CHAPTER 6

WAREHOUSING STRATEGY

Defines the warehouse strategy as part of the information technology strategy of the enterprise. The traditional Information Strategy Plan (ISP) addresses operational computing needs thoroughly but may not give sufficient attention to decisional information requirements. A data warehouse strategy remedies this by focusing on the decisional needs of the enterprise.

We start this chapter by presenting the components of a Data Warehousing strategy. We follow with a discussion of the tasks required to define a strategy for an enterprise.

6.1 STRATEGY COMPONENTS

At a minimum, the data warehouse strategy should include the following elements:

Preliminary Data Warehouse Rollout Plan

Not all of the user requirements can be met in one data warehouse project—such a project would necessarily be large, and dangerously unmanageable. It is more realistic to prioritize the different user requirements and assign them to different warehouse rollouts. Doing so allows the enterprise to divide the warehouse development into phased, successive rollouts, where each rollout focuses on meeting an agreed set of requirements.

The iterative nature of such an approach allows the warehousing temp to extend the functionality of the warehouse in a manageable manner. The phased approach lowers the overall risk of the data warehouse project, while delivering increasing functionality to the users.

Preliminary Data Warehouse Architecture

Define the overall data warehouse architecture for the pilot and subsequent warehouse rollouts to ensure the scalability of the warehouse. Whenever possible, define the initial technical architecture of each rollout.

By consciously thinking through the data warehouse architecture, warehouse planners can determine the various technology components (e.g., MDDB. RDBMS, tools) those are required for each rollout.

Short-listed Data Warehouse Environment and Tools

There are a number of tools and warehousing environments from which to be chosen. Create a short - list for the tools and environments that appear to meet a warehousing needs. A standard set of tools will lessen tool integration problems and will minimize the learning required for both the warehousing team and the warehouse users.

Below are the tasks required to create the enterprise's warehousing strategy. Note that the tasks described below can typically be completed in three to five weeks, depending on the availability of resource persons and the size of the enterprise.

6.2 DETERMINE ORGANIZATIONAL CONTEXT

An understanding of the organization helps to establish the context of the project and may highlight aspects of the corporate culture that my ease or complicate the warehousing project. Answers to organizational background questions are typically obtained from the Project Sponsor, the CIO, or the Project Manager assigned to the warehousing effort.

Typical organizational background questions include:

- **Who is the Project Sponsor for this project?**

  The Project Sponsor sets the scope of the warehousing project. He or she also plays a crucial role in establishing the working relationship among warehousing team members, especially if third parties are involved. Easy access to warehousing data may also be limited to the organizational scope that is within the control or authority of the Project Sponsor.

- **What are the IS or IT groups in the organization, which are involved in the data warehousing effort?**

  Since data warehousing is very much a technology-based endeavor, the IS or IT groups within the organization will always be involved in any warehousing effort. It is often insightful to understand the bulk of the work currently performed within the IS or IT departments. If the IS or IT groups are often fighting fires or are very busy deploying operational systems, data warehousing is unlikely to be high on the list of IT priorities.

- **What are the roles and responsibilities of the individuals who have been assigned to this effort?**

  It is helpful to define the roles and responsibilities of the various individuals involved in the data warehousing project. This practice sets common, realistic expectations and improves understanding and communication within the team. In cases where the team is composed of external parties (especially where several vendors are involved), a clear definition of roles becomes critical.

6.3 CONDUCT PRELIMINARY SURVEY OF REQUIREMENTS

Obtain an inventory of the requirements of business users through individual and group interview with the enduser community. Whenever possible, obtain layouts of the current management reports (and their planned enhancements).

The requirements inventory represents the breadth of information that the warehouse is expected to eventually provide. While it is important to get a clear picture of the extent of requirements, it is not necessary to detail all the requirements in depth at this point. The objective is to understand the user needs enough to prioritize the requirements. This is a critical input for identifying the scope of each data warehouse rollout.

Interview Categories and Sample Questions

The following questions, arranged by category, should be useful as a starting point for the interview with intended end users of the warehouse:

- **Functions.** What is the mission of your group or unit? How do you go about fulfilling this mission? How do you know if you've been successful with your mission? What are the key performance indicators and critical success factors?
- **Customers.** How do you group or classify your customers? Do these groupings change over time? Does your grouping affect how you treat your customers? What kind of information do you track for each type of client? What demographic information do you use, if any? Do you need to track customer data for each customer?
- **Profit.** At what level do you measure profitability in your group? Per agent? Per customer? Per product? Per region? At what level of detail are costs and revenues tracked in your organization? How do you track costs and revenues now? What kind of profitability reports do you use or produce now?
- **Systems.** What systems do you use as part of your job? What systems are you aware of in other groups that contain information you require? What kind of manual data transformations do you have to perform when data are unavailable?
- **Time.** How many months or years of data you need to track? Do you analyze performance across years? At what level of detail do you need to see figures? Daily? Weekly? Monthly? Quarterly? Yearly? Do you need to see figures? How soon do you need to see data (e.g., do you need yesterday's data today?) How soon after week-end, month-end, quarter-end, and year-end do you need to see the previous period figures?
- **Queries and reports.** What reports do you use now? What information do you actually use in each of the reports you now receive? Can we obtain samples, of these reports? How often are these reports produced? Do you get them soon enough, frequently enough? Who makes these reports for you? What reports do you produce for other people?
- **Product.** What products do you sell, and how do you classify them? Do you have a product hierarchy? Do you analyze data for all products at the same time, or do you analyze one product type at a time? How do you handle changes in product hierarchy and product description?
- **Geography.** Does your company operate in more than one location? Do you divide your market into geographical areas? Do you track sales per geographic region?

Interviewing Tips

Many of the interviewing tips enumerated below may seem like common sense. Nevertheless, interviewers are encouraged to keep the following points in mind:

- **Avoid making commitments about warehouse scope.** It will not be surprising to find that some of the queries and reports requested by interviewes cannot be supported by the data that currently reside in the operational databases. Interviewers should keep this in mind and communicate this potential limitation to their interviewees. The interviewers cannot afford to make commitments regarding the warehouse scope at this time.
- **Keep the interview objective in mind.** The objective of these interviews is to create an inventory of requirements. There is no need to get a detailed understanding of the requirements at this point.
- **Don't overwhelm the interviewees.** The interviewing team should be small; two people are the ideal number—one to ask questions, another to take notes. Interviewees may be intimidated if a large group of interviewers shows up.
- **Record the session if the interviewee lets you.** Most interviewees will not mind if interviewers bring along a tape recorder to record the session. Transcripts of the session may later prove helpful.
- **Change the interviewing style depending on the interviewee.** Middle-Managers more likely to deal with actual reports and detailed information requirements. Senior executives are more likely to dwell on strategic information needs. Change the interviewing style as needed by adapting the type of questions to the type of interviewee.
- **Listen carefully.** Listen to what the interviewee has to say. The sample interview questions are merely a starting point—follow-up questions have the potential of yielding interesting and critical of yielding interesting and critical information. Take note of the terms that the interviewee uses. Popular business terms such as "profit" may have different meanings or connotations within the enterprise.
- **Obtain copies of reports, whenever possible.** The reports will give the warehouse team valuable information about source systems (which system produced the report), as well as business rules and terms. If a person manually makes the reports, the team may benefit from talking to this person.

6.4 CONDUCT PRELIMINARY SOURCE SYSTEM AUDIT

Obtain an inventory of potential warehouse data sources through individual and group interview with key personnel in the IT organization. While the CIO no doubt has a broad, high-level view of the systems in the enterprise, the best resource persons for the source system audit are the DBAs and system administrators who maintain the operational systems.

Typical background interview questions, arranged by categories, for the IT department include:

- **Current architecture.** What is the current technology architecture of the organization? What kind of systems, hardware, DBMS, network, end-user tools, development tools, and data access tools are currently in use?
- **Source system relationships.** Are the source systems related in any way? Does one system provide information to another? Are the systems integrated in any manner? In cases where multiple systems have customer and product records, which one serves as the "master" copy?

- **Network facilities.** Is it possible to use a single terminal or PC to access the different operational systems, from all locations?
- **Data quality.** How much cleaning, scrubbing, de-duplication, and integration do you suppose will be required? What areas (tables or fields) in the source systems are currently known to have poor data quality?
- **Documentation.** How much documentation is available for the source systems? How accurate and up-to-date are these manuals and reference materials? Try to obtain the following information whenever possible: copies of manuals and reference documents, database size, batch window, planned enhancements, typical backup size, backup scope and backup medium, data scope of the system (e.g., important tables and fields), system codes and their meanings, and keys generation schemes.
- **Possible extraction mechanisms.** What extraction mechanisms are possible with this system? What extraction mechanisms have you used before with this system? What extraction mechanisms will not work?

6.5 IDENTIFY EXTERNAL DATA SOURCES (IF APPLICABLE)

The enterprise may also make use of external data sources to augment the data from internal source systems. Example of external data that can be used are:

- Data from credit agencies.
- Zip code or mail code data.
- Statistical or census data.
- Data from industry organizations.
- Data from publications and news agencies.

Although the use of external data presents opportunities for enriching the data warehouse, it may also present difficulties because of difference in granularity. For example, the external data may not be readily available at the level of detail required by the data warehouse and may require some transformation or summarization.

6.6 DEFINE WAREHOUSE ROLLOUTS (PHASED IMPLEMENTATION)

Divide the data warehouse development into phased, successive rollout. Note that the scope of each rollout will have to be finalized as part of the planning for that rollout. The availability and quality of source data will play a critical role in finalizing that scope.

As stated earlier, applying a phased approach for delivering the warehouse should lower the overall risk of the data warehouse project while delivering increasing functionality and data to more users. It also helps manage user expectations through the clear definition of scope for each rollout.

Figure 6.1 is a sample table listing all requirements identified during the initial round of interviews with end users. Each requirement is assigned a priority level. An initial complexity assessment is made, based on the estimated number of source systems, early data quality assessments, and the computing environments of the source systems. The intended user group is also identified.

No.	Requirement	Priority	Complexity	Users	Rollout No.
1	Customer Profitability	High	High	Customer Service	1
2	Product Market Share	High	Medium	Product Manager	1
3	Weekly Sales Trends	Medium	Low	VP, Sales	2
	—	—	—	—	—

Figure 6.1. Sample Rollout Definition

More factors can be listed to help determine the appropriate rollout number for each requirement. The rollout definition is finalized only when it is approved by the Project Sponsor.

6.7 DEFINE PRELIMINARY DATA WAREHOUSE ARCHITECTURE

Define the preliminary architecture of each rollout based on the approved rollout scope. Explore the possibility of using a mix of relational and multidimensional databases and tools, as illustrated in Figure 6.2.

At a minimum, the preliminary architecture should indicate the following:

- **Data warehouses and data mart.** Define the intended deployment of data warehouses and data marts for each rollout. Indicate how the different databases are related (i.e., how the databases feed one another). The warehouse architecture must ensure that the different data marts are not deployed in isolation.

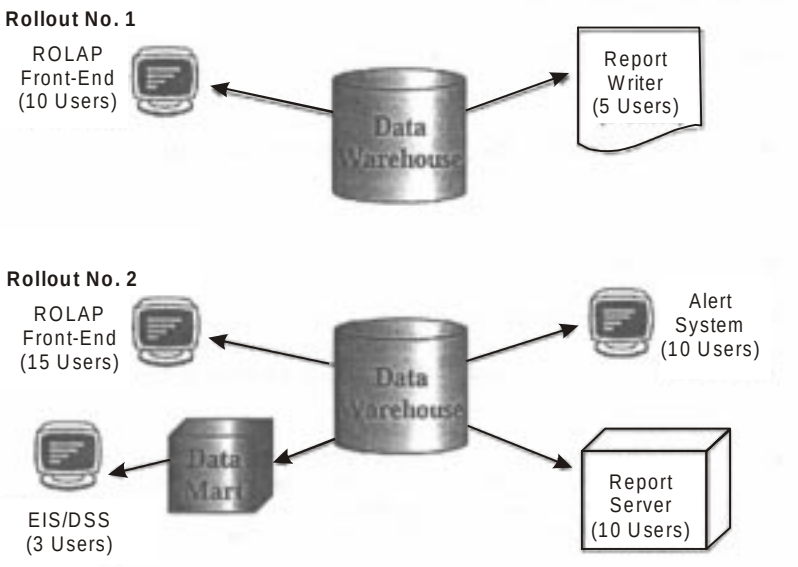


Figure 6.2. Sample Preliminary Architecture per Rollout

- **Number of users.** Specify the intended number of users for each data access and retrieval tool (or front-end) for each rollout.
- **Location.** Specify the location of the data warehouse, the data marts, and the intended users for each rollout. This has implications on the technical architecture requirements of the warehousing project.

6.8 EVALUATE DEVELOPMENT AND PRODUCTION ENVIRONMENT AND TOOLS

Enterprises can choose from several environments and tools for the data warehouse initiative, select the combination of tools that best meets the needs of the enterprise. At present, no single vendor provides an integrated suite of warehousing tools. There are, however, clear leaders for each tool category.

Eliminate all unsuitable tools, and produce a short-list from which each rollout or project will choose its tool set (see Figure 6.3). Alternatively, select and standardize on a set of tools for all warehouse rollouts.

No.	Tool Category	Short-listed Tools	Evaluation Criteria	Weights (%)	Preliminary Scores
1	Data Access	Tool A	Criterion 1	30%	
	and Retrieval		Criterion 2	30%	78%
			Criterion 3	40%	
		Tool B	Criterion 1	30%	
			Criterion 2	30%	82%
			Criterion 3	40%	
		Tool C	Criterion 1	30%	
			Criterion 2	30%	84%
			Criterion 3	40%	
2	RDBMS				

Figure 6.3. Sample Tool Short-List

In Summary

A data warehouse strategy at a minimum contains:

- Preliminary data warehouse rollout plan, which indicates how the development of the warehouse is to be phased:
- Preliminary data warehouse architecture, which indicates the likely physical implementation of the warehouse rollout; and
- Short-listed options for the warehouse environment and tools.

The approach for arriving at these strategy components may vary from one enterprise to another, the approach presented in this chapter is one that has consistently proven to be effective.

Expect the data warehousing strategy to be updated annually. Each warehouse rollout provides new learning and as new tools and technologies become available.

CHAPTER 7

WAREHOUSE MANAGEMENT AND SUPPORT PROCESSES

Warehouse Management and Support Processes are designed to address aspects of planning and managing a data warehouse project that are critical to the successful implementation and subsequent extension of the data warehouse. Unfortunately, these aspects are all too often overlooked in initial warehousing deployments.

These processes are defined to assist the project manager and warehouse driver during warehouse development projects.

7.1 DEFINE ISSUE TRACKING AND RESOLUTION PROCESS

During the course of a project, it is inevitable that a number of business and technical issues will surface. The project will quickly be delayed by unresolved issues if an issue tracking and resolution process is not in place. Of particular importance are business issues that involve more than one group of users. These issues typically include disputes over the definition of business terms and the financial formulas that govern the transformation of data.

