



Contrôle terminal de BDBIO - session 1

UCBL - Département informatique (2021/2022)

Pour assurer l'anonymat, ne notez pas votre nom ou numéro étudiant sur la copie.
Documents papier autorisés. Durée : 1h30.
Remplissez les cases sur la dernière feuille, de préférence au stylo noir. Les questions avec un symbole ♣ ont 0, 1 ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Panique à château Rocfort ! Des animaux volent régulièrement des fromages, et les habitants ont décidé de recenser ces vols dans les différentes pièces. Évidemment, les occupants ne s'étant pas concertés au préalable, vous récupérez, en tant que responsable de la sécurité du château, différentes bases de données...

Une première source de données relationnelles recense les vols perpétrés par des souris. Ci-dessous son schéma et les données de la table VOLS :

FROMAGES (nomF, lait, forme, affinage)
SOURIS (idS, nomS)
VOLS (#nomF, #idS, #nomL, date, quantité)
LIEUX (nomL, surface, étage)

nomF	idS	nomL	date	quantité
Comté	1	cave	02/02/2022	25
Raclette	1	cuisine	21/02/2022	30
Raclette	2	cuisine	27/02/2022	20
Tome des Bauges	3	donjon	27/02/2022	15
Tomme de brebis	4	cave	28/02/2022	10
Camembert fermier	5	donjon	02/03/2022	15

Une seconde source contient des données au format JSON, la troisième source en XML recense aussi des vols commis par des souris et la dernière source (graphe RDF) liste des vols qui ont eu lieu dans la cuisine.

```

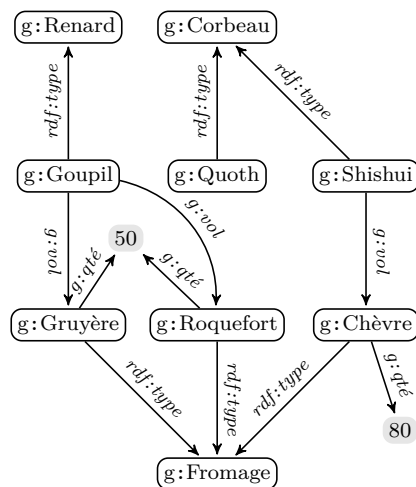
28 {
29   "vols": [
30     {
31       "_id": "Croutard (rat)",
32       "caractéristiques": "double personnalité",
33       "vols_cave": [{"objet": "Chèvre",
34                     "type": "fromage",
35                     "qté": 45
36                   }],
37       "vols_cuisine": [{"objet": "Pomme",
38                          "type": "fruit",
39                          "qté": 1
40                        }],
41     },
42     {
43       "_id": "Rémy (rat)",
44       "caractéristiques": "odorat fin",
45       "vols_donjon": [{"objet": "Comté",
46                        "type": "fromage",
47                        "qté": 40
48                      },
49                       {"objet": "Raclette",
50                          "type": "fromage",
51                          "qté": 80
52                        }],
53     },
54     {
55       "_id": "Biquette (chèvre)",
56       "vols_cave": [{"objet": "Brie",
57                      "type": "fromage",
58                      "qté": 150
59                    }],
60     }
61   ]

```

```

13 <stats>
14 <vols>
15   <vol fromage="1" qté="10" lieu="cave" />
16   <vol fromage="2" qté="15" lieu="cuisine" />
17   <vol fromage="2" qté="20" lieu="cuisine" />
18 </vols>
19 <fromages>
20   <fromage id="1">Séchon</fromage>
21   <fromage id="2">Pélardon</fromage>
22 </fromages>
23 </stats>

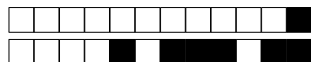
```



Exercice 1 – Compréhension des sources (6 points)

Afin de vous familiariser avec les sources, répondez aux questions suivantes.

Question 1 En relationnel, deux lieux avec le même nom (e.g., chambre) peuvent-ils se trouver à 2 étages différents ?



A Oui Non

Question 2 En relationnel, est-ce qu'une souris peut voler le même jour et dans le même lieu deux quantités différentes d'un même fromage ?

A Oui Non

Question 3 Dans le diagramme E/A qui produirait le schéma relationnel fourni, comment sont reliées les entités dérivées des relations SOURIS et FROMAGES ?

- A Association réflexive
- B Association binaire sans cardinalité maximale en 1
- C Association binaire avec cardinalité maximale en 1
- D Spécialisation
- E Association ternaire sans cardinalité maximale en 1
- F Association ternaire avec cardinalité maximale en 1
- G Entité faible
- H Autre réponse

Question 4 Dans la source RDF, quel problème de modélisation identifiez-vous ?

Question 5 Dans la source JSON, quel problème de modélisation identifiez-vous ?

