Task 2 : Enrich your Knowledge Graph through SPARQL

Construct queries

Query1

The first CONSTRUCT query enriches the knowledge graph by defining triples that include, for each concert, the track that is played first in the program entry. The constructed triples contain the concert's and track's title. A new semantic relationship is defined, the ":playedFirstInConcert" which provides with the specific information of the first track to be played in a concert. This can simplify a query that is trying to retrieve this information, while the selection of the first track is meaningful, because the track that opens a concert is usually of special interest.

Query2

The second CONSTRUCT query identifies all events scheduled after 2023. It creates triples that contain this information as ":eventAfter2023". In addition, it includes the full information given by the knowledge graph regarding the event, namely date, time, concert, theater and orchestra. That way, the data obtain a new semantic knowledge that corresponds to a subset of events planed after a specific date.

Query3

The third CONSTRUCT query creates triples that represent a new semantic knowledge associated to the composers who created a track at an age older than 60. This facilitates a query for obtaining the names and information of older composers, without having to repeat the filtering for age. This subset of data can be interesting knowledge to be acquired by the user who run a query because they represent a group of the most experienced composers. Prior to setting up the query, a property 'age' was defined as a column in the table Composer of the music-ontology database. This was done in postgresql by manually altering the table and inserting a new column.

Update query

At first, a SPARQL query was executed on Protege with the aim to select one Composer as the Best (arbitrary selection by using the primary key), delete their last name and insert it again with the suffix LastName + "_TheBest". Since Ontop is only read RDF system, there is no option to write on the table, which would allow for INSERT or DELETE. Therefore, a screenshot with the query is included in the zip file. However, the actual update of the table was done in pgAdmin4 GUI where the following query was executed:

```
UPDATE composer
SET lastName = lastName || '_TheBest'
WHERE composer_id = 1000;
```