An Overview on Data Warehousing Problems

Pooja Jadon, Assistant Professor
Department of Computer Science and Engineering, Vivekananda Global University, Jaipur
Email Id- pooja.jadon@vgu.ac.in

***Abstract:** The term "Data Warehousing" refers to structures, methods, and tools for combining data from many databases or other information sources into a single repository, referred to as a "data warehouse," that can be queried or analyzed directly. One of the most pressing problems in database research and business is providing integrated access to numerous, dispersed, heterogeneous databases and other information sources. Data warehousing has been a popular term in the database business in recent years, although it has received little attention from the database research community. We motivate the idea of a data warehouse in this article, describe a basic data warehousing architecture, and suggest a number of technical problems emerging from the design that we think would be good exploratory study subjects.*

***KEYWORDS:** Business, Data Warehouse, Information, Problems, Query.*

1. INTRODUCTION

One of the most pressing problems in database research and business is providing integrated access to numerous, dispersed, heterogeneous databases and other information sources. Most methods to the data integration issue in the research community are based on the following fairly basic two-step process:

- Accept a query, identify the suitable collection of information sources to respond to the question, and create sub queries or instructions for each information source.
- Gather information from many sources, apply necessary translation, filtering, and merging, and provide the final response to the user or application (hereafter called the client).

Because information is pulled from the sources only when queries are asked, we refer to this process as a lazy or on-demand method to data integration (Because the module that decomposes queries and aggregates results is frequently referred to as a mediator, this method is also known as a mediated approach). Figure 1 shows the working of data warehousing.

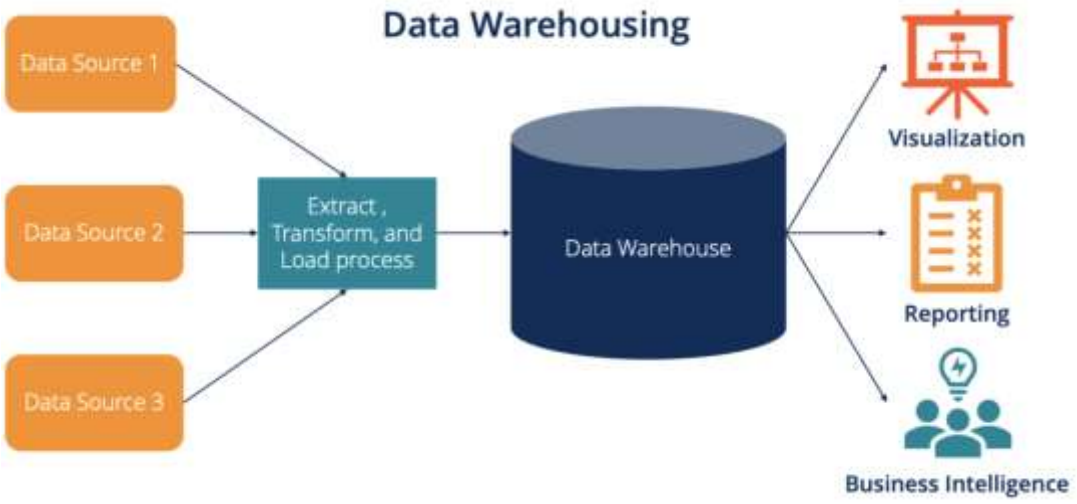

Figure 1: Illustrates the working of data warehousing[1]

An eager or in-advance approach to data integration is a logical alternative to a sluggish method. In a zealous manner:

- Relevant information from each source is retrieved ahead of time, translated and filtered as needed, combined with relevant information from other sources, and stored in a (logically) centralized repository.
- When a query is submitted, it is assessed immediately at the repository, without the need to consult the original data sources.

Because the repository acts as a warehouse for storing the data of interest, this technique is often referred to as data warehousing.

For information that changes often, customers with unexpected requirements, and queries that operate across huge quantities of data from a large number of sources, a lazy approach to integration is suitable. The lazy approach, on the other hand, may result in query processing inefficiency and delay, particularly when queries are issued multiple times, information sources are slow, expensive, or infrequently unavailable, and significant processing is required for the translation, filtering, and merging steps. The sluggish method is simply not possible in situations when information sources do not permit ad-hoc searches[2]–[6].

The combined information is accessible for rapid querying and analysis by customers in the warehousing method. As a result, the storage strategy is suitable for:

- Clients who need particular, predictable parts of the information provided.
- Clients that need high query speed (data is accessible locally at the warehouse), but not necessarily the most up-to-date state of the data.
- Situations in which native applications at the information sources need high performance (large multi-source queries are executed at the warehouse instead).

Clients, who want private copies of material so they may modify, annotate, summarize, and so on, or clients who wish to preserve information that isn't kept at the sources (such as historical information).

