TME RDF et SPARQL

Kaan Yagci kaan@kaanyagci.com

L'énoncé de cette TME peut être trouvé ici

Exercices

Exercice facultatif

Construire des requêtes SPARQL à partir des motifs de l'exercice 2 du TD et les tester sur les triplets du fichier **Ex2.ttl**.

Exercice obligatoire

Reprendre les exercices 3 et 4 du TD en utilisant, respectivement, les triplets des fichiers Ex3.ttl et Ex4.ttl. Vous pouvez utiliser les fichiers qx_Ex3.spl et qx_Ex4.spl fournis. Écrivez chaque requête dans un fichier séparé.

Exercice 3

Q1.a : Extraire l'ensemble des IRI des sujets

```
A COMPLETER
```

Q1.b : Idem en retournant les nom locaux

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT DISTINCT ?nom WHERE {
?nom ?relation ?personne.
}
```

Résultat attendu

```
---------------
| nom         |
===============
```

```
| :nyc        |
| :mi         |
| :J.Horrow   |
| :monica     |
| :liz        |
| :cmu        |
| :larry      |
| :ucsb       |
| :luke       |
| :robert     |
| :suzan      |
| :john       |
| :Jim        |
| :pittsburgh |
| :richard    |
| :dan        |
| :mit        |
---------------
```

Q2 : Même question avec les prédicats

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?relation
WHERE
{
    ?person ?relation ?person1
}
```

Résultat attendu

```
-----------------
| relation      |
=================
| rdf:type      |
| :studiedAt    |
| :supervisedBy |
| :friend       |
| :livesIn      |
| :hasFather    |
| :hasMother    |
```

```
| :hasBrother   |
| :locatedAt    |
| :hasDegree    |
| rel:ChildOf   |
-----------------
```

Q3 : Les villes citées dans des triples de cette base.

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?ville
WHERE
{
    ?attribute :livesIn|:locatedAt ?ville
}
```

Résultat attendu

```
------------------
| ville          |
==================
| :santaBarbara  |
| :pittsburgh    |
| :westLafayette |
| :nyc           |
------------------
```

Q4 : Les personnes qui ont étudié dans la même université que l'un de leur parents (sans prendre en compte la propriété childOf).

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?personne
WHERE
{
    ?personne a ?personType .
```

```
    FILTER (?personType IN (:Person, :Artist)) .
    ?personne :hasMother|:hasFather ?parent .
    ?personne :studiedAt ?place .
    ?parent :studiedAt ?place
}
```

Résultat attendu

```
------------
| personne |
============
| :liz     |
| :john    |
| :larry   |
------------
```

Q5 : Les personnes qui ont étudié dans une université où leur deux parents ont étudié.

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?personne
WHERE
{
    ?personne a ?personType .
    FILTER (?personType IN (:Person, :Artist)) .
    ?personne :hasMother ?mother .
    ?personne :hasFather ?father .
    ?personne :studiedAt ?place .
    ?mother :studiedAt ?place .
    ?father :studiedAt ?place
}
```

Résultat attendu

```
------------
| personne |
============
| :liz     |
------------
```

Q6 : Donnez les personnes qui ont étudié dans une université où aucun de leur parent n'a étudié.

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?personne
WHERE
{
    ?personne a ?personType .
    FILTER (?personType IN (:Person, :Artist)) .
    ?personne :hasMother ?mother .
    ?personne :hasFather ?father .
    ?personne :studiedAt ?place .
    ?mother :studiedAt ?pplace .
    ?father :studiedAt ?pplace .
    FILTER NOT EXISTS {?personne :studiedAt ?pplace}
}
```

Résultat Attendu

```
------------
| personne |
============
------------
```

Q7 : Donnez les noms et les universités des personnes qui ont au moins un frère ou une soeur

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?personne ?universite
WHERE
{
    ?personne a ?personType .
    FILTER (?personType IN (:Person, :Artist)) .
    ?personne ?siblingPredicate ?sibling
    FILTER (?siblingPredicate IN (:hasBrother, :hasSister)) .
```

