Project Report

Constructing an Ontology and Knowledge Graph of Our 22 Favorite English Songs

Data and Project

Our project aims at building an RDF knowledge graph centered around our 22 most favorite English songs. By collecting data from diverse sources such as Wikipedia pages and music information websites, we not only aim to demonstrate our proficiency in leveraging the knowledge acquired from seminars but also test our understanding of RDF and its application in organizing music-related information.

The data was selected based on our personal music preferences, which compelled us to manually collect it. Furthermore, the unexpected withdrawal of one of our partners just before the project's initiation resulted in the workload originally assigned to three individuals being shouldered by only two. Consequently, we had to restrict the dataset to 22 entries. However, despite its relatively small size, we have managed to ensure the integrity and comprehensiveness of the relevant information, resulting in a substantial turtle file with over 650 lines.

This is an introduction of the data and explanation of the reason why we choose the following properties for the data:

1. Song Name: Each song is represented as a distinct resource, allowing efficient navigation and retrieval of song-specific information.
2. Release Date: Accurate release date information enriches the knowledge graph, enabling historical analysis and exploration of musical trends.
3. Artist: Singer information is incorporated as resources, establishing connections between songs and the vocalists who performed them.
4. Writer: Songwriters are linked to the corresponding songs, acknowledging their creative contributions.
5. Genre: Each song is associated with its respective genre, providing insights into diverse musical styles and influences.
6. Producer: Producers are connected to songs, recognizing their influence and contributions to the production process.
7. Record Label: Songs are linked to record labels, highlighting industry support and the role of labels in promotion and distribution.

Ontology

In addition to utilizing shared vocabularies like "schema", we created our own ontology named "myOnto" specifically tailored for our 22 favorite songs. Apart from incorporating external classes, properties, and resources, we also introduced our own custom classes, properties, and resources within "myOnto".

Within our ontology, we defined three self-created classes, applying restrictions on cardinality and data types for two of them. For instance, the class "myOnto:Person" includes the restriction that "Each person should have no more than 1 WikipageID, and the WikipageID's data type should be integer."

Moreover, we established two properties in "myOnto" namely, "myOnto:hasName" and "myOnto:hasWikipageID". Additionally, we utilized built-in properties in OWL to specify that a property is functional or/and inverse-functional.

(Please refer to the attached data_model.graphml)

Resources

Our project has obtained resources by combining existing resources from other repositories, such as "schema" and "dbr," with resources we have created ourselves. The majority of our resources, including songs, artists, songwriters, producers, and record labels, are sourced from "dbr" as they are well-known and readily available in established repositories. This approach allows us to work more efficiently and avoid duplicating efforts in creating redundant resources.

However, to showcase what we have learned from the course and reflect the latest music trends, we have also defined our own resources within mySemantics. These resources are derived from our personal playlists, encompassing newly released songs as well as lesser-known older songs that cannot be found in "dbr." We gathered information for these resources from reputable sources such as Wikipedia, Spotify, and other online platforms.

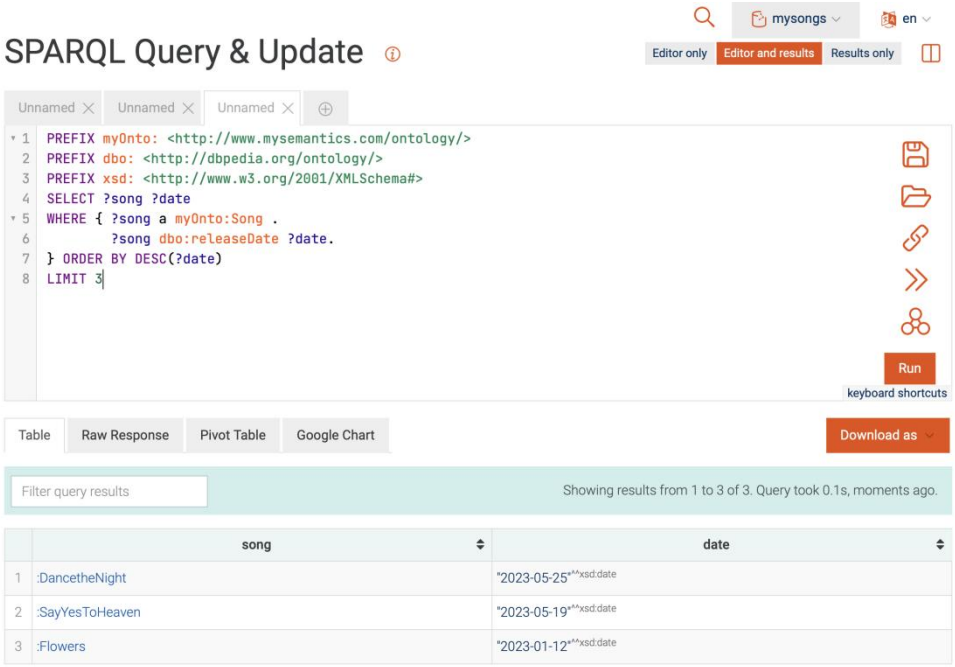
When creating a new resource, particularly a song that is not present in "dbr," we follow a specific procedure. Firstly, we establish a binding of a prefix (an empty string) to the namespace represented by mySemantics. This enables us to uniquely identify and reference the new resource within mySemantics. Subsequently, we assign the value of the new resource (the song) to a new variable, thereby creating a fresh resource within mySemantics. Finally, we add triples to define various properties and relationships associated with the new resource. For instance, we employed this process to define a recently released song named "Say Yes to Heaven," specifying its type, name, release date, artist, writer, genre, record label, producer, label, and comment.

By amalgamating existing resources and generating our own within mySemantics, we strive for efficiency by leveraging established repositories while also incorporating unique and up-to-date information from our personal sources.

