Hans Journal of Data Mining 数据挖掘, 2023, 13(4), 335-351
Published Online October 2023 in Hans. https://www.hanspub.org/journal/hjdm
https://doi.org/10.12677/hjdm.2023.134033

Hans汉斯

基于RDB-RDF模式映射的数据转换方法研究

高　金，李征宇，孙　平

沈阳建筑大学计算机科学与工程学院，辽宁 沈阳

收稿日期：2023年9月3日；录用日期：2023年10月3日；发布日期：2023年10月11日

摘　要

随着语义web的发展，现代web希望数据能够采用资源描述框架(RDF)的格式，这是一种机器可读的形式，能够在无需人工干预的情况下共享和重用数据。但是目前大多数数据仍然存储在关系数据库中，现有的将关系数据转换为资源描述框架的方法由于映射不佳，未能产生预期的结果，因此，本文提出了一种基于RDB-RDF模式映射的数据转换方法，从形式化定义出发，使用模式映射，借助于映射描述，结合数据物化和按需映射，避免数据全部转储的方法，使SPARQL查询转换为SQL查询时简单便捷，提高转换效率和数据检索时间。此外本方法还对关系数据库进行了扩充，能够实现对象关系数据库转换为资源描述框架。最后给出方法的整体思路，各项结果表明，新的方法既能够保持语义，又能够提高速度，实现了比传统方法更加易于理解的映射方法。

关键词

按需映射，映射描述，RDB-RDF，SPARQL-SQL

Research on Data Transformation Method Based on RDB-RDF Schema Mapping

Jin Gao, Zhengyu Li, Ping Sun

School of Computer Science and Engineering, Shenyang Jianzhu University, Shenyang Liaoning

Received: Sep. 3rd, 2023; accepted: Oct. 3rd, 2023; published: Oct. 11th, 2023

Abstract

With the development of semantic web, modern web expects data to be in Resource Description Framework (RDF) format, which is a machine-readable form that enables sharing and reusing data without human intervention. However, most of the data are still stored in relational databases, and existing methods for converting relational data to Resource Description Framework fail to produce the desired results due to poor mapping, therefore, in this paper, we propose a

文章引用: 高金, 李征宇, 孙平. 基于 RDB-RDF 模式映射的数据转换方法研究[J]. 数据挖掘, 2023, 13(4): 335-351.
DOI: 10.12677/hjdm.2023.134033

data conversion method based on RDB-RDF schema mapping, using from formal definitions, schema mapping with the help of mapping descriptions, and combining data materialization and on-demand mapping, to avoid all data dumping, the method makes the conversion of SPARQL query to SQL query simple and convenient and improves the conversion efficiency and data retrieval time. In addition, this method also extends the relational database, which can realize the conversion of object-relational database to resource description framework. Finally, the overall idea of the method is given, and the results show that the new method is able to maintain the semantics and improve the speed, and realize the mapping method which is easier to understand than the traditional method.

Keywords

On-Demand Mapping, Mapping Description, RDB-RDF, SPARQL-SQL

1. 引言

近年来，语义 web [1]成为了目前主要的学术研究之一，在语义 web 的基础之上许多关键技术都能更进一步发展，语义 web 的目的不仅仅是简单的提取原始数据，而是将数据连接在一起，利用添加元数据的方法来使机器理解数据概念和不同数据间的逻辑关系[2] [3]。同时将现实世界中的各种概念和实体以结构化的形式进行表达，让机器也明白数据的语义以及数据之间的关系。

关系数据库是当前使用的最为广泛的一种数据库，各方面都具有很好的优势，但因为数据欠缺一定的语义能力，所以要把关系数据库转换为具有语义的资源描述框架，因此研究将关系数据库转换为资源描述框架有很好的研究价值。

目前对于将关系数据库转换为资源描述框架的研究有很多，比较多采用的方法是 W3C 提供的 DM [4] (直接映射)和 R2RML 映射语言，其中 R2RML 是推荐标准，但这个方法需要手工编辑映射过程[5]，并掌握和使用各种映射工具和映射语言，还需对本体模型和数据库结构等特别熟悉[6] [7] [8] [9]。目前的传统方案所使用的映射方式实现的效率并不佳，甚至未能达到预期效果[10] [11] [12] [13]，此外很多的转换方法是将整个关系数据库直接转换为 RDF 格式[14] [15] [16]，这样会造成占据存储空间大，转换效率低的问题，还需要对关系数据库间的模式信息进行进一步的分析并提取，来保证转换的数据的语义保持能力[17]，所以本文提出了新的方法——基于 RDB-RDF 模式映射的数据转换方法从而实现关系数据库到资源描述框架的转换。

本文避免了数据的全部转储，采用对用户的每个 SPARQL 查询转换为 SQL 查询的方式查询关系数据库并提取 RDB 数据的方法。此外，在转换时将实例和模式的关注点分离开来，先利用二者形式化定义来完成基础的模式映射，并在映射过程中引入映射描述，从而简化转换过程。数据物化和按需映射相结合，逐步完成数据物化。

2. 相关概念

关系数据库(Relational database)：实体和实体之间的联系的集合能够构成一个关系数据库，用行和列组成的二维表去管理数据，执行具体操作时使用 SQL 来实现。

RDF(S)：即元数据模型 RDF (Resource Description Framework)和 RDF 模式(RDF Schema，简称 RDFS)，能够在 Web 中描述任何有用信息，并且能够为这些信息赋予确定的语义。

数据物化：数据物化是将静态源数据库转换为 RDF 表示的过程。描述了如何通过属于映射和三重映射将关系数据库转换为 RDF。生成的 RDF 知识库可以在三重存储中物化，然后用 SPARQL 进行查询[18]。

按需映射：按需是动态的，只需要考虑当前的 SPARQL [19] [20]查询，只涉及 SPARQL 查询中指定的三元组的数据[21]。

3. 基于 RDB-RDF 模式映射的数据转换方法

为了更好的保留数据的完整语义信息，更好的完成整个映射过程，本节通过关系数据库以及 RDF(S)的显著特点，提出二者对应的形式化定义。

3.1. 形式化定义

3.1.1. 关系数据库的形式化定义

关系数据库模式由关系模式(表的结构)和完整性约束两部分组成。在关系数据库模式这个内容中，实体和实体间的联系都是用关系来表示，关系模式定义了关系(表)的结构、属性(字段)及其数据类型等完整性约束的定义。

定义 1：关系数据库模型可以用一个四元组来表示 $RDBM=(B,C,Inh,Ins)$。

B 表示关系数据库中的基本概念的有限集。$B=Tab\cup Col\cup D$，Tab 表示的是关系数据库中的所有数据表有限集。Col 代表数据表中字段的有限集。D 表示数据类型的有限集。

