

Examen MIF04 - Gestion de données pour le Web - session 1 - 7 janvier 2020

Durée : 2h

Documents autorisés

Exercice 1:

(6 points)

On considère le graphe RDF suivant (représenté graphiquement dans la figure 1) :

```
1 @prefix p: <http://example.com/rdf/p#> .
2 @prefix b: <http://example.com/rdf/b#> .
3 @prefix s: <http://example.com/rdf/s#> .
4 @prefix rel: <http://example.com/rdf/rel#> .
5 @prefix t: <http://example.com/rdf/t#> .
6 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
7 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
8
9 p:david rel:auteur b:triumph;
10   rel:nom "David Brin".
11 p:greg rel:auteur b:chaos;
12   rel:nom "Greg Bear".
13 p:benford rel:auteur b:fear;
14   rel:nom "Gregory Benford".
15
16 b:triumph rel:annee "1999";
17   rel:titre "Foundation's Triumph";
18   rel:serie s:sndfund.
19 b:chaos rel:annee "1998";
20   rel:titre "Foundation and Chaos";
21   rel:serie s:sndfund.
22 b:fear rel:annee "1997";
23   rel:titre "Foundation's Fear";
24   rel:serie s:sndfund.
25
26 s:sndfund rel:titre "The Second Foundation";
27   rel:serie s:foundation.
28
29 rel:titre rdfs:subPropertyOf rdfs:label.
30 rel:nom rdfs:subPropertyOf rdfs:label;
31   rdfs:domain t:personne.
32 rel:auteur rdfs:range t:livre.
```

Dans la suite de l'exercice, on supposera que les préfixes déclarés ci-dessus sont prédéfinis pour toutes les requêtes.

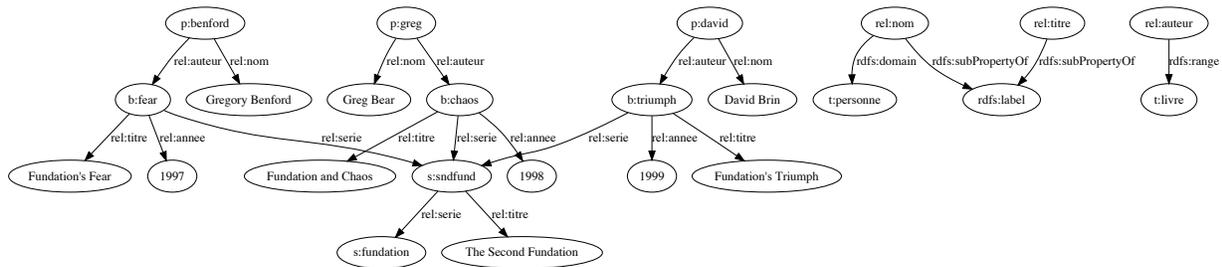


FIGURE 1 – représentation graphique

On considère les règles de déduction suivantes :

$$\frac{S P O \text{ et } P \text{ rdfs:domain } T}{S \text{ rdf:type } T} \text{ (Domain)}$$

$$\frac{S P O \text{ et } P \text{ rdfs:range } T}{O \text{ rdf:type } T} \text{ (Range)}$$

$$\frac{X \text{ rdf:type } C \text{ et } C \text{ rdfs:subClassOf } D}{X \text{ rdf:type } D} \text{ (SC)}$$

$$\frac{S P O \text{ et } P \text{ rdfs:subPropertyOf } Q}{S Q O} \text{ (SP)}$$

$$\frac{B \text{ rel:serie } S1 \text{ et } S1 \text{ rel:serie } S2}{B \text{ rel:serie } S2} \text{ (RSerie)}$$

Par la suite, on nommera TD (comme “triplets déduits”) l’ensemble des triplets du graphe saturé qui ne sont pas dans le graphe de départ.

1. Donner l’ensemble des **prédicats** des triplets de TD .
2. Donner la liste des triplets qui peuvent être déduits par la règle (SP) .
3. Donner la taille de TD .
4. Soit la requête SPARQL suivante :

```
SELECT ?b WHERE {
  ?b rdf:type t:livre.
  ?b rel:serie s:fundation.
}
```

Donner le résultat de l’exécution de cette requête sur le graphe saturé.

