Semantic Technologies and Linked Data – Project

Formula 1 Domain

A.Y 2020/2021

Marco Zenere

Table of Contents

Motivation Scenario

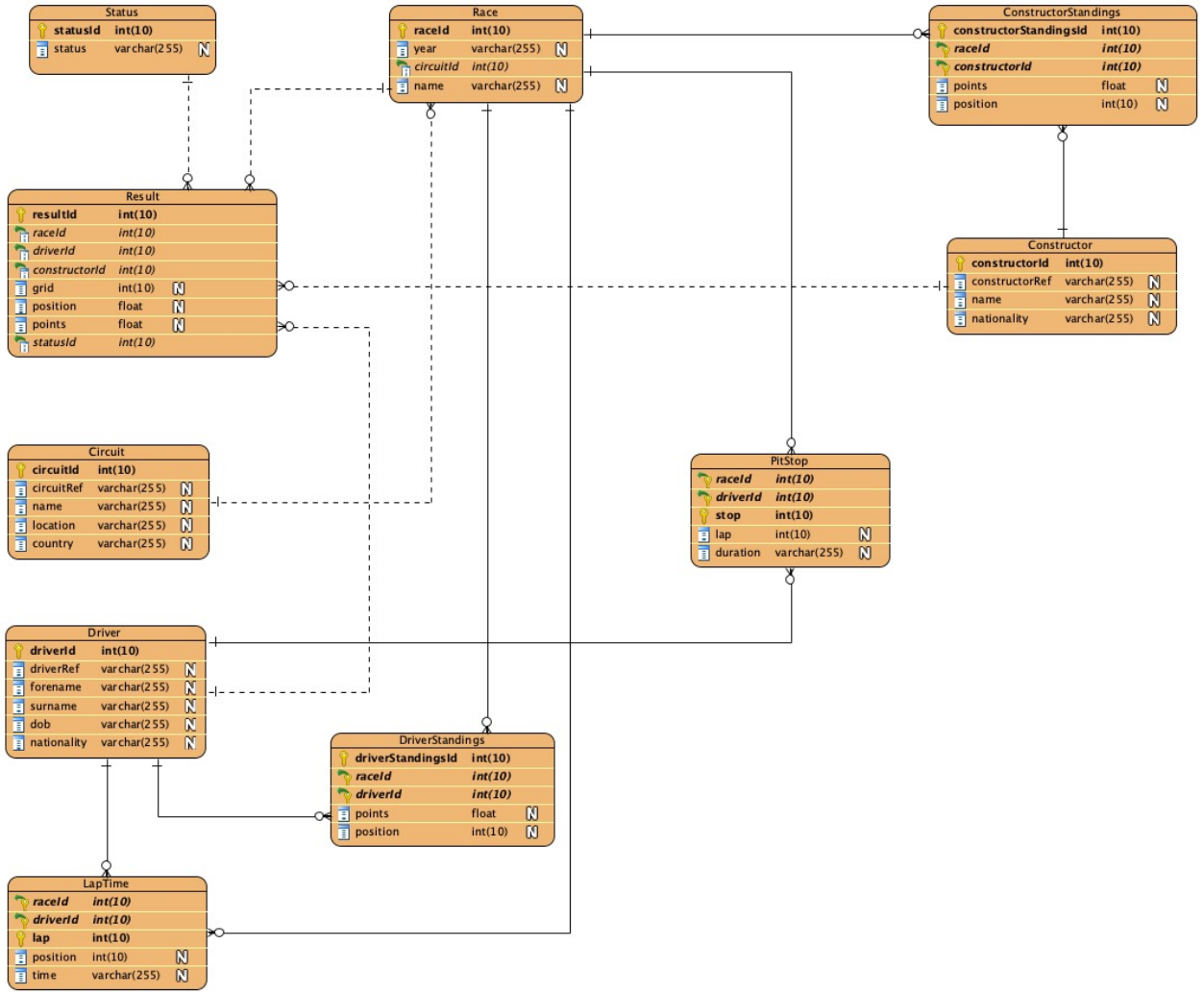
- Born in 1950

- “… the highest class of international car racing for single–seater formula racing approved by FIA”(Wikipedia)

- From the data point of view, it is an incredibly vast domain
- Data from cars amd circuits sensors, supply chain, statistics, …
- “Every F1 car contains 300 sensors which generate 1.1 million telemetry data points per second … ”(AWS)

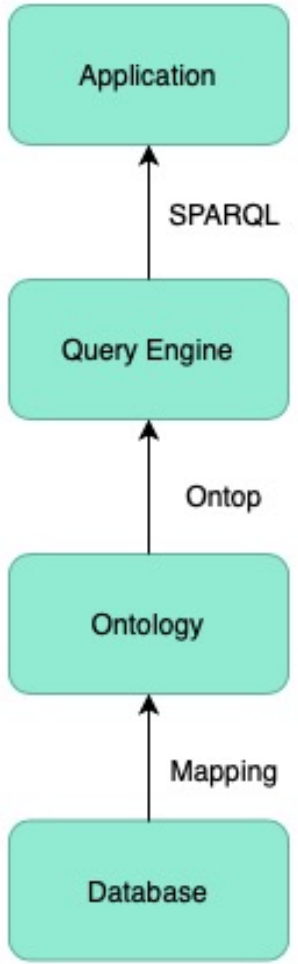
Dataset

- Formula 1 doesn't provide officialy the data
- A fan named Chris Newell provides F1 data from 1950 to 2017
- The original dataset has 13 tables

