

# Music Ontology Extended - Report

## Elaborato 2 - 3

Martina Cavallucci  
martina.cavallucci@studio.unibo.it

22 giugno 2021

### Indice

<b>Introduzione</b>	<b>3</b>
<b>1 Ontologie utilizzate</b>	<b>4</b>
1.1 Music Ontology (MO)	4
1.1.1 FOAF	4
1.1.2 Event Ontology	4
1.1.3 Timeline Ontology	4
1.1.4 FRBR	4
<b>2 Elementi di estensione ontologia</b>	<b>6</b>
2.1 Classi modellate	6
2.1.1 Tassonomia strumenti musicali	6
2.1.2 Tassonomia tipi di locazione	9
2.1.3 Classi concerto, festival, performance, tour di concerto	10
2.1.4 Classe vendor	10
2.2 Object Property modellate	11
2.3 Data Property modellate	13
<b>3 SWRL</b>	<b>15</b>
3.1 InstrumentSkill	15
3.2 WorkGenre	15
3.3 FullTicket	15
<b>4 SPARQL</b>	<b>16</b>
4.1 Prefissi utilizzati	16
4.2 QUERY 1 - Concerti con posti esauriti	16
4.3 QUERY 2 - Numero di artisti in Band	17
4.4 QUERY 3 - I biglietti di concerto più costosi del 2021 in USA	18

4.5	QUERY 4 - Il venditore che ha venduto più biglietti in Italia . . . . .	19
<b>5</b>	<b>Esempi</b>	<b>20</b>
5.0.1	Esempio Artista musicale . . . . .	20
5.1	Esempio concerto . . . . .	20
5.2	Esempio strumenti musicale . . . . .	21
<b>6</b>	<b>Conclusioni</b>	<b>22</b>
6.0.1	Vantaggi . . . . .	22
6.0.2	Sviluppi futuri . . . . .	23

## Elenco delle figure

1	Tassonomia Strumenti musicali riferimento . . . . .	7
2	Tassonomia Strumenti musicali riferimento . . . . .	8
3	Tassonomia Strumenti musicali . . . . .	9
4	Tassonomia tipi di locazione . . . . .	10
5	Classi concerto, tour di concerto, performance . . . . .	10
6	Classe Vendor . . . . .	11
7	Output - query 1 . . . . .	17
8	Output - query 2 . . . . .	17
9	Output - query 3 . . . . .	18
10	Output - query 4 . . . . .	19
11	Artista ChrisMartin . . . . .	20
12	Concerto Coldplay . . . . .	21
13	Strumento musicale chitarra . . . . .	21

## Introduzione

Il progetto **Music Ontology Extended (MOE)** ha l'obiettivo di ampliare la specifica dell'ontologia **Music Ontology (MO)** che fornisce i concetti e le proprietà principali per descrivere la musica come artisti, album, tracce sul Web Semantico. La motivazione della scelta di implementazione di tale estensione deriva dall'interesse che mostro verso questo tema e la curiosità riguardo al comprendere i concetti che ruotano intorno al tema della musica. In particolare il progetto si concentra sull'introduzione di concetti come un concerto, la gerarchia di strumenti musicali, i generi musicali e le proprietà ad esse connesse. Tale attività sarà frutto di un'implementazione che segue le principali tecnologie viste durante il corso quali: *RDF*, *RDFS*, *OWL*, *SPARQL*, *SWRL*, *SQWRL*.

# 1 Ontologie utilizzate

In questa sezione si descriveranno le principali ontologie utilizzate, specificandone classi, proprietà. Tale comprensione sarà utile per poter inserire i nuovi concetti riutilizzando quelli già esistenti nella Knowledge base.

## 1.1 Music Ontology (MO)

L'ontologia è stata creata nel 2006 e revisionata nel 2013, i suoi autori sono: Yves Raimond, Thomas Gängler, Frédéric Giasson, Kurt Jacobson, George Fazekas, Simon Reinhardt, Alexandre Passant.

Tale ontologia si basa su 4 ontologie principali:

### 1.1.1 FOAF

FOAF (acronimo di friend of a friend - Amico di un amico) è un'ontologia comprensibile dal computer atta a descrivere persone, con le loro attività e le relazioni con altre persone e oggetti. Chiunque può usare FOAF per descriversi. FOAF permette a gruppi di persone di descrivere quel fenomeno noto come social network senza la necessità di un database centralizzato.

### 1.1.2 Event Ontology

Questa ontologia è incentrata sulla nozione di evento, vista qui come il modo in cui gli agenti cognitivi classificano le regioni tempo/spazio arbitrarie.

Questa ontologia si è già dimostrata utile in un'ampia gamma di contesti, grazie alla sua semplicità e usabilità: dai discorsi in una conferenza, alla descrizione di un concerto, o agli accordi suonati in un brano Jazz (se utilizzato con l'ontologia Timeline), festival, ecc.

