

Foundations of Linked Data

Bonus Tasks

Task 2: Querying Linked Data

Thomas Wehr, Prof. Dr. Andreas Harth

May 16, 2025

This is the second out of the three bonus tasks. All bonus tasks are voluntary. To pass this task, the correct solutions must be submitted until **June 8, 2025, 23:59h**. Students who pass all three tasks in time are awarded a 0.3 / 0.4 grade improvement in the final exam of the course.

Administrative Information

Following the inverted classroom principle, if you have problems or further questions, please post these into the forum on StudOn. That way your colleagues can also contribute to and benefit from the answers.

2 Task: Description and Objective

This task will cover lectures 4 and 5 of the Foundations of Linked Data course. The subtasks are ordered accordingly, so that you can step-by-step solve the task. The aim of this task is to make you familiar with querying RDF documents.

2.1 RDFLib

For this task, you will write SPARQL requests. For that, it would be beneficial to have a testing environment to check, if the written SPARQL request outputs the expected values. While there are several tools available, we recommend to use the python package RDFLib¹.

2.2 SPARQL requests

As you have learned, SPARQL requests can be used to retrieve RDF data as table results (SELECT), as Boolean (ASK), or as RDF data (CONSTRUCT). In the last task, you

¹<https://rdflib.readthedocs.io/en/stable/>

have created an RDF document containing information about a Bundesliga player.

Your task will be to answer several questions about this document with SPARQL requests. When you think that your SPARQL request is correct, save it as a file with the file extension `.rq` and the name given by the task title. For instance, the first task's file name should be `datatype.rq`.

2.2.1 datatype (datatype.rq)

Return all RDF Literals with integer or decimal values. (SELECT ?label)

2.2.2 Nested Count (nested.rq)

Return the number of triples the RDF document contains. You will need a nested SELECT clause. (SELECT ?num)

2.2.3 year (year.rq)

Return the first 4 characters of the players birthdate. (SELECT ?year)

2.2.4 Pick (pick.rq)

Return the 8th triple from your graph after ordering the result by descending subject and ascending predicate. (SELECT ?s ?p ?o)

2.2.5 Math (math.rq)

Calculate $|91054 \times -3 \div 33|$ and return the rounded result. (SELECT ?value)

2.2.6 Ask (ask.rq)

Validate this statement "the RDF document contains a team with the name 'Borussia Dortmund'". Instead of SELECT, you should use ASK to return True/False. It is hard to check, if the ASK query returns the correct value or was incorrectly written. You might check the result using a SELECT query before. For the submission, however, only the ASK request should be saved in `ask.rq`. (ASK)

2.2.7 Construct (construct.rq)

Construct a triple where the player's sport number increases by 23. Here you must use the CONSTRUCT term. CONSTRUCT is another SPARQL term that returns triple patterns instead of a table result. (CONSTRUCT {?s ?p ?o .})

Remark: when you have understood how to work with CONSTRUCT, you should also grasp the concept of INSERT/DELETE (DATA) which are the terms used to update

graphs. But as we load our graph from an RDF document and are not querying a SPARQL endpoint, we can only update the temporary graph.

2.3 Upload your results to StudOn

Upload all `.rq` files to the same directory you uploaded your solution for the first task. Then, just like in the first task, upload a text file with the name `"(idm).txt"` to StudOn containing a single line of the URL to your `fid.ttl`.

6 out of 7 SPARQL queries must be correct to pass the second bonus task.

Key Takeaways of this Task

At the end of this task you should be familiar with `RDFLib` and SPARQL requests.