Mapping Relational Database to Resource Description Framework using Direct Mapping Method

Murtada K.Elbashir[1], Mohamed A.Aboelhassan[1], Abubaker A.Ali[1]

[1]Faculty of Mathematical and Computer Sciences, University of Gezira, Wad Madani, Sudan.

ABSTRACT

The Resource Description Framework (RDF) is one of the World Wide Web Consortium (W3C) specifications and it is part of W3C semantic web activities. Its main idea is to make the statements about web resources in the form of subject–predicate–object expressions. Relational databases are considered to be the main sources for the web. To integrate them into semantic web, they should be mapped to RDF. The aim of this study is to design a tool that converts relational databases to RDF based on direct mapping method. The designed tool can express the Meta data of relational database in a minimally constrained, flexible, but meaningful way so that web data can be exchanged and integrated without loss of semantics.

Keyword: Relational Database, Resource Description Framework (RDF), Semantic Web.

INTRODUCTION

When the web was started in 1994 the main objective of its inventor, Tim Berners-Lee is to make the exchange of data and scientific documents quick and easy for scientists in the laboratories, and later to the whole world. This is what has already happened, when the web begun to spread gradually until everyone was talking about web services and engages in it. The first generation of web is called Web 1.0 was started in 1994 and gradually diminished until 2001. In this generation been using the web in e-publishing, The process of publishing on the World Wide Web (Web) is limited to those who have experience in programming and also to organizations and companies, It was only a few of the individuals are those who create a page or site for them on the web (Aghaei *et al*,2012).

The second generation of the web called Web 2.0 and also called two-dimensional web is defined by Dale Dougherty in 2004 as a read-write web and we still use various tools in many fields. This generation of web generations focused on the social side of the web, in the first generation the internet user was a consumer of information and rarely able to participate in building a web content. But now By virtue of technology, which facilitated the process of publishing on the Internet, such as Wikis, blogs, YouTube and other tools of the second generation of the Web become the individual's ability to build a site in a few minutes and share his thoughts on the Web as if it was used text editor program (Aghaei *et al*,2012).

The third generation of the web is Web 3.0, which is a development of the second generation of the Web. The most important aspect of Web 3.0 is a semantic it is the idea of the web innovator and the aim of this idea is to make the semantic web global intermediary to exchange data, information and the human knowledge. Semantic Web According to its inventor can transform the vast amount of data and information sources available on the Internet from than just units made up of bits system to data understood by computer programs that are created specifically for this purpose. We want from the Web with semantics that make the machine understand what: did mean page on the web? ; did mean the links in the page? If we do so the future programs can give smart results and serve our needs backed by a kind of artificial intelligence that is means more intelligent Web from the today's Web (Aghaei *et al*,2012).

In order to achieve these objectives the Semantic Web depends on a set of tools namely: advanced programming languages such as Extensible Markup Language (XML), Resource Description Framework (RDF) and Ontology Web language (OWL) (Pomponio and Viale,2013)**.**

The main concern of the semantic web technologies is the meaning of the data and not its structure. While other technologies such as relational databases and the World Wide Web itself concentrate on the structure of the data. Technically, there are three standards that are essential for the semantic Web these standards are:

1- RDF, which is the data modeling language for the Semantic Web. RDF is an XML-based language that describes the information that is contained in a Web resource. All Semantic Web information is stored and represented in an RDF.

2- SPARQL (SPARQL Protocol and RDF Query Language), which is the query language of the Semantic Web. It is specifically designed to query data across various systems.

3- OWL (Web Ontology Language), which is the schema language, or the knowledge representation of the Semantic Web. With OWL concepts can be defined composably, this will enable the reusability of these concepts as much and as often as possible. Composability means that each concept is carefully defined so that it can be selected and assembled in various combinations with other concepts as needed for many different applications and purposes (Polleres, 2010).

Semantic data are stored in a repository, which its data model is a part of the semantic web architecture. Data are stored in Semantic Web in a structure called RDF Schema (RDFS). Web Ontology Language (OWL) is used to interpret the data in RDF. Data in RDF Schema not only has literals but also a semantic meaning attached to it. This meaning has different informal hierarchies and formal taxonomies in its knowledge domain (space-domain) (Mallede *et al*, 2013).

This leads to the introduction of a consensually shared view of concepts called Ontologies. Ontology is a formal, explicit specification of a shared conceptualization. The specifications use relations, functions, constraints, and axioms to conceptualize the abstract model. Ontology definition refers to the fact that the expressions must be machine readable; hence, natural language is excluded. RDF was designed for situations where Web data need to be processed and exchanged by applications without the intervention of people. The ability to exchange data between different applications means that the data may be made available to applications other than those for which they were originally intended (Obitko *et al*, 2004).

In this work we will study the technique of RDF which allows relational database content to be shared and reused across applications, enterprise, and community boundaries on the web and to enhance reasoning by SPARQL (the query language of the semantic web) queries, This means exploiting of relational data which locked in relational databases and put it in a machine-readable format to make it readily interpreted by machines. Then a tool that converts Relational Database to Resource Description Framework will be designed.

