计算机科学
COMPUTER SCIENCE

基于关系数据库的时态 RDF 建模究

韩啸, 章哲庆, 严丽

引用本文

韩啸, 章哲庆, 严丽. 基于关系数据库的时态 RDF 建模[J]. 计算机科学, 2022, 49(11): 90-97.

HAN Xiao, ZHANG Zhe-qing, YAN Li. Temporal RDF Modeling Based on Relational Database[J]. Computer Science, 2022, 49(11): 90-97.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.211100065

基于关系数据库的时态 RDF 建模

韩 啸 章哲庆 严 丽
南京航空航天大学计算机科学与技术学院 南京 211106
(han_xiao1996@163.com)

摘 要 随着时态数据的不断增加,时态知识图谱的概念得到了普及,如何高效地表示时态知识图谱已成为一个重要的研究方向。RDF(Resource Description Framework)虽然在传统知识图谱建模中被广泛运用,但其只能表示静态语义,缺乏表示时态知识图谱的能力,因此已有几种针对时态知识图谱的时态 RDF 模型被提出。但这些模型都只是将时态信息简单地附加在谓语或整个三元组上,缺少对时态信息所属对象的准确定位。为了更好地表示时态知识图谱,文中提出了一个新的时态 RDF 表示模型-tRDF。该模型首先根据宾语的不同类型,选择性地将时态信息附加在宾语或谓语上;其次,结合时态数据库的概念,给出了一种基于关系数据库 PostgreSQL 的 tRDF 数据存储方法;最后,从数据存储的时间和空间两个方面对所提出的 tRDF 数据存储方法进行了验证。实验结果表明,所提方案能有效地表示时态知识图谱。

关键词:RDF;时态扩展;时态 RDF;时态知识图谱;时态数据库
中图法分类号 TP399

Temporal RDF Modeling Based on Relational Database

HAN Xiao,ZHANG Zhe-qing and YAN Li
College of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 211106,China

Abstract With the increase of temporal data,the concept of temporal knowledge graph is popularized,and how to represent temporal knowledge graph efficiently has become an important research direction. Although resource description framework(RDF) is widely used in traditional knowledge graph modeling,it can only represent static semantics and lacks the ability to represent temporal knowledge graph. Therefore,several temporal RDF models have been proposed for temporal knowledge graph,but all these models simply attach temporal information to the predicate of RDF or the whole triple,and lack the accurate positioning of the object to which the temporal information belongs. In order to better represent temporal knowledge graph,firstly,this paper proposes a new temporal RDF representation model called tRDF,which attaches temporal information to the object or predicate according to the type of object. Secondly,by combining the concept of temporal database,this paper presents a tRDF data storage method based on the relational database,PostgreSQL. Finally,the proposed tRDF data storage method is verified from two aspects,the time of storing and the size of space. Experimental results show that the proposed scheme can effectively represent temporal knowledge graph.

Keywords RDF,Temporal expansion,Temporal RDF,Temporal knowledge graph,Temporal database

1 引言

谷歌公司于 2012 年正式提出知识图谱的概念,其本质是从文本数据中抽取语义,形成特定的实体以及实体间的关系,从而构成一个直观的、容易理解的关系网络。知识图谱不仅能以图的形式为用户提供可视化的数据管理方式,还具备语义推理能力,极大地提高了搜索的准确性。因知识图谱对关系数据强大的表示和管理能力,它被广泛应用于社交网络[1]以及智能搜索[2]等领域。此外,知识图谱在金融、医疗以及地理信息等领域的知识构建工作中也起到了关键作用[3-4]。RDF 是由 W3C(World Wide Web Consortium)提出的一种用于描述 Web 上的资源以及这些资源之间关系的数据模型。由于 RDF 的语法形式与知识图谱的知识构成十分吻合,因此 RDF 已经成为知识图谱的主要表示形式,并被称为知识图谱的基石之一。随着知识图谱规模的不断扩大,如何存储海量的 RDF 数据成为了研究的重点。在过去的研究中,常用的

到稿日期:2021-11-05 返修日期:2022-02-22
基金项目:江苏省基础研究计划(BK20191274)
This work was supported by the Basic Research Program of Jiangsu Province,China(BK20191274).
通信作者:严丽(yanli@nuaa.edu.cn)

存储方法主要有以下 3 种：基于内存、基于磁盘和基于数据库。在上述 RDF 数据存储方法中，数据库技术凭借其成熟的理论和众多的产品支持，成为了管理 RDF 数据的主要手段。其中，关系型数据库更是占主导地位，它借助 SQL 语句强大的数据管理能力来存储和查询 RDF 数据[5-11]。此外，使用非关系型数据库管理 RDF 数据的技术也在不断地发展[11-13]。

时态属性是描述事物发展过程中动态变化的重要属性，带有时态属性的数据被称为时态数据。随着网络上时态数据的不断增多，对传统知识图谱进行时态补全的研究也在不断深入，时态知识图谱得以进入大众的视野[14-17]。虽然 RDF 可以有效地表示传统知识图谱，但遗憾的是，Web 资源并不是静态不变的，现有的 RDF 模型只能表示静态语义，缺少表示时态信息的能力，因此传统 RDF 模型在表示时态知识图谱时显得力不从心。为了更好地表示时态知识图谱，学术界和工业界都在探索新的时态 RDF 模型。文献[18-19]首次提出了时态 RDF 的概念模型，并给出了模型相应的语法和语义。文献[20]引入了非确定时态三元组的概念，并在前人研究的基础上提出了几种不同的模型。文献[21]为 RDF 三元组添加时态信息，从而将其扩展为四元组形式。目前的大多数研究将时态信息以时间戳的形式直接附加到整个 RDF 三元组之后或 RDF 的谓语上。通过研究发现，当时态 RDF 三元组的宾语为字面量时，时态信息通常表示三元组主语的属性值的有效时间；当时态 RDF 表示两个资源之间的关系时，时态信息则表示谓语的持续时间。因此，现有的时态 RDF 模型虽然可以表示时态知识图谱，但缺乏对时态信息所表示目标的准确区分，对时态知识图谱的表示能力还不够完善。另一方面，随着时态知识图谱规模的扩大，时态 RDF 数据也随之急剧增多。虽然存储传统 RDF 数据的研究已经十分成熟，但是鲜有针对如何存储时态 RDF 数据的研究，目前的工作试图填补这一空白。

