

En este laboratorio aprenderemos a usar R2RML, un lenguaje declarativo que permite traducir entre bases de datos relacionales y datos RDF. R2RML puede ser usado en varias formas, algunas de las cuales son: transformar archivos CSV a RDF, hacer una copia de una base relacional a una a RDF, o crear un endpoint SPARQL que permita consultar una base de datos relacional, traduciendo de manera transparente las consultas SPARQL a consultas SQL. En este laboratorio vamos a probar varias de estas funcionalidades.

Para este laboratorio deberás conectarte a servidor n14.degu.cl mediante SSH. Tu nombre de usuario es tu nombre guión tu apellido y tu contraseña es tu nombre (cámbiala la primera vez que entres). Al entrar encontrarás dos carpetas, una (ontop) con el programa llamado Ontop, que usaremos en el laboratorio y la otra (lab10) con algunos ejemplos de archivos de mapeo y consultas.

Traduciendo los datos a RDF Los datos que están en la base de datos corresponden a los resultados de las elecciones presidenciales en Estados Unidos. Hay tres tablas:

- **candidates:** contiene los candidatos de las elecciones. Cada candidato tiene un nombre (**name**) y algunos tienen un partido al que representan (**party**).
- **state:** contiene los estados de Estados Unidos. Cada condado tiene un nombre (**name**), el número de votos electorales (**electoral_votes**), el momento en el que se cerró la urna (**poll_closed_at**) y el número de candidatos (**number_of_candidates**).
- **votos:** Cada voto tiene el candidato por el cual se votó (**candidate**), el condado (**county**), el estado (**state**) y el número de votos (**number_of_votes**). Es necesario indicar que hay condados con el mismo nombre, por ejemplo hay ocho condados llamados Butler. El nombre del condado es único por cada estado.

Si quieres puedes inspeccionar con mayor detalle los datos conectándote con la consola de SQL, con el nombre de usuario, base de datos y contraseña “webdatos”. Es decir, con el comando:

```
psql -U webdatos -h localhost
```

Para comenzar vamos a hacer una traducción directa. Para ello utilizaremos el archivo mapping-0.ttl en el directorio lab10. Para ejecutar la traducción utilizamos el siguiente comando dentro de la carpeta ontop:

```
./ontop materialize --driver-class postgresql \  
    --disable-reasoning \  
    --url jdbc:postgresql:webdatos \  
    --username webdatos \  
    --password webdatos \  
    --mapping ../lab10/mapping-0.ttl \  
    --output ../datos-0.ttl \  
    --format turtle
```

Esta consulta va a generar el archivo datos.ttl en tu carpeta personal con el resultado de convertir todos los datos a RDF usando un mapeo directo. Es decir, por cada fila de una tabla habrá un recurso que tendrá un atributo por cada uno de las columnas no nulas de la tabla.

Consultando los datos con SPARQL Ahora haremos el ejercicio de consultar la base de datos relacional usando SPARQL. Para ello, en vez de usar la opción materialize del comando ontop, usaremos la opción query. En archivo query-1.sparql de la carpeta lab10 tiene una consulta que funciona para el mapping-0.ttl que hemos probado anteriormente. Para ejecutar esta consulta prueba con:

```
./ontop query --driver-class postgresql \  
    --disable-reasoning \  
    --url jdbc:postgresql:webdatos \  
    --query query-1.sparql
```

```
--mapping ../mapping-0.ttl \
--username webdatos \
--password webdatos \
--query ../lab10/query-0.sparql \
--output result-0.csv
```

La consulta busca las primeras diez filas del resultado de hacer un join natural entre las tres tablas, ordenadas por los nombres del estado, del condado, y del partido (considerando sólo los candidatos que tienen partido). Nota: al parecer Ontop tiene problemas al hacer consultas que requieran usar optional. Por eso estamos seleccionando sólo aquellos candidatos que tienen partido.

Ejercicios

1. En la traducción entregada la representación de las votaciones obtenidas por D. Trump en Autauga County, Alabama son:

```
<http://ex.org/votes/D.%20Trump/Alabama/Autauga%20County>
  a <http://ex.org/Votes> ;
  ex:candidate "D. Trump" ;
  ex:county "Autauga County" ;
  ex:state "Alabama" ;
  ex:number_of_votes "18110" .
```

Modifica la traducción para cambiar los nombres del candidato y del estado por sus respectivas IRIs. Es decir, los datos deberían ser:

```
<http://ex.org/votes/D.%20Trump/Alabama/Autauga%20County>
  a <http://ex.org/Votes> ;
  ex:candidate <http://ex.org/candidate/D.%20Trump> ;
  ex:county "Autauga County" ;
  ex:state <http://ex.org/state/Alabama> ;
  ex:number_of_votes "18110" .
```

Además, modifica la consulta SPARQL para que entregue los mismos resultados que entregaba antes.

2. Modifica la traducción para que todos los tipos numéricos tengan el tipo de datos apropiado. Por ejemplo, en número de votos debiera ser "18110"^^xsd:integer en vez de "18110".
3. Los votos no están normalizados, porque existe una dependencia funcional entre los condados y los estados. Modifica la traducción para normalizar los votos, de modo que el resultado sea:

```
<http://ex.org/votes/D.%20Trump/Alabama/Autauga%20County>
  a <http://ex.org/Votes> ;
  ex:candidate <http://ex.org/candidate/D.%20Trump> ;
  ex:county <http://ex.org/county/Alabama/Autauga%20County> ;
  ex:number_of_votes "18110"^^xsd:integer .
```

```
<http://ex.org/county/Alabama/Autauga%20County>
  a <http://ex.org/County> ;
  ex:state <http://ex.org/state/Alabama> .
```

Escribe una consulta que obtenga los mismos resultados que las anteriores sobre este nuevo esquema.

4. Escribe una consulta que calcule los votos totales de D. Trump y H. Clinton, sumando todos los condados.

Entrega: Debes entregar en U-Cursos un archivo `mapping-i.ttl` y un archivo `query-i.sparql` para cada ejercicio *i* de este laboratorio.