

Report

In this assignment were implemented basic and more complex queries using SPARQL via the GraphDB environment. Regarding the task 1, the option A was chosen and thus the film ontology was used for the implementation. For the task 2, first the DBPedia was explored and then five queries were constructed from the film ontology provided by DBPedia. Finally, in task 3 the results were compared with and without reasoning on different queries. In the following tables are contained the number of the query, the query in natural language and a screenshot showing the SPARQL query and the result obtained.







Task 1

In task 1 are presented the queries implemented in film ontology.

Q2: Return all films with their title

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://semantics.id/ns/example/film#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select ?film ?title
where {
 ?film rdf:type :Film;
 rdfs:label ?title
}

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 select ?film ?title
5 where {
6 ?film rdf:type :Film;
7 rdfs:label ?title
8 }
9







keyboard shortcuts

TableRaw ResponsePivot TableGoogle ChartDownload as

Filter query results

Showing results from 1 to 5 of 5. Query took 0.1s, moments ago.

	film	title
1	ex:film_1	"Christopher Robin"
2	ex:film_2	"Son of a Gun"
3	ex:film_3	"Beauty and the Beast"
4	ex:film_4	"Dune"
5	ex:film_5	"Dune"

1






Q3: Is there a film named “Dune”

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX : <http://semantics.id/ns/example/film#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 ask {
 ?film rdf:type :Film;
 rdfs:label "Dune"
 }

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 ask {
5     ?film rdf:type :Film;
6     rdfs:label "Dune"
7 }
8

```

Run

keyboard shortcuts

Query took 0.1s, moments ago.

YES

Q5: Give all information about “Dune” movie released

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX : <http://semantics.id/ns/example/film#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 describe ?film
 where {
 ?film rdfs:label "Dune";
 :releaseYear 1984.
 }

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 describe ?film
5 where {
6     ?film rdfs:label "Dune";
7         :releaseYear 1984.
8 }
9

```

Run

keyboard shortcuts

TableRaw ResponsePivot TableGoogle Chart

Download asVisual

Filter query results

Showing results from 1 to 9 of 9. Query took 0.1s, moments ago.

	subject	predicate	object
1	ex:film_5	rdf:type	rdfs:Resource
2	ex:film_5	rdf:type	:Film
3	ex:film_5	rdf:type	:Artwork
4	ex:film_5	rdf:type	owl:NamedIndividual
5	ex:film_5	rdfs:label	"Dune"
6	ex:film_5	:hasActor	:kyle_maclachlan
7	ex:film_5	:hasPerformer	:kyle_maclachlan
8	ex:film_5	:hasGenre	:genre_science_fiction
9	ex:film_5	:releaseYear	"1984"^^xsd:integer

Q7: Return all writers and the film studios for which they have worked

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://semantics.id/ns/example/film#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

Construct{

 ?writer :works ?studio

}

where {

 ?film :hasScriptWriter ?writer.

 ?film :hasFilmStudio ?studio.

}

1

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

PREFIX : <http://semantics.id/ns/example/film#>

3

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

4

5

Construct{

6

?writer :works ?studio

7

}

8

where {

9

?film :hasScriptWriter ?writer.

10

?film :hasFilmStudio ?studio.

11

}

12

Run

keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Visual

Filter query results

Showing results from 1 to 6 of 6. Query took 0.1s, moments ago.

	subject	predicate	object
1	ex:writer_1	:works	ex:WaltDisneyPictures
2	ex:writer_2	:works	ex:WaltDisneyPictures
3	ex:julius_avery	:works	ex:EntertainmentOne
4	ex:writer_3	:works	ex:EntertainmentOne
5	ex:writer_4	:works	ex:WaltDisneyPictures
6	ex:writer_5	:works	ex:WaltDisneyPictures

Q9: Return all actors and the genre of the movie/movies that they played in

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://semantics.id/ns/example/film#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

Construct{

 ?actor :preferGenre ?genre

}

where {

 ?film :hasGenre ?genre.

 ?film :hasActor ?actor.

}

1

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

PREFIX : <http://semantics.id/ns/example/film#>

3

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

4

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

5

6

Construct{

7

?actor :preferGenre ?genre

8

}

9

where {

10

?film :hasGenre ?genre.

