wireless network to gather near-real-time point of sale (POS) data from each one. The results were astounding because they accurately forecasted demand and identified problems quickly. Total sales immediately increased 10 percent. In addition, due to the more accurate machine servicing, overtime and other costs decreased 46 percent. In addition, each salesperson was able to service up to 42 percent more vending machines.

The test was so successful that planning began to expand it to encompass the entire enterprise (60,000 machines), using an *active data warehouse*. Eventually, the data warehousing solution will ideally expand across corporate boundaries into the entire Coca-Cola Bottlers network so that the more than one million vending machines in Japan will be networked, leading to immense cost savings and higher revenue.

Sources: Adapted from K.D. Schwartz, "Decisions at the Touch of a Button," *Teradata Magazine*, **teradata.com/t/page/117774/index.html** (accessed April 2006); K.D. Schwartz, "Decisions at the Touch of a Button," *DSS Resources*, March 2004, pp. 28–31, **dssresources.com/cases/coca-colajapan/index.html** (accessed April 2006); and Teradata Corp., *Coca-Cola Japan Puts the Fizz Back in Vending Machine Sales*, **teradata.com/t/page/118866/index.html** (accessed April 2006).

Clearly defining the business objective, gathering project support from management end users, setting reasonable time frames and budgets, and managing expectations are critical to a successful data warehousing project. A data warehousing strategy is a blueprint for the successful introduction of the data warehouse. The strategy should describe where the company wants to go, why it wants to go there, and what it will do when it gets there. It needs to take into consideration the organization's vision, structure, and culture. See Matney (2003) for the steps that can help in developing a flexible and efficient support strategy. When the plan and support for a data warehouse are established, the organization needs to examine data warehouse vendors. (See Table 5.1 for a sample list of vendors; also see The Data Warehousing Institute **[twdi.com]** and *DM Review* **[dmreview.com]**.) Many vendors provide software demos of their data warehousing and BI product.

TABLE 5.1 Sample List of Data Warehousing Vendors

Vendor	*Product Offerings*
Computer Associates (**cai.com**)	Comprehensive set of data warehouse (DW) tools and products
DataMirror Corp. (**datamirror.com**)	DW administration, management, and performance products
Data Advantage Group, Inc. (**dataadvantagegroup.com**)	Metadata software
Dell Computer Corp. (**dell.com**)	DW servers
Embarcadero Technologies (**embarcadero.com**)	DW administration, management, and performance products
Business Objects (**businessobjects.com**)	Data cleansing software
Harte-Hanks, Inc. (**harte-hanks.com**)	Customer relationship management (CRM) products and services
Hewlett-Packard Company (**hp.com**)	DW servers
Hummingbird Ltd. (**hummingbird.com**)	DW engines and exploration warehouses
Hyperion Solutions Corp. (**hyperion.com**)	Comprehensive set of DW tools, products, and applications
IBM (**ibm.com**)	DW tools, products, and applications
Informatica Corp. (**informatica.com**)	DW administration, management, and performance products
Microsoft Corp. (**microsoft.com**)	DW tools and products
Oracle (including PeopleSoft and Siebel) (**oracle.com**)	DW, ERP and CRM tools, products, and applications
SAS Institute, Inc. (**sas.com**)	DW tools, products, and applications
Siemens (**siemens.com**)	DW servers
Sybase, Inc. (**sybase.com**)	Comprehensive set of DW tools and applications
Teradata (**teradata.com**)	DW tools, products, and applications

DATA WAREHOUSE VENDORS

McCloskey (2002) cited six guidelines that need to be considered when developing a vendor list: financial strength, ERP linkages, qualified consultants, market share, industry experience, and established partnerships. We can collect additional data from trade shows and corporate Web sites, as well as by submitting requests for specific product information. Van den Hoven (1998) differentiated three types of data warehousing products. The first type handles functions such as locating, extracting, transforming, cleansing, transporting, and loading the data into the data warehouse. The second type is a data management tool—a database engine that stores and manages the data warehouse as well as the metadata. The third type is a data access tool that provides end users with access to analyze the data in the data warehouse. This may include query generators, visualization, EIS, OLAP, and data mining capabilities.

DATA WAREHOUSE DEVELOPMENT APPROACHES

Many organizations need to create the data warehouses used for decision support. Two competing approaches are employed. The first approach is that of Bill Inmon, who is often called "the father of data warehousing." Inmon supports a top-down development approach that adapts traditional relational database tools to the development needs of an enterprise-wide data warehouse, also known as the *EDW approach*. The

TABLE 5.2 Contrasts Between the Data Mart and EDW Development Approaches

Effort	*Data Mart Approach*	*EDW Approach*
Scope	One subject area	Several subject areas
Development time	Months	Years
Development cost	$10,000 to $100,000+	$1,000,000+
Development difficulty	Low to medium	High
Data prerequisite for sharing	Common (within business area)	Common (across enterprise)
Sources	Only some operational and external systems	Many operational and external systems
Size	Megabytes to several gigabytes	Gigabytes to petabytes
Time horizon	Near-current and historical data	Historical data
Data transformations	Low to medium	High
Frequency of update	Hourly, daily, weekly	Weekly, monthly
Technology		
Hardware	Workstations and departmental servers	Enterprise servers and mainframe computers
Operating system	Windows and Linux	Unix, Z/OS, OS/390
Databases	Workgroup or standard database servers	Enterprise database servers
Usage		
Number of simultaneous users	10s	100s to 1,000s
User types	Business area analysts and managers	Enterprise analysts and senior executives
Business spotlight	Optimizing activities within the business area	Cross-functional optimization and decision making

Sources: Adapted from J. Van den Hoven, "Data Marts: Plan Big, Build Small," in *IS Management Handbook*, 8th ed., CRC Press, Boca Raton, FL, 2003; and T. Ariyachandra and H. Watson, "Which Data Warehouse Architecture Is Most Successful?" *Business Intelligence Journal*, Vol. 11, No. 1, First Quarter 2006, pp. 4–6.

second approach is that of Ralph Kimball, who proposes a bottom-up approach that employs dimensional modeling, also known as the *data mart approach*.

