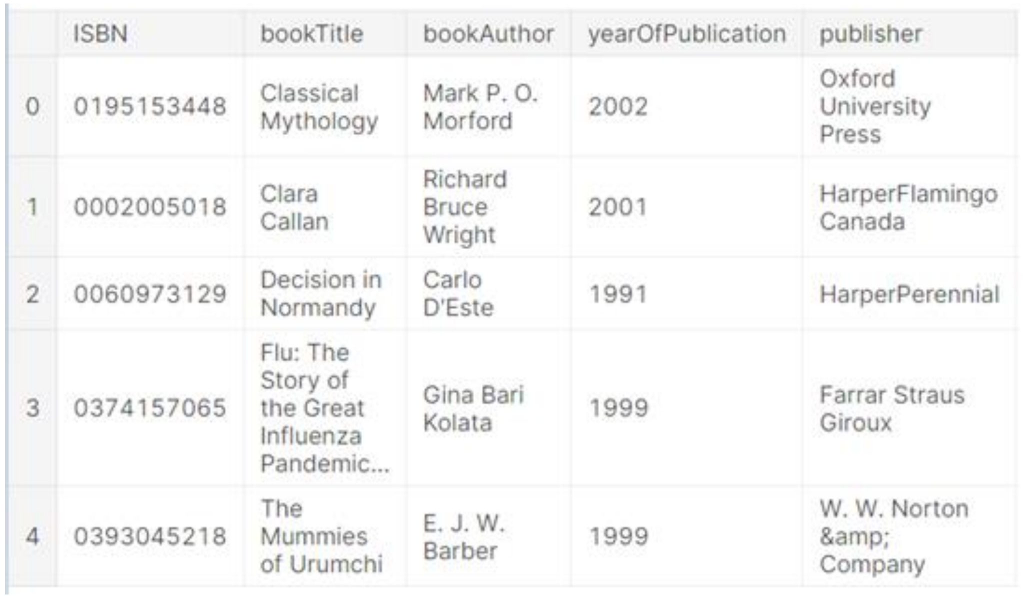
Knowledge Graphs

Transformation of a static dataset into a knowledge graph

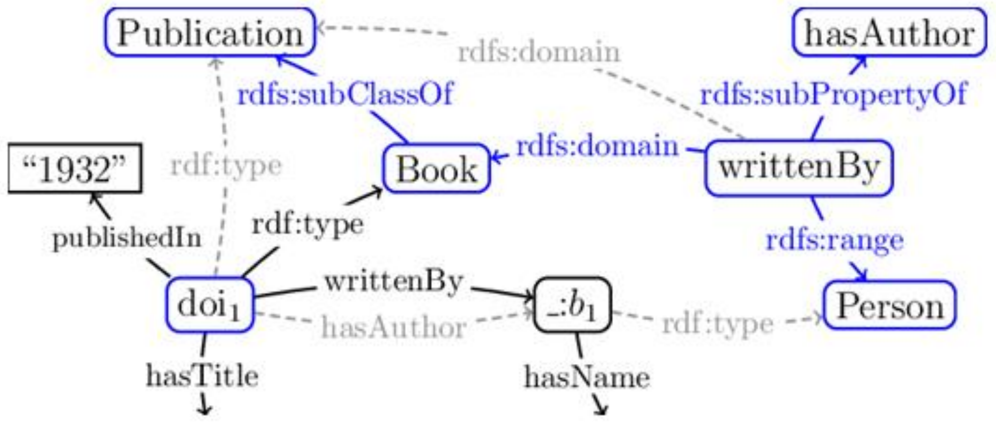
Illia Tesliuk

Problem Statement

Transform a tabular dataset (e.g., CSV, XML) into a knowledge graph (e.g., RDF graph)



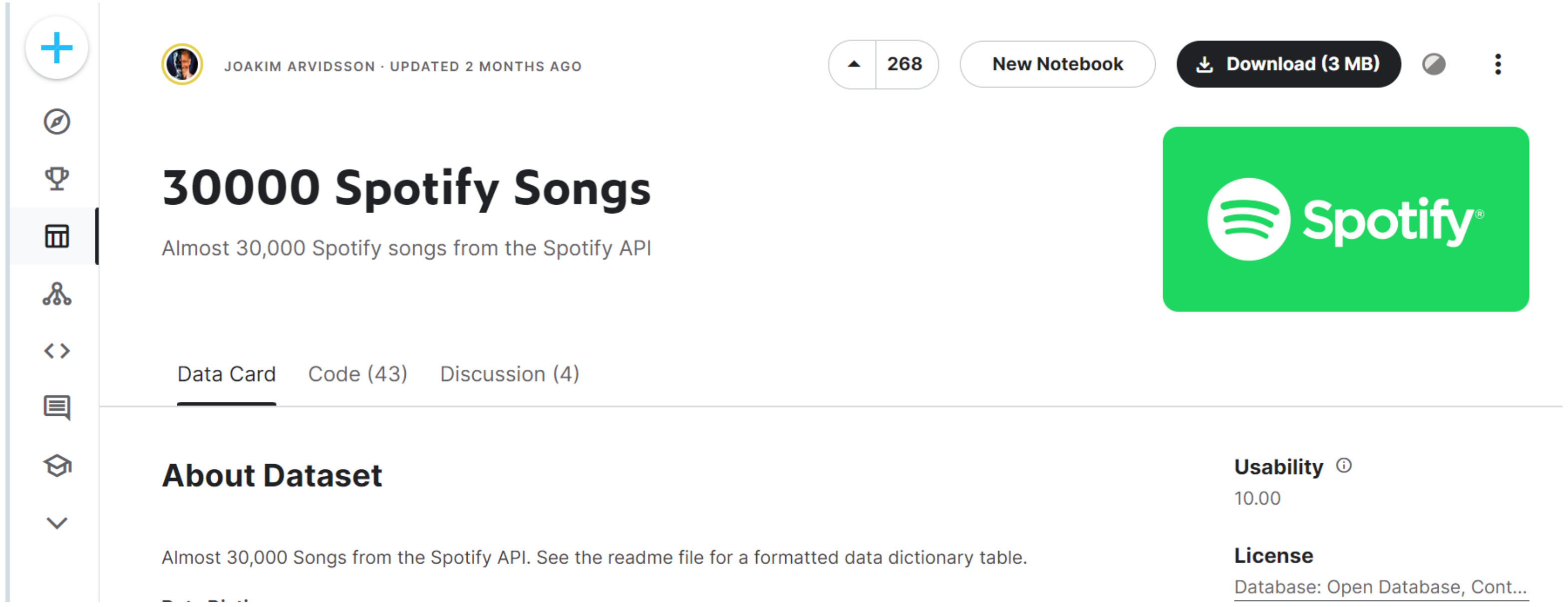
	ISBN	bookTitle	bookAuthor	yearOfPublication	publisher
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company



Motivation

- Knowledge graphs:
  - capture semantic relationships between different entities, so more meaningful interpretation of the information can be obtained
  - support query languages (e.g. SPARQL) that enable efficient retrieval of specific patterns from the graph
  - provide a means to integrate data from diverse sources by establishing links between related entities
  - support inferencing and reasoning capabilities
  - are well-suited for representing domain-specific knowledge in various fields

Dataset: 30,000 Spotify Songs


JOAKIM ARVIDSSON · UPDATED 2 MONTHS AGO

268 New Notebook Download (3 MB)

30000 Spotify Songs

Almost 30,000 Spotify songs from the Spotify API

Data Card Code (43) Discussion (4)

About Dataset

Almost 30,000 Songs from the Spotify API. See the readme file for a formatted data dictionary table.

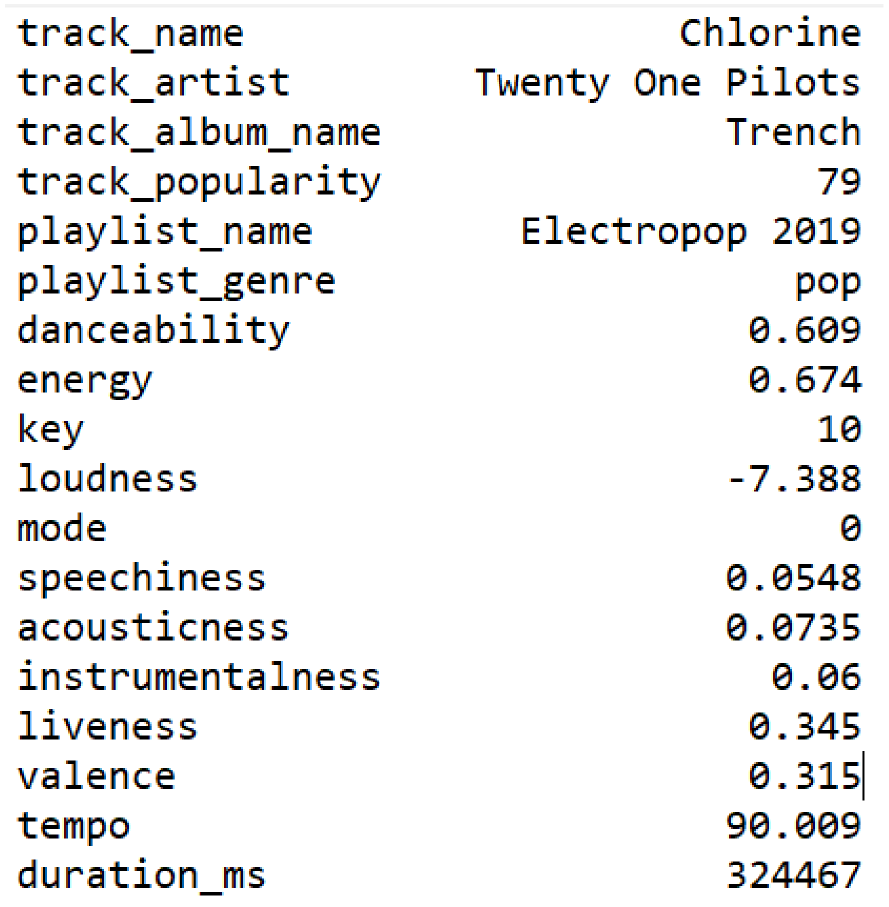
Usability 10.00

License Database: Open Database, Cont...

Source: https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs

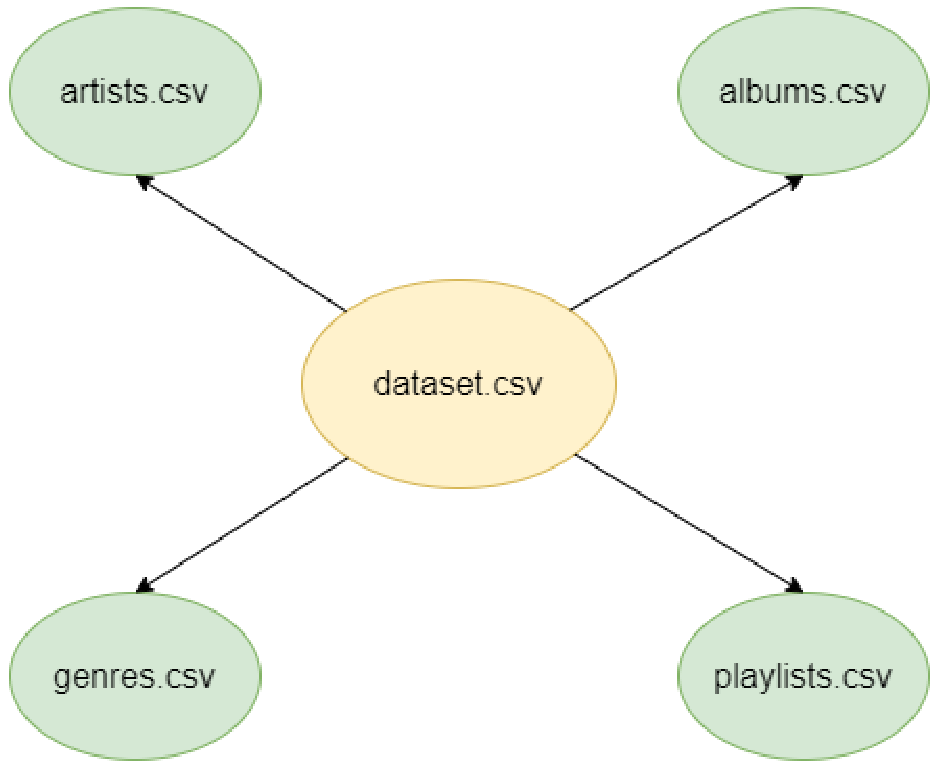
Dataset: 30,000 Spotify Songs

- **Danceability** - how "danceable" the song is
- **Energy** – scale of intensity, activity
- **Speechiness** - presence of spoken words in a track
- **Acousticness** – uses acoustic/electrical instruments
- **Liveness** – studio/live recording
- **Valence** - musical "positiveness", happy or sad
- **Mode** – major/minor
- **Loudness** – average value in dB



```
track_name                          Chlorine
track_artist               Twenty One Pilots
track_album_name                      Trench
track_popularity                          79
playlist_name                Electropop 2019
playlist_genre                           pop
danceability                           0.609
energy                                 0.674
key                                       10
loudness                              -7.388
mode                                       0
speechiness                           0.0548
acousticness                          0.0735
instrumentalness                        0.06
liveness                               0.345
valence                                0.315
tempo                                 90.009
duration_ms                           324467
```

