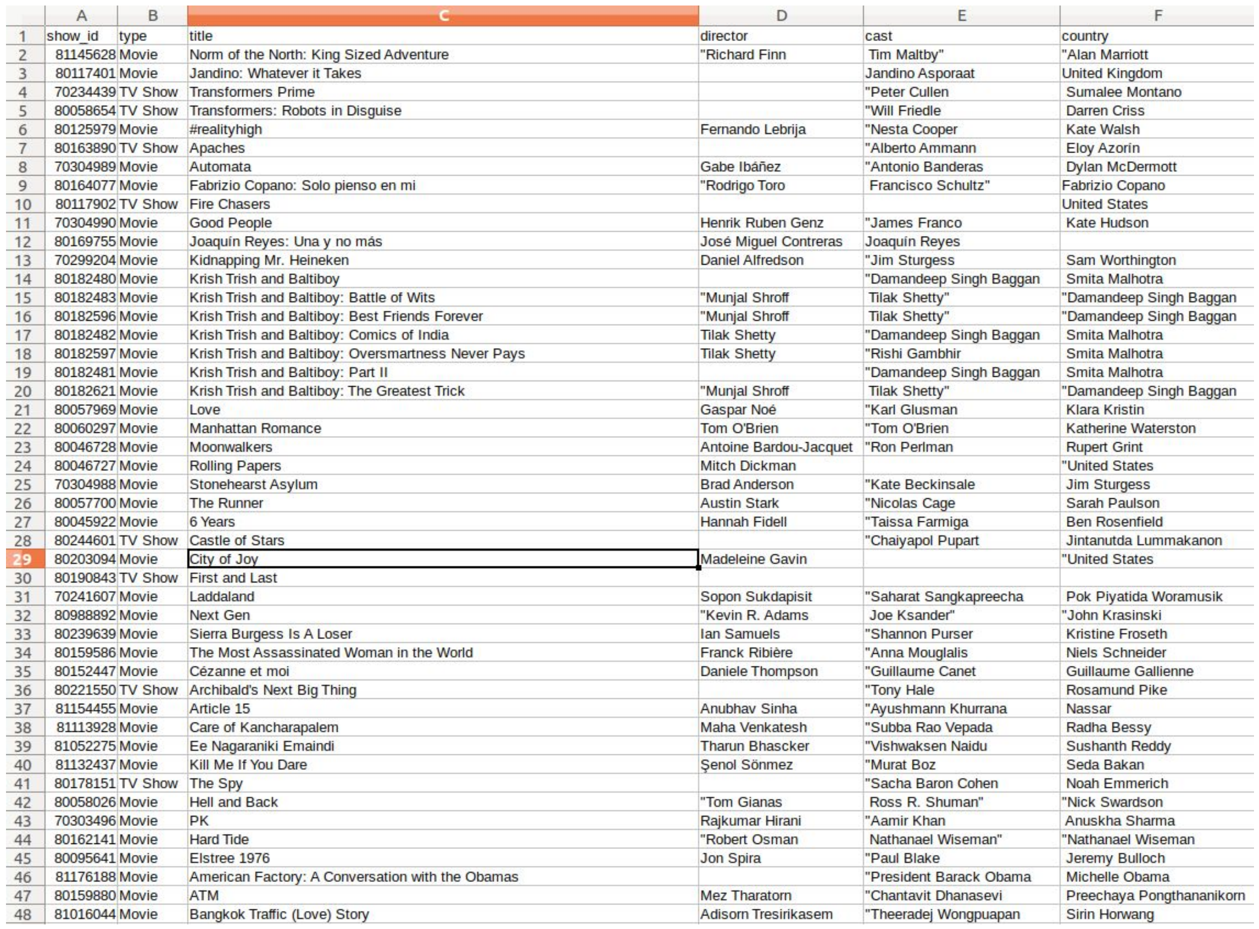
KNOWLEDGE GRAPH FOR NETFLIX DATASET

by Aman Roy

Step 1 :-

Analyzing the Netflix.csv data and creating the ontology.



	A	B	C	D	E	F
1	show_id	type	title	director	cast	country
2	81145628	Movie	Norm of the North: King Sized Adventure	"Richard Finn	Tim Maltby"	"Alan Marriott
3	80117401	Movie	Jandino: Whatever it Takes		Jandino Asporaat	United Kingdom
4	70234439	TV Show	Transformers Prime		"Peter Cullen	Sumalee Montano
5	80058654	TV Show	Transformers: Robots in Disguise		"Will Friedle	Darren Criss
6	80125979	Movie	#realityhigh	Fernando Lebrija	"Nesta Cooper	Kate Walsh
7	80163890	TV Show	Apaches		"Alberto Ammann	Eloy Azorín
8	70304989	Movie	Automata	Gabe Ibáñez	"Antonio Banderas	Dylan McDermott
9	80164077	Movie	Fabrizio Copano: Solo pienso en mi	"Rodrigo Toro	Francisco Schultz"	Fabrizio Copano
10	80117902	TV Show	Fire Chasers			United States
11	70304990	Movie	Good People	Henrik Ruben Genz	"James Franco	Kate Hudson
12	80169755	Movie	Joaquín Reyes: Una y no más	José Miguel Contreras	Joaquín Reyes	
13	70299204	Movie	Kidnapping Mr. Heineken	Daniel Alfredson	"Jim Sturgess	Sam Worthington
14	80182480	Movie	Krish Trish and Baltiboy		"Damandeep Singh Baggan	Smita Malhotra
15	80182483	Movie	Krish Trish and Baltiboy: Battle of Wits	"Munjal Shroff	Tilak Shetty"	"Damandeep Singh Baggan
16	80182596	Movie	Krish Trish and Baltiboy: Best Friends Forever	"Munjal Shroff	Tilak Shetty"	"Damandeep Singh Baggan
17	80182482	Movie	Krish Trish and Baltiboy: Comics of India	Tilak Shetty	"Damandeep Singh Baggan	Smita Malhotra
18	80182597	Movie	Krish Trish and Baltiboy: Oversmartness Never Pays	Tilak Shetty	"Rishi Gambhir	Smita Malhotra
19	80182481	Movie	Krish Trish and Baltiboy: Part II		"Damandeep Singh Baggan	Smita Malhotra
20	80182621	Movie	Krish Trish and Baltiboy: The Greatest Trick	"Munjal Shroff	Tilak Shetty"	"Damandeep Singh Baggan
21	80057969	Movie	Love	Gaspar Noé	"Karl Glusman	Klara Kristin
22	80060297	Movie	Manhattan Romance	Tom O'Brien	"Tom O'Brien	Katherine Waterston
23	80046728	Movie	Moonwalkers	Antoine Bardou-Jacquet	"Ron Perlman	Rupert Grint
24	80046727	Movie	Rolling Papers	Mitch Dickman		"United States
25	70304988	Movie	Stonehearst Asylum	Brad Anderson	"Kate Beckinsale	Jim Sturgess
26	80057700	Movie	The Runner	Austin Stark	"Nicolas Cage	Sarah Paulson
27	80045922	Movie	6 Years	Hannah Fidell	"Taissa Farmiga	Ben Rosenfield
28	80244601	TV Show	Castle of Stars		"Chaiyapol Pupart	Jintanutda Lummakanon
29	80203094	Movie	City of Joy	Madeleine Gavin		"United States
30	80190843	TV Show	First and Last			
31	70241607	Movie	Laddaland	Sopon Sukdapisit	"Saharat Sangkapreecha	Pok Piyatida Woramusik
32	80988892	Movie	Next Gen	"Kevin R. Adams	Joe Ksander"	"John Krasinski
33	80239639	Movie	Sierra Burgess Is A Loser	Ian Samuels	"Shannon Purser	Kristine Froseth
34	80159586	Movie	The Most Assassinated Woman in the World	Franck Ribière	"Anna Mouglalis	Niels Schneider
35	80152447	Movie	Cézanne et moi	Daniele Thompson	"Guillaume Canet	Guillaume Gallienne
36	80221550	TV Show	Archibald's Next Big Thing		"Tony Hale	Rosamund Pike
37	81154455	Movie	Article 15	Anubhav Sinha	"Ayushmann Khurrana	Nassar
38	81113928	Movie	Care of Kancharapalem	Maha Venkatesh	"Subba Rao Vepada	Radha Bessy
39	81052275	Movie	Ee Nagaraniki Emaindi	Tharun Bhascker	"Vishwaksen Naidu	Sushanth Reddy
40	81132437	Movie	Kill Me If You Dare	Şenol Sönmez	"Murat Boz	Seda Bakan
41	80178151	TV Show	The Spy		"Sacha Baron Cohen	Noah Emmerich
42	80058026	Movie	Hell and Back	"Tom Gianas	Ross R. Shuman"	"Nick Swardson
43	70303496	Movie	PK	Rajkumar Hirani	"Aamir Khan	Anuskha Sharma
44	80162141	Movie	Hard Tide	"Robert Osman	Nathanael Wiseman"	"Nathanael Wiseman
45	80095641	Movie	Elstree 1976	Jon Spira	"Paul Blake	Jeremy Bulloch
46	81176188	Movie	American Factory: A Conversation with the Obamas		"President Barack Obama	Michelle Obama
47	80159880	Movie	ATM	Mez Tharatorn	"Chantavit Dhanasevi	Preechaya Pongthananikorn
48	81016044	Movie	Bangkok Traffic (Love) Story	Adisorn Tresirikasem	"Theeradej Wongpuapan	Sirin Horwang