C 表示关系数据库中约束关系的集合。$C=Pcons\cup Fcons\cup Ucons\cup Ncons$, $Pcons$ 表示所有主键约束的集合，$Fcons$ 表示外键约束的有限集，$Ucons$ 表示所有唯一约束的有限集，$Ncons$ 表示所有非空约束的有限集。

Inh 表示关系数据库中继承关系的有限集，$Inh=SingleI\cup MultiI$，其中 $SingleI$ 表示所有单继承关系的集合，$MultiI$ 表示所有多继承关系的集合。

Ins 表示数据库中所有数据表存储实例数据的集合。

3.1.2. RDF(S)模型的形式化定义

RDF 语句由 RDF 声明构成，每个声明由主语、谓语和宾语这样的三元组的形式表示。RDF Schema 是一种轻量级的本体语言，能够提供具有固定含义的建模语言，包括类和子类、属性和子属性、定义域和值域等约束情况。RDF(S)的形式化定义如下。

定义 2：资源描述框架模型可以用一个三元组 $RM=(RB,RA,RI)$ 来表示。

RB 表示的是 RDF(S)所有基本属性的有限集，$RB=RC\cup RD\cup RP$。RC 表示了 RDF(S)中的所有类，RD 表示所有基本数据类型，RP 表示所有属性的集合。

RA 表示 RDF(S)中的公理集合，是由两两不相交的集合来组成的。$RA=CAxiom\cup PAxiom$，类公理 $CAxiom$ 集中包含了本体中定义的所有类公理；属性公理 $PAxiom$ 集中包含本体中定义的所有属性公理。

RI 表示所有实例的有限集。

全篇的关系数据库以房地产对象关系数据库为依据，关系数据库中存储的数据表结构如图 1 所示。builds 表为建筑物盘，包含建筑物盘号 Id、开发商 builderCom 和项目名称 projectName；build 表为楼栋，包含楼栋号 Id、盘号 buildsId；project 表为项目表，包含项目号 Id、开发商 builderCom、项目名称 name 和项目日期 date；builder 表为开发商表，包含开发商 Company 和开发商所在地址 town；state 表为房屋状态表，包含房屋号 Id 和房屋状态 roomstate；takebuilds 表为关联关系表，buildsId 和 builderCom 是该表的主

键也是外键，作为外键分别指向表 builds 的主键和表 builder 的主键；house 表为房屋表，包含房屋号 Id，房屋地址 addr 和房屋楼层 floor；图中虚线框中的内容为 house 表的继承关系，分别为 commercial_house、residential_house 和 public_rental_house。Commercial_house 表代表商业房，包含房屋号 Id，房屋收益 earnings 和房屋用途 use；residential_house 代表住宅房，包含房屋号 Id 和房屋产权 property_right；public_rental_house 为公租房，包含房屋号 Id，房屋居住权 residence 和房屋补贴 subsidy。

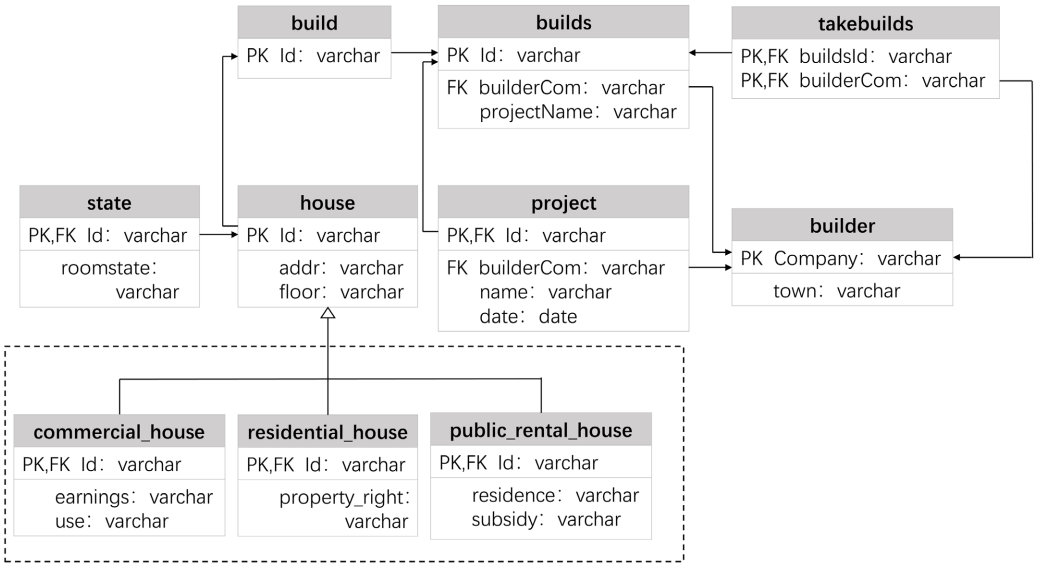

Figure 1. Structure of relational database
图 1. 关系数据库的结构图

3.2. 模式映射

整个系统的映射过程分为模式层面和实例层面进行转换，首先介绍模式层面的映射情况。

3.2.1. RDF Schema 的构建规则

根据关系数据库和 RDF(S)的形式化定义以及提取到的语义信息，提出构建 RDF Schema 的详细规则。在本节中 RDF(S)模型利用提出的三元组 $RM=(RB,RA,RI)$ 表示，关系数据库模型用 $RDBM=(B,C,Inh,Ins)$ 表示，φ 表示映射过程。

规则 1 (基本实体表映射)：

$$\forall t \in baseTable \rightarrow \varphi(t) \in RC$$

对于数据库中任意一个基本实体表可以直接映射成为 RDF Schema 中类。如图 1 中的 builder 表可以直接映射成为 RDF Schema 中的类，映射结果为：

<rdfs:Class rdf:ID="builder"/>

规则 2 (继承关系映射)：

$\forall t1 \in SingleITable \cup MultiITable$ AND $t2 \in parent(t1) \rightarrow \varphi(t1) \in RC$ AND $\varphi(t2) \in RC$ AND $\varphi(i) \in CAxiom \varphi(t1) = \{\varphi(t2)\}$

对于任一包含单继承或者多继承关系的实体表，映射时需要保留数据表包含的继承语义。以图 1 中 commercial_house 表为例，该表是 house 表的子表。则保留继承语义时先按照规则 1 创建父表，再创建子表，并将子表与父表按照继承关系进行连接，映射结果为：

<rdfs:Class rdf:ID="house"/>

<rdfs:Class rdf:ID="commercial_house"/>

<rdfs:Class rdf:about="#house">

<rdfs:subClassOf rdf:resource="#commercial_house"/>

</rdfs:Class>

规则 3 (基础数据类型映射)：

$$\forall c \in Col \quad \text{AND} \quad type(c) \in D \rightarrow \varphi(c) \in RP$$