11

?film :hasActor ?actor.

12

}

13

Save

Open

Link

Expand

Network

Run

Press Alt+Enter keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Visual

Filter query results

Showing results from 1 to 16 of 16. Query took 0.1s, moments ago.

	subject	predicate	object
1	ex:ewan_mcgregor	:preferGenre	:genre_animation
2	ex:hayley_atwell	:preferGenre	:genre_animation
3	ex:ewan_mcgregor	:preferGenre	:genre_family
4	ex:hayley_atwell	:preferGenre	:genre_family
5	ex:alicia_vikander	:preferGenre	:genre_action

Q10: Select film studios that were established after 1960

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://semantics.id/ns/example/film#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select ?studio

where {

 ?studio a :FilmStudio.

 ?studio :establishedDate ?date.

 FILTER (?date > "1960-01-01"^^xsd:date)

}

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5
6 select ?studio
7 where {
8     ?studio a :FilmStudio.
9     ?studio :establishedDate ?date.
10    FILTER (?date > "1960-01-01"^^xsd:date)
11 }
12
13

```

Run

keyboard shortcuts

TableRaw ResponsePivot TableGoogle Chart

Download as

Filter query results

Showing results from 1 to 2 of 2. Query took 0.1s, moments ago.

	studio
1	ex:EntertainmentOne
2	ex:WaltDisneyPictures

Q13: Select titles of the movies and genre for which genre is “Action” or “Family”

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://semantics.id/ns/example/film#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select ?title ?genre_name

where {

 ?film rdf:type :Film;

 rdfs:label ?title.

 ?film :hasGenre ?genre.

 ?genre rdfs:label ?genre_name.

 filter(?genre_name = "Action" || ?genre_name = "Family")

}

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
5
6 select ?title ?genre_name
7 where {
8     ?film rdf:type :Film;
9         rdfs:label ?title.
10    ?film :hasGenre ?genre.
11    ?genre rdfs:label ?genre_name.
12    filter(?genre_name = "Action" || ?genre_name = "Family")
13 }
14

```

Run

keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

⚠ Showing results from 1 to 3 of 3. Query took 0.1s, yesterday at 20:26.

	title	genre_name
1	"Son of a Gun"	"Action"
2	"Christopher Robin"	"Family"
3	"Beauty and the Beast"	"Family"

Q14: Select name of actors who plays in movies ordered by birthdate (ascending)

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://semantics.id/ns/example/film#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select ?name_actor

where {

?actor a :Actor;
 :fullName ?name_actor;
 :dateOfBirth ?birthdate

} order BY ASC(?birthdate)

1

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

PREFIX : <http://semantics.id/ns/example/film#>

3

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

4

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

5

6

select ?name_actor

7

where {

8

?actor a :Actor;

9

:fullName ?name_actor;

10

:dateOfBirth ?birthdate

11

} order BY ASC(?birthdate)

12

13

Run

Press Alt+Enter keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

Showing results from 1 to 6 of 6. Query took 0.1s, moments ago.

	name_actor
1	"Kyle Merritt MacLachlan"
2	"Ewan McGregor"
3	"Hayley Atwell"
4	"Dan Stevens"
5	"Emma Watson"
6	"Zendaya Maree Stoermer Coleman"

Q16: Select names of the movies and the number of the actors per movies without including the movie "Dune"

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://semantics.id/ns/example/film#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select ?movie_name (count(?actor) as ?number_of_actors)
where {
    ?movie a :Film;
    rdfs:label ?movie_name.
    ?movie :hasActor ?actor
    filter(?movie_name != "Dune")
} Group by ?movie_name
```


1

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

PREFIX : <http://semantics.id/ns/example/film#>

3

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

4

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

5

6

select ?movie_name (count(?actor) as ?number_of_actors)

7

where {

8

?movie a :Film;

9

rdfs:label ?movie_name.

10

?movie :hasActor ?actor

11

filter(?movie_name != "Dune")

12

} Group by ?movie_name

13

14

15

Save

Folder

Link

Share

Connect

Run

Press Alt+Enter keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

Showing results from 1 to 3 of 3. Query took 0.1s, yesterday at 20:36.

	movie_name	number_of_actors
1	"Christopher Robin"	"2"xsd:integer
2	"Son of a Gun"	"2"xsd:integer
3	"Beauty and the Beast"	"3"xsd:integer

Q17: List all actors and crews together with the title of movies that they are involved in, ordered by their name

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://semantics.id/ns/example/film#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select ?person ?movie_title
where {
  ?movie a :Film
  { ?movie :hasActor ?person.
    ?person :fullName ?name_person.
    ?movie rdfs:label ?movie_title}
  union
  { ?movie :hasCrew ?person.
    ?person :fullName ?name_person.
    ?movie rdfs:label ?movie_title}
}order by ?name_person
```

