Decision support at Lands' End—An evolution

by G. G. Bustamente
K. Sorenson

A decision support system with over one billion rows of data has been developed at Lands' End using the IBM DATABASE 2™ (DB2®) relational database management system. This corporate database is a subset of an Information Warehouse™ framework and functions as both a decision support system server and an application enabler. The corporate database uses operational data gathered from order processing and customer mailing systems. Weekly processes reformat these real-time data for loading into the corporate database. This paper discusses some of the business requirements that guided the development of the corporate database, and also describes the database design process, tool selection, and implementation experiences.

Lands' End, Inc., is a direct marketer of traditionally styled quality clothing for adults and children through its catalog. Known for its excellent customer service, with every item guaranteed, Lands' End** constantly seeks new ways to improve customer satisfaction while continuing to increase sales and reduce costs. With over $800 million in sales in 1993, Lands' End services approximately 17 million customers, primarily by phone but also by mail and electronic media.

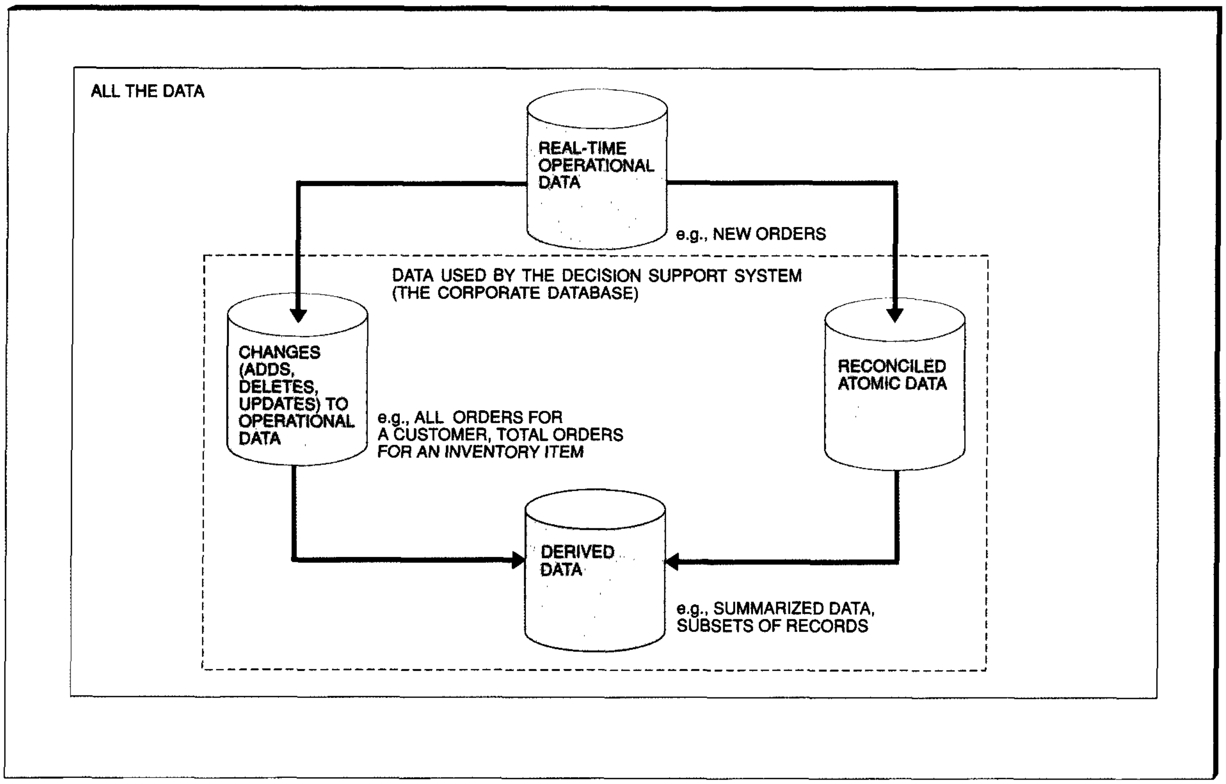
Because of the cost to produce and deliver a catalog to its customers, Lands' End must precisely target a customer for a mailing. Variables such as lifetime customer value, geographic location, recent customer sales, and others are factored into a selection process to identify customers to receive a catalog.

In its 30 years of existence and as a result of its phenomenal growth, Lands' End has accumulated a vast store of information. Lands' End processes as many as 80000 orders a day. Each order represents several records that are archived and migrated from the operational order processing system to magnetic tape.

Before the modernization of these processes occurred, a number of problems existed. Single-threaded processing of information, constituting over 125 uncompressed cartridge tapes, caused contention with other processes that required those same files. Contention occurred with other client requests for information on the same files. Many of the requests required the reading of multiple files, thus prohibiting access to the tape drives for long periods of time. To address the disadvantages of tape processing, subsets of the tape files were created on direct access storage devices (DASD) as an interim solution.

