International Journal of Engineering & Technology, 7 (2.33) (2018) 84-88

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Sparql query processing in relational databases

Ju Ri Kim [1] *, Zhanfang Zhao [2], Sung-Kook Han [3]

[1] College of Liberal Arts, WonKwang University, City Iksan, JeonBuk, 54538, Korea
[2] College of Information Engineering, Hebei GEO University, Hebei, 050031, China
[3] Department of Computer Engineering, WonKwang University, City Iksan, JeonBuk,54538, Korea
**Corresponding author E-mail: cyanic@wku.ac.kr*

Abstract

Background/Objectives: The mapping RDB to RDF has become important to populate Linked Data more efficiently. This paper shows how to implement SPARQL endpoint in RDB using a conceptual level mapping approach.
Methods/Statistical analysis: Many diverse approaches and related languages for mapping RDB to RDF have been proposed. The prominent achievements of mapping RDB to RDF are two standard draft Direct Mapping and R2RML proposed by W3C RDB2RDF Working Group. This paper analyzes these conventional mapping approaches and proposes a new approach based on schema mapping. The paper also presents SPARQL query processing in RDB.
Findings: There are distinct differences between instance level mapping and conceptual level mapping for RDB2RDF. Data redundancy of instance level mapping causes many inevitable problems during mapping procedure. The conceptual level mapping can provide straightforward and efficient way. The ER model in RDB and RDF model in Linked Data have obvious similarity. The ER model describes entities and relationships, which is the conceptual schema of RDB. RDF model consists of three parts: subject, predicate and object, which is the standard model for data interchange on the Web. The entities in ER model and subjects in RDF model are all the things that can be anything in the real world. Both the relationships in ER model and predicates in RDF model describe the relations between things.
Since RDB and RDF share the similar modeling approach at the schema level, it is reasonable that mapping approach should be based on RDB schema. This kind of conceptual level mapping also can provide efficient SPARQL query processing in RDB.
Improvements/Applications: The paper realizes SPARQL query processing in RDB, which is based on conceptual level mapping. The query experiments show that it is a concise and efficient way to populate Linked Data.

Keywords: *SPARQL, RDF; Triple; Linked Open Data; Schema Mapping.*

1. Introduction

The Linked Open Data (LOD) has emerged as a powerful enabler to realize Web of open, shared, interlinked data and, ultimately, the vision of Semantic Web [1-2]. To realize Web of Data based on LOD, LOD data sets from the diverse domains play the key role in expanding the global data space. Since the vast amount of useful data are still stored in relational databases (RDB), one of the most efficient ways to populate LOD sets is to map data in relational databases into RDF (Resource Description Framework) of Linked Data. Due to the importance of mapping RDB to RDF (RDB2RDF), the various mapping approaches have been proposed [5-7]. Most of mapping approaches provide the SPARQL query to process the RDB data. However, the conventional mapping RDB data to RDF has not yielded the significant achievement as was predicted, and has revealed some limitations and problems3. This paper proposes a new modeling method based on schema translation for RDB to RDF. Since the conceptual schema of RDB is similar to ontological modeling of a certain domain, RDB schema mapping into RDF is more effective than the conventional instance-based direct mapping approaches, which can overcome some shortcomings of instance-based mapping approaches.
In the following section, we discuss the conventional mapping methods and their features, and review related technique and mapping language. In section [3], we introduce the triple modeling of relational databases and propose schema-based mapping model. The mapping language is provided. Section 4 discusses the SPARQL query procedure of schema-based mapping approach and introduces the processing of SPARQL query by examples. Section 5 concludes with the features of schema-based mapping method.

