ISSN: 2321-7782 (Online)
Volume 3, Issue 5, May 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Techniques of Mapping XML Data: A Survey

Sonam S. Kharade[1]
Dept. of Computer Science Technology
Dept. of Technology, Shivaji University,
Kolhapur – India

Chetan J. Awati[2]
Dept. of Computer Science Technology
Dept. of Technology, Shivaji University,
Kolhapur – India

Abstract: XML, the Extensible Mark up Language, has gone from the most recent trendy expression to a settled in eBusiness innovation in record time. XML gets from a reasoning that the data has a place with its creators and that content suppliers are best served by an information design that does not tie them to specific script dialects, and conveyance motors yet gives an institutionalized, seller free, whereupon distinctive writing and conveyance instruments might unreservedly finish. XML is usually sustained by SQL database framework. Mapping leads to normalization of data and provides good performance for document types and queries. Mapping also plays vital role in reducing the number of joins. This paper reveals survey related to mapping XML data.

Keywords: XML, Database, Query processing, schemas, Normalization.

I. Introduction

Extensible Markup Language (XML) is quick rising as the prevailing standard for speaking to information on the Web. XML is the universal standard format used for data exchange. XML is very much a part of the future of Web, and part of the future for all electronic information. Since the start of XML technology a decade ago, there has been tremendous amount of interest and effort in supporting XML and its associated languages, such as XPath, XQuery, XSLT and SQL/XML, from both research and industry in the database community [17].

XML is a standard for describing how data is structured. This makes it much simpler to move structured data from place to place or from one program to another. Though, it doesn't determine how data on a particular topic ought to be structured, it does provide syntax for writing such specifications, called XML Schemas or DTDs (document type definitions). Making a document smarter and portable is an important feature of XML. For storing XML data, Relational database system is most commonly used.

Mapping nested elements to flattened tables is the key problem for supporting XML on SQL databases [1]. XML mapping is a property on a content control that links or binds the content of the content control to an XML element in a data store that is stored with the document. By utilizing XML mapping, XML data is divided from the presentation of the document so it is less demanding to modify both data and organizing automatically. This separation of data from formatting enables you to make more robust documents than you could with past versions of Word.

As there are large, complex and different types of queries a fixed mapping is unlikely to work. An XML document contains a root element following the nested elements. Elements can be either attributes or sub-elements. Mapping these nested elements is a problem. Mapping can be done in several ways. Many mapping schemes have been proposed. The existing mappings are namely STORED system [11], normalization mapping [8], [11], [18], and node-encoding mappings [9], [13], [14] and mapping XML to a wide sparse table [1]. Mapping XML data into relational tables include scanning and parsing of document then storing all information into relational tables. Whereas, mapping XML to a wide sparse table resembles normalization [1].

Despite the fact that standardization mappings give outline and standardization based enhancements, they can just bolster a limited number of use cases. In particular, we identify three use cases that are generally seen in big business applications however can't be bolstered by standardization mappings productively.

1) *Flexible schema*: XML's pattern less property is one of its real points of interest over other construction first information models. In standardization mappings, blueprint plan of essential outside keys obliges full learning of the XML pattern, and can't be connected to this utilize endeavor applications regularly utilize numerous little XML reports instead of a couple of vast ones [5].

2) *Many small XML documents*: Standardization mappings regularly shred one XML report over numerous tables, which cause fracture of little XML reports. As an outcome, the expense of XML reproduction can be high in case. While the XML pattern can be compressed from the introductory report set [11], later overhauls may bring about lavish outline advancement and table repartitioning.

3) *Many optional elements/attributes*: By information standardization, XML qualities are put away as extra sections in the tables of their guardian components. This method, then again, brings about numerous sparse tables and acquires stockpiling overhead.

A novel mapping that takes after standardization and holds all diagram and standardization based improvements for XML with adaptable patterns was proposed. The thought is to guide XML to a physical wide sparse table, in which standardization is communicated as coherent sub-tables. Coherent sub-tables are communicated as meta-information which is kept up in primary memory. Upon diagram advancement, coherent sub-tables develop in primary memory, which evades the expense of physical table repartitioning. At inquiry accumulate time, coherent sub-tables are seen as regular standardized tables; also, joins on one table are dispensed with.

II. Literature Survey

There has been a lot of work done related to mapping in database. Deutsch, Fernandez and Suciu propose the STORED system for mapping between semi-structured data and the relational data model [11]. The procedure depends on a mapping between the semi structured information model and the social information model, communicated in a query language called STORED. They concentrate on an information mining procedure which at the same time solves schema revelation and capacity mapping by distinguishing profoundly bolstered tree designs for capacity in relations. At the point using the data mining system a STORED mapping can be produced naturally when a semi structured information example is given. For this purpose an automatic storage generation algorithm was also developed. It changes over every query design into a data tree and expands the data mining calculations to the new data items.

The goal in [18] was to investigate the use of relational database systems to process queries on semi-structured documents. Recursive queries, addressing the issue of developing the outcome in XML and assessing the social methodology utilizing genuine DTDs were handled. In this approach the interpretation from document schemas to relational schemas is performed manually rather than automatically.

In database systems, schema and normalization have a significant effect on query preparing. Query minimization [2], [10] aims to use schemas and constraints to minimize tree patterns at query compile time. XML queries are usually represented through XPath expressions by visualizing some parts of the documents. One of the problems related to XPath query optimization was its minimization. A solution to this was the development of an algorithm for computing the least XPath queries implemented in [2]. Also, identification of fascinating tractable case and an ad hoc algorithm capable of handling minimization of queries in polynomial time has been proposed. This paraphrase covered the issue of the minimization for tree patterns belonging to the fragment of XPath and its remedy.

Since SQL database supports XML, mapping nested elements to flattened tables is an issue. Various mapping techniques were proposed namely normalization and node-encoding mappings. Normalization mappings shred nested structures into normalized tables, where each table corresponds to one or more elements/attributes types [8], [11], [18]. Node-encoding mappings represent XML elements by numerical encodings of node locations in the XML tree, e.g., Dewey ID or preorder/postorder [19].

Mapping XML to a wide sparse table resembles normalization for XML with flexible schemas [1]. Logical sub-tables were created by storing the elements & physically represented using interpreted storage in order to retain normalization. Annotated DG (Data Guide) was implemented in order to eliminate joins and reduce the cost of schema evolution.

Utilizing diagrams to rearrange query builds is a clear enhancement. Query minimization [2], [10] points to utilize diagrams and requirements to minimize tree designs at query execution time. This thought can be reached out by the mix of basic rundowns (on the off chance that the construction is not accessible) and lists, in which XML components are connected to the comparing synopsis hubs [6], [12], [15].

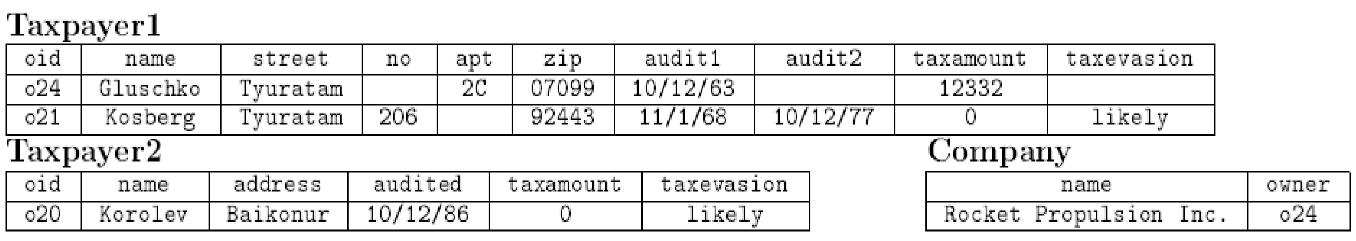
This empowers the motor to rearrange query designs at incorporate time and to rapidly get to occasions at execution time. Then again, structure records are by all account not the only schema based improvements. Compositions or auxiliary rundowns can be combined with basic stockpiling too, i.e., schema driven capacity. To comprehend the distinction, consider hub encoding mappings in which all XML components are put away in a solitary turn table. In those frameworks, regardless of the possibility that inquiries are improved, finding the coordinated components can't abstain from getting to numerous insignificant components, in light of the fact that those components are not physically isolated.

Mapping driven XML stockpiling strategies proposed in the writing are taking into account archive part, e.g., by tag names [14] or by ways [4]. The advantage is that once a question expression is disentangled to solid ways, just comparing components need to be checked. In addition, such procedures encourage pressure [3], [16], as it has been watched that values under the same root-to-hub ways are comparative.

III. Strategies

A. STORED

It is considered as a declarative query language for determining storage mapping from semi-structured data model to relational model. It includes mapping between the semi structured information model and the social information model, communicated in a query language called STORED. Here the system consequently creates the overflow mapping essential to assure that any semi structured instance is stored losslessly. The advantage of this mapping is that it updates insertions automatically. The insertions in the semi structured data can be automatically rewritten as insertions into the relational and overflow stores [11]. An algorithm was also developed in order to generate automatic STORED overflow mapping for a given relational mapping, so that it can abuse a DTD.



Taxpayer1

oid	name	street	no	apt	zip	audit1	audit2	taxamount	taxevasion
o24	Gluschko	Tyuratam		2C	07099	10/12/63		12332	
o21	Kosberg	Tyuratam	206		92443	11/1/68	10/12/77	0	likely

Taxpayer2

oid	name	address	audited	taxamount	taxevasion
o20	Korolev	Baikonur	10/12/86	0	likely

Company

name	owner
Rocket Propulsion Inc.	o24

Fig: 2 Relational storage

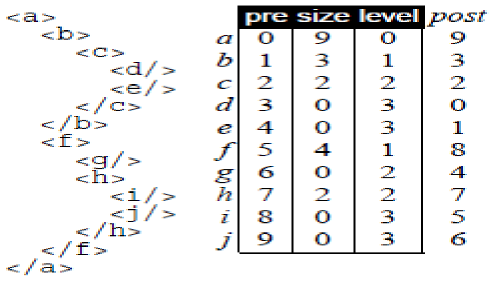
Here we consider the data to be a tree. Object identifiers in the text representation are optional. If no object identifier is specified, an object is assigned a unique identifier automatically. These assumptions are consistent with XML.

B. Normalization mapping

Since an XML document consists of nodes in a nested format, Normalization mapping shred nested structures into standardized tables, where every table relates to one or more components or trait sorts. The nested structure is caught by primary, foreign key connections, so various navigations in XML queries are assessed by essential primary, foreign key joins. Normalization mapping has achieved good query efficiency [17]. The key features of data normalization are it reduces redundancy and avoid update peculiarities.

C. Node-encoding mapping

It described a relational encoding of XML fragments that is a true isomorphism with respect to the tree structure. The encoding is based on preorder and postorder traversal ranks to encode the XML tree structure. Here, we use an equivalent encoding variant in which the location of a node v in the document tree is represented as the triple hpre(v); size(v); level(v)i, recording v's preorder rank, the number of nodes in the subtree below v, and the distance from the tree's root, respectively. (From this, the postorder rank may be recovered via post(v) = pre(v)+size(v)□ level(v) [9]. The preorder rank pre (v) simultaneously serves as a node identifier. XML fragment and the encoding we assign to this fragment. The system maintains further tables to capture more node properties (e.g., tag name, node kind, text content), Our tree representation exhibits a number of useful characteristics, among which element (or tree) construction through pasting of encodings and highly efficient XPath processing with staircase join are especially relevant in the XQuery context.



```
<a>
  <b>
    <c>
      <d/>
      <e/>
    </c>
  </b>
  <f>
    <g/>
    <h>
      <i/>
      <j/>
    </h>
  </f>
</a>
```

	pre	size	level	*post*
a	0	9	0	9
b	1	3	1	3
c	2	2	2	2
d	3	0	3	0
e	4	0	3	1
f	5	4	1	8
g	6	0	2	4
h	7	2	2	7
i	8	0	3	5
j	9	0	3	6

Fig: 2 XML Node Encoding

Another XML database framework based absolutely on social database innovation has been depicted. It gives a full XQuery usage supporting both archive and arrangement request (counting such highlights as XQuery modules and recursive client characterized capacities). We have additionally delineated a redesign plan that addresses the troublesome assignment of proficiently keeping up a range based XML numbering plan under basic upgrades. The primary thoughts behind this component are a page-wise indirection plan to limit the effect of hub movements to a solitary consistent page, and the utilization of deltas to record changes in the size property of XML tree hubs, which abstains from locking the root hub amid the whole exchange.

IV. Conclusion

The main objective of the review paper was to throw some light on the mapping strategies. We also discussed the various concepts and their strengths and weaknesses associated. Different authors presented different techniques for the purpose of mapping. They also showed different ways for storing XML data and how one approach is better than the other.

Acknowledgement

I express my sincere thanks to Chetan J. Awati whose supervision, inspiration and valuable guidance helped me a lot to complete my review paper. His guidance proved to be the most valuable to overcome all the hurdles in the fulfilment of this paper.

References

1. L. J. Chen et al. "Mapping XML to a wide sparse table," in Proc.IEEE VOL.26. NO. 6. JUNE 2014.
2. S. Amer-Yahia, S. Cho, L. V. S. Lakshmanan, and D. Srivastava, Minimization of tree pattern queries," in *Proc. SIGMOD*, Santa Barbara, CA, USA, 2001.
3. A. Arion, A. Bonifati, G. Costa, S. D'Aguanno, I. Manolescu, and A. Pugliese, "Efficient query evaluation over compressed XML data," in *Proc. EDBT*, Heraklion, Greece, 2004.
4. A. Arion, A. Bonifati, I. Manolescu, and A. Pugliese, "Path summaries and path partitioning in modern XML databases," *World Wide Web*, vol. 11, no. 1, pp. 117–151, 2008.
5. A. Balmin, K. S. Beyer, F. Özcan, and M. Nicola, "On the path to efficient XML queries," in *Proc. VLDB*, Seoul, Korea, 2006.
6. A. Barta, M. P. Consens, and A. O. Mendelzon, "Benefits of path summaries in an XML query optimizer supporting multiple access methods," in *Proc. 31st Int. Conf. VLDB*, 2005.
7. K. S. Beyer *et al*. "System RX: One part relational, one part XML," in *Proc. SIGMOD*, Baltimore, MD, USA, 2005.
8. P. Bohannon, J. Freire, P. Roy, and J. Siméon, "From XML schema to relations: A cost-based approach to XML storage," in *Proc. 18th ICDE*, San Jose, CA, USA, 2002.
9. P. A. Boncz, T. Grust, M. van Keulen, S. Manegold, J. Rittinger, and J. Teubner, "MonetDB/XQuery: A fast XQuery processor powered by a relational engine," in *Proc. SIGMOD*, Chicago, IL, USA, 2006.
10. Z. Chen, H. V. Jagadish, L. V. S. Lakshmanan, and S. Paparizos, "From tree patterns to generalized tree patterns: On efficient evaluation of XQuery," in *Proc. 29th Int. Conf. VLDB*, Berlin, Germany, 2003.
11. A. Deutsch, M. F. Fernández, and D. Suciu, "Storing semistructured data with STORED," in *Proc. SIGMOD*, New York, NY, USA,1999.
12. G. H. L. Fletcher, D. V. Gucht, Y. Wu, M. Gyssens, S. Brenes, and J. Paredaens, "A methodology for coupling fragments of XPath with structural indexes for XML documents," *Inf. Syst.*, vol. 34, no. 7, pp. 657–670, Nov. 2009.
13. D. Florescu and D. Kossmann, "Storing and querying XML data using an RDMBS," *IEEE Data Eng. Bull.*, vol. 22, no. 3, pp. 27–34, 1999.
14. H. Georgiadis and V. Vassalos, "XPath on steroids: Exploiting relational engines for XPath performance," in *Proc. SIGMOD*, Beijing, China, 2007.
15. R. Goldman and J. Widom, "DataGuides: Enabling query formulation and optimization in semistructured databases," in *Proc. 23rd Int. Conf. VLDB*, San Francisco, CA, USA, 1997.
16. H. Liefke and D. Suciu, "XMill: An efficient compressor for XML data," in *Proc. SIGMOD*, Dallas, TX, USA, 2000.
17. Z. H. Liu and R. Murthy, "A decade of XML data management: An industrial experience report from oracle," in *Proc. IEEE ICDE*, Shanghai, China, 2009.
18. J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. DeWitt, and J. F. Naughton, "Relational databases for querying XML documents: Limitations and opportunities," in *Proc. 25th Int. Conf. VLDB*, Edinburgh, Scotland, U.K., 1999.
19. I. Tatarinov, S. Viglas, K. S. Beyer, J. Shanmugasundaram, E. J. Shekita, and C. Zhang. "Storing and querying ordered XML using a relational database system," in *Proc. SIGMOD*, Madison, WI, USA, 2002.

AUTHOR(S) PROFILE

Sonam S. Kharade received the B.E. degree in Computer Science & Engineering from Bharati Vidyapeeth College of Engineering, Kolhapur in 2010. Currently, she is pursuing M.Tech. in Computer Science and Technology in Department of Technology, Shivaji University, Kolhapur.

Chetan J. Awati received the M.Tech. degree in Computer Science and Technology from Department of Technology, Shivaji University, Kolhapur in 2011. Currently, he is an Assistant Professor in Department of Technology, Shivaji University, Kolhapur.