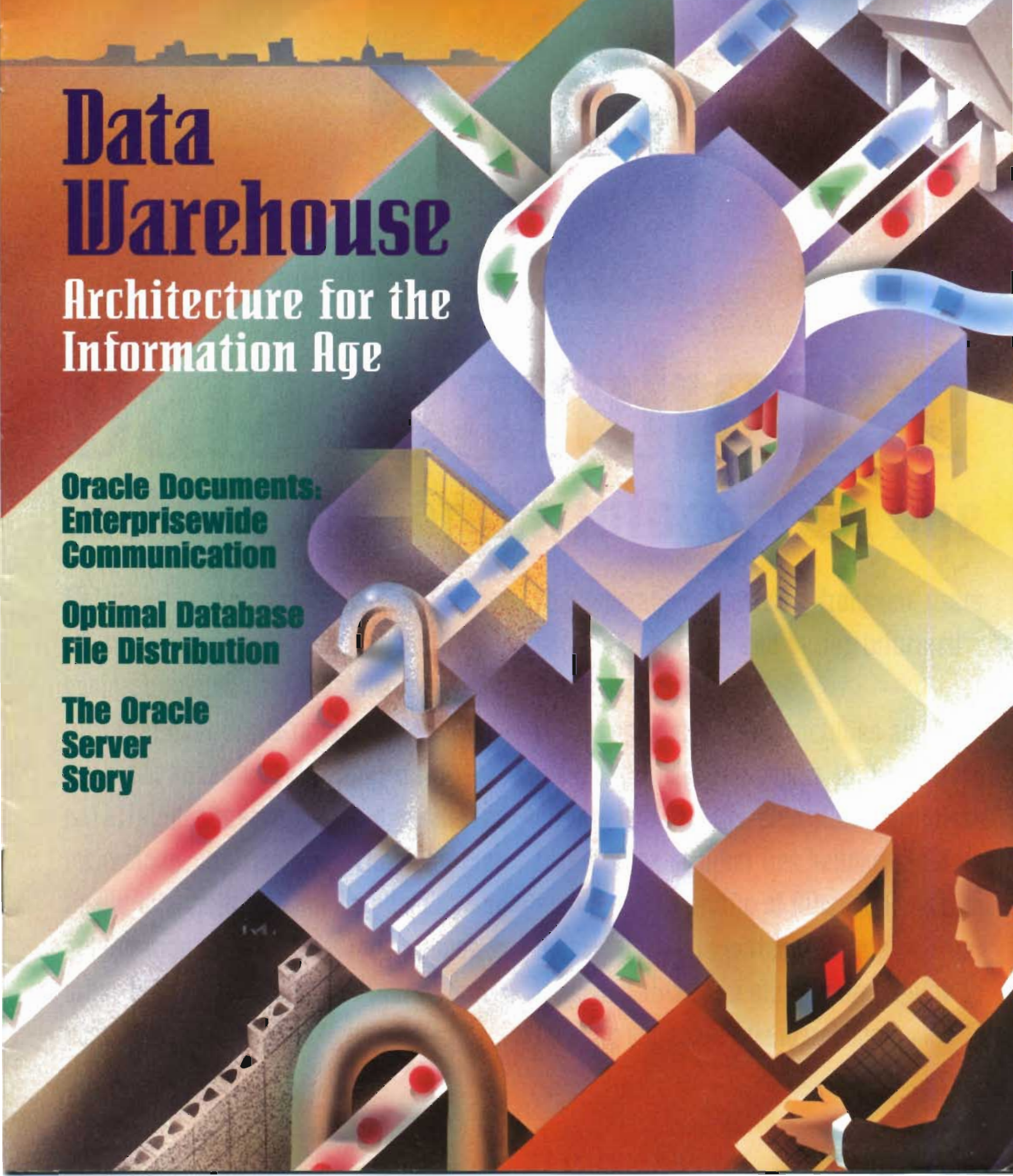
ORACLE®
MAGAZINE
VOLUME IX/NUMBER 1
JANUARY/FEBRUARY 1995/$4.95



Data Warehouse

Architecture for the Information Age

Oracle Documents: Enterprisewide Communication

Optimal Database File Distribution

The Oracle Server Story

Data Warehouse

Architecture for the Information Age

Thanks to computer technology, corporations can now store vast quantities of business-related data. The next step is to turn that data into strategic information. With advances in relational-database, parallel-processing, and distributed technology, information technology may finally live up to its name by allowing any organization to build a data warehouse in order to gain better insight into its business. With easy access to integrated data from operational systems, front-line users can make better—and better-informed—business decisions.