SPARQL

1. What are the 3 newest songs in the RDF graph?

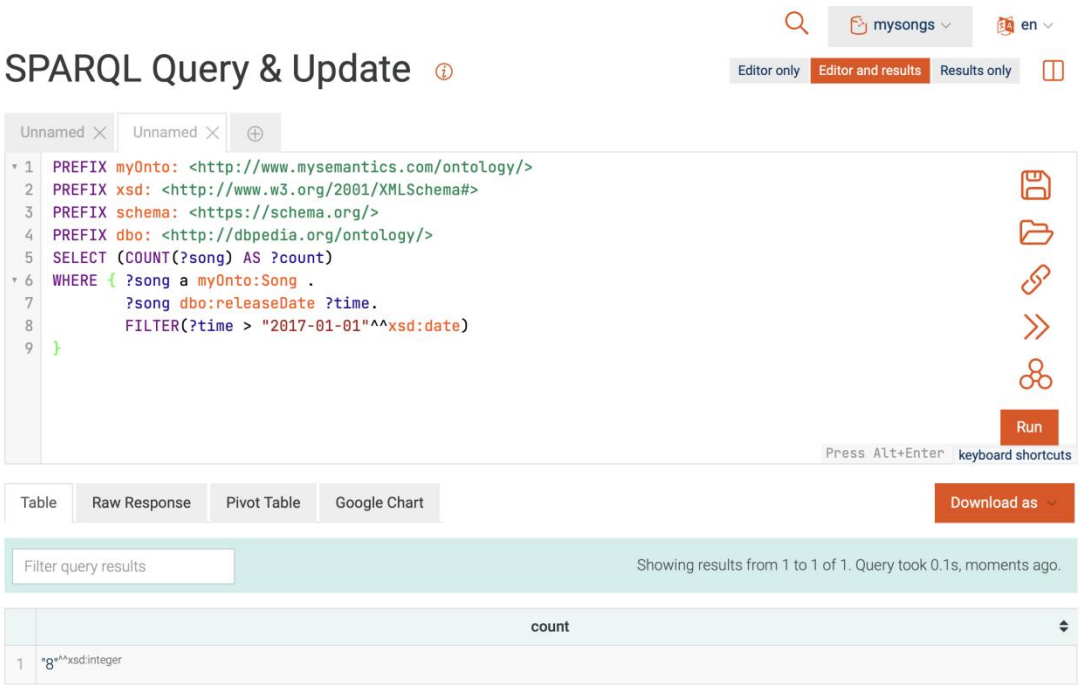
```
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?song ?date
WHERE { ?song a myOnto:Song .
        ?song dbo:releaseDate ?date.
} ORDER BY DESC(?date)
LIMIT 3
```



The answer indicates that there are three songs in the RDF graph that were released in the year 2023. This suggests that the RDF graph contains information about songs and their release dates, and the system has identified the three most recent songs based on their release dates.

2. How many songs in the graph were released after 2017?

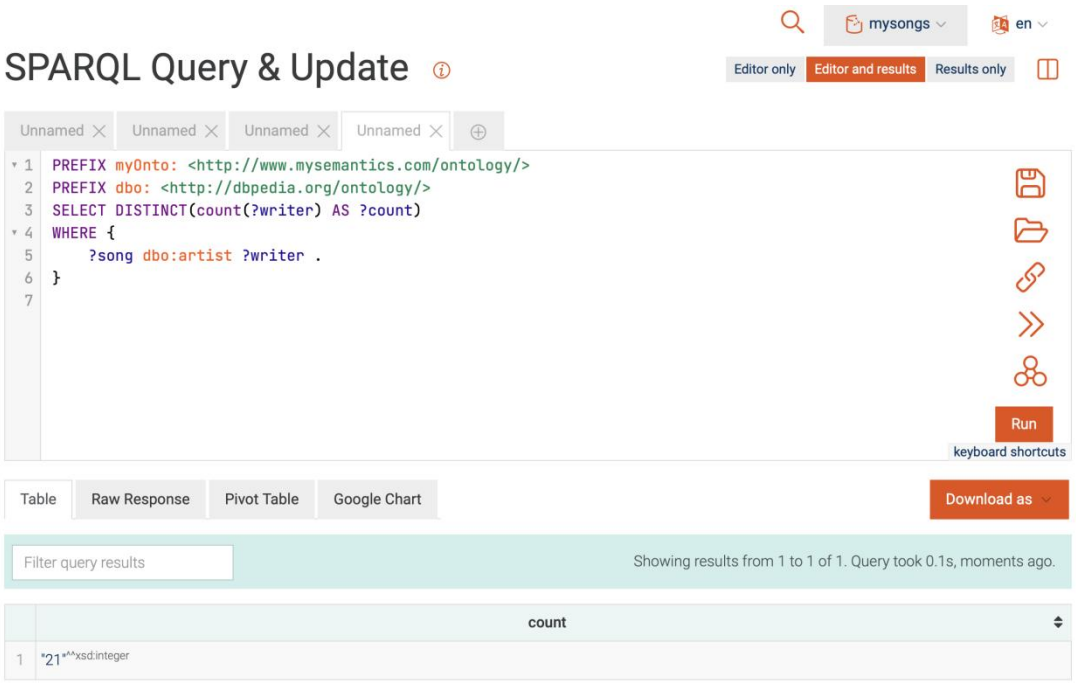
```
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX schema: <https://schema.org/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT (COUNT(?song) AS ?count)
WHERE { ?song a myOnto:Song .
        ?song dbo:releaseDate ?time.
        FILTER(?time > "2017-01-01"^^xsd:date)
}
```



The answer indicates that there are 8 songs in the RDF graph that have a release date after the year 2017. This suggests that the RDF graph contains information about songs and their release dates, and songs released after the year 2017 account for about one third of the total number of songs in the ontology.

