

# Musico Update

---

## General Description

In order to creating information to be suggested to the end user, the entities that have been added and modified on the Musico ontology aim, primarily, to augment the information between concepts revolving around the figure of the musician. The basic idea is to increase the amount of data that can later be related to the musico:Artist class. In this way, relationships have been created that relate previously disconnected concepts in a way that further enriches the knowledge graph. With these new relationships, it is possible to do automatic reasoning to infer new information that can be suggested to the musician to improve his experience. This information can be used to suggest new music genres or instrument to play, new people to meet, or new venues to perform in.

## Classes

- **ADDED** - *Artist* :
  - Class used to represent an artist, both HumanMusician and MusiciansGroup

## Object Properties

- **ADDED** - *played\_genre*:
  - Used to indicate what genre is played in a particular MusicVenue
    - *Domain*: schema:MusicVenue
    - *Range*: mo:Genre
- **ADDED** - *popular\_mood*:
  - Used to indicate the emotion aroused most by a MusicalWork, a Genre, or at a particular time of day
    - *Domain*: musico:MusicalWork mo:Genre schema:Time
    - *Range*: obo:MFOEM\_000193
- **ADDED** - *suggested\_user*:
  - Used to indicate if two people should get to know each other
    - *Domain*: musico:Artist
    - *Range*: musico:Artist
- **ADDED** - *used\_to\_play*:
  - Used to indicate what genre of music was played with a certain instrument
    - *Domain*: mo:Instrument
    - *Range*: mo:Genre
- **MODIFIED** - *musico:plays\_genre*:
  - Changed the domain with the addition of the Artist class
    - *Domain*: musico:Artist

- *Range*: mo:Genre

## Data Properties

- **ADDED** *foundation\_day*:
  - Property representing the founding date of a musical group
    - *Domain*: musico:MusiciansGroup
    - *Range*: xsd:date

## Rules

Listed below are rules constructed to infer new information using the new relationships to populate the GraphDB by Ontotext database.

```
Id: workMood:
  p <rdf:type> <musico:MusicianParticipation>
  p <musico:played_musical_work> w
  p <musico:expressed_mood> m
  -----
  w <musico:popular_mood> m
```

```
Id: genreMood:
  p <rdf:type> <mo:MusicalWork>
  w <mo:genre> g
  w <musico:popular_mood> m
  -----
  g <musico:popular_mood> m
```

```
Id: instrGenre
  p <rdf:type> <musico:MusicianParticipation>
  p <musico:played_instrument> i
  p <musico:played_musical_work> w
  w <mo:genre> g
  -----
  i <musico:used_to_play> g
```

```
Id: musicianGroupGenre
  a <musico:has_group> mg
  a <musicoo:plays_genre> g
  -----
  mg <musicoo:plays_genre> g
```

```
Id: venueGenre
  p <rdf:type> <musico:MusicianParticipation>
  p <musico:involved_event> e
  s <owl:sameAs> e
  s <schema:address> v
  p <musico:played_musical_work> w
  w <mo:genre> g
  -----
  v <musico:played_genre> g
```

```
Id: suggestedUser
  u <rdf:type> <musicoo:HumanMusician>
  o <rdf:type> <musicoo:HumanMusician>
  u <foaf:based_near> c
  o <foaf:based_near> c
  u <musicoo:plays_genre> g
  o <musicoo:plays_genre> gg
  gg <mo:similar_to> g [Constraint g != gg]
  u p o [Constraint p != <owl:sameAs>,
        p != <foaf:knows>,
        p !=< musico:suggested_user>]
-----
  u <musico:suggested_user> o [Constraint u != o]
```