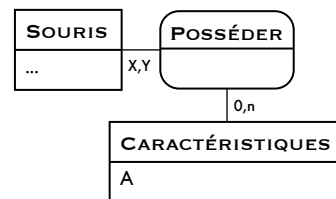
Question 6 Combien de tuples sont retournés par la requête suivante sur les données relationnelles ?
SELECT DISTINCT nomF, nomL FROM VOLS WHERE nomL LIKE 'c_%e';

- A 6
- B 1
- C 5
- D 2
- E 3
- F 0
- G 4
- H Autre réponse

Question 7 ♣ Quelles contraintes sont valides sur le document XML ?

- A <!ATTLIST vol qté (xsd:integer)>
- B <!ELEMENT fromage (#PCDATA)>
- C <!ATTLIST vol fromage ID>
- D <!ELEMENT stats (vols, fromage)>
- E <!ATTLIST fromage id CDATA>
- F Aucune de ces réponses n'est correcte

Vous souhaitez ajouter des informations dans la source relationnelle en repartant du diagramme E/A. Une souris peut posséder des caractéristiques représentées par un nom (e.g., couleur des poils, type de personnalité). Il peut être nécessaire de stocker plusieurs valeurs pour la même caractéristique (e.g., un pelage noir et blanc). Répondez aux questions suivantes en utilisant l'extrait de diagramme ci-contre.



Question 8 Que mettez-vous dans CARACTÉRISTIQUES à la place de A ?

- A nomC #idS
- B nomC #idS
- C nomC
- D nomC idS
- E nomC
- F nomC idS
- G Aucune réponse n'est correcte

Question 9 Quelle cardinalité mettez-vous pour X,Y ?



- 0,n C (1,n) E (0,1) G (0,n) I *Aucune réponse n'est correcte*
 B 1,1 D 1,n F 0,1 H (1,1)

Question 10 Où faut-il mettre la valeur de la caractéristique ?

- A Dans SOURIS C Dans CARACTÉRISTIQUES E Nulle part, elle est implicite
 B Dans POSSÉDER D Dans une autre association *Aucune réponse n'est correcte*

Exercice 2 – Interrogation des sources (6 points)

Écrivez les requêtes suivantes dans le langage demandé sur la source correspondante.

Question 11 Le nom des souris qui n'ont jamais volé de fromage ni à la cave ni à l'étage 1 (SQL). Le résultat sera trié par ordre alphabétique.

Question 12 Pour chaque (nom de) fromage dont on a volé plus de 100 grammes, affichez le nombre de lieux différents où ce fromage a été chapardé (SQL).

Question 13 Les animaux qui ont volé deux variétés différentes de fromage, mais en même quantité (SPARQL).

Question 14 Pour chaque lieu, donnez la liste des fromages volés et la quantité totale subtilisée (XQUERY). Le résultat sera de la forme `<lieu nom="cuisine" qté="35"><f>Séchon</f><f>Pélardon</f></lieu>`. La fonction `sum(val1, val2, ...)` permet de calculer la somme des paramètres `val1, val2, etc.`

Exercice 3 – Intégration de données (8 points)

Pour faciliter l'utilisation des données, vous les intégrez en utilisant le document JSON comme BD centralisée. Pour répondre aux questions suivantes, vous utiliserez les langages de requête appropriés pour extraire les données d'une source et pour peupler la BD globale. Le reste du programme sera codé en pseudo-langage (syntaxe libre, mais suffisamment explicite pour que le programme soit implémentable). Pour le niveau de détail, utilisez des appels de fonctions pour simplifier le code (e.g., si vous devez trier un tableau, écrivez `tabSorted = sort(tab)` accompagné d'un commentaire mais n'écrivez pas un algorithme complet de tri de tableau!). Utilisez des commentaires, par exemple pour expliquer comment vous résolvez les conflits.

Question 15 Écrivez tout d'abord un programme informatique pour intégrer les données relationnelles dans le document JSON.

Question 16 Écrivez ensuite un programme informatique pour intégrer les données RDF dans le document JSON.

Exercice 4 – Bonus (1 point)

Question 17 ♣ Parmi ces propositions, lesquelles sont des vrais noms de fromages ?



+1/4/57+

Sablé de Wissant

Vieux-Lille

Poivre d'âne

Amour de Nuits

Bleu de Gex

Question 18 Dans quelle pièce la plus grande quantité de fromages a-t-elle été volée ?

Cuisine

B Donjon

C Oubliettes

D Cave

Question 19 Quelle espèce a volé la plus grande quantité de fromages ?

A Chèvre

B Souris

C Corbeau

D Renard

Rat



Vous rendez uniquement les feuilles suivantes, sur lesquelles vous **remplissez** les cases pour répondre.

