IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-6, June 2018 E-ISSN: 2347-2693

A Model for Mapping Semantic Web Data with Heterogeneous Data Sources Using SPARQL

R. Gupta[1*], S.K. Malik[2]

[1] USIC&T, GGSIP University, New Delhi, India
[2] USIC&T, GGSIP University, New Delhi, India

Corresponding Author: rupal.gupta07@gmail.com, Tel.: +91-8791716629

Available online at: www.ijcseonline.org

Accepted: 10/Jun/2018, Published: 30/Jun/2018

***Abstract*—** Semantic Web is extending web2.0 to web3.0 with an idea of incorporating intelligence or meaning to the existing web. Relational databases have been playing a crucial role in software development since many years. Also rapidly developing semantic web is based on mapping and compatibility of the existing data on the web which may be either in relational or non-relational form. RDF & Web Ontology are two major representations in semantic web, and SPARQL is a query language, used to query data from different semantic web resources like RDF/OWL/LOD (Linked Open Data). SPARQL processing and execution is playing a crucial role in mapping data from different sources. In this paper, first, a brief literature survey is being presented and discussed, focusing on SPARQL usage in mapping. Second, it focuses on various concerns of SPARQL query processing and execution in different domains along with SQL conversions and illustrations. Third, it presents analysis of various approaches and tools for the migration of data into semantic web from other data sources supported by a proposed model for information processing.

***Keywords*—**SPARQL, RDF, OWL, LOD, Hadoop, Relational Databases (RDB), D2RQ, Twinkle, Jena fuseki, Jena ARQ, DBpedia.

I. Introduction

Semantic web, the future smart web, focuses on the need of processing or publishing various kinds of data which may be structured, semi structured or unstructured. For data publishing in semantic web, the three key technologies viz. SPARQL [1], RDF [2] and ontology are playing a significant role. Data on the web is heterogeneous and in different formats which is growing rapidly as the size of web gets doubled after few months. The challenge is to bring all this data on the same platform like triple format stored as RDF data and then to process it. For this purpose, it is required to map and process relational / non-relational data through semantic web technologies. For handling large datasets, RDF graphs are most preferable due to their flexibility and ability to store data in machine understandable form.

SPARQL is a query language used in semantic web for fetching the desired data from RDF graph. SPARQL query processing and performance for ever increasing size of RDF has become a major challenge. For storing and managing large RDF graphs effectively and efficiently, Hadoop[1] framework is widely used to handle big data so it can also handle millions and trillions of triples stored in RDF format. It assists large RDF graph data with features like reliability and high fault tolerance with replication (high availability). Large RDF graphs can be stored in Hadooop distributed file system (HDFS) and can be processed through Hadoop processing model using MapReduce [3] or Apache Spark[2] [4]. This processing framework also supports distributed SPARQL processing using cluster based approach.

In this paper, the usage of SPARQL Query in transformation is being analyzed along with its semantics and execution on various tools, which is demonstrated using D2RQ[3] tool and mysql database with the help of illustrations which shows how the data stored in mysql database may be mapped and viewed using SPARQL query. The objective is to present an analysis of various tools and technologies which can be used for mapping or transformation of data from one domain to other through classification of approaches [5] and is supported by a model of information processing. The emphasizes is on an alternate approach for common semantic query language (SPARQL) which can be directly used to query relational and non-relational databases without a need to convert them into OWL/RDF [6]. The major discussions in this paper may be summarized as follows:

Section 1 is the introduction which discusses general perspectives and objectives of the work done.

Section 2 presents the background which has literature survey on SPARQL processing and mapping data in

semantic web along with concerned discussions and key definitions. It includes: various key components of semantic web – RDF, OWL, SPARQL along with their semantics and tools for development. Also various concerns are being discussed for large RDF data and its processing using Hadoop and related technologies.
Section 3 covers the main focus of the paper which includes: a brief analysis of various tools and approaches of mapping RDB into RDF/OWL with the concerns of SPARQL processing on non-Relational data stored in Hadoop or LOD[4]. The analysis is demonstrated using D2RQ tool and is supported by a proposed model.

II. Background

A. Three most significant Components of Semantic Web

Semantic web comprises of various components or technologies in which RDF [2], OWL [7] and SPARQL [8] are the one of the most significant ones, where RDF represents basic data format in terms of triples and graph, OWL represent rich data taxonomy in terms of classes, object properties and relationships, SPARQL represents the query model to fetch the semantic web data. These are detailed as below:

1) RDF

RDF is a flexible and universal graph-like data model used in Semantic web data representation. It is recommended by W3C and is treated as a standard for representing information about arbitrary resources through IRIs (Information Resource Identifiers) [9]. The RDF data model is represented using a triple pattern, which consists of a subject, a predicate and an object known as S, P and O respectively [10]. Assuming disjoint, countable infinite sets I (RDF IRIs), B (Blank Node) and L (Literals), The RDF triple t can be seen as $t=\{S,P,O\} \in (I \cup B) \times I \times (I \cup B \cup L)$ [11]. It may also be treated as a resource identifier with attribute or property values, which can be easily represented using RDF Graphs [5] [12].

a) RDF Formats

A RDF data can be represented in various formats like N-Triple format, N3, RDF/XML, RDFa and Turtle etc. Among these mentioned, RDF/XML is the W3C recommended standard format of storing data in RDF [13]. Assume that data represented here is: The article with ISBN 008796671X has the author "Rupal Gupta" and has the title "SPARQL: A literature Survey". Rupal Gupta's designation is "Assistant Professor, TMU". Some of the data representations like N-Triples, RDF/XML, N3 and Turtle format representations as code snippets are as follows:

- N-Triples

(File Format: .nt, for example- rdfdata.nt) [13]
Code Snippet

```
<urn:isbn:008796671X>
<http://tmu.org/dc/elements/1.1/author>
<http://www.w3.org/ccsit/Rupal/card#i> .
<urn:isbn:008796671X>
<http://tmu.org/dc/elements/1.1/title> " SPARQL: A
literature Survey " .
<http://www.w3.org/ccsit/Rupal/card#i>
<http://www.w3.org/2006/vcard/title>
" Assistant Professor, TMU " .
```

- RDF/XML

(File Format: .rdf, for example-rdfdata.rdf) [13]
Code Snippet

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
xmlns:dc="http://tmu.org/dc/elements/1.1/"
xmlns:v="http://www.w3.org/2006/vcard/">
<rdf:Description rdf:about="urn:isbn:008796671X">
<dc:title> SPARQL: A literature Survey </dc:title>
<dc:author
rdf:resource="http://www.w3.org/ccsit/Rupal/card#i"/>
</rdf:Description>
<rdf:Description
rdf:about="http://www.w3.org/ccsit/Rupal/card#i">
<v:title> Assistant Professor, TMU </v:title>
</rdf:Description>
</rdf:RDF>
```

- N3 (Notation 3 Format)

(File Format: .n3, for example- rdfdata.n3) [13]
Code Snippet

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix v: <http://www.w3.org/2006/vcard/> .
<http://www.w3.org/People/Rupal/card#i>
v:title " Assistant Professor, TMU " .
<urn:isbn:008796671X>
dc:author <http://www.w3.org/People/Rupal/card#i> ;
dc:title " SPARQL: A literature Survey " .
```

- Turtle Format

(File Format: .ttl, for example – rdfdata.ttl) [13]
Code Snippet

```
@prefix a: <http://rupalipu.com/ns/addressbook#> .
@prefix d: <http://rupalipu.com/ns/data#> .
d:i0078 a:firstName "Rupal" .
d:i0078 a:lastName "Gupta" .
d:i0078 a:Tel "8791716629" .
d:i0078 a:email "rupal.gupta07@gmail.com" .
d:i0078 a:email "rupal.computers@tmu.ac.in" .
d:i0077 a:firstName "Ruchika" .
d:i0077 a:lastName "Gupta" .
d:i0077 a:Tel "8791176761" .
```

d:i0077 a:email "ruchika21@gmail.com" .
d:i8661 a:firstName "Apoorv" .
d:i8661 a:lastName "Gupta" .
d:i8661 a:email "apoorv21jamia@hotmail.com" .
The above RDF snippets assist to better understand and analyze the data representations.

b) RDF Graphs

A set of RDF triples in a combination is treated as a RDF graph, which is a graphical representation of RDF triples. Thus RDF graph G = {t1,t2,….,tn} , where t1, t2… tn are the triples represented using as {s, p, o} [9]. For example-Ruchika follows Rupal , Rupal follows Saurabh, Rupal likes Pizza, Ruchika likes Pizza, Saurabh likes Burger, Rupal likes Burger all are triples and can be easily visualized using the RDF graph in figure 1 as below:

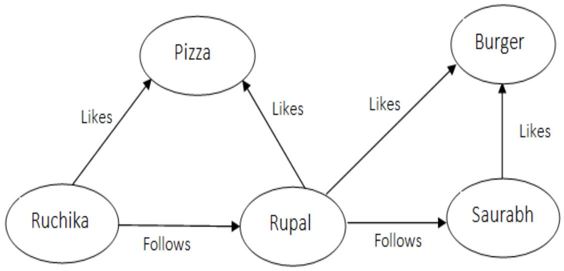


Figure 1. A sample RDF Graph

Since the existing data is generated with a rapid speed, so it is advisable to store RDF data in a distributed system and this distributed system environment may be broadly categorized into three categories [9]: Standalone distributed RDF store, Classical centralized RDF store deployed on cluster nodes and Fully distributed platform for big RDF data processing like Hadoop. For handling large datasets, Bigdata has been widely used and supported by many tools and frameworks like Hadoop [14]. Hadoop is an open source framework which supports to handle and process large amount of data.

2) OWL

The Ontology Web Language (OWL) is a basic representation of semantic data which is an extension of RDF/RDFS and is also a W3C recommendation [7]. It has a stronger syntax which provides more vocabulary along with formal semantics [15]. It may be considered as a basic and rich representation of semantic web data in terms of classes and its properties along with their associations. It may be examined as metadata which explicitly represent the semantics of data in machine executable form instead of parsing the data only for the display purpose [15]. The current version of OWL is OWL2.0 developed by the W3C OWL Working Group in 2012 [16]. Ontology may be considered as the backbone for embedding semantics. It provides a common and shared domain theory which is a key asset for web semantics. OWL may be used to maintain specific knowledge of a domain and to represent complex and rich knowledge about its concepts, group of concepts, relations among concepts and the individuals. There are various tools which support ontology development and Protégé[5] is most widely used.

3) SPARQL

SPARQL is a protocol and a query language which is able to fetch and manipulate the data stored in RDF data format. Every SPARQL query is based on triple pattern which may be viewed as graph model named as BGP (Basic Graph Pattern) [17]. The latest version of SPARQL is SPARQL1.1 [8] which is released by W3C in year 2013. Initial version (SPARQL1.0) was only to fetch the data but the current version is having various new features like aggregate functions, Insert/ Update/Delete i.e. manipulating RDF data. There are many tools which support SPARQL and RDF data development like Jena ARQ[6], Apache Jena Fuseki Server[7], Twinkle[8] and others like Blazegraph[9], Sesame[10] etc which enables SPARQL queries to be executed.

a) SPARQL and SQL

The SPARQL query syntax follows the select-from-where clause approach, just like in SQL queries. It is quite easy to understand a SPARQL query by comparing it with SQL query syntax. There are many algorithms for conversion of SPARQL to SQL [18] where it has been noticed that conversion approach helps in various mappings from different domains. Automatic mapping is done from SPARQL to SQL in many mapping tools like D2RQ but efficiency is a major issue along with completeness concern. A brief comparison of both queries with the illustrations has been presented in table 1 as below:

Table1. Illustrations of SQL and its mapping with SPARQL with different clause.

SNO.	**SQL**	**SPARQL**

1.	SELECT faculty, course
	FROM LabReview
		SELECT ?faculty ?course
		WHERE {
		?x rdf:type foaf:Person .
		?x foaf:faculty ?faculty .
		?x foaf:course ?course }

2.	SELECT fname
	FROM LabReview
	WHERE fname = 'RUPAL'
		SELECT ?fname
		WHERE {
		?x rdf:type foaf:Person .
		?x foaf:fname ?fname .
		?x foaf:fname "RUPAL" .}

3.	SELECT fname, mail_id
	FROM LabReview
	WHERE fname like '%S%'
	And mail_id like '%@hotmail%'
	Order by fname
		SELECT ?fname ?mail_id
		WHERE {
		?x rdf:type foaf:Person .
		?x foaf:mail_id ?mail_id .
		?x foaf:fname ?fname .
		FILTER regex(?fname,"S")
		FILTER regex(?mail_id, "@hotmail") }
		Order by ?fname

There are various other features of SPARQL 1.1 like INSERT, DELETE etc have been illustrated as below [13].

Adding Telephone number to person “Apoorv” with reference to RDF data shown in turtle format which has been shown above in rdfdata.ttl code Snippet.

SPARQL Syntax for Inserting data in RDF (SPARQL Code Snippet):-

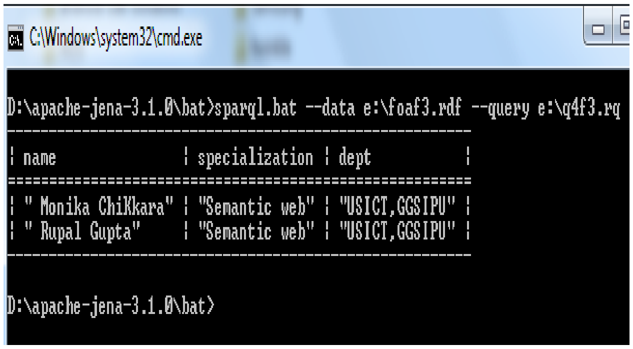
```
INSERT DATA
{
d:i8661 b:Tel "9837088135" .
b:Person a rdfs:Class .
}
WHERE {}
```

Similarly DELETE clause may also be performed on data and UPDATE may be performed with DELETE-INSERT method (If data needs to be changed, then it must delete previous data and insert a new modified one). Some new features have been added to SPARQL 1.1 like Group-Concat(), SPARQL HTTP Protocol Specification etc.

b) *Tools Used for SPARQL Execution*

- Jena ARQ

Jena ARQ[6] is used for free text search and is a SPARQL Query Engine that supports to retrieve RDF data for knowledge representation. A snapshot of SPARQL query execution using ARQ processor is been presented in figure 2.



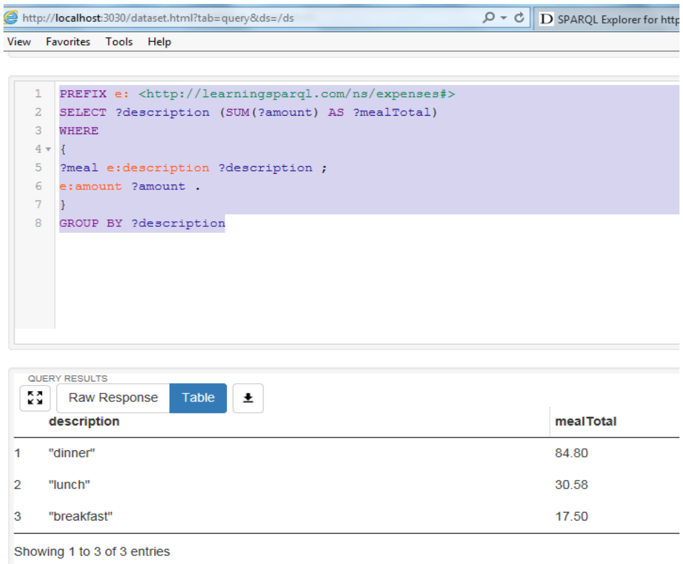
```
D:\apache-jena-3.1.0\bat>sparql.bat --data e:\foaf3.rdf --query e:\q4f3.rq
---------------------------------------------------
| name             | specialization | dept          |
===================================================
| " Monika Chikkara" | "Semantic web" | "USICT,GGSIPU" |
| " Rupal Gupta"     | "Semantic web" | "USICT,GGSIPU" |
---------------------------------------------------

D:\apache-jena-3.1.0\bat>
```

Figure 2. Snapshot of SPARQL Execution using JENA ARQ using SPARQL.bat

Jena ARQ provides support for SPARQL 1.1 features like Update, property function for custom processing, grouping and aggregate functions and remote accessing to SPARQL endpoint. Jena ARQ also supports for advanced SPARQL use like sub-SELECT, Negation, Construct Quad (added in Jena 3.0.1) and Property Paths [11].

- Jena Fuseki Server

Apache Jena Fuseki Server is a standalone SPARQL Server which provides support for querying through SPARQL1.1 and also provides support for server monitoring and administration. The current version of Apache Jena Fuseki server is Fuseki2 and is shown in below figure 3.



```
PREFIX e: <http://learningsparql.com/ns/expenses#>
SELECT ?description (SUM(?amount) AS ?mealTotal)
WHERE
{
?meal e:description ?description ;
e:amount ?amount .
}
GROUP BY ?description
```

	description	mealTotal
1	"dinner"	84.80
2	"lunch"	30.58
3	"breakfast"	17.50

Showing 1 to 3 of 3 entries

Figure 3. Snapshot of SPARQL Execution using JENA Fuseki Server

- SPARQL Query on DBpedia

DBpedia[11] is an open source online tool which is treated as a database of structured contents created by Wikipedia. It allows querying from Wikipedia data resources through SPARQL and is a good GUI tool for fetching desired information in different storage like XML, Spreadsheet, JSON, TSV, CSV and others. DBpedia can also be used as resource for taking dataset having millions of triples and so is treated as a benchmark for querying and performance analysis as well. It may also utilize for SPARQL intermediate conversion into SQL, Query execution plan generation and viewing optimized plan.

DBpedia also support additional feature in iSPARQL tab with interesting features of visualizing SPARQL Query using BGP Model. This feature of SPARQL can be utilized in performance analysis with different shapes of SPARQL Query especially in Distributed Environment. A LUBM[12] benchmark Query2 on university domain (presented in APPENDIX) is being viewed with DBpedia and helpful for analyzing the triples patterns used in SPARQL queries. The execution model is shown below in figure 4.

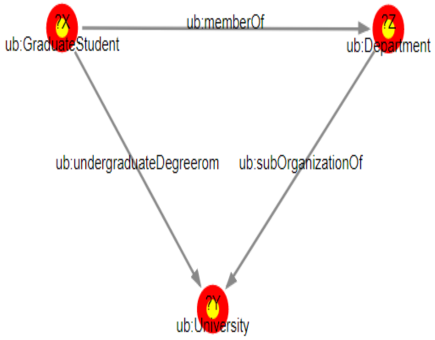
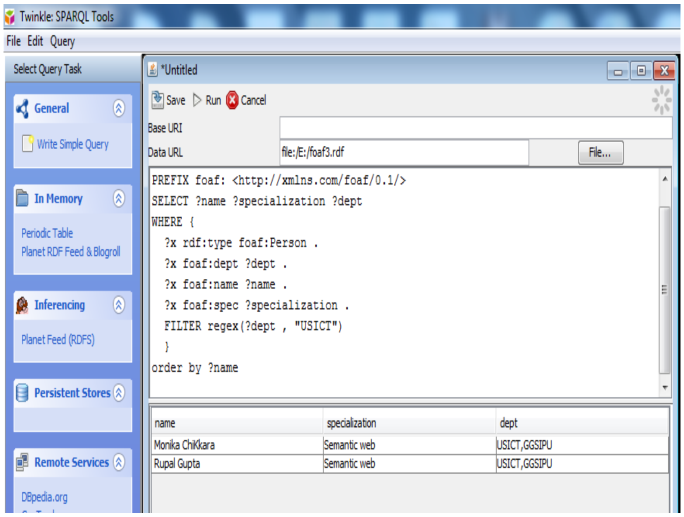


Figure 4. Visualization of LUBM SPARQL Query2 with DBpedia Tool using isparql

- Twinkle

Twinkle[8] is a GUI Tool which is distributed under GNU Public License which wraps a ARQ query Engine for SPARQL Query execution. Twinkle Tool is used for editing storing and manipulating SPARQL queries and its results. SPARQL execution on Twinkle tool is been shown in figure 5.



Base URI

Data URL file:/E:/foaf3.rdf

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?specialization ?dept
WHERE {
  ?x rdf:type foaf:Person .
  ?x foaf:dept ?dept .
  ?x foaf:name ?name .
  ?x foaf:spec ?specialization .
  FILTER regex(?dept , "USICT")
  }
order by ?name
```

name	specialization	dept
Monika ChiKkara	Semantic web	USICT,GGSIPU
Rupal Gupta	Semantic web	USICT,GGSIPU

Figure 5. Snapshot of SPARQL Query execution using TWINKLE tool

There are still many other tools which support SPARQL as per the need and popularity of RDF/OWL with other platforms.

III. Literature Survey and Related Work

The literature survey is categorized into two broad categories. One focuses on how SPARQL query can be utilized for mapping existing data in different format into semantic web data and other focuses on different tools that are used to perform the mapping with different platforms.

Cuzzocrea et al. [3] presents a critical survey on how Big RDF graphs management may be done by Map Reduced based algorithms and elaborate it with the relevance of RDF query Processing. The paper focuses on research direction and scope of indexing, fragmentation, integration, privacy preservation and analytics of Big RDF Data. Naacke et al. [4] concluded that as per the growing size of open linked data graphs at a rapid pace, processing the data efficiently is very much required and for this purpose a brief study of SPARQL Query processing strategies over in-memory cluster computer engine using Apache Spark is being presented. Five strategies are compared over different types of joins and partitioning, (SPARQL RDD, SPARQL DL, SPARQL SQL, SPARQL Hybrid RDD, and SPARQL Hybrid DF). Further the testing is being done of large triples using LUBM, DBpedia and WAVdir dataset. Schatzle et al. [9] represent the concepts of extended vertical partitions of RDF schemas that use semi-join based processing. The proposed prototype of S2RDF is a Hadoop based SPARQL query processor for large scale RDF data in distributed environment, implemented on the top of Apache Spark and the analysis is being done using S2RDF, H2RDF, Sempala, PigSPARQL and Virtuoso tool. Nikolaos et al. [19] presents H2RDF+ Tool, which is capable of storing and fetching large RDF data in a fully distributed RDF environment. A scalable adaptive decision about centralized and distributed join execution of SPARQL has been represented and effective results are being found on Hadoop environment using HBase indexes.

Franck et al. [5] represent two broad categories, R2RML and Non- R2RML approaches for transformation from relational database to RDF. Total seventeen tools in both categories is being analyzed on four major axis. Panawong et al. [20] presented an automatic method for generating database from ontology. An application development platform has been analyzed using OAM (Ontology Application Management) tool, which is used for creating, adopting ontology for semantic web applications.

Bellini et al. [21] presents the challenge of navigation of accessible RDF stores on internet. For the same Linked open graph (LOG), a web based tool may be used for processing, analyzing and navigating on multiple SPARQL endpoints. LOG, LODLive and Gruff tool have been analyzed on different parameter and LOG is found as the best among them. Oh et al. [22] presented an efficient query processing system which uses a job optimized and map only query planner for better performance. It is concluded that by utilizing a careful design HBase storage schema, a RDF data can be input to Map phase so that rearranging was not needed to evaluate the query. It has been concluded that by the proper use of abstract RDF data may be done to find out which pattern the result lied in, the amount of inputs may be reduced for Map-side jobs resulting in better performance [22].

Hartig and Pirro [11] studies the problem of extending the scope of property paths features in SPARQL to query distributed linked data and proposed a different interpretations of property paths over the web via the context based query semantics. Paper defines how the SPARQL language can be used for accessing Linked data on WWW. It focuses on reachability-based query semantics for property paths and differentiates them from navigation on web. Garcia and Wang [23] analyze the techniques and tools to overcome working for the large RDF datasets on Amazon Cloud. Also represents how Big Data Technology concept of Elastic MapReduce can be applied to process large Dataset with parallelization and distributed file system. The demonstration and evaluation is performed with three open source parsers, Apache Any23, Apache Jena ARQ and Semantic Web's Nx Parser with different file size and CPU count.

Mogotlane and Domneu [24] represents how ontologies were automatically constructed from oracle database using Protégé tool with DataMaster and OntoBase Plug-ins. The result is being visualized via OntoGraf and OWLViz and a comparative study of both plug-in is done. Anyanwu [25] shares the views that Map-Reduce approach can be used for SPARQL Multi Query optimization to increase the performance.

IV. Semantic Web Data Mapping Approaches

B. CLASSIFICATION OF MAPPING RELATIONAL DATABASE INTO SEMANTIC WEB

The classification of approaches can be broadly divided into two categories R2RML and Non-R2RML mapping [5]. Any approach starts with three things why, what and how to perform and define the task. For mapping classification approaches, it concerns as Motivation (Why it requires), Mapping Description (What is being defined), Mapping Implementation and Access Plan (How the task will perform and how can the data be accessed). The general description of all key steps is shown in the Table 2 given below. Here, the prime focus is on SPARQL query usage for mapping and retrieval along with mapping implementation and access plan where SPARQL usage has been noticed.

1) Mapping Implementation

- Data Materialization (ETL): Just like in Data warehousing ETL (Extract Transform Load) process, Data materialization is a transformation approach through which source database transformed into RDF data or OWL, statically. Various mapping rules exist through which the whole content of the database can be converted into RDF graphs and then the data is loaded into triple store [26]. It hardly supports large data sets and is a major drawback, also it does not work for frequent changing database and the generated RDF dump through ETL [12] needs to be regenerated for the consistency of data [5].
- On-Demand Mapping [27]: In this approach the whole relational database instance is not required to be transformed into RDF/OWL, but SPARQL queries are executed directly on database instance and at runtime conversion for SPARQL to SQL performed [18]. This means that Semantic query is to be converted into relational query to perform the task. This approach removes the major drawbacks of data materialization but has its own issues in transformation process of query into other with better performance. On-Demand Mapping is a dynamic approach and ETL is static in nature. It is efficient and effective as database mapping is performed on every execution of SPARQL query with transformation of query into SQL [6].

2) Access Plan/ Data Retrieval

- Query-Based Access (SPARQL): In this approach, the access plan or data retrieval is performed either using transformation or via conversion of a query into other domain. As SPARQL is a query language recommended by W3C to express queries on RDF data [12] [5], so SPARQL is also performing this task with both methods. Different mapping tools are using SPARQL query for retrieving data in transformation (ETL) and mapping (On-Demand Mapping).
- Linked Data approach: This approach uses http dereferences and the information is treated as resources using identifiers and mapping result is published as linked data. Linked open data also uses RDF graphs for visualization of data and resources and SPARQL can also be used for information retrieval [28].

Table 2. Classification of approaches (Key focus area / Steps) for transformation [5] [34]

S.No	**Key Focus (Steps)**	**Classification of Approaches**	**Description**
1	Motivation (Initial motivation to perform the task)	Ontology Learning	• Extracts concepts of ontology and relation from schema. • Prototype based mapping method.
		Generic-Purpose Mapping Language	• Uses complex mapping methods such as regular expressions or NLP. • Handles simple as well as complex mapping.
		Transformation Engine	• Implements query processing engine to process SPARQL Query or transforms whole RDB to RDF • Manage huge number of concurrent requests. • Can handle complex queries.

2	Mapping description (Way of mapping)	Direct Mapping	• Local ontology mapping. • Automatic creation of URIs. • Better to use when no domain Ontology exists.
		Domain Semantics-Driven Mapping	• Transformative mapping approach. • Reduces gap between RDB and RDF. • Ontology and databases are designed separately.
3	Mapping implementation (Way of translation into RDF/Ontology instances)	Data Materialization	• Uses ETL Process which is a static transformation approach. • Implementation of complex queries may be done.
		On-Demand Mapping	• Reverse approach of data materialization. • Run-time mapping is performed using SPARQL queries. • Best suited for distributed data environment. • Data consistency is maintained.
4	Access plan / Data Retrieval (Retrieval of data from transformed data through Mapping)	Query Based Access	• Irrespective of any transformation approach, SPARQL query is used to fetch the transformed data. • Query implementation is performed either directly or through conversion into SQL.
		Linked Data	• Uses RDF graphs to represents linked data. • Uses HTTP Get method for dereference URI to a logical entity.

C. ANALYSIS OF TOOLS USED FOR MAPPING RDB AND RDF/OWL

There are different tools available for mapping through classification of approaches. Some of tools discussed are open source work in together with other domains as per the requirement. There are some built-in tools used as a plug-ins to perform the transformation. The detailed description of them is shown below in Table 3.

Table 3. Description of tools used for mapping RDB into RDF/OWL.

S.NO	Tool Used For Mapping	Description
1	**OnTop**[13]	• Open Source ODBA system that conceptually provide ontologies that defines vocabulary, models and domain by hiding the source data [29]. • Fast tool packed which explore relational database as virtual graphs by linking them into ontology using R2RML mapping. • It includes Quest (a SPARQL Engine) for querying and support to protégé and Sesame.
2	**D2RQ**[3]	• Open source RDB to RDF Query based transformation engine used for Mapping with relational database as virtual RDF graphs. • Mapping may be performed using SPARQL endpoint, Linked Data (using http dereferencing), generating RDF dumps, and Jena API based access.
3	RDB2ONT	• It uses metadata and structured constraint defined in databases [5] and preserves all constraints while generating ontology [30]. • Contains two components OWL builder (Uses Java JDBC and ODBC API to extract metadata, structural constraints) and OWL writer (Uses file system to write generated OWL and gives users the flexibility of choosing their own namespace URIs and location to store)
4	R2O / ODE Master	• It is an incorporated framework which consists of R2O and ODE Master. • R2O is based on XML syntax, allows the description of complex mapping between existing ontology and relational tables whereas ODE Master is a processor that generates the semantic web instance from relational database based upon mapping description articulated in R2O document [31].

5	**Triplify**	• It is a generic purpose mapping language and a RDF2RDF transformation engine, which uses both query based and transformation based approach. • It focuses mainly on small to medium web applications which are generally less than 100MB database content [5]. It supports update logs for RDF resources and is useful for crawling engines.
6	**Relational. OWL**	• It is an abstract way of OWL based representation format for relational data and schema components. • It uses JDBC and Jena framework, to automatically extract the semantics from relational database virtually and transformed further into RDF/OWL [32].
7	**Virtuoso RDF Views**	• This tool provided by virtuoso openlink software, that derives a Semantic Web of Linked Data from existing data. It is targeted to meet enterprise needs regarding data management, access and integration [5]. • Using Virtuoso universal server and high performance virtual database engine it was possible to build a SPARQL to SQL relational layer mapping into server. • Virtuoso RDF Views also provide supports to RDF Quad Store used for linked data.
8	Data Master	• It is a protégé plug-in used to import relational data from the databases which supports JDBC drivers. It imports database Schema as OWL classes or schema instances of Relational.OWL classes. • This plug-in is supported by protege3.X version and comes as a built-in plug-ins [24].
9	OntoBase	• OntoBase is a Protégé plug-in, available for the automatic conversion of relational databases into ontologies [24]. • The main advantage is that it reduces the cost of reverse-engineering for mapping ontologies and relational database.
10	**Ultrawrap**	• It is a direct RDB2RDF mapping framework which based on SQL views to present relational data as RDF triples [33]. • Virtual RDF may be presented through SPARQL-to-SQL transformation at runtime. It utilizes SQL features in transformation. [5].

The tools mentioned above for mapping has been described which is helpful for readers to analyze and compare them. In above Table3 Serial number {1,2,5,6,7,10} are highlighted which is focusing that these tools/approaches uses query based implementation, that support SPARQL Query processing for information retrieval either directly from RDF/OWL or via transformation of query in to other compatible query structure. An analysis of few mentioned above is represented by Gupta and Malik, [34] focusing on the motivation, mapping description, purpose and level of automation with SPARQL utility and semantic web language used for mapping.

D. Mapping of relational database using D2RQ Tool

For the illustrations of mapping the database with the name "labreview" is being created in mysql with seven relations shown in figure 6.

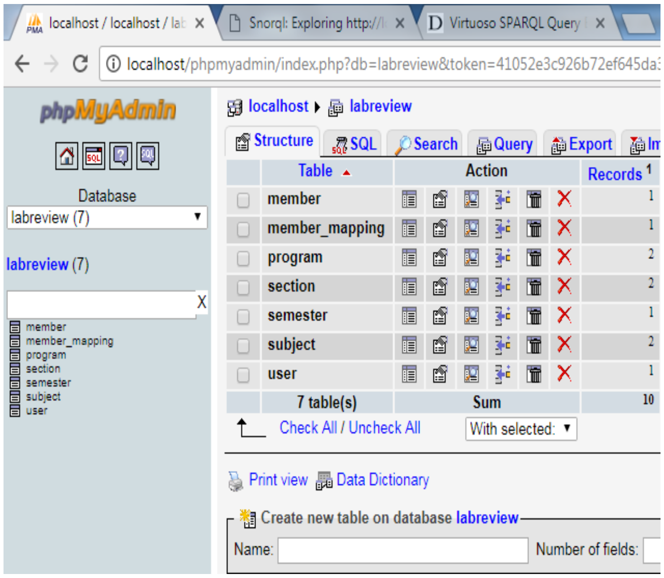


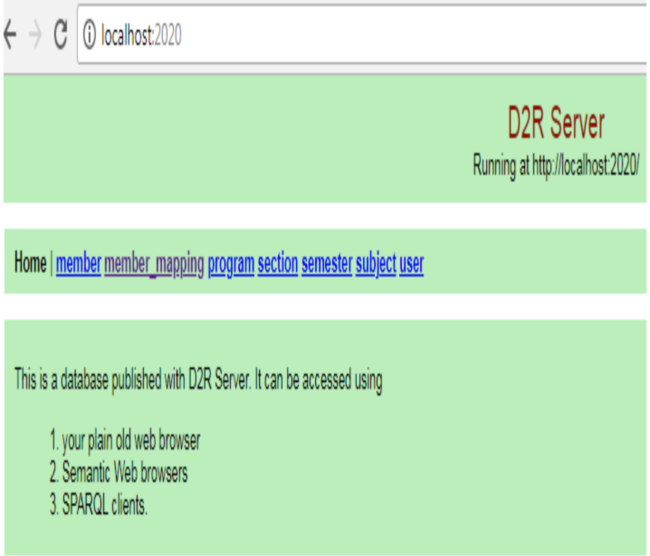
	Table ▲	Action	Records 1
□	member		1
□	member_mapping		1
□	program		2
□	section		2
□	semester		1
□	subject		2
□	user		1
	7 table(s)	Sum	10

Figure 6. Snapshot of database "labreview" created in mysql

Following are the key steps of how to work with D2R Server on the machine using above mysql database tables:

Step 1: Download and install extract d2rq-0.8.1.zip or tar file as per your system specification from d2rq.com. (Ensure JRE version1.5 or higher must exist first)
Step 2: Download a JDBC driver from the database vendor from which mapping need to be performed. The .jar file of driver must be uploaded there in lib directory of D2R server.
Step 3: Get into d2rq folder through command prompt and generate a mapping file with the command syntax.
Command: generate-mapping –u root jdbc:mysql:///labreview

`**Step 4:** Start D2R server with the syntax.
Command: d2r-server –u root jdbc:mysql:///labreview
Step 5: Access and test the D2R server through the browser using 2020 port on localhost shown in figure 7.


localhost:2020
D2R Server
Running at http://localhost:2020/
Home | member member_mapping program section semester subject user

This is a database published with D2R Server. It can be accessed using

1. your plain old web browser
2. Semantic Web browsers
3. SPARQL clients.

Figure 7. Snapshot of D2RQ Server homepage showing the relations in upper tabs after mapping from "labreview" database shown in figure 6.

Step 6: Run a SPARQL query through the browser or can also be run through command prompt using d2r-query command. ('Note- that mapping1.ttl is a filename in the command that has to be generated using generate-mapping with –o option and filename with step2.').
Command: d2r-query mapping1.ttl "SELECT * { ?s ?p ?o } LIMIT 10"

Step 7: Generate an RDF dump using command.
Command: dump–rdf mapping.ttl –o dump.nt
Above are the steps through which the relational database can be accessed and also be transformed into RDF dump using D2R server. D2RQ can perform the task through command line or GUI.

E. Mapping / Processing Non-Relational Data in Semantic web (using SPARQL)

This section focuses on mapping of semi-structured and unstructured data through SPARQL using LOD and Hadoop framework. Here SPARQL processing for mapping these two domains of data has been explored. Although Hadoop may handle various types of data, but is quite useful for handling heterogeneous data in it and so is placed in this section of mapping Non-Relational data into Semantic web.

1) SPARQL and LOD (Linked Open Data)

As SPARQL Query is the standard way to query semantic web data and it has been seen with the discussion in previous sections, how query based approach is more efficient to get data from relational databases. And for the same SPARQL query is playing magnificently good with translation into SQL query. In this section the usage of SPARQL query is being analyzed with the latest technologies of present era of computer science. Linked Open Data is playing a significant role in social networks and interlinking of data with different sources. The idea of Linked data has been taken from semantic web technologies and used to assist social networks domain.
Through Linked data environment it is quite simple to share data across the web. Further usage of semantic web technologies such as RDFS, OWL, and SPARQL are handy to build applications around that data. SPARQL can also be used as a protocol and query language for Linked Open Data (LOD). LOD is a blend of Linked Data and Open Data, which collectively linked and uses open sources of data. LOD is inferring knowledge out of the data which is interlinked [35]. DBpedia is a case of large linked dataset, which makes available the content of wikipedia in RDF. Another linked open dataset is wikidata[14], which is a collaborative knowledgebase hosted by Wikimedia foundation. Hartig and Perez [36] proposed LDQL (Linked Data Query Language) and compare it with SPARQL query processing to fetch web of linked data.

2) SPARQL Processing on Hadoop framework

As data is growing at very fast speed and Bigdata technology is handling large data efficiently and effectively. Big data technology handles and process different type of data (structured/unstructured/semi-structured) focusing 4V's (Volume, Variety, Velocity and Varacity), and for the same Hadoop framework is supporting and performing well with its cluster based approach along with support of NoSQL databases. Semantic web and Bigdata technologies, both are the latest technologies and having lots of scope for integration of existing technologies of both for better results. Hadoop is having its own framework and technology to query data from Hadoop cluster. On the basis of the literature survey represented in this paper, it is been analyzed that SPARQL query can also be supported by Hadoop technologies. The main techniques/tools that support semantic web query language are MAP-REDUCE, PigSPARQL, SPARQL with Apache Spark. Sempala is another tool used for SPARQL query processing on Hadoop cluster which uses query based approach for SPARQL-to-SQL-to-Hadoop pattern with selective queries to perform the task. It is noticed that SPARQL query processing with MapReduce is taking more time so it is better to explore execution with either spark or impala for better results. Similarly, Semantic web data may also be mapped with NoSQL databases like Neo4J, HBase, MongoDB etc. SPARQL query compatibility may be explored for processing as it is seen that SPARQL is capable to fetch information from databases and other different sources across the Web. Only the issue is compatibility and fast transformation of SPARQL into others.

V. Sparql Usage for Data Retrieval And Mapping (A Proposed Model for Information Processing)

SPARQL query is having a major role in mapping data from different data sources and this is shown in the proposed model using a block diagram below in figure 8. The block diagram explains how relational and non relational data can be mapped using SPARQL query / tools in semantic web. Further it has been analyzed that various tools are working on query based approach for better mapping with SPARQL query conversion.

The below model also proposes the solution for handling large RDF datasets using Hadoop, which is an open source framework and platform that assists large datasets effectively using multi-node clustering. HDFS is a file system and MapReduce/Spark is a processing framework for storing and processing large data in Hadoop system. Sqoop and Flume are used for the data transformation from relational databases and non-relational data respectively. Both transformed the data into HDFS and different data (Structured/ Semi-Structured and Unstructured) may be uploaded into HDFS. Further, PIG which is a scripting language is written in Pig Latin, Hive is a query language used in Hadoop for processing columnar dataset and HBase is used for processing NoSQL databases. These three standalone systems are working on the top of HDFS system for processing and managing data.

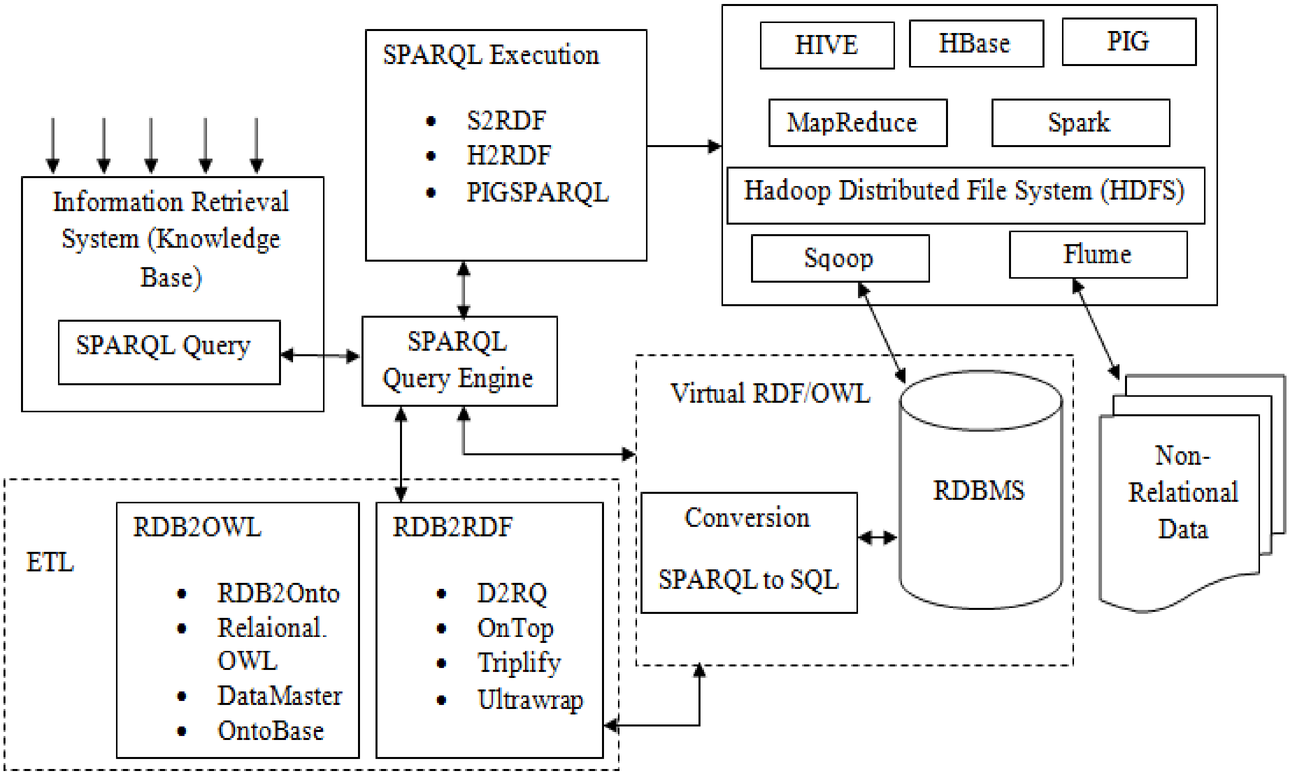


Figure 8. A Proposed Model for Information Processing from Relational and non-Relational data Using SPARQL.

SPARQL Query may also be used to map data stored in Hadoop system. Various tools is being reviewed in literature survey, different tools are using SPARQL query and join based approaches for mapping like S2RDF

(Processing SPARQL through Spark) [9], H2RDF+ (Using MapReduce and HBase Indexes) [19], PigSPARQL (SPARQL with PIG) [37]. Further from large RDF data stored in framework, some intelligent information can also be retrieved using various data mining techniques [38].

All conversions which are using Query Based technique is having advantage on others transformation majorly due to durability and maintenance of existing systems.

VI. Conclusion And Future Work

In this paper, the SPARQL Query Usage has been analyzed for performing mapping with other data sources available in different formats and storage. Also, a brief analysis is presented on transformation of relational database into ontology or RDF. The approaches of transformation/mapping have been discussed and different tools have been analyzed like D2RQ, OnTop, Triplify, Virtuoso RDF Views etc. Direct query based methods have been presented along with some plug-ins used with protégé tool in transformation and mapping. SPARQL runtime transformation with other query is found better and efficient due to the benefit of managing consistency of data all time, as there is no need to generate RDF dump or OWL for every change. An illustration of D2RQ server is also shown with mysql database with transformation as well as On-Demand Mapping approach. Further the compatibility of SPARQL query with other domains like LOD and Hadoop has been revisited. SPARQL usage has been explored using MapReduce, Spark, and other tools. A Model for information processing has been presented which may be treated as a common platform for managing and mapping data into semantic web using SPARQL engine utility. The storage of semantic web data on Hadoop framework and its processing is having a wide scope to explore like RDF fragmentation, Storing with indexes, and tool like S2RDF, H2RDF+, Jena With HBase, PigSPARQL and others may be analyzed for future work. Integration of semantic web and big data technologies also has a very wide research scope to explore.

References

[1] E. Prud'hommeaux, A. Seaborne, “SPARQL Query Language for RDF,” W3C Recommendations , Jan , 2008.

[2] RDF Working Group, “RDF,” W3C Working Group, February 2014.

[3] A. Cuzzocrea, R. Buyya, V. Passanisi and G. Pilato, "MapReduce based Algorithms for managing Big RDF Graphs: State-of-art analysis, paradigms and future directions," in Cluster, Cloud and Grid Computing (CCGRID), 2017.

[4] H. Naacke, O. Curé and B. Amann, "SPARQL query processing with Apache Spark," arXiv.org Database, 2016.

[5] M. Franck, F. Moantagnat and C. Zucker, "A Survey of RDB to RDF translation approaches and Tools," laboratory informatiue, signaux et systems de Sophia antipolis, 2014.

[6] M. Hazber, R. Li, X. Gu, G. Xu and Y. Li, "Semantic SPARQL query in a relational database based on ontology construction," in International conference on Semantics, Knowledge and Grids, IEEE, 2015.

[7] OWL Working Group, "Web Ontology Language," 2012.

[8] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language," 21 March 2013.

[9] A. Schatzle, M. Przyjaciel, S. Skilevic and G. Lausen, "S2RDF:RDF Querying with SPARQL on Spark," in VLDB Endowment, 2016.

[10] M. Grobe, "RDF, Jena, SPARQL and the Semantic Web," in SIGUCCS, Indianapolis, Indiana, USA, 2009.

[11] O. Hartig and G. Pirro, "SPARQL with Property Paths on the Web," Semantic Web Journal IOS Press, 2016.

[12] D. Spanos, P. Stavrou and N. Mitrou, "Bringing relational databases into the Semantic Web: A survey," Semantic Web Journal IOS Press, pp. 169-209, 2012.

[13] B. DuCharme, Learning SPARQL: Querying and.updating with SPARQL 1.1, O’REILLY, 2nd Edition., 2013.

[14] T. White, Hadoop: The Definitive Guide, April: O'REILLY, 2015.

[15] J. Cardoso and A. Pinto, "‘The Web Ontology Language (OWL) and its applications," IGI Global, 2015.

[16] B. Motik, P. Patel-Schneider and B. Grau, "OWL 2 Web Ontology Language," 2012.

[17] O. Hartig and R. Heese, "The SPARQL Query Graph Model for Query Optimization," in In Proceedings of the 4th European Semantic Web Conference (ESWC), Innsbruck, Austria, 2007.

[18] A. Chebotko, S. Lu, H. Jamil and F. Fotouhi, "Semantics Preserving SPARQL-to-SQL Query Translation for Optional Graph Patterns,," Technical Report TR-DB-052006-CLJF., 2016.

[19] P. Nikolaos, K. Ioannis, T. Dimitrios, K. Panagiotis and K. Nectarios, "H2RDF+: High-performance distributed joins over large-scale RDF graphs," in Big Data, 2013 IEEE International Conference on, 2013.

[20] J. Panawong, T. Ruangrajitpakorn and M. Buranarach, "An Automatic Database Generation and Ontology Mapping from OWL File," JIST, 2016.

[21] B. Bellini, P. Nesi and A. Venturi, "Linked open graph: Browsing multiple SPARQL entry points to build your own LOD views," Journal of Visual Languages & Computing, vol. 25, no. 6, pp. 703-716, December 2014.

[22] H. Oh, S. Chun, S. Eom and K. Lee, "Job-Optimized Map-Side Join Processing using MapReduce and HBase with Abstract RDF Data," in International Conference on Web Intelligence and Intelligent Agent Technology, 2015.

[23] T. Garcia and T. Wang, "Analysis of Big Data Technologies and Method - Query Large Web Public RDF Datasets on Amazon Cloud Using Hadoop and Open Source Parsers," in Seventh International Conference Semantic Computing (ICSC), Irvine, CA, USA, 2013.

[24] K. D. Mogotlane and J. Domneu, "Automatic Conversion of Relational databases into Ontologies: A Comparative Analysis of PROTÉGÉ Plug-ins Performances," International Journal of web & Semantic Technology (IJWest), 2016.

[25] K. Anyanwu, "A vision for SPARQL multi-query optimization on MapReduce," in 29th International Conference on Data Engineering Workshops (ICDEW), Brisbane, QLD, Australia, 2013.

[26] J. Lu, F. Cao, L. Ma ,Y. Yu1 and Y. Pan, “An Effective SPARQL Support over Relational Databases”, Springer-Verlag, Berlin Heidelberg, 2008.

[27] D. E. Spanos, P. Stavrou and N. Mitrou, "Bringing Relational Databases into the Semantic Web: A Survey," Semantic Web IOS Press, 2012.

[28] T. Health and C. Bizer, "Linked Data: Evolving the web into a Global Data Space," 2011.

[29] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro and G. Xiao, "Ontop: Answering SPARQL Queries over Relational Databases," Semantic Web Journal,IOS Press., 2016.

[30] Q. Trinh, K. Barker and R. Alhajj, "RDB2ONT: A Tool for Generating OWL Ontologies From Relational Database Systems," in Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, 2006.

[31] J. Barrasa, O. Corcho and A. Perez, "R2O, an Extensible and Semantically Based Databaseto-ontology Mapping Language," in Second Workshop on Semantic Web and Databases, Springer-Verlag, Canada, 2004.

[32] C. P. d. Laborda and S. Conrad, "Relaional.OWL- A Data and schema representation format based on OWL," in Second Asia-Pacific Conference on Conceptual Modeling (APCCM2005),, 2005.

[33] G. Bumans, "Relational Database information availability to Semantic Web technologies," 2014.

[34] R. Gupta and S. Malik, "SPARQL Usage for Mapping Semantic Web Data (OWL/RDF) from Relational Database: A Revisit," in System Modeling and Advancement in Research Trends (SMART-2017), 2017.

[35] S. B. V. T. Aver, “Linked Open Data-Creating Knowledge out of the Interlinked Data", Springer, 2014.

[36] O. Hartig and J. Pérez, “LDQL: A Query Language for the Web of Linked Data”, Journal of Web Semantics, pp 9-29, 2016.

[37] A. Schätzle, M. Przyjaciel-Zablocki, T. Hornung and G. Lausen, "PigSPARQL: a SPARQL query processing baseline for big data," in ISWC-PD '13 Proceedings of the 12th International Semantic Web Conference (Posters & Demonstrations Track), Sydney, Australia, 2013.

[38] M.Ahmed , “Semantic Based Intelligent Information Retrieval through Data Mining and Ontology”, International Journal of Computer Sciences and Engineering. pp. 210-217, 2017.

APPENDIX

LUBM[12] - Lehigh University Benchmark with SWAT project presented 14 SPARQL Queries which are treated as benchmark for SPARQL.

Query 2-

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X, ?Y, ?Z
WHERE
{?X rdf:type ub:GraduateStudent .
  ?Y rdf:type ub:University .
  ?Z rdf:type ub:Department .
  ?X ub:memberOf ?Z .
  ?Z ub:subOrganizationOf ?Y .
  ?X ub:undergraduateDegreeFrom ?Y}
```