Ontology Design

- Id
  - movieId
  - showId
- Movie
- Show
- NetflixContent
- Title
- Director
- Cast
- Country

- Date
- releaseYear
- Rating
- Duration
    - timeDuration
    - showDuration
- genre
- Description

Note :- All in red leads to Data Property while in black are Concepts

Classes :-

	Rigid	**Unity**	**Identity**
Movie	**Yes**	**Yes**	**Yes**
Show	**Yes**	**Yes**	**Yes**
NetflixContent	**Yes**	**Yes**	**Yes**
Director	**No**	**Yes**	**Yes**
Cast	**No**	**Yes**	**Yes**
Country	**Yes**	**Yes**	**Yes**
Rating	**Yes**	**Yes**	**No**
Genre	**Yes**	**Yes**	**No**

Properties :-

Property	Characteristics(Inverse Relation)	Domain	Range	Rigidity
hasId	-	NetflixContent	String	
hasMovieId	-	Movie	String	
hasShowId	-	Show	String	
hasDirectorRole	isDirectorRoleOf	NetflixContent	DirectorRole	

hasCastRole	isCastRoleOf	NetflixContent	CastRole	
isReleasedInCountry	isReleasedCountryOf	NetflixContent	Country	
hasDuration	-	NetflixContent	String/Int	
hasTimeDuration	-	Movie	Int	
hasShowDuration	-	Show	String	
hasRated	isRatingOf	NetflixContent	Rating	
hasDate	-	NetflixContent	DateTime	
hasGenre	isGenreOf	NetflixContent	genre	
hasDescription	-	NetflixContent	String	
hasCasted	isCastedBy	Cast	CastRole	
hasDirected	isDirectedBy	Director	DirectorRole	
hasName	-	Cast/Director	String	

Note :- **All in red are data property while in black are object property.**

TBoxes Statements :-

1. A director is someone who hasDirected some DirectorRole.
2. A cast is someone who hasCasted some DirectorRole.
3. Every movie has one MovieId.
4. Every show has one ShowId.
5. A NetflixContent is something that has atleast one DirectorRole and atleast one CastRole
6. Movie and Show are disjoint classes.
7. Every NetflixContent has atleast one genre.
8. DirectorRole and CastRole are disjoint.
9. Every NetflixContent has some country where it is released.
10. Every NetflixContent has only one rating.

ODPs Used

Two type of ODPs are used namely :-

1. Agent-Role ODP

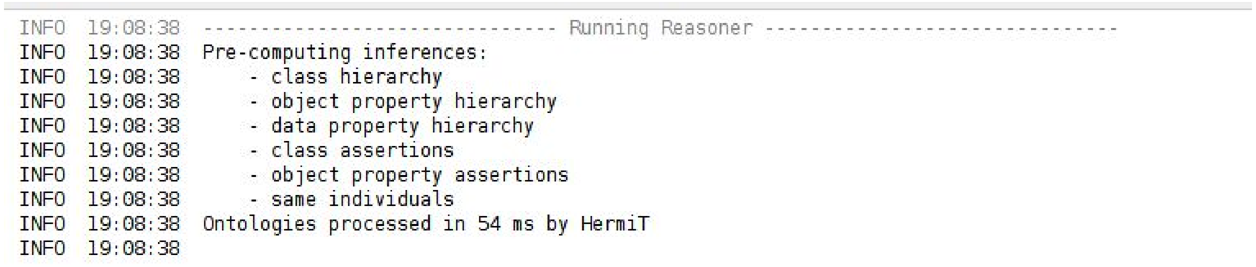
    a. Agents are director and cast which performs role of director role and cast role respectively.
2. Event-ODP
    a. Agent Role ODP is combined with Event ODP where Event is movie and show which has both location and duration information.

Hierarchy Explanation :-

1. Every netflix content will always be a netflix content and is either a movie or a show. Also all movies and shows will always sustain to be movies and shows. This is the reason why movies and shows are subclasses of netflix content.
2. Every movie or show has data properties which are listed in terms of various data properties as show in red in the table.
3. Apart from that every other class has no subclasses and also provides information regarding movies and shows.
4. The ontology has primarily designed in order to answer the below competency questions which it fulfills :-

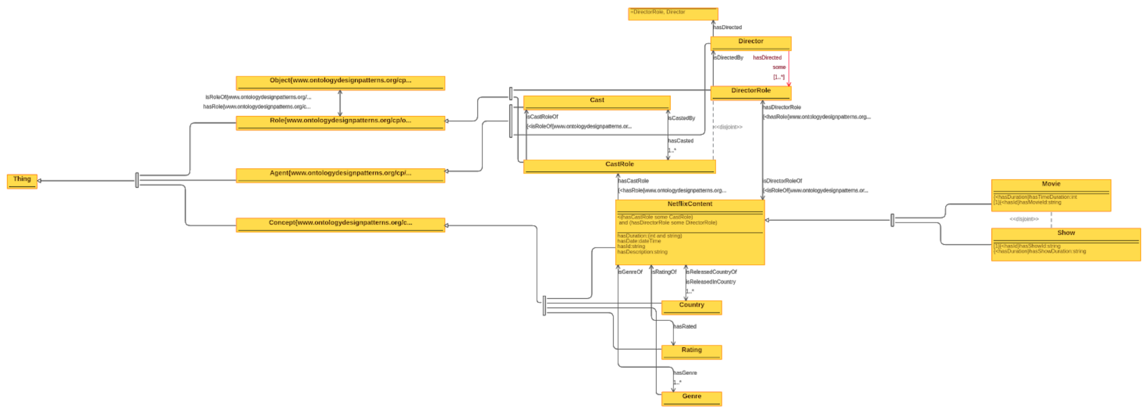
Competence Questions :-
1. List all the DirectorRole of a Director ?
2. List all the movies directed by a director ?
3. List all the movies or shows in which a cast was casted?
4. List all the movies or shows which was directed by a director ?
5. List all the movies released in a country ?

