ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

Querying XML Documents by Using Relational Database System

Hina A. Kore[1], Sayali D. Hivarkar[2], Neha K. Pathak[3], Ravindra S. Bakle[4], Prof.Sapna S. Kaushik[5]

Student, Computer Department, DBNCOET, Yavatmal/Sant Gadge Baba Amravati University, Amravati, India[1,2,3,4]

Assistant Professor, Computer Department, DBNCOET, Yavatmal/Sant Gadge Baba Amravati University, Amravati, India[5]

Abstract: An XML database is a data persistence software system that allows data to be stored in XML format. This data can then be queried, exported and serialized into the desired format. XML databases are usually associated with document-oriented databases.One reason for the use of XML in databases: the increasingly common use of XML for data transport, which has meant that "data is extracted from databases and put into XML documents and vice-versa". It may prove more efficient and easier to store the data in XML format.The XML database is a rapidly growing technology which is poised to replace many existing technologies used for data storage. It uses xml and many of its derived technologies, like DTD, XSL, XSLT, Xpath, Xquery, etc., as its framework. XML document is self-describing and is compatible on all kind of platforms because it is in text format. This makes it a very powerful technology. Due to its simplicity and compatibility on all kinds of platforms , XML database is rapidly becoming the de facto standard of passing data over the internet. The general purpose based on the XML documents is to access the data by using XPath and other techniques. But for general user it's not possible to remember all the XML queries. So our aim is to Design a SQL Interface for XML databases. This Interface will accept sql queries, convert them into XPATH expressions and will retrieve data from XML Documents.

Keywords*:* XML, Xpath, SQL.

I. INTRODUCTION

An XML database is a data persistence software system that allows data to be stored in XML format. This data can then be queried, exported and serialized into the desired format. XML databases are usually associated with document oriented databases. Two major classes of XML database exist:

A. XML-enabled

These may either map XML to traditional, accepting XML as input and rendering XML as output, or more recently support native XML types within the traditional database. This term implies that the database processes the XML itself.

B. Native XML

The internal model of such databases depends on XML and uses XML documents as the fundamental unit of storage, which is, however, not necessarily stored in the form of text files. One reason for the use of XML in databases: the increasingly common use of XML for data transport, which has meant that "data is extracted from databases and put into XML documents and vice-versa". It may prove more efficient and easier to store the data in XML format.

The formal definition from the XML: DB initiative states that a native XML database:

1. Defines a (logical) model for an XML document — as opposed to the data in that document — and stores and retrieves documents according to that model. At a minimum, the model must include elements, attributes, PCDATA, and document order. Examples of such models include the XPath data model, the XML Info set, and the models implied by the DOM and the events in SAX 1.0.
2. Has an XML document as its fundamental unit of (logical) storage, just as a relational database has a row in a table as its fundamental unit of (logical) storage.
3. Need not have any particular underlying physical storage model. For example, NXDs can use relational, hierarchical, or object-oriented database structures, or use a proprietary storage format.Additionally, many XML databases provide a logical model of grouping documents, called "collections". Databases can set up and manage many collections at one time. In some implementations, a hierarchy of collections can exist, much in the same way that an operating system's directory-structure works. All XML databases now support at least one form of querying syntax. Minimally, just about all of them support XPath for performing queries against documents or collections of documents. XPath provides a simple pathing system that allows users to identify nodes that match a particular set of criteria. In addition to XPath, many XML databases support XSLT as a method of transforming documents or query-results retrieved from the database. XSLT provides a declarative language written using an XML grammar. It aims to define a set of XPath filters that can transform documents (in part or in whole) into other formats including plain text, XML, or HTML. Many XML databases also support XQuery to perform querying. XQuery includes XPath as a node-selection method, but extends XPath to provide transformational capabilities. Traditional RDBMS vendors (who traditionally had SQL-only engines), are now

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

shipping with hybrid SQL and XQuery engines. Hybrid SQL/XQuery engines help to query XML data alongside the relational data, in the same query expression. This approach helps in combining relational and XML data.

II. XPATH

XPath 2.0 is the current version of the XPath language defined by the World Wide Web Consortium, W3C. It became a recommendation on 23 January 2007. XPath is used primarily for selecting parts of an XML document. For this purpose the XML document is modelled as a tree of nodes. XPath allows nodes to be selected by means of a hierarchic navigation path through the document tree. XPath 2.0 is used as a sublanguage of XSLT 2.0, and it is also a subset of XQuery 1.0. All three languages share the same data model, type system, and function library, and were developed together and published on the same day.

A. Data model

Every value in XPath 2.0 is a sequence of items. The items may be nodes or atomic values. An individual node or atomic value is considered to be a sequence of length one. Sequences may not be nested .Nodes are of seven kinds, corresponding to different constructs in the syntax of XML: elements, attributes, text nodes, comments, processing instructions, namespace nodes, and document nodes.

B. Type system

The type system of XPath 2.0 is noteworthy for the fact that it mixes strong typing and weak typing within a single language. Operations such as arithmetic and boolean comparison require atomic values as their operands. If an operand returns a node, then the node is automatically atomized to extract the atomic value. If the input document has been validated against a schema, then the node will typically have a type annotation, and this determines the type of the resulting atomic value. If no schema is in use, the node will be untyped, and the type of the resulting atomic value will be untyped Atomic.