This special section includes an overview of the data warehouse, with a sidebar on Oracle's parallel-everything architecture, a look at what lies beyond the data warehouse, a profile of Alliance BlueCross BlueShield's data-warehouse implementation, and finally, a step-by-step guide to creating your own data warehouse.–*Editors*

BY KELLI WISETH

Data Warehouse

Architecture for the Information Age

The last two decades of business computing have focused on building the foundation for automating production, streamlining mundane tasks, and capturing data at blazing-fast speeds. These functions developed into an industry that was appropriately referred to as *data processing*. In the '90s, however, *information technology* (IT) replaced data processing as the preferred term for corporate computing. As participants in a global economy as well as managers of technology, information workers of this decade must focus on the strategic value of *information*, confronting issues of using data rather than simply processing it.

That's where the data warehouse comes in. Thanks to advances in relational-database, parallel-processing, and distributed technology, IT is now poised to live up to its name. A client/server or host-based architecture in which users on any number of platforms or terminals access a wealth of information on a host system, the data warehouse leverages the investments most companies have already made in operational and legacy systems and provides a solid foundation for corporatewide decision-support systems (DSSs) and other strategic business activities.

Integrated Information

In simple terms, a data warehouse is a large database that holds integrated data from an organization's operational databases—typically, its online transaction processing (OLTP) systems. One of the key words in that loose definition, the word that lies at the heart of the data warehouse, is *integrated*. The architects of a data warehouse must transform and integrate operational data and possibly data from outside their company and then place it all in the data warehouse, where it will be used as a strategic tool by the users on the front lines—the ones making key business decisions based on the information available to them.

This transformation and integration process can be the most challenging part of building a data warehouse, given the nature of most operational systems and the design requirements of the data processed in them. Aaron Zornes, senior vice president and director of application-development strategies at META Group, a Westport, Connecticut, consulting firm, characterizes operational systems as "vertically challenged, or *stovepiped*, applications—top-to-bottom, closed applications that automate a specific function, such as customer billing, order entry, or expense reporting."

Information in the data warehouse is by necessity of a very different design. In his book *Building the Data Warehouse*, Bill Inmon, vice president of Prism Solutions (Sunnyvale, California) and recognized industrywide as the originator of the data-warehouse concept, describes a data warehouse as "a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process."

The move toward data-warehouse architecture is fueled by the same business drivers that are affecting organizations everywhere—chief among them the need to compete in a global economy, bringing better products or services to the shrinking market faster than the competition without increasing the cost of the product or service. Changes in regulatory policies in industries such as insurance, utilities, and banking have also eroded established markets, heightened the level of competition, and chipped away at profit margins.

Meeting such challenges depends on a company's ability to leverage its investment in people, resources, and technology. The data held captive in a company's operational systems is one such resource, but generally speaking, it rarely serves as a resource in its original state. It's by pulling data from the operational systems and integrating it into a data warehouse that an organization turns operational data into a strategic tool. Integrated, analytical information—rather than raw data—enables an organization to make mission-critical, strategic business decisions.

Accessing Operational Systems

Decision-support analysts began dipping into legacy systems years ago, using various so-called data dippers—desktop GUI clients—to access mainframe data. Such methods don't produce the strategic results possible with a data-warehouse architecture, however. Given the appropriate tool, it was relatively easy for users to connect to and access operational data via data dippers, but getting usable information out was another matter, given the nature of data structure and content in an operational system. For example, to achieve maximum OLTP performance, such a system might have data dispersed on multiple systems and might represent different update intervals. In addition, a data element in one system might have a different meaning in the context of another system.

The bottom line is that, even though business analysts could dip into their store of data, it didn't effectively inform their decision-making process. If the product managers for a retail clothing chain were trying to cull the necessary information from an operational system to help them decide which month to discount winter coats in a certain region of the country, they would challenge the limits of the tool (and their analyst's patience). Particularly if they then decided that they really would like to see the same analysis, but this time looking only at wool coats sold in women's sizes 6 through 10.

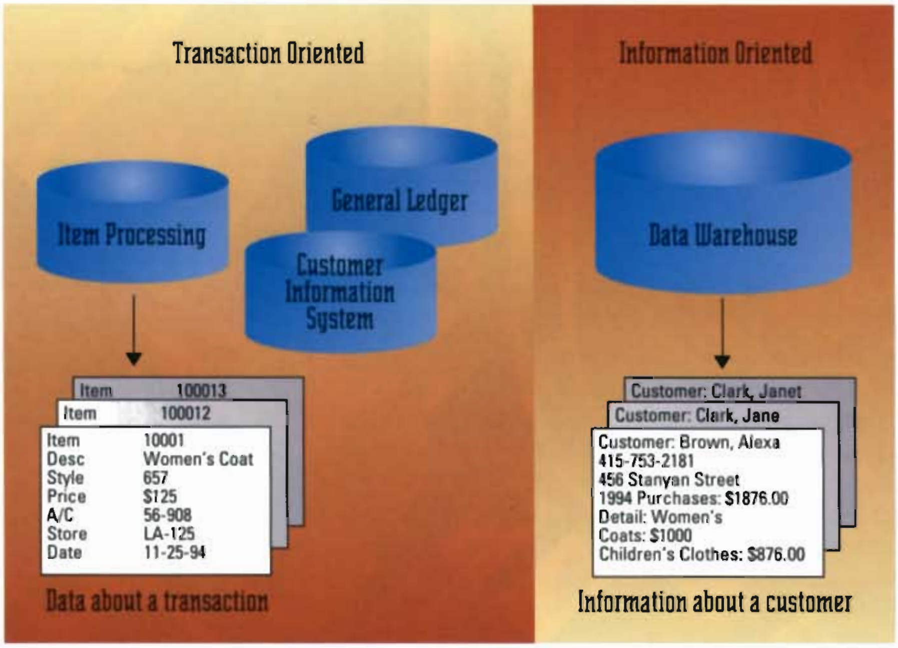
In addition to issues of data quality, there is the always thorny subject of performance: Running complex queries against a database designed for high-input OLTP can slow down your system. With a data warehouse, on the other hand, users can easily slice and dice the information it houses in a variety of ways (see "Dressing Up Data," on page 46).

