PressAcademia Procedia - (PAP), ISSN: 2459-0762

PressAcademia Procedia

2nd World Conference on Technology, Innovation and Entrepreneurship

May 12- 14, 2017, Istanbul, Turkey. Edited by Sefer Şener

EXPANSION OF A RELATIONAL DATABASE TO SUPPORT SEMANTIC WEB QUERIES

DOI: 10.17261/Pressacademia.2017.594
PAP-WCTIE-V.5-2017(32)-p.234-239

Ylber Januzaj[1], Samedin Krrabaj[2], Artan Luma[3], Arben Hajra[4], Besnik Selimi[5]
[1]University of Mitrovica. ylber.januzaj@umib.net
[2]University of Prizren. . samedin.krrabaj@uni-prizren.com
[3]South East European University. b.selimi@seeu.edu.mk

ABSTRACT

Nowadays various types of data, especially those with public character, are stored and represented in relational databases. While this way of storing data is a practice for most of the institutions, it turns them into isolated silos with low level of accessibility and interoperability in the web. Indeed not all datasets are readable from the World Wide Web, therefore to increase their access it is necessary to provide mapping from relational databases to any serialization format of Resource Description Framework (RDF), as one of the best practices for distributing and interlinking data on the web. While there are several ways to accomplish the mapping process, the most appropriate and fastest way is by using the Protégé Plugins application. In this paper we present this mapping method, and describe the complete conversion of a database from SQL Server relational Database to RDF using different scenarios. By using Sesame Repository, as the local repository for our dataset, several SPARQL queries will be generated on top of our data. As a main contribution this paper emphasizes the advantages and performance of querying and using serialized RDF statements versus those stored in the relational databases.

Keywords: Linked data, semantic technologies, RDF, Protégé.

1. INTRODUCTION

Lately, technology research especially those of the World Wide Web have reached a high point of research and development, however there is still a lot to be discovered. As mentioned in [1],[5] we claim there are a lot of changes not only in the functional way but in the organizational way as well. These changes have occurred following the developments of Semantic Web, but what is more important is that we still consider that the amount of data on the Semantic Web remains collateral, when compared to the traditional Web. Based on all these changes that occurred in the World Wide Web, the need to work with dynamic data in our Web site, arises.[3],[4],[14] Unfortunately, not all records are appropriate to be used in Web sites, so the importance to import data from a relational database to ontologies is frequently required. There are many priorities of RDF over a relational database, and we will mention the most important ones. [11] We all know that life-science is developing rapidly, and there are hundreds of subjects that have developed a data-warehouse with information that needs to be either private or public. If we have a large database company, for instance, which needs to create a large database, they hire a team of programmers which consist of beginner level programmers up to the most advanced ones. All of these programmers need approximately one year to tune the database, and after a while changes will happen, for example schema changes, format changes, and the requests for new data sources will grow up. In RDF no changes will happen, because triples are triples, and there is no need to change any schema, or any tables. The only thing that is required to do, is updating the queries. Second, SPARQL is more stable than SQL since SQL has many dialect changes from one implementation to another. And finally RDF has a better performance in grouped data. Let's imagine a website which responds to a single record, where each record consists of many fields, and after we have a normalized SQL database we will have 40-50 tables with many-to-many relations, and in order to display each page we need to join them. In RDF there is a simple way to call the records. If displaying a record is needed, for instance one in the 30th table, all we need to know in order to display it, is the URI of that record. Exhuming the relational data and making them available as RDF triples is not an unknown field, and there are many negotiability in this area. There are many tools which do the import, from an existing relational database to RDF, but in this paper we will present the automatic mapping way with Protégé Tools from an existing SQL relational database, which uses the ‘ready’ plug-ins which need to be configured to achieve a successful connectivity and import. First, we will show the configuration of Protégé tools, which are necessary to connect to the SQL Server in the distance. The connection will be done using IP addresses. Then, after we have successfully connected, we will do a complete import of tables, including the data that is in them. Next, the automatically created Classes will be detected, along with Slots and Instances. These Slots and Instances will be able to be modified the way we want to, but we should always be careful because later all these Classes, Slots, and Instances will be used to query the data. At the end of the paper we will show some queries which will be implemented on Sesame server, where as an endpoint, will be our local machine, and after this point we consider that they are able to query from the Web sites and PHP programmers.

