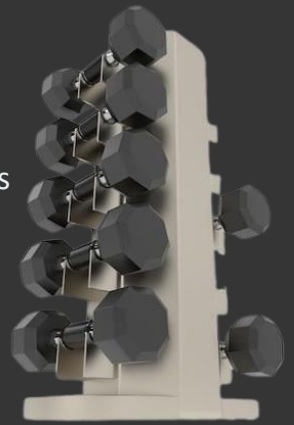
Gym Ontology with Protegé

Carmella Sta Ana Pana - 1794019
Engineering in Computer Science
Sapienza University of Rome

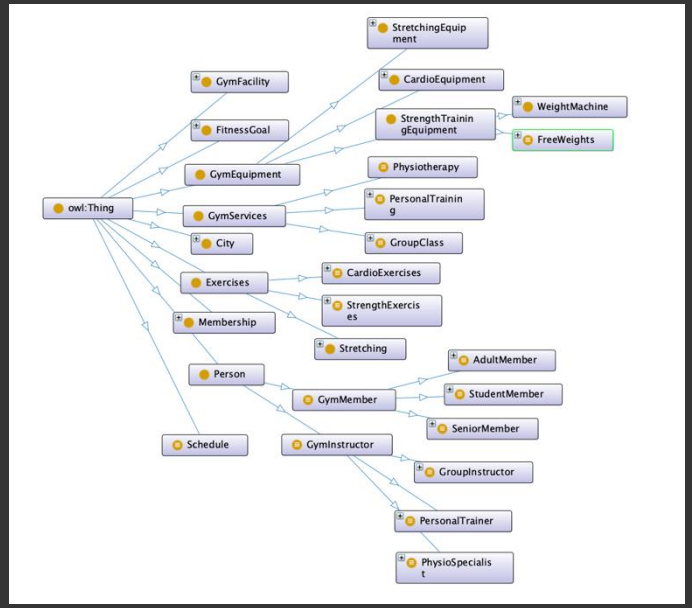
The ontology

- The purpose of this project is to model a simple ontology about gym-related entities, relationships and services.
- Built and developed in Protegé, a common ontology editor useful for creating ontologies in structured data formats.
- By capturing all relevant information, the ontology enables data-driven decisions regarding gym-related operations.



Structure: Classes

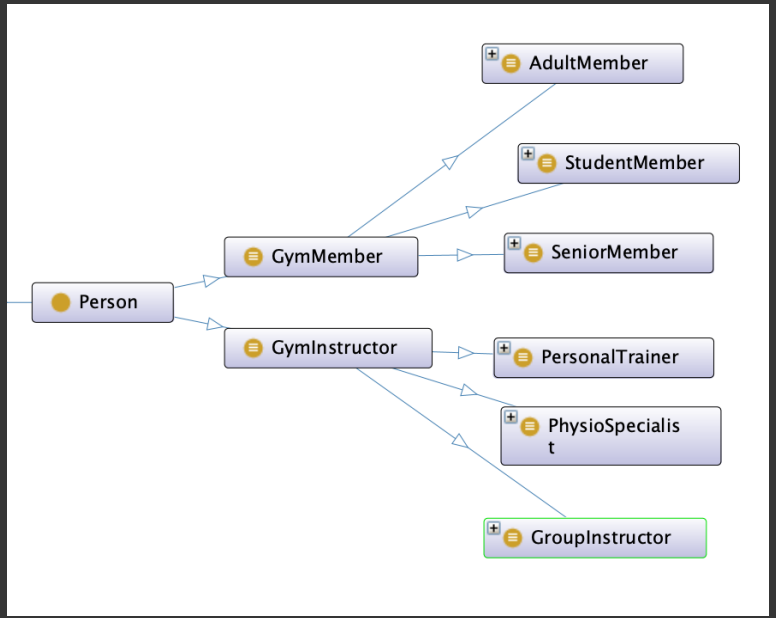
Metrics	
Axiom	2,028
Logical axiom count	1,657
Declaration axioms count	371
Class count	28
Object property count	25
Data property count	16
Individual count	303
Annotation Property count	0



- 27 classes have been defined, that encapsule the main aspects of gym-related concepts.
- Some of the main classes, which most are subclasses of *owl:Thing* (fundamental class that represents the most general concept) are **GymFacility**, **GymServices**, **Exercises**, **Person** and its subclasses (**GymInstructor** and **GymMember**) etc.

