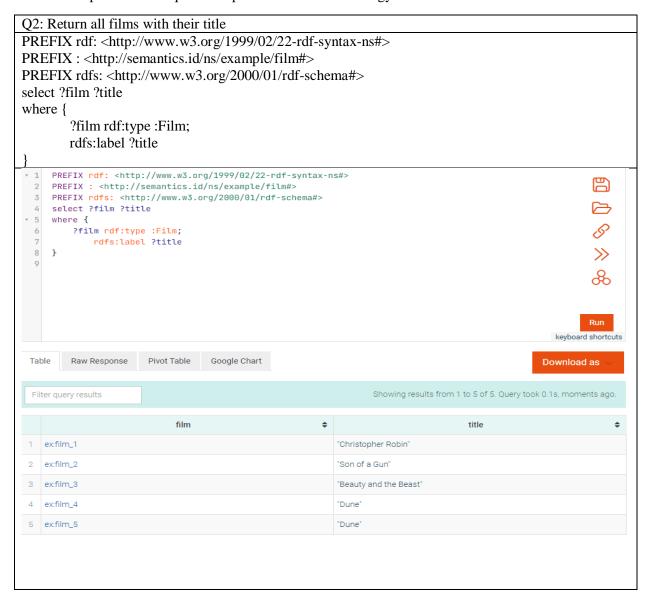
Report

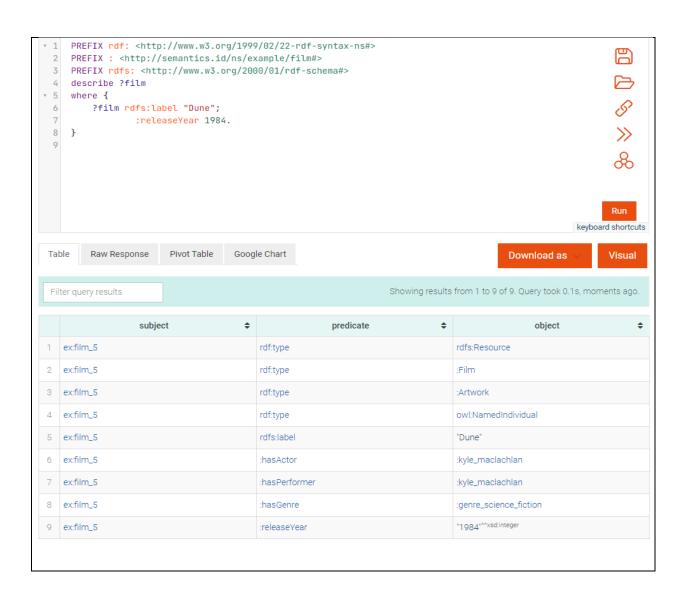
In this assignment were implemented basic and more complex queries using SPARQL via the GraphDB environment. Regarding the task 1, the option A was chosen and thus the film ontology was used for the implementation. For the task 2, first the DBPedia was explored and then five queries were constructed 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.

