

Semantic Web - Winter 2020 Assignment 4

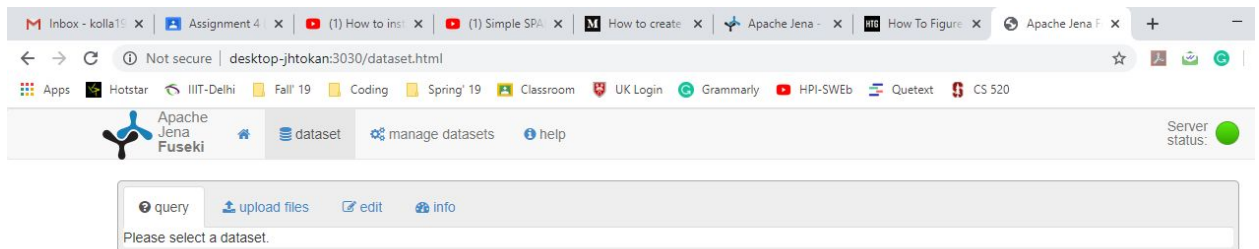
Q1)

(a) The code for this part is in CSVtoTriples.java . OpenCSV parser is used for reading the csv file and making triples. All the triples are made with title as subject and remaining all as properties and objects. For handling the spaces in the title names, replace() method from String class is used.

(b) Fuseki installation

```
C:\WINDOWS\system32\cmd.exe
2020-05-11 15:44:50] Server INFO Apache Jena Fuseki 3.14.0
2020-05-11 15:44:51] Config INFO FUSEKI_HOME=C:\Users\hanum\Desktop\apache-jena-fuseki-3.14.0\apache-jena-fuseki-3.14.0\
2020-05-11 15:44:51] Config INFO FUSEKI_BASE=C:\Users\hanum\Desktop\apache-jena-fuseki-3.14.0\apache-jena-fuseki-3.14.0\run
2020-05-11 15:44:51] Config INFO Shiro File: file:///C:/Users/hanum/Desktop/apache-jena-fuseki-3.14.0\apache-jena-fuseki-3.14.0\run\shiro.ini
2020-05-11 15:44:51] Config INFO Configuration file: C:\Users\hanum\Desktop\apache-jena-fuseki-3.14.0\apache-jena-fuseki-3.14.0\run\config.ttl
2020-05-11 15:44:51] riot WARN [line: 5, col: 9 ] Bad IRI: <C:\Users\hanum\Desktop\apache-jena-fuseki-3.14.0\apache-jena-fuseki-3.14.0\run\c
onfig.ttl#> Code: 4/UNWISE_CHARACTER in PATH: The character matches no grammar rules of URIs/IRIs. These characters are permitted in RDF URI Referen
ces, XML system identifiers, and XML Schema anyURIs.
2020-05-11 15:44:52] Server INFO Started 2020/05/11 15:44:52 IST on port 3030
```

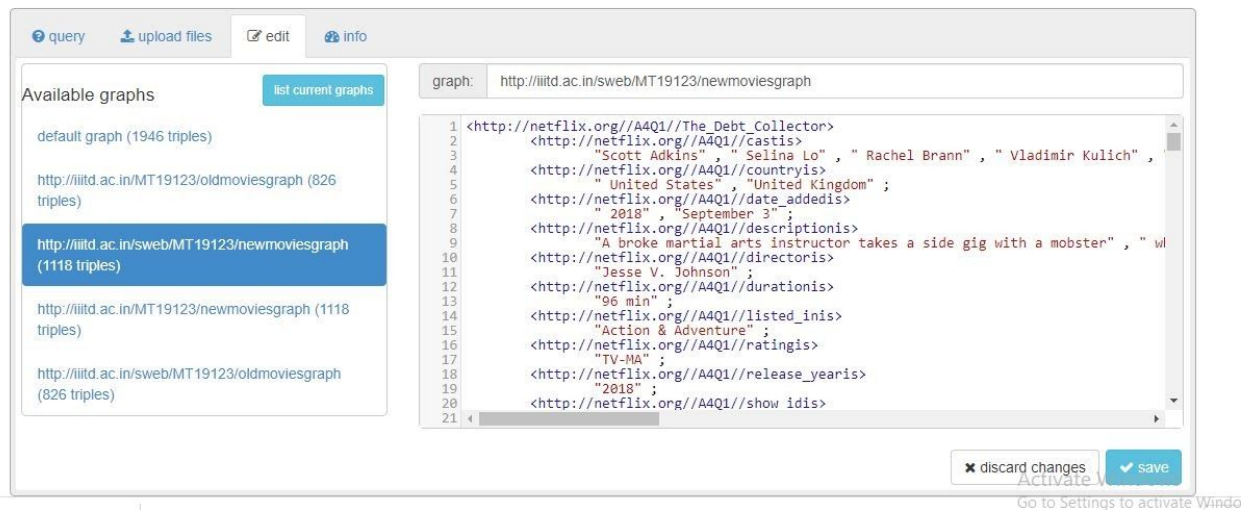
Accessing fuseki server with host name



Configuration file



(c) Default graphs and required named graphs are created.



Q2)

- a) We use filter here for getting the names which contains Shetty and for getting codirectors we use OPTIONAL.

