RDBMS Information Using Xml Query

Sangana kotireddy[1],Yadhukrishna M R[2], Thangamma K C[3]
[1, 2]Department of Information Science Department
[3]Department of Computer Science Department
[1, 2, 3]Coorg Institute of Technology, Ponnampet, Kodagu, Karnataka, India

Abstract*-In the company of the speedy mounting recognition of XML to characterize information, how to build a superior make use of XML information in relational information with databases is valuable of study. store XML information as text in relational databases is a established policy which cannot reproduce the quality of XML format. In this paper, a mechanism for XML information storage and query in relational databases is proposed. XML information can be store in relational tables and XQuery expressions can be evaluate as a part of SQL for XML information query. XQuery grammar tree and Query tree model for XML information query in relational information with databases is to be had to gain additional resourceful piece while querying XML information.. Finally, experiments cancel the strategy of XML storage and run the algorithm on real XML information of datasets to show the effectiveness compared with other mechanism.*

Keywords-Grammar Tree, , XQuery, XQuery evaluation, XML Yable, Structural Index.

I. INTRODUCTION

In the company of the speedy mounting recognition of XML, it has turn out to be a pattern format to store information in a lot of area and share information involving them. although XML has be use in different domains for information exchange and demonstration, a large pact of XML information appear and how to direct XML information powerfully has suitable a hotspot between researchers.

Relational information with databases are the most trendy in information managements. As relational information with databases are budding quickly, people understand that XML is a useful information format. As a result, offthe-shelf RDBMS start on to support XML information type, such as SQL Server, DB2 and Oracle. How to put together XML information into relational information with database is attract extra people for research.

Within order to use XML information in relational information with databases and construct the good use of XML information and relational information in information with databases, how to store, query and update XML information in RDBMS is worthy of study. Query optimizing is an significant crisis in examine of XML. XPath is an key language for XML query which is offered by W3C as a standard. The simplest query in Native XML information with database is by XQuery which is superset of XPath and XQuery is also presented by W3C as a standard.

In universal, querying XML information in XML support information with databases include following steps: grammar scrutiny, rewrite, optimizing and preparation.

Analyzing action will parse XPath expression to a model. Optimizing step will rephrase XPath model as an best example and planning move will make plan of XML query. Query unpolluted XML information by XQuery and XPath have be paid concentration to by researchers for several years while XML has been available as model by W3C. The main problem in XPath query giving out is the identical of structural relationship. C.Zhang et al.[2], combine with the established thoughts of integration algorithm and period encoding, future Multi- Predicate Merge Join (MPMGJN) .

MPMGJN reduce the evaluation involving XML nodes by their document order correlation. the function of stack acting an important role in XPath query algorithm. Nevertheless, MPMGJN and StackTree may bring into being very large intermediary results.

To solve this crisis, N. Bruno et al.[4] projected twig join algorithm that map XPath query tree into simultaneous stacks,and then by recursion scrutiny structures in stream pushes the nodes similar the structural relationship into stacks, finally gets the result in a leaf-to-root path.

In addition, generally of alive XPath query algorithm assume the XML information is external the DBMS, so individuals algorithms would have a very huge cost on time and space to retrieve XML information. This retrieve technique has turn out to be the bottleneck of structural join. In paper [4] and [5] structural index was projected and was used in XPath query algorithm, on the other hand it didn't resolve the trouble in query rewrite base on structural index.

This paper will focus on put into operation of XML query based on relation tables in RDBMS[5][6]. Firstly, XML information is store in relation tables plus structural indexes are build for XML information. Secondly, XQuery expressions are parsed to grammar tree. as well as valuation of XQuery will be perform by twig join algorithms.

II. STRUCTURAL INDEXES AND XML TABLE

In direct to store XML information in relational information with database, an attribute for relative named “xml” will be further in information with database.

