188.399 Introduction to Semantic Systems 2022W

Assignment 2 - Querying Knowledge Graphs with SPARQL (15 pts)

Instructions

Deadline

Make sure to upload your results **by November 18th, 2022**!

What you should hand in

Please upload your work to TUWEL by November 18th, 2022. Please name all your files with the following format: A[x]_[family name]_[student number], where x represents the number of the assignment. Your submission should follow the specified naming convention and include:

1. A short report (e.g., A2_Mustermann_1234_Report.pdf) listing all the implemented queries. On the next page, each query is given a **query code** Q[own]x, x= {1,21}. For **each** query you address please list:
   - The query code (e.g., Q1, Qown1);
   - For the queries that you propose, please also add a description in natural language of what the query should do (similar to the descriptions we provided);
   - Your solution in terms of the SPARQL query (please make sure this is in text – not image - format and can be copied and pasted);
   - A screenshot showing the SPARQL query and the result obtained (as a proof that you ran the query).

Questions

Please post general questions in the TUWEL discussion forum of the course. You can also discuss problems and issues you are facing there. We appreciate it if you help other students out with general problems or questions regarding the tools used. For obvious reasons, however, please do not post any solutions there.

You can also contact Filip Kovacevic and Laura Waltersdorfer directly (with specific questions) at:

filip.kovacevic@tuwien.ac.at
laura.waltersdorfer@tuwien.ac.at

Please use the subject line ISS_2022_<your subject> to minimize the probability that your Email gets lost in our inbox.

188.399 Introduction to Semantic Systems 2022W

Assignment 2 - Querying Knowledge Graphs with SPARQL (15 pts)

Introduction

In this assignment, you will exercise important skills and technologies needed for developing a basic semantic application enabled by a knowledge graph. In particular, you will follow these two tasks (each task is described in detail in the next section):

1. You will exercise the main SPARQL features by implementing increasingly complex queries either on a didactic ontology (option A) or the ontology that you created in assignment 1 (option B).
2. You will write SPARQL queries to extract information from the DBPedia KG.

Tasks description

Task 1 (Option A): Basic SPARQL on a didactic ontology (5 points)

For this task, please download and install the GraphDB triple store [1]. Create your own repository within GraphDB and upload the **film.ttl** ontology [2]. The ontology schema is shown in Figure 1. Optionally you can use the WebVowl editor [3] to visualize and interactively inspect the ontology.