如果数据表中的字段类型是基础的数据类型，那么这个字段就映射成为 RDF(S)中的数值属性，这个字段映射后的定义域为字段所在的数据表映射成为的类，值域为字段的数据类型。以 builder 表中包含的 Company 属性为例，该属性的数据类型是基础数据类型。映射结果可以表示为：

<rdfs:Class rdf:ID="builder"/>

<rdf:Property rdf:ID="Company">

<rdfs:domain rdf:resource="#builder"/>

<rdfs:range rdf:resource="&xsd;string"/>

</rdf:Property>

规则 4 (引用关系表映射)：

$$\forall t \in ReferenceTable \quad \text{AND} \quad FK(t,c) \rightarrow \varphi(t) \in RC \quad \text{AND} \quad \varphi(c) \in RP$$

引用表中包含外键约束的字段，可以看作是两个实体间的聚合关系。映射时可以将外键约束所在字段映射成为对象属性。例如 builds 表中的 builderCom 字段是该表的外键，指向表 builder 的主键，这时就将 builder 作为 builderCom 属性的属性类型，映射结果如下：

<rdfs: Class rdf: ID="builds"/>

<rdfs: Class rdf: ID="builder"/>

<rdf: Property rdf: ID="builderCom">

<rdfs: domain rdf: resource=" #builds"/>

<rdfs: range rdf: resource="#builder"/>

</rdf:Property>

规则 5 (关联关系表映射)：

$$\forall t \in RelationTable \rightarrow \varphi(Col(t)) \in RP$$

关联关系表达两个实体之间的关系，没有单独拥有的字段，所以直接映射成为对象属性。如 takebuilds 表，该表将不被映射成为 RDF Schema 类，而是将其映射成为 RDF Schema 中的对象属性，表达两个实体间的联系。外键引用的表即为该属性的定义域和值域，映射结果如下：

<rdf: Property rdf: ID="takebuilds">

<rdfs: domain rdf: resource="#builds"/>

<rdfs: range rdf: resource="#builder"/>

</rdf: Property>

3.2.2. RDF Schema 图的构建

接下来提出将关系数据库模式转换为 RDF 图的形式。RDF 图是以三元组的形式显现出来的，所以在

关系数据库映射为 RDF 图的过程中，将满足以下映射规则(图 2 为映射规则的图式情况)：

1) 数据库名称映射到命名空间。数据库的名称映射到 RDF 名称空间；
2) 表名映射为主体。RDB 表的名称映射到 RDF 主体；
3) 列映射为谓词。RDB 表中的列映射到 RDF 图中的谓词；
4) 列值作为对象。Table.column 的单元格映射到 RDF 图的对象；
5) 行作为实例。RDB 表中的每一行都映射到其相应的 RDF 三元组。

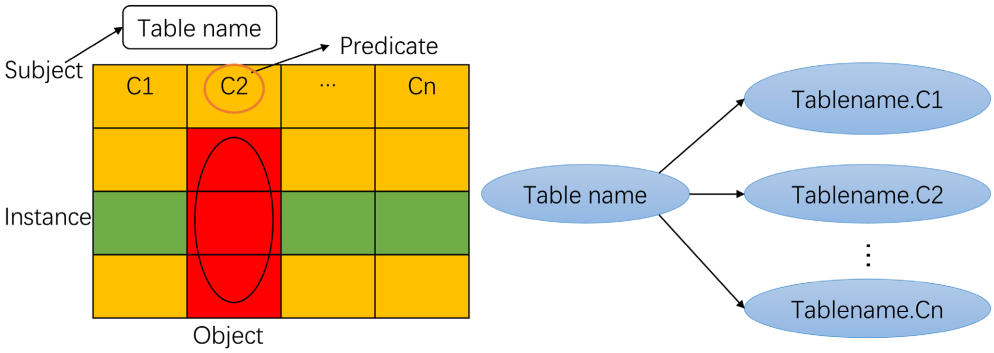

Figure 2. RDB-RDF mapping rules
图 2. RDB-RDF 映射规则

谓词类型：要建立表之间的关系，需要同时使用主键和外键，有些情况下，表之间的关系没有明确指定，而是隐含为概念模式。根据谓词的功能，他们被分为属性谓词和链接谓词两种。整个关系数据库中的部分数据信息如表 1(a)~(i)所示。

1) 属性谓词：表示基本概念，并包含以值项表示为 TableName.Ci 的文字值。例如，builds.Id 表示属性谓词，其中 builds 是基元表，Id 是表 1(a)中表示列的标题。

2) 链接谓词：链接谓词派生自外键和主键关系。根据目标表的不同，链接谓词分为内部链接和外部链接。内部链接谓词用于递归关系，而外部链接谓词用于不同的表。链接谓词的对象值将连接表达式表示为 TableName.Ci = TableName.Cj。

Table 1. RDB relation table
表 1. RDB 关系表

(a) Builds

Id	builderCom	ProjectName
56,846	沈阳市万科西盛置业有限公司	万科西盛花园项目(二期)
5329	沈阳红宇房地产开发有限公司	秀水花城(二期)

(b) Build

Id	buildsId	builderCom
451,965	56,846	沈阳市万科西盛置业有限公司
451,983	56,846	沈阳市万科西盛置业有限公司
184,277	5329	沈阳红宇房地产开发有限公司
184,278	5329	沈阳红宇房地产开发有限公司

(c) House

Id	buildId	buildsId	addr	floor
9,325,570	451,965	56,846	于洪区西湖街 11-2 号	第 6 层
8,959,968	451,983	56,846	于洪区西湖街 13-3 号	第 20 层
2,859,973	184,277	5329	苏家屯区乔松路 36-1 号	第 2 层
2,859,907	184,278	5329	苏家屯区乔松路 36 号	第 1 层

(d) Project

Id	builderCom	name	date
56,846	沈阳市万科西盛置业有限公司	万科西盛花园(二期)	2018-10-23
5329	沈阳红宇房地产开发有限公司	秀水花城(二期)	2009-10-27

(e) Builder

Company	town
沈阳市万科西盛置业有限公司	于洪区
沈阳红宇房地产开发有限公司	苏家屯区

(f) State

Id	roomstate
9,325,570	已发证
8,959,968	已发证
2,859,973	已售
2,859,907	查封

(g) Commercial_house

Id	earnings	use
5,119,617	28.8 万元	厂区租用
53203	17.8 万元	写字楼

(h) Residential_house

Id	property_right
9325570	70 年
2859973	70 年

(i) Public_rental_house

Id	residence	subsidy
4,218,956	3 年	1.44 万元
{15CDCDA6-1314-4B6C-9AB7-8AC12242F8C8}	3 年	0.72 万元