The lazy and warehouse methods are both feasible answers to the data integration issue, and each is suitable for different situations. The database research community has mainly concentrated on lazy integration methods. The research issues connected with the warehousing method are discussed in this article.

From a Business Perspective:

Before we go into the research issues surrounding data warehousing, it's worth noting that the database industry has shown a lot of interest in the subject in recent years. Most major suppliers claim to provide at least some data warehousing capabilities, while many small businesses specialize only in data warehousing solutions. Despite significant advancements in commercial data warehousing techniques and solutions, the majority of current systems are still rigid and feature-limited. A genuinely universal, efficient, inflexible, and scalable data warehousing architecture, we think, requires a number of technological advancements, which are outlined below.

The significance of data warehousing in the commercial sector seems to be related to a need for businesses to consolidate all of their data in one location for in-depth analysis, as well as a desire to divorce such analysis from on-line transaction processing systems. One of the main applications of data warehouses is analytical processing that includes highly sophisticated queries (sometimes with aggregates) and few or no updates, which is referred to as decision support. As a result, the terms data warehousing and decision support are often used interchangeably. Given that decision support is often the objective of data warehousing, warehouses may be optimized for decision support, and vice versa. However, since decision support is such a wide topic, we have concentrated this article on research problems related to the warehousing method to integration.

A Data Warehousing System's Architecture:

The information sources are shown at the bottom of the graphic. Although typical disk shapes connote traditional database systems, non-traditional data sources such as at les, news wires, HTM L and SGML documents, knowledge bases, legacy systems, and so on may be included in the general case. A wrap-per/monitor is connected to each information source. The wrapper component of this module is in charge of translating data from the source's native format into the format and data model used by the warehousing system, while the monitor component is in charge of automatically detecting and reporting changes of interest in the source data to the integrator.

The new or updated data is transmitted to the integrator when a new information source is connected to the warehousing system, or when relevant information at a source changes. The integrator is in charge of putting the data in the warehouse, which may involve filtering, summarizing, or combining it with data from other sources. It may be required for the integrator to

acquire more information from the same or other information sources in order to effectively integrate new information into the warehouse.

An off-the-shelf or special-purpose database management system may be used in the data warehouse. Despite the fact that we show a single, centralized warehouse, the warehouse may easily be built as a distributed database system, and data parallelism or dispersion may be required to achieve the needed performance.

The architecture and fundamental functions we've outlined are broader than what most commercial data warehousing systems have to offer. Current systems, in particular, assume that the sources and the warehouse subscribe to a single data model (normally relational), that information propagation from the sources to the warehouse is done in a batch process (possibly off-line), and that queries from the integrator to the information sources are never required[7]–[10].

Problems in Research:

We now detail a variety of research issues that emerge from the warehousing strategy, based on the basic architecture for data warehousing outlined earlier.

1. *Wrapper/Monitors:*

The wrapper/monitor components are responsible for two tasks that are intertwined:

- *Translation:*

Making the underlying information source app seem as though it follows the warehousing system's data model. If the information source is a set of at lies but the warehouse model is relational, for example, the wrapper/monitor must provide an interface that displays the data from the information source as if it were relational. The translation issue exists in virtually all data integration methods, both lazy and eager, and is not exclusive to data warehousing. A translator or wrapper is often a component that converts an information source into a common integrating model.

- *Detecting changes:*

Monitoring the data source for changes that are important to the warehouse and informing the integrator of such changes. This capability is based on translation because, like the data itself, changes to the data must be translated from the information source's format and model into the warehousing system's format and model.

2. *Integrator:*

Assume that the warehouse has been loaded with its initial set of data obtained from the information sources. The continuous duty of the integrator is to receive change alerts from the wrapper/monitors for the information sources and reflect these changes in the data warehouse.

At a suitably abstract level, the data in the warehouse may be viewed as a materialized view (or collection of views), where the basic data exists at the information sources. Viewing the issue in this manner, the role of the integrator is simply to conduct materialized view maintenance. Indeed, there is a strong relationship between the view maintenance issue and data warehousing. However, there are a variety of reasons why traditional view maintenance methods cannot be utilized, and each of these reasons emphasizes a research issue related with data warehousing.

- Data warehouses also tend to include highly aggregated and summarized information. Although in certain instances aggregations may be describable in a standard view definition language, the expressiveness of aggregates and summary operators in such languages are restricted, thus more expressive view definition languages may be required. Furthermore, effective view maintenance in the context of aggregation and summary information app ears to be an ongoing issue.

- The information sources updating the base data typically run independently from the warehouse where the view is kept, and the base data may com e from older systems that are unable or unwilling to engage in view maintenance. Most materialized view maintenance methods depend on the fact that base data up dates are tightly linked to the view maintenance machinery, and view modification happens inside the same transaction as the updates.

- In a data warehouse, the views may not need to be refreshed after every modification or set of modifications to the base data. Rather, large batch up dates to the base data may be considered, in which case efficient view maintenance techniques may involve different algorithms than are used for conventional view maintenance.

3. *Warehouse Specification:*

In the last section we established an analogy between maintenance of a data warehouse and materialized view maintenance. We also indicated that it is useful to provide capabilities for specifying integrators in a high-level fashion, rather than implementing each integrator from scratch. Hence, in a ideal architecture, the contents of the data warehouse are specified as a set of view definitions, from which the warehouse updating tasks performed by the integrator and the change detection tasks required of the wrapper/monitors are deduced automatically.

For traditional view maintenance, methods have been developed to automatically create active data- base rules for updating SQL-defined views. Each rule is "triggered" by the notice of an update that may impact the view and the rule changes the view accordingly. A similar technique may be used to data warehousing if a rule-driven integrator is utilized. Each integrator rule is triggered by a change notice (potentially of a particular kind) from a wrapper/monitor. Similar to the view maintenance rules, integrator rules must update the warehouse to match the base data up dates.

4. *Optimizations:*

In this part we describe three improvements that may enhance the speed of the architecture: filtering irrelevant changes at the sources, storing extra data at the warehouse for "self-maintainability," and effectively handling numerous materialized views.

- Update Filtering
- Self-maintainability
- Multiple View Optimization

5. *Miscellaneous:*

We quickly mention a few additional significant problems that emerge in a data warehousing setting.

- *Warehouse management:* We have concentrated mainly on issue s connected with the "stationary state" of a data warehousing system. However, problems related with warehouse architecture, loading, and metadata management are significant as well. (In fact, it is these issues that have gotten the greatest attention from a significant part of the data storage sector to date.
- *Source and warehouse evolution:* A warehousing design must smoothly manage changes to the information sources: schema modifications, as well as the addition of new information sources and the removal of existing ones. In addition, it is probable that customers may seek schema modifications at the ware- house itself. All of these changes should be handled with as little interruptions or adjustments to other components of the warehousing system as feasible.

DISCUSSION

Data warehousing, often known as corporate data warehousing, is an electronic technique of organizing, analyzing, and reporting data. Data warehousing, for example, enables data mining, which aids companies in identifying data trends that may lead to increased sales and profitability. A data warehouse stores and feeds BI and analytics with current and historical data for the whole company. Data warehouses utilize a database server to extract data from an organization's databases, as well as data modeling, data lifecycle management, data source integration, and other features. SQL Server Data Warehouse is a functionality of SQL Server that is available on-premises. It's a specialized service in Azure that lets you create a data warehouse that can store a lot of data, scale up and down, and is completely managed. Traditional data warehouse solutions such as Azure SQL Data Warehouse are often utilized.

CONCLUSION

In the field of combining numerous, dispersed, heterogeneous information sources, data warehousing is a feasible and in some instances better alternative to conventional research methods. Traditional methods seek, analyze, and integrate information from sources as queries are presented. In the data warehousing method, information is requested, processed, and integrated constantly, so the information is immediately accessible for direct querying and analysis at the warehouse.

Although the idea of data warehousing is already predominant inside the database sector, we believe there are a number of significant open survey problems, defined above, that need to be remedied to start realizing the powerful, and efficient data warehousing processes of the future.

REFERENCES:

[1] "data-warehousing1-1024x505." https://corporatefinanceinstitute.com/resources/knowledge/other/data-warehousing/ (accessed Jul. 15, 2018).

[2] "A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing," *Int. J. Comput. Sci. Issues*, 2010.

[3] A. Wibowo, "Problems and available solutions on the stage of Extract, Transform, and Loading in near real-time data warehousing (a literature study)," 2015, doi: 10.1109/ISITIA.2015.7220004.

[4] J. Widom, "Research problems in data warehousing," 1995, doi: 10.1145/221270.221319.

[5] A. Cuzzocrea, "Data warehousing and OLAP over Big Data: A survey of the state-of-the-art, open problems and future challenges," *Int. J. Bus. Process Integr. Manag.*, 2015, doi: 10.1504/IJBPIM.2015.073665.

[6] N. Tasić, Ž. Đurić, D. Malešević, R. Maksimović, and N. Radaković, "Automation of process performance management in a company," *Teh. Vjesn.*, 2018, doi: 10.17559/TV-20151010074417.

[7] D. Stodder, "Customer analytics in the age of social media," 2012.

[8] M. Pathak, S. Singh, and S. Oberoi, "Impact of Data Warehousing and Data Mining in Decision Making," *Int. J. Comput. Sci. Inf. Technol.*, 2013.

[9] Y. Cui and J. Widom, "Lineage tracing for general data warehouse transformations," *VLDB J.*, 2003, doi: 10.1007/s00778-002-0083-8.

[10] Q. Yuan, "Mega freight generators in my backyard: A longitudinal study of environmental justice in warehousing location," *Land use policy*, 2018, doi: 10.1016/j.landusepol.2018.04.013.