Knowing how these two models are alike and how they differ helps us understand the basic data warehouse concepts (e.g., see Breslin, 2004). We show some of the advantages and disadvantages of both approaches in Table 5.2. We describe these approaches in detail next.

The Inmon Model: The EDW Approach

Inmon's approach emphasizes top-down development, employing established database development methodologies and tools, such as entity-relationship diagrams (ERD), and an adjustment of the spiral development approach. The EDW approach does not preclude the creation of data marts. The EDW is the ideal in this approach because it provides a consistent and comprehensive view of the enterprise. Murtaza (1998) presented a framework for developing EDW.

The Kimball Model: The Data Mart Approach

Kimball's data mart strategy is a "plan big, build small" approach. A data mart is a subject-oriented or department-oriented data warehouse. It is a scaled-down version of a data warehouse that focuses on the requests of a specific department, such as marketing or sales. This model applies dimensional data modeling, which starts with tables. Kimball advocated a development methodology that entails a bottom-up approach, which in the case of data warehouses means building one data mart at a time.

Which Model Is Best?

There is no one-size-fits-all strategy to data warehousing. An enterprise's data warehousing strategy can evolve from a simple data mart to a complex data warehouse in response to user demands, the enterprise's business requirements, and the enterprise's maturity in managing its data resources. For many enterprises, a data mart is frequently

TABLE 5.3 Comparison of the Essential Characteristic Differences Between the Inmon and Kimball Development Approaches

Characteristic	*Inmon*	*Kimball*
Methodology and Architecture		
Overall approach	Top-down	Bottom-up
Architecture structure	Enterprise-wide (atomic) data warehouse "feeds" departmental databases	Data marts model a single business process, and enterprise consistency is achieved through a data bus and conformed dimensions
Complexity of the method	Quite complex	Fairly simple
Comparison with established development methodologies	Derived from the spiral methodology	Four-step process; a departure from relational database management system (RDBMS) methods
Discussion of physical design	Fairly thorough	Fairly light
Data Modeling		
Data orientation	Subject- or data driven	Process oriented
Tools	Traditional (entity-relationship diagrams [ERD], data flow diagrams [DFD])	Dimensional modeling; a departure from relational modeling
End-user accessibility	Low	High
Philosophy		
Primary audience	IT professionals	End users
Place in the organization	Integral part of the corporate information factory	Transformer and retainer of operational data
Objective	Deliver a sound technical solution based on proven database methods and technologies	Deliver a solution that makes it easy for end users to directly query the data and still get reasonable response times

Sources: Adapted from M. Breslin, "Data Warehousing Battle of the Giants: Comparing the Basics of Kimball and Inmon Models," *Business Intelligence Journal*, Vol. 9, No. 1, Winter 2004, pp. 6–20; and T. Ariyachandra and H. Watson, "Which Data Warehouse Architecture Is Most Successful?" *Business Intelligence Journal*, Vol. 11, No. 1, First Quarter 2006.

a convenient first step to acquiring experience in constructing and managing a data warehouse while presenting business users with the benefits of better access to their data; in addition, a data mart commonly indicates the business value of data warehousing. Ultimately, obtaining an EDW is ideal (see Application Case 5.4). However, the development of individual data marts can often provide many benefits along the way toward developing an EDW, especially if the organization is unable or unwilling to invest in a large-scale project. Data marts can also demonstrate feasibility and success in providing benefits. This could potentially lead to an investment in an EDW. Table 5.3 summarizes the most essential characteristic differences between the two models.

Application Case 5.4

HP Consolidates Hundreds of Data Marts into a Single EDW

In December 2005, Hewlett-Packard Co. (HP) planned to consolidate its 762 data marts around the world into a single EDW. HP took this approach both to get a superior sense of its own business and to determine how best to serve its customers. Mark Hurd, HP's president and chief executive, stated that "there was a thirst for analytic data" inside the company that had unfortunately led to the creation of many data marts. Those data silos were very expensive to design and maintain, and they did not produce the enterprise-wide view of internal and customer information that HP wanted. In mid-2006, HP started to consolidate the data in the data marts into the new data warehouse. All the disparate data marts will ultimately be eliminated.

Sources: Adapted from C. Martins, "HP to Consolidate Data Marts into Single Warehouse," *Computerworld*, December 13, 2005; C. Martins, "HP to Consolidate Data Marts into Single Warehouse," *InfoWorld*, December 13, 2005; and C. Martins, "HP to Consolidate Data Marts into One Warehouse," *ITWorld Canada*, December 14, 2005.

ADDITIONAL DATA WAREHOUSE DEVELOPMENT CONSIDERATIONS

Some organizations want to completely outsource their data warehousing efforts. They simply do not want to deal with software and hardware acquisitions, and they do not want to manage their information systems. One alternative is to use hosted data warehouses. In this scenario, another firm—ideally, one that has a lot of experience and expertise—develops and maintains the data warehouse. However, there are security and privacy concerns with this approach. See Technology Insights 5.5 for some details.

DATA WAREHOUSE STRUCTURE: THE STAR SCHEMA

A typical data warehouse structure is shown in Figure 5.1. While there are many variations on architecture, the most important one is the star schema. The data warehouse design is based on the concept of *dimensional modeling*. **Dimensional modeling** is a retrieval-based system that supports high-volume query access. The star schema is the means by which dimensional modeling is implemented. A star schema contains a central fact table surrounded by several dimension tables. The fact table contains a large number of rows that correspond to observed business or facts. A fact table contains the attributes needed to perform decision analysis, descriptive attributes used for query reporting, and foreign keys to link to dimension tables. The decision analysis attributes consist of performance measures, operational metrics, aggregated measures, and all the other metrics needed to analyze the organization's performance. In other words, the fact table primarily addresses *what* the data warehouse supports for decision analysis.

