

# M164 CS2 Knowledge Technologies

## Homework 1

<b>Exercise 1 (Wikidata)</b>	<b>2</b>
<b>Exercise 2</b>	<b>3</b>
<b>Exercise 3 (linkedopendata &amp; geonames)</b>	<b>6</b>
Kallikratis:	7
Geonames:	7
<b>Exercise 4 (Schema.org)</b>	<b>8</b>
With Inferencing	8
Without Inferencing	9
<b>Exercise 5</b>	<b>10</b>

## Exercise 1 (Wikidata)

For the purposes of the first exercise the greek entity Gregory's Micromeals was chosen. There was already an entry made in Wikidata but there was very little (and semi-inaccurate) information added. The title was missing an apostrophe and the labels, descriptions and aliases were inaccurate/missing. For example, the Greek description was just the word "restaurant" in English. Also, there were additional statements added regarding the creation and history of the chain. All the information that was added was based on their [official website](https://www.gregorys.gr/en). The pointer to the wikidata page is [Q62273834](https://www.wikidata.org/wiki/Q62273834).

Below you can find the SPARQL queries that were used:

1. 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.

```
SELECT ?item ?itemLabel ?politicalPartyLabel ?educatedAtLabel
WHERE {
  ?item wdt:P27 wd:Q41 .
  ?item wdt:P39 wd:Q4377230 .
  OPTIONAL{?item wdt:P102 ?politicalParty .}
  OPTIONAL{
    ?item wdt:P69 ?educatedAt .
    ?educatedAt wdt:P31 ?type FILTER(?type = wd:Q3918) .
  }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "el,en". }
}
```

[Link to wikidata query](#)

2. 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).

```
SELECT DISTINCT ?item ?itemLabel (COUNT(?person) AS ?count)
WHERE {
  ?item wdt:P31 wd:Q3918 . # instance of university
  ?item wdt:P17 wd:Q41 . # country Greece
  OPTIONAL{?item wdt:P159 ?headquartersLocation .} # optional
  headquarters location
  OPTIONAL{
    ?person wdt:P69 ?item . # person educated at university
    ?person wdt:P27 wd:Q41 . # person is greek
    ?person wdt:P106 wd:Q36180 # person is writer
  }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "el,en". }
} GROUP BY ?item ?itemLabel ORDER BY DESC(?count)
```

In this example I used writer(wd:Q36180) and not author(wd:Q482980) because author returned very few results. For example, only 10 authors are turned in total but more than 255 writers.

[Link to wikidata query](#)

## Exercise 2

1. Give the official name and population of each municipality (δήμος) of Greece.

```
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
SELECT ?official_municipality_name ?population
WHERE { ?municipality rdf:type gag:Δήμος .
        ?municipality gag:έχει_επίσημο_όνομα ?official_municipality_name .
        ?municipality gag:έχει_πληθυσμό ?population . }
```

2. 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.

```
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
SELECT ?municipality ?official_municipality_name ?
official_regional_unit_name
?official_municipality_unit_name
WHERE { ?municipality rdf:type gag:Δήμος .
        ?municipality gag:έχει_επίσημο_όνομα ?official_municipality_name .
        ?municipality gag:ανήκει_σε ?regional_unit .
        ?regional_unit gag:έχει_επίσημο_όνομα ?official_regional_unit_name .
        ?municipality_unit gag:ανήκει_σε ?municipality .
        ?municipality_unit gag:έχει_επίσημο_όνομα ?
official_municipality_unit_name . }
ORDER BY ?municipality
```

3. For each municipality of the region Crete with population more than 5,000 people, give its official name and its population.

```
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
SELECT ?official_municipality_name ?population
WHERE { ?region rdf:type gag:Περιφέρεια .
        ?region gag:έχει_επίσημο_όνομα "ΠΕΡΙΦΕΡΕΙΑ ΚΡΗΤΗΣ" .
        ?regional_unit gag:ανήκει_σε ?region .
        ?municipality gag:ανήκει_σε ?regional_unit .
        ?municipality gag:έχει_πληθυσμό ?population .
        FILTER(?population > 5000)
        ?municipality gag:έχει_επίσημο_όνομα ?official_municipality_name .}
```

4. For each municipality of Crete for which we have no seat (έδρα) information in the dataset, give its official name.

```
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
```

```

SELECT ?official_municipality_name
WHERE { ?region rdf:type gag:Περιφέρεια .
?region gag:έχει_επίσημο_όνομα "ΠΕΡΙΦΕΡΕΙΑ ΚΡΗΤΗΣ" .
?regional_unit gag:ανήκει_σε ?region .
?municipality gag:ανήκει_σε ?regional_unit .
FILTER NOT EXISTS { ?municipality gag:έχει_έδρα ?seat . }
?municipality gag:έχει_επίσημο_όνομα ?official_municipality_name .}

