**PMS 509 Knowledge Technologies**

**Fall 2016-2017**

**Homework I**

**Out: October 16, 2017**

**Due: November 19, 2017 at 24:00.**

**Total marks: 500**

**Kind request:** Please write your answers in English. Writing in English will be a nice exercise for you and it can be useful in your further studies, work etc.

**Exercise 1 (Search Engine Knowledge Graphs)**

In the introductory lecture we discussed how search engines like Google are using large knowledge graphs and related to technologies in order to understand the semantics of user queries and answer them more precisely.

In this exercise you are asked to experiment with existing search engines like Google and Bing and find out how well they answer queries with a temporal dimension. Examples of such queries are the following:

* Who was the prime minister of Germany in 2015?
* Which politicians were prime ministers of Great Britain during the year 2016?
* When did World War II took place?
* Who was the prime minister of Germany when the reunification of Germany took place?

Suggest knowledge graph representations and techniques that would help current search engines answer these queries precisely. You might want to consider different classes of queries with a temporal dimension and suggest techniques accordingly.

This exercise is an open research question ☺ You are free to use any relevant resources on the Web (papers, Web pages etc.) but please make sure to give them credit in your answers.

**[100 marks]**

**Exercise 2 (Wikidata)**

One of the large knowledge bases or ontologies we discussed in the introductory lecture is Wikidata ([https://www.wikidata.org/wiki/Wikidata:Main\_Page](https://www.wikidata.org/wiki/Wikidata%3AMain_Page)). In this exercise you will become familiar with Wikidata by examining its contents, posing SPARQL queries and adding information to it. More specifically, you have to do the following:

* Become familiar with Wikidata by browsing its web site. Take the tutorials and understand how Wikidata represents knowledge (i.e., its knowledge model). Maybe you would also like to read the following papers: <http://korrekt.org/papers/Wikidata-CACM-2014.pdf> and <http://korrekt.org/papers/Wikidata-RDF-export-2014.pdf>.
* Choose your favourite Greek entity (a football team, a mountain, a politician, a university etc.) and add information about it in Wikidata (the entity may already exist). You do not need to add lots of information although this is encouraged, but enough so that you appreciate how this is done. In your answer explain to us what you added to Wikidata and give us relevant pointers. If you enjoy this, become a Wikidata contributor!
* Read about the SPARQL query service of Wikidata. Then use this service to pose the following queries:
	+ Find all the prime ministers of Greece known to Wikidata. Output their name, the party or parties they have been members of and the university (-ies) that they have graduated from (assuming that they have a university degree ☺).
	+ Find all the Greek universities known to Wikidata. Output their name, the city that they are located in and the number of Greek authors that have graduated from them (order answers by this number).

**[50 marks]**

**Exercise 3 (Querying the Greek administrative geography dataset using SPARQL)**

Our group has initiated the development of a linked open data portal of interest to Greece (<http://www.linkedopendata.gr/>). In the context of this effort, we have developed an ontology and a corresponding dataset for the new administrative system of Greece known as the Kallikratis plan (<http://en.wikipedia.org/wiki/Administrative_divisions_of_Greece>). This exercise involves posing SPARQL 1.1 queries against this ontology and dataset.

First, we ask you to understand the Kallikratis ontology gag-ontology.rdf given at the Web page of the exercises for the course (<http://cgi.di.uoa.gr/~pms509/projects.htm>). See also the brief documentation available there.

Then consider the dataset for Kallikratis given at the same Web page. Load the dataset in Sesame and use SPARQL 1.1 to express the following queries:

* Give the official name and population of each municipality (δήμος) of Greece.
* For each municipality (δήμος) of Greece, give its official name, the official name of the regional unit (περιφερειακή ενότητα) it belongs to, and the official name of each municipal unit (δημοτική ενότητα) in it. Organize your answer by municipality.
* For each municipality of the region Crete with population more than 5,000 people, give its official name and its population.
* For each municipality of Crete for which we have no seat (έδρα) information in the dataset, give its official name.
* For each municipality of Crete, give its official name and all the administrative divisions of Greece that it belongs to according to Kallikratis. Your query should be the simplest one possible, and it should not use any explicit knowledge of how many levels of administration are imposed by Kallikratis.
* For each region of Greece, give its official name, how many regional units belong to it, the official name of each regional unit (περιφερειακή ενότητα) that belongs to it, and how many municipalities belong to that regional unit.
* Check the consistency of the dataset regarding stated populations: the sum of the populations of all administrative units A of level L must be equal to the population of the administrative unit B of level L+1 to which all administrative units A belong to. (You have to write one query only.)
* Give the decentralized administrations (αποκεντρωμένες διοικήςεις) of Greece that consist of more than two regional units. (You cannot use SPARQL 1.1 aggregate operators to express this query.)