本文利用时间标签法对传统 RDF 模型进行时态扩展，提出了一种新的名为 tRDF 的时态 RDF 模型，以便更好地表示时态知识图谱，然后给出了利用关系型数据库存储 tRDF 数据的方法。本文的主要贡献如下：

(1)构造了一个新的时态 RDF 模型，称之为 tRDF，它根据 RDF 三元组的宾语是字面量还是资源，分别向三元组的宾语或谓语部分添加时间戳；给出了 tRDF 数据模型的语法和语义，并通过一个实例更好地解释了 tRDF 模型。

(2)给出了基于关系型数据库 PostgreSQL 的 tRDF 数据的存储规则与实现算法。

(3)对提出的 tRDF 模型存储方法进行实验分析，并与其他存储方法进行对比。实验结果表明，本文提出的 tRDF 模型能准确地表示时态知识图谱，给出的存储方法能有效地存储 tRDF 数据。

本文第 2 节简要概述了时态 RDF 模型、传统 RDF 存储技术以及时态数据库的相关工作；第 3 节提出了一个新的时态 RDF 模型，并给出了相应的存储方案和实现算法；第 4 节通过对比实验对提出的存储方法进行了分析验证；最后总结全文并展望未来。

2　相关工作

2.1　时态 RDF 模型

随着大量时态数据涌入网络，时态知识图谱的概念逐渐进入人们的视野。由于语义网中携带时态信息的数据越来越多，只能描述静态资源的传统 RDF 在表示时态资源时力有不逮。许多学者和从业者开始研究传统 RDF 模型的时态扩展，并提出了多种时态 RDF 模型。经过研究和总结，目前对传统 RDF 模型进行时态扩展的方式主要有以下 3 种。

(1)版本控制法[18-19]：基于快照技术，当 RDF 三元组随时间发生变化时生成新的快照，并将之前的快照保存。

(2)时间标签法[20]：通过给传统 RDF 三元组添加时间戳来实现时态扩展。

(3)三元组扩展法[21]：对 RDF 的语法进行重新定义，添加表示时态信息的内容，如四元组结构。

在上述方法中，时间标签法由于不会破坏 RDF 原有的三元组结构和可扩展性，因此被广泛地用于时态 RDF 的建模中。该方法所涉及的时间概念主要分为以下两类[18]：事务时间和有效时间。事务时间指数据在数据库中存在的时间；有效时间指数据在现实中存在的时间，可以是过去的时间、现在的时间，也可以是未来的时间。有效时间主要有时间点$\{T_1, T_2, \cdots, T_n\}$和时间间隔$[Ts, Te]$两种表现形式。

文献[18-19]采用版本控制的方法首次提出时态 RDF 的概念模型，并给出了该模型相应的语法和语义，但缺少对非确定三元组的分析。文献[20]利用时间标签法，在 RDF 谓语上添加时间戳，提出了新的时态 RDF 模型，并引入了非确定时态三元组的概念。文献[21]利用三元组扩展法提出了带有时态信息的 RDF 四元组结构。近年来，研究者基于时间标签法提出了一些新的时态 RDF 模型。除有效时间外，文献[22]引入了更新次数，从而构建了双时态 RDF；文献[23]提出了一种基于邻域矩阵的时态 RDF 模型，并通过一维编码方式压缩时态信息。此外，针对时态 RDF 查询的研究也愈来愈多。文献[24] 提出了一个获取本体上的时态信息的框架；文献[25]基于本体实现对时态 RDF 数据的查询；文献[26-27]利用图算法对时态 RDF 数据的索引进行优化，从而提高了查询的效率。

2.2　传统 RDF 数据的存储

RDF 是知识图谱的表示形式，对知识图谱的管理实际上就是对 RDF 数据的存储。本节首先简要介绍几种现有的 RDF 数据存储方法，然后对其中最常用的数据库方法做详细的说明。到目前为止，存储 RDF 数据的研究已经较为成熟，主要的存储方法有以下几种。

(1)基于内存：操作系统直接为 RDF 数据分配内存空间，RDF 数据以三元组的形式存储在内存中。

(2)基于磁盘：这种方法类似于内存存储，最大的区别就是存储空间从内存转移到磁盘。

(3)基于数据库:使用数据库技术管理 RDF 数据。

基于内存和基于磁盘的存储方法将 RDF 数据以三元组的形式直接装入内存或以文件的形式存储到本地硬盘上。这两种方法很好地保留了 RDF 数据原始的三元组形式和语义,但前者受文件大小限制,后者的性能则受读写频率的影响。因此,使用数据库存储大量 RDF 数据成为了管理 RDF 数据的主流方法。其中关系型数据库由于技术成熟,数据管理能力强,因此被广泛应用于 RDF 数据存储研究中。RDF-3X[5]将 RDF 三元组都存储在一张表中。4Store[6]和 RDB2RDF[7]根据 RDF 的属性创建不同的表来存储 RDF 数据。文献[8]以 MySQL 数据库为存储后端,优化了 RDF 的存储效率。文献[9]从集中式和分布式等多个角度对现有的 RDF 存储方法进行了比较分析。文献[10]利用对象关系数据库映射 RDF 数据。尽管关系型数据库能有效地存储 RDF 数据,但这种方法的效率会随着数据规模的不断扩大而降低,而非关系型数据库可以有效地解决这个问题,因此非关系型数据库也越来越受到国内外学者的重视。文献[12]利用非关系型数据库中热门的 HBase 来存储大规模 RDF 数据。文献[13]利用图形数据库来存储 RDF 数据。此外,针对模糊领域的 RDF,文献[28-29]给出了详细的基于关系型数据库和非关系型数据库的存储方法。