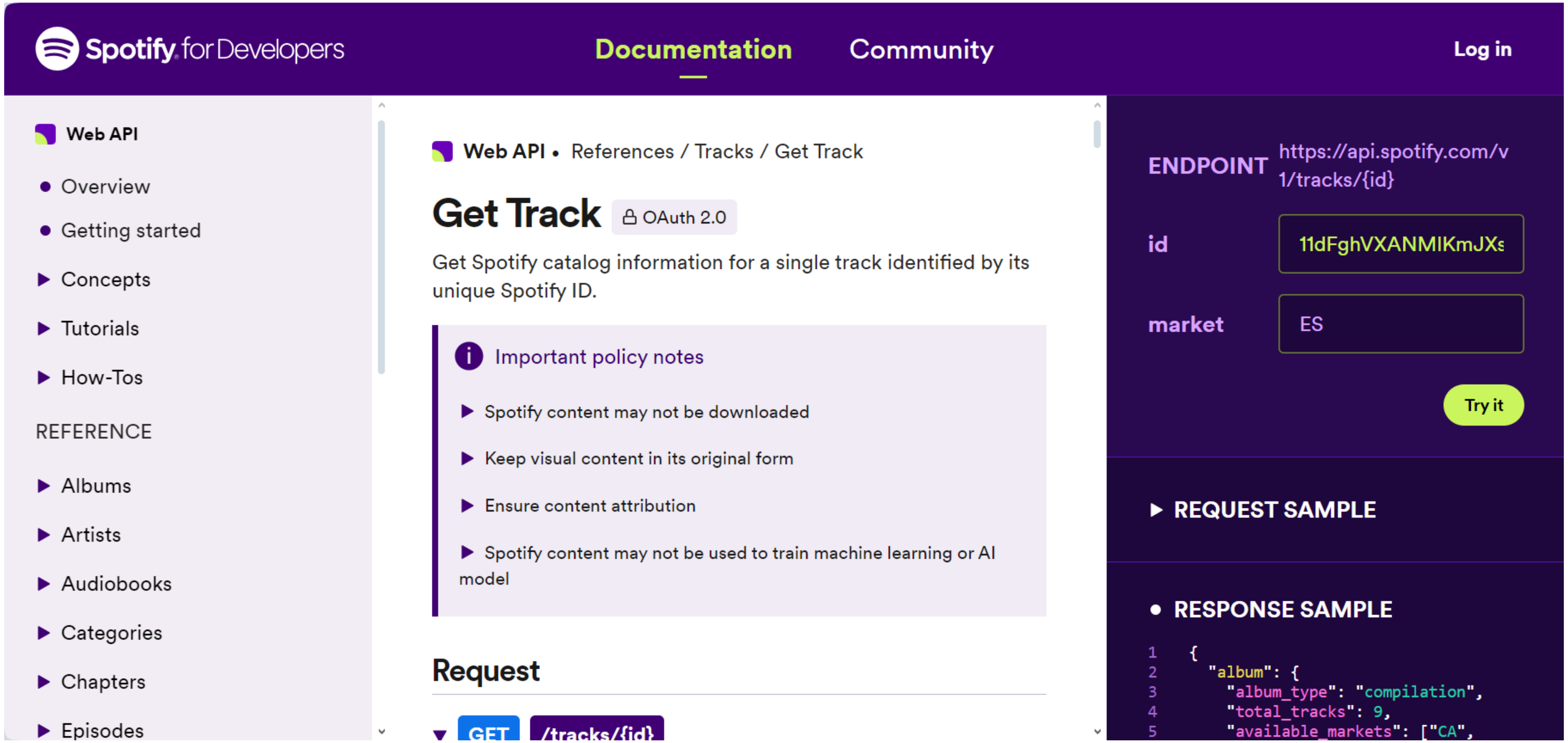
Dataset: split



Spotify Recommendation System

- **Content-based filtering** is a part of Spotify's recommendation system used to enhance personalization
  - Collects metadata (release date, label, etc.)
  - Performs raw audio analysis to translate sound into measurable sonic characteristics
  - Extracts semantic information from the lyrics with the help of NLP
- Combines the outputs of several independent algorithms to generate higher-level vectors (think of these as mood, genre, style tags, etc.)
- Final recommendations are made by combining results of content-based and user-based analysis

Spotify API



Spotify for Developers | Documentation | Community | Log in

Web API

- Overview
- Getting started
- ▶ Concepts
- ▶ Tutorials
- ▶ How-Tos

REFERENCE

- ▶ Albums
- ▶ Artists
- ▶ Audiobooks
- ▶ Categories
- ▶ Chapters
- ▶ Episodes

Web API • References / Tracks / Get Track

Get Track OAuth 2.0

Get Spotify catalog information for a single track identified by its unique Spotify ID.

Important policy notes

- ▶ Spotify content may not be downloaded
- ▶ Keep visual content in its original form
- ▶ Ensure content attribution
- ▶ Spotify content may not be used to train machine learning or AI model

Request

GET /tracks/{id}

ENDPOINT	https://api.spotify.com/v1/tracks/{id}
id	11dFghVXANMIKmJXs
market	ES

Try it

▶ REQUEST SAMPLE

● RESPONSE SAMPLE

```
{
  "album": {
    "album_type": "compilation",
    "total_tracks": 9,
    "available_markets": ["CA",
```

Source: https://developer.spotify.com

Ontologies

- Music Ontology (mo):
  - http://purl.org/ontology/mo/
  - Provides main concepts and properties for describing music
  - Classes: **Track**, **Record**, **MusicArtist**, **Genre**
  - Properties: **duration**
- Playlist Ontology (po):
  - http://purl.org/net/po#
  - An ontology for describing playlists, playlist entries and songs
  - Classes: **Playlist**

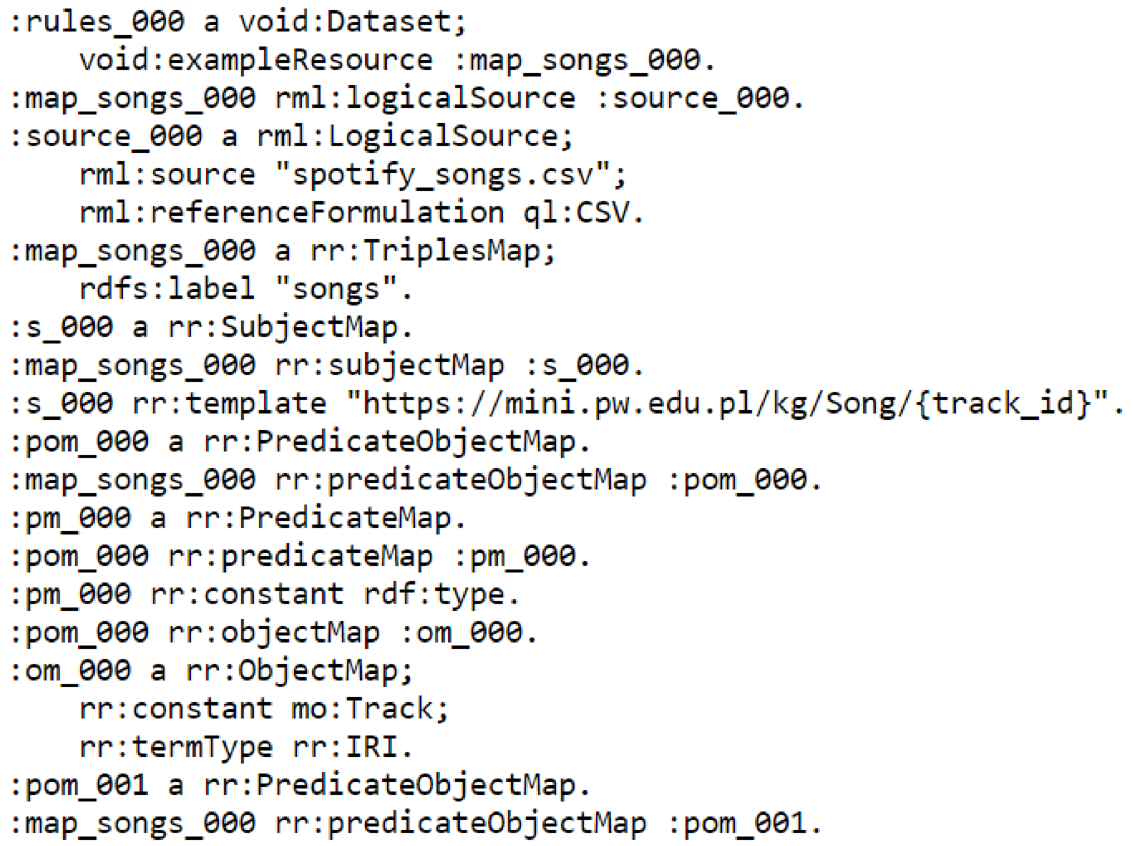
Custom Ontology (spo)

```
spo:genre a rdf:Property ;
    rdfs:label "genre"@en ;
    rdfs:domain plo:Playlist ;
    rdfs:range mo:Genre .

spo:danceability a rdf:Property ;
    rdfs:label "danceability"@en ;
    rdfs:domain mo:Track ;
    rdfs:range rdfs:Literal .

spo:popularity a rdf:Property ;
    rdfs:label "popularity"@en ;
    rdfs:domain mo:Track ;
    rdfs:range rdfs:Literal .
```

```
@prefix spo: <https://mini.pw.edu.pl/kg/spo#> .

spo:artist a rdf:Property ;
    rdfs:label "artist"@en ;
    rdfs:domain mo:Track ;
    rdfs:range mo:MusicArtist .

spo:album a rdf:Property ;
    rdfs:label "album"@en ;
    rdfs:domain mo:Track ;
    rdfs:range mo:Record .

spo:playlist a rdf:Property ;
    rdfs:label "playlist"@en ;
    rdfs:domain mo:Track ;
    rdfs:range plo:Playlist .
```

RML (RDF Mapping Language)

- Designed for expressing mappings between different data formats, (CSV, XML) and RDF graphs.
- Provides a way to define how the data in non-RDF formats should be mapped to RDF triples of subjects, predicates and objects.



```
:rules_000 a void:Dataset;
    void:exampleResource :map_songs_000.
:map_songs_000 rml:logicalSource :source_000.
:source_000 a rml:LogicalSource;
    rml:source "spotify_songs.csv";
    rml:referenceFormulation ql:CSV.
:map_songs_000 a rr:TriplesMap;
    rdfs:label "songs".
:s_000 a rr:SubjectMap.
:map_songs_000 rr:subjectMap :s_000.
:s_000 rr:template "https://mini.pw.edu.pl/kg/Song/{track_id}".
:pom_000 a rr:PredicateObjectMap.
:map_songs_000 rr:predicateObjectMap :pom_000.
:pm_000 a rr:PredicateMap.
:pom_000 rr:predicateMap :pm_000.
:pm_000 rr:constant rdf:type.
:pom_000 rr:objectMap :om_000.
:om_000 a rr:ObjectMap;
    rr:constant mo:Track;
    rr:termType rr:IRI.
:pom_001 a rr:PredicateObjectMap.
:map_songs_000 rr:predicateObjectMap :pom_001.
```

YARRRML

- Human readable text-based representation for declarative generation rules
- It is a subset of [YAML], a widely used data serialization language designed to be human-friendly.
- User defines set of YARRRML rules in YAML file, and a parser converts them to RML rules

```
mappings:
  songs:
    sources:
      - ['spotify_songs.csv~csv']
    s: song:$(track_id)
    po:
      - [a, mo:Track]
      - [foaf:name, $(track_name)]
      - p: mo:duration
        o:
          value: $(duration_ms)
          datatype: xsd:double
      - p: spo:popularity
        o:
          value: $(track_popularity)
          datatype: xsd:integer
      - p: spo:danceability
        o:
          value: $(danceability)
          datatype: xsd:double
```

YARRRML Parser

This library allows to convert YARRRML rules to RML or R2RML rules.

RMLMapper

Maven Central v6.5.1

The RMLMapper executes RML rules to generate Linked Data. It is a Java library, which is available via the command line (API docs online). The RMLMapper loads all data in memory, so be aware when working with big datasets.

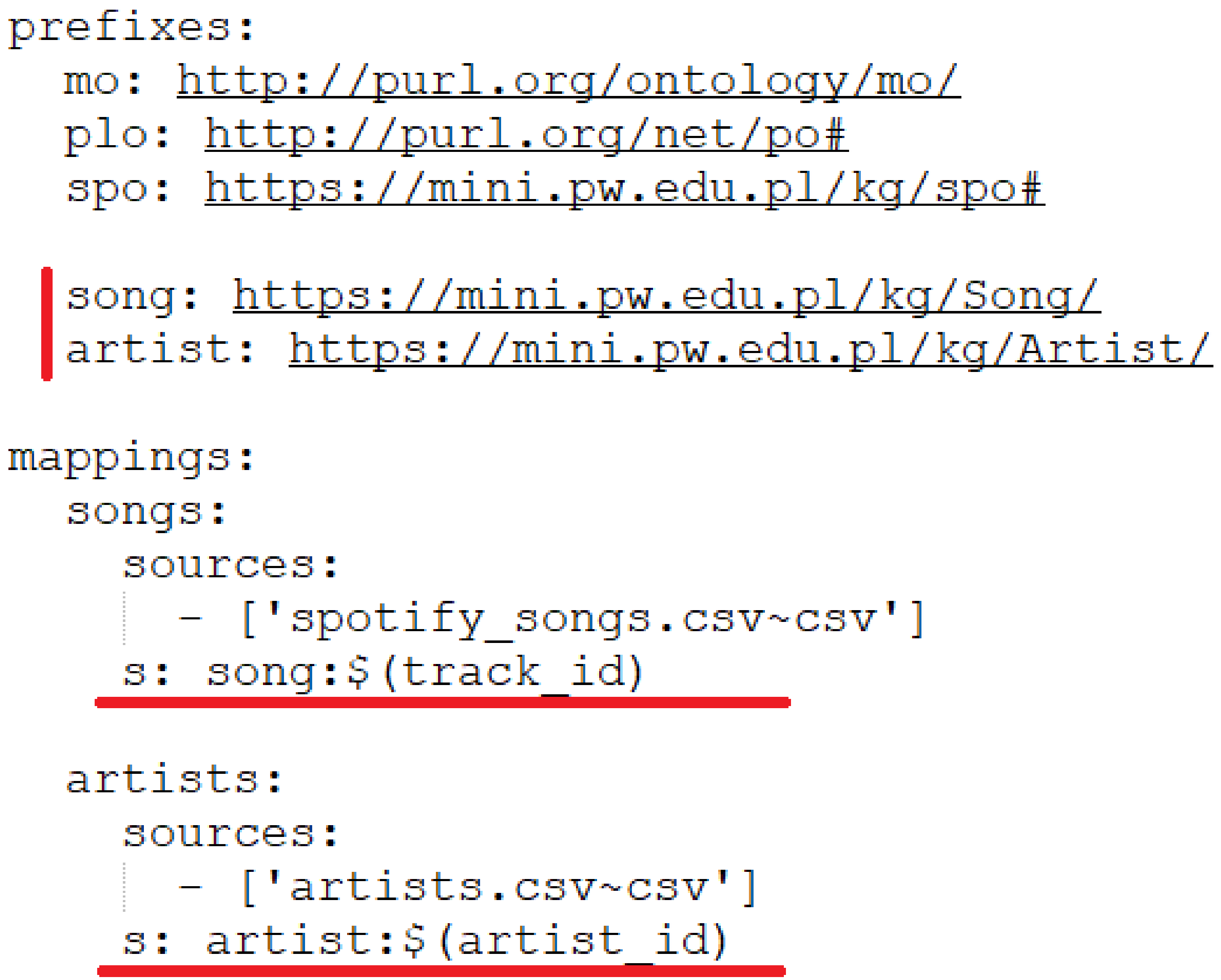
```
yarrrml-parser -i rules.yml -o rules.rml.ttl
java -jar /path/to/rmlmapper.jar -m rules.rml.ttl
```