Surrounding the central fact tables (and linked via foreign keys) are *dimension tables*. The **dimension tables** contain classification and aggregation information about

TECHNOLOGY INSIGHTS 5.5

How About a Hosted Data Warehouse?

A hosted data warehouse has nearly the same, if not more, functionality as an on-site data warehouse, but it does not consume computer resources on client premises. A hosted data warehouse offers the benefits of BI minus the cost of computer upgrades, network upgrades, software licenses, in-house development, and in-house support and maintenance.

A hosted data warehouse offers the following benefits:

- Requires minimal investment in infrastructure
- Frees up capacity on in-house systems
- Frees up cash flow
- Makes powerful solutions affordable
- Enables powerful solutions that provide for growth
- Offers better quality equipment and software
- Provides faster connections
- Enables users to access data from remote locations
- Allows a company to focus on core business
- Meets storage needs for large volumes of data

Despite its benefits, a hosted data warehouse is not necessarily a good fit for every company. Large companies with revenue upwards of $500 million could lose money if they already have underused internal infrastructure and IT staff. Furthermore, companies that see the paradigm shift of outsourcing applications as loss of control of their data are not prone to use a business intelligence service provider (BISP). Finally, the most significant and common argument against implementing a hosted data warehouse is that it may be unwise to outsource sensitive applications for reasons of security and privacy.

Sources: Partly adapted from M. Thornton and M. Lampa, "Hosted Data Warehouse," *Journal of Data Warehousing*, Vol. 7, No. 2, 2002, pp. 27–34; and M. Thornton, "What About Security? The Most Common, but Unwarranted, Objection to Hosted Data Warehouses," *DM Review*, Vol. 12, No. 3, March 18, 2002, pp. 30–43.

the central fact rows. Dimension tables contain attributes that describe the data contained within the fact table; they address *how* data will be analyzed. Dimension tables have a one-to-many relationship with rows in the central fact table. Some examples of dimensions that would support a product fact table are location, time, and size. The star schema design provides extremely fast query-response time, simplicity, and ease of maintenance for read-only database structures. According to Raden (2003), setting up a star schema for real-time updating could be a straightforward approach, as long as a few rules are followed. We show a star schema example in Figure 5.9.

The **grain** (also known as granularity) of a data warehouse defines the highest level of detail that is supported. The grain indicates whether the data warehouse is highly summarized or also includes detailed transaction data. If the grain is defined too high, then the warehouse may not support detail requests to *drill down* into the data. **Drill-down** analysis is the process of probing beyond a summarized value to investigate each of the detail transactions that comprise the summary. A low level of granularity will result in more data being stored in the warehouse. Larger amounts of detail may affect the performance of queries by making the response times longer. Therefore, during the scoping of a data warehouse project, it is important to identify the right level of granularity that will be needed. See Tennant (2002) for a discussion of granularity issues in metadata.

DATA WAREHOUSING IMPLEMENTATION ISSUES

Implementing a data warehouse is generally a massive effort that must be planned and executed according to established methods. There are, however, many facets to the project lifecycle, and no single person can be an expert in each area. Here we discuss

FIGURE 5.9 Star Schema

specific ideas and issues as they relate to data warehousing. Inmon (2006) provided a set of actions that a data warehouse systems programmer may use to tune a data warehouse.

Solomon (2005) provided some guidelines regarding the critical questions that must be asked, some risks that should be weighted, and some processes that can be followed to help ensure a successful data warehouse implementation. He compiled a list of 11 major tasks that could be performed in parallel:

1. Establishment of service-level agreements and data-refresh requirements
2. Identification of data sources and their governance policies
3. Data quality planning
4. Data model design
5. ETL tool selection
6. Relational database software and platform selection
7. Data transport
8. Data conversion
9. Reconciliation process

10. Purge and archive planning
11. End-user support

Following these guidelines should increase an organization's chances for success. Given the size and scope of an enterprise-level data warehouse initiative, failure to anticipate these issues greatly increases the risks of failure.

Hwang and Xu (2005) conducted a major survey of data warehousing success issues. The results established that data warehousing success is a multifaceted construct, and Hwang and Xu proposed that a data warehouse be constructed while keeping in mind the goal of improving user productivity. Extremely significant benefits of doing so include prompt information retrieval and enhanced quality information. The survey results also indicated that success hinges on factors of different dimensions.

People want to know how successful their BI and data warehousing initiatives are in comparison to those of other companies. Ariyachandra and Watson (2006a) proposed some benchmarks for BI and data warehousing success. Watson et al. (1999) researched data warehouse failures. Their results showed that people define a "failure" in different ways, and this was confirmed by Ariyachandra and Watson (2006a). The Data Warehousing Institute (**tdwi.org**) has developed a data warehousing maturity model that an enterprise can apply in order to benchmark its evolution. The model offers a fast means to gauge where the organization's data warehousing initiative is now and where it needs to go next. The maturity model consists of six stages: prenatal, infant, child, teenager, adult, and sage. Business value rises as the data warehouse progresses through each succeeding stage. The stages are identified by a number of characteristics, including scope, analytic structure, executive perceptions, types of analytics, stewardship, funding, technology platform, change management, and administration. See Eckerson (2004) for complete details.

Weir (2002) described some of the best practices for implementing a data warehouse, which include the following guidelines:

- The project must fit with corporate strategy and business objectives.
- There must be complete buy-in to the project by executives, managers, and users.
- It is important to manage user expectations about the completed project.
- The data warehouse must be built incrementally.
- Build in adaptability.
- The project must be managed by both IT and business professionals.
- Develop a business/supplier relationship.
- Only load data that have been cleansed and are of a quality understood by the organization.
- Do not overlook training requirements.
- Be politically aware.

