International Journal of Information Sciences and Application.
ISSN 0974-2255 Volume 10, Number 1 (2018), pp. 23-36

http://www.irphouse.com

Comparative Study of RDB to RDF Mapping using D2RQ and R2RML Mapping Languages

[1]Runumi Devi, [2]Rohit Singh, [3]Vikram Pratap Singh

JSS Academy of Technical Education, Noida, Uttar-Pradesh, India

Abstract

Making data that reside in relational databases (RDB) accessible to the semantic web has been an active field in the domain of Computer Science. RDB to Resource Description Framework(RDF) mapping language plays a major role in bridging the conceptual gap between the relational model and RDF.RDB-to-RDF process can be referred to as a method of converting relational data into RDF or exposing relational data on semantic web so that it can be queried through Simple Protocol and RDF Query Language(SPARQL). SPARQL is the query language used to query RDF data on semantic web. However, semantic web developers face difficulty while choosing between the existing mapping languages for converting their data in a machine-readable format. The goal and contribution of this paper is to provide a comparison between two mapping languages-Database to RDF Query(D2RQ) and RDB to RDF Mapping Language(R2RML) considering type of mapping, their expressiveness, how mapping is being implemented and the data retrieval method used. Our comparison is based on use case which will help users while selecting a mapping language.

Keywords: RDF, SPARQL,D2RQ,OWL

1. INTRODUCTION

Tim Berners-Lee[2] had proposed a concepts-Linked Data principles recommending some best practices that can be used to expose data, make data sharable and connect pieces of data, information and knowledge on the semantic web. Linked data can be obtained by using RDB-to-RDF techniques. These techniques exposes relational data by converting into machine-readable format.

Furthermore, integration of heterogeneous data by making their semantics explicit has become a need of the hour.This requires data to be elevated to a format that capture and

expose semantics in an explicit and machine readable format. Under this circumstances, there is a need to study and implement various RDB-to-RDF methods for choosing the appropriate one.

Several studies have been conducted in the domain of relational database to RDF mapping, thatexplores the approaches along diverse axes: how mapping is described, how data is translated into RDF and how query is performed on RDF data.

2. RDB to RDF Mapping languages

Classification Sahoo et al[3] have discussed a reference framework that enables categorization of various RDB to RDF mapping approaches. The components of the reference framework include creation of mapping, mapping representation and accessibility, mapping implementation, query implementation, application domain and data integration.

Michel et al[1] have presented various strategies for producing additional knowledge from relational database that can be used to create semantically rich data thereby making data more usable, linkable and sharable.Hert etal[9] provided feature-based comparison of RDB-RDF mapping languages with the help of case studies.

2.1 Direct Mapping:

Direct mapping[9] language creates ad-hoc ontology that reflect the relational database. It extract ontological knowledge in the form of RDF Schema(RDFS) or Web Ontology Language(OWL) from Structured Query Language(SQL) Data Definition Language schema for translating relational database into RDF.It involves creation of Uniform Resource Identifier(URI) following rules defined by Tim Berners-Lee. A table is considered as an ontological concept and is translated to class with URI "namespace/database/table". Each column of a table is considered as ontological property and is assigned with URI as "namespace/database/table/column". Each row of a table is considered as resource whose URI is formed by using primary key as "namespace/database/table/primaryKey" or "namespace/database/table#primaryKey". Each cell having a literal value is considered as object of a data property whereas cell with a foreign key constraint is translated into object of an object property.

Usually Direct Mapping method is applied either when there is no domain ontology exists for mapping the relational schema, or when there is a urgent necessity of making the data sources available in a machine-readable format, with few concerns for semantic interoperability.

2.2 Domain Semantics-Driven mapping

In case of direct mapping[1], a mapping is created automatically which can be later customized manually.Unlike direct mapping, domain semantics-driven mapping frequently use direct mapping generatedautomatically as starting point for complex

domain-specific mappings.

Both direct and domain semantics-driven mapping, which is also termed as transformative mapping, are supported by R2RML. The language not only provide information for the processor to data materialization but also to rewrite SPARQL queries into SQL. Apart from the features mentioned, the language also supports features like: unique identifiers generation, data type conversions, renaming of column, tables to classes, many-to-many relation into simple triples.

3. D2RQ as mapping language

D2RQ platform: It is a system that facilitate users to access relational databases as virtual, read-only graphs which includes D2RQ mapping language, D2RQ engine and D2R server.[4]

D2RQ language: It is a declarative language that is used to map relational database schema to RDF vocabularies and OWL ontologies. The language is written in Turtle syntax and supports highly normalized table structures containing data instances spreading over multiple tables. Conditional mapping of of D2RQ allows specific information accessible using construct d2rq:condition.

