Semantic Technologies and Linked Data Project

Marco Zenere June 17th 2021

1 Motivation Scenario and Domain Description

Formula 1 is the highest class of international car racing for single-seater formula racing cars approved by the Féderation Internationale de l'Automobile (FIA) [1] .Born in 1950, today it has many fans from all over the world and from a 2019 market research [2], it is estimated to be 490 million nowadays. The domain is incredibly vast considering the data point of view and to substantiate this thesis, an agreement was recently reached with Amazon AWS to manage the large amount of data and enrich the show by providing additional information to fans during the races. The dataset used for this project is unfortunately not the one provided by the F1 company, but is instead a toy from a website made by Chris Newell [3] [4]. It contains data from 1950 to 2017 and consists of tables describing constructors, race drivers, lap time, pit stops and more. Some tables and columns are discarded from the original dataset to facilitate their management and also because they are not very interesting. The ER schema designed for this project can be seen below.

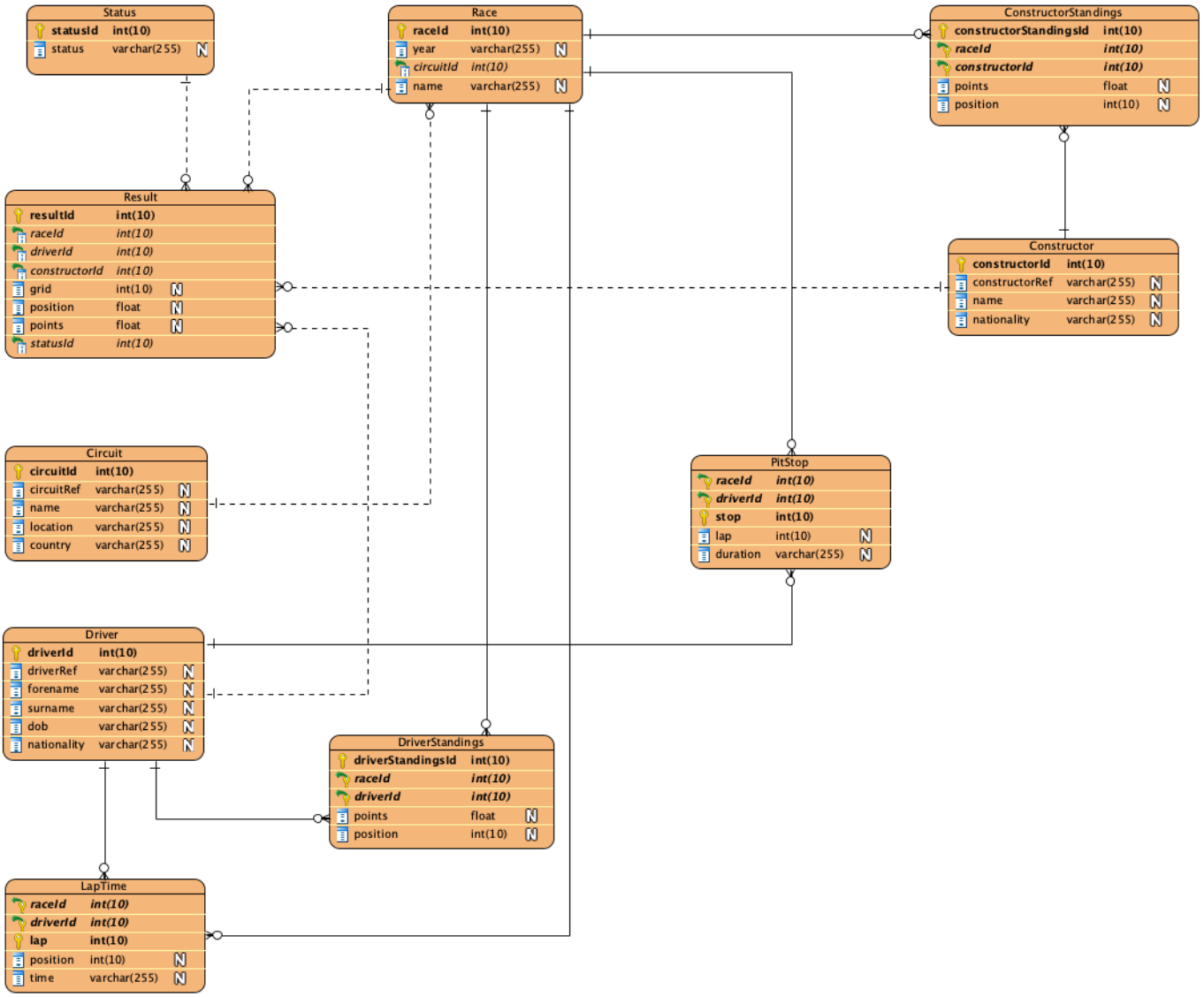

Figure 1: ER Schema

The relative ontology of the domain consists of 11 concepts and below there is a graphical visualization to see how each concept is related to the others.

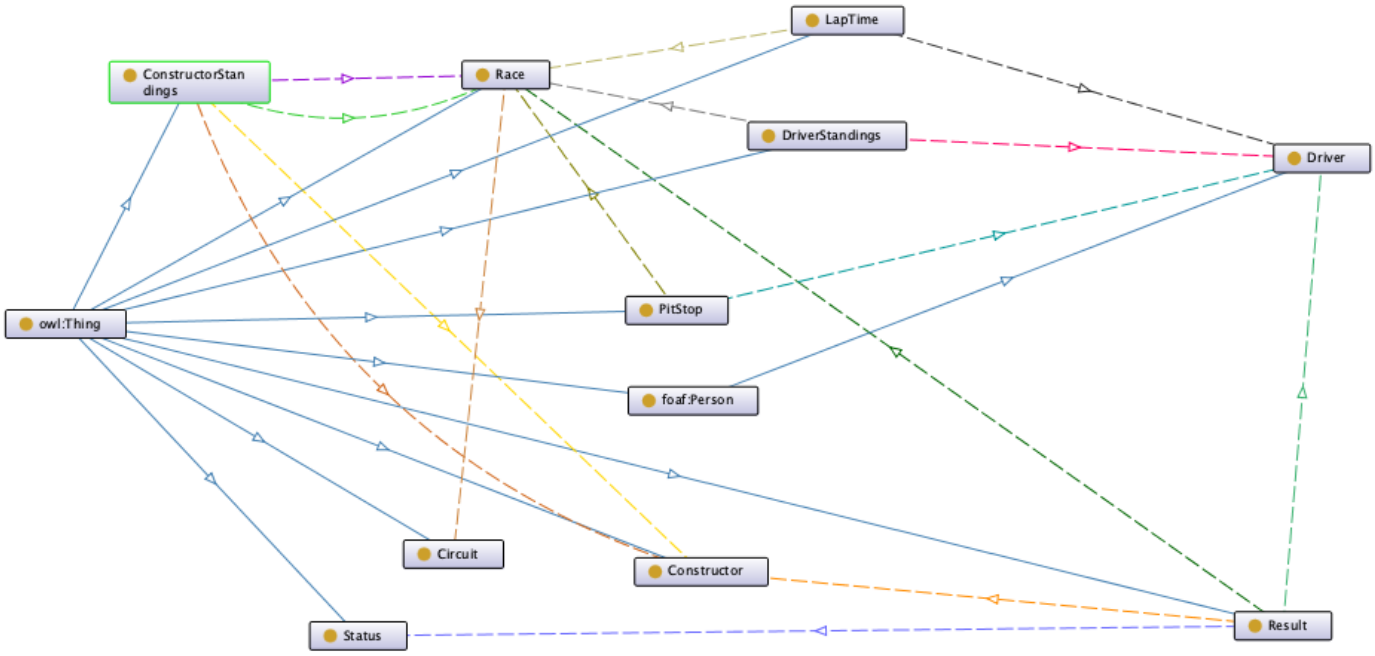

Figure 2: Domain Ontology

2 System Architecture

The system developed is composed by four layers and starting from the lowest layer of the architecture we have:

- Database Layer: It stores the data of the domain.
- Ontology Layer: Designed in Protégé, this layer represents the ontology of the domain. It is populated using mapping technique provided by OnTop.
- Query Engine Layer: Provided by OnTop, gives the capability to access to data using SPARQL language.
- Application Layer: Through Jupyter notebook written in Python, it gives the possibility to process data from the ontology.