```

5. 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.

```

PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
SELECT ?official_region_name ?regional_unit_count ?
official_regional_unit_name
?municipality_count
WHERE { ?regional_unit gag:ανήκει_σε ?region .
?region gag:έχει_επίσημο_όνομα ?official_region_name .
?regional_unit gag:έχει_επίσημο_όνομα ?
official_regional_unit_name .
{
SELECT ?region (COUNT(?regional_unit) as ?
regional_unit_count)
WHERE { ?regional_unit rdf:type gag:Περιφερειακή_Ενότητα .
?region rdf:type gag:Περιφέρεια .
?regional_unit gag:ανήκει_σε ?region . }
GROUP BY ?region
} .
{
SELECT ?regional_unit (COUNT(?municipality) as ?
municipality_count)
WHERE { ?municipality rdf:type gag:Δήμος .
?regional_unit rdf:type gag:Περιφερειακή_Ενότητα .
?municipality gag:ανήκει_σε ?regional_unit . }
GROUP BY ?regional_unit
} . }
ORDER BY ?region;

```

6. 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.)

```

PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>

```

```

SELECT ?population ?sum_population ?parent_name
WHERE { ?parent_unit gag:έχει_πληθυσμό ?population .
        ?parent_unit gag:έχει_επίσημο_όνομα ?parent_name .
        {
            SELECT ?parent_unit (SUM(?unit_population) as ?
sum_population)
            WHERE { ?unit gag:ανήκει_σε ?parent_unit .
                    ?unit gag:έχει_πληθυσμό ?unit_population .
            } GROUP BY ?parent_unit } .
}

```

7. 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.)

```

PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
SELECT ?official_d_name
WHERE { ?d rdf:type gag:Αποκεντρωμένη_Διοίκηση .
        ?d gag:έχει_επίσημο_όνομα ?official_d_name .
        ?regional_unit1 rdf:type gag:Περιφερειακή_Ενότητα .
        ?regional_unit1 gag:ανήκει_σε+ ?d .
        ?regional_unit2 rdf:type gag:Περιφερειακή_Ενότητα .
        ?regional_unit2 gag:ανήκει_σε+ ?d .
        ?regional_unit3 rdf:type gag:Περιφερειακή_Ενότητα .
        ?regional_unit3 gag:ανήκει_σε+ ?d .
        FILTER(?regional_unit1 != ?regional_unit2)
        FILTER(?regional_unit2 != ?regional_unit3) }
GROUP BY ?official_d_name

```

## Exercise 3 (linkedopendata & geonames)

1. Find all information that Geonames has for “Dimos Chania” (you have to use only Geonames here, not the Kallikratis dataset).

```

PREFIX gn:<http://www.geonames.org/ontology#>
SELECT ?property ?value
WHERE { ?x ?property ?value .
        ?x geonames:name "Dimos Chania".
}

```

2. Find all information held by Geonames for municipalities in the regional unit of Chania (περιφερειακή ενότητα Χανίων).

```
SELECT ?name ?property ?value
WHERE { ?chania gag:έχει_επίσημο_όνομα "ΠΕΡΙΦΕΡΕΙΑΚΗ ΕΝΟΤΗΤΑ
ΧΑΝΙΩΝ" .
      ?munK gag:ανήκει_σε ?chania .
      ?munK rdf:type gag:Δήμος .
      ?munK owl:sameAs ?munG .
      ?munG geonames:name ?name .
      ?munG ?property ?value .
} ORDER BY ?name
```

3. 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.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
PREFIX geonames:<http://www.geonames.org/ontology#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
```

```
SELECT ?nameK ?populationK ?populationG
WHERE { ?municipalityK rdf:type gag:Δήμος .
      ?municipalityK gag:έχει_επίσημο_όνομα ?nameK .
      ?municipalityK gag:έχει_πληθυσμό ?populationK .
      ?municipalityK owl:sameAs ?municipalityG .
      ?municipalityG geonames:population ?populationG .
}
```

The results are slightly different between each dataset. That can be due to datasets not being updated at the same time and frequency as populations change.

4. 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.

### **Kallikratis:**

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
```

```
SELECT ?unit
```

```
WHERE { ?unit rdfs:subClassOf gag:Διοικητική_Μονάδα . }
```

## Geonames:

```
PREFIX geonames:<http://www.geonames.org/ontology#>
```

```
SELECT DISTINCT ?featureCode  
WHERE { ?x geonames:countryCode "GR" .  
        ?x geonames:featureClass geonames:A .  
        ?x geonames:featureCode ?featureCode .}
```

Kallikratis results are personalized to Greece (Δήμος, Περιφέρεια etc) compared to geonames results where divisions have a more generic meaning and name (first order, second order etc).