根据上述规则以及关系数据库结构，可以将 RDB 的结构映射成为 RDF 图的关系图，整体的关系图如图 3 所示。在模式层次上，不引入复杂的结构，生成这样的概念模式图。链接谓词借助于属性谓词中虚线包含的关系表达式(后称为连接表达式)，有效地解决了许多复杂的映射问题，例如外键关系。根据链接谓词而引入的连接表达式使得原有的隐式实现的外键关系显示的表示出来，有助于解决 RDB 到 RDF 映射期间产生的各种复杂的问题。

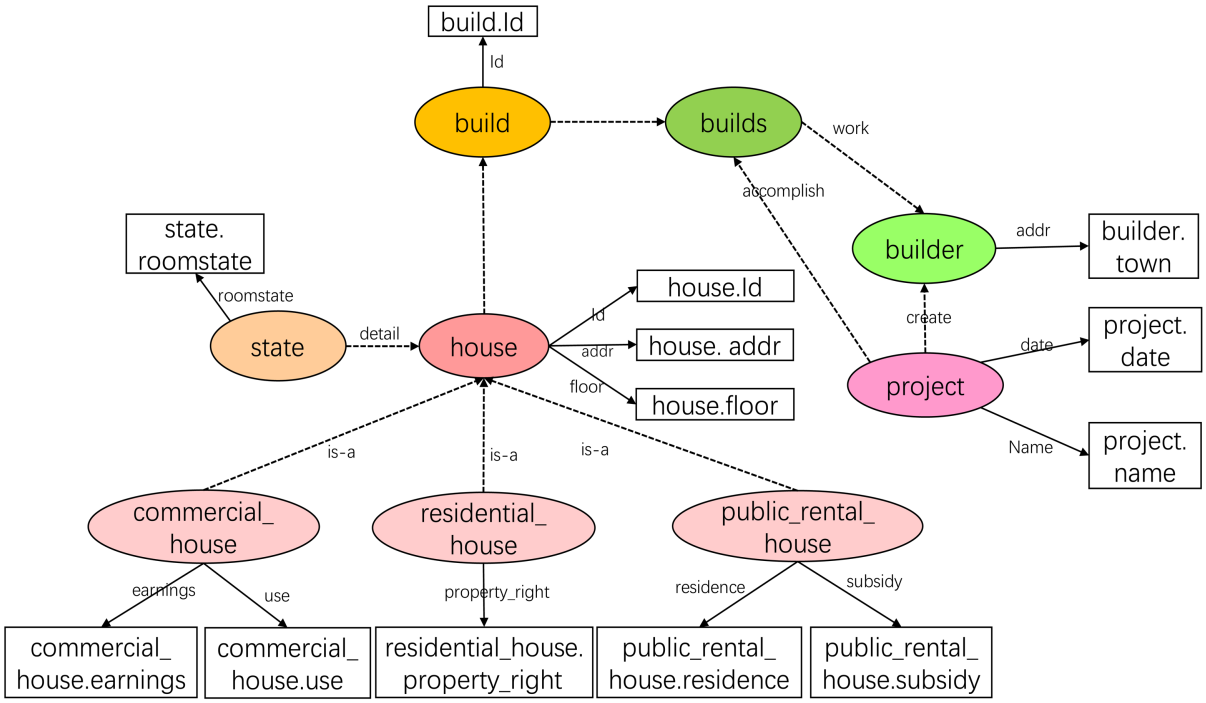

Figure 3. Overall system diagram
图 3. 系统整体关系图

映射描述：关系图 3 显示了 RDF 关系图的所有的映射信息，包含各数据表之间的联系，根据外键的联系产生的数据表之间的关系。映射描述其实就是对数据库映射情况的具体解释，将数据表之间的联系用表达式的形式展示出来，便于后续查询转换的理解使用。具体的表之间的映射描述如表 2(a)，表 2(b)所示。

Table 2. Mapping description table
表 2. 映射描述表

(a) Column mapping

RDF.property	RDB.column
Id	build.Id
Id	house.Id
addr	house.addr
floor	house.floor
roomstate	state.roomstate

Continued

name	project.name
date	project.date
town	builder.town
earnings	commercial_house.earnings
use	commercial_house.use
property_right	residential_house.property_right
residence	public_rental_house.residence
subsidy	public_rental_house.subsidy

(b) Relation mapping

RDF.relation	RDB.relation	Join.expression
detail	detail	house.Id = state.Id
work	work	builds.builderCom = builder.Company
accomplish	accomplish	builds.Id = project.Id
create	create	project.builderCom = builder.Company

映射描述在 SPARQL 到 SQL 的转换中如何发挥作用可通过实例表明，例如，一个用户提出了一个问题，“秀水花城项目所创设的开发商地址在哪里?”，对于这个问题生成 SPARQL 查询转换为 SQL 查询时，如图 4 所示。由于模式层面映射，从 RDF 关系图找到对应的查询模式图，构造 SPARQL 语句很容易，接着借助映射描述表中的谓词含义，“秀水花城”作为 name 分别具有对象值术语 project.Name 和链接谓词 create 代表的连接表达式 project.builderCom = builder.Company。在一个序列中，谓词名称介于?p 和?o 被推导为 builder.Town，这样的过程就能够生成 SQL 语句。

