Musewerx
SERVICE AT YOUR PACE

WHITE PAPER

Data Warehousing

Middleware, Analytics and Reporting

This white paper focuses on Data Warehousing Middleware, Analytics and Reporting.

Introduction to Data Warehouse

A Data Warehouse is a relational database that is designed for query and analysis rather than for transaction processing. It usually contains historical data derived from transaction data, but it can include data from other sources. It separates analysis workload from transaction workload and enables an organization to consolidate data from several sources.

In addition to a relational database, a Data Warehouse environment includes an extraction, transportation, transformation, and loading (ETL) solution, an online analytical processing (OLAP) engine, client analysis tools, and other applications that manage the process of gathering data and delivering it to business users.

Characteristic of Data Warehouse

Subject Oriented

Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a warehouse that concentrates on sales. Using this warehouse, you can answer questions like "Who was our best customer for this item last year?" This ability to define a data warehouse by subject matter, sales in this case makes the Data Warehouse subject oriented.

Integrated

Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated.

Non-Volatile

Nonvolatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable you to analyze what has occurred.

Time Variant

In order to discover trends in business, analysts need large amounts of data. This is very much in contrast to online transaction processing (OLTP) systems, where performance requirements demand that historical data be moved to an archive. A Data Warehouse's focus on change over time is what is meant by the term time variant.

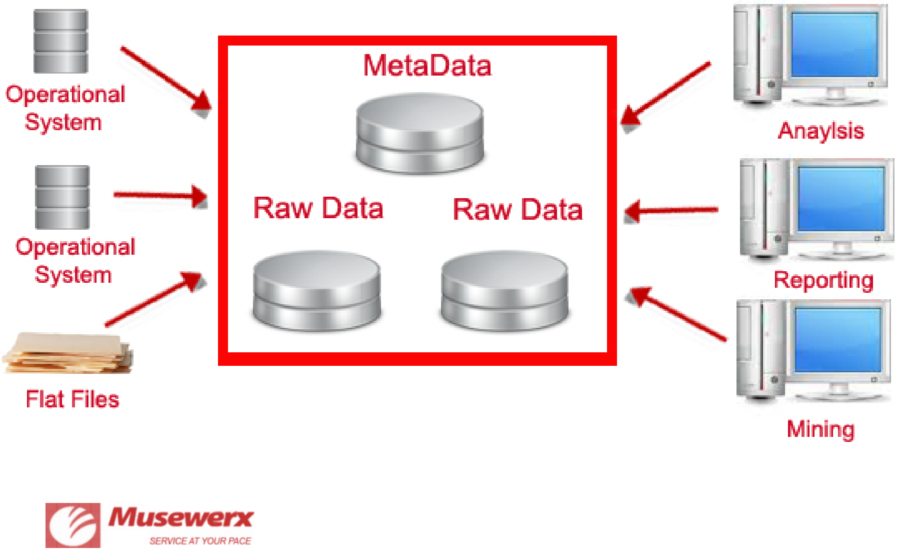

Figure 1- ***Data Warehouse Architecture***

Development in Data Warehousing

Evolving business requirements—for predictive analytics, operational business intelligence, faster reaction to events and deeper analysis of options—result in demanding technical requirements. Some of the key requirements for the newest generation of Data Warehouses are:

- **Frequent update of data:** instead of weekly or nightly batch update, today's Data Warehouse often requires online update frequently or continuously; stringent data latency requirements are becoming more common (e.g. data must be online and available for query within x hours, minutes or seconds on receipt); high volume ingest requirements are also becoming more common;
- **Mixed workload:** instead of a steady diet of periodic scheduled reporting or analysis, today's Data Warehouse often must meet multiple concurrent service levels including interactive query; large long running jobs; and, a mix of concurrent query and update;
- **Many concurrent users:** instead of a few users feeding a relatively small complement of management decision makers, data warehouses must now often support large communities of employees, partners or customers concurrently accessing and updating data;

- **High data availability:** as enterprises come to rely on the Data Warehouse to support operational business processes, data availability requirements rise and often attain mission critical status; thus, the Data Warehouse often will need to operate on a continuous or near- continuous schedule;

- **Intensive Analytics:** as business strategies rise in sophistication, and data volumes continue to increase, there is a rising demand to perform more and more analysis of the data—and to accomplish it in place in the data warehouse. Just as the tolerance for down time decreases, because business needs data access all the time, the tolerance for delay in analysis also decreases. Thus, no one wants to wait while data is exported to another system for analysis. Instead, the requirement is increasingly to analyze the data in place in the Data Warehouse.