Also below is the validity of ontology after running through Hermit Reasoner.



```
INFO  19:08:38  ------------------------------ Running Reasoner ------------------------------
INFO  19:08:38  Pre-computing inferences:
INFO  19:08:38      - class hierarchy
INFO  19:08:38      - object property hierarchy
INFO  19:08:38      - data property hierarchy
INFO  19:08:38      - class assertions
INFO  19:08:38      - object property assertions
INFO  19:08:38      - same individuals
INFO  19:08:38  Ontologies processed in 54 ms by HermiT
INFO  19:08:38
```

The Explanation Diagram



=DirectorRole, Director
hasDirected
Director
isDirectedBy
hasDirected
some
[1..*]
DirectorRole
Object{www.ontologydesignpatterns.org/cp...
isRoleOf{www.ontologydesignpatterns.org/...
hasRole{www.ontologydesignpatterns.org/c...
Role{www.ontologydesignpatterns.org/cp/o...
Cast
isCastRoleOf
{<isRoleOf{www.ontologydesignpatterns.or...
isCastedBy
hasCasted
1..*
<<disjoint>>
hasDirectorRole
{<hasRole{www.ontologydesignpatterns.org...
Thing
Agent{www.ontologydesignpatterns.org/cp/...
CastRole
hasCastRole
{<hasRole{www.ontologydesignpatterns.org...
isDirectorRoleOf
{<isRoleOf{www.ontologydesignpatterns.or...
NetflixContent
<(hasCastRole some CastRole) and (hasDirectorRole some DirectorRole)
hasDuration:(int and string)
hasDate:dateTime
hasId:string
hasDescription:string
Movie
{<hasDuration}hasTimeDuration:int
[1]{<hasId}hasMovieId:string
<<disjoint>>
Show
[1]{<hasId}hasShowId:string
{<hasDuration}hasShowDuration:string
Concept{www.ontologydesignpatterns.org/c...
isGenreOf
isRatingOf
isReleasedCountryOf
isReleasedInCountry
1..*
Country
hasRated
Rating
hasGenre
1..*
Genre

Ontology file is Netflix.owl. (Can be opened using Protege).

Step 2 :-
Creating the knowledge graph from Netflix.owl

Add all the libraries and jar files in a workspace and run Create_KG.java which should generate a knowledge graph file named “Ontology.ttl”.

Step 3 :- Linking the dataset to other datasets and creating a five star graph.

a) *Here are the individuals which have been linked with other datasets like “geonames” and “dbpedia”*

"a" , "b"
"http://dbpedia.org/resource/Aamir_Khan" ,
"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Aamir_Khan"

"http://dbpedia.org/resource/Adam_Conover" ,
"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Adam_Conover"

"http://dbpedia.org/resource/Anushka_Sharma" ,
"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Anushka_Sharma"

"http://dbpedia.org/resource/Ayushmann_Khurrana" ,
"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Ayushmann_Khurrana"

"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Denmark" ,
"https://www.geonames.org/countries/DK/denmark.html"

"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#India" , "https://www.geonames.org/countries/IN/india.html"

"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Indonesia" , "https://www.geonames.org/countries/ID/indonesia.html"

"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Ireland" , "https://www.geonames.org/countries/IE/ireland.html"

"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#Pakistan" , "https://www.geonames.org/countries/PK/pakistan.html"

"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#United_Kingdom" , "https://www.geonames.org/countries/GB/united-kingdom.html"

"http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#United_States" , "https://www.geonames.org/countries/US/united-states.html".

Linking can be done using protege by below methods :-

1. **owl:sameAs (adding extra dbpedia or geonames IRI for every individual)**
2. **By changing the IRI of individuals to dbpedia or geonames IRI.**

The above KG is a 5 star linked dataset as it has :-

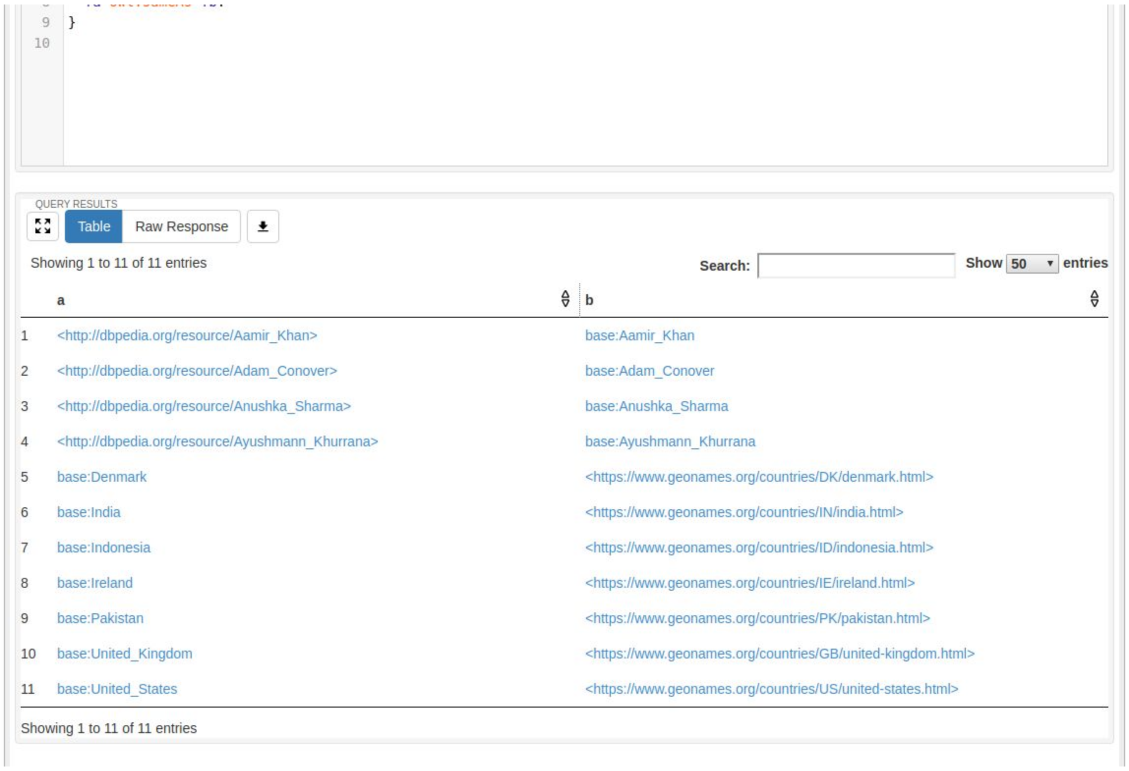
1. **Sparql endpoints.**
2. **Is in RDF format.**
3. **Linked with at least two external datasets i.e. dbpedia and geonames.**

b) will generate the ontology file named ontology.ttl.

Step 4 :- Setup Of Fuseki Server for SparQL query

Run Fuseki bash script and upload knowledge graphs to run SparQL queries. The named graphs and the fuseki server running on hostname.