There are many risks in data warehouse projects. Most of them are also found in other IT projects, but data warehousing risks are more serious because data warehouses are expensive, large-scale projects. Each risk should be assessed at the inception of the project. Adelman and Moss (2001) described some of these risks, including the following:

- No mission or objective
- Quality of source data unknown
- Skills not in place
- Inadequate budget
- Lack of supporting software
- Source data not understood

Copyright

- Weak sponsor
- Users not computer literate
- Political problems or turf wars
- Unrealistic user expectations
- Architectural and design risks
- Scope creep and changing requirements
- Vendors out of control
- Multiple platforms
- Key people leaving the project
- Loss of the sponsor
- Too much new technology
- Having to fix an operational system
- Geographically distributed environment
- Team geography and language culture

Practitioners have unearthed a wealth of mistakes that have been made in the development of data warehouses. Watson et al. (1999) also discussed how such mistakes could lead to data warehouse failures (also see Barquin et al., 1997). Turban et al. (2006) listed failure factors that include cultural issues being ignored, inappropriate architecture, unclear business objectives, missing information, unrealistic expectations, low levels of data summarization, and low data quality.

When developing a successful data warehouse, it's important to carefully consider these issues:

- *Starting with the wrong sponsorship chain.* You need an executive sponsor who has influence over the necessary resources to support and invest in the data warehouse. You also need an executive *project driver*, someone who has earned the respect of other executives, has a healthy skepticism about technology, and is decisive but flexible. And you need an IS/IT manager to head up the project (the "you" in the project).
- *Setting expectations that you cannot meet and frustrating executives at the moment of truth.* There are two phases in every data warehousing project: Phase 1 is the selling phase, in which you internally market the project by selling the benefits to those who have access to needed resources. Phase 2 is the struggle to meet the expectations described in Phase 1. For a mere $1 to $7 million, you can, hopefully, deliver.
- *Engaging in politically naive behavior.* Do not simply state that a data warehouse will help managers make better decisions. This may imply that you feel they have been making bad decisions until now. Sell the idea that they will be able to get the information they need to help in decision making.
- *Loading the warehouse with information just because it is available.* Do not let the data warehouse become a data landfill. This would unnecessarily slow down the use of the system. There is a trend toward real-time computing and analysis. Data warehouses must be shut down to load data in a timely way.
- *Believing that data warehousing database design is the same as transactional database design.* In general, it is not. The goal of data warehousing is to access aggregates rather than a single or a few records, as in transaction-processing systems. Content is also different, as is evident in how data are organized. DBMS tend to be nonredundant, normalized, and relational, whereas data warehouses are redundant, not normalized, and multidimensional.
- *Choosing a data warehouse manager who is technology oriented rather than user oriented.* One key to data warehouse success is to understand that the users must get what they need, not advanced technology for technology's sake.

- *Focusing on traditional internal record-oriented data and ignoring the value of external data and of text, images, and, perhaps, sound and video.* Data come in many formats and must be made accessible to the right people at the right time and in the right format. They must be catalogued properly.
- *Delivering data with overlapping and confusing definitions.* Data cleansing is a critical aspect of data warehousing. It includes reconciling conflicting data definitions and formats organization-wide. Politically, this may be difficult because it involves change, typically at the executive level.
- *Believing promises of performance, capacity, and scalability.* Data warehouses generally require more capacity and speed than is originally budgeted for. Plan ahead to scale up.
- *Believing that your problems are over when the data warehouse is up and running.* DSS/BI projects tend to evolve continually. Each deployment is an iteration of the prototyping process. There will always be a need to add more and different data sets to the data warehouse, as well as additional analytic tools for existing and additional groups of decision makers. High energy and annual budgets must be planned for because success breeds success. Data warehousing is a continuous process.
- *Focusing on ad hoc data mining and periodic reporting instead of alerts.* The natural progression of information in a data warehouse is: (1) *Extract* the data from legacy systems, cleanse them, and feed them to the warehouse; (2) *support* ad hoc reporting until you learn what people want; and (3) *convert* the ad hoc reports into regularly scheduled reports.

This process of learning what people want in order to provide it seems natural, but it is not optimal or even practical. Managers are busy and need time to read reports. *Alert systems* are better than periodic reporting systems and can make a data warehouse mission critical. Alert systems monitor the data flowing into the warehouse and inform all key people who have a need to know as soon as a critical event occurs.

Sammon and Finnegan (2000) revealed the outcome of a study of four mature users of data warehousing technology. Their practices were captured in an outline of 10 organizational requisites for applying data warehousing. They think that organizations might potentially use this representation to internally evaluate the chances of the success of a data warehousing project and to recognize the parts that need attention prior to beginning implementation. A summary of their prerequisites model is as follows:

- A business-driven data warehousing initiative
- Executive sponsorship and commitment
- Funding commitment based on realistically managed expectations
- A project team
- Attention to source data quality
- A flexible enterprise data model
- Data stewardship
- A long-term plan for automated data extraction methods/tools
- Knowledge of data warehouse compatibility with existing systems
- Hardware/software proof of concept

Wixom and Watson (2001) defined a research model for data warehouse success that identified seven important implementation factors that can be categorized into three criteria (i.e., organizational issues, project issues, and technical issues):

1. Management support
2. Champion
3. Resources

4. User participation
5. Team skills
6. Source systems
7. Development technology

In many organizations, a data warehouse will be successful only if there is strong senior management support for its development and if there is a project champion. Although this would likely be true for any IT project, it is especially important for a data warehouse. The successful implementation of a data warehouse results in the establishment of an architectural framework that may allow for decision analysis throughout an organization and in some cases also provides comprehensive SCM by granting access to an organization's customers and suppliers. The implementation of Web-based data warehouses (called *Webhousing*) has facilitated ease of access to vast amounts of data, but it is difficult to determine the hard benefits associated with a data warehouse. *Hard benefits* are defined as benefits to an organization that can be expressed in monetary terms. Many organizations have limited IT resources and must prioritize projects. Management support and a strong project champion can help ensure that a data warehouse project will receive the resources necessary for successful implementation. Data warehouse resources can be a significant cost, in some cases requiring high-end processors and large increases in direct-access storage devices (DASD). Web-based data warehouses may also have special security requirements to ensure that only authorized users have access to the data.