2. Materials and methods

Many diverse approaches [8-9], techniques, languages [4], [15], [16], [17] and corresponding tools [13-14] for mapping RDB to RDF have been proposed over the last decade. The W3C RDB2RDF Working Group has proposed two standards to support LOD data publication: Direct Mapping and R2RML (Relational Database to RDF Mapping Language) [10-11]. Direct Mapping is the recommended approach to directly translate RDB data and its schema to RDF representation. R2RML is a generic language for describing a set of customized mappings that transform RDB data into RDF datasets. However, the proposed mapping has revealed the difficulties in handling the complex relationships of RDB tables, foreign key constraints and SQL query generation [12-13].
Direct Mapping is instance-based approaches, which means that it would generate large number of RDF graphs. Figure 1 shows an example of Direct Mapping method.

(A) Relational Table

ID	Name	DID	Director	Age
201601	John	A01		35
201617	Maddy	A01	201601	29

(B)The RDF Graphs Corresponding To A Tuple in (A).

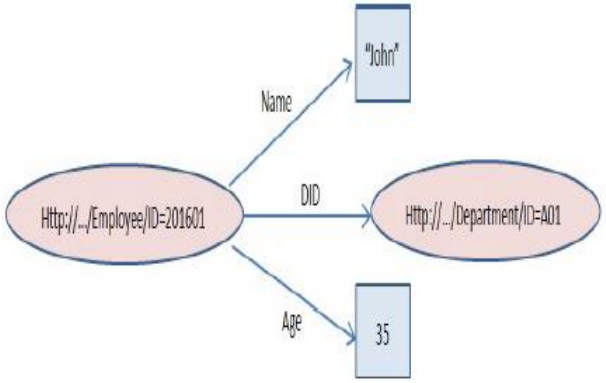


Fig. 1: Direct Mapping Approach.

In Figure 1, subfigure (a) shows a relational table "Employee" with two tuples. Subfigure (b) shows a group of RDF graphs that correspond to one tuple in the relational table. These groups of RDF graphs have the same subject that generated from the primary key name and primary key value of relational table. The predicates in RDF graphs generated from the column names of non-primary key, and the objects in RDF graphs generated from the cell values of relational table. The objects have two types. If the columns in relational table are ordinary columns, the objects generated from cell values are leaf nodes like the values of name and age. If the columns are foreign keys like DID column, the object generated from the foreign key column value is also a subject of another triple structure such as the node of department "http://.../Department/ID=A01".

As shown in Figure 1, each tuple in relational table generates a set of RDF graphs, which means the instance-based mapping approach will cause a lot of problems when the data update operation occurs. Schema-based mapping approach can effectively avoid the generation of such problems.

ever necessary ethical clearance obtained must be stated.

3. Triple modeling of relational databases

This section will analysis the characteristics of ER and RDF models, construct the triple model of relational databases, and propose schema-based mapping approach from RDB to RDF. The mapping model is described by examples.

3.1. ER and RDF models

ER model: the Entity-Relationship (ER) model defines the conceptual schema of RDB. The ER model presents a data schema in graphical way, which is usually depicted in a graphical way as boxes (entities) that are connected by edges (relationships). An entity is a thing capable of an independent object of a domain that can be uniquely identified. The relationships express the associations and dependencies between entities.

Noted that the diagrams included properties, relationships and entities may be called as entity-property-relationship diagrams, rather than ER model. An ER model is typically implemented as a database. Each row in a table represents one instance of an entity, each field in a table represents an attribute, and a PK-FK pair represents a relationship.

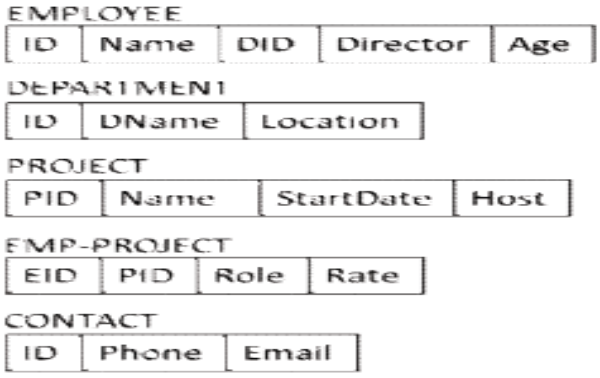
RDF model: RDF (Resource Description Framework) is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ. LOD instances uses a RDF graph-based model based on the concepts of resources and triples. A resource that can be anything can be identified by a URI (Uniform Resource Identifier). RDF triples consist of three components: a subject, a predicate and an object. A set of RDF triples consists of an RDF graph that forms a directed labelled graph. Nodes represent subjects and objects of RDF triples, while predicates are arcs pointing from the subject to the object of the triple.

