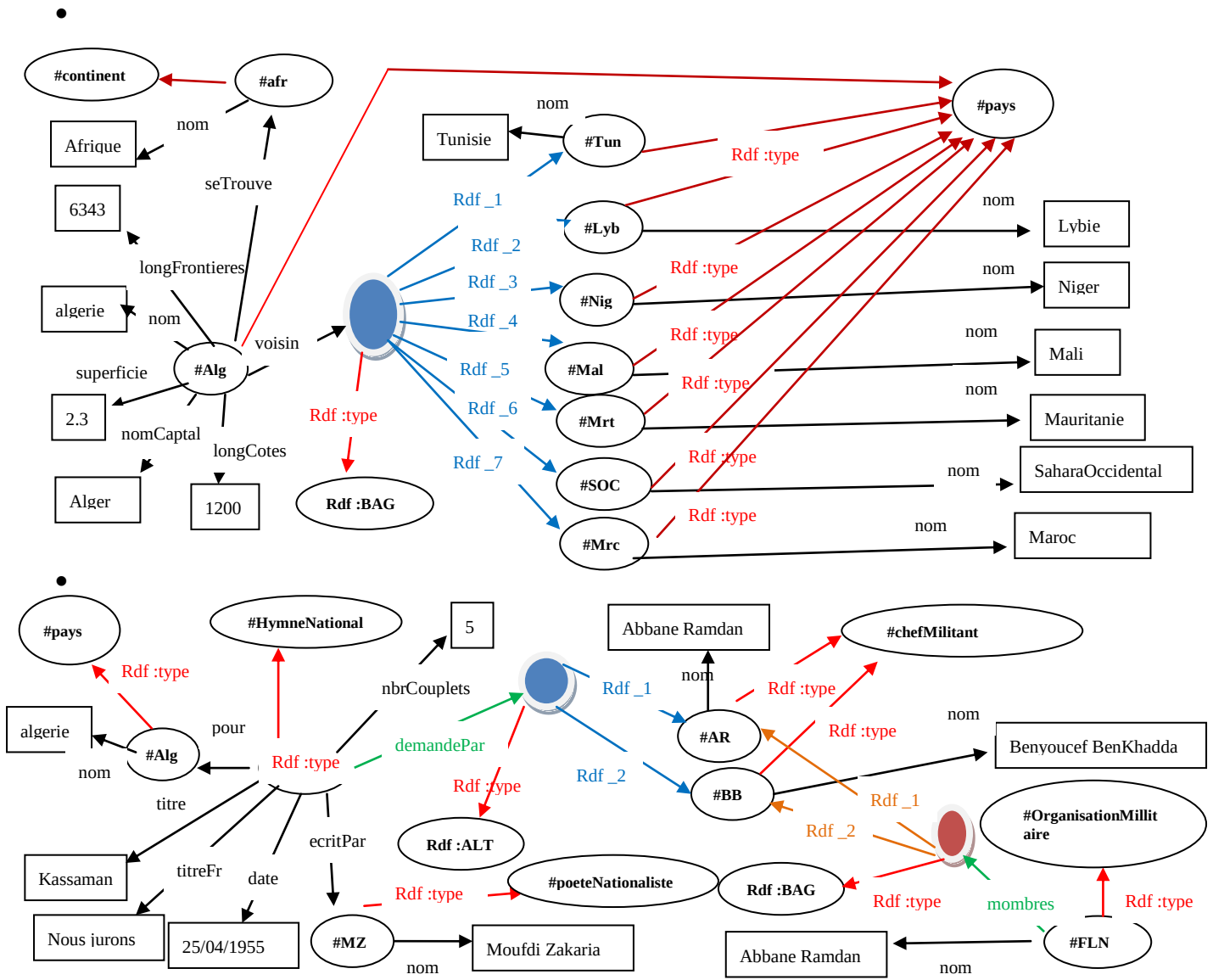


Exercice 1 (4pts) :



Exercice 2 (6pts) :

Donner le rôle et le résultat des requêtes SPARQL suivantes. (05.+0.5)*3

<pre>SELECT ?n WHERE {?pers :nom ?n. ?pers :age ?a. ?pers :possede ?v.}</pre>	<pre>SELECT ?n WHERE { ?pers :nom ?n. ?pers :possede ?v. ?v :deType ?t. ?t :nom "Renault Megane".}</pre>	<pre>SELECT ?n WHERE {?pers :nom ?n. ?pers :possede ?v. ?v :puissance "Essence".}</pre>								
Les noms de personnes possédant une ou plusieurs voitures ayant la valeur de l'age	Les noms de personnes possédant une ou plusieurs voitures du type 'Renault Megane'	Les nom des personnes possadant une ou des voitures de type energie 'Essecnce'								
<table><tr><th>?n</th></tr><tr><td>Farid</td></tr><tr><td>saleh</td></tr></table>	?n	Farid	saleh	<table><tr><th>?n</th></tr><tr><td></td></tr></table>	?n		<table><tr><th>?n</th></tr><tr><td></td></tr><tr><td></td></tr></table>	?n		
?n										
Farid										
saleh										
?n										
?n										