1

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

PREFIX : <http://semantics.id/ns/example/film#>

3

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

4

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

5

6

select ?person ?movie_title

7

where {

8

?movie a :Film

9

{?movie :hasActor ?person.

10

?person :fullName ?name_person.

11

?movie rdfs:label ?movie_title}

12

union

13

{?movie :hasCrew ?person.

14

?person :fullName ?name_person.

15

?movie rdfs:label ?movie_title}

16

}order by ?name_person

Run

keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

Showing results from 1 to 18 of 18. Query took 0.1s, moments ago.

	person	movie_title
1	ex:writer_2	"Christopher Robin"
2	ex:alicia_vikander	"Son of a Gun"
3	ex:bill_condon	"Beauty and the Beast"
4	ex:dan_stevens	"Beauty and the Beast"
5	ex:emma_watson	"Beauty and the Beast"

Task 2

In task 2 in order to access the DBpedia SPARQL endpoint, the command: *service* `<https://dbpedia.org/sparql>{}` was used. The service keyword instructs a federated query processor a portion of SPARQL query to be executed against a remote SPARQL endpoint. Finally, the ontology that was used in order to implement the following queries, is the Film ontology (dbo:Film, where dbo is the prefix `<http://dbpedia.org/ontology/>`), as mentioned above.

Q18: How many South Korean movies are listed in DBPedia

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

select (count (distinct ?film) as ?count)
where {
 service <https://dbpedia.org/sparql>{
 ?film rdf:type dbo:Film.
 ?film dbp:country dbr:South_Korea
 }
}

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 PREFIX dbr: <http://dbpedia.org/resource/>
4 PREFIX dbp: <http://dbpedia.org/property/>
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
7
8 select (count (distinct ?film) as ?count)
9 where {
10 service <https://dbpedia.org/sparql>{
11 ?film rdf:type dbo:Film.
12 ?film dbp:country dbr:South_Korea
13 }
14 }

Run

keyboard shortcuts

TableRaw ResponsePivot TableGoogle Chart

Download as

Filter query results

Showing results from 1 to 1 of 1. Query took 0.3s, moments ago.

	count
1	*132^^xsd:integer

Q19: Find all movies released after year 2000

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>

11

```
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?film ?date
where {
  service <https://dbpedia.org/sparql>{
    ?film rdf:type dbo:Film .
    ?film dbp:releaseDate|dbp:released|dbo:releaseDate|dbo:premiereDate ?date
    filter(?date >= "2001-01-01"^^xsd:date)
  }
}
```



```
1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 PREFIX dbr: <http://dbpedia.org/resource/>
4 PREFIX dbp: <http://dbpedia.org/property/>
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
7 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
8
9 select distinct ?film ?date
10 where {
11   service <https://dbpedia.org/sparql>{
12     ?film rdf:type dbo:Film .
13     ?film dbp:releaseDate|dbp:released|dbo:releaseDate|dbo:premiereDate ?date
14     filter(?date >= "2001-01-01"^^xsd:date)
15   }
16 }
```

Table Raw Response Pivot Table Google Chart Download as 1 2 3 4 5

Filter query results		Showing results from 1 to 1,000 of 8,701. Query took 0.2s, moments ago.	
	film		date
1	http://dbpedia.org/resource/The_Bourne_Legacy_(novel)		"2004-06-22"^^xsd:date
2	http://dbpedia.org/resource/The_Last_Patriot		"2008-07-01"^^xsd:date
3	http://dbpedia.org/resource/What_the_Dead_Know		"2007-03-13"^^xsd:date
4	http://dbpedia.org/resource/Naked_Ambition:_An_R_Rated_Look_at_an_X_F		"2007-11-11"^^xsd:date
5	http://dbpedia.org/resource/Stone_Cold_(Baldacci_novel)		"2007-11-06"^^xsd:date
6	http://dbpedia.org/resource/Frozen_Fire_(novel)		"2006-09-07"^^xsd:date

