

Exploring Spatial Knowledge with Knowledge Graphs

In these weeks you learned how to:

- Construct a Knowledge Graph
- Query Wikidata
- Write federated queries to link entities and enrich your knowledge graph
- Model and query geospatial data using GeoSPARQL

The final project allows you to apply all of your newly acquired skills to answer a complex geospatial question and visualize the results on a map.

>> This is an individual assignment <<

In this project, you will explore how to use knowledge graphs to answer complex and interesting questions that combine geospatial calculations with Knowledge Graphs. The objective is to learn how to integrate spatial data with Linked Data to generate meaningful insights, visualize the results on a map, and deepen your understanding of knowledge graph construction and applications.

Steps to complete the project:

1. **Define a Question:** Formulate an interesting question that combines spatial calculations (e.g., distance, intersection) on certain locations (e.g., countries, cities, rivers) with data extracted from Wikidata about these locations (e.g. population, GDP, spoken language, religion, etc.). Your goal is to formulate an interesting spatial question, such as:
 - Which city within 20 km of the Alps has the most tourists each year?
 - Which capital cities are within 100 km of an active volcano?
2. **Find Geospatial Data:** Locate one or several CSV datasets that include the geographical locations of the relevant entities in WKT (Well-Known Text) format. For example, a CSV dataset of the capitals of the world with their geographical location.
3. **Convert to a Knowledge Graph:** Use Ontorefine to transform your CSV dataset into a Knowledge Graph, ensuring the geospatial data is appropriately represented using the GeoSPARQL vocabulary. The Knowledge Graph will be stored locally in the GraphDB triple store.

4. **Link to Wikidata:** Write a federated SPARQL query from GraphDB to link your entities to Wikidata and retrieve relevant data for your question. This involves writing a CONSTRUCT/INSERT query to enrich your knowledge graph with the information from Wikidata. For example, one can link the capitals to Wikidata to retrieve their population or their GDP.
5. **Answer Your Question:** Use SPARQL to query your completed knowledge graph while using geospatial computations (GeoSPARQL functions) to obtain the answer to your question.
6. **Visualize Results on a Map:** Show the results of the query on a map, ensuring the visualization clearly represents the insights derived from your query. For example, you can use the YASGUI SPARQL editor that allows to visualise SPARQL results on a map in 2D or 3D: <https://yasgui.triply.cc>. To run YASGUI locally to query your local graph, you can follow the instructions on their Github repository: <https://github.com/TriplyDB/Yasgui>
7. **Document the process that led to the implementation.** We ask you to:
 - a. Motivate your choices of the question that you outlined.
 - b. Ensure replication of your project (if I can't replicate it, I can't grade it). Make sure to clearly include a link to download the following resources:
 - i. CSV dataset(s): its source, how to download it, and the changes you made to that file (if any)
 - ii. The SPARQL queries used for linking and answering the question
 - iii. The completed RDF dataset after linking the entities to Wikidata and inserting the information required for the question
 - c. Record a video for presenting your question and dataset, and demonstrating the results of your SPARQL query visualized on a map (2 minutes max).

Be clear, structured and concise. The number of pages has no relevance at all.

The quality of your documentation (report + video) will significantly affect your grade.

>> Deadline for submitting the project: Friday January 17, 2025 <<

How to submit the project: On eCampus (mini-project submission). Make sure that all the necessary documents, resources and queries are available in one compressed folder. Make sure that the documentation and the included files in the compressed file are enough for me to replicate the project (try to follow your own steps to see whether anything is missing).