2. INITIAL STATE (MAPPING)

In [13],[15] is emphasized that obstacles of Semantic Web applicants are non-semantic information which are required for transformation in Semantic Web language (e.g., OWL, RDF). The goal of our project is to ensure that we will cover all OWL constructs, and support all OWL entities such as Classes, Properties and Individuals.

2.1. Ontologies Describing the Database Schema

Ontology is a language which serves to subscribe semantic web data, or also known as the vocabulary of semantic web. In our project, after a conversation we have ontologies which system reserves for us. We will nominate them in vocabulary, which is globally known for easy access, like Dublin Core (DC)[1], Friend of a Fried (FOAF)[2], The Bibliographic Ontology Specification (BIBO)[3], etc., for having an easy interoperability with other resources. In this context, the possibility to enrich them with other data from different repositories will be considered through ontology alignment and also by creating ongoing links by different services like sameas.org[4]. The database schema is described through ontologies in different ways, depending on the form that it is required and on the form that the application work.[1],[4],[7] It is important because in some cases it will be required to import all the database with the possibility to make “live” changes on it, in some other cases it will be required to import the database not in “live” form, because if we try to make changes in our database, then we should update the importation because we cannot see the changes that occur in database. In the next part, the advantages and benefits of an RDF repository will be described through the conversion of a bookstore database. There are thousands of records saved on a relational database, and we will see the delay of the importance. We will also see the OWL ontologies and the frame ontologies that will describe our database.

2.2. Using Protégé Plugin

Protégé is developed by Stanford Center for Biomedical Information Research at the Stanford University School of Medicine, and it is an open source project. [2],[6] There are many developed versions of Protégé but the most appropriate version to perform for this paper is 3.4.4. As we mentioned in the previous sections there are many tools that enable us to convert from relational database to RDF, as D2RQ[5], Triplify[6], etc. Our choice is Protégé because its comprehensive features, functionalities and a well-designed GUI (Graphical User Interface).

2.2.1. First Steps

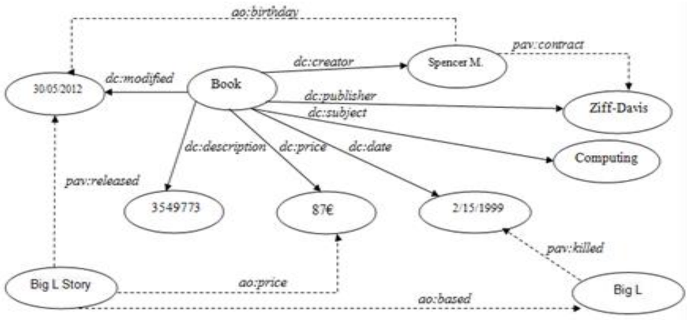
The goal is to convert the data from relational database to RDF in a way to be published and used as open data. For this purpose we want to share something that is on both sides’ interest, citizens and the provider. There is chosen a Bookstore database, from a small Bookstore in our city. The database contains several entities, in different relationships, like Authors, Publishers, Subjects, and Titles. We will focus on a few of them and make these information public to the citizens, and at the same time the Bookstore employees will have the opportunity to extend the data at any time. Having such data bring several advantages to people (readers, buyers), more up to data information, interoperability with other repositories, enriching the data with various other data, etc.

2.2.2. Dealing with Ontology

In this section we will see the Ontology which will be created after we make a complete con-version from the relational database to the RDF.

[1] http://dublincore.org/documents/dces/
[2] http://xmlns.com/foaf/spec/
[3] http://bibliontology.com/
[4] http://sameas.org/
[5] http://d2rq.org/
[6] http://triplify.org/

Figure 1: Sketch of Books Ontology



In [12] is emphasized that an important component in this context is the definition of Classes, Properties and Individuals. In the following section we will show the importance of these components:

- *Classes* tab – building class hierarchy

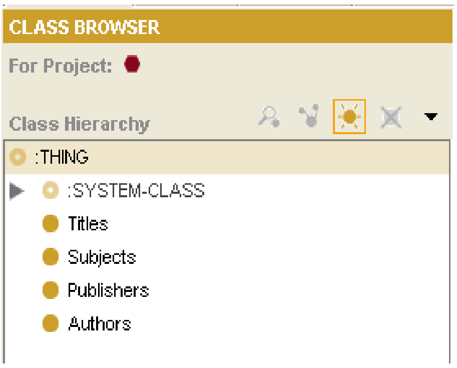
- *Properties* tab – creating object properties, data properties and assign domains

- *Individuals* tab – creating individuals

2.2.3. Classes

Resources which are separated in groups and are in RDF are called Classes. In our project we will see these classes which in relational database were known as entities. As it is mentioned in [6], after a successful conversion from the relational database to the RDF statements, we need to define the classes. These classes are created automatically but we can manipulate with those classes for having specific customization, by choosing the adequate class we want to work on. In Figure 2 is showed a print-screen sample of the classes created automatically by Protégé. This is our imported database ontology which has owl:Thing as parent of all classes.