Personnel, or "clients," from merchandising, finance, and marketing analyzed the data by product, fiscal period, and customer. To perform their analysis, a lengthy process of data capture was necessary that restricted the amount of analysis time before the next project. Time constraints of-

Figure 1 Lands' End Information Warehouse implementation structure



ten did not allow iterative processing of the decision support data. Because of the static structure of the files, the unavailability of data, or the cost of changing the current system, several client requests to enhance this early decision support system were not feasible.

The creation of a corporate database at Lands' End encompassed the entire set of information available to the company with the exception of the day-to-day, real-time operational data. To support the use of the corporate database, a DATABASE 2* (DB2*)-based decision support system was developed, consistent with the three-level approach used by the IBM Information Warehouse* framework described in Reference 1.[1] The three-level approach uses derived data (data that have been summarized and joined from multiple tables[2]), such as summarized data for a specific product, and reconciled data (adds, deletes, and updates to real-time operational data), such as all the orders for a specific customer. Figure 1 illustrates the approach used.

This paper discusses the development of the decision support system (DSS), reports on the experiences learned, and concludes with some thoughts about future directions that Lands' End will investigate.

The corporate database

When they were originally designed, the processes that analyzed information at Lands' End were state-of-the-art processes. To remain competitive, however, the need for timely decision making and the ability to respond to rapidly changing market opportunities became the catalyst for modernization of these processes. Five factors led to the decision to develop a corporate DSS database within the Information Warehouse framework (see Figure 1).

The first factor was the operational complexity of accessing data in different locations. Some of the data were on tape files, other data were on VSAM (virtual storage access method) files, while still other data were located on IDMS**, a network database. Processing this information in a meaningful way required the technical knowledge of an information services (IS) analyst. Only clients that have the requisite technical skills could be expected to gather and process their own information.

The second factor was that much of the DSS information was stored on sequential media, DASD, and tapes, and not indexed to support ad hoc queries. This required processing much more of the data than was necessary, with a corresponding burden on computer resources.

The third factor was the long turnaround time for queries. This is somewhat the result of factors one and two but is mentioned here because it is separately perceived as important by a client. After making a request for information from an IS analyst, a client should not be concerned with where the data reside or if the data are efficiently indexed. The satisfaction of the client with IS services is a function of the timeliness with which the request is answered and the accuracy and consistency of the information provided.

The fourth factor was the proliferation of file descriptions and the amount of redundant data. There was no centralized data administrator function and thus the benefits of a central point of control over the definition of data were not fully realized. Different areas of IS and client departments had their own definition of data items. This led to confusion, increased application program maintenance, and inconsistent reporting.

Changing technology was the fifth factor that led to the development of a corporate database. The processes in existence at the start of this project were state-of-the-art at the time of their development, but time changed the definition of state-of-the-art. Thirty years ago when Lands' End started, there was no such thing as a gateway concept, industrial strength relational databases, or powerful workstations with query tools.

Dan Rourke, the Lands' End Vice President of IS, recognized the time had come to take advantage of these advances in technology. He challenged the staff to apply the latest technologies to solve Lands' End's business problems. This included developing a platform that would serve as both an application enabler and a DSS server. One of the results of this challenge was the decision to develop a corporate database in an Information Warehouse framework using the IBM DB2 platform.

The corporate database would enable Lands' End to achieve the following goals:

- Improve flexibility, timeliness, and accuracy of the corporate decision reporting
- Position IS in a more proactive posture for supporting the needs of Lands' End management to develop and enhance market strategies
- Effectively utilize technology for rapid client responses
- Improve the ability to proactively respond to competitive trends
- Implement a process that is flexible and comprehensive to support new and improved marketing functions beyond the year 2000
- Reduce IS involvement in supporting the analysis and reporting needs of business units
- Provide the ability to quickly respond to new business ventures and acquisitions

Benefits of implementation

There are numerous benefits that were realized by the development of the corporate database.

The customer. Because Lands' End will know more about the historical relationship it has with a customer, catalogs that target the customer's specific needs are used for direct mail contact.

Location of data. Because the atomic level data[3] and summaries of the data are located on a single, relational platform, the data are easier to access. Given the plethora of tools designed for client use in accessing DB2, an IS analyst that provides information to a client is no longer required. The client is separated from the technical complexity of the platform on which the data reside. A by-product of this is a savings in not having to maintain the data in several locations, a practice that consumes resources and is operationally complex. Adding new, atomic level data to the corporate database does not require changing existing programs in most cases.