YARRRML: sources

```
prefixes:
  mo: http://purl.org/ontology/mo/
  plo: http://purl.org/net/po#
  spo: https://mini.pw.edu.pl/kg/spo#

mappings:
  songs:
    sources:
      - ['spotify_songs.csv~csv']

  artists:
    sources:
      - ['artists.csv~csv']

  playlists:
    sources:
      - ['playlists.csv~csv']

  albums:
    sources:
      - ['albums.csv~csv']

  genres:
    sources:
      - ['genres.csv~csv']
```

YARRRML: subjects

- Column values are retrieved using the following syntax: **$(csv_column_name)**



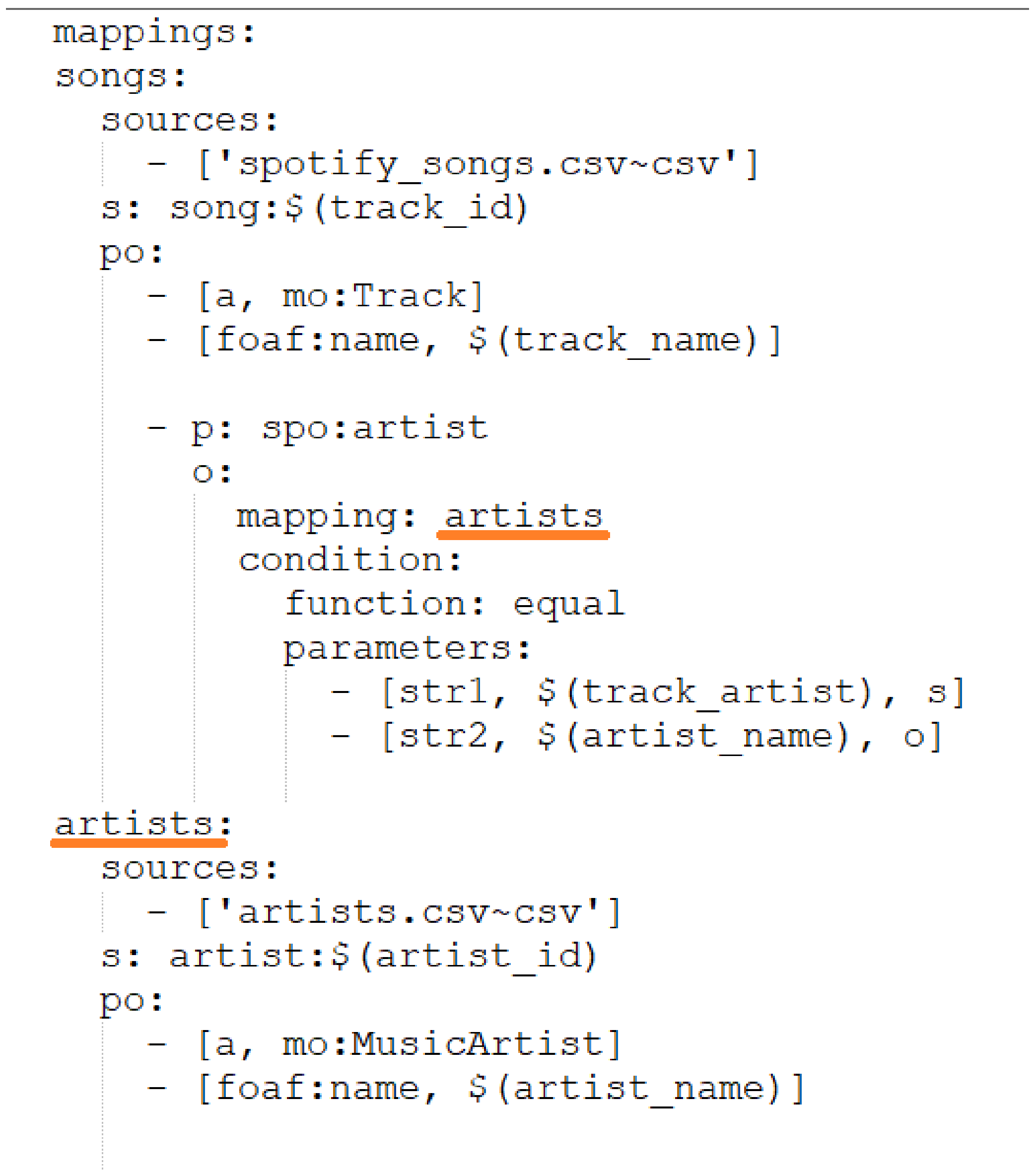
```
prefixes:
  mo: http://purl.org/ontology/mo/
  plo: http://purl.org/net/po#
  spo: https://mini.pw.edu.pl/kg/spo#

  song: https://mini.pw.edu.pl/kg/Song/
  artist: https://mini.pw.edu.pl/kg/Artist/

mappings:
  songs:
    sources:
      - ['spotify_songs.csv~csv']
    s: song:$(track_id)

  artists:
    sources:
      - ['artists.csv~csv']
    s: artist:$(artist_id)
```

YARRRML: predicates, objects

```
mappings:
  songs:
    sources:
      - ['spotify_songs.csv~csv']
    s: song:$(track_id)
    po:
      - [a, mo:Track]
      - [foaf:name, $(track_name)]
      - p: mo:duration
        o:
          value: $(duration_ms)
          datatype: xsd:double
      - p: mo:bpm
        o:
          value: $(tempo)
          datatype: xsd:double
      - p: spo:popularity
        o:
          value: $(track_popularity)
          datatype: xsd:integer
      - p: spo:danceability
        o:
          value: $(danceability)
          datatype: xsd:double
```

YARRRML: links between entities

- 'spotify_songs.csv' contains artist names in the '**track_artist**' column
- 'artists.csv' saves names in the '**artist_name**' column
- YARRRML allows to create a mapping between 2 sources and join data by comparing these two column

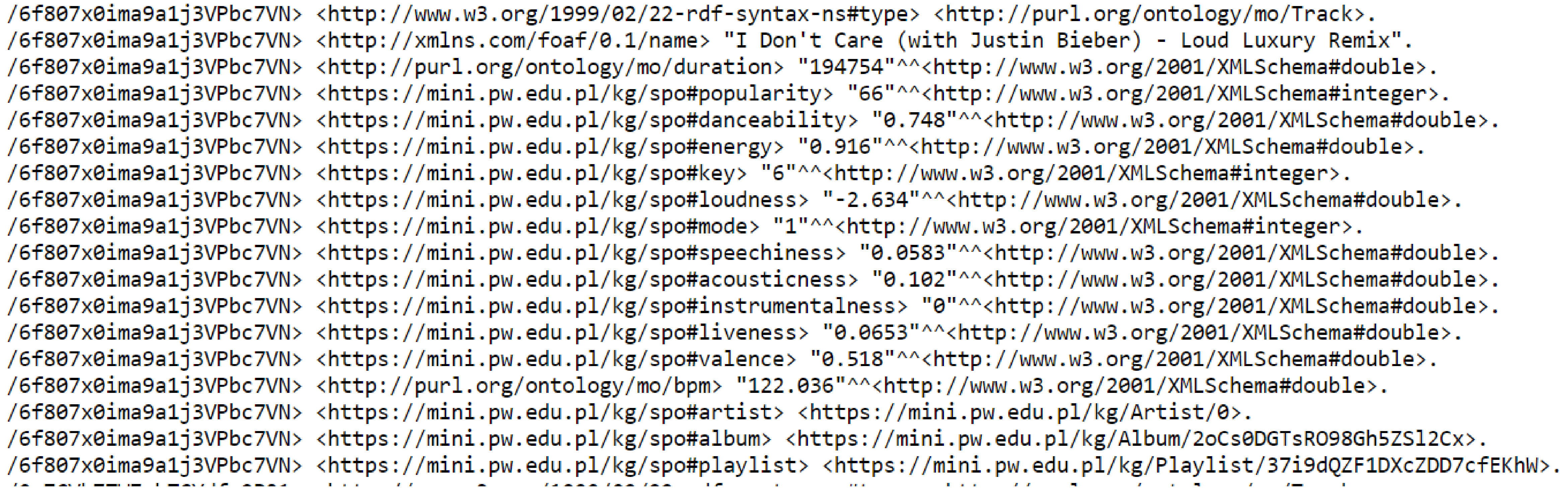


```
mappings:
songs:
  sources:
    - ['spotify_songs.csv~csv']
  s: song:$(track_id)
  po:
    - [a, mo:Track]
    - [foaf:name, $(track_name)]

    - p: spo:artist
      o:
        mapping: artists
        condition:
          function: equal
          parameters:
            - [str1, $(track_artist), s]
            - [str2, $(artist_name), o]

artists:
  sources:
    - ['artists.csv~csv']
  s: artist:$(artist_id)
  po:
    - [a, mo:MusicArtist]
    - [foaf:name, $(artist_name)]
```

Results

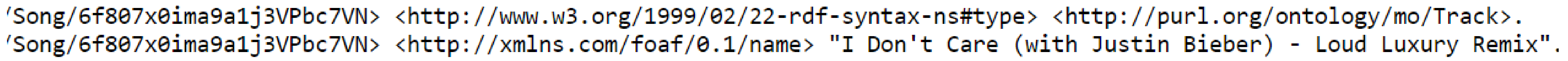
- Resulting knowledge graph consists of:
  - 605,253 triples
  - 28,356 songs
  - 10,693 artists
  - 22,543 albums
  - 471 playlists
  - 30 genres

Results: song



```
/6f807x0ima9a1j3VPbc7VN> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Track>.
/6f807x0ima9a1j3VPbc7VN> <http://xmlns.com/foaf/0.1/name> "I Don't Care (with Justin Bieber) - Loud Luxury Remix".
/6f807x0ima9a1j3VPbc7VN> <http://purl.org/ontology/mo/duration> "194754"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#popularity> "66"^^<http://www.w3.org/2001/XMLSchema#integer>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#danceability> "0.748"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#energy> "0.916"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#key> "6"^^<http://www.w3.org/2001/XMLSchema#integer>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#loudness> "-2.634"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#mode> "1"^^<http://www.w3.org/2001/XMLSchema#integer>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#speechiness> "0.0583"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#acousticness> "0.102"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#instrumentalness> "0"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#liveness> "0.0653"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#valence> "0.518"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <http://purl.org/ontology/mo/bpm> "122.036"^^<http://www.w3.org/2001/XMLSchema#double>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#artist> <https://mini.pw.edu.pl/kg/Artist/0>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#album> <https://mini.pw.edu.pl/kg/Album/2oCs0DGTsRO98Gh5ZSl2Cx>.
/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#playlist> <https://mini.pw.edu.pl/kg/Playlist/37i9dQZF1DXcZDD7cfEKhW>.
```

Results: main fields

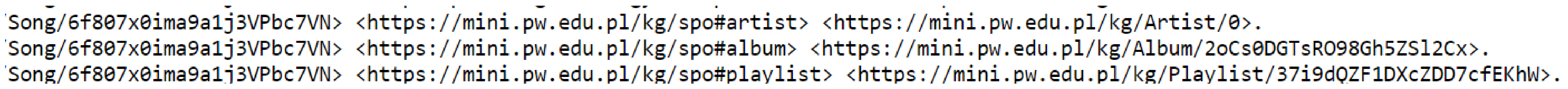


```
/Song/6f807x0ima9a1j3VPbc7VN> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Track>.
/Song/6f807x0ima9a1j3VPbc7VN> <http://xmlns.com/foaf/0.1/name> "I Don't Care (with Justin Bieber) - Loud Luxury Remix".
```

Results: song properties

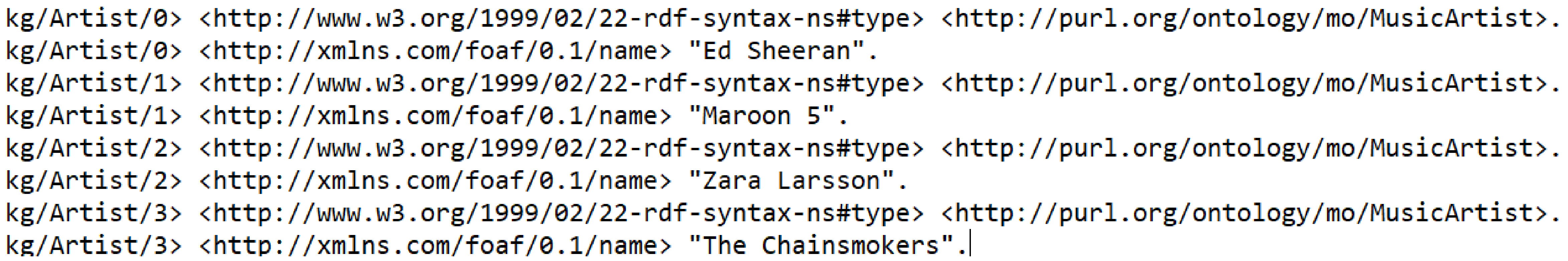
```
'Song/6f807x0ima9a1j3VPbc7VN> <http://purl.org/ontology/mo/duration> "194754"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#popularity> "66"^^<http://www.w3.org/2001/XMLSchema#integer>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#danceability> "0.748"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#energy> "0.916"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#key> "6"^^<http://www.w3.org/2001/XMLSchema#integer>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#loudness> "-2.634"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#mode> "1"^^<http://www.w3.org/2001/XMLSchema#integer>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#speechiness> "0.0583"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#acousticness> "0.102"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#instrumentalness> "0"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#liveness> "0.0653"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#valence> "0.518"^^<http://www.w3.org/2001/XMLSchema#double>.
'Song/6f807x0ima9a1j3VPbc7VN> <http://purl.org/ontology/mo/bpm> "122.036"^^<http://www.w3.org/2001/XMLSchema#double>.
```