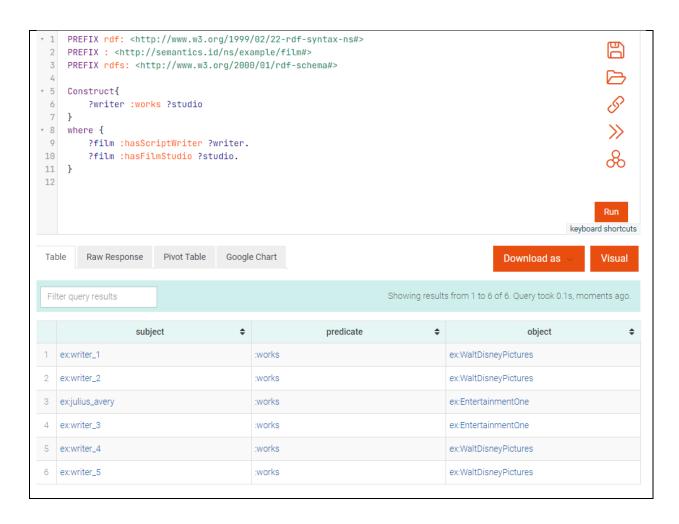
Task 1

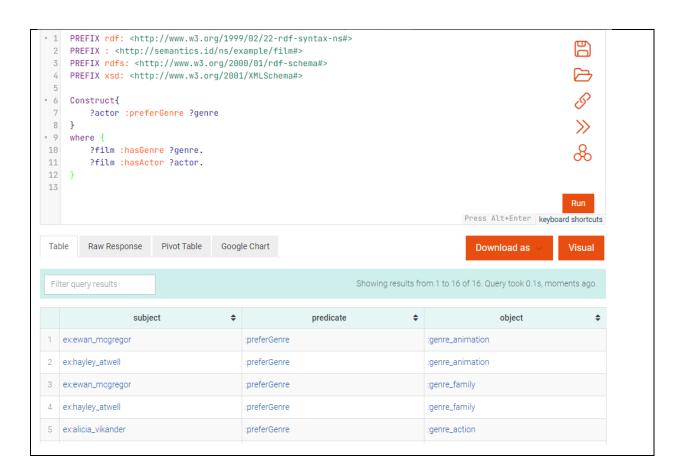
In task 1 are presented the queries implemented in film ontology.



```
Q3: Is there a film named "Dune"
PREFIX rdf: <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>
PREFIX : <a href="http://semantics.id/ns/example/film#">PREFIX : <a href="http://semantics.id/ns/example/film#">http://semantics.id/ns/example/film#</a>
PREFIX rdfs: <a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>>
ask {
            ?film rdf:type :Film;
           rdfs:label "Dune"
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
  2 PREFIX : <http://semantics.id/ns/example/film#>
      PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
+ 4 ask {
  5
          ?film rdf:type :Film;
                rdfs:label "Dune"
  6
      }
                                                                                                                                keyboard shortcuts
                                                                                                                  Query took 0.1s, moments ago.
                                                                   YES
```









```
Q13: Select titles of the movies and genre for which genre is "Action" or "Family"

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

PREFIX: <a href="http://semantics.id/ns/example/film#">http://semantics.id/ns/example/film#</a>

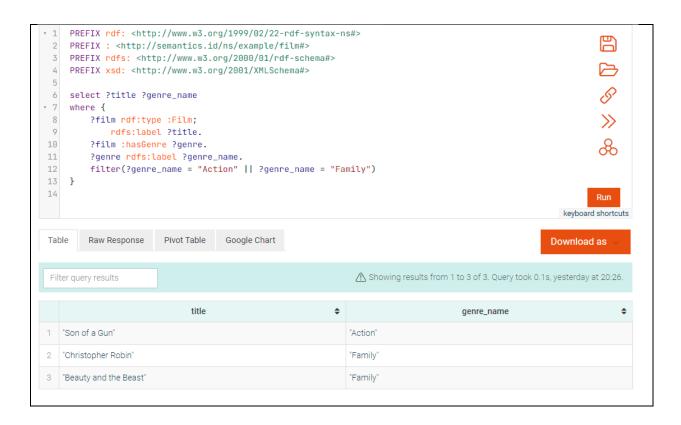
PREFIX rdfs: <a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>

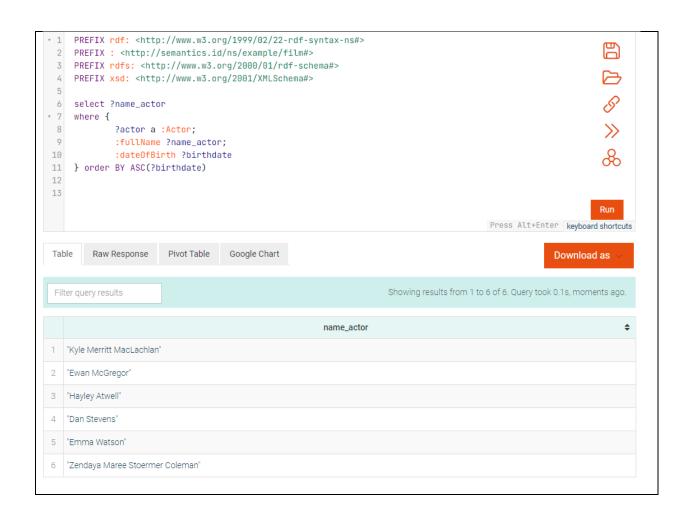
PREFIX xsd: <a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a>

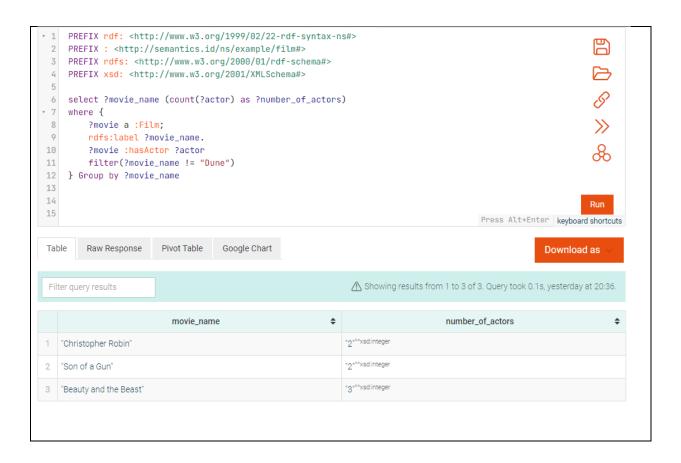
select ?title ?genre_name

where {

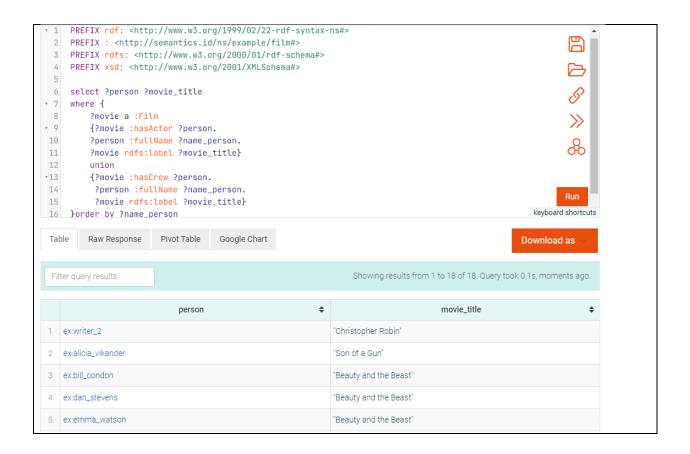
    ?film rdf:type :Film;
    rdfs:label ?title.
    ?film :hasGenre ?genre.
    ?genre rdfs:label ?genre_name.
    filter(?genre_name = "Action" || ?genre_name = "Family")
}
```