RELATED WORKS

Juan *et al*. propose that the success of the Semantic Web depends on enabling access to relational databases and their content by semantic methods. They survey direct mapping approaches which try to bridge the gap between relational databases and the Semantic Web in an easy and automatic way (Juan *et al*., 2009).

Sahoo *et al.* pointed out that researchers and practitioners have provided different mechanisms with which to tackle the RDB2RDF conversion process. However, most of the current RDB2RDF tools provide different proprietary mapping languages for the mapping process (Sahoo *et al.*, 2009).

In 2007 the W3C created the RDB2RDF Working Group4 to standardize languages for mapping relational database schemas into RDF and OWL. Sahoo *et al.* conducted a wide scope review, addressing theoretical articles, proofs of concept, domain-specific projects as well as generic mapping tools. The goal of their survey was not to get into the details of each approach, but to provide the RDB2RDF researchers with a comprehensive overview of the different approaches that had been investigated so far, in order to serve as a basis for the definition of R2RML (Sahoo *et al.*, 2009).

Konstantinos *et al*. propose that the large volume of data residing in relational databases led to create methodologies and tools able to map Relational Databases with the Resource Description Framework. They present Relational Database to Ontology Transformation Engine (RDOTE), which is a framework for easily transporting data residing in Relational Databases into the Semantic Web. RDOTE is available under GNU/GPL license and provides friendly graphical interfaces, as well as enough expressivity for creating custom RDF dumps. RDOTE provides the Semantic Web research community and domain experts with the necessary means for easily enriching Ontology schemata with the vast amount of data currently residing in relational databases.

It also enables quick instantiations of new ontology schemata for testing and experimentation. By allowing easy transportation of legacy data into semantically aware data structures, RDOTE aspires to bring the Semantic Web vision one step closer (Konstantinos *et al*., 2010).

Hert *et al.* proposed a feature-based comparison framework that they have applied to state of the art mapping languages. Their framework is derived from the use cases and requirements described by the W3C RDB2RDF Working Group. The mapping languages are sorted into four categories: direct mapping, read-only general-purpose mapping, read-write general-purpose mapping, special-purpose mapping.

In their research they focused on the comparison of the mapping language features and expressiveness, and it does not address the implementations proposed by their authors or the way queries are rewritten (Hert *et al*., 2011).

Edgard *et al.* propose a process that transforms data stored in relational databases (RDBs) into sets of Resource Description Framework (RDF) triples, which is known as triplificationor RDB2RDF.they introduce an Eclipse plug-in that supports the entire conversion process. Its architecture takes into consideration the specificities of the triplification process by providing a modular structure that encapsulates the stable and well-understood components separately from the volatile, change-prone mapping strategies (Edgard *et al*., 2013).

Relational Database:

The relational database consists of different relational objects that are grouped into relational schema called database schema. The database schema S(T1, T2, … Tn), where n is the number of relational tables, and T refer to the table objects under the schema. The relational data is stored in tables T(A1, A2,…, Am), where A is the column or attributes of the table and m is the number of these columns or attributes. Each column or attribute has its own domain and range. The primary and foreign keys are considered as database constraints during mapping. Each table T consists of a set of tuples t1, t2, …,tn where n is the number of tuples in a table. Each tuple t is defined as a set of values (v1, v2, …,vn), where vi is the value corresponding to column attribute Ai . The Individual attribute values in a tuple are represented using the attribute and value pair as t(Ai, Vi). The definition of a relationship in relational databases is a situation that exists between two relational tables indicated by a foreign key constraint. The foreign keys are used to establish a reference from any row in a table to exactly one row in a (potentially different) table. The direct graph conveys these references, as well as each value in the row.

There are many types of relationships in the Relational databases one of them is the binary relationship which exists between two tables, the other types may consist of a group of binary relations ;that may form a pattern that involve three (ternary), four (quaternary) or more tables that are commonly referred as Nary relationships. The tables which are involved in a relationship are classified as strong or weak tables depending on where the foreign key is placed. A strong table is indicated by a primary key database constraint using one or more column attributes while a weak table uses a foreign key to refer to the strong table. Binary relationships are represented using a foreign key constraint that involve one or many cardinalities each side to form a one to one, one to many and many to many relationship (Mallede *et al*,2013).

MATERIALS AND METHODS

Since the process of converting data stored in RDBs to RDF is a critical step in the move to the Web of Data, our aim here is to design a tool that does the conversion based on direct mapping method. Vast amount of data in enterprises and on the web resides in relational databases.

The conversion will make these data of the relational databases available in a machine-readable format and can be integrated with semantic web applications.

Many research efforts are taken as a base for designing the tool. These researches are reviewed carefully because RDF is an integration platform for data from multiple sources. For semantic web to interact with relational databases W3C Released tow standard methods for mapping relational database to resource description framework these standards are direct mapping and R2RML. Direct Mapping is an automatic default mapping and R2RML is a mapping language where users can customize the mappings. These two standards enable more and more relational data to be available in the Linked Data cloud and part of Semantic Web applications.