### 1.1.3 Timeline Ontology

Questa ontologia è incentrata sulla nozione di linea temporale, vista qui come un modo per identificare una spina dorsale temporale. Questa ontologia, utilizzata con l'ontologia Evento, può essere utilizzata per annotare sezioni di un segnale, un video o qualsiasi oggetto temporale.

### 1.1.4 FRBR

Vocabolario per descrivere opere, espressioni, manifestazioni e oggetti e le loro relazioni, come definito dai requisiti funzionali per i record bibliografici.

Il vocabolario di MO definisce 54 classi, tra le principali troviamo:

Nome	Definizione	Classe genitore	Classi figli
<b>Arrangiamento</b>	Un evento di arrangiamento	EventOntology#Event	Strumentazione, orchestrazione
<b>Artista Musicale</b>	Una persona, un gruppo di persone o un computer, il cui lavoro creativo musicale mostra sensibilità e immaginazione	foaf#Agent	Artista singolo, Gruppo Musicale
<b>Artista Singolo</b>	Singola persona il cui lavoro creativo musicale mostra sensibilità e immaginazione.	foaf#Person, Artista Musicale	
<b>Attività</b>	Un periodo di attività, che definisce quando un artista era musicalmente attivo.	EventOntology#Event	
<b>Composizione</b>	Un evento di composizione. Assume come agente il compositore stesso. Produce un lavoro musicale, o un'espressione musicale	EventOntology#Event	
<b>Espressione Musicale</b>	La realizzazione intellettuale o artistica di un'opera sotto forma di notazione alfanumerica musicale, coreografica, suono. O qualsiasi combinane di tali forme.	frbr/core#Expression	Libretto, Lyrics, Segnale, Suono
<b>Etichetta</b>	Nome commerciale di un'azienda che produce opere musicali o espressioni di opere musicali.		
<b>Evento di rilascio</b>	Un evento di rilascio, in un luogo particolare in un momento particolare.	EventOntology#Event	
<b>Festival</b>	Evento musicale/artistico della durata di diversi giorni.	EventOntology#Event	
<b>File audio</b>	Un file audio, che può essere disponibile su un file system locale o tramite http, ftp.	Medium, foaf#Document	
<b>Genere</b>	Uno stile musicale espressivo.		
<b>Gruppo Musicale</b>	Gruppo di musicisti	foaf#Group, Music Group	
<b>Lavoro musicale</b>	Distinta creazione musicale intellettuale o artistica.	frbr/core#Work	Movimento
<b>Manifestazione musicale</b>	Entità correlata all'edizione, produzione e pubblicazione di un'espressione musicale	frbr/core#Manifestation	Release, Traccia

Nome	Definizione	Classe genitore	Classi figli
<b>Movimento</b>	Un movimento è una parte autonoma di un'opera musicale.	Lavoro musicale	
<b>Orchestrazione</b>	L'orchestrazione include, oltre alla strumentazione, la gestione di gruppi di strumenti e il loro equilibrio e interazione.	Arrangiamento	
<b>Prestazione</b>	Un evento di performance, potrebbe includere come agenti i performer, ingegneri, direttori e ascoltatori.	EventOntology#Event	
<b>Segnale</b>	Una sottoclasse di espressione musicale, che rappresenta un segnale.	Espressione musicale	Segnale digitale, Segnale analogico
<b>Suono</b>	Una sottoclasse di espressione musicale, che rappresenta un suono. Realizzazione di un lavoro musicale durante una performance musicale.	Espressione musicale	
<b>Traccia</b>	Una traccia su un particolare disco	Manifestazione musicale	

## 2 Elementi di estensione ontologia

### 2.1 Classi modellate

#### 2.1.1 Tassonomia strumenti musicali

Tra le classi presenti nell'ontologia Music Ontology si sono aggiunte le sotto-classi appartenenti alle classe *Instrument*. In particolare l'ontologia principale non disponeva di sotto specificazioni che però diventano utili per descrivere lo strumento suonato dagli artisti.

Si sono aggiunte le sotto-classi con le loro relative sotto-classi come si nota dalla figura 1 e 2