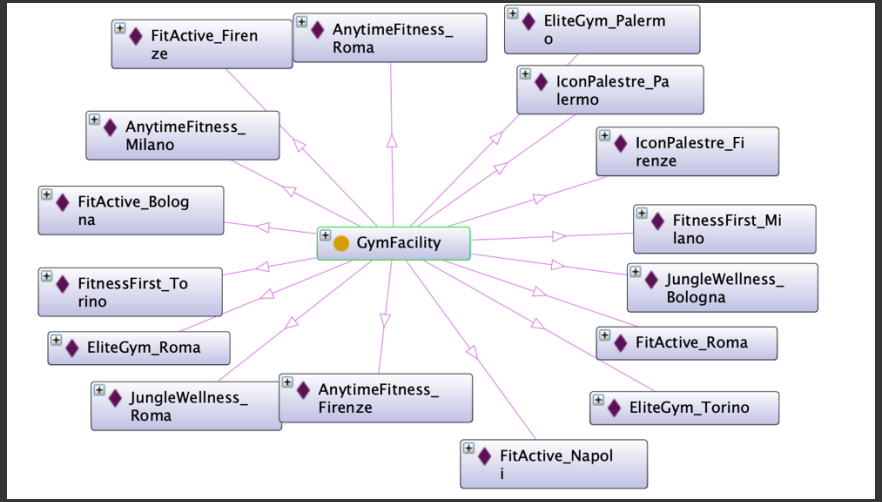
Classes: Person

- The **Person** entity represents two kinds of people that can be associated to a gym facility:
- its **GymInstructors,** which are the people that work at the facility, divided into 3 different categories (PersonalTrainer, PhysioSpecialist, GroupInstructor);
- and also its **GymMembers**, in other words the people who are enrolled in the various gym facilities, and are divided into subclasses based on the members' age (AdultMember, SeniorMember, StudentMember).



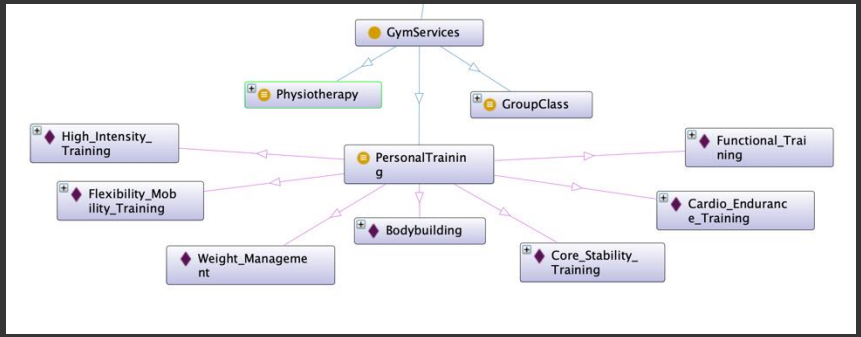
Classes: GymFacility

- The **GymFacility** is a representation of the gym structures and facilities.
- It is associated to the classes City (which represents the location of each instance), Person (the people related to the facility) and Membership (different types of membership a facility can offer) through object properties



