

Data Warehousing with the SAS/Warehouse Administrator

Tony Fisher SAS Institute Inc., Cary, NC

ABSTRACT

Few topics have taken hold of the computer industry the way that data warehousing has over the past few years. The SAS System has long been used to provide data warehouse functionality. The SAS System provided this functionality even before the term data warehouse was coined.

This paper will cover:

- ☐ why has data warehousing become such an important issue
- ☐ an explanation of data warehouse architecture and how existing tools in the SAS System may be employed in the design and implementation of that architecture.
- ☐ a description of the SAS/Warehouse Administrator, a new tool from SAS Institute that will facilitate faster and easier data warehouse implementations. This paper presentation is primarily a demonstration of the SAS/Warehouse Administrator.

Data warehousing consists of a series of data refinements and reformat in an effort to move data from a operational design to a design that more closely reflects decision support applications and a business model. The paper will explore data warehousing concepts in general and then will drill down into how the SAS System can be used to make the concepts a reality.

INTRODUCTION

Buzzwords and new concepts come and go in the Information Processing industry, but some ideas seem to have value that ensures a more lasting permanence. Data warehousing is one of those ideas. Data warehousing has moved beyond being just the latest hot new trend to becoming a mainstream, integral part of many businesses. There are many reasons why this has happened:

- ☐ Operational systems were not conducive to decision support applications. Queries were difficult to form, because operational physical database models were designed to get information into the system, rather than out of the system.
- ☐ Decision support activities negatively impacted the performance of operational systems. Decision support analysis has a more random pattern of hardware utilization, thus causing sporadic, inconsistent response time for operational system users. Sometimes this could have serious consequences, such as system outages or poor customer service.
- ☐ End users found that the information they were getting from different operational systems was not consistent and credible. Data values that should be identical were different or they were in different formats or units. Additionally, because information was not derived from the same place, solutions were not as easily shared between departments or divisions.
- ☐ A long term, historical view of the data was difficult to perform on an operational system, since most operational systems were designed to keep current, up to the moment data. Retaining large amounts of older data for decision support purposes impacted the performance of operational users.
- ☐ Other external trends, such as the increasing affordability of disk space and newer modeling technologies, such as star schema, promoted the exploration and research of data warehouse architectures at many businesses.

For these and other reasons, many businesses have found that data warehousing has supplied the missing ingredient in their use of information technology to promote better decision-making.

DATA WAREHOUSE ARCHITECTURE AND THE SAS SYSTEM

The usefulness of the information in the data warehouse and the ease by which it can be accessed is heavily dependent on the overall architecture and physical design of the warehouse. Some data warehouse architecture topics to consider are enumerated in the following sections.

Data Flow

To build a data warehouse, you must understand the major phases that data passes through as it flows from the operational environment to the warehouse environment. There are four different phases:

- ☐ Extraction
- ☐ Transformation
- ☐ Population or loading into the warehouse environment

☐ Summarization/archival

Before you can begin to map the data flow, you must identify the operational data sources that feed the data warehouse. Operational data sources may exist on legacy systems or be distributed across the enterprise network. Operational data sources may reside in a number of different formats, although data in Data Base Management Systems or flatter file formats, such as VSAM, are common.

Operational data sources may easily be defined to SAS Software by creating views. SAS views can be used to define the layout, or schema, the table or file you wish to access. The table typically resides in another location, or in another format. Views have the advantage of shielding the application from complicated access methods or data structures. Additionally, views may be used to limit what data is presented to the application. For example, views may be created on to DBMS tables with SAS/Access software. Views on to other file formats, such as VSAM, may be created with the DATA step in SAS Software or with the new External File Interface tool in version 6.12 of the SAS System.

Once the operational data sources are defined, a method of extracting data from the operational data sources must be created. An important consideration in the extraction phase is deciding how much data should be extracted from the operational system. Are the warehouse tables to be totally refreshed with a full load of data from the operational system or are the warehouse tables updated with just the transactions from the operational system that have changed?

PROC SQL is a highly flexible method of extracting data from operational systems. The SAS System implements an ANSI-compliant version of the language that can be used not only to extract data from operational data sources, but also to join, subset, and concatenate data. Powerful WHERE clause processing permits you to extract data based on your subset criteria. Additionally, PROC SQL allows you to pass the SQL statements directly to the DBMS you are accessing, thus allowing the DBMS to evaluate and optimize the SQL in its native state.

The SAS System also has an interactive SQL component called the SQL Query Window. The SQL Query Window allows you to easily define views and tables to local or remote Database Management Systems for extraction purposes.