Figure 2: Class Browser

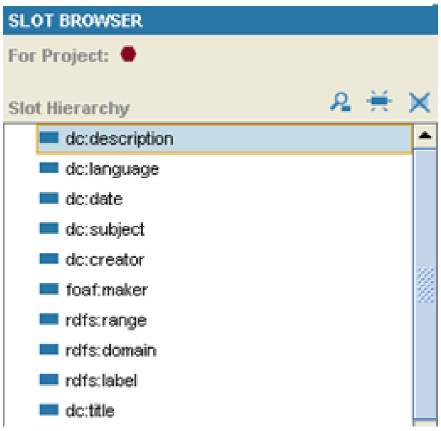


2.2.4. Properties

A property is nothing else but a relation between the subject and an object, which needs to be related by a property to enables us to see the necessary information. By Protégé, both types of Properties, the object and data properties can be manipulated in the same way. Also, below you can see a print-screen example of the properties created also automatically. We should mention that these properties in the relational database were attributes, for example the attribute Title from the table Titles, is being displayed as Titles. In the Figure 3 have a list of these properties. By default, during the conversation process, the properties are automatically created from the table attributes, and so they get the same name.

However, in section 2.1 is mentioned that the best practice is the implementation of well know ontologies and vocabularies. For this purpose there is applied Dublin Core in *dc:description, dc:date, dc:creator*, etc. instead of the attributed names that came from the table.

Figure 3. Slot Browser with Properties



SLOT BROWSER

For Project: ●

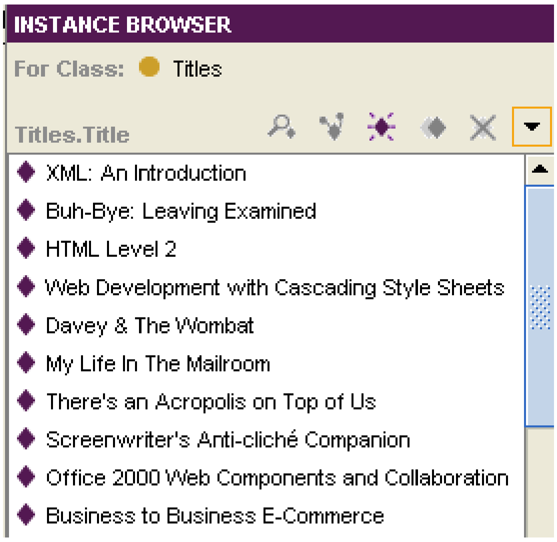
Slot Hierarchy

- dc:description
- dc:language
- dc:date
- dc:subject
- dc:creator
- foaf:maker
- rdfs:range
- rdfs:domain
- rdfs:label
- dc:title

2.2.5. Individuals

Unlike the class and property, individuals is the exact place where the information is stored. The relational database after the conversion to the RDF, will be totally converted with the records on it. These records on the database, after the conversion, are known as Individuals and they are connected to the Properties. [6] As default they will be displayed as Instance_Nr, where dhe Nr will refer to the number of the records, and in the configuration of the individual we can choose to display in the form that we want. For example for the Class Title, we can choose to display the individuals as Title, PubDate, ISBN, Info or whatever way they are described. We can also modify them, delete, rename, moreover we can insert a new instance as a new individual, and we can query it like it was converted from the relational database. Figure 4 will show an example of the Individuals after converted by the Protégé – OWL.

Figure 4. Instance Browser including Converted instances



INSTANCE BROWSER

For Class: ● Titles

Titles.Title

- XML: An Introduction
- Buh-Bye: Leaving Examined
- HTML Level 2
- Web Development with Cascading Style Sheets
- Davey & The Wombat
- My Life In The Mailroom
- There's an Acropolis on Top of Us
- Screenwriter's Anti-cliché Companion
- Office 2000 Web Components and Collaboration
- Business to Business E-Commerce

2.2.5. Reasoner Usage

After the ontology creation, must be sure that if this ontology is consequent or not by using a reasoner. Protégé – OWL is distributed with a reasoner called HermiT which needs to be configured and installed as a plug-in. So, after the check procedure, if there is no any error then we can conclude that the ontology is consequent. We can explain this error with an example: consider we have the Titles.PubDate property value, and if we try to put string values on it, then after we make a reasoner check we will see that we have an error. The error will say ontology is inconsequent, and this is because in the Titles.PubDate we were required to put data type values and not string values.

