A Semi-Structured Data Cartridge for Relational Databases

Fei Sha         Georges Gardarin         Laurent Némirovski

Laboratoire PRiSM, Université de Versailles St-Quentin
45 avenue des Etats-Unis, 78035 Versailles, France
E-mail: firstname.lastname@prism.uvsq.fr

Abstract

In this paper, we present our experiences on managing semi-structured data in relational databases. We use a Java package built on top of relational database for store and query semi-structured data. In this work, we introduced a more complete data model, a flexible storage scheme, and a compact metedata structure for processing regular path expressions.

1. Introduction

Research on semi-structured data is a hot topic in recent database research. In this paper, we present our experiences on managing semi-structured data in relational databases.

2. Data model

Our data model is a labeled directed graph model. In this model, a composite object is represented by an internal node and an atomic object or a value is represented by a leaf node. Nodes are connected by edges. Each edge has a source node, a label, a link type, and a target node. Link type describes the nature of an edge. This is particularly useful for modeling XML attribute and reference. An order is introduced among edges originated from same node for handling ordered collections.

3. Implementation

For storing semi-structured data in relational databases, we introduce a flexible storage scheme. A link table is used to maintain the graph structures and one or more value tables are used for storing values and atomic objects. This storage scheme does not make any assumption or restriction on data to be stored in the system. A detailed description of the storage scheme is given in [1].

4. Longest word language

The longest word language is a metadata we introduced for pruning regular path expressions. The longest word language takes all labels occurred in the database as letters of its alphabet. A word formed of a sequence of letters corresponds to a data path in the database. In the LWL, only the words corresponding to longest path in the database are retained. This allows the language to have a very compact size. When a regular path expression appears in a query, a patterns matching is performed in the LWL to find all data paths qualified by the regular path expression.

The LWL reflects exact structures of data stored in the database. When data are added or removed from the database, the LWL is updated accordingly. For this, an incremental update mechanism is proposed.

5. Conclusion

In our experience, we elaborated a new approach for managing semi-structured data in relational databases. In our system, we introduced a rich data model that allows modeling XML attributes and references and ordered collections, a flexible storage scheme that allows storing irregular data without restriction, and a metadata structire, the longest word language, for processing regular path expressions. An integrated demonstration is described in [2].

References