General Description of Direct Mapping:

The direct mapping defines a simple transformation, providing a basis for defining and comparing more intricate transformations (Auer *et al*, 2010). It defines an RDF graph representation of the data in a relational database and takes relational database (data and schema) as input. Its output will be a RDF graph in the form of subject-predicate-object, which is called the direct graph. The direct graph is a formula for creating an RDF graph from the rows of each table and view in a database schema. The direct mapping maps database tables into classes. Direct mapping create for each class an RDF Repository object. The structure of the RDF repository is based on the triple format subject-predicate-object (Sequeda *et al*, 2012). Thus the research methodology is divided into four phases these phases as illustrated in the figure bellow:

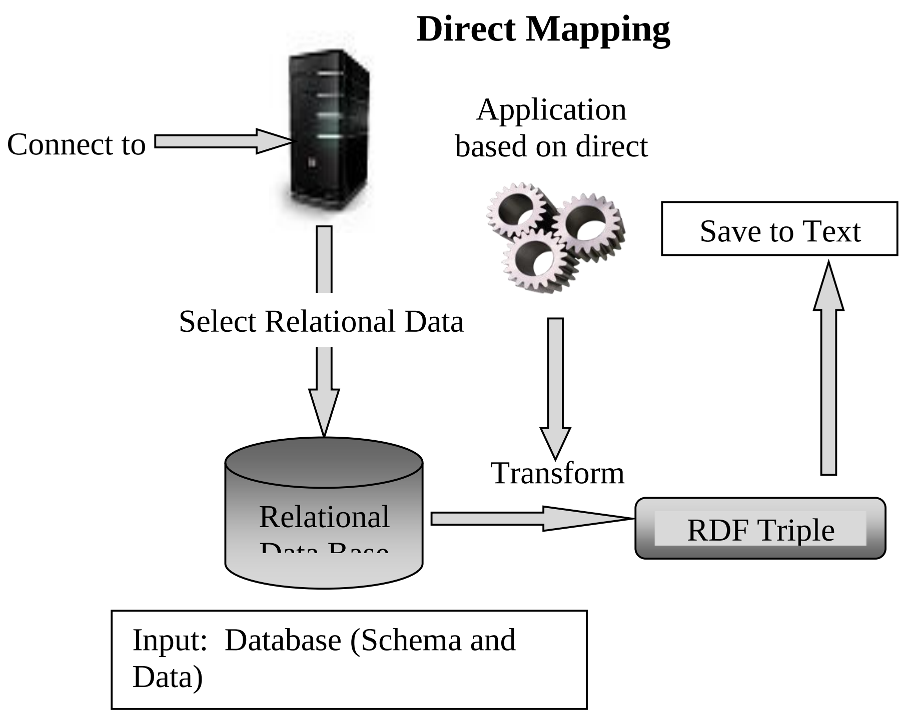

Figure 1: the phases of direct mapping method.

Mapping Relational Database to Resource Description Framework using Direct Mapping Method

The final conversion to a RDF will be accomplished using the following four phases:

Phase 1: connect to a database server using the server name, user name and password.

Phase 2: if the connection to the database server is successful, then all the databases in the given server will be loaded in a list, then the required database will be selected forms this list

Phase 3: Transform the selected relational database to RDF by using the proposed application, which is based on direct mapping method.

Phase 4: finally the generated direct graph can be saved in a txt file, (Mallede *et al*, 2013) and (Sequeda *et al*, 2012).

Description of mapping relational Database table to RDF class:

The same name of the table in relational database is used to map the class that represents it in the generated RDF during the mapping process. The class name is used to map subsequent relational columns into semantic class properties. These classes represent repositories of data to hold the relational data after the end of mapping process.

The structure of the classes in RDF triples in the form of subject-predicate-object and each row in the relational table produces a set of triples with a common subject.

The subject is an Internationalized Resource Identifier (IRI), which is complement to Uniform Resource Identifiers (URIs). URIs is string of characters used to identify the name of resources. IRI is formed from the combination (base IRI, table name, primary key column, and primary key value). Each row implies a set of triples with a shared subject when there is no primary key, in such case the subject will be a blank node.

The predicate for each column is an IRI is a combination (base IRI, table name, and the column name). The RDF literals are considered as the values, which are formed from the lexical form of the column value.

Each foreign key produces a triple with a predicate composed of the foreign key of column names, the referenced table, and the referenced column names.

The object of these triples is the row identifier for the referenced triple. These reference row identifiers must coincide with the subject used for the triples which are generated from the referenced row.

The direct mapping does not generate triples for NULL values. It is not known how to link the standard SQL semantics of the NULL values of the source RDB with the behavior of the obtained RDF graph.

Description of Mapping Rules:

As mentioned earlier that each row in the database produces a set of RDF triples with a subject, predicate, and object, these triples are composed as follows **:**

- **Shared Subject**: A row RDF node, which may be an IRI or a Blank Node, is generated for each row.
- **Table Triples**: A triple generated by the row with the following:
  - Predicate: the rdf: type property.
  - Object: the object represents the table IRI for the table.
- **Literal Triples:** Each column with a non-null value (the column(s) that are part of the primary key are inclusive), and that either is not the only part of a foreign key or is the only part of a foreign key that references a candidate key, generates a triple with the following:
  - Predicate: the column IRI for the column.
  - Object: represents the RDF literal with an XML Schema data type corresponding to the SQL data type of that value. String data types are expressed as an RDF plain literal.

