International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

Use of XML as Data Format in Distributed Heterogeneous Database

Yukta Lapsiya[1]

Abstract— *The rate of generation of data is increasing exponentially and requires a tremendous amount of storage devices. It is impossible to store the data in the traditional centralized systems. Distributed Heterogeneous Database provides a solution to store the data in a decentralized manner across various systems. With decentralized systems, the data stored must be syntactically similar. XML, a widely used mark-up language, allows to store the data in a standardized manner with a high flexibility. This paper aims at providing the fundamentals to achieve heterogeneous database integration with XML, particularly looking at previous works and proposing a method*

Key Words: XML, Heterogeneous Database, Distributed Data Storage, Decentralized Systems, Distributed Heterogeneous Database

I. INTRODUCTION

We live in a world where around 2.5 quintillion bytes [4] of data is generated every day. This data has to be stored for further use and there could only be so many centralized systems for the same. Heterogeneous Database provides a solution that is cost-effective and provides an alternative to store large volumes of data on distributed systems. However, a major limitation of heterogeneous databases is the type of data stored. With multiple systems storing data from a variety of sources, it becomes complex to deal with the available data. This can be overcome using XML format for data storage.

"Extensible Markup Language is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable." [5] XML is famous and widely used due to it being simple, general and high usability in various fields. It is easy to store, transform and manipulate the data stored in the form of XML. Due to its benefits, it is used in a wide range of applications and provides flexibility while maintaining a standard format of data. The data stored in the form of XML allows the heterogeneous databases to work in a systematic manner and makes the process of data exchange simple.

The improvements in storage device technologies and its availability at a cheap cost supports the use of distributed heterogeneous databases. XML allows to store the data in a standard format and makes it simple to perform tasks on the same. It can be used to develop various applications using a large database at an efficient cost with high performance. A data warehouse could be built within an enterprise to store historical data that can be further used for analysis purposes and enable the decision-making factor which is very important to take business decisions. All the factors increase the use of heterogeneous databases.

II. RELATED WORKS

A. *Why to choose XML?*

We prefer XML over other data formats for the following 4 reasons: [3]

1. Flexibility with Internet Applications: XML is widely used in many Internet-based applications. More and more applications are being developed that make use of XML Schema. It provides scalability and a standard that can be used across multiple platforms easily.
2. Impact on database and data sharing: XML is a nimble tool and has a great impact on the database. Due to its popularity, many DBMS vendors are providing support services for XML management.
3. Impact on Data warehouse creation: Storing high amounts of data encourages use of data warehouses for analysis purposes. XML can be used to store the data in a concise manner.
4. Expandability and storage capacity: XML comprehends diverse assortment of expression and partition of information. It has a solid capacity of expandability and marvelous information execution capacity.

B. *Comparison of XML with other alternatives*

An un-federal heterogeneous database uses a source database that connects to the other available databases. The transformation of data done by this database must ensure that the transformation meets the requirements of the destination database and the transformation process must be as efficient as possible. Multiple conversion alternatives can be used for this transformation of data. We compare the major ones: Software tools, third-party database, database components and XML with advantages and disadvantages of each of them. The following table provides the comparison:

Transformation methods	Advantage	Disadvantage
Software tools	Low cost, easy to implement, high execution speed	There are restrictions on the source, target database type, versatility is low
Third-party database	Require less conversion modules, good expandability	Manual intervention is needed, implementation is complex
Database components	Low cost, easy to implement	There are strict restrictions on data types, generic is not high
XML	Low cost, easy to implement, high versatility	The need to establish the mapping of XML and databases

Table – 1: Example of XML document [1]

The major disadvantage of XML is the need of a mapping XML with a database which we will discuss further in the paper.

C. *Understanding XML*

The following figure provides an example of an XML document:

```
<book>
  <title>XML</title>
    <chapter>
      <section>
        <head>introduction</head>
          <subsection>data model</subsection>
      </section>
      <section>
        <head>tree pattern</head>
        <subsection>twig</subsection>
      </section>
      <section>
        <head>search</head>
      </section>
      <author>Lu</author>
      <year>2013</year>
    </chapter>
</book>
```

Fig – 1: Example of XML document [2]

XML stores data in the form of XML documents that is a collection of fields. An XML document may look similar to an HTML; however, the prior follows a strict rule of labels (i.e., every opened label must have its corresponding closing label)

D. *Working with XML data*

An XML document unification can be divided into 3 stages [2]:

1. **Export:** The required data is extracted from the database nodes based on rules and logical structure of the application.
2. **Import:**

a. Validation of XML files: The extracted data may not be of an XML type or might be corrupt. To ensure that once is dealing only with XML files, it is necessary to validate that all the files are in the XML format.

b. Assembling data: This process deals with mapping the database table and XML data. It is also responsible for transformation of the data in the required format of the destination database nodes.

c. Saving in database: Once the data is assembled, it forwarded to appropriate data nodes for storage purpose.

3. **Transmitting:** For the data transmission, different policies must be implied to ensure correct transfer of data. Care must be taken to deal with any missing files, broken connections or other possible errors. Furthermore, authentication and encryption might be adopted in case the database system is very large or can be accessed by an unauthorized user

III. METHODOLOGY

A. *Architecture*

The following figure provides a basic architecture of an heterogeneous database based on XML.

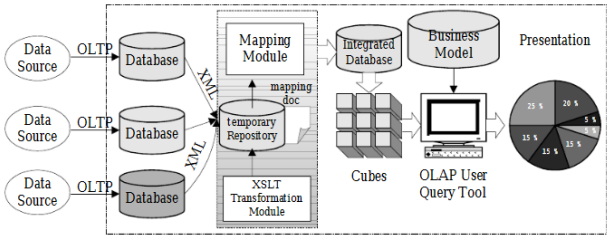

Fig – 2: Heterogeneous Database Architecture [3]

The temporary repository acts as the source database and connects all the databases on the server with the external world. It provides responses to the queries by mapping them to the corresponding database query and collecting all the required data from various database nodes before sending it back to the user.

B. *Mapping query to XML*

1. **Template based mapping:** Mapping involves using an SQL query to refer to XML data. It deals with extraction of the interested data from the XML document. A template-driven mapping is a shallow map that can be used and is shown in the following figure

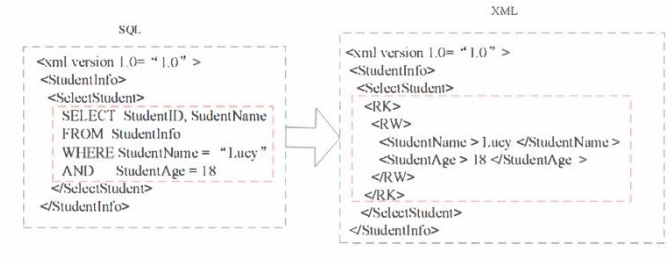

Fig – 3: Template Mapping [2]

However, this type of mapping does not provide any support for mapping of object-oriented databases and works only between RDBMS and XML documents.

2. **Model based mapping:** To overcome the limitation of template-based mapping, this approach suggests to develop a model for transfer between two data formats. It relies on building two-way models consisting of tables and objects mapped with each other. This table is referred for transformation of the data and can provide support to a range of databases: relational database, object-oriented database, XML documents.

C. *Data Mapping between relational database and XML document*

A relational database can be transformed into an XML document using the following set of rules [1]:

1. For each table, create a new element
2. For each column of the table, create a sub-element
3. For each column as primary key, create a new sub-element

The resulting format will be an XML document which can be used to trace back to the relational database if necessary. This is achieved by the following rules [1]:

1. For each element type containing element or mixed content, create a table and a primary key field.
2. For each element type containing mixed content, create a separate table, to store the data without analysis, and through the parent element primary key to link to the parent table.
3. For each single value attribute of this element type and the sub-element which only contains the content not being analyzed and emerges only once, create a field in the table. If the element type or attribute is optional, the field can be empty.
4. For the sub-elements with multi-valued attributes and appear multiple times, to create a separate table to store values, and through the parent element primary key to link to the parent table. For each sub-element with element or mixed content, to connect the parent element table and sub-element table through the primary key of the parent element.

Using these rules, it is possible to create a mapping model between a relational database and XML document.

IV. CONCLUSION

The reliability of data is ever-increasing and there is a need for an alternative to the traditional data storage systems. The various issues in handling the data in traditional systems can be overcome by using Heterogeneous Databases based on XML. A standard format of data improves the quality of the system and overall development of technology. This paper presents available technology that can be used to integrate an XML-based heterogeneous database and provides information on how one can use XML documents instead of relational databases. While the comparison of both systems is out of scope of this paper, with advancements in the field, heterogeneous databases can provide a reliable alternative to the traditional centralized database.

REFERENCES

[1]Yunpeng, L., & Meiyun, X. (2010, September). Research of heterogeneous database integration based on XML. In 2010 International Conference on Mechanical and Electrical Technology (pp. 793-796). IEEE.

[2]Kartikey Borle, Amit Bendarkar, Rajan Bhirud, Aditya Bambole, Prof. Rachna Somkunwar (2017, March). A Survey on Heterogeneous Data Exchange Using XML. IRJET

[3]Tseng, Frank. (2005). XML-Based Heterogeneous Database Integration For Data Warehouse Creation.. 9th Pacific Asia Conference on Information Systems: I.T. and Value Creation, PACIS 2005. 48.

[4]How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read – https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#29cad37860ba

[5]XML – Wikipedia: https://en.wikipedia.org/wiki/XML