Indexed data. Database index utilization is monitored. Highly used indexes are optimized, low-use indexes are dropped. Temporary indexes are created for planned, one-time requests or for short-term iterative analysis. When an index is used, fewer rows of data are read, which reduces the amount of computer resources needed to answer a query while throughput of the system is increased.

Numerous benefits were realized by development of the corporate database.

Application enabler. Locating the data in an Information Warehouse framework and indexing the data facilitates the use of the corporate database as an application enabler. Readily accessible data from a relational platform allows rapid application development. Lands' End is already realizing these benefits in the development of their executive information system (EIS) applications.

Shorter turnaround time for queries. Locating the data in a corporate database, providing the client tools to access the data, measuring those accesses to optimize them, and reducing the need to request the services of IS, all contribute to a shorter elapsed turnaround time for the client. It is also feasible for the client to iteratively query the data based on the results of the last query. What would have traditionally turned into a second or third request by the client of IS is now serviced by the client alone. The tools and methods used to access information in an Information Warehouse framework enable the client to be more productive by making it practical to spend more time performing analysis and less time in the capture and analysis of the data.

Clarification of data items. The inconsistencies and inaccuracies in reporting are reduced, and sometimes eliminated, by agreements on common data item definitions, domains,[4] and data formats.

Legacy systems. The cost of maintaining mature (legacy) systems will be reduced over time as clients become more familiar with and use the corporate database and reap its advantages.

System development and architecture

After reviewing the literature and with the benefit of the IS staff experience, only one other platform was seriously considered as an alternative to DB2. The NCR Corp. Teradata** machine that existed at the time the decision had to be made was examined. After several months of evaluation, DB2 was selected as the strategic platform for eight primary reasons:

1. Industry software support of DB2 is pervasive. Third-party software written specifically for DB2 is now available for human resource, accounts payable, and general ledger applications. Data dictionaries, change management, application development, performance monitors, database administration, utilities, gateways, client/server tools, and even electronic mail extensions are available for DB2.
2. DB2 has internally supported referential integrity.
3. The skills, training, and experiences of the IS staff would be leveraged by the current use of DB2. DB2 was already being used for a financial decision support system.
4. Lands' End would not be tied to a specific hardware vendor. DB2 can be run on the hardware of various mainframe vendors.
5. DB2 supports the Distributed Relational Database Architecture* (DRDA*) protocol for distributed processing.
6. The difference in performance between the two choices (IBM and NCR) was considered negligible when sample queries were executed both serially and concurrently. Several million rows of Lands' End DSS data were part of the test.
7. Lands' End already uses IDMS and DB2. Acquiring the Teradata machine would have added a third database management system to the environment. The operational complexity of managing all three would have required additional staffing.
8. I/O parallelism in DB2 Version 3.1 was expected to significantly reduce the elapsed time of queries.

Database design

In this section both the logical and physical design of the database are discussed, concluding with performance monitoring and tuning.

Logical design. The logical design of the corporate database consisted of two phases. The first phase was led by the data administrator. The second phase utilized the services of consultants who, together with the data administrator, worked with an Operating System/2* (OS/2*) -based data modeling tool.

The first phase consisted of a data modeling effort in which common data items were identified and their meaning and characteristics defined. The Texas Instruments, Inc., Information Engineering Facility** (IEF**) computer-assisted software engineering (CASE) software was used by the data administrator to act as a repository of the data modeling effort. The output of this phase, a logical data model, was the result of the combined efforts of the data administrator, the IS business analysts, and clients.

For the second phase, consultants were hired who produced an inventory of existing customer order processing and promotional data with the goal of identifying the atomic level data that would be available for a new DSS system. The consultants introduced the Silverrun-RDM** (Relational Data Modeler from Computer Systems Advisers) OS/2-based data modeling software to Lands' End. While the consultants conducted a data modeling effort, the data administrator learned to use the Silverrun-RDM software by documenting the existing financial DSS as well as new corporate database reference tables. The data administrator also transcribed the results of the first modeling effort from IEF to the Silverrun-RDM product.

More meetings were held to further define the common data items and refine their definitions. The output of the second phase was a relational data model, both an on-line version and a printed copy, a list of common data items including their description and domain, and a file of data definition language (DDL) for use by the database administrators in creating the physical DB2 objects.

Finally, the data models from the first and second phases were compared. Ninety to 95 percent of the tables and attributes matched.

Physical design. The physical design was performed by database administrators, IS business analysts, and a consultant from IBM. The volume of data, roughly 150 gigabytes or over one billion rows, was an overriding consideration in the physical design. Index choice, denormalization, data partitioning and placement, and DDL management issues were particularly addressed in this phase.

