Advanced Geomatics - Exam Exercises

Andrea Antonello - Unibz

2024-06-11

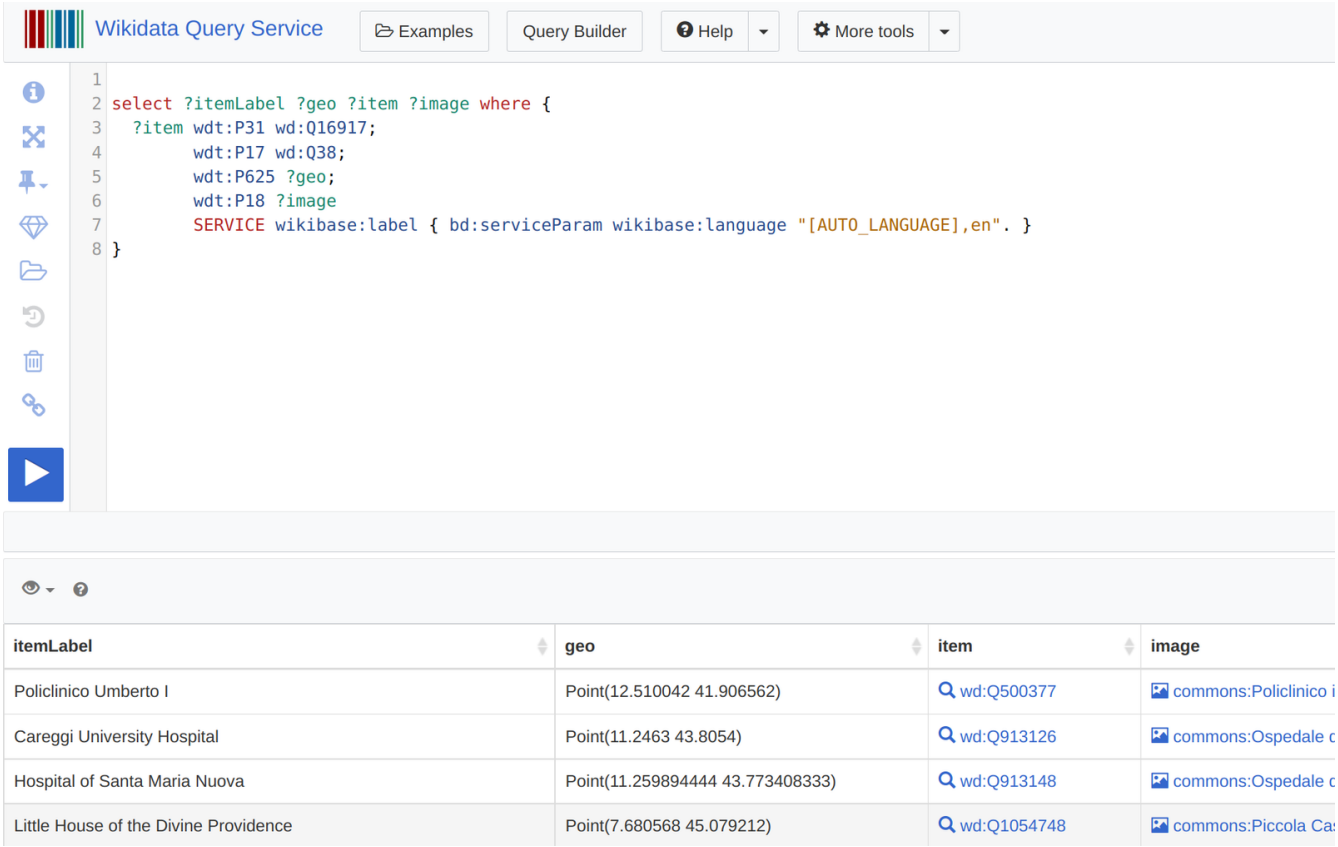
1. How to run a sparql query in the browser

Head to the wikidata query service: https://query.wikidata.org/

Insert the query in the editor:

```
select ?itemLabel ?geo ?item ?image where {
    ?item wdt:P31 wd:Q16917;
        wdt:P17 wd:Q38;
        wdt:P625 ?geo;
        wdt:P18 ?image
        SERVICE wikibase:label {
            bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
        }
}
```

