

Knowledge Graph Analysis

Exercise Sheet 1

Dr. Asja Fischer, Prof. Jens Lehmann

February 6, 2017

1 IN CLASS

1. Model the dataset below with an RDF graph:

Name	Director	Release Date	Actor
Pulp Fiction	Quentin Tarantino	1994	John Travolta
Taste of Cherry	Abbas Kiarostami	1997	Homayoun Ershadi
Saturday Night Fever	John Badham	1977	John Travolta

Festival Name	Year	Winning Movie	Country
Cannes Film Festival	1994	Pulp Fiction	USA
Cannes Film Festival	1997	Taste of Cherry	Iran

2. Model *Germany* with a dataset which contains its language, chancellor's name, continent, currency, population, total area and capital city. Explain which should be literal values, which should be resources and why.
3. Write the statement below in Turtle syntax using abbreviation and grouping whenever possible:

```
<http://dbpedia.org/resource/John_Travolta>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://dbpedia.org/ontology/Actor> .
<http://dbpedia.org/resource/John_Travolta>
```

```

    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Producer> .
  <http://dbpedia.org/resource//John_Travolta>
    <http://dbpedia.org/ontology/starring>
      <http://dbpedia.org/resource/Basic_(film)> .
  <http://dbpedia.org/resource//John_Travolta>
    <http://dbpedia.org/ontology/starring>
      <http://dbpedia.org/resource/Shout_(film)> .

```

4. Write SPARQL queries to answer the movie related questions below using the displayed Turtle data.

```

@prefix ex: <http://example.org/movies/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

```

```

ex:m1  rdf:type ex:Movie;
       ex:genre ex:Drama;
       ex:year  "2006"^^xsd:gYear;
       rdfs:label "Marie_Antoinette";
       ex:country ex:USA;
       ex:director ex:p1;
       ex:actor ex:p2 .

ex:p1  rdf:type ex:Director;
       foaf:familyName "Coppola";
       foaf:givenName  "Sofia";
       ex:birthYear  "1971"^^xsd:gYear .

ex:p2  rdf:type ex:Actor;
       foaf:familyName "Dunst";
       foaf:givenName  "Kirsten";
       ex:birthYear  "1982"^^xsd:gYear .

ex:p5  rdf:type ex:Actor;
       foaf:familyName "De_Niro";
       foaf:givenName  "Robert";
       ex:birthYear  "1943"^^xsd:gYear .

ex:m2  rdf:type ex:Movie;
       ex:genre ex:Crime;
       ex:year  "1995"^^xsd:gYear;
       rdfs:label "Heat";
       ex:country ex:USA;
       ex:director ex:p3;
       ex:actor ex:p4 , ex:p5.

```

```

ex:p3  rdf:type ex:Director;
        foaf:familyName "Mann";
        foaf:givenName  "Michael";
        ex:birthYear  "1943"^^xsd:gYear .
ex:p4  rdf:type ex:Actor;
        foaf:familyName "Pacino";
        foaf:givenName  "Al";
        ex:birthYear  "1940"^^xsd:gYear .

```

- Names of all movies.
- Names of movies and directors sorted descending by the year the movie appeared.
- Names and directors of all movies before 1996.
- Names all movies whose genre is Crime.
- Names of all actors who are above 50 (at 2016).
- Names of all movies whose directors are above 70 (at 2016).

5. Consider the following RDF document on planets:

```

@prefix ex: <http://example.org/planets/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Sun ex:radius "1.392e6"^^xsd:double ;
ex:satellite ex:Merkur, ex:Venus, ex:Earth, ex:Mars .
ex:Mercury ex:radius "2439.7"^^xsd:double .
ex:Venus ex:radius "6051.9"^^xsd:double .
ex:Earth ex:radius "6372.8"^^xsd:double ;
ex:satellite ex:Mond .
ex:Mars ex:radius "3402.5"^^xsd:double ;
ex:satellite ex:Phobos, ex:Deimos .
ex:Moon ex:name "Mond@de", "Moon@en" ;
ex:radius "1737.1"^^xsd:double .
ex:Phobos ex:name "Phobos" .
ex:Deimos ex:name "Deimos" .