The figure below provides a graphical representation of the system architecture.

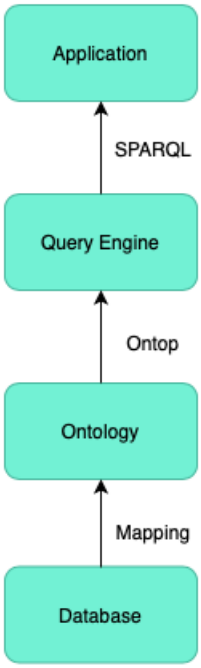

Figure 3: System Architecture

2.1 Additional information

In the beginning, the intention was to do the mapping using R2RML language instead of the one provided by Ontop. After reading carefully the Ontop tutorial, the decision was moved to this one for its simplicity and more human readability feature. An aspect to take into account is the other possibility of mapping provided by direct mapping. Ontop provides a functionality called "bootstrap ontology and mapping" that takes in input a relational database and creates the ontology and the mapping based on the semantics encoded in the relational schema. The direct mapping solution was tested but the result obtained didn't satisfy the expectation and also there was the idea of having an ontology that doesn't have the same schema as the database.

3 How to run the project

The following sections will show you how to set up all the required software in order to run the project. The sections are sorted in the same order as the steps you need to do for running correctly the project.

3.1 How to setup the database

The Database folder of the project contains a sql file that can be used to create all required tables of the project by simply import the file using a database design tool as MySQL Workbench or pgAdmin. Unfortunately, due to an error of MySQL Workbench, the tables don't contain already the data, so they must be imported manually (dataset in the data folder).

3.2 How to setup Ontop-cli

After downloading Ontop, run the following command supposing the user is in the project folder:

*$ ~/PATH_TO_ONTOP/ontop endpoint --ontology=Ontology/F1_ontology.owl *

--mapping=Ontology/F1_ontology.obda --properties=Ontology/F1_ontology.properties

3.3 How to run the Application/Notebook

In order to run the notebook, the user has to install:

- Jupyter / Jupyter lab
- Anaconda / Miniconda

The notebook requires some libraries in order to run the code written in it. In the project directory there is a yml file for creating a conda virtual enviroment that has all the required libraries and is ready to use for running the notebook. To create correctly the virtual environment, run the following commands supposing the user is in the project folder:

$ conda env create -f f1.yml

$ conda activate f1

(f1)$ conda install ipykernel

(f1)$ ipython kernel install –user –name=F1

(f1)$ conda deactivate

Be sure before running the code to select the correct kernel. Normally, Jupyter asks the user which kernel wants to use for running the notebook but in case it doesn't happen simply go to "Kernel -> Change kernel" and then select the correct one.