- **Reference Triples:** Columns that part of a foreign key and with non-null values in the row, and it generate triples with the following:

Mapping Relational Database to Resource Description Framework using Direct Mapping • Method

  - Predicate: the column IRI for the columns that are part of the foreign key.
  - Object: the row RDF node for the corresponding referenced row (according to the foreign key).

Direct Graph Definition:

A column in a table forms a literal property IRI, which consists of the concatenation of (the percent-encoded form of the table name, the hash character '#', the percent-encoded form of the column name).A foreign key in a table composes a reference property IRI, which consists of the concatenation of (the percent-encoded form of the table name, the string '#ref-' for each column in the foreign key).

Any input database with a given schema has a direct graph defined as the union of the table graphs for each table in a database schema. The table graph is the union of the row graphs for each row in a table. The row graph is an RDF graph consisting of the triples (The row type triple, reference triples for each <column name list> in a table's foreign keys where none of the column values is NULL, a literal triple for each column in a table will be part of RDF graph if the column value is non-NULL). The row type triple represents an RDF triple with the following (Subject: the row node for the row, Predicate: the RDF IRI rdf:type, Object: represents the table IRI for the table name). The literal triple is an RDF triples with the following (Subject: the row node for the row, Predicate: represents the literal property IRI for the column, Object: the R2RML natural RDF literal representation of the column value as defined in R2RML, Natural Mapping of SQL Values).

The reference triple is an RDF triples with the following (Subject: the row node for the row, Predicate: represents the reference property IRI for the columns, Object: represents the row node for the referenced row).

The algorithms for accomplishing the mapping as described in the previous sections can be described as following:

• **Domain Data Knowledge** *Referencing Table_ T, Referenced Table_ T',* *Column attribute_ A, primary key_ pk(T), foreign key_ fk(T).* *Begin* *If ((A in (fk(T))) AND (A in (pk(T))))* *then .* *<owl:Classrdf:ID="T" >* *<rdfs:subClassOfrdf:resource="#T'"/>* *</owl:Class>* *End if .* *End.*	• **map_Database() Algorithm** *Procedure map_Database (S)* *Input: Schema_ S* *Begin* *map_Tables(S).* *map_Columns(S).* *map_Constraints(S).* *map_Relationships(S).* *End.*
• **map_Tables() Algorithm** *Procedure map_Tables(S)* *Input: Schema_ S* *Output: Class_ C, RDF_Repository C_RDF, OWL Class* *Begin* *For each table "Ti" in S loop.* *Create Class Table "Ci".*	• **map_Columns() Algorithm** *Procedure map_Columns(S)* *Input: Schema_ S, Table_ T, Column attribute_ A* *Output: Property_ P, OWL:DatatypeProperty* *Begin* *For each table "Ti" in S loop .*

Create RDF Repository "Ci_RDF" using Class Table "Ci" and a TRIPLE type attribute. *<owl:Classrdf:ID="Ci" />* *End loop .* *End.*	*For each Column "Aj" in Ti loop.* *Get mapped Class Table "Ci" of "Ti".* *Set "Aj" as Property Column has "Aj".* *get_&xsd;type_equivalent (Aj) .* *<owl:DatatypePropertyrdf:ID="hasAj">* *<rdfs:domainrdf:resource="#Ci" />* *<rdfs:rangerdf:resource ="&xsd;type_equivalent" />* *</owl:DatatypeProperty>* *End loop .* *End loop.* *End.*

- Mapping Relational Database to Resource Description Framework using Direct Mapping Method

• **map_Constraints() Algorithm** *Procedure map_Constraints(S)* *Input: Schema_ S, Table_ T, Referenced Table_ T', Column attribute_ A, primary key_ pk(T), foreign key_ fk(T), UNIQUE_ unq(A), NOT NULL_ nn(A), and CHECK _ck(A)* *Output: RDFS subClassOf, Property P, OWL cardinality properties* *Begin* *For each table "Ti" in S loop .* *For Column "Aj" in "Ti" loop.* *Get mapped Class Table Ci of Ti.* *If (Aj in (pk(Ti))) then .* *<owl:InverseFunctionalPropertyrdf:resource="# hasAj "/>* *<rdfs:subClassOf>* *<owl:Restriction>* *<owl:maxCardinality rdf:datatype="&xsd:nonNegativeInteger">* *</owl:maxCardinality>*	*rdf:datatype="&xsd:nonNegativeInteger">1* *</owl:minCardinality>* *</owl:Restriction>* *Else* *if (ck(Aj)) then .* *<rdfs:subClassOf>* *<owl:Restriction>* *<owl:onPropertyrdf:resource="#hasAj" />* *<owl:hasValuerdf:datatype="&xsd;string" > v(Aj)* *</owl:hasValue>* *</owl:Restriction>*

<table>
<tr>
<td rowspan="2"></owl:Restriction>
</rdfs:subClassOf>
Else
if (Aj in (fk(Ti))) then .
If (Aj in (pk(T’i))) then
<rdfs:subClassOfrdf:resource="#C’"
/>
End if .
<owl:
ObjectPropertyrdf:ID=”hasA”>
<rdfs:domainrdf:resource=”#C” />
<rdfs:rangerdf:resource=”#C’” />
</owl: ObjectProperty>
Else
if (unq(Aj)) then .
<owl:InverseFunctionalPropertyrdf:r
esource=”# has Aj ”/>
Else
if (nn(Aj) and (!pk(Aj)) then .