An individual on the project should be designated to track and follow up the resolution of each issue as it arises. Extremely urgent issues (i.e., issues that may cause project delay if left unresolved) or issues with strong political overtones can be brought to the attention of the Project Sponsor, who must use his or her clout to expedite the resolution process.

Figure 7.1 shows a sample issue logs that tracks all the issues that arise during the course of the project.

The following issue tracking guidelines will prove helpful:

- **Issue description.** State the issue briefly in two to three sentences. Provide a more detailed description of the issue as a separate paragraph. If there are possible resolutions to the issue, include these in the issue description. Identify the consequences of leaving this issue open, particularly and impact on the project schedule.

No.	Issue Description	Urgency	Raised By	Assigned To	Date Opened	Date Closed	Resolved By	Resolution Description
1	Conflict over definition of "Customer"	High	MWH	MCD	Feb 03	Feb 05	CEO	Use CorPlan's definition
2	Currency exchange rates are not tracked in GL	High	MCD	RGT	Feb 04			

Figure 7.1. Sample Issue Log

- **Urgency.** Indicate the priority level of the issue: high, medium, or low. Low-priority issues that are left unresolved may later become high priority. The team may have agreed on a resolution rate depending on the urgency of the issue. For example, the team can agree to resolve high-priority issues within three days, medium-priority issues within a week, and low-priority issues within two weeks.
- **Raised by.** Identify the person who raised the issue. If the team is large or does not meet on a regular basis, provide information on how to contact the person (e.g., telephone number, e-mail address). The people who are resolving the issue may require additional information or details that only the issue originator can provide.
- **Assigned to.** Identify the person on the team who is responsible for resolving the issue. Note that this person does not necessarily have answer. However, he or she is responsible for tracking down the person who can actually resolve the issue. He or she also follows up on issues that have been left unresolved.
- **Date opened.** This is the date when the issue was first logged.
- **Date closed.** This is the date when the issue was finally resolved.
- **Resolved by.** The person who resolves the issue must have the required authority within the organization. User representatives typically resolve business issues. The CIO or designated representatives typically resolve technical issues, and the project sponsor typically resolves issues related project scope.
- **Resolution description.** State briefly the resolution of this issue in two or three sentences. Provide a more detailed description of the resolution in a separate paragraph. If subsequent actions are required to implement the resolution, these should be stated clearly and resources should be assigned to implement them. Identify target dates for implementation.

Issue logs formalize the issue resolution process. They also serve as a formal record of key decisions made throughout the project.

In some cases, the team may opt to augment the log with yet another form–one form for each issue. This typically happens when the issue descriptions and resolution descriptions are quite long. In this case, only the brief issue statement and brief resolution descriptions are recorded in the issue log.

7.2 PERFORM CAPACITY PLANNING

Warehouse capacity requirements come in the following forms: space required, machine processing power, network bandwidth, and number of concurrent users. These requirements increase with each rollout of the data warehouse.

During the stage of defining the warehouse strategy, the team will not have the exact information for these requirements. However, as the warehouse rollout scopes are finalized, the capacity requirements will likewise become more defined.

Review the following capacity planning requirements basing your review on the scope of each rollout.

There are several aspects to the data warehouse environment that make capacity planning for the data warehouse a unique exercise. The first factor is that the workload for the data warehouse environment is very variable. In many ways trying to anticipate the DSS workload requires imagination. Unlike the operational workload that has an air of regularity to it, the data warehouse DSS workload is much less predictable. This factor, in and of itself, makes capacity planning for the data warehouse a chancy exercise.

A second factor making capacity planning for the data warehouse a risky business is that the data warehouse normally entails much more data than was ever encountered in the operational environment. The amount of data found in the data warehouse is directly related to the design of the data warehouse environment. The designer determines the granularity of data that in turn determines how much data there will be in the warehouse. The finer the degree of granularity, the more data there is. The coarser the degree of granularity, the less data there is. And the volume of data not only affects the actual disk storage required, but the volume of data affects the machine resources required to manipulate the data. In very few environments is the capacity of a system so closely linked to the design of the system.

A third factor making capacity planning for the data warehouse environment a nontraditional exercise is that the data warehouse environment and the operational environments do not mix under the stress of a workload of any size at all. This imbalance of environments must be understood by all parties involved-the capacity planner, the systems programmer, management, and the designer of the data warehouse environment.

Consider the patterns of hardware utilization as shown by Figure 7.2.

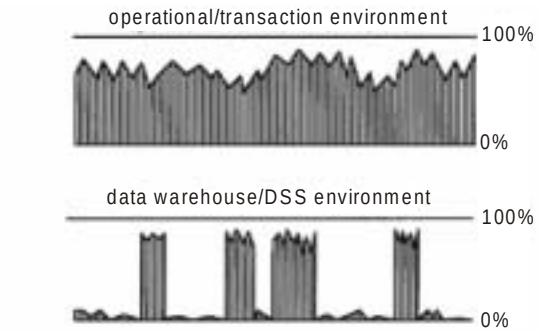


Figure 7.2. The Fundamentally Different Patterns of Hardware Utilization between the Data Warehouse Environment and the Operational Environment

In Figure 7.2 it is seen that the operational environment uses hardware in a static fashion. There are peaks and valleys in the operational environment, but at the end of the day hardware utilization is predictable and fairly constant. Contrast the pattern of hardware utilization found in the operational environment with the hardware utilization found in the data warehouse/DSS environment.

In the data warehouse, hardware is used in a binary fashion. Either the hardware is being used constantly or the hardware is not being used at all. Furthermore, the pattern is such that it is unpredictable. One day much processing occurs at 8:30 a.m. The next day the bulk of processing occurs at 11:15 a.m. and so forth.

There are then, very different and incompatible patterns of hardware utilization associated with the operational and the data warehouse environment. These patterns apply to all types of hardware CPU, channels, memory, disk storage, etc.

Trying to mix the different patterns of hardware leads to some basic difficulties. Figure 7.3 shows what happens when the two types of patterns of utilization are mixed in the same machine at the same time.

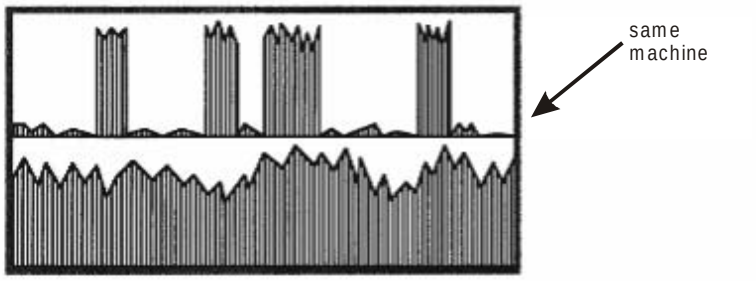


Figure 7.3. Trying to Mix the Two Fundamentally Different Patterns of the Execution in the Same Machine at the Same Time Leads to Some Very Basic Conflicts

The patterns are simply incompatible. Either you get good response time and a low rate of machine utilization (at which point the financial manager is unhappy), or you get high machine utilization and poor response time (at which point the user is unhappy.) The need to split the two environments is important to the data warehouse capacity planner because the capacity planner needs to be aware of circumstances in which the patterns of access are mixed. In other words, when doing capacity planning, there is a need to separate the two environments. Trying to do capacity planning for a machine or complex of machines where there is a mixing of the two environments is a nonsensical task. Despite these difficulties with capacity planning, planning for machine resources in the data warehouse environment is a worthwhile endeavor.

Time Horizons

As a rule there are two time horizons, the capacity planner should aim for–the one-year time horizon and the five-year time horizon. Figure 7.4 shows these time horizons.

The one-year time horizon is important in that it is on the immediate requirements list for the designer. In other words, at the rate that the data warehouse becomes designed and populated, the decisions made about resources for the one year time horizon will have to be lived with. Hardware, and possibly software acquisitions will have to be made. A certain

amount of "burn-in" will have to be tolerated. A learning curve for all parties will have to be survived, all on the one-year horizon. The five-year horizon is of importance as well. It is where the massive volume of data will show up. And it is where the maturity of the data warehouse will occur.

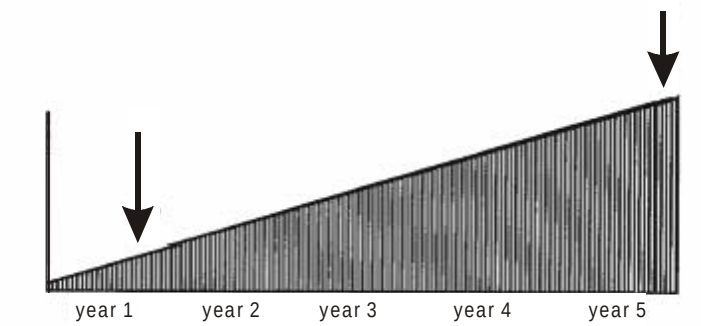


Figure 7.4. Projecting Hardware Requirements out to Year 1 and Year 5

An interesting question is – " why not look at the ten year horizon as well?" Certainly projections can be made to the ten-year horizon. However, those projections are not usually made because:

- It is very difficult to predict what the world will look like ten years from now,
- It is assumed that the organization will have much more experience handling data warehouses five years in the future, so that design and data management will not pose the same problems they do in the early days of the warehouse, and
- It is assumed that there will be technological advances that will change the considerations of building and managing a data warehouse environment.

DBMS Considerations

One major factor affecting the data warehouse capacity planning is what portion of the data warehouse will be managed on disk storage and what portion will be managed on alternative storage. This very important distinction must be made, at least in broad terms, prior to the commencement of the data warehouse capacity planning effort.

Once the distinction is made, the next consideration is that of the technology underlying the data warehouse. The most interesting underlying technology is that of the Data Base Management System-DBMs. The components of the DBMs that are of interest to the capacity planner are shown in Figure 7.5.

The capacity planner is interested in the access to the data warehouse, the DBMs capabilities and efficiencies, the indexing of the data warehouse, and the efficiency and operations of storage. Each of these aspects plays a large role in the throughput and operations of the data warehouse.

Some of the relevant issues in regards to the data warehouse data base management system are:

- How much data can the DBMs handle? (**Note:** There is always a discrepancy between the theoretical limits of the volume of data handled by a data base management system and the practical limits.)

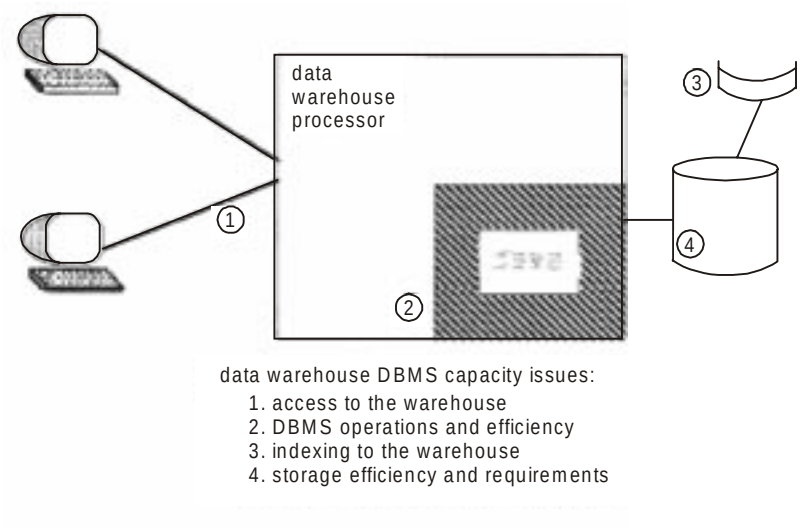


Figure 7.5

- How can the data be stored? Compressed? Indexed? Encoded? How are null values handled?
- Can locking be suppressed?
- Can requests be monitored and suppressed based on resource utilization?
- Can data be physically denormalized?
- What support is there for metadata as needed in the data warehouse? and so forth. Of course the operating system and any teleprocessing monitoring must be factored in as well.

Disk Storage and Processing Resources

The two most important parameters of capacity management are the measurement of disk storage and processing resources. Figure 7.6 shows those resources.

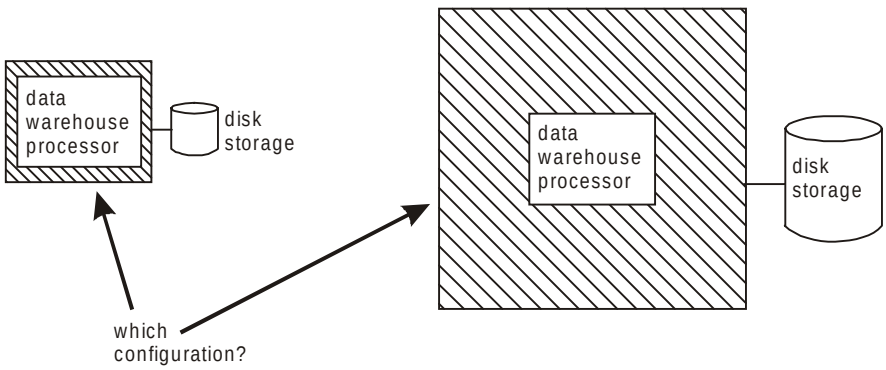


Figure 7.6. The Two Main Aspects of Capacity Planning for the Data Warehouse Environment are the Disk Storage and Processor Size

The question facing the capacity planner is – how much of these resources will be required on the one-year and the five-year horizon. As has been stated before, there is an indirect (yet very real) relationship between the volume of data and the processor required.

Calculating Disk Storage

The calculations for space are almost always done exclusively for the current detailed data in the data warehouse. (If you are not familiar with the different levels of data in the warehouse, please refer to the Tech Topic on the description of the data warehouse.) The reason why the other levels of data are not included in this analysis is that:

- They consume much less storage than the current detailed level of data, and
- They are much harder to identify.

Therefore, the considerations of capacity planning for disk storage center around the current detailed level. The calculations for disk storage are very straightforward. Figure 7.7 shows the elements of calculation.

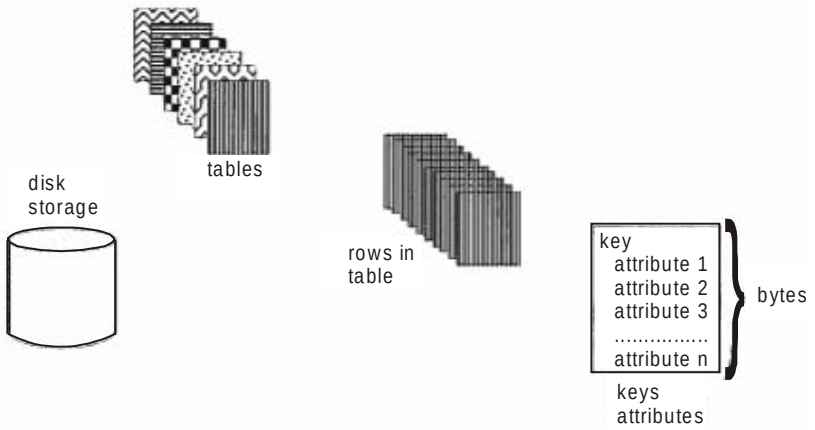


Figure 7.7. Estimating Disk Storage Requirements for the Data Warehouse

To calculate disk storage, first the tables that will be in the current detailed level of the data warehouse are identified. Admittedly, when looking at the data warehouse from the standpoint of planning, where little or no detailed design has been done, it is difficult to divise what the tables will be. In truth, only the very largest tables need be identified.