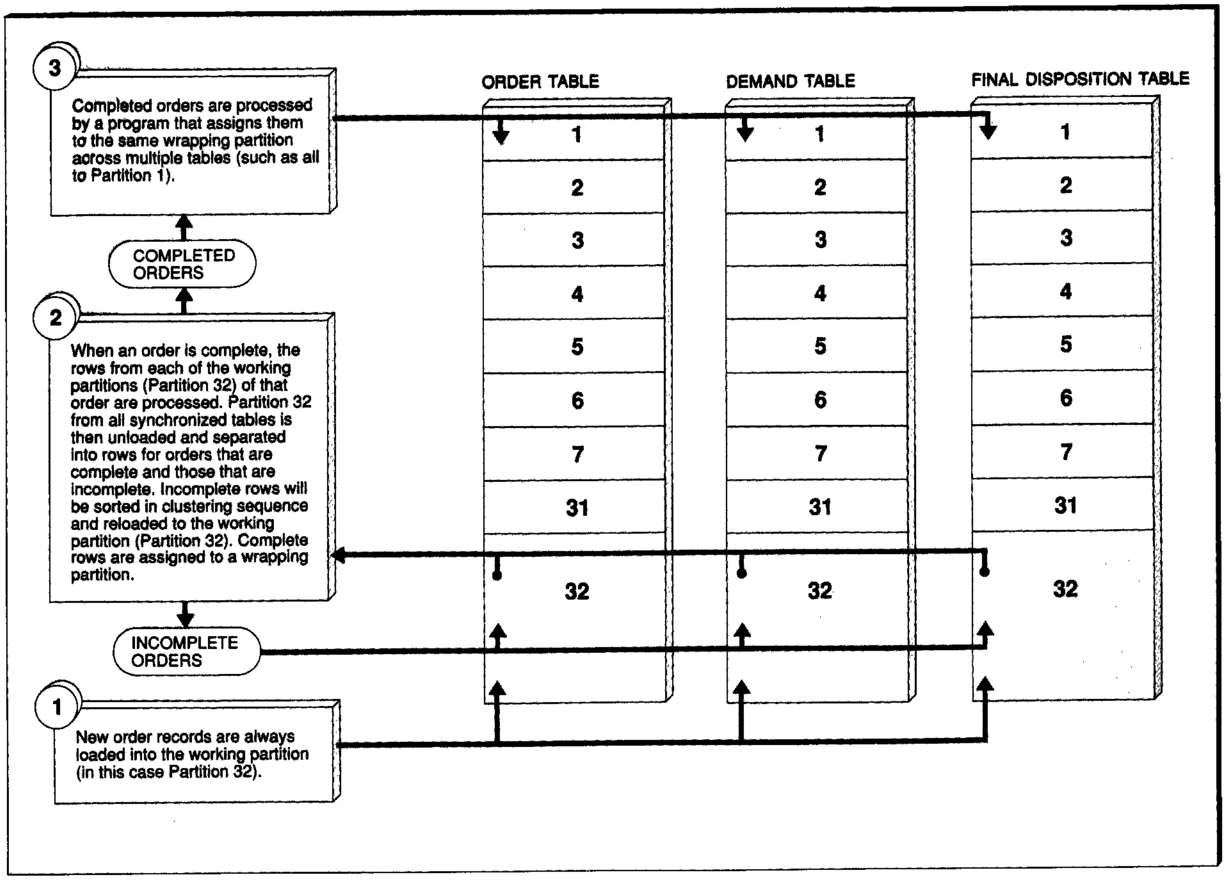
Indexes. Business analysts met with clients to determine the definition of indexes on the tables. Likely access paths were identified for application-specific and ad hoc querying as well as for common joins. Because many of the access paths were not application-specific, identifying useful indexes required intensive analysis. Senior applications analysts were interviewed to determine commonly requested columns, and to combine these columns into composite keys that would receive a high degree of matching to consecutive index key columns. These candidates for indexing were then evaluated according to six main criteria:

1. The estimated probability or frequency of queries using the index
2. The likelihood that the DB2 optimizer, which creates an access plan, would choose the index if created
3. The client and application requirements for the fastest available access
4. The disk space required to store the index
5. The processor (CPU) resources required to maintain the index during utility processing such as loading or reorganizing a table
6. The CPU resources saved by having the index in place for querying as opposed to scanning the entire table

Denormalization. Because of the sheer volume of information, it was necessary to denormalize in some cases from the third normal form.[5] Not doing so would have required additional joins of very large tables, more CPU utilization, and significantly more disk space.

Data partitioning. Large volumes of data required extra resources to manage it efficiently. Following standard recommendations for large tables, and based on our own experience, tables larger than two million rows were partitioned. This was necessary to manage the data in chunks,

Figure 2 Synchronized partition wrapping scenario



usually one partition, when loading. Loading several of the tables would take place on a weekly, and later, nightly, basis.

