Lab 3 – SPARQL

`data.open.ac.uk` is the RDF data platform of the Open University in the UK. It includes data about people, courses, articles, course material, etc. The URL of its SPARQL endpoint is `https://data.open.ac.uk/query`.

T1: Using the form at `https://data.open.ac.uk/query`, write a SPARQL query to figure out all the types of entities in those datasets (i.e. all the things that are objects of triples with the predicate `rdf:type`. How many are there?

T2: Using the same process, find the lists of IRIs for a 100 entities of type `http://xmlns.com/foaf/0.1/Person`

T3: Inspecting one of the IRIs from the previous results, figure out what IRI is used for the predicate linking entities of type person and their full name. Change the previous query to list the names of 100 people.

T4: Using the predicates `http://xmlns.com/foaf/0.1/givenName` and `http://xmlns.com/foaf/0.1/familyName`, change the previous query so to show the given name and family name in different variables for the list of 100 people.

T5: The `http://purl.org/dc/terms/creator` predicate connects a book or article (subject) to a person who authored it (object). The `http://purl.org/dc/terms/subject` predicate connects a book or article (subject) with its topic (object). `http://data.open.ac.uk/topic/library/artificial_intelligence` is the IRI of the topic artificial intelligence. Using those, write a query getting the names (`http://www.w3.org/2000/01/rdf-schema#label`) of people who have written something on the topic of artificial intelligence.

T6: Transform the previous query into a construct query. The constructed triples will have for subject the IRI of people having written something about artificial intelligence, for predicate the IRI `http://example.com/expertIn`, and for object the IRI `http://data.open.ac.uk/topic/library/artificial_intelligence`.

T7: Change the previous query so that it works for all topics, not only for artificial intelligence (limiting to 100 triples in the results).

T8: Send the previous query to the SPARQL enpoint using the command line `curl`. You can use a tool like `https://www.urlencoder.org/` to URL encode the query. In what RDF syntax is the result?

T9: Send the query again through `curl`, but getting the results in the turtle syntax.