Integrating Data from Relational Databases Using RDF: Position Paper for W3C Workshop on RDF Access to Relational Databases

Author: Ashok Malhotra, Oracle Corporation

Introduction

We all know that huge amounts of invaluable data are stored in Relational databases. Many of us feel that we should be able to do a much better job of extracting useful knowledge from this data than we are able to do today. There are two basic problems:

- Relational databases are relatively flat with little semantic information. Due to normalization, data is often fragmented and although Key and Foreign Key relationships are indicated in the catalog, these are merely hints about the semantics of the data..
- Databases are often insular, built for a single purpose with access restricted to a particular group whereas insight often comes from combining data from several databases across organizational boundaries such as a company and the databases of its suppliers.

The approach we recommend to solving this problem is to put a semantic cover over the several Relational databases relevant to a particular problem or situation. This would allow users to write queries to extract information that is more semantically meaningful from across several databases.. The semantic cover would, essentially, be an ontology that may be represented in OWL[1], with links to the sources of the data in the different databases. Writing such a semantic cover is not simple. Apart from the fact that database semantics are often written on bits of paper or encoded in table and column names, we need to define the relationships between the data in the different databases. These relationships can take many forms:

1) Corresponding or analogous data in one database might be represented in a different number of tables in another database.

2) Corresponding or analogous data in one database might be represented in a different number of columns in another database.

3) Corresponding or analogous data in one database might be represented in a different number of rows in another database.

4) Corresponding or analogous data in one database might be represented using different values in another database (e.g., one database uses "m" and "f" for "male" and "female", while the other might use "0" and "1", respectively), or the same values in the various

databases might mean different things (e.g., one database uses "1" and "0" for "male" and "female", while the other might use "0" and "1", respectively), or the values in the various databases might be incompatible in ways that make the mapping very difficult (e.g., one database uses "0" and "1" for "male" and "female", respectively, which the other database uses values from "0" through "9" to represent medically-significant variations of human sexes, such as types of hermaphrodites).

Other relationships are possible. The above are merely some examples.

After the relationships in the relational databases have been understood, the ontology, including the classes and subclasses in the semantic cover, can be defined. A single class that is built from information contained in several databases maps to an SQL[4] query on each of the databases as well as an integrating query to gather together all the information for that class. A common case is where the data for a single class is extracted from several databases and then joined together using property columns.

Thus, each class or subclass in the semantic cover is mapped to one or more SQL queries on the underlying databases. These queries may be linked queries in that the results from one query may be used as parameters to the next query. Also, data from several queries may be combined to provide the data for a single class or subclass.

After this step, queries can be written against the semantic cover ontology using SPARQL[3], the query language for RDF[2]. Using the information discussed above, the SPARQL queries can be mechanically translated into SQL queries on the underlying databases and the results obtained from the various databases integrated to yield the final result. The reason for translating the SPARQL queries into SQL is to take advantage of existing Relational database engines, which are highly optimized and extremely efficient at extracting data from Relational databases. It also takes advantage of existing SQL language skills and is the shortest route to achieving the vision outlined in this paper. Unfortunately, we do not have space in this position paper to illustrate the translation of SPARQL to SQL. This will be the subject of a longer paper at a future conference. .

Summary

- First, the Relational databases are selected.
- Next, an ontology or semantic cover is designed for the combined databases.
- Each class or subclass in this ontology is associated with one or more (possibly linked) SQL queries over the databases perhaps followed by queries to integrate the information obtained from .the earlier queries.
- SPARQL queries on the semantic cover are translated into several SQL queries over the constituent databases.

References

[1] OWL: ***OWL Web Ontology Language Reference***, Mike Dean and Guus Schreiber, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/ .

[2] RDF: *RDF/XML Syntax Specification (revised)*, Dave Beckett and Brian McBride, Editors W3C Recommendation, 10 February 2004**,** See http://www.w3.org/RDF/

[3] SPARQL: *SPARQL Query Language for RDF*, *W3C Candidate Recommendation 14 June 2007.* See http://www.w3.org/TR/rdf-sparql-query/

[4] SQL: ISO (International Organization for Standardization). *ISO/IEC 9075-2:1999, Information technology --- Database languages --- SQL --- Part 2: Foundation (SQL/Foundation).* [Geneva]: International Organization for Standardization, 1999. See http://www.iso.org/iso/en/ISOOnline.frontpage