Question 1 : A

Question 2 : A

Question 3 : A B C D E F G

Question 4 : 0 0.25 0.5 *Reservé*

Un fromage possède une quantité fixe, et un vol (prédicat) concerne directement un fromage, donc impossible de voler une quantité différente de ce fromage. Il faudrait qu'un vol soit une ressource, liée au voleur, au fromage et possédant une quantité.

Question 5 : 0 0.25 0.5 *Reservé*

Les propriétés concernant les vols incluent le nom du lieu (e.g., "vols_cave", "vols_cuisine"), ce qui ne permet pas de requêter facilement les documents. Le lieu devrait être une propriété des sous documents décrivant chaque vol. De plus, l'espèce est stockée avec le nom comme valeur, deux propriétés auraient été préférables.

Question 6 : A B C D F G H

Question 7 : A C D E F

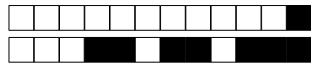
Question 8 : A B D E F G

Question 9 : B C D E F G H I

Question 10 : A B C D E

Question 11 : 0 0.25 0.5 0.75 1 1.25 *Reservé*

```
SELECT nomS FROM SOURIS
WHERE idS NOT IN (SELECT idS FROM VOLS v NATURAL JOIN LIEUX l
WHERE étage = 1 OR l.nomL = 'cave')
ORDER BY nomS DESC;
```



Question 12 :

0	0.25	0.5	0.75	1	1.25	Reservé
---	------	-----	------	---	------	---------

```
SELECT nomF, COUNT(DISTINCT nomL) AS nb
FROM VOLS
GROUP BY nomF
HAVING SUM(quantité) > 100;
```

Question 13 :

0	0.25	0.5	0.75	1	1.25	Reservé
---	------	-----	------	---	------	---------

```
SELECT DISTINCT ?a
WHERE { ?a g :vol ?f1 .
        ?a g :vol ?f2 .
        ?f1 g :qté ?q1 .
        ?f2 g :qté ?q2 .
        FILTER(?q1 = ?q2 && ?f1 != ?f2)
}
```

Question 14 :

0	0.25	0.5	0.75	1	1.25	Reservé
---	------	-----	------	---	------	---------

```
for $v in //vol
  let $lieu := $v/@lieu
  let $q := fn :sum($v/@qté)
  group by $lieu
  return <lieu nom="{@$lieu}" qté="{@$q}" >
    {
      for $fid in $v/@fromage
      let $f := //fromage[@id=$fid]/text()
      return <f>{$f}</f>
    }
</lieu>
```



Question 15 :

0

0.5

1

1.5

2

2.5

3

3.5

Reservé

```
6 json = connexion(source2.json)
7 bdrel = connexion(source1)
8
9 req = "select idS, nomS from Souris group by idS, nomS"
10 pour s dans bdrel.executer(req):
11     nom = s['nomS'] + " (souris)"
12     dict = {_id : nom}
13     if nom in json: # la souris existe dans le doc, on récupère son doc
14         dict = json.vols.get(nom)
15     req2 = "select nomF, nomL, SUM(quantité) AS qté from Vols where idS={s['idS']}
16           group by nomF, nomL"
17     vols = bdrel.executer(req2)
18     pour v dans vols :
19         prop_vol_lieu = "vols_" + vols
20         if not dict[prop_vol_lieu]:
21             dict[prop_vol_lieu] = []
22         dict[prop_vol_lieu].append({
23             objet: v['nomF'],
24             type: "fromage",
25             qté: v['qté']
26         })
27     json.vols.update(dict) # ajout ou update du dict de la souris
```



Question 16 :

0

0.5

1

1.5

2

2.5

3

3.5



Réservé

```
31 json = connexion(source2.json)
32 bdrdf = connexion(source4)
33
34 # insertion des animaux
35 req = "SELECT ?animal ?espece
36 WHERE { ?animal rdf:type ?esp .
37 }"
38 resultats = bdrdf.executer(req)
39 pour res dans resultats:
40   nom_esp = res['animal'] + " (" + res['espece'] + ")"
41   if not json.get(nom_esp): # animal inexistant, insertion
42     dict = {"_id": nom_esp}
43     json.vols.update(dict)
44 # insertion des vols
45 req = "SELECT ?animal ?espece ?fromage ?qte
46 WHERE { ?animal rdf:type ?esp .
47   ?animal g:vol ?fromage.
48   ?fromage rdf:type g:Fromage .
49   ?fromage g:qté ?qte .
50 }"
51 resultats = bdrdf.executer(req)
52 pour res dans resultats:
53   nom_esp = res['animal'] + " (" + res['espece'] + ")"
54   dict = json.get(nom_esp)
55   if not dict["vols_cuisine"]:
56     dict["vols_cuisine"