Usually there are a finite number of those tables in even the most complex of environments.

Once the tables are identified, the next calculation is how many rows will there be in each table. Of course, the answer to this question depends directly on the granularity of data found in the data warehouse. The lower the level of detail, the more the number of rows.

In some cases the number of rows can be calculated quite accurately. Where there is a historical record to rely upon, this number is calculated. For example, where the data warehouse will contain the number of phone calls made by a phone company's customers and where the business is not changing dramatically, this calculation can be made. But in other cases it is not so easy to estimate the number of occurrences of data.

One approach is to look across the industry and see what other companies have experienced. This approach is quite effective if you can find out information that other companies are willing to share and if the company has a similar profile. Unfortunately, often times a comparative company is hard to find.

A third approach is to estimate the number of occurrences based on business plans, economic forecasts, and using the advice of specialized industry consultants. This is the least accurate method and the most expensive, but sometimes it is the only choice.

The number of occurrences of data is a function of more than just the business. The other factors are:

- The level of detail the data will be kept at, and
- The length of time the data will be kept.

After the number of rows are divised, the next step is to calculate the size of each row. This is done by estimating the contents of each row–the keys and the attributes. Once the contents of the row are taken into consideration, the indexes that are needed are factored in.

As a matter of practice, very little if any free space is left in the data warehouse because data is not updated in the warehouse. In most circumstances, any free space in a data warehouse is wasted. The total disk requirements then are calculated by adding all the requirements mentioned.

Processor Requirements

In order to make sense of the estimation of the processor requirements for the data warehouse, the work passing through the data warehouse processor must be divided into one of three categories—background processing, predictable DSS processing, and unpredictable DSS processing. Figure 7.8 shows these three categories.

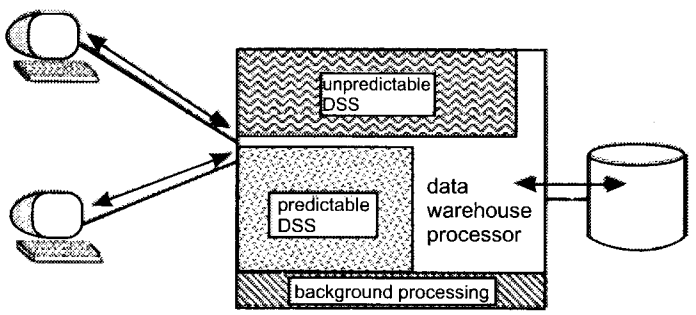


Figure 7.8. The Three Types of Processing that Occur in the Data Warehouse

Background processing is that processing that is done on a predictable, (usually) batch basis. Typical of background processing is extract processing, data warehouse loads, monitors, sorts/merges, restructuring, index creations, etc. Background processing is that utilitarian processing necessary to the data warehouse but not directly associated with a query or an analysis of data warehouse data.

Background processing can be run at off peak times and can be spread evenly throughout the day. There is seldom much of a time constraint for background processing.

Predictable DSS processing is that processing that is regularly done, usually on a query or transaction basis. Predictable DSS processing may be modeled after DSS processing done today but not in the data warehouse environment. Or predictable DSS processing may be projected, as are other parts of the data warehouse workload.

The parameters of interest for the data warehouse designer (for both the background processing and the predictable DSS processing) are:

- The number of times the process will be run,
- The number of I/Os the process will use,
- Whether there is an arrival peak to the processing,
- The expected response time.

These metrics can be arrived at by examining the pattern of calls made to the DBMS and the interaction with data managed under the DBMS.

The third category of process of interest to the data warehouse capacity planner is that of the unpredictable DSS analysis. The unpredictable process by its very nature is much less manageable than either background processing or predictable DSS processing.

However, certain characteristics about the unpredictable process can be projected (even for the worst behaving process.) For the unpredictable processes, the:

- Expected response time (in minutes, hours, or days) can be outlined,
- Total amount of I/O can be predicted, and
- Whether the system can be quiesced during the running of the request can be projected.

Once the workload of the data warehouse has been broken into these categories, the estimate of processor resources is prepared to continue. The next decision to be made is whether the eight-hour daily window will be the critical processing point or whether overnight processing will be the critical point. Usually the eight-hour day from 8:00 a.m. to 5:00 p.m. as the data warehouse is being used is the critical point. Assuming that the eight-hour window is the critical point in the usage of the processor, a profile of the processing workload is created.

The Workload Matrix

The workload matrix is a matrix that is created as the intersection of the tables in the data warehouse and the processes (the background and the predictable DSS processes) that will run in the data warehouse. Figure 7.9 shows a matrix formed by tables and processes.

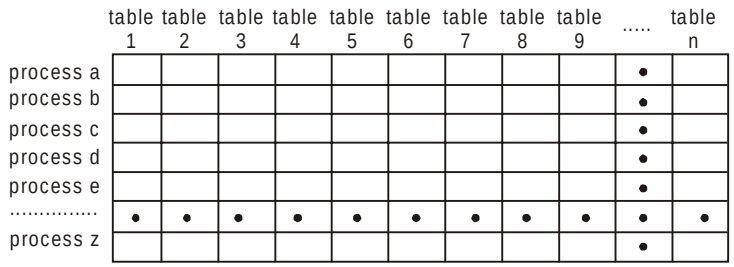


Figure 7.9. A Matrix Approach Helps to Organize the Activities

The workload matrix is then filled in. The first pass at filling in the matrix involves putting the number of calls and the resulting I/O from the calls the process would do if the process were executed exactly once during the eight hour window. Figure 7-10 shows a simple form of a matrix that has been filled in for the first step.

	table 1	table 2	table 3	table 4	table 5	table 6	table 7	table 8	table 9	.....	table n
process a	2/5			2/10		5/15				•	3/9
process b	1/3		3/9						1/2	•	
process c		5/25					4/12		1/5	•	2/5
process d	1/6		4/8		1/2	3/9		3/9		•	1/2
process e	1/5		3/4		2/2	6/12		6/12		•	
.....	•	•	•	•	•	•	•	•	•	•	•
process z	1/5	5/25	4/8		1/2				5/15	•	5/15

number of calls I/O's per access

Figure 7.10. The Simplest Form of a Matrix is to Profile a Individual Transactions in Terms of Calls and I/O's.

For example, in Figure 7.10 the first cell in the matrix–the cell for process **a** and table 1- contains a "2/5". The 2/5 indicates that upon execution process **a** has two calls to the table and uses a total of 5 I/Os for the calls. The next cell–the cell for process **a** and table 2 - indicates that process **a** does not access table 2. The matrix is filled in for the processing profile of the workload as if each transaction were executed once and only once. Determining the number of I/Os per call can be a difficult exercise. Whether an I/O will be done depends on many factors:

- The number of rows in a block,
- Whether a block is in memory at the moment it is requested,
- The amount of buffers there are,
- The traffic through the buffers,
- The DBMS managing the buffers,
- The indexing for the data,
- The other part of the workload,
- The system parameters governing the workload, etc.

There are, in short, MANY factors affecting how many physical I/O will be used. The I/Os an be calculated manually or automatically (by software specializing in this task). After the single execution profile of the workload is identified, the next step is to create the actual workload profile. The workload profile is easy to create. Each row in the matrix is multiplied by the number of times it will execute in a day. The calculation here is a simple one. Figure 7.11 shows an example of the total I/Os used in an eight-hour day being calculated.

	table 1	table 2	table 3	table 4	table 5	table 6	table 7	table 8	table 9	.....	table n
process a	2500			5000		7500				•	4500
process b	1000		3000						666	•	
process c		2500					1200		500	•	500
process d	750		1000		250	1250		1250		•	250
process e	700		566		283			1750		•	
.....	•	•	•	•	•	•	•	•	•	•	•
process z	750	3750	1000		300				2250	•	2250
total I/O	15000	85000	12250	10250	5750	27750	19500	6750	8926	•	87560

total I/Os

Figure 7.11. After the Individual Transactions are Profiled, the profile is Multiplied by the Number of Expected Executions per Day

At the bottom of the matrix the totals are calculated, to reach a total eight-hour I/O requirement. After the eight hour I/O requirement is calculated, the next step is to determine what the hour-by-hour requirements are. If there is no hourly requirement, then it is assumed that the queries will be arriving in the data warehouse in a flat arrival rate. However, there usually are differences in the arrival rates. Figure 7.12, shows the arrival rates adjusted for the different hours in the day.

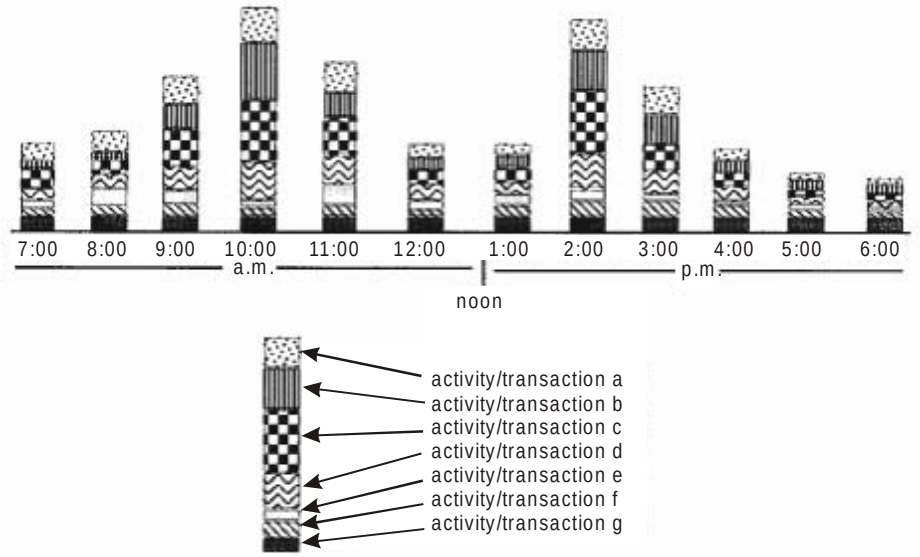


Figure 7.12. The Different Activities are Tallied at their Processing Rates for the Hours of the Day

Figure 7.12, shows that the processing required for the different identified queries is calculated on an hourly basis. After the hourly calculations are done, the next step is to identify the "high water mark. "The high water mark is that hour of the day when the most demands will be made of the machine. Figure 7.13, shows the simple identification of the high water mark.

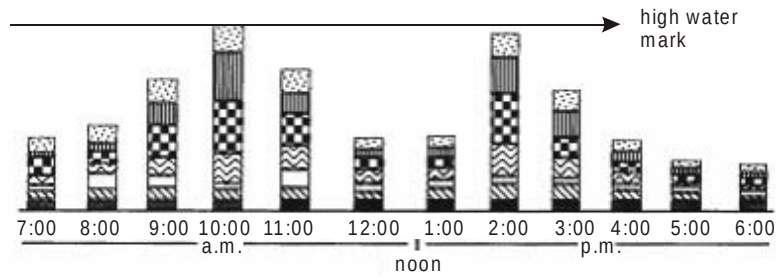


Figure 7.13. The High "Water Mark" for the Day is Determined

After the high water mark requirements are identified, the next requirement is to scope out the requirements for the largest unpredictable request. The largest unpredictable request must be parameterized by:

- How many total I/Os will be required,
- The expected response time, and
- Whether other processing may or may not be quiesced.

Figure 7.14, shows the specification of the largest unpredictable request.

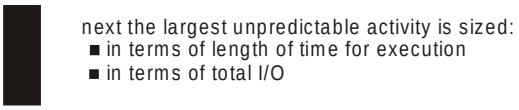


Figure 7.14

After the largest unpredictable request is identified, it is merged with the high water mark. If no quiescing is allowed, then the largest unpredictable request is simply added as another request. If some of the workload (for instance, the predictable DSS processing) can be quiesced, then the largest unpredictable request is added to the portion of the workload that cannot be quiesced. If all of the workload can be quiesced, then the unpredictable largest request is not added to anything.

The analyst then selects the larger of the two–the unpredictable largest request with quiescing (if quiescing is allowed), the unpredictable largest request added to the portion of the workload that cannot be quiesced, or the workload with no unpredictable processing. The maximum of these numbers then becomes the high water mark for all DSS processing. Figure 7.15 shows the combinations.

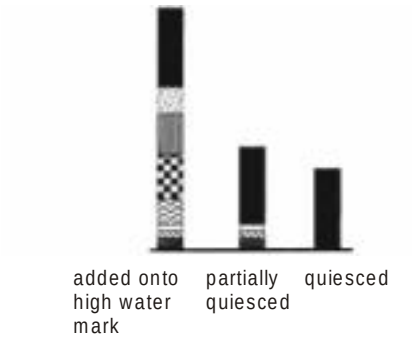


Figure 7.15. The Impact of the Largest Unpredictable Request is Estimated

The maximum number then is compared to a chart of MIPS required to support the level of processing identified, as shown in Figure 7.16.

Figure 7.16, merely shows that the processing rate identified from the workload is matched against a machine power chart.

Of course there is no slack processing factored. Many shops factor in at least ten percent. However, factoring an unused percentage may satisfy the user with better response time, but costs money in any case.

The analysis described here is a general plan for the planning of the capacity needs of a data warehouse. It must be pointed out that the planning is usually done on an iterative basis. In other words, after the first planning effort is done, another more refined version soon follows. In all cases it must be recognized that the capacity planning effort is an estimate.

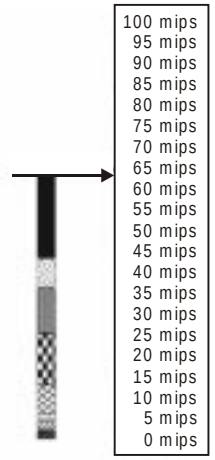


Figure 7.16. Matching the High Water Mark for all Processing Against the Required MIPS

Network Bandwidth

The network bandwidth must not be allowed to slow down the warehouse extraction and warehouse performance. Verify all assumptions about the network bandwidth before proceeding with each rollout.

7.3 DEFINE WAREHOUSE PURGING RULES

Purging rules specify when data are to be removed from the data warehouse. Keep in mind that most companies are interested only in tracking their performance over the last three to five years. In cases where a longer retention period is required, the end users will quite likely require only high-level summaries for comparison purposes. They will not be interested in the detailed or atomic data.