Results: entity links



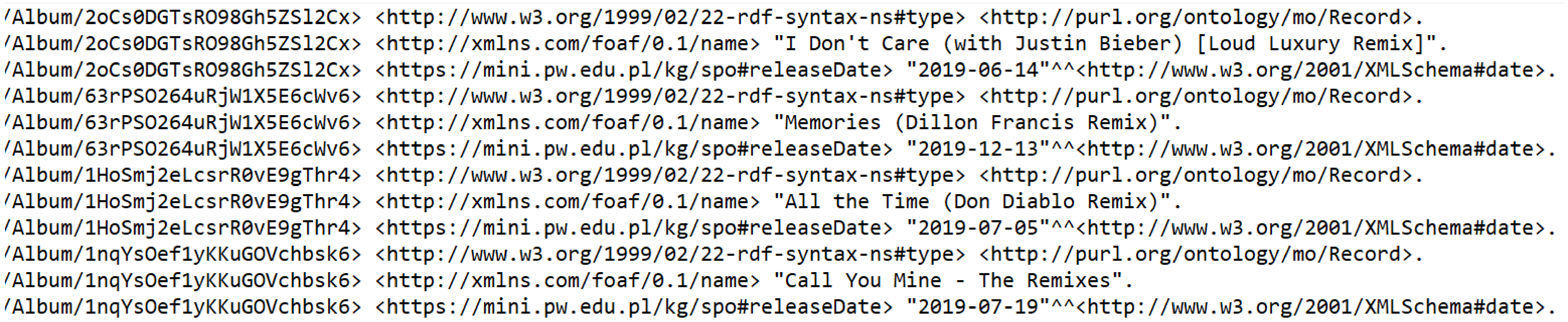
```
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#artist> <https://mini.pw.edu.pl/kg/Artist/0>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#album> <https://mini.pw.edu.pl/kg/Album/2oCs0DGTsRO98Gh5ZSl2Cx>.
'Song/6f807x0ima9a1j3VPbc7VN> <https://mini.pw.edu.pl/kg/spo#playlist> <https://mini.pw.edu.pl/kg/Playlist/37i9dQZF1DXcZDD7cfEKhW>.
```

Results: artist



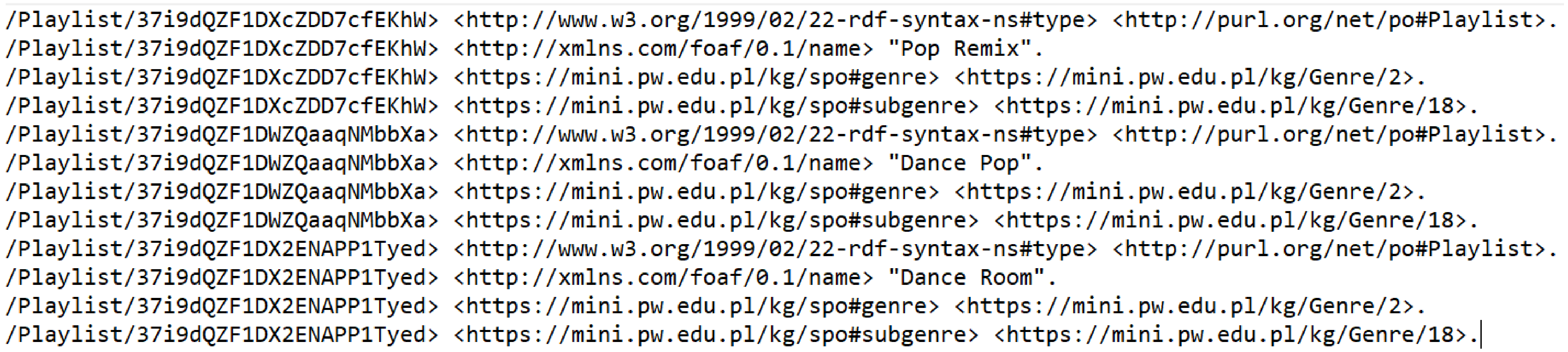
```
kg/Artist/0> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/MusicArtist>.
kg/Artist/0> <http://xmlns.com/foaf/0.1/name> "Ed Sheeran".
kg/Artist/1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/MusicArtist>.
kg/Artist/1> <http://xmlns.com/foaf/0.1/name> "Maroon 5".
kg/Artist/2> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/MusicArtist>.
kg/Artist/2> <http://xmlns.com/foaf/0.1/name> "Zara Larsson".
kg/Artist/3> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/MusicArtist>.
kg/Artist/3> <http://xmlns.com/foaf/0.1/name> "The Chainsmokers".
```

Results: album



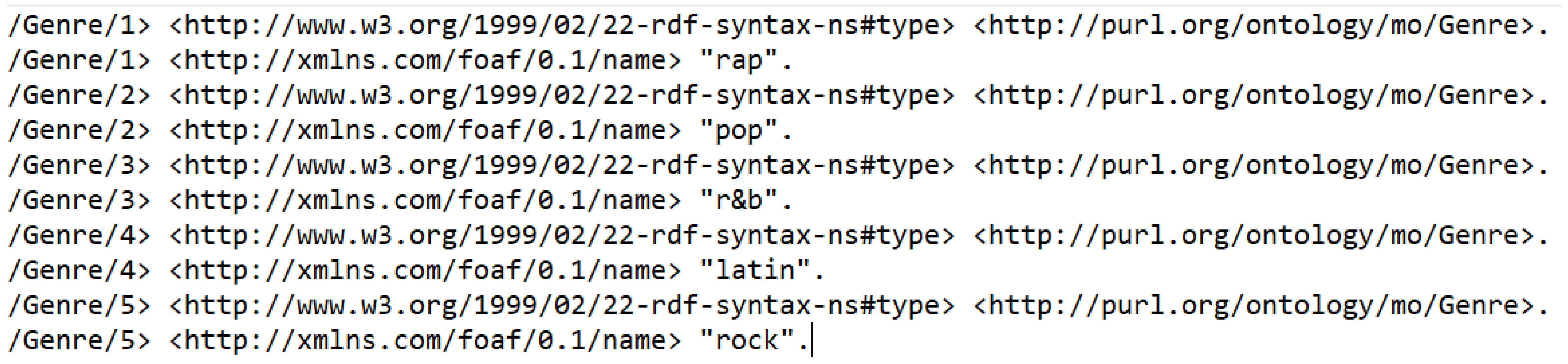
```
/Album/2oCs0DGTsRO98Gh5ZSl2Cx> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Record>.
/Album/2oCs0DGTsRO98Gh5ZSl2Cx> <http://xmlns.com/foaf/0.1/name> "I Don't Care (with Justin Bieber) [Loud Luxury Remix]".
/Album/2oCs0DGTsRO98Gh5ZSl2Cx> <https://mini.pw.edu.pl/kg/spo#releaseDate> "2019-06-14"^^<http://www.w3.org/2001/XMLSchema#date>.
/Album/63rPSO264uRjW1X5E6cWv6> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Record>.
/Album/63rPSO264uRjW1X5E6cWv6> <http://xmlns.com/foaf/0.1/name> "Memories (Dillon Francis Remix)".
/Album/63rPSO264uRjW1X5E6cWv6> <https://mini.pw.edu.pl/kg/spo#releaseDate> "2019-12-13"^^<http://www.w3.org/2001/XMLSchema#date>.
/Album/1HoSmj2eLcsrR0vE9gThr4> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Record>.
/Album/1HoSmj2eLcsrR0vE9gThr4> <http://xmlns.com/foaf/0.1/name> "All the Time (Don Diablo Remix)".
/Album/1HoSmj2eLcsrR0vE9gThr4> <https://mini.pw.edu.pl/kg/spo#releaseDate> "2019-07-05"^^<http://www.w3.org/2001/XMLSchema#date>.
/Album/1nqYsOef1yKKuGOVchbsk6> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Record>.
/Album/1nqYsOef1yKKuGOVchbsk6> <http://xmlns.com/foaf/0.1/name> "Call You Mine - The Remixes".
/Album/1nqYsOef1yKKuGOVchbsk6> <https://mini.pw.edu.pl/kg/spo#releaseDate> "2019-07-19"^^<http://www.w3.org/2001/XMLSchema#date>.
```

Results: playlist



```
/Playlist/37i9dQZF1DXcZDD7cfEKhW> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/net/po#Playlist>.
/Playlist/37i9dQZF1DXcZDD7cfEKhW> <http://xmlns.com/foaf/0.1/name> "Pop Remix".
/Playlist/37i9dQZF1DXcZDD7cfEKhW> <https://mini.pw.edu.pl/kg/spo#genre> <https://mini.pw.edu.pl/kg/Genre/2>.
/Playlist/37i9dQZF1DXcZDD7cfEKhW> <https://mini.pw.edu.pl/kg/spo#subgenre> <https://mini.pw.edu.pl/kg/Genre/18>.
/Playlist/37i9dQZF1DWZQaaqNMbbXa> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/net/po#Playlist>.
/Playlist/37i9dQZF1DWZQaaqNMbbXa> <http://xmlns.com/foaf/0.1/name> "Dance Pop".
/Playlist/37i9dQZF1DWZQaaqNMbbXa> <https://mini.pw.edu.pl/kg/spo#genre> <https://mini.pw.edu.pl/kg/Genre/2>.
/Playlist/37i9dQZF1DWZQaaqNMbbXa> <https://mini.pw.edu.pl/kg/spo#subgenre> <https://mini.pw.edu.pl/kg/Genre/18>.
/Playlist/37i9dQZF1DX2ENAPP1Tyed> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/net/po#Playlist>.
/Playlist/37i9dQZF1DX2ENAPP1Tyed> <http://xmlns.com/foaf/0.1/name> "Dance Room".
/Playlist/37i9dQZF1DX2ENAPP1Tyed> <https://mini.pw.edu.pl/kg/spo#genre> <https://mini.pw.edu.pl/kg/Genre/2>.
/Playlist/37i9dQZF1DX2ENAPP1Tyed> <https://mini.pw.edu.pl/kg/spo#subgenre> <https://mini.pw.edu.pl/kg/Genre/18>.
```

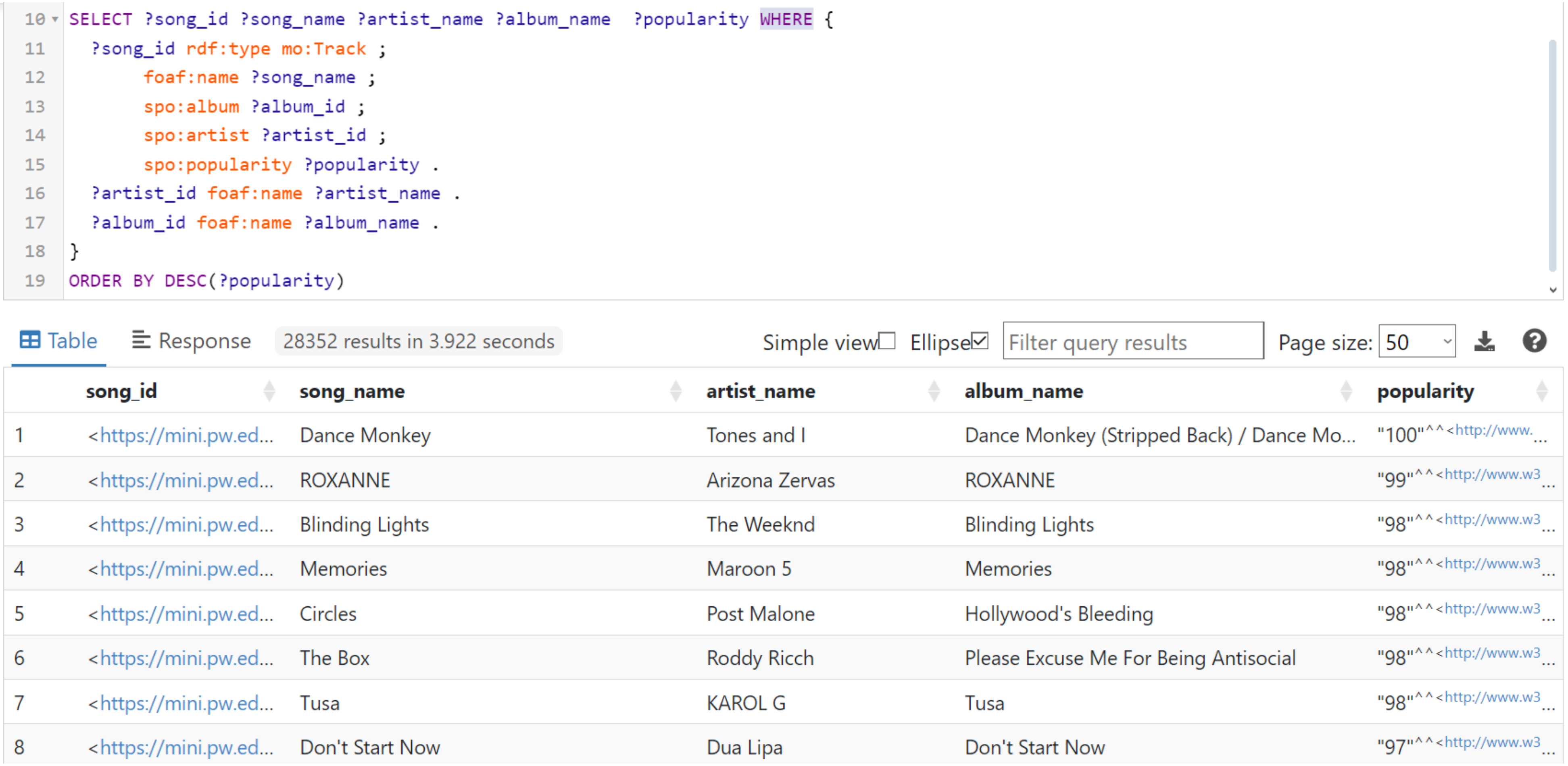
Results: genres



```
/Genre/1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Genre>.
/Genre/1> <http://xmlns.com/foaf/0.1/name> "rap".
/Genre/2> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Genre>.
/Genre/2> <http://xmlns.com/foaf/0.1/name> "pop".
/Genre/3> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Genre>.
/Genre/3> <http://xmlns.com/foaf/0.1/name> "r&b".
/Genre/4> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Genre>.
/Genre/4> <http://xmlns.com/foaf/0.1/name> "latin".
/Genre/5> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://purl.org/ontology/mo/Genre>.
/Genre/5> <http://xmlns.com/foaf/0.1/name> "rock".
```