Three different partitioning schemes were used. Customer-related tables, containing data that would not be purged, used a simple scheme of filling each partition until it was full. The exact number of rows that would fit in a partition was previously calculated. A DB2 table was created that identified the table and partition capacity. This table is read by a program that manipulates the data prior to loading, prefixing the row with the correct partition number. This table also identifies the active partition for each table. Two tables required a specific number of months of history and were handled with a chronological wrapping scenario in which the oldest partition is overlaid with the newest (or most recent) reconciled data from the operational systems.

Business requirements spurred the creation of a previously undocumented synchronized partition wrapping scheme (see Figure 2). This scenario has the unique requirement that several related tables (each in its own tablespace) have their wrapping synchronization based on criteria other than the age of the data. For Lands' End, it was a requirement that all data for a customer order, which exists in multiple tables, be purged together. Item 1 in Figure 2 shows that to accomplish this through partition wrapping, it was necessary that all rows relating to an order be located in the same partition, even though the creation of these rows may occur months apart. To accomplish the partition alignment of an order across multiple tables, a single working partition is used in each table to hold the rows belonging to orders that are not yet complete. Items 2 and 3 in Figure 2 illustrate that once an order is complete, all of the rows related to that order (across multiple tables) will be moved into the current wrapping partition.

Data placement. The final outcome of the physical design resulted in the creation of close to 5000 linear VSAM data sets. The number is high as a result of the large number of highly partitioned tables. To minimize disk contention, maximize throughput, and plan for input/output parallelism, the placement of each of these data sets was individually considered. To assist in this process, the database administrators created spreadsheets. A spreadsheet was developed for calculating the size of tables and each of the four types of indexes. Other spreadsheets were created to monitor the disk requirements at the database level (there are over 70 DB2 databases in the corporate database) broken down by data and indexes. Data sets were spread over the maximum number of disk controllers and volumes. To minimize contention on a given DASD volume, the data and indexes for a given partition are located on separate disk volumes and are offset by a disk controller whenever possible. This was practical because a large number of DASD volumes were acquired at one time.

DDL management. The database administrators used the output of the Silverrun-RDM software to create the physical DB2 objects. The Silverrun-RDM product produced a file of database definition language (DDL) statements that was uploaded to two partitioned data sets on the host processor, one for indexes and the other for tables. The database administrators further separated the DDL using a naming convention that incorporated the logical database number. Finally, the DDL was edited to conform to standard naming conventions.

Database performance monitoring and tuning. Because of the ad hoc nature of DSS queries, no firm estimates were available regarding the number and type of queries that could access the corporate database. Therefore, there was no way to calculate in advance the appropriate DB2 initialization parameters (DSNZPARMS), the proper size for the DB2 temporary database (DSNDB07) or the number and size of the active logs. Guidelines in the DB2 manuals seem to apply to relatively small, operational systems.

To determine these initial values, information was gathered from listening to cassette tapes of the International Database User Group (IDUG) presentations (which were extremely valuable), reading technical periodicals, conversations with other users, IBM support personnel, and the IBM customer electronic link (IBMLink*). Benchmarking queries and drawing upon our experiences with another database management system also proved useful. Information from all of the above sources was used in determining the initial settings for the DSNZPARMS and for sizing DSNDB07 and the active logs.

The staff of three database administrators used information gathered from the IBM System Management Facility (SMF), statistics trace classes 1, 3, and 4 as well as accounting trace classes 1, 2, and 3. Insight for DB2 from Legent Corp. was used as a performance monitor that can access trace classes 1, 2, and 3 with the DB2 Instrumentation Facility Interface. The query language for the monitor was used to create customized requests to collect information to further analyze DB2 performance. These customized requests supplemented the standard set of requests that came with the DB2 monitoring product. Understanding what type of information was available and how to apply it was seen as key to maintaining optimal DB2 performance.

Access

Concurrent to the logical and physical design of the corporate database was the evaluation and selection of a workstation-based query tool and a gateway to DB2. This process was led by another team.

Query tool. Because Lands' End had decided to build applications on the OS/2 platform, the availability of tools was somewhat limited at that time. Believing that tools are replaceable, Lands' End concentrated on functionality and compatibility with whatever DB2 gateway was selected. The Forest and Trees** product from Trinzic Corporation was selected after numerous vendor presentations and analysis of industry literature.

Using the Forest and Trees graphical user interface query product on an OS/2 workstation, an EIS application was developed that provides a snapshot of the company's critical success factors.[6] This application displays second-level derived data on various performance factors available at the click of a mouse. Development of the information-rich screens took days rather than the weeks it would have taken using traditional, mainframe-based screen painters.

The tools available for query result in a development process that is highly iterative. Since prototypes can be created in only a few hours, clients can review preliminary designs and suggest modifications and enhancements early in the development process. This improves communication between IS and the client, reduces rework, and assures the quality of the final product.

