**XRecursive:** Connecting XML with Relational Databases

Mohammed Adam Ibrahim Fakharaldien, Universiti Malaysia Pahang, Malaysia

Jasni Mohamed Zain, Universiti Malaysia Pahang, Malaysia

Norrozila Sulaiman, Universiti Malaysia Pahang, Malaysia

Tutut Herawan, Universiti Malaysia Pahang, Malaysia

**ABSTRACT**

Storing XML documents in a relational database is a promising solution because relational databases are mature and scale very well. They have the advantages that in a relational database XML data and structured data can coexist making it possible to build application that involve both kinds of data with little extra effort. This paper proposes an alternative method named Xrecursive for mapping XML (eXtensible Markup Language) documents to RDB (Relational Databases). The Xrecursive method does not need a DTD (Document Text Definition) or XML schema. Further, it can be applied as a general solution for any XML data. The steps and algorithm of Xrecursive are given in details to describe how to use the storing structure to storage and query XML documents in relational database. The authors report their experimental results on a real database, showing that the performance of their Xrecursive algorithm achieves better results in terms of storage size, insertion time, mapping time, and reconstruction time as compared with the SUCXENT and XParent methods. In overall, Xrecursive performs better in term of query performances as compared to the both methods.

Keywords: Databases, eXtensible Markup Language (XML), Mapping, Relational Database (RDB), Structured Data, Structured Query Language (SQL)

**INTRODUCTION**

The eXtensible Markup Language (XML) is used for representing data through the internet, but this technology needs a suitable medium for storing these data. At present, three common technologies can be used to store and retrieve XML documents, i.e., native XML database, Object Oriented Database (OODB) and Relational Database (RDB). With the trend of increasing amount of XML documents on the World Wide Web, it is critical to have efficient mechanism to store and query XML documents to exploit the full power of this technology. Now day’ XML emerged as a major standard for representing data on the World Wide Web while the dominant storage mechanism for structured data is the relational databases, which has been an efficient tool for storing, searching, retrieving data from different collection of data. The ability to map XML data in relational databases is difficult mission and challenging in the world of all IT organization so there is a need to develop an
interfaces and tools for mapping and storing XML data in relational databases.

The extensible Markup Language (XML) is quickly becoming the de facto standard for data exchange over the Internet (Zafari, Hasami, & Shiri, 2010) and now it plays a central role in data management, transformation and exchange. Since its introduction to industry in the late 1990s, XML (Grandi, Mandreoli, Tiberio, & Bergonzini, 2003) has achieved widespread support and adoption among all the leading software tools, server, and database vendors. As importantly, XML has become the lingua franca for data by lowering the cost of processing, searching, exchanging, and re-using information. XML provides a standardized, self-describing means for expressing information in a way that is readable by humans and easily verified, transformed, and published, the hot topic is to seek the best way for storing XML documents in order to get high query processing efficiency (Sainan, Caifeng, & Liming, 2009). In addition, data can be transmitted to remote services anywhere on the Internet using XML-based Web services to take advantage of the new ubiquity of connected software applications. The openness of XML (Augeri, Bulutoglu, Mullins, Baldwin, & Baird, 2007) allows it to be exchanged between virtually any hardware, software, or operating system. Simply put, XML opens the door for information interchange without restriction. Today, the dominant storage mechanism for structured enterprise data is the relational database, which has proven itself an efficient tool for storing, searching for, and retrieving information from massive collections of data. Relational databases specialize in relating individual data records grouped by type in tables. Developers can join records together as needed using SQL (Structured Query Language) and present one or more records to end-users as meaningful information. The relational database model revolutionized enterprise data storage with its simplicity, efficiency, and cost effectiveness. Relational databases have been prevalent in large corporations since the 1980s, and they will likely remain the dominant storage mechanism for enterprise data in the foreseeable future. Despite these strengths, relational databases lack the flexibility to seamlessly integrate with other systems, since this was not historically a requirement of the database model (Reed, 2008). In addition, although relational databases share many similarities, there are enough differences between the major commercial implementations to make developing applications to integrate multiple products difficult. Among the challenges are differences in data types, varying levels of conformance to the SQL standard, proprietary extensions to SQL, and so on. For the storage of XML document, the key issue is transforming the tree structure of an XML document into tuples in relational tables (Yue, Ren, & Qian, 2008). Nowadays, there are more and more data presented as XML document, the need of storing them persistently in a database has increased rapidly while the native–XML databases usually have limited support for relational databases. In recent years, with the popularity of relational databases (RDB), approaches based on RDB (Sybase Corporation, 1999; Yoshikawa, Amagasa, Shimura, & Uemura, 2001; Jiang, Lu, Wang, & Yu, 2002a, 2002b; Kyung-Soo, 2001; Rys, 2000) to store and manipulate XML data as relational tables but still there is need to manage XML data and relational data seamlessly with similar storage and retrieval efficiencies simultaneously. XML and Relational databases cannot be kept separately because XML is becoming the universal standard data format for the representation and exchanging the information whereas most existing data lies in RDBMS and their power of data capabilities cannot be degraded so the solution to this problem a new efficient methods for storing XML documents in relational database is required. A new efficient method for storing XML document in relational database is proposed in this paper to face these problems.

The rest of this paper is organized as follows. First, we review the XML databases. Then, we describe the proposed XRecursive algorithm. Afterwards, we describe the experimental and comparison results. Finally the conclusion of this work is described.
Related Content

The Data Swarm: A Next Step for Distributed Data Analytics
Jeffrey Smith and Manjeet Rege (2016). *International Journal of Information Retrieval Research* (pp. 52-64).
[www.irma-international.org/article/the-data-swarm/142825/](www.irma-international.org/article/the-data-swarm/142825/)

The Impact of the Mode of Data Representation for the Result Quality of the Detection and Filtering of Spam

The Description and Relation of WS-CDL and BPEL
[www.irma-international.org/chapter/description-relation-cdl-bpel/69154/](www.irma-international.org/chapter/description-relation-cdl-bpel/69154/)

Self-Monitoring Scale
[www.irma-international.org/chapter/self-monitoring-scale/69737/](www.irma-international.org/chapter/self-monitoring-scale/69737/)

A Hybrid Classification Approach Based on Decision Tree and Naïve Bays Methods
[www.irma-international.org/article/a-hybrid-classification-approach-based-on-decision-tree-and-nave-bays-methods/127364/](www.irma-international.org/article/a-hybrid-classification-approach-based-on-decision-tree-and-nave-bays-methods/127364/)