Query 1

- Select songs and sort them in a descending order based on the popularity



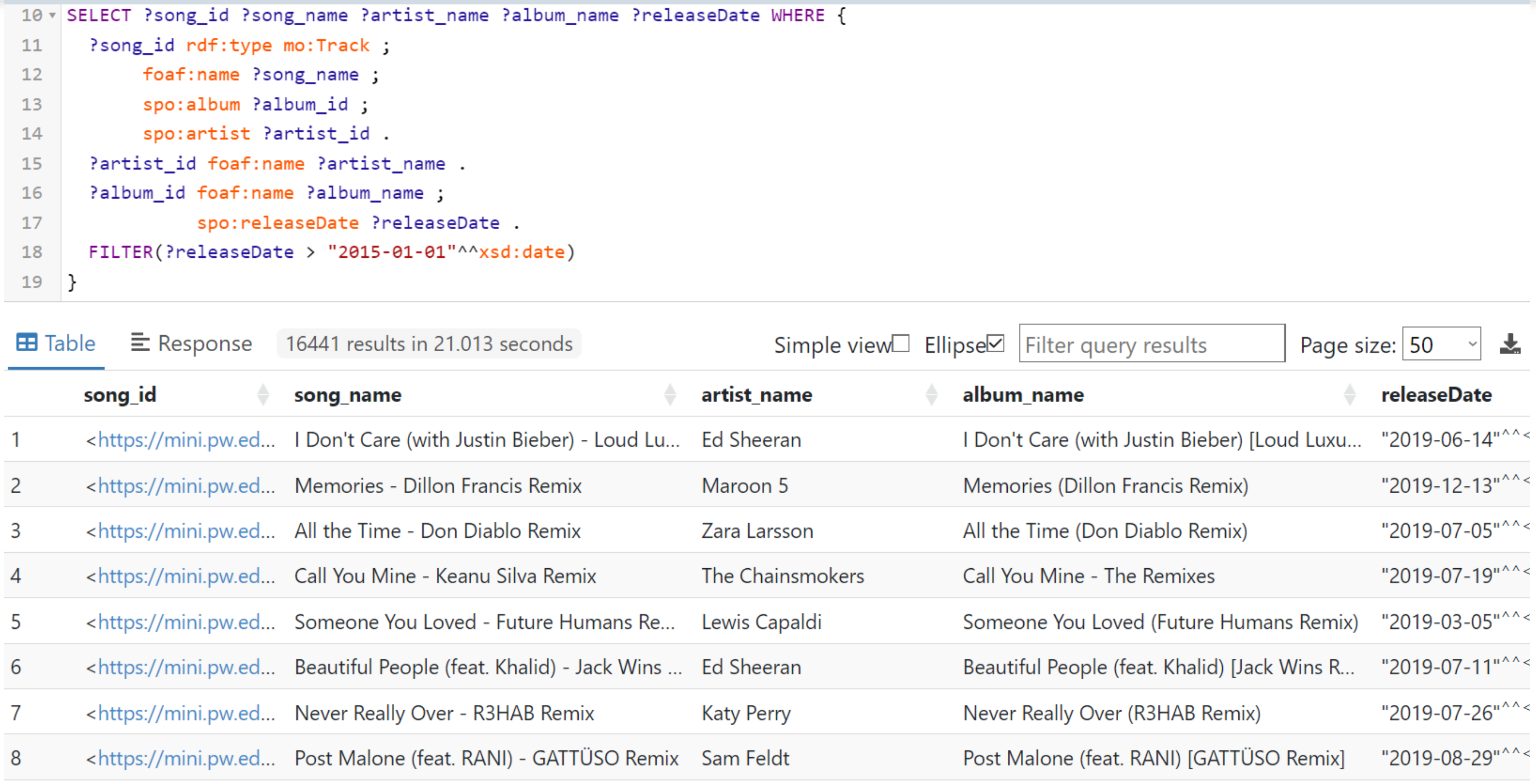
```
SELECT ?song_id ?song_name ?artist_name ?album_name  ?popularity WHERE {
  ?song_id rdf:type mo:Track ;
       foaf:name ?song_name ;
       spo:album ?album_id ;
       spo:artist ?artist_id ;
       spo:popularity ?popularity .
  ?artist_id foaf:name ?artist_name .
  ?album_id foaf:name ?album_name .
}
ORDER BY DESC(?popularity)
```

28352 results in 3.922 seconds

	song_id	song_name	artist_name	album_name	popularity
1	<https://mini.pw.ed...	Dance Monkey	Tones and I	Dance Monkey (Stripped Back) / Dance Mo...	"100"^^<http://www....
2	<https://mini.pw.ed...	ROXANNE	Arizona Zervas	ROXANNE	"99"^^<http://www.w3...
3	<https://mini.pw.ed...	Blinding Lights	The Weeknd	Blinding Lights	"98"^^<http://www.w3...
4	<https://mini.pw.ed...	Memories	Maroon 5	Memories	"98"^^<http://www.w3...
5	<https://mini.pw.ed...	Circles	Post Malone	Hollywood's Bleeding	"98"^^<http://www.w3...
6	<https://mini.pw.ed...	The Box	Roddy Ricch	Please Excuse Me For Being Antisocial	"98"^^<http://www.w3...
7	<https://mini.pw.ed...	Tusa	KAROL G	Tusa	"98"^^<http://www.w3...
8	<https://mini.pw.ed...	Don't Start Now	Dua Lipa	Don't Start Now	"97"^^<http://www.w3...

Query 2

- Select songs released after 2015-01-01



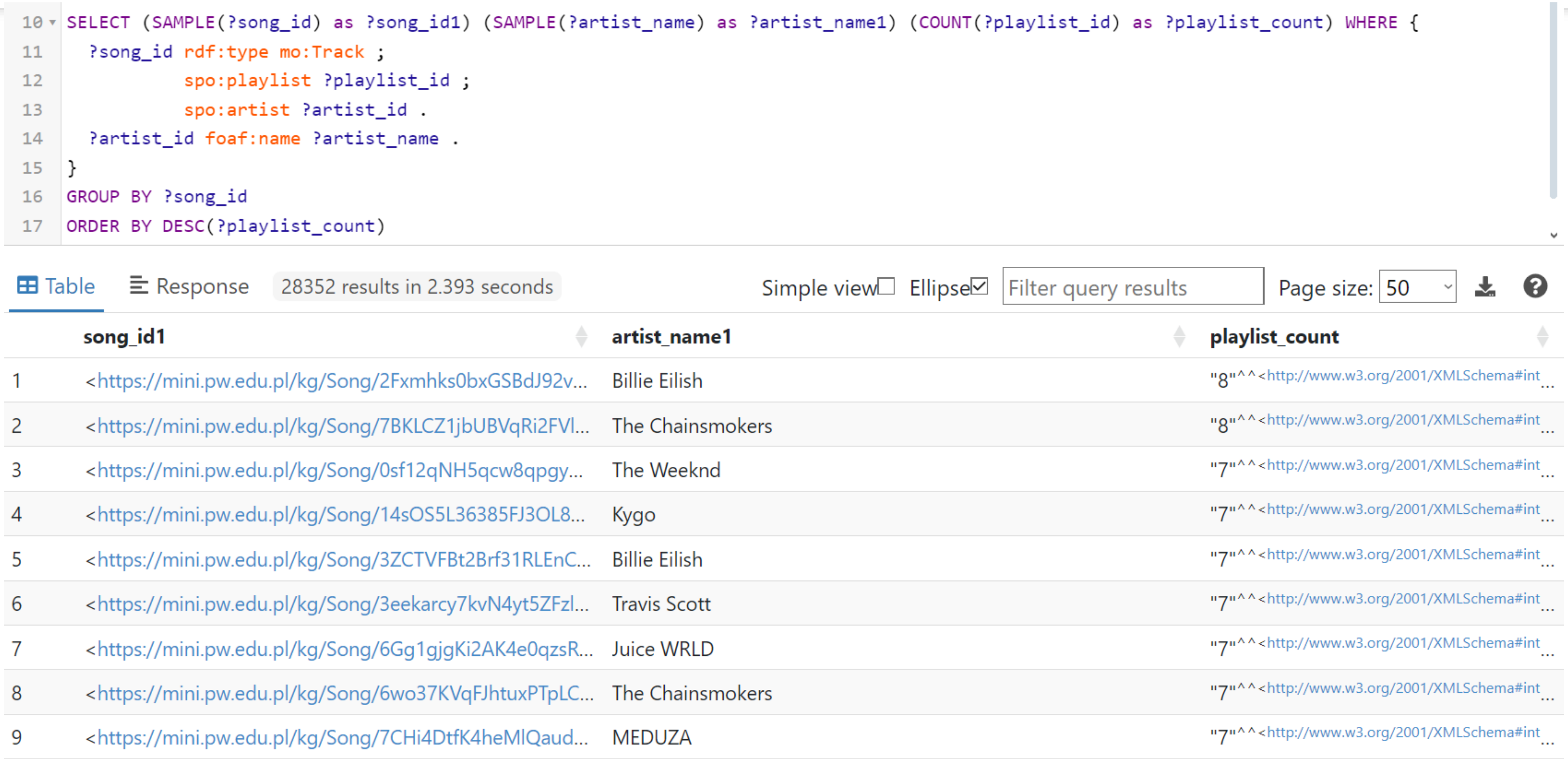
```
SELECT ?song_id ?song_name ?artist_name ?album_name ?releaseDate WHERE {
  ?song_id rdf:type mo:Track ;
       foaf:name ?song_name ;
       spo:album ?album_id ;
       spo:artist ?artist_id .
  ?artist_id foaf:name ?artist_name .
  ?album_id foaf:name ?album_name ;
            spo:releaseDate ?releaseDate .
  FILTER(?releaseDate > "2015-01-01"^^xsd:date)
}
```

Table | Response | 16441 results in 21.013 seconds | Simple view ☐ | Ellipse ☑ | Filter query results | Page size: 50

	song_id	song_name	artist_name	album_name	releaseDate
1	<https://mini.pw.ed...	I Don't Care (with Justin Bieber) - Loud Lu...	Ed Sheeran	I Don't Care (with Justin Bieber) [Loud Luxu...	"2019-06-14"^^<
2	<https://mini.pw.ed...	Memories - Dillon Francis Remix	Maroon 5	Memories (Dillon Francis Remix)	"2019-12-13"^^<
3	<https://mini.pw.ed...	All the Time - Don Diablo Remix	Zara Larsson	All the Time (Don Diablo Remix)	"2019-07-05"^^<
4	<https://mini.pw.ed...	Call You Mine - Keanu Silva Remix	The Chainsmokers	Call You Mine - The Remixes	"2019-07-19"^^<
5	<https://mini.pw.ed...	Someone You Loved - Future Humans Re...	Lewis Capaldi	Someone You Loved (Future Humans Remix)	"2019-03-05"^^<
6	<https://mini.pw.ed...	Beautiful People (feat. Khalid) - Jack Wins ...	Ed Sheeran	Beautiful People (feat. Khalid) [Jack Wins R...	"2019-07-11"^^<
7	<https://mini.pw.ed...	Never Really Over - R3HAB Remix	Katy Perry	Never Really Over (R3HAB Remix)	"2019-07-26"^^<
8	<https://mini.pw.ed...	Post Malone (feat. RANI) - GATTÜSO Remix	Sam Feldt	Post Malone (feat. RANI) [GATTÜSO Remix]	"2019-08-29"^^<

