# Building & Mining Knowledge Graphs

## (KEN4256)

## Assignment 1 part 2:

### Mapping, Linking and Querying

Due date: March 6 18:00 CET

## Description

You have already learned how to build a music Knowledge Graph (KG). This part of the assignment, you will transform tabular data to a KG and link it to open linked data, and query the resulting KG.

You should perform the tasks in groups. The group members should remain the same as in Assignment 1 part 1.

## Data

Attached is the music csv generated from
https://www.kaggle.com/datasets/leonardopena/top-spotify-songs-from-20102019-by-year

| Index | title | artist | top genre | year |
|---|---|---|---|---|
| 0 | Hey, Soul Sister | Train | neo mellow | 2010 |
| 1 | Love The Way You Lie | Eminem | detroit hip hop | 2010 |
| 2 | TiK ToK | Kesha | dance pop | 2010 |
| 3 | Bad Romance | Lady Gaga | dance pop | 2010 |
| 4 | Just the Way You Are | Bruno Mars | pop | 2010 |

……

# Tasks

1. Write YARRRML or RML mappings to convert the given CSV data to a KG with following the RDF structure and prefix-namespace

```
@prefix ex: <http://example.org/> .
@prefix schema: <https://schema.org/>
@prefix mo: <http://purl.org/ontology/mo/>
```

   a. Triples represent information of  the songs, e.g.

```
ex:443 a schema:MusicRecording .
ex:443 schema:byArtist ex:Ed_Sheeran .
ex:443 schema:genre ex:pop .
ex:443 schema:datePublished "2016"^^xsd:gYear.
ex:443 rdfs:label "Shape of You"@en .
```

   b. Triples represent information of the artists, e.g.

```
ex:Ed_Sheeran a schema:Person .
ex:Ed_Sheeran rdfs:label "Ed Sheeran"@en .
```

   c. Triples represent information of the genre, e.g.

```
ex:pop a mo:Genre .
ex:pop rdfs:label "pop"@en .
```

2. Linking KG to DBpedia using LIMES
   a. All entities with an rdfs:label should be linked to DBpedia entities
   b. Add triples with `owl:sameAs` to represent your linking
3. Add the birth date and birth place information of each artist to your KG by querying DBpedia. An example of retrieving all DBpedia information of Ed Sheeran.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
SELECT distinct *
WHERE
{
  ?subject rdfs:label "Ed Sheeran"@en .
  ?subject owl:sameAs ?s.
  SERVICE <https://dbpedia.org/sparql> {
    ?s ?p ?o. }

}
```

4 Provide SPARQL queries to answer the following questions using the resulting integrated KG:

a. Return a list of artists and their names who produce songs with genres other than "pop" and "dance pop."
b. Return a list of songs released in 2016 by artists born before 1990.
c. Who is the artist that has produced the greatest number of songs?
d. Return a list of artists born in the USA, sorted by the number of songs they have produced.
e. Find artists whose song names contain the word "love" and sort the artists by the resulting number of songs.

You may use the results of multiple separate queries to answer each question if you prefer, although it is possible to answer each using a single query.
To start a local SPARQL endpoint, install the rdflib-endpoint Python package with pip:

*pip install "rdflib-endpoint==0.1.6"*

To start SPARQL endpoint on http://localhost:8000

*rdflib-endpoint serve [local triple files]*

# Materials might help you

- https://dbpedia.org/page/Chun-Li_(song) or https://www.wikidata.org/wiki/Q51840972
  You should learn from these links:
  - Type of Song/Artist/Genre in DBpedia/Wikidata
  - Links between artists, songs and genres
- https://rv.aksw.org/Projects/LIMES LIMES could be used to link KG to KG
  - http://dice-group.github.io/LIMES/#/ this manual could help you when you try to link KG to DBpedia
- Lab 3 slides https://canvas.maastrichtuniversity.nl/courses/12396/modules/items/450668

# Deliverables

**One student per group** submit the following file (also on Student Portal):

- Your report explaining your methodology and results
- YARRRML and/or RML file
- Additional links generated by Limes
- Your final KG (as ntriples format)
- A **single** text file (.txt extension) containing all your SPARQL queries **clearly marked with the question number that it answers**

## Questions and comments:

Remzi Celebi remzi.celebi@maastrichtuniversity.nl
Xu Wang: xu.wang@maastrichtuniversity.nl