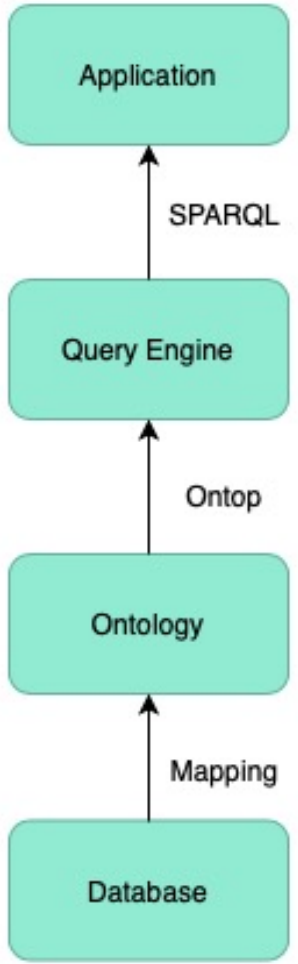


Implementation

- System architecture is composed by 4 layers:
  - Database Layer
  - Ontology Layer
  - Query Engine
  - Application Layer

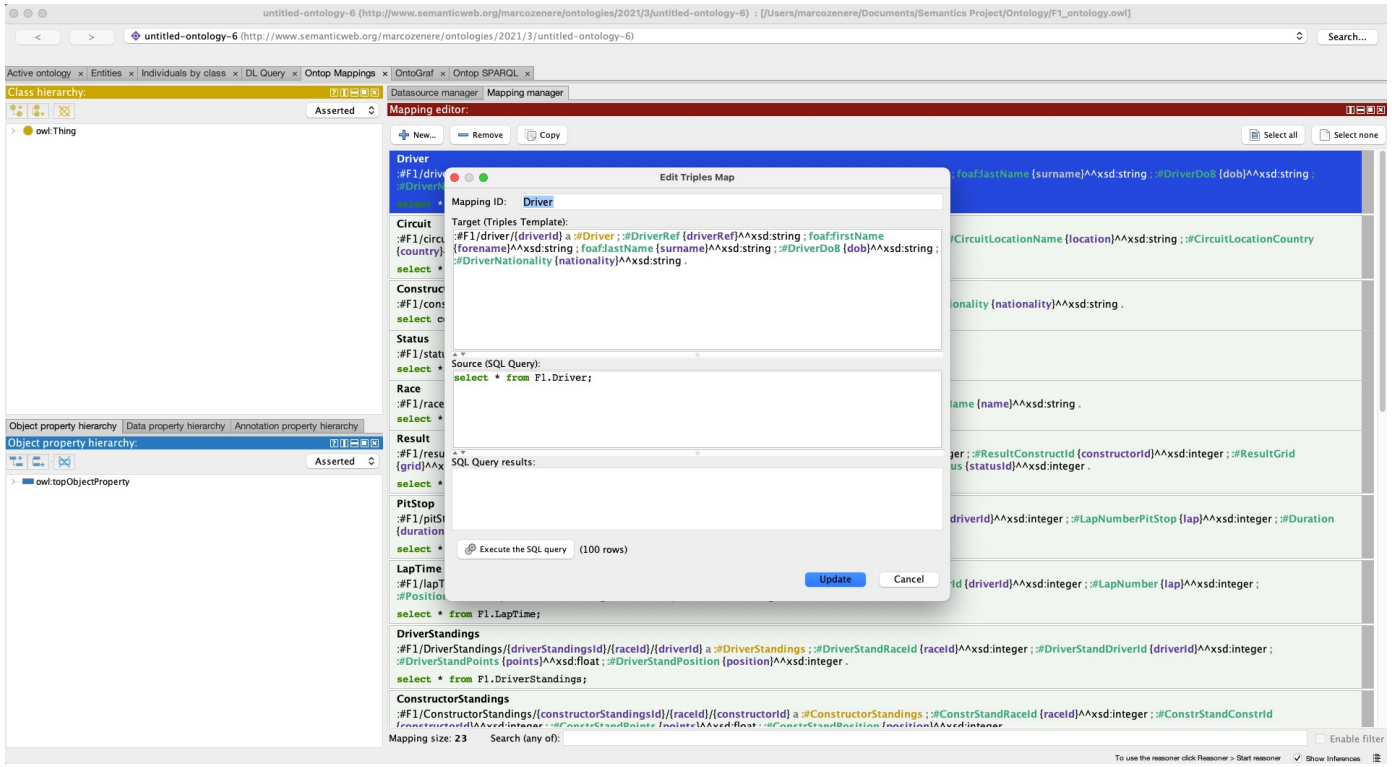


- Database hosted in MySQL
- Ontology in RDFS / OWL designed in Protégé
- Ontop used as Reasoner
- Application written in Python