Query 3

- Select songs and sort them based on the number of appearances in playlists



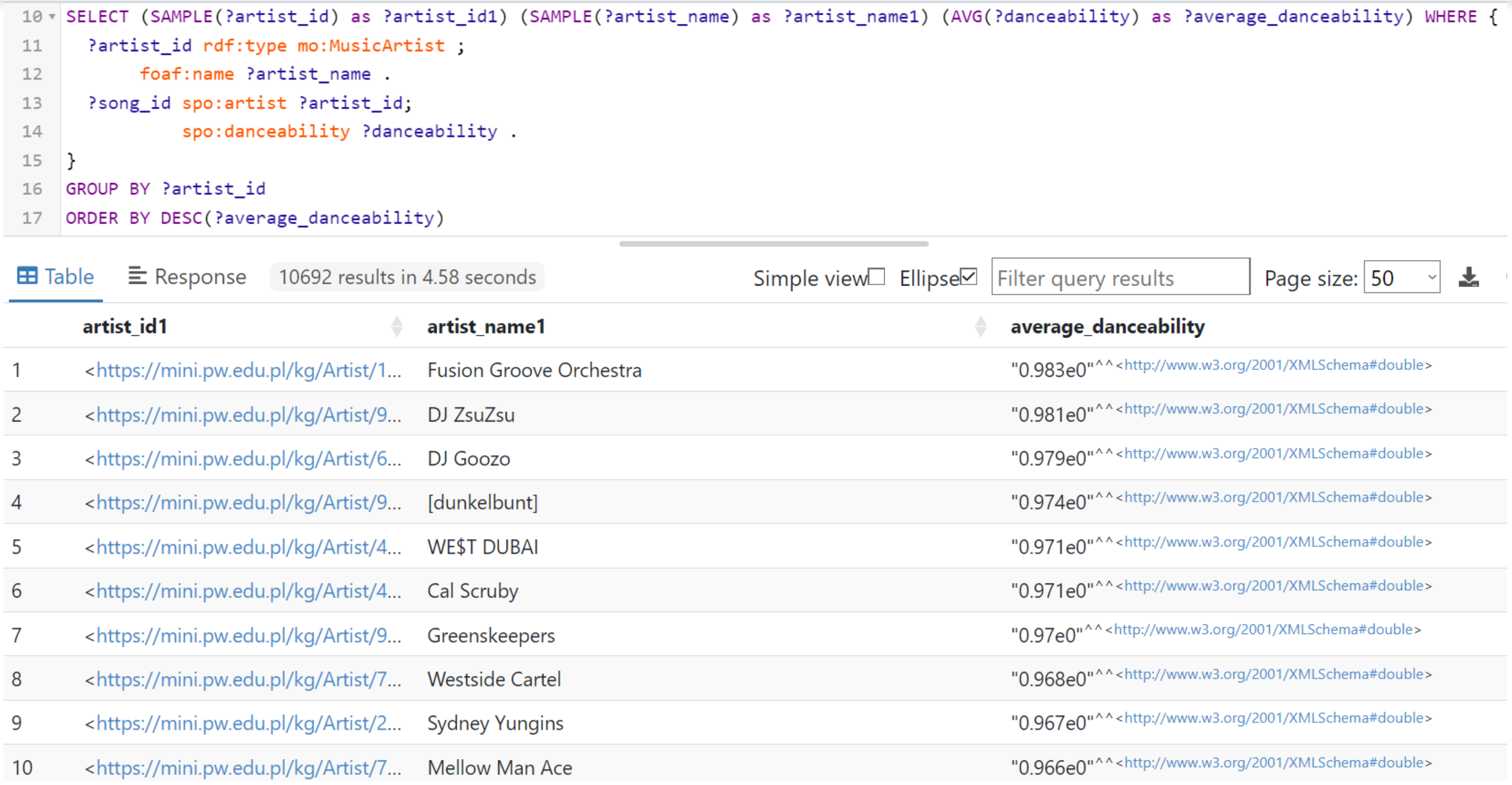
```
SELECT (SAMPLE(?song_id) as ?song_id1) (SAMPLE(?artist_name) as ?artist_name1) (COUNT(?playlist_id) as ?playlist_count) WHERE {
  ?song_id rdf:type mo:Track ;
           spo:playlist ?playlist_id ;
           spo:artist ?artist_id .
  ?artist_id foaf:name ?artist_name .
}
GROUP BY ?song_id
ORDER BY DESC(?playlist_count)
```

Table | Response | 28352 results in 2.393 seconds | Simple view ☐ | Ellipse ☑ | Filter query results | Page size: 50

	song_id1	artist_name1	playlist_count
1	<https://mini.pw.edu.pl/kg/Song/2Fxmhks0bxGSBdJ92v...	Billie Eilish	"8"^^<http://www.w3.org/2001/XMLSchema#int...
2	<https://mini.pw.edu.pl/kg/Song/7BKLCZ1jbUBVqRi2FVl...	The Chainsmokers	"8"^^<http://www.w3.org/2001/XMLSchema#int...
3	<https://mini.pw.edu.pl/kg/Song/0sf12qNH5qcw8qpgy...	The Weeknd	"7"^^<http://www.w3.org/2001/XMLSchema#int...
4	<https://mini.pw.edu.pl/kg/Song/14sOS5L36385FJ3OL8...	Kygo	"7"^^<http://www.w3.org/2001/XMLSchema#int...
5	<https://mini.pw.edu.pl/kg/Song/3ZCTVFBt2Brf31RLEnC...	Billie Eilish	"7"^^<http://www.w3.org/2001/XMLSchema#int...
6	<https://mini.pw.edu.pl/kg/Song/3eekarcy7kvN4yt5ZFzl...	Travis Scott	"7"^^<http://www.w3.org/2001/XMLSchema#int...
7	<https://mini.pw.edu.pl/kg/Song/6Gg1gjgKi2AK4e0qzsR...	Juice WRLD	"7"^^<http://www.w3.org/2001/XMLSchema#int...
8	<https://mini.pw.edu.pl/kg/Song/6wo37KVqFJhtuxPTpLC...	The Chainsmokers	"7"^^<http://www.w3.org/2001/XMLSchema#int...
9	<https://mini.pw.edu.pl/kg/Song/7CHi4DtfK4heMlQaud...	MEDUZA	"7"^^<http://www.w3.org/2001/XMLSchema#int...

Query 4

- Select artists and sort them in a descending order based on the average danceability of their songs



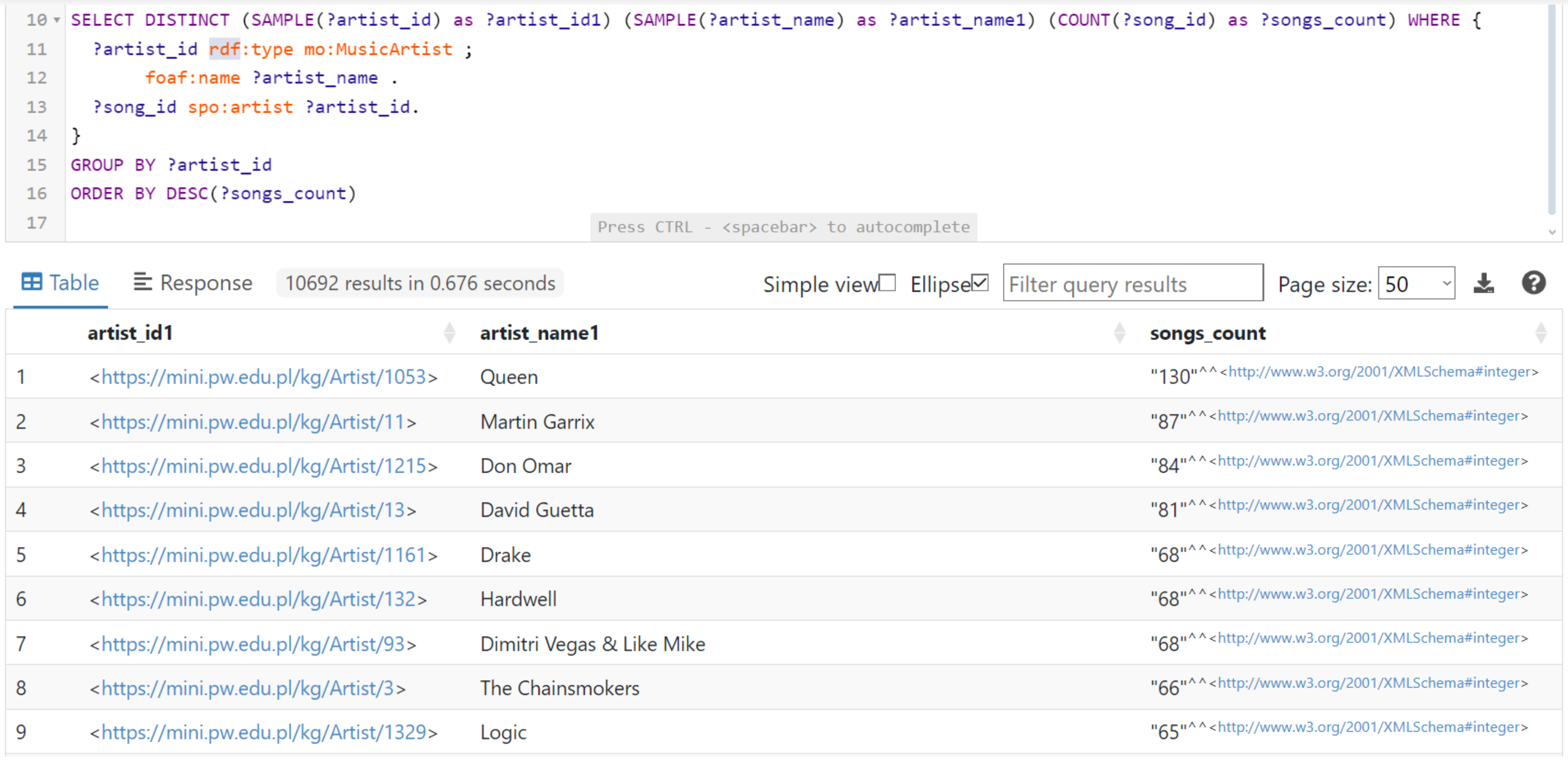
```
SELECT (SAMPLE(?artist_id) as ?artist_id1) (SAMPLE(?artist_name) as ?artist_name1) (AVG(?danceability) as ?average_danceability) WHERE {
  ?artist_id rdf:type mo:MusicArtist ;
       foaf:name ?artist_name .
  ?song_id spo:artist ?artist_id;
           spo:danceability ?danceability .
}
GROUP BY ?artist_id
ORDER BY DESC(?average_danceability)
```

Table Response 10692 results in 4.58 seconds Simple view ☐ Ellipse ☑ Filter query results Page size: 50

	artist_id1	artist_name1	average_danceability
1	<https://mini.pw.edu.pl/kg/Artist/1...	Fusion Groove Orchestra	"0.983e0"^^<http://www.w3.org/2001/XMLSchema#double>
2	<https://mini.pw.edu.pl/kg/Artist/9...	DJ ZsuZsu	"0.981e0"^^<http://www.w3.org/2001/XMLSchema#double>
3	<https://mini.pw.edu.pl/kg/Artist/6...	DJ Goozo	"0.979e0"^^<http://www.w3.org/2001/XMLSchema#double>
4	<https://mini.pw.edu.pl/kg/Artist/9...	[dunkelbunt]	"0.974e0"^^<http://www.w3.org/2001/XMLSchema#double>
5	<https://mini.pw.edu.pl/kg/Artist/4...	WE$T DUBAI	"0.971e0"^^<http://www.w3.org/2001/XMLSchema#double>
6	<https://mini.pw.edu.pl/kg/Artist/4...	Cal Scruby	"0.971e0"^^<http://www.w3.org/2001/XMLSchema#double>
7	<https://mini.pw.edu.pl/kg/Artist/9...	Greenskeepers	"0.97e0"^^<http://www.w3.org/2001/XMLSchema#double>
8	<https://mini.pw.edu.pl/kg/Artist/7...	Westside Cartel	"0.968e0"^^<http://www.w3.org/2001/XMLSchema#double>
9	<https://mini.pw.edu.pl/kg/Artist/2...	Sydney Yungins	"0.967e0"^^<http://www.w3.org/2001/XMLSchema#double>
10	<https://mini.pw.edu.pl/kg/Artist/7...	Mellow Man Ace	"0.966e0"^^<http://www.w3.org/2001/XMLSchema#double>

Query 5

- Select artists and sort them in a descending order based on the number of released songs



```
SELECT DISTINCT (SAMPLE(?artist_id) as ?artist_id1) (SAMPLE(?artist_name) as ?artist_name1) (COUNT(?song_id) as ?songs_count) WHERE {
  ?artist_id rdf:type mo:MusicArtist ;
       foaf:name ?artist_name .
  ?song_id spo:artist ?artist_id.
}
GROUP BY ?artist_id
ORDER BY DESC(?songs_count)
```

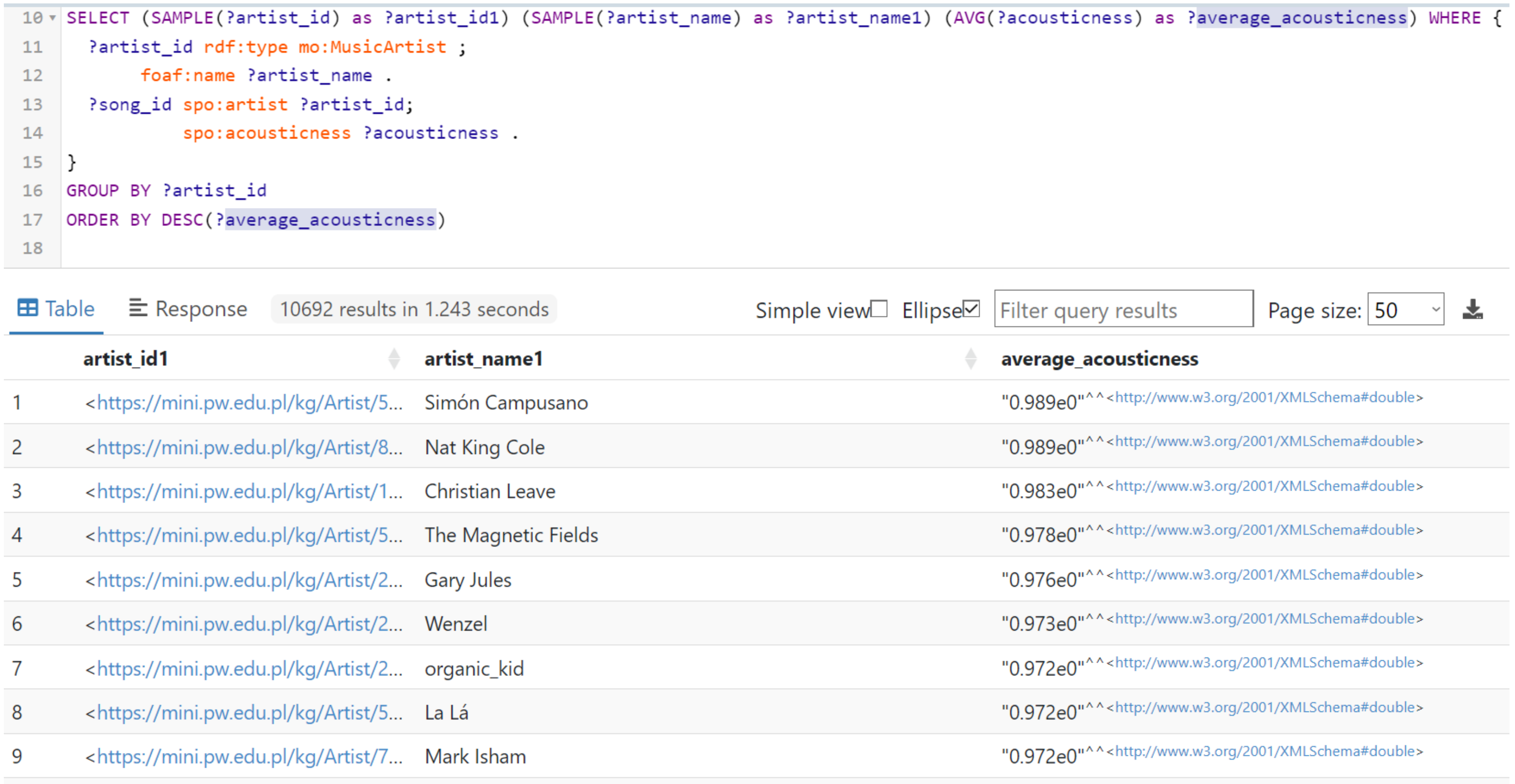
Press CTRL - <spacebar> to autocomplete

Table | Response | 10692 results in 0.676 seconds | Simple view ☐ | Ellipse ☑ | Filter query results | Page size: 50