- Aereophones
- Chordophones
- Electrophones
- Percussion

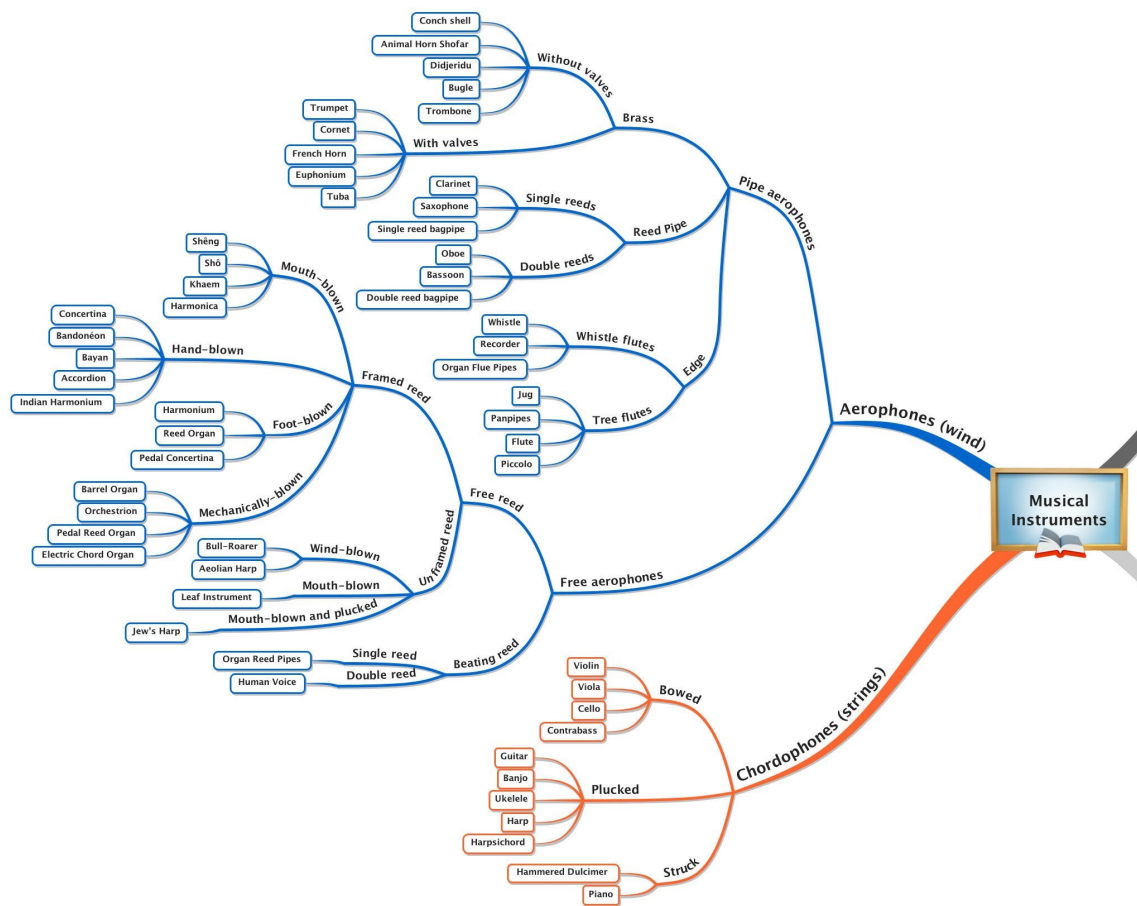


Figura 1: Tassonomia Strumenti musicali riferimento

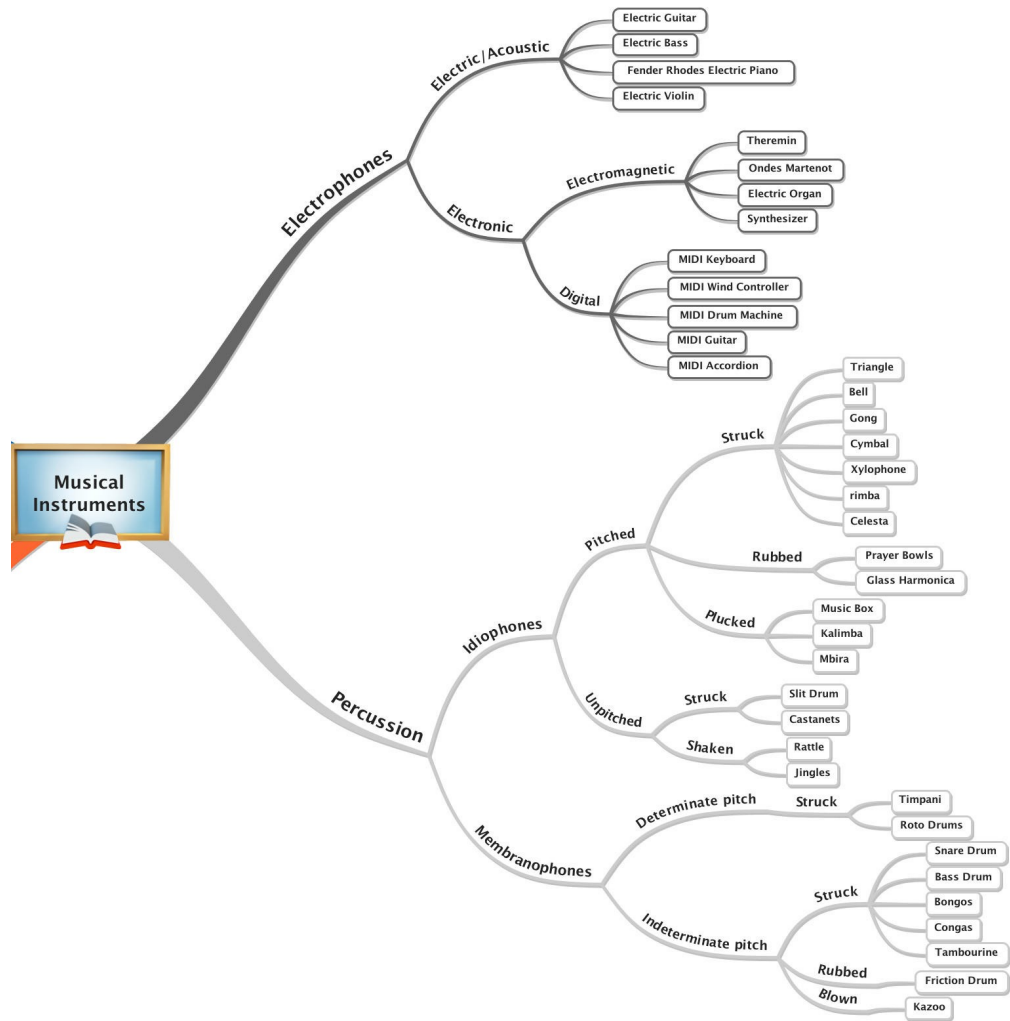


Figura 2: Tassonomia Strumenti musicali riferimento

In figura 10 si mostra uno snapshot in Protégè della gerarchia creata. Non esistono ontologie degli strumenti musicali disponibili, perciò ho pensato di integrarla con una tassonomia proposta trovata in rete.



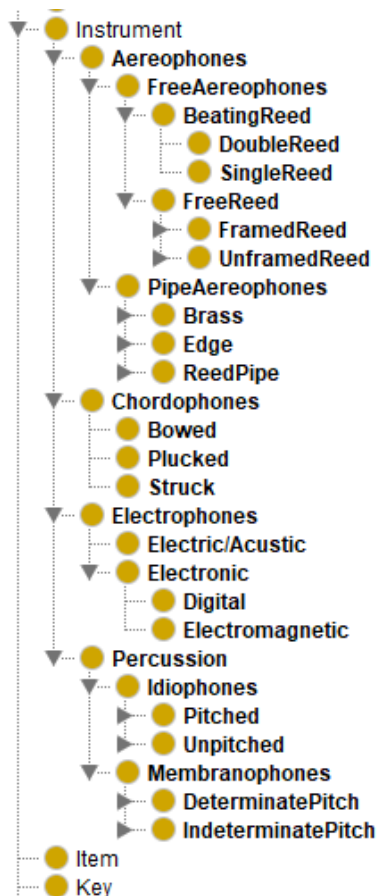


Figura 3: Tassonomia Strumenti musicali

### 2.1.2 Tassonomia tipi di locazione

All'interno dell'ontologia MO non è presente un modo di rappresentare gli spazi nella quale manifestazioni musicali vengono eseguite. Si è deciso di espandere questo concetto, esprimendo le classi di:

- Palazzo;
- Piazza;
- Parco;
- Stadio;
- Teatro.

In figura 4 si mostra uno snapshot in Protégè della gerarchia creata.

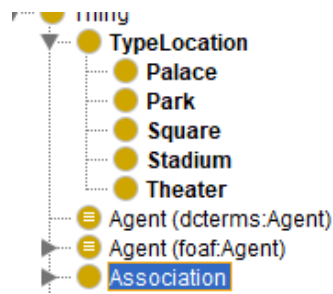


Figura 4: Tassonomia tipi di locazione

