

Lab: Knowledge Graphs

After this lab, you should have a good understanding of standardized Semantic Web technologies, including URI, RDF, RDF(S), SPARQL, and Rules, which can easily interact with knowledge in knowledge graphs (KGs).

Reading Materials

- Lecture slides
- Appendix A and Chapter 10 in Hogan, Aidan et al. [Knowledge graphs](#). ACM Computing Surveys (Csur) 54, no. 4 (2021).
- The W3C documentatons:
 - [RDF 1.1 Primer](#).
 - [RDF Schema 1.1](#).
 - [RDF 1.1 Turtle](#).
 - [SPARQL 1.1 Query Language](#).
 - [SPARQL 1.1 Query Results JSON Format](#)

Submission

You should submit a document that contains the answers to the tasks. **Note:** Any form of plagiarism, including using AI to generate answers, will result in failing the lab.

Programming Tool

In this lab it is suggested to use Visual Studio Code (VS Code) and its extensions and libraries for RDF and SPARQL.

1. RDF and RDF(S)

RDF extension in VS Code: Please install the extension [Stardog RDF Grammars](#), which include syntax highlighting support for the different RDF languages.

Submissions: You should hand in a document containing the answers to the questions in the tasks 1.1 and 1.2, and the Python program and the generated *.ttl* file for the task 1.3.

1.1 Get familiar with the RDF and RDF(S)

In this task you will get familiar with RDF and RDF(S) by looking at a music dataset. The details of the music dataset can be found [here](#). The document also provides a nice introduction to data modeling using RDF and RDF(S). Download the *beatles.ttl* and *music_schema.ttl* from [Stardog tutorials repository](#), and open them in VS Code. The *music_schema.ttl* contains a simple schema for music dataset. The *beatles.ttl* gives a tiny music dataset.

Spend some time to understand the dataset and address the exercises below:

- 1) What are the namespace references and URIs (i.e., the namespace prefixes) of the music dataset and schema?
- 2) Draw the directed labelled graph of the triples in the “beatles.ttl”, where subjects and URI-identified objects are represented as nodes (ellipses), literal objects are represented as nodes (boxes), and predicates are labeled edges in the graph.

Note: Draw the graphs on paper or using diagram software which you are familiar with.

1.2 Understand the RDF triples in Open Linked Data

In this task you will look at the RDF triples in YAGO. [YAGO](#) is a large knowledge base with general knowledge about people, cities, countries, movies, and organizations [1]. Go to https://yago-knowledge.org/resource/The_Beatles. Spend some time to understand the triples in the dataset and answer the questions:

- 1) Which ontology is used in this RDF dataset to describe the data?
- 2) Choose three triples in https://yago-knowledge.org/resource/The_Beatles and write the information described by the triples in natural language text.

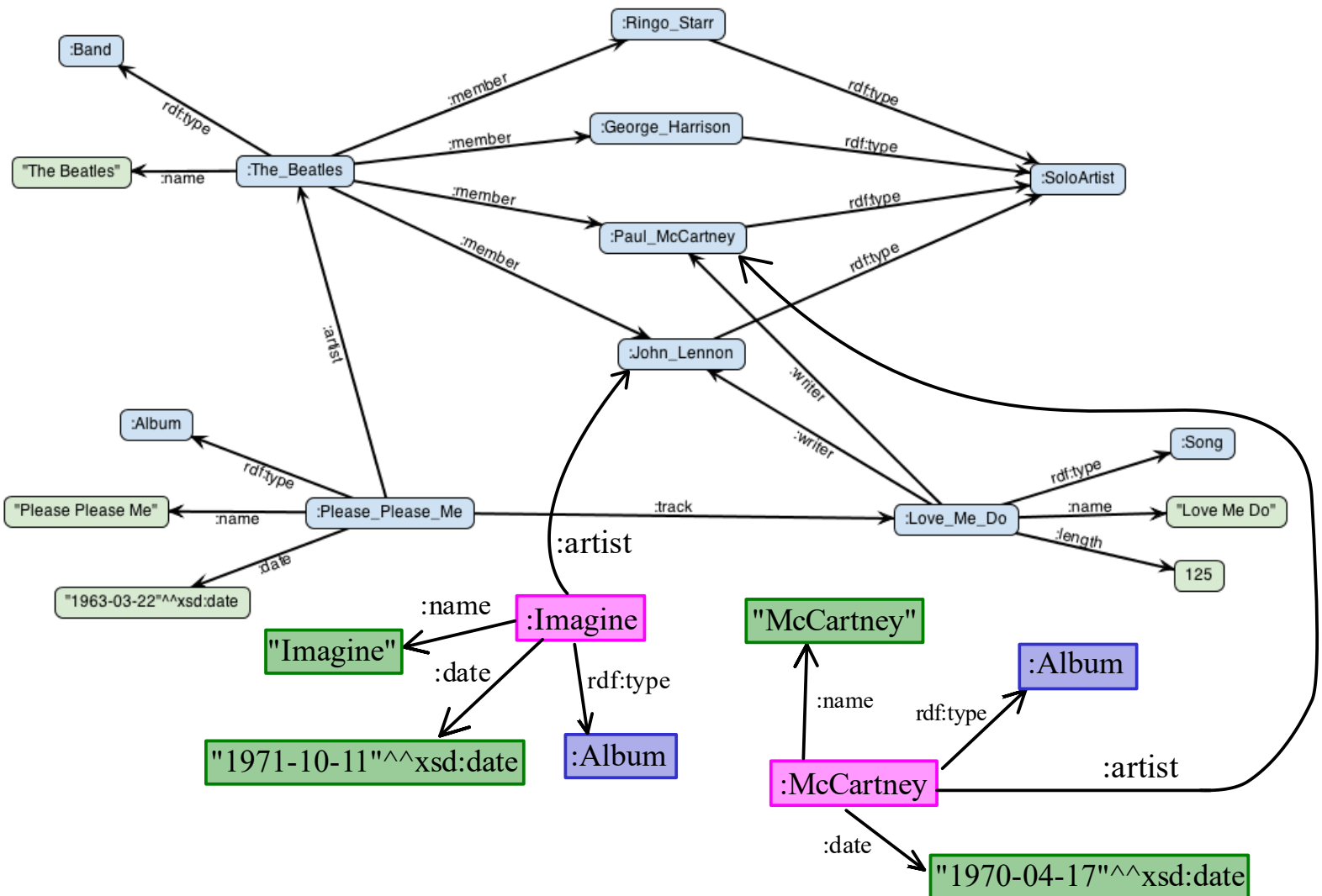
1.3 Create a RDF Graph Dataset using RDFLib

