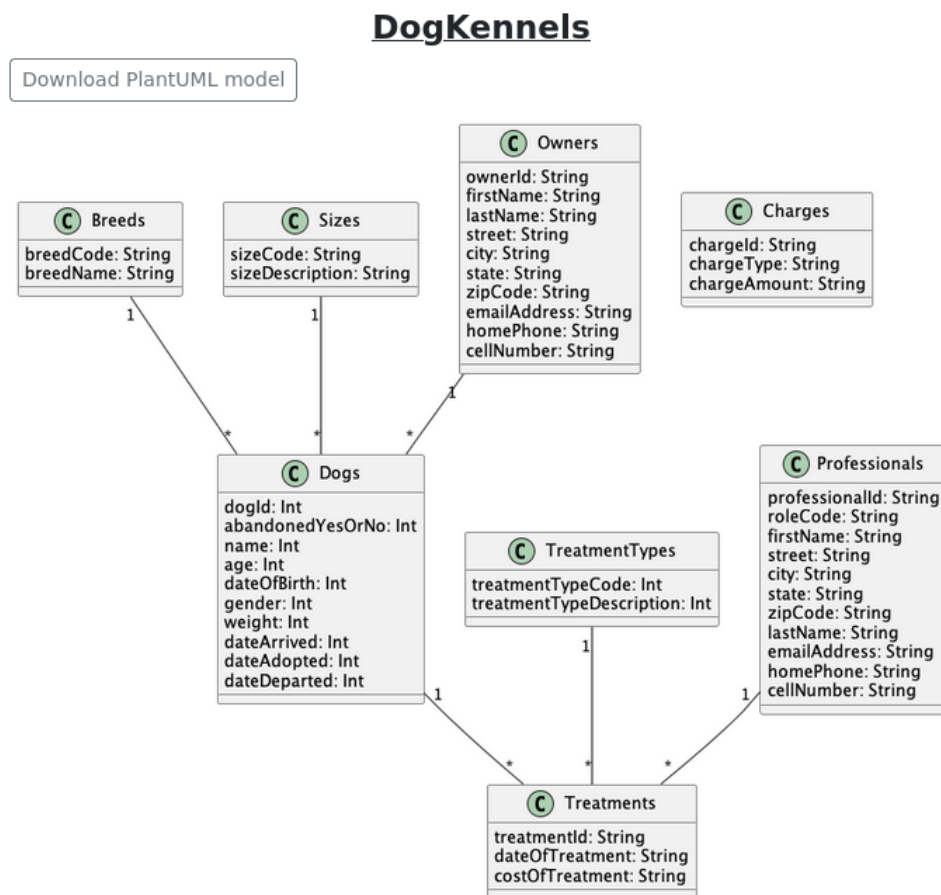


TP 4 - IAGL : OCL

TRINQUART Matthieu et SANCHEZ Martin

1 Exercice 1



1.1 What is the average age of all the dogs?

URL : <http://oclcopus.a2sistemas.com:8001/nl2ocl/question/9408/>

Réponse :

```
1      /*On fait la somme de tout les ages (en utilisant la fonction sum de
      OCL), qu'on divise par le nombre de chien total (en utilisant la
      fonction size de OCL)*/
2      Martin: Dogs.allInstances.age->sum() / Dogs.allInstances.age->size()
3
4
5      /*La même chose qu'au dessus*/
6
7      Matthieu: Dogs.allInstances->collect(age)->sum()/Dogs.allInstances->
      collect(age)->size()
8
9      /*On a pris celui la car l'écriture de la requête est plus claire (Mais
      les 2 requêtes sont en soit indentique)*/
10     Final : Dogs.allInstances.age->sum() / Dogs.allInstances.age->size()
```