2.3 时态数据库

近年来,随着时态数据的增加,时态数据库的理论研究也日趋成熟。文献[30]将时态数据库中的时间分类为有效时间、事务时间和自定义时间,其中有效时间和事务时间同前文的介绍,自定义时间指用户根据自身需要而定义的时间,其语义由用户自行解释。时态数据库根据所支持的时间类型可分为 3 种类型[30]:只支持事务时间的数据库被称为回滚数据库;只支持有效时间的数据库被称为历史数据库;同时支持事务时间和有效时间的数据库被称为双时态数据库。

关系数据库普遍利用 SQL(Structured Query Language)来管理数据库中的数据。SQL:2011 发布于 2011 年 12 月,它在 SQL:2008 的基础上增加了对时态数据的支持,提出了创建和操作时态表的最新标准,即能管理一个或多个与时态相关联的表[31]。在 SQL:2011 提出之后,学术界涌现了大量的对传统数据库管理系统进行时态扩展的研究。文献[32]提出了一种新的关系型数据库设计框架,该框架可以支持当前信息和历史信息的访问;文献[33]对分布式数据库管理系统进行了时态扩展;文献[34]针对时态关系数据库中的时间不确定性问题提出了解决方案;文献[35]提出了应用于数据库的时间建模方法。虽然时态数据库的理论研究已经取得了很大的进展,但现在还没有成熟的时态数据库产品问世。

3 时态 RDF 建模及存储

为了更好地表示时态知识图谱,本文使用时间标签法对传统的 RDF 模型进行时态扩展,构建新的时态 RDF 模型-tRDF,并利用关系型数据库 PostgreSQL 存储 tRDF 数据。

3.1 时态 RDF 模型-tRDF

现有的时态 RDF 模型只是将时态信息以时间戳的形式附加在 RDF 三元组的谓语上或者整个三元组之后,缺乏对时态信息所指代对象的准确定位。本文首先根据 SQL:2011 提出的闭-开周期将时态信息修改为左闭右开的格式;其次,根据宾语是资源还是字面量,选择性地将时态信息添加到谓语或宾语上,以便更准确地区分时态信息所指代的目标,实现对传统 RDF 数据的时态扩展。本文将这种新的时态 RDF 模型称为 tRDF。下文将给出一组传统 RDF 和 tRDF 的实例图(受篇幅影响,图中均不显示前缀),来更好地解释所提出的时态 RDF 模型。

首先,传统 RDF 模型的实例如图 1 所示。

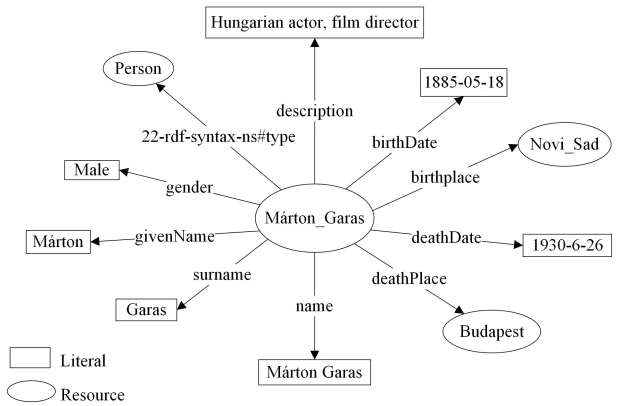

图 1 个人信息 RDF 图

Fig. 1 RDF graph of personal information

经过时态扩展后的 tRDF 模型实例如图 2 所示。

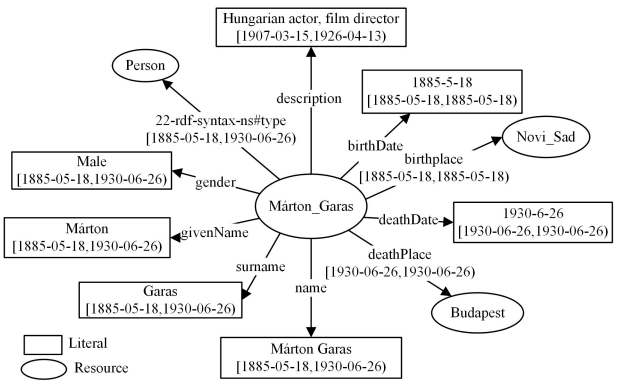

图 2 个人信息 tRDF 图

Fig. 2 tRDF graph of personal information

3.1.1 tRDF 语法

为了保持 RDF 模型的三元组结构及其可扩展性,本文利用时间标签法为传统 RDF 进行时态扩展。为了更精确地表示时态信息,本文根据宾语是资源或者字面量,将时态信息分别附加在三元组(S,P,O)的谓语 P 或者宾语 O 上。为了不失一般性,本文模型采用 SQL:2011 提出的闭-开时间模型,将时间间隔写为$[Ts,Te)$(必须有 $Ts\leqslant Te$)的形式来表示时态信息,其中 Ts 表示开始时间,Te 表示结束时间。这样就得到了对应的 *tRDF* 模型的语法,如定义 1 所示。