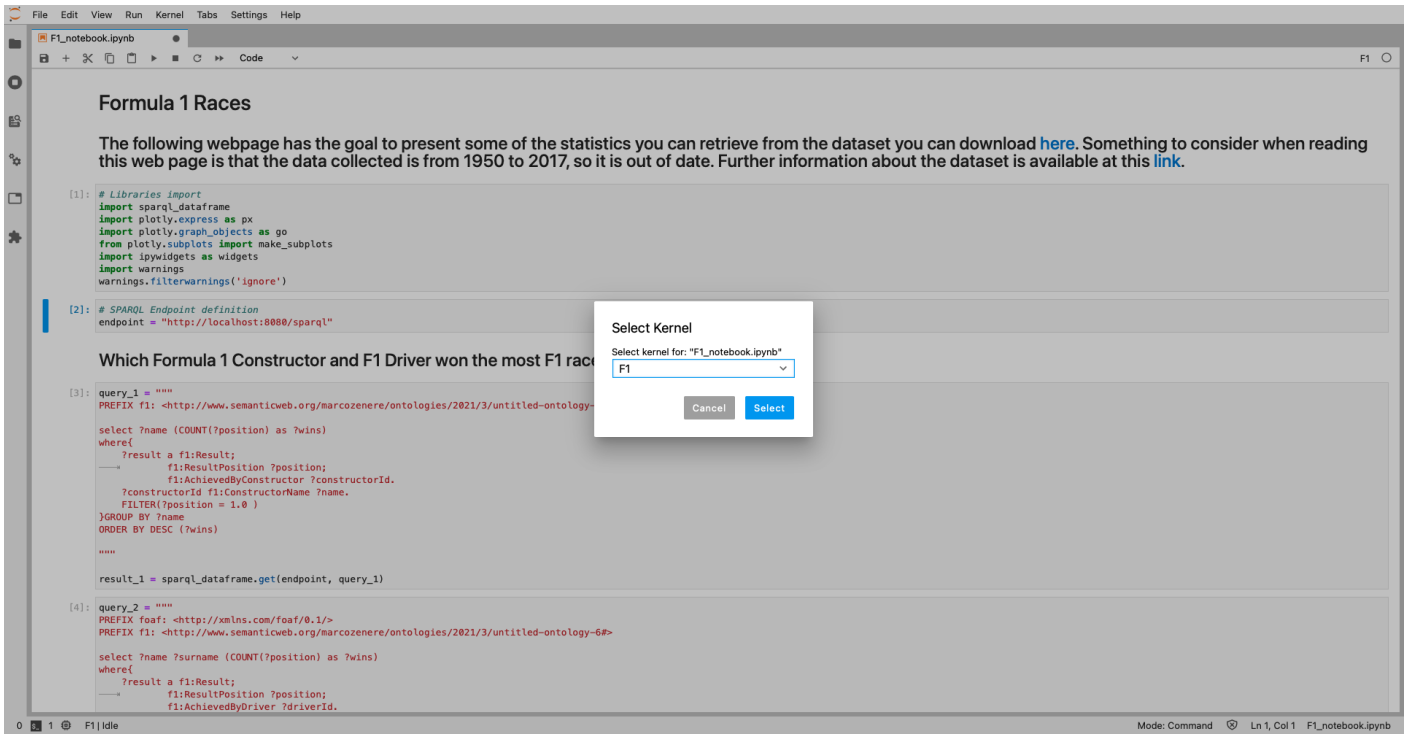

Figure 4: Kernel Selection

To run the notebook ,considering the user is in project folder, in Jupyter:

$ jupyter notebook

instead with Jupyter lab:

$ jupyter lab

3.4 Additional Information

- To run Ontop-cli, a jdbc driver is needed in order to connect with the database. The project was tested with a MySQL instance and in order to provide to the user the same experience, in the project folder is provided the jdbc driver that has to be put in jdbc folder in ontop-cli directory.
- To visualize the charts in the notebook, an additional package for Jupyter needs to be installed and can be found at this link. Follow the section named "Jupyter Notebook support" in case you installed Jupyter notebook otherwise "JupyterLab support".

4 Visualization

The user experience is provided in the html file, where there are the visualization of the sparql queries done for this project. The user can interact with the charts and gain knowledge of Formula 1 statistics without knowing what there are behind the scene.