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: 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



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.



```
Q19: Find all movies released after year 2000

PREFIX foaf: <a href="http://xmlns.com/foaf/0.1/">http://xmlns.com/foaf/0.1/>
PREFIX dbo: <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/>
PREFIX dbr: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/></a>
```

```
PREFIX dbp: <a href="http://dbpedia.org/property/">http://dbpedia.org/property/</a>
PREFIX rdf: <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>
PREFIX rdfs: <a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>
PREFIX xsd: <a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a>
select distinct ?film ?date
where {
   service <a href="https://dbpedia.org/sparql">https://dbpedia.org/sparql</a>
       ?film rdf:type dbo:Film.
       ?film dbp:releaseDate|dbp:released|dbo:releaseDate|dbo:premiereDate ?date
       filter(?date >= "2001-01-01"^xsd:date)
   }
  7 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#>
        PREFIX xsd: <a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a>
   9 select distinct ?film ?date
  →10 where {
            service <https://dbpedia.org/sparql>{
                 ?film rdf:type dbo:Film .
                 ?film dbp:releaseDate|dbp:released|dbo:releaseDate|dbo:premiereDate ?date
   13
   14
                 filter(?date >= "2001-01-01"^^xsd:date)
                                                                                                                                Run
   15
                                                                                                                        keyboard shortcuts
            Raw Response
                              Pivot Table
                                            Google Chart
                                                                                                                          1 2 3 4 5 >
   Table
                                                                                                    Download as
                                                                        Showing results from 1 to 1,000 of 8,701. Query took 0.2s, moments ago.
    Filter query results
                                    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
     http://dbpedia.org/resource/Naked_Ambition:_An_R_Rated_Look_at_an_X_F "2007-11-11" "xsd:date
                                                                         '2007-11-06"^^xsd:date
   5 http://dbpedia.org/resource/Stone_Cold_(Baldacci_novel)
                                                                        "2006-09-07"^^xsd:date
   6 http://dbpedia.org/resource/Frozen_Fire_(novel)
```