定义 1(tRDF 语法) 当宾语是资源时,有 tRDF 三元组为$(S,P[Ts,Te),O)$;当宾语是字面量时,有 *tRDF* 三元组为

$(S,P,O[Ts,Te))$，必须有 $Ts,Te\in T$ 且 $Ts\leqslant Te$。

在定义 1 中，(S,P,O)是传统 RDF 模型的三元组形式。图 2 给出了 Márton_Garas 的个人信息。从图中可以清晰地看到，当宾语是资源时，时间戳就被添加在 RDF 的谓语上，$P[Ts,Te)$就是 tRDF 三元组的谓语部分，表示主语 S 和宾语 O 之间的关系在时间$[Ts,Te)$上有效；当宾语是字面量时，时间戳就被添加在 RDF 的宾语上，$O[Ts,Te)$构成了 tRDF 三元组的宾语部分，表示主语 S 的属性值的有效时间；T 表示一个时间集合，是 Ts 和 Te 可取的时间域；另外，Ts，Te 和 T 的数据格式都为“yyyy-MM-dd”，表示具体的某年某月某日。需要注意的是，为不失一般性，模型使用时间间隔来表示时态信息，但并不代表不能表示时间点，规定当 $Ts=Te$ 时，$[Ts,Te)$就表示一个时间点。如图 2 中所示的出生日期、出生地点、死亡日期和死亡地点等默认是时间点的数据，与规定的一样，通过使 $Ts=Te$ 来令时间间隔表示某一时刻。由此可见，本文提出的 tRDF 模型可以很好地实现对传统 RDF 数据的时态扩展，从而更精准地表示了时态知识图谱。

3.1.2　tRDF 语义

tRDF 的语义主要包含解释、满足和蕴含 3 个方面。其中，时态解释是对传统 RDF 的语义中解释的扩展，下面先给出传统 RDF 语义的解释。

定义 2（RDF 语义解释[36]）　RDF 的解释 I 由以下元素组成：

1）一个非空集合 IR，为 I 的资源集合；

2）一个集合 IP，为 I 的属性集合；

3）一个集合 IL，为 I 的字面量集合；

4）一个映射 $IEXT$，将 IP 映射到 $IR\times IR$，即$\langle x,y\rangle$的集合，其中 x 和 y 在 IR 中；

5）一个映射 IS，将 V 中的 IRI 映射到 $IR\cup IP$；

6）一个映射 ILR，将 V 中的字面量映射到 IR。

（1）时态解释：使用表达式或逻辑关系运算符来解释数据模型的语义。

定义 3（时态解释）　tRDF 模型的解释 TI 通过在定义 2 中添加如下时态元素：

1）一个 IR 的子集 T，为时态信息集合；

2）一个标志 OR，表示宾语是资源；

3）一个 IP 的子集 BP，为宾语是字面量时不携带时态信息的谓语集合；

4）一个 IP 的子集 TP，为宾语是资源时携带时态信息的谓语集合，TP 中添加时态相关内容，如 $startTime$ 和 $endTime$ 属性；

5）一个 IR 的子集 BO，为宾语是资源时不携带时态信息的宾语集合；

6）IL 中添加时态相关内容，如 $startTime$ 和 $endTime$ 属性；

7）一个映射 PT，将 $TP\times(T\cap OR)\times(T\cap OR)$映射到 IP；

8）一个映射 ILR，将 $IL\times T\times T$ 映射到 IR。

（2）时态满足：表示解释 TI 和时态三元组之间的基本语义关系。

定义 4（时态满足）　对于给定的 tRDF 模型 TM 的解释 TI，TI 满足某个 tRDF 三元组 $tm\in TM$，记作 $TI\models tm$，当且仅当：

当宾语是资源时：

1）$\forall Ts,Te\in T,(S,P,O)\in(TI(Ts)\wedge TI(Te))$都有 $TI\models(S,P[Ts,Te),O)$；

2）$\forall Ts,Te\in T,(S,subP,O)\in(TI(Ts)\wedge TI(Te))$都有 $TI\models(S,subP[Ts,Te),O)$。

当宾语是字面量时：

1）$\forall Ts,Te\in T(S,P,O)\in(TI(Ts)\wedge TI(Te))$都有 $TI\models(S,P,O[Ts,Te))$；

2）$\forall Ts,Te\in T(S,P,subO)\in(TI(Ts)\wedge TI(Te))$都有 $TI\models(S,P,subO[Ts,Te))$。

若 $\forall tm\in TM$，都有 $TI\models tm$，则称时态解释 TI 满足 tRDF 模型 TM，记作 $TI\models TM$。

（3）时态蕴含：表达了两个事物之间的逻辑关系。

定义 5（时态蕴含）　当宾语是资源时，对于给定的 $tRDF$ 时态解释 TI，如果有 $TI\models(S,P[Ts,Te),O)$且 P'是 P 的子属性，则有 $TI\models(S,P'[Ts,Te),O)$；当宾语是字面量时，对于给定的 tRDF 时态解释 TI，如果有 $TI\models(S,P,O[Ts,Te))$且 O'是 O 的子宾语，则有 $TI\models(S,P,O'[Ts,Te))$。

3.2　tRDF 数据存储

如前文所述，关系型数据库被广泛用于研究传统 RDF 数据的存储。在此基础上，本文利用关系数据库 PostgreSQL 结合 SQL:2011 提出的时态数据库的概念来设计存储 tRDF 数据的方法。本节首先介绍我们创建的数据库模式，然后给出了 tRDF 数据到 PostgreSQL 的映射规则与算法。

