



# INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

## CONVERTING XML DOCUMENT TO SQL QUERY

MISS. ANUPAMA V. ZAKARDE<sup>1</sup>, DR. H. R. DESHMUKH<sup>2</sup>, A. S. MAHALLE<sup>3</sup>, S. A. KARALE<sup>3</sup>

1. M. E. First Year, Department of Computer Science & Engineering, IBSS College of Engineering, Amravati.

2. Prof and HOD, Department of Computer Science & Engineering, IBSS College of Engineering, Amravati.

3. Asst. Prof, Department of Computer Science & Engineering, IBSS College of Engineering, Amravati.

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

**Abstract:** Developing techniques for managing and querying the growing body of XML data is becoming increasingly important. A popular approach to evaluating XML document is to translate them to relational queries and then to use a relational database system to evaluate the result. We demonstrate that it is possible to translate path expression queries (an important class of XML queries) into a single SQL query, even in the presence of recursion in the XML schema and the XML query. Experiments with a commercial relational database system show that the SQL queries output by our algorithms can be far more efficient than the queries output by previous translation algorithms. The Xindice Core server is a database server designed from the ground up to store XML Data. The Xindice server is what term by the XML is: DB Initiative as a Native XML Database.

**Keywords:** XML, SQL queries, Xindice, SAX parser, XML parser, DOM, JDBC, RDBMS



PAPER-QR CODE

Corresponding Author: MISS. ANUPAMA V. ZAKARDE

Access Online On:

[www.ijpret.com](http://www.ijpret.com)

How to Cite This Article:

Anupama Zakarde, IJPRET, 2014; Volume 2 (9): 1044-1051

## INTRODUCTION

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.

This project gives us an idea of converting XML document to SQL query using Xindice as backend to store XML document & source code "MyParser". MyParser consists of SAX parser. SAX is a common interface implemented for many different XML parser (and things that pose as XML parser), just as a JDBC is a common interface implemented for many different relational databases (and things that pose as relational databases).

**2. XINDICE:** The Xindice Core server is a database server designed from the ground up to store XML data. All data that goes into and out of the server is XML. The query language used is XPath and the programming APIs support DOM and SAX. When working with XML data and Xindice there is no mapping between different data models. You simply design your data as XML and store it as XML. Xindice is a tool. The job it is best at is simply storing XML data. If you have lots of XML data then Xindice may just be the right tool.

## 3. SYSTEM ANALYSIS:

**3.1 Objective:** In MyParser, we focus on XML SAX Parsing technique. By using data structure such as Array, we develop

a parser based on SAX Parsing technique which parse the given document and convert it into SQL Select Query. We have also implemented SQL clauses such as where, between, order by, like, null and not null. For converting XML document into clauses of SQL Query, for that firstly we parse the XML document and then retrieve that data and store that data in the form of array and then by applying some conditions we extract the XQL Query.

