# Part I: Collection of situations

## S1. An invoice is a commercial document issued by a seller to a buyer (WebProtégé or WebVOWL).

\*Note: In order to describe Part I, the document should include functional representation, graphical representation and OWL Code for all the n-ary relation patterns modeled.

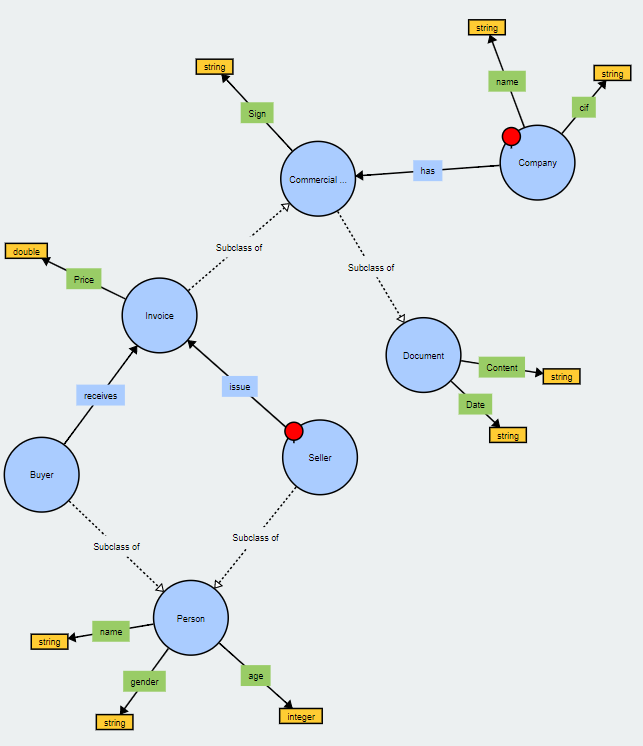
<https://webprotege.stanford.edu/#projects/b84590b9-9ea4-458e-91ea-1dec44376227/edit/Classes>

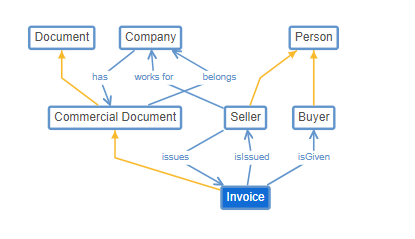
In this situation we consider both the Buyer and the Seller as a subclass of the class Person. The relation between both is not disjoint, as a Seller can have another relationship with another instance of invoice, and a Buyer can eventually be a Seller issuing an Invoice. We see an Invoice as a Document with relation to some company (even if the Seller is a contractor, that would still be his company), but as in a company there are multiple kinds of commercial documents, we decided to have three classes: Document, which is the parent of Commercial Document, which is the parent of Invoice.

* Functional representation:

Invoice(Commercial Document, Seller, Buyer)

* Graphical representation:





* OWL code:

## S6. Dealers offer cars to customers for a certain price (WebProtégé).

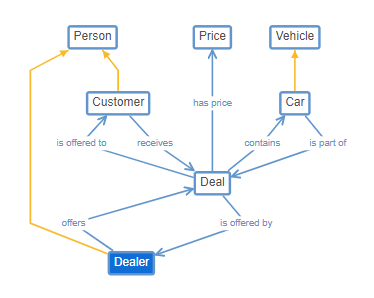
<https://webprotege.stanford.edu/#projects/dfec7708-494b-4206-ad6d-0abefe21896f/edit/Classes>

The following situation models the situation of a Dealer (which is a subclass of Person) offering a Deal to a Customer (which is also a subclass of Person). Customer and Dealer will not be disjointed for the same reason that in the previous situation Seller and Buyer were not disjointed. The deal consists of a Car (which is a kind of Vehicle) and a Price.

