ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Study of Data Warehouse Architecture

Inderpal Singh
Department of Computer Science and Engineering, DAV Institute of Engineering and Technology, Jalandhar, India
Er.inderpal13@gmail.com

Abstract

Data warehousing is the essential elements of decision support, which has increasingly become a focus of the database industry. Many commercial products and services are now available, and all of the principal database management system vendors now have offerings in these areas. Decision support places some rather different requirements on database technology compared to traditional on-line transaction processing applications. This paper provides an overview of data warehousing with an emphasis on their new requirements and alsodefine back end tools for extracting, cleaning and loading data into a data warehouse, front-end client tools for querying and data analysis and tools for metadata management and for managing the warehouse.

Keywords: Data warehousing, decision support, on-line transaction processing, database and front-end client tools.

Introduction

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process.

Subject-Oriented: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

Integrated: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

Time-Variant: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

Non-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

Overall Architecture

The data warehouse architecture is based on a relational database management system server that functions as the central repository for informational data. Operational data and processing is completely separated from data warehouse processing. This central information repository is surrounded by a number of key components designed to make the entire environment functional, manageable and accessible by both the operational systems that source data into the warehouse and by end-user query and analysis tools.

Typically, the source data for the warehouse is coming from the operational applications. As the data enters the warehouse, it is cleaned up and transformed into an integrated structure and format. The transformation process may involve conversion, summarization, filtering and condensation of data. Because the data contains a historical component, the warehouse must be capable of holding and managing large volumes of data as well as different data structures for the same database over time.

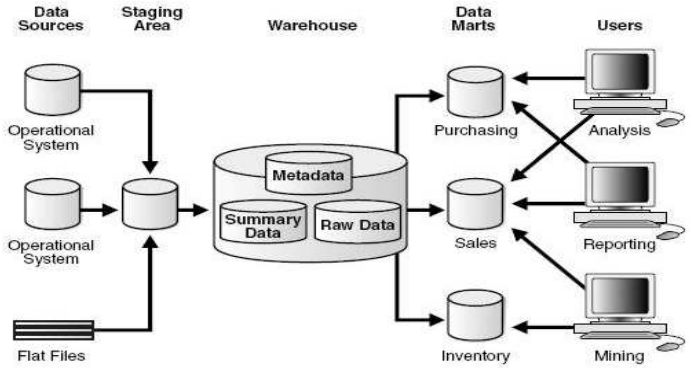

Fig 1. Architecture of data warehouse

ISSN: 2277-9655
Impact Factor: 1.852

Major Components Of Data Warehousing

a. Data Warehouse Database

The central data warehouse database is the cornerstone of the data-warehousing environment. This database is almost always implemented on the relational database management system (RDBMS) technology. However, this kind of implementation is often constrained by the fact that traditional RDBMS products are optimized for transactional database processing. Certain data warehouse attributes, such as very large database size, ad hoc query processing and the need for flexible user view creation including aggregates, multi-table joins and drill-downs, have become drivers for different technological approaches to the data warehouse database. These approaches include:

- Parallel relational database designs for scalability that include shared-memory, shared disk, or shared-nothing models implemented on various multiprocessor configurations (symmetric multiprocessors or SMP, massively parallel processors or MPP, and/or clusters of uni- or multiprocessors).
- An innovative approach to speed up a traditional RDBMS by using new index structures to bypass relational table scans.
- Multidimensional databases (MDDBs) that are based on proprietary database technology; conversely, a dimensional data model can be implemented using a familiar RDBMS. Multi-dimensional databases are designed to overcome any limitations placed on the warehouse by the nature of the relational data model. MDDBs enable on-line analytical processing (OLAP) tools that architecturally belong to a group of data warehousing components jointly categorized as the data query, reporting, analysis and mining tools. Sourcing, Acquisition, Cleanup and Transformation Tools

A significant portion of the implementation effort is spent extracting data from operational systems and putting it in a format suitable for informational applications that run off the data warehouse.

The data sourcing, cleanup, transformation and migration tools perform all of the conversions, summarizations, key changes, structural changes and condensations needed to transform disparate data into information that can be used by the decision support tool. They produce the programs and control statements, including the COBOL programs, MVS job-control language (JCL), UNIX scripts, and SQL data definition language (DDL) needed to move data into the data warehouse for multiple operational systems. These tools also maintain the meta data. The functionality includes:

- Removing unwanted data from operational databases
- Converting to common data names and definitions
- Establishing defaults for missing data
- Accommodating source data definition changes

The data sourcing, cleanup, extract, transformation and migration tools have to deal with some significant issues including:

- Database heterogeneity. DBMSs are very different in data models, data access language, data navigation, operations, concurrency, integrity, recovery etc.
- Data heterogeneity. This is the difference in the way data is defined and used in different models - homonyms, synonyms, unit compatibility (U.S. vs metric), different attributes for the same entity and different ways of modeling the same fact.

These tools can save a considerable amount of time and effort. However, significant shortcomings do exist. For example, many available tools are generally useful for simpler data extracts. Frequently, customized extract routines need to be developed for the more complicated data extraction procedures.

b. Meta Data

Meta data is data about data that describes the data warehouse. It is used for building, maintaining, managing and using the data warehouse. Meta data can be classified into:

- Technical meta data, which contains information about warehouse data for use by warehouse designers and administrators when carrying out warehouse development and management tasks.
- Business meta data, which contains information that gives users an easy-to-understand perspective of the information stored in the data warehouse.

Equally important, meta data provides interactive access to users to help understand content and find data. One of the issues dealing with meta data relates to the fact that many data extraction tool capabilities to gather meta data remain fairly immature. Therefore, there is often the need to create a meta data interface for users, which may involve some duplication of effort.

Meta data management is provided via a meta data repository and accompanying software. Meta data repository management software, which typically runs on a workstation, can be used to map the source data to the target database; generate code for data transformations; integrate and transform the data; and control moving data

ISSN: 2277-9655
Impact Factor: 1.852

to the warehouse.

As user's interactions with the data warehouse increase, their approaches to reviewing the results of their requests for information can be expected to evolve from relatively simple manual analysis for trends and exceptions to agent-driven initiation of the analysis based on user-defined thresholds. The definition of these thresholds, configuration parameters for the software agents using them, and the information directory indicating where the appropriate sources for the information can be found are all stored in the meta data repository as well.

c. Access Tools

The principal purpose of data warehousing is to provide information to business users for strategic decision-making. These users interact with the data warehouse using front-end tools. Many of these tools require an information specialist, although many end users develop expertise in the tools. Tools fall into four main categories: query and reporting tools, application development tools, online analytical processing tools, and data mining tools.

d. Query and Reporting Tools

Query and reporting tools can be divided into two groups: reporting tools and managed query tools. Reporting tools can be further divided into production reporting tools and report writers. Production reporting tools let companies generate regular operational reports or support high-volume batch jobs such as calculating and printing paychecks. Report writers, on the other hand, are inexpensive desktop tools designed for end-users.

Managed query tools shield end users from the complexities of SQL and database structures by inserting a metalayer between users and the database. These tools are designed for easy-to-use, point-and-click operations that either accept SQL or generate SQL database queries. Often, the analytical needs of the data warehouse user community exceed the built-in capabilities of query and reporting tools. In these cases, organizations will often rely on the tried-and-true approach of in-house application development using graphical development environments such as PowerBuilder, Visual Basic and Forte. These application development platforms integrate well with popular OLAP tools and access all major database systems including Oracle, Sybase, and Informix.

e. OLAP Tools

OLAP tools are based on the concepts of dimensional data models and corresponding databases, and allow users to analyze the data using elaborate, multidimensional views. Typical business applications include product performance and profitability, effectiveness of a sales program or marketing campaign, sales forecasting and capacity planning. These tools assume that the data is organized in a multidimensional model.

A critical success factor for any business today is the ability to use information effectively. Data mining is the process of discovering meaningful new correlations, patterns and trends by digging into large amounts of data stored in the warehouse using artificial intelligence, statistical and mathematical techniques.

f. Data Marts

The concept of a data mart is causing a lot of excitement and attracts much attention in the data warehouse industry. Mostly, data marts are presented as an alternative to a data warehouse that takes significantly less time and money to build. However, the term data mart means different things to different people. A rigorous definition of this term is a data store that is subsidiary to a data warehouse of integrated data. The data mart is directed at a partition of data (often called a subject area) that is created for the use of a dedicated group of users. A data mart might, in fact, be a set of denormalized, summarized, or aggregated data. Sometimes, such a set could be placed on the data warehouse rather than a physically separate store of data. In most instances, however, the data mart is a physically separate store of data and is resident on separate database server, often a local area network serving a dedicated user group. Sometimes the data mart simply comprises relational OLAP technology, which creates highly denormalized dimensional model (e.g., star schema) implemented on a relational database. The resulting hypercubes of data are used for analysis by groups of users with a common interest in a limited portion of the database.

These types of data marts, called dependent data marts because their data is sourced from the data warehouse, have a high value because no matter how they are deployed and how many different enabling technologies are used, different users are all accessing the information views derived from the single integrated version of the data.

Unfortunately, the misleading statements about the simplicity and low cost of data marts sometimes result in organizations or vendors incorrectly positioning them as an alternative to the data warehouse. This viewpoint defines independent data marts that in fact, represent fragmented point solutions to a range of business problems in the enterprise. This type of implementation should be rarely deployed in the context of an overall technology or applications architecture. Indeed, it is missing the ingredient that is at the heart of the data warehousing concept -- that of data integration. Each independent data mart makes its own assumptions about how to consolidate the data, and the data across several data marts may not be consistent.

ISSN: 2277-9655
Impact Factor: 1.852

Moreover, the concept of an independent data mart is dangerous -- as soon as the first data mart is created, other organizations, groups, and subject areas within the enterprise embark on the task of building their own data marts. As a result, you create an environment where multiple operational systems feed multiple non-integrated data marts that are often overlapping in data content, job scheduling, connectivity and management. In other words, you have transformed a complex many-to-one problem of building a data warehouse from operational and external data sources to a many-to-many sourcing and management nightmare.

g. Data Warehouse Administration and Management

Data warehouses tend to be as much as 4 times as large as related operational databases, reaching terabytes in size depending on how much history needs to be saved. They are not synchronized in real time to the associated operational data but are updated as often as once a day if the application requires it.

In addition, almost all data warehouse products include gateways to transparently access multiple enterprise data sources without having to rewrite applications to interpret and utilize the data. Furthermore, in a heterogeneous data warehouse environment, the various databases reside on disparate systems, thus requiring inter-networking tools. The need to manage this environment is obvious.

Managing data warehouses includes security and priority management; monitoring updates from the multiple sources; data quality checks; managing and updating meta data; auditing and reporting data warehouse usage and status; purging data; replicating, subsetting and distributing data; backup and recovery and data warehouse storage management.

Information Delivery System

The information delivery component is used to enable the process of subscribing for data warehouse information and having it delivered to one or more destinations according to some user-specified scheduling algorithm. In other words, the information delivery system distributes warehouse-stored data and other information objects to other data warehouses and end-user products such as spreadsheets and local databases. Delivery of information may be based on time of day or on the completion of an external event. The rationale for the delivery systems component is based on the fact that once the data warehouse is installed and operational, its users don't have to be aware of its location and maintenance. All they need is the report or an analytical view of data at a specific point in time. With the proliferation of the Internet and the World Wide Web such a delivery system may leverage the convenience of the Internet by delivering warehouse-enabled information to thousands of end-users via the ubiquitous worldwide network.

In fact, the Web is changing the data warehousing landscape since at the very high level the goals of both the Web and data warehousing are the same: easy access to information. The value of data warehousing is maximized when the right information gets into the hands of those individuals who need it, where they need it and they need it most. However, many corporations have struggled with complex client/server systems to give end users the access they need. The issues become even more difficult to resolve when the users are physically remote from the data warehouse location. The Web removes a lot of these issues by giving users universal and relatively inexpensive access to data. Couple this access with the ability to deliver required information on demand and the result is a web-enabled information delivery system that allows users dispersed across continents to perform a sophisticated business-critical analysis and to engage in collective decision-making.

Back End Tools and Utilities

Data warehousing systems use a variety of data extraction and cleaning tools, and load and refresh utilities for populating warehouses.

a. Data Extraction

Data extraction from “foreign” sources is usually implemented via gateways and standard interfaces (such as Information Builders EDA/SQL, ODBC, Oracle Open Connect, Sybase Enterprise Connect, Informix Enterprise Gateway).

b. **Data Cleaning**

Since a data warehouse is used for decision-making, it is important that the data in the warehouse be correct. However, since large volumes of data from multiple sources are involved, there is a high probability of errors and anomalies in the data.. Therefore, tools that help to detect data anomalies and correct them can have a high payoff.

Some examples where data cleaning becomes necessary are: inconsistent field lengths, inconsistent descriptions, inconsistent value assignments, missing entries and violation of integrity constraints. Not surprisingly, optional fields in data entry forms are significant sources of inconsistent data.

There are three related, but somewhat different, classes of data cleaning tools. Data migration tools allow simple transformation rules to be specified; e.g., “replace the string gender by sex”. Warehouse Manager from Prism is an example of a popular tool of this kind. Data scrubbing tools use domain-specific knowledge (e.g.,

ISSN: 2277-9655
Impact Factor: 1.852

postal addresses) to do the scrubbing of data. They often exploit parsing and fuzzy matching techniques to accomplish cleaning from multiple sources. Some tools make it possible to specify the "relative cleanliness" of sources. Tools such as Integrity and Trillum fall in this category. Data auditing tools make it possible to discover rules and relationships (or to signal violation of stated rules) by scanning data. Thus, such tools may be considered variants of data mining tools. For example, such a tool may discover a suspicious pattern (based on statistical analysis) that a certain car dealer has never received any complaints.

c. **Load**

After extracting, cleaning and transforming, data must be loaded into the warehouse. Additional preprocessing may still be required: checking integrity constraints; sorting; summarization, aggregation and other computation to build the derived tables stored in the warehouse; building indices and other access paths; and partitioning to multiple target storage areas. Typically, batch load utilities are used for this purpose. In addition to populating the warehouse, a load utility must allow the system administrator to monitor status, to cancel, suspend and resume a load, and to restart after failure with no loss of data integrity.