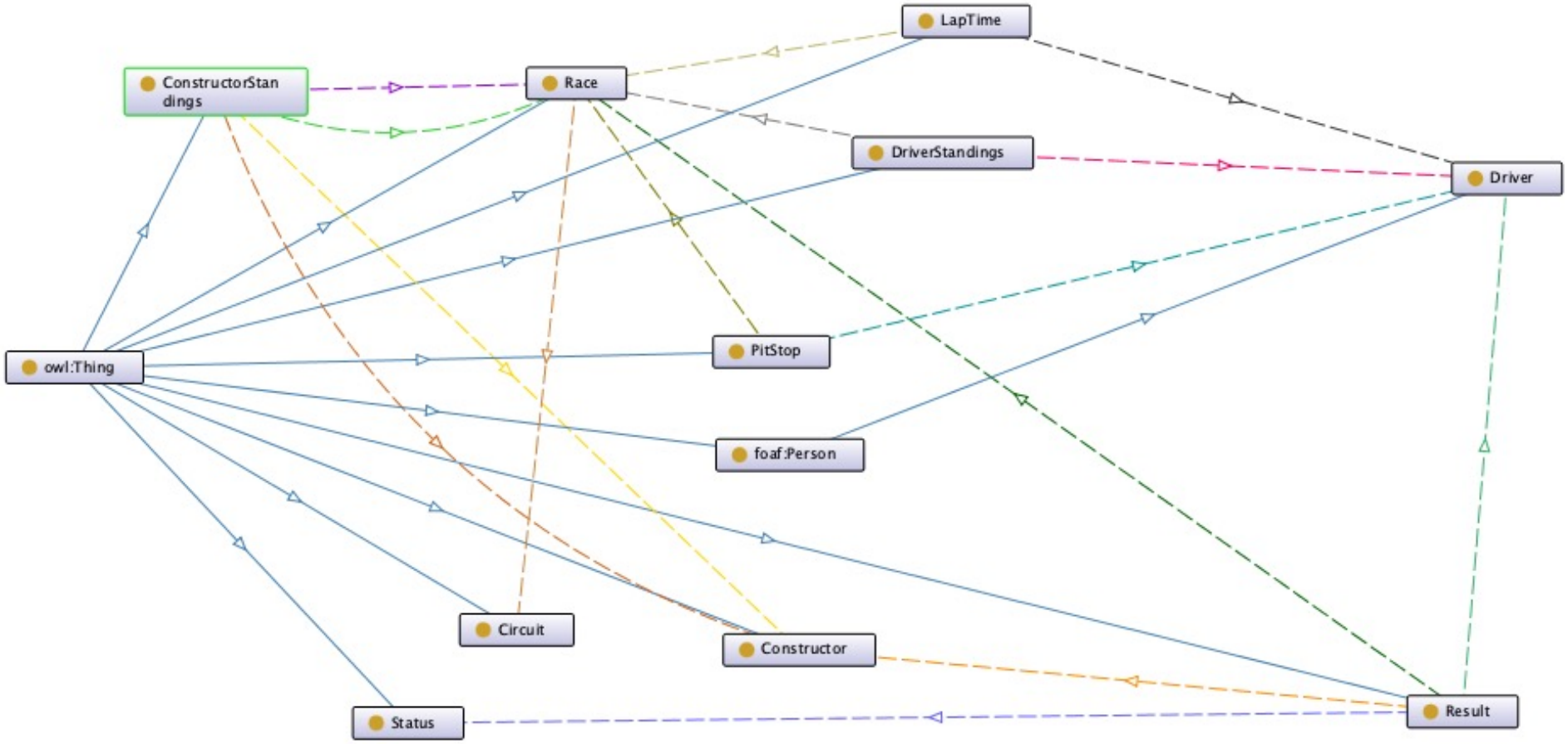


Mapping

- Ontop plugin in Protégé provides the capability to generate ontology and mapping (Direct Mapping) from a database
- Ontop Mappings has a better human readability feature and simpler to write than R2RML



Ontology


LapTime
ConstructorStan dings
Race
DriverStandings
Driver
owl:Thing
PitStop
foaf:Person
Circuit
Constructor
Status
Result

Application

- Jupyter Notebook written in Python
- Sparql Dataframe for connecting to the sparql endpoint and manage the query result

```
# 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)
```

Application Visualization

- The application includes charts to improve the readability of the results
- All the plots are generated using Plotly library, which provides interesting features to have interactive figures

Conclusion

- The mapping solution used in this project works great for the purpose I wanted to achieve
- It would be interesting to have more data in order to have a more complex domain to understand the limits of the mapping choice
- The project gave me a deeper knowledge of the topics encountered during the lectures, and gave me the possibility to try technologies I have never tried before

Live Demo