Open Data Project

OBJECTIVES

This is a hands-on practice in order to exploit and benefit from graph-based formalisms. The specific goals are as follows:

- Comprehend the benefits of graph modelling for automating processes (such as data integration, recommendations, etc.),
- Use either a property graph or knowledge graph (RDF, RDFS and / or OWL) to model your data,
- Design a database able to store and manage such data,
- Gain experience on how to extract non-graph data and enrich it with the needed semantics before being stored in your graph database,
- Exploit the resulting repository with graph analytics,
- Decide the needed metadata to be stored in the repository in order to enable and enhance the exploitation processes.

Optionally, you may create a proof of concept of this system.

PRACTICE STATEMENT

For this practice you need to identify at least two (open) data sources to be integrated in your system (i.e., an integrated repository). The integrated data, together with some metadata gathered, should then be used to automate a certain exploitation analytical process. For example, you may focus on **recommendations**, but if you prefer to focus on any other kind of analysis, this is possible.

The course practice consists of three main tasks:

Task 1: Business idea and identification of two data sources.

- Decide what the system should do. **Write a brief purpose statement**. For example, *recommend movies to users*.
- You should identify the sources from where you want to extract data. It is important **you identify what data is relevant from each source** (e.g., one source may store data about movies, actors, etc. and another one information about the

ratings given to movies by different users and their personal characteristics). **Identify the source schema and the data elements you are interested in**.

- Decide and **justify the most appropriate graph family** for the problem. Either property graphs or knowledge graphs (in this case, you must also choose the appropriate language).

Task 2: Creation of the integrated repository (and the metadata repository, if any). This stage includes the definition of the integration / global schema.

- **Design the integration schema**. Data extracted from the data sources will have to be mapped to this schema. Let us call this integrated view the **integration graph**. What concepts and relationships must be captured?
- **Design the data flows**. How is data from the sources going to be extracted and transformed? Bear in mind graph instances (i.e., the ABOX) should be aligned with the schema/**integration graph** (TBOX) created in the previous item. As we saw in the labs, you may expect that most sources you will find are not graph data. Therefore, you will have to transform them into triples. Specify the transformations to generate graph data from such sources.
- **Decide the needed metadata in order to automate your exploitation process**. For example, after recommending a movie to a user we may store if she likes / dislike the movie so that we can apply reinforced learning based on this added knowledge. Remember though there are many kinds of metadata and there is a broad list of options, so you are encouraged to explore and be creative in this task.

Task 3: Exploitation idea. Goals and algorithms.

- **Refine the purpose statement from task 1**. Add information about the specific algorithm(s) used in your pipeline. For example, decide what algorithm(s), its (their) parametrization and attributes/variables use for recommending movies. If we are analysing graph data remember to include graph-specific variables so that *using a graph is justified* (and also, you benefit from aspects you could not get from a relational or csv data source). Include a discussion about the benefit of such analysis in terms of the project.
- **Precisely describe the data analysis conducted**. You must use the usual steps to describe data analysis: preparation, creation of the model (including its parametrization) and validation tasks. Note that you can use any algorithm (in the literature you can find a plethora of graph-based algorithms), even data mining (DM) or machine learning (ML) algorithms on top of graphs (in this case, be sure it makes sense to start from a graph to conduct DM or ML).
    - In case you are interested in recommendations, be aware there are several different kinds of recommendations: user-based (if you keep track of previous interactions with your system, you may use them for better

recommendations) or collaborative recommendations (see what other users did and take this into account when recommending). Another option is to consider cold-starts (there is no additional info about the current user). You can learn more about recommenders here (from the Mining Massive Datasets book: http://www.mmds.org/).

Creating a proof of concept (PoC) of your system is not mandatory. However, you will gain much more experience by doing so. Therefore, even if not mandatory, you are suggested to develop a PoC of your system.

DELIVERABLES

First, use the team creator event to register your joint project into OD.

Once done, by the deadline stated in this event, one person of the group must upload a document giving answers to Task 1-3. There is no need for very large documents. Simply, be precise and concise. Therefore**, a maximum length of 6 content pages per group is set**.

In the case you implemented a PoC, the document should include a link to the github or project page you are using (if it is a private repository, please, create a user and state login / passwords credentials in this document).

TIMELINE

To help you at the beginning of the project, we will set one online session where we will present the main project ideas and you can ask questions. See the course schedule and look for the “project” session. In this session you should work with your group and clarify questions with the lecturer. Later, as you progress with the project, each group, if needed, can schedule an individual 1-1 session with the lecturer to clarify their doubts.

Importantly, please check the corresponding event in Learn-SQL to see the project deadline.

PROJECT TEAMS

Create your team of three yourself. Register the team created in the corresponding event of Learn-SQL.

EVALUATION CRITERIA

The project will be evaluated according to the following criteria:

Conciseness

The document fits in 6 pages and explains all the main details requested.

Understandability

You provide enough details as to assess your solution.

Soundness

There are no contradictions about the choices made and the inherent advantages of the underlying theory chosen.

Proof of Concept

If you provide a PoC you will be evaluated out of 12 (instead of 10). Whatever mark above 10 you get, it will be automatically transferred to the final exam as a surplus.