User participation in the development of data and access modeling is a critical success factor in data warehouse development. During data modeling, expertise is required to determine what data are needed, define business rules associated with the data, and decide what aggregations and other calculations may be necessary. Access modeling is needed to determine how data are to be retrieved from a data warehouse, and it assists in the physical definition of the warehouse by helping to define which data require indexing. It may also indicate whether dependent data marts are needed to facilitate information retrieval. The team skills needed to develop and implement a data warehouse include in-depth knowledge of the database technology and development tools used. Source systems and development technology, as mentioned previously, reference the many inputs and the processes used to load and maintain a data warehouse.

MASSIVE DATA WAREHOUSES AND SCALABILITY

In addition to flexibility, a data warehouse needs to support scalability. The main issues pertaining to scalability are the amount of data in the warehouse, how quickly the warehouse is expected to grow, the number of concurrent users, and the complexity of user queries. A data warehouse must scale both horizontally and vertically. The warehouse will grow as a function of data growth and the need to expand the warehouse to support new business functionality. Data growth may be a result of the addition of current cycle data (e.g., this month's results) and/or historical data.

Hicks (2001) described huge databases and data warehouses. Wal-Mart is continually increasing the size of its massive data warehouse. Wal-Mart is believed to use a warehouse with hundreds of terabytes of data to study sales trends and track inventory and other tasks. The U.S. Department of Defense is using a 5 petabyte data warehouse and repository to hold medical records for 9 million military

personnel. Because of the storage required to archive its news footage, CNN also has a petabyte-sized data warehouse.

Given that the size of data warehouses is expanding at an exponential rate, *scalability* is an important issue. Good scalability means that queries and other data-access functions will grow (ideally) linearly with the size of the warehouse. See Rosenberg (2006) for approaches to improve query performance. In practice, specialized methods have been developed to create scalable data warehouses. Scalability is difficult when managing hundreds of terabytes or more. Terabytes of data have considerable inertia, occupy a lot of physical space, and require powerful computers. Some firms use parallel processing, and others use clever indexing and search schemes to manage their data. Some spread their data across different physical data stores. As more data warehouses approach the petabyte size, better and better solutions to scalability continue to be developed.

Hall (2002) also addressed scalability issues. AT&T is an industry leader in deploying and using massive data warehouses. With its 26-terabyte data warehouse, AT&T Labs can detect fraudulent use of calling cards and investigate calls related to kidnappings and other crimes. It can also compute millions of call-in votes from TV viewers selecting the next American Idol.

For a sample of successful data warehousing implementations, see Edwards (2003). Jukic and Lang (2004) examined the trends and specific issues related to use of offshore resources in the development and support of data warehousing and BI applications. Davison (2003) indicated that IT-related offshore outsourcing had been growing at 20 to 25 percent per year. When considering offshoring data warehousing projects, careful consideration must be given to culture and security (for details, see Jukic and Lang, 2004).

Section 5.6 Review Questions

1. List the benefits of data warehouses.
2. List several criteria for selecting a data warehouse vendor and describe why they are important.
3. Does a bottom-up data warehouse development approach use an enterprise data model?
4. Describe the major similarities and differences between the Inmon and Kimball data warehouse development approaches.
5. List the different types of data warehouse architectures.

5.7 REAL-TIME DATA WAREHOUSING

Data warehousing and BI tools traditionally focus on assisting managers in making strategic and tactical decisions. Increased data volumes and accelerating update speeds are fundamentally changing the role of the data warehouse in modern business. For many businesses, making fast and consistent decisions across the enterprise requires more than a traditional data warehouse or data mart. Traditional data warehouses are not business critical. Data are commonly updated on a weekly basis, and this does not allow for responding to transactions in near-real-time.

More data, coming in faster and requiring immediate conversion into decisions, means that organizations are confronting the need for real-time data warehousing. This

is because decision support has become operational, integrated BI requires closed-loop analytics, and yesterday's ODS will not support existing requirements.

In 2003, with the advent of real-time data warehousing, there was a shift toward using these technologies for operational decisions. **Real-time data warehousing (RDW)**, also known as **active data warehousing (ADW)**, is the process of loading and providing data via the data warehouse as they become available. It fairly recently evolved from the EDW concept. The active traits of an RDW/ADW supplement and expand traditional data warehouse functions into the realm of tactical decision making. People throughout the organization who interact directly with customers and suppliers will be empowered with information-based decision making at their fingertips. Even further leverage results when an ADW provides information directly to customers and suppliers. The reach and impact of information access for decision making can positively affect almost all aspects of customer service, SCM, logistics, and beyond. E-business has become a major catalyst in the demand for active data warehousing (see Armstrong, 2000). For example, online retailer Overstock.com, Inc. (**overstock.com**), connected data users to a real-time data warehouse. At Egg plc, the world's largest purely online bank, a customer data warehouse is refreshed in near-real-time. See Application Case 5.5.

Application Case 5.5

Egg Plc Fries the Competition in Near-Real-Time

Egg plc (**egg.com**) is the world's largest online bank. It provides banking, insurance, investments, and mortgages to more than 3.6 million customers, through its Internet site. In 1998, Egg selected Sun Microsystems to create a reliable, scalable, secure infrastructure to support its more than 2.5 million daily transactions. In 2001, the system was upgraded to eliminate latency problems. This new customer data warehouse (CDW) used Sun, Oracle, and SAS software products. The initial data warehouse had about 10 terabytes of data and used a 16-CPU server. The system provides near-real-time data access. It provides data warehouse and data mining services to internal users, and it provides a requisite set of customer data to the customers themselves. Hundreds of sales and marketing campaigns are constructed using near-real-time data (within several minutes). And better, the system enables faster decision making about specific customers and customer classes.