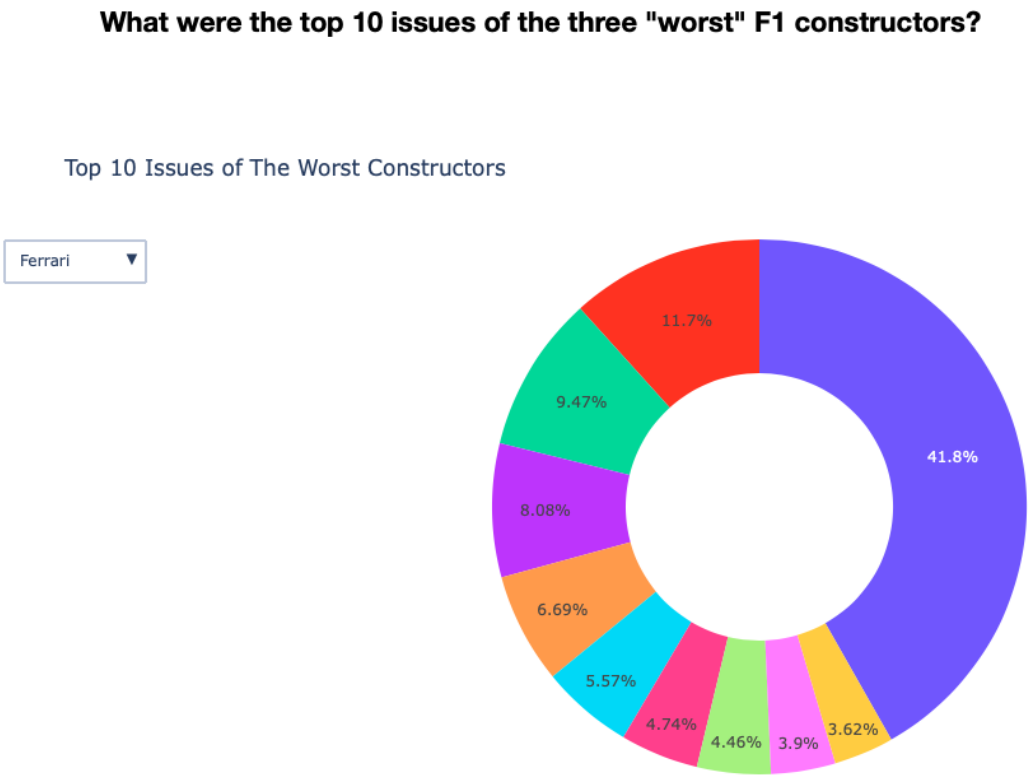

Figure 5: Spippet of the Notebook

By looking at the code, the SPARQL queries are done by using sparql_dataframe library which only requires the endpoint of the domain and the query itself. The result returned by the function is a pandas dataframe that is simple to manage in python.

```
# SPARQL Endpoint definition
endpoint = "http://localhost:8080/sparql"
```

Which Formula 1 Constructor and F1 Driver won the most F1 races?

```
query_1 = """
PREFIX f1: <http://www.semanticweb.org/marcozenere/ontologies/2021/3/untitled-ontology-6#>

select ?name (COUNT(?position) as ?wins)
where{
    ?result a f1:Result;
            f1:ResultPosition ?position;
            f1:AchievedByConstructor ?constructorId.
    ?constructorId f1:ConstructorName ?name.
    FILTER(?position = 1.0 )
}GROUP BY ?name
ORDER BY DESC (?wins)

"""

result_1 = sparql_dataframe.get(endpoint, query_1)
```

Figure 6: Spippet of the Code

All the charts in the notebook are generated using Plotly library, which gives the possibility to interact a bit with the figures.

5 Conclusion and Lesson Learned

This was the first time I developed a project in this field, so everything was new except for database and python implementation. Some design choice was made due to my basic knowledge on this topic and could be interesting to try in the future with a different mapping choice. I didn't see any limitation in the mapping language I used during the development, but I suppose there are some limitations due to its simplicity. It would be interesting to dig deep to see whether it is true or not with additional data and consequently with a bigger and more complex ontology. The project gave me the possibility to test with real data the knowledge learned from the lectures and also gave me the possibility to try new technologies and techniques as the mapping solution provided by Ontop.

References

[1] Wikipedia, *Formula 1 Wikipedia Webpage: https://en.wikipedia.org/wiki/Formula_ One*

[2] Formula 1, *Formula 1 Market Research: https://www.formula1.com/en/latest/article.formula-1s-tv-and-digital-audiences-grow-for-the-second-year-running.OqTPVNthtZKFbKqBaimKf.html*

[3] Ergast, *Ergast Webpage: http://ergast.com/mrd/*

[4] Kaggle, *Kaggle Webpage of the dataset: https://www.kaggle.com/cjgdev/formula-1-race-data-19502017*