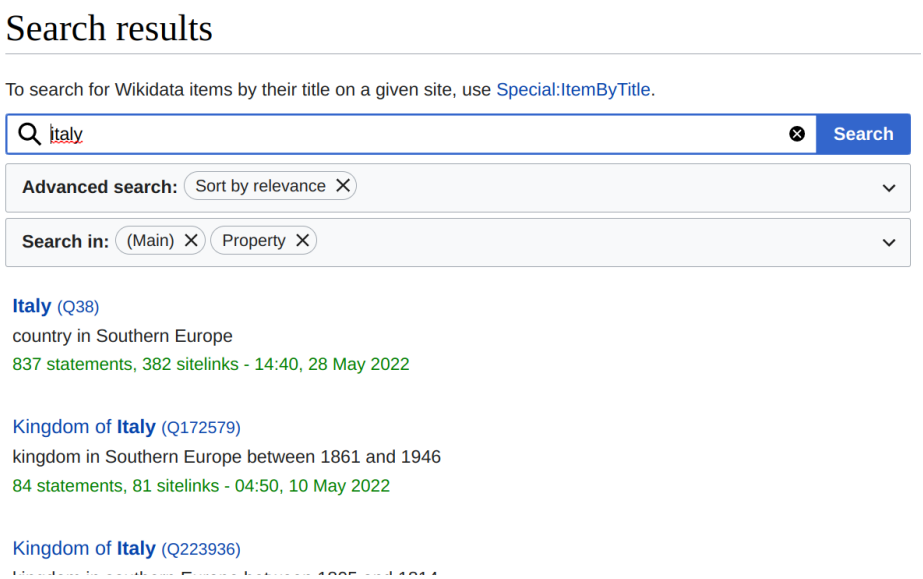
Push the run button:



Where do I find that Q38 is Italy? Where do I find the codes?

Head to wikidata: https://www.wikidata.org/

And use the search box:



The identifier is shown in brackets.

If you are interested in how wikimedia queries work, there are many articles around the net, for example:

- https://towardsdatascience.com/how-to-extract-knowledge-from-wikipedia-data-science-style-35f50f095d1a
- https://librarycarpentry.org/lc-wikidata/05-intro_to_querying/index.html

2. How to run a sparql query in python

```
# import the http requests library to get stuff from the internet
import requests
# import the url parsing library to urlencode the query
import urllib.parse

# define the query to launch
endpointUrl = "https://query.wikidata.org/sparql?query=";

# define the query to launch
query = """
select ?itemLabel ?geo ?item ?image where {
    ?item wdt:P31 wd:Q16917;
        wdt:P17 wd:Q38;
        wdt:P625 ?geo;
        wdt:P18 ?image
        SERVICE wikibase:label {
            bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
        }
}
"""

# URL encode the query string
encoded_query = urllib.parse.quote(query)
# prepare the final url
url = f"{endpointUrl}{encoded_query}&format=json"

# run the query online and get the produced result as a dictionary
r=requests.get(url)
result = r.json()
print(result)
```

3. Exam exercises

Please complete the exam exercise assigned to your group in the form of a QGIS script.

The script needs to comply with following rules:

- it has to work standalone, i.e. additional own written libraries are **not permitted**
- pip installed libraries are **permitted**
- variables that depend on the computer of the user have to be declared at the top of the script
- it has to run in the QGIS python editor, of the version used in class or major

3.1. Group 1: Places in Italy that have an elevation minor than 10 meters

Query:

```
SELECT ?place ?placeLabel ?placeDescription ?location ?elev ?image
WHERE
{
  ?place p:P2044/psv:P2044 ?placeElev.
  ?place wdt:P17 wd:Q38.
  ?placeElev wikibase:quantityAmount ?elev.
  ?placeElev wikibase:quantityUnit ?unit.
  bind(0.01 as ?km).
  filter( (?elev < ?km*1000 && ?unit = wd:Q11573)
       || (?elev < ?km*3281 && ?unit = wd:Q3710)
       || (?elev < ?km      && ?unit = wd:Q828224) ).
  ?place wdt:P625 ?location.
  optional { ?place wdt:P18 ?image }
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en" }
}
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
- create a nice pdf that contains:
  - the resulting layer (with labels)
  - the right country borders from the naturalearth dataset
  - the layer of openstreetmap
- color cities and villages with a different color than generic places

3.2. Group 2: Distribution of names of settlements ending in "-ow" or "-itz" in Germany

Query:

```
SELECT ?item ?itemLabel ?elev ?coord
WHERE
{
    ?item wdt:P31/wdt:P279* wd:Q486972;
                wdt:P17 wd:Q183;
                rdfs:label ?itemLabel;
                wdt:P2044 ?elev;
                wdt:P625 ?coord;

