DSCI 558 Building Knowledge Graphs

Homework 3: SPARQL

Released: Sep 19, 2022
Due: Sep 26, 2022 @ 23:59 PST

Ground Rules

This homework must be done individually. You can ask others for help with the tools, however, the submitted homework must be your own work.

Summary

In this homework, you must write sparql queries for Wikidata as well as for a custom knowledge graph. You will use two different types of query engines: an online tool for Wikidata and a locally run instance of Apache Jena for the custom triple store.

Submission Instructions

You must submit (via Blackboard) the following files/folders in a single .zip archive named **Firstname_Lastname_hw03.zip**:

- **wikidata_1.txt**: Query for Task 1.1.
- **wikidata_1.png/jpeg**: snapshots of executing each of Task 1.1 queries on Wikidata SPARQL endpoint.
- **wikidata_2.txt**: Query for Task 1.2.
- **wikidata_2.png/jpeg**: snapshots of executing each of Task 1.2 queries on Wikidata SPARQL endpoint.
- **jena_1.txt**: a collection of all queries for Task 2.2.
- **jena_1.png/jpeg**: snapshots of executing each of Task 2.2 queries on Apache Jena server.
- **jena_2.txt**: a collection of all queries for Task 2.3.
- **jena_2.png/jpeg**: snapshots of executing each of Task 2.3 queries on Apache Jena server.

Task 1: Wikidata - 4 points

In this task, you will use Wikidata's query service (SPARQL endpoint) to execute queries. The service is available here: https://query.wikidata.org/ Write SPARQL queries to accomplish the following sub-tasks (each is worth 2 points). Execute your queries in the GUI and include the results (screenshots) in your submission, along with a text file containing all answers. **All queries should be limited to 10 results**.

Task 1.1 (2 pts)

*The **Ballon d'Or** is an annual football award presented by French news magazine France Football since 1956. **Between 2010 and 2015**, in an agreement with FIFA, the award was temporarily merged with the FIFA World Player of the Year and known as the **FIFA Ballon d'Or**.*

Get **Player URI**, **Player Name**, and **their number of Ballon d'Or awards** of the players who have won Ballon d'Or since 2000 in descending order of number of Ballon d'Or awards.

- There might be missing awards in Wikidata.
    - 2020 - no award ceremony due to COVID
    - 2016, 2017 - missing records in Wikidata

Task 1.2 (2 pts)

Get ~~**Book URI**~~**Series URI**, ~~**Book Name**~~**Series Name**, **Author URI**, and **Author Name** of authors who have ~~more than 10 book series (e.g., Harry Potter)~~**a book series that contain more than 10 books** in descending order of number of series. Here, show only the top ~~20~~**10** results.

== Get Book Series URI, Book Series Name, Author URI, Author Name of book series that contains more than 10 books in descending order of number of books in the series. Here, show only the top 10 results.

e.g., examples of book series name and author name of book series that contains more than 10 books.

bookseriesLabel	authorLabel	count
Gor	John Norman	37
Karl May's Gesammelte Reiseerzählungen	Karl May	33
Anita Blake: Vampire Hunter	Laurell K. Hamilton	28
Jeżycjada	Małgorzata Musierowicz	22
Harry Bosch	Michael Connelly	22

Task 2: Apache Jena - 6 points

In this task, you will work with RDF triples over Apache Jena.

Task 2.1 (2 pts)

First, download and install Apache Jena Fuseki (SPARQL server) from: https://jena.apache.org/download/index.cgi. Simply download the compressed file called "apache-jena-fuseki-4.6.1.tar.gz" and unzip it, then start the server via command line:

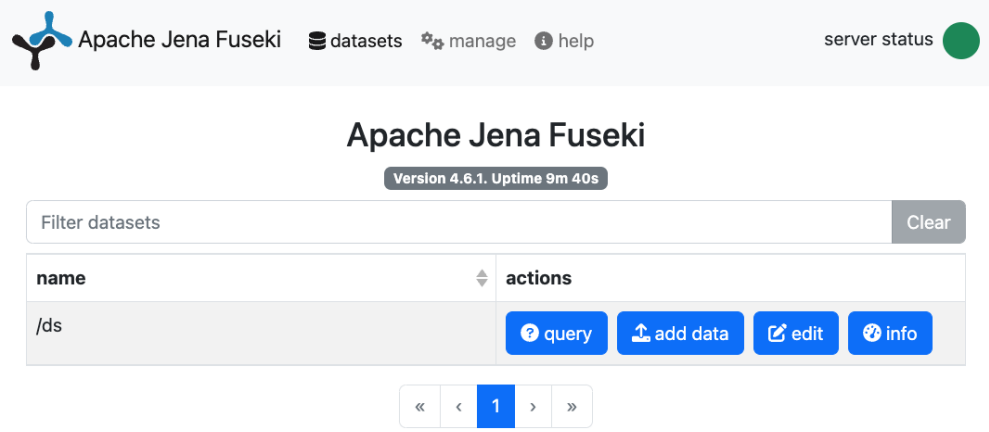
```
./apache-jena-fuseki-4.6.1/fuseki-server
```

You need to download a triple store from this url and load it into a local Jena server by:

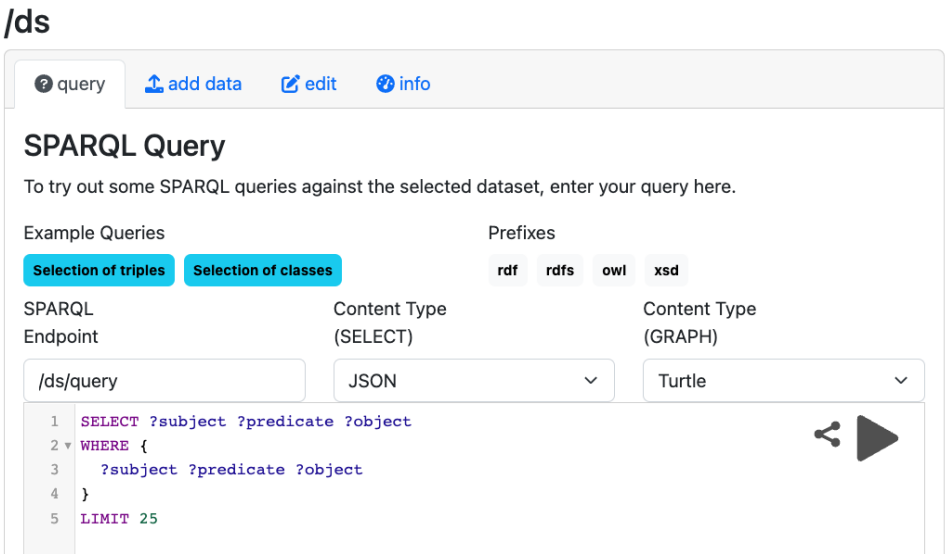
```
./apache-jena-fuseki-4.6.1/fuseki-server --file=<path to .nt
file>/data.nt --localhost /ds
```

where <ds> is any name you wish to give to your server.

You can navigate to the server by typing the following in your browser: http://localhost:3030/. Click on Query to reach the tool which allows you to execute SPARQL queries on ds.



Checkout what a triple store looks like using example queries!



Tips

- You may need to install JAVA if you don't have: https://www.oracle.com/java/technologies/downloads/#java11

Then, you will write and execute queries for the following sub-tasks on Apache Jena and include the results (screenshots) in your submission, along with a text file containing all answers.

Task 2.2 (2 pts)

Get **names of films** and **comma-separated names of all actors who starred in them**. Sort by movie name in ascending order of first alphabet (might be start from special character such as ", $, *) and limit to top 20 results.

Task 2.3 (2 pts)

Get **names of female actors who were born before 1970** and **their date of birth**, sorted by birth date in ascending order and limit to top 20 results.

Note that you do not have the data about gender and date-of-birth for actors in the provided graph. You will need to retrieve these values from an additional SPARQL endpoint using a federated query extension (the SERVICE keyword, read more about it here: https://www.w3.org/TR/sparql11-federated-query/). Hint: use DBpedia.

Updates

- Task 1.2
    - You can just keep only one author for multiple authors, or merge it with a comma if you can. But, we don't expect you to make multiple rows for the same book by different authors.
    - I understand there must be several ways to get results for this task. KG's subgraph for this task is so large and complex that there must be several ways to solve this task. We will check the adequacy of your query. There can be multiple answers to this task.
        - **Results can be different from screenshot. There can be more advanced results.**
- Task 2.2
    - If you just sort the film name by "order by" syntax, it will automatically sort results by alphabetical order.