Define the mechanisms for archiving or removing older data from the data warehouse. Check for any legal, regulatory, or auditing requirements that may warrant the storage of data in other media prior to actual purging from the warehouse. Acquire the software and devices that are required for archiving.

7.4 DEFINE SECURITY MANAGEMENT

Keep the data warehouse secure to prevent the loss of competitive information either to unforeseen disasters or to unauthorized users. Define the security measures for the data warehouse, taking into consideration both physical security (i.e., where the data warehouse is physically located), as well as user-access security.

Security management deals with how system integrity is maintained amid possible man-made threats and risks, intentional or unintentional. Intentional man-made threats include espionage, hacks, computer viruses, etc. Unintentional threats include those due to accidents or user ignorance of the effects of their actions. Security management ranges from identification of risks to determination of security measures and controls, detection of violations, and analysis of security violations. This section describes the process steps involved in security management, and discusses factors critical to the success of security management.

Determine and Evaluate of IT Assets

Three types of assets must be identified:

- **Physical.** Computer hardware and software resources, building facilities, and resources used to house sensitive assets or process sensitive information;
- **Information.** Sensitive data pertaining to the company's operations, plans, and strategies. Examples are marketing and sales plans, detailed financial data, trade secrets, personnel information, IT infrastructure data, user profiles and passwords, sensitive office correspondence, minutes of meetings, etc. Lately, there is also concern about protecting company logos and materials posted on the public Internet; and
- **People.** Vital individuals holding key roles, whose incapacity or absence will impact the business in one way or another.

After you identify company assets, the next step is to determine their security level. Depending on the company's requirements, assets may be classified into two, three, or more levels of security, depending on the value of the asset being protected. We recommend having only two levels for organizations with minimal security threats: public and confidential. A three-level security classification scheme can be implemented if security needs are greater: public, confidential, and restricted.

Beware of having too many security levels, as this tends to dilute their importance in the eyes of the user. A large multinational IT vendor used to have four levels of security: public, internal use only, confidential, confidential restricted, and registered confidential. Today, they have cut it down to three: public, internal use only, and confidential. Employees were getting confused as to the differences between the secured levels, and the procedures associated with each one. Having too many security levels proved expensive in terms of employee education, security facilities, and office practices - the costs were often greater than the potential losses from a security violation.

Analyze Risk

Every effective security management system reflects a careful evaluation of how much security is needed. Too little security means the system can easily be compromised intentionally or unintentionally. Too much security can make the system hard to use or degrade its performance unacceptably. Security is inversely proportional to utility - if you want the system to be 100 percent secure, don't let anybody use it. There will always be risks to systems, but often these risks are accepted if they make the system more powerful or easier to use.

Sources of risks to assets can be *intentional* (criminals, hackers, or terrorists; competitors; disgruntled employees; or self-serving employees) or *unintentional* (careless employees; poorly trained users and system operators; vendors and suppliers).

Acceptance of risk is central to good security management. You will never have enough resources to secure assets 100 percent; in fact, this is virtually impossible even with unlimited resources. Therefore, identify all risks to the system, then choose which risks to accept and which to address via security measures. Here are a few reasons why some risks are acceptable:

- The threat is minimal;
- The possibility of compromise is unlikely;

- The value of the asset is low;
- The cost to secure the asset is greater than the value of the asset;
- The threat will soon go away; and
- Security violations can easily be detected and immediately corrected.

After the risks are identified, the next step is to determine the impact to the business if the asset is lost or compromised. By doing this, you get a good idea of how many resources should be assigned to protecting the asset. One user workstation almost certainly deserves fewer resources than the company's servers.

The risks you choose to accept should be documented and signed by all parties, not only to protect the IT organization, but also to make everybody aware that unsecured company assets do exist.

Define Security Practices

Define in detail the following key areas of security management:

- **Asset classification practices**. Guidelines for specifying security levels as discussed above;
- **Risk assessment and acceptance**. As above;
- **Asset ownership.** Assignment of roles for handling sensitive assets;
- **Asset handling responsibilities.** The different tasks and procedures to be followed by the different entities handling the asset, as identified above;
- **Policies regarding mishandling of security assets;**
- **How security violations are reported and responded to;**
- **Security awareness practices.** Education programs, labeling of assets; and
- **Security audits.** Unannounced checks of security measures put in place to find out whether they are functioning.

Implement Security Practices

At this phase, implement the security measures defined in the preceding step. You can do this in stages to make it easier for everybody to adapt to the new working environment. Expect many problems at the start, especially with respect to user resistance to their security tasks, such as using passwords. Staged implementation can be performed:

- **By department,** starting with the most sensitive assets. The natural first choice would be the IT department.
- **By business function or activity,** starting with those that depends upon (or create) the most sensitive assets. You might begin with all Business Planning activities, followed by Marketing, Human Resources, etc.
- **By location,** especially if prioritized sensitive assets are mostly physical. This approach is easiest to implement. However, its effectiveness is doubtful for information assets residing in networked computer systems. You might start with the IT data center, then gradually widen the secured area to encompass the entire business facility.
- **By people,** starting with key members of the organization.

Monitor for Violations and Take Corresponding Actions

An effective security management discipline depends on adequate compliance monitoring. Violations of security practices, whether intentional or unintentional, become more frequent and serious if not detected and acted upon. A computer hacker who gets away with the first system penetration will return repeatedly if he knows no one can detect his activities. Users who get away with leaving confidential documents on their desks will get into bad habits if not corrected quickly.

There are two major activities here: *detecting* security violations and *responding* to them. With respect to sensitive assets, it is important to know:

- Who has the right to handle the assets (user names);
- How to authenticate those asset users (password, IDs, etc.);
- Who has tried to gain access to them;
- How to restrict access to allowed activities; and
- Who has tried to perform actions beyond those that are allowed.

Document the response to security violations, and follow up immediately after a violation is detected. The IT organization should have a Computer Emergency Response Team to deal with security violations. Members of this team should have access to senior management so that severe situations can easily be escalated.

Responses can be built into your security tools or facilities to ensure that the response to a violation is immediate. For example, a password checking utility may be designed to lock out a user name immediately after three invalid password entries. Alarms can be installed around the data center facility so that if any window or door is forced open, security guards or police are immediately notified.

A critical part of this activity is the generation of reports for management that discuss significant security violations and trends of minor incidences. The objective is to spot potential major security violations before they cause serious damage.

Re-evaluate IT Assets and Risks

Security management is a discipline that never rests. Some major changes that would require a reassessment of the security management practice are:

- Security violations are rampant
- Organizational structure or composition changes
- Business environment changes
- Technology changes
- Budget allocation decreases

Additional precautions are required if either the warehouse data or warehouse reports are available to users through an intranet or over the public Internet infrastructure.

7.5 DEFINE BACKUP AND RECOVERY STRATEGY

Define the backup and recovery strategy for the warehouse, taking into consideration the following factors:

- **Data to be backed up.** Identify the data that must be backed up on a regular basis. This gives an indication of the regular backup size. Aside from warehouse data and metadata, the team might also want to back up the contents of the staging or de-duplication areas of the warehouse.
- **Batch window of the warehouse.** Backup mechanisms are now available to support the backup of data even when the system is online, although these are expensive. If the warehouse does not need to be online 24 hours a day, 7 days a week, determine the maximum allowable down time for the warehouse (i.e., determine its batch window). Part of that batch window is allocated to the regular warehouse load and, possibly, to report generation and other similar batch jobs. Determine the maximum time period available for regular backups and backup verification.
- **Maximum acceptable time for recovery.** In case of disasters that result in the loss of warehouse data, the backups will have to be restored in the quickest way possible. Different backup mechanisms imply different time frames for recovery. Determine the maximum acceptable length of time for the warehouse data and metadata to be restored, quality assured, and brought online.
- **Acceptable costs for backup and recovery.** Different backup mechanisms imply different costs. The enterprise may have budgetary constraints that limit its backup and recovery options.

Also consider the following when selecting the backup mechanism:

- **Archive format.** Use a standard archiving format to eliminate potential recovery problems.
- **Automatic backup devices.** Without these, the backup media (e.g., tapes) will have to be changed by hand each time the warehouse is backed up.
- **Parallel data streams.** Commercially available backup and recovery systems now support the backup and recovery of databases through parallel streams of data into and from multiple removable storage devices. This technology is especially helpful for the large databases typically found in data warehouse implementations.
- **Incremental backups.** Some backup and recovery systems also support incremental backups to reduce the time required to back up daily. Incremental backups archive only new and updated data.
- **Offsite backups.** Remember to maintain offsite backups to prevent the loss of data due to site disasters such as fires.
- **Backup and recovery procedures.** Formally define and document the backup and recovery procedures. Perform recovery practice runs to ensure that the procedures are clearly understood.

7.6 SET UP COLLECTION OF WAREHOUSE USAGE STATISTICS

Warehouse usage statistics are collected to provide the data warehouse designer with inputs for further refining the data warehouse design and to track general usage and acceptance of the warehouse.

Define the mechanism for collecting these statistics, and assign resources to monitor and review these regularly.

In Summary

The capacity planning process and the issue tracking and resolution process are critical to the successful development and deployment of data warehouses, especially during early implementations.

The other management and support processes become increasingly important as the warehousing initiative progress further.

CHAPTER 8

DATA WAREHOUSE PLANNING

The data warehouse planning approach presented in this chapter describes the activities related to planning one rollout of the data warehouse. The activities discussed below build on the results of the warehouse strategy formulation described in Chapter 6.

Data warehouse planning further details the preliminary scope of one warehouse rollout by obtaining detailed user requirements for queries and reports, creating a preliminary warehouse schema design to meet the user requirements, and mapping source system fields to the warehouse schema fields. By so doing, the team gains a thorough understanding of the effort required to implement that one rollout.

A planning project typically lasts between five to eight weeks, depending on the scope of the rollout. The progress of the team varies, depending (among other things) on the participation of enterprise resource persons, the availability and quality of source system documentation, and the rate at which project issues are resolved.

Upon completion of the planning effort, the team moves into data warehouse implementation for the planned rollout. The activities for data warehouse implementation are discussed in Chapter 9.

8.1 ASSEMBLE AND ORIENT TEAM

Identify all parties who will be involved in the data warehouse implementation and brief them about the project. Distribute copies of the warehouse strategy as background material for the planning activity.

Define the team setup if a formal project team structure is required. Take the time and effort to orient the team members on the rollout scope, and explain the role of each member of the team. This approach allows the project team members to set realistic expectations about skill sets, project workload, and project scope.

Assign project team members to specific roles, taking care to match skill sets to role responsibilities. When all assignments have been completed, check for unavoidable training requirements due to skill-role mismatches (i.e., the team member does not possess the appropriate skill sets to properly fulfill his or her assigned role).

If required, conduct training for the team members to ensure a common understanding of data warehousing concepts. It is easier for everyone to work together if all have a common goal and an agreed approach for attaining it. Describe the schedule of the planning project to the team. Identify milestones or checkpoints along the planning project timeline. Clearly explain dependencies between the various planning tasks.

Considering the short time frame for most planning projects, conduct status meetings at least once a week with the team and with the project sponsor. Clearly set objectives for each week. Use the status meeting as the venue for raising and resolving issues.

8.2 CONDUCT DECISIONAL REQUIREMENTS ANALYSIS

Decisional Requirements Analysis is one of two activities that can be conducted in parallel during Data Warehouse Planning; the other activity being Decisional Source System Audit (described in the next section). The object of Decisional Requirements Analysis is to gain a thorough understanding of the information needs of decision-makers.

Decisional Requirements Analysis is Working Top-Down

Decisional requirements analysis represents the top-down aspect of data warehousing. Use the warehouse strategy results as the starting point of the decisional requirements analysis; a preliminary analysis should have been conducted as part of the warehouse strategy formulation.

Review the intended scope of this warehouse rollout as documented in the warehouse strategy document. Finalize this scope by further detailing the preliminary decisional requirements analysis. It will be necessary to revisit the user representatives. The rollout scope is typically expressed in terms of the queries or reports that are to be supported by the warehouse by the end of this rollout. The project sponsor must review and approve the scope to ensure that management expectations are set properly.

Document any known limitations about the source systems (e.g., poor data quality, missing data items). Provide this information to source system auditors for their confirmation. Verified limitations in source system data are used as inputs to finalizing the scope of the rollout—if the data are not available, they cannot be loaded into the warehouse.

Take note that the scope strongly influences the implementation time frame for this rollout. Too large a scope will make the project unmanageable. As a general rule, limit the scope of each project or rollout so that it can be delivered in three to six months by a full-time team of 6 to 12 team members.

Conducting Warehouse Planning Without a Warehouse Strategy

It is not unusual for enterprises to go directly into warehouse planning without previously formulating a warehouse strategy. This typically happens when a group of users is clearly driving the warehouse initiative and are more than ready to participate in the initial rollout as user representatives. More often than not, these users have already taken the initiative to list and prioritize their information requirements.

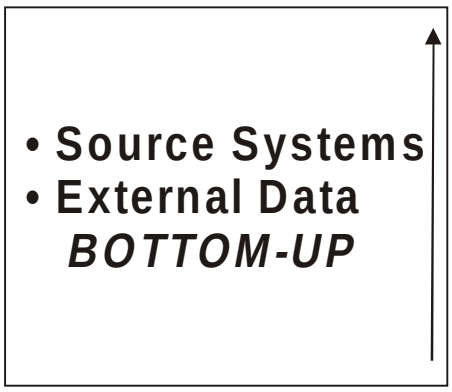
In this type of situation, a number of tasks from the strategy formulation will have to be conducted as part of the planning for the first warehouse rollout. These tasks are as follows:

- **Determine organizational context.** An understanding of the organization is always helpful in any warehousing project, especially since organizational issues may completely derail the warehouse initiative.
- **Define data warehouse rollouts.** Although business users may have already predefined the scope of the first rollout, it helps the warehouse architect to know what lies ahead in subsequent rollouts.
- **Define data warehouse architecture.** Define the data warehouse architecture for the current rollout (and if possible, for subsequent rollouts).
- **Evaluate development and production environment and tools.** The strategy formulation was expected to produce a short-list of tools and computing environments for the warehouse. This evaluation will be finalized during planning by the actual selection of both environments and tools.

8.3 CONDUCT DECISIONAL SOURCE SYSTEM AUDIT

The decisional source system audit is a survey of all information systems that are current or potential sources of data for the data warehouse.

A preliminary source system audit during warehouse strategy formulation should provide a complete inventory of data sources. Identify all possible source systems for the warehouse if this information is currently unavailable.



Data Sources can be Internal or External

Data sources are primarily internal. The most obvious candidates are the operational systems that automate the day-to-day business transactions of the enterprise. Note that

aside from transactional or operational processing systems, one often-used data source in the enterprise general ledger, especially if the reports or queries focus on profitability measurements.