The client uses an OS/2 workstation, and DB2 processes the query.

Gateway. The gateway selected had to be compatible with the maximum number of query tools. After research and demonstrations, the CICS*-based (Customer Information Control System*) gateway from Micro Decisionware was selected. Because of time constraints, it was decided that a later study will be conducted to benchmark and test the applicability to Lands' End of using the IBM DRDA protocol to take full advantage of distributed processing capabilities in DB2.

The final hardware and software configuration consists of a client sitting at an OS/2 workstation using the Forest and Trees client/server query tool. From OS/2 on the workstation, the query is routed to the server on the local area network (LAN) on which reside some of Micro Decisionware's gateway software. The gateway then routes the request to the host processor. The mainframe gateway software routes the request to DB2. DB2 performs the query and routes the answer set back to the client following the same path in reverse (see Figure 3).

A typical query might work like this:

Business Question:

> What was the return rate for the women's division, grouped by category, in our Christmas catalog. Compare this year to last year.

SQL Query:

```
SELECT      CATALOG_YEAR,
            CATALOG_DESC,
            PRODUCT_CATEGORY_DESC,
            SUM(RETURNS)/SUM(SHIPMENTS)
FROM        SHIPMENT_TBL D,
            RETURN_TBL R
WHERE       D.ORDER          = R.ORDER
AND         DIVISION         = 'WOMENS'
AND         CATALOG_YEAR IN (1992, 1993)
AND         CATALOG_ID       = 'XMAS'
GROUP BY    CATEGORY_YEAR,
            CATALOG_DESC,
            PRODUCT_CATEGORY_DESC;
```

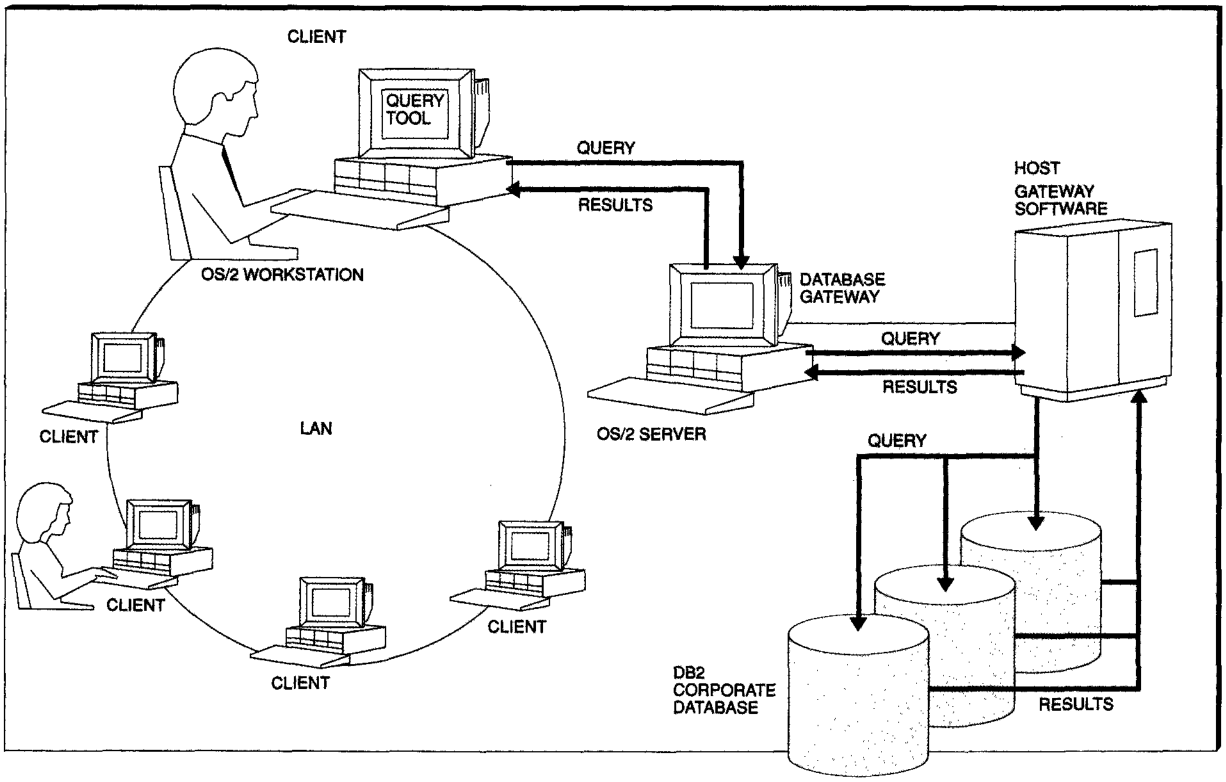
This query would be built by a tool on the client's workstation and passed to DB2 via the gateway. DB2 would process approximately one gigabyte (one billion bytes) of data, summarize it to an answer set of approximately 10000 bytes, and pass it back to the client via the gateway. On the workstation, the client would use available tools to format the information for presentation.