In addition, Data Warehouses continue to grow in every direction at a rapid pace: most successful Data Warehouses today increase rapidly in both data volume and usage levels. Thus, even in modest initiatives in smaller enterprises and departments, the data warehouse often grows to present quite stringent requirements in order to deliver and sustain business value over a period of just a few years. As a further implication of the continual rapid increase in scale, solutions must be even more cost effective; deployment and expansion must be rapid and easily accomplished; Data Warehouse management must be more automated; and, skill requirements per unit of data must continually decrease.

Middleware

Middleware is software that lets systems talk to one another while hiding the complexities of network connectivity. Middleware is an important Data Warehouse component since it's the means by which applications communicate with the Data Warehouse. Middleware technology lets clients talk to servers, but more critically, it shields the application programmer from the complexity of retrieving and combining data.

In a data warehousing environment, the Middleware services are the set of programs and routines that do the following:

- Pull data from the source (or sources).
- Make sure that the data is correct.
- Move the data around the environment from platform to platform.
- Handle any necessary data transformations.
- Load the data into the Data Warehouse's database (or databases).

Middleware Services

In a more formal sense, the items in the preceding list are handled by these Middleware services:

- Data selection and extraction
- Data quality assurance
- Data movement,
- Data mapping and transformation
- Data quality assurance

- Data movement,
- Data loading (into the Data Warehouse)

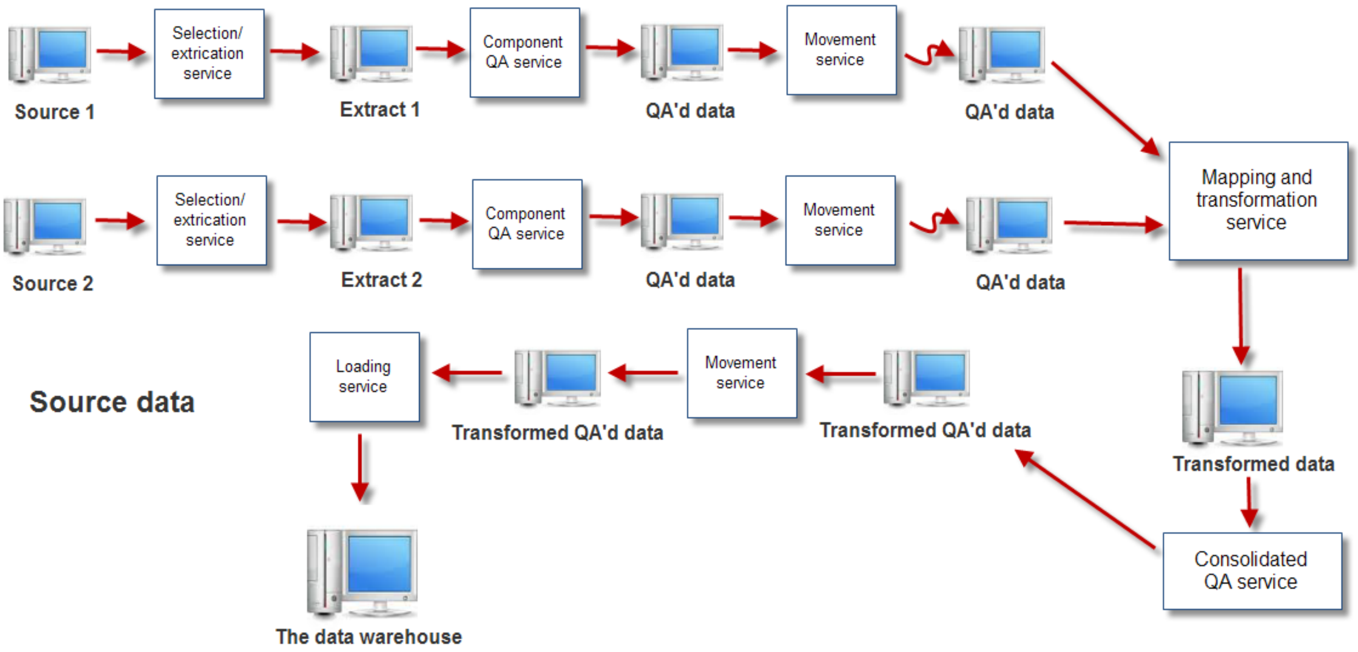

Figure 2: ***The Data Warehousing Middleware services flow together, from end to end, from data sources to the Data Warehouse.***

Specialty Middleware Services

For more sophisticated and/or simplified Data Warehousing needs, you can deploy a set of specialty Middleware services. These services include:

- Replication and change data capture.
- Virtual, or federated, data access (also known as Enterprise Information Integration, or EII).