ENDNOTES

[1] http://hadoop.apache.org/
[2] https://spark.apache.org
[3] http://d2rq.org/
[4] https://en.wikipedia.org/wiki/Linked_data#Linked_open_data
[5] https://protege.stanford.edu/
[6] https://jena.apache.org/documentation/query/
[7] https://jena.apache.org/documentation/fuseki2/.
[8] http://www.ldodds.com/projects/twinkle/
[9] https://www.blazegraph.com/.
[10] https://www.w3.org/2001/sw/wiki/Sesame.
[11] http://wiki.dbpedia.org/
[12] http://swat.cse.lehigh.edu/projects/lubm/
[13] https://www.w3.org/2001/sw/wiki/Ontop.
[14] https://www.wikidata.org/wiki/Wikidata:Main_Page

Authors Profile

Mr. Rupal Gupta is a Reseach Scholar at USIC&T, Guru Gobind Singh Indraprashtha University, Delhi, India. His area of interests are Semantic Web, SPARQL Query Processing, Big Data and Data Mining. He received his Master’s, MCA from UPTU, Lucknow and M.Tech.(IT) from USIT, GGSIPU, New Delhi . He has published papers in IEEE conferences. He is currently working as an Assistant Professor at Teerthanker Mahaveer University, Moradabad and having 11 years of teaching experience.

Dr. Sanjay Kumar Malik completed his Ph.D. in the area of Semantic Web from USIC&T, GGSIP University, Delhi. He is currently working as Associate Prof. in University School of Information, Communication and Technology, GGSIP University. He has more than 18 years of industry and academic experience in India and abroad (Dubai and USA). His areas of research interest are Semantic Web and Web Technolgies. He has several research papers in reputed international conferences (India/abroad) and Journals. He has been session chair in several international IEEE/Springer conferences and honoured with third best researcher award, 2011 by GGSIP University for his contributions in research.

.