3.2.1　数据库模式的标准定义

本文通过对广泛使用的垂直模式进行范式分解，建立了以下 5 张表：命名空间表、主语表、谓语表、宾语表和三元组声明表。数据库模式的详细定义如下。

定义 6（数据库设计模式）　PostgreSQL 数据库模式是一个六元组 $P=(N,COL,DT,PK,FK,L)$。

1）N 是一个非空有限的名称集合，$N=TN\cup DN$，TN 是实体表名集合，DN 是数据类型名称集合；

2）COL 是一个表列名的非空有限集合，有 $\forall t\in TN$，$\exists COL(t)$；

3）DT 是一个表属性列的数据类型的集合，有 $\forall c\in COL(t)$，$\exists DT(c)\in DN$；

4）PK 是表主键的集合，$\forall t\in TN$，$\exists PK(t)\in COL(t)$；

5）FK 是一个表外键的集合，$\forall t\in TN$，$\exists\ n(n\geqslant 0)$个 $FK(t)\subseteq COL(t)$，外键的取值范围为 $value(FK(t_i))\subseteq value(PK(t_j))\cup\{Null\}$，其中 $value(*)$表示 $*$ 的取值范围；

6）L 是一个表之间关系的集合，$L\subseteq TN\times TN$，表之间的关系由外键 $FK(t_i)$对主键 $PK(t_j)$的引用来表示，对于 $\forall t_i$，$t_j\in TN$，都有表 t_i通过外键 $FK(t_i)$对表 t_j的主键 $PK(t_j)$进行引用，记作 $FK(t_i)\rightarrow PK(t_j)$。

各表的具体结构如表1—表5所列。

表1 命名空间表

Table 1 Namespace

ID(PK)	Prefix
1	http://dbpedia.org/resource
—	—

命名空间表由主键 *ID* 和 *Prefix* 列构成，用于存储三元组的前缀 *IRI*。由于 tRDF 数据中存在大量重复的 *IRI*，将 *IRI* 与主语、谓语和宾语分离有利于节省空间。命名空间表通过主键 *ID* 链接到主语表、谓语表和宾语表，并将 *IRI* 存储到 *Prefix* 列中。

表2 主语表

Table 2 Subjects

ID(PK)	NS_ID(FK)	Resource
1	1	Fiatau_Penitala_Teo
—	—	—

主语表由主键 *ID*、外键 *NS_ID* 和 *Resource* 列组成，用于存储 tRDF 三元组的主语部分，并通过主键 *ID* 与声明表相关联。*NS_ID* 列作为外键实现对主语前缀在命名空间表中的 *ID* 的引用。*Resource* 列存储了不带前缀的主语。

表3 谓语表

Table 3 Predicates

ID(PK)	NS_ID(FK)	Property	PTs	PTe
1	2	22-rdf-syntax-ns#type	1965-11-07	1996-10-21
—	—	—	—	—

谓语表由主键 *ID*、外键 *NS_ID*、*Property*、*PTs* 和 *PTe* 列组成，用于存储 tRDF 三元组的谓语部分，并通过主键 *ID* 与声明表相关联。*NS_ID* 列作为外键实现了对谓语前缀在命名空间表中 *ID* 的引用，*Property* 列存储了不带前缀的谓语。当宾语为资源时，*PTs* 和 *PTe* 为时间间隔的开始时间和结束时间；当宾语是字面量时，*PTs* 和 *PTe* 允许为空。

表4 宾语表

Table 4 Objects

ID(PK)	NS_ID(FK)	Object	OTs	OTs
1	3	Person		
—	—	—	—	—

宾语表由主键 *ID*、外键 *NS_ID*、*Object*、*OTs* 和 *OTe* 列组成，用于存储 tRDF 三元组的宾语部分，并通过主键 *ID* 与声明表相关联。*NS_ID* 列作为外键实现了对宾语前缀在命名空间表中 *ID* 的引用。当宾语是字面量时，记录的 *NS_ID* 对应于命名空间表中表示空前缀的 *ID*。*Object* 列存储不带前缀的宾语。当宾语为资源时，*OTs* 和 *OTe* 允许为空；当宾语为字面量时，*OTs* 和 *OTe* 是时间间隔的开始时间和结束时间。

表5 声明表

Table 5 Statements

ID(PK)	Sid(FK)	Pid(FK)	Oid(FK)
1	1	1	1
—	—	—	—

声明表由主键 *ID*、外键 *Sid*、外键 *Pid* 和外键 *Oid* 组成，使用整数来存储 tRDF 三元组的声明。声明表使用外键 *Sid*，*Pid* 和 *Oid* 来引用主语、谓语和宾语，极大地节省了空间。

这些表通过主键和外键相互连接，表之间的具体关系如图3所示。

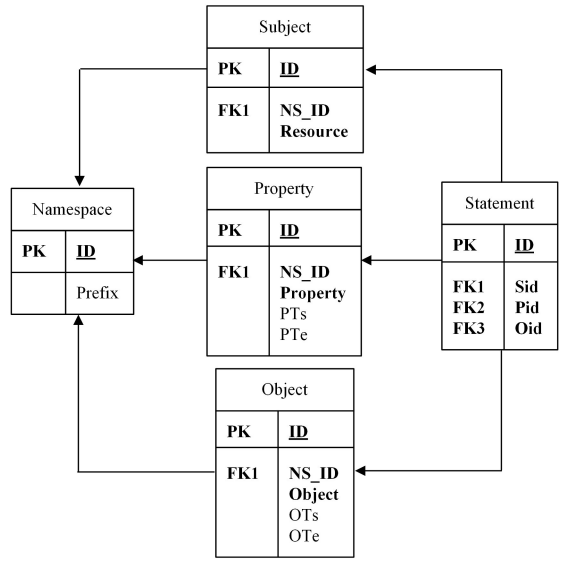

图3 数据库模式图

Fig. 3 Database schema graph

3.2.2 映射规则与算法

基于3.2.1节设计的数据库模式，本节给出了 tRDF 数据到 PostgreSQL 的映射规则和实现算法。首先将 tRDF 三元组的主语、谓语和宾语分割为前缀 *N*、不带前缀的主语 *ES*、不带前缀的谓语 *EP*、不带前缀的宾语 *EO*、时间信息 *Ts* 和 *Te*，然后将得到的数据存入对应的表，数据存储规则如下。