The client may be able to speed up this query by using one of the predefined summary tables. This could reduce the DB2 access by 90 percent or more. The client can also call upon the corporate database business support group in IS to get assistance with constructing and executing this query in the most efficient way.

Expected limitations. The ability of clients to submit queries (such as the one previously described) from their workstation can provide some challenges. While this freedom empowers the client to generate information in a timely fashion, it also presents unique problems in administering a database of this size.

A query for a valid business question that is not properly qualified or that creates a Cartesian product,[7] can become a runaway query. These queries can consume high volumes of CPU cycles or create intensive I/O against DB2 objects, both of

Figure 3 Flow for a typical query



which can have a negative impact on other queries. Our current approach to this challenge is to monitor DB2 threads and cancel runaway queries when they are identified. Adjusting the dispatching priorities of the various DB2 address spaces helped to control the effect of the DSS workload on other non-DSS processes.

Implementation experiences

Designing and loading a large DSS with data that are several years old and creating a client/server access to the data was a learning experience for all involved. Several issues were encountered.

Two problems arose as a result of data integrity. First, whenever an organization sets out to siphon several years of data into a single platform, data integrity problems are encountered. When file and data item formats changed throughout the years, not all changes were rippled through all the historical data. Logical errors when creating the historical data were not discovered until the data were loaded to a single platform.

Second, the sheer volume of data being processed offers a much higher probability that problems with data integrity and program bugs will be encountered. We found that when preparing one billion rows of data, the "one-in-a-million" problem occurs 1000 times. Also, more program bugs, which would normally take years to encounter, are found and must be corrected during the development phase of the project life cycle. Both of these situations were encountered and had major, negative impacts on the project schedule.

Loading a billion rows of data to DB2 takes significant elapsed time, DASD work space for the utilities, and DASD space for the DB2 internal sort database (DSNDB07). The amount of DASD space needed and a plan to have the DASD in place should be determined as early as practical. Loading the data in multiples of partitions, instead of all at once, can reduce the demand for disk work space but could lead to more frequent maintenance of nonpartitioning indexes.

During the physical design phase of the project, estimates were made for DASD space. When computing the DASD space requirements for indexes, Lands' End relied heavily on the guidelines in the IBM DATABASE 2 Version 2 Release 3 Administration Guide. In the guidelines, a multiplier is recommended based on the design types of tight, medium, and loose. Because DSS tends to be a loose design (i.e., requirements are not as strictly defined), we chose to use the loose design multiplier. This multiplier specified that we should account for an extra 40 percent of DASD space for indexes. After all the space calculations had been completed by the database administrators using the formulas in the Administration Guide, the index overhead in reality was approximately 75 percent. The difference between recommendation and reality required the rushed acquisition of additional DASD.

Some of the factors that contributed to the index ratio being approximately 75 percent were:

1. DSS databases are by nature not meant to be application-specific. Thus, if indexes cannot be applied to satisfy the specific requirements of an application, they must at least be flexible enough to satisfy the majority of ad hoc requests.
2. Atomic level data contain a greater number of perspectives by which data may be predicated or summarized. Indexes must be applied to these perspectives to achieve efficient access.
3. The large volume of data requires that even access paths having only moderate frequency of use must be indexed to provide reasonable response times to client queries.
4. The use of system-managed referential integrity in most cases requires that foreign keys be indexed. With these indexes it is possible to enforce referential constraints while executing the load utility and not incur a tablespace scan (read the entire table without utilizing an index) of a dependent table.

The definition of "loose" in the DB2 manuals needs to be adjusted upward. As more businesses put larger and larger demands on DB2, the implications of managing more rows and larger indexes will need to be more fully addressed.

Some of the major third-party software vendors have significant restrictions on their DB2 utilities that can cause excessive overhead. Referential integrity is one area where problems can occur, forcing the execution of DB2 utilities that check referential integrity. The overhead of this checking could greatly diminish the value from such utilities.

To provide the clients and developers with metadata at the workstation level, the data administrator created a data directory. The data directory used Forest and Trees to provide a graphical user interface to clients and developers to extract information from Silverrun-RDM and the DB2 catalog. Information about tables and columns, including descriptions and domains, is available complete with help information. Lands' End will be exploring the acquisition of a true data dictionary in the future.

The future of DSS at Lands' End

Lands' End will continue to leverage the benefits of its Information Warehouse framework by looking for new opportunities to extract value from its corporate database.