<owl:Restriction>
<owl:minCardinality
• Create_Relationships()
Algorithm
Procedure Create_Relationships(T,
T’, TYPE)
Output: RDFS subClassOf, OWL
Class, ObjectProperty
Begin
If (fk(T) = pk(T’)) then .
Create Class C _T_T’.
setpk(T) as Property hasP of Class
C_T_T’ .
setpk(T’) as Property hasP’ of Class
C_T_T’ .
Create RDF Repository
C_T_T’_RDF
using Class Table C_T_T’ and a
TRIPLE
type attribute.
Get mapped Class Table C of T.
Get mapped Class Table C’ of T’.
If (TYPE = ‘subClass’) then
<owl:Classrdf:ID="C"></td>
<td></rdfs:subCl
assOf>
End if .
End loop.
End loop.
End.</td>
</tr>
<tr>
<td>• Chek
_Relations
hips()
Algorithm
Procedure
Check_Rela
tionships(T,
T’)
Input:
Table_ T,
primary
key_ pk(T),
foreign
key_
fk(T),NOT
NULL
_nn(T)
Begin
If (fk(T) =
pk(T’) and
(fk(T) =
nn(fk(T)))
then .
CreateRelat</td>
</tr>
</table>

<table>
<tr>
<td rowspan="2"><rdfs:subClassOf
rdf:resource="#C'" />
</owl:Class>
Else
if (TYPE = 'Transitive') then
<owl:ObjectPropertyrdf:ID="pk(T)"
>
<rdf:typerdf:resource
="owl;TransitiveProperty"/>
<rdfs:domainrdf:resource="#C" />
<rdfs:rangerdf:resource="#C'" />
</owl:ObjectProperty>
End if .
End.</td>
<td>ionship(T,T
', subClass)
.
End if.
End</td>
</tr>
<tr>
<td>• Chek _Transitive Chains() Algorithm
Procedure Check_TransitiveChains(T, T')
Input: Table_ T, Column attribute _ A, primary key _pk(T),foreign key _fk(T)
Begin
For each column Ai in T' loop .
If (Ai in fk(T')) then
.
For each table Ti in S loop .
If ((Ai in pk(Ti)) and (Ti != T)) then .
create_Rela</td>
</tr>
</table>

	tionships(T, Ti, Transitive) . *End if.* *End loop.* *End if.* *End loop* *End.* • **Chek _Disjointness() Algorithm** *Procedure Check_Disjointness(T, T')* *Input: Table _T, Column attribute _A, primary key _pk(T),foreign key _fk(T)* *Output: RDFS subClassOf, OWL Class, disjointWith* *Begin* *For each column Ai in T' loop .* *If (Ai in fk(T')) then .* *For each table Ti in S loop .* *If ((Ai in pk(Ti)) and (Ti != T)) then .* *If ((ALL) fk(Ti) NOT in*

	(ALL) pk *(T)) then .* *<owl:Classr* *df:ID="Ti">* *<rdfs:subCl* *assOfrdf:res* *ource="#T"/* *>* *<owl:disjoin* *tWithrdf:res* *ource=" #T* *"/>* *</owl:Class* *>* *End if .* *End loop.* *End if.* *End loop.* *End.*

RESULTS AND DISCUSSION

In this study a tool that Maps Relational Data Base to Resource Description Framework using Direct Mapping Method was developed. The proposed tool is composed out of several interfaces. In this section we present some these interfaces with their description.

In the first interface (see figure 2), the type of databases management system that needs to be converted to RDF can be selected.

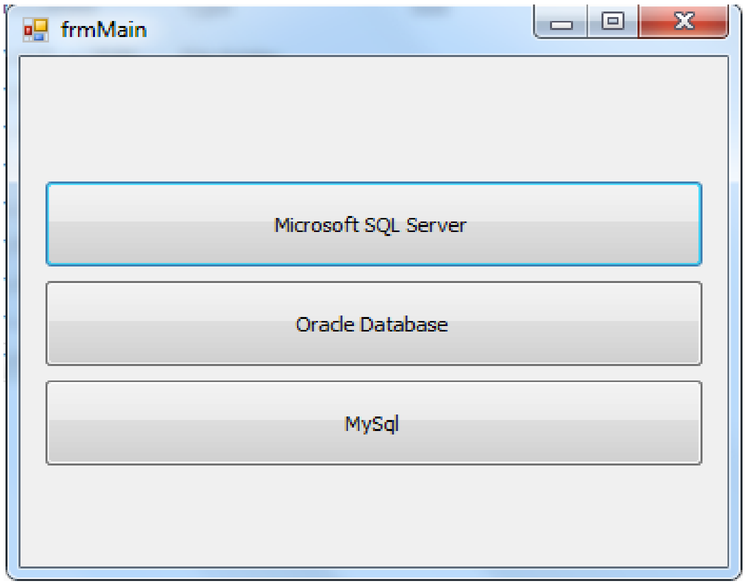

Figure 2: select the database management system.

The user can log in specific database through a log in form using the server name, the user name and the password. The login form of the proposed tool is depicted in figure 3.