Here is the screenshot of the same



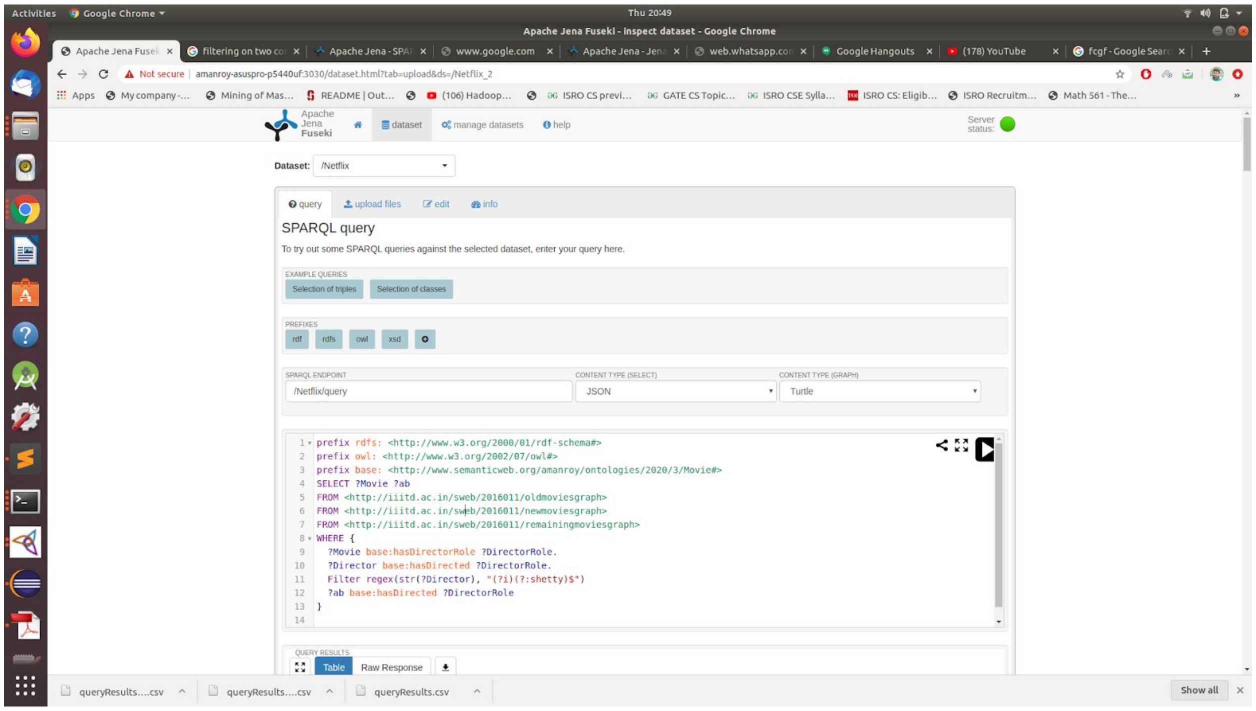
```
}
```

QUERY RESULTS

Showing 1 to 11 of 11 entries

	a	b
1	<http://dbpedia.org/resource/Aamir_Khan>	base:Aamir_Khan
2	<http://dbpedia.org/resource/Adam_Conover>	base:Adam_Conover
3	<http://dbpedia.org/resource/Anushka_Sharma>	base:Anushka_Sharma
4	<http://dbpedia.org/resource/Ayushmann_Khurrana>	base:Ayushmann_Khurrana
5	base:Denmark	<https://www.geonames.org/countries/DK/denmark.html>
6	base:India	<https://www.geonames.org/countries/IN/india.html>
7	base:Indonesia	<https://www.geonames.org/countries/ID/indonesia.html>
8	base:Ireland	<https://www.geonames.org/countries/IE/ireland.html>
9	base:Pakistan	<https://www.geonames.org/countries/PK/pakistan.html>
10	base:United_Kingdom	<https://www.geonames.org/countries/GB/united-kingdom.html>
11	base:United_States	<https://www.geonames.org/countries/US/united-states.html>

Showing 1 to 11 of 11 entries



Apache Jena Fuseki - inspect dataset - Google Chrome

amanroy-asuspro-p5440uf:3030/dataset.html?tab=upload&ds=/Netflix_2

Dataset: /Netflix

SPARQL query

To try out some SPARQL queries against the selected dataset, enter your query here.