Similarity of ER and RDF models: there are similarity between ER model and RDF mode from conceptual model perspective. The entity in ER model and subject in RDF are all the abstraction for a thing that can be anything in the real world. The relationship in ER and predicate in RDF are all the relationship description of things. Both of the two models can express the facts of real world. Therefore, it is feasible to build conceptual level mapping (also known as schema level mapping) from ER model to RDF model, which may be called as triple model of relational databases

3.2. Triple modeling of relational databases

Since RDB and RDF share the similar modeling approach at the schema level, it is reasonable that mapping approach should be based on RDB schema. This provides a consistent way of mapping and makes it possible to preserve information and conceptual structures of RDB in the process of the mapping. The typical example containing complex relationships among the tables in Figure 2 shows that theschema-based mapping is more succinct and effective than the conventional direct mapping.

(A) RDB Schema



(B) RDF Graph By Schema-Based Mapping

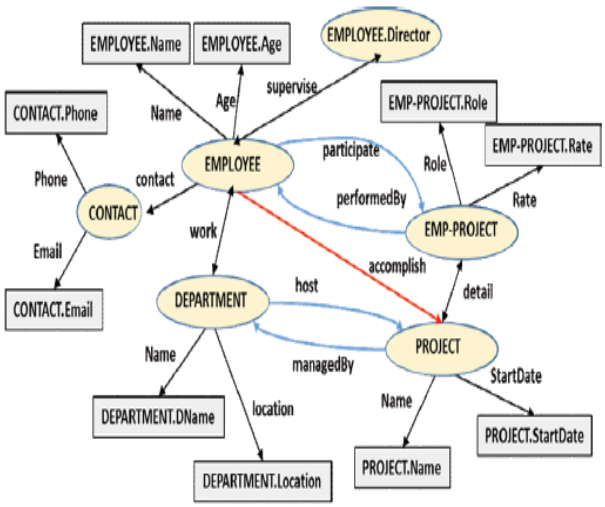


Fig. 2: A Typical Example of Schema-Based Mapping.

In Figure 2, subfigure (a) shows the RDB Schema, including five entities: EMPLOYEE, DEPARTMENT, PROJECT, EMP-PROJECT and CONTACT. The relationships are shown as following:

- One: M relationship. EMPLOYEE and CONTACT, DEPARTMENT and EMPLOYEE, DEPARTMENT and PROJECT, they all have 1: M relationship.
- One: M recursive relationship. EMPLOYEE and EMPLOYEE have recursive relationship. ID is primary key, and Director is foreign key.
- M: N relationship. In RDB Schema, M: N relationship is divided into two 1: M relationships. For example, EMPLOYEE and PROJECT have M: N relationship, which is divided into two, 1: M relationships: EMPLOYEE and EMP-PROJECT, PROJECT and EMP-PROJECT.

In Figure 2, subfigure (b) shows the RDF graph by Schema-based mapping. The oval nodes in RDF graph are subjects such as EMPLOYEE, "EMPLOYEE. Director", EMP-RPOJECT and CONTACT. Rectangular nodes in RDF graph are objects such as "EMPLOYEE. Name", "EMPLOYEE. Age" and "EMP-PROJECT. Role". The edges in RDF graph are relationships, which are classified as following:

- Entity Predicate. Entity predicates are the ordinary column names in relational table, which are neither primary key nor foreign key such as "name", "age" and "phone".
- Self-Link Predicate. Self-link predicates express the recursive relation between relational tables such as "supervise".
- Table-Link Predicate. Table-link predicates express the non-recursive relation between two relational tables such as "detail", "accomplish", "participate", "performed By" and "host".

