Semantic Web (CSE632)
Winter 2020
Assignment 4

Instructions

1. This assignment is not a group activity. Each student has to work on it by himself/herself and submit the assignment.
2. Plagiarism check and corresponding policies will be strictly enforced. Students can be selected at random for a "viva" on the assignment and if the responses are not satisfactory, they will get a 0 in the assignment.
3. All the text and images should go into a single pdf file. All the code should be in a directory named "code" and the name of the files should be RollNo_Q#.java, where # is the question number. Include a jar file of the source code. All the SPARQL queries should be put in a text file and each SPARQL query should have a description and the question number. This should go into a directory named "queries". Create a zip file with the name RollNo_HW4.zip and submit it.
4. Provide comments in the code and document every step of your code (except variable assignment).
5. If the instructions from points 3 and 4 are not followed, there will be a penalty of 10% reduction in the points.
6. In all the questions, the term "movie" has been used to mean **both** the movies and tv shows from the Netflix csv file.

Questions **Max points: 50**

1. **35 pt**
    a. Redo the conversion of data from the Netflix csv file from Assignment-1 into triples by using the schema described in the ontology that was created in Assignment-3.
    b. Install Apache Jena Fuseki (https://jena.apache.org/documentation/fuseki2/). All the configuration related parameters should go into a separate configuration file and this file should be clearly documented.
    c. Construct the following SPARQL queries and run them using the Java API (https://jena.apache.org/documentation/rdfconnection/).
        i. Create two named graphs in the store with the following IRIs: http://iiitd.ac.in/sweb/your-rollno/newmoviesgraph and http://iiitd.ac.in/sweb/your-rollno/oldmoviesgraph
        ii. Load the triples into the default graph
        iii. Move the movies whose release year is on or after 2016 to the new movies graph and the rest of them to the old movies graph. After this operation, there should not be any triples in the default graph.

d. Make this a 5-star Linked Data by linking this dataset to at least two other datasets from Linked Open Data (for example, DBpedia, GeoNames, LinkedMDB, etc.). Explain to which datasets you linked to, how you linked them, and why you consider your dataset as a 5-star Linked Data.
e. Add a frontend to it using either brwsr (https://github.com/Data2Semantics/brwsr) or Pubby (https://github.com/cygri/pubby). The former is in Python and the latter is in Java.

For Fuseki and the Linked Data browser (brwsr/Pubby), use your machine's hostname instead of localhost when accessing them through the browser.

Take screenshots to show that you have installed Fuseki, made your dataset into Linked Data and also to show that one of the frontends is running successfully. Add these images to a pdf file.

2. Construct the following SPARQL queries and run them using the SPARQL over HTTP feature (https://jena.apache.org/documentation/fuseki2/soh.html) of Fuseki. In each of the following cases, provide the complete command including the query you have used, along with the result file. A single file should be provided for all the queries and separate files for the results. Name the result file as "Q2-subquestion".txt. **15 pt**
   a. Retrieve the name of all the movies that have a director with the last name as "Shetty". If present, retrieve the name of the co-director as well.
   b. Retrieve the names of all the movies that are categorized as either a comedy or a drama or both.
   c. Retrieve the names of all the movies that are released in the United States between the year 2010 and 2020 and that have the word "couple" in their description.
   d. Write a SPARQL query involving property paths to retrieve two movies that have at least one director in common and have a run time of greater than or equal to 60 minutes.
   e. Construct a SPARQL query of your choice on your movie dataset that satisfies the following star pattern (join on the subject).