

Sistemas de Información para la Web

Introducción a SPARQL

Omar Teixeira González, UO281847

30/11/2023



Tabla de contenidos

Tabla de contenidos	1
Documentación sobre SPARQL	2
Consulta elegida	2
Resultado esperado	2
Realización de la consulta	3
Versión inicial	3
Versión general con consulta anidad y con <i>timeout</i>	4
Versión general sin consulta anidada y con <i>timeout</i>	5
Versión general, sin <i>timeout</i> pero más “fea”	6

Documentación sobre SPARQL

Consulta elegida

Intérpretes que hayan trabajado en películas americanas, del S. XXI, premiadas y que hayan tenido más de 1.000 millones de ingresos junto con todos los intérpretes con los que hayan trabajado en esas películas ordenados por número de “conocidos” descendente. Pista: necesitas usar la función GROUP_CONCAT.

Resultado esperado

Los siguientes resultados son para las películas de Avatar, Avengers: Endgame y Top Gun: Maverick, a modo de representación de lo que debería de ser la salida obtenida de la consulta.

?actor	?cast
Sam Worthington	Zoe Saldaña, Sigourney Weaver, Michelle Rodríguez, Stephen Lang, Giovanni Ribisi, Joel David Moore, C. C. H. Pounder, Wes Studi, Laz Alonso
Robert Downey Jr	Chris Evans, Chris Hemsworth, Mark Ruffalo, Scarlett Johansson, Jeremy Renner, Don Cheadle, Paul Rudd, Karen Gillan, Bradley Cooper, Brie Larson, Tessa Thompson, Danai Gurira, Benedict Wong, Jon Favreau, Benedict Cumberbatch, Chris Pratt, Zoe Saldaña, Elizabeth Olsen, Tom Holland, Chadwick Boseman, Anthony Mackie, Sebastian Stan, Evangeline Lilly, Letitia Wright, Dave Bautista, Pom Klementieff, Vin Diesel, Gwyneth Paltrow, Josh Brolin
Tom Cruise	Val Kilmer, Miles Teller, Jennifer Connelly, Glen Powell, Lyliana Wray, Ed Harris, Monica Barbaro, Jon Hamm, Charles Parnell

Realización de la consulta

Versión inicial

La siguiente query consiste en la primera versión de la consulta global, realizando las partes correspondientes a la obtención de actores que hayan trabajado en películas americanas del siglo XXI, que hayan ganado premios y que tengan más de 1000 millones de recaudación.

Para realizar esta parte, se ha buscado en WikiData “El Señor de los Anillos: El Retorno del Rey”, ya que cumplía con todos los requisitos, película estadounidense, publicada en el siglo XXI, con premios, y con una recaudación superior a 1000 millones de dólares. A raíz de esto se han ido sacando las restricciones pertinentes de forma que el resultado fuera una lista con todos los actores y actrices que cumplen dichas características.

```
SELECT ?actorLabel
WHERE {
  # P31 -> instancia de
  # P161 -> miembro del reparto
  # P495 -> país de origen
  # P166 -> premio recibido
  # P577 -> fecha de publicación
  # P2142 -> recaudación
  # Q11424 -> película
  # Q30 -> Estados Unidos
  ?movie wdt:P31 wd:Q11424;
    wdt:P161 ?actor;
    wdt:P495 wd:Q30;
    wdt:P166 ?award;
    wdt:P577 ?date;
    wdt:P2142 ?boxOffice.

  FILTER(YEAR(?date) > 2000).
  FILTER(?boxOffice > 1000000000).

  ?actor rdfs:label ?actorLabel.

  FILTER(LANG(?actorLabel)="en").
}
GROUP BY ?actorLabel
```

