

Descripción del escenario y consultas SeRQL

Supongamos un escenario en el que disponemos de tres bases de datos distribuidas en la Web. Cada base de datos posee una de las tablas que se muestran en la Figura expuesta a continuación.

Supongamos que un agente inteligente recoge la información de estas tablas y la vuelca en un repositorio RDF de Sesame para su posterior manipulación de forma combinada.

La base para poder combinar estos datos está en que los códigos de aeropuertos y compañías aéreas son globales y, por tanto, se podrían tomar fácilmente como recursos para crear un grafo RDF global que enlace las tres tablas.

Suponiendo este escenario, y teniendo en cuenta que los datos ya han sido previamente volcados en Sesame, se realizarán una serie de consultas SeRQL.

Vuelo						
numVuelo	companya	origen	destino	escala	fecha	hora

Aeropuerto			
codigoAeropuerto	nombreAeropuerto	ciudad	pais

CompanyaAerea	
codigoCompanya	nombreCompanya

- 1) **Consulta que devuelve el sujeto y valor de todas las triples con la propiedad o:numVuelo.**

>consulta

using namespace

o = <http://www.avanthotel.com/ontoturismo#>

select X, Y

from {X} o:numVuelo {Y}

>resultados

X

Y

<http://www.avanthotel.com/vuelos#1> "IB1727"

<http://www.avanthotel.com/vuelos#2> "UX6076"

<http://www.avanthotel.com/vuelos#3> "JK5772"

<http://www.avanthotel.com/vuelos#4> "IB1749"

<http://www.avanthotel.com/vuelos#5> "UX6024"

5 results found in 3 ms.

2) **Consulta que devuelve todos los sujetos de la clase o:Vuelo.**

```
>consulta
using namespace
  o = <http://www.avanhotel.com/ontoturismo#>

select X
from {X} rdf:type {o:Vuelo}

>resultados
X
http://www.avanhotel.com/vuelos#1
http://www.avanhotel.com/vuelos#2
http://www.avanhotel.com/vuelos#3
http://www.avanhotel.com/vuelos#4
http://www.avanhotel.com/vuelos#5
5 results found in 12 ms.
```

3) **Se extraen todos los registros de la clase o:CompanyiaAerea con todas las propiedades que posee esta clase.**

```
>consulta
using namespace
  o = <http://www.avanhotel.com/ontoturismo#>

select X,Y,Z
from
{X} rdf:type {o:CompanyiaAerea};
o:codigoCompanyia {Y};
o:nombreCompanyia {Z}

>resultados
X                                     Y   Z
http://www.avanhotel.com/companyias#1 "IB" "Iberia"
http://www.avanhotel.com/companyias#2 "UX" "Air Europa"
http://www.avanhotel.com/companyias#3 "JK" "Spanair"
3 results found in 25 ms.
```

4) **Se extraen todas las propiedades del registro de la clase o:CompanyiaAerea cuyo código es igual a _IB_.**

```
>consulta
using namespace
  o = <http://www.avanhotel.com/ontoturismo#>

select X,Y,Z
from
{X} rdf:type {o:CompanyiaAerea};
o:codigoCompanyia {Y};
o:nombreCompanyia {Z}
where Y like "IB"
```

>resultados

X

Y Z

<http://www.avanthotel.com/companias#1> "IB" "Iberia"

1 results found in 7 ms.

- 5) Supongamos que el agente es capaz de entender que el significado de la columna 'companyia' de la tabla Vuelo es equivalente al de la columna 'codigoCompanyia' de la tabla CompanyiaAerea. El agente es capaz entonces de combinar ambas tablas realizando un 'mapping' semántico entre ambas columnas mediante una consulta.

Se realizará dicha consulta, y se extraerá todos los números de vuelo (numVuelo) de la clase o:Vuelo seguidos del nombre de la compañía aérea (o:nombreCompanyia) que opera el vuelo.

>consulta

using namespace

o = <<http://www.avanthotel.com/ontoturismo#>>

select NV, NCA

from

{X} rdf:type {o:Vuelo};

o:numVuelo {NV};

o:companyia {CA},

{Y} rdf:type {o:CompanyiaAerea};

o:codigoCompanyia {CCA};

o:nombreCompanyia {NCA}

where CA=CCA

>resultados

NV NCA

["IB1727"](#) ["Iberia"](#)

["IB1749"](#) ["Iberia"](#)

["UX6076"](#) ["Air Europa"](#)

["UX6024"](#) ["Air Europa"](#)

["JK5772"](#) ["Spanair"](#)

5 results found in 9 ms.

Javier Fernández Rivera

www.aurea.es