```
    ?personne :studiedAt ?universite
}
```

Résultat attendu

```
-------------------------
| personne | universite |
=========================
| :liz     | :cmu       |
| :john    | :ucsd      |
-------------------------
```

Question subsidiaire : Donnez les noms et les universités des personnes qui ont au moins un frère ou une soeur et qui n'ont pas de frère ou de soeur qui ont étudié à la même université qu'elles.

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?personne ?universite
WHERE
{
    ?personne a ?personType .
    FILTER (?personType IN (:Person, :Artist)) .
    ?personne ?siblingPredicate ?sibling
    FILTER (?siblingPredicate IN (:hasBrother, :hasSister)) .
    ?personne :studiedAt ?universite .
    FILTER NOT EXISTS { ?sibling :studiedAt ?universite }
}
```

Résultat attendu

```
-------------------------
| personne | universite |
=========================
| :liz     | :cmu       |
| :john    | :ucsd      |
-------------------------
```

Q8 : Extraire les personnes qui étudient dans une ville différente de celle où ils habitent.

```
base <http://example.org>
```

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?personne
WHERE
{
    ?personne a ?personType .
    FILTER (?personType IN (:Person, :Artist)) .
    ?personne :studiedAt ?enseignement .
    ?personne :livesIn ?bled
    FILTER NOT EXISTS { ?enseignement :locatedAt ?bled }
}
```

Résultat attendu

```
------------
| personne |
============
| :liz     |
------------
```

Q9 : Extraire les personnes qui sont ami(e)s d'un(e) ami(e) de Liz.
Remarque : ne pas retourner Liz et les ami(e)s direct(e)s de Liz!

```
base <http://example.org>
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix rel: <http://www.perceive.net/schemas/relationship/>
prefix : <http://example.org/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix univ: <http://www.faculty.ac>

SELECT distinct ?amisDeAmisDeLiz
WHERE
{
{
    {?amisDeLiz :friend :liz} UNION
    {:liz :friend ?amisDeLiz}
} .
{
    {?amisDeLiz :friend ?amisDeAmisDeLiz} UNION
    {?amisDeAmisDeLiz :friend ?amisDeLiz}
} .
FILTER NOT EXISTS { {?amisDeAmisDeLiz :friend :liz } UNION {:liz :friend ?amisDeAmisDeLiz }}
```

```
FILTER (?amisDeAmisDeLiz NOT IN (:liz))
}
```

Résultat Attendu

```
-------------------
| amisDeAmisDeLiz |
===================
| :john           |
-------------------
```

Exercice 4

Q1 : Noms des employés avec leur job suivant l'ordre alphabétique décroissant de leur nom.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>

SELECT ?name ?job
WHERE {
    ?name a f:emp .
    ?name f:Job ?job
} ORDER BY ?name
```

Résultat attendu :

```
------------------------
| name   | job         |
========================
| :dan   | "engineer"  |
| :john  | "architect" |
| :larry | "singer"    |
| :liz   | "doctor"    |
------------------------
```

Q2 : Les 3 premiers employés suivant l'ordre alphabétique de leurs noms de famille (surname).

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>
```

```
SELECT ?surname
WHERE {
    ?name a f:emp .
    ?name foaf:surname ?surname
} ORDER BY ?surname LIMIT 3
```

Résultat attendu

```
-----------
| surname |
===========
| "dan"   |
| "john"  |
| "larry" |
-----------
```

Q3 : Les 3 premiers employés, qui ont un nom de famille (surname) et qui sont les mieux payés.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>

SELECT ?surname ?salaire
WHERE {
    ?name a f:emp .
    ?name foaf:surname ?surname .
    ?name f:Sal ?salaire
} ORDER BY DESC (?salaire) LIMIT 3
```

Résultat attendu

```
----------------------
| surname | salaire |
======================
| "liz"   | "42"    |
| "dan"   | "33"    |
| "john"  | "33"    |
----------------------
```

Q4 : Les employés qui ne sont ni médecin ni chanteur.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>