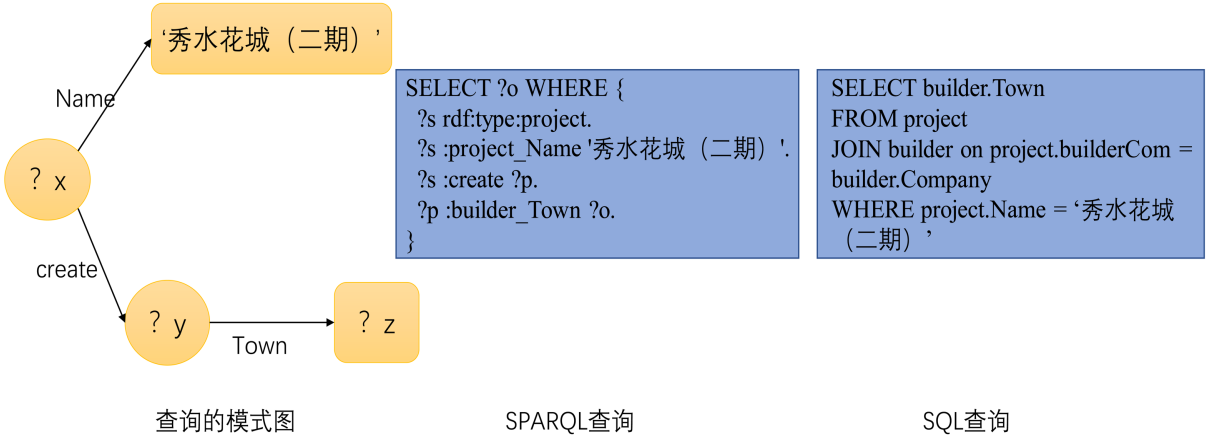

Figure 4. SPARQL converted to SQL instance
图 4. SPARQL 转换为 SQL 的实例

连接表达式以高效和简单的方式促进复杂表关系的规范化，以便进行数据库管理。总体来说，映射

描述很容易实现，因为有映射描述，所以将 SPARQL 转换为 SQL 的过程将会变得简单容易实现。

3.3. SPARQL to SQL

要实现按需映射，动态地生成 RDF 数据，就需要将每个用户提出的问题从 SPARQL 查询的形式转换为 SQL 查询语句，作用于 RDB 上。首先分析 SPARQL 查询语句，将查询语句中的三元组模式提取出来，获取到查询语句的主体、谓词和对象，接下来用上文定义的映射描述，将 SPARQL 转换为 SQL，作用于 RDB 数据上，从而能够在 RDB 数据中得到想要的数据信息，将得到的数据利用数据物化转换为 RDF 三元组格式，并存储在三元存储库中，这样就得到了 RDF 数据，再将 SPARQL 查询重新运用在此三元组数据上，就能得到想要的最终查询结果。按需映射只需要一段信息，就能将 SPARQL 查询转换为 SQL 查询。算法 SPARQLtoSQL 具体实现过程如下：

算法 1. SPARQLtoSQL convert

```
Input: SparqlQuery              // SPARQL 查询
Output: SqlQuery            // SQL 查询
Begin
    query ←parseSparqlQuery(sparqlQuery);      // 解析 SPARQL 查询
  sqlQuery=" ";                                // 初始化 SQL 查询字符串
    for t in query.triplePatterns:                    // t 为三元组模式
       subject = t.subject;
        predicate = t.predicate;
        object = t.object;                            // 获取 RDF 三元组的主谓宾

       tableName ← obtaintableName (subject);                 // 获取表、列名
columnName   ← obtaincolumnName (predicate);

sqlCondition = createSQLCondition(tableName, columnName, object);       // 生成 SQL 查询条件
       sqlQuery += sqlCondition;                      // 添加到 SQL 查询字符串
      Join_expression ← obtainexpression (RDF);      // 根据映射描述设计的连接表达式
      sqlQuery = "SELECT * FROM tableName " +Join_expression + " WHERE " + sqlQuery; // 生成 SQL
      return sqlQuery
End
```

3.4. RDB-RDF 转换算法

系统提出了基于 RDB-RDF 模式映射的数据转换方法，整体思路为从一个空的 RDF 三元组存储开始，对每个用户提出的 SPARQL 查询，首先将这个 SPARQL 查询应用到三元组存储库上，若没有答案，则说明目前是空状态，可以开始整个映射流程。我们借助映射描述，引入连接表达式，生成包含 SPARQL 查询中指定的三元组的 SQL 查询，将这些查询应用于 RDB，并将回答的 RDB 数据转换为三元组，我们将这些三元组存储在三元存储库中，最后，再对此三元存储库重新应用 SPARQL 查询，从而得到最终答案。在这个过程中，数据物化是逐步完成的，也避免了数据全部转换，而且需要存储的数据也大幅减少。也因为有连接表达式的存在，使得 SPARQL 向 SQL 的转换更加灵活简单。全文的主要贡献是避免了数据的全部转储，用到的是部分转换而不是整个数据转换。图 5 说明了基于模式的 RDB-RDF 映射的系统结构设计。

在转换过程中，每次的查询转换都是基于最新的 RDB 数据，也解决了传统方法转换时 RDB 数据更新造成的数据延迟问题。整体实现如算法 2 所示：

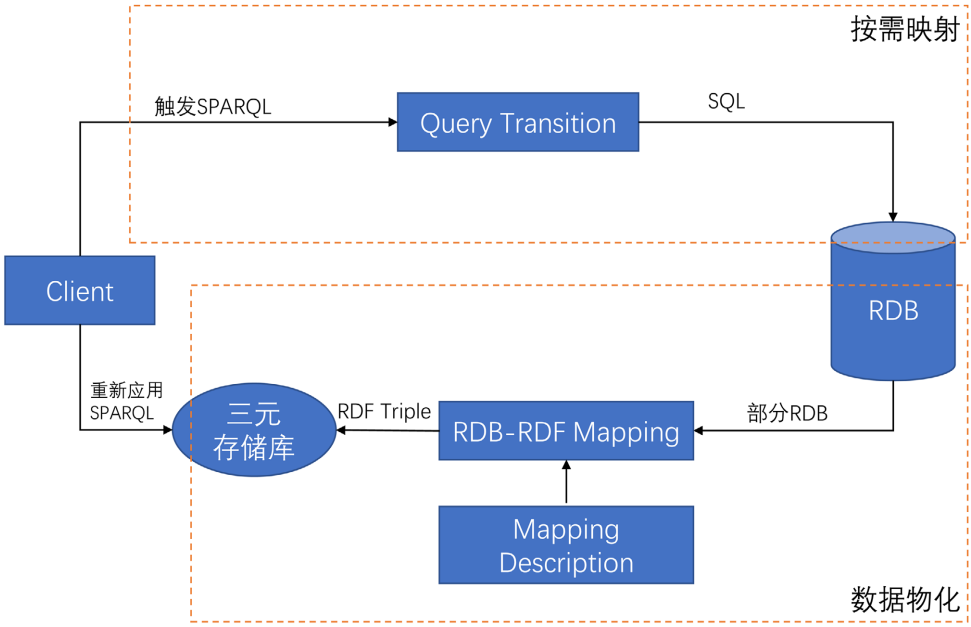

Figure 5. RDB-RDF mapping system architecture diagram
图 5. RDB-RDF 映射系统架构图

算法 2. RDBtoRDF_Conversion

```
Input: SparqlQuery              // SparqlQuery 是一个 SPARQL 查询
Input: TE                  //TE(物化的)三重存储(初始化 TE 为空)
Output: result             //result 是 SparqlQuery 查询对应的结果集

Begin
   result ← Apply_Sparql_on (TE);       //初始化
   IF result = ∅ Then
       rdfModel ← convert_SchemaToRDF(RDB);   // 关系数据库模式转换为 RDF 模式
   Endif
        SqlQuery ← convert_SparqlToSql(SparqlQuery);     //SPARQL 查询转换为 SQL 查询
        resultSet ← executeSqlQuery(RDB);     //执行 SQL 查询
        resultModel ← materializeToRDF(resultSet,rdfModel);     //将 RDB 结果物化为 RDF 形式
        TE ← TE∪resultModel;                                    //将结果存储在 TE 中
   result ← Apply_Sparql_on (TE);                               //重复请求
End
```