C. Path expressions

The location paths of XPath 1.0 are referred to in XPath 2.0 as path expressions. Informally, a path expression is a sequence of steps separated by the "/" operator, for example a/b/c (which is short for child::a/child::b/child::c). More formally, however, "/" is simply a binary operator that applies the expression on its right-hand side to each item in turn selected by the expression on the left hand side. So in this example, the expression a selects all the element children of the context node that are named <a>; the expression child::b is then applied to each of these nodes, selecting all the <b> children of the <a> elements; and the expression child::c is then applied to each node in this sequence, which selects all the <c> children of these <b> elements. The "/" operator is generalized in XPath 2.0 to allow any kind of expression to be used as an operand. For example, a function call can be used on the right-hand side. The typing rules for the operator require that the result of the first operand is a sequence of nodes. The right hand operand can return either nodes or atomic values. If the result consists of nodes, then duplicates are eliminated and the nodes are returned in document order, and ordering defined in terms of the relative positions of the nodes in the original XML tree. Other operators available in XPath 2.0 include the following:

D. Operators Effect

+, -, *, div, mod, div: Arithmetic on numbers, dates, and durations.

=,!=, <, >, <=,>=:General comparison: compare arbitrary sequences. The result is true if any pair of items, one from each sequence, satisfies the comparison

eq, ne, lt, gt, le, ge: Value comparison: compare single items is Compare node identity: true if both operands are the same node.

<<, >>: Compare node position, based on document order union, intersect, except Compare sequences of nodes, treating them as sets, returning the set union, intersection, or difference and, or boolean conjunction and disjunction. Negation is achieved using the not() function. to defines an integer range, for example 1 to 10 instance of determines whether a value is an instance of a given type cast as converts a value to a given type castable as tests whether a value is convertible to a given type.

III. SQL

SQL Stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

A. Why we combine SQL with XML?

As we know SQL is the basic language for Database management. Managing database depends upon Structural query language. The question arrives why we are using SQL for XML Documents? As we know the Xpath, Xquery and many more are the basic techniques for handling XML data. But the general users of the database always not aware about the Xpath and many more techniques for accessing the data. The simple solution on that is to use the SQL for accessing the XML data. User can't recognize the specific notation for XML data. So We will access the XML data by using Relational database query.

B. How it will work?

On the database server many more users are connected at the same time, so there will be the many client system which will extract data from the Database system. We will extract the XML data from XML database server from SQL client. The user will extract the data from database by using SQL language. But now the question arrives that,How the SQL query access the XML data?

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

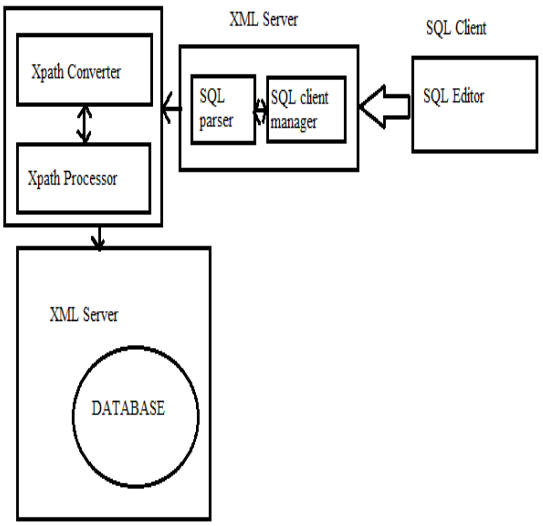

Fig. Working of XML Data extracting

1.SQL Server
The client will send the request through SQL client to XML server by using Relational Database Query.

2.XML server
a)Xpath Converter
The Xpath converter will convert the SQL query in Xpath expression which will extract the XML data from the Database i.e XML server.

b)Xpath Processor
It will decide the type of request done by client and proceed it as per request.The output will display on sql screen.

IV. FEATURES

1. It will provide an visual interface for an XML database.
2. Users will be able to upload XML Documents into this database.
3. This Interface will display list of all XML Documents present in database.
4. We can extract specific part of an XML document.
5. User can fire a SQL query on the opened document.
6. The SQL-To-XML converter will automatically convert it into XPATH expression.
7. The XPATH expression will be used to retrieve data from the document.
8. The retrieved xml data will be displayed in SQL Editor.

V. CONCLUSION

XML databases are rapidly becoming the de facto standard of transferring data over the internet. Also, because of their ability to store heterogeneous data, they are used in a wide variety of fields like e-publishing, digital libraries, finance industry, etc. Xml standard is evolving day and night and new standard of xml are being created for almost every field. Open source base and simplicity has made it best tool to pass information. We live in an era of information technology and xml databases is the best vehicle, we can ride on.

REFERANCES

[1] A General Technique for Querying XML Documents using a Relational Database System by Jayavel Shanmugasundaram,Eugene Shekita,Jerry Kiernan.
[2] Distributed XML Database Systems by Marko Smiljanić, Henk Blanken, Maurice van Keulen, Willem Jonker.
[3] XML and Databases Ronald Bourret Independent consultant, Felton, CA.
[4] Editor Martin Bryan, "Guidelines for using XML for Electronic Data Interchange," January 1998, XML/EDI Grop.