Lab 5 – SPARQLing Wikidata

CC7220-1 – September 9, 2024

Wikidata is a dataset proposed to better organise the data of Wikipedia (used for lists, tables, infoboxes, etc.) in a central, language-agnostic, structured repository. Wikidata is used not only by Wikipedia, but also Google[1], Alexa[2], Siri[3], etc. The dataset is published using Semantic Web standards and provides a SPARQL endpoint that we will use for today's lab to try out some SPARQL queries.

Navigate to `http://query.wikidata.org/`. Here you will find the SPARQL interface for Wikidata. Try this query, which returns the Spanish names of Chilean cities:

```
SELECT ?cityLabel
WHERE {
  ?city wdt:P31/wdt:P279* wd:Q515 ;
     wdt:P17 wd:Q298 ;
     rdfs:label ?cityLabel .
  FILTER (lang(?cityLabel)="es")
}
```

Note the following:

- Rather than use identifiers like `ex:Bejing`, which are biased to a particular language (e.g., would it be better to define the identifier for the Chinese city as `ex:北京市`?), Wikidata uses numeric identifiers like `wd:Q956`; as an additional benefit, when a city changes its name from Peking to Bejing, for example, it is not necessary to change the identifier or maintain multiple identifiers. The downside is that the query is more difficult to read but you can hover over the IRI term in the query interface to get an explanation.

- In order to find identifiers for entities (like `wd:Q515`) you can write `wd:` in the interface, hold Ctrl and press Spacebar; this opens a search box were you can type (e.g.) `chile` and get suggested identifiers. On the other hand, for properties, you can type `wdt:` and then hold Ctrl and press Spacebar.

- Wikidata uses its own properties for some properties already defined by the standards. The property `wdt:P31` denotes INSTANCE-OF, which serves the same role as `rdf:type`. The property `wdt:P279` denotes SUB-CLASS-OF which serves the same role as `rdfs:subClassOf`. The expression `wdt:P31/wdt:P279*` is a property path that matches nodes connected to `?city` by one `wdt:P31` edge and zero-or-many `wdt:P279` edges; this pattern matches instances of classes and their transitive sub-classes. For example, in the previous question, `?city` must be an instance of *city* (`wd:515`), or (transitive) subclasses like *capital city* (`wd:Q5119`), etc.

- Wikidata provides human-readable labels for entities in many languages; the filter asks for Spanish labels.

With this in mind, please provide queries to answer the following. When a question mentions your favourite entity of type x, the choice is yours, but if your choice provides empty results, you should select your other favourite entity of type x. You should return names in English where applicable (it will often give more results than for Spanish). When looking for instances of something, use `wdt:P31/wdt:P279*` to consider subclasses. Avoid returning duplicate rows. You can return additional variables alongside those requested (e.g., to return the Wikidata IDs). In the first couple of questions, we will give hints about property ids to help get you started.

1. [10 MARKS] Find the names of video games set on Mars.

[1] https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/44818.pdf

[2] https://www.wired.com/story/inside-the-alexa-friendly-world-of-wikidata/

[3] https://io9.gizmodo.com/siri-erroneously-told-people-stan-lee-was-dead-1827322243

2. [10 MARKS] Find the names of the *other* videogames made by the developer of your favourite video game (the results should not include your favourite video game).

3. [10 MARKS] Find the names of videogames made by Chilean developers.

4. [10 MARKS] Find the name of the highest grossing video game and its revenue (you can assume there are no ties).

5. [10 MARKS] Find the names of developers who have made at least three puzzle video games. Be sure to include recursive sub-types of the puzzle genre, such as *falling block puzzle game*. Order the results by the number of puzzle games developed, decreasing.

6. [10 MARKS] Find the names of exclusive video games for the Sega Dreamcast (only released on that platform) and how many *transitive* sequels they had (returning 0 if they had none[4]).

SUBMIT: A single file `lab5.sparql` containing the query required to answer each question. Ensure to upload the *queries* only, not the results. Indicate the question to which each query responds with a preceding comment (e.g., `"# Q1"` without the quotes).

[4] Some sequels might be missing ... you could add them to Wikidata if you wished!