doi:10.3969/j.issn.1005-3026.2015.03.010

RDF在关系数据库中的存储研究

佟 强[1],张 富[2],程经纬[2],马宗民[2]
(1.东北大学 软件学院,辽宁 沈阳 110819;2.东北大学 信息科学与工程学院,辽宁 沈阳 110819)

摘 要:资源描述框架(RDF)是Web资源信息的规范性描述语言,如何存储RDF数据成为当前重要的研究问题.通过深入分析RDF数据的特点,提出了一种新的基于关系数据库的RDF存储模式,给出了RDF在关系数据库中的存储规则,提供了相应的存储实例.基于提出的存储方法,实现了相应的自动存储原型系统,通过实验进一步验证存储方法和原型系统的可行性,并与已有存储模式进行了理论对比分析.

关 键 词:语义网;资源描述框架(RDF);关系数据库;存储;信息保持
中图分类号:TP 301 **文献标志码**:A **文章编号**:1005-3026(2015)03-0346-04

Research on Storage of RDF in Relational Database

TONG Qiang[1], *ZHANG Fu*[2], *CHENG Jing-wei*[2], *MA Zong-min*[2]
(1. School of Software, Northeastern University, Shenyang 110819, China; 2. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: ZHANG Fu, E-mail: zhangfu@ise.neu.edu.cn)

Abstract: Resource description framework (RDF) is a normative language to describe the Web resource information. How to store RDF data is becoming an important research issue. By analyzing the characteristics of RDF data, a storage model of RDF was proposed based on relational databases. The storage rules and a storage example were given. On the basis of the proposed storage approach, a prototype storage tool was implemented, and the experiments shown that the approach and the tool are feasible. Also, the theory comparative analyses with the existing storage modules were done.

Key words: semantic Web; resource description framework (RDF); relational database; storage; information preserving

资源描述框架(resource description framework,RDF)是由万维网联盟W3C提出的用于描述Web资源信息的规范标准语言[1].常见的RDF数据存储模式包括:基于文件形式、基于专门的存储形式以及基于数据库存储[2].基于数据库存储方式是将RDF数据存储在数据库中[3-4].

在上述三类RDF数据存储方法中,基于关系数据库多年来具有较为成熟的理论和技术支持,利用关系数据库进行存储成为解决RDF数据存储的有效手段.目前,常见的基于关系数据库的RDF数据存储模式大致包括:水平模式[5-6]、通用/垂直模式[7],以及专有/二元模式[8].然而,基于水平模式进行存储时,可能导致表中空值较多,造成空间浪费且增加了维护和查询的代价;基于通用/垂直模式设计简单且模式稳定,但是对于每个查询都必须搜索整个数据库,影响查询效率,并且设计的相应SQL查询语句复杂且容易出错;基于专有/二元模式进行存储时,在RDF类和属性较多的情况下,需要创建很多表,而且随着RDF数据的变化需要动态创建和删除数据库中的表.

为此,本文通过深入分析RDF数据的特点,提出一种新的基于关系数据库的RDF存储模式,给出RDF在关系数据库中详细的存储规则,提供了相应的存储实例.基于提出的存储方法,实现了

收稿日期:2014-07-02
基金项目:国家自然科学基金资助项目(61073139,61202260,61370075);辽宁省教育厅科学研究一般项目(L2013098);中央高校基本科研业务费专项资金资助项目(N140404010,N140404005).
作者简介:佟 强(1975-),男,辽宁锦州人,东北大学讲师,博士研究生;马宗民(1965-),男,山东金乡人,东北大学教授,博士生导师.

相应的自动存储原型系统，通过实验进一步验证了存储方法和原型系统的可行性，并与已有存储模式进行理论对比分析.

1　基于关系数据库的 RDF 存储方法

通过分析 RDF 实例数据的特点，提出一种基于关系数据库的 RDF 存储模式.

1.1　RDF 的形式化表示

资源描述框架的基本思想是：所有信息都被称为“资源”；被描述的“资源”具有一些“属性”，而这些属性各有其“值”；对资源的描述可以通过“陈述”来进行. RDF 采用三元组陈述 < 主体、谓词、客体 > 来描述 Web 上的资源. 下面给出 RDF 的形式化定义.

定义 1（RDF）　一个 RDF 数据集 R 可以表示为 $R=(R_S, R_T)$，其中：R_S 是一个标识符集，包括类资源标识符集合 C，属性资源标识符集合 P，数据类型标识符集合 D，以及实例资源标识符集合 T；R_T 是一个定义在标识符集 R_S 上的三元组陈述集合.

