Task 1: Linked Open Data Cloud

Have a look at the Linked Open Data Cloud: lod-cloud.net.

1.1 What does the Linked Open Data Cloud show?

> *The Linked Open Data Cloud shows datasets that have been published in the Linked Data format and described in Datahub (e.g. https://old.datahub.io/dataset/dbpedia). Each node represents a dataset and is assigned a field such as "geography" or "media". An edge from one dataset node to another dataset node denotes that the first dataset links to instances of the second dataset.*

1.2 Can you name central nodes and detect important subgroups of nodes in the cloud?

> *Some nodes such as DBpedia, Data.gov.uk and Freebase have a lot of incoming links, which demonstrates their important role in the Linked Data community.*
> *The Comprehensive Knowledge Archive Network (CKAN) links to many other nodes, as it is a management system for the storage and distribution of open data.*
> *There is a subgroup of many interlinked life science related datasets that are rather isolated from the rest of the Linked Open Data Cloud.*

Task 2: Knowledge Graphs

Find differences and similarities of the information about Hannover in the following knowledge graphs and resources:

DBpedia	http://dbpedia.org/page/Hanover
Wikidata	https://www.wikidata.org/wiki/Q1715
Google Knowledge Graph	https://www.google.com/search?q=hannover
Wikipedia Infobox	https://en.wikipedia.org/wiki/Hanover

DBPedia has the most facts about Hannover, but many of them are hard to interpret (e.g. "dbp:decPrecipitationMm").
Wikidata facts are enriched with meta information such as temporal validity times and provenance information.
The Google Knowledge Graph is focused on information of public interest, such as weather and travel information.
The information present in the Wikipedia infobox can be found both in DBpedia and Wikidata. This indicates that those two knowledge graphs extract data from Wikipedia infoboxes.

Task 3: Understanding SPARQL queries

For each of the following queries:
- Explain which results you expect from the query.
- Execute the query at https://dbpedia.org/sparql and check the results.

What do you notice when you execute Query 3.3, compared to the other queries?

Query 3.1
```
SELECT ?leader WHERE {
      ?city rdfs:label "Hanover"@en .
      ?city dbo:leaderName ?leader .
}
```
Who is the current leader of Hannover?
> *http://dbpedia.org/resource/**Stefan_Schostok***

Query 3.2
```
SELECT ?a ?b ?c WHERE {
      ?a ?b ?c .
      ?c rdfs:label "Hanover"@en .
}
```
Which are the triples stored in DBpedia that refer to Hannover as an object?
This query is particularly useful to get an overview of frequent properties.
> *<http://dbpedia.org/resource/**Charlotte_Buff,***
> *http://dbpedia.org/ontology/**deathPlace,***
> *http://dbpedia.org/resource/Hanover>, ...*

Query 3.3
```
SELECT ?subject WHERE {
      ?subject dbo:abstract ?abstract .
      FILTER regex(?abstract, "Hanover") .
      FILTER regex(?abstract, "computer", "i") .
}
```
Which DBpedia items have an abstract text that mentions "Hanover" (case-sensitive) and "computer" (case-insensitive)?
> *http://dbpedia.org/page/**Hermann_Gummel**,*
> *http://dbpedia.org/page/**Hanover_Public_School_District**, ...*

Query 3.3 needs the longest execution time by far. This is because it is using free text search in the Wikipedia abstracts stored in DBpedia. Due to the text search, we can also not say for sure whether the texts refer to Hannover in Germany or Hanover in the USA.

Query 3.4
```
SELECT ?player WHERE {
      ?team rdfs:label "Hannover 96"@en .
      ?player dbo:team ?team .
      ?player dbo:height ?height .
      FILTER(?height > 1.95) .
}
```
Which (former) members of the Hannover 96 football team are taller than 1.95?

> *http://dbpedia.org/page/**Per_Mertesacker**,*
> *http://dbpedia.org/page/**Philipp_Tschauner**, ...*

Task 4: Writing SPARQL queries

For each of the following queries given in natural language::
- Write a SPARQL query to retrieve the needed information.
- Execute your query at https://dbpedia.org/sparql and check the results.

Query 4.1
What is the total population ("`dbo:populationTotal`") of Hannover?
```
SELECT ?population WHERE {
      ?city rdfs:label "Hanover"@en .
      ?city dbo:populationTotal ?population .
}
```
> *"**518386**"^^<http://www.w3.org/2001/XMLSchema#nonNegativeInteger>*

Query 4.2
Which entities have a label ("`rdfs:label`") that ends with "Hannover"? Show their URIs and labels.
```
SELECT ?subject ?label WHERE {
      ?subject rdfs:label ?label .
      FILTER(regex(?label, "Hannover$"))  .
}
```
> *http://dbpedia.org/resource/**Ernest_August** | "**Ernst August von Hannover**"@de, ...*

Query 4.3
Which persons were born and died in Hannover?
```
SELECT ?person WHERE {
      ?city rdfs:label "Hanover"@en .
      ?person dbo:birthPlace ?city .
      ?person dbo:deathPlace ?city .
}
```
> *http://dbpedia.org/page/**Ernst_Brandes**,*
> *http://dbpedia.org/page/**Fritz_Haarmann**, ...*

To make sure that only persons (and not e.g. famous animals) are returned, the line
"`?person rdf:type dbo:Person .`" can be added to the query.

<u>Query 4.4</u>
Which districts of Hannover have a total population of less than 20,000?
```
SELECT ?district WHERE {
      ?city rdfs:label "Hanover"@en .
      ?district dbo:district ?city .
      ?district dbo:populationTotal ?population .
      FILTER(?population < 20000) .
}
```
*http://dbpedia.org/page/**Harenberg**,*
*http://dbpedia.org/page/**Gehrden**, ...*

More Information

Wikidata SPARQL endpoint:
https://query.wikidata.org

Query 3.1 in Wikidata:
```
SELECT ?leader WHERE {
      ?city rdfs:label "Hanover"@en .
      ?city wdt:P6 ?leader .
}
```

Query 3.1 in Wikidata (with unique subject identifier):
```
SELECT ?city ?leader WHERE {
  ?city wdt:P6 ?leader .
  <https://en.wikipedia.org/wiki/Hanover>        schema:about ?city ;
}
```

Wikidata SPARQL query examples with explanations:
https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial

YAGO demos:
https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/demo/

Google Knowledge Graph Search API (executable with key only):
https://developers.google.com/knowledge-graph/