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

PREFIX foaf: <a href="http://xmlns.com/foaf/0.1/">http://xmlns.com/foaf/0.1/></a>

PREFIX dbo: <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a>

PREFIX dbr: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a>

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

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

select distinct ?film ?actor ?director

where {
```

```
service <a href="mailto:service">https://dbpedia.org/sparql>{</a>
      ?film rdf:type dbo:Film.
      ?film dbp:director ?director.
      ?film dbo:starring ?actor.
      filter(?actor != dbr:Tom_Hanks && ?director = dbr:Steven_Spielberg)
     PREFIX foaf: <http://xmlns.com/foaf/0.1/>
     PREFIX dbo: <http://dbpedia.org/ontology/>
 3 PREFIX dbr: <http://dbpedia.org/resource/>
     PREFIX dbp: <http://dbpedia.org/property/>
 5 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
     PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 8
     select distinct ?film ?actor ?director
v 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)
                                                                                                                         Run
 15
                                                                                                                 keyboard shortcuts
 Table
          Raw Response
                           Pivot Table
                                        Google Chart
                                                                                                               Download as
                                                                     Showing results from 1 to 167 of 167. Query took 0.2s, moments ago.
 Filter query results
                      film
                                                               actor
                                                                                                        director
                                                                                                                              $
1 http://dbpedia.org/resource/Always_(1989_film) http://dbpedia.org/resource/Audrey_Hepburn
                                                                                        http://dbpedia.org/resource/Steven_Spielberg
   http://dbpedia.org/resource/Always_(1989_film) http://dbpedia.org/resource/Brad_Johnson_(actor http://dbpedia.org/resource/Steven_Spielberg
   http://dbpedia.org/resource/Always_(1989_film) http://dbpedia.org/resource/Holly_Hunter
                                                                                        http://dbpedia.org/resource/Steven_Spielberg
    http://dbpedia.org/resource/Always_(1989_film) http://dbpedia.org/resource/Richard_Dreyfuss
                                                                                        http://dbpedia.org/resource/Steven Spielberg
   http://dbpedia.org/resource/Always_(1989_film) http://dbpedia.org/resource/John_Goodman
                                                                                        http://dbpedia.org/resource/Steven_Spielberg
```

```
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)
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/>
     PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
     PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 8
 9
    select distinct ?film ?name_movie_Brad_Pitt ?name_movie_Bradley_Cooper
+10
·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.}
                                                                                                             Run
 15
              union
                                                                                                       keyboard shortcuts
 Table
         Raw Response
                         Pivot Table
                                     Google Chart
                                                                                                    Download as

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

 Filter query results
                    film
                                       ٠
                                                  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
                                          "Vice (2018 film)"@en
 4 http://dbpedia.org/resource/Vice (2018 film)
 5 http://dbpedia.org/resource/Two-
                                          "Two-Fisted Tales (film)"@en
   Fisted_Tales_(film)
```

} GROUP BY ?film HAVING (?actor > "3"^^xsd:integer) 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# 9 select ?film (count(?ac) as ?actor) +10 where { •11 service <https://dbpedia.org/sparql>{ ?film dbo:starring ?ac. 12 13 ?film dbo:director dbr:Alfred_Hitchcock 14 15 } GROUP BY ?film keyboard shortcuts 14 HAVING (Section > "Z"AAved integen) Table Raw Response Pivot Table Google Chart Download as ⚠ Showing results from 1 to 39 of 39. Query took 0.2s, yesterday at 22:08. Filter query results film actor "5"^^xsd:integer 1 http://dbpedia.org/resource/Vertigo_(film) "4"^^xsd:integer 2 http://dbpedia.org/resource/Secret_Agent_(1936_film) 3 http://dbpedia.org/resource/Dial_M_for_Murder "5"^^xsd:integer "8"^^xsd:integer 4 http://dbpedia.org/resource/Saboteur_(film)__Saboteur__1 "6"^^xsd:integer 5 http://dbpedia.org/resource/Foreign_Correspondent_(film)

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).