3. STORING RDF TRIPLES

After a successful conversion, the serialized data need to be stored in an RDF Repository. At the moment there is a relatively large list of choices for this purpose, where each of them has its advantages and disadvantages. Among these, the most know tools are AllegroGraph[7], OpenLink Virtuoso[8], Sesame[9] , 4store[10] , etc. For some practical reasons, in this project our choice has been Sesame, which includes parsing, storing, inferencing and querying such data. Thus, after running the Sesame, locally or in a server, we are ready to upload our data, as .N3 format.[8],[9],[10] In this way a successful upload guarantee that our data are stored and ready for consuming by other people. For this purpose we are offering a SPARQL endpoint as one of the best practices for consuming the data.

3.1. SPARQL Endpoint

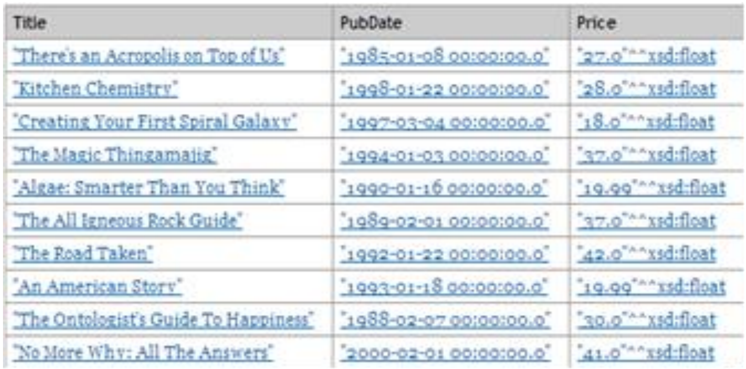
In the following section we will show the possibility of retrieving data from our repository by querying the SPARQL endpoint deployed at the Sesame server. The current link where the datastore is saved locale looks like this: http://localhost:8080/openrdf-workbench/repositories/bookstore. In the Listing 1 is showed an example for illustrating the retrieved data with a simple SPARQL query. So, the Title, Published Year and the Price for all the books will be listed.

Listing 1.

```
SELECT ?Title ?PubDate ?Price
WHERE
{
?p dc:title ?Title.
?p dc:date ?PubDate.
?p rdfs:domain ?Price.
}
 LIMIT 10
```

The retrieved data of the query from the Listing1 are showed in the *Figure 5*.

Figure 5: Table after SPARQL Query Execution



Title	PubDate	Price
"There's an Acropolis on Top of Us"	"1985-01-08 00:00:00.0"	"27.0"^^xsd:float
"Kitchen Chemistry"	"1998-01-22 00:00:00.0"	"28.0"^^xsd:float
"Creating Your First Spiral Galaxy"	"1997-03-04 00:00:00.0"	"18.0"^^xsd:float
"The Magic Thingamajig"	"1994-01-03 00:00:00.0"	"37.0"^^xsd:float
"Algae: Smarter Than You Think"	"1990-01-16 00:00:00.0"	"19.99"^^xsd:float
"The All Igneous Rock Guide"	"1989-02-01 00:00:00.0"	"37.0"^^xsd:float
"The Road Taken"	"1992-01-22 00:00:00.0"	"42.0"^^xsd:float
"An American Story"	"1993-01-18 00:00:00.0"	"19.99"^^xsd:float
"The Ontologist's Guide To Happiness"	"1988-02-07 00:00:00.0"	"30.0"^^xsd:float
"No More Why: All The Answers"	"2000-02-01 00:00:00.0"	"41.0"^^xsd:float

Thus, everyone can perform various SPARQL queries for getting any kind of information. Everything is accessible without any special hardware or software requirements. The data can be consumed, integrated and visualized everywhere and for any purpose.

[7] http://franz.com/agraph/allegrograph/
[8] http://virtuoso.openlinksw.com/
[9] http://openrdf.org/
[10] http://4store.org/

4. CONCLUSION

In our paper we presented the advantages of serialized RDF statements versus Relational Databases using Data Master Protégé Plug-in, which allows us to convert the database from relational model to the RDF statements by using the Java mechanisms. All this import is done by using the JDBC driver which creates a kind of a bridge between the Protégé application and the SQL Server. We presented the ontologies, manipulation with them and a consummation through the SPARQL endpoint deployed in the SESAME Server. Offering the data as Linked Open data, especially by applying a SPARQL endpoint, bring several benefits to the readers, buyers and to the bookstores also. At any time these data can be used and visualized by anyone for any purpose. In addition, the data will be enriched with other data stored in different repositories.

REFERENCES

[1] Csongor Nyulas, Martin O'Connor and Samson Tu, "DataMaster – a Plug-in for Importing Schemas and Data from Relational Databases into Protégé," in Stanford Medical Informatics, Stanford University School of Medicine, Stanford, CA 94305, 2007.

[2] Farid Cerbah, "Learning Highly Structured Semantic Repositories from Relational Databases The RDBToOnto Tool" in Dassault Aviation, DPR/ESA, 78 quai Marcel Dassault 92552 Saint-Cloud , France, 2008.

[3] Li Ma, Jing Mei, Yue Pan, Krishna Kullkarni, Achille Fokue, Anand Ranganathan, "Semantic Web Technol-ogies and Data Management" in IBM China Research Laboratory, IBM Software Group, Ibm Watson Research Center, Bei Jing 100094, China San Jose, CA, 2012.

[4] Juan F. Sequeda, Syed Hamid Tirmizi and Daniel P. Miranker, "SQL Databases are a Moving Target Position Paper for W3C Workshop on RDF Access to Relational Databases" in The University of Texas at Aus-tin, Department of Computer Sciences, Austin, USA, 2007.

[5] Simeon Polfliet, Ryutaro Ichise, "Automated Mapping Generation for Converting Databases into Linked Data" in Ensimag engineering school Grenoble Institute of Technology (INPG) Grenoble, France, Principles of Informatics Research Division National Institute of Informatics, Tokyo, Japan, 2010 .

[6] Adam Jurcik,"Development of Visualization Plug-in for Protege" in Masarykova Univerzita Fakulta In-formatiky, Brno, 2010.

[7] Juan F. Sequeda, Marcelo Arenas and Daniel P. Miranker, "On Directly Mapping Relational Databases to RDF and OWL" in The University of Texas at Austin, PUC Chile, Lyon, France, 2012.

[8] Rafael Valencia-García, Francisco García-Sánchez, Dagoberto Castellanos-Nieves, and Jesualdo Tomás Fernández-Breis, " OWLPath: An OWL Ontology-Guided Query Editor" in Ieee Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, 2011.

[9] A. Bernstein and E. Kaufmann, "Gino—A guided input natural language ontology editor," in Proc. Int. Se-mantic Web Conf., vol. 4273, Lecture Notes in Computer Science, I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, Eds.pp. 144–157, UK, 2006.

[10] H. Namgoong and H.-G. Kim, "Ontology-based controlled natural language editor using CFG with lexical dependency" in Proc. ISWC/ASWC, vol. 4825, Lecture Notes in Computer Science, K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds., pp. 353–366, Korea, 2007.

[11] H.-G. Kim, B.-H. Ha, J.-I. Lee, and M.-K. Kim, "A multi-layered application for the gross description using semantic web technology" in Int. J. Med. Inf., vol. 74, no. 5, pp. 399–407, Hindawi Publishing Corporation, 2005.

[12] Lei ZHANG, Jing LI, "Automatic Generation of Ontology Based on Database" in School of Computer Sci-ence and Technology, China University of Mining and Technology, Xuzhou 22111, Jiangsu Province, China, 2012.

[13] Rai Ghawi, Nadine Cullot, "Database-to-Ontology Mapping Generation for Semantic Interoperability" in Laboratory LE2I, University of Burgundy, Dijon France, 2007.

[14] Nikolaos Konstantinou, Dimitrios-Emmanuel Spanos, Michael Chalas, Emmanuel Solidakis and Nikolas Mitrou," VisAVis: An Approach to an Intermediate Layer between Ontologies and Relational Database Con-tents" in National Technical University of Athens, Division of Communications, Electronics and Information Systems, Heroon Polytechniou, Athens, Greece, 2006.

[15] Martin J. O'Connor, Christian Halaschek-Wiener, Mark A. Musen, "Mapping Master: a Flexible Approach for Mapping Spreadsheets to OWL" in Stanford Center for Biomedical Informatics Research Stanford, USA, 2010.