Project Submission

Program: CESS

Course Code: CSE 488

Course Name: Ontologies and the Semantic Web

Submitted to

Dr. Ensaf Hussein Mohamed

Done by Team 1

Abdel-Rahman Ibrahim El Said Ahmed Megahed	18P7423
Khaled Medhat Mahmoud Khalifa	18P3557
Mariam Yasser Abdelmageed Meckawy	18P3059
Mohamed Magdy Mostafa	18P5160

Ain Shams University
Faculty of Engineering
Spring Semester – 2023

Table of Contents

Table of Figures

1. Ontology Topic

Our chosen topic for the ontology creation revolves around the renowned music band, Metallica. Formed in 1981, Metallica has gained worldwide recognition for their contributions to the heavy metal genre. The band currently consists of four talented members: James Hetfield, who serves as the rhythm guitarist and vocalist; Kirk Hammett, the lead guitarist; Lars Ulrich, the drummer; and Robert Trujillo, the bassist. Over the years, Metallica has released numerous studio albums, showcasing their musical prowess. In addition to the studio recordings, the band has also performed live versions of their songs, which have sparked debates among fans about their superiority compared to the original studio tracks.

We were drawn to this particular topic of Metallica for our ontology due to our deep appreciation for the heavy metal genre as a whole, and our admiration for the unique contributions and artistry of this band. Metallica's rich discography and their ability to captivate audiences through their energetic performances make them an ideal subject for our ontological exploration.

2. Ontology Details

In the following section, we'll discuss the details of our ontology, including the classes, object properties, data properties, instances, and cardinality.

i. Classes

We have a total of 3 classes in our ontology: Band, MusicAlbum, and Person. The instances of each are:

Band:

- Metallica

MusicAlbum:

- Ride The Lightning
- Master Of Puppets
- And Justice For All

Person:

- James Hetfield
- Kirk Hammett
- Lars Ulrich
- Robert Trujillo

ii. Object Properties

Regarding the object properties, we have a total of 2.

hasAlbum

Describes what albums does the band have

- **Domain:** Band
- **Range:** MusicAlbum
- **Cardinality: Band** has a minimum of 1 **MusicAlbum**

hasMember

Describes which members does the band have

- **Domain:** Band
- **Range:** Person
- **Cardinality: Band** has a minimum of 2 **Person**

iii. Data Properties

We have a total of 4 data properties:

hasRole

Describes the role of each member

- **Domain:** Person
- **Range:** xsd:string
- **Values:** Bassist, Drummer, Lead Guitarist, Rhythm Guitarist, Vocalist

hasSong

Describes the songs of each album

- **Domain:** MusicAlbum
- **Range:** xsd:string

placeFormedIn

Describes the place in which the band was formed in

- **Domain:** Band
- **Range:** xsd:string

timeFormedIn

Describes the time in which the band was formed in

- **Domain:** Band
- **Range:** xsd:int

3. Protégé

In this subsection, we delve into the work we accomplished in the application Protégé to develop an ontology for our project. Protégé, a widely-used ontology development tool, provided us with a robust platform to design and construct a formal representation of our domain knowledge.

Creating an ontology serves as a pivotal step in structuring and organizing information within a specific domain. It enables us to define concepts, their properties, and the relationships that exist among them. By employing Protégé, we leveraged its intuitive interface and powerful features to create a comprehensive ontology tailored to our project's requirements.

Our efforts in Protégé involved a meticulous process of analyzing our domain, identifying key entities, determining their attributes, and specifying the relationships between them. Through an iterative approach, we refined and expanded our ontology, ensuring its accuracy, completeness, and adherence to standard ontology design principles.

By constructing this ontology, we established a shared understanding of the domain, facilitating effective communication and knowledge sharing among project stakeholders. The ontology serves as a semantic framework that underlies our project's data model, enabling us to infer new information, query the data effectively, and support various knowledge-driven functionalities.

Throughout this subsection, we will explore the ontology development process in Protégé in detail. We will discuss the conceptualization of domain entities, the definition of their properties and relationships, and the organization of the ontology's structure. Furthermore, we will highlight the considerations and decisions made during the development process, shedding light on the rationale behind our ontology design choices.

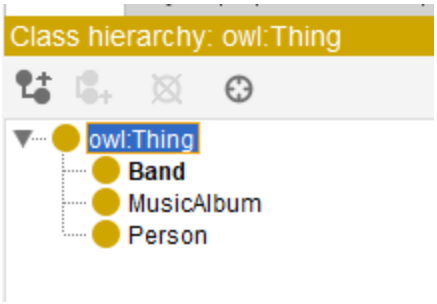


Figure 2 Ontology class hierarchy

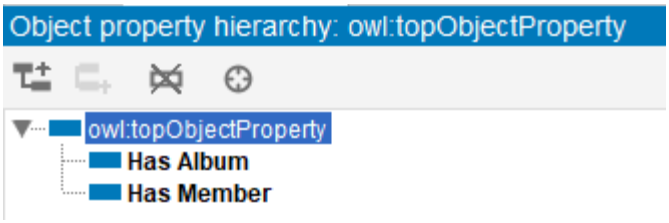


Figure 1 Ontology object property

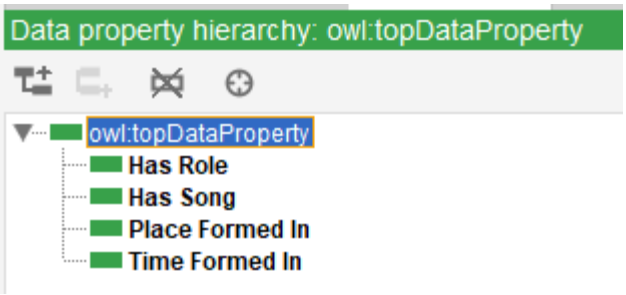


Figure 3 Ontology data property

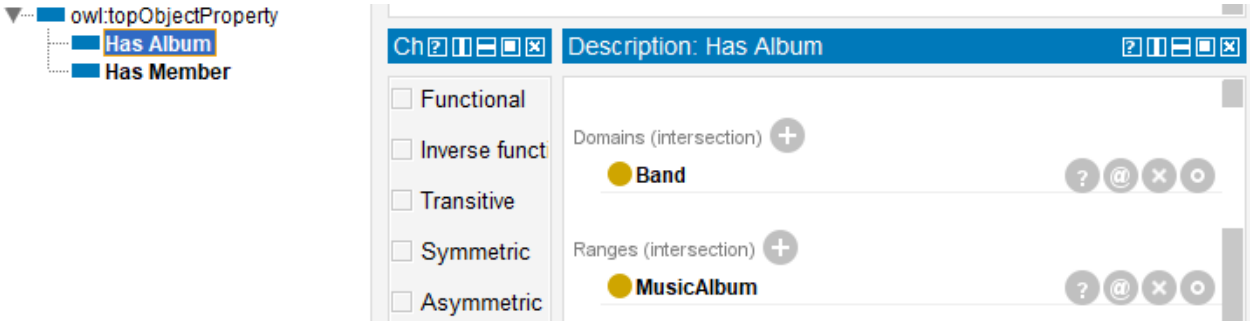


Figure 4 Has Album property

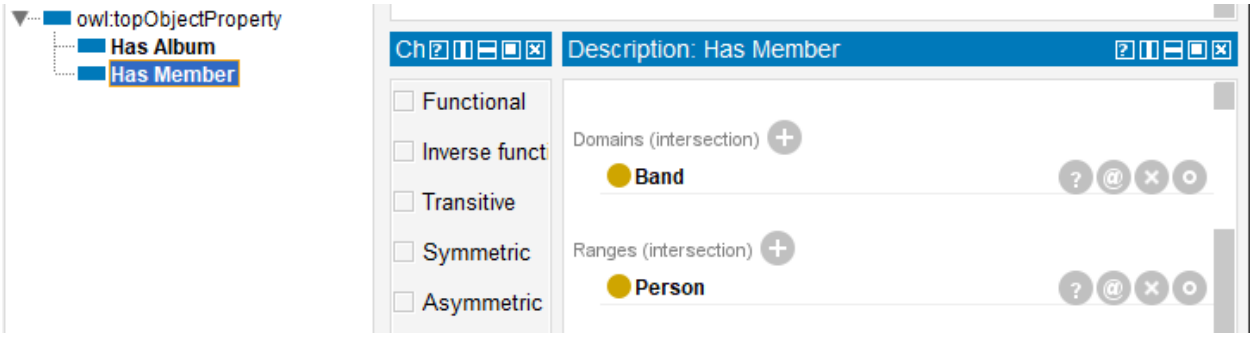


Figure 5 Has Member property

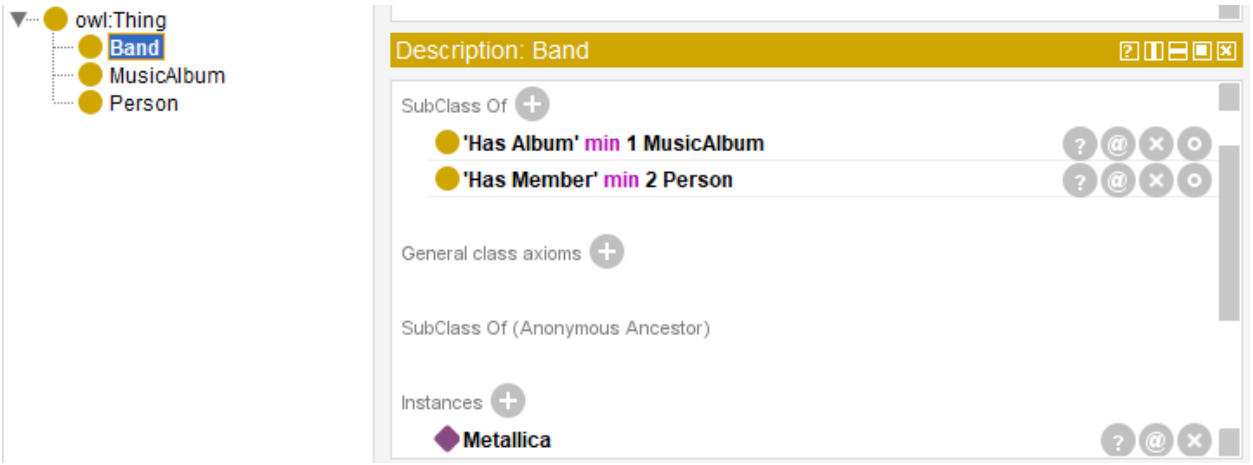


Figure 6 Band class

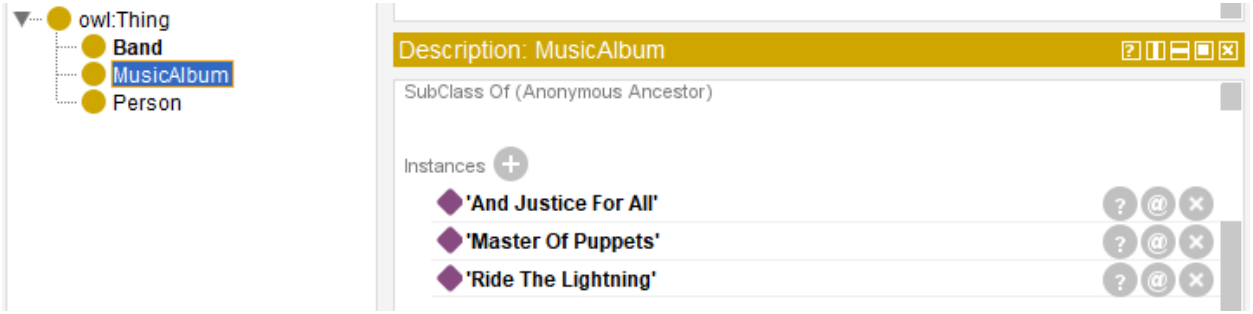


Figure 7 MusicAlbum class

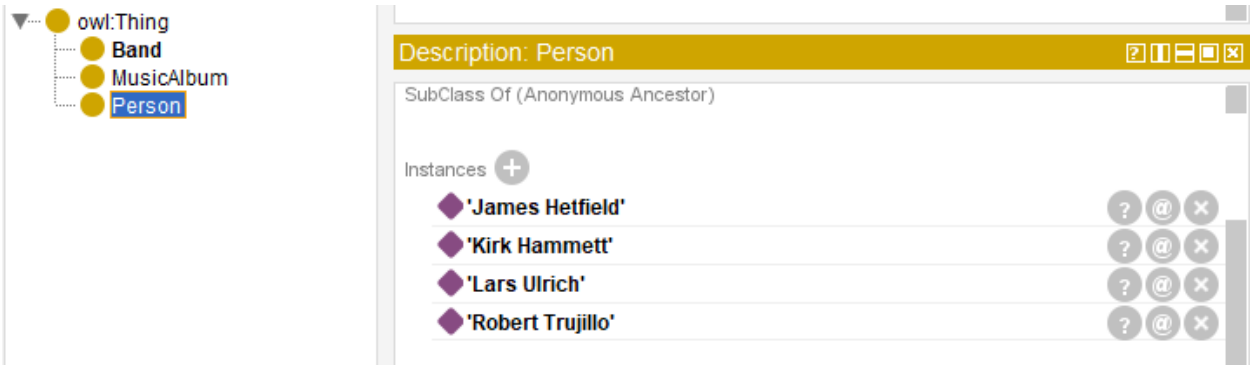


Figure 8 Person class

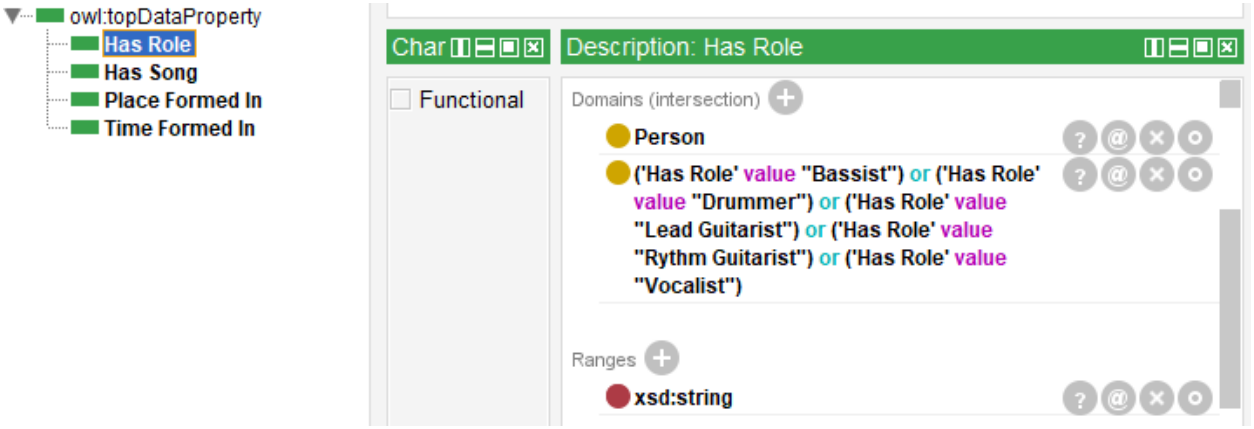


Figure 9 Has Role property

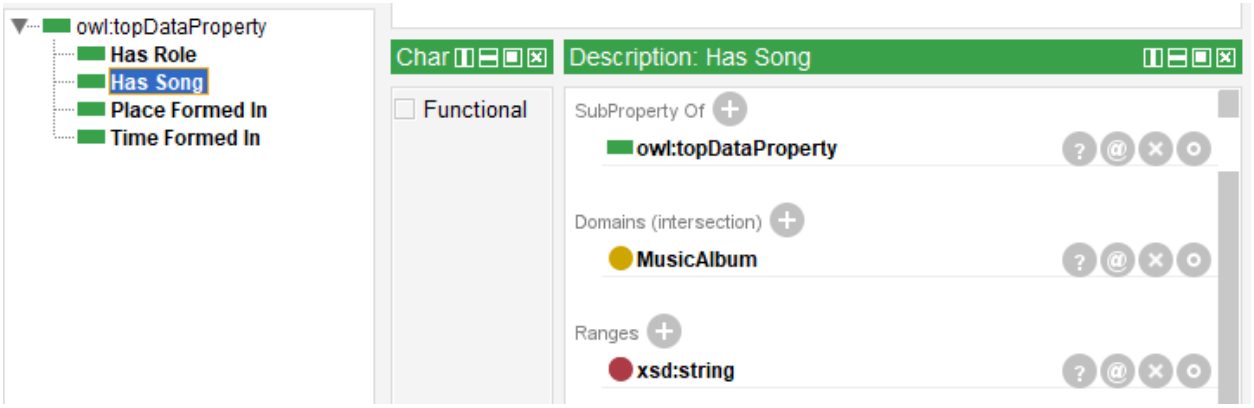


Figure 10 Has Song property

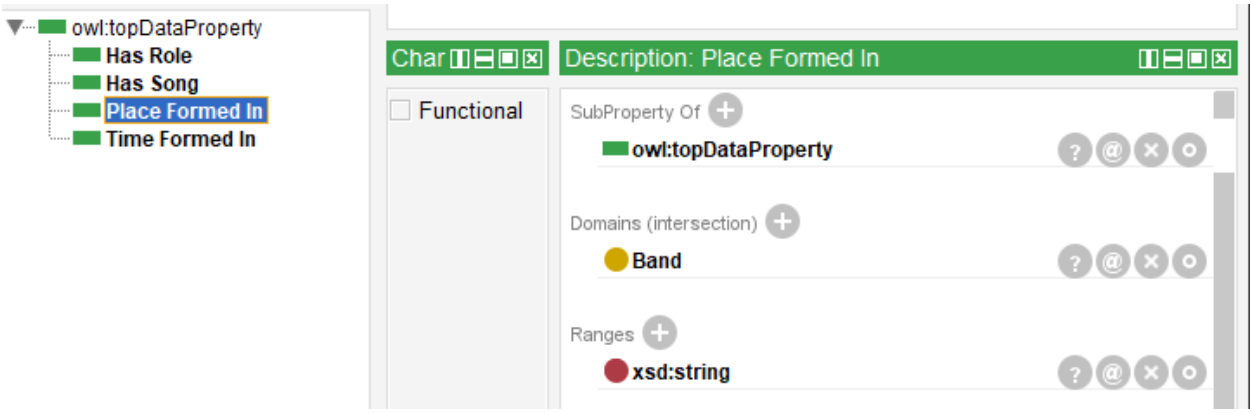


Figure 11 Place Formed In property

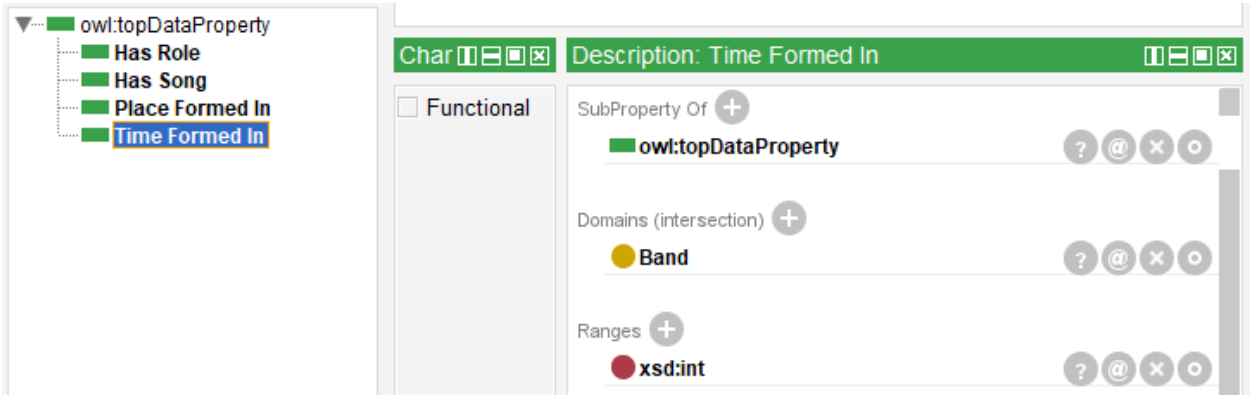


Figure 12 Time Formed In property

4. RDF Graphs

In this subsection, we discuss the RDF (Resource Description Framework) graph that we have created as part of our report. The RDF graph serves as a structured representation of our data, allowing us to model and interconnect various entities and their relationships.

We conducted two iterations of creating the RDF graph, each with a specific focus. The first iteration involved constructing the RDF graph without incorporating the individuals, solely emphasizing the schema and ontology. This initial step allowed us to define the classes, properties, and relationships that form the foundation of our data model. By establishing a clear structure, we laid the groundwork for organizing and categorizing our information effectively.