D2R Server uses the D2RQ mapping language for capturing mappings between database schemas and RDFS schemas or OWL ontologies. It gives a specification for identifying resources and generating property values from database content. ClassMap is the main object of D2RQ. Mapping from a set of entities within the database, to class or a group of similar classes of resources is represented by ClassMap. Property Bridges of ClassMap specify how resource descriptions are created. D2RQ allows property values to be created directly from database values or by translation tables or patterns. Conditional mappings, mapping of many-to-many relations and highly normalized table structures are supported on ClassMap and PropertyBridgelevel.[7]

D2RQ Engine, a plug-in for Apache Jena Framework used for building semantic web application, can be used as a component that rewrite Jena API calls. The query results obtained on performing these API calls against the database are passed to the higher layers of the Jena Framework**.** The functional component D2RQ platform is illustrated in Fig1

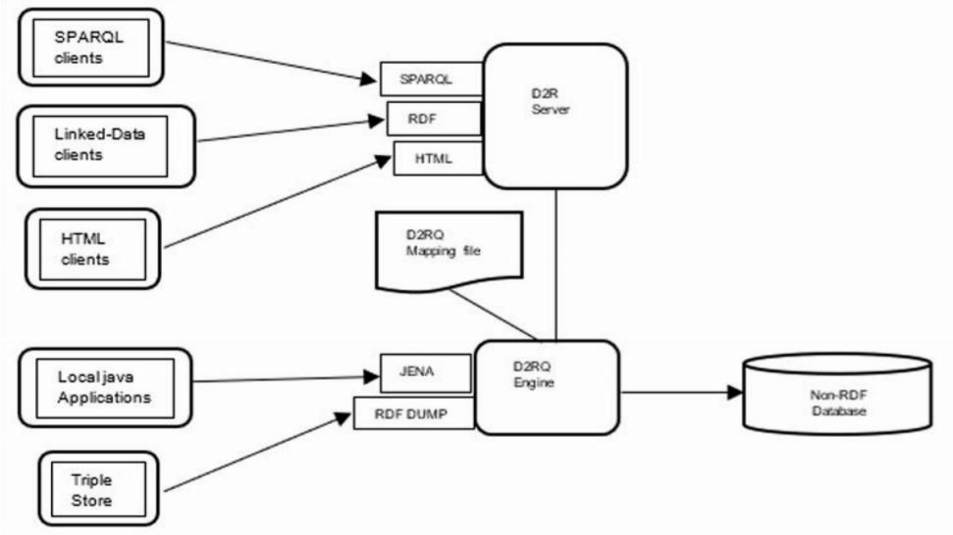

Fig 1. Functional Component of D2RQ platform

The customizable D2RQ Mapping is used by D2R Server for mapping database content into RDF format and enable the data to be searched and browsed. This mapping is used for rewriting the request on-the-fly from the web into SQL queries allowing publishing of RDF[8] from large database. Therefore, replication of data by creating a RDF triple store can be eliminated.

Bizer et al [6]provides an environment to access non-RDF relational databases as virtual RDF graph using D2RQ languages. Although D2RQ performs well but its performance deteriorates if SPARQL includes Optional, Filter and Limit feature. It does not have reasoning capability which can be added by using Jena. It is read- only and cannot handle structure like VIEWS of database.

4. R2RML as mapping language

Das Souripriya et al [5] and Neto, Luís Eufrasio T., et al [12] describe R2RML as a language that is used to generate customized mappings from relational databases to RDF datasets. The generated mappings enable users to have an RDF view of existing relational data which is expressed in a structure and target vocabulary of the choice of author. D2RQ system use R2RML mappings for providing relational data as RDF graph[1].As illustrated in Fig 2, R2RML Engine is a software which takes as input relational database(RDB) and R2RML mapping document and produces RDF graph as output as per the mapping document.

The main component of R2RML mapping is TriplesMaps. Each TriplesMap refer to a logical table from the database. A base table, a view or a valid SQL query of the input schema can be the logical table.

A TriplesMap consists of one SubjectMap and one or more PredicateObjectMap structure(s) that map logical table row with a set of RDF triples. The rules of generation of subject for a row is being contained in SubjectMap. Similarly rules for generation of predicate and object are defined in the PredicateMap and ObjectMap structure respectively.