3.3. Schema-based mapping approach from RDB to RDF

Schema-based mapping is to realize conceptual level mapping from ER mode to RDF model, also known as mapping from RDB to RDF, which preserves the structure information of RDB and generates instances only when SPARQL query operation occurs. The mapping rules are as follows.

- Table name to subject. A table name in RDB corresponds to a subject in RDF graph.
- Column name to predicate. A column name in RDB corresponds to a predicate in RDF graph.
- Ordinary columns (non-primary key and non-foreign key) correspond to entity predicates.
- PK-FK columns correspond to self-link predicates or table-link predicates.
- Cell value to object. A data in cell of RDB corresponds to object in RDF graph.
- In the mapping process, there are still some rules as follows.
- Make full use of common ontology vocabularies such as FOAF, DC, and common metadata vocabularies in the "schema.org" website. Common ontology and metadata vocabularies have explicit meanings about defined concepts, so using common vocabularies can promote the sharing of knowledge and integration of data from different RDF sources. In addition, the common vocabularies are formal because it is machine-readable and interpreted which conductive to the application of RDF data. For example, the "EMPLOYEE.name" corresponds to "FOAF.name"; the "BOOK.author" corresponds to "DC.creator".
- Provide as many as possible, replaceable, human-readable predicates. For some column names, there are no common ontology vocabularies that can be referenced. As many predicates as possible should be provided for meet requirement of different users.

For example, the relation "accomplish" between EMPLOYEE and PROJECT, can be replaced by "participate" and "performedBy" between EMPLOYEE and EMP-PROJECT. The relation "host" between DEPARTMENT and PROJECT can be replaced by "managedBy" relation.

- Link Predicate should bind object information. Link Predicates consist of self-link predicates and table-link predicates, which correspond to join operation of many tables in RDB. Object information expresses the conditional expression of join operation of RDB. The object information will be used for transformation engine from SPARQL to SQL.

For example, relation "supervise" binds the expression "EMPLOYEE.ID = EMPLOYEE.Director". Relation "accomplish" binds the expression "EMPLOYEE.ID = EMP-PROJECT.EID and PROJECT.ID = EMP-PROJECT.PID".

3.4. Description of mapping language

The mapping language is in JSON format, which describe the mapping rule in detail and generate mapping file. The mapping file will provide the service for transformation engine when SPARQL query occurs. The part of mapping file of Figure 2 is shown in Figure 3.

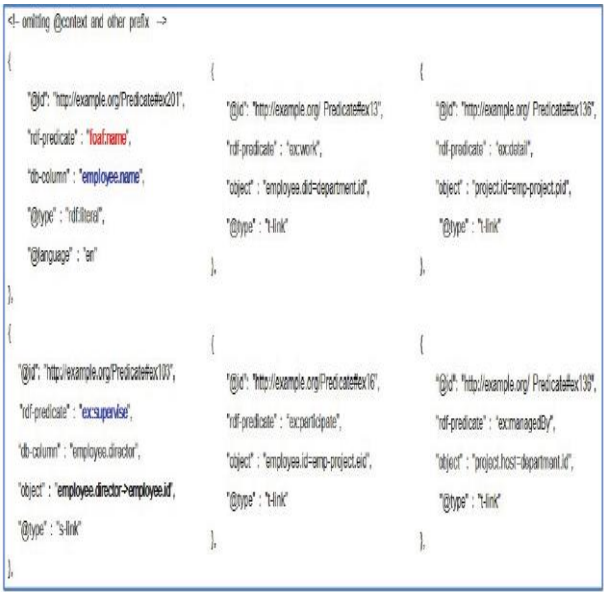


Fig. 3: Mapping File Corresponds to the RDF Graph in Figure 2

Figure 3 is only the part of whole mapping file, which is description about predicates. Each pair of braces describes the mapping information of a predicate, which include ID of predicate, rdf-predicate, corresponding column name in RDB, datatype and language.