* Functional representation:

Invoice(Commercial Document, Seller, Buyer)

* Graphical representation:



* OWL code:

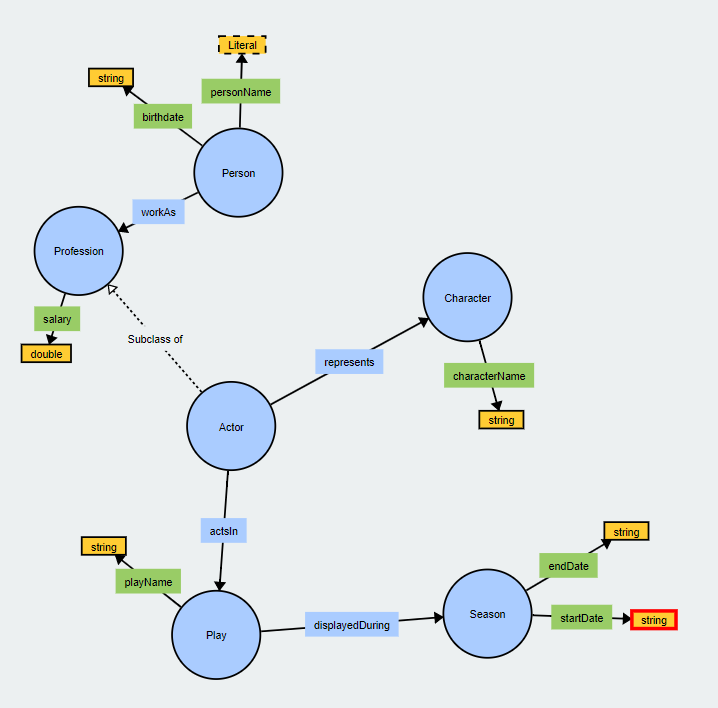
## S7. An actor plays a character in a play during a season (WebVOWL).

In this case, we will treat an Actor as a subclass of a more general Profession class. The actor represents a character, and acts in a Play, which is part of a Season.

* Functional representation:

Play(Actor, Character, Season)

* Graphical representation:



* OWL code:

## S8. Tourists use bicycles from shared bicycle services, picking them up and returning them at points of interest (WebProtégé).

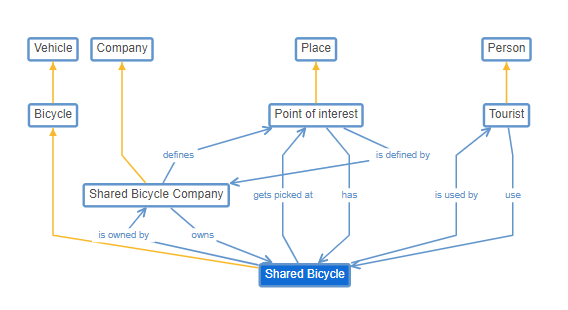
<https://webprotege.stanford.edu/#projects/1403b9df-4803-4599-a04e-907f8b8451aa/edit/Classes>

Tourists, which are a subclass of Person, use Shared Bicycles as a service provided by a Shared Bicycle Company. Share Bicycles are a type of Bicycles, which are a type of Vehicles. The same way, a Shared Bicycle Company is a child of the class Company. The Shared Bicycles are picked at Points of Interest, defined by the Shared Bicycle Company, which are Places.

* Functional representation:

SharedBicycle(SharedBicycleCompany, Tourist, Bicycle, PointOfInterest)

* Graphical representation:



* OWL code:

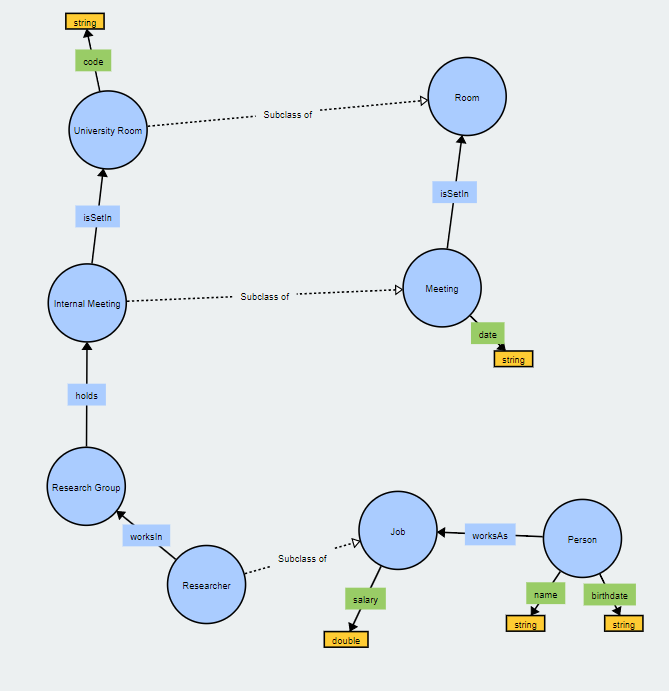
## S9. Research groups hold internal meetings at specific dates and times in a meeting room (WebVOWL).