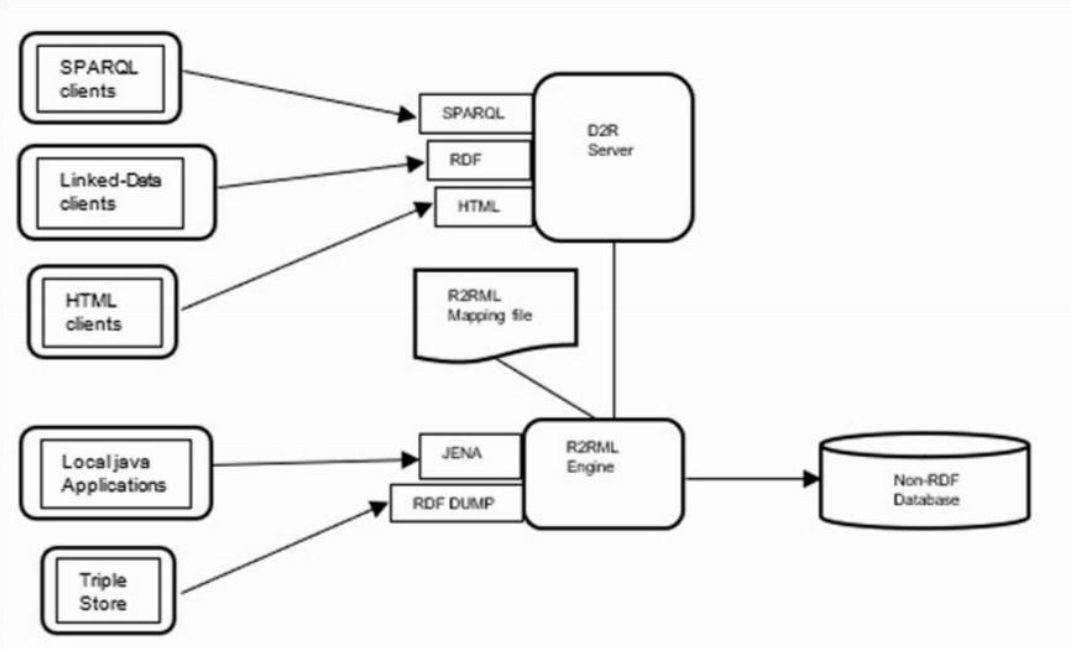

Fig 2. R2RML Architecture

The output of an R2RML mapping is an RDF dataset. The RDF triples in the dataset are the result of applying the mapping rules of each TriplesMap to the rows of its logical table.

By default, all RDF triples are in the default graph of the RDF dataset. A TriplesMap can contain optional rules that place some or all of the triples into named graphs instead.

5. Use Case Evaluation

In this section a sample database is being used as a case study for showing how the two mapping languages- D2RQ and R2RML differ along four major dimensions such as i) their mapping type, ii) how much expressive the languages are by considering mapping descriptions iii) how mapping is being implemented i.e how and when data is converted into RDF and iv) the way data is retrieved(query-based, linked-data, graph dump). As a use case we have used one sample relational database , having five table: person, employee, department, project, and workson as described in Fig3.