Lands' End will also strive to make available to its clients the latest in industrial strength, workstation-based query and analysis tools. As more and more 32-bit client/server tools become available, Lands' End will evaluate them for increased functionality and productivity.

To improve client satisfaction and increase throughput, Lands' End has positioned itself to take advantage of I/O parallelism by spreading its data out as much as possible, acquiring high end mainframe hardware and maintaining DB2 software at its most current version and release.

Hardware compression of data is another area Lands' End will explore. Benchmarks for various types of processing that include queries, loading, update, copying, and deleting will be performed during evaluation of this technology.

Finally, to provide the client with the latest DSS information, Lands' End has the goal of migrating the atomic level order processing and promotional data to the DSS system on a daily rather than weekly schedule.

Concluding remarks

Decision support processing at Lands' End has evolved from a single threaded, primarily tape-based disparate group of data to a refined collection of common data items located on a relational platform where multiple connectivity choices exist. Lands' End is reaping the benefits of its corporate database. Client queries that would have taken hours, or sometimes days, of elapsed time to satisfy are now taking minutes. The relationship between the clients and IS has been strengthened by including the clients in the process of developing the corporate database.

In the process of creating a corporate database, Lands' End underwent an intense data modeling effort, developed creative data partitioning schemes, and for the first time, explored the dynamic world of client/server. The development effort has expanded the capabilities and horizons of those involved. Lands' End will need these abilities to continue to meet the challenges from both competitors and changing technology in order to position itself for future success.

Acknowledgments

The authors wish to acknowledge several people whose support made the corporate database project a success. In 1992 Lands' End named Dan Rourke the Vice President of Information Services. The development of the corporate database was a direct result of his vision of an Information Warehouse framework that would act as a DSS server and application enabler. Carolyn Dow of IBM provided invaluable guidance and technical advice in developing this article. Mark Koshalek, database administrator, Kevin Thompson, business support group manager, Tami Van Dreese, programmer/analyst, and Steve Andrews, data administrator, all contributed to the content of this document.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Lands' End, Inc., Computer Associates, NCR Corp., Texas Instruments Incorporated, Computer Systems Advisers, or Trinzic Corporation.

Cited references and notes

1. *Information Warehouse Architecture I*, SC26-3244, IBM Corporation (April 1993); available through IBM branch offices.
2. A *join* is a "relational operation that allows retrieval of data from two or more tables based on matching column values." *IBM DATABASE 2 Version 2 Release 3 SQL Reference*, SC26-4380, IBM Corporation (March 1992), p. 430; available through IBM branch offices.
3. "The atomic level (DSS) data is detailed data that is stored as of some moment in time, . . . is as granular as data in the (DSS) environment can become. It is nonredundant with other atomic data, and the structures of the atomic level are oriented toward the major subjects of the enterprise." W. H. Inmon, *Data Architecture: The Information Paradigm*, QED Information Sciences, Inc., Wellesley, MA (1989), p. 14.
4. "A domain is a pool of values from which the actual values appearing in a given column are drawn." *Ibid.*, p. 65.
5. "A relation R is in third normal form (3NF) if and only if it is in 2NF and every nonkey attribute is nontransitively dependent on the primary key." C. J. Date, *An Introduction to Database Systems*, third edition, Addison-Wesley Publishing Co., Reading, MA (1981), p. 248.
6. B. Ives, S. L. Jarvenpaa, and R. O. Mason, "Global Business Drivers: Aligning Information Technology to Global Business Strategy," *IBM Systems Journal* **32**, No. 1, 143–161 (1993).
7. A Cartesian product is a join of tables in which "each row of one table is combined with every row of another, one at a time. The resulting table has the same number of rows as the number of rows in the first table, times the number of rows in the other tables." B. L. Larson, *The Database Experts' Guide to DATABASE 2*, Multiscience Press, Inc., New York (1988), p. 8.

Accepted for publication February 11, 1994.

Gerald G. Bustamente *Lands' End Information Services, 1 Lands' End Lane, Dodgeville, Wisconsin 53595.* Mr. Bustamente is one of three database administrators who are responsible for maintaining the corporate database at Lands' End. Prior to joining Lands' End, Mr. Bustamente was a database administrator for the State of Wisconsin after having become a senior programming analyst. Mr. Bustamente has a B.S. in business management from Cardinal Stritch College in Milwaukee, Wisconsin.

Kai Sorenson *Lands' End Information Services, 1 Lands' End Lane, Dodgeville, Wisconsin 53595.* Mr. Sorenson is an applications business systems specialist in the Finance/Corporate Database business support group. Mr. Sorenson was project leader for the corporate database project described in this paper. Prior to joining Lands' End, Mr. Sorenson was an application developer with First Wisconsin National Bank. Mr. Sorenson has a B.S. in business management from the University of Wisconsin—Parkside.

Reprint Order No. G321-5540.