

TP - Séance 2 a

Interrogation RDF avec SPARQL

Téléchargez et décompressez les fichiers de données (et de requêtes) joints à ce TP.
La commande `sparql` de Jena permet d'évaluer des requêtes SPARQL :

```
1 sparql --data=<fichier.rdf> --query=<fichier.sparql>
```

- `<fichier.rdf>` indique le fichier interrogé
- `<fichier.sparql>` contient la requête SPARQL à exécuter.

1 Interrogation SPARQL

Exercice 1 Affichez et évaluez la requête `queries/rois/q0.sparql` sur le document `data/roisdbpedia.ttl`.

Exercice 2 Définissez et évaluez les requêtes SPARQL suivantes sur le graphe RDF `data/roisdbpedia.ttl` :

1. Les noms des rois avec leur date de naissance.
2. Les noms des rois avec optionnellement, si présente, leur date de naissance.
3. Les noms des reines mariées avec les noms des époux.
4. Les noms des reines mariées qui sont nées la même année que leurs époux, avec les noms de leurs époux (il faut utiliser la fonction `year`).
5. Les noms des rois sans fils (deux solutions).
6. Expliquez le résultat de la requête suivante :

```
1 PREFIX asws: <http://www.asws.com/rois#>  
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
3 SELECT ?n  
WHERE { { ?a asws:nom ?n } MINUS { ?b asws:fils ?f } }
```

7. Les noms des rois sans enfants (ni fils, ni filles, ni enfants).

Exercice 3 SPARQL permet d'interroger n'importe quelle base de triplets RDF, y compris les triplets de schéma RDFS. Définissez et évaluez les requêtes SPARQL suivantes sur le schéma RDFS `data/rois_schema.ttl` :

1. Toutes les propriétés du schéma.
2. Tous les couples (classe, sous-classe).
3. Toutes les classes, en spécifiant le graphe source dans la partie FROM de la requête. Remarquer que le chemin du graphe source indiqué dans la partie FROM est considéré par défaut relatif au répertoire du fichier de la requête. Pour changer la base de ce chemin utiliser l'option `--base = chemin` de la commande `sparql`.

2 Interrogation de plusieurs graphes

Se rappeler qu'une requête SPARQL peut être évaluée sur plusieurs sources de données (plusieurs parties FROM), nommées ou pas nommées.

Exercice 4 Expliquez la réponse de la requête `queries/rois/q1.sparql`. Que peut-on conclure ? En particulier est-ce que l'évaluation faite par la commande `sparql` tient compte des inférences ?

Exercice 5 Expliquez la réponse de la requête `queries/rois/q1named.sparql`.

Exercice 6 Définissez et évaluez les requêtes SPARQL suivantes sur le graphe RDF `data/roisdbpedia.ttl` et le schéma RDFS `data/rois_schema.ttl` :

1. Toutes les instances des sous-classes de la classe `asws:Homme`.
2. Tous les triplets (`?a ?p ?b`) apparaissant dans l'un des deux graphes (`data/roisdbpedia.ttl` ou `data/rois_schema.ttl`, comme graphes nommés) tels que `?b` apparait comme sujet dans l'autre graphe.

3 Expressions de chemins

Exercice 7 Dans cet exercice on utilisera un nouveau graphe RDF, `data/succdbpedia.ttl`, dont les triplets sont un extrait des relations de succession entre monarques de France. Ces triplets utilisent le prédicat `:successeur`. Le triplet `:a :successeur :b` indique que le monarque `:b` a été un successeur direct de `:a`. (Remarquer que parfois un même monarque a plusieurs successeurs, par exemple quand il a été monarque de plusieurs royaumes).

Définissez et évaluez les requêtes SPARQL suivantes sur les graphes RDF `data/roisdbpedia.ttl` et `data/succdbpedia.ttl` et le schéma RDFS `data/rois_schema.ttl` :

1. Tous les (URIs des) successeurs directs et indirects du monarque de nom "Louis XII". La réponse pourra être facilement vérifiée en visualisant le graphe `data/succdbpedia.ttl`.
2. Les URIs des "personnes". Les "personnes" sont toutes les ressources appartenant à une sous-classe directe ou indirecte de la classe `asws:Personne`. Écrivez la requête de telle sorte que son corps (la partie WHERE) contienne un seul triplet et ne contienne aucune ressource anonyme.

NB. Se rappeler que la commande `sparql` répond à une requête en tenant compte uniquement des triplets explicitement présents dans les sources, aucun triplet n'est inféré par les contraintes RDFS. Il faudra donc reconstruire avec la requête la chaîne des propriétés de sous-classe.

NB. La requête est à évaluer sur les graphes `data/roisdbpedia.ttl` et `data/rois_schema.ttl`. Veillez à écrire une requête qui calcule la bonne réponse pour n'importe quel contenu de ces deux documents RDF/S.

3. Les paires de "personnes" mariées (URIs) (cf. ci-dessus pour la définition de "personnes"). Le conjoint d'une personne est représenté soit par le prédicat `:epoux` soit par le prédicat `:epouse`. Pour chaque personne `n` avec un conjoint `m` la requête doit retourner à la fois `n m` et `m n` (paires de URIs). Tenir compte du fait qu'on pourrait avoir par exemple `:n :epouse :m`, dans les données, sans avoir `:m :epoux :n`. Écrire la requête d'abord sans UNION puis avec UNION.
4. Les couples de "personnes" (URIs) parent - enfant (cf. ci-dessus pour la définition de "personnes"). Remarquer qu'un enfant peut être relié à son parent par les prédicats `:fils` ou `fille` ou `enfant`, ou à l'inverse par les prédicats `:pere`, `:mere` ou `:parent`.

5. Les couples de "personnes" (URIs) ancêtre - descendant (cf. ci-dessus pour la définition de "personnes"). Un descendant est un enfant ou un enfant d'un enfant etc, etc, sans limite ; on n'est pas descendant de soi même. Les deux URIs retournés par la requête doivent être parmi les "personnes", mais cette restriction ne s'applique pas à la chaîne de descendance.

Question : Est-ce que SPARQL (sans inférence) peut exprimer la variante de la requête dans laquelle on demande que toute la chaîne de descendance soit faite de "personnes" ?

Exercice 8 Requêtes d'agrégation

1. Le nombre de successeurs par roi ou reine.
2. Le nombre d'enfants par roi ou reine.
3. Le nombre d'époux par reine.
4. Le nombre de roi et de reines par dynastie.
5. Le nombre de descendants et d'ancêtres par roi ou reine.