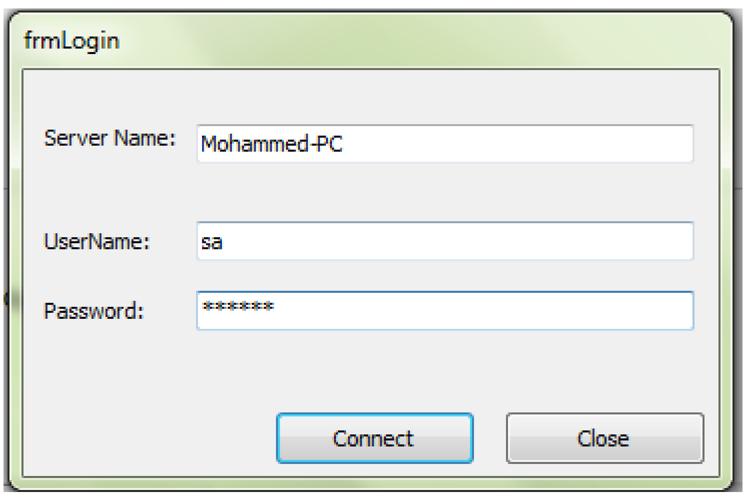

Figure 3: log in form.

If connection to the server succeeded then the interface in figure 4 will appear.

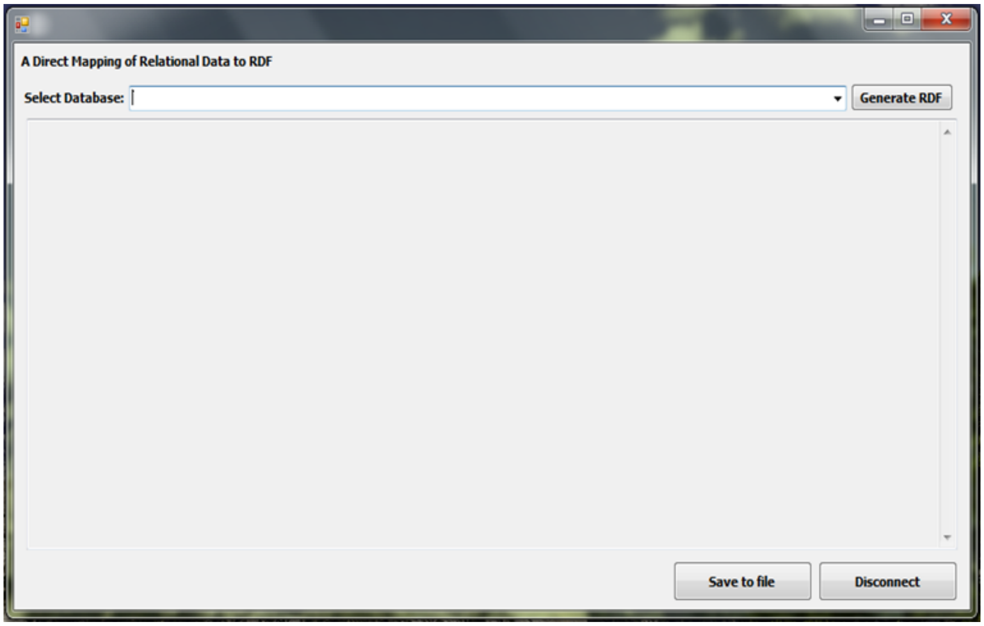

Figure 4: when connection succeeded.

Upon clicking on the combo box in the user interface of figure 4 then all the data bases that are stored in the server used for in the login step will appear. The appearance of the data bases is depicted in the interface that is shown in figure 5.

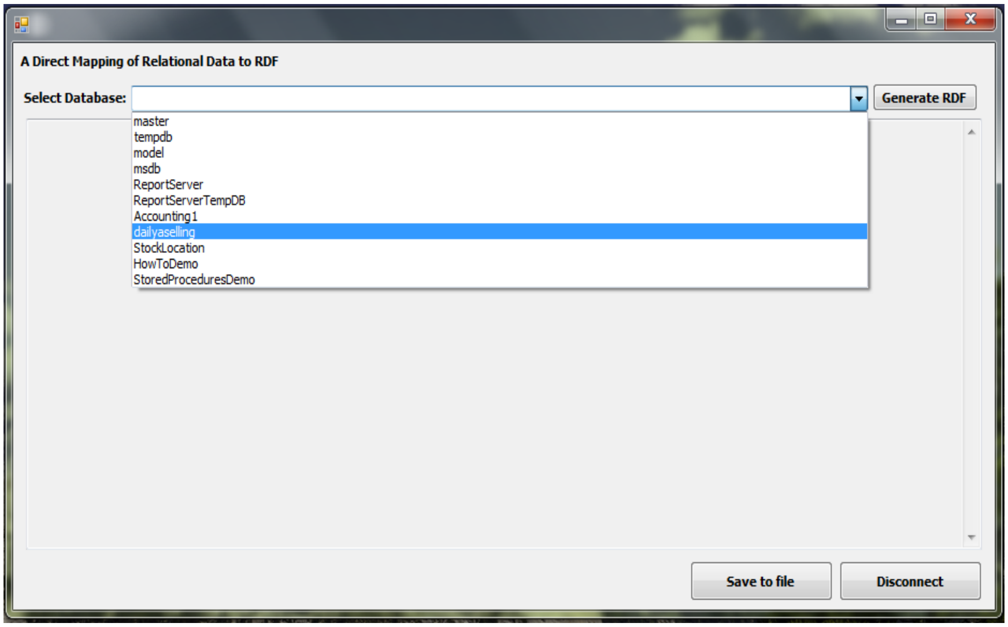

Figure 5: all databases.

Users can select a data base from the list and then click on generate RDF button to transform it to RDF format. Then the resulted RDF will appear in the user interface as shown in figure 6

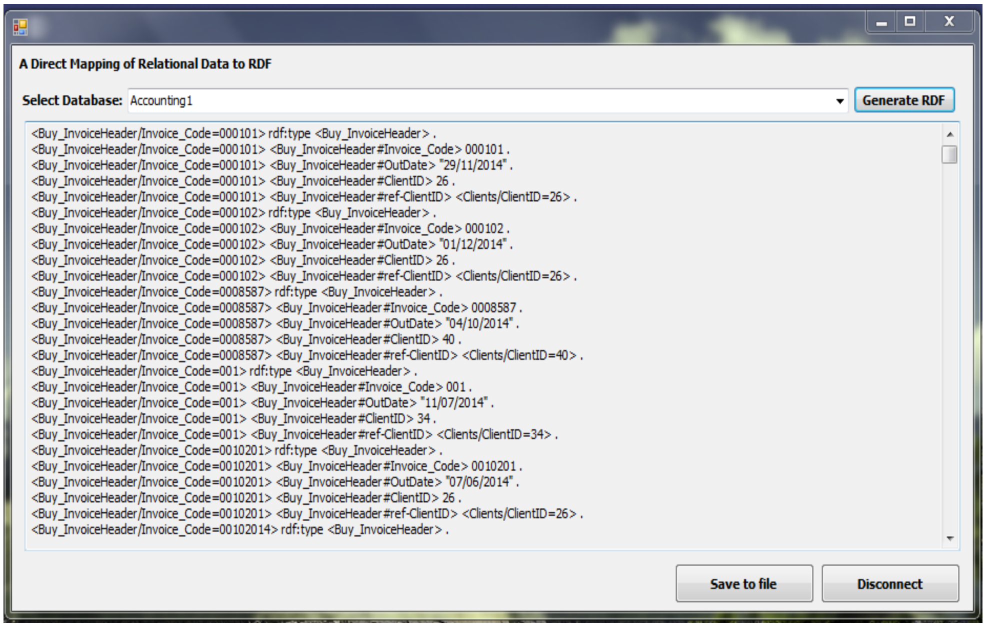

Figure 6: generated RDF**.**

The generated RDF can be saved in a txt file by clicked on save to file button then a dialog box in which the location of the file should be determined will appear as shown in figure 7.

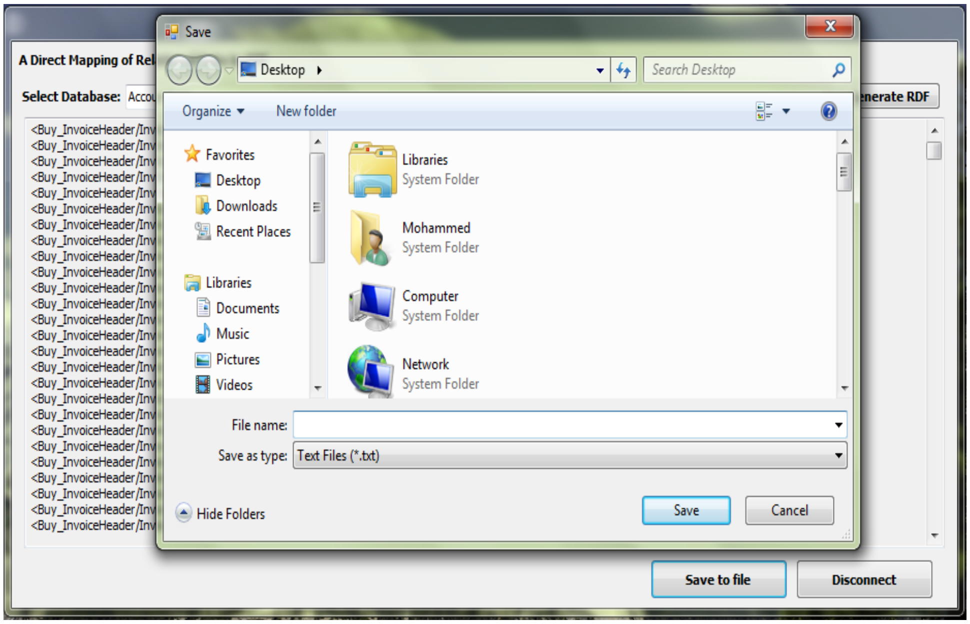

Figure 7: saving generated RDF in a txt file.

Discussion:

Based on the study on 2007(70%) of web site backed by relational data bases (Bin *et al*,2007) which contained 500 times more data than directly available and that three quarters of these databases are managed by relational databases management systems. To make this huge amount of data available in a machine-readable format (which interpreted by machines) and to integrate it in semantic web applications, it is necessary to transform the relational databases that hold them to RDF. Therefore the success of the Semantic Web depends on enabling access to relational databases and their content.

The generated RDF can be more expressive and data represented in RDF can be interpreted, processed and reasoned using machines through software agents. RDF can be enabled to effectively integrate data from multiple sources through the use of URIs for entities along with the ability to link them together using predicates.

The RDF format can capture metadata and structure of relational data base and also it can describe simple data structures to complete vocabularies/Ontologies to processing and inference rules. It can represent instance data, data structures, and schema.

CONCLUSION

The recent years are characterized by increasing use of semantic technologies both on a global scale (Semantic Web) as well as locally within enterprises, supported by the development of format and standards such as RDF, SPARQL, OWL and many others. The number and performance of tools for providing data can be interpreted be machines has grown and continues to grow. However, majority of data continue to reside in relational databases because they are efficient in terms of processing time and data volume, and they have precise definition. So a tool that convert relational database to semantic standards is needed. The aim of this research is to develop a tool that can convert RDB to RDF using the direct mapping method. The developed tool can generate the mapping file automatically by converting table to class and column to predicate. The use of this tool will result in notable advantages concerning documentation of the RDB data compared with textual and graphical documentation. Possible advantages is that the textual and graphical documentation is human readable only. The documentation of the developed tool will be machine and human readable and traceable.

REFERENCES

Aghaei, S., Nematbakhsh, M. A., & Farsani, H. K. (2012). Evolution of the world wide web: from Web 1.0 to Web 4.0. International Journal of Web & Semantic Technology, 3(1), 1-10.

Bin, H., Mitesh, P., Zhen, Z., & Kevin, C. (2007). Accessing the Deep Web: a survey. Communications of the ACM, 50(5), 94-101.

Hert, M., Reif, G., & Gall, H. C. (2011, September). A comparison of RDB-to-RDF mapping languages. In Proceedings of the 7th International Conference on Semantic Systems (pp. 25-32). ACM.

Mallede, W. Y., Marir, F., Vassilev, V. T., & Jing, Y. (2013). Algorithms for mapping RDB Schema to RDF for Facilitating Access to Deep Web. In Proceedings of the First International Conference on Building and Exploring Web Based Environments (pp. 32-41).

Marx, E., Salas, P., Breitman, K., Viterbo, J., & Casanova, M. A. (2013). RDB2RDF: A relational to RDF plug-in for Eclipse, 43(4), 435-447.

Obitko, M., Snasel, V., Smid, J., & Snasel, V. (2004, September). Ontology Design with Formal Concept Analysis. In CLA (Vol. 110).

Polleres, A. (2010). Semantic web technologies: From theory to standards. In 21st National Conference on Artificial Intelligence and Cognitive Science, NUI Galway.

Pomponio, L.,Viale, P. (2013, December). Semantic Web in Action.

S. Auer, L. Feigenbaum, D. Miranker, A. Fogarolli, and J. Sequeda. (2010, June). Use cases and requirements for mapping relational databases to RDF.

Sahoo, S. S., Halb, W., Hellmann, S., Idehen, K., Thibodeau Jr, T., Auer, S., ... & Ezzat, A. (2009). A survey of current approaches for mapping of relational databases to RDF.

Sequeda, J. F., Tirmizi, S. H., Corcho, O., & Miranker, D. P. (2009). Direct mapping SQL databases to the semantic web: A survey. Univeristy of Texas, Department of Computer Sciecnces TR-09-04.

Sequeda, J., Priyatna, F., & Villazón-Terrazas, B. (2012). Relational Database to RDF Mapping Patterns. In WOP.

Vavliakis, K. N., Grollios, T. K., & Mitkas, P. A. (2010, November). Rdote-transforming relational databases into semantic web data. In Proceedings of the ISWC (pp. 121-124)

Gezira of Eng. & applied Sci. 11-(1)1-10 (2016)

مجلة الجزيرة للهندسة والعلوم التطبيقية المجلد الحادي عشر، العدد الثاني 1437هـ/2016م

الملخص

إطار وصف الموارد(RDF) هو واحد من مواصفات اتحاد شبكة الويب العالمية (W3C)، التي يعتبر الويب الدلالي جزء من أنشطتها الأساسية. تقوم فكرة الـRDF الرئيسية على جعل العبارات والجمل الموجودة في موارد الانترنت تأخذ التعبير (subject–predicate–object). تعتبر قواعد البيانات العلائقية المصدر الرئيس للبيانات والمعلومات الموجودة على الويب و لإعطاء هذه المعلومات الصفة الدلالية لابد من تحويل هذه البيانات إلى تعبيرات الـRDF. الهدف من هذه الدراسة هو تصميم أداة تقوم بتحويل قواعد البيانات العلائقية لـRDF اعتماداً على طريقة التحويل المباشرة (Direct Mapping Method). هذه الأداة يمكنها التعبير عن البيانات الفوقية لقواعد البيانات العلائقية بأقل قيود و مرونة علاوةً على كونها ذات معنى و بالتالي فإنه يمكن تبادل ودمج بيانات الويب دون فقدان لمعانيها الدلالية.