Following the extraction phase is the transformation phase. The term "transformation" is an umbrella term for all the tasks that prepare data prior to loading it in the warehouse. The transformation phase can encompass many different functions:

- ☐ Data validation, also known as data scrubbing or data cleaning. This is either removing or recoding operational data for invalid, missing, or out-of-range values.
- ☐ Data integration. This is the process of reconciling data that may be in different formats to a single, consistent format in the warehouse.
- ☐ Data addition. This is the creation of new data values prior to loading into warehouse. An example could be the addition of a time stamp or deriving new values from a formula.
- ☐ Data restructuring. This can include denormalizing or joining data to optimize queries.
- ☐ Data summarization. This is the process of aggregating data to reduce the number of transactions entering the warehouse.

The SAS System is ideally suited to the transformation of data. The data step contains a rich set of mathematical, statistical, and other types of functions that can automate the process of transforming data. For example, the PUT and INPUT functions can be used to convert numeric data to character and character data to numeric. Sophisticated string functions can permit you to transform existing data values or create new data values.

SAS System formats are convenient tools to use to validate or transform values on a table-driven basis. Input data can be changed or discarded based on its value in the format table. Additionally, the DATA step may be used to apply logic to input data and transform or discard data based on IF-THEN/ELSE or SELECT processing.

The DATA step also contains facilities to concatenate, interleave, or match merge data. PROC SQL may also be used to combine or join tables for denormalization purposes.

Following the transformation phase is the load or population phase. Here the transformed data is actually loaded into the warehouse tables. If the data contains just those changed transactions from the operational system, then those transactions must be applied to the target warehouse tables in either an append or update fashion.

The SAS System implements efficient loading via the PROC COPY, PROC DATASETS, or PROC APPEND procedures. Additionally, PROC DBLOAD in SAS/ACCESS software may be used to load data into many different DBMS physical stores.

The last phase of data flow is summarization/archiving. Summarization is discussed in more detail in the following section. Archival is the aging out of either detail or summary warehouse data to a less expensive storage media, typically tape. This can be done during each load of a warehouse table, or performed in a separate job.

Granularity of Warehouse Data

The data warehouse may contain many different aggregations of data. The volume of transactions that are loaded in some warehouses is such that it may be inefficient to perform decision support analysis against those large quantities. Therefore, some summarization of data may be employed to make queries more efficient. Additionally, many types of analysis frequently require a view of the "big picture", i.e., analysis of data that spans several months or years.

Most data warehouses contain detail data, lightly summarized data and heavily summarized data. Detail data, also known as atomic data, is data on an individual transaction basis. This is the data that is usually loaded into the warehouse from the initial load phase. Not surprisingly, detail data is almost always the most current data in the warehouse. Some warehouse implementations keep only one load's worth of data, typically one day or one week. Other implementations may keep several weeks or months of detail data.

Lightly and heavily summarized data typically span longer time periods -- months or even years. Decision support analysis against summary tables is more efficient due to lower quantities of data, however, the warehouse administrator must carefully plan how the data is to be summarized to ensure that the end user's queries are optimized for that summarization level. Also, the warehouse administrator must determine what statistics are to be computed in the summary tables and how those tables are to be ordered or subset.

The SAS System contains useful tools to perform summarization, the data step itself being the most flexible. Additionally, PROC SUMMARY and PROC SQL may be used to compute a wide variety of summary statistics against warehouse data, for example, mean, count, sum, and standard deviation.

As noted earlier, the process of summarization is actually one of the data flow phases. Data at the detail level is typically aged out and summarized into the summary tables when it reaches its retention date thereby keeping the volume of data at the detail level relatively constant, while still retaining the essentials of that data for historical reporting and analysis.

A new data storage format that may be used to easily explore data in a summarized fashion is the new multi-dimensional database format or MDDB. The MDDB database provides for the pre-summarization of large amounts of data in a matrix-like structure. Pre-summarizing the data makes it available for immediate use with MDDB viewers and other business objects. Users may then rapidly navigate and drill through the data, looking at multiple relationships concurrently.

Location of Warehouse Data

One aspect in designing the architecture of the data warehouse is planning the physical location of the data. In the older, monolithic model of the data warehouse, the central data warehouse was kept on some larger system, typically MVS or Unix, with most users accessing the warehouse directly.

In the newer, distributed model, the metadata, described in more detail below, defines the locations of the warehouse tables that may be dispersed on numerous systems throughout the enterprise. This model has advantage of placing data on the computing platform that is the best fit for that data. For example, warehouse data that requires a high level of security may best be suited for placement on the corporate mainframe. Data that is subject to a high degree of slicing and dicing decision support analysis may best reside closer to the end user, for example, on a desktop PC. Naturally, network load and other performance factors must be taken into account when configuring a distributed system.

