RDF(S) Store in Object-Relational Databases

Zongmin Ma, Nanjing University of Aeronautics and Astronautics, China

https://orcid.org/0000-0001-7780-6473

Daiyi Li, Nanjing University of Aeronautics and Astronautics, China

Jiawen Lu, Nanjing University of Aeronautics and Astronautics, China

Ruizhe Ma, University of Massachusetts, Lowell, USA*

Li Yan, Nanjing University of Aeronautics and Astronautics, China

ABSTRACT

The Resource Description Framework (RDF) and RDF Schema (RDFS) recommended by World Wide Web Consortium (W3C) provide a flexible model for semantically representing data on the web. With the widespread acceptance of RDF(S) (RDF and RDFS for short), a large number of RDF(S) is available. Databases play an important role in managing RDF(S). However, there are few studies on using object-relational databases to store RDF(S). In this paper, the authors propose the formal definitions of RDF(S) model and object-relational databases model. Then they introduce the approach for storing RDF(S) in object-relational databases based on the formal definitions. They implement a prototype system to demonstrate the feasibility of the approach and test the performance and semantic retention ability of this prototype system with the benchmark dataset.

KEYWORDS

Object-Relational Databases, PostgreSQ, RDF(S), Storage

1. INTRODUCTION

The Semantic Web has been proposed by Tim Berners-Lee to provide a common framework for information sharing across multiple domains (Crasso *et al*., 2012). With the Semantic Web, data are provided with data semantic meaning (through metadata), and concepts and entities in the real world can be represented in a machine-readable and structured form. The Resource Description Framework (RDF) proposed by the World Wide Web Consortium (W3C) is a model of representing metadata of resources on the Web. RDF Schema (RDF(S)) as well as Web Ontology Languages (OWL) are the description of vocabulary semantics used in RDF datasets. RDF and RDF Schema (collectively known as RDF(S)) are the core of the Semantic Web. Nowadays, RDF(S) have been increasingly applied

*Corresponding Author

in a wide range of Web-based application scenarios, such as semantic data integration (Arsic *et al.*, 2019), semantic search (Xiong, Power and Callan, 2017; Zheng *et al.*, 2019), semantic analysis of Big Data (Smiatacz, 2018; Shen, Hu and Tzeng, 2017), decision making (Rubio-Largo *et al.*, 2017; Zhou *et al.*, 2017) and so on. Currently, RDF(S) has become the de-facto standard of representing and handling data semantics. In particular, knowledge graphs (KGs) mostly adopt RDF mode to represent massive instances, and now are widely investigated and applied in diverse domains for the semantic and intelligent processing of massive data (Song *et al.*, 2019).

With the rapid increase in the number of RDF(S) on the Web, it has become increasingly important to efficiently store massive amounts of RDF(S). The storage of RDF(S) (Ma, Capretz and Yan, 2016) often supports efficient queries of RDF data, mainly because the storage structure of RDF(S) not only directly determines the integrity of storage semantics, but also greatly affects its query efficiency (Ma *et al.*, 2016; Ma, *et al*, 2018). At present, there have been many studies on RDF(S) storage methods, which can be roughly divided into the following three categories:

1) Memory-based storage (e.g., Sesame (Broekstra, *et al.*, 2002) and BitMat (Atre, *et al.*, 2008)). With this category of methods, memory space is directly allocated for RDF data and indexing technology is generally utilized for quick data process. Note that these methods are limited by the size of computer memory and are only suitable for storing a small number of RDF datasets;

2) Disk-based storage (e.g., YARS2 (Harth, *et al.*, 2007) and System II (Wu, *et al.*, 2009)). With this category of methods, the storage location is transferred from memory to hard disk. These methods meet the storage requirements of large-scale RDF datasets in space, but frequent reads and writes to disks greatly reduce storage performance;

3) Database-based storage (e.g., Jena-TDB (Wilkinson, *et al.*, 2003), 4Store (Harris, *et al.*, 2009), Virtuoso (Erling and Mikhailov, 2007), BigOWLIM/OWLIM-SE (Bishop *et al.*, 2011), SPARQLcity/SPARQLverse[1], MarkLogic[2], and Clark and Parsia[3]). This category of methods uses database technology to store RDF data. In addition to some commercial systems, there are some developed prototypes such as RDF-3X (Neumann and Weikum, 2010), SW-Store (Abadi, *et al.*, 2009) and RDFox[4].

With the mature technology and powerful data management capability of relational databases (RDBs), research on RDF(S) storage methods based on RDB have achieved some results (Ma, Capretz and Yan, 2016; Fan, Yan and Ma, 2020). However, due to the use of a two-dimensional table storage structure at the bottom of the RDB, which does not match the structure of RDF(S), the RDF(S) storage methods based on RDBs cannot effectively store the semantic information of RDF(S), resulting in incomplete storage data semantics and low query efficiency. To store massive RDF data, NoSQL (not only SQL) databases are applied to store RDF data (Cudre-Moroux *et al.*, 2013). However, although NoSQL-based storage is highly efficient for massive RDF(S) data, it is recognized that there is no unified standard for NoSQL databases, and different databases use different query languages, each with its own advantages and disadvantages (Edwards, 2022). Considering the cost, familiarity and technical maturity, for the storage of non-massive RDF data, traditional databases rather than NoSQL databases are still the first choice due to the mature theoretical basis and powerful data management capabilities.

Object-relational databases (ORDBs) are based on RDBs and combine object-oriented features. Therefore, they not only support the integrity constraints and SQL standards of RDBs, but also utilize object-oriented features to handle complex data relationships. At present, although ORDBs are widely applied in various domains such as geographic information management (Ackere *et al.* 2019), software engineering (Gregory, 2019), multimedia (Khanduja and Chkraverty, 2019) and other fields, there is relatively little research on using ORDBs to store RDF(S). In (Alexali *et al.*, 2001), a tool called RDF Suite was implemented based on an ORDB, which can be used for RDF(S) storage and querying. This RDF(S) storage method achieves the separation of RDF(S) schema information

Related Content

On the Representation of Temporal Dynamics
Salvatore T. Marchand Gove N. Allen (2003). *Advanced Topics in Database Research, Volume 2 (pp. 37-53).*
www.irma-international.org/chapter/representation-temporal-dynamics/4340

Designing Information Systems Capabilities to Create Business Value: A Theoretical Conceptualization of the Role of Flexibility and Integration
Christoph Schlueter Langdon (2006). *Journal of Database Management (pp. 1-18).*
www.irma-international.org/article/designing-information-systems-capabilities-create/3355

The Graph Traversal Pattern
Marko A. Rodriguezand Peter Neubauer (2012). *Graph Data Management: Techniques and Applications (pp. 29-46).*
www.irma-international.org/chapter/graph-traversal-pattern/58605

Electronic Tools for Online Assessments: An Illustrative Case Study from Teacher Education
Jon Margerum-Leysand Kristin M. Bass (2009). *Database Technologies: Concepts, Methodologies, Tools, and Applications (pp. 1291-1308).*
www.irma-international.org/chapter/electronic-tools-online-assessments/7973

Representing Classes of Things and Properties in General in Conceptual Modelling: An Empirical Evaluation
Graeme Shanks, Daniel Moody, Jasmina Nuredini, Daniel Tobinand Ron Weber (2010). *Journal of Database Management (pp. 1-25).*
www.irma-international.org/article/representing-classes-things-properties-general/42083