1.2 What is the age of the oldest dog?

URL : <http://oclcopus.a2sistemas.com:8001/nl2ocl/question/9410/>

Réponse :

```
1      /*On trie les chiens au fonction de leur age (du plus jeune au plus vieux)
      avec la fonction sortBy de OCL et on récupère le dernier chien de la
      collection (Donc le plus vieux) avec la fonction last() de OCL*/
2      Martin : Dogs.allInstances->sortBy(age)->last()).age
3
4      /*On récupère la valeur max de la collection d'age avec la fonction max()
      de OCL*/
5      Matthieu : Dogs.allInstances->collect(age)->max()
6
7
8      /*Celui là est meilleur car elle utilise directement la fonction max() de
      OCL et est donc plus optimisé */
9      Final : Dogs.allInstances->collect(age)->max()
```

1.3 What are the date and the operating professional's first name of each treatment ?

URL : <http://oclcorpus.a2sistemas.com:8001/nl2ocl/question/9369/>

Réponse :

```
1
2  /*On créer un Tuple avec la dateOfTreatment et le firstName du professionnel
   courant*/
3  Martin : Treatments.allInstances->collect(Tuple{date=dateOfTreatment,
   invalidSessionprofessional=Professionals.firstName})
4
5  /*La même chose qu'au dessus*/
6  Matthieu : Treatments.allInstances->collect(Tuple{Date = dateOfTreatment,
   FirstName = Professionals.firstname})
7
8
9
10 /*Les deux fonctions sont les mêmes*/
11 Final : Treatments.allInstances->collect(Tuple{Date = dateOfTreatment,
   FirstName = Professionals.firstname})
```

1.4 Which dogs are owned by someone who lives in Virginia? List the owner's first name and the dog's name.

URL : <http://oclcorpus.a2sistemas.com:8001/nl2ocl/question/9378/>

Réponse :

```
1  /*Pour chaque instance de chien on utilise la fonction select() de OCL pour
   sélectionner que les instances avec un propriétaire qui habite au
   Virginia. Puis on crée un tuple avec le nom du propriétaire et le nom du
   chien*/
```

```

2 Martin : Dogs.allInstances->select(d | d.Owners.state = 'Virginia')->
    collect(Tuple{ ownerName=Owners.firstName , dogsName=name })
3
4 /*La même chose qu'au dessus*/
5 Matthieu : Dogs.allInstances->select(p | p.Owners.state = 'Virginia')->
    collect(Tuple{NameDog = p.name , NameOweners = p.Owners.firstName})
6
7
8 /*Les deux requêtes sont identiques*/
9 Final : Dogs.allInstances->select(d | d.Owners.state = 'Virginia')->collect
    (Tuple{ ownerName=Owners.firstName , dogsName=name })

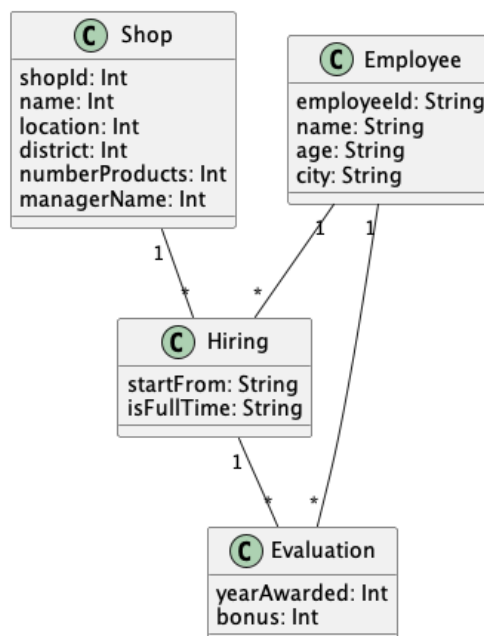
```