- Donner les requetes SPARQL qui donnent : **1pts*3**

<pre>SELECT DISTINCT ?n ?a WHERE {?pers :nom ?n. ?pers :possede ?v1. ?pers :possede ?v2. FILTER (?v1!=?v2). OPTIONAL {?pers :age ?a.} }</pre>	<pre>SELECT DISTINCT ?n WHERE {?uriMrq :nom "Renault". ?uriType :deMarque ?uriMrq. ?uriV :deType ?uriType. ?pers :possede ?uriV. ?pers :nom ?n.}</pre>	<pre>SELECT DISTINCT ?n WHERE {?pers :nom ?n. ?pers :possede ?v1. ?pers :possede ?v2. FILTER (?v1!=?v2). ?v1 :deType ?t1.?v2 :deType ?t2. ?t1 :deMarque ?m1.?t2 :deMarque ?m2. FILTER (?m1!=?m2). ?v1 :puissance ?p1. ?v2 :puissance ?p2. FILTER (?p1!=?p2).}</pre>
Les noms et l'age si possible des personnes ayant plus d'une voitures.	noms des personnes ayant des voitures de la marque « Renault	Les noms des personnes ayant au moins deux voitures de marques différentes et de types de puissances différentes

Exercice 3 (5pts)

Une ontologie OWL existante contenant les éléments suivants :

Les classes : personne, homme, metier, filiere .

- Ajouter la classe femme (*avec les relations possibles avec les autres classes*)

```
<owl:class rdf:ID='femme'> 0.75 pts
<rdfs:subClassOf rdf:resource='personne' />
<owl:disjointWith rdf:resource='homme' />
</owl:class>
```

- Ajouter la classe pour les métiers medecin .

```
<owl:class rdf:ID='medecin'> 0.5 pts
<rdfs:subClassOf rdf:resource='metier' />
</owl:class>
```

- une personne doit avoir un nom ,ajouter la propriété nom .

```
<owl:DatatypeProperty rdf:ID='nom'> 0.5 pts
<rdfs:domain rdf:resource="#Personne" />
<rdfs:range rdf:resource="&xsd:String" />
</owl:DatatypeProperty>
```

- Ajouter les relations :enfantDe,aMetier ,etudie ,parentDe .

```
owl:ObjectProperty rdf:ID='enfantDe'>0.75 pts
<rdfs:domain rdf:resource="#personne" />
<rdfs:range rdf:resource="#personne" />
</owl:ObjectProperty>
```

```
owl:ObjectProperty rdf:ID='aMetier'>0.75 pts
<rdfs:domain rdf:resource="#personne" />
<rdfs:range rdf:resource="#metier" />
</owl:ObjectProperty>
```

```
owl:ObjectProperty rdf:ID='aMetier'>0.75 pts
<rdfs:domain rdf:resource="#personne" />
<rdfs:range rdf:resource="#filiere" />
</owl:ObjectProperty>
```

```
< Owl: ObjectProperty rdf:ID='parentDe'>0.5 pts
<owl:inverseOf rdf:resource="#enfantDe" />
</owl:ObjectProperty>
```

- ajouter les définitions des classes suivantes :

- o Etudiant.

```
<owl:class rdf:ID='Etudiant'> 1 pts
<owl:equivalentClass>
  <owl:Restriction >
    <owl:onProperty rdf:resource=''#etudier' />
    <owl:minCardinality >1</owl:maxCardinality>
  </owl:Restriction >
```

100

</owl:class>

</owl:class>

</owl:class>

</owl:class>

- Branche $\in$ 1 partie De Arbre 0.5pts

- Feuille $\sqsubseteq$ 1 partieDe.Branche. 0.5pts
- Herbivore $\equiv$ Animal $\sqcap \forall$ mange. (Plante $\vee \forall$ partieDe.Plante). 0.75pts
- Carnivore $\equiv$ Animal $\sqcap \exists$ mange.Animal. 0.75pts
- Lion $\sqsubseteq$ Carnivore $\sqcap \forall$ mange.Herbivore. 0.75pts
- Girafe $\sqsubseteq$ Herbivore $\sqcap \forall$ mange.Feuille. 0.75pts
- PlanteSavoureuse $\equiv$ Plante $\sqcap \exists$ mangePar. (Herbivore $\vee$ Carnivore). 0.75pts
- Ajouter les fait de la ABOX suivante :
 - *Giri* est une girafe qui mange une feuille d'un arbre.
 - Girafe (Giri) 0.25pts
 - Feuille(laFeuille) 0.25pts
 - Arbre(lArbre) 0.25pts
 - partieDe(laFeuille ,partieDe(lArbre)) 0.25pts
 - Mange(Giri, laFeuille) 0.25pts
 - *Lionel* est un lion qui mange une gazelle .
 - Lion(Lionel) 0.25pts
 - Herbivore(laGazele) 0.25pts
 - Mange (Lionel, laGazelle) 0.25pts

Bonne copie

Mr. Bougherara.S