3. How many songwriters are in the RDF graph?

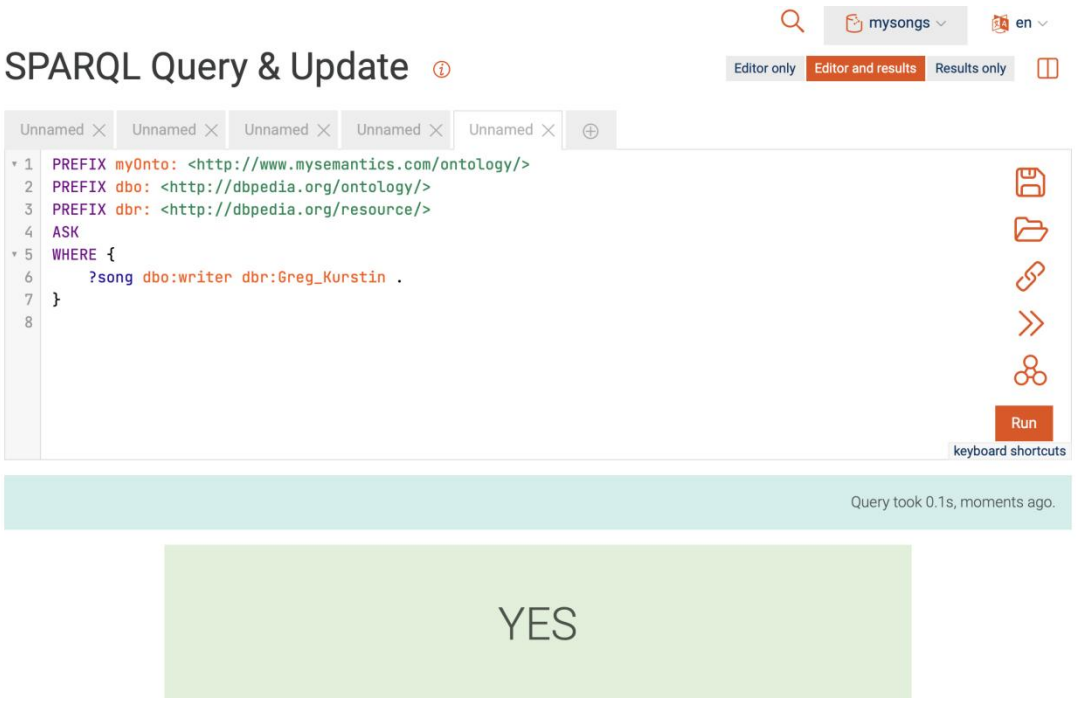
```
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT(count(?writer) AS ?count)
WHERE {
   ?song dbo:artist ?writer .
}
```



The answer to the query "How many songwriters are in the RDF graph?" indicates that there are 21 songwriters in the RDF graph. This suggests that the RDF graph contains information about songwriters associated with the songs in the graph.

4. Is there any song written by Greg Kurstin in the RDF graph?

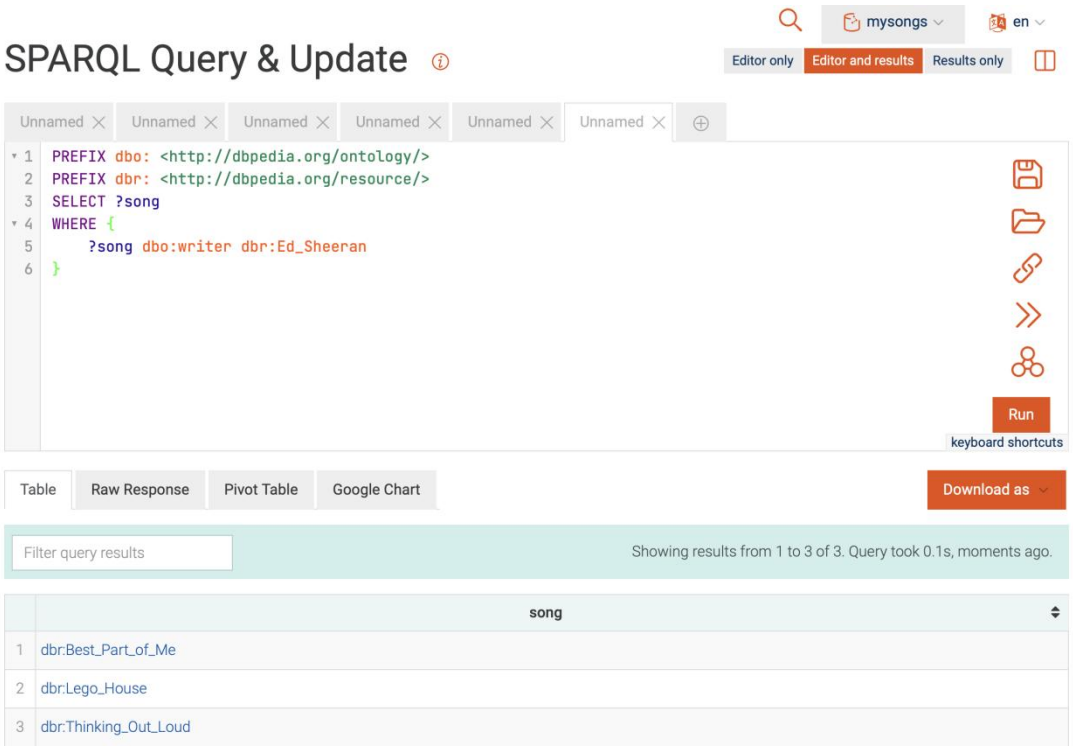
```
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
ASK
WHERE {
   ?song dbo:writer dbr:Greg_Kurstin .
}
```



The answer states that there is indeed a song in the RDF graph that is written by Greg Kurstin. This suggests that the RDF graph contains information about songs and their associated songwriters, and it has identified a song written by Greg Kurstin within the graph.

5. Find all songs written by Ed Sheeran.

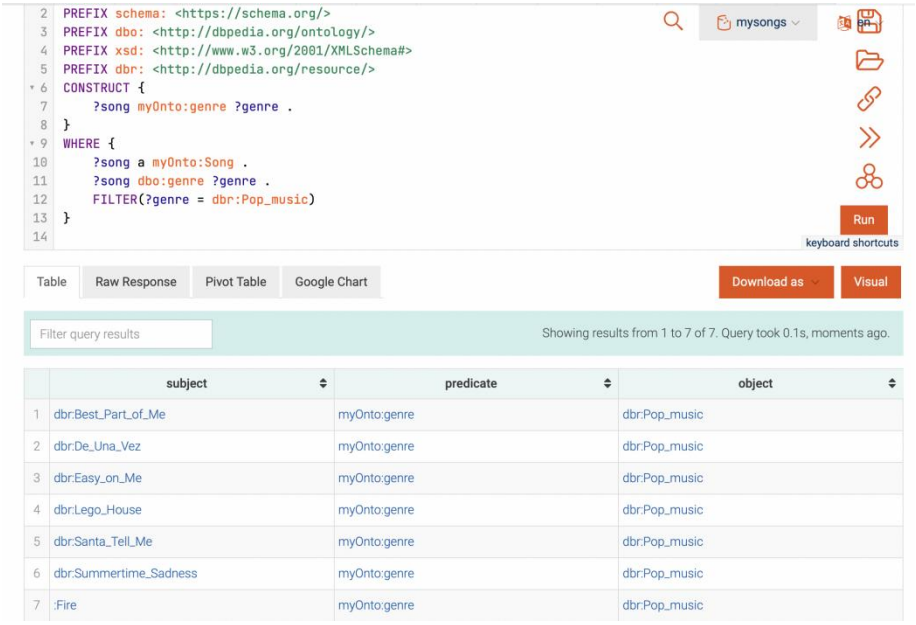
```
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?song
WHERE {
   ?song dbo:writer dbr:Ed_Sheeran
}
```