A Research Group hold an Internal Meeting, which is a child of the class Meeting, which take place in a Room and a Date. The Internal Meetings of the Research Group are held in a University Room, which is a subclass of Room.

* Functional representation:

InternalMeeting(ResearchGroup, Date, Time, MeetingRoom)

* Graphical representation:



* OWL code:

## P1. Peter received a low mark in the Italian course and Susan a high mark in such a course (WebProtégé or WebVOWL).

<https://webprotege.stanford.edu/#projects/f8fc06a5-9724-47fe-ad97-5fe89754d01e/edit/Classes>

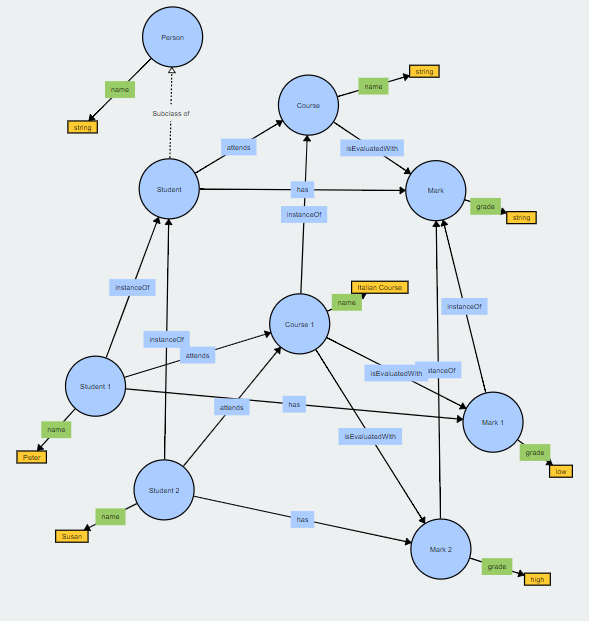
On the one hand, we have the general classes: Student is a subclass of Person, and a Student studies in a Course, which is evaluated with a Mark and can be more specific, as an Italian Course. On the other hand, we have the instances: Peter and Susan are instances of the class Student, and both study in the same instance of the class Italian Course. Both have associated a Mark with the curse, which is high in the case of Susan and low in the case of Peter.

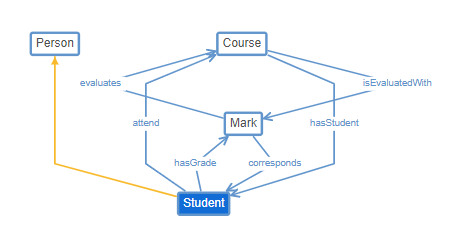
* Functional representation:

Mark(Student, Course, Value)

* Graphical representation:

\*Note: Due to the lack of functionality in the given tool to create instances, we have represented them as an objectProperty called “instanceOf”.





* OWL code:

## P2. Kate lost 5 kilos thanks to the Dukan diet, and her sister has gained 10 kilos with the Atkins diet (WebProtégé or WebVOWL).

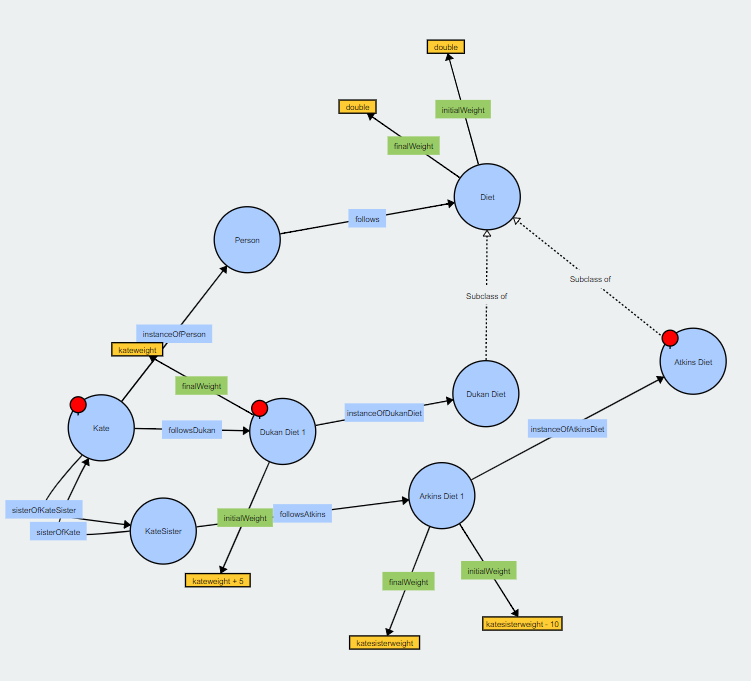
Again, we have the general classes, which include Person, which can follow a Diet, which can be of different types. In our case, we have two disjointed classes, Dukan Diet and Atkins Diet.

We also have instances, Kate and her sister (who we do not know the name) are Persons, and they both follow an instance of the Dukan Diet and the Atkins Diet respectively. Both Diets have an initial and final weight, which are attributes of their parent class, Diet.

* Functional representation:

WeightDiff(Person, Diet, Value, Trend)

* Graphical representation:



* OWL code:

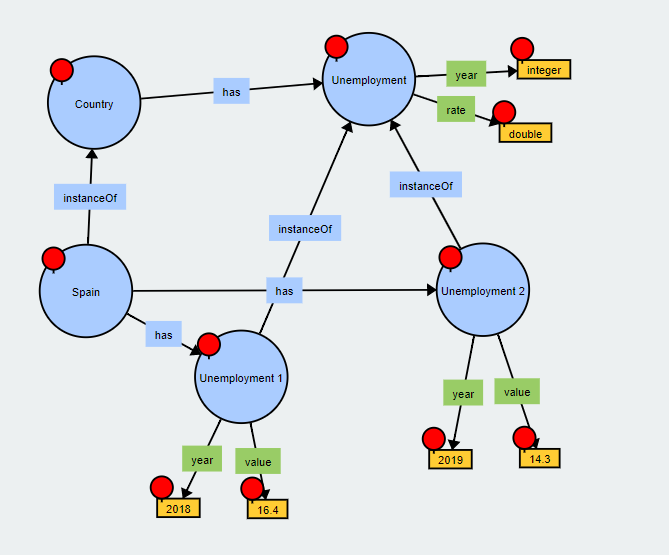
## P3. The unemployment rate in Spain is high, however is dropping (WebProtégé or WebVOWL).

The last of the relationships represent a Country with an Unemployment rate in the general sense. In this case our instance is Spain, and for the decreasing of the rate we have considered creating two instances of the class Unemployment, one representing the unemployment rate in 2018, which was 16.4%, and another one representing the unemployment rate in 2019, which was 14.3%. This way, we see it is a high rate that is dropping along the time.

* Functional representation:

Unemployment(Country, Value, Trend)

* Graphical representation:



* OWL code:

# Part II: Develop an ontology network in Education domain.

\*Note: Education: The ontology must model (at least) elements related to academic degree, courses, subjects, teachers, educational materials, as well as assignments, exercises, exams among others.

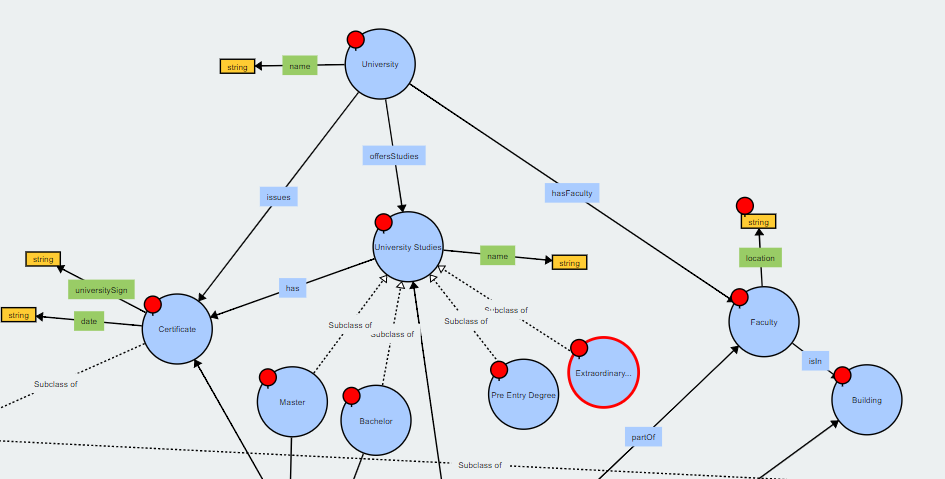
\*Note: In order to describe Part I, the document should include functional representation, graphical representation and OWL Code for all the n-ary relation patterns modeled.