Q20: Find all movies directed by Steven Spielberg where Tom Hanks is not playing

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```






```
select distinct ?film ?actor ?director
where {
```

```

    service <https://dbpedia.org/sparql>{
    ?film rdf:type dbo:Film .
    ?film dbp:director ?director.
    ?film dbo:starring ?actor.
    filter(?actor != dbr:Tom_Hanks && ?director = dbr:Steven_Spielberg)
    }
}

```

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 PREFIX dbr: <http://dbpedia.org/resource/>
4 PREFIX dbp: <http://dbpedia.org/property/>
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
7
8 select distinct ?film ?actor ?director
9 where {
10 service <https://dbpedia.org/sparql>{
11 ?film rdf:type dbo:Film .
12 ?film dbp:director ?director.
13 ?film dbo:starring ?actor.
14 filter(?actor != dbr:Tom_Hanks && ?director = dbr:Steven_Spielberg)
15 }
16 }

Run

keyboard shortcuts

Table Raw Response Pivot Table Google Chart

Download as

Filter query results

Showing results from 1 to 167 of 167. Query took 0.2s, moments ago.

	film	actor	director
1	http://dbpedia.org/resource/Always_(1989_film)	http://dbpedia.org/resource/Audrey_Hepburn	http://dbpedia.org/resource/Steven_Spielberg
2	http://dbpedia.org/resource/Always_(1989_film)	http://dbpedia.org/resource/Brad_Johnson_(actor)	http://dbpedia.org/resource/Steven_Spielberg
3	http://dbpedia.org/resource/Always_(1989_film)	http://dbpedia.org/resource/Holly_Hunter	http://dbpedia.org/resource/Steven_Spielberg
4	http://dbpedia.org/resource/Always_(1989_film)	http://dbpedia.org/resource/Richard_Dreyfuss	http://dbpedia.org/resource/Steven_Spielberg
5	http://dbpedia.org/resource/Always_(1989_film)	http://dbpedia.org/resource/John_Goodman	http://dbpedia.org/resource/Steven_Spielberg

Q21: List all movies and the english titles of the movies order by the title (descending) that are directed, produced and acted by Brad Pitt or Bradley Cooper

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

```

```

select distinct ?film ?name_movie_Brad_Pitt ?name_movie_Bradley_Cooper
where {
    service <https://dbpedia.org/sparql>{
    {
        ?film dbo:director|dbo:starring|dbo:producer dbr:Brad_Pitt;

```

```

    rdfs:label ?name_movie_Brad_Pitt.}
union
{
    ?film dbo:director|dbo:starring|dbo:producer dbr:Bradley_Cooper;
    rdfs:label ?name_movie_Bradley_Cooper.
}
FILTER (lang(?name_movie_Bradley_Cooper) = "en" || lang(?name_movie_Brad_Pitt) = "en")
}
} order by Desc(?name_movie_Brad_Pitt ) Desc(?name_movie_Bradley_Cooper)

```

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 PREFIX dbr: <http://dbpedia.org/resource/>
4 PREFIX dbp: <http://dbpedia.org/property/>
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
7
8
9 select distinct ?film ?name_movie_Brad_Pitt ?name_movie_Bradley_Cooper
10 where {
11 service <https://dbpedia.org/sparql>{
12 {
13 ?film dbo:director|dbo:starring|dbo:producer dbr:Brad_Pitt;
14 rdfs:label ?name_movie_Brad_Pitt.}
15 union
16 {
17 ?film dbo:director|dbo:starring|dbo:producer dbr:Bradley_Cooper;
18 rdfs:label ?name_movie_Bradley_Cooper.}
19 }
20 }
21 } order by Desc(?name_movie_Brad_Pitt) Desc(?name_movie_Bradley_Cooper)

Run

keyboard shortcuts

Table Raw Response Pivot Table Google Chart

Download as

Filter query results

Showing results from 1 to 107 of 107. Query took 0.2s, yesterday at 22:03.

	film	name_movie_Brad_Pitt	name_movie_Bradley_Cooper
1	http://dbpedia.org/resource/World_War_Z_(film)	"World War Z (film)"@en	
2	http://dbpedia.org/resource/War_Machine_(film)	"War Machine (film)"@en	
3	http://dbpedia.org/resource/Voyage_of_Time	"Voyage of Time"@en	
4	http://dbpedia.org/resource/Vice_(2018_film)	"Vice (2018 film)"@en	
5	http://dbpedia.org/resource/Two-Fisted_Tales_(film)	"Two-Fisted Tales (film)"@en	

Q22: Select all movies that are directed by Alfred Hitchcock and have more than 3 actors

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select ?film (count(?ac) as ?actor)
where {
    service <https://dbpedia.org/sparql>{
        ?film dbo:starring ?ac.
        ?film dbo:director dbr:Alfred_Hitchcock
    }
}