If external data sources are also available, these may be integrated into the warehouse.

DBAs and IT Support Staff are the Best Resource Persons

The best resource persons for a decisional source system audit of internal systems are the database administrators (DBAs), system administrators and other IT staff who support each internal system that is a potential source of data. With their intimate knowledge of the systems, they are in the best position to gauge the suitability of each system as a warehouse data source.

These individuals are also more likely to be familiar with any data quality problems that exist in the source systems. Clearly document any known data quality problems, as these have a bearing on the data extraction and cleansing processes that the warehouse must support. Known data quality problems also provide some indication of the magnitude of the data cleanup task.

In organizations where the production of managerial reports has already been automated (but not through an architected data warehouse), the DBAs and IT support staff can provide very valuable insight about the data that are presently collected. These staff members can also provide the team with a good idea of the business rules that are used to transform the raw data into management reports.

Conduct individual and group interviews with the IT organization to understand the data sources that are currently available. Review all available documentation on the candidate source systems. This is without doubt one of the most time-consuming and detailed tasks in data warehouse planning, especially if up-to-data documentation of the existing systems is not readily available.

As a consequence, the whole-hearted support of the IT organization greatly facilitates this entire activity.

Obtain the following documents and information if these have not yet been collected as part of the data warehouse strategy definition:

- **Enterprise IT architecture documentation.** This refers to all documentation that provides a bird's eye view of the IT architecture of the enterprise, including but not limited to:
  - System architecture diagrams and documentation—A model of all the information systems in the enterprise and their relationships to one another.
  - Enterprise data model—A model of all data that currently stored or maintained by the enterprise. This may also indicate which systems support which data item.
  - Network architecture—A diagram showing the layout and bandwidth of the enterprise network, especially for the locations of the project team and the user representatives participating in this rollout.
- **User and technical manuals of each source system.** This refers to data models and schemas for all existing information systems that are candidate's data sources.

If extraction programs are used for ad hoc reporting, obtain documentation of these extraction programs as well. Obtain copies of all other available system documentation, whenever possible.

- **Database sizing.** For each source system, identify the type of database used, the typical backup size, as well as the backup format and medium. It is helpful also to know what data are actually backed up on a regular basis. This is particularly important if historical data are required in the warehouse and such data are available only in backups.
- **Batch window.** Determine the batch windows for each of the operational systems. Identify all batch jobs that are already performed during the batch window. Any data extraction jobs required to feed the data warehouse must be completed within the batch windows of each source system without affecting any of the existing batch jobs already scheduled. Under no circumstances will the team want to disrupt normal operations on the source systems.
- **Future enhancements.** What application development projects, enhancements, or acquisition plans have been defined or approved for implementation in the next 6 to 12 months, for each of the source systems? Changes to the data structure will affect the mapping of source system fields to data warehouse fields. Changes to the operational systems may also result in the availability of new data items or the loss of existing ones.
- **Data scope.** Identify the most important tables of each source system. This information is ideally available in the system documentation. However, if definitions of these tables are not documented, the DBAs are in the best position to provide that information. Also required are business descriptions or definitions of each field in each important table, for all source systems.
- **System codes and keys.** Each of the source systems no doubt uses a set of codes for the system will be implementing key generation routines as well. If these are not documented, ask the DBAs to provide a list of all valid codes and a textual description for each of the system codes that are used. If the system codes have changed over time, ask the DBAs to provide all system code definitions for the relevant time frame. All key generation routines should likewise be documented. These include rules for assigning customer numbers, product numbers, order numbers, invoice numbers, etc. check whether the keys are reused (or recycled) for new records over the years. Reused keys may cause errors during reduplication and must therefore be thoroughly understood.
- **Extraction mechanisms.** Check if data can be extracted or read directly from the production databases. Relational databases such as oracle or Sybase are open and should be readily accessible. Application packages with proprietary database management software, however, may present problems, especially if the data structures are not documented. Determine how changes made to the database are tracked, perhaps through an audit log. Determine also if there is a way to identify data that have been changed or updated. These are important inputs to the data extraction process.

8.4 DESIGN LOGICAL AND PHYSICAL WAREHOUSE SCHEMA

Design the data warehouse schema that can best meet the information requirements of this rollout. Two main schema design techniques are available:

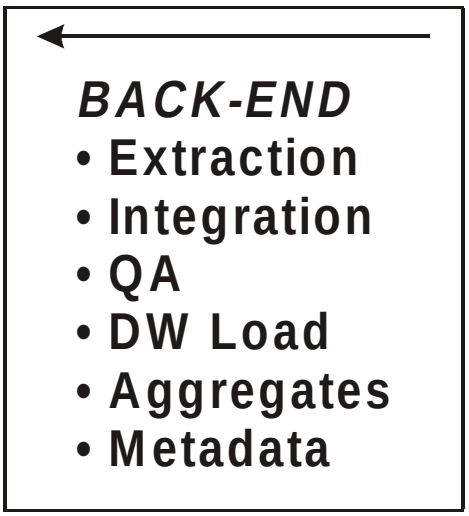
- **Normalization.** The database schema is designed using the normalization techniques traditionally used for OLTP applications;
- **Dimensional modeling.** This technique produces demoralized, star schema designs consisting of fact and dimension tables. A variation of the dimensional star schema also exists (i.e., snowflake schema).

There are ongoing debates regarding the applicability or suitability of both these modeling techniques for data warehouse projects, although dimensional modeling has certainly been gaining popularity in recent years. Dimensional modeling has been used successfully in larger data warehousing implementations across multiple industries. The popularity of this modeling technique is also evident from the number of databases and front-end tools that now support optimized performance with star schema designs (e.g., Oracle RDBMS 8, R/olap XL).

A discussion of dimensional modeling techniques is provided in Chapter 12.

8.5 PRODUCE SOURCE-TO-TARGET FIELD MAPPING

The Source-To-Target Field Mapping documents how fields in the operational (source) systems are transformed into data warehouse fields. Under no circumstances should this mapping be left vague or open to misinterpretation, especially for financial data. The mapping allows non-team members to audit the data transformations implemented by the warehouse.



Many-to-Many Mappings

A single field in the data warehouse may be populated by data from more than one source system. This is a natural consequence of the data warehouse's role of integrating data from multiple sources.

The classic examples are customer name and product name. Each operational system will typically have its own customer and product records. A data warehouse field called customer name or product name will therefore be populated by data from more than one systems.

Conversely, a single field in the operational systems may need to be split into several fields in the warehouse. There are operational systems that still record addresses as lines of text, with field names like address line 1, address line[2], etc. these can be split into multiple address fields such as street name, city, country and Mail/Zip code. Other examples are numeric figures or balances that have to be allocated correctly to two or more different fields.

To eliminate any confusion as to how data are transformed as the data items are moved from the source systems to the warehouse database, create a source-to-target field mapping that maps each source field in each source system to the appropriate target field in the data warehouse schema. Also, clearly document all business rules that govern how data values are integrated or split up. This is required for each field in the source-to-target field mapping.

The source-to-target field mapping is critical to the successful development and maintenance of the data warehouse. This mapping serves as the basis for the data extraction and transformation subsystems. Figure 8.1 shows an example of this mapping.

			TARGET	No.	1	2	3	4	5	6	7	
				Schema	R1	R1	R1	R1	R1	R1	R1	
SOURCE				Table	TT1	TT1	TT1	TT2	TT2	TT2	TT2	
No.	System	Table		Fields	TF1	TF2	TF3	TF4	TF5	TF6	TF7	
1	SS1	ST1		SF1								
2	SS1	ST1		SF2								
3	SS1	ST1		SF3								
4	SS1	ST1		SF4								
5	SS1	ST2		SF5								
6	SS1	ST2		SF6								
7	SS2	ST2		SF7								
8	SS2	ST3		SF8								
9	Ss2	ST3		SF9								
10	SS2	ST3		SF10								
...	...	...		...	...	...	...	...	...	...	...	

SOURCE: SS1 = Source System1.ST1= Source Table 1. SF1 = Source Field 1
TARGET: R1 = Rollout1.TT1 = Target Table1. TF1 = Target Field 1

Figure 8.1. Sample Source-to-Target Field Mapping.

Revise the data warehouse schema on an as-needed basis if the field-to-field mapping yields missing data items in the source systems. These missing data items may prevent the warehouse from producing one or more of the requested queries or reports. Raise these types of scope issues as quickly as possible to the project sponsors.

Historical Data and Evolving Data Structures

If users require the loading of historical data into the data warehouse, two things must be determined quickly:

- **Changes in schema.** Determine if the schemas of all source systems have changed over the relevant time period. For example, if the retention period of the data

warehouse is two years and data from the past two years have to be loaded into the warehouse, the team must check for possible changes in source system schemas over the past two years. If the schemas have changed over time, the task of extracting the data immediately becomes more complicated. Each different schema may require a different source-to-target field mapping.

- **Availability of historical data.** Determine also if historical data are available for loading into the warehouse. Backups during the relevant time period may not contain the required data item. Verify assumptions about the availability and suitability of backups for historical data loads.

These two tedious tasks will be more difficult to complete if documentation is out of data or insufficient and if none of the IT professionals in the enterprise today are familiar with the old schemas.

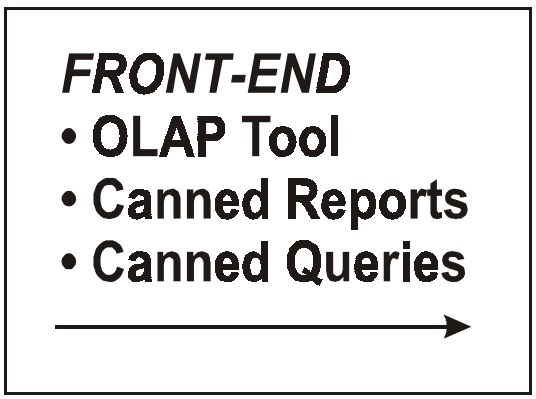
8.6 SELECT DEVELOPMENT AND PRODUCTION ENVIRONMENT AND TOOLS

Finalize the computing environment and tool set for this rollout based on the results of the development and production environment and tools study during the data warehouse strategy definition. If an exhaustive study and selection had been performed during the strategy definition stage, this activity becomes optional.

If, on the other hand, the warehouse strategy was not formulated, the enterprise must now evaluate and select the computing environment and tools that will be purchased for the warehousing initiative. This activity may take some time, especially if the evaluation process requires extensive vendor presentations and demonstrations, as well as site visits. This activity is therefore best performed early on to allow for sufficient time to study and select the tools. Sufficient lead times are also required for the delivery (especially if importation is required) of the selected equipment and tools.

8.7 CREATE PROTOTYPE FOR THIS ROLLOUT

Using the short-listed or final tools and production environment, create a prototype of the data warehouse.



A prototype is typically created and presented for one or more of the following reasons:

- **To assists in the selection of front-end tools.** It is sometimes possible to ask warehousing vendors to present a prototype to the evaluators as part of the selection

process. However, such prototypes will naturally not be very specific to the actual data and reporting requirements of the rollout.

- **To verify the correctness of the schema design.** The team is better served by creating a prototype using the logical and physical warehouse schema for this rollout. If possible, use actual data from the operational systems for the prototype queries and reports. If the user requirements (in terms of queries and reports) can be created using the schema, then the team has concretely verified the correctness of the schema design.
- **To verify the usability of the selected front-end tools.** The warehousing team can invite representatives from the user community to actually use the prototype to verity the usability of the selected front-end tools.
- **To obtain feedback from user representatives.** The prototype is often the first concrete output of the planning effort. It provides users with something that they can see and touch. It allows users to experience for the first time the kind of computing environment they will have when the warehouse is up. Such an experience typically triggers a lot of feedback (both positive and negative) from users. It may even cause users to articulate previously unstated requirements.

  Regardless of the type of feedback, however, it is always good to hear what the users have to say as early as possible. This provides the team more time to adjust the approach or the design accordingly.

  During the prototype presentation meeting, the following should be made clear to the business users who will be viewing or using the prototype:
- **Objective of the prototype meeting.** State the objectives of the meeting clearly to properly orient all participants. If the objective is to select a tool set, then the attention and focus of users should be directed accordingly.
- **Nature of data used.** If actual data from the operational systems are used with the prototype, make clear to all business users that the data have not yet been quality assured. If dummy or test data are used, then this should be clearly communicated as well. Users who are concerned with the correctness of the prototype data have unfortunately sidetracked many prototype presentations.
- **Prototype scope.** If the prototype does not yet mimic all the requirements identified for this rollout, then say so. Don't wait for the users to explicitly ask whether the team has considered (or forgotten!) the requirements they had specified in earlier meetings or interviews.

8.8 CREATE IMPLEMENTATION PLAN OF THIS ROLLOUT

With the scope now fully defined and the source-to-target field mapping fully specified, it is now possible to draft an implementation plan for this rollout. Consider the following factors when creating the implementation plan:

- **Number of source systems, and their related extraction mechanisms and logistics.** The more source systems there are, the more complex the extraction and integration processes will be. Also, source systems with open computing environments present fewer complications with the extraction process than do proprietary systems.

- **Number of decisional business processes supported.** The larger the number of decisional business processes supported by this rollout, the more users there are who will want to have a say about the data warehouse contents, the definition of terms, and the business rules that must be respected.
- **Number of subject areas involved.** This is a strong indicator of the rollout size. The more subject areas there are, the more fact tables will be required. This implies more warehouse fields to map to source systems and, of course, a larger rollout scope.
- **Estimated database size.** The estimated warehouse size provides an early indication of the loading, indexing, and capacity challenges of the warehousing effort. The database size allows the team to estimate the length of time it takes to load the warehouse regularly (given the number of records and the average length of time it takes to load and index each record).
- **Availability and quality of source system documentation.** A lot of the team's time will be wasted on searching for or misunderstanding the data that are available in the source systems. The availability of good-quality documentation will significantly improve the productivity of source system auditors and technical analysts.
- **Data quality issues and their impact on the schedule.** Unfortunately, there is no direct way to estimate the impact of data quality problems on the project schedule. Any attempts to estimate the delays often produce unrealistically low figures, much to the concentration of warehouse project managers. Early knowledge and documentation of data quality issues will help the team to anticipate problems. Also, data quality is very much a user responsibility that cannot be left to IT to solve. Without sufficient user support, data quality problems will continually be a thorn in the side of the warehouse team.
- **Required warehouse load rate.** A number of factors external to the warehousing team (particularly batch windows of the operational systems and the average size of each warehouse load) will affect the design and approach used by the warehouse implementation team.
- **Required warehouse availability.** The warehouse itself will also have batch windows. The maximum allowed down time for the warehouse also influences the design and approach of the warehousing team. A fully available warehouse (24 hours × 7 days) requires an architecture that is completely different from that required by a warehouse that is available only 12 hours a day, 5 days a week. These different architectural requirements naturally result in differences in cost and implementation time frame.
- **Lead time for delivery and setup of selected tools, development, and production environment.** Project schedules sometimes fail to consider the length of time required to setup the development and production environments of the warehousing project. While some warehouse implementation tasks can proceed while the computing environments and tools are on their way, significant progress cannot be made until the correct environment and tool sets are available.