```

Give SPARQL queries returning the following in tabular format:

1. Object circling around the sun or a satellite of the sun.
2. Objects having a satellite with an English name.
3. Objects having a satellite which are themselves satellite of another object.
4. Objects having a satellite with an English name, which are also satellite of another object with more than 3000 kilometers diameter.

5. Objects with two or more satellites (you can assume that different URIs refer to different objects).

2 AT HOME

1. Install Fuseki (go to <https://jena.apache.org/download> to download, then start "fuseki-server" on the command line and go to <http://localhost:3030>). Load the data from Q4 and Q5 and try the corresponding SPARQL queries! Experiment with leaving out and/or adding further triple patterns.
2. Describe the Knowledge Graph Analysis (KGA) course at University of Bonn in Turtle including the university (Bonn), the lecturers (Prof. Lehmann, Dr. Fischer), the semester (SW 16/17), the track (Intelligent Systems) and the name of the research department organising the lecture (Smart Data Analytics, SDA).
3. Draw the RDF graph of:
{ (ex:a1, rdf:type, ex:person), (ex:a1, ex:name, ex:John Travolta), (ex:a1, ex:age, "62"), (ex:a1, ex:role, ex:Vincent Vega),
(ex:a2, rdf:type, ex:person), (ex:a2, ex:name, ex:Bruce Willis), (ex:a2, ex:role, ex:Butch Coolidge),
(ex:a3, rdf:type, ex:person), (ex:a3, ex:name, ex:Uma Thurman), (ex:a3, ex:age, "46"),
(ex:a3, ex:role, ex:Mia Wallace)
(ex:a2, ex:kills, ex:a1), (ex:a3, ex:likes, ex:a1), (ex:a2, ex:neverMet, ex:a3)}
4. Create the following SPARQL queries for DBpedia!

You can test the queries on <http://dbpedia.org/sparql> or <http://live.dbpedia.org/sparql>. Please note that DBpedia data in this endpoint is subject to change (in particular the live version will be frequently updated). In some cases, this can result in the queries below no longer working and/or the changes in properties or classes.

For each question, we provide you with the DBpedia entities, which are not in the RDF, RDFS or OWL namespace (e.g. `rdf:type` is not listed). The following prefixes should be used (note that the DBpedia endpoint already has those predefined as listed in <http://dbpedia.org/sparql?nsdecl> so you do not need to manually add them in your SPARQL queries):

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX res: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX yago: <http://dbpedia.org/class/yago/>
```

We distinguish between individuals (specific entities e.g. Angela Merkel), classes (sets of individuals) and properties (connecting individuals) here, so you can more easily define your queries.

- a) How tall is Claudia Schiffer?
- Classes: –
 - Properties: dbo:height
 - Individuals: res:Claudia_Schiffer
- b) Give me all female Russian astronauts.
- Classes: yago:RussianCosmonauts, yago:FemaleAstronauts
 - Properties: –
 - Individuals: –
- c) How many monarchical countries are there in Europe?
- Classes: yago:EuropeanCountries
 - Properties: dbo:governmentType
 - Individuals: –
- d) Which states of Germany are governed by the Social Democratic Party?
- Classes: yago:StatesOfGermany
 - Properties: dbp:rulingParty
 - Individuals: res:Social_Democratic_Party_of_Germany
- e) Which monarchs of the United Kingdom were married to a German?
- Classes: yago:MonarchsOfTheUnitedKingdom
 - Properties: dbo:spouse, dbo:birthPlace
 - Individuals: res:Germany
- f) Which countries have places with more than two caves?
- Classes: dbo:Cave, dbo:Country
 - Properties: dbo:location
 - Individuals:
- g) Give me all cities in New Jersey with more than 100000 inhabitants.
- Classes: dbo:City
 - Properties: dbo:isPartOf, dbp:populationTotal
 - Individuals: res:New_Jersey
- h) Is proinsulin a protein?
- Classes: dbo:Protein
 - Properties: –
 - Individuals: res:Proinsulin
- i) Is Frank Herbert still alive?

- Classes: –
- Properties: dbo:deathDate
- Individuals: res:Frank_Herbert

j) Which mountain is the highest after the Annapurna?

- Classes: dbo:Mountain
- Properties: dbo:elevation
- Individuals: res:Annapurna