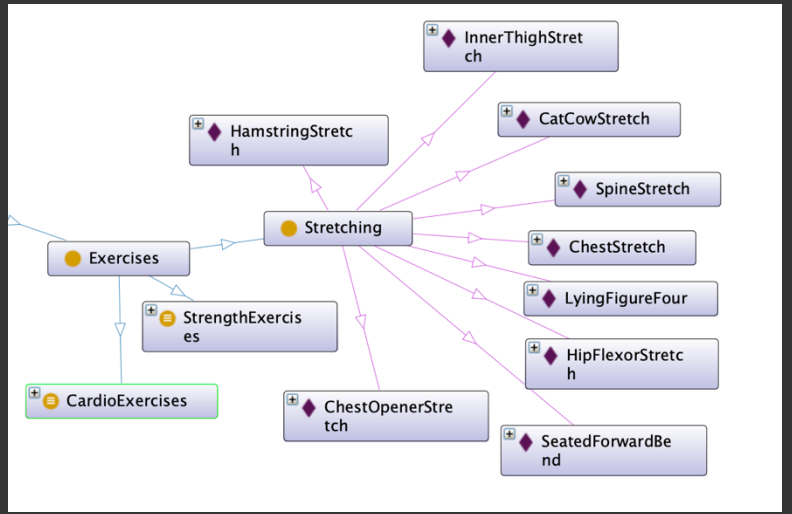
Classes: GymServices

- The **GymServices** class represent possible types of services that a gym facility can offer to its members.
- In this case it is restricted to the kinds of instructors we've seen before, so the possible services provided are:
  Physiotherapy
  PersonalTraining
  GroupClass
- Each of these has their own instances that specify more specifically the service or training offered



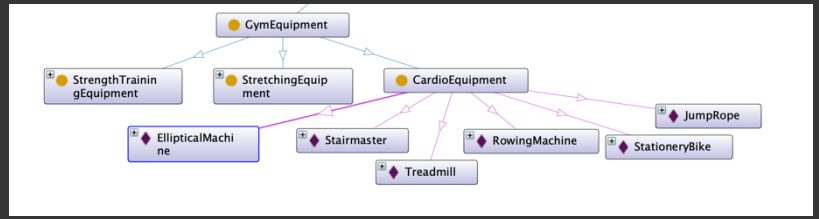
Classes: Exercises

- A person going to the gym must have a set of exercises to execute in order to achieve particular fitness goals
- This is encapsulated through an **Exercises** entity
- For simplicity, I distinguished them only between 3 kinds: CardioExercises, StrengthExercises and Stretching

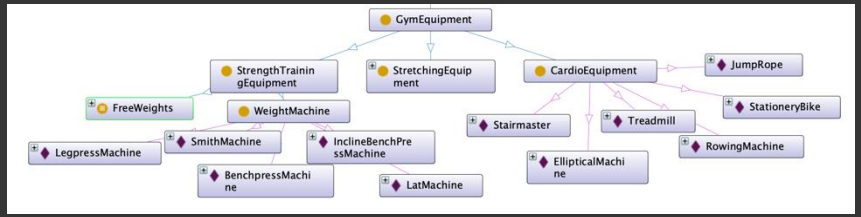


Classes: GymEquipment

- The **GymEquipment** class represents a distinction between the variety of machines or tools that can be found in a gym facility and can be used to perform some of the Exercises

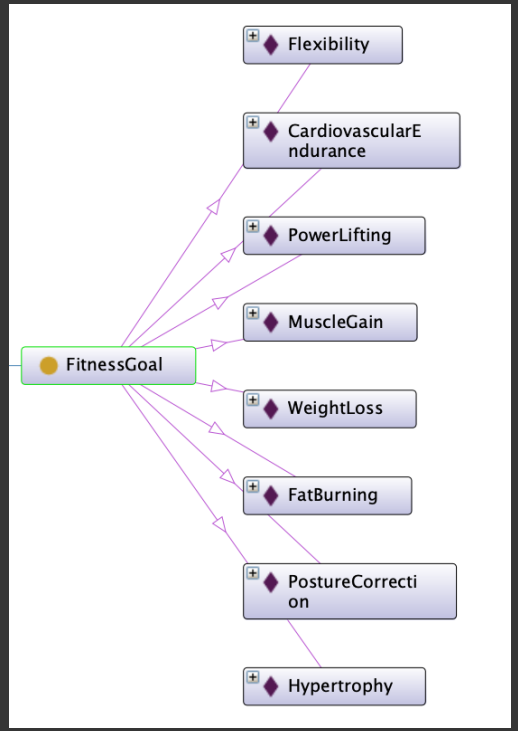


- Again for simplicity, the instances are categorized into three main subclasses only:
  CardioEquipment
  StretchingEquipment
  StrengthTrainingEquipment