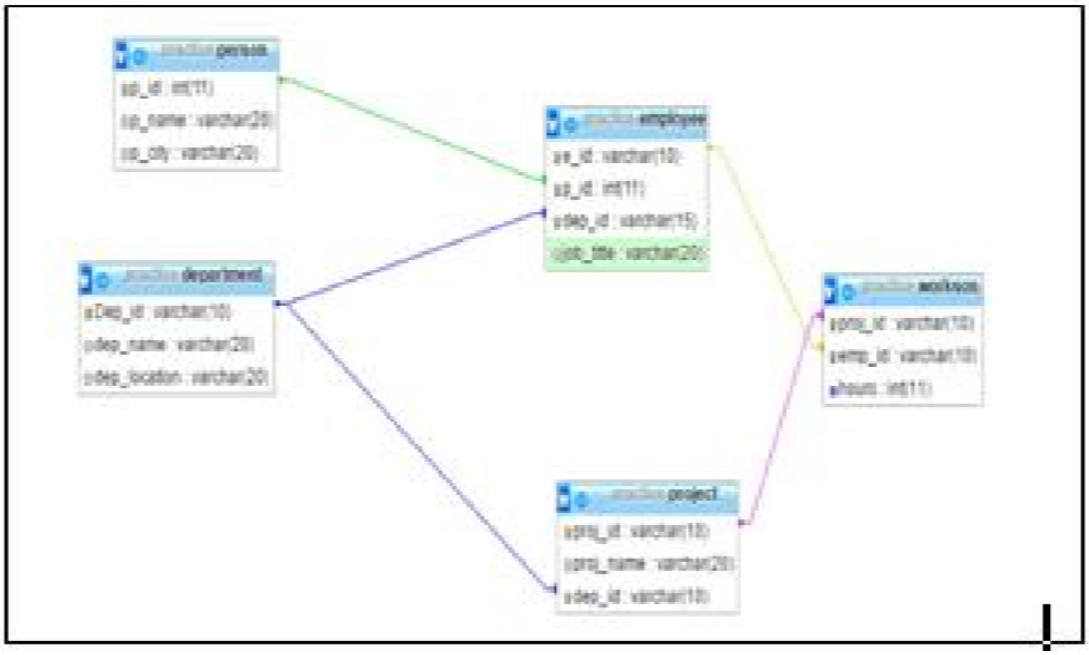

Fig 3. Sample relational database

6. Generation of mapping

The D2RQ and R2RML mapping languages are implemented on relational database[fig]and mapping file is being generated. The type of mapping file vary based on the mapping languages used. The mapping file being generated is observed for different constraints of relational database.

6.1 IS-A Relationship

Fig4 below shows the IS-A relationship between employee and person:

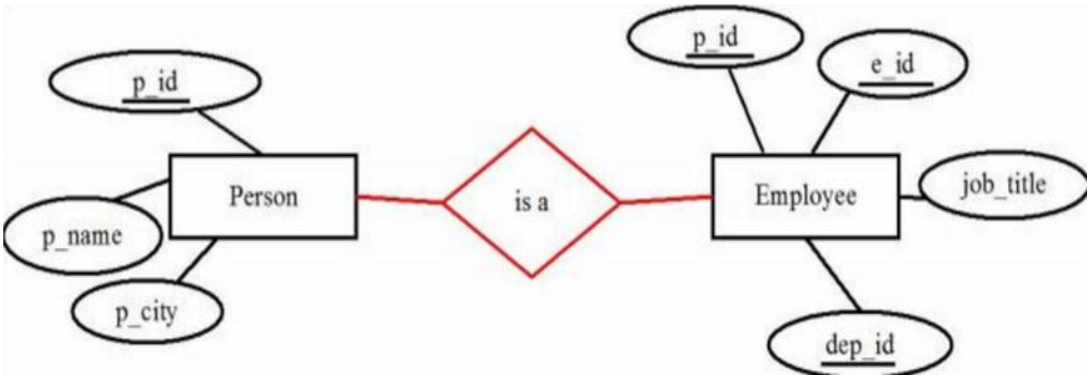

Fig 4. IS-A relationship

A section of the mapping file being generated are shown Fig. 5 where "d2rq:join is the d2rq:classmap in D2RQ and rr:joinCondition in R2RML is used for representing the IS-A relationship between employee to the person.[4]

D2RQ

```
map:employee_p_id__ref a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:employee;
    d2rq:property vocab:employee_p_id;
    d2rq:refersToClassMap map:person;
    d2rq:join "employee.p_id => person.p_id";
```

R2RML

```
rr:predicateObjectMap [
    rr:predicate vocab:employee_p_id;
    rr:objectMap [
        rr:parentTriplesMap map:person;
        rr:joinCondition [ rr:child "`p_id`"; rr:parent "`p_id`"; ];
    ];
];
```

Fig 5. Mapping of IS-A relationship

6.2: N Relationship

Fig6 below shows the 1:N relationship between employee and Department:

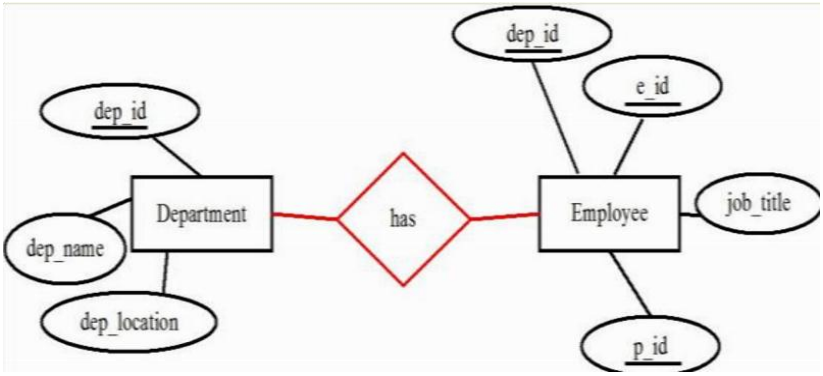

Fig. 6. One-to-Many Relationship

A section of the mapping file being generated are shown **Fig.7** where The relationship of one to many of the RDB which is connected by the column dep_id of the department table with the dep_id of employee table in the R2RML ttl file is shown.