The SAS System has two products that make distribution of warehouse data easy -- SAS/CONNECT software and SAS/SHARE software. SAS/CONNECT software provides communication links to permit the transport of data between different hosts, the execution of programs on remote hosts, and the transparent access of data residing on a remote host. SAS/SHARE provides similar transparent access of remote data, but permits that data to be shared by multiple users.

Physical Storage of Warehouse Data

There are numerous aspects in the design of the physical storage of data. You must decide what format the data resides in, for example, DBMS tables, SAS datasets, multi-dimensional databases, etc. SAS datasets are excellent data stores, both for their physical characteristics and their orientation toward decision support. Multi-dimensional databases are most effective as summary tables or datamarts, since they easily allow the slicing and dicing of data along various hierarchies or dimensions.

A new data store with release 6.12 of the SAS System is the Scalable Performance Data Server, or SPDS. SPDS software will split queries into several sub-queries to be executed in parallel. The results from each sub-query is then joined before presenting the answer set to the user. This has the effect of greatly speeding up turnaround time for long queries.

Indexing, compression, and partitioning schemes are also important factors to consider when maximizing the performance of the data warehouse.

Queries may be both easier to form and run quicker if the physical structure of the warehouse tables is designed using a star schema model. A star schema is a physical model based on a central fact table surrounded by some number of dimension tables. The fact table typically contains the measured facts, such as transaction dollar amounts, with the other columns being foreign keys into the dimension tables. The dimension tables contain the attributes of that particular dimension. For example, there could be a customer dimension table that consists of a customer id, which functions as the key, along with customer name, address, location, and other customer attributes.

Queries against a star schema model typically require fewer joins than similar queries against tables configured in the traditional entity relation model.

Metadata

Metadata, or data about data, is one of the most important aspects of data warehouse architecture. Not only does the metadata allow the warehouse administrator to document the inventory and processes of the warehouse, but it also permits the end user or business analyst to easily search for topics of interest in the warehouse. Typically, the analyst or administrator will use a hierarchical or directory-based tool to navigate the metadata.

Some of types of metadata that may be stored in the warehouse are:

- ☐ Definition of the business subjects in the warehouse
- ☐ Table names, locations and attributes.
- ☐ Column information
- ☐ Owner/custodian of the data
- ☐ Definition of how tables are summarized
- ☐ Free-form information, i.e., business metadata
- ☐ Definition of how the data was extracted and transformed prior to entering the warehouse.

The SAS/Warehouse Administrator product, described in more detail later, contains a robust metadata component that not only provides the necessary function of documenting the warehouse, but is also used to generate the code that loads the warehouse.

SAS/WAREHOUSE ADMINISTRATOR

In 1995, at SAS Institute, we recognized that it would be helpful to supply a framework that would gather together many of the tools in the SAS System that lend themselves toward data warehousing. This toolset, known as SAS/Warehouse Administrator, will be generally available in the second half of 1996. The SAS/Warehouse Administrator will make it even easier to build, maintain, and access a data warehouse by bringing the strengths of the SAS System relative to data warehousing together into one consistent interface plus adding additional capabilities not currently available in other SAS system products.

The features available to users of SAS/Warehouse Administrator are grouped into three broad categories:

- ☐ **Manage**
All facilities to build and maintain the data warehouse. Building refers to both the initial build of the data warehouse and subsequent iterative modifications and enhancements to the data warehouse. Metadata is created/changed with this feature.
- ☐ **Organize**
Data warehouse management capabilities for day-to-day operations. Metadata may be changed with this feature, for example, by the Job Scheduler.
- ☐ **Exploit**
Features that make the data warehouse accessible to end users. Metadata is not changed with this feature.

The site administrator of SAS/Warehouse Administrator product, also known as the Data Warehouse Administrator (DWA), will primarily use the features to manage and organize the data warehouse environment. A decision support analyst or business user may use the manage and organize features, for example, to build a personal data warehouse, but will primarily use the exploit features to access existing warehouses built by the DWA.

To facilitate data warehouse definition of data entities and data flow between those entities, some of the SAS/Warehouse Administrator supports:

- ☐ defining subjects in the data warehouse,
- ☐ defining different types of data collections, i.e. detail data, summaries, data marts,
- ☐ identifying sources of operational data,
- ☐ defining extraction techniques,
- ☐ defining transformations to transform operational data to warehouse data,
- ☐ defining different summarization techniques, and populating the target files in the warehouse,
- ☐ inserting user exits in the load jobs.

As the warehouse is defined, metadata describing the warehouse entities and relationships will be automatically generated and maintained by SAS/Warehouse Administrator.