Data Warehouse Analytics and Reporting

Analytics

Analytics is a term that refers to the various modes of using information to make decisions. Traditionally we used to call this decision support, and it was mostly accomplished through static green-bar paper reports and some ad hoc querying. But in the last decade, the decision-support tools (Data Warehouse and Business Intelligence tools) have become more and more sophisticated for data access, Data Analysis, Data Manipulation, Data Mining, Forecasting, Trend Analysis and other metric-based presentations such as Scorecards and Dashboards. Nowadays, they even include packaged analytical applications for specific business domains, such as supply chain analysis, sales channel analysis, performance analysis, etc

Regulatory compliance, increased competition, and other pressures have created an insatiable need for companies to accumulate and analyze large, fast-growing quantities of data such as:

- Telecommunications call detail records (CDRs)
- IT/Network event history
- Financial trade (quote and tick) history
- Web logs & click streams for marketing and fraud analytics
- Compliance and other historical logs

This presents a major market opportunity for enterprise software vendors and software as a service (SaaS) companies. They can profit by creating analytic data management features or entirely new applications that put customers on a faster path to better data-driven decision making.

The New Growth Opportunity for Software Vendors: Data Analytics

Twenty years ago, the business transaction processing software market was in transition. Instead of building and maintaining ERP, CRM and other applications internally using large, expensive IT teams, companies began purchasing commercial off-the-shelf (COTS) applications from new vendors like SAP, JD Edwards and PeopleSoft. A multi-billion dollar software business was born.

Ten years later, in the early 2000s, the trend was further transformed by Salesforce.com and others who innovated the Software as a Service (SaaS) model and enabled companies to implement new applications without traditional data center costs or time-consuming cap-ex budget approval processes.

The major market opportunity today for enterprise software vendors and SaaS companies is in the analytic application arena. The funding for these initiatives is available (as evidenced by the billions spent annually on custom data warehousing technology and services), and software companies can profit by putting customers on a faster path to better data-driven decision making.

Such solutions enable customers to gain insights for a competitive edge, to reduce risk exposure, and/or increase profitability. Enterprise software and SaaS vendors have benefited greatly from underlying database management systems allowing architects and developers to focus on their core competencies at the application level without worrying about the underlying data management. Historically, these traditional OLTP row-oriented engines could handle the transactional data entry nature of most applications. However, today's analytic applications must continuously and simultaneously load and query against massive volumes of information.

Reporting

Many organizations would like to take advantage of reporting on real-time data from business critical systems but are concerned about the impact to the end user community of those applications. Data entry databases should not be used for reporting, conversely, reporting databases should not be used for data entry. Here's why: A database that is designed to be speedy for data-entry will be slow for reporting and vice versa. This is because they are optimized for speed through incompatible approaches.

In a nutshell, data entry databases are fastest when they are "normalized". Data warehouse databases are fastest when they are "denormalized".

To elaborate, reporting systems need to report data quickly. To do so they allow redundancies in the data. These redundancies save the reports from spending time searching for the data when the report is requested.

In contrast, data-entry systems need to update data quickly. To do so they minimize redundant data by pulling out the redundancies and storing them elsewhere. They link them through an ID. For instance, rather than storing the customer's name and address repeatedly with each order, they make a single entry of the name and address and they assign the customer an ID. They then store just the ID on each of the orders. This allows the system to update a subsequent change of address by modifying or inserting a single entry, rather than having to modify many, many orders from that customer. Storage structures built like this are said to be “normalized”. This decreases data entry time and also decreases the data storage requirements.

Different Types of Reports

The conventional method of allocating reporting tools according to responsibility for different types of reports has proved to be a hindrance time and time again. This section will examine why this is the case and how the problems can be tackled:

- Standard versus Ad-hoc
- Condensed versus Detailed
- Key Figures versus Master Data
- Replication in the Data Warehouse versus Direct Access

If you facilitate a smooth transition from ad-hoc reporting to reusable reports, the construction of a reporting landscape will be considerably easier. If top-down and bottom-up approaches are interpreted as two metaphors for navigation around the same data, it is not necessary to divide up reports into key figure reports and master data reports. A good Data Warehouse also makes real-time reporting possible.

References

- Data Analytics-The New Growth Opportunity For Software Developers
- IBM DB2 97 and Smart Analytics System
- Data Warehouses, Metadata, and Middleware Baseline Techniques ™
- Analytic Architecture-Approaches to Supporting Analytics Users and Workloads