

Práctica 4: Introducción a Apache Jena y SPARQL

Para reducir este documento se adjuntan los resultados obtenidos en ficheros json del formato *ej<num_ejercicio>.json*.

Ejercicio 1: Obtener las empresas de la industria de las películas ubicadas en los Estados Unidos.

```
prefix dbo:<http://dbpedia.org/ontology/>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select distinct ?company
where
{
  ?company rdf:type dbo:Company.
  {
    {
      ?company dbo:industry dbr:Film_industry.
    } UNION
    {
      ?company dbo:industry dbr:Film.
    }
  }
  {
    {
      ?company dbo:locationCountry dbr:United_States_of_America.
    } UNION
    {
      ?company dbo:locationCountry dbr:United_States.
    }
  }
  {
    ?company dbo:location ?location.
    { ?location dbo:country dbr:United_States. } UNION
    { ?location dbo:country dbr:United_States_of_America. }
  }
}
```

Explicación

- Primero se obtienen todos los objetos de clase *Company* (**rdf:type dbo:Company**).
- Se filtra por industria filmográfica (en este grupo también tanto la clase principal **dbr:Film_industry** como la clase **dbr:Film_industry** para que empresas como *Lucasfilm* o *Walt Disney* aparezcan)
- Por último se filtra por país (hay que tener cuidado debido a que existe tanto **dbr:United_States_of_America** como **dbr:United_States**).

IMPORTANTE: muchas empresas no tienen cubierto el campo **locationCountry**, por lo que es necesario buscar la ciudad o área (**location**) y comprobar que pertenece a Estados Unidos.

La consulta simple sería la siguiente:

```
prefix dbo: <http://dbpedia.org/ontology/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select distinct ?company
where
{
    ?company rdf:type dbo:Company;
              dbo:industry dbr:Film_industry;
              dbo:locationCountry dbr:United_States.
}
```

La consulta sigue un poco lo comentado anteriormente: buscamos los objetos de clase *Company*, filtramos por el campo *industry* para quedarnos con las compañías filmográficas y filtramos las que sean estadounidenses gracias al campo *locationCountry*.

El resultado de esta consulta es el archivo *ej1_simple.json*.

Ejercicio 2: Obtener, de las empresas anteriores, aquellas cuyo fundador sea un actor.

```
prefix dbo: <http://dbpedia.org/ontology/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select distinct ?company
where
{
    ?actor rdf:type dbo:Actor;
           foaf:gender "male"@en.

    ?company rdf:type dbo:Company;
              dbo:foundedBy ?actor.

    {
        {
            ?company dbo:industry dbr:Film_industry.
        } UNION
        {
            ?company dbo:industry dbr:Film.
        }
    }

    {
        {
            ?company dbo:locationCountry dbr:United_States_of_America.
        } UNION
        {
            ?company dbo:locationCountry dbr:United_States.
        } UNION
        {
            ?company dbo:location ?location.
            { ?location dbo:country dbr:United_States. } UNION
            { ?location dbo:country dbr:United_States_of_America. }
        }
    }
}
```

Explicación

- Partimos de la consulta anterior.
- Buscamos todos las personas que sean actores.
- Filtramos las compañías que fueran fundadas por alguno de ellos (**dbo:foundedBy ?actor.**)

Si nos basamos en la consulta simple del ejercicio 1:

```
prefix dbo:<http://dbpedia.org/ontology/>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select distinct ?company
where
{
    ?actor  rdf:type dbo:Actor;
           foaf:gender  "male"@en.

    ?company rdf:type dbo:Company;
             dbo:industry dbr:Film_industry;
             dbo:locationCountry dbr:United_States;
             dbo:foundedBy ?actor.
}
```

Las modificaciones sufridas son las mismas que en el caso de la consulta compleja: se buscan a todos los actores (elementos que tenga como *rdf:type* el *dbo:Company*) y se añade una nueva tripleta para filtrar los elementos de tipo compañía que tenga como fundador (*foundedBy*) al actor.

No se han obtenido resultados.