            FILTER (lang(?itemLabel) = "de") .
            FILTER regex (?itemLabel, "(ow|itz)$").
}
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
- create a nice pdf that contains:
  - the resulting layer (with labels)
  - the right country borders from the naturalearth dataset
  - the layer of openstreetmap
  - places should be colored differently by elevation: 0→50→100→500→1000→2000→3000 and up

3.3. Group 3: Battles in Italy and Germany

Query:

```
SELECT ?label ?coord ?subj ?year
WHERE
{
        ?subj wdt:P31 wd:Q178561 .
        {?subj wdt:P17 wd:Q38} UNION {?subj wdt:P17 wd:Q183}.
        ?subj wdt:P625 ?coord .
        OPTIONAL {?subj wdt:P580 ?d1}
        OPTIONAL {?subj wdt:P585 ?d2}
        OPTIONAL {?subj wdt:P582 ?d3}
        BIND(IF(!BOUND(?d1),(IF(!BOUND(?d2),?d3,?d2)),?d1) as ?date)
        BIND(YEAR(?date) as ?year)
        ?subj rdfs:label ?label filter (lang(?label) = "en")
}
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
  - create a nice pdf that contains:
  - the resulting layer (with labels)
  - the right country borders from the naturalearth dataset
  - the layer of openstreetmap
  - places should be colored differently by year: < 0 → 0 → 1000 → 1500 → 2000 and up
- print out which country had more battles between the years 0 and 1000

3.4. Group 4: List of lakes in Italy and Germany

Query:

```
SELECT ?item ?itemLabel ?itemDescription ?area ?elev ?image ?coord WHERE {
    ?item (wdt:P31/wdt:P279*) wd:Q23397.
    {?item wdt:P17 wd:Q38} UNION {?item wdt:P17 wd:Q183}.
    ?item wdt:P625 ?coord.
    ?item wdt:P2046 ?area.
    ?item wdt:P2044 ?elev
    OPTIONAL {?item wdt:P18 ?image.}
    SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
}
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
- create a nice pdf that contains:
  - the resulting layer (with labels)
  - the right country borders from the naturalearth dataset
  - the layer of openstreetmap
  - places should be colored differently by area: 0 → 10 → 100 → 1000 and up
- print out the number of lakes of both countries between:
  - 500 and 1000 meters
  - over 2000 meters

3.5. Group 5: Disasters in Italy and Germany

Query:

```
SELECT ?item ?itemLabel ?itemDescription ?date ?geo ?type ?typeLabel (SAMPLE(?_image) AS ?image) WHERE {
    ?type wdt:P279* wd:Q3839081 .
    {?item wdt:P17 wd:Q38} UNION {?item wdt:P17 wd:Q183}.
    ?item wdt:P585 ?date.
    ?item wdt:P31 ?type ;
        wdt:P625 ?geo .
    SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
    OPTIONAL { ?item wdt:P18 ?_image }
}
GROUP BY ?item ?itemLabel ?itemDescription ?date ?geo ?type ?typeLabel
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
- create a nice pdf that contains:
    - the resulting layer (with labels)
    - the right country borders from the naturalearth dataset
    - the layer of openstreetmap
    - places should be colored differently by disaster type

3.6. Group 6: The highest mountains in the universe

```
SELECT ?elevation ?unitLabel ?itemLabel ?itemDescription ?coord
WHERE
{
    ?psv_triples wikibase:quantityAmount ?elevation .
    filter(?elevation > 8000)
    ?psv_triples wikibase:quantityUnit ?unit .

    ?p_triples psv:P2044 ?psv_triples .
    ?p_triples a wikibase:BestRank .

    ?item p:P2044 ?p_triples .

    ?item wdt:P625 ?coord .

    SERVICE wikibase:label {
        bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
    }
}
ORDER BY DESC(?elevation)
```

- create a geopackage based on the result, both the georaphic and aphanumeric part, excluding mountains on other planets
- create a nice pdf that contains:
  - the resulting layer (with labels)
  - the country borders from the naturalearth dataset
  - the layer of openstreetmap
- color the points by elevation, dividing in 0→3000→5000→7000→8000→and up
- list the names of the 2 highest mountains

3.7. Group 7: Italian mountains higher than 4000 meters

```
SELECT ?item ?itemLabel ?coord ?elev ?picture
{
?item p:P2044/psn:P2044/wikibase:quantityAmount ?elev ;  # normalized height
        wdt:P625 ?coord ;
        wdt:P17 wd:Q38 ;
        wdt:P18 ?picture
FILTER(?elev > 4000)