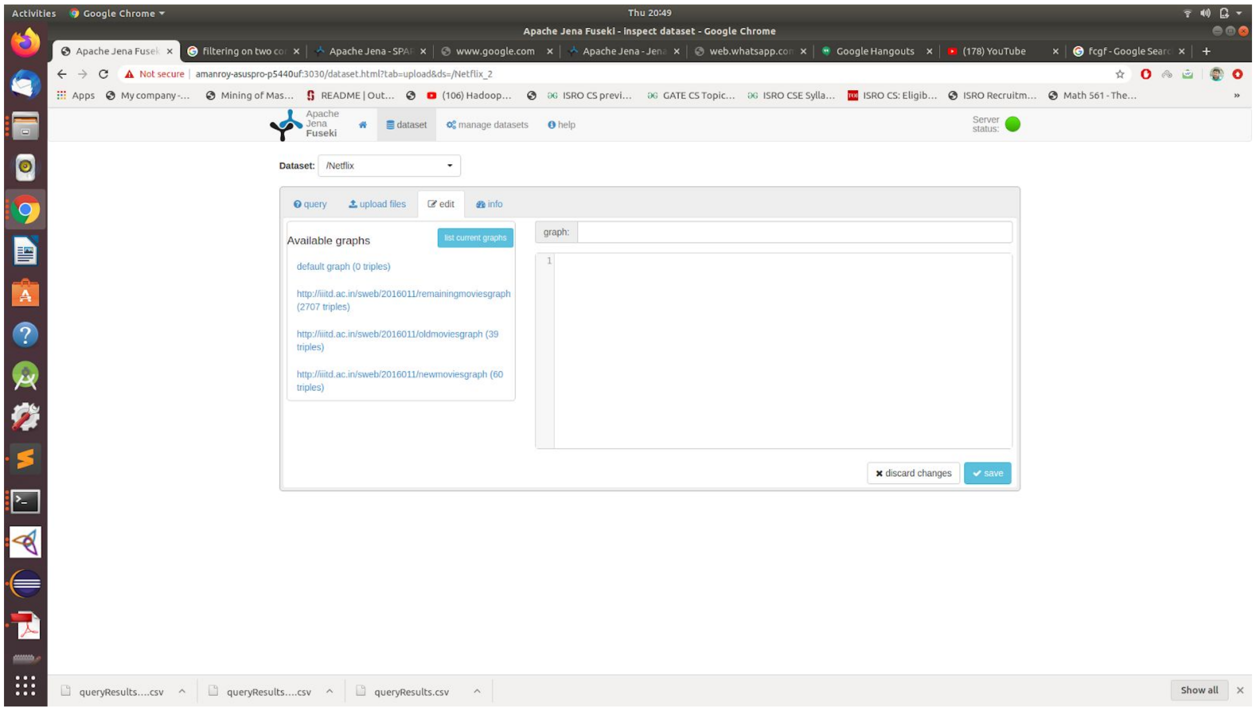
SPARQL ENDPOINT: /Netflix/query

CONTENT TYPE (SELECT): JSON

CONTENT TYPE (GRAPH): Turtle

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix base: <http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#>
SELECT ?Movie ?ab
FROM <http://iiitd.ac.in/sweb/2016011/oldmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph>
WHERE {
  ?Movie base:hasDirectorRole ?DirectorRole.
  ?Director base:hasDirected ?DirectorRole.
  Filter regex(str(?Director), "(?i)(?:shetty)$")
  ?ab base:hasDirected ?DirectorRole
}
```

Step 5 :- Splitting Graph into Named Graphs


Apache Jena Fuseki - inspect dataset - Google Chrome
amanroy-asuspro-p5440uf:3030/dataset.html?tab=upload&ds=/Netflix_2
Apache Jena Fuseki
dataset
manage datasets
help
Server status:
Dataset: /Netflix
query
upload files
edit
info
Available graphs
list current graphs
default graph (0 triples)
http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph (2707 triples)
http://iiitd.ac.in/sweb/2016011/oldmoviesgraph (39 triples)
http://iiitd.ac.in/sweb/2016011/newmoviesgraph (60 triples)
graph:
discard changes
save
queryResults....csv
queryResults....csv
queryResults.csv
Show all

I have created three named graphs i.e.

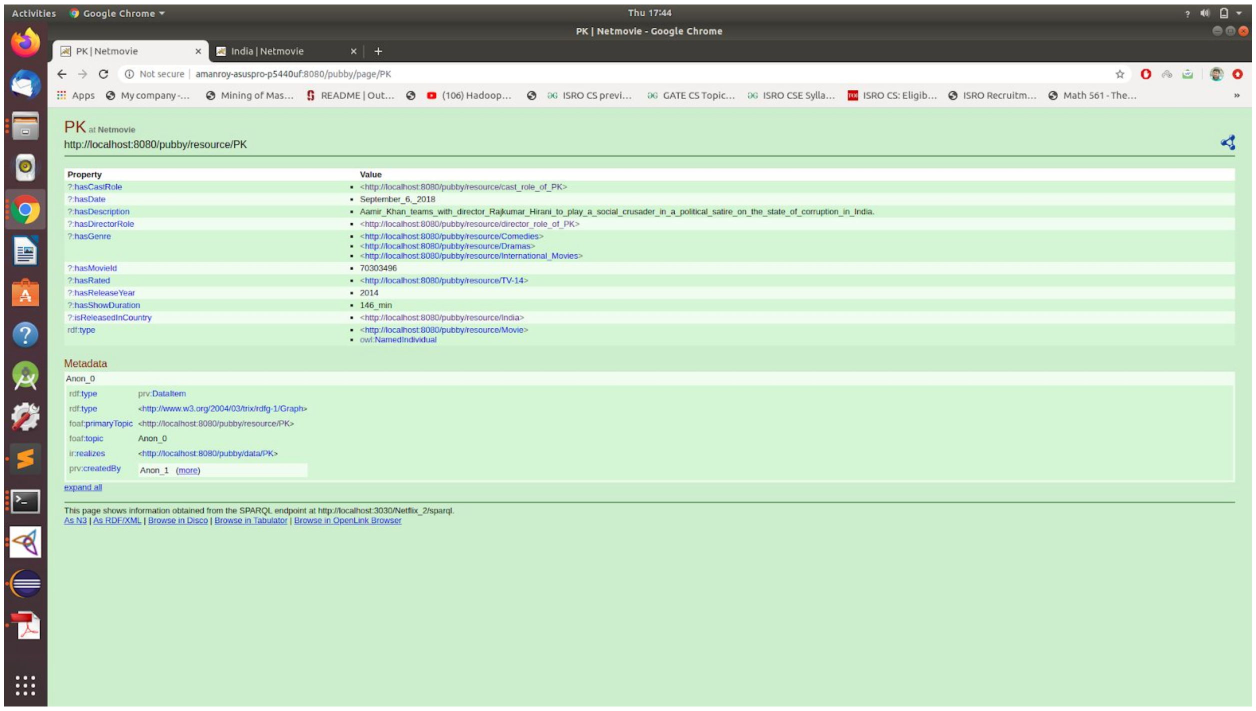
i) **<http://iiitd.ac.in/sweb/2016011/oldmoviesgraph>**

ii) **<http://iiitd.ac.in/sweb/2016011/newmoviesgraph>**

iii) **<http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph>**

The first two graphs contain triples as defined in the question and the remaining graphs contain all the triples which are not present in the first two graphs. Also after moving triples to these named graphs, default graphs become empty at the end.

Step 6 :- Give knowledge Graph a front end using Pubby


PK | Netmovie - Google Chrome
PK | Netmovie
India | Netmovie
Not secure
amanroy-asuspro-p5440uf:8080/pubby/page/PK

PK at Netmovie
http://localhost:8080/pubby/resource/PK

Property	Value
?:hasCastRole	<http://localhost:8080/pubby/resource/cast_role_of_PK>
?:hasDate	September_6,_2018
?:hasDescription	Aamir_Khan_teams_with_director_Rajkumar_Hirani_to_play_a_social_crusader_in_a_political_satire_on_the_state_of_corruption_in_India.
?:hasDirectorRole	<http://localhost:8080/pubby/resource/director_role_of_PK>
?:hasGenre	<http://localhost:8080/pubby/resource/Comedies> <http://localhost:8080/pubby/resource/Dramas> <http://localhost:8080/pubby/resource/International_Movies>
?:hasMovieId	70303496
?:hasRated	<http://localhost:8080/pubby/resource/TV-14>
?:hasReleaseYear	2014
?:hasShowDuration	146_min
?:isReleasedInCountry	<http://localhost:8080/pubby/resource/India>
rdf:type	<http://localhost:8080/pubby/resource/Movie> owl:NamedIndividual

Metadata

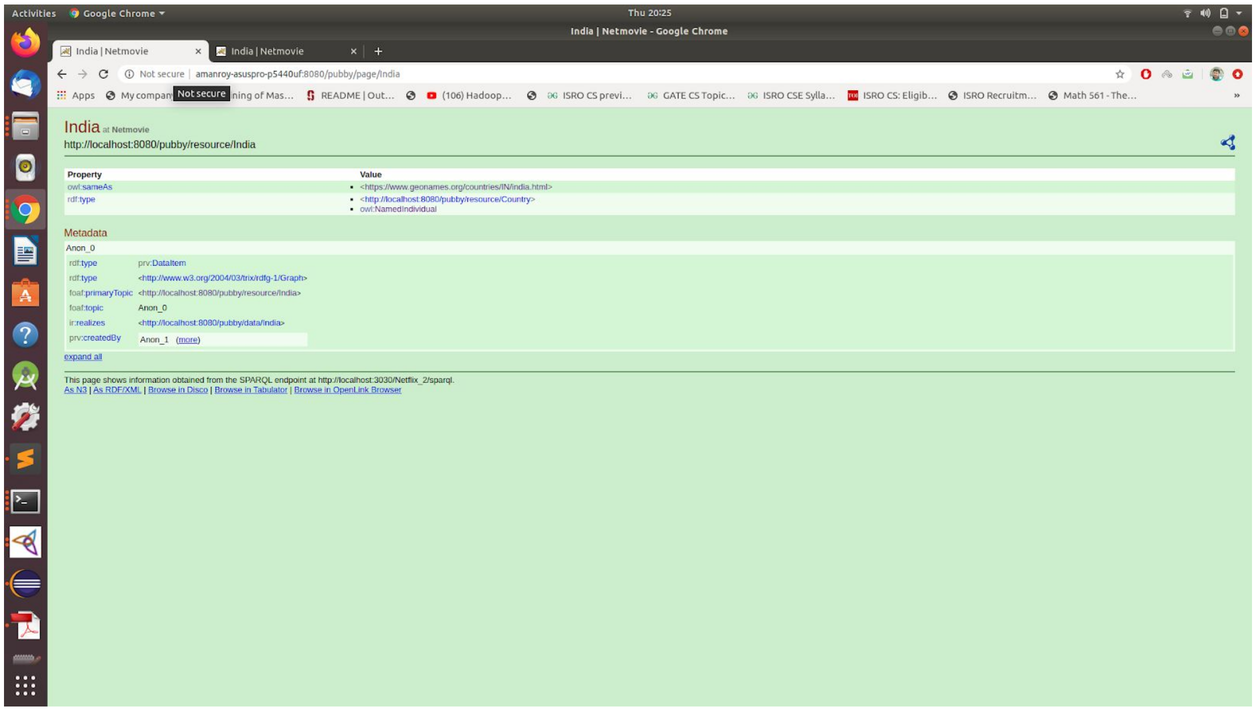
Anon_0

rdf:type	prv:DataItem
rdf:type	<http://www.w3.org/2004/03/trix/rdfg-1/Graph>
foaf:primaryTopic	<http://localhost:8080/pubby/resource/PK>
foaf:topic	Anon_0
ir:realizes	<http://localhost:8080/pubby/data/PK>
prv:createdBy	Anon_1 (more)

expand all

This page shows information obtained from the SPARQL endpoint at http://localhost:3030/Netflix_2/sparql.
As N3 | As RDF/XML | Browse in Disco | Browse in Tabulator | Browse in OpenLink Browser

IsreleasedInCountry **India** is linked to Geonames.org link


India | Netmovie - Google Chrome
India | Netmovie
India | Netmovie
Not secure
amanroy-asuspro-p5440uf:8080/pubby/page/India

India at Netmovie
http://localhost:8080/pubby/resource/India

Property	Value
owl:sameAs	<https://www.geonames.org/countries/IN/india.html>