**3.2 Requirement:** Converting XML Document to SQL Query needs the following software:

- 1) Apache Tomcat
- 2) JDK 1.7.0
- 3) Xindice 1.1
- 4) JCreator

#### 4. SYSTEM DESIGN:

##### 4.1 Data Flow Diagram:

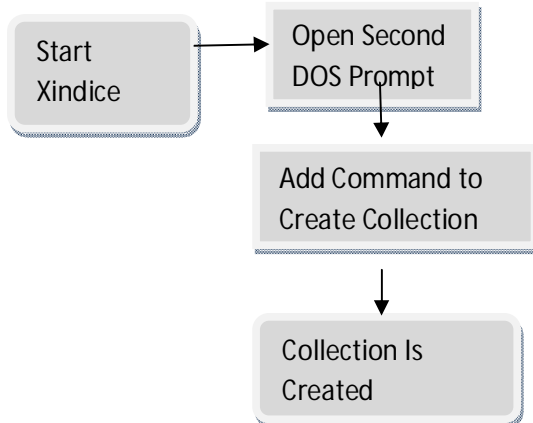
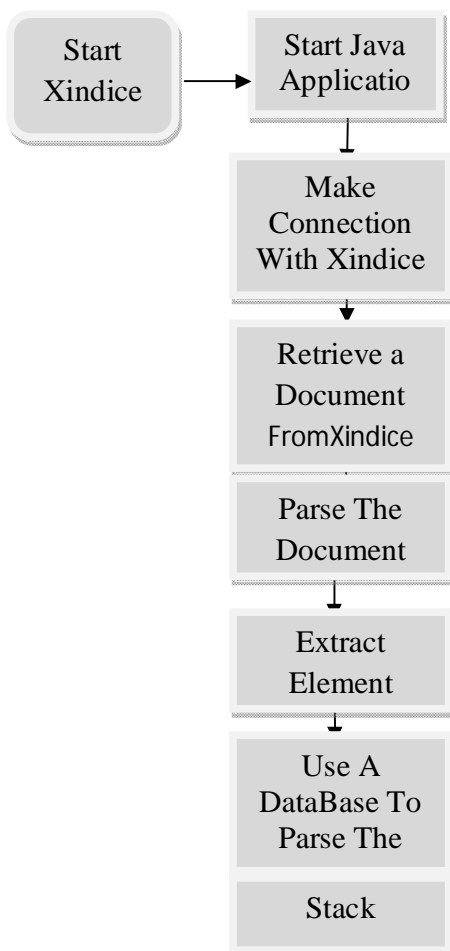


Fig.1: DFD for Creating Collections



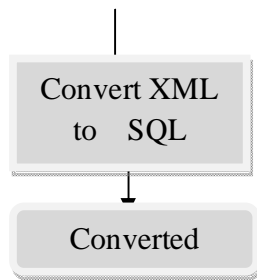


Fig.2: Parse XML Document

Firstly start Xindice Server the open MyParser program in JCreator, make connection with Xindice using tomcat. Then retrieve a document using Retrieve Document command of Xindice. Parse the document using SAX Parser, extract element and use database (stack) to parse the document then it will be converted into SQL Query.

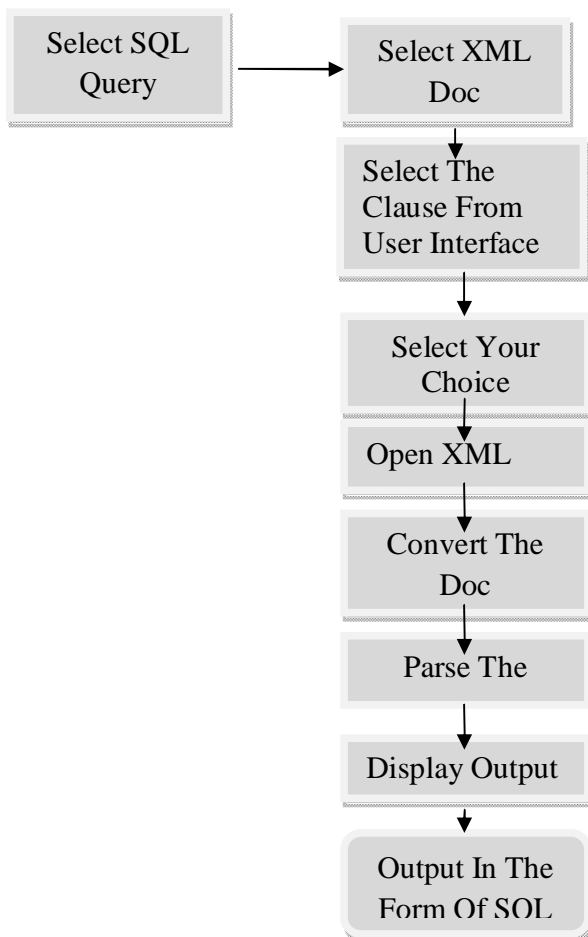


Fig.3: DFD of XML to SQL

Fig.3 gives DFD for Converting XML document to SQL Query. Select SQL Query option from interface then select XML document which you want to convert to SQL Query. And then select the clause (where, between, order by, like, null, not null) from user interface. Convert the selected document to Array, then parse the Array and display the output in the SQL form.

