chem-bla-ics

Getting CAS registry numbers out of WikiData

Egon Willighagen

Published April 10, 2015

Citation

Willighagen, E. (2015, April 10). Getting CAS registry numbers out of WikiData. *Chem-bla-ics*. https://doi.org/10.59350/2cfqr-fwe26

Keywords

Wikidata, Chemistry, Cas, Ldf

Abstract

I have promised my Twitter followers the SPARQL query you have all been waiting for. Sadly, you had to wait for it for more than two months. I'm sorry about that.

Copyright

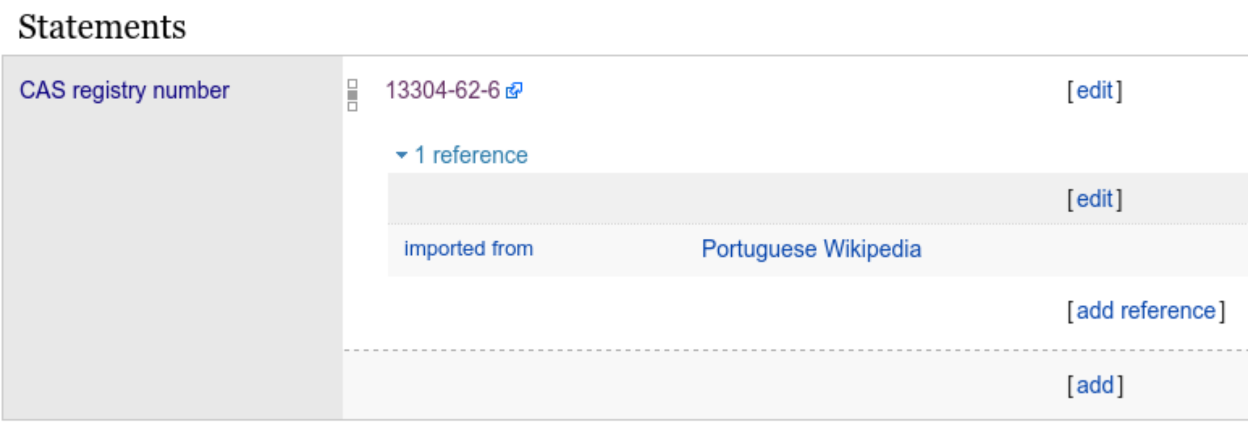
I have promised my Twitter followers the SPARQL query you have all been waiting for. Sadly, you had to wait for it for more than two months. I'm sorry about that. But, here it is:

```
PREFIX wd: <http://www.wikidata.org/entity/>

SELECT ?compound ?id WHERE {
  ?compound wd:P231s [ wd:P231v ?id ] .
}
```

What this query does is ask for all things (let's call whatever is behind the identifier is a "compound"; of course, it can be mixtures, ill-defined chemicals, nanomaterials, etc) that have a CAS registry identifier. This query results in a nice table of Wikidata identifiers (e.g. Q47512 is acetic acid) and matching CAS numbers, 16298 of them.

Because Wikidata is not specific to the English Wikipedia, CAS numbers from other origin will show up too. For example, the CAS number for N-benzylacrylamide (Q10334928) is provided by the Portuguese Wikipedia:



Statements

CAS registry number: 13304-62-6

1 reference

imported from: Portuguese Wikipedia

I used Peter Ertl's cheminfo.org (doi:10.1186/s13321-015-0061-y) to confirm this compound indeed does not have an English page, which is somewhat surprising.

The SPARQL query uses a predicate specifically for the CAS registry number (P231). Other identifiers have similar predicates, like for PubChem compound (P662) and Chemspider (P661). That means, Wikidata can become a community crowdsource of identifier mappings, which is one of the things Daniel Mietchen, me, and a few others proposed in this H2020 grant application (doi:10.5281/zenodo.13906). The SPARQL query is run by the Linked Data Fragments platform, which you should really check out too, using the Bioclipse manager I wrote around that.

The full Bioclipse script looks like:

```
wikidataldf = ldf.createStore(
  "http://data.wikidataldf.com/wikidata"
)
```

```
// P231 CAS
identifier = "P231"
type = "cas"

sparql = """
PREFIX wd:

SELECT ?compound ?id WHERE {
  ?compound wd:${identifier}s [ wd:${identifier}v ?id ] .
}
"""
mappings = rdf.sparql(wikidataldf, sparql)

// recreate an empty output file
outFilename = "/Wikidata/${type}2wikidata.csv"
if (ui.fileExists(outFilename)) {
  ui.remove(outFilename)
  ui.newFile(outFilename)
}

// safe to a file
for (i=1; i<=mappings.rowCount; i++) {
  wdID = mappings.get(i, "compound").substring(3)
  ui.append(
    outFilename,
    wdID + "," + mappings.get(i, "id") + "\n"
  )
}
```

BTW, of course, all this depends on work by many others including the core RDF generation with the Wikidata Toolkit. See also the paper by Erxleben *et al.* (PDF).