## Task A:

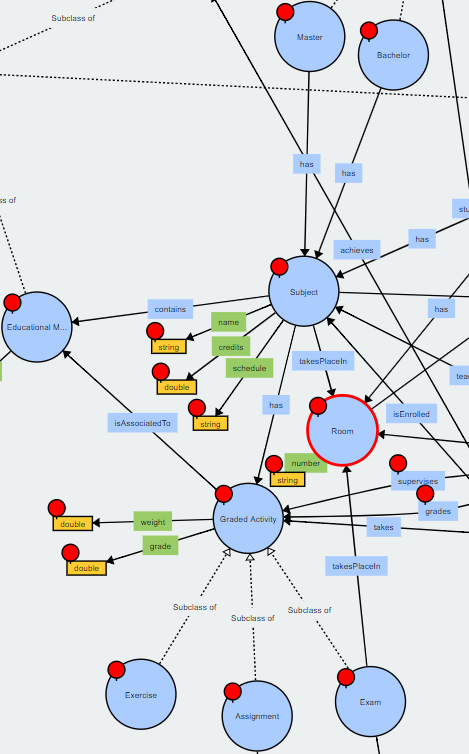
<https://webprotege.stanford.edu/#projects/c2b5bf2a-41c9-4f7e-8539-89340b01689b/edit/Classes>

We will proceed to explain the rationale behind the Education ontology. As it is too complex to explain all of it at once, we will break it down in different parts.

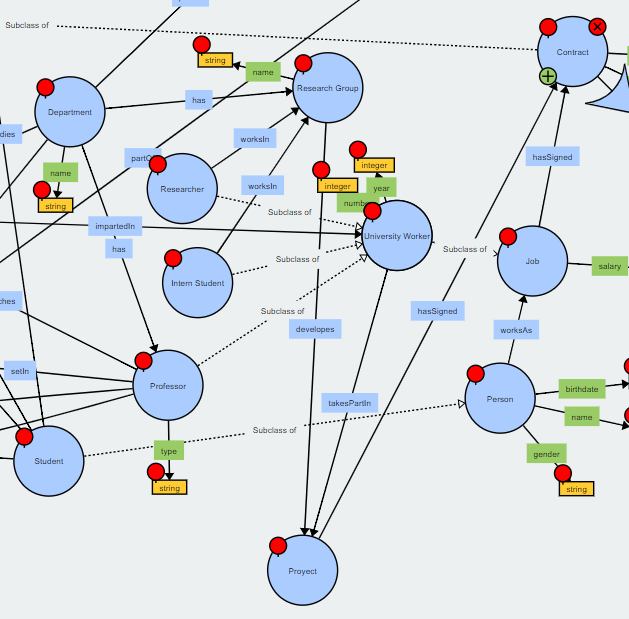
First of all, we have a University that offers different kind of studies (here taken into account Master’s degree, Bachelor’s degree, Pre-Entry Degree and Extraordinary Courses). In the completion of any of these studies, the student will receive a Certificate. Moreover, a University consists of multiple Faculties, which are in different Buildings.