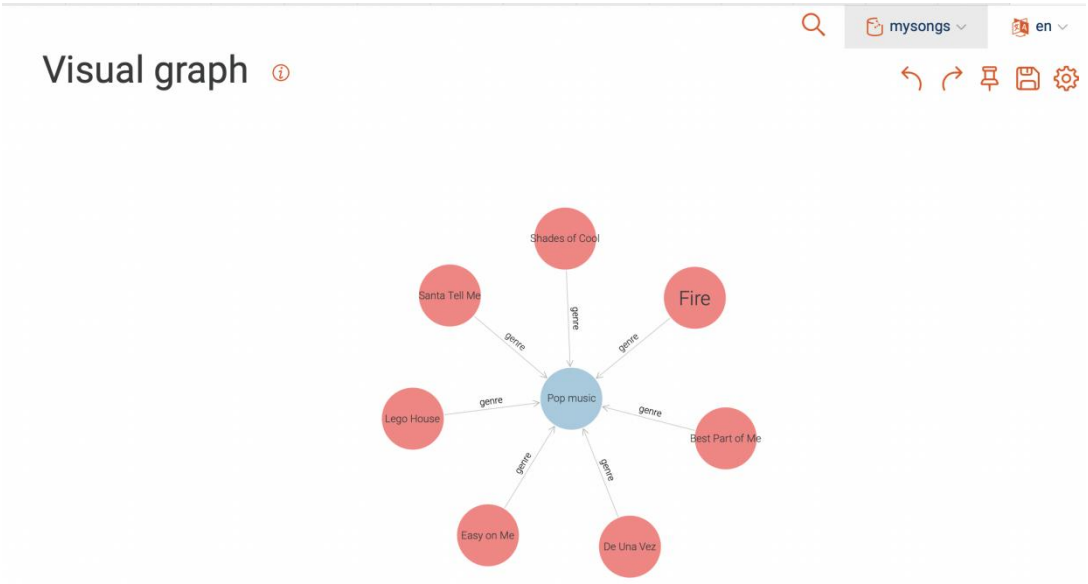
The answer provides three song titles: "Best Part of Me," "Lego House," and "Thinking Out Loud." Based on this answer, it can be inferred that the RDF graph includes information about the songwriters of each song and allows for querying and retrieving songs written by a specific songwriter, in this case, Ed Sheeran.

6. Construct a subgraph containing only the genre pop music. This graph should contain nodes representing the songs. Visualize the graph.

```
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX schema: <https://schema.org/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dbr: <http://dbpedia.org/resource/>
CONSTRUCT {
   ?song myOnto:genre ?genre .
}
WHERE {
   ?song a myOnto:Song .
   ?song dbo:genre ?genre .
   FILTER(?genre = dbr:Pop_music)
}
```



	subject	predicate	object
1	dbr:Best_Part_of_Me	myOnto:genre	dbr:Pop_music
2	dbr:De_Una_Vez	myOnto:genre	dbr:Pop_music
3	dbr:Easy_on_Me	myOnto:genre	dbr:Pop_music
4	dbr:Lego_House	myOnto:genre	dbr:Pop_music
5	dbr:Santa_Tell_Me	myOnto:genre	dbr:Pop_music
6	dbr:Summertime_Sadness	myOnto:genre	dbr:Pop_music
7	:Fire	myOnto:genre	dbr:Pop_music



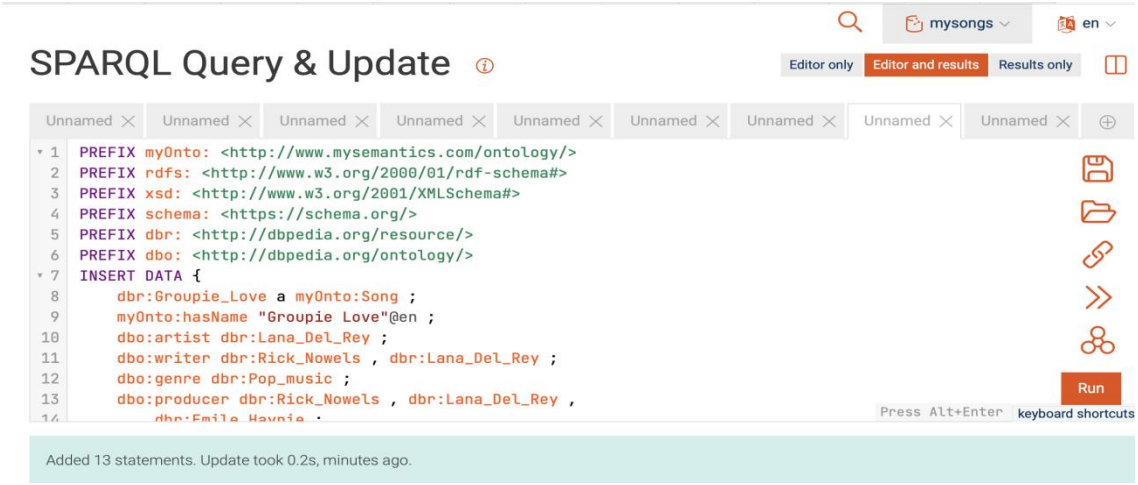
The answer states that the constructed subgraph contains seven songs associated with the genre "pop music", accounting for almost one third of the total number of songs in the ontology. This implies that the subgraph represents a subset of a larger RDF graph, focusing only on songs that are classified under the "pop music" genre.

The subgraph consists of seven nodes, each representing a song. The nodes are connected to the "pop music" genre through the dbo:genre property.

By creating this subgraph, it becomes easier to analyze and work specifically with songs that fall within the "pop music" genre, providing a more focused view of the data.

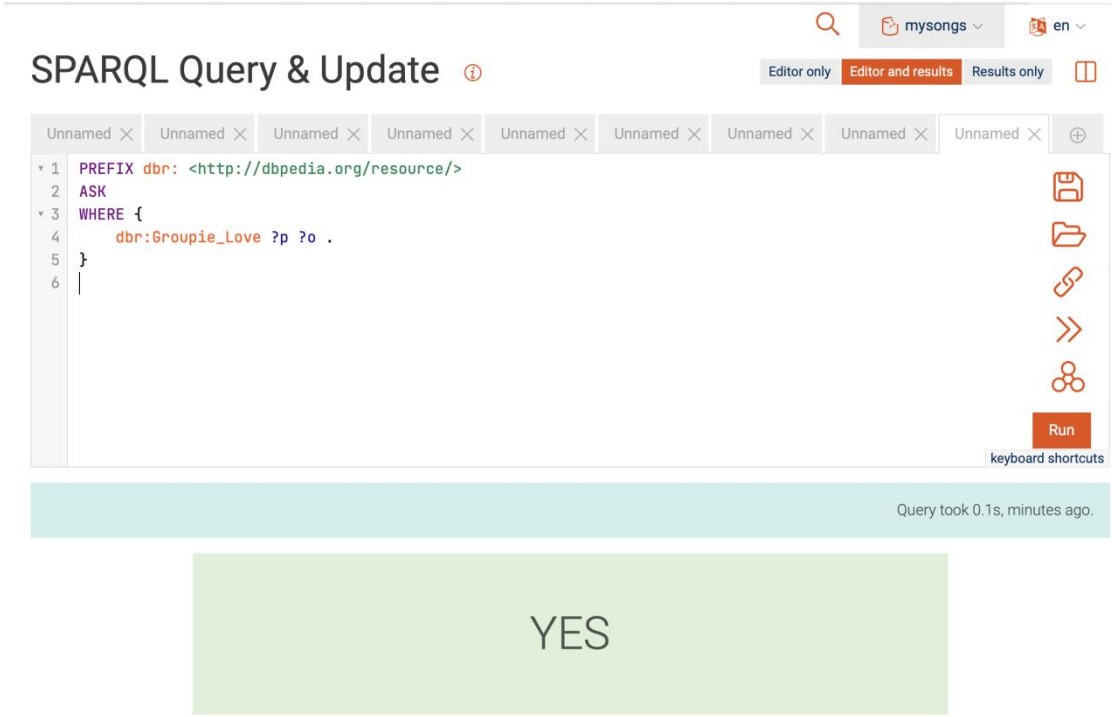
7. Write the SPARQL statement to add the following song to the graph.
Song name: Groupie Love; Release date: 2017-07-28; Artist: Lana Del Rey; Writer: Rick Nowels; Lana Del Rey; Genre: Pop music; Producer: Rick Nowels; Lana Del Rey; Emile Haynie; Record Label: Interscope Records.

```
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX schema: <https://schema.org/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
INSERT DATA {
   dbr:Groupie_Love a myOnto:Song ;
   myOnto:hasName "Groupie Love"@en ;
   dbo:artist dbr:Lana_Del_Rey ;
   dbo:writer dbr:Rick_Nowels , dbr:Lana_Del_Rey ;
   dbo:genre dbr:Pop_music ;
   dbo:producer dbr:Rick_Nowels , dbr:Lana_Del_Rey ,
      dbr:Emile_Haynie ;
   dbo:recordLabel dbr:Interscope_Records ;
   dbo:releaseDate "2017-07-28"^^xsd:date ;
   rdfs:label "Groupie Love"@en;
   rdfs:comment "A song by American singer-songwriter Lana Del Rey."@en.
}
```



We checked that the data is added to the Graph:

```
PREFIX dbr: <http://dbpedia.org/resource/>
ASK
WHERE {
   dbr:Groupie_Love ?p ?o .
}
```

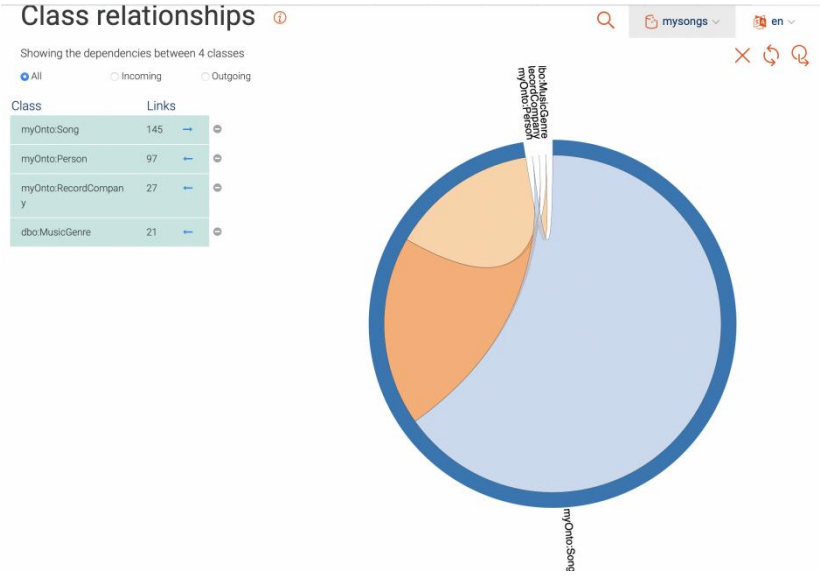


The statement "the song is successfully added to the graph" indicates that the SPARQL statement to add the song "Groupie Love" with its associated details was executed successfully. This means that the provided song details have been inserted into the RDF graph.
Analyzing the answer further, we can understand that the RDF graph now contains a new node representing the song "Groupie Love" with various properties and relationships defined. The song is associated with the provided attributes such as the song name, release date, artist, writer(s), genre, producer(s), and record label.

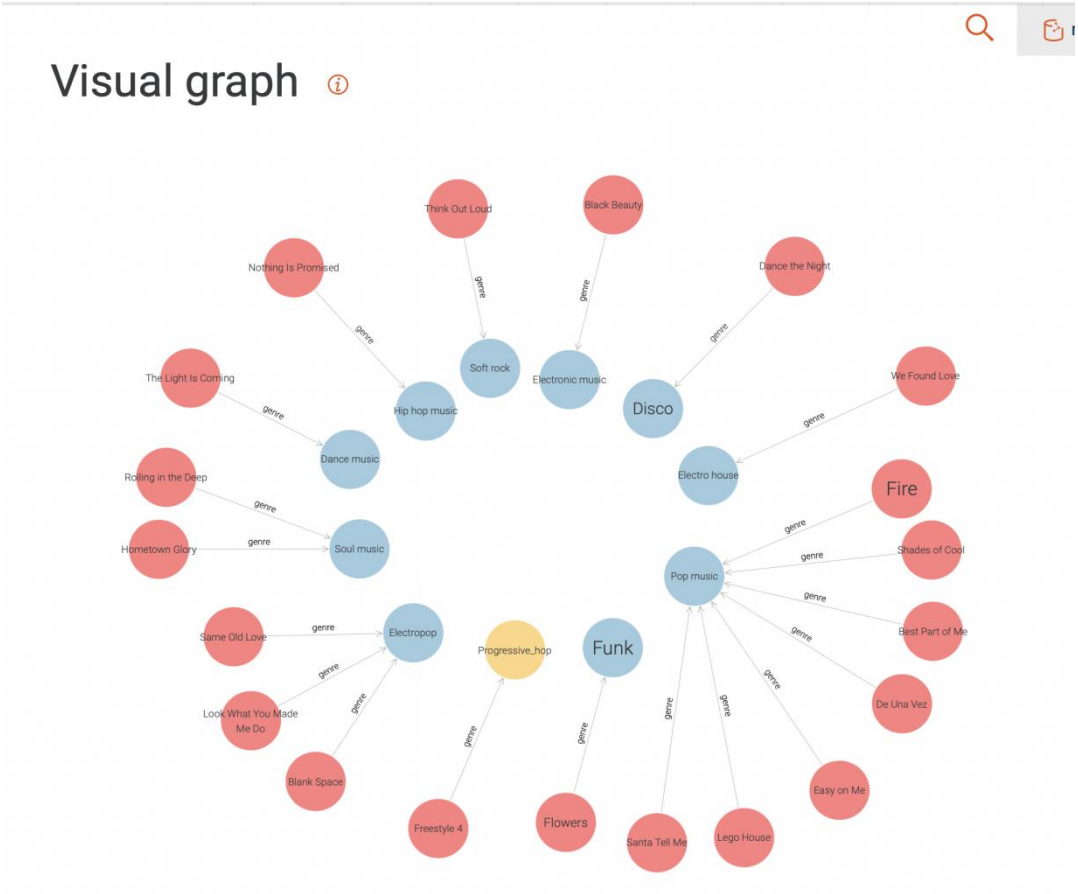
By successfully adding the song to the graph, it becomes part of the dataset and can be queried or processed along with other existing songs and their associated information. This allows for further analysis, exploration, and utilization of the song data within the context of the RDF graph.

The following visualizations depicting essential characteristics of our knowledge graph, which can help us gain an overview of our knowledge graph.

This visualization showcases the class hierarchy within our knowledge graph. It provides a clear depiction of the hierarchical structure of the classes. We generated this visualization using Graph DB, accessed by navigating to the Explore section and selecting the option for Class hierarchy.



The above visualization illustrates the interconnections between our classes, shedding light on their relationships. Notably, the class Song stands out with the highest number of links, totaling 145, while the class MusicGenre exhibits the fewest links among the four classes. This emphasizes that our knowledge graph centers around songs as a primary focus. We generated this visualization by utilizing Graph DB, accessing the Explore section, and specifically selecting the option for Class relationships.



This visualization demonstrates the genres with corresponding songs. It tells us that there are 11 music genres in the knowledge graph, and the ontology covers a variety of different music

genres. Of the genres, songs belongin to pop music are the most. This visualization was created using Graph DB in the following steps:

First, use SPARQL to query all the genres and corresponding songs.
This visualization showcases the relationship between genres and their associated songs within our knowledge graph. It provides insights into the diversity of music genres covered in the ontology, revealing a total of 11 distinct genres. Among these genres, pop music stands out with the highest representation of songs.

To create this visualization, we employed Graph DB and followed the following steps:

1. Utilized SPARQL to query and retrieve all genres along with their corresponding songs. Details are as follows:

```
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX schema: <https://schema.org/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dbr: <http://dbpedia.org/resource/>
CONSTRUCT {
   ?song myOnto:genre ?genre .
}
WHERE {
   ?song a myOnto:Song .
   ?song dbo:genre ?genre .
}
```

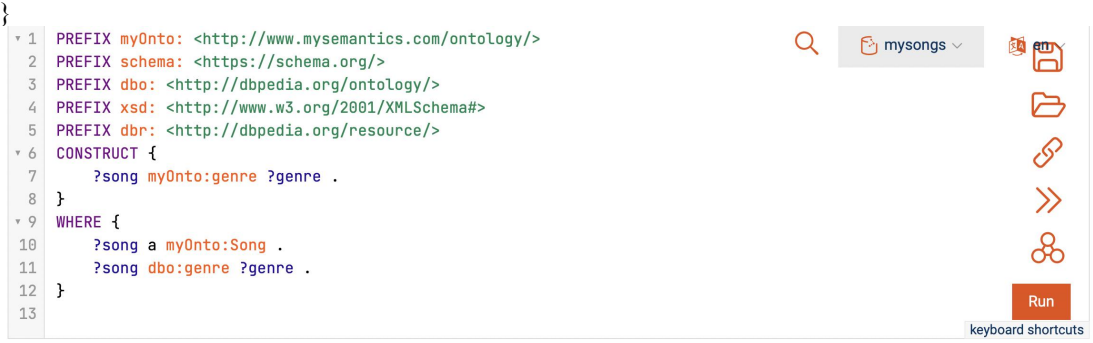


Table | Raw Response | Pivot Table | Google Chart — Download as — Visual

Filter query results — Showing results from 1 to 22 of 22. Query took 0.1s, today at 20:53.

	subject	predicate	object
1	dbr:Best_Part_of_Me	myOnto:genre	dbr:Pop_music
2	dbr:Blank_Space	myOnto:genre	dbr:Electropop
3	dbr:De_Una_Vez	myOnto:genre	dbr:Pop_music
4	dbr:Easy_on_Me	myOnto:genre	dbr:Pop_music
5	dbr:Freestyle_4	myOnto:genre	dbr:Progressive_hop
6	dbr:Hometown_Glory	myOnto:genre	dbr:Soul_music
7	dbr:Lego_House	myOnto:genre	dbr:Pop_music

8	dbr:Look_What_You_Made_Me_Do	myOnto:genre	dbr:Electropop
9	dbr:Nothing_Is_Promised	myOnto:genre	dbr:Hip_hop_music
10	dbr:Rolling_in_the_Deep	myOnto:genre	dbr:Soul_music
11	dbr:Same_Old_Love	myOnto:genre	dbr:Electropop
12	dbr:Santa_Tell_Me	myOnto:genre	dbr:Pop_music
13	dbr:Summertime_Sadness	myOnto:genre	dbr:Pop_music
14	dbr:The_Light_Is_Coming	myOnto:genre	dbr:Dance_music
15	dbr:Thinking_Out_Loud	myOnto:genre	dbr:Soft_rock
16	dbr:We_Found_Love	myOnto:genre	dbr:Electro_house
17	:BlackBeauty	myOnto:genre	dbr:Electronic_music
18	:DancetheNight	myOnto:genre	dbr:Disco
19	:Fire	myOnto:genre	dbr:Pop_music
20	:Flowers	myOnto:genre	dbr:Funk
21	:SayYesToHeaven	myOnto:genre	dbr:Soft_rock
22	dbr:Groupie_Love	myOnto:genre	dbr:Pop_music

2. We click Visual button on the query page and get a visualization like this.

Optional Tasks

SHACL

Please refer to the attached file: shacl.ipynb.

RDF* and SPARQL*

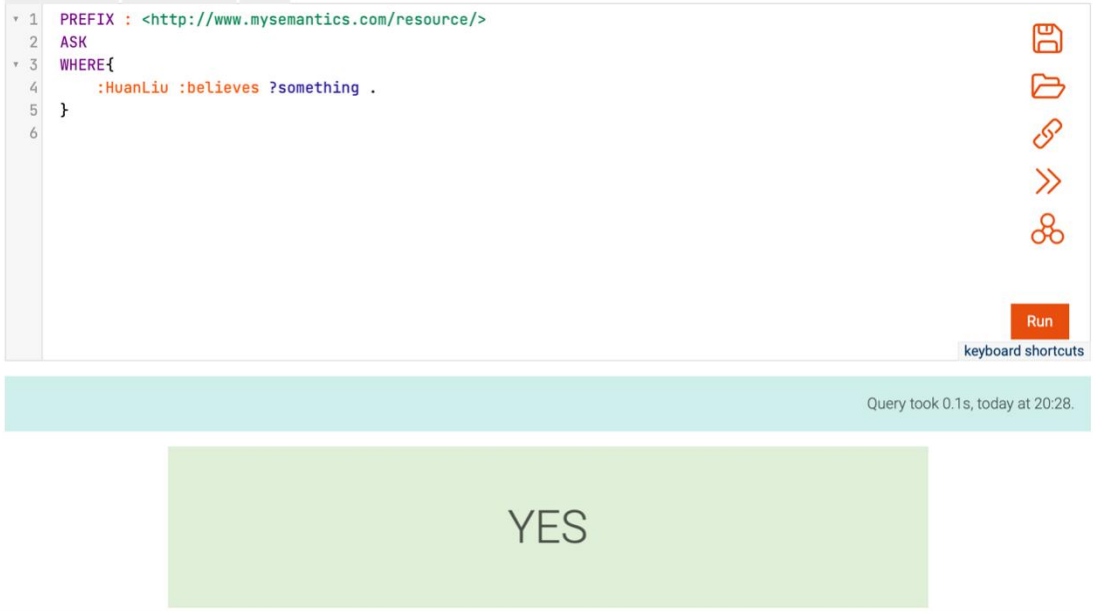
Insert data

PREFIX : <http://www.mysemantics.com/resource/>
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
INSERT DATA{
  :HuanLiu a myOnto:Person .
  :HuanLiu :believes <<dbr:Best_Part_of_Me :sungBy dbr:Ed_Sheeran>>
}

Check whether the data has been successfully added

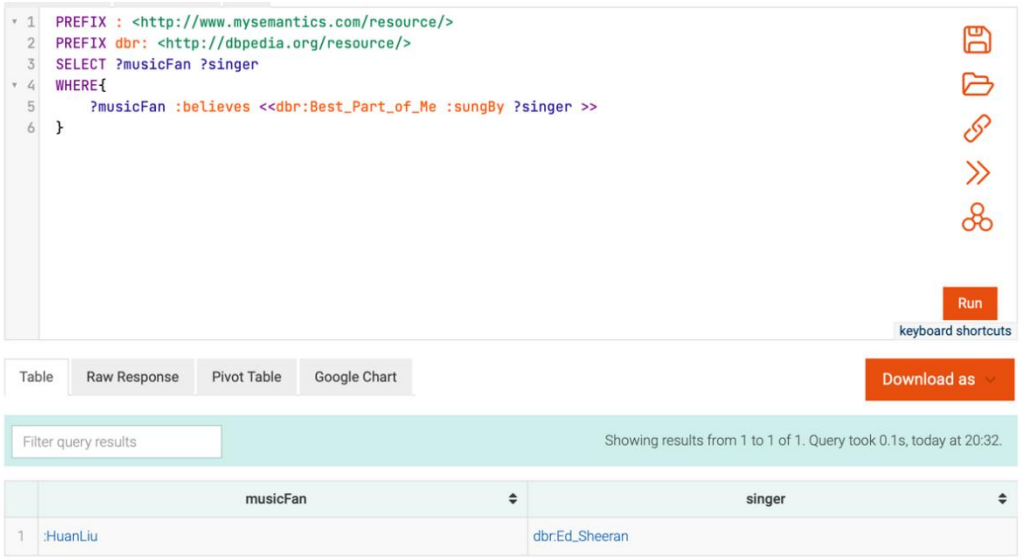
```
PREFIX : <http://www.mysemantics.com/resource/>
ASK
WHERE{
   :HuanLiu :believes ?something .

}
```



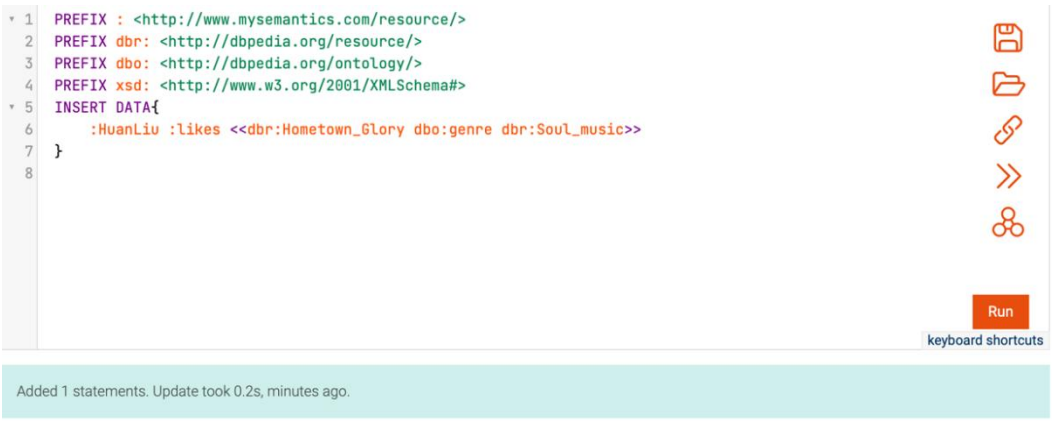
Which music fan believes that the song “Best Part of Me” is sung by which singer?

```
PREFIX : <http://www.mysemantics.com/resource/>
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT ?musicFan ?singer
WHERE{
        ?musicFan :believes <<dbr:Best_Part_of_Me :sungBy ?singer >>
}
```



Insert data

PREFIX : <http://www.mysemantics.com/resource/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
INSERT DATA{
:HuanLiu :likes <<dbr:Hometown_Glory dbo:genre dbr:Soul_music>>
}



Who likes which song and which genre does this song belong to?

PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX : <http://www.mysemantics.com/resource/>
SELECT ?musicFan ?song ?genre
WHERE{
?musicFan :likes <<?song dbo:genre ?genre>>
}

Insert data

PREFIX : <http://www.mysemantics.com/resource/>
PREFIX myOnto: <http://www.mysemantics.com/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
INSERT DATA{
:Xiaojing a myOnto:Person .
:Xiaojing :listensTo <<dbr:Easy_on_Me dbo:releaseDate "2021-10-15"^^xsd:date>>
}

Which music fan listens to a song released on 2021/10/15 and what is the name of the song?

PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX : <http://www.mysemantics.com/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?musicFan ?song
WHERE{
?musicFan :listensTo <<?song dbo:releaseDate "2021-10-15"^^xsd:date>>
}

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX : <http://www.mysemantics.com/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?musicFan ?song
WHERE{
    ?musicFan :listensTo <<?song dbo:releaseDate "2021-10-15"^^xsd:date>>
}
```

Run

keyboard shortcuts

Table | Raw Response | Pivot Table | Google Chart | Download as

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, today at 20:40.

	musicFan	song
1	:Xiaojing	dbr:Easy_on_Me

Construct a graph about the opinion about a song and visualize it.

```
PREFIX : <http://www.mysemantics.com/resource/>
CONSTRUCT{
   ?musicFan :believes ?something .
}
WHERE{
   ?musicFan :believes ?something .
}
```

```
PREFIX : <http://www.mysemantics.com/resource/>
CONSTRUCT{
    ?musicFan :believes ?something .
}
WHERE{
    ?musicFan :believes ?something .
}
```

Run

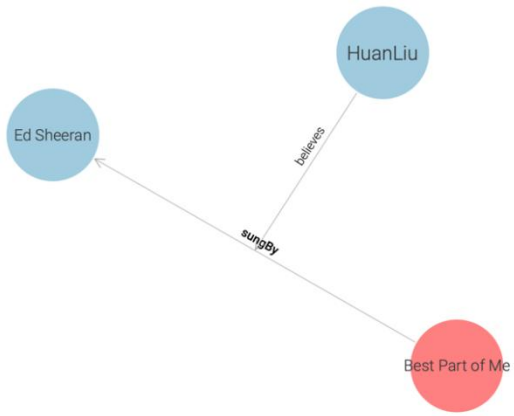
keyboard shortcuts

Table | Raw Response | Pivot Table | Google Chart | Download as | Visual

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, minutes ago.

	subject	predicate	object
1	:HuanLiu	:believes	<< dbr:Best_Part_of_Me :sungBy dbr:Ed_Sheeran >>



Construct a graph about someone listens to something.

```
PREFIX : <http://www.mysemantics.com/resource/>
CONSTRUCT{
   ?musicFan :listensTo ?something .
}
WHERE{
   ?musicFan :listensTo ?something .
}
```

```
PREFIX : <http://www.mysemantics.com/resource/>
CONSTRUCT{
    ?musicFan :listensTo ?something .
}
WHERE{
    ?musicFan :listensTo ?something .
}
```

Run

keyboard shortcuts

Table | Raw Response | Pivot Table | Google Chart

Download as | Visual

Filter query results

Showing results from 1 to 1 of 1. Query took 0.1s, minutes ago.

	subject	predicate	object
1	:Xiaojing	:listensTo	<< dbr:Easy_on_Me dbo:releaseDate "2021-10-15"^^xsd:date >>