### 2.1.3 Classi concerto, festival, performance, tour di concerto

Le classi principali che sono state aggiunte all'ontologia MO, sono quelle che descrivono degli eventi musicali con un pubblico. Tutte le classi sono sotto-classi di Event. In particolare sono state aggiunte le seguenti classi:

- Concerto: evento musicale dove agiscono uno o più artisti musicali;
- Tour di concerto: insieme di concerti eseguiti da un gruppo musicale o da un artista singolo;
- Festival: evento musicale nella quale si esibiscono diversi artisti;
- Performance: evento musicale generale.

In figura 5 si mostra uno snapshot in Protégè delle classi create.

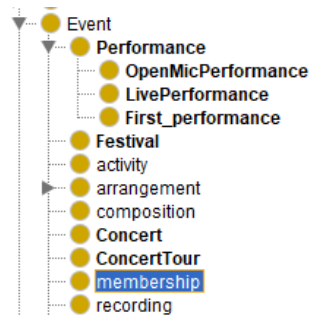


Figura 5: Classi concerto, tour di concerto, performance

### 2.1.4 Classe vendor

La classe vendor specifica l'organizzazione o la persona che si occupa di vendere i biglietti dei concerti. Essa è stata modellata come sotto-classe di foaf:Agent. In figura 6 si mostra uno snapshot in Protégè della classe creata.

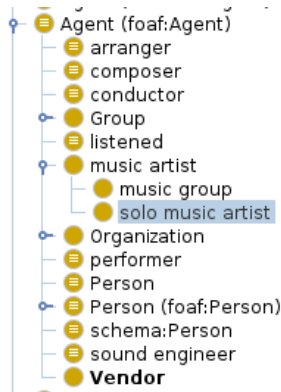


Figura 6: Classe Vendor

## 2.2 Object Property modellate

- **ConcertHomePage:**
  - Descrizione: descrive l'home page di riferimento per uno specifico concerto;
  - Dominio: Concert, ConcertTour;
  - Range: foaf:homepage
- **RefersToPerformance:**
  - Descrizione: Collega un biglietto di una performance alla performance.
  - Dominio: Ticket
  - Range: mo:Performance
  - Inverso di : RefersToTicket
- **RefersToTicket**
  - Descrizione: Collega una performance ad un biglietto.
  - Dominio: mo:Performance
  - Range: Ticket
  - Inverso di : RefersToPerformance
- **TypeLocationIn:**
  - Descrizione: Collega un determinato tipo di locazione con l'Evento.
  - Dominio: moe: TypeLocation
  - Range: mo:Event
  - Inverso di : hasTypeLocation
- **hasTypeLocation:**

- Descrizione: Collega un determinato Evento al tipo di locazione in cui si è svolto.
  - Dominio: mo:Event
  - Range: moe: TypeLocation
  - Inverso di : TypeLocationIn
- **VendorIn:**
    - Descrizione: Collega un determinato Venditore di biglietti alla performance per la quale vende biglietti.
    - Dominio: moe: Vendor
    - Range: mo:Performance
    - Inverso di : hasTicketVendor
- **hasTicketVendor:**
    - Descrizione: Collega una determinata Performance con la persona/azienda che esegue il venditori di biglietti del concerto.
    - Dominio: mo:Performance
    - Range: moe: Vendor
    - Inverso di : VendorIn
- **time\_Event:**
    - Descrizione: Collega un determinato evento con il suo istante temporale nella quale esso ha inizio.
    - Dominio: Event
    - Range: time:TemporalEntity
    - Equivalente di: mo:time
- **hasGenre:**
    - Descrizione: Collega una determinata Performance ad un genere musicale alla quale si riferisce.
    - Dominio: mo:Performance
    - Range: mo:Genre
    - Inverso di: GenreOf
- **GenreOf:**
    - Descrizione: Collega un determinato genere alla Performance della quale si vuole specificare il genere.
    - Dominio: mo:Genre

- Range: mo:Performance
- Inverso di: hasGenre
- **hasAvailableTicket:**
  - Descrizione: Specifica che una determinata Performance non ha ancora raggiunto il limite di posti disponibili.
  - Dominio: mo:Performance
  - Inverso di: hasFullTicket
- **hasFullTicket:**
  - Descrizione: Specifica che una determinata Performance ha ancora raggiunto il limite di posti disponibili.
  - Dominio: mo:Performance
  - Inverso di: hasAvailableTicket
- **TicketOfConcert:**
  - Descrizione: Collega uno specifico ticket al concerto per la quale si riferisce.
  - Dominio: mo:Ticket
  - Range: mo:Concert
- **CompetenceRegion:**
  - Descrizione: Collega uno specifico venditore di ticket con la sua area geografica di vendita.
  - Dominio: mo:Vendor
  - Range: event:place

## 2.3 Data Property modellate

- **ConcertName:**
  - Descrizione: Specifica il nome di un determinato concerto.
  - Dominio: mo:Concert
  - Range: rdfs:Literal
- **ConcertTourName:**
  - Descrizione: Specifica il nome di un determinato tour concerto.
  - Dominio: mo:ConcertTour
  - Range: rdfs:Literal
- **MaxTicket:**
  - Descrizione: Specifica il numero massimo di posti prenotabili in un concerto.

- Dominio: mo:Performance
- Range: xsd:int
- **NumberSeat:**
  - Descrizione: Specifica il numero di posti fino a quel momento prenotati. Non possono superare MaxTicket.
  - Dominio: mo:Performance
  - Range: xsd:int
- **PerformanceName:**
  - Descrizione: Specifica il nome di una performance.
  - Dominio: moe:FirstPerformance, moe:LivePerformance, moe:OpenMicPerformance
  - Range: rdfs:Literal
- **TicketNumber:**
  - Descrizione: Specifica il numero/identificativo di un biglietto.
  - Dominio: moe:Ticket
  - Range: rdfs:Literal
- **TicketPrice:**
  - Descrizione: Specifica il costo di un biglietto.
  - Dominio: moe:Ticket
  - Range: rdfs:Literal
- **TicketSold:**
  - Descrizione: Specifica il numero di biglietti venduti da un Venditore.
  - Dominio: moe:Vendor
  - Range: xsd:int

## 3 SWRL

Per permettere al reasoner di inferire maggiori informazioni riguardanti l'ontologia sono state realizzate alcune regole SWRL.

### 3.1 InstrumentSkill

```
1 instrument(?perf, ?inst) ^ performed(?ag, ?perf) ->  
   hasInstrumentSkill(?ag, ?inst)
```

Questa regola serve per inferire se in una determinata Performance (concerto, festival, etc.) si è suonato un certo strumento musicale e un determinato Agent (Persona, gruppo musicale) si è esibito in tale Performance allora significa che hanno competenze sullo strumento utilizzato. Nel caso in cui la regola fosse vera allora un determinato Agent verrà descritto con un object property *hasInstrumentSkill* indicando come oggetto lo strumento in questione.

### 3.2 WorkGenre

```
1 hasGenre(?perf, ?gen) ^ performed(?ag, ?perf) -> WorkGenre(?ag, ?gen)
```

Questa regola serve per inferire se una determinata Performance (concerto, festival, etc) ha un determinato genere musicale e un determinato Agent (Persona, gruppo musicale) si è esibito in tale Performance allora significa che l'Agent segue un gruppo musicale per la quale lavora. Nel caso in cui la regola fosse vera allora un determinato Agent verrà descritto con un object property *WorkGenre* indicando come oggetto il genere in questione.

### 3.3 FullTicket

```
1 Event(?perf) ^ NumberSeat(?perf, ?seat) ^ MaxTicket(?perf, ?seat) ->  
   hasFullTicket(?perf, ?perf)
```

Questa regola serve per inferire se una determinata Performance (concerto, festival, etc) ha un numero di posti disponibili uguale al numero massimo di posti disponibili allora tale Performance è al completo. Nel caso in cui la regola fosse vera allora una determinata Performance verrà descritta con un object property *hasFullTicket*.

## 4 SPARQL

SPARQL (Simple Protocol and RDF Query Language) è un linguaggio di interrogazione per RDF. SPARQL trova i risultati dell'interrogazione attraverso il pattern matching all'interno del grafo che deve soddisfare ciò che è stato specificato nella clausola WHERE della query. Le parole chiave di SPARQL sono: SELECT, FROM, WHERE, GROUP BY, LIMIT, OPTIONAL, FILTER, ORDER BY.

### 4.1 Prefissi utilizzati

I prefissi utilizzati per svolgere le query sono i seguenti:

```
1 PREFIX owl: <http://www.w3.org/2002/07/owl#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 PREFIX moe: <http://www.semanticweb.org/martina/ontologies/2021/5/moe#>
5 PREFIX mo: <http://purl.org/ontology/mo/>
6 PREFIX time: <http://www.w3.org/2006/time#>
7 PREFIX event: <http://purl.org/NET/c4dm/event.owl#>
```

### 4.2 QUERY 1 - Concerti con posti esauriti

*Concerti per la quale sono terminati i biglietti in Italia nel mese di Maggio 2021 in ordine decrescente per il numero di posti disponibili*

Tale query permette di capire quali sono i Concerti che hanno riscontrato più successo. Viene utilizzata la SELECT il cui scopo è quello di selezionare i campi che si vogliono mostrare. Nella clausola WHERE vengono filtrati i concerti nel lasso temporale selezionato e nel luogo stabilito. Mentre nell'ORDER BY viene deciso il modo in cui i risultati vengono ordinati.

```
1 SELECT ?concert ?numSeat WHERE {
2     ?concert rdf:type moe:Concert.
3     ?concert moe:NumberSeat ?numSeat.
4     ?concert moe:hasFullTicket ?concert.
5     ?concert event:place ?place.
6     FILTER(?place = moe:Italy).
7     ?concert moe:time_Event ?month.
8     FILTER(?month = moe:May2021).
9 }
10 ORDER BY DESC (?numSeat)
```



Output:

?concert	?numSeat
moe:ConcertoGianna	2000
moe:ConcertoLigabue	1000
moe:ColdplayInItaly	500

Figura 7: Output - query 1

### 4.3 QUERY 2 - Numero di artisti in Band

*Band con il loro numero di artisti in ordine crescente*

Tale query permette di capire quali sono le band con il maggior numero di artisti. Viene utilizzata la SELECT il cui scopo è quello di selezionare i campi che si vogliono mostrare. Nella clausola WHERE vengono filtrati i membri di un particolare gruppo musicale. Il risultato viene raggruppato per gruppo musicale sulla quale con la clausola COUNT si contano i membri. Infine si ordina in ordine crescente il risultato in base al numero di membri.

```

1 SELECT ?group (COUNT(?member) as ?members)
2 WHERE {
3     ?group rdf:type mo:MusicGroup.
4     ?member mo:member_of ?group .
5 }
6 GROUP BY ?group
7 ORDER BY ASC (?members)

```

Output:

?group	?members
moe:TheDoors	4
moe:Coldplay	4
moe:GreenDay	3
moe:ThirtySecondsToMars	2
moe:Nirvana	2

Figura 8: Output - query 2

#### 4.4 QUERY 3 - I biglietti di concerto più costosi del 2021 in USA

*Mostrare i biglietti con il relativo prezzo che sono i più costosi (soglia maggiore di 300 dollari) venduti negli Stati Uniti nel 2021*