D2RQ

```
map:employee_dep_id__ref a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:employee;
    d2rq:property vocab:employee_dep_id;
    d2rq:refersToClassMap map:department;
    d2rq:join "employee.dep_id => department.Dep_id";
```

R2RML

```
rr:predicateObjectMap [
    rr:predicate vocab:employee_dep_id;
    rr:objectMap [
        rr:parentTriplesMap map:department;
        rr:joinCondition [ rr:child "`dep_id`"; rr:parent "`Dep_id`"; ];
    ];
];
```

Fig 7. Mapping of 1:N relationship

6.3M:N (Many to Many) Relationship

Fig.8 below shows the IS-A relationship between employee and Department:

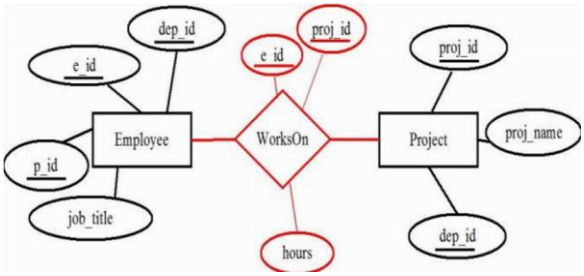

Fig.8 Many-to-many relationship

6.3.1 IN D2RQ

The relationship of many to many of the RDB is shown by the workson table. The mapping generated by D2RQ corresponding to Many-to-many relationship is illustrated in Fig 9:

```
# Table workson
map:workson a d2rq:ClassMap;
   d2rq:dataStorage map:database;
   d2rq:class vocab:workson;
   d2rq:uriPattern "workson/@@workson.emp_id|urlify@@/@@workson.proj_id|urlify@@";
   d2rq:classDefinitionLabel  "workson";
   .
map:workson__label a d2rq:PropertyBridge;
   d2rq:belongsToClassMap  map:workson;
   d2rq:property rdfs:label;
   d2rq:pattern "workson #@@workson.emp_id@@/@@workson.proj_id@@";
   .
map:workson_hours a d2rq:PropertyBridge;
   d2rq:belongsToClassMap  map:workson;
   d2rq:property vocab:workson_hours;
   d2rq:column "workson.hours";
   d2rq:datatype xsd:integer;
   d2rq:propertyDefinitionLabel  "workson hours";
   .
map:workson_proj_id__ref a d2rq:PropertyBridge;
   d2rq:belongsToClassMap  map:workson;
   d2rq:property vocab:workson_proj_id;
   d2rq:refersToClassMap  map:project;
   d2rq:join "workson.proj_id => project.proj_id";
   .
map:workson_emp_id__ref a d2rq:PropertyBridge;
   d2rq:belongsToClassMap  map:workson;
   d2rq:property vocab:workson_emp_id;
   d2rq:refersToClassMap  map:employee;
   d2rq:join "workson.emp_id => employee.e_id";
   .
```

joins the project and workson table

joins the workson and employee table

Fig.9 Mapping of m:n relation in D2RQ

6.3.2 IN R2RML

The relationship of many to many of the RDB is shown by the workson table. The mapping generated by D2RQ corresponding to Many-to-many relationship is illustrated in Fig 10:

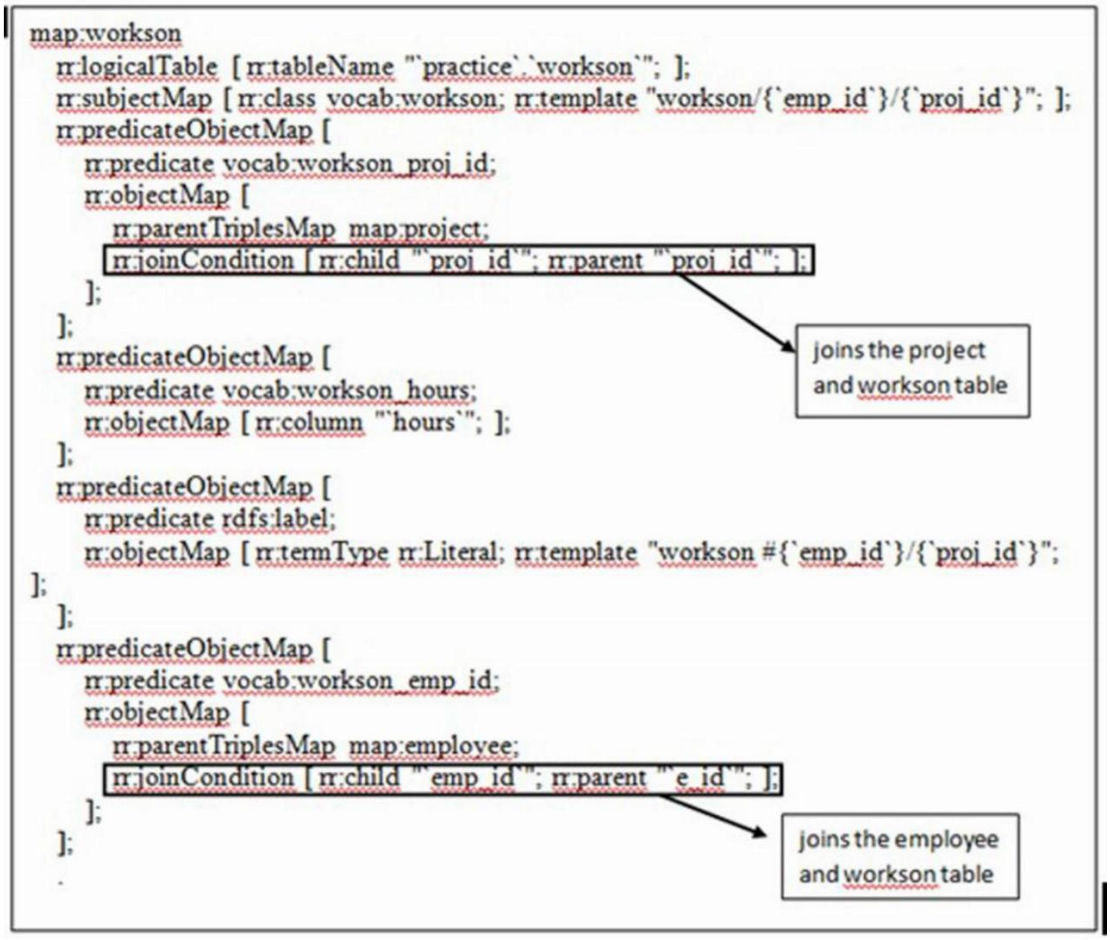

Fig. 10 Mapping of m:n relation in R2RML

7. Publishing relational database data on semantic web

The Semantic Web allows data to be stored along with well-defined meaning by using RDF data model. For transforming data into RDF, .is a global information space consisting of inter-linked data about resources. There are two access paradigms to the Semantic Web: browsing and searching. Using a Semantic Web browser like Tabulator (slides), a surfer can follow links from data about one resource to data about other resources. The second access paradigm is searching. Using the SPARQL query language and protocol, a client application can query data-sources for information about resources.D2R Server is a tool for publishing the content of relational databases on the Semantic Web.D2R Server allows a RDF representation of the database to be browsed and searched.

Both the mapping languages allow the relational database data to be made available to the user on the semantic web by publishing the data using the turtle file being generated and D2R server as demonstrated in Fig11

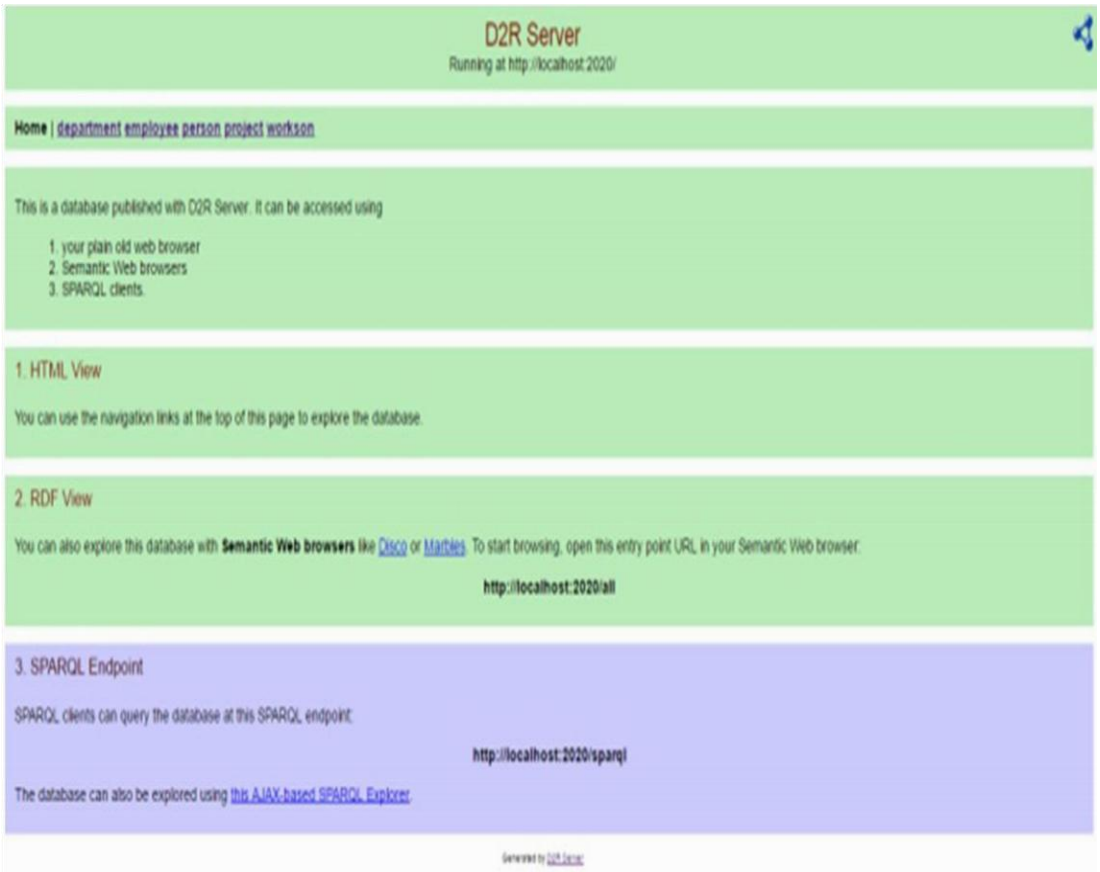

Fig.11 Publishing file generated by D2RQ on D2R-Server

8. Mapping implementation

The dump-rdf tool is used to transform the contents of database into a single RDF file. This transformation can be achieved with or without a mapping file. In case of mapping file, translation will be as per mapping specified in the file otherwise default mapping is used for translation. The sample of RDF file being created in case of D2RQ is shown in **fig.12**

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:map="http://localhost:2020/#"
    xmlns:vocab="http://localhost:2020/vocab/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"  xml:base="http://localhost:2020/" >
  <rdf:Description rdf:about="project/d1/proj2">
    <vocab:project_dep_id rdf:resource="department/d1"/>
    <vocab:project_proj_name>InnoM</vocab:project_proj_name>
    <rdfs:label>project #d1/proj2</rdfs:label>
    <vocab:project_proj_id>proj2</vocab:project_proj_id>
    <rdf:type rdf:resource="vocab/project"/>
  </rdf:Description>
  <rdf:Description rdf:about="vocab/workson_hours">
    <rdfs:label>workson hours</rdfs:label>
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="workson/e1/proj2">
    <vocab:workson_emp_id rdf:resource="employee/d2/e1/1111"/>
    <vocab:workson_proj_id rdf:resource="project/d1/proj2"/>
    <rdfs:label>workson #e1/proj2</rdfs:label>
    <vocab:workson_hours rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">4</vocab:workson_hours>
    <rdf:type rdf:resource="vocab/workson"/>
  </rdf:Description>
  <rdf:Description rdf:about="workson/e2/proj2">
    <vocab:workson_emp_id rdf:resource="employee/d2/e2/1112"/>
    <vocab:workson_proj_id rdf:resource="project/d1/proj2"/>
    <rdfs:label>workson #e2/proj2</rdfs:label>
    <vocab:workson_hours rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">55</vocab:workson_hours>
    <rdf:type rdf:resource="vocab/workson"/>
  </rdf:Description>
  <rdf:Description rdf:about="vocab/project">
    <rdfs:label>project</rdfs:label>
```

Fig.12 Sample of rdf file generated by R2RML

9. Data Retrieval :

The transformed relational data which is in RDF form is accessed by using the SPARQL query language. It is a data-oriented query language without having any inference in itself. A sample query that retrieve all triples of RDF model is

Querying using SPARQL