4. 实验评估分析

在提出基于 RDB-RDF 模式映射数据转换的方法之后，需要对算法的整体效果进行评估，从性能分析和语义保持两个方面对构建的方法进行评估。

4.1. 数据集

本系统选用了关系数据库(扩充了对象关系数据库)作为构建实验的数据库。选取的数据集是具体的房地产数据集，经过一系列的清洗筛选最终得到实验所需的数据集。利用该数据集建立了三个不同规模的数据源。表 3 给出了三个数据源的详细信息。表中的每一列代表相应数据表中的数据的数量。

Table 3. Information on the number of data sources selected for experiments
表 3. 实验选取数据源数量信息

名称	builder	project	builds	takebuilds	build	house	state
source1	16	16	16	16	73	8171	4
source2	32	32	32	32	202	13,704	4
source3	65	60	66	66	404	27,197	4

4.2. 性能分析

4.2.1. 构建 RDF(S)所用时间

设置了三个不同规模的数据源，分布构建 RDF(S)，记录不同数据源构建得到 RDF(S)所花费时间。对每个数据源都进行多次构建实验，然后多次所得时间的平均值进行记录，最终得到不同规模数据源构建 RDF(S)所用时间如图 6 所示。

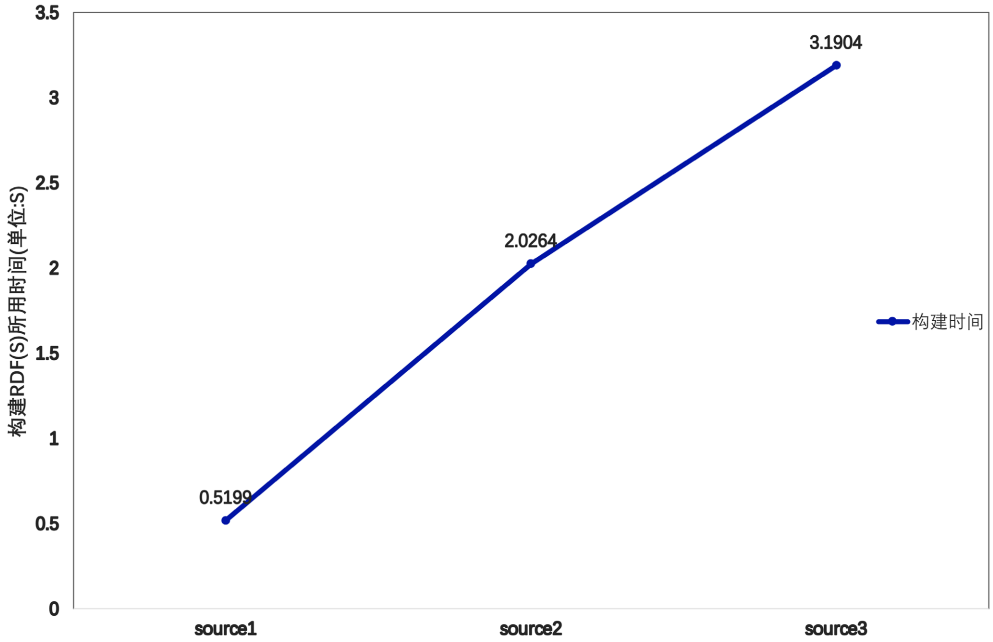

Figure 6. Time spent on building RDFS by RDB-RDF approach
图 6. RDB-RDF 方法构建 RDFS 花费时间

4.2.2. 对比分析

将传统的方法(直接映射 DM 和 R2RML)与提出的基于 RDB-RDF 模式映射的数据转换方法对不同规模的数据源的构建过程所用时间进行对比，得到的结果如表 4 所示。图 7 为不同方法对于不同数据源构建所用时间的折线图，直观地看出本文提出的方法节省了时间，比传统的方法耗时少。

4.3. 查询结果对比

本文的方法不仅能实现将关系数据库转换为资源描述框架，还能实现传统的方法(直接映射 DM 和 R2RML)无法实现的将对象关系数据库转换为资源描述框架，对关系数据库进行了扩充，实现了这种转换，并且实验表明，这种转换方法能够实现这二者的无损转换。

Table 4. Comparison of data conversion time
表 4. 数据转换时间对比

Method	Data transformation time (s)		
	source1	source2	source3
DM	0.7941	3.2476	4.7597
R2RML	0.6238	2.5523	3.7406
RDB-RDF	0.5199	2.0264	3.1904

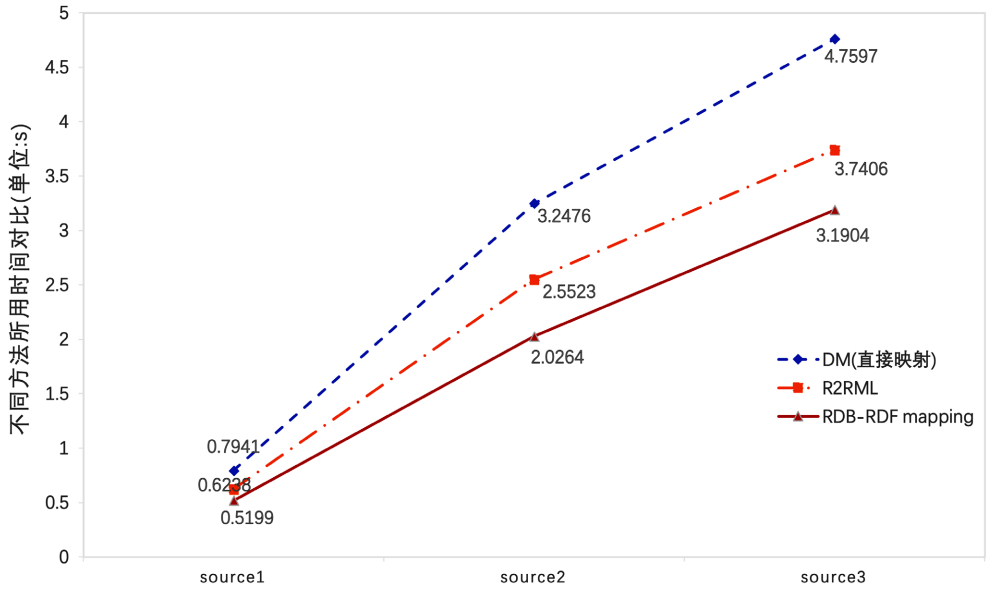

Figure 7. Comparative analysis of time spent on data conversion by different methods
图 7. 不同方法数据转换所用时间对比分析

所以为了检验这个映射过程是否成功将关系数据库和对象关系数据库成功映射为 RDF 实例，需要对关系数据库中包含的实例数据与映射得到的 RDF(S)文件中包含的实例信息进行对比。所以本文从不同的查询角度出发设计多组查询条件，对映射得到的实例与关系数据库中的存储的实例数据进行对比。本文设计了一组等价的 SPARQL 查询和 SQL 查询，分别施加在生成的 RDF 文档和原关系数据库上来判断数据转换的准确性。设计的查询条件如下：