In tale query si fa uso della clausola FILTER per controllare il prezzo dei biglietti, l'area di vendita e l'anno. E' necessario controllare che un certo concerto si riferisca ad un certo ticket senno non è possibile reperire il nome del concerto al quale fa riferimento.

```
1 SELECT ?price ?ticket ?concert
2 WHERE {
3     ?concert rdf:type moe:Concert.
4     ?ticket rdf:type moe:Ticket.
5     ?ticket moe:TicketPrice ?price.
6     ?concert moe:RefersToTicket ?ticket.
7     FILTER(?price > 300).
8     ?concert event:place ?place.
9     FILTER(?place = moe:USA).
10    ?concert moe:time_Event ?year.
11    FILTER(?year = moe:2021).
12
13 }
14 ORDER BY DESC (?price)
```

Output:

?price	?ticket	?concert
725	moe:TSTMTicket	moe:ConcertTSTM
560	moe:NirvanaTicketConcert	moe:NirvanaConcert2021

Figura 9: Output - query 3

## 4.5 QUERY 4 - Il venditore che ha venduto più biglietti in Italia

*Restituire il venditore che ha venduto più biglietti in Italia*

In tale query si è fatto uso della clausola MAX per poter restituire solamente il numero massimo di ticket vendut, inoltre si è filtrato per area geografica.

```
1 SELECT ?vendor (Max(?ticketSold) as ?SoldTicket)
2 WHERE {
3     ?vendor rdf:type moe:Vendor.
4     ?vendor moe:CompetenceRegion ?region
5     FILTER(?region = moe:Italy).
6     ?vendor moe:TicketSold ?ticketSold
7 }
8 GROUP BY ?vendor
```

Output:

?vendor	?SoldTicket
moe:TuttoConcerti	50000.0

Figura 10: Output - query 4

## 5 Esempi

Per lo svolgimento del progetto si è utilizzato il reasoner Pellet che ha permesso di verificare le inferenze.

### 5.0.1 Esempio Artista musicale

Come si nota in Figura 11 se un determinato Artista partecipa ad una Performance con un certo strumento musicale allora viene inferito che esso ha skill in quello specifico strumento musicale. Inoltre se si è esibito in Performance con un certo genere musicale anche da tale informazione che esso lavora nel genere musicale di riferimento di quella performance.

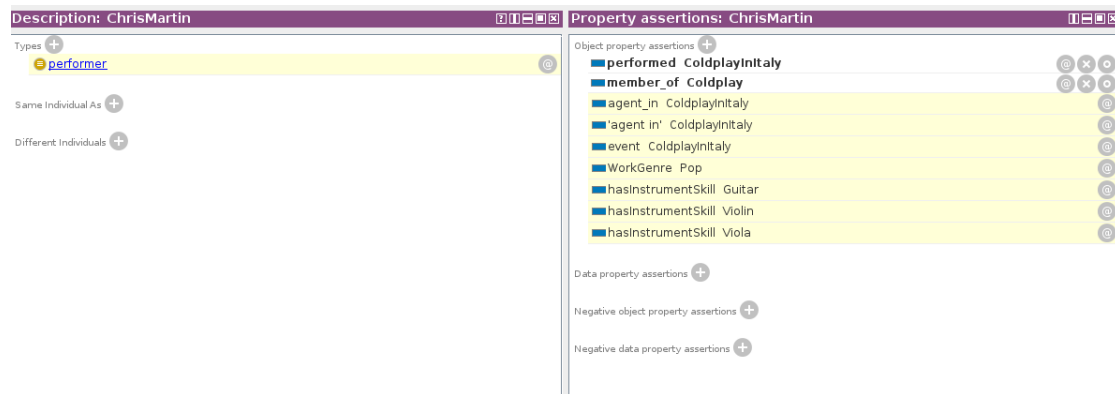


Figura 11: Artista ChrisMartin

### 5.1 Esempio concerto

Come si nota in Figura 12 se vengono specificati il numero massimo di biglietti disponibili e quelli attualmente prenotati, in base alla regola definita in SWRL allora tale concerto ha i posti esauriti.

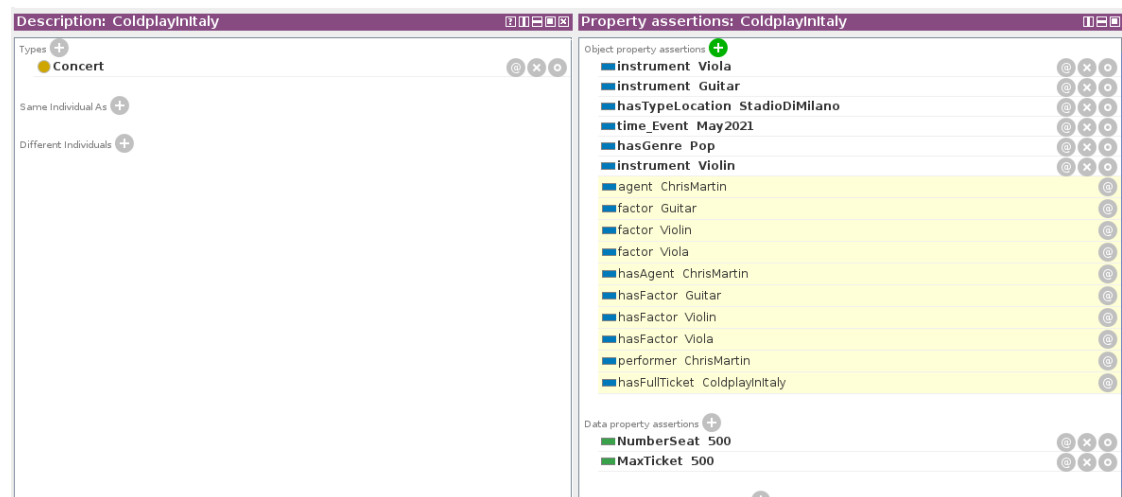


Figura 12: Concerto Coldplay

## 5.2 Esempio strumenti musicale

Come si nota in figura 13 se viene specificato che un certo strumento musicale è stato suonato in una Performance (Concerto) allora esso diventa uno degli strumenti passivi della quale si è fatto uso. Per ogni strumento musicale abbiamo tutti gli eventi alla quale è referenziato.

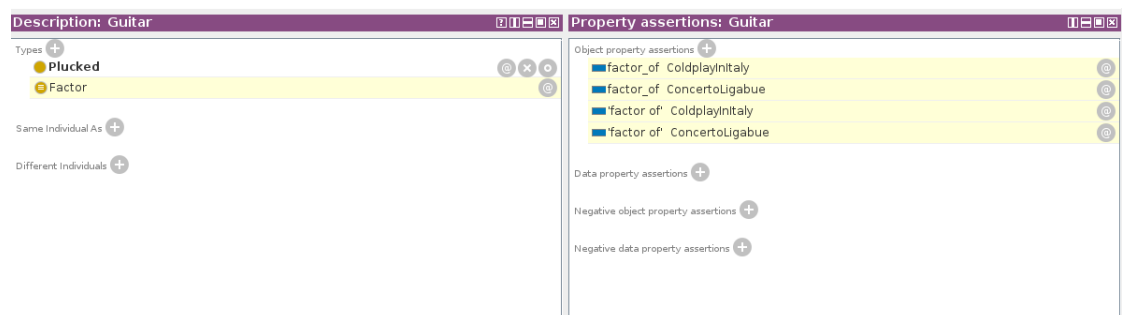


Figura 13: Strumento musicale chitarra

## 6 Conclusioni

L'obiettivo del progetto e della presente relazione era quello di esporre le conoscenze nell'ambito del Web Semantico. In particolare il progetto mira a esercitarsi sulla conoscenza acquisita in tali tecnologie e linguaggi:

- RDF : descrivere triple in RDF.
- RDFS/OWL : descrivere attraverso RDFS e OWL le caratteristiche di un dominio d'interesse prendendo ontologie già esistenti e aggiungendo a quest'ultime nuove informazioni.
- SPARQL : realizzare interrogazioni in SPARQL per estrapolare informazioni riguardo l'ontologia realizzata.
- SWRL : realizzare regole SWRL che aggiungano ulteriori informazioni non definibili attraverso OWL.

Il lavoro svolto sull'ontologia della musica mi è stato d'aiuto per capire un dominio di cui non avevo la conoscenza completa, e mi ha data la possibilità di espanderlo con elementi essenziali che non erano compresi all'interno di MO.

### 6.0.1 Vantaggi

Il vantaggio principale di tale ontologia è quello di poter collegare informazioni che prima erano difficilmente collegabili. In particolare con tale ontologia è possibile:

- Collegare Artisti ai Concerti che eseguono
- Collegare Artisti agli strumenti che sono in grado di suonare
- Collegare Concerti con il loro Venditore
- Collegare Artisti con i biglietti dei loro concerto, inferendo informazioni sul prezzo e dei posti disponibili.
- Collegare i Venditori di biglietti con i Ticket
- Controllare se un concerto ha i biglietti esauriti o meno

Tale informazioni derivano dai vantaggi che il Web Semantico offre in particolare l'integrazione tra informazioni provenienti da più fonti, la possibilità di ridurre l'ambiguità semantica, identificare le informazioni davvero rilevanti.

Tali risultati vengono raggiunti grazie a proprietà caratterizzanti degli strumenti del web semantico come:

- facilità di integrazione di diversi domini: tramite l'importazione di ontologie e la creazioni di relazione tra esse.

- controllo sulle inconsistenze: il reasoner infatti grazie ad una descrizione del dominio dettagliata ricava in maniera automatica le eventuali inconsistenze della'ontologia controllando proprietà, domini, range e classi.
- inferenza di nuove informazioni: le tecnologie del Web semantico sono in grado di creare nuove informazioni tramite deduzioni logiche.

### **6.0.2 Sviluppi futuri**

Una possibile ulteriore estensione dell'ontologia potrebbe essere quella di collegare gli Artisti con i libri in cui vengono citati, oppure di cui se ne fa una biografia.