- **Time frame required for IT infrastructure upgrades.** Some IT infrastructure upgrades (e.g., network upgrade or extension) may be required or assumed by the warehousing project. These dependencies should be clearly marked on the project schedule. The warehouse Project Manager must coordinate with the infrastructure Project Manager to ensure that sufficient communication exists between all concerned teams.
- **Business sponsor support and user participation.** There is no way to overemphasize the importance of Project Sponsor support and end user participation. No amount of planning by the warehouse Project Manager (and no amount of effort by the warehouse project team) can make up for the lack of participation by these two parties.
- **IT support and participation.** Similarly, the support and participation of the database administrators and system administrators will make a tremendous difference to the overall productivity of the warehousing team.
- **Required vs. existing skill sets.** The match (or mismatch) of personnel skill sets and role assignments will likewise affect the productivity of the team. If this is an early or pilot project, then training on various aspects of warehousing will most likely be required. These training sessions should be factored into the implementation schedule as well and, ideally, should take place before the actual skills are required.

8.9 WAREHOUSE PLANNING TIPS AND CAVEATS

The actual data warehouse planning activity will rarely be a straightforward exercise. Before conducting your planning activity, read through this section for planning tips and caveats.

Follow the Data Trail

In the absence of true decision support systems, enterprises have, over the years, been forced to find stopgap or interim solutions for producing the managerial or decisional reports that decision-makers require. Some of these solutions require only simple extraction programs that are regularly run to produce the required reports. Other solutions require a complex series of steps that combine manual data manipulation, extraction programs, conversion formulas, and spreadsheet macros.

In the absence of a data warehouse, many of the managerial reporting requirements are classified as ad hoc reports. As a result, most of these report generation programs and processes are largely undocumented and are known only by the people who actually produce the reports. This naturally leads to a lack of standards (i.e., different people may apply different formulas and rules to the same data item), and possible inconsistencies each time the process is executed. Fortunately, the warehouse project team will be in a position to introduce standards and consistent ways of manipulating data.

Following the data trail (see Figure 8.2) from the current management reports, back to their respective data sources can prove to be a very enlightening exercise for data warehouse planners.

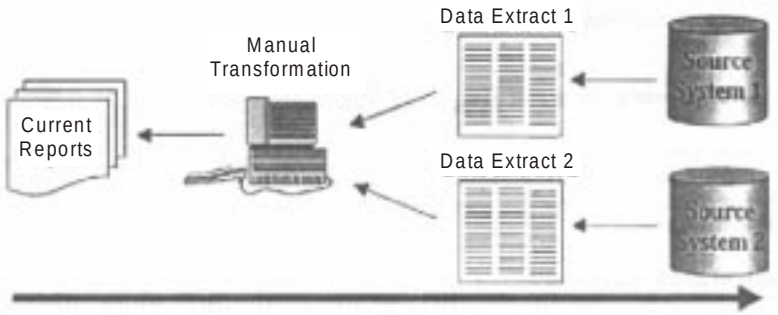


Figure 8.2. Follow the Data Trail

Through this exercise, the data warehouse planner will find:

- **All data sources currently used for decisional reporting.** At the very least, these data sources should also be included in the decisional source system audit. The team has the added benefit of knowing before hand which fields in these systems are considered important.
- **All current extraction programs.** The current extraction programs are a rich input for the source-to-target field mapping. Also, if these programs manipulate or transform or convert the data in any way, the transformation rules and formulas may also prove helpful to the warehousing effort.
- **All undocumented manual steps to transform the data.** After the raw data have been extracted from the operational systems, a number of manual steps may be performed to further transform the data into the reports that enterprise managers. Interviews with the appropriate persons should provide the team with an understanding of these manual conversion and transformation steps (if any).

Apart from the above items, it is also likely that the data warehouse planner will find subtle flaws in the way reports are produced today. It is not unusual to find inconsistent use of business terms formulas, and business rules, depending on the person who creates and reads the reports. This lack of standard terms and rules contributes directly to the existence of conflicting reports from different groups in the same enterprise, i.e., the existence of “different versions of the truth”.

Limitations Imposed by Currently Available Data

Each data item that is required to produce the reports required by decision-makers comes from one or more of the source systems available to the enterprise. Understandably, there will be data items that are not readily supported by the source systems.

Data limitations generally fall into one of the following types.

Missing Data Items

A data item is considered missing, if no provisions were made to collect or store this data item in any of the source systems. This omission particularly occurs with data items that may have no bearing on the day-to-day operations of the enterprise but will have tactical or managerial implications.

For example, not all loan systems record the industry to which each loan customer belongs; from an operational level, such information may not necessarily be considered critical. Unfortunately, a bank that wishes to track its risk exposure for any given industry will not be able to produce an industry exposure report if customer industry data are not available at the source systems.

Incomplete (optional) Data Items

A data item may be classified as "nice to have" in the operational systems, and so provisions are made to store the data, but no rules are put in place to enforce the collection of such data. These optional data items are available for some customer products, accounts, or orders but may be unavailable for others.

Returning to the above example, a loan system may have a field called customer industry, but the application developers may have made the field optional, in recognition of the fact that data about a customers industry are not ready available in cases such as this, only customers with actual data can be classified meaningfully in the report.

Wrong Data

Errors occur when data are stored in one or more source systems but are not accurate. There are many potential reasons or causes for this, including the following ones:

- **Data entry error.** A genuine error is made during data entry. The wrong data are stored in the database.
- **Data item is mandatory but unavailable.** A data item may be defined as mandatory but it may not be readily available, and the random substitution of other information has no direct impact on the day-to-day operations of the enterprise. This implies that any data can be entered without adversely affecting the operation processes.

Returning to the above example, if customer industry was a mandatory customer data item and the person creating the customer record does not know the industry to which the customer belongs, he is likely to select, at random, any of the industry codes that are recognized by the system. Only by so doing will he or she be able to create the customer record.

Another data item that can be randomly substituted is the social security number, especially if these numbers are stored for reference purposes only, and not for actual processing. Data entry personnel remain focused on the immediate task of creating the customer record which the system refuses to do without all the mandatory data items. Data entry personnel are rarely in a position to see the consequences to recording the wrong data.

Improvements to Source Systems

From the above examples, it is easy to see how the scope of a data warehousing initiative can be severely compromised by data limitations in the source systems. Most pilot data warehouse projects are thus limited only to the data that are available. However, improvements can be made to the source systems in parallel with the warehousing projects. The team should therefore properly document any source system limitations that are encountered. These documents can be used as inputs to upcoming maintenance projects on the operational systems.

A decisional source system audit report may have a source system review section that covers the following topics:

- **Overview of operational systems.** This section lists all operational systems covered by the audit. A general description of the functionality and data of each operational system is provided. A list of major tables and fields may be included as an appendix. Current users of each of the operational systems are optionally documented.
- **Missing data items.** List all the data items that are required by the data warehouse but are currently not available in the source systems. Explain why each item is important (e.g., cite reports or queries where these data items are required). For each data item, identify the source system where the data item is best stored.
- **Data quality improvement areas.** For each operational system, list all areas where the data quality can be improved. Suggestions as to how the data quality improvement can be achieved can also be provided.
- **Resource and effort estimate.** For each operational system, it might be possible to provide an estimate or the cost and length of time required to either add the data item or improve the data quality for that data item.

In Summary

Data warehouse planning is conducted to clearly define the scope of one data warehouse rollout. The combination of the top-down and bottom-up tracks gives the planning process the best of both worlds—a requirements-driven approach that is grounded on available data.

The clear separation of the front-end and back-end tracks encourages the development of warehouse subsystems for extracting, transporting, cleaning, and loading warehouse data independently of the front-end tools that will be used to access the warehouse.

The four tracks converge when a prototype of the warehouse is created and when actual warehouse implementation takes place.

Each rollout repeatedly executes the four tracks (top-down, bottom-up, back-end), and the scope of the data warehouse is iteratively extended as a result Figure 8.3 illustrates the concept.

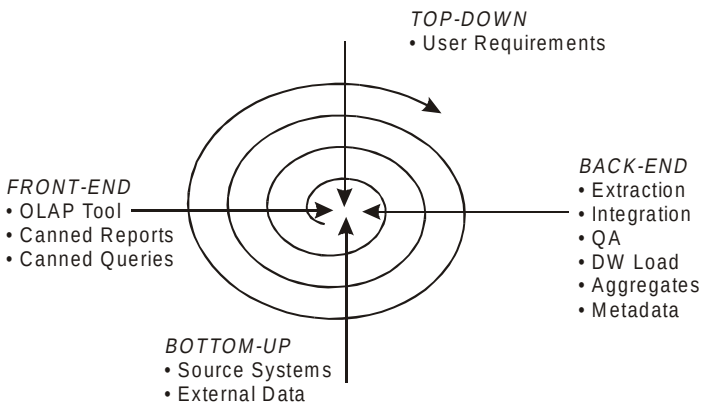


Fig. 8.3

CHAPTER 9

DATA WAREHOUSE IMPLEMENTATION

The data Warehouse implementation approach presented in this chapter describes the activities related to implementing one rollout of the date warehouse. The activities discussed here are built on the results of the data warehouse planning described in the previous chapter.

The data warehouse implementation team builds or extends an existing warehouse schema based on the final logical schema design produced during planning. The team also builds the warehouse subsystems that ensure a steady, regular flow of clean data from the operational systems into the data warehouse. Other team members install and configure the selected front-end tools to provide users with access to warehouses data.

An implementation project should be scoped to last between three to six months. The progress of the team varies, depending (among other things) on the quality of the warehouse design, the quality of the implementation plan, the availability and participation of enterprise resource persons, and the rate at which project issues are resolved.

User training and warehouse testing activities take place towards the end of the implementation project, just prior to the deployment to users. Once the warehouse has been deployed, the day-to-day warehouse management, maintenance, and optimization tasks begin. Some members of the implementation team may be asked to stay on and assist with the maintenance activities to ensure continuity. The other members of the project team may be asked to start planning the next warehouse rollout or may be released to work on other projects.

9.1 ACQUIRE AND SET UP DEVELOPMENT ENVIRONMENT

Acquire and set up the development environment for the data warehouse implementation project. This activity includes the following tasks, among others: install the hardware, the operating system, the relational database engine; install all warehousing tools; create all necessary network connections; and create all required user IDs and user access definitions.

Note that most data warehouses reside on a machine that is physically separate from the operational systems. In addition, the relational database management system used for data warehousing need not be the same database management system used by the operational systems.

At the end of this task, the development environment is set up, the project team members are trained on the (new) development environment, and all technology components have been purchased and installed.

9.2 OBTAIN COPIES OF OPERATIONAL TABLES

There may be instances where the team has no direct access to the operational source systems from the warehouse development environment. This is especially possible for pilot projects, where the network connection to the warehouse development environment may be available.

Regardless of the reason for the lack of access, the warehousing team must establish and document a consistent, reliable, and easy-to-follow procedure for obtaining copies of the relevant tables from the operational systems. Copies of these tables are made available to the warehousing team on another medium (most likely tape) and are restored on the warehouse server. The creation of copies can also be automated through the use of replication technology.

The warehousing team must have a mechanism for verifying the correctness and completeness of the data that are loaded onto the warehouse server. One of the most effective completeness checks is meaningful business counts (e.g., number of customers, number of accounts, number of transactions) that are computed and compared to ensure data completeness. Data quality utilities can help assess the correctness of the data.

The use of copied tables as described above implies additional space requirements on the warehouse server. This should not be a problem during the pilot project.

9.3 FINALIZE PHYSICAL WAREHOUSE SCHEMA DESIGN

Translate the detailed logical and physical warehouse design from the warehouse planning stage into a final physical warehouse design, taking into consideration the specific, selected database management system.

The key considerations are :

- **Schema design.** Finalize the physical design of the fact and dimension tables and their respective fields. The warehouse database administrator (DBA) may opt to divide one logical dimension (e.g., customer) into two or more separate ones (e.g., a customer dimension and a customer demographic dimension) to save on space and improve query performance.
- **Indexes.** Identify the appropriate indexing method to use on the warehouse tables and fields, based on the expected data volume and the anticipated nature of warehouse queries. Verify initial assumptions made about the space required by indexes to ensure that sufficient space has been allocated.
- **Partitioning.** The warehouse DBA may opt to partition fact and dimension tables, depending on their size and on the partitioning features that are supported by the database engine. The warehouses DBA who decides to implement partitioned views must consider the trade-offs between degradation in query performance and improvements in warehouse manageability and space requirements.

9.4 BUILD OR CONFIGURE EXTRACTION AND TRANSFORMATION SUBSYSTEMS

Easily 60 percent to 80 percent of a warehouse implementation project is devoted to the back-end of the warehouse. The back-end subsystems must extract, transform, clean, and load the operational data into the data warehouse. Understandably, the back-end subsystems vary significantly from one enterprise to another due to differences in the computing environments, source systems, and business requirements. For this reason, much of the warehousing effort cannot simply be automated away by warehousing tools.

Extraction Subsystem

The first among the many subsystems on the back-end of the warehouse is the data extraction subsystem. The term extraction refers to the process of retrieving the required data from the operational system tables, which may be the actual tables or simply copies that have been loaded into the warehouse server.

Actual extraction can be achieved through a wide variety of mechanisms, ranging from sophisticated third-party tools to custom-written extraction scripts or programs developed by in house IT staff. Third-party extraction tools are typically able to connect to mainframe, midrange and UNIX environments, thus freeing their users from the nightmare of handling heterogeneous data sources. These tools also allow users to document the extraction process (i.e., they have provisions for storing metadata about the extraction).

These tools, unfortunately, are expensive. For this reason, organizations may also turn to writing their own extraction programs. This is a particularly viable alternative if the source systems are on a uniform or homogenous computing environment (e.g., all data reside on the same RDBMS, and they make use of the same operating system). Custom-written extraction programs, however, may be difficult to maintain, especially if these programs are not well documented. Considering how quickly business requirements will change in the warehousing environment, ease of maintenance is an important factor to consider.

Transformation Subsystem

The transformation subsystem literally transforms the data in accordance with the business rules and standards that have been established for the data warehouse.

Several types of transformations are typically implemented in data warehousing.

- **Format changes.** Each of the data fields in the operational systems may store data in different formats and data types. These individual data items are modified during the transformation process to respect a standard set of formats. For example, all data formats may be changed to respect a standard format, or a standard data type is used for character fields such as names, addresses.
- **De-duplication.** Records from multiple sources are compared to identify duplicate records based on matching field values. Duplicates are merged to create a single record of a customer; a product, an employee, or a transaction, Potential duplicates are logged as exceptions that are manually resolved. Duplicate records with conflicting data values are also logged for manual correction if there is no system of record to provide the “master” or “correct” value.

