

---

# Querying data on the Semantic Web

This session will be about querying data in RDF, using the SPARQL query language. It starts with observing existing queries to understand the constructs of SPARQL queries, then follows with questions to be answered by writing a SPARQL query to a deployed endpoint.

## Observing example queries

In this part, you simply observe the existing examples from [Wikidata Query Service](#).

- Use the button labelled "Examples".
- Start looking at the first example, listing notable cats. Note that the `SERVICE` clause here uses special features of Wikidata that are not available in standard SPARQL endpoints. To execute the query, click on the blue button with a white triangle on the left side.
- Put your mouse on top of `wdt:P31`. What do you read? Do the same for `wd:Q146`.
- Delete `Q146` and place your cursor after `wd:.` Type `Ctrl+Space`. Then you should have a little popup that says "Type to search for an entity". Type "cow" and select the entity that corresponds to a kind of cattle and press enter. What is the Q number of this entity?
- Execute the query. How many results there are?
- Select the example query for *horses (with some info about them)*. This query shows multiple SPARQL constructs. Try to understand them.
- The example query named "Cats, with pictures" uses a special feature of Wikidata that is not part of the language SPARQL. At the beginning of the query, there is comment saying `#defaultView:ImageGrid`. This turns any URL of pictures in the result of a query into a displayed image.
- The example query "Map of hospitals" uses another feature of Wikidata. At the beginning of the query, there is comment saying `#defaultView:Map`. Execute the query and see what happens. Then delete the comment and execute again. What allows Wikidata to determine the location of a result?
- Take a close look at the query "Items about authors with a Wikispecies page". What SPARQL constructs are used here that are also SQL constructs?

- The query "Recent events" shows other features, like `FILTER`, `BOUND`, comparators, operators, `DATATYPE`, `BIND` . . . `AS`, `LANG`.
- You can play a bit more with the example queries.

## Challenge yourself

Use DBpedia SPARQL endpoint to solve the exercises : <http://dbpedia.org/sparql>

1. Find 50 example concepts in the DBpedia dataset.
2. Find the resource of Madonna (the singer). Tip: don't forget to use the language tag in your query: "Madonna"@en
3. Describe Madonna using one of the URIs retrieved by the previous query.
4. List the people and their names who are born in Brussels.
5. Are there people who were born in Brussels and died in Paris? (use an ASK query)
6. Find 20 people (URI and place of death) who were born in Ghent, but died elsewhere.
7. Give a list of countries and their French ('FR') labels.
8. Choose two cities, and make everyone born in one city know all persons born in the other and vice versa. (use a CONSTRUCT query)

## Some useful references

- [RDF Playground](#) from University of Chile, with RDF syntax checker and visualiser, SPARQL editor and engine, and more things.
- [SPARQL Playground](#) from the Swiss institute of bioinformatics, with more and more complicated examples