2 Exercice 2

2.1 Sort all the shops by number products in descending order, and return the name, location and district of each shop.

EmployeeHireEvaluation

Download PlantUML model

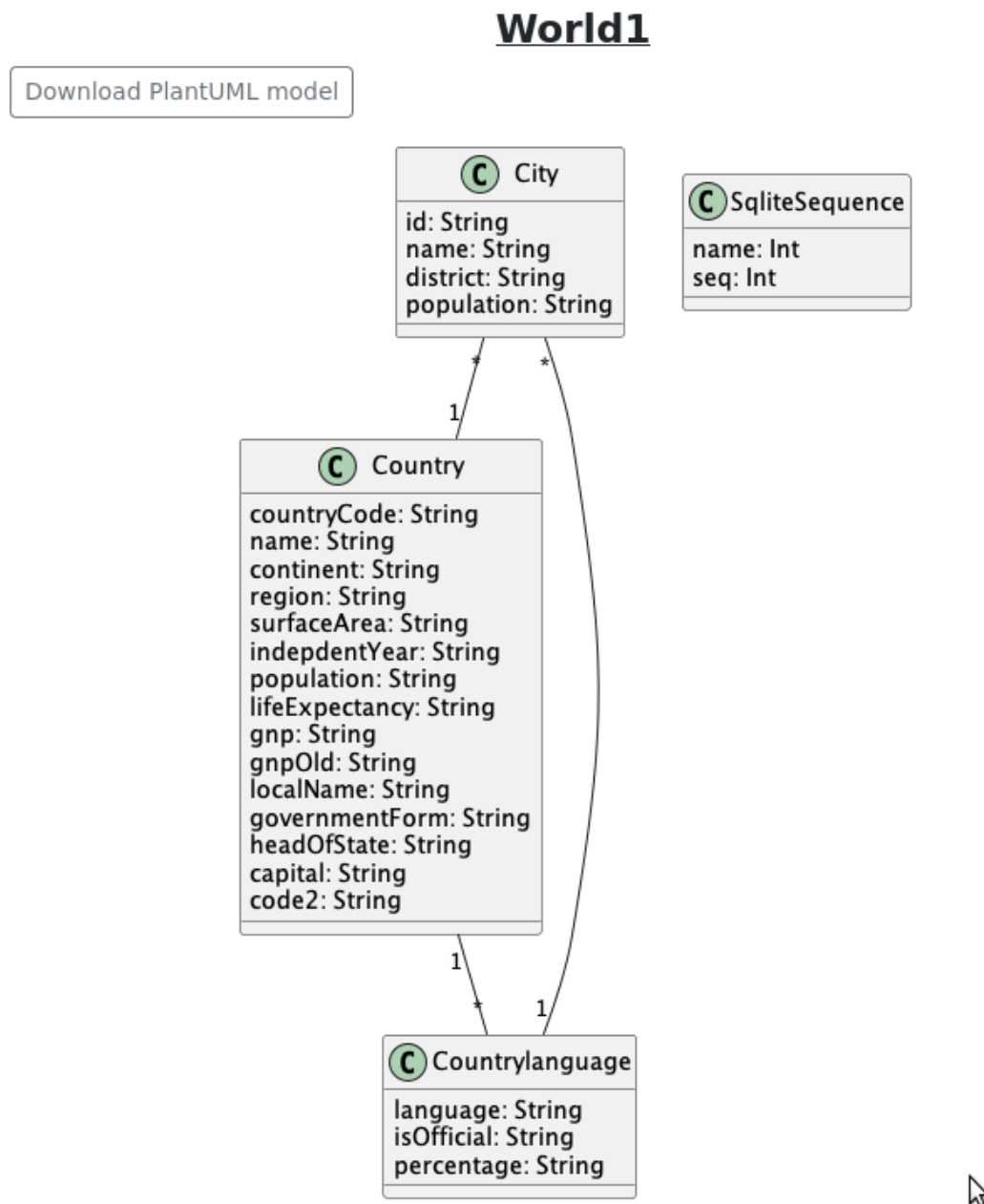


URL : <http://oclcorpus.a2sistemas.com:8001/nl2ocl/question/8696/>

Réponse :

```
1      /*On trie les Shops en fonction de leur numéro de production et on crée
      un tuple avec le nom, la localisation et district du shop*/
2      Martin : Shop.allInstances->sortedBy(numberProducts)->collect(Tuple{
      name = name , location = location , district = district})
3
4      /*La même chose qu'au dessus*/
5      Matthieu : Shop.allInstances->sortedBy(numberProducts).collect(Tuple{
      NameShop = name, Location = location , District = district })
6
7      /*Les deux requêtes sont identiques*/
8      Final : Shop.allInstances->sortedBy(numberProducts)->collect(Tuple{name
      = name , location = location , district = district})
```

