CHAMROUK Laïla
MAJDOUL Kaoutar
M2 GL

HAI914I-NoSQL : Projet Rendu n°2

Lien du dépôt :
https://github.com/lailachm/NoSQL_HAI914I/tree/main/MiniProjet/qengine-master

Le dictionnaire et les indexs

Le dictionnaire et les indexs sont créés dans le fichier `MainRDFHandler.java`. Pour ce deuxième rendu, nous avons réutiliser les résultats du premier rendu afin d'accéder aux données, et de résoudre les requêtes.

L'accès aux données

Le travail effectué pour ce rendu a permis de manipuler les données du dictionnaire, et d'utiliser les indexs afin de répondre aux requêtes. Pour cela, nous avons travaillé dans le fichier `Main.java`, plus particulièrement dans la fonction `processAQuery()` qui permet le traitement d'une requête.

Nous appelons dans un premier temps la fonction `parseData()` qui permet de récupérer le dictionnaire et l'index sur lesquels nous allons évaluer les requêtes.

```
try {
    parseData();
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Pour chaque élément du dictionnaire, nous allons comparer les objets et prédicats du statement de la requête que nous étudions. Si l'objet/le prédicat du statement comparé est dans le dictionnaire, on l'ajoute à la liste correspondante que nous utiliserons ensuite pour la comparaison à l'index.
Cela permet de s'assurer que la requête est bien conforme au dictionnaire.

CHAMROUK Laïla
MAJDOUL Kaoutar
M2 GL

```
for (Integer element : dico.keySet()) { // pour chaque élément du dictionnaire
    int keyDico = element;
    String valueDico = dico.get(element);

    int numOfStatement = 0;
    for (StatementPattern pattern : patterns) { // pour chaque statement de la requête
        String predicateInQuery = pattern.getPredicateVar().getValue().toString();
        String objectInQuery = pattern.getObjectVar().getValue().toString();
        if (predicateInQuery.equals(valueDico)) {// si le prédicat apparaît dans le dico
            listOfPredicatesInQueryAndInDico.put(numOfStatement, keyDico);// on ajoute à la liste correspondante
                                                                          // la clé du dico qui correspond à
                                                                          // la valeur du prédicat
        }
        if (objectInQuery.equals(valueDico)) {
            listOfObjectsInQueryAndInDico.put(numOfStatement, keyDico);
        }
        numOfStatement++;

    }
}
```

Ensuite, nous comparons les objets et prédicats de chaque statements à l'index SPO. Nous avons choisi l'index SPO car il nous permettait d'accéder à chaque valeur plus facilement et de façon plus logique. Ainsi, pour chaque statement, si l'objet O et le prédicat P sont un index de l'indexSPO, on ajoute le sujet S à une liste qui contiendra tous les sujets répondant au statement de la requête.

```
while (keyForLists < patterns.size()) {
    int predicateInStatement = 9999999; // le prédicat garde cette valeur s'il n'apparaît pas dans le dico
    if (listOfPredicatesInQueryAndInDico.get(keyForLists) != null) {// au statement n°keyForLists, si le
                                                                    // prédicat existe, il instancie la variable
                                                                    // qui sera comparé à l'index
        predicateInStatement = listOfPredicatesInQueryAndInDico.get(keyForLists);
    }
    int objectInStatement = 9999999;
    if (listOfObjectsInQueryAndInDico.get(keyForLists) != null) {
        objectInStatement = listOfObjectsInQueryAndInDico.get(keyForLists);
    }

    for (HashMap<Integer, Integer> bigKeyIndex : indexSPO.keySet()) {// indexSPO est de la forme
                                                                     // {subject=predicate}=object.
        int object = indexSPO.get(bigKeyIndex);// dans ce contexte, bigKeyIndex(clé) est {subject=predicate}, et
                                               // sa valeur est object
        for (Integer smallKeyIndex : bigKeyIndex.keySet()) {
            int predicate = bigKeyIndex.get(smallKeyIndex);
            int subject = smallKeyIndex;

            /*
             * Pour chaque index, si l'objet et le prédicat de l'index match l'objet et le
             * prédicat du statement, on ajoute le sujet à la liste
             * subjectsThatAnswerToStatementInQuery
             */
            if (objectInStatement != 9999999 && predicateInStatement != 9999999) {
                if (objectInStatement == object && predicateInStatement == predicate) {
                    subjectsThatAnswerToStatementInQuery.add(subject);
                }
            }

        }
    }
    keyForLists++;
}
```

CHAMROUK Laïla
MAJDOUL Kaoutar
M2 GL

Ce n'est pas encore terminé puisqu'il faut maintenant obtenir une liste contenant les sujets répondant à la requête entière, c'est-à-dire à tous les statements de la requête.
Pour cela, nous reprenons la liste précédente contenant les sujets qui répondent aux statements. Ainsi, si le sujet répond à tous les statements, il apparaît autant de fois dans la liste qu'il y a de statement dans la requête étudiée.
Les sujets répondant à la requête entière sont ensuite ajoutés à une nouvelle liste et affichés à l'aide d'un print.

```
LinkedHashSet<Integer> subjectsThatAnswerToQuery = new LinkedHashSet<Integer>();
for (int subject : subjectsThatAnswerToStatementInQuery) {
    int occurences = Collections.frequency(subjectsThatAnswerToStatementInQuery, subject);
    if (occurences == patterns.size()) { // si le sujet apparaît dans la liste autant de fois qu'il y a de
                                         // statements, c'est qu'il répond à la requête
        subjectsThatAnswerToQuery.add(subject);
    }
}

System.out.println("Les sujets qui répondent à tous les statements : " + subjectsThatAnswerToQuery);

for (int s : subjectsThatAnswerToQuery) {
    for (int key : dico.keySet()) {
        if (s == key) {
            System.out.println("Subject answering the query (match in dico) : " + dico.get(key));
        }
    }
}
```

Un exemple d'exécution :

Requête

```
SELECT ?v0 WHERE {
    ?v0 <http://schema.org/birthDate> "1988-09-24" .
    ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/userId> "9764726" . }
```

Résultat

```
******QUERY N°3******
-----------------------------
Le dictionnaire est le suivant : {0=http://db.uwaterloo.ca/~galuc/wsdbm/User0, 1=http://schema.org/birthDate, 2="1988-09-24", 3=http://db.uwaterloo.ca/~galuc/wsdbm/userId, 4="976
-----------------------------

-----------------------------
L'index SPO : {{0=1}=2, {0=3}=4, {7=3}=8, {5=3}=6, {9=1}=10, {9=3}=11, {15=3}=16, {12=1}=13, {12=3}=14, {19=3}=20, {17=3}=18, {23=3}=24, {21=3}=22, {25=1}=2, {26=3}=4, {26=1}=2}
-----------------------------

Les objets qui apparaissent dans la requête et le dico : {0=2, 1=4}
Les prédicats qui apparaissent dans la requête et le dico : {0=1, 1=3}
Les sujets qui répondent aux statements de la requête : [0, 25, 26, 0, 26]
Les sujets qui répondent à tous les statements : [0, 26]
Subject answering the query (match in dico) : http://db.uwaterloo.ca/~galuc/wsdbm/User0
Subject answering the query (match in dico) : http://db.uwaterloo.ca/~galuc/wsdbm/User11
******END QUERY N°3******
```