

Web Sémantique — Examen SUJET A

Cours #2, #3 et #4 — Triplets, OWL, SPARQL, déploiement

Durée : 1h30
Sans documents
Barème : 20 points

Nom : _____ Prénom : _____ Sujet : A

Consignes : Lisez chaque question attentivement. Répondez dans les cadres prévus. Toute réponse non justifiée ne rapportera que la moitié des points. La clarté et la précision sont évaluées. Quelques questions font appel à la réflexion — ne restez pas bloqué, passez à la suite. **Vous avez le sujet A.**

Exercice 1 — Vrai ou Faux (6 points — 12 × 0,5 pt)

Pour chaque affirmation, entourez **V** (vrai) ou **F** (faux). Une réponse fausse enlève 0,25 pt. Pas de réponse = 0.

	Affirmation	V	F
1	En RDF, un triplet est composé d'un sujet, d'un prédicat et d'un objet.	<input type="checkbox"/>	<input type="checkbox"/>
2	L'unification de <code>père(X, jean)</code> et <code>père(paul, Y)</code> donne $\{X/paul, Y/jean\}$.	<input type="checkbox"/>	<input type="checkbox"/>
3	Le chaînage avant part d'un but et remonte les règles jusqu'aux faits.	<input type="checkbox"/>	<input type="checkbox"/>
4	En OWL, \exists <code>a_mangé.Viande</code> signifie « ne mange que de la viande ».	<input type="checkbox"/>	<input type="checkbox"/>
5	Une ontologie OWL peut être incohérente sans violer la syntaxe RDF.	<input type="checkbox"/>	<input type="checkbox"/>
6	SPARQL permet d'interroger des graphes RDF mais pas de créer de nouveaux triplets.	<input type="checkbox"/>	<input type="checkbox"/>
7	La requête <code>SELECT ?s WHERE { ?s rdf:type :Homme }</code> retourne tous les hommes.	<input type="checkbox"/>	<input type="checkbox"/>
8	En Turtle, <code>rdf:type</code> est équivalent à <code>a</code> .	<input type="checkbox"/>	<input type="checkbox"/>
9	Le raisonneur Hermit est capable de classier une ontologie OWL 2.	<input type="checkbox"/>	<input type="checkbox"/>
10	\forall <code>a_ingredient.Fromage</code> signifie qu'au moins un ingrédient est du fromage.	<input type="checkbox"/>	<input type="checkbox"/>
11	Apache Jena Fuseki est un serveur SPARQL écrit en Python.	<input type="checkbox"/>	<input type="checkbox"/>
12	Un graphe RDF peut contenir des littéraux comme objets.	<input type="checkbox"/>	<input type="checkbox"/>

Exercice 2 — Triplets, Prolog, raisonnement (5 points)

Question 2.1 — Traduction en triplets

(1,5 pt)

Traduisez les phrases suivantes en triplets RDF (utilisez des préfixes si nécessaire) :

1. « *Montpellier est une ville.* »
2. « *Montpellier est située dans le Sud de la France.* »
3. « *Toutes les villes du Sud ont du soleil.* » (écrivez la règle Prolog)

Question 2.2 — Chaînage avant

(2 pts)

Soit les faits et règles suivants :

```
est_un(medor, chien).
est_un(felix, chat).
est_un(medor, animal).
aime(medor, felix).

amitie(X) ← aime(X, Y) ∧ est_un(Y, animal).
ennemi(X) ← est_un(X, chat) ∧ ¬aime(chien_du_voisin, X).
```

1. Appliquez le chaînage avant. Quels nouveaux faits déduisez-vous ? (1 pt)
2. Ajoutez une règle copain qui vérifie que deux individus s'aiment mutuellement. (0,5 pt)
3. La deuxième règle peut-elle provoquer un problème ? Expliquez. (0,5 pt)

Question 2.3 — Unification

(1,5 pt)

Pour chaque paire, indiquez si l'unification réussit et, si oui, donnez la substitution :

1. `aime(medor, X)` et `aime(Y, felix)`
2. `est_un(Z, animal)` et `est_un(medor, Z)`
3. `pere(X, mere(X))` et `pere(jean, mere(paul))`

Exercice 3 — OWL : classes, restrictions, cohérence (5 points)

Question 3.1 — Hiérarchie et héritage

(1,5 pt)

Organisez les classes suivantes en hiérarchie OWL (\sqsubseteq) :

Animal · Mammifère · Oiseau · ChauveSouris · Moineau · Rapace · Aigle

Contrainte : une chauve-souris est un mammifère qui vole. Un aigle est un rapace. Un rapace est un oiseau.

Question 3.2 — Restrictions

(2 pts)

Traduisez en OWL (utilisez \sqcap , \sqcup , \neg , \exists , \forall , \leq , \geq) :

1. « Les chauves-souris sont des mammifères qui volent. » (0,5 pt)
2. « Un moineau est un oiseau qui ne mange que des graines. » (0,5 pt)
3. « Un aigle est un rapace qui chasse au moins un petit mammifère. » (0,5 pt)
4. « Les rapaces diurnes sont des rapaces qui chassent uniquement le jour. » (0,5 pt)

Question 3.3 — Cohérence

(1,5 pt)

Cette ontologie est-elle cohérente ? Justifiez.

```
Moineau  $\sqsubseteq$  Oiseau
Moineau  $\sqsubseteq$   $\forall$  mange.Végétal
Aigle  $\sqsubseteq$  Rapace
Rapace  $\sqsubseteq$  Oiseau
Rapace  $\sqsubseteq$   $\exists$  chasse.Mammifère
Aigle  $\sqsubseteq$   $\exists$  aime.MangeurDeSerpents
titi  $\sqsubseteq$  Moineau
titi  $\sqsubseteq$   $\exists$  a_pour_amie. $\neg$ Moineau
titi  $\sqsubseteq$   $\neg(\exists$  a_pour_amie.Chose) ← question piège
```

Exercice 4 — SPARQL (4 points)

Soit la base RDF suivante (préfixe : `http://ex.org/`) :

```
@prefix : <http://ex.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:Montpellier rdf:type :Ville ; :dans :Sud ; :population "302000" .
:Paris       rdf:type :Ville ; :dans :Nord ; :population "2161000" .
:Lyon        rdf:type :Ville ; :dans :Sud ; :population "522000" .
:Marseille   rdf:type :Ville ; :dans :Sud ; :population "861000" .
:Toulouse    rdf:type :Ville ; :dans :Sud .
:Sud         rdf:type :Region ; :soleil true .
:Nord        rdf:type :Region ; :soleil false .
```

Question 4.1 — SELECT

(1 pt)

Écrivez une requête SPARQL qui retourne toutes les villes du Sud :

Question 4.2 — FILTER

(1 pt)

Écrivez une requête qui retourne les villes de plus de 500 000 habitants, classées par population décroissante :

Question 4.3 — CONSTRUCT

(1 pt)

Écrivez une requête CONSTRUCT qui crée une relation `:a_du_soleil` entre chaque ville et `true/false` selon sa région :

Question 4.4 — Culture web sémantique

(1 pt)

Citez deux différences entre SQL et SPARQL (hors syntaxe) :