Versión general con consulta anidada y con *timeout*

Una vez hecho lo anterior se probó a expandir la consulta para cumplir la segunda parte, de forma que tras leer la documentación de WikiData sobre consultas anidadas y sobre el uso de GROUP_CONCAT, se llegó a la siguiente versión, la cual no parecía funcionar debido a un timeout, pero en apariencia parecía cumplir todo lo solicitado.

```
SELECT ?actorLabel (GROUP_CONCAT(?coActorLabel;separator=", ") as ?cast)
WHERE {
  # P31 -> instancia de
  # P161 -> miembro del reparto
  # P495 -> país de origen
  # P166 -> premio recibido
  # P577 -> fecha de publicación
  # P2142 -> recaudación
  # Q11424 -> película
  # Q30 -> Estados Unidos
  ?movie wdt:P31 wd:Q11424;
    wdt:P161 ?actor;
    wdt:P495 wd:Q30;
    wdt:P166 ?award;
    wdt:P577 ?date;
    wdt:P2142 ?boxOffice.

  FILTER(YEAR(?date) > 2000).
  FILTER(?boxOffice > 1000000000).

  ?actor rdfs:label ?actorLabel.

  FILTER(LANG(?actorLabel)="en").

  {
    SELECT ?coActorLabel
    WHERE {
      ?movie wdt:P161 ?coActor;

      FILTER(?coActor != ?actor).

      ?coActor rdfs:label ?coActorLabel.

      FILTER(LANG(?coActorLabel)="en").
    }
  }
}
GROUP BY (?actorLabel)
ORDER BY DESC(COUNT(DISTINCT ?coActor))
```

Versión general sin consulta anidada y con *timeout*

Esta versión supone una forma similar de realizar lo anterior, lo único que no se realiza la consulta anidada, cabe mencionar que había probado con esta versión previamente a la anterior, pero como no obtuve resultados aparentes, pensé que la causa era la necesidad de realizar la consulta anidada (algo que ya he visto que no).

```
SELECT ?actorLabel (GROUP_CONCAT(?coActorLabel;separator=", ") as ?cast)
WHERE {
  # P31 -> instancia de
  # P161 -> miembro del reparto
  # P495 -> país de origen
  # P166 -> premio recibido
  # P577 -> fecha de publicación
  # P2142 -> recaudación
  # Q11424 -> película
  # Q30 -> Estados Unidos
  ?movie wdt:P31 wd:Q11424;
    wdt:P161 ?actor;
    wdt:P161 ?coActor;
    wdt:P495 wd:Q30;
    wdt:P166 ?award;
    wdt:P577 ?date;
    wdt:P2142 ?boxOffice.

  FILTER(YEAR(?date) > 2000).
  FILTER(?boxOffice > 1000000000).

  ?actor rdfs:label ?actorLabel.
  ?coActor rdfs:label ?coActorLabel.

  FILTER(?coActor != ?actor).
  FILTER(LANG(?actorLabel)="en").
  FILTER(LANG(?coActorLabel)="en").
}
GROUP BY (?actorLabel)
ORDER BY DESC(COUNT(DISTINCT ?coActor))
```

Versión general, sin *timeout* pero más “fea”

Por último, esta versión de la consulta devuelve los valores solicitados, si bien no devuelve los valores de las etiquetas, haciendo que los resultados correspondientes sean más feos y menos legibles frente al resultado buscado.

```
SELECT ?actor (GROUP_CONCAT(?coActor;separator=", ") as ?cast)
WHERE {
  # P31 -> instancia de
  # P161 -> miembro del reparto
  # P495 -> país de origen
  # P166 -> premio recibido
  # P577 -> fecha de publicación
  # P2142 -> recaudación
  # Q11424 -> película
  # Q30 -> Estados Unidos
  ?movie wdt:P31 wd:Q11424;
    wdt:P161 ?actor;
    wdt:P161 ?coActor;
    wdt:P495 wd:Q30;
    wdt:P166 ?award;
    wdt:P577 ?date;
    wdt:P2142 ?boxOffice.

  FILTER(YEAR(?date) > 2000).
  FILTER(?boxOffice > 1000000000).

  FILTER(?coActor != ?actor).
}
GROUP BY (?actor)
ORDER BY DESC(COUNT(DISTINCT ?coActor))
```