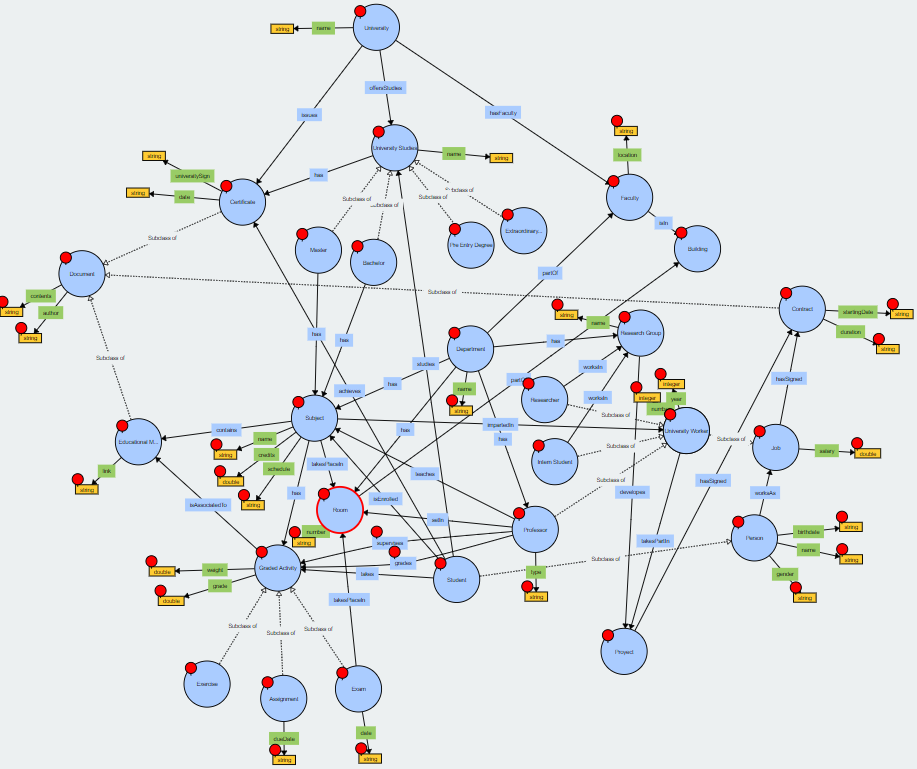
Masters and Bachelors have Subjects, which contain Educational Material as a part of their syllabus. Also, Subjects use Graded Activities (that have part of the mentioned Educational Material associated) to evaluate the course. These Graded Activities can be Exercises, Assignments or Exams (which take place in a Room).

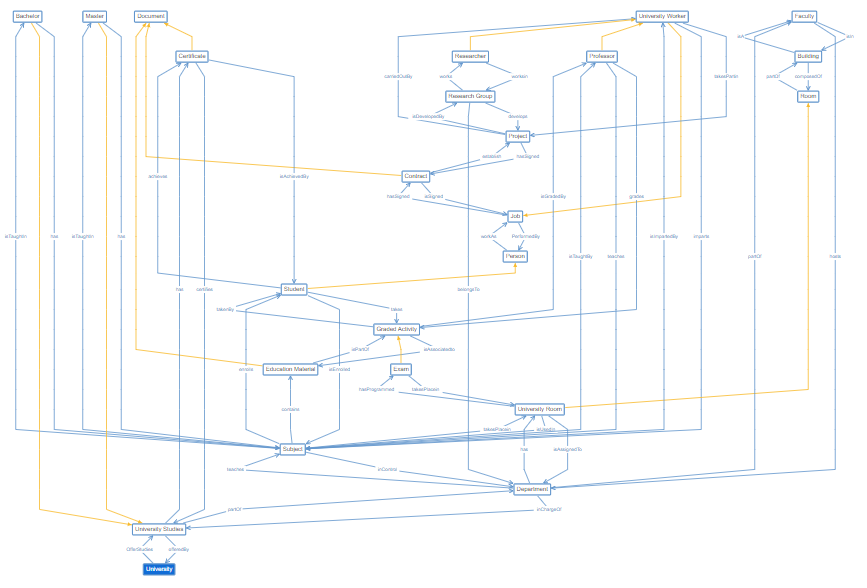


The previously mentioned Subjects correspond to a Department. These Departments are part of the Faculties and have Professors and Research Groups. A Professor is a University Worker that teaches a Subject, has an office in a Room and grades Graded Activities. Other University Workers are Researchers or Interns, which collaborate in Research Groups. On the other hand, Students study some University Studies, are enrolled in Subjects and eventually achieve their corresponding Certificate.



The whole ontology would look like this:





## Task B: