Chapter 2

Chapter 1

XML Data Management in Object Relational Database Systems

Zhen Hua Liu
Oracle Corporation, USA

Anguel Novoselsky
Oracle Corporation, USA

Vikas Arora
Oracle Corporation, USA

ABSTRACT

Since the advent of XML, there has been significant research into integrating XML data management with Relational DBMS and Object Relational DBMS (ORDBMS). This chapter describes the XML data management capabilities in ORDBMS, various design approaches and implementation techniques to support these capabilities, as well as the pros and cons of each design and implementation approach. Key topics such as XML storage, XML Indexing, XQuery and SQL/XML processing, are discussed in depth presenting both academic and industrial research work in these areas.

INTRODUCTION

Ever since XML has been introduced, a primary challenge associated with XML has been to persist and query XML content. As a result, there has been significant research into integrating XML data management in relational and object-relational databases. From a user point of view, it is desirable to have integrated access to both relational content and XML content in one platform and to leverage the full breadth of mature database technology to manage XML data.

There are two aspects of integrating XML data management into ORDBMS. One aspect is related to the XML storage and query processing in the database. Various techniques have been developed for storing XML in an ORDBMS including decomposed relational storage form, ***aggregated storage*** form, such as storing XML in a tree representation, a BLOB (Binary Large Object), or a CLOB (Character Large Object). Efficient XML query and

DOI: doi
DOI: 10.4018/978-1-61520-727-5.ch002

update techniques for each form have been studied extensively.

A second aspect is to provide capabilities to facilitate interoperability between the Relational and XML data models. The SQL/XML standard plays a key role in this by providing users complete relational and XML duality - the capability of generating XML from relational data so that XQuery/XPath/XSLT can be used to query relational data, the capability of providing relational access over XML data so that SQL can be used to query XML data, and the capability of embedding XQuery/XSLT and SQL in each other to query and update both XML and relational data together.

In this chapter, we discuss the rationale and benefits of XML data management in ORDBMS, its capabilities of managing both XML and relational data in one platform, the industrial XML applications built by leveraging these capabilities, the design and implementation techniques to support such capabilities effectively and efficiently. We analyze different design approaches from the wealth of the literature in XML processing in the last decade. To conclude, we present the current ongoing and future work in XML data management in ORDBMS.

1 RATIONALE OF XML DATA MANAGEMENT IN ORDBMS

In this section we outline the motivation of building XML data management capabilities in ORDBMS from the perspective of user, from the perspective of data modeling and from the perspective of data engineering.

1.1 User Perspective

There is a general trend for users to increasingly prefer to manage all of their data in one platform for the benefits of easier data management, integrated query services, and sharing a common infrastructure for all types of data. This is more so with relational databases that are widely adopted as mature platforms for managing structured data. With data extensibility mechanisms, such as user defined types, user defined functions and extensible indexes that are added into relational databases, the RDBMS has evolved into an ORDBMS with object-relational support (Stonebraker, Brown & Moore, 1999) that is capable of managing more complex data that may not fit into the relational model. Using an ORDBMS, users are able to manage various kinds of non-relational data such as object hierarchies in one platform. XML data management is a continuation of this trend with users looking for a single data management platform to avoid the overheads of partitioning and integrating data across multiple repositories. Major database vendors such as Oracle, IBM and Microsoft, in recognition of this trend have released XML capabilities in their products to support industrial strength applications that span all forms of data including relational data and XML data.

The other aspect is that XML has been used as a data exchange language to exchange data. Organizations have used XML as a portable way of exchanging data, in particular, exchanging data from and to relational systems with well-defined XML schema. This is associated with the need to generate XML from relational data at the data sending side and the need to extract relational data from XML at the data receiving side. To support this, it is critical for an ORDBMS to provide support for bridging the XML and relational data models.

1.2 Data Model Analysis Perspective

Another motivation for integrating XML data management in an ORDBMS arises from the perspective of data modeling. Although XML is based on a document centric hierarchical tree based data model, XML data in practice has been broadly classified into *document centric XML* and *data centric XML*. There also exists a wide spectrum