	artist_id1	artist_name1	songs_count
1	<https://mini.pw.edu.pl/kg/Artist/1053>	Queen	"130"^^<http://www.w3.org/2001/XMLSchema#integer>
2	<https://mini.pw.edu.pl/kg/Artist/11>	Martin Garrix	"87"^^<http://www.w3.org/2001/XMLSchema#integer>
3	<https://mini.pw.edu.pl/kg/Artist/1215>	Don Omar	"84"^^<http://www.w3.org/2001/XMLSchema#integer>
4	<https://mini.pw.edu.pl/kg/Artist/13>	David Guetta	"81"^^<http://www.w3.org/2001/XMLSchema#integer>
5	<https://mini.pw.edu.pl/kg/Artist/1161>	Drake	"68"^^<http://www.w3.org/2001/XMLSchema#integer>
6	<https://mini.pw.edu.pl/kg/Artist/132>	Hardwell	"68"^^<http://www.w3.org/2001/XMLSchema#integer>
7	<https://mini.pw.edu.pl/kg/Artist/93>	Dimitri Vegas & Like Mike	"68"^^<http://www.w3.org/2001/XMLSchema#integer>
8	<https://mini.pw.edu.pl/kg/Artist/3>	The Chainsmokers	"66"^^<http://www.w3.org/2001/XMLSchema#integer>
9	<https://mini.pw.edu.pl/kg/Artist/1329>	Logic	"65"^^<http://www.w3.org/2001/XMLSchema#integer>

Query 6

- Select artists and sort them in a descending order based on the acousticness of their songs



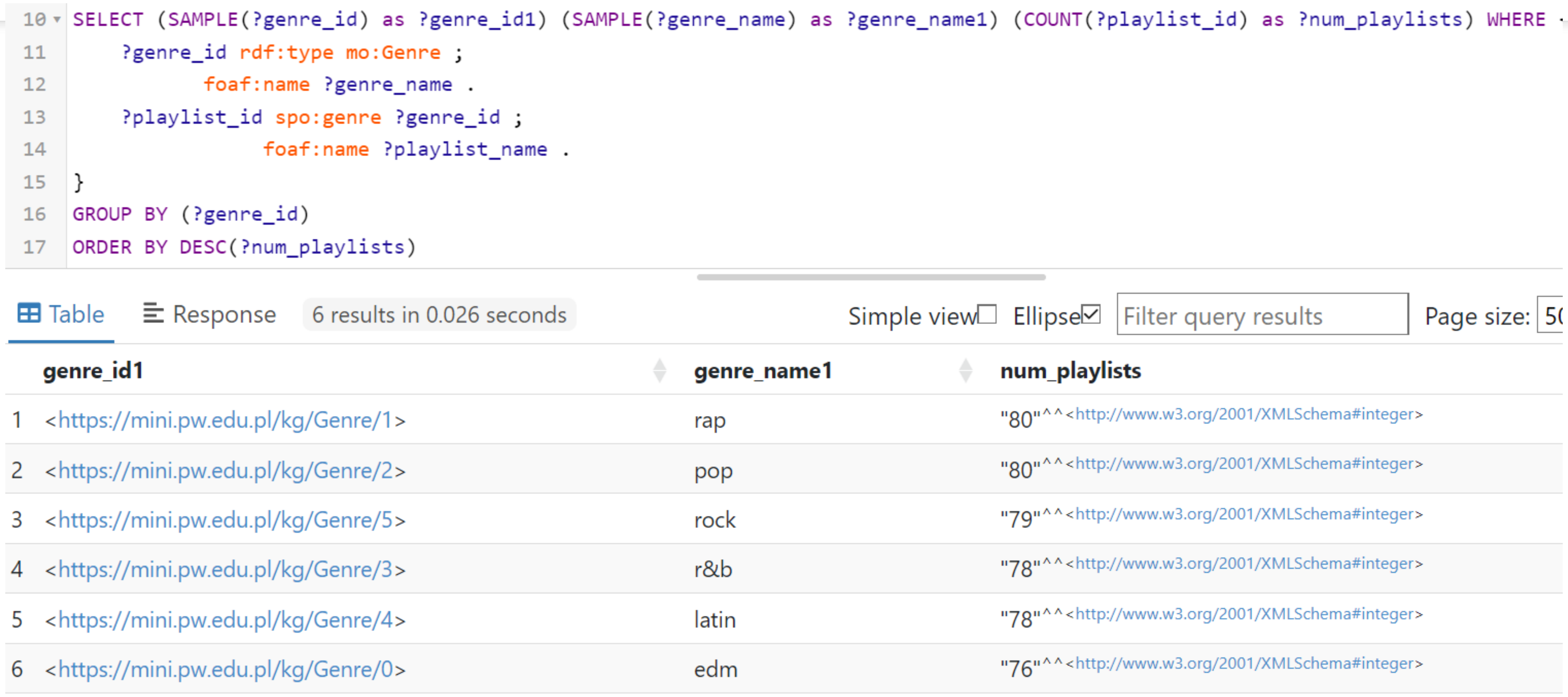
```
SELECT (SAMPLE(?artist_id) as ?artist_id1) (SAMPLE(?artist_name) as ?artist_name1) (AVG(?acousticness) as ?average_acousticness) WHERE {
  ?artist_id rdf:type mo:MusicArtist ;
       foaf:name ?artist_name .
  ?song_id spo:artist ?artist_id;
           spo:acousticness ?acousticness .
}
GROUP BY ?artist_id
ORDER BY DESC(?average_acousticness)

```

Table   Response   10692 results in 1.243 seconds   Simple view ☐ Ellipse ☑ Filter query results Page size: 50

	artist_id1	artist_name1	average_acousticness
1	<https://mini.pw.edu.pl/kg/Artist/5...	Simón Campusano	"0.989e0"^^<http://www.w3.org/2001/XMLSchema#double>
2	<https://mini.pw.edu.pl/kg/Artist/8...	Nat King Cole	"0.989e0"^^<http://www.w3.org/2001/XMLSchema#double>
3	<https://mini.pw.edu.pl/kg/Artist/1...	Christian Leave	"0.983e0"^^<http://www.w3.org/2001/XMLSchema#double>
4	<https://mini.pw.edu.pl/kg/Artist/5...	The Magnetic Fields	"0.978e0"^^<http://www.w3.org/2001/XMLSchema#double>
5	<https://mini.pw.edu.pl/kg/Artist/2...	Gary Jules	"0.976e0"^^<http://www.w3.org/2001/XMLSchema#double>
6	<https://mini.pw.edu.pl/kg/Artist/2...	Wenzel	"0.973e0"^^<http://www.w3.org/2001/XMLSchema#double>
7	<https://mini.pw.edu.pl/kg/Artist/2...	organic_kid	"0.972e0"^^<http://www.w3.org/2001/XMLSchema#double>
8	<https://mini.pw.edu.pl/kg/Artist/5...	La Lá	"0.972e0"^^<http://www.w3.org/2001/XMLSchema#double>
9	<https://mini.pw.edu.pl/kg/Artist/7...	Mark Isham	"0.972e0"^^<http://www.w3.org/2001/XMLSchema#double>

Query 7

- Sort genres in descending order based on the number of their playlists

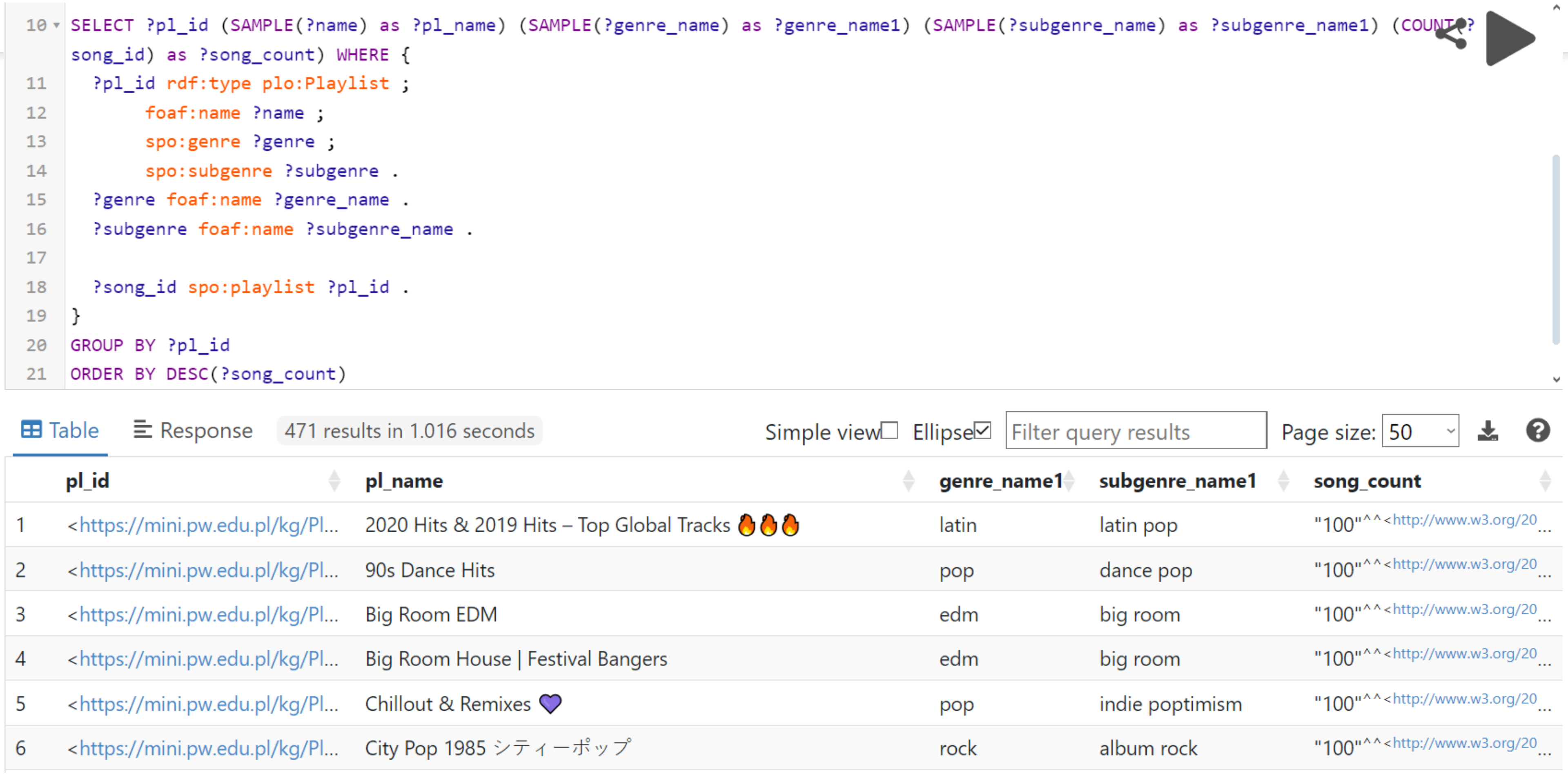


```
SELECT (SAMPLE(?genre_id) as ?genre_id1) (SAMPLE(?genre_name) as ?genre_name1) (COUNT(?playlist_id) as ?num_playlists) WHERE
    ?genre_id rdf:type mo:Genre ;
           foaf:name ?genre_name .
    ?playlist_id spo:genre ?genre_id ;
                foaf:name ?playlist_name .
}
GROUP BY (?genre_id)
ORDER BY DESC(?num_playlists)
```

	genre_id1	genre_name1	num_playlists
1	<https://mini.pw.edu.pl/kg/Genre/1>	rap	"80"^^<http://www.w3.org/2001/XMLSchema#integer>
2	<https://mini.pw.edu.pl/kg/Genre/2>	pop	"80"^^<http://www.w3.org/2001/XMLSchema#integer>
3	<https://mini.pw.edu.pl/kg/Genre/5>	rock	"79"^^<http://www.w3.org/2001/XMLSchema#integer>
4	<https://mini.pw.edu.pl/kg/Genre/3>	r&b	"78"^^<http://www.w3.org/2001/XMLSchema#integer>
5	<https://mini.pw.edu.pl/kg/Genre/4>	latin	"78"^^<http://www.w3.org/2001/XMLSchema#integer>
6	<https://mini.pw.edu.pl/kg/Genre/0>	edm	"76"^^<http://www.w3.org/2001/XMLSchema#integer>