Sources: Compiled from "Egg's Customer Data Warehouse Hits the Mark," *DM Review*, Vol. 15, No. 10, October 2005, pp. 24–28; Sun Microsystems, *Egg Banks on Sun to Hit the Mark with Customers*, September 19, 2005, **sun.com/smi/Press/sunflash/ 2005-09/sunflash.20050919.1.xml** (accessed April 2006); and ZD Net UK, *Sun Case Study: Egg's Customer Data Warehouse*, **whitepapers.zdnet.co.uk/0,39025945,60159401p-39000449q,00.htm** (accessed April 2006).

As business needs evolve, so do the requirements of the data warehouse. At this basic level, a data warehouse simply reports what happened. At the next level, some analysis occurs. As the system evolves, it provides prediction capabilities, which lead to the next level of operationalization. At its highest evolution, the ADW is capable of making events happen (e.g., activities such as creating sales and marketing campaigns or identifying and exploiting opportunities). See Figure 5.10 for a graphic description of this evolutionary process.

Teradata Corp. provides the baseline requirements to support an EDW. It also provides the new traits of active data warehousing required to deliver data freshness,

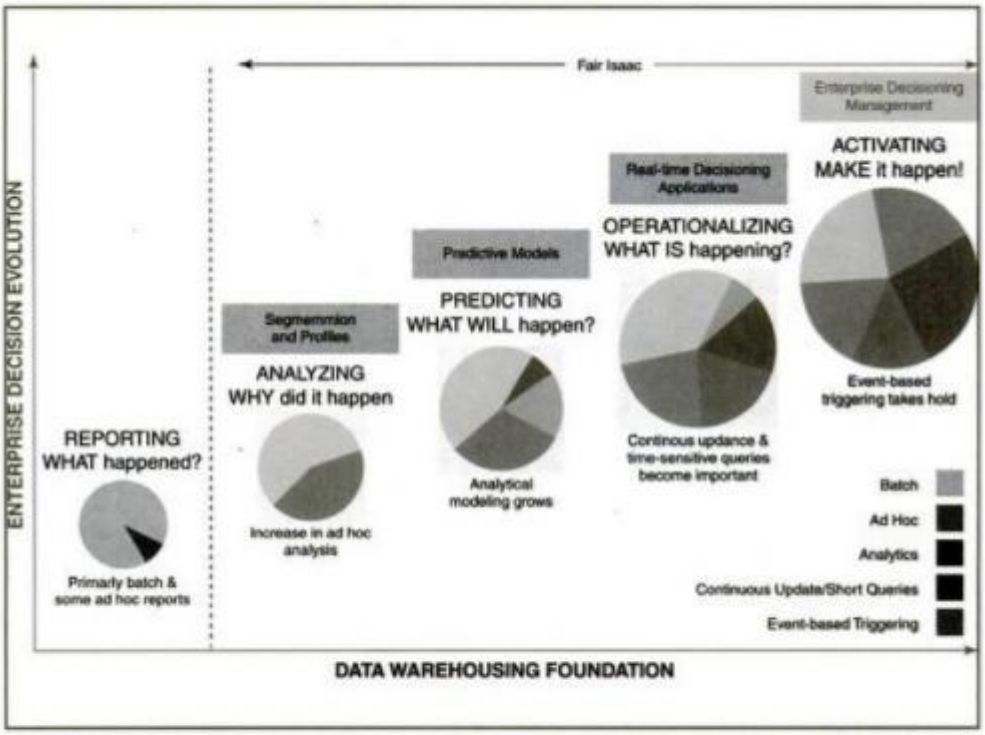


FIGURE 5.10 Enterprise Decision Evolution

Source: Courtesy of Teradata Corporation, a Division of NCR Corp. Used with permission.

performance, and availability and to enable enterprise decision management (see Figure 5.11 for an example).

An ADW offers an integrated information repository to drive strategic and tactical decision support within an organization. Real-time data warehousing upholds that instead of extracting operational data from an OLTP system in nightly batches into an ODS, data are assembled from OLTP systems as and when events happen and are moved at once into the data warehouse. This permits the instant updating of the data warehouse and the elimination of an ODS. At this point, tactical and strategic queries can be made against the RDW to use immediate as well as historical data.

According to Basu (2003), the most distinctive difference between a traditional data warehouse and an RDW is the shift in the data acquisition paradigm. Some of the business cases and enterprise requirements that led to the need for data in real-time include the following:

- A business often cannot afford to wait a whole day for its operational data to load into the data warehouse for analysis.
- Until now, data warehouses have captured snapshots of an organization's fixed states instead of incremental real-time data showing every state change and almost analogous patterns over time.
- With a traditional hub-and-spoke architecture, retaining the metadata in sync is difficult. It is also costly to develop, maintain, and secure many systems as opposed to one huge data warehouse so that data are centralized for BI/BA tools.

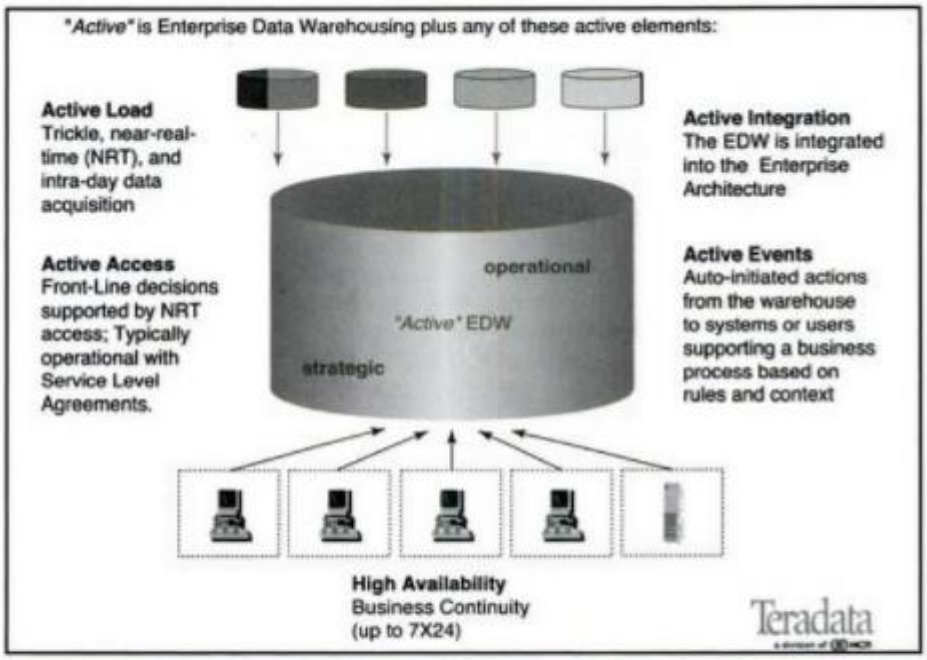


FIGURE 5.11 The Teradata Active EDW

Source: Courtesy of Teradata Corporation, a Division of NCR Corp. Used with permission.

- In cases of huge nightly batch loads, the necessary ETL setup and processing power for large nightly data warehouse loading might be very high, and the processes might take too long. An EAI with real-time data collection can reduce or eliminate the nightly batch processes.

Despite the benefits of an RDW, developing one can create its own set of issues. These problems relate to architecture, data modeling, physical database design, storage and scalability, and maintainability. In addition, depending on exactly when data are accessed, even down to the microsecond, different versions of the truth may be extracted and created, which can confuse team members. For some details, refer to Basu (2003) and Terr (2004).

Real-time solutions present a remarkable set of challenges to BI activities. Although it is not ideal for all solutions, real-time data warehousing may be successful if the organization develops a sound methodology to handle project risks, incorporate proper planning, and focus on quality assurance activities. Understanding the common challenges and applying best practices can reduce the problem levels that are often a part of implementing complex data warehousing systems that incorporate BI/BA methods. Details and real implementations are discussed by Burdett and Singh (2004) and Wilk (2003). Also see Akbay (2006) and Ericson (2006).

See Technology Insights 5.7 for some details on how the real-time concept evolved. The flight management dashboard application at Continental Airlines (see the opening vignette) illustrates the power of real-time BI in accessing a data warehouse for use in face-to-face customer interaction situations. The operations staff use the real-time system to identify issues in the Continental flight network. As another example, UPS invested $600 million so it could use real-time data and processes. The investment was expected to cut 100 million delivery miles and save 14 million gallons of fuel annually by

TECHNOLOGY INSIGHTS 5.7

The Real-Time Realities of Active Data Warehousing

By 2003, the role of data warehousing in practice was growing rapidly. Real-time systems, though a novelty, were the latest buzz, along with the major complications of providing data and information instantaneously to those who need them. Many experts, including Peter Coffee, *eWeek*'s technology editor, believe that real-time systems must feed a real-time decision-making process. Stephen Brobst, CTO of the Teradata division of NCR, indicated that active data warehousing is a process of evolution in how an enterprise uses data. *Active* means that the data warehouse is also used as an operational and tactical tool. Brobst provided a five-stage model that fits Coffee's experience (2003) of how organizations "grow" in their data utilization (see Brobst et al., 2005). These stages (and the questions they purport to answer) are reporting (What happened?), analysis (Why did it happen?), prediction (What will happen?), operationalizing (What is happening?), and active warehousing (What do I want to happen?). The last stage, active warehousing, is where the greatest benefits may be obtained. Many organizations are enhancing centralized data warehouses to serve both operational and strategic decision making.

Sources: Adapted from P. Coffee, "'Active' Warehousing," *eWEEK*, Vol. 20, No. 25, June 23, 2003, p. 36; and Teradata Corp., *Active Data Warehousing*, **teradata.com/t/page/87127/index.html** (accessed April 2006).

managing its real-time package flow technologies (see Malykhina, 2003). In Table 5.4, we show a comparison of the traditional and active data warehousing environments.

Real-time data warehousing, near-real-time data warehousing, zero-latency warehousing, and active data warehousing are different names used in practice to describe the same concept. Gonzales (2005) presented different definitions for ADW. According to Gonzales, ADW is only one option that provides blended tactical and strategic data on-demand. The architecture to build an ADW is very similar to the corporate information factory architecture developed by Bill Inmon. The only difference between a corporate information factory and an ADW is the implementation of both data stores in a single environment. However, an SOA based on XML and Web services provide another option for blending tactical and strategic data on-demand.

TABLE 5.4 Comparison Between Traditional and Active Data Warehousing Environments

Traditional Data Warehouse Environment	*Active Data Warehouse Environment*
Strategic decisions only	Strategic and tactical decisions
Results sometimes hard to measure	Results measured with operations
Daily, weekly, monthly data currency acceptable; summaries often appropriate	Only comprehensive detailed data available within minutes is acceptable
Moderate user concurrency	High number (1,000 or more) of users accessing and querying the system simultaneously
Highly restrictive reporting used to confirm or check existing processes and patterns; often uses predeveloped summary tables or data marts	Flexible ad hoc reporting, as well as machine-assisted modeling (e.g., data mining) to discover new hypotheses and relationships
Power users, knowledge workers, internal users	Operational staffs, call centers, external users

Sources: Adapted from P. Coffee, "'Active' Warehousing," *eWEEK*, Vol. 20, No. 25, June 23, 2003, p. 36; and Teradata Corp., *Active Data Warehousing*, **teradata.com/t/page/87127/index.html** (accessed April 2006).

One critical issue in real-time data warehousing is that not all data should be updated continuously. This may certainly cause problems when reports are generated in real-time because one person's results may not match another person's. For example, a company using Business Objects Web Intelligence noticed a significant problem with real-time intelligence. Real-time reports are all different when produced at slightly different times (see Peterson, 2003). Also, it may not be necessary to update certain data continuously (e.g., course grades that are three or more years old).