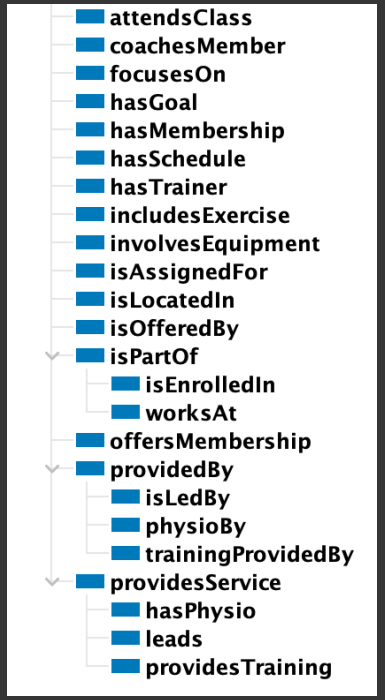


Classes: FitnessGoal

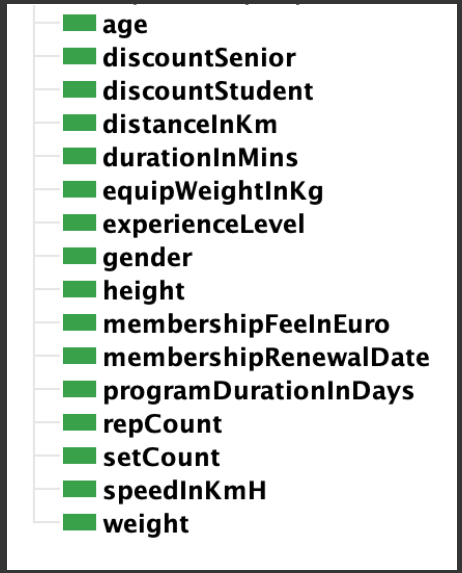
- The **FitnessGoal** class represents the goals that either the exercises focus on or that a gym member that are enrolled in gyms have.
- Some of the individuals defined in this class are WeightLoss, MuscleGain, PostureCorrection etc.



Object and Data Properties

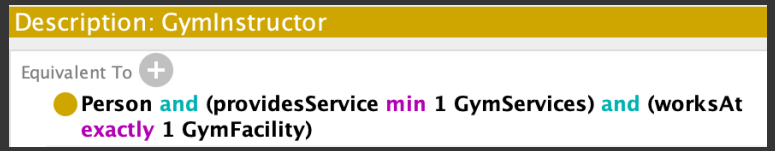


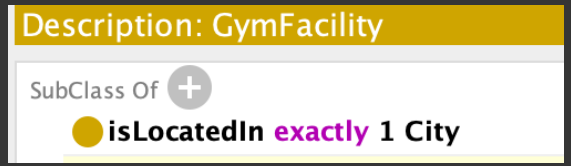
- The object properties listed (on the left) are the main relationships that exist between the defined classes
- They help represent different types of gym-related scenarios
- For example, involvesEquipment is a relationship between Exercises and GymEquipment
- On the right instead we have the data properties which are the attributes that provide more information about the classes.



Class axioms

Some of the class axioms described in the ontology



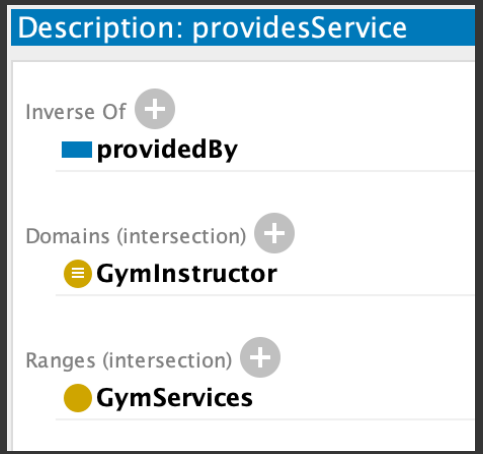


- Exercises ≡ CardioExercises ⊔ StrengthExercises ⊔ Stretching
  Exercises ⊑ ∃focusesOn.FitnessGoal
  Exercises ⊑ ∃involvesEquipment.FitnessGoal
- CardioExercises ⊑ Exercises
  CardioExercises ⊑ ∃distanceInKm.xsd:decimal ⊓ ∃speedInKmH.xsd:decimal
  CardioExercises ⊑ focusesOn.CardiovascularEndurance ⊓ focusesOn.FatBurning ⊓ focusesOn.WeightLoss
  CardioExercises ⊑ ∃involvesEquipment.CardioEquipment
- GymMember ⊑ Person
  GymMember ≡ Person ⊓ (∃hasMembership.Membership) ⊓ (∃isEnrolledIn.GymFacility)
- PersonalTrainer ⊑ GymInstructor
  PersonalTrainer ≡ GymInstructor ⊓ (∃providesTraining.PersonalTraining)
- AdultMember ⊑ GymMember
  AdultMember ≡ GymMember ⊓ (18 ≤ age < 60)
- GymFacility⊑≥1 isLocatedIn.City ⊓ ≤1 isLocatedIn.City

[…]

Object property axioms

Some examples of the object properties' axioms



- focusesOn:
  Domain(focusesOn)=Exercises
  Range(focusesOn)=FitnessGoal

- offersMembership ≡ isOfferedBy $^{-}$
  Domain(offersMembership) = GymFacility
  Range(offersMembership) = Membership

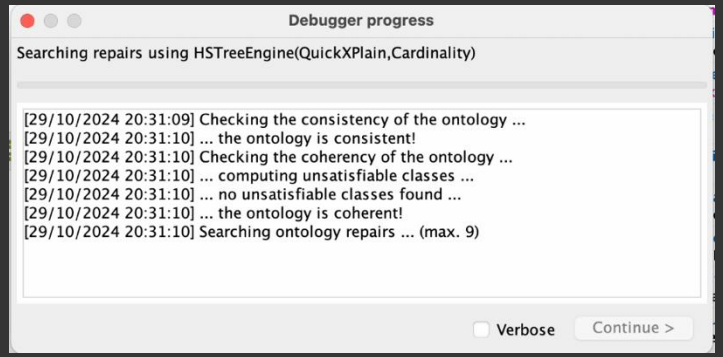
- isPartOf:
  Domain(isPartOf) = Person
  Range(isPartOf) = GymFacility

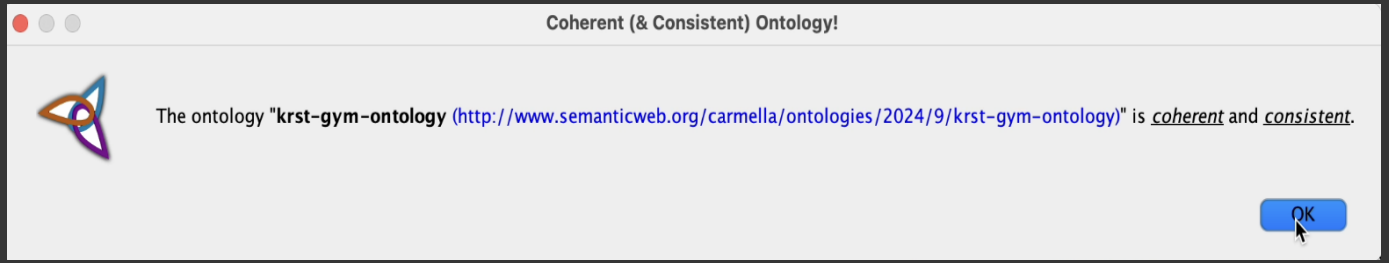
- providedBy:
  providedBy ≡ providesService^{-}
  Domain(providedBy) = GymService
  Range(providedBy) = GymInstructor

[…]

Consistency and Coherency

By using Protegé's "Ontology Debugger" plug-in, the built ontology is proven to be consistent and coherent.





SPARQL and DL queries

- A few SPARQL queries have been written to explore and retrieve data from the ontology.
- There are two sets of queries, each work on a version of the ontology:

  - the first set queries the version of the ontology with only explicitly defined data
  - the other set of queries on the other hand works on the ontology that is enriched with inferred data: the inferences are obtained by using the Protegé's "Pellet Reasoner" plugin, and then all is saved in a separate ontology

- The DL queries over the ontology have been written with the help of the DL query plug-in

(The next slides show only a few of each of these sets of queries. To view all the queries written, open file "queries.pdf".)

SPARQL queries: no inferences

- **Find all the group instructors in the city of Torino who offer Dance Classes.**

```
SELECT ?instructor ?gym

WHERE {
        ?instructor rdf:type c:GroupInstructor .
        ?instructor c:worksAt ?gym .
        ?gym rdf:type c:GymFacility .
        ?gym c:isLocatedIn c:Torino .
        ?instructor c:leads c:DanceClass .

}
```

instructor	gym
SerenaGrasso	FitnessFirst_Torino

SPARQL queries: no inferences

- **Find the gyms in Torino or Bologna that offers a Student Discount membership greater than 10%.**

```
SELECT  DISTINCT ?facility (STR(?discount) as ?discounts)  ?membership
WHERE {
   ?facility rdf:type ?GymFacility .
   ?facility c:offersMembership ?membership .
   ?membership c:discountStudent ?discount .
   ?facility c:isLocatedIn ?city .
   FILTER(?discount > 10) .
   FILTER (?city = c:Bologna || ?city = c:Torino) .
}
```

facility	discounts	membership
JungleWellness_Bologna	"12"	Monthly_JW_Bologna
JungleWellness_Bologna	"30"	Annual_JW_Bologna
EliteGym_Torino	"15"	Monthly_EG_Torino

SPARQL queries: no inferences

- **Calculate the final membership fees for all Senior Members.**

```
SELECT ?seniorMember (STR(?membershipFee) AS ?membershipFeeStr) (STR(?discountSenior) AS
?discountSeniorStr) (STR((?membershipFee - ((?discountSenior / 100) * ?membershipFee))) AS ?finalFeeStr)
WHERE {
  ?seniorMember rdf:type c:SeniorMember .
  ?seniorMember c:hasMembership ?membership .

  ?membership c:membershipFeeInEuro ?membershipFee .
  ?membership c:discountSenior ?discountSenior .
}
```

seniorMember	membershipFeeStr	discountSeniorStr	finalFeeStr
BarbaraGalli	"500.0"	"20"	"400.00"
StefaniaErcole	"500.0"	"20"	"400.00"
ElenaDeSantis	"400.0"	"10"	"360.00"
VincenzoMorelli	"220.0"	"21"	"173.800"
GiorgiaFontana	"610.0"	"24"	"463.600"
AntonioRusso	"70.0"	"40"	"42.00"
LeonardoGreco	"140.0"	"15"	"119.000"
NicolaMoretti	"55.0"	"20"	"44.00"
RenatoEsposito	"70.0"	"30"	"49.00"
TizianaRusso	"50.0"	"15"	"42.500"

SPARQL queries: no inferences

- **List all CardioExercises involving CardioEquipment with a distanceInKm greater than 5 km.**

```
SELECT ?exercise ?equipment (str(?distance) as ?dist)
WHERE {
   ?exercise rdf:type c:CardioExercises .
   ?exercise c:involvesEquipment ?equipment .
   ?exercise c:distanceInKm ?distance .
   ?equipment rdf:type c:CardioEquipment .
   FILTER(?distance > 5.0)
}
```

exercise	equipment	dist
Cycling_20_15	StationeryBike	"15.0"
Rowing_15_8	RowingMachine	"8.0"

SPARQL queries: with inferences

- **Find the exercises that involve less than 3 equipments.**

```
SELECT ?exercise (STR(COUNT(?equipment)) AS ?equipmentCount) ?exerciseType
WHERE {
    ?exercise rdf:type ?exerciseType .
    ?exerciseType rdfs:subClassOf c:Exercises .
    ?exercise c:involvesEquipment ?equipment .
}
GROUP BY ?exercise ?exerciseType
HAVING (COUNT(?equipment) <3 )
```

exercise	equipmentCount	exerciseType
SpineStretch	"1"	Stretching
HamstringStretch	"1"	Stretching
JumpRope_2	"1"	CardioExercises
StairClimbing_3_0.2	"1"	CardioExercises
Running_10_5	"1"	CardioExercises
Cycling_20_15	"1"	CardioExercises
Rowing_15_8	"1"	CardioExercises
Elliptical_8_4	"1"	CardioExercises
Rowing_2_2	"1"	CardioExercises
OverheadPress_3x8	"2"	StrengthExercises
Deadlift_3x5	"2"	StrengthExercises
Squat_5x5	"1"	StrengthExercises

SPARQL queries: with inferences

- **Find the exercises that involve less than 3 equipments.**

```
SELECT ?exercise (STR(COUNT(?equipment)) AS ?equipmentCount) ?exerciseType
WHERE {
    ?exercise rdf:type ?exerciseType .
    ?exerciseType rdfs:subClassOf c:Exercises .
    ?exercise c:involvesEquipment ?equipment .
}
GROUP BY ?exercise ?exerciseType
HAVING (COUNT(?equipment) <3 )
```

exercise	equipmentCount	exerciseType
SpineStretch	"1"	Stretching
HamstringStretch	"1"	Stretching
JumpRope_2	"1"	CardioExercises
StairClimbing_3_0.2	"1"	CardioExercises
Running_10_5	"1"	CardioExercises
Cycling_20_15	"1"	CardioExercises
Rowing_15_8	"1"	CardioExercises
Elliptical_8_4	"1"	CardioExercises
Rowing_2_2	"1"	CardioExercises
OverheadPress_3x8	"2"	StrengthExercises
Deadlift_3x5	"2"	StrengthExercises
Squat_5x5	"1"	StrengthExercises

SPARQL queries: with inferences

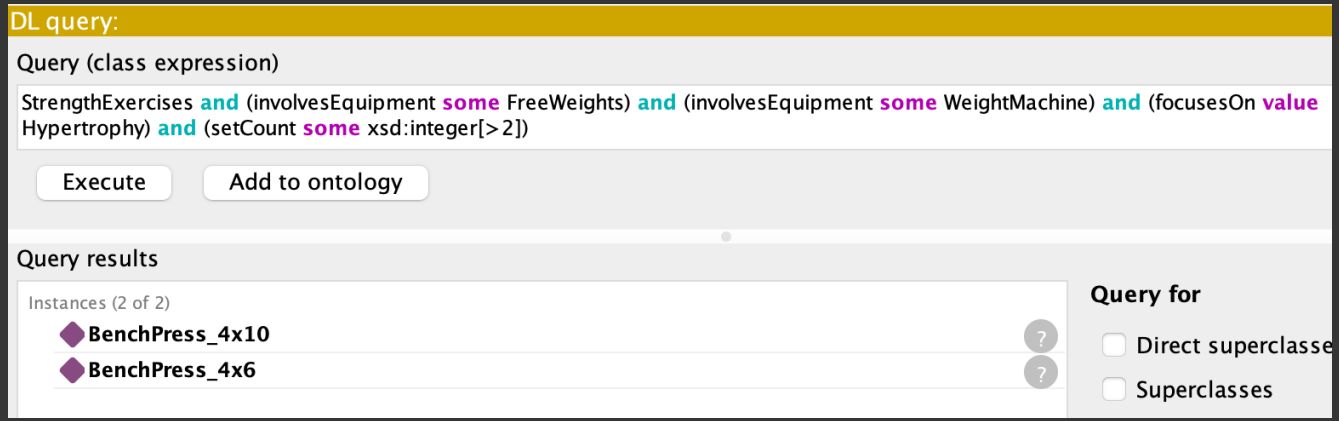
- **Find free weights that weigh more than 10kg and which exercises are they involved in that focuses on Muscle Gain or Power Lifting**

```
SELECT ?equipment ?exercise ?focus
WHERE {
    ?equipment rdf:type c:FreeWeights .
    ?equipment c:equipWeightInKg ?weight .
    ?exercise rdf:type c:StrengthExercises .
    ?exercise c:involvesEquipment ?equipment .
    ?exercise c:focusesOn ?focus .
    FILTER(?weight >= 10.0 && (?focus = c:MuscleGain || ?focus = c:PowerLifting))
}
```