Query:-

prefix Netflix: <http://netflix.org//A4Q1//>

```
SELECT    DISTINCT    ?title    ?director    ?coDirector    FROM
<http://iiitd.ac.in/web/MT19123/newmoviesgraph>
FROM <http://iiitd.ac.in/web/MT19123/oldmoviesgraph>
Where {
    ?title Netflix:directoris ?director . FILTER regex(?director,"Shetty") .
    OPTIONAL { ?title Netflix:directoris ?coDirector . FILTER
(!regex(?coDirector,"Shetty")).}
}
LIMIT 500
```

Query Command:-

```
C:\Users\hanum\Desktop\fuseki\bin>ruby s-query
--service=http://DESKTOP-JHTOKAN:3030/Netflix.csv
--query=C:\Users\hanum\Desktop\2a.rq>Q2-a.txt
```

(b) FILTER is used to retrieve the movies of comedy and drama genre.

Query:-

prefix Netflix: <http://netflix.org//A4Q1//>

```
SELECT      ?title      ?genre      FROM
<http://iiitd.ac.in/sweb/MT19123/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/MT19123/oldmoviesgraph>
WHERE {
  ?title Netflix:listed_inis ?genre
    FILTER
  (regex(?genre,"Comedies")||regex(?genre,"Dramas")||regex(?genre,"Comedy
  "))
}
LIMIT 500
```

Query command:-

```
C:\Users\hanum\Desktop\fuseki\bin>ruby s-query
--service=http://DESKTOP-JHTOKAN:3030/Netflix.csv
--query=C:\Users\hanum\Desktop\2b.rq>Q2-b.txt
```

(c) Query

prefix Netflix: <http://netflix.org//A4Q1//>

```
SELECT      ?title      ?releaseDate      ?country      ?description      FROM
<http://iiitd.ac.in/sweb/MT19123/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/MT19123/oldmoviesgraph>
WHERE {
  ?title Netflix:release_yearis ?releaseDate .
  ?title Netflix:countryis ?country .
  ?title Netflix:descriptionis ?description .
    FILTER ( ?releaseDate >= "2010" && ?releaseDate <= "2020" &&
  regex(?country,"United States") && regex(?description,"couple"))
}
LIMIT 500
```

Query command

```
C:\Users\hanum\Desktop\fuseki\bin>ruby
--service=http://DESKTOP-JHTOKAN:3030/Netflix.csv
--query=C:\Users\hanum\Desktop\2c.rq>Q2-c.txt
```

(d) Query

```
prefix Netflix: <http://netflix.org//A4Q1//>
SELECT distinct ?movie1 ?movie2 ?director1 ?director2 ?duration FROM
<http://iiitd.ac.in/sweb/MT19123/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/MT19123/oldmoviesgraph>
Where {
  ?movie1 Netflix:directoris ?director1.
  OPTIONAL { ?movie2 Netflix:directoris ?director2 FILTER
((?director2=?director1) && (?movie1 != ?movie2) && ?director1!="""
&& ?director2!=""" ) }
  ?movie1 Netflix:durationis ?duration Filter ( ?duration>="60" && ?movie1 !=""
&& ?movie2 !="" )
}
```

Query command:-

```
C:\Users\hanum\Desktop\fuseki\bin>ruby
--service=http://DESKTOP-JHTOKAN:3030/Netflix.csv
--query=C:\Users\hanum\Desktop\2d.rq>Q2-d.txt
```

(e) Query

```
prefix Netflix: <http://netflix.org//A4Q1//>

SELECT DISTINCT ?Movie ?Country ?Rating ?genre ?duration FROM
<http://iiitd.ac.in/sweb/MT19123/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/MT19123/oldmoviesgraph>
Where {
  ?Movie Netflix:countryis ?Country . FILTER
(regex(?Country,"India")||regex(?Country,"United Kingdom")).
  ?Movie Netflix:ratingis ?Rating . FILTER regex(?Rating,"TV") .
  ?Movie Netflix:listed_inis ?genre . FILTER
(regex(?genre,"Documentaries")||regex(?genre,"Dramas")) .
  ?Movie Netflix:durationis ?duration . FILTER regex(?duration,"102 min") .
}
LIMIT 500
```

Query command:-

```
C:\Users\hanum\Desktop\fuseki\bin>ruby
--service=http://DESKTOP-JHTOKAN:3030/Netflix.csv
--query=C:\Users\hanum\Desktop\2e.rq>Q2-e.txt
```

s-query

```
C:\Users\hanum\Desktop\fuseki\bin>ruby s-query --service=http://DESKTOP-JHTOKAN:3030/Netflix.csv --query=C:\Users\hanum\Desktop\2d.rq>Q2-d.txt

C:\Users\hanum\Desktop\fuseki\bin>ruby s-query --service=http://DESKTOP-JHTOKAN:3030/Netflix.csv --query=C:\Users\hanum\Desktop\2a.rq>Q2-a.txt

C:\Users\hanum\Desktop\fuseki\bin>ruby s-query --service=http://DESKTOP-JHTOKAN:3030/Netflix.csv --query=C:\Users\hanum\Desktop\2b.rq>Q2-b.txt

C:\Users\hanum\Desktop\fuseki\bin>ruby s-query --service=http://DESKTOP-JHTOKAN:3030/Netflix.csv --query=C:\Users\hanum\Desktop\2c.rq>Q2-c.txt

C:\Users\hanum\Desktop\fuseki\bin>ruby s-query --service=http://DESKTOP-JHTOKAN:3030/Netflix.csv --query=C:\Users\hanum\Desktop\2e.rq>Q2-e.txt

C:\Users\hanum\Desktop\fuseki\bin>
```