- **Splitting up fields.** A data item in the source system may need to be split up into one or more fields in the warehouse. One of the most commonly encountered problems of this nature deals with customer addresses that have simply been stored as several lines of text. These textual values may be split up into distinct fields; street number, street name, building name, city, mail or zip code, country, etc.
- **Integrating fields.** The opposite of splitting up fields is integration. Two or more fields in the operational systems may be integrated to populate one warehouse field.
- **Replacement of values.** Values that are used in operational systems may not be comprehensible to warehouse users. For example, system codes that have specific meanings in operational systems are meaningless to decision-makers. The transformation subsystem replaces the original with new values that have a business meaning to warehouse users.
- **Derived values.** Balances, ratios, and other derived values can be computed using agreed formulas. By pre-computing and loading these values into the warehouse, the possibility of miscomputation by individual users is reduced. A typical example of a pre-computed value is the average daily balance of bank accounts. This figure is computed using the base data and is loaded as it is into the warehouse.
- **Aggregates.** Aggregates can also be pre-computed for loading into the warehouse. This is an alternative to loading only atomic (base-level) data in the warehouse and creating in the warehouse the aggregate records based on the atomic warehouse data.

The extraction and transformation subsystems (see Figure 9.1) create load images, i.e., tables and fields populated with the data that are to be loaded into the warehouse. The load images are typically stored in tables that have the same schema as the warehouse itself. By doing so, the extraction and transformation subsystems greatly simplify the load process.

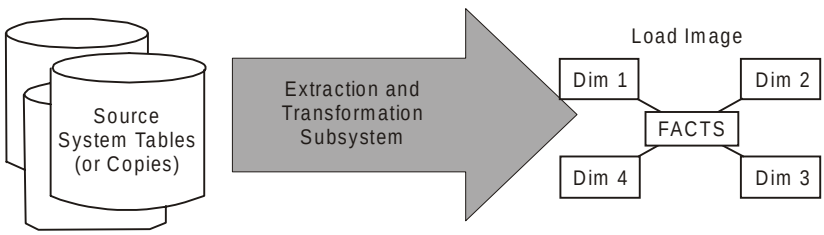


Figure 9.1. Extraction and Transformation Subsystems

9.5 BUILD OR CONFIGURE DATA QUALITY SUBSYSTEM

Data quality problems are not always apparent at the start of the implementation project, when the team is concerned more about moving massive amounts of data rather than the actual individual data values that are being moved. However, data quality (or to be more precise, the lack of it) will quickly become a major, show-stopping problem if it is not addressed directly.

One of the quickest ways to inhibit user acceptance is to have poor data quality in the warehouse. Furthermore, the perception of data quality is in some ways just as important

as the actual quality of the data warehouse. Data warehouse users will make use of the warehouse, only if they believe that the information they will retrieve from it is correct. Without user confidence in the data quality, a warehouse initiative will soon lose support and eventually die off.

A data quality subsystem on the back-end of the warehouse therefore is critical component of the overall warehouse architecture.

Causes of Data Errors

An understanding of the causes of data errors makes these errors easier to find. Since most data errors originate from the source systems, source system database administrators and system administrators, with their day-to-day experiences working with the sources systems are very critical to the data quality effort.

Data errors typically result from one or more of the following causes:

- **Missing values.** Values are missing in the sources systems due to either incomplete records or optional data fields.
- **Lack of referential integrity.** Referential integrity in source systems may not be enforced because of inconsistent system codes or codes whose meanings have changed over time.
- **Different units of measure.** The use of different currencies and units of measure in different source systems may lead to data errors in the warehouse if figures or amounts are not first converted to a uniform currency or unit of measure prior to further computations or data transformation.
- **Duplicates.** De-duplication is performed on source system data prior to the warehouse load. However, the de-duplication process depends on comparison of data values to find matches. If the data are not available to start with, the quality of the de-duplication may be compromised. Duplicate records may therefore be loaded into the warehouse.
- **Fields to be split up.** As mentioned earlier, there are times when a single field in the sources system has to be split up to populate multiple warehouse fields. Unfortunately, it is not possible to manually split up the fields one at a time because of the volume of the data. The team often resorts to some automated form of field splitting, which may not be 100 percent correct.
- **Multiple hierarchies.** Many warehouse dimensions will have multiple hierarchies for analysis purposes. For example, the time dimension typically has day-month-quarter-year hierarchy. This same time dimension may also have a day-week hierarchy and a day-fiscal, month-fiscal, quarter-fiscal year hierarchy. Lack of understanding of these multiple hierarchies in the different dimensions may result in erroneous warehouse loads.
- **Conflicting or inconsistent terms and rules.** The conflicting or inconsistent use of business terms and business rules may mislead warehouse planners into loading two distinctly different data items into the same warehouse field, or vice versa. Inconsistent business rules may also cause the misuse of formulas during data transformation.

Data Quality Improvement Approach

Below is an approach for improving the overall data quality of the enterprise.

- **Assess current level of data quality.** Determine the current data quality level of each of the warehouse sources systems. While the enterprise may have a data quality initiative that is independent of the warehousing project, it is best to focus the data quality efforts on warehouse sources systems—these systems obviously contains data that are of interest to enterprise decision—makers.
- **Identify key data items.** Set the priorities of the data quality team by identifying the key data items in each of the warehouse source systems. Key data items, by definition, are the data items that must achieve and maintain a high level of data quality. By prioritizing data items in this manner, the team can target its efforts on the more critical data areas and therefore provides greater value to the enterprise.
- **Define cleansing tactics for key data items.** For each key data item with poor data quality, define an approach or tactic for cleaning or raising the quality of that data item. Whenever possible, the cleansing approach should target the source systems first, so that errors are corrected at the source and not propagated to other systems.
- **Define error-prevention tactics for key data items.** The enterprise should not stop at error-correction activities. The best way to eliminate data errors is to prevent them from happening in the first place. If error-producing operational processes are not corrected, they will continue to populate enterprise databases with erroneous data. Operational and data-entry staff must be made aware of the cost of poor data quality. Reward mechanisms within the organization may have to be modified to create a working environment that focuses on preventing data errors at the source.
- **Implement quality improvement and error-prevention processes.** Obtain the resources and tools to execute the quality improvement and error-prevention procession. After some time, another assessment may be conducted, and a new set of key data items may be targeted for quality improvement.

Data Quality Assessment and Improvements

Data quality assessments can be conducted at any time at different points along the warehouse back-end. As shown in Figure 9.2, assessments can be conducted on the data while it is in the source systems, in warehouse loads images or in the data warehouse itself.

Note that while data quality products assist in the assessment and improvement of data quality, it is unrealistic to expect any single program or data quality product to find and correct all data quality errors in the operational systems or in the data warehouse. Nor is it realistic to expect data quality improvements to be completed in a matter of months. It is unlikely that an enterprise will ever bring its databases to a state that is 100 percent error free.

Despite the long-term nature of the effort, however, the absolute worst thing that any warehouse Project Manager can do is to ignore the data quality problem in the vain hope that it will disappear. The enterprise must be willing and prepared to devote time and effort to the tedious task of cleaning up data errors rather than sweeping the problem under the rug.

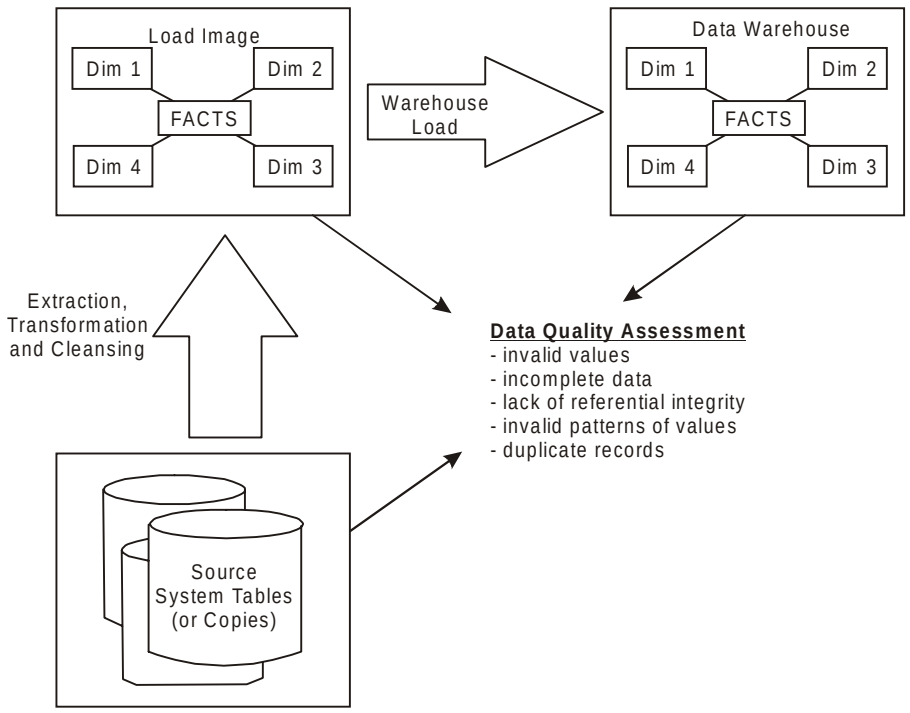


Figure 9.2. Data Quality Assessments at the Warehouse Back-End

Correcting Data Errors at the Source

All data errors found are, under ideal circumstances, corrected at the source, i.e., the operational system database is updated with the correct values. This practice ensures that subsequent data users at both the operational and decisional levels will benefit from clean data.

Experience has shown, however, that correcting data at the source may prove difficult to implement for the following reasons:

- **Operational responsibility.** The responsibility for updating the source system data will naturally fall into the hands of operational staff, who may not be so inclined to accept the additional responsibility of tracking down and correcting past data-entry errors.
- **Correct data are unknown.** Even if the people in operations know that the data in a given record are wrong, there may be no easy way to determine the correct data. This is particularly true of customer data (e.g., a customer's social security number). The people in operations have no other resource but to approach the customers one at a time to obtain the correct data. This is tedious, time-consuming, and potentially irritating to customers.

Other Considerations

Many of the available warehousing tools have features that automate different areas of the warehouse extraction, transformation, and data quality subsystems.

The more data sources there are, the higher the likelihood of data quality problems. Likewise, the larger the data volume, the higher the number of data errors to correct.

The inclusion of historical data in the warehouse will also present problems due to changes (over time) in system codes, data structures, and business rules.

9.6 BUILD WAREHOUSE LOAD SUBSYSTEM

The warehouse load subsystem takes the load images created by the extraction and transformation subsystems and loads these images directly into the data warehouse. As mentioned earlier, the data to be loaded are stored in tables that have the same schema design as the warehouse itself. The load process is therefore fairly straightforward from a data standpoint.

Basic Features of a Load Subsystem

The load subsystem should be able to perform the following:

- **Drop indexes on the warehouse.** When new records are inserted into an indexed table, the relational database management system immediately updates the index of the table in response. In the context of a data warehouse load, where up to hundreds of thousands of records are inserted in rapid succession into one single table, the immediate re-indexing of the table after each insert results in a significant processing overhead. As a consequence, the load process slows down dramatically. To avoid this problem, drop the indexes on the relevant warehouse table prior to each load.
- **Load dimension records.** In the source systems each record of a customer, product, or transaction is uniquely identified through a key. Likewise, the customers, products, and transactions in the warehouse must be identifiable through a key value. Source system keys are often inappropriate as warehouse keys, and a key generation approach is therefore used during the load process. Insert new dimension records, or update existing records based on the load images.
- **Load fact records.** The primary key of a Fact table is the concatenation of the keys of its related dimension records. Each fact record therefore makes use to the generated keys of the dimension records. Dimension records are loaded prior to the fact records to allow the enforcement of referential integrity checks. The load subsystem therefore inserts new fact records or updates old records based on the load images. Since the data warehouse is essentially a time series, most of the records in the Fact table will be new records.
- **Compute aggregate records, using base fact and dimension records.** After the successful load of atomic or base level data into the warehouse, the load subsystem may now compute aggregate records by using the base-level fact and dimension records. This step is performed only if the aggregates are not pre-computed for direct loading into the warehouse.
- **Rebuild or regenerate indexes.** Once all loads have been completed, the indexes on the relevant tables are rebuilt or regenerated to improve query performance.
- **Log load perceptions.** Log all referential integrity violations during the load process as load exceptions. There are two types of referential integrity violations: (*a*) missing key values one of the key fields of the fact record does not have a value: and (*b*) wrong key values the key fields have values, but one or more of them do

not have a corresponding dimension record. In both cases, the warehousing team has option of (*a*) not loading the record until the correct key values are found or (*b*) loading the record, but replacing the missing or wrong key values with hard-coded values that users can recognize as a load exception.

The load subsystem, as described above, assumes that the load images do not yet make use of warehouse keys; i.e., the load images contain only source system keys. The warehouse keys are therefore generated as part of the load process.

Warehousing teams may opt to separate the key generation routines from the load process. In this scenario, the key generation routine is applied on the initial load images (i.e., the load images created by the extraction and transformation subsystems). The final load images (with warehouse keys) are then loaded into the warehouse.

Loading Dirty Data

There are ongoing debates about loading dirty data (i.e., data that fail referential integrity checks) into the warehouse. Some teams prefer to load only clean data into the warehouse, arguing that dirty data can mislead and misinform. Others prefer to load all data, both clean and dirty, provided that the dirty data are clearly marked as dirty. Depending on the extent of data errors, the use of only clean data in the warehouse can be equal to or more dangerous than relying on a mix of clean and dirty data. If more than 20 percent of data are dirty and only 80 percent clean data are loaded into the warehouse, the warehouse users will be making decisions based on an incomplete picture.

The use of hard-coded values to identify warehouse data with referential integrity violations on one dimension allows warehouse users to still make use of the warehouse data on clean dimensions.

Consider the example in Figure 9.3. If a Sales Fact record is dependent on Customer, Data (Time dimension) and Product and if the Customer key is missing, then a “Sales per Product” report from the warehouse will still produce the correct information.

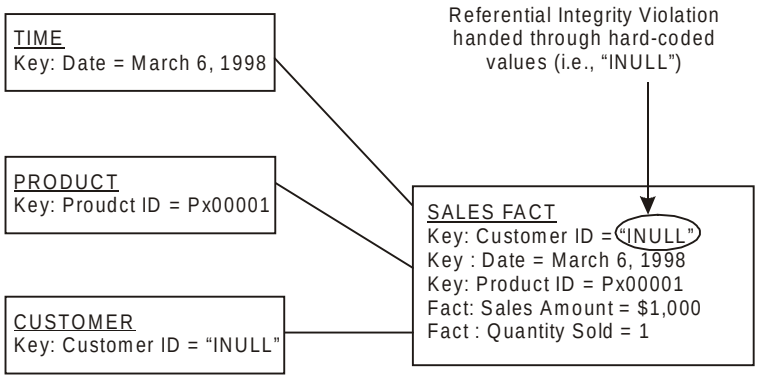


Figure 9.3. Loading Dirty Data

When a “sales per customer” report is produced (as shown in Figure 9.4), the hard-coded value that signifies a referential integrity violation will be listed as a customer ID, and the user is aware that the corresponding sales amount cannot be attributed to valid customer.