1) 对于基本实体表中实例数据的映射情况进行检验

Q1：SELECT Company,Town FROM builder；(查询开发商名称与地址)

2) 对继承实体表中实例数据的映射情况进行检验

Q2：SELECT COUNT(Id) FROM public_rental；(查询 public_rental 表中有多少房屋)

3) 对引用关系表中实例数据的映射情况进行检验

Q3：SELECT project.Name FROM builder

JOIN project ON builder.Company = project.builderCom

WHERE builder.Company = '沈阳富景房产开发有限公司'；(查询沈阳富景房产开发的项目名称是什么)

Q4: SELECT state.roomstate FROM state

JOIN house ON house.Id = state.Id

WHERE house.Id = 3712971；(查询房屋号为 3712971 目前的房屋状态是什么)

4) 对关联关系表中实例数据的映射情况进行检验

Q5: SELECT COUNT(takebuilds.Id) FROM takebuilds

JOIN builder ON takebuilds.Company = builder.Company

JOIN builds ON takebuilds.Id = builds.Id

WHERE builder.Company = '沈阳华凌房地产有限公司'；(查询沈阳华凌房地产开发的盘有几个)

对于关系数据库进行 Q1~Q5 查询，得到的查询结果如表 5 所示。本文利用 Q1~Q5 的查询条件对生成的 RDF(S)文件进行 SPARQL 查询。查询得到的统计结果与表中的结果是完全一致的，这就表明在对象关系数据库中存储的实例数据进行映射的过程中没有发生数据的丢失。

Table 5. Database query results
表 5. 数据库查询结果

Q1	Q2	Q3	Q4	Q5
65	895	居住商业(二期)	已售	2

本文选取 Q1 和 Q5 两个查询条件作为示例进行进一步分析。Q1 的查询条件对于 builder 数据表中包含多少个开发商进行查询，该数据表中包含 65 个开发商。图 8 给出了对关系数据库查询 builder 数据表中部分开发商详细信息的结果。

company	Town
亚产(沈阳)房地产开发有限公司	皇姑区
恒大鑫源(沈阳)置业有限公司	于洪区千山西路北侧
沈阳万科朗汇置业有限公司	沈阳市铁西区卫工北街44号
沈阳万科西盛置业有限公司	于洪区
沈阳万科金域蓝湾房地产开发有限公司	东陵区
沈阳东北总部基地开发建设有限公司	沈北新区
沈阳东大兴科置业有限公司	和平区
沈阳东环置业有限公司	铁西区云峰北街
沈阳中冶京诚置业有限公司	沈北新区
沈阳中韩科技产业团地发展有限公司	大东区
沈阳九洲福尔房地产开发有限公司	和平区
沈阳五洲国际工业博览城置业有限公司	于洪区
沈阳其仕和悦房地产有限公司	沈阳市沈河区方家栏路
沈阳凯盈汇源房地产开发有限公司	沈河区
沈阳加州阳光花园房屋开发有限公司	于洪区怒江北街39号

Figure 8. Partial results obtained from Q1 query on database
图 8. 对数据库进行 Q1 查询得到的部分结果

根据提出的构建规则，builder 数据表作为基本实体表会被映射称为 builder 类，而表中所有的数据会被映射成为 builder 类的实例及其相应属性。所以对于得到的 RDF(S)文件中的数据进行相应查询得到的结果如图 9 所示，查询语句如下：

SELECT ?p ?o WHERE {

?s rdf:type:builder.

?s :builder_Company ?p.

?s :builder_Town ?o.

}

根据 SPARQL 查询语句可以得知，图 9 中的 p 列表示 Company 属性的值，o 列表示 Town 属性的值。比较图 8 与图 9 中的信息可以发现，数据是一致的，说明 builder 表中数据被构建成了 RDF 实例。

p	o
亚产(沈阳)房地产开发有限公司	皇姑区
恒大鑫源(沈阳)置业有限公司	于洪区千山西路北侧
沈阳万科朗汇置业有限公司	沈阳市铁西区卫工北街44号
沈阳万科西盛置业有限公司	于洪区
沈阳万科金域蓝湾房地产开发有限公司	东陵区
沈阳东北总部基地开发建设有限公司	沈北新区
沈阳东大兴科置业有限公司	和平区
沈阳东环置业有限公司	铁西区云峰北街
沈阳中冶京诚置业有限公司	沈北新区
沈阳中韩科技产业团地发展有限公司	大东区
沈阳九洲福尔房地产开发有限公司	和平区
沈阳五洲国际工业博览城置业有限公司	于洪区
沈阳其仕和悦房地产有限公司	沈阳市沈河区方家栏路
沈阳凯盈汇源房地产开发有限公司	沈河区
沈阳加州阳光花园房屋开发有限公司	于洪区怒江北街39号

Figure 9. Results obtained by performing Q1 query on RDF file
图 9. 对 RDF 文件进行 Q1 查询得到的结果

同样对于Q5查询语句得到的结果进行详细分析。Q5 主要对于关联关系表表达的语义信息进行查询，该语句查询得到开发商“沈阳华凌房地产有限公司”开发的项目有 2 个，图 10 给出了这两个项目的 Id 和 projectName 的详细信息。

Id	projectName
49602	居住、商业（五期-7）
64546	居住、商业（五期-1）0009

Figure 10. Detailed results obtained by performing Q5 query on the database
图 10. 对数据库进行 Q5 查询得到的详细结果

根据提出的构建规则，takebuilds 是关联实体表，外键分别引用 builds.Id 和 builder.Company，takebuilds 表会被映射成为 RDF Schema 中属性，所以对于开发商 builder.Company 为“沈阳华凌房地产有限公司”开发的所有项目进行查询的 SPARQL 语句如下：

SELECT ?p ?o WHERE {

?s rdf:type:builder.

?s :builder_Company '沈阳华凌房地产有限公司'.

?p :takebuilds ?s.

?p :builds_projectName ?o.