In this task, you will create a small RDF dataset about your family or any famil(ies), such as those featured in 'Game of Thrones' using [RDFLib](#). Please use the terminology, including classes and properties specified in the family ontology given in the ontology lab, to describe your dataset. The namespace URI of the family ontology is <http://example.com/owl/families>. Please save your RDF dataset in Turtle format.

RDFLib is a pure Python package for working with RDF. Read [Getting started with RDFLib](#) and understand how to install it and how it works. You can also start with the example code *familydata.py* provided in Canvas.

1.1

- 1) The namespace prefixes links the RDF dataset and dataset schema through its respective Uniform Resource Identifier (URI) to retrieve the corresponding resource. They hold definitions of the classes and relations in the knowledge graph.
- 2) The basic Beatles graph found at <<https://docs.stardog.com/tutorials/rdf-graph-data-model>> + the relations to the albums "Imagine" and "McCartney".



1.2

- 1) Ontologies used for "The Beatles" dataset: owl, rdfs, yago and schema
- 2) `yago:The_Beatles ----yago:director----> yago:Brian Epstein:`
The beatles had a director named Brian Epstein.
`The_Beatles ----yago:locationCreated----> yago:Liverpool &`
`The_Beatles ----yago:dateCreated----> "1960-01-01T00:00:00..."^^xsd:dateTime :`
The band was formed in Liverpool in 1960.

2. SPARQL Queries

SPARQL extension in VS Code: Please install the extension [SPARQL Notebook](#), which allows to document SPARQL queries and make them execute as notebook code cells. Notebooks are a form of interactive computing, in which users write and execute code, visualize the results, and share insights.

Submissions: You should hand in the completed SPARQL notebooks for the task 2.1, the program for the task 2.2, and the answers to the questions in the task 2.2.

2.1 Get Familiar with SPARQL

Download and unzip the *music.ttl.gz* from [Stardog tutorials repository](#) and the SPARQL notebook *music.sparqlbook* from the Canvas. Finish the exercises given in the notebook.

Note: The Stardog SPARQL tutorial using the music dataset can be found at [here](#).

2.2 Query Open Linked Data from Program

SPARQLWrapper is a simple Python wrapper to remotely execute queries via SPARQL endpoint. Read its [documentation](#) and understand how to install it and how it works. You can also start with the example code *remotequery.py* provided in Canvas. The SPARQL query result is usually serialized in a JSON format. The JSON format details can be found at [here](#).

Write a program to execute SPARQL queries against YAGO via its endpoint at <https://yago-knowledge.org/sparql/query> and process the results in Python. Your program should create queries, execute the queries and process and display the results for each query to answer the questions below. The classes and properties from the ontologies to describe the data can be found [here](#), both the information on the webpage and the design document.

Note: There is a 1 minute timeout to ensure a responsive service for everyone at YAGO SPARQL endpoint.

- 1) List the people who influenced The Beatles.
- 2) List the people influenced by The Beatles, either directly or indirectly (through a chain of influence), and show their nationality if available in the dataset.
- 3) List the movies from 1960 to 1970 directed by people who have won the same award as The Beatles. Tip: use the function `YEAR(?date)` to extract the year from the `?date`.
- 4) List the people who have been members of The Beatles and their family members. Tip: use navigational graph patterns to find all the family members.

3. Reasoning

Submissions: You should hand in the Prolog program which can work on the RDF dataset you created in task 1.3.

Starting with the example Prolog code *prolog-test.pl* provided in Canvas, you should extend the program to include the rules to,

- 1) traverse a family tree and gather all ancestors of a person;
- 2) calculate the generational distance between two people;

which be applied to the RDF dataset you created in task 1.3, and perform some meaningful reasoning tasks. More information on the SWI-Prolog Semantic Web Library 3.0 can be found [here](#).

References

- [1] Suchanek, F. M., Alam, M., Bonald, T., Chen, L., Paris, P. H., & Soria, J. (2024, July). Yago 4.5: A large and clean knowledge base with a rich taxonomy. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 131-140).