The data warehouse environment is organized around the major subjects of a business enterprise, not process-oriented applications. SAS/Warehouse Administrator will permit flexible definition of warehouse data by subject area. Each subject will support a number of different data collections that may reside as SAS data sets or database tables, such as Oracle or Sybase. Data is collected into a detail level then summarized into one or more summary groups. These summary groups may then be broken down into time-based summary levels, for example, weekly, monthly, or yearly summaries. Additionally, custom-built data marts may be created from the detail or summary levels of one or more subjects to provide a more departmental or user-specific view of the data. These datamarts may be stored in any storage format, including the MDDB format. SAS/Warehouse Administrator also supports the storage of not only data, but information. Information marts are the storage vehicle for reports, charts, graphs, stored queries, and may even be used as application launchers from within the warehouse environment. The SAS/Warehouse Administrator stores this information in a logical container known as an Information Mart (or InfoMart.)

SAS/Warehouse Administrator will support access to operational data in the following formats:

- ☐ SAS data set or view
- ☐ any DBMS supported by a SAS/ACCESS engine
- ☐ external files flat files or VSAM

Operational data sources can be distributed across the network and are accessed using SAS/CONNECT, SAS/SHARE, ODBC, or the 6.11 filename engines (FTP, SOCKET). The DWA can specify whether or not to extract a full copy of the data or just extract the data that has changed since the last extraction. Additionally, other subset criteria may be applied during the extraction phase for data scrubbing purposes. User-written extraction programs will be supported.

After defining operational sources and extraction techniques, a number of different transformation techniques may be chosen to assist the DWA in integrating the warehouse data in a consistent fashion. Tools such as an expression builder will permit the DWA to transform data entering the warehouse from an operational source. As with extractions, SAS/Warehouse Administrator users will be able to write their own transformations and register them in the metadata.

The data summarization component summarizes detail data into day, week, bi-week, month, quarter, and year levels for historical reporting or long term trend analysis. Each site can use its fiscal time instead of wall calendar time to define levels that are meaningful to its business. Closely integrated with metadata, the data summarization component allows users to point and click to choose which summary levels to keep and what statistics to maintain for the data. Non-time-based summarization is also supported. Traditional summarization requires that all data that is to be summarized is available at the time of summarization. To remove that limitation, the data summarization component supports incremental summarization that allows warehouse users to easily access many different kinds of summarized results as data is periodically loaded into the warehouse.

Population is the physical loading of data warehouse tables at the detail level. SAS/Warehouse Administrator will support either full refresh or append of warehouse tables. Warehouse tables may reside on one platform or be distributed across multiple platforms.

The SAS/Warehouse Administrator features can also be used to build "personal" data warehouses as well as one or more corporate data warehouses. To do this, the SAS/Warehouse Administrator supports the concept of environments to allow you to partition your data warehouse activities along department or functional lines. Like warehouse tables, the warehouses themselves may be centralized on one platform or distributed across multiple platforms.

When the data warehouse is ready for test or production use, the DWA will use the Warehouse Administrator features to facilitate the day to day operations of the data warehouse. Specifically, it provides the DWA with user interfaces for:

- ☐ generation of code from metadata definitions,
- ☐ scheduling of generated code as jobs,
- ☐ job log viewer to view SAS logs and output, and
- ☐ exception handling to handle data or processing exceptions.

The code generator is called to convert the previously defined metadata into generated SAS code that will load data into the warehouse from the operational sources. That code may then be scheduled and executed as a job via the Job Scheduler or submitted immediately in the current SAS session.

The Job Scheduler, an independent feature, will be packaged with SAS/Warehouse Administrator. It will enable the scheduling and submission of jobs through a platform independent user interface. The user may use their native host scheduler, such as Unix CRON, or interface with the SAS Domain Server, a component of SAS/CONNECT software that provides scheduling services. A variety of scheduling criteria will be supported, for example, running jobs on certain days of the week or month. A job log viewer will be provided to enable the viewing of scheduled or completed jobs scheduled through the job scheduler. Relevant information for each job, for example, return code, will be displayed. The user will be able to view specific job logs and/or output.

Once you build a Data Warehouse, you certainly want business users to access the warehouse easily. So that a business user can find out what data or information a data warehouse contains, the SAS/Warehouse Administrator product will contain a metadata browser allow the end user to explore the warehouse, looking for subject areas of interest. Once the business user has found subjects of interest, they will most likely wish to query those subjects for further information or to perform data analysis. SAS Macros will be supplied that will allow the end user to easily access the desired warehouse data from their own programs or applications.

In summary, SAS/Warehouse Administrator will give customers a competitive advantage by greatly reducing the time requirements to build and manage a data warehouse.

CONCLUSION

Storing enterprise-wide information in a data warehouse allows organizations to become much more responsive and customer-driven by turning transaction-oriented data into useful information. With its powerful data access, management, analysis, and presentation abilities, the SAS System is the ideal tool for data warehousing. The addition of the SAS/Warehouse Administrator toolset further strengthens the SAS user's ability to build, maintain, and exploit data warehouses.