规则1 将前缀 *N* 插入命名空间表的 *Prefix* 列。注意，当宾语是字面量时，根据命名空间表的结构设置，*Prefix* 允许为空。

规则2 将规则1得到的命名空间表中的前缀 *ID* 和对应的不带前缀的主语 *ES* 分别插入主语表的 *NS_ID* 和 *Resource* 列中。每个记录对应的 *ID* 作为主键被声明表引用。

规则3 将规则1得到的命名空间表中的前缀 *ID* 和对应的不带前缀的谓语 *EP* 分别插入到谓语表的 *NS_ID* 和 *Property* 列中。当宾语为资源时，将时间信息 *Ts* 和 *Te* 插入到谓语表的 *PTs* 和 *PTe* 列中。将每个记录对应的 *ID* 作为主键被声明表引用。

规则4 将规则1得到的命名空间表中的前缀 *ID* 和对应的不带前缀的谓语 *EO* 分别插入到宾语表的 *NS_ID* 和 *Object* 列中。当宾语为字面量时，将时间信息 *Ts* 和 *Te* 插入宾语表的 *OTs* 和 *OTe* 列中。每个记录对应的 *ID* 作为主键被声明表引用。

规则5 当规则3和规则4涉及时间信息操作时，*SQL*：2011会自动根据时间间隔对数据进行丢弃、合并、覆盖、插入操作。

规则6 将规则2、规则3和规则4返回的每个 tRDF 三元组的主语、谓语和宾语的 *ID* 插入到声明表的 *Sid*，*Pid* 和 *Oid* 列中。

具体实现算法如算法 1 所示。

算法 1　tRDFToPostgreSQL 存储算法

```
输入：tRDF 三元组 nt
输出：PostgreSQL 中的数据
/* 首先分析三元组 nt，获取主语、谓语和宾语 */
1.  subject←nt.getSubject().toString()
2.  predicate←nt.getPredicate().toString()
3.  objectN←nt.getObject()
4.  object←objectN toString()
    /* 以宾语为资源为例，从谓语上获取时态信息 */
5.  predicate1←predicate.substring(0,predicate.indexOf("["));
6.  T←predicate.substring(predicate.indexOf("[")+1,predicate.
    length()-1);
7.  ts←T.substring(0,T.indexOf(","));
8.  te←T.substring(T.indexOf(",")+1,T.length());
    /* 以主语为例，分割前缀，谓语和宾语的操作与之类似，不再赘
    述 */
9.    S←subject.split("/");
10.  easyS←S[S.length-1];
11.  regexS←"/"+easyS;
12.  namespaceS←subject.substring(0,subject.indexOf(regexS));
    /* 接着，根据映射规则存入数据 */
    /* 若命名空间表中不存在前缀记录，则将其插入，并返回对应
    ID */
13. if(isNotExistN(namespaceS)) then
14.  String sqlInsertNamespace←"INSERT INTO namespace(pre-
    fix) VALUES(?);";
    /* 当主语表中无该记录时，插入前缀 ID 以及无前缀主语 */
15. if(isNotExistS(S_ID,easyS)) then
16.  String sqlInsertsubject←"INSERT INTO subject(ns_id,re-
    source)VALUES(?,?);";
    /* 向谓语表中插入数据时还需考虑时态信息 */
17. if(isNotExistP(P_ID,easyP,ts,te)) then
18.  String sqlInsertproperty←""NSERT INTO property(ns_id,
    property,pts,pts) VALUES(?,?,?,?);";
19. if(isNotExistO(O_ID,easyO)) then
20.  String sqlInsertobject←"INSERT INTO object(ns_id,object)
    VALUES(?,?);";
    /* 当声明表中无该三元组记录时，插入 tRDF 三元组的主语、
    谓语和宾语在主语表、宾语表和谓语表中的对应 ID */
21. if(isNotExistStm(Sid,Pid,Oid)) then
22.  String sqlInsertstatement←"INSERT INTO statement(sid,pid,
    oid)VALUES(?,?,?);";
    /* 宾语是字面量时操作与上述类似，不再赘述 */
```

4　实验验证

为了验证所提方法的可行性，本实验基于 JDK13 版本的 Eclipse 平台和 12.3 版本的 PostgreSQL 开发，并在具有 Intel(R) Core(TM) i5-4210H 2.9 GHz 处理器、8.00 GB RAM 和 Windows 10 操作系统的系统上完成。

4.1　数据集

本实验使用的数据集是从 DBpedia 中获取的个人信息数据集，其中包含 10310048 个 RDF 三元组。由于时态 RDF 的标准并未统一，因此从 DBpedia 获得的数据集并不是时态 RDF 数据集，需要根据第 3 节给出的模型对其进行扩展，形成相应的 tRDF 数据集。然后，将扩展后的 tRDF 数据集划分为不同规模，以达到在不同大小的 tRDF 数据集上测试方法性能的目的。表 6 列出了每个数据集的基本内容。

表 6　实验数据集

Table 6　Experimental datasets

Dataset	Number of tRDF triples	Sizeof dataset /MB
Dataset1	102606	13.7
Dataset2	1145487	153
Dataset3	5332453	712
Dataset4	10310048	1372

4.2　实验结果分析

按照第 3 节提出的存储方法，实验将上述 4 个不同大小的数据集存储到 PostgreSQL 数据库中，并从存储时间和存储空间两个方面比较了验证方法的可行性。为了保证实验的准确性，本文实验数据均取 5 次实验结果的平均值。