```
Q23: Select all the performers from the movies with their names

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

PREFIX: <a href="http://semantics.id/ns/example/film#">http://semantics.id/ns/example/film#</a>

PREFIX ex: <a href="http://semantics.id/ns/example#">http://semantics.id/ns/example#</a>

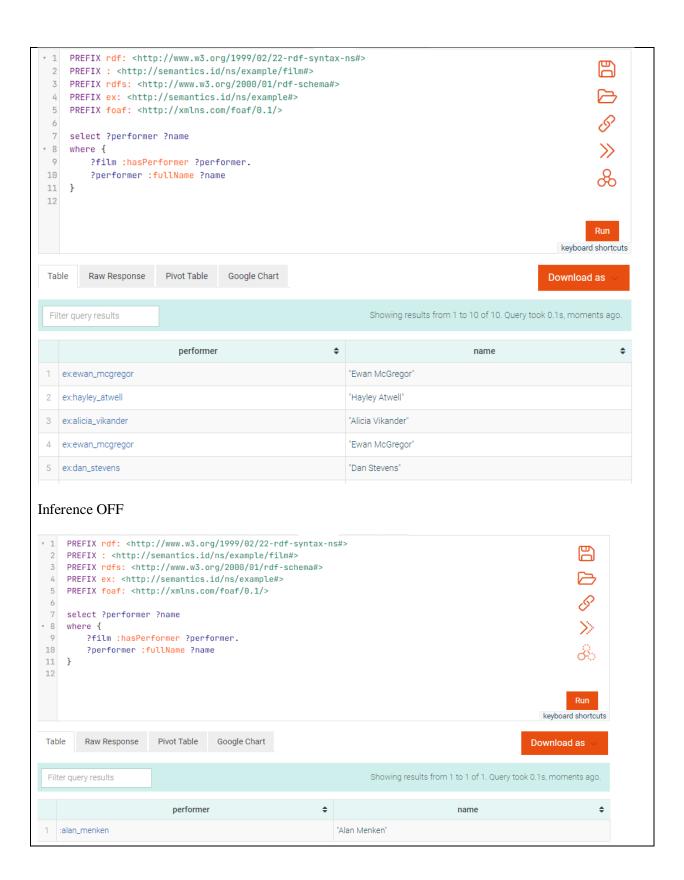
PREFIX foaf: <a href="http://xmlns.com/foaf/0.1/">http://xmlns.com/foaf/0.1/</a>

select ?performer ?name

where {
    ?film:hasPerformer ?performer.
    ?performer:fullName ?name
}
```

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.

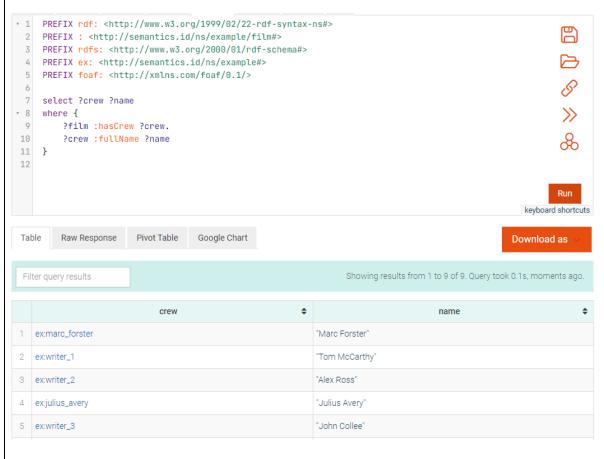
Inference ON



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 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 has Crew has 3 subproperties named has Director, has Script Writer and has Composer and thus without inference, the system is not able to see this relationship between the properties.

Inference ON



Inference OFF



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://semantics.id/ns/example# PREFIX foaf: http://xmlns.com/foaf/0.1/ select ?person ?name ?birthdate where { ?person a foaf:Person. ?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

