

A Java Based XML-Driver for Retrieving Data from the XML Document Using SQL-Query

S.Karthic

*Department of Computer Science and Engineering
KPR Institute of Engineering and Technology, Coimbatore, TamilNadu, India*

S.Kandasamy

*Department of Computer Science and Engineering
KPR Institute of Engineering and Technology, Coimbatore, TamilNadu, India*

Abstract- In the recent trend data transfer across the different platform is a major issue due to different technologies and operating systems in the market. To improve this cross platform communication, all the major companies agreed on a common format for data transfer as XML which is stated by W3C. Recently, a new concept called xQuery(using XQJ API) has been added which helps to parse the XML contents. Instead of using xQuery to parse the XML we will use the standard SQL to parse the XML contents with our java based XML driver. In this project we use the XML as Database and the standard SQL is used to parse the contents from the database, so that programmers need not learn new technologies

Keywords – XML, DOM, SAX, XQuery ,Parsing, DTD, Schema

I. INTRODUCTION

eXtensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. When the application programmer's needs retrieve the data from the XML file they can make use of XQuery which in turn uses SAX & DOM parser, depending on the XQuery API implementation.

Simple API for XML (SAX), an event-driven, serial-access mechanism for accessing XML documents. This protocol is frequently used by servlets and network-oriented programs that need to transmit and receive XML documents, because it is the fastest and least memory-intensive mechanism that is currently available for dealing with XML documents.

A DOM is a standard tree structure, where each node contains one of the components from an XML structure. The two most common types of nodes are ELEMENT nodes and TEXT nodes. Using DOM functions lets you create nodes, remove nodes, change their contents, and traverse the node hierarchy.

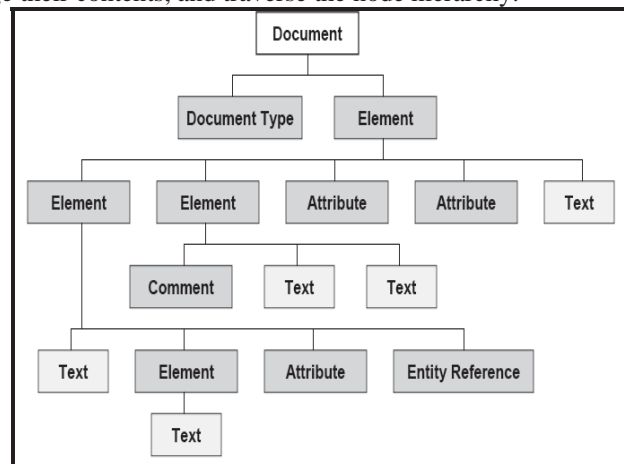


Figure 1 Simple DOM Structure

XQuery is a specification for a query language that allows a user or programmer to extract information from an XML file or any collection of data that is in an XML format. In order to achieve the above, the programmer has to learn a new syntax to extract the information. On the other hand, We have SQL, which is one of the most familiar querying language. In this project, we will provide a link between SQL and XML, by creating standard rules (API) and implementation.

II. PROPOSED SYSTEM

The proposed system aims to provide a link between XML and SQL. The above diagram shows the existing system.. The proposed combines Java Application connecting to the database using the JDBC API and connecting to an XML DB using XQJ API techniques. The input provided would be an SQL statement and the back end would be an XML file. This will minimize the difficulty associated with the xQuery usage.

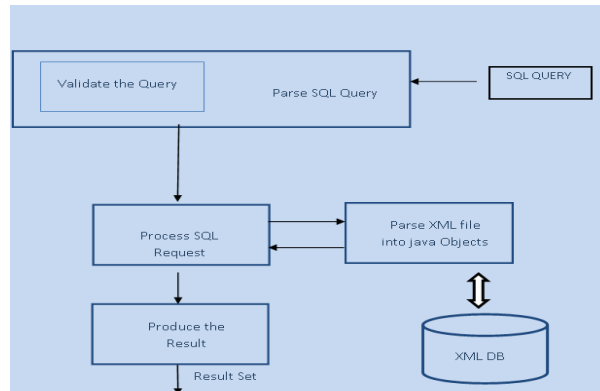


Figure 2. System Architecture

The architecture shows four main parts:

- Query Validation
- Query Processor
- XML Parser
- XML with TABLE Mapping

A. Query Validation

The input is given to the system in the form of an SQL Query. Before processing, the query has to be validated according to the SQL standards. This module will check for the semantics of the Query and check for the availability of the table which is in the form an XML document. If the query is valid and the expected XML file is available, then a successful message will be generated for the next processing to continue. Otherwise, an error message will be generated in the form of an exception. The scope of the validation module is to check only the DML commands syntax.

B. Query Processor

If the Query is valid then it has to be parsed into tokens to identify the kind of operation to be performed. The query request could be made for any of the Data Manipulation activities like SELECT, INSERT, UPDATE or DELETE. If the given SQL input is for the SELECT, then the ResultSet is created. If the given input is for INSERT, then it results in a success or failure message depending on the execution. For the rest of the operations, it returns the number of rows that gets affected in the XML database. If no rows are retrieved for the SELECT, then it results in NULL. And if no rows get affected, it results the value as 0.

C. XML Parser.

As DOM model is being used for parsing, the XML contents are converted into a tree like structure, i.e. in the form of Java Objects. Any operation which is performed on the XML content, will actually be performed on the DOM content from which the result is generated. While updating any content, the updation will be reflected in the DOM object first. Later on, the entire DOM object and its values are serialized into XML Database file.

D. XML with TABLE Mapping

The main idea behind this project is mapping the XML file with normal data base tables with rows and columns. Let's consider the following content in the form of table and how it can be represented in the form XML document.

Book ID	Book Name	Book Author	ISBN Number	Book Price
101	C++	Riche	92928383933	200\$
102	Java	Gosling	75775577575	300\$
103	J2EE	Mathew	34567788963	150\$
104	.Net	Dennis	25044444444	100\$

Figure 3. Sample Book Table

The representation of table in XML format for a row is given below

```
<?xml version="1.0"encoding="UTF-8" ?> <books>
  <book>
    <book_id>101</book_id>
    <book_name>C++</book_name>
    <book_author>Richie</book_author>
    <book_price>300$</book_price>
  </book>
</books>
```

Pseudocode

1. read_sql_query and read_xml_source_name
2. load_xml_content
3. decision_to_validate(sql_query)
4. if ok
5. parse_query(sql_query)
6. decide_to_parse_xml(xml_source_name)
7. if ok
8. process_query()
9. else
10. create_xml__exception()
11. endif
12. else
13. create_query_invalid_exception()
14. endif

III. EXPERIMENTAL RESULTS & ANALYSIS

The experimentation is done with the software namely: Windows XP, SQL 2000, Text pad 5, Jdk1.6.0, WordNet

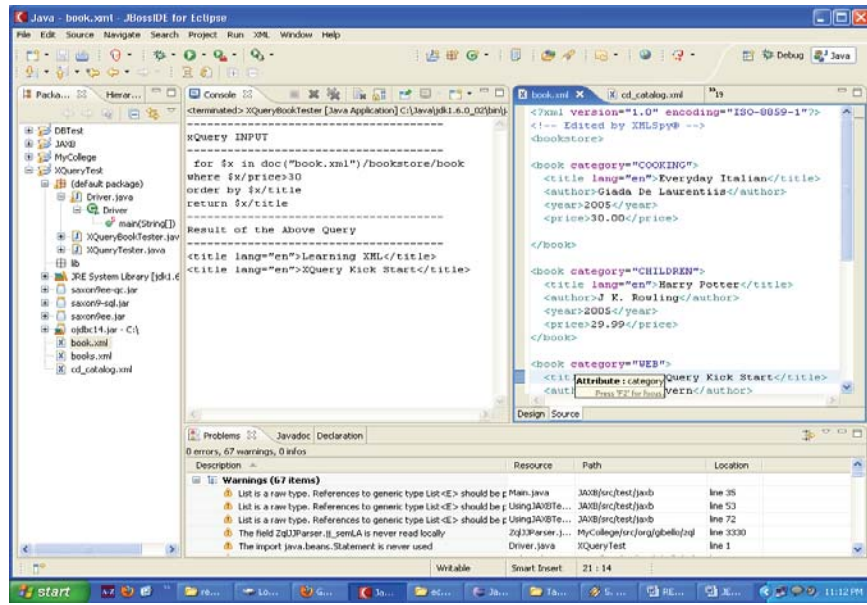


Figure 4 xQuery Result Page

Figure 4 shows the xQuery input ,which is a request to get all the books having the price value greater than 30 and result has to be sorted with respect to the title of the book

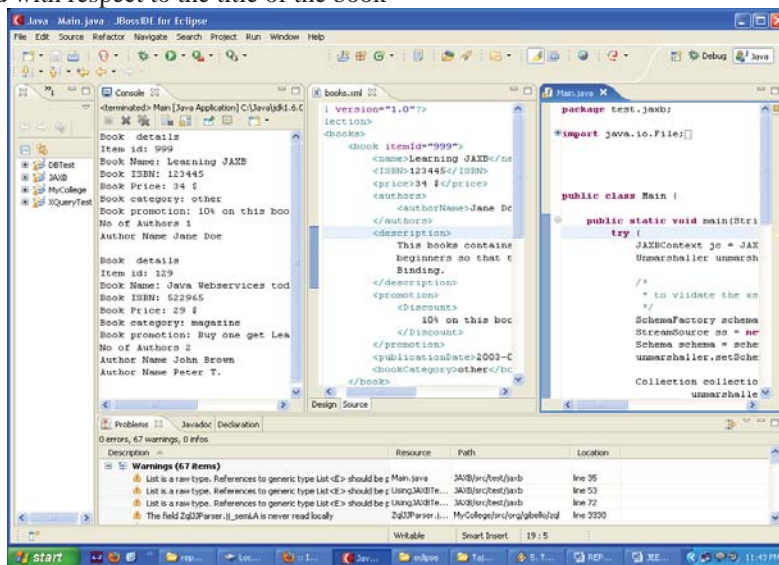


Figure 5 XML Parsed output

Figure 5 shows the parsed output of the XML. Here books.xml file is containing the actual data. The XML parser module used to extract the information from the XML file using the DOM object. Now traversing the DOM objects contents the actual data taken out. The DOM itself is having a method to validate the XML file with its Schema. If the xml is invalid with respect to the XSD stated it will generate the parser exception.