The followed by XML information can be store as columns in tables. XML information in table is not store as text like SQLServer, except stored in XML tables. XML information in columns of XML tables is not available as XML fragment except stored as a pointer linked with one XML fragment or document, such as “sportSold” column in XML table shown by Table I.

TABLE I. Xml Information Stored In Relational Information With Database

sportStore ID	sportStor eName	sportSold
11	Abc	XMLType(rID=17003,do cID=1
12	Xyz	XMLType(rID=17003,do cID=2)
13	Abc1	XMLType(rID=17003,do cID=3)

A. XML Tables

XML Table are used to store up XML information of an XML document in relational model. All XML information in relational information with database can be store in one XML table and illustrious by SSID as foreign key of the table.

Table II shows XML table for document named sss.xml.Columns in XML table are named nodeid, type, name, value and path. When value of item in type column is DOCUMENT,the value in nodeid is identifying docID. As a result, XML tables are stored in relational information with database as fixed schema.

TABLE II. SSS.XML (XML TABLE)

Nodeid	type	name	val	Path
01	DOCUMENT	Null	Null	null
02	ELEMETNT	1	Null	Null
03	Elemetnt	2	Null	01

B. XML information with Structural Indexes

Structural Indexes are attributes, paths and elements of XML information, where it can characterize structural information of XML documents and attain XML nodes to form query results.

Structural Indexes base on XML tables will be use for XML query in this paper. During XML table, column name path record paths of related XML nodes, and every path is tell upturned. Data of XML nodes can be obtain by path quickly, as a result, index can be build on column path. Such index is named Structural Summary Index and as well named Path Index.

Mapping the path of XML nodes to column named nodeid in XML table can obtain values of nodes where labels are stored in interrelated column name, as shown in Fig. 1.

Structural outline index can speediness up XML query based on XPath since the index can get query result for additional continuous PC relation in XPath rapidly at the same time as user inputs a path for XML query, he can get gathering of nodes values by mapping column path to nodeid.

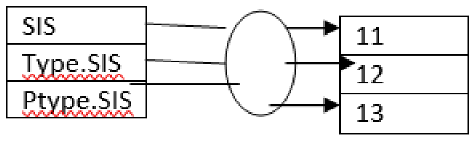

Fig. 1 Structural Summary Index

III. XQUERY ANALYSIS AND GRAMMAR TREE

Using XQuery and SQL terms to query XML information in relational information with database, XQuery terms will be represent by tree model named XQuery grammar tree. Then, evaluateXQuery based on the representation and query result for XML information would be obtain.

The query results will use as components of SQL terms.

A.XML/SQL Expression

XQuery is obtainable by W3C as standard of xml information query.

XML/SQL provide formal appearance for xml query in SQL.

For example, expression:

select XmlQuery (‘/SIS/SPORT’, SportSold) from sportStore where sportStoreID =1);

conform to SQL/XML for querying xml information from relational information with database by SQL. Function XmlQuery is provide for query xml information in xml column of relational information with database. In the handing out of query, terms conform to SQL/XML will model by grammar tree.

B.Tree Grammer for XQuery

XQuery terms will be analyze first to XQuery grammar tree. XQuery grammar tree includes following types of nodes:

- "XmlQuery" will be root node
- Clause nodes include five XQuery clauses: ""Return", "Where","Order by", "Let", and "FOR"
- Path nodes include "ReletivePath" and "AbsolutePath"
- Step nodes note each step for evaluating of XQuery.

XQuery expression:
for $sport in //sport
let $stitle:=$sport/stitle
where id=1
return $stitle

. grammar tree for XQuery can convey all components of XPath, such as, predicates,
node test,axis for conforming to XQuery.

C. Grammar Tree for Filling XQuery

Grammar tree for XQuery is beginning XQuery model. In the replica, all part of XQuery expression can be considered as nodes in the tree model. unluckily, information of xml query, such as target columns, mode of order, cannot be describe fully in XQuery grammar tree. since a result,grammar tree will be extensive to Query Tree model for xml query executing every kind of nodes will explain components of XQuery expression.
Thus, expressions for XQuery can be represented by Query trees.

For example, expression:
/SIS/SPORT/name[last()] shows a information structure named stepList is use to perform algorithms for structural join to query XML information base on Query tree model. StepList exist as link list, whose nodes are steps in XQuery terms, such as, predicates and axis names. For example, "for" clause in XQuery expression:

for $sport in //sport

can be filled in the node "ForClause" and "//sport" can be filled in the stepList of the node Querying XML information is a processing of evaluating results in Query tree with two stages: preparing and executing.

A. Preparing Stage
Here, parameters for filling Query tree and other structure of XML query will be prepared. For example, parameters of functions to fill Query tree, such as *execInitExpr,*will be prepared in this stage. Whereas all parameters have been set some values, XQuery terms will be executed in executing stage.

B. Executing Stage
According to parameters provide in prepare for stage, XQuery terms for querying XML information are executed in this stage.
Executing stage includes two tasks: XPath evaluation and XQuery evaluation. Usually, XPath expressions are added in XQuery expressions.

As a result, XPath evaluation is an vital performance in querying XML information. Indexes, such as value indexes, path indexes and structural indexes, can be used to speed up evaluation of XPath.

In this paper, path indexes will be used for XPath evaluation. In XPath terms, parent-child (PC) step appears regularly, as a result, path indexes can process this value of axis rapidly.
For example, XPath expression:
/sport/sports/item can obtain collection of XML nodes: {10,11,12} by path indexes openly and structural join is not necessarily used in each step.
Algorithm for structural join usually processes parent-child step while it cannot deal with ancestor-decedent (AD) step,such as "//".
Two parts of the expression are divided by "//", they are "/sport/item/" and "/author/first". These two sub XPath expressions can be evaluated by path indexes respectively and algorithm for structural join, such as TwigStack and PathStack [3],will be used to get final query results. Either algorithm is fit for native XML information query and cannot be used to query XML information in relational informationbases directly. Thus, an appropriative algorithm shown in Algorithm 1 will be improved in this paper to query XML information by XML Query tree model based on structural indexes.

Next task is evaluating XQuery according to following steps:
a) Executing "in" sub clauses of outer shell "for" clauses and "let" clauses. Then, a sequence of results will be obtained.

b) Sequence of results will include XML nodes. Each node will be assigned to a binding variable “$i”
c) Deciding whether conform to “where” clauses for each node represented by “$i”
d) Executing return clauses for nodes meeting conditions.
e) Packing query results to some format fitting for algorithms of joining.

IV. CONCLUSION

This paper obtainable an effectively mechanism for XML information storage in relational information with database using XML tables. Based on XML table, structural indexes are built for higher efficiency query of XML information in relational information with databases. In order to implement XML query by XQuery expressions, XQuery grammar tree and Query tree model was proposed in this paper. By these models and appropriative algorithm for XPath evaluation,XQuery terms can be executed quickly and expediently to get concluding query results. Experiments demonstrate that using our mechanisms can get perfect results in XML information storage and query in relational information with databases.

REFERENCES

[1] S. Al-Khalifa, H. V. Jagadish, J. M. Patel, Y. Wu, N. Koudas,and D. Srivastava. Structural joins: A primitive for efficient XML query pattern matching.

[2] C. Zhang, J. F. Naughton, D. J. DeWitt, Q. Luo, and G. M. Lohman. On supporting containment queries in relational informationbase management systems.

[3] Lu Qin, Je_rey Xu Yu, and Bolin Ding. TwigList:Make twig pattern matching fast.

[4] N. Bruno, N. Koudas, and D. Srivastava. Holistic twig joins: optimal XML pattern matching.

[5] GALAX System. Available: http://www.galaxquery.org/

[6] N. Grimsmo, T. A. Bjørklund, and M. L. Hetland. Fast optimal twig joins. PVLDB, 3(1):894–905, 2010.

[7] PostgreSQL: http://www.postgresql.org/