1.2　基于关系数据库的 RDF 存储方法

本节提出一种基于关系数据库的 RDF 存储模式. 给定一个 RDF 模型 $R=(R_S, R_T)$，下面的规则 1 ~6 说明了 RDF 模型 R 在关系数据库中的存储方法.

规则 1（存储 RDF 类资源）：给定 RDF 模型 R 中的类资源标识符集合 $C \in R_S$，为每个类 $c \in C$ 创建相应的类关系表 Class_Table.

规则 2（存储 RDF 文字类型属性资源）：给定 RDF 模型 R 中的属性资源标识符集合 $P \in R_S$ 以及相应的属性三元组描述 R_T，如果属性 $p \in P$ 的值为文字类型值（Literal）时（例如字符串），将该属性映射为相应类表中的一个属性列.

说明：属性 $p \in P$ 的 RDF 数据类型相对应的 SQL 数据类型，作为表中这一属性列的类型. 例如：一个属性 Age 的数据类型为 xsd：positiveInteger，然而在 SQL 中没有 positiveInteger，因此，映射后的 Age 属性列用 INTEGER 作为它的类型，并且加上 CHECK 限制：CHECK（Age > 0）.

规则 3（存储 RDF 非文字类型属性资源）：给定 RDF 模型 R 中的属性资源标识符集合 $P \in R_S$ 以及相应的属性三元组描述 R_T，如果属性 $p \in P$ 的值为资源时，将该属性映射为相应类表中的一个外键属性列.

规则 4（存储 RDF 个体实例资源）：给定 RDF 模型 R 中的个体实例资源标识符集合 $T \in R_S$ 以及相应的属性三元组描述 R_T，将每个个体 $i \in T$ 映射为相应类表中的一个元组.

规则 5（存储 RDF 多值属性）：给定 RDF 模型 R 中的属性标识符集合 $P \in R_S$，如果属性 $p \in P$ 为多值属性时，创建相应的多值属性表 Multi_Pro_Table.

说明：Multi_Pro_Table 包含两个字段（ProID 和 Value），分别用于存储多值属性以及相应的值，其中，ProID 取值来自于规则 2 和规则 3 中该属性 p 相应的属性列的取值.

规则 6（存储 RDF 数据类型）：给定 RDF 模型 R 中的数据类型标识符集合 $D \in R_S$，映射为对应的 SQL 数据类型.

基于上述规则，可以将 RDF 存储在关系数据库中. 从存储过程可以看出，存储相当于实现从 RDF 到关系数据库的转换. 目前还没有一种通用的标准方法用于证明两种模式转化的正确性. 基于文献[9]中提到的信息容量保持原理，可类似证明上述存储过程是信息容量保持的存储.

1.3　RDF 存储实例

为了说明上述规则，图 1 给出了一个 RDF 实例. 根据 1.2 节提出的存储规则，可以将图 1 中的 RDF 数据存储在关系数据库中，得到如表 1 所示的关系数据库模式.

表 1　图 1 中的 RDF 在关系数据库中的存储
Table 1　The storage of RDF in Fig. 1 in relational database

No	Relational Schemas
1	Company（cid vchar, cnumber vchar）
2	Department（did vchar, dnumber vchar, total int, belong_to vchar REFERENCES Company）
3	Leader（lid vchar, lnumber vchar, age int, position vchar, manage vchar REFERENCES Department, supervise vchar REFERENCES Employee）
4	Employee（eid vchar, enumber vchar, age vchar, email vchar, work _ in vchar REFERENCES Company）
5	Multi_Pro_Table（ProID vchar, Value vchar）

2　存储原型系统

基于上一节提出的 RDF 在关系数据库中的

存储方法,本节设计和实现了一个相应的存储工具 RDF2RDB. 该系统以 RDF/XML 形式描述的 RDF 数据集作为输入,输出对应的关系数据库模型. 下面简要介绍 RDF2RDB 的总体结构并给出一个例子的运行界面. RDF2RDB 是采用 java 语言在 MyEclipse 7.0 平台上自行开发的工具. 在具体实现过程中,主要使用 JDK1.6.0_02 开发包,利用 java.awt 包及 javax. swing 包进行图形用户界面的实现,并将结果存储在 MySQL 中. RDF2RDB 的总体结构包括:1)解析模块,RDF2RDB 利用 Jena API[7] 和 SPARQL 查询语言[1]对以 RDF/XML 形式描述的 RDF 数据集进行解析;2)存储模块,该模块以解析模块得到的三元组作为输入,首先定义一些分解规则从三元组中提取出类、属性、实例及约束等,然后根据第 1 节中的存储方法将 RDF 数据集中的元素分别映射为关系数据库中对应的表或者记录,从而实现 RDF 在关系数据库中的存储;3)显示模块,生成 SQL 脚本文件.