"Retail needs to know who's buying and what promotions work and don't work," says Kevin Strange, research director at the Gartner Group, a consulting group based in Santa Clara, California. "Banking has been using data warehousing to target credit cards and loans, based on demographic analysis. Insurance companies are looking for parameters for fraud and patterns of health care. A company like General Mills might use information in the data warehouse to analyze sales patterns for Cheerios cereal, and, discovering that stores in Los Angeles aren't selling Cheerios, they could direct coupons to that area to move the inventory."

Although data warehousing has typically been associated with industries such as financial institutions, utilities, insurance, and retail, the merits of this architectural approach can benefit virtually any organization, according to Prism Solutions' Inmon. He cites as an example the Ontario (Canada) Lottery's implementation of a data warehouse to keep track of promotions as compared to sales. Because turning operational data into strategic information is the whole point of the data warehouse, any organization that has high volumes of operational data—many transactions, large customer files, and unit item sales—should be able to take advantage of a data warehouse.

Inmon also notes that although sales and marketing is typically a prime area for an initial data-warehousing project—because data warehousing enables those departments to pull information from many sources and quickly build successful implementations—data warehousing is not just for sales and marketing. Controlling expenses within any business becomes much easier if you can look at those expenses in a variety of ways.

FIGURE 1: OLTP data differs from data-warehouse information



Item 100013

Item 100012

Item 10001
Desc Women's Coat
Style 657
Price $125
A/C 56-908
Store LA-125
Date 11-25-94

Customer: Clark, Janet

Customer: Clark, Jane

Customer: Brown, Alexa
415-753-2181
456 Stanyan Street
1994 Purchases: $1876.00
Detail: Women's
Coats: $1000
Children's Clothes: $876.00

OLTP Versus the Data Warehouse

Let's say those retail-chain product managers want to clear out their inventory of winter coats and need to find out which of their stores are likely to sell which kinds of coats and at what prices. A typical operational system for processing customer orders might be deep with detail about orders, but it provides no long-term historical perspective. Operational systems are constantly changing as clerks enter new orders. Furthermore, the chain might have no outside demographic information about its customers, so it wouldn't know what category of consumer was buying particular coats.

If the company had a data warehouse, however, customer information extracted from various databases in the chain's operational systems—the customer-information system (CIS), accounts receivable, and sales, for example—would be accessible as a single integrated topic, or subject, called Customers.

Whether mandated from the top levels of the organization as part of a major infrastructure project (see the sidebar "Alliance BlueCross BlueShield") or initiated by a single department within the company to meet specific needs, a successful data warehouse is one that is modeled through the collaborative effort of many parties, including DSS analysts and database designers. With the data-warehouse model in hand, the designers then decide from which of the company's existing systems—OLTP databases, departmental relational-database applications, or legacy systems—to obtain the raw data for integration into the warehouse and then program the appropriate applications to extract the data from those systems. Data from sources external to the company—for a retail chain, these might include TRW reports or survey data from market-research firms, for example—also might be consolidated into the warehouse.

In the process of integrating all the appropriate information from the various sources while building the data warehouse, the designers must reconcile differences among naming conventions, measurements, encoding structures, and physical attributes of the data, so that the consolidated information is meaningful. Users can then obtain from the data-warehouse system a read-only snapshot summary of the extracted collection of information about a particular subject—coat sales, for example. The data warehouse contains these subject snapshots from various points in time, typically spanning a period of 5 to 10 years—as opposed to the operational system, which usually holds data for only 30 to 90 days worth of transactions (see Figure 1).

As seen in Figure 1, a retail chain's data warehouse might contain a subject database for its customers that puts together contact and credit information from the CIS database and compiles a purchasing history from a combination of the information in the accounts-receivable system and the order-processing system. In addition, a data-warehouse customer database might include demographic information based on regional location, credit information, or information from other external sources. With this system, a simple query would give the chain's product managers exactly the information they needed to make a profitable business decision.