 **[150 marks]**

**Exercise 4 (Greek administrative geography and GeoNames)**

GeoNames (<http://www.geonames.org/>) is a gazetteer that collects both geospatial and thematic information for various placenames around the world. GeoNames data is available through various Web services but it is also published as linked data (<http://www.geonames.org/ontology/documentation.html>).

In order to make accessible to users of the Kallikratis dataset mentioned above, the rich geographical information held by Geonames, we have interlinked the two datasets by creating owl:sameAs links for each municipality in the Kallikratis dataset. For example, the assertion

<http://geo.linkedopendata.gr/gag/id/9325> owl:sameAs <http://sws.geonames.org/8133762/>

links the Kallikraties entity “ΔΗΜΟΣ ΧΑΝΙΩΝ” with the same entity in the Geonames dataset. Your job in this exercise is to use these links to answer the queries given below. You should use the SPARQL endpoint for the Kallikratis dataset available at <http://geo.linkedopendata.gr/gag-endpoint/> (instead of your own Sesame deployment as you did in Exercise 3) and try its various ways of returning answers to your queries.

* Find all information that Geonames has for “Dimos Chania” (you have to use only Geonames here, not the Kallikratis dataset).
* Find all information held by Geonames for municipalities in the regional unit of Chania (περιφερειακή ενότητα Χανίων).
* For every municipality of the region of Crete according to Kallikratis, find its population and its population given by Geonames. Is the population information in the two datasets the same? Discuss the quality of the results.
* What kind of hierarchical administrative information for Greece is provided by Geonames and how does it compare to the Kallikratis dataset? Explain your answer using appropriate SPARQL queries on the joint datasets and their results.

**[50 marks]**

**Exercise 5 (**[**http://schema.org**](http://schema.org) **)**

As we have discussed in class, <http://schema.org> is a major effort from the top search engine companies (Google, Bing, Yahoo and Yandex) to help web designers annotate their pages with structured information which can then be used by search engines for better indexing of these web pages. You can read about this effort at <https://developers.google.com/search/> and <http://schema.org/> .

As you can see <http://schema.org/> provides an ontology for annotating Web pages. This exercise asks you to write queries that navigate this ontology and are evaluated using RDFS reasoning. First browse the ontology starting from the page <https://schema.org/docs/schemas.html>. You should also read about the data model and other information about this ontology at <http://schema.org/docs/documents.html>. Then download the recent version of the core ontology <https://schema.org/version/3.3/> as triples, store it in a Sesame repository that supports inferencing, and use SPARQL 1.1 to express the following queries:

* Find all subclasses of class Place (note that <http://schema.org/> prefers to use the equivalent term “type” for “class”).
* Find all the superclasses of class Place.
* Find all properties defined for the class Place together with all the properties inherited from its superclasses.
* Find all classes that are subclasses of class Thing and are found in at most 2 levels of subclass relationships away from Thing.
* Finally, express the above queries on the ontology and dataset but without the use of inferencing.

**[50 marks]**

**Exercise 5 (Using** [**http://schema.org**](http://schema.org) **to annotate Web pages)**

Now that we have understood what <http://schema.org/> is, let us use it to annotate the Web page of the Acropolis Museum (http:// <http://www.theacropolismuseum.gr/> ). First of all read about this task on <http://schema.org/docs/gs.html> and familiarize yourself with the relevant technologies of Microdata, RDFa and JSON-LD. You can also see examples of using <http://schema.org/> on the main web page of each of its elements (e.g., see examples of using the class Place at the bottom of the page <https://schema.org/Place> ). Google recommends to use JSON-LD to annotate Web pages (see <https://developers.google.com/search/docs/guides/intro-structured-data>).

Your job in this exercise is to use the format of JSON-LD to prepare a new version of the Web page of the Acropolis Museum. You do not need to copy all the functionality of the current page, only some useful information from it together with information from some of the pages it links to (e.g., <http://www.theacropolismuseum.gr/el/content/istoria-0> ).

You should use the Google tool available at <https://search.google.com/structured-data/testing-tool/u/0/> to verify your code. The JSON-LD code should be embedded in your HTML code to have a fully functional Web page for the Acropolis Museum which can be explored using your favorite browser.

**[100 marks]**