In the second iteration, we enhanced the RDF graph by incorporating the individuals, which represent specific instances or examples within our data domain. By including these individuals, we added real-world context to our RDF graph, making it more representative of the actual data we are working with. This step further enriches our understanding of the relationships and connections between entities, enabling us to derive meaningful insights and draw conclusions.

Throughout this subsection, we will delve into the details of both iterations of the RDF graph creation process, examining the structure, elements, and interdependencies of the graph. By exploring both the schema-focused and instance-focused representations, we aim to provide a comprehensive overview of our data model, facilitating a deeper understanding of the information and its underlying structure.

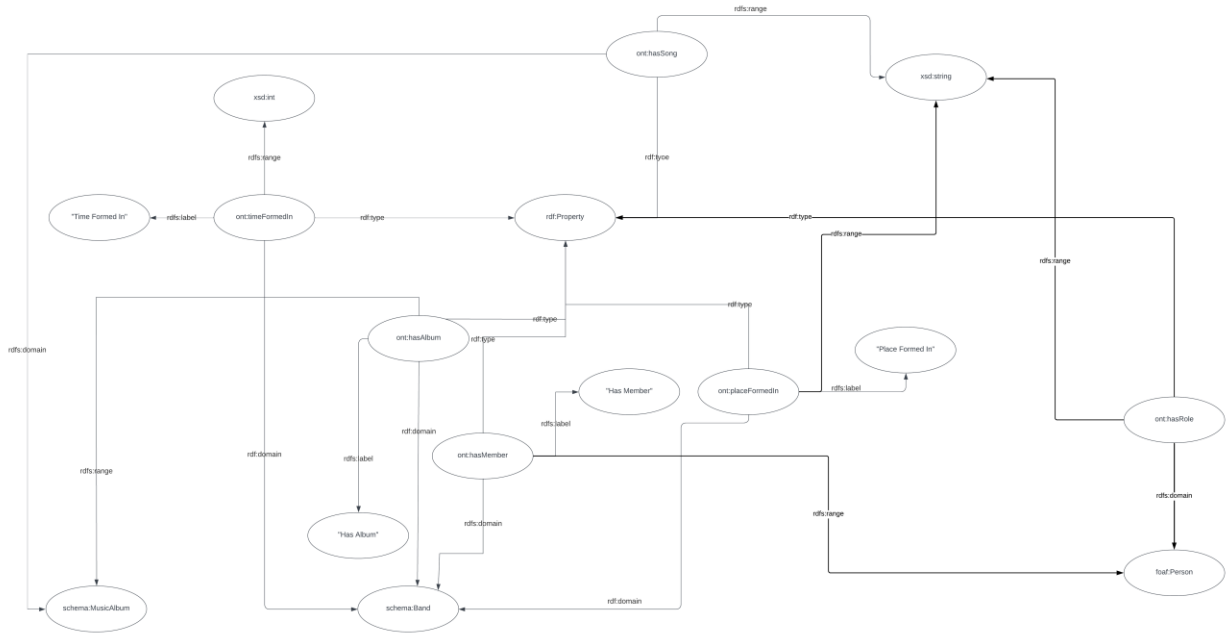


Figure 13 The RDF Graph Showing only the Classes and the Properties.

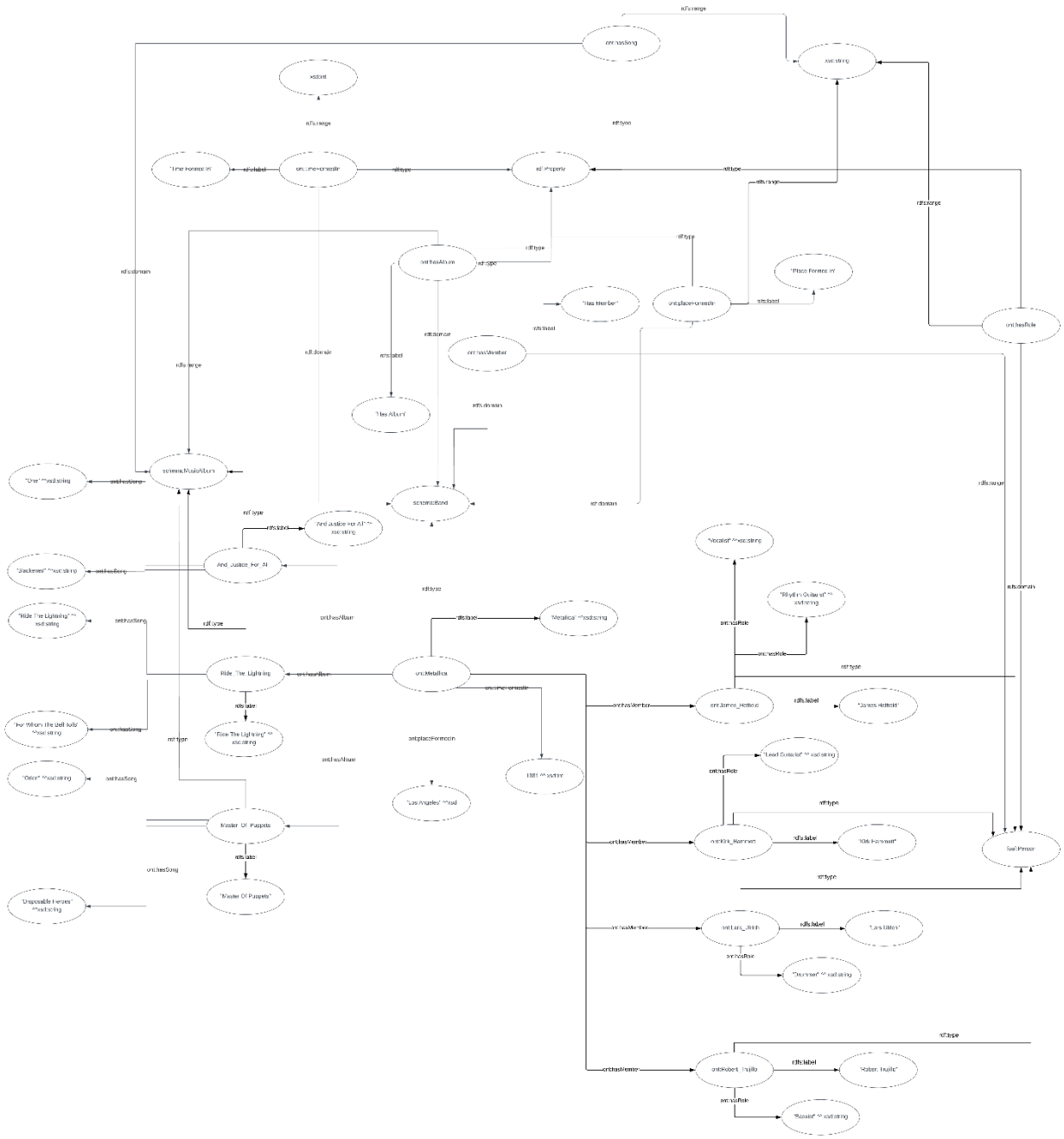


Figure 14 The Final RDF Graph showing the Classes, Properties, and Individuals.

5. Website Input and Output

4.1 Query 1

Query to show all band members and each one's role.

SPARQL Query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ont: <http://www.semanticweb.org/khale/ontologies/2023/4/untitled-ontology-4#>

SELECT ?person ?role
WHERE {
 ?person rdf:type ont:Person.
 ont:Metallica ont:hasMember ?person.
 ?person ont:hasRole ?role.
}
```

Figure 15 Query 1

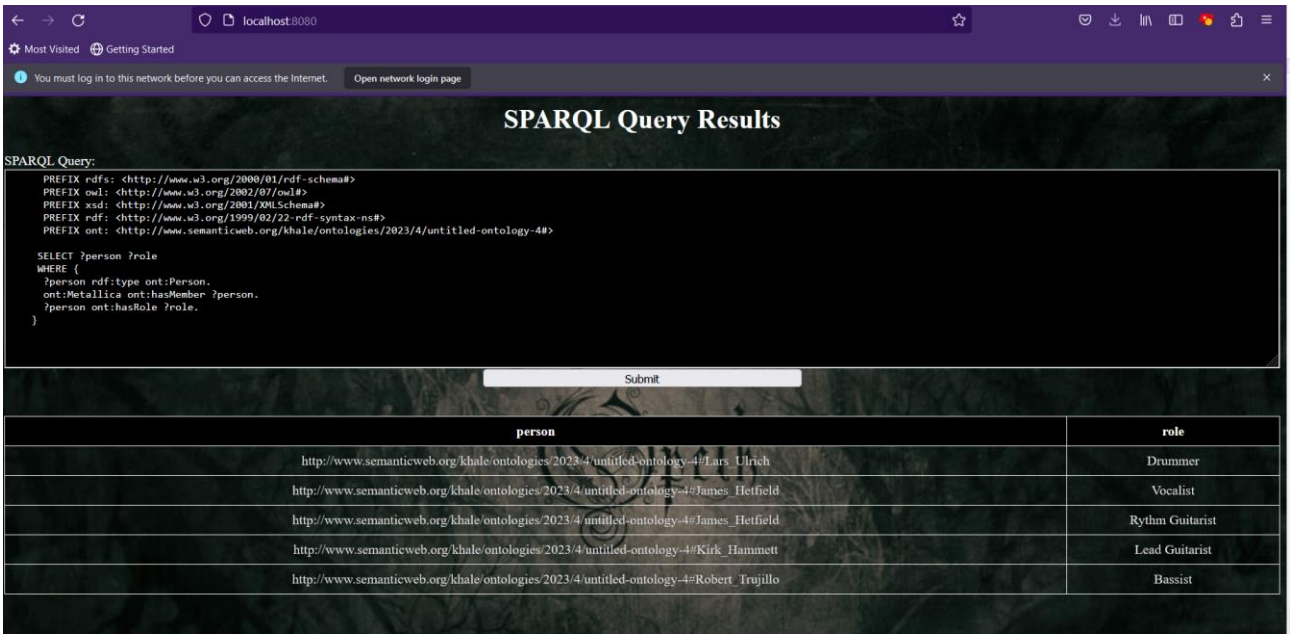


Figure 16 Query 1 output

4.2 Query 2

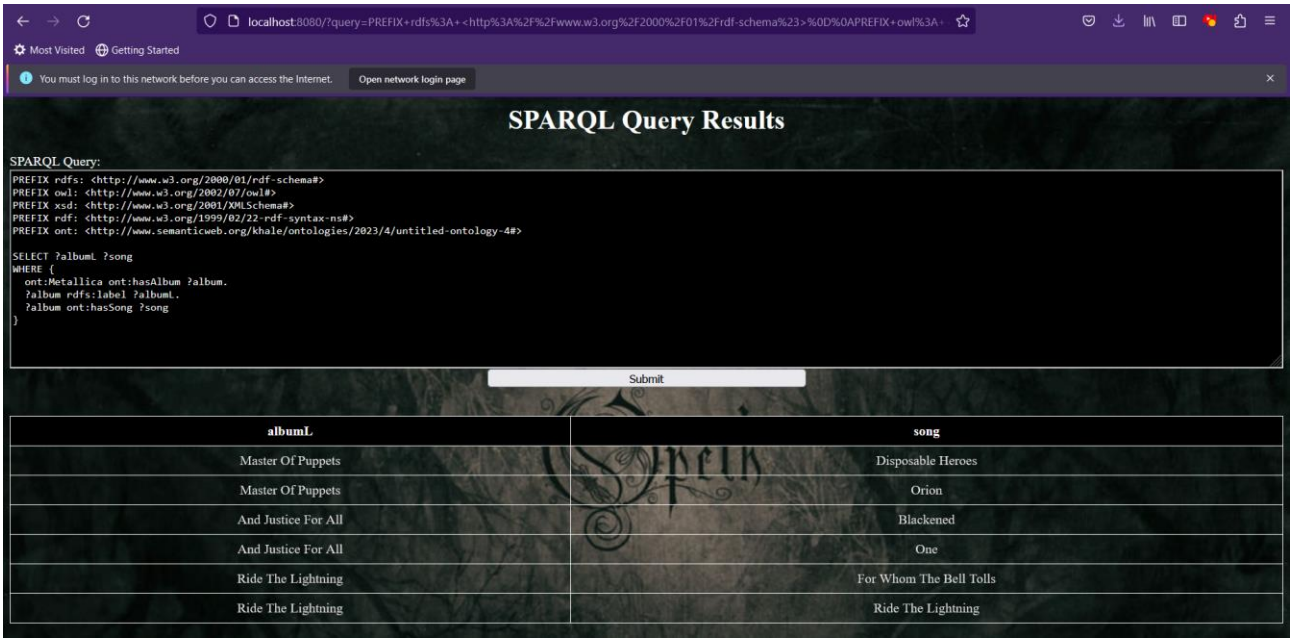
Query to show all the band's albums and their songs.

SPARQL Query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ont: <http://www.semanticweb.org/khale/ontologies/2023/4/untitled-ontology-4#>

SELECT ?albumL ?song
WHERE {
  ont:Metallica ont:hasAlbum ?album.
  ?album rdfs:label ?albumL.
  ?album ont:hasSong ?song
}
```

Figure 17 Query 2



albumL	song
Master Of Puppets	Disposable Heroes
Master Of Puppets	Orion
And Justice For All	Blackened
And Justice For All	One
Ride The Lightning	For Whom The Bell Tolls
Ride The Lightning	Ride The Lightning

Figure 18 Query 2 output

4.3 Query 3

Query to show all members with role “Guitarist”.

SPARQL Query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ont: <http://www.semanticweb.org/khale/ontologies/2023/4/untitled-ontology-4#>

SELECT ?personL
WHERE {
  ont:Metallica ont:hasMember ?person.
  ?person ont:hasRole ?role.
  ?person rdfs:label ?personL.
  FILTER REGEX( ?role , "Guitarist$", "i").
}
```

Figure 19 Query 3

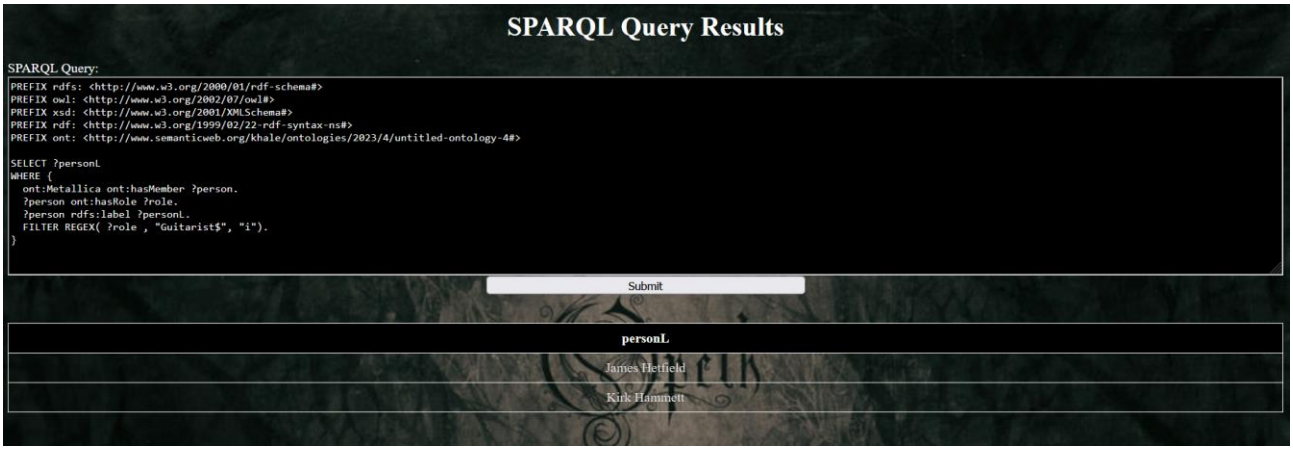


Figure 20 Query 3 output

4.4 Query 4

Query to show place and time the band was formed in.

SPARQL Query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ont: <http://www.semanticweb.org/khale/ontologies/2023/4/untitled-ontology-4#>

SELECT ?bandL ?placeFormedIn ?timeFormedIn
WHERE {
  ?band rdf:type ont:Band.
  ?band rdfs:label ?bandL.
  ?band ont:placeFormedIn  ?placeFormedIn.
  ?band ont:timeFormedIn ?timeFormedIn.
}
```

Figure 21 Query 4

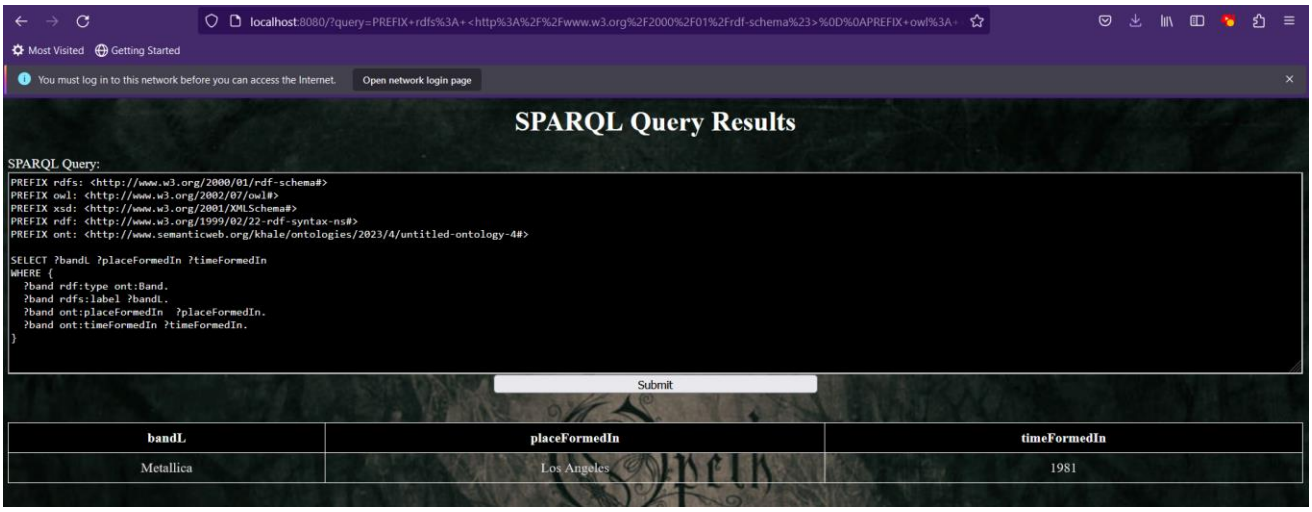


Figure 22 Query 4 output

4.5 Query 5

Query to show all the songs in the album which contains the song “Orion”.

SPARQL Query:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ont: <http://www.semanticweb.org/khale/ontologies/2023/4/untitled-ontology-4#>

SELECT ?albumL ?song
WHERE {
  ont:Metallica ont:hasAlbum ?album.
  ?album ont:hasSong "Orion".
  ?album rdfs:label ?albumL.
  ?album ont:hasSong ?song.
}
```

Figure 23 Query 5

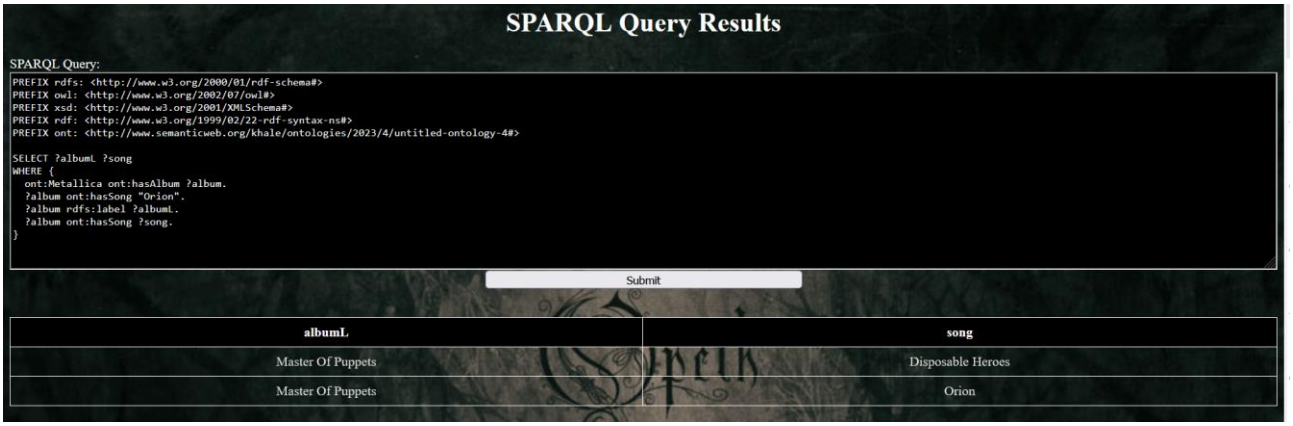


Figure 24 Query 5 output