Ejercicio 3: Obtener el nombre de los fundadores de compañías de la industria aeroespacial que no sean estadounidenses.

```
prefix dbo:<http://dbpedia.org/ontology/>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?founder, ?company
where
{
    ?company rdf:type dbo:Company;
             dbo:foundedBy ?founder.
    { ?company dbo:industry dbr:Aerospace_industry } UNION
    { ?company dbo:industry dbr:Aerospace_manufacturer }

    FILTER ( !EXISTS{
        ?company dbo:locationCountry dbr:United_States_of_America.
    })
    FILTER ( !EXISTS{
        ?company dbo:locationCountry dbr:United_States.
    })
    FILTER ( !EXISTS{
        ?company dbo:location ?location.
        { ?location dbo:country dbr:United_States. } UNION
        { ?location dbo:country dbr:United_States_of_America. }
    })
    FILTER ( !EXISTS{
        ?company dbo:locationCity ?locationCity.
        { ?locationCity dbo:country dbr:United_States. } UNION
        { ?locationCity dbo:country dbr:United_States_of_America. }
    })
}
```

Esta consulta es muy parecida a la primera.

Explicación

- Se filtran las compañías por industria (campo **industry**) aerospacial: **dbo:Aerospace_industry** y **dbo:Aerospace_manufacturer** (para que aparezcan empresas como Boeing).
- Obtenemos el elemento correspondiente a *dbo:foundedBy* (almacenamos el valor en una variable *?founder*).
- Para buscar compañías no estadounidenses se cogen los mismos filtros que en los ejercicios anteriores pero negándolos (**FILTER (!EXISTS{ <filtros> })**).

Una consulta más simple “equivalente” sería:

```
prefix dbo:<http://dbpedia.org/ontology/>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select distinct ?founder, ?company
where
{
  ?company rdf:type dbo:Company;
    dbo:foundedBy ?founder;
    dbo:industry dbr:Aerospace_manufacturer.
  MINUS {
    ?company dbo:locationCountry dbr:United_States.
  }
}
```

La base es la misma pero utilizando *MINUS* para buscar las compañías cuyo *dbo:locationCountry* son sea *dbr:United_States*. El resultado de esta consulta es el archivo *ej3_simple.json*.

Ejercicio 4: Obtener las empresas/compañías ubicadas en los Estados Unidos cuya persona clave naciera en una ciudad que tenga menos habitantes que empleados tiene la empresa.

```
prefix dbo:<http://dbpedia.org/ontology/>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xsd:<http://www.w3.org/2001/XMLSchema#>

select distinct ?boss ?birthPlace ?employees ?populationTotal ?company
where
{
  ?company rdf:type dbo:Company;
    dbo:keyPerson ?boss;
    dbo:numberOfEmployees ?employees.
  ?boss dbo:birthPlace ?birthPlace.
  ?birthPlace dbo:populationTotal ?populationTotal
  {
    {
      ?company dbo:locationCountry dbr:United_States_of_America.
    } UNION
    {
      ?company dbo:locationCountry dbr:United_States.
    } UNION
    {
      ?company dbo:location ?location.
      { ?location dbo:country dbr:United_States. } UNION
      { ?location dbo:country dbr:United_States_of_America. }
    }
  }
  FILTER (?populationTotal < ?employees)
}
```

Explicación

- Para realizar el filtro de compañías estadounidenses se realiza el filtro presentado en los ejercicios previos.
- Se obtiene el jefe (**dbo:keyPerson**) y el número de empleados de la compañía **?company** (**dbo:numberOfEmployees**).
- Del objeto “*Person*” se obtiene el lugar de nacimiento (**?birthPlace**).
- Del objeto “*City*” se obtiene la cantidad de población (**dbo:populationTotal**).
- Para finalizar, se filtra la cantidad de población respecto a los empleados (“*menos habitantes que empleados*”)

Para terminar se muestra una consulta más simple “equivalente”:

```
prefix dbo:<http://dbpedia.org/ontology/>
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix xsd:<http://www.w3.org/2001/XMLSchema#>

select distinct ?boss ?birthPlace ?employees ?populationTotal ?company
where
{
  ?company rdf:type dbo:Company;
    dbo:keyPerson ?boss;
    dbo:numberOfEmployees ?employees;
    dbo:locationCountry dbr:United_States.
  ?boss dbo:birthPlace ?birthPlace.
  ?birthPlace dbo:populationTotal ?populationTotal.
  FILTER (?populationTotal < ?employees)
}
```

NOTA

Se realizaron consultas más complejas haciendo uso del *UNION* para poder obtener “todos” los resultados que pedían las consultas. Como se ha comentado con el profesor, muchos elementos de la web semántica no tienen todas las propiedades (hay empresas que no tienen la propiedad *dbo:foundedBy*), utilizan valores diferentes para una propiedad (por ejemplo para el fundador pueden utilizar un literal del tipo “*David Álvarez*” o un elemento del tipo *person* como *dbr:William_Boeing*) o diferentes nombres para un país (*dbr:United_States_of_America* o *dbr:United_States*).