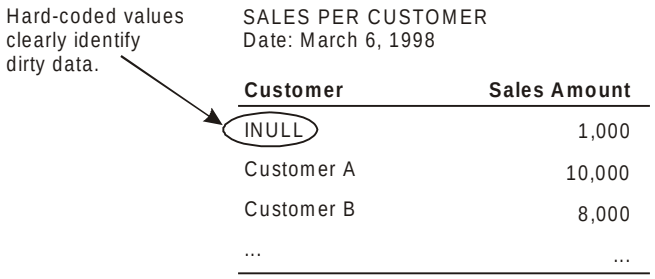


Figure 9.4. Sample Report with Dirty Date Identified Through Hard-Coded Value.

By handling a referential integrity violation during warehouse loads in the manner described above. The users get full picture of the facts on clean dimensions and are clearly aware when dirty dimensions are used.

The Need for Load Optimization

The time required for a regular warehouse load is often of great concern to warehouse designers and project managers. Unless the warehouse was designed and architects to be fully available 24 hours a day, the warehouse will be offline and unavailable to its users during the load period.

Much of the challenge in building the load subsystem therefore lies in optimizing the load process to reduce the total time required. For this reason, parallel load features in later releases of relational database management systems and parallel processing capabilities in SMP and MPP machines are especially welcome in data warehousing implementations.

Test Loads

The team may like to test the accuracy and performance of the warehouse load subsystem on dummy data before attempting a real load with actual load images. The team should know as early as possible how much load optimization work is still required.

Also, by using dummy data, the warehousing team does not have to wait for the completion of the extraction and transformation subsystems to start testing the warehouse load subsystem.

Warehouse load subsystem testing of course, is possible only if the data warehouse schema is already up and available.

Set Up Data Warehouse Schema

Create the data warehouse schema in the development environment while the team is constructing or configuring the warehouse back-end subsystems (i.e., the data extraction and transformation subsystems, the data quality subsystem, and the warehouse load subsystem).

As part of the schema setup, the warehouse DBA must do the following:

- **Create warehouse tables.** Implement the physical warehouse database design by creating all base-level fact and dimension tables, core and custom tables, and aggregate tables.

- **Build Indexes.** Build the required indexes on the tables according to the physical warehouse database design.
- **Populate special referential tables and records.** The data warehouse may require special referential tables or records that are not created through regular warehouse loads. For example, if the warehouse team uses hard-coded values to handle load with referential integrity violations, the warehouse dimension tables must have records that use the appropriate hard-coded value to identify fact records that have referential integrity violations. It is usually helpful to populate the data warehouse with test data as soon as possible. This provides the front-end team with the opportunity to test the data access and retrieval tools; even while actual warehouse data are not available. Figure 9.5 presents a typical data warehouse scheme.

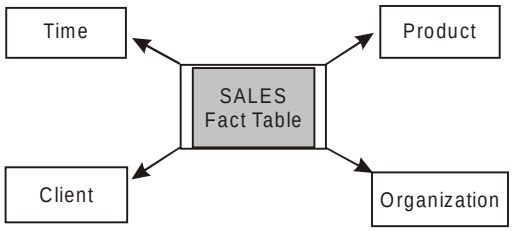


Figure 9.5. Sample Warehouse Schema

9.7 SET UP WAREHOUSE METADATA

Metadata have traditionally been defined as "data about data." While such a statement does not seem very helpful, it is actually quite appropriate as a definition-metadata describe the contents of the data warehouse, indicate where the warehouse data originally came from, and document the business rules that govern the transformation of the data.

Warehousing tools also use metadata as the basis for automating certain aspects of the warehousing project. Chapter 13 in the Technology section of this book discusses metadata in depth.

9.8 SET UP DATA ACCESS AND RETRIEVAL TOOLS

The data access and retrieval tools are equivalent to the tip of the warehousing iceberg. While they may represent as little as 10 percent of the entire warehousing effort, they all are that users of the warehouse. As a result, these tools are critical to the acceptance and usability of the warehouse.

Acquire and Install Data Access and Retrieval Tools

Acquire and install the selected data access tools in the appropriate environments and machines. The front-end team will find it prudent to first install the selected data access tools on a test machine that has access to the warehouse. The test machine should be loaded with the software typically used by the enterprise. Through this activity, the front-end team may identify unforeseen conflicts between the various software programs without causing inconvenience to anyone.

Verify that the data access and retrieval tools can establish and hold connections to the data warehouse over the corporate network. In the absence of actual warehouse data, the team may opt to use test data in the data warehouse schema to test the installation of front-end tools.

Build Predefined Reports and Queries

The prototype initially developed during warehouse planning is refined by incorporating user feedback and by building all predefined reports and queries that have been agreed on with the end-users.

Different front-end tools have different requirements for the efficient distribution of predefined reports and queries to all users. The front-end team should therefore perform the appropriate administration tasks as required by the selected front-end tools.

By building these predefined reports and queries, the data warehouse implementation team is assured that the warehouse schema meets the decisional information requirements of the users.

Support staff, who will eventually manage the warehouse Help Desk should participate in this activity, since this participation provides excellent learning opportunities.

Set Up Role or Group Profiles

Define role or group profiles on the database management system. The major database management systems provide the use of a role or a group to define the access rights of multiple users through one role definition.

The warehousing team must determine the appropriate role definitions for the warehouse. The following roles are recommended as a minimum:

- **Warehouse user.** The typical warehouse user is granted Select rights on the production warehouse tables. Updates are granted on the warehouse dimension records.
- **Warehouse administrator.** This role is assigned to users strictly for the direct update of data warehouse dimension records. *Select* and *update* rights are granted on the warehouse dimension record.
- **Warehouse developer.** This role applies to any warehouse implementation team member who works on the back-end of the warehouse. Users with this role can create development warehouse objects but cannot modify or update the structure and content of production warehouse tables.

Set Up User Profiles and Map These to Role Profiles

Define user profiles for each warehouse user and assign one or more roles to each user profile to grant the user access to the warehouse. While it is possible for multiple users to use the same user profile, this practice is greatly discouraged for the following reasons:

- **Collection of warehouse statistics.** Warehouse statistics are collected as part of the warehouse maintenance activities. The team will benefit from knowing (*a*) how many users have access to the warehouse. (*b*) which users are actually making use of the warehouse, and (*c*) how often a particular user makes use of the warehouse.
- **Warehouse security.** The warehousing team must be able to track the use of the warehouse to a specific individual, not just to a group of individuals. Users may also be less careful with Ids and passwords if they know these are shared. Unique user Ids are also required should the warehouse start restricting or granting access based on record values in warehouse tables (e.g., a branch manager can see only the records related to his or her branch).

- **Audit Trail.** If one or more users have Update access to the warehouse, distinct, users IDs will allow the warehouse team to track down the warehouse user responsible for each update.
- **Query performance complaints.** In case where a query on the warehouse server is slow or has stalled, the warehouse administrator will be better able to identify the slow or stalled query when each user has a distinct user ID.

9.9 PERFORM THE PRODUCTION WAREHOUSE LOAD

The production data warehouse load can be performed only when the load images are ready and both the warehouse schema and metadata are setup.

Prior to the actual production warehouse load, it is a good practice to conduct partial loads to get some indication of the total load time. Also, since the data warehouse schema design may require refinement, particularly when the front-end tools are first setup, it will be easier and quicker to make changes to the data warehouse schema when very little data have been loaded. Only when the end users have had a chance to provide positive feedback should large volumes of data be loaded into the warehouse.

Data warehouses are not refreshed more than once every 24 hours. If the user requirements call for up-to-the-minute information for operational monitoring purposes, then a data warehouse is not the solution; these requirements should be addressed through an Operational Data Store.

The warehouse is typically available to end-users during the working day. For this reason, warehouse loads typically take place at night or over a weekend.

If the retention period of the warehouse is several years; the warehouse team should first load the data for the current time period and verify the correctness of the load. Only when the load is successful should the team start loading historical data into the warehouse. Due to potential changes to the schemas of the sources systems over the past few years, it is natural for the warehouse team to start from the most current period and work in reverse chronological order when loading historical data.

9.10 CONDUCT USER TRAINING

The IT organization is encouraged to fully take over the responsibility of conducting user training, contrary to the popular practice of having product vendors or warehouse consultant to assist in the preparation of the first warehousing classes. Doing so will enable the warehousing team to conduct future training courses independently.

Scope of User Training

Conduct training for all intended users of this rollout of the data warehouses. Prepare training materials if required. The training should cover the following topics:

- **What is a warehouse?** Different people have different expectations of what a data warehouse is. Start the training with a warehouse definition.
- **Warehouse scope.** All users must know the contents of the warehouses. The training should therefore clearly state what is not supported by the current warehouse rollout. Trainers might need to know what functionality has been deferred to later phases, and why.
- **Use of front-end tools.** The users should learn how to use the front-end tools. Highly usable front-ends should require fairly little training. Distribute all relevant user documentation to training participants.

- **Load timing and publication.** Users should be informed of the schedule for warehouse loads (e.g., "the warehouse is loaded with sales data on a weekly basis, and a special month-end load is performed for the GL expense data"). Users should also know how the warehouse team intends to publish the results of each warehouse load.
- **Warehouse support structure.** Users should know how to get additional help from the warehousing team. Distribute Help Desk phone numbers, etc.

Who Should Attend the Training?

Training should be conducted for all intended end users of the data warehouse. Some senior managers, particularly those who do not use computers every day, may ask their assistants or secretaries to attend the training in their place. In this scenario, the senior manager should be requested to attend at least the portion of the training that deals with the warehouse scope.

Different Users Have Different Training Needs

An understanding of the users computing literacy provides insight to the type and pace of training required. If the user base is large enough, it may be helpful to divide the trainees into two groups - a basic class and an advanced class. Power users will otherwise quickly become bored in a basic class, and beginners will feel overwhelmed if they are in an advanced class. Attempting to meet the training needs of both types of users in one class may prove to be difficult and frustrating.

At the end of the training, it is a good practice to identify training participants who would require post-training follow up and support from the warehouse implementation team. Also ask participants to evaluate the warehouse training, constructive criticism will allow the trainers to deliver better training in the future.

Training as a Prerequisite to Testing

A subset of the users will be requested to test the warehouse. This user subset may have to undergo user training earlier than others, since user training is a prerequisite to user testing. Users cannot adequately test the warehouse if they do not know what is in it or how to use it.

9.11 CONDUCT USER TESTING AND ACCEPTANCE

The data warehouse, like any system, must undergo user testing and acceptance. Some considerations are discussed below:

Conduct Warehouse Trails

Representatives of the end-user community are requested to test this warehouse rollout. In general, the following aspects should be tested.

- **Support of specified queries and reports.** Users test the correctness of the queries and reports of this warehouse rollout. In many cases, this is achieved by preparing the same report manually (or through existing mechanisms) and comparing this report to the one produced by the warehouse. All discrepancies are accounted for, and the appropriate corrections are made. The team should not discount the possibility that the errors are in the manually prepared report.

- **Performance/response time.** Under the most ideal circumstances, each warehouse query will be executed in less than one or two seconds. However, this may not be realistically achievable, depending on the warehouse size (number of rows and dimensions) and the selected tools. Warehouse optimization at the hardware and database levels can be used to improve response times. The use of stored aggregates will likewise improve warehouse performance.
- **Usability of client front-end·** Training on the front-end tools is required, but the tools must be usable for the majority of the users.
- **Ability to meet report frequency requirements.** The warehouse must be able to provide the specified queries and reports at the frequency (i.e., daily, weekly, monthly, quarterly, or yearly) required by the users.

Acceptance

The rollout is considered accepted when the testing for this rollout is completed to the satisfaction of the user community. A concrete set of acceptance criteria can be developed at the start of the warehouse rollout for use later as the basis for acceptance. The acceptance criteria are helpful to users because they know exactly what to test. It is likewise helpful to the warehousing team because they know what must be delivered.

In Summary

Data warehouse implementation is without question the most challenging part of data warehousing. Not only will the team have to resolve the technical difficulties of moving, integrating, and cleaning data, they will also face the more difficult task of addressing policy issues, resolving organizational conflicts, and untangling logistical delays.

In general, the following areas present more problems during warehouse implementation and bear the more watching.

- **Dirty data.** The identification and cleanup of dirty data can easily consume more resources than the project can afford.
- **Underestimated logistics.** The logistics involved in warehousing typically require more time than originally expected. Tasks such as installing the development environment, collecting source data, transporting data, and loading data are generally beleaguered by logistical problems. The time required to learn and configure warehousing tools likewise contributes to delays.
- **Policies and political issues.** The progress of the team can slow to a crawl if a key project issue remains unresolved for too long.
- **Wrong warehouse design.** The wrong warehouse design results in unmet user requirements or inflexible implementations. It also creates rework for the schema as well as all the back-end subsystems; extraction and transformation, quality assurance, and loading.

At the end of the project, however, a successful team has the satisfaction of meeting the information needs of key decision-makers in a manner that is unprecedented in the enterprise.

PART IV : TECHNOLOGY

A quick browse through the Process section of this book makes it quite clear that a data warehouse project requires a wide array of technologies and tools. The data warehousing products market (particularly the software segment) is a rapidly growing one; new vendors continuously announce the availability of new products, while existing vendors add warehousing-specific features to their existing product lines.

Understandably, the gamut of tools makes tool selection quite confusing. This section of the book aims to lend order to the warehousing tools market by classifying these tools and technologies. The two main categories, understandably, are:

- **Hardware and Operating Systems.** These refer primarily to the warehouse servers and their related operating systems. Key issues include database size, storage options, and backup and recovery technology.
- **Software.** This refers primarily to the tools that are used to extract, clean, integrate, populate, store, access, distribute, and present warehouse data. Also included in this category are metadata repositories that document the data warehouse. The major database vendors have all jumped on the data warehousing bandwagon and have introduced, or are planning to introduce, features that allow their database products to better support data warehouse implementations.

In addition, this section of the book focuses on two key technology issues in data warehousing:

- **Warehouse Schema Design.** We present an overview of the dimensional modeling techniques, as popularized by Ralph Kimball, to produce database designs that are particularly suited for warehousing implementations.
- **Warehouse Metadata.** We provide a quick look at warehouse metadata—what it is, what it should encompass, and why it is critical in data warehousing.