

Exercise Sheet 12

Submit until Tuesday, February 9 at 2:00pm

Exercise 1 (10 points)

Write a program that reads a SPARQL query from a given file, translates it to a corresponding SQL query, and executes that query on a given Sqlite3 database. The idea of the translation and the (simple) Python interface to Sqlite3 were explained in the lecture; see also the code provided on the Wiki. Your program should support all SPARQL queries of the form

SELECT <variables> WHERE { <triples> }

where *<variables>* is a space-separated list of variables, and *<triples>* is a dot-separated list of triples, where the three components of each triple (subject, predicate, object) are separated by TABs, and each component can be either a variable (e.g. *?x*) or a string literal (e.g. *Johnny Depp*). Also support the optional *LIMIT <k>*, which (if appended to the end of the query) restricts the output to (any) *k* results.

Exercise 2 (5 points)

Consider the query *German female actors born in Asia*. Formulate the appropriate SPARQL query on the Freebase dataset linked on the Wiki (use *grep* to figure out the names of the entities and relations needed for the query). Run that query (without LIMIT) using your program from Exercise 1, and report the number of query results and the query time in the table on the Wiki. Your program should accept the SPARQL query from the command line as input. Along with the code, also commit a file containing your SPARQL query and corresponding SQL query to the SVN.

Exercise 3 (5 points)

Repeat Exercise 2 for a SPARQL query of your choice, using *LIMIT 10*. That query should have at least two triples in its WHERE clause, and it should have more than one hundred matches without the limit.

Add your code and the files with the SPARQL queries to a new sub-directory *sheet-13* to our SVN, and make sure that everything runs through without errors on Jenkins. Tell us about your *experiences* one more time, and reflect on how proficient you have become in information retrieval, linear algebra, coding in general and Python in particular.