2.2 What are the names of all the countries that became independent after 1950 ?



URL : <http://oclcopus.a2sistemas.com:8001/nl2ocl/question/9124/>

Réponse :

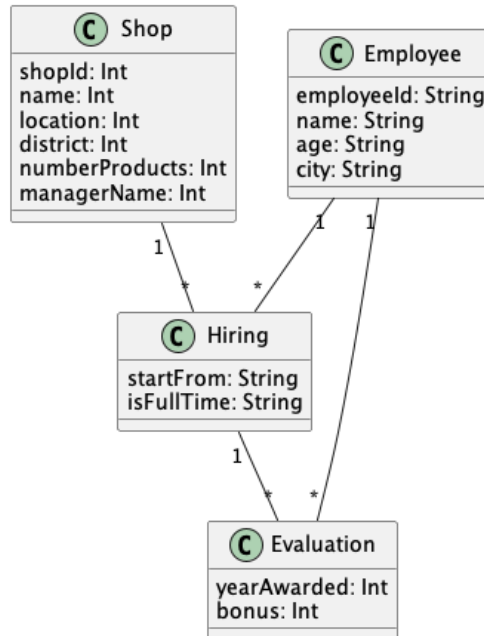
```
1 /*On selectionne le name de chaque instance qui à un indepndentYear supè  
   rieur à '1950' en utilisant la fonction select() de OCL*/
```

```
2      Martin : Country.allInstances->select(c | c.indepdentYear > '1950').
      name
3      /*Même chose qu'au dessus*/
4      Matthieu : Country.allInstances->select(c| c.indepdentYear > '1950')->
      collect(c.name)
5
6      /*La première requête est plus claire et lisible*/
7      Final : Country.allInstances->select(c | c.indepdentYear > '1950').name
```

2.3 What is total bonus given in all evaluations ?

EmployeeHireEvaluation

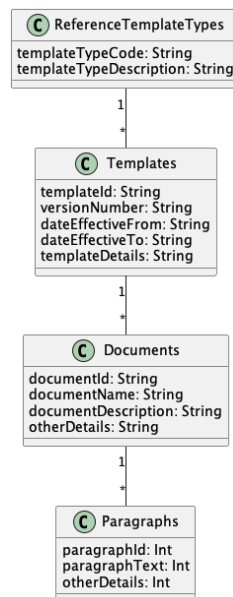
Download PlantUML model



URL : <http://oclcampus.a2sistemas.com:8001/nl2ocl/question/8711/>

```
1      /*Sur chaque instance on fait la somme de tout les bonus (avec la
2      fonction sum()*/
3
4      Martin : (Evaluation->allInstances.bonus)->sum()
5
6
7      /*Même chose qu'au dessus*/
8      Matthieu : Evaluation->allInstances()->collect(bonus)->sum()
9
10
11     /*La première requête est plus claire et lisible*/
12     Final : (Evaluation->allInstances.bonus)->sum()
```


2.4 What is the version number and template type code for the template with version number later than 5 ?



URL : <http://oclcorg.a2sistemas.com:8001/nl2ocl/question/8749/>

```

1      /*Pour toutes les instances On prend un Tuple quand la version est supè
      rieur a 5*/
2      Martin : Template->allInstances->collect( Tuple{version=versionNumber,
      templateTypeCode=ReferenceTemplatesTypes.templateTypeCode})->select(
      version > 5)
3
4      /*Pour toutes les instances On prend un Tuple quand la version est supè
      rieur a 5*/
5      Matthieu : Templates->allInstances()->select(t| t.versionNumber < 5)->
      collect(t | Tuple{VersionNumber = t.versionNumber, TemplateTypeCode =t.
      ReferenceTemplateTypes.templateTypeCode})
6
7      /*La première requête est plus compréhensible*/
8      Final : Templates->allInstances()->select(t| t.versionNumber < 5)->
      collect(t | Tuple{VersionNumber = t.versionNumber, TemplateTypeCode =t.
      ReferenceTemplateTypes.templateTypeCode})
  
```