Hardware and Software Issues

Issues you must consider when building a data warehouse include the design of the system and the data, the selection of database software and toolsets, and the choice of the hardware platform on which to run them. As always, behind each of these is the issue of cost: finding the optimal solution at the best price.

Building a data warehouse is much easier today, thanks to parallel implementations of RDBMSs and symmetric multiprocessor (SMP) or massively parallel processor (MPP) open-systems hardware from vendors such as Hewlett-Packard, Sequent, Digital, nCube, and IBM. The combination of parallel software and hardware provides the means to process complex queries rapidly at a low cost on readily available systems (see the sidebar "Oracle7 Release 7.1 Parallel Everything Architecture"). In fact, META Group's Zornes cites the availability of cost-effective midrange UNIX and LAN RDBMS servers as a key enabler of the data warehouse. Parallelism also offers the mission-critical level of availability—better than 99 percent—found on the mainframe, but more cost effectively, with remaining processors in a parallel system covering for any processor that might fail.

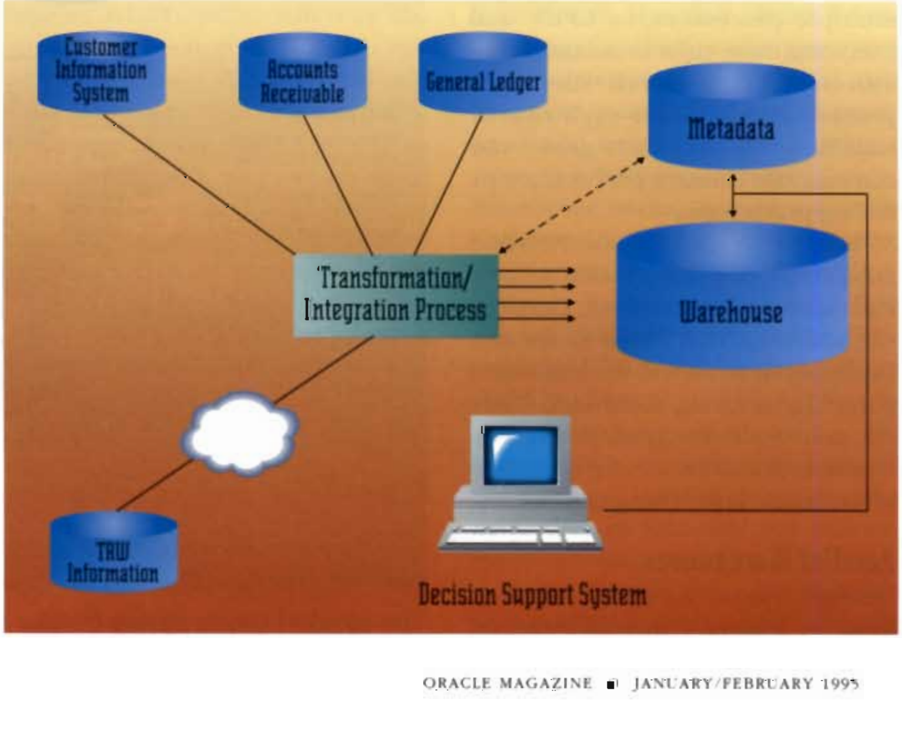
Software issues also include choosing your suite of end-user query tools, either off-the-shelf or by building your own, and choosing tools that let you put the data-warehouse architecture together.

Building the Structure

Data sources for the warehouse can include an eclectic collection of legacy and server systems. For example, in Figure 2, the General Ledger system might be Oracle7 running Oracle Financials on an HP 9000 server; the Customer Information System could be running on an IBM AS/400 system; and the Accounts Receivable system might be your company's original IMS database running on a mainframe. To build an Oracle-based data warehouse, you would extract data from each of the non-Oracle sources by using the Oracle gateway designed for the source, and you would extract the data directly from the Oracle7 server running Oracle Financials.

But building an effective data warehouse that will serve as your corporatewide decision-support platform requires more than simply dumping operational-systems data into one very large database. Given the distinct

FIGURE 2: A view of the data-warehouse architecture



Oracle7 Release 7.1 Parallel Everything Architecture

With its parallel-everything architecture, Oracle7 Release 7.1 is poised to provide the foundation for the data warehouse. Oracle7 is the first commercial relational database system available on both symmetric multiprocessor (SMP) and massively parallel processor (MPP) systems, such as those from nCube, Hewlett-Packard, IBM, Sequent, Pyramid, and more than a dozen other vendors.

As the name implies, the Oracle7 Release 7.1 Parallel Everything architecture brings parallel-processing capability to all functions in the RDBMS environment, from loading data, creating indexes, and querying to backing up and recovering the database. Parallel processors offer significant improvements in computing performance by splitting tasks among multiple processors, or CPUs, and executing these tasks or subtasks concurrently. In general, the more processors, the faster each overall task runs; as workloads grow, you can maintain constant performance by adding processors.

Because Oracle7 Release 7.1 takes advantage of parallel architecture transparently, programmers, developers, and users don't have to do anything special to exploit the hardware's parallel-processing capabilities. Nor do you need to modify existing Oracle applications to take advantage of the parallel-processing capabilities.

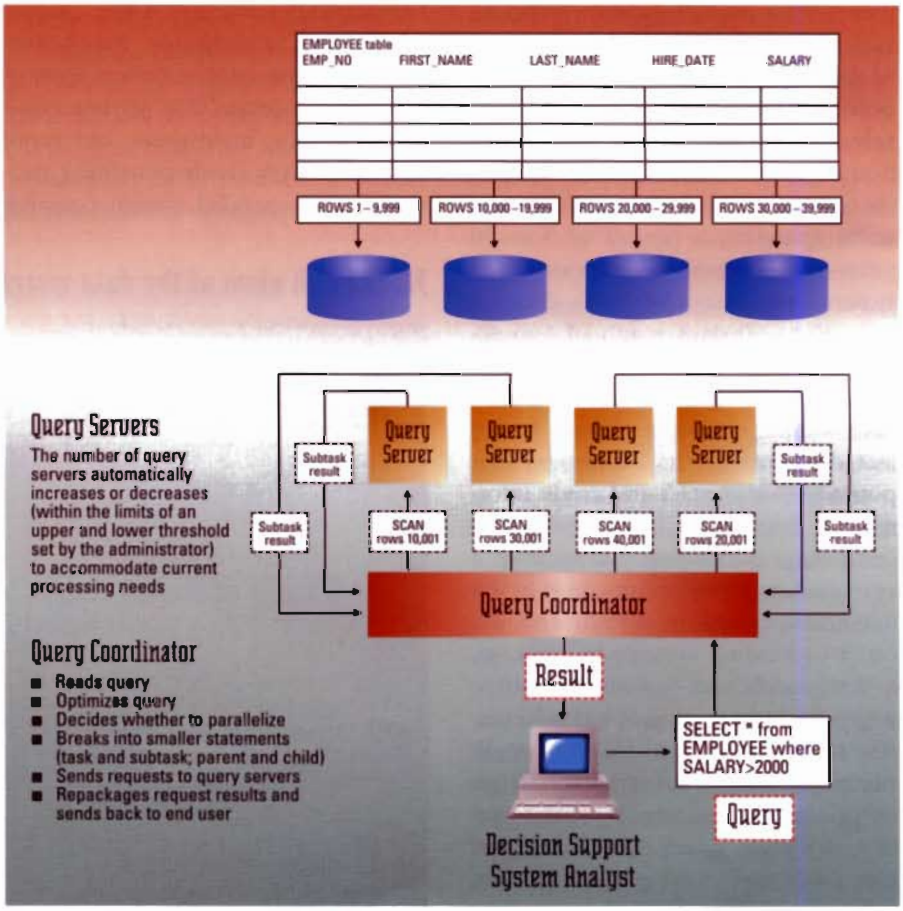
Parallel Warehouses

Oracle7 Release 7.1's parallel-everything architecture provides several key benefits for data-warehouse environments. For starters, it enables users to greatly reduce the time-consuming task of loading the data warehouse—and warehouses can contain hundreds of gigabytes of data—by using the parallel load utility, which uses all available processors simultaneously to load rows of a table. After the table is loaded, Oracle7 Release 7.1 creates its index, again using multiple processors to index segments of the table at the same time.

Most important, however, is the parallel query option. Typical decision-support queries often draw upon information in several large, very often nonindexed tables. In a nonmultiprocessor environment, such queries can become runaway queries and take many hours to process, bogging down the system to the point where the cost of retrieving the information outweighs any potential benefit it may offer. The parallel query option available in Oracle7 Release 7.1 improves response time for decision-support queries by breaking up each query into multiple units of work that can be processed simultaneously by the multiple processors or nodes of the hardware.

What Does a Parallel Query Do?

The parallel query option comprises a query coordinator (QC) and a pool of query servers. The QC intercepts



SELECT * from EMPLOYEE where SALARY>2000

Query Servers

The number of query servers automatically increases or decreases (within the limits of an upper and lower threshold set by the administrator) to accommodate current processing needs

Query Coordinator

- Reads query
- Optimizes query
- Decides whether to parallelize
- Breaks into smaller statements (task and subtask; parent and child)
- Sends requests to query servers
- Repackages request results and sends back to end user

The parallel query option breaks a query into multiple units of work queries and decides whether to split them into multiple streams (see the figure on page 38).

If necessary, the QC breaks the SQL query into smaller, task-and-subtask (or parent-and-child) statements. The QC then sends these to available query servers for processing; the query servers pass the results of the subtasks back to the tasks, and so on, until all constituent tasks are complete. The results return to the QC, which repackages the information and returns the answer to the DSS analyst via a query tool.

MPP or SMP cluster hardware can accommodate multiple query servers per processor or node. Thus, in the figure on page 38, the query servers pictured may be running on a single node in an SMP cluster. The number of query servers per node isn't static but is dynamically increased and decreased in order to provide optimal CPU utilization. For example, if a heavy-duty query is using only 50 percent of Node A and is using 75 percent of Node B, Oracle7 Release 7.1 will initiate additional query servers on Node A.

Likewise, Oracle7 Release 7.1 partitions queries across all CPUs dynamically. Unlike a static partitioning scheme, which assigns portions of a database to specific CPUs, the Oracle7 parallel query option uses available CPUs equally to process queries. (In a static scheme, if a query touches only one section of a particular database and that section is assigned to CPU No. 3, CPUs Nos. 1, 2, and 4 will be idle while CPU No. 3 does all the work.) Dynamic partitioning results in less time spent processing the same query.

Available Now

The parallel query option is available with Oracle7 Release 7.1. It does not require Oracle Parallel Server. The parallel query option with Oracle Parallel Server is designed to run on clustered or MPP hardware; without Oracle Parallel Server, the parallel query option is designed to run on SMP hardware.—*KW*

differences and possible anomalies among each of the source databases, you must do a great deal of cleaning up—scrubbing—of data before you can populate the data warehouse. In fact, the bulk of the work in building a data warehouse lies in thoroughly analyzing the source data and processes; you must understand the processes of the operational systems in order to accurately capture the business rules (see the sidebar "Beyond the Data Warehouse").

Figure 2 shows a generic architectural view of the data-warehouse environment. Metadata, shown as a separate database in the figure, is a key element of the data-warehouse architecture; it holds all of the data about the data, as mapped between the source and the target system. An Oracle7 database houses the metadata in an Oracle table. Metadata correlates the data from the source into the target. It holds information about the operational data elements (where they come from, for example), the data definitions for the target database and the data warehouse, and the transformation/integration logic.

Transforming and integrating the data might involve converting values or converting data among various platforms and databases. For example, a column in an Oracle7 database of financial information might be labeled CUSTOMER, whereas the same information in a DB2 G/L system might have a different label and data definition: alphanumeric, say, rather than simply defined as a character string as it is in the Oracle7 database. In this case, to integrate the data into the target, you might convert one of these datatypes to the same type as the other in order to be able to include information about customers from both sources.

With the metadata in place, you can extract data from the operational systems; scrub or otherwise repair it; and then summarize, sort, and organize it before loading it into your data warehouse. Much of the work in building the data warehouse is in upfront analysis of your operational systems and the data they contain. The various operational systems in a company, having been developed at different times over the years, most likely have inconsistent naming conventions, measurements, and coding schemes. Modeling the data that will be housed in the data warehouse is therefore the most critical part of this process (see the sidebar "Seven Steps to Building a Data Warehouse").

Tools of the Trade

You'll find many tools available to help build a data warehouse. You can use various CASE tools, such as those available as part of Oracle's Cooperative Development Environment (CDE), to develop the initial requirements for the warehouse. With participation from your decision-support analysts and others in your organization, you can then construct graphical maps of the proposed data model; other CASE tools can generate end-user applications based on the information created and captured during the development process.

But the real challenge to a successful warehouse lies in effectively extracting the data from the source databases, integrating that data into the target data warehouse, and managing the metadata that ties the two together. That's where third-party tools come into play. For example, Warehouse Manager, from Prism Solutions, automatically generates code to integrate, transform, and map data from source databases to mainframe and server databases running a variety of RDBMSs, including Oracle. Warehouse Manager; Passport, from Carleton (Burlington, Massachusetts); the Extract Tool Suite, from Evolutionary Technologies (Austin, Texas); and Integrity Programming Environment, from Vality Technology (Boston, Massachusetts) are all software tools designed to manage the burdensome aspects of warehouse maintenance, including managing the extraction of data and the maintenance of metadata.

Each of these products supports many different databases as a source data warehouse does not provide integration with operational systems to allow users to act upon decisions being made. This type of integration requires corporations to go beyond the traditional data warehouse.

Details Make the Difference

Corporations can often summarize data to serve the needs of a particular group. Summary data often has limited value, however, because it reflects the specific business process of that particular group. For example, summarization of sales activity by week, even at a store level, cannot account for daily occurrences such as advertising, weather, competitor activity, or other unscheduled events that may have a short-term impact on customer behavior. The finer the granularity of the data available to a company, the broader the potential use—and the more accurate the view.

Data Versus Applications

As a business changes, it must develop new applications to implement new strategies and organizational structures. But the data a company uses to run the business doesn't change. Therefore, companies need to separate their data-management infrastructure from the applications used to access it. Companies should develop an "infostructure" for data that is flexible in accessibility and use, so that their most valuable resource doesn't get locked away from users as the system evolves.

Not Just a Warehouse

A data warehouse is the beginning of an evolution—data has to be efficiently gathered into the warehouse, new applications must be fully integrated into the new environment, and the entire client/server environment must be effectively managed as a single entity. AT&T calls this complete, fully integrated environment an Enterprise Information Factory. An Enterprise Information Factory goes beyond the current notion of a DSS by integrating operational applications. To accomplish this, businesses must carefully design and implement the following critical elements of the data-warehouse environment:

- Data movement and transformation between legacy transaction systems and the new factory infrastructure. The customer information used by new operational applications must be more current than the information that has traditionally been available with batch-updated DSSs.
- A business-information directory. This can help manage the complexity of the new environment—where data structures, new application code, and underlying business-process rules must be synchronized.

Converging Environments

To make this converged environment successful, companies must take several vital steps:

- Link the project to key business challenges. Input and support from senior management and end users are key to accurately forecasting and developing a road map to success.
- Develop the proper data model with consistency among functional organizations. This model must allow the data warehouse to evolve to a converged environment without major changes or restarts.
- Plan for customer data that is as detailed as possible—even to the level of individual transaction records.
- Plan for rapid growth. As users begin to recognize the power of the information they now have within their grasp, they will demand more access to data. And as the business grows, the number of transactions that will need to be captured and managed will also grow.

Going beyond the data warehouse to a carefully converged Enterprise Information Factory based on detailed data requires more careful planning and implementation but enables business to go beyond strategic decision-making to taking action. And the faster a company can act, the more competitive it can be.

Robert Tuttle is director of Decision-Enabling Worldwide Marketing for AT&T.

and a target; potential users should evaluate these and any other products in the context of their own information-technology infrastructure.

Working With the Warehouse

The writing is on the wall. Organizations that plan to ring in the new millennia with a healthy bank account recognize the strategic value of information, whether their bottom line currently needs bolstering or not. Such organizations also recognize that their OLTP operational systems are a ready source of data that they can transform into strategic information in a data warehouse.

Implementing a data-warehouse strategy is no longer cost-prohibitive or technologically difficult, given the availability of parallel software architectures and high-performance SMP and MPP open-systems hardware. These systems provide the processing power necessary to handle the complex queries that decision-support analysis usually entails.

Again, the fundamental challenge in designing the data-warehouse architecture is the analysis of your organization's existing operational systems and the mapping of that data to the data-warehouse target. The complexity of this task depends in large part on how well your company's systems were designed to begin with. Nonetheless, you can pick from a wide range of tools for extracting data from mainframe or legacy systems. Toolsets are also readily available for transforming and integrating the data and for managing the metadata. When implemented as part of a corporation's core IT infrastructure, a data warehouse can benefit virtually every department in the company—from marketing and sales to purchasing to customer service to the actual warehouse itself—by providing strategic information that gives greater insight into business trends and business opportunities. ◘

Kelli Wiseth is a San Francisco-based freelance writer who specializes in network solutions and other enterprise issues.

Alliance BlueCross BlueShield

With almost 800,000 members, three million claims to process each year, and more than 350 commercial health care providers competing in its marketplace, Alliance BlueCross BlueShield (ABCBS) has to move fast these days just to stay in one place. Although satisfying customer needs while remaining profitable may be typical business challenges, the health care industry faces additional obstacles in the U.S., given the ongoing national debate over health care access and cost.

In the meantime, ABCBS remains committed to providing access to affordable, quality health care. The company has controlled costs over the past few years by replacing straight indemnity programs—go to any doctor, file a claim, get reimbursed—with managed benefits programs such as BlueChoice HMO (health maintenance organization) and Alliance PPO (preferred provider organization).

Managing these managed-benefits programs is no small undertaking, with vast amounts of data about health care providers, members, claims, and more to track. Executives at ABCBS realized that they needed a solid foundation of information on which to make strategic decisions about their product line. With vital support from the executive level, the company adopted what it calls a Strategic Data Initiative (SDI), whose mission statement is "to identify, define, maintain, and support a decision-making environment for Alliance BlueCross BlueShield by providing the information needed to measure and manage the enterprise."

The Strategic Data Initiative

One of the key components of the SDI, the data warehouse provides the architectural foundation for modeling, mapping, summarizing, and integrating enterprisewide data into a consistent database of historical information. In the process of building the first generation of its data warehouse, ABCBS is getting a handle on all the differences among the data structures in its operational systems. "We spent over 30,000 hours among 20 people. We spent most of the development efforts getting data out of the old system," says John Ladley, project director for ABCBS.

Although the data-warehouse project was first proposed several years ago, it wasn't officially launched until late in 1993. "Corporate management wanted access to information; they wanted a uniform set of measures and business definitions. And they wanted to make the information available to as many people as appropriate," explains Ladley.

ABCBS chose Oracle7 Release 7.1 because of its parallel-everything architecture. The company's evolving information infrastructure will be a 200-gigabyte Oracle7 data warehouse that enables line-of-business managers to measure product performance, analyze claims, and so forth. At press time, ABCBS's data warehouse is between 20 and 50 gigabytes and focuses on three subject areas; ultimately, the warehouse will be a foundation element for every business unit in the company.

