

Hands-on session: Querying Wikidata

- Go to Wikidata (https://www.wikidata.org/wiki/Wikidata:Main_Page), do a few searches in the search bar and navigate to related items or properties to get familiar with Wikidata.
- Go to <https://query.wikidata.org/>. Test a few queries from the examples set.
- Write the SPARQL queries to obtain the following information. You can find the entities identifiers you need by looking for them in Wikidata.
 - List of Albert Einstein’s children with their birth date and birth place.
 - Subproperties of the property `student`.
 - List of students of Einstein, of the students of his students, etc.
 - List of singers (occupation `singer`) having French *and* German citizenship.
 - List of singers having French *or* German citizenship.
 - List of paintings from European painters that are located in France.
 - List of French presidents with the start date of their presidency.
 - List of presidents of the French Fifth Republic with the start date of their presidency.
 - Number of presidents of the French Fifth Republic.
 - List of proteins encoded by some gene located on chromosome Y.

Tips:

- To get labels in the answer, add `?variableLabel` to the `SELECT` clause for each `?variable` in `SELECT` clause and the following at the end of the `WHERE` clause (see examples).
`{SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }}`
- Pay attention to prefixes, especially for Wikidata properties : `Pxxx` turns into `wd:Pxxx` when used as the subject or object of a triple but into `wdt:Pxxx` when used as the predicate. In particular you should not use the same variable in predicate and object or subject position.
- All details about RDF encoding (statements reification) and prefixes are available here: https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format
- See Table 1 for property paths constructors. Don’t forget parentheses when needed.

Table 1: Simple property paths constructors

<code>path1/path2</code>	path1 followed by path 2
<code>^path1</code>	backwards path (object to subject)
<code>path1 path2</code>	path1 or path2
<code>path1*</code>	path1 repeated zero or more times
<code>path1+</code>	path1 repeated one or more times