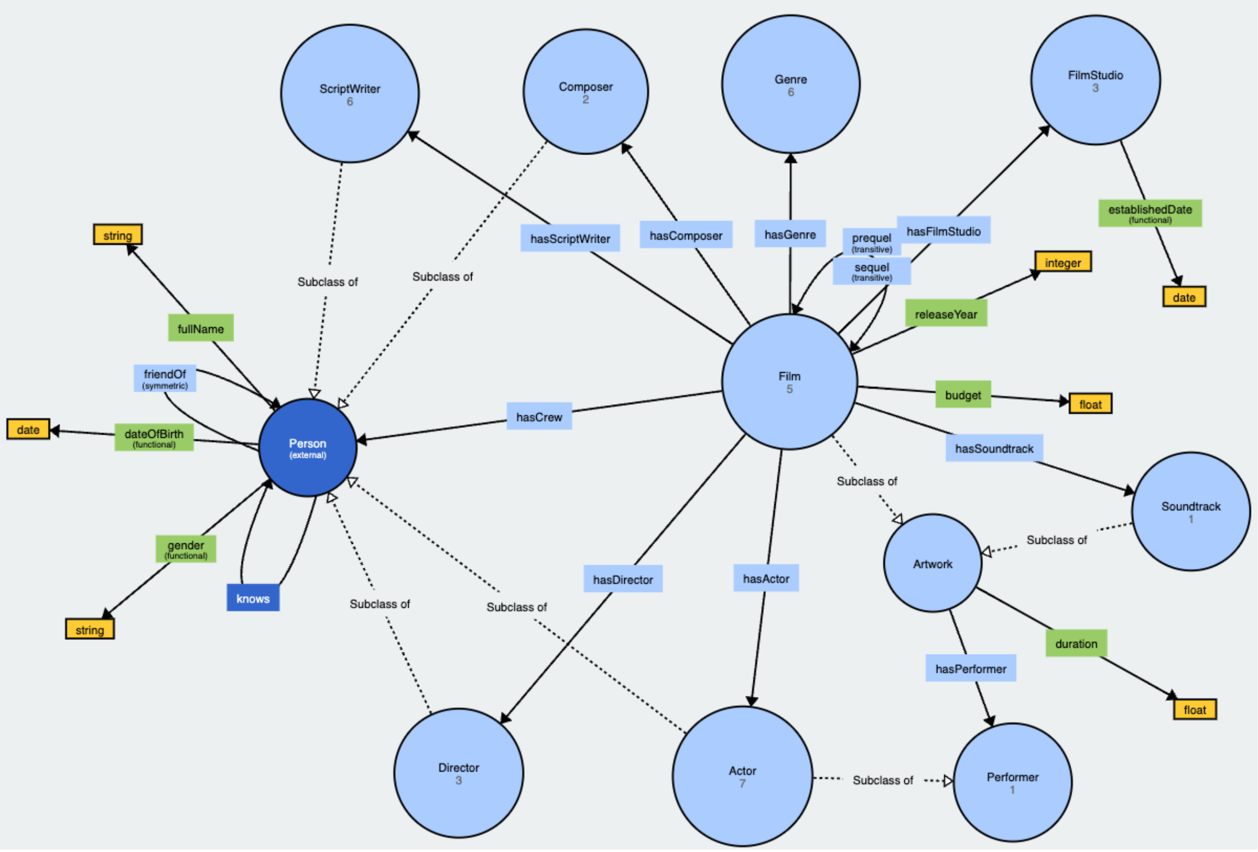

Figure 1: Schema of film.ttl ontology.

Please choose from the following queries as indicated and implement them as SPARQL queries. For each correctly implemented query you will be awarded 0.5 point (incorrectly implemented queries will lead to 0 points).

SELECT queries – please implement **one of** these queries:

- Q1: Return all films
- Q2: Return all films with their title

ASK queries – please implement **one of** these queries:

- Q3: Is there a film named "Dune"?
- Q4: Is there a film named "Dune" released before 1984?

DESCRIBE queries – please implement **one of** these queries:

- Q5: Give me all information about "Dune" movie released in 1984
- Q6: Give me all information about actors which are playing in "Dune" released in 1984

SPARQL CONSTRUCT queries – please implement **one of** these queries and additionally build a query of **your own** with code **Q9**

- Q7: Return all writers and the film studios for which they have worked
- Q8: Return all actors and directors who know each other from movies

FILTER queries – please implement **one of** these queries and additionally build a query of **your own** with code **Q13**

- Q10: Select film studios that were established after 1960
- Q11: Select unique name of actors who play in movies between 2014 and 2020
- Q12: Select name of actors who play in movies released after 2016 and with title containing string "Beauty"

ORDER and GROUP queries – please implement **one of** these queries and additionally build a query of **your own** with code **Q16**

- Q14: Select name of actors who plays in movies ordered by birthdate (ascending)
- Q15: Count the average number of ScriptWriters involved in each movie!

UNION queries – please implement this query **without** utilizing inference mechanisms**:**

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

Task 1 (Option B): Basic SPARQL on your own ontology (5 points)

Similar to option A of Task 1, please download and install GraphDB then create a repository. Please upload the ontology you created during Assignment 1. You will need to make sure your ontology contains a number of instances (5-10) to make querying meaningful.

Please think of and implement 10 queries similar to those described at option A, as follows:

- **Qown1**: A simple **SELECT** query
- **Qown2**: A simple **ASK** query
- **Qown3**: A simple **DESCRIBE** query
- **Qown4, Qown5**: Two **SPARQL CONSTRUCT** queries
- **Qown6, Qown7**: Two queries using **FILTER**

- **Qown8, Qown9**: Two queries using **ORDER** and **GROUP**
- **Qown10**: A query using **UNION**

Task 2: Querying knowledge graphs on the web (5 points)

Please access the DBPedia SPARQL endpoint (http://dbpedia.org/sparql OR http://yasgui.org/) and implement **all of** the following queries.

- Q18: How many South Korean movies are listed in DBPedia?
- Q19: Find all movies released after year 2000.
- Q20: Find all movies directed by Steven Spielberg where Tom Hanks is not playing

Additionally, please propose and implement 2 complex queries of your own including (any combination of) FILTER, ORDER, GROUP, UNION. Name these **Q21** and **Q22**.

Each query from Task 2 is worth 1 point.

Note: you might need to first explore DBpedia's schema [4] to understand how to phrase your queries.

Task 3: Querying with/without inference (5 points)

Write 3 queries of your own (**Q23, Q24, Q25**) that provide different answers when the inference in the triple store is enabled. The execution of each query should be influenced by a **different** entailment pattern (e.g., Q23 could be influenced by pattern RDFS related to property domains; Q24 could be influenced by the RDFS pattern related to sub-properties etc). In your report, provide the queries, their results obtained with and without reasoning and explain which entailment pattern explains the obtained results. Feel free to extend your data (your own or the movie ontology) with additional elements if needed to demonstrate your queries. If you do so, please document these additions in your document.

Hint: In GraphDB you can enable/disable reasoning in the right side of the SPARQL query panel, with the sign ">>".

Each query and adequate explanation from Task 3 are worth 1.5 points (you will receive 0.5 points to round up to 5 points).

Tools:
[1] GraphDB triple store: https://www.ontotext.com/products/graphdb/
[3] WebVOWL: https://service.tib.eu/webvowl/ ; Run editor => Click "Ontology" Tab => Under "Custom Ontology" add http://semantics.id/ns/example/film into the URL => Click "Visualize"

Resources:
[2] film.ttl ontology (also in TUWEL): http://semantics.id/ns/example/film
[4] DBPedia ontology: http://mappings.dbpedia.org/server/ontology/ ; DBPedia class hierarchy: http://mappings.dbpedia.org/server/ontology/