```

```

}
} GROUP BY ?film
HAVING (?actor > "3"^^xsd:integer)

```

```

1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX dbo: <http://dbpedia.org/ontology/>
3 PREFIX dbr: <http://dbpedia.org/resource/>
4 PREFIX dbp: <http://dbpedia.org/property/>
5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
6 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
7 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
8
9 select ?film (count(?ac) as ?actor)
10 where {
11   service <https://dbpedia.org/sparql>{
12     ?film dbo:starring ?ac.
13     ?film dbo:director dbr:Alfred_Hitchcock
14   }
15 } GROUP BY ?film
16 HAVING (?actor > "3"^^xsd:integer)

```

Run keyboard shortcuts

Table Raw Response Pivot Table Google Chart

Download as

Filter query results Showing results from 1 to 39 of 39. Query took 0.2s, yesterday at 22:08.

	film	actor
1	http://dbpedia.org/resource/Vertigo_(film)	"5"^^xsd:integer
2	http://dbpedia.org/resource/Secret_Agent_(1936_film)	"4"^^xsd:integer
3	http://dbpedia.org/resource/Dial_M_for_Murder	"5"^^xsd:integer
4	http://dbpedia.org/resource/Saboteur_(film)__Saboteur__1	"8"^^xsd:integer
5	http://dbpedia.org/resource/Foreign_Correspondent_(film)	"6"^^xsd:integer

Task 3

In task 3, 3 queries were implemented that provide different results with and without the inference in the triple store. The execution of each query were influenced by different entailment pattern (subclass, property domain and range, subproperties).

<p>Q23: Select all the performers from the movies with their names</p> <p>PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX : <http://semantics.id/ns/example/film#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX ex: <http://semantics.id/ns/example#> PREFIX foaf: <http://xmlns.com/foaf/0.1/></p> <pre>select ?performer ?name where { ?film :hasPerformer ?performer. ?performer :fullName ?name }</pre>
<p>The entailment pattern, which was used in the query 23, is the RDFS pattern related to subclasses. As can be seen from the two screenshots the results are completely different from each other. According to the first screenshot with inference, all the results are printed compared to the screenshot 2, without inference, in which they are not. The reason why this is happening is that the class Actor is subclass to the class Performer thus without inference, the system is not able to see this class-subclass relationship. Therefore, when the inference is on, all performers including the actors are printed in contrast to inference off where only the performers are printed.</p> <p>Inference ON</p>


```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX ex: <http://semantics.id/ns/example#>
5 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
6
7 select ?performer ?name
8 where {
9     ?film :hasPerformer ?performer.
10    ?performer :fullName ?name
11 }
12

```

Run

keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

Showing results from 1 to 10 of 10. Query took 0.1s, moments ago.

	performer	name
1	ex:ewan_mcgregor	"Ewan McGregor"
2	ex:hayley_atwell	"Hayley Atwell"
3	ex:alicia_vikander	"Alicia Vikander"
4	ex:ewan_mcgregor	"Ewan McGregor"
5	ex:dan_stevens	"Dan Stevens"