#### 4.2. Algorithm of MyParser:

- 1) Get XML file.
- 2) Select a document to convert into SQL.
- 3) Append "#" as a delimiter at the end of file.
- 4) Get start tag using String functions.
- 5) Add start tag to Data Structure.
- 6) Get end tag and check with the last tag in the Data Structure.
- 7) If start and end tag matches, remove it from the Data Structure.
- 8) Repeat steps 4 to 7 until Data Structure becomes null.
- 9) Fetch the SQL Query.

#### 4.3. Xindice Commands:

##### ❖ Collection Management Commands:

1. **Adding a Collection:** Adds a new collection to the database. When adding a collection under an existing collection hierarchy all parent collections must already exist in the database.
2. **Deleting a Collection:** Deletes a collection or sub collection from the database. If deleting a collection that has sub collections and documents they will also be deleted.
3. **List Collections:** List all collections under the collection context given.

##### ❖ Document Management Commands:

1. **Adding a Document:** Adds a document to a collection or nested collection.
2. **Adding Multiple Documents:** Adds multiple documents to a collection or a nested collection. This command requires two arguments: the collection to store the documents under and the directory path containing the documents.
3. **Retrieving a Document:** Retrieves an existing document from a collection or nested collection within the database. The complete path where the document will be stored is required.

**5. SYSTEM EXECUTION DETAILS:** This project gives result (output) in the form of SQL Query and Clauses of SQL.

### 5.1 SQL Clauses

1) **The WHERE Clause:** The WHERE clause is used to extract only those records that fulfill a specified criterion. The WHERE clause is used to filter records.

2) **Like Clause:** The SQL LIKE clause is used to compare a value to similar values using wildcard operators. There are two wildcards used in conjunction with the LIKE operator:

- ❖ The percent sign (%)
- ❖ The underscore (\_)

The percent sign represents zero, one, or multiple characters. The underscore represents a single number or character. The symbols can be used in combinations.

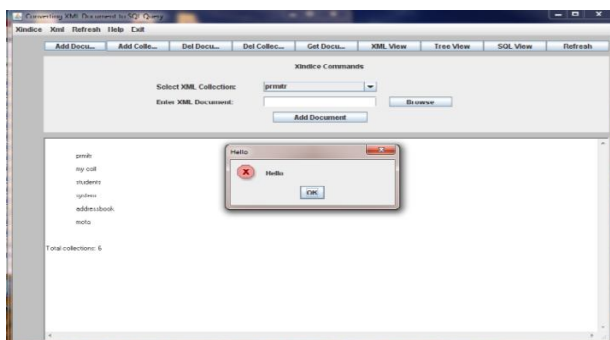
3) **The BETWEEN Clause:** The BETWEEN operator selects a range of data between two values. The values can be numbers, text, or dates.

4) **Null and Not Null Clause:** The SQL NULL is the term used to represent a missing value. NULL value is different than a zero value or a field that contains spaces.

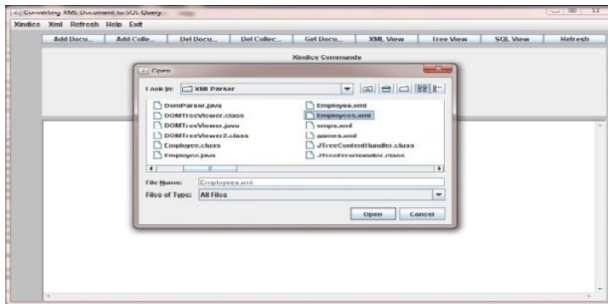
5) **Order By Clause:** The SQL ORDER BY clause allows you to sort the records in your result set. The SQL ORDER BY clause can only be used in [SQL SELECT statements](#).

### 6. RESULT:

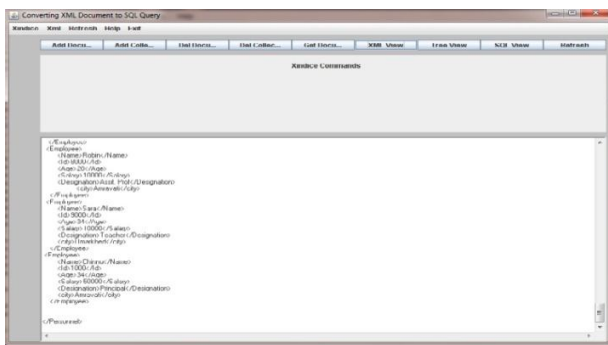
#### Screen shot 1: Open dialog box for adding document



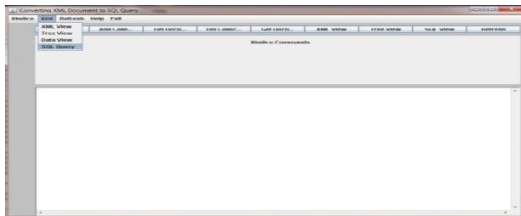
Screen shot 2: Open XML document



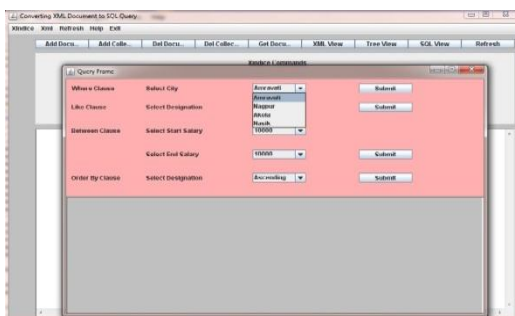
Screen shot 3: Showing the XML view



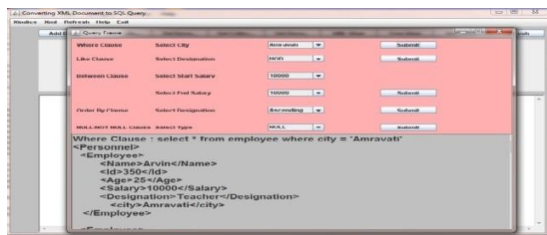
Screen shot 4: Selecting SQL Query from interface



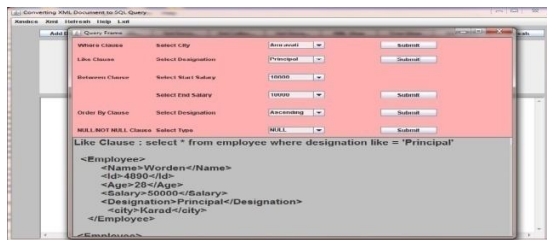
Screen shot 5: Selecting choice for "Where" clause



Screen shot 6: Converted XML document to SQL Where clause



Screen shot 7: Converted XML document to SQL "Like" clause



**7. CONCLUSION:** XQuery is a query and functional programming language that is designed to query collections of XML data. XQuery provides the means to extract and manipulate data from XML documents or any data source that can be viewed as XML, such as relational databases or office documents. The language is based on the XQuery and XPath Data Model (XDM) which uses a tree-structured model of the information content of an XML document, containing seven kinds of nodes: document nodes, elements, attributes, text nodes, comments, processing instructions, and namespaces.

## 8. REFERENCES:

1. Out *XML Data Management* edited by Akmal B. Chaudhri, Awais Rashid, and Roberto Zicari (Addison-Wesley, ISBN 0201844524).
2. Explore Dr. Richard Edwards' work on XML Repositories and minx, his generic architecture for storing XML Documents in a Relational Database.
3. M. Nicola and J. Johan, "XML parsing:" A threat to database performance in proceeding of the 12<sup>th</sup> international conference on information and knowledge Management, November 2003.
4. Yusof Mohod Kamir, Mat Atar, "High Performance of DOM Technique in XML for data Retrieval", 2009 International Conference on Information and MultimediaTechnology cICMIT'09. Page(s): 303-305."