If the predicate is Entity Predicate, the value of "@type" is a RDF datatype like "rdf:literal". If the predicate is Link Predicate, the value of "@type" is "s-link" or "t-link", which corresponds to recursive relation or PK-FK join relation respectively. Furthermore, the "object" item provides the conditional expression of join operation in RDB.

The mapping file records the mapping details from RDB to RDF, which provides the mapping rules to transformation engine when query operation is conducted. By using mapping file, the transformation engine translates SPARQL grammar into SQL grammar, and executes query against RDB.

4. Sparql query processing

This section introduces the basic procedure of SPARQL query, and shows the transformation result by query examples.

For SPARQL query, the first thing is to generate mapping file that is a global mapping view. Then transformation engine translates SPARQL query statements into SQL statement and execute query against RDB. At last, the transformation engine translates the query result of SQL into JSON-LD format. The translation of SPARQL query based on schema-based mapping into SQL is straightforward, since each node of RDF graph represents the table of RDB and predicates represent column or link relation.

The following some query examples explain the processing of query.

Fig. 4: SPARQL Query Processing in RDB.

In Figure 4, subfigure (a) shows the translation for the query of 1: N relations. The query problem is "What is the name of department where John works in?" Predicate "work" is a table-link predicate with the conditional expression "EMPLOYEE.DID = DEPARTMENT.ID" Subfigure (b) shows the translation for M: N relations. The query problem is "What are the names of projects that John accomplished?" Predicate "accomplish" binds the conditional expression "EMPLOYEE.ID = EMP-PROJECT.EID and EMP-PROJECT.PID = PROJECT.ID"

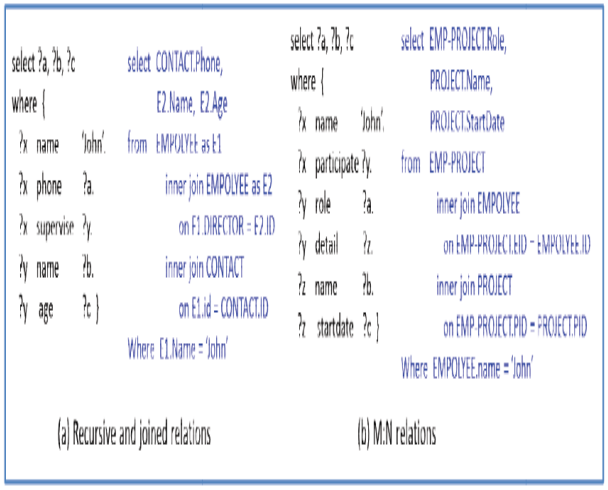


Fig. 5: SPARQL Query Processing in RDB.

Figure 5 shows two complex query transformations: recursive, and join operation between three tables. Subfigure (a) display the query "What are John's phone, name of his director and age of his director respectively?" There is a recursive relation between "ID" and "Director" in relational table. Subfigure (b) shows a M: N relations between EMPLOYEE, PROJECT and EMP-PROJECT. Its query is "What are the name and start date of the projects that John participated? In addition, what are the John's roles in these projects? " In fact, the relation "accomplish" in subfigure (b) of Figure 4 and the relation "participate" in subfigure (b) of Figure 5 are identical relation in RDB. Mapping rules generate multiple predicates for the relation that cannot refer to the common vocabularies. At last, the engine will provide the query result in JSON-LD form.

The SPARQL query processing reveals the superiority of schema-based mapping approaches. It only realize SPARQL query by means of transformation engine, and do not generate data redundancy. The generated mapping file is succinct, efficient and machine-readable, which is the bridge between mapping model and query engine.