Real-time requirements change the way we view the design of databases, data warehouses, OLAP, and data mining tools because they are literally updated concurrently while queries are active. But the substantial business value in doing so has been demonstrated, so it is crucial that organizations adopt these methods in their business processes. Careful planning is critical in such implementations.

Section 5.7 Review Questions

1. What is an RDW?
2. List the benefits of an RDW.
3. What are the major differences between a traditional data warehouse and an RDW?
4. List some of the drivers for RDW.

5.8 DATA WAREHOUSE ADMINISTRATION AND SECURITY ISSUES

Data warehouses provides a distinct competitive edge to enterprises that effectively create and use them. Due to its huge size and its intrinsic nature, a data warehouse requires especially strong monitoring in order to sustain satisfactory efficiency and productivity. The successful administration and management of a data warehouse entails skills and proficiency that go past what is required of a traditional database administrator (DBA). A **data warehouse administrator (DWA)** should be familiar with high-performance software, hardware, and networking technologies. He or she should also possess solid business insight. Because data warehouses feed BI systems and DSS that help managers with their decision-making activities, the DWA should be familiar with the decision-making processes so as to suitably design and maintain the data warehouse structure. It is particularly significant for a DWA to keep the existing requirements and capabilities of the data warehouse stable while simultaneously providing flexibility for rapid improvements. Finally, a DWA must possess excellent communications skills. See Benander et al. (2000) for a description of the key differences between a DBA and a DWA.

Security and privacy of information is a main and significant concern for a data warehouse professional. The U.S. government has passed regulations (e.g., the Gramm-Leach Bliley privacy and safeguards rules, the Health Insurance Portability and Accountability Act of 1996 [HIPAA]), instituting obligatory requirements in the management of customer information. Hence, companies must create security procedures that are effective yet flexible to conform to numerous privacy regulations. According to Elson and LeClerc (2005), effective security in a data warehouse should focus on four main areas:

1. Establishing effective corporate and security policies and procedures. An effective security policy should start at the top, with executive management, and should be communicated to all individuals within the organization.
2. Implementing logical security procedures and techniques to restrict access. This includes user authentication, access controls, and encryption technology.

TECHNOLOGY INSIGHTS 5.8

Ambeo Delivers Proven Data Access Auditing Solution

Since 1997, Ambeo (**ambeo.com**; now Embarcadero Technologies, Inc.) has deployed technology that provides performance management, data usage tracking, data privacy auditing, and monitoring to Fortune 1000 companies. These firms have some of the largest database environments in existence. Ambeo data access auditing solutions play a major role in an enterprise information security infrastructure.

The Ambeo technology is a relatively easy solution that records everything that happens in the databases, with low or zero overhead. In addition, it provides data access auditing that identifies exactly who is looking at data, when they are looking, and what they are doing with the data. This real-time monitoring helps quickly and effectively identify security breaches.

Sources: Adapted from "Ambeo Delivers Proven Data Access Auditing Solution," *Database Trends and Applications*, Vol. 19, No. 7, July 2005; and Ambeo, *Keeping Data Private (and Knowing It): Moving Beyond Conventional Safeguards to Ensure Data Privacy*, **ambeo.com/why_ambeo_white_papers.html** (accessed April 2006).

3. Limiting physical access to the data center environment.
4. Establishing an effective internal control review process with an emphasis on security and privacy.

See Technology Insights 5.8 for a description of Ambeo's important software tool that monitors security and privacy of data warehouses. Finally, keep in mind that accessing a data warehouse via a mobile device should always be performed cautiously. In this instance, data should only be accessed as read-only.

In the near term, data warehousing developments will be determined by noticeable factors (e.g., data volumes, increased intolerance for latency, the diversity and complexity of data types) and less noticeable factors (e.g., unmet end-user requirements for dashboards, balanced scorecards, master data management, information quality). Given these drivers, Agosta (2006) suggested that data warehousing trends will lean toward simplicity, value, and performance.

Section 5.8 Review Questions

1. What steps can an organization take to ensure the security and confidentiality of customer data in its data warehouse?
2. What skills should a DWA possess? Why?

5.9 RESOURCES, LINKS, AND THE TERADATA UNIVERSITY NETWORK CONNECTION

The use of this chapter and most other chapters in this book can be enhanced by the tools described in the following sections.

RESOURCES AND LINKS

We recommend looking at the following resources and links for further reading and explanations:

- The Data Warehouse Institute (**tdwi.com**)

- *DM Review* parentheses (**dmreview.com**)
- DSS Resources parentheses (**dssresources.com**)

CASES

All major MSS vendors (e.g., MicroStrategy, Microsoft, Oracle, IBM, Hyperion, Cognos, Exsys, Fair Issac, SAP, Information Builders) provide interesting customer success stories. Academic-oriented cases are available at the Harvard Business School Case Collection (**harvardbusinessonline.hbsp.harvard.edu**), Business Performance Improvement Resource (**bpir.com**), Idea Group Publishing (**idea-group.com**), Ivy League Publishing (**ivylp.com**), ICFAI Center for Management Research (**icmr.icfai.org/casestudies/icmr_case_studies.htm**), KnowledgeStorm (**knowledgestorm.com**), and other sites. For additional case resources, see Teradata University Network (**teradatauniversitynetwork.com**). For data warehousing cases, we specifically recommend the following from the Teradata University Network (**teradatauniversitynetwork.com**): *Continental Airlines Flies High with Real-Time Business Intelligence*, *Data Warehouse Governance at Blue Cross and Blue Shield of North Carolina*, *3M Moves to a Customer Focus Using a Global Data Warehouse*, *Data Warehousing Supports Corporate Strategy at First American Corporation*, *Harrah's High Payoff from Customer Information*, and *Whirlpool*. We also recommend the Data Warehousing Failures Assignment, which consists of eight short cases on data warehousing failures.

VENDORS, PRODUCTS, AND DEMOS

A comprehensive list is available at **(dmreview.com)**. Vendors are listed in Table 5.1. Also see **technologyevaluation.com**.

PERIODICALS

We recommend the following periodicals: