An Analysis and Study on Data Warehouse Server

[1]Nirav S. Shukla, [2]Parag B.Makwana, [3]Riddhi K. Raval, [4]Shraddha R.Mandaviya

[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor,
[1]Shri Chimanbhai Patel Institute of Computer Applications,
Ahmedabad, India.
[2]Shree Swaminarayan College of Computer Science,
Bhavnagar, India
[3]Shree Swaminarayan College of Computer Science,
Bhavnagar, India
[4]Shree Swaminarayan College of Computer Science,
Bhavnagar, India

Abstract: Data warehouses and on-line analytical processing (OLAP) tools have become essential elements of decision support systems. Traditionally, data warehouses are refreshed periodically (for example, nightly) by extracting, transforming, cleaning and consolidating data from several operational data sources. The data in the warehouse is then used to periodically generate reports, or to rebuild multidimensional (data cube) views of the data for on-line querying and analysis

IndexTerms: data warehouse, meta data, data extraction, cleaning, data mart, dbms, data transformation.

I. INTRODUCTION

1.1. Data warehouse Design and Construction.

There are five step are discuss in below

- How to design and construct data warehouse
- Three tier data warehouse architecture
- Back –end tools and utitlies of data warehouse.
- The metadata repository
- Types of warehouse server for OLAP Processing.

2. Design and construction of data warehouses

A business analysis framework for data warehouse design. The basic steps involved in the design process are also described. The Design of a Data Warehouse: A Business Analysis Framework "What can business analysts gain from having a data warehouse?" First, having a data warehouse may provide a competitive advantage by presenting relevant information from which to measure performance and make critical adjustments in order to help win over competitors.
Second, a data warehouse can enhance business productivity because it is able to quickly and efficiently gather information that accurately describes the organization. Third, a data warehouse facilitates customer relationship management because it provides a consistent view of customers and items across all lines of business, all departments, and all markets. Finally, a data warehouse may bring about cost reduction by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner. To design an effective data warehouse we need to understand and analyze business needs and construct a business analysis framework. The construction of a large and complex information system scan be viewed as the construction of a large and complex building, for which the owner, architect, and builder have different views. These views are combined to form a complex framework that represents the top-down, business-driven, or owner's perspective, as well as the bottom-up, builder-driven, or implementer's view Four different views regarding the design of a data warehouse must be considered: the top-down view, the data source view, the data warehouse view, and the business query view

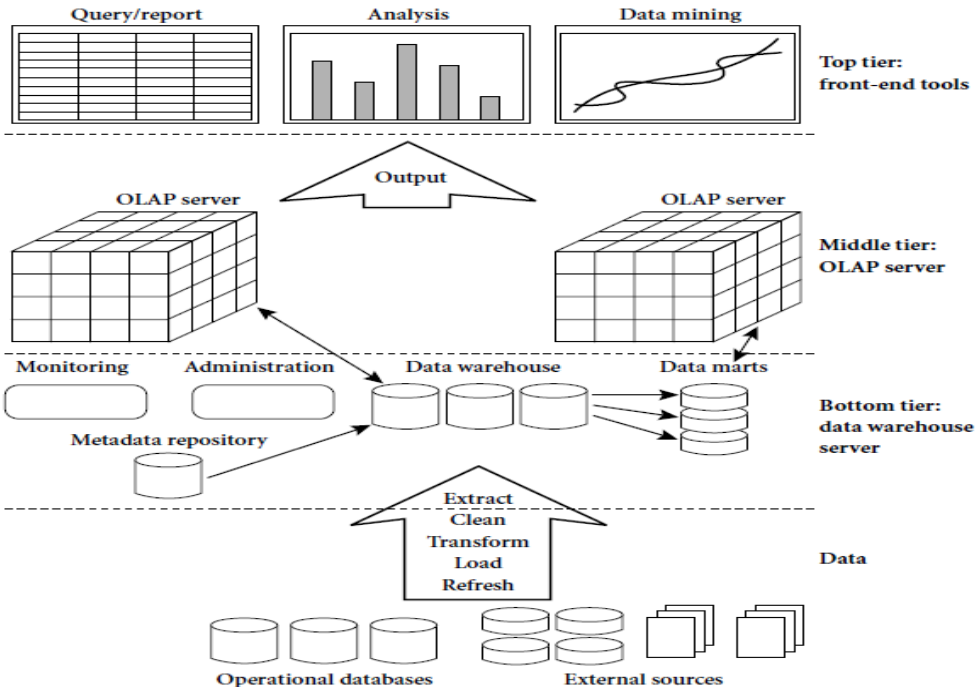
3.The Process of Data Warehouse Design

A data warehouse can be built using a top-down approach, a bottom-up approach, or a combination of both. The top-down approach starts with the overall design and planning. It is useful in cases where the technology is mature and well known, and where the business problems that must be solved are clear and well understood. The bottom-up approach starts with experiments and prototypes. This is useful in the early stage of business modeling and technology development It allows an organization to move forward at considerably less expense and to evaluate the benefits of the technology before making significant commitments. In the combined approach, an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach. From the software engineering point of view,

the design and construction of data warehouse may consist of the following steps: planning, requirement study, problem analysis, warehouse design, data integration and testing, and finally deployment of the data warehouse .Large software systems can be developed using two methodologies: the waterfall method or the spiral method.[1] The waterfall method performs a structured and systematic analysis at each step before proceeding to the next, which is like a waterfall, falling from one step to the next.[2] The spiral method involves the rapid generation of increasingly functional systems, with short intervals between successive releases. This is considered a good choice for data warehouse development, especially for data marts, because the turnaround time is short, modifications can be done quickly, and new designs and technologies can be adapted in a timely manner. In general, the warehouse design process consists of the following steps: Choose a business process to model, for example, orders, invoices, shipments, inventory, account administration, sales, or the general ledger. If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed. However, if the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen. Choose the grain of the business process. The grain is the fundamental, atomic level of data to be represented in the fact table for this process, for example, individual transactions, individual daily snapshots, and so on. Choose the dimensions that will apply to each fact table record. Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status. Choose the measures that will populate each fact table record. Typical measures are numeric additive quantities like dollars sold and units sold. data warehouse construction is a difficult and long-term task, its implementation scope should be clearly defined. The goals of an initial data warehouse implementation should be specific, achievable, and measurable. This involves determining the time and budget allocations, the subset of the organization that is to be modeled, the number of data sources selected, and the number and types of departments to be served. Once a data warehouse is designed and constructed, the initial deployment of the warehouse includes initial installation, roll-out planning, training, and orientation. Platform upgrades and maintenance must also be considered. Data warehouse administration includes data refreshment, data source synchronization, planning for disaster recovery, managing access control and security, managing data growth, managing database performance, and data warehouse enhancement and extension. Scope management includes controlling the number and range of queries, dimensions, and reports; limiting the size of the data warehouse; or limiting the schedule, budget, or resources.

4. Three-Tier Data Warehouse Architecture

Data warehouses often adopt three-tier architecture. The bottom tier is a warehouse database server that is almost always a relational database system. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources These tools and utilities perform data extraction, cleaning, and transformation (e.g., to merge similar data from different



sources into a unified format), as well as load and refresh functions to update the data warehouse The data are extracted using application program interfaces known as gateways. A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server. Examples of gateways include ODBC (Open Database Connection) and OLEDB (Open Linking and Embedding for Databases) by Microsoft and JDBC (Java Database Connection). This tier also contains a metadata repository, which stores information about the data warehouse and its contents. The middle tier is an OLAP server that is typically implemented using either (1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or (2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations. The top tier is a front-end client layer,

which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on). From the architecture point of view, there are three data warehouse models: the enterprise warehouse, the data mart, and the virtual warehouse.

5. Data Warehouse Back-End Tools and Utilities

Data warehouse systems use back-end tools and utilities to populate and refresh their data. These tools and utilities include the following functions: Data extraction, which typically gathers data from multiple, heterogeneous, and external sources. Data cleaning, which detects errors in the data and rectifies them when possible Data transformation, which converts data from legacy or host format to warehouse format Load, which sorts, summarizes, consolidates, computes views, checks integrity, and builds indices and partitions Refresh, which propagates the updates from the data sources to the warehouse Besides cleaning, loading, refreshing, and metadata definition tools, data warehouse systems usually provide a good set of data warehouse management tools. Data cleaning and data transformation are important steps in improving the quality of the data and subsequently, of the data mining results.

6. Metadata Repository

Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects. a metadata repository within the bottom tier of the data warehousing architecture. Metadata are created for the data names and definitions of the given warehouse. Additional metadata are created and captured for time stamping any extracted data, the source of the extracted data, and missing fields that have been added by data cleaning or integration processes. A description of the structure of the data warehouse, which includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents Operational metadata, which include data lineage (history of migrated data and the sequence of transformations applied to it), currency of data (active, archived, purged), and monitoring information (warehouse usage statistics, error reports, and audit trails).The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports. The mapping from the operational environment to the data warehouse, which includes source databases and their contents, gateway descriptions, data partitions ,data extraction, cleaning, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control). Data related to system performance, which include indices and profiles that improve data access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles. Business metadata, which include business terms and definitions, data ownership information, and charging policies. A data warehouse contains different levels of summarization, of which metadata is one type. Other types include current detailed data (which are almost always on disk), older detailed data (which are usually on tertiary storage), lightly summarized data and highly summarized data (which may or may not be physically housed).

Metadata play a very different role than other data warehouse data and are important for many reasons. For example, metadata are used as a directory to help the decision support system analyst locate the contents of the data warehouse, as a guide to the mapping of data when the data are transformed from the operational environment to the data warehouse environment, and as a guide to the algorithms used for summarization between the current detailed data and the lightly summarized data, and between the lightly summarized data and the highly summarized data. Metadata should be stored and managed persistently (i.e., on disk).

7 Types of OLAP Servers: ROLAP versus MOLAP

OLAP servers present business users with multidimensional data from data warehouses or data marts, without concerns regarding how or where the data are stored. However, the physical architecture and implementation of OLAP servers must consider data storage issues. Implementations of a warehouse server for OLAP processing include the following:

1.7.1 Relational OLAP (ROLAP) servers: These are the intermediate servers that stand in between a relational back-end server and client front-end tools. They use a relational or extended-relational DBMS to store and manage warehouse data, and OLAP middleware to support missing pieces. ROLAP servers include optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services. ROLAP technology tends to have greater scalability than MOLAP technology. The DSS server of Microstrategy, for example, adopts the ROLAP approach.

1.7.2 Multidimensional OLAP (MOLAP) servers: These servers support multidimensional views of data through array-based multidimensional storage engines. They map multidimensional views directly to data cube array structures. The advantage of using a data cube is that it allows fast indexing to pre computed summarized data. Notice that with multidimensional data stores, the storage utilization may be low if the data set is sparse. In such cases, sparse matrix compression techniques should be explored. Many MOLAP servers adopt a two-level storage representation to handle dense and sparse data sets: denser sub cubes are identified and stored as array structures, whereas sparse sub cubes employ compression technology for efficient storage utilization.

1.7.3 Hybrid OLAP (HOLAP) servers: The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP. For example, a HOLAP server may allow large volumes of detail data to be stored in a relational database, while aggregations are kept in a separate MOLAP store. The Microsoft SQL Server 2000 supports a hybrid OLAP server.

*1.7.4 Specialized SQL server*s: To meet the growing demand of OLAP processing in relational databases, some database system vendors implement specialized SQL servers that provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment

References

[1]Paulraj Ponniah.. Data Warehousing Fundamentals.
[2] Ralph kimbal . The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling