

# SHARE MARKET MANAGEMENT SYSTEM BASED KEYWORD QUERY PROCESSING ON XML DATA

Darsana C.S.<sup>1</sup>, Roshni P.<sup>2</sup>, Chandini K.<sup>3</sup> and Surekha Mariam Varghese<sup>4</sup>

Department of Computer Science and Engineering, Mar Athanasius College of  
Engineering, Kothamangalam, Kerala

## **ABSTRACT**

*Nowadays, many research has been focused on XML database systems. In share market, the value of a share changes over time. For managing market values of a share, traditional databases are commonly used. Main such example is relational database. But use of such databases are not efficient since it cannot handle more complex applications and do not have the ability to be scalable. In this paper we implement a Share Market Management System using an XML database. The main function of Share Market Management System ( SMMS ) is to store, retrieve and update the information about share market values. Keyword-based query solution is implemented to obtain the results. Then its performance is evaluated by considering the compilation, parsing and execution metric as total time.*

## **KEYWORDS**

*XML Data; Native XML Database; XQuery; Keyword query;*

## **1. INTRODUCTION**

Stock market is the market in which shares are issued and traded, which is also known as the equity or share market. It is one of the most vital areas of a market economy. The stock market is the important source for companies to raise money. This is where businesses are publicly traded. With the development of online systems, online share marketing has become a trend. These sites provide current share market value details.

XML is a standard for storing and exchanging information over Internet. XML documents contain both the data and the informative relationship structuring of that data in a way that both machines and people can read. All its information can be send from one party to other. So they are self describing. The proposed system use XML database for storing share market information as documents. The information include equity name, open and closing value, rating and offer price. The data can be queried, transformed and exported. XML is a good method to handle sparse data.

In traditional share-market applications, details of equities are usually kept in relational database. Even-though they provides flat storage and retrieval, they possess many problems when dealing with redundant and sparse. The details include equity name, open, high and low values, number of shares traded, total turnover. Sometimes some features may be absent. In a relational database

store, values are also stored redundantly. This can cause null values in database and a lot of space wastage. To overcome these problems, we use an XML database system named BaseX. BaseX is a native and light-weight XML database management system which is specialized in storing, visualizing and querying large XML documents.

A brief description of XML database is presented in section 2. Section 3 explains BaseX, a tool to process XML documents. In section 4 the proposed method is explained. Performance analysis of this method is evaluated in Section 5. The paper is concluded in section 6

## 2. XML DATABASE

An XML database is a data storage software system. It stores data in eXtendedMarkup Language format. These data can then be queried, transformed, exported and returned to a caller. XML databases are document-oriented databases which falls in the category of NoSQL database. A NoSQL database provides a mechanism for storage and retrieval of data in a form other than tabular representation. XML - enabled and Native XML are the main two type of XML database. Attractive feature of XML is it's simplicity. Information coded in XML is easy to read and understand. Computers process it easily. XML is in W3C standard and is extensible, because there is no fixed set of tags. Based on our requirement we can create and use new tags. XML is a self describing language. XML documents can be stored without schemas and they contain meta data. Any XML tag can possess an unlimited number of attributes such as author or version. XML contains machine-readable context information. Tags, attributes and element structure provide context information. It opens up new possibilities for highly efficient search engines, intelligent data mining, agents, etc. XML can embed multiple data types. XML documents can contain any possible data type — from multimedia data (image, sound, video) to active components (Java applets, ActiveX). It is easy to map existing data structures like relational databases or file systems to XML. Null values and multiple values are managed efficiently by XML language. So XML is a good solution for dealing with websites having semi-structure.

### 2.1. NATIVE XML DATABASE

Native XML database is based on the container format, not in table format. It stores large amount of XML data and document. Native XML database is usually queried by the XPath-expressions. Native XML database has advantage over the XML-enabled database. It is highly capable to store, query and maintain the XML document than XML-enabled database.

A sample for an XML document is shown below. This is only a portion of document in the database we have considered("equity.xml")

```
<?xml version="1.0"?>
<share>
<security id="500510">
  <securityname>LT</securityname>
  <securitygroup>A</securitygroup>
  <open>1,474</open>
  <high>1,498</high>
  <low>1,400</low>
```

```
<noshares>1,028,765</noshares>
<turnover>14,840.39</turnover>
<trades>41,101</trades>
</security>
<security id="500285">
<securityname>SPICEJET</securityname>
<securitygroup>B</securitygroup>
<open>43.8</open>
<high>43.1</high>
<low>46.75</low>
<noshares>24,055,017</noshares>
<turnover>10,932.52</turnover>
<trades>44,857</trades>
</security>
<security id="503283">
<securityname>GEOJIT</securityname>
<securitygroup>B</securitygroup>
<open>36.40</open>
<high>54.40</high>
<low>43.40</low>
<noshares>44,055,32</noshares>
<turnover>42,265.74</turnover>
<trades>44,745</trades>
</security>
</share>
```

The data considered above is dynamically fetched from the stock exchange site and is transformed as XML Data.

### 3. BASEX

XML data handling is not an easy task even though it is highly readable. So tools are needed to make the XML content easy to explore, visualize and edit.

BaseX is a native and light-weight XML database management system and XQuery processor. BaseX is used for storing, querying, and visualizing large XML documents and collections. This platform independent DBMS uses standardized query languages such as XPath and XQuery. It also uses a tabular representation of XML tree structures to store XML documents. BaseX offers a Client-Server architecture to handle concurrent read and write operations of multiple users.

#### 3.1. XQUERY

XQuery Processing is the important part of BaseX. This is fast and efficient. XQuery defines a query that transforms collection of unstructured and structured data in XML format. XQuery is an expression oriented programming language and contains a superset of XPath. XPath expression addresses specific portions of an XML document. It gives FLWOR expression. That is, FOR,

LET, ORDER BY, WHERE, RETURN. BaseX gives java bindings also.

```
Let us consider an XQuery for the above XML document
for $v in doc("equity.xml")/share/security
where $v/open > 40
return data($v/securityname)
```

The result for the above XQuery will be,

*LT  
SPICEJET*

The above is an example for a FLWOR expression in XQuery. The names of equities having open values higher than 40 are displayed in the result. This gives user the ability to quickly and easily know about securities or equities. The visualization of XML database is shown in the figure Fig 1

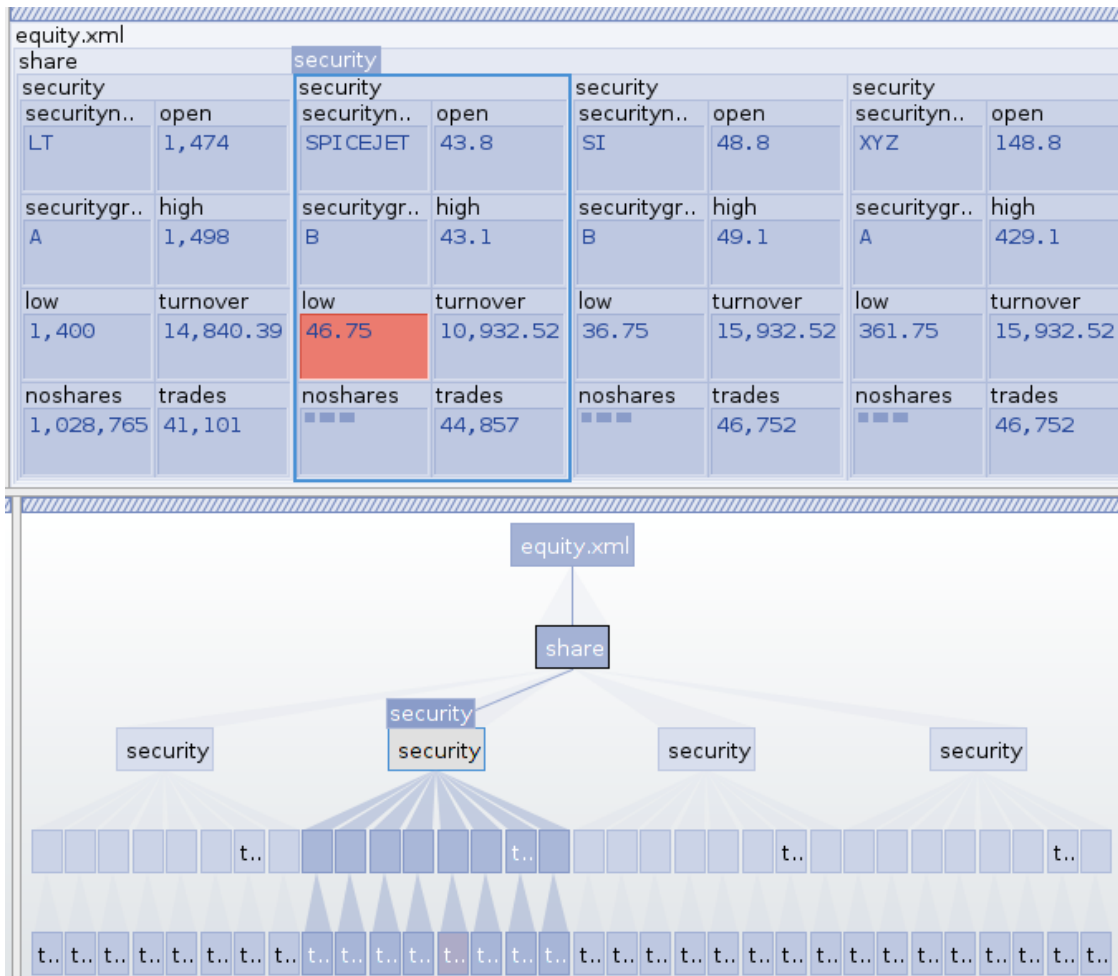


Figure 1. Map and Tree Visualization of XML data

#### 4. THE METHOD

In this paper, we are concentrating on different operations such as storage, retrieval ,update of shares. In Share Market Management System, the main operation is the retrieval of equity information based on some search queries. For example, if a person wants to know the securities whose open value is above some particular value, he can easily get that by giving the opening value as the search query. The interface accepts a keywords set from user, and then generates a FLWOR expression(Fig 2).[2]. The generated FLWOR expression is then sent to BaseX (Native XML Database). XQuery engine performs query on XML database, then sends user the result (Fig 3).

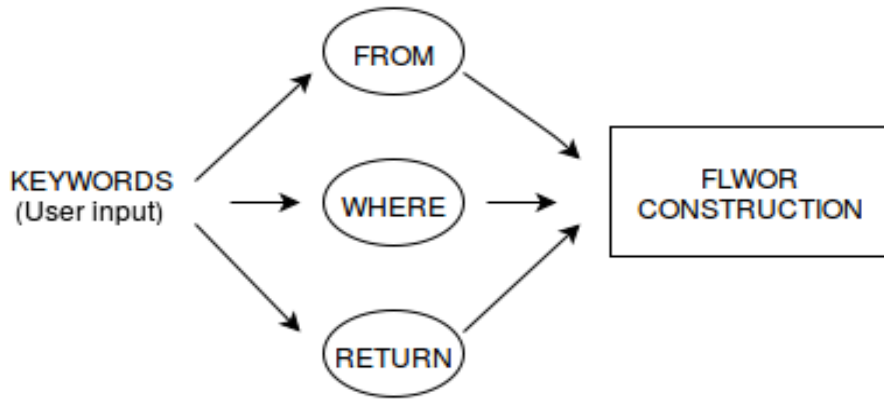


Figure 2. FLWOR Construction for XMLQuery

We have implemented it using the BaseX and Eclipse IDE. The XQuery is compiled and executed within the Eclipse IDE. The client version of the XQuery interpreter is used. The XQuery module is executed and the results are displayed in the window of the Eclipse.

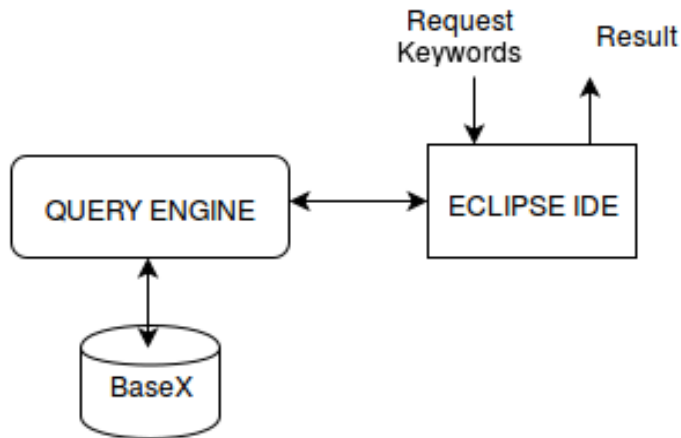


Figure 3. Query Processing

#### 4.1. STEPS

The stock market values are dynamically fetched from the stock market site, transformed and stored in "equity.xml". The details include equity name, open, high and low values, number of shares traded, total turnover. Later, an XML database was created using BaseX tool. After creation, the tree structure view of the XML database was examined.

A client program was written to communicate with the BaseX database. The FLWOR (For, Let, Where, Order by and Return) query is constructed after accepting keywords from the user. The client program in Eclipse IDE (An integrated development environment) was connected to a running BaseX server instance, executing database commands and evaluating XQuery expressions.

### 5. PERFORMANCE EVALUATION

A series of tests were conducted to evaluate the performance by varying the number of inputs. The general conclusions drawn from these experiments are represented in graph. For simplicity, we have considered small datasets only. It is shown in the Figure.4

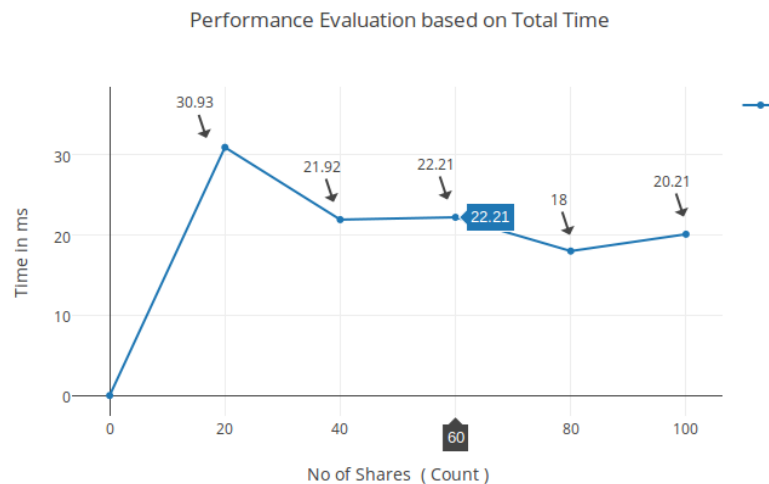


Figure 4. Performance Evaluation

### 6. CONCLUSION

The main function of Share Market Management System ( SMMS ) is to store, update and retrieve the information about share market values. Keyword-based query solution is implemented to obtain the results. The FLWOR query is constructed based on keywords from the user. The client program in Eclipse IDE was connected to a running BaseX server instance, executing database commands and evaluating XQuery expressions.

## REFERENCES

- [1] Zafari, H. ; Dept. of Comput. Eng., Islamic Azad Univ. Malayer Branch, Malayer, Iran ; Hasani, K. ; Shiri, M.E. XLight, An Efficient Relational Schema to Store and Query XML Data, Data Storage and Data Engineering (DSDE), 2010 International Conference.
- [2] ZHuaJiang ; Coll. of Comput. Sci. & Technol., Wuhan Univ. of Technol., Wuhan, China ; Qing Yang A Keyword-Based Query Solution for Native XML Database, Internet Technology and Applications (iTAP), 2011 International Conference.
- [3] Wahid, N. ; Dept. of Comput. Sci. & Eng., La Trobe Univ.,Melbourne, VIC, Australia ; Pardede, E. Single Transition Constraint for XML Update Validation , Network-Based Information Systems (NBIS), 2012 15th International Conference.
- [4] Draxler, S.; Stevens, G.; Boden, A. Keeping the Development Environment Up to Date A Study of the Situated Practices of Appropriating the Eclipse IDE , Software Engineering, IEEE Transactions on, On page(s): 1061 - 1074 Volume: 40, Issue:11, Nov. 1 2014
- [5] T. Grust and M. van Keulen.Tree awareness for relational DBMS Kernels: Staircase Join. In H. M. Blanken, T. Grabs,H.-J. Schek, R. Schenkel, and G. Weikum, editors. Intelligent Search on XML Data, volume 2818 of Lecture Notes in Computer Science, , pages 231-245. Springer, 2003

## AUTHORS

Darsana CS. is currently pursuing M.Tech in Computer Science and Engineering in Mar Athanasius College of Engineering. She completed her B.Tech from Government Engineering College,Waynad , Kerala. Her area of research is Database Transactions.



Roshni P is currently pursuing M.Tech in Computer Science and Engineering in Mar Athanasius College of Engineering. She completed her B.Tech from Federal Institute of Science and Tech. Angamaly Kerala. Her areas of research are Database and Big data Analytics.



Chandini K is currently pursuing M.Tech in Computer Science and Engineering in Mar Athanasius College of Engineering. She completed her B.Tech from LBS Engineering College ,Kasargod Kerala. Her areas of research are Machine Learning and Image Processing.



Surekha Mariam Varghese is currently heading the Department of Computer Science and Engineering, M.A. College of Engineering, Kothamangalam, Kerala, India. She received her B-Tech Degree in Computer Science and Engineering in 1990 from College of Engineering, Trivandrum affiliated to Kerala University and M-Tech in Computer and Information Sciences from Cochin University of Science and Technology, Kochi in 1996. She obtained Ph.D in Computer Security from Cochin University of Science and Technology, Kochi in 2009. She has around 25 years of teaching and research experience in various institutions in India. Her research interests include Network Security, Database Management, Data Structures and Algorithms, Operating Systems, Machine Learning and Distributed Computing. She has published 17 papers in international journals and international conference proceedings. She has been in the chair for many international conferences and journals.