equipment	exercise	focus
Barbell_20	BenchPress_4x6	MuscleGain
Plates_10	InclineBenchPress_4x10	MuscleGain
Plates_30	BenchPress_4x10	MuscleGain
Plates_30	Deadlift_3x5	MuscleGain
Kettlebell_16	KettlebellSwing_4x15	MuscleGain
Kettlebell_24	KettlebellGobletSquat_4x6	MuscleGain
Barbell_15	InclineBenchPress_4x10	PowerLifting
Barbell_15	OverheadPress_3x8	PowerLifting
DumbbellSet_12	DumbbellChestPress_4x10	PowerLifting
DumbbellSet_40	DumbbellShoulderPress_3x6	PowerLifting
Barbell_20	Deadlift_3x5	PowerLifting

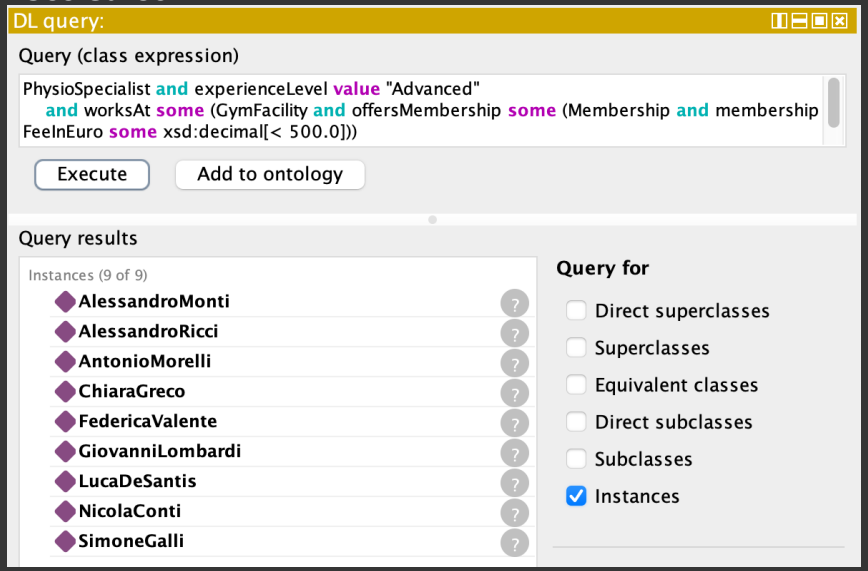
DL Queries

- Retrieve all StrengthExercises that involve both FreeWeights and WeightMachines, that focus on Hypertrophy, and have a set count greater than 2



DL Queries

- **Find the physio-specialists with advanced experience level that work at facilities which offer memberships less than 500 euros.**



References

- **Documentation:**
  - Protegé 5 documentation, https://protegeproject.github.io/protege/
  - W3C SPARQL Query Language, https://www.w3.org/TR/sparql11-query/

- **Tools:**
  - Protegé, https://protege.stanford.edu/about.php
    - Plug-ins:
      - SPARQL query version 6.0.0
      - Pellet Reasoner version 2.2.0
      - Ontology Debugger version 0.2.2
      - DL Query version 4.0.1

- **Slides:**
  - https://github.com/pietro-nardelli/sapienza-ppt-template
    [Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)]

Thank you for the attention.