4.2.1　存储时间

将 4 个不同大小的数据集按如下两种方案存储：垂直存储方案和本文设计的存储方案。所花的时间如图 4 所示。

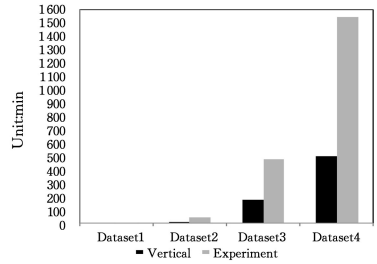

图 4　存储时间

Fig. 4　Time of storage

两种方案存储 Dataset1 花费的时间分别为 0.69 min 和 3.43 min，由于时间过短，在图 4 中表现不明显。从图 4 可以看出，随着 tRDF 数据集规模的增大，数据存储的时间也随之增加。一方面，当数据集中三元组的个数小于 100 万时，无论是垂直模式还是本文提出的存储模式，存储时间都随着三元组的个数呈线性增长；由于关系型数据库存在插入百万数据时效率明显降低的缺陷，因此当三元组的数量超过百万时，两种方法的存储时间都剧烈增大，存储效率明显降低。另一方面，由于垂直模式直接将三元组存储在一个表中，而本文所采用的存储方法需要先分析数据，再将信息存到不同的表中，因此本实验方法的存储时间长于垂直模式，但两者所用时间的比值并不受数据集规模的影响。

4.2.2　存储空间

针对不同规模的 4 个数据集，分别以文件形式、数据库垂直模式和本文所给出的存储方法 3 种不同方式进行存储，所占空间大小如图 5 所示。从图 5 可以看出，无论哪种存储方法，占用的内存大小都随着数据集规模呈线性增长。由于垂直模式中所有内容都存储在一张表中，因此这种存储方法所占空间在任何大小的数据集上都是最大的；当数据集中的三元组个数小于十万条时，3 种存储方法所占内存相近，但当数据规模达到数十万甚至数百万时，本文方法在三者中使用的

空间最少，存储性能也最高。一方面，这是因为 tRDF 三元组有很多重复的前缀，在本文方法中，命名空间表用于存储唯一的前缀，而其他表中的前缀只需要存储命名空间表中对应的 ID。另一方面，在声明表中利用主语表、谓语表和宾语表中对应的 ID 来表示三元组，整数的使用也极大地减少了内存消耗。因此，本文方法凭借其合理的设计可以更有效地利用空间。

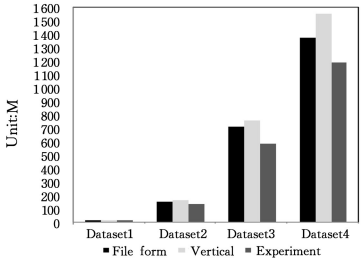

图 5 存储空间

Fig. 5 Space of storage

以上实验结果证实了本文提出的 tRDF 模型能够有效地表示时态知识图谱，针对所提模型而给出的存储方法也能高效地存储 tRDF 数据。

结束语 随着网络上时态数据的增多，研究时态知识图谱新的表示形式是十分必要的。鉴于时态 RDF 模型还未有统一的标准以及现有的模型缺乏对时态信息所属对象的准确定位的现状，本文首先提出了一种新的名为 tRDF 的时态 RDF 模型，并对其语法和语义进行了详细的定义；其次结合时态数据库的概念，提出了一种利用关系型数据库 PostgreSQL 高效存储 tRDF 数据的方法，并给出了相应的映射规则与算法。最后通过大量的实验，从存储的时间和空间等多个角度验证了该方法的可行性。由于本文注重的是时态 RDF 的建模和存储研究，针对 tRDF 数据查询的研究还不够深入。在未来的工作中，我们将针对完善 tRDF 数据模型查询语言的研究，来提高 tRDF 数据的管理效率。此外，我们还将探索非关系型数据库的使用可能，继续寻求更高效的时态知识图谱管理方法。

参 考 文 献

[1] JUPP S, MALONE J, BOLLEMAN J, et al. The EBI RDF platform: linked open data for the life sciences[J]. Bioinformatics, 2014, 30(9): 1338-1339.

[2] RANZINGER R, AOKI-KINOSHITA K F, CAMPBELL M P, et al. GlycoRDF: an ontology to standardize glycomics data in RDF[J]. Bioinformatics, 2015, 31(6): 919-925.

[3] REESE J T, UNNI D R, CALLAHAN T J, et al. KG-COVID-19: a framework to produce customized knowledge graphs for COVID-19 response[J]. ScienceDirect, 2020, 2(1): 100155.

[4] STADLER C, LEHMANN J, HöFFNER K, et al. Linkedgeodata: A core for a web of spatial open data[J]. Semantic Web, 2012, 3(4): 333-354.

[5] NEUMANN T, WEIKUM G. RDF-3X: a RISC-style engine for RDF[J]. Proceedings of the VLDB Endowment, 2008, 1(1): 647-659.

[6] HARRIS S, LAMB N, SHADBOLT N. 4store: The design and implementation of a clustered RDF store[C]// Proceedings of the 5th International Workshop on Scalable Semantic Web Knowledge Base Systems. Washington DC: CEUR, 2009: 94-109.

[7] SALAS P E, MARX E, MERA A, et al. RDB2RDF plugin: relational databases to RDF plugin for eclipse[C]// Proceedings of the 1st Workshop on Developing Tools as Plug-ins. South Pacific: ACM, 2011: 28-31.

[8] BORNEA M A, DOLBY J, KEMENTSIETSI-DIS A, et al. Building an efficient RDF store over a relational database[C]// Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2013: 121-132.