```
Select ?s ?p ?o where {
        ?s ?p ?o .
}
```

where

?s represents Subject

?p represents Predicate

?o represents Object

10. Result and Analysis

Query is performed on the RDF file being generated to retrieve the result and their running time is also calculated. Fig 13 and fig14 shows the running time of five different queries for both D2RQ and R2RML.

Query No.	SPARQL Query	D2RQ(time in ms)	R2RML(time in ms)
Q1	SELECT ?sub ?pre ?obj WHERE { <http://localhost:2020/person/1112> ?pre ?obj}	3017	2119
Q2	select ?s ?pre ?city where{?s ?pre ?city}	2374	2168
Q3	PREFIX vocab: <http://localhost:2020/vocab/> SELECT ?obj WHERE { ?sub vocab:person_p_name ?obj . FILTER regex(?obj, "jo", "i") }	2278	2150
Q4	SELECT ?p_id WHERE { ?y <http://localhost:2020/vocab/person_p_name> "john". ?y <http://localhost:2020/vocab/person_p_id> ?p_id. }	2189	2111
Q5	PREFIX vocab: <http://localhost:2020/vocab/> SELECT ?sub ?obj WHERE { ?sub vocab:person_p_id ?obj . FILTER (?obj >= 1111) }	2320	2154

Fig.13. Query wise time in D2RQ and R2RML

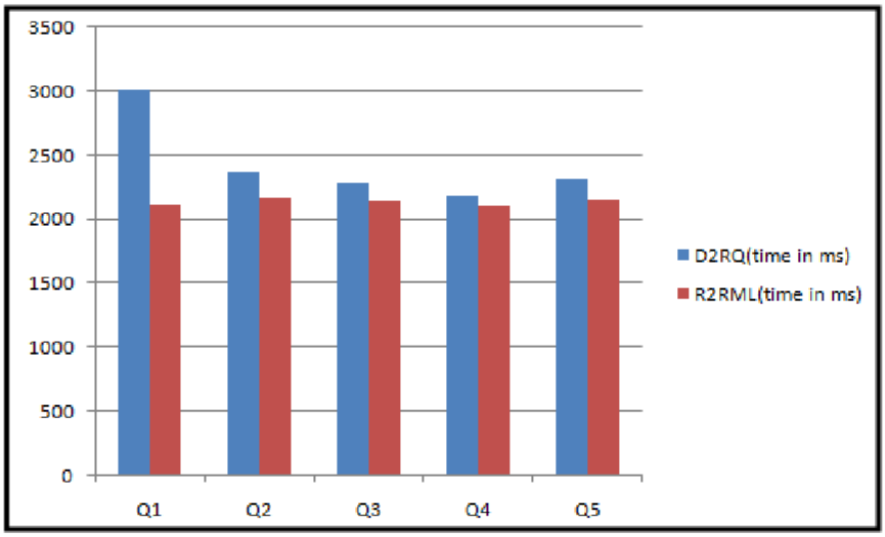

Fig 14. Time Analysis Graph

11. CONCLUSIONS

In this paper we have presented two different mapping languages for mapping the content being stored in relational databases to ontology: D2RQ and R2RML. We focused on four main aspects-mapping type, mapping description, mapping implementation and data retrieval. It is observed that there are lot of similarities between D2RQ and R2RML mapping languages. Both the mapping languages apply the same logic for mapping relational database to ontology. In both the languages rows are mapped to ontology individuals, columns that are foreign keys are mapped to object property and other columns are mapped to data properties of ontology. However, mapping file generated by R2RML mapping language offers the facility to write SQL query for creating views of table/tables. The table referring each view are further mapped to the ontology concepts.

REFERENCES

[1] Michel, Franck, Johan Montagnat, and Catherine Faron-Zucker. “A survey of RDB to RDF translation approaches and tools”. Diss. I3S, 2014.

[2] Berners-Lee, Linked Data, in Design Issues of the WWW, (2006).

[3] Sahoo, Satya S., et al. "A survey of current approaches for mapping of relational databases to RDF." *W3C RDB2RDF Incubator Group Report* (2009): 113-130

[4] “The D2RQ Platform – Accessing Relational Databases as Virtual RDF ...”,http://d2rq.org/[last accessed on 03/04/2018]

[5] Das Souripriya, Sundara Seema, and Richard Cyganiak. “R2RML: RDB to RDF Mapping Language - World Wide Web Consortium”,https://www.w3.org/TR/r2rml//[last accessed on 08/05/2017]

[6] Bizer, Christian, and Richard Cyganiak. "D2r server-publishing relational databases on the semantic web." *Poster at the 5th international semantic web conference*. Vol. 175. 2006.

[7] Bizer, C., Cyganiak, R.: D2R Server – Publishing Relational Databases on the Semantic Web. In: ISWC (2006)

[8] C. Bizer. D2R MAP -- A Database to RDF Mapping Language. In *Proceedings of the 12th International World Wide Web Conference*, 2003.

[9] Hert, Matthias, Gerald Reif, and Harald C. Gall. "A comparison of RDB-to-RDF mapping languages." *Proceedings of the 7th International Conference on Semantic Systems*. ACM, 2011.

[10] Neto, Luís Eufrasio T., et al. "R2RML by assertion: A semi-automatic tool for generating customised R2RML mappings." *Extended Semantic Web Conference*. Springer Berlin Heidelberg, 2013.

[11] Eisenberg, Vadim, and YaronKanza. "D2RQ/update: updating relational data via virtual RDF." *Proceedings of the 21st International Conference on World Wide Web*. ACM, 2012.

[12] J. Sequeda, S. H. Tirmizi, O. Corcho, and D. P. Miranker. Survey of directly mapping SQL databases to the semantic web.Knowledge Eng. Review, 26(4):445–486, 2011.