# Exercise 4 (Schema.org)

## With Inferencing

1. Find all subclasses of class Place (note that <http://schema.org/> prefers to use the equivalent term “type” for “class”).

```
PREFIX ns: <http://schema.org/>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
SELECT ?x  
WHERE { ?x rdfs:subClassOf ns:Place }
```

2. Find all the superclasses of class Place.

```
PREFIX ns: <http://schema.org/>  
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT ?x
WHERE { ns:Place rdfs:subClassOf ?x }
```

3. Find all properties defined for the class Place together with all the properties inherited from its superclasses.

```
PREFIX ns: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?property
WHERE { ?property rdf:type rdf:Property .
        ?place rdf:type rdfs:Class .
        ?property ns:domainIncludes ?place . }
```

4. Find all classes that are subclasses of class Thing and are found in at most 2 levels of subclass relationships away from Thing.

```
PREFIX ns: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?x
WHERE {
    { ?x rdfs:subClassOf ns:Thing }
    UNION
    { ?subClassOfThing rdfs:subClassOf ns:Thing .
      ?x rdfs:subClassOf ?subClassOfThing. }
}
```

## Without Inferencing

1. Find all subclasses of class Place (note that <http://schema.org/> prefers to use the equivalent term “type” for “class”).

```
PREFIX ns: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?x
WHERE { ?x rdfs:subClassOf* ns:Place }
```

2. Find all the superclasses of class Place.

```
PREFIX ns: <http://schema.org/>
```



```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?x
WHERE { ns:Place rdfs:subClassOf* ?x }
```

3. Find all properties defined for the class Place together with all the properties inherited from its superclasses.

```
PREFIX ns: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?property
WHERE { ?property rdf:type rdf:Property .
        ?place rdf:type rdfs:Class .
        ?property ns:domainIncludes ?place . }
```

4. Find all classes that are subclasses of class Thing and are found in at most 2 levels of subclass relationships away from Thing.

```
PREFIX ns: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?x
WHERE {
    { ?x rdfs:subClassOf ns:Thing }
    UNION
    { ?subClassOfThing rdfs:subClassOf ns:Thing .
      ?x rdfs:subClassOf ?subClassOfThing. }
}
```

## Exercise 5

For this exercise the Acropolis museum restaurant and one temporary exhibition event were used. These two scripts are meant for the header of the html files of the corresponding items.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "Restaurant",
  "acceptsReservations": true,
  "hasMenu": true,
```

```

"image":
"http://www.theacropolismuseum.gr/sites/default/files/styles/subcategoryimage/public/
acropolis_museum_mg_8279_photographed_by_giorgos_vitsaropoulos.jpg?
itok=oHRU9Txx",
"name": "Acropolis Museum Cafe & Restaurant",
"location": {
  "@type": "Museum",
  "name": "Acropolis Museum",
  "address": "Acropolis Museum, 15 Dionysiou Areopagitou Street, Athens
11742"
},
"servesCuisine": "Traditional greek recipes",
"telephone": "+30 210 9000915",
"priceRange": "5-25 euro per person",
"currenciesAccepted": "EUR",
"openingHours": "Monday Tuesday - Thursday 9:00 a.m. - 6:00 p.m., Friday
9:00 a.m. - 12 midnight, Saturday/Sunday 8:00 a.m. - 8 p.m.",
"paymentAccepted": "cash, credit cards",
"address": "Acropolis Museum,15 Dionysiou Areopagitou Street,Athens 11742"
}

```

</script>

```

<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "ExhibitionEvent",
  "name": "Archaic Colors",
  "location": {
    "@type": "Museum",
    "name": "Acropolis Museum",
    "address": "Acropolis Museum, 15 Dionysiou Areopagitou Street, Athens
11742"
  },
  "startDate": "31-07-2012",
  "endDate": "31-12-2019",
  "description": "The Acropolis Museum conducts research on its unique
collection of archaic statues, which retain their colors to a small or
large degree, and has opened a very extensive discussion with the public
and various experts on color, its technical issues, its detection using
new technologies, its experimental use on marble surfaces, its digital
reconstruction, its meaning, as well as the archaic period's aesthetic
perception of color. So far, scientific research into the color found on
ancient sculpture has made great progress and reached surprising
conclusions that to a large degree refute the stereotypical assumptions
regarding ancient sculpture. It turns out that color, far from being just
a simple decorative element, added to the sculpture's aesthetic quality.",

```

```
"url": "https://www.theacropolismuseum.gr/en/content/archaic-colors-1",  
"image":  
"https://www.theacropolismuseum.gr/sites/default/files/styles/activity_image/public/peploforos_front_0.png?itok=7ig7jZz7"  
}  
</script>
```