[9] OEZSU M T. A survey of RDF data management systems[J]. Frontiers of Computer Science, 2016, 10(3): 418-432.

[10] LU J W, YAN L. Mapping Method from Object-relational Database to RDF(S)[J]. Chinese Computer Science, 2021, 48(10): 145-151.

[11] MA Z M, CAPRETZ M, YAN L. Storing massive Resource Description Framework(RDF) data: A survey[J]. The Knowledge Engineering Review, 2016, 31(4): 391-413.

[12] SUN J L, JIN Q. Scalable RDF store based on HBase and MapReduce[C]// Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering. Chengdu: IEEE, 2010: 633-636.

[13] SHAO B, WANG H X, LI Y T. Trinity: A Distributed Graph Engine on a Memory Cloud[C]// Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2013: 505-516.

[14] HOFFART J, SUCHANEK F M, BERBERIC K, et al. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia[J]. Artificial Intelligence, 2013, 194(JAN.): 28-61.

[15] GOAL R, KAZEMI S, BRUBAKER M, et al. Diachronic Embedding for Temporal Knowledge Graph Completion[C]// Proceedings of the AAAI Conference on Artificial Intelligence. New York: AAAI, 2020: 3988-3995.

[16] WANG J Y, DI X F, LIU J M, et al. A Constraint Framework for Uncertain Spatiotemporal Data in RDF Graphs[C]// Proceedings of the 15th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. Kunming: Springer, 2019: 727-375.

[17] BAI L Y, WANG J Y, DI X F, et al. Fixing the inconsistencies in fuzzy spatiotemporal RDF graph[J]. Information Sciences, 2021, 578(2021): 166-180.

[18] CLAUDIO G, HURTADO C A, VAISMAN A A. Temporal RDF[C]// Proceedings of the Second European Conference on The Semantic Web: Research and Applications. Berlin: Springer, 2005: 93-107.

[19] CLAUDIO G, HURTADO C A, VAISMAN A A, et al. Introducing time into RDF[J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(2): 207-218.

[20] PUGLIESE A, UDREA O, SUBREHMAN-IAN V S. Scaling RDF with time[C]// Proceedings of the 17th International Conference on World Wide Web. New York: ACM, 2008: 605-614.

[21] KOUBARAKIS M, KYZIRAKOS K. Modeling and Querying

Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL[C]// Proceedings of the Semantic Web: Research and Applications, 7th Extended Semantic Web Conference. Berlin: Springer, 2010: 425-439.

[22] ZHANG F, WANG K, LI Z, et al. Temporal Data Representation and Querying Based on RDF[J]. IEEE Access, 2019, 7: 85000-85023.

[23] CHEN Y Y, YAN L, ZHANG Z Q, et al. Temporal RDF Model and Index Method Based on Neighborhood Structure[J]. Chinese Computer Science, 2021, 48(10): 167-176.

[24] BRANDT S, ELEM G K, RYZHIKOV V, et al. A Framework for Temporal Ontology-Based Data Access: A Proposal[C]// Proceedings of the European Conference on Advances in Databases and Information Systems. Nicosia: Springer, 2017: 161-173.

[25] ELEM G K, XIAO G, RYZHIKOV V, et al. Ontop-temporal: A Tool for Ontology-based Query Answering over Temporal Data [C]// Proceedings of the 27th ACM International Conference. Indiana: ACM, 2018: 1927-1930.

[26] YAN L, ZHAO P, MA Z M. Indexing temporal RDF graph[J]. Computing, 2019, 101(10): 1457-1488.

[27] ZHAO P, YAN L. A methodology for indexing temporal RDF data[J]. Journal of Information Science and Engineering, 2019, 35(4): 923-934.

[28] FAN T Y, YAN L, MA Z M. Mapping fuzzy RDF(S) into fuzzy object-oriented databases[J]. International Journal of Intelligent Systems, 2019, 34(10): 2607-2632.

[29] FAN T Y, YAN L, MA Z M. Storing and querying fuzzy RDF (S) in HBase databases[J]. International Journal of Intelligent Systems, 2020, 35(4): 751-780.

[30] O'CONNOR M J, DAS A. A Lightweight Model for Representing and Reasoning with Temporal Information in Biomedical Ontologies [C]// Proceedings of the 3rd International Conference on Health Informatics. Barcelona: DBLP, 2010: 90-97.

[31] KULKARNI K, MICHELS J. Temporal features in SQL: 2011 [J]. ACM SIGMOD Record, 2012, 41(3): 34-43.

[32] GAO Q, LEE M L, DOBBIE G, et al. A Semantic Framework for Designing Temporal SQL Databases[C]// Proceedings of the 37th International Conference on Conceptual Modeling. Xi'an: Springer, 2018: 382-396.

[33] LU W, ZHAO Z H, WANG X Y. A lightweight and efficient temporal database management system in TDSQL[C]// Proceedings of the 45th International Conference on Very Large Data Bases. Los Angeles: VLDB, 2019: 2035-2046.

[34] ANSELMA L, PIOVESAN L, TERENZIANI P. Dealing with temporal indeterminacy in relational databases: An AI methodology[J]. AI Communications, 2019, 32(3): 1-15.

[35] Al-FEDAGHI S. Conceptual Temporal Modeling Applied to Databases[J]. International Journal of Advanced Computer Science and Applications, 2021, 12(1): 524-534.

[36] RDF 1.1 Primer[EB/OL]. (2014-02-25) [2021-11-01]. http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/.

HAN Xiao, born in 1996, postgraduate. His main research interests include semantic web and temporal RDF data.

YAN Li, born in 1964, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. Her main research interests include big data, knowledge graph, spatiotemporal information processing and NoSQL database.

（责任编辑：喻藜）