}

该 SPARQL 语句查询得到的结果如图 11 所示。

p	o
db:builds/49602	居住、商业（五期-7）
db:builds/64546	居住、商业（五期-1）0009

Figure 11. Detailed results obtained by performing Q5 query on RDF file
图 11. 对 RDF 文件进行 Q5 查询得到的详细结果

通过分析 SPARQL 语句可知，图 11 中的 p 列表示 builds 类的 Id，o 列表示该实例的 projectName 属性。经对比，图 10 与图 11 中得到的信息完全相同。这表明映射得到的 RDF(S)文件完整保留了关系数据库的语义信息。数据转换的精确性较好。以上内容证明提出的 RDB-RDF 模式映射的数据转换方法能够实现对象关系数据库的无损转换，传统的方法只能实现的是关系数据库的转换，本文的方法两个层面都能够实现转换。

5. 总结

本文提出了一种基于 RDB-RDF 模式映射的数据转换方法，转换时首先根据关系数据库和 RDF(S)数据的特点，给出二者的形式化定义，将模式层面和实例层面关注点分离开来，分别研究转换方法，使整个转换过程分块进行，更容易理解；同时引入映射描述和连接表达式，使得 SPARQL 转换为 SQL 查询的过程变得简单快速，整个方法结合了数据物化和按需映射，不需要转换整个数据库中的数据，只针对 SPARQL 查询中的部分数据，减少了存储空间。同时本文还对关系数据库进行了扩充，实现了对象关系数据库转换为资源描述框架，这是传统方法不能够做到的内容。实验表明这种方法具备语义完整性，整个过程更加高效。本文的思路还是存在许多的不足，未来的工作方向可以立足于多个关系数据库来实现，设立能够交互的关系数据库，从而实现多个异构数据的融合，进一步满足用户的需要。

参考文献

[1] Berners-Lee, T., Hendler, J. and Lassila, O. (2001) The Semantic Web. *Scientific American*, **284**, 34-43. https://www.scientificamerican.com/article/the-semantic-web https://doi.org/10.1038/scientificamerican0501-34

[2] Nesrine, L., Mimoun, M., Ahmed, L., *et al.* (2017) On Demand ETL of RDB to RDF Mapping for Linked Enterprise Data. *International Journal of Strategic Information Technology and Applications*, **8**, 91-100. http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJSITA.2017070106 https://doi.org/10.4018/IJSITA.2017070106

[3] Arenas-Guerrero, J., Scrocca, M., Iglesias-Molina, A., *et al.* (2021) Knowledge Graph Construction with R2RML and RML: An ETL System-Based Overview.

[4] A Direct Mapping of Relational Data to RDF. https://www.w3.org/TR/rdb-direct-mapping/

[5] R2RML: RDB to RDF Mapping Language. https://www.w3.org/TR/r2rml/

[6] Unbehauen, J., Stadler, C. and Auer, S. (2013) Optimizing SPARQL-to-SQL Rewriting. *Proceedings of International Conference on Information Integration and Web-Based Applications & Services*, Vienna, 2-4 December 2013, 324-330. https://doi.org/10.1145/2539150.2539247

[7] Zhang, F., Ma, Z.M. and Yan, L. (2011) Construction of Ontologies from Object-Oriented Database Models. *Inte-*

grated Computer Aided Engineering, **18**, 327-347.

[8] Idrissi, B.E. (2022) RDF/OWL Storage and Management in Relational Database Management Systems: A Comparative Study. *Journal of King Saud University—Computer and Information Sciences*, **34**, 7604-7620. https://doi.org/10.1016/j.jksuci.2021.08.018

[9] 张永威, 张岩, 唐新余, 等. 关系型数据的知识抽取和 RDF 转换框架及实现[J/OL]. 计算机工程与应用, 2022, 58(17): 213-223. https://kns.cnki.net/kcms2/article/abstract?v=3uoqIhG8C44YLTlOAiTRKibYlV5Vjs7iJTKGjg9uTdeTsOI_ra5_XQaS6ni5SKIh-2f6XO3TvG8CVmb6O0b3JVH9tVEoYw7r&uniplatform=NZKPT

[10] Siow, E., Tiropanis, T. and Hall, W. (2016) SPARQL-to-SQL on Internet of Things Databases and Streams. In: Groth, P., Simperl, E., Gray, A., *et al*., Eds., *The Semantic Web—ISWC* 2016, Vol. 9981, Springer International Publishing, Cham, 515-531. https://link.springer.com/10.1007/978-3-319-46523-4_31 https://doi.org/10.1007/978-3-319-46523-4_31

[11] Pequeno, V.M., Vidal, V.M.P., Casanova, M.A., *et al*. (2014) Specifying Complex Correspondences between Relational Schemas and RDF Models for Generating Customized R2RML Mappings. In: *Proceedings of the* 18*th International Database Engineering & Applications Symposium on IDEAS*'14, ACM Press, Porto, 96-104. http://dl.acm.org/citation.cfm?doid=2628194.2628233 https://doi.org/10.1145/2628194.2628233

[12] Sengupta, K., Haase, P., Schmidt, M., *et al*. (2013) Editing R2RML Mappings Made Easy.

[13] Sequeda, J., Tirmizi, S., Corcho, O. and Miranker, D. (2011) Survey of Directly Mapping SQL Databases to the Semantic Web. *The Knowledge Engineering Review*, **26**, 445-486. https://www.cambridge.org/core/journals/knowledge-engineering-review/article/abs/survey-of-directly-mapping-sql-databases-to-the-semantic-web/0688CCA9A831376C4EE5DCE09382563B

[14] 鲁佳文, 严丽. 对象关系数据库到 RDF(S)的映射方法[J]. 计算机科学, 2021, 48(10): 145-151.

[15] Hazber, M.A.G., Li, R., Xu, G., *et al*. (2016) An Approach for Automatically Generating R2RML-Based Direct Mapping from Relational Databases. In: Che, W., Han, Q., Wang, H., *et al*., Eds., *Social Computing*, Vol. 623, Springer, Singapore, 151-169. http://link.springer.com/10.1007/978-981-10-2053-7_15 https://doi.org/10.1007/978-981-10-2053-7_15

[16] Mathur, S.N., O'Sullivan, D. and Brennan, R. (2018) Milan: Automatic Generation of R2RML Mappings.

[17] 王嘉庆, 杨卫东, 何亦征. 关系数据库的实体间关系提取方法的研究[J]. 计算机应用与软件, 2019, 36(10): 10-16, 38.

[18] Michel, F., Montagnat, J. and Faron-Zucker, C. (2014) A Survey of RDB to RDF Translation Approaches and Tools.

[19] SPARQL by Example. https://www.w3.org/2009/Talks/0615-qbe/

[20] Abatal, A., Alaoui, K., Alaoui, L., *et al*. (2019) SQL2SPARQL4RDF: Automatic SQL to SPARQL Conversion for RDF Querying. In: *Proceedings of the* 4*th International Conference on Big Data and Internet of Things*, ACM, Rabat, 1-9. https://doi.org/10.1145/3372938.3372968

[21] Natarajan, S., Vairavasundaram, S., Teekaraman, Y., *et al*. (2021) Schema-Based Mapping Approach for Data Transformation to Enrich Semantic Web. *Wireless Communications and Mobile Computing*, **2021**, Article ID: 8567894. https://www.hindawi.com/journals/wcmc/2021/8567894/ https://doi.org/10.1155/2021/8567894