5. Conclusion

Making the data stored in RDB accessible to the Linked Data has been a promising research field during the last decade. Translating RDB to RDF so that it can be queried through SPARQL, is deemed to the important processing of RDB2RDF mapping. The classic mapping method, Direct Mapping, has few insurmountable defects when mapping occurs from RDB to RDF. In addition, the recommended mapping language R2RML still has some problems that need to be resolved. To date there is no a widely accepted RDB2RDF mapping method that can provide efficient and perfect mapping scheme.

This paper proposes a new and instinctive mapping method at the conceptual schema level. The proposed schema-based mapping can achieve more coherent mapping than the conventional direct mapping approaches by dissolving the operational differences. The paper shows how SPARQL query can be processed in RDB by means of the proposed schema-based mapping. The experiments show the schema-based mapping method is a succinct and efficient mapping approach.

6. Acknowledgment

- This paper was supported by WonKwang University in 2017

References

[1] Heath T, Bizer C.: Linked data: Evolving the web into a global data space [J]. Synthesis lectures on the semantic web: theory and technology, 2011, 1(1), pp.1-136.

[2] Bizer, C.: The emerging web of linked data. IEEE Intelligent Systems, 2009, 24 (5), pp. 87-92.

[3] Michel, F., Montagnat, J., Faron-Zucker, C.: A Survey of RDB to RDF Translation Approaches and Tools. Rapport de Recherche, ISRN I3S/RR 2013-04-FR, 2014.

[4] Hert, M., Reif, G., Gall, H.C.: A Comparison of RDB-to-RDF Mapping Languages. In Proceedings of the seventh International Conference on Semantic Systems, ACM, 2011, pp. 25-32.

[5] W3C RDB2RDF Incubator Group. A Survey of Current Approaches for Mapping of Relational Databases to RDF, Technical report, 2009.

[6] Chebotko, A., et al.: Semantics Preserving SPARQL-to-SQL Translation, Data & Knowledge Eng., 2009, pp. 973-1000.

[7] W3C RDB2RDF Incubator Group.:A Survey of Current Approaches for Mapping of Relational Databases to RDF, Technical report, 2009.

[8] Thuy P T T, Thuan N D, and Han Y, et al.: RDB2RDF: completed transformation from relational database into RDF ontology[C]//Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication. ACM, 2014, pp. 88.

[9] Chen L, Yao N.: Publishing linked data from relational databases using traditional views[C]//Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on. IEEE, 2010, 6, pp. 9-12.

[10] Arenas M, Bertails A, Prud E, et al.: A direct mapping of relational data to RDF [J], 2012.

[11] Das S, Sundara S, Cyganiak R.: R2RML: RDB to RDF mapping language [J], 2012.

[12] Sequeda J F, Arenas M, Miranker D P.: On directly mapping relational databases to RDF and OWL[C]//Proceedings of the 21st international conference on World Wide Web. ACM, 2012, pp. 649-658.

[13] De Medeiros L F, Priyatna F, Corcho O.: MIRROR: Automatic R2RML mapping generation from relational databases[C]//International Conference on Web Engineering, Springer, Cham, 2015,pp. 326-343.

[14] Priyatna F, Corcho O, Sequeda J.:Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph[C]//Proceedings of the 23rd international conference on World wide web, ACM, 2014,pp. 479-490.

[15] Rodriguez-Muro M, Rezk M.: Efficient SPARQL-to-SQL with R2RML mappings [J]. Web Semantics: Science, Services and

Agents on the World Wide Web, 2015, 33, pp. 141-169.
[16] Neto L E T, Vidal V M P, and Casanova M A, et al.: R2RML by assertion: A semi-automatic tool for generating customised R2RML mappings[C]//Extended Semantic Web Conference. Springer, Berlin, Heidelberg, 2013, pp. 248-252.
[17] Dimou A, Sande M V, Colpaert P, et al.: Extending R2RML to a source-independent mapping language for RDF[C]//Proceedings of the 2013th International Conference on Posters & Demonstrations Track-Volume 1035. CEUR-WS. Org, 2013, pp. 237-240.