Query 8

- Select playlists and sort them in a descending order based on the songs count



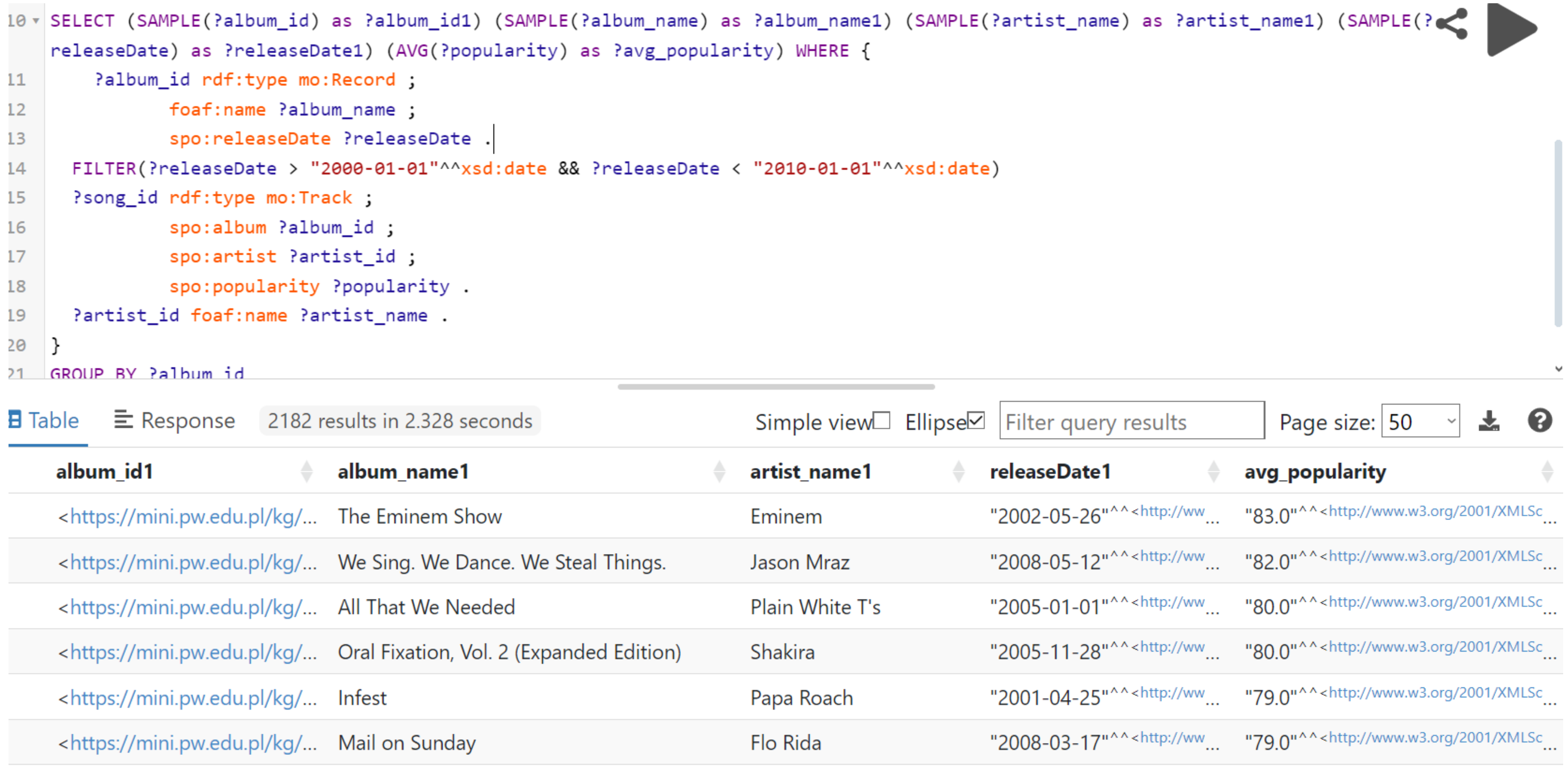
```
SELECT ?pl_id (SAMPLE(?name) as ?pl_name) (SAMPLE(?genre_name) as ?genre_name1) (SAMPLE(?subgenre_name) as ?subgenre_name1) (COUNT(?
song_id) as ?song_count) WHERE {
  ?pl_id rdf:type plo:Playlist ;
       foaf:name ?name ;
       spo:genre ?genre ;
       spo:subgenre ?subgenre .
  ?genre foaf:name ?genre_name .
  ?subgenre foaf:name ?subgenre_name .

  ?song_id spo:playlist ?pl_id .
}
GROUP BY ?pl_id
ORDER BY DESC(?song_count)
```

Table | Response | 471 results in 1.016 seconds | Simple view ☐ | Ellipse ☑ | Filter query results | Page size: 50

	pl_id	pl_name	genre_name1	subgenre_name1	song_count
1	<https://mini.pw.edu.pl/kg/Pl...	2020 Hits & 2019 Hits – Top Global Tracks 🔥🔥🔥	latin	latin pop	"100"^^<http://www.w3.org/20...
2	<https://mini.pw.edu.pl/kg/Pl...	90s Dance Hits	pop	dance pop	"100"^^<http://www.w3.org/20...
3	<https://mini.pw.edu.pl/kg/Pl...	Big Room EDM	edm	big room	"100"^^<http://www.w3.org/20...
4	<https://mini.pw.edu.pl/kg/Pl...	Big Room House | Festival Bangers	edm	big room	"100"^^<http://www.w3.org/20...
5	<https://mini.pw.edu.pl/kg/Pl...	Chillout & Remixes 💜	pop	indie poptimism	"100"^^<http://www.w3.org/20...
6	<https://mini.pw.edu.pl/kg/Pl...	City Pop 1985 シティーポップ	rock	album rock	"100"^^<http://www.w3.org/20...

Query 9

- Select albums that were released in 2000-2010 and sort them based on the average popularity of their songs



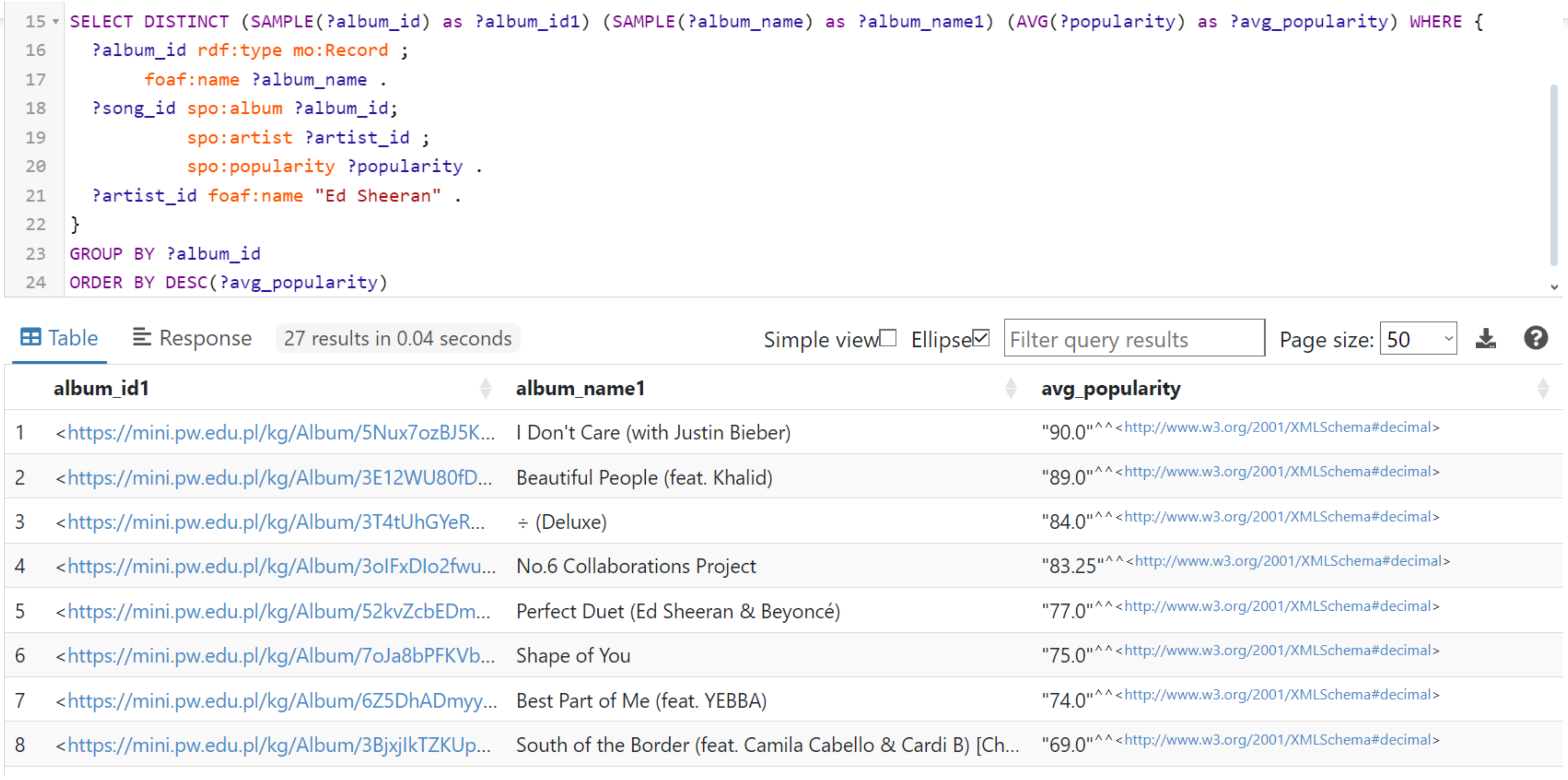
```
SELECT (SAMPLE(?album_id) as ?album_id1) (SAMPLE(?album_name) as ?album_name1) (SAMPLE(?artist_name) as ?artist_name1) (SAMPLE(?
releaseDate) as ?releaseDate1) (AVG(?popularity) as ?avg_popularity) WHERE {
    ?album_id rdf:type mo:Record ;
           foaf:name ?album_name ;
           spo:releaseDate ?releaseDate .
  FILTER(?releaseDate > "2000-01-01"^^xsd:date && ?releaseDate < "2010-01-01"^^xsd:date)
  ?song_id rdf:type mo:Track ;
           spo:album ?album_id ;
           spo:artist ?artist_id ;
           spo:popularity ?popularity .
  ?artist_id foaf:name ?artist_name .
}
GROUP BY ?album_id
```

Table   Response   2182 results in 2.328 seconds   Simple view   Ellipse ✓   Filter query results   Page size: 50

album_id1	album_name1	artist_name1	releaseDate1	avg_popularity
<https://mini.pw.edu.pl/kg/...	The Eminem Show	Eminem	"2002-05-26"^^<http://ww...	"83.0"^^<http://www.w3.org/2001/XMLSc...
<https://mini.pw.edu.pl/kg/...	We Sing. We Dance. We Steal Things.	Jason Mraz	"2008-05-12"^^<http://ww...	"82.0"^^<http://www.w3.org/2001/XMLSc...
<https://mini.pw.edu.pl/kg/...	All That We Needed	Plain White T's	"2005-01-01"^^<http://ww...	"80.0"^^<http://www.w3.org/2001/XMLSc...
<https://mini.pw.edu.pl/kg/...	Oral Fixation, Vol. 2 (Expanded Edition)	Shakira	"2005-11-28"^^<http://ww...	"80.0"^^<http://www.w3.org/2001/XMLSc...
<https://mini.pw.edu.pl/kg/...	Infest	Papa Roach	"2001-04-25"^^<http://ww...	"79.0"^^<http://www.w3.org/2001/XMLSc...
<https://mini.pw.edu.pl/kg/...	Mail on Sunday	Flo Rida	"2008-03-17"^^<http://ww...	"79.0"^^<http://www.w3.org/2001/XMLSc...

Query 10

- Select the most popular albums of ‘Ed Sheeran’



```
SELECT DISTINCT (SAMPLE(?album_id) as ?album_id1) (SAMPLE(?album_name) as ?album_name1) (AVG(?popularity) as ?avg_popularity) WHERE {
  ?album_id rdf:type mo:Record ;
       foaf:name ?album_name .
  ?song_id spo:album ?album_id;
           spo:artist ?artist_id ;
           spo:popularity ?popularity .
  ?artist_id foaf:name "Ed Sheeran" .
}
GROUP BY ?album_id
ORDER BY DESC(?avg_popularity)
```

Table   Response   27 results in 0.04 seconds   Simple view☐ Ellipse☑ Filter query results   Page size: 50

	album_id1	album_name1	avg_popularity
1	<https://mini.pw.edu.pl/kg/Album/5Nux7ozBJ5K...	I Don't Care (with Justin Bieber)	"90.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
2	<https://mini.pw.edu.pl/kg/Album/3E12WU80fD...	Beautiful People (feat. Khalid)	"89.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
3	<https://mini.pw.edu.pl/kg/Album/3T4tUhGYeR...	÷ (Deluxe)	"84.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
4	<https://mini.pw.edu.pl/kg/Album/3oIFxDIo2fwu...	No.6 Collaborations Project	"83.25"^^<http://www.w3.org/2001/XMLSchema#decimal>
5	<https://mini.pw.edu.pl/kg/Album/52kvZcbEDm...	Perfect Duet (Ed Sheeran & Beyoncé)	"77.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
6	<https://mini.pw.edu.pl/kg/Album/7oJa8bPFKVb...	Shape of You	"75.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
7	<https://mini.pw.edu.pl/kg/Album/6Z5DhADmyy...	Best Part of Me (feat. YEBBA)	"74.0"^^<http://www.w3.org/2001/XMLSchema#decimal>
8	<https://mini.pw.edu.pl/kg/Album/3BjxjIkTZKUp...	South of the Border (feat. Camila Cabello & Cardi B) [Ch...	"69.0"^^<http://www.w3.org/2001/XMLSchema#decimal>

References

- Heyvaert, P., De Meester, B., & Dimou, A. (2018). Generating Linked Data with YARRRML. Retrieved from https://rml.io/yarrrml/tutorial/getting-started/
- Brunner, K. (2022, March). RDF, RML, YARRRML: A basic tutorial to create Linked Data from a relational database table. Katharina Brunner. Retrieved from https://katharinabrunner.de/2022/03/rdf-rml-yarrrml-kglab-morph-kgc/
- Torabi, N. (2023b, August 28). The inner workings of Spotify's AI-powered music recommendations: How Spotify Shapes your playlist. Medium. https://neemz.medium.com/the-inner-workings-of-spotifys-ai-powered-music-recommendations-how-spotify-shapes-your-playlist-a10a9148ee8d
- Pastukhov, D. (2022, February 9). How Spotify's algorithm works? A Complete Guide to spotify recommendation system [2022] | Music Tomorrow Blog. https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022

Thank you for attention