Inference OFF

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX ex: <http://semantics.id/ns/example#>
5 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
6
7 select ?performer ?name
8 where {
9     ?film :hasPerformer ?performer.
10    ?performer :fullName ?name
11 }
12

```

Run

keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, moments ago.

	performer	name
1	:alan_menken	"Alan Menken"

Q24: Select all the crew members from the movies with their names

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://semantics.id/ns/example/film#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX ex: <http://semantics.id/ns/example#>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

```
select ?crew ?name
where {
  ?film :hasCrew ?crew.
  ?crew :fullName ?name
}
```

The entailment pattern, which was used in the query 24, is the RDFS pattern related to sub-properties. As can be seen from the two screenshots the results are completely different from each other. According to the first screenshot with inference, all the results are printed compared to the screenshot 2, without inference, in which they are not. The reason why this is happening is that the property hasCrew has 3 subproperties named hasDirector, hasScriptWriter and hasComposer and thus without inference, the system is not able to see this relationship between the properties.

Inference ON

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX ex: <http://semantics.id/ns/example#>
5 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
6
7 select ?crew ?name
8 where {
9   ?film :hasCrew ?crew.
10  ?crew :fullName ?name
11 }
12
```

Run keyboard shortcuts

Table Raw Response Pivot Table Google Chart Download as

Filter query results		Showing results from 1 to 9 of 9. Query took 0.1s, moments ago.	
	crew		name
1	ex:marc_forster		"Marc Forster"
2	ex:writer_1		"Tom McCarthy"
3	ex:writer_2		"Alex Ross"
4	ex:julius_avery		"Julius Avery"
5	ex:writer_3		"John Collee"

Inference OFF

```

1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://semantics.id/ns/example/film#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX ex: <http://semantics.id/ns/example#>
5 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
6
7 select ?crew ?name
8 where {
9     ?film :hasCrew ?crew.
10    ?crew :fullName ?name
11 }
12

```

Run

keyboard shortcuts

TableRaw ResponsePivot TableGoogle Chart

Download as

Filter query results

No results. Query took 0.1s, moments ago.

crew	name
No data available in table	

Q25: Select all people (actors, directors, writers, etc) with their names and date of birth

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX : <http://semantics.id/ns/example/film#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX ex: <http://semantics.id/ns/example#>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

```

select ?person ?name ?birthdate
where {
    ?person a foaf:Person.
    ?person :fullName ?name.
    ?person :dateOfBirth ?birthdate
}

```

The entailment pattern, which was used in the query 25, is the RDFS pattern related to property range. As can be seen from the two screenshots the results are completely different from each other. According to the first screenshot with inference, all the results are printed compared to the screenshot 2, without inference, in which they are not. The reason why this is happening is that when the reasoning is disabled, can not be detected the range Person from the properties in order then to provide the information about the actors, directors, writers, etc.

Inference ON

1

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

3

PREFIX : <http://semantics.id/ns/example/film#>

4

5

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

6

7

PREFIX ex: <http://semantics.id/ns/example#>

8

9

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

10

11

12

13

select ?person ?name ?birthdate

where {

?person a foaf:Person.

?person :fullName ?name.

?person :dateOfBirth ?birthdate

}

Run

keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

Showing results from 1 to 7 of 7. Query took 0.1s, moments ago.

	person	name	birthdate
1	ex:dan_stevens	"Dan Stevens"	"1982-10-10" ^{^^xsd:date}
2	ex:emma_watson	"Emma Watson"	"1990-04-15" ^{^^xsd:date}
3	ex:ewan_mcgregor	"Ewan McGregor"	"1971-03-31" ^{^^xsd:date}
4	ex:hayley_atwell	"Hayley Atwell"	"1982-04-05" ^{^^xsd:date}
5	ex:writer_1	"Tom McCarthy"	"1966-06-07" ^{^^xsd:date}

Inference OFF

1

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

2

3

PREFIX : <http://semantics.id/ns/example/film#>

4

5

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

6

7

PREFIX ex: <http://semantics.id/ns/example#>

8

9

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

10

11

12

13

select ?person ?name ?birthdate

where {

?person a foaf:Person.

?person :fullName ?name.

?person :dateOfBirth ?birthdate

}

Run

keyboard shortcuts

Table

Raw Response

Pivot Table

Google Chart

Download as

Filter query results

No results. Query took 0.1s, moments ago.

	person	name	birthdate
No data available in table			