5. Donner un triplet tel que si on l’ajoute au graphe initial, on puisse déduire les triplets suivants :

```
s:sndfund rdf:type t:serie.
s:fundation rdf:type t:serie.
```

Exercice 2:

(9 points)

On considère une collection `rayons` dans MongoDB. Dans cette collection, chaque document représente le contenu du rayon d'un supermarché sous forme d'une liste de produits. Un exemple de tel document est donné ci-dessous :

```
1 {
2   "_id": "1234",
3   "allée": "3",
4   "produits": [
5     {
6       "code": "5678-9012",
7       "prix": 5.2,
8       "quantite": 12,
9       "emplacement": 18,
10      "categorie": "conserve"
11    },
12    {
13      "code": "3456-7890",
14      "prix": 7.1,
15      "quantite": 35,
16      "emplacement": 5,
17      "categorie": "sucre"
18    },
19    /* ... */
20  ]
21 }
```

Soient les deux fonctions `map1` et `reduce1` suivantes :

```
1 var map1 = function() {
2   var s = 0;
3   for(var i in this.produits) {
4     if (this.produits[i].categorie === "conserve") {
5       s = s + 1;
6     }
7   }
8   emit(1, s);
9 }
10
11 var reduce1 = function(k, values) {
12   var s = 0;
13   for(var i = 0; i < values.length; i++) {
14     s = s + values[i];
15   }
16   return s;
17 }
```

1. Expliquer *en une phrase* ce qui est calculé par le *job* lancé par la commande :
`db.rayons.mapReduce(map1, reduce1, {out : {inline : 1}})`

2. Écrire les fonctions `map2`, `reduce2` et `finalize2` correspondant à un *job* qui donne, pour chaque catégorie, la différence entre le prix le plus élevé et le prix le plus faible parmi tous les produits de cette catégorie. On rappelle l'existence du phénomène de *re-reduce* et le fait que la fonction `finalize2` sera appelée une fois par valeur de clé, après exécution des `reduce2`. Ce *job* sera lancé avec la commande

```
db.rayons.mapReduce(map2, reduce2,
                    {out : {inline : 1}, finalize : finalize2})
```

3. Suite à certaines erreurs de saisie, certains produits se retrouvent dans plusieurs catégories. On souhaite obtenir les références (code) de ces produits, ainsi que les allées dans lesquels ils apparaissent. Écrire les fonctions `map3`, `reduce3` et `finalize3` correspondant à un *job* permettant de connaître les produits concernés. Ce *job* associera la valeur `undefined` aux produits n'ayant qu'une catégorie. Il associera le tableau de leurs allées aux produits ayant différentes catégories. Le code devra fonctionner même en présence de *re-reduce*. Ce *job* sera lancé avec la commande

```
db.rayons.mapReduce(map3, reduce3,
                    {out : {inline : 1}, finalize : finalize3})
```

Exercice 3:

(5 points)

On considère une collection `trafic` contenant les mesures des capteurs de circulation routière. Chaque document de la collection correspond à une mesure d'un capteur. Un exemple de tel document est donné ci-dessous :

```
1 {
2   "_id": "123456",
3   "capteur": "rd463-18",
4   "ts": 1577881235,
5   "speed": 65
6 }
```

Le champ `ts` est l'heure à laquelle la mesure est prise, donnée en nombre de secondes écoulées depuis le 1^{er} janvier 1970 à 00h00. Le champ `speed` donne la vitesse de passage du véhicule en km/h.

Écrire les fonctions `mapt`, `reducet` et `finalizet` permettant pour chaque tranche de 5 minutes de connaître quel capteur a enregistré la vitesse moyenne la plus élevée.

Remarques :

- En cas d'égalité, on pourra choisir arbitrairement le capteur renvoyé (parmi ceux ayant enregistré la vitesse moyenne la plus élevée).
- Une tranche de 5 minutes est identifiée par son heure de début (au même format que le champ `ts`). La fonction `tranche_5_min` ci-dessous peut être utilisée pour calculer la tranche d'une heure donnée :

```
var tranche_5_min = function(h) { return h - h % (5*60); }
```