A Data Warehouse in Action

First, ABCBS applications clean up and scrub data from provider, claims, membership, and billing operational systems, in both legacy IMS files and DB2, and integrate the data on the mainframe. It takes anywhere between eight and ten hours to build the staging files on the mainframe.

To make the data warehouse as transparent as possible to OLTP systems, all the data transformation occurs on the operational end. "Data in the warehouse is presummarized for performance-type reporting," says Ladley, "but we also keep the detail level available for dimensional analysis." Ladley points out that this has the benefit of keeping the whole system modular; changes need be made only on the mainframe part of the system. When the system regenerates the warehouse data, it automatically reflects the changes.

Before downloading the data, ABCBS produces a report and looks at the numbers to make sure that the information is accurate. Loading the data into the data warehouse is a relatively simple matter of performing an Oracle utility load.

ABCBS uses Prism Solutions' Warehouse Manager to perform the transformations, integrate the data, and manage the metadata. The company also developed an end-user GUI query tool, using Powersoft Corporation's PowerBuilder. Development and implementation occurred with relatively few snags, says Ladley. "We had a few incompatibilities [because of version problems] between PowerBuilder and the Oracle RDBMS, but we were able to work around them."

The data warehouse not only enables ABCBS's decision-support analysts to study membership trends and analyze financial performance by product line but allows users to find out how certain products are doing in different parts of the state.

"We feel very confident that the system is going to meet all expectations," Ladley says. "People are looking forward to this system. From the top down, the whole company supports the SDI, and the data warehouse in particular." —*KW*

Seven Steps to Building a Data Warehouse

Experts in data-warehouse technology point out that nothing highlights the GIGO—garbage in, garbage out—maxim better than a poorly implemented data warehouse. It's a mistake to underestimate the amount of analysis you must do of your current corporate operational systems in order to understand how to transform and integrate your data before populating the data warehouse. You should conduct in-depth examinations of business rules and policies; this is often accomplished by reverse-engineering the application code that runs your business. (As a byproduct, this analysis may enable an organization to uncover anomalies in its legacy production systems.)

There are many different ways to build the warehouse, but every implementation involves work in four key areas: analysis of the data sources; definition of the transformation and integration processes that need to occur on that data; construction of the warehouse itself; and implementation of the tools users will employ to get into the warehouse and pull out the information they need. Although there's no cookie-cutter method, the following steps outline a generic approach to building a data warehouse.

1 Determine the needs of your end users and model the data that the data warehouse should contain. For example, a retailer may want to be able to target existing customers for specific ongoing promotional activities. To do this, the business or marketing analysts will need to query the database and glean demographic profiles of existing customers and their purchasing patterns. Depending on the frequency of the promotions, the analysts may also want that information organized by various time periods. CASE tools can help by presenting a high-level, graphical entity-relationship diagram showing the proposed data model in an easy-to-understand format.

2 Identify the necessary data sources from among the many corporate data sources that are available to your company, based on the most apparent needs of your end users. Appropriate sources for a retail-chain business, for example, might be the customer-information system housed on a mainframe, the accounts-receivable system running on an SMP server, and demographic information on a networked CD-ROM purchased from a market-research firm.

3 Analyze the corporate data sources in depth, documenting the functions and processes that these data sources capture. J.D. Welch, a consultant for Prism Solutions and the designer of one of the first commercial data warehouses, for a California telecommunications company, notes that one of the most important aspects of designing a data warehouse is understanding the rules that drive the business. "You have to understand how the data behaves so you can decompose those business processes until you get to your data elements," Welch says. "It's like doing analysis for any system, except it's information-oriented instead of process-oriented."

4 Use the information about the corporate data sources to decide the transformation/integration logic necessary to create the proposed data-warehouse data models (the target) from the source. Issues to be concerned with are how much data you will extract and transform; whether the integration involves complete files or files containing changed data only; and how frequently the transformation should take place. Welch says that the "significant business events" should determine the frequency of transformation or update. Whether updates are event- or time-driven, you need a mechanism that lets the data warehouse know that something happened that needs to be captured.

5 Create the metadata, which identifies the source data, describes the transformation and integration that needs to occur, and defines the data model for the warehouse. CASE repositories should allow data definitions, business rules, and detailed logic to be modeled for distributed systems development. Metadata becomes the business analyst's guide to what is in the data warehouse and how it got there.

Several third-party products, including Carleton's Passport, Evolutionary Technologies' Extract Tool Suite, and Prism's Warehouse Manager, are designed to streamline and manage this most critical aspect of data-warehouse development. These products can also assist, to varying degrees, with the next two steps.

6 Create the physical data-warehouse database and then populate the warehouse from the various sources. Gateway software enables the extraction of data from a variety of legacy systems. (Oracle7 Release 7.1's replication feature enables you to populate the data warehouse directly from another Oracle database.)

7 Generate the necessary end-user applications or in some other way provide your end users with query tools that give them access to the information in the data warehouse.—*KW*