The load utilities for data warehouses have to deal with much larger data volumes than for operational databases. There is only a small time window (usually at night) when the warehouse can be taken offline to refresh it. Sequential loads can take a very long time, e.g., loading a terabyte of data can take weeks and months! Hence, pipelined and partitioned parallelism are typically exploited [6]. Doing a full load has the advantage that it can be treated as a long batch transaction that builds up a new database. While it is in progress, the current database can still support queries; when the load transaction commits, the current database is replaced with the new one. Using periodic checkpoints ensures that if a failure occurs during the load, the process can restart from the last checkpoint.

However, even using parallelism, a full load may still take too long. Most commercial utilities (e.g., RedBrick Table Management Utility) use incremental loading during refresh to reduce the volume of data that has to be incorporated into the warehouse. Only the updated tuples are inserted. However, the load process now is harder to manage. The incremental load conflicts with ongoing queries, so it is treated as a sequence of shorter transactions (which commit periodically, e.g., after every 1000 records or every few seconds), but now this sequence of transactions has to becoordinated to ensure consistency of derived data and indices with the base data.

d. **Refresh**

Refreshing a warehouse consists in propagating updates on source data to correspondingly update the base data and derived data stored in the warehouse. There are two sets of issues to consider: when to refresh, and how to refresh. Usually, the warehouse is refreshed periodically (e.g., daily or weekly). Only if some OLAP queries need current data (e.g., up to the minute stock quotes), is it necessary to propagate every update. The refresh policy is set by the warehouse administrator, depending on user needs and traffic, and may be different for different sources.

Refresh techniques may also depend on the characteristics of the source and the capabilities of the database servers. Extracting an entire source file or database is usually too expensive, but may be the only choice for legacy data sources. Most contemporary database systems provide replication servers that support incremental techniques for propagating updates from a primary database to one or more replicas. Such replication servers can be used to incrementally refresh a warehouse when the sources change. There are two basic replication techniques: data shipping and transaction shipping.

In data shipping (e.g., used in the Oracle Replication Server, Praxis OmniReplicator), a table in the warehouse is treated as a remote snapshot of a table in the source database. After rowtriggers are used to update a snapshot log table whenever the source table changes; and an automatic refresh schedule (or a manual refresh procedure) is then set up to propagate the updated data to the remote snapshot.

In transaction shipping (e.g., used in the Sybase Replication Server and Microsoft SQL Server), the regular transaction log is used, instead of triggers and a special snapshot log table. At the source site, the transaction log is sniffed to detect updates on replicated tables, and those log records are transferred to a replication server, which packages up the corresponding transactions to update the replicas. Transaction shipping has the advantage that it does not require triggers, which can increase the workload on the operational source databases. However, it cannot always be used easily across DBMSs from different vendors, because there are no standard APIs for accessing the transaction log.

Such replication servers have been used for refreshing data warehouses. However, the refresh cycles have to be properly chosen so that the volume of data does not overwhelm the incremental load utility.

The Three Major Advantages are

1. Integrating data from multiple sources;
2. Performing new types of analyses; and

ISSN: 2277-9655
Impact Factor: 1.852

3. Reducing cost to access historical data.

Other benefits may include:
1. Standardizing data across the organization, a "single version of the truth";
2. Improving turnaround time for analysis and reporting;
3. Sharing data and allowing others to easily access data;
4. Supporting ad hoc reporting and inquiry;
5. Reducing the development burden on IS/IT; and
6. Removing informational processing load from transaction-oriented databases.

The Three Major Disadvantages Are

The major disadvantage is that a data warehouse can be costly to maintain and that becomes a problem if the warehouse is underutilized. It seems that managers have unrealistic expectations about what they will get from having a data warehouse.

Conclusion

This paper provides an overview of data warehousing with an emphasis on their new requirements and also define back end tools for extracting, cleaning and loading data into a data warehouse, front-end client tools for querying and data analysis and tools for metadata management and for managing the warehouse.

Reference

[1] http://www.1keydata.com
[2] Devlin, B.A., and P.T. Murphy, "An architecture for a business and information system," IBM Systems Journal, Vol. 27, No 1. 1988.
[3] Power, D., "What are the advantages and disadvantages of Data Warehouses?" DSS News, Vol. 1, No. 7, July 31, 2000.
[4] Zhuge, Y., H. Garcia-Molina, J. Hammer, J. Widom, “View Maintenance in a Warehousing Environment, Proc. of SIGMOD Conf., 1995.
[5] Roussopoulos, N., et al., “The Maryland ADMS Project: Views R Us.” Data Eng. Bulletin, Vol. 18, No.2, June 1995.