SERVICE wikibase:label { bd:serviceParam wikibase:language "it" }
}
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
- create a nice pdf that contains:
  - the resulting layer (with labels)
  - the country borders from the naturalearth dataset
  - the layer of openstreetmap
- color the points by elevation, dividing into an interval that makes sense for the data's range
- list the names of the highest and the lowest mountains

3.8. Group 8: Largest cities per country

```
SELECT DISTINCT ?city ?cityLabel ?population ?country ?countryLabel ?loc WHERE {
  {
    SELECT (MAX(?population_) AS ?population) ?country WHERE {
      ?city wdt:P31/wdt:P279* wd:Q515 .
      ?city wdt:P1082 ?population_ .
      ?city wdt:P17 ?country .
    }
    GROUP BY ?country
    ORDER BY DESC(?population)
  }
  ?city wdt:P31/wdt:P279* wd:Q515 .
  ?city wdt:P1082 ?population .
  ?city wdt:P17 ?country .
  ?city wdt:P625 ?loc .
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "en" .
  }
}
ORDER BY DESC(?population)
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
- create a nice pdf that contains:
  - the resulting layer (with labels)
  - the country borders from the naturalearth dataset
  - the layer of openstreetmap
- color the points by population, dividing into an interval that makes sense for the data's range
- list the names of the 5 most populated cities

3.9. Plus points for the brave: The map of the metro of Stockhold

Query:

```
SELECT ?station ?stationLabel ?subwayLine ?coords ?line ?line_number ?layer ?rgb WHERE {
    VALUES ?search {
    wd:Q272926
#Metro system search
    }
    ?search wdt:P527 ?lignes.#What are the lines of that subway?
    ?lignes wdt:P5817 wd:Q55654238.#Lines that are running
    ?lignes wdt:P559 ?termini.#What are the termini of the lines of that subway
    ?station wdt:P31/wdt:P279* wd:Q928830; #subway stations or likes
    wdt:P5817 wd:Q55654238;#That are running
    wdt:P361|wdt:P16 ?search;#That are part of the searched subway
    wdt:P625 ?coords; wdt:P81|wdt:P1192 ?subwayLine; wdt:P197 ?pred.
    ?pred wdt:P625 ?coords_pred; wdt:P81|wdt:P1192 ?subwayLine_pred.#closest_stations
```

```
    ?station p:P197 _:b1.
    _:b1 ps:P197 ?pred ; pq:P5051 ?towards;
    pq:P81|pq:P1192 ?line_pq.
    ?pred wdt:P5817 wd:Q55654238.#Neighbours are running
    optional{?subwayLine wdt:P1671 ?line_number.}#subway line number
FILTER(?subwayLine_pred = ?lignes)#Take only the lines if correspondance on the same line
FILTER(?subwayLine = ?lignes)#Take only the lines if subway is on that line
FILTER(?subwayLine = ?line_pq)#Take only the lines if next station on the same line
FILTER(?towards = ?termini) #Take termini of the station
    ?pred p:P625 ?node_pred.
    ?node_pred psv:P625/wikibase:geoLatitude ?lat1 ; psv:P625/wikibase:geoLongitude ?lon1.
    ?node_pred a wikibase:BestRank.
    ?station p:P625 ?node_station.
    ?node_station psv:P625/wikibase:geoLatitude ?lat2 ; psv:P625/wikibase:geoLongitude ?lon2.
    ?node_station a wikibase:BestRank.
    BIND(CONCAT("LINESTRING(", STR(?lon1), " ", STR(?lat1), ",", STR(?lon2), " ", STR(?lat2), ")") AS ?str)#Construction
of colored lines
    BIND(STRDT(?str, geo:wktLiteral) AS ?line )
    ?subwayLine wdt:P465 ?rgb.#Subway line color
    SERVICE wikibase:label {
    bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en,fr".
    ?subwayLine rdfs:label ?layer.
    ?pred rdfs:label ?predLabel.
    ?station rdfs:label ?stationLabel.
    } } GROUP BY ?station ?stationLabel ?subwayLine ?rgb ?predLabel ?layer ?coords ?line ?line_number
order by xsd:integer(?line_number ) ?line_number ?layer
```

- create a geopackage based on the result, both the georaphic and aphanumeric part
- create a nice pdf that contains:
  - the resulting line layer
  - the resulting points layer (with labels)
  - all colored by the rgb field found in the result
  - the layer of openstreetmap