SELECT *
WHERE {
    ?name f:Job ?job .
    MINUS { ?name f:Job "singer"}  .
    MINUS { ?name f:Job "doctor"}
}
```

Résultat attendu

```
-----------------------
| name  | job         |
=======================
| :dan  | "engineer"  |
| :john | "architect" |
-----------------------
```

Q5 : Les employés dont le nom commence par l.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>

SELECT ?name
WHERE {
    ?name a f:emp .
    FILTER (strStarts(str(?name) , concat(str(:) , str("l"))))
}
```

Résultat attendu

```
----------
| name   |
==========
| :larry |
| :liz   |
----------
```

Q6.a : Le plus grand salaire (sans utiliser ORDER BY et LIMIT). Retourner l'employé et son salaire.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>

SELECT ?person ?maxSalaire WHERE {
    ?person f:Sal ?maxSalaire .
    {
        SELECT  (MAX(?salaire) AS ?maxSalaire)
        WHERE {
            ?person f:Sal ?salaire
        }
    }
}
```

Résultat attendu

```
------------------------
| person  | maxSalaire |
========================
| :robert | "48"       |
------------------------
```

Q6.b : Le plus grand salaire . Retourner l'employé et son salaire.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>

SELECT ?person ?salaire
WHERE {
    ?person f:Sal ?salaire .
} ORDER BY DESC (?salaire) LIMIT 1
```

Résultat attendu

```
---------------------
| person  | salaire |
=====================
| :robert | "48"    |
---------------------
```

Q7 : Les couples d'employés qui gagnent le même salaire (chaque couple d'employés doit apparaître une seule fois). Indice : utiliser la

fonction str(?v) pour extraire la chaîne de caractères à partir de ?v.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>

SELECT DISTINCT ?emp1 ?emp2 ?salaire1 ?salaire2
WHERE {
    ?emp1 a f:emp .
    ?emp2 a f:emp .
    ?emp1 f:Sal ?salaire1 .
    ?emp2 f:Sal ?salaire2 .
    FILTER (?salaire1 = ?salaire2)
 # Pour eviter les couples doublons on va utiliser une comparaison au lieu d'une inégalité
    FILTER (str(?emp2) > str(?emp1))
}
```

Résultat attendu

```
--------------------------------------
| emp1 | emp2  | salaire1 | salaire2 |
======================================
| :dan | :john | "33"     | "33"     |
--------------------------------------
```

Q8 : Les couples d'employés tels que la différence entre leurs salaires est supérieure à 5. Indice : utiliser xs:integer(?l) pour convertir le literal ?l en entier. Ajouter le préfixe prefix xs: http://www.w3.org/2001/XMLSchema#

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>
PREFIX xs: <http://www.w3.org/2001/XMLSchema#>
PREFIX fn:  <http://www.w3.org/2005/xpath-functions#>

SELECT ?emp1 ?salaire1 ?emp2 ?salaire2
WHERE {
    ?emp1 a f:emp .
    ?emp2 a f:emp .
    ?emp1 f:Sal ?salaire1 .
    ?emp2 f:Sal ?salaire2
```

```
    FILTER (?emp1 != ?emp2)
 # On n'a pas besoin de la valeur absolue parce que sinon on va avoir des couples (a,b) et (b,
    FILTER (xs:integer(?salaire1) - xs:integer(?salaire2) > 5)
}
```

Résultat attendu

```
----------------------------------------
| emp1    | salaire1 | emp2  | salaire2 |
========================================
| :liz    | "42"     | :dan  | "33"     |
| :liz    | "42"     | :john | "33"     |
| :robert | "48"     | :dan  | "33"     |
| :robert | "48"     | :john | "33"     |
| :robert | "48"     | :liz  | "42"     |
----------------------------------------
```

Q9 : Le nombre de départements.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX f: <http://www.cems.uwe.ac.uk/empdept/concept/>
PREFIX : <http://www.abc.org/>
SELECT COUNT(DISTINCT ?deptName)
WHERE {
    ?pname f:Dept ?deptName
}
```

Résultat attendu

```
------
| .1 |
======
| 4  |
------
```