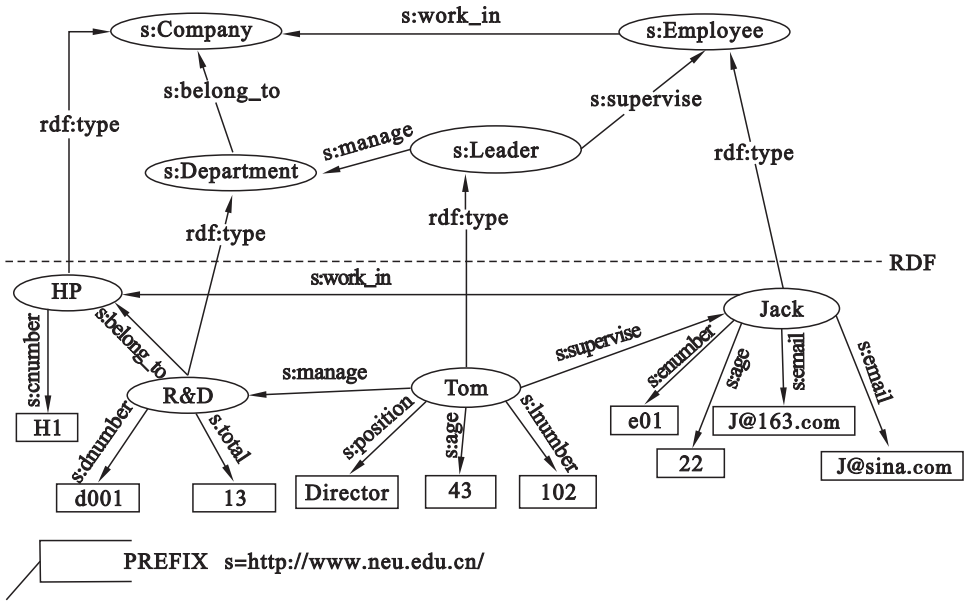

图 1 一个模拟某公司部分信息的 RDF 文档

Fig. 1 A RDF document modeling parts of the reality at a company

为了说明 RDF2RDB 的有效性,下面给出一个例子的 RDF2RDB 运行界面. 给定图 1 中的 RDF 文档,通过 RDF2RDB 可以将其存储在数据库 MySQL 中. 图 2 给出了 RDF2RDB 存储过程的截屏,其中输入的 RDF/XML 语法描述的 RDF 数据集、解析结果以及存储后的结果分别显示在图形界面的左、中和右区域.

从存储结果可以看出,本文提出的存储模式

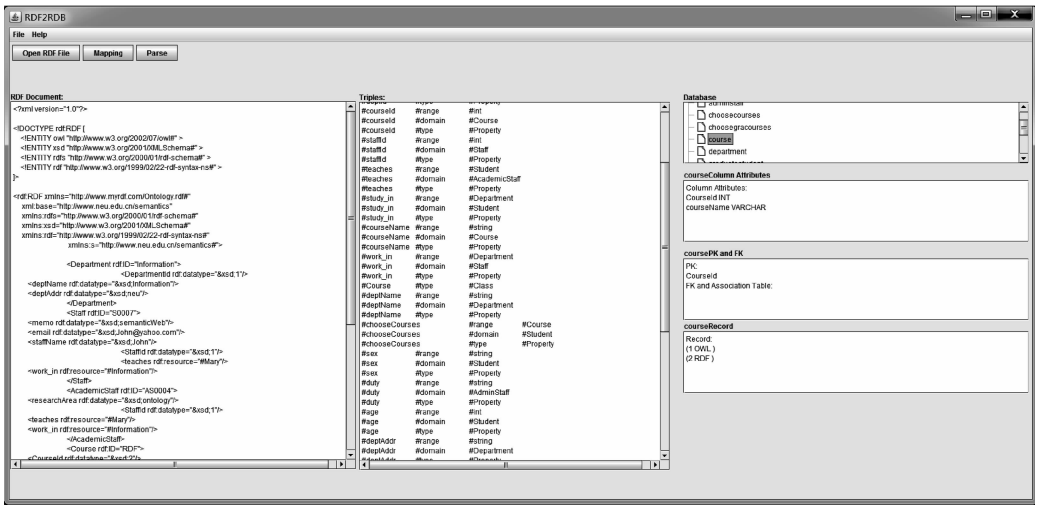

图 2 RDF2RDB 的运行结果

Fig. 2 Screen snapshot of RDF2RDB

和原型系统能够实现 RDF 在关系数据库中的完整存储. 此外,与本文开头部分提到的已有方法相比:1）与水平模式相比,由于水平模式在存储 RDF 时,仅创建 1 张关系表,每个 RDF 属性被映射为表中的一列,RDF 中的每个实例都是该表中的 1 条记录. 该水平模式结构虽然简单,但容易造成表中空值较多,浪费存储空间. 此外,当 RDF 数据量大的时候,由于表中的属性列以及元组太多,增加了维护代价. 再者,每次查询数据库需要搜索整个表的属性和元组,增加了查询代价. 2）与垂直模式相比,垂直模式仅创建 1 张三元组表,表中只包含三个属性列(subject,predicate,object),表中的每一条记录都对应于一个 RDF 三元组. 该模式结构稳定并实现了 RDF 数据的直接存储,但是不能很好地体现 RDF 数据集中的类、属性以及个体等资源之间的关系,同时对于每个查询必须搜索三元组表中的所有记录并进行自连接,影响查询效率,同时设计的相应 SQL 查询语句复杂且容易出错. 3）与二元模式相比,二元模式为 RDF 中的每一个类和属性创建单独的表,而且类表中仅包含一列属性,用于存储属于该类的个体实例. 该模式实现了类和属性的区分,但并未体现类和属性之间的关联. 此外,在 RDF 属性较多的情况下,需要创建很多属性表,占用大量存储空间. 再者,随着 RDF 属性的变化,需要动态创建和删除表,增加了维护的代价. 与上述三种模式相比较,本文提出的存储模式具有如下特点:1）减少了属性表的数量,本文提出的存储模式为每个类创建一个类表,并且将属性和个体实例增加到相应的类表中,进而减少属性表的数量,节约存储空间;2）RDF 资源之间的关系体现明显,在本文提出的存储模式下,实现了 RDF 类、属性以及个体实例等资源的区分,并且考虑了它们之间的关系;3）表结构相对稳定,随着 RDF 属性的变化,只需添加或删除属性所在类表的列属性即可,不需要动态创建和删除表;4）查询自连接减少,在本文提出的存储模式下,查询时只需要查询属性和个体相应的类表即可,减少了查询自连接的数量. 基于以上分析,本文提出的基于关系数据库的 RDF 存储模式及工具,与已有的存储模式相比,能够减少表的数量,进而节约存储空间,同时减少了查询自连接的数量,并充分考虑了 RDF 资源信息之间的关联.

3　结　　语

本文通过深入分析 RDF 的特点,根据属性种类的不同,提出一种新的基于关系数据库的 RDF 存储模式. 给出了详细的存储规则,提供了相应的存储实例. 基于提出的存储方法,实现了相应的自动存储工具,通过实验进一步验证存储方法和原型系统的可行性,并与已有存储模式进行理论对比分析.

进一步的工作包括:深入研究 RDF 查询问题,并与已有的存储模式在查询效率方面进行对比,通过大量实验进一步完善存储框架.

参考文献:

[1] Schreiber G,Raimond Y. RDF 1.1 primer[EB/OL]. [2014 - 02 - 25]. http://www. w3. org/TR/2014/NOTE - rdf11 - primer - 20140225/.

[2] Faye D,Cure O,Blin G. A survey of RDF storage approaches [J]. *ARIMA Journal*,2012,15:11 - 35.

[3] Abadi D J,Marcus A,Madden S R,et al. Scalable semantic web data management using vertical partitioning [C]// Proceedings of the 33rd International Conference on Very Large Data Bases. Vienna:ACM,2007:411 - 422.

[4] Thtoharis Y. Benchmarking database representations of RDF/S store[C]//Proceedings of the 4th International Semantic Web Conference. Galway:Springer,2005:685 - 701.

[5] Agrawal R, Somani A, Xu Y. Storage and querying of ecommerce data[C]//Proceedings of the 27th International Conference on Very Large Data Bases. Rome: ACM, 2002: 149 - 158.

[6] Bornea M A,Dolby J,Kementsietsidis A,et al. Building an efficient RDF store over a relational database [C]// Proceedings of the SIGMOD. New York:ACM,2013:121 - 132.

[7] Apache Jena Project Team. Apache Jena[EB/OL]. [2014 - 12 - 16]. https://jena. apache. org/.

[8] Pan Z,Heflin J. DLDB: extending relational database to support semantic web queries [C]//Proceedings of the International Workshop on Practical and Scalable Semantic Systems. Sanibel Island:IEEE,2003:43 - 48.

[9] Miller R J, Ioannidis Y E, Ramakrishnant R. The use of information capacity in schema integration and translation [C]//Proceedings of the VLDB Endowment. Dublin:ACM, 1993:120 - 133.