rdf:type	<http://localhost:8080/pubby/resource/Country> owl:NamedIndividual

Metadata

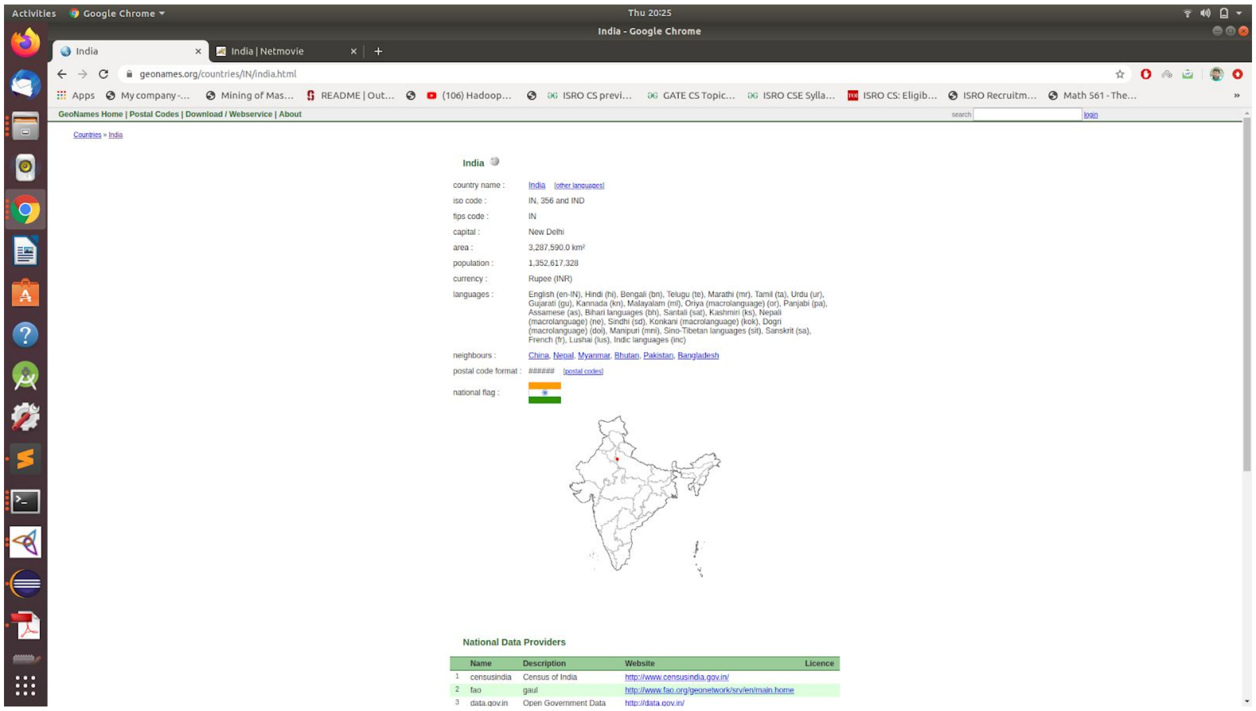
Anon_0

rdf:type	prv:DataItem
rdf:type	<http://www.w3.org/2004/03/trix/rdfg-1/Graph>
foaf:primaryTopic	<http://localhost:8080/pubby/resource/India>
foaf:topic	Anon_0
ir:realizes	<http://localhost:8080/pubby/data/India>
prv:createdBy	Anon_1 (more)

expand all

This page shows information obtained from the SPARQL endpoint at http://localhost:3030/Netflix_2/sparql.
As N3 | As RDF/XML | Browse in Disco | Browse in Tabulator | Browse in OpenLink Browser

Geoname link takes to Geoname page.



India - Google Chrome

geonames.org/countries/IN/india.html

GeoNames Home | Postal Codes | Download / Webservice | About

Countries » India

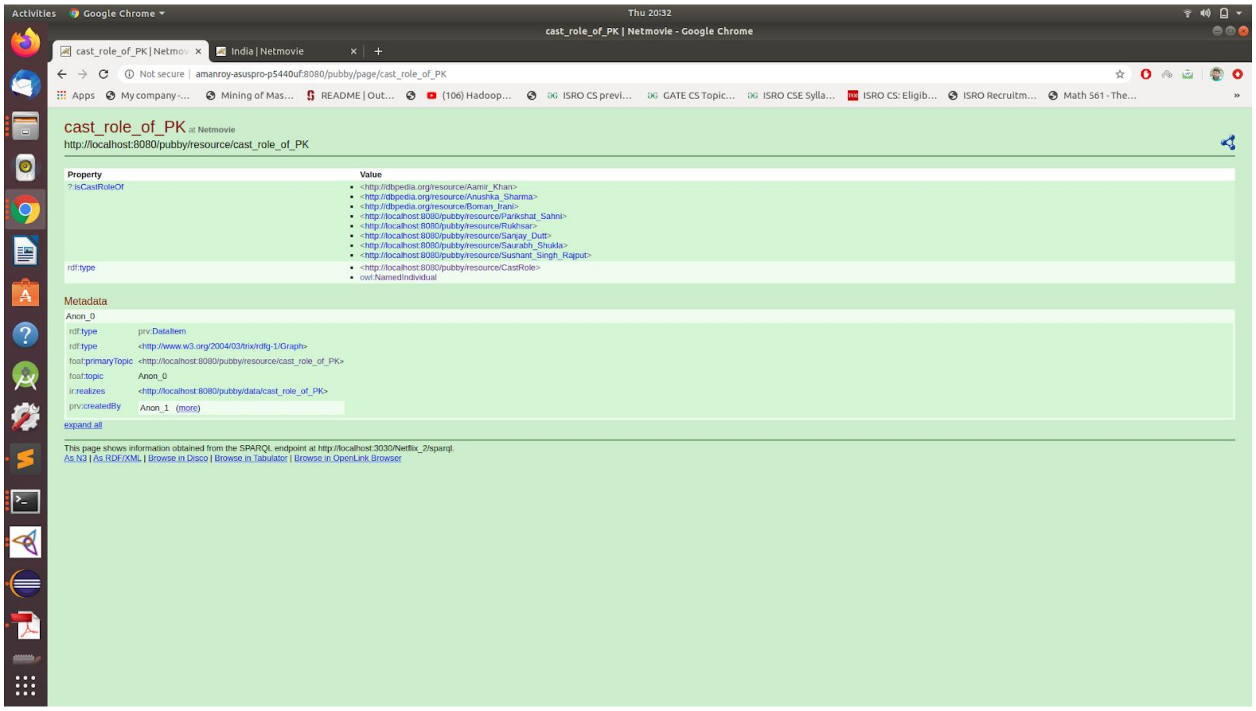
India

country name :	India [other languages]
iso code :	IN, 356 and IND
fips code :	IN
capital :	New Delhi
area :	3,287,590.0 km²
population :	1,352,617,328
currency :	Rupee (INR)
languages :	English (en-IN), Hindi (hi), Bengali (bn), Telugu (te), Marathi (mr), Tamil (ta), Urdu (ur), Gujarati (gu), Kannada (kn), Malayalam (ml), Oriya (macrolanguage) (or), Panjabi (pa), Assamese (as), Bihari languages (bh), Santali (sat), Kashmiri (ks), Nepali (macrolanguage) (ne), Sindhi (sd), Konkani (macrolanguage) (kok), Dogri (macrolanguage) (doi), Manipuri (mni), Sino-Tibetan languages (sit), Sanskrit (sa), French (fr), Lushai (lus), Indic languages (inc)
neighbours :	China, Nepal, Myanmar, Bhutan, Pakistan, Bangladesh
postal code format :	###### [postal codes]
national flag :	

National Data Providers

	Name	Description	Website	Licence
1	censusindia	Census of India	http://www.censusindia.gov.in/	
2	fao	gaul	http://www.fao.org/geonetwork/srv/en/main.home	
3	data.gov.in	Open Government Data	http://data.gov.in/	

hasCastRole is linked to CastRole which has dbpedia entries for different actors



cast_role_of_PK | Netmovie - Google Chrome

Not secure | amanroy-asuspro-p5440uf:8080/pubby/page/cast_role_of_PK

cast_role_of_PK at Netmovie
http://localhost:8080/pubby/resource/cast_role_of_PK

Property	Value
?:isCastRoleOf	<http://dbpedia.org/resource/Aamir_Khan> <http://dbpedia.org/resource/Anushka_Sharma> <http://dbpedia.org/resource/Boman_Irani> <http://localhost:8080/pubby/resource/Parikshat_Sahni> <http://localhost:8080/pubby/resource/Rukhsar> <http://localhost:8080/pubby/resource/Sanjay_Dutt> <http://localhost:8080/pubby/resource/Saurabh_Shukla> <http://localhost:8080/pubby/resource/Sushant_Singh_Rajput>
rdf:type	<http://localhost:8080/pubby/resource/CastRole> owl:NamedIndividual

Metadata

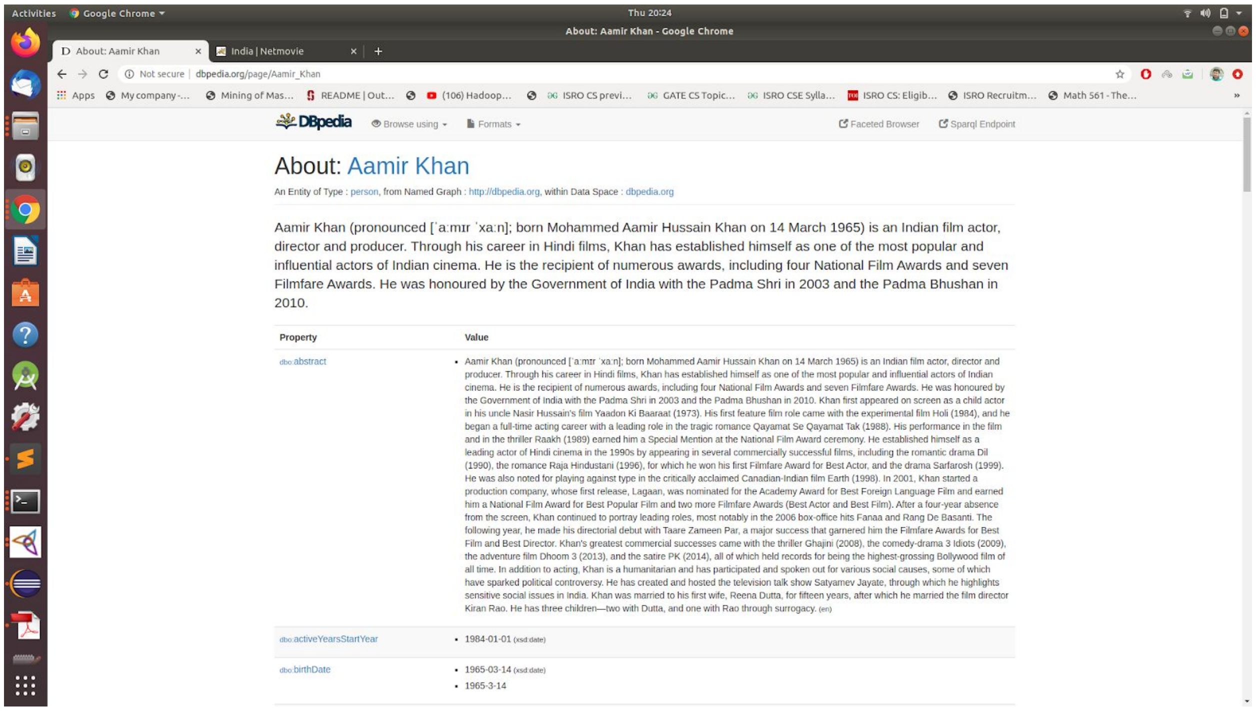
Anon_0

rdf:type	prv:DataItem
rdf:type	<http://www.w3.org/2004/03/trix/rdfg-1/Graph>
foaf:primaryTopic	<http://localhost:8080/pubby/resource/cast_role_of_PK>
foaf:topic	Anon_0
ir:realizes	<http://localhost:8080/pubby/data/cast_role_of_PK>
prv:createdBy	Anon_1 (more)

expand all

This page shows information obtained from the SPARQL endpoint at http://localhost:3030/Netflix_2/sparql.
As N3 | As RDF/XML | Browse in Disco | Browse in Tabulator | Browse in OpenLink Browser

Clicking on Aamir Khan link opens dbpedia page of Aamir Khan


About: Aamir Khan

An Entity of Type : person, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

Aamir Khan (pronounced [ˈaːmɪr ˈxaːn]; born Mohammed Aamir Hussain Khan on 14 March 1965) is an Indian film actor, director and producer. Through his career in Hindi films, Khan has established himself as one of the most popular and influential actors of Indian cinema. He is the recipient of numerous awards, including four National Film Awards and seven Filmfare Awards. He was honoured by the Government of India with the Padma Shri in 2003 and the Padma Bhushan in 2010.

Property	Value
dbo:abstract	• Aamir Khan (pronounced [ˈaːmɪr ˈxaːn]; born Mohammed Aamir Hussain Khan on 14 March 1965) is an Indian film actor, director and producer. Through his career in Hindi films, Khan has established himself as one of the most popular and influential actors of Indian cinema. He is the recipient of numerous awards, including four National Film Awards and seven Filmfare Awards. He was honoured by the Government of India with the Padma Shri in 2003 and the Padma Bhushan in 2010. Khan first appeared on screen as a child actor in his uncle Nasir Hussain's film Yaadon Ki Baaraat (1973). His first feature film role came with the experimental film Holi (1984), and he began a full-time acting career with a leading role in the tragic romance Qayamat Se Qayamat Tak (1988). His performance in the film and in the thriller Raakh (1989) earned him a Special Mention at the National Film Award ceremony. He established himself as a leading actor of Hindi cinema in the 1990s by appearing in several commercially successful films, including the romantic drama Dil (1990), the romance Raja Hindustani (1996), for which he won his first Filmfare Award for Best Actor, and the drama Sarfarosh (1999). He was also noted for playing against type in the critically acclaimed Canadian-Indian film Earth (1998). In 2001, Khan started a production company, whose first release, Lagaan, was nominated for the Academy Award for Best Foreign Language Film and earned him a National Film Award for Best Popular Film and two more Filmfare Awards (Best Actor and Best Film). After a four-year absence from the screen, Khan continued to portray leading roles, most notably in the 2006 box-office hits Fanaa and Rang De Basanti. The following year, he made his directorial debut with Taare Zameen Par, a major success that garnered him the Filmfare Awards for Best Film and Best Director. Khan's greatest commercial successes came with the thriller Ghajini (2008), the comedy-drama 3 Idiots (2009), the adventure film Dhoom 3 (2013), and the satire PK (2014), all of which held records for being the highest-grossing Bollywood film of all time. In addition to acting, Khan is a humanitarian and has participated and spoken out for various social causes, some of which have sparked political controversy. He has created and hosted the television talk show Satyamev Jayate, through which he highlights sensitive social issues in India. Khan was married to his first wife, Reena Dutta, for fifteen years, after which he married the film director Kiran Rao. He has three children—two with Dutta, and one with Rao through surrogacy. (en)
dbo:activeYearsStartYear	• 1984-01-01 (xsd:date)
dbo:birthDate	• 1965-03-14 (xsd:date) • 1965-3-14

These images show that front-end is working succesfully.

<u>Bonus Part</u>

Some queries to run on the above graphs on Fusceki Server

a)

Copy the query in query.rq file and run the command in fusceki's bin folder.

Run the command :-

sudo ruby ./s-query --service=http://amanroy-asuspro-p5440uf:3030/Netflix --query=query.rq >> Q2-a.txt

Query
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix base: <http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#>
SELECT ?Movie ?ab
FROM <http://iiitd.ac.in/sweb/2016011/oldmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/newmoviesgraph>

```
FROM <http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph>
WHERE {
  ?Movie base:hasDirectorRole ?DirectorRole.
  ?Director base:hasDirected ?DirectorRole.
  Filter regex(str(?Director), "(?i)(?:shetty)$")
  ?ab base:hasDirected ?DirectorRole
}
```

b)

Copy the query in query.rq file and run the command in fusceki's bin folder.

Run the command :-

```
sudo ruby ./s-query --service=http://amanroy-asuspro-p5440uf:3030/Netflix --query=query.rq >>
Q2-b.txt
```

Query

```
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix base: <http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#>
SELECT distinct ?Movie ?Genre
FROM <http://iiitd.ac.in/sweb/2016011/oldmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph>
WHERE {
  ?Movie base:hasGenre ?Genre.
  Filter regex(str(?Genre), "(?i)(?:Comedies|comedy|drama)")
}
```

c)

Copy the query in query.rq file and run the command in fusceki's bin folder.

Run the command :-

```
sudo ruby ./s-query --service=http://amanroy-asuspro-p5440uf:3030/Netflix --query=query.rq >>
Q2-c.txt
```

Query

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix base: <http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#>
```

```
SELECT ?Movie ?Year ?country
FROM <http://iiitd.ac.in/sweb/2016011/oldmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph>
WHERE {
  ?Movie base:hasReleaseYear ?Year.
  ?Movie base:isReleasedInCountry ?country.
  ?Movie base:hasDescription ?description.
  Filter (xsd:int(?Year) > 2010 && xsd:int(?Year) < 2020 && regex(str(?description),
"(?i)(?:couple)") && regex(str(?country), "(?i)(?:united_state)") )
}
```

d)

Copy the query in query.rq file and run the command in fusceki's bin folder.

Run the command :-

```
sudo ruby ./s-query --service=http://amanroy-asuspro-p5440uf:3030/Netflix --query=query.rq >>
Q2-d.txt
```

Query

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

prefix owl: <http://www.w3.org/2002/07/owl#>
prefix base: <http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#>
SELECT ?Movie ?Movie1 ?Director ?Director1 ?time ?time1
FROM <http://iiitd.ac.in/sweb/2016011/oldmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph>
WHERE {
  ?Movie base:hasDirectorRole ?DirectorRole.
  ?Director base:hasDirected ?DirectorRole.
  ?Movie1 base:hasDirectorRole ?DirectorRole1.
  ?Director1 base:hasDirected ?DirectorRole1.
  ?Movie base:hasTimeDuration ?time.
  ?Movie1 base:hasTimeDuration ?time1
  Filter (xsd:int(?time) >= 60 && xsd:int(?time1) >= 60 &&
str(?DirectorRole)!=str(?DirectorRole1) && str(?Movie) > str(?Movie1) && str(?Director) =
str(?Director1))
}
```

e)

Copy the query in query.rq file and run the command in fusceki's bin folder.

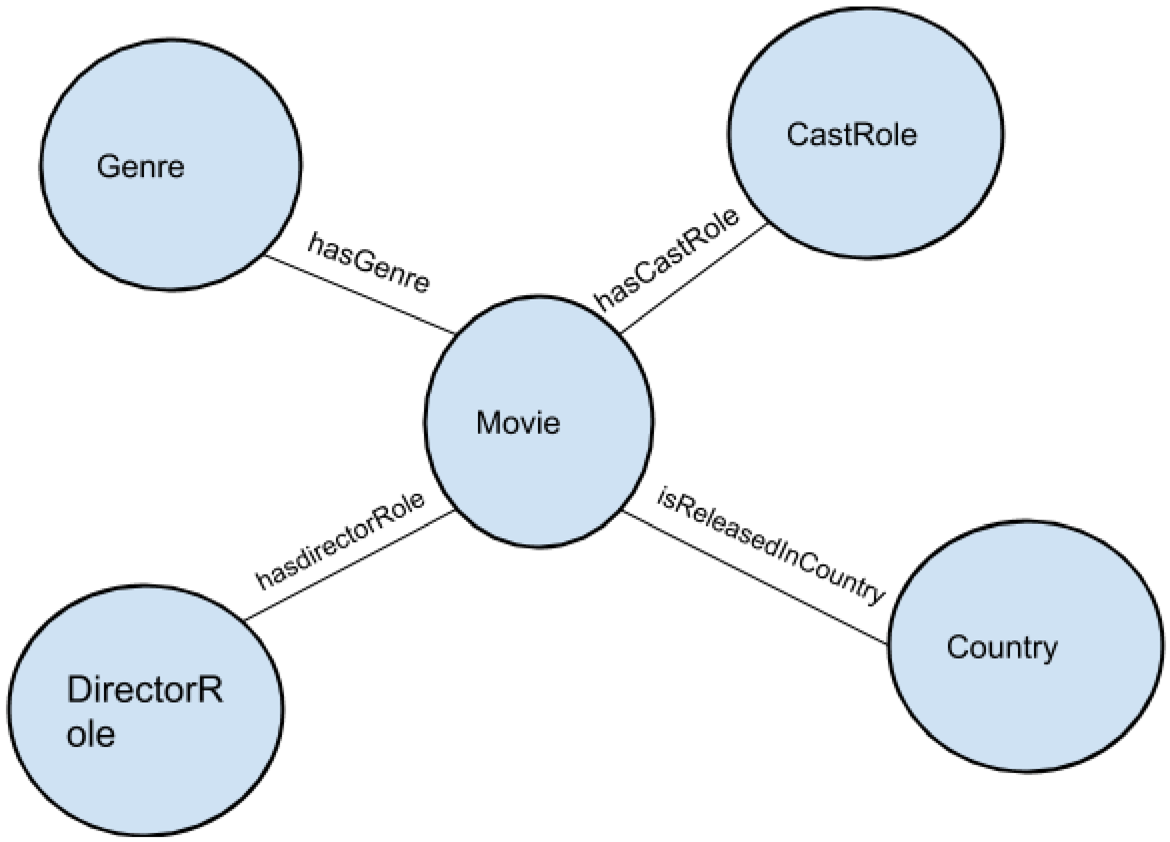
Run the command :-

```
sudo ruby ./s-query --service=http://amanroy-asuspro-p5440uf:3030/Netflix --query=query.rq >>
Q2-e.txt
```

Query

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix base: <http://www.semanticweb.org/amanroy/ontologies/2020/3/Movie#>
SELECT distinct ?Movie ?CastRole ?DirectorRole ?Country ?Genre
FROM <http://iiitd.ac.in/sweb/2016011/oldmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/newmoviesgraph>
FROM <http://iiitd.ac.in/sweb/2016011/remainingmoviesgraph>
WHERE {
  ?Movie base:hasCastRole ?CastRole.
  ?Movie base:hasDirectorRole ?DirectorRole.
  ?Movie base:isReleasedInCountry ?Country.
  ?Movie base:hasGenre ?Genre.
}
```

This is the diagram for the star pattern.


Genre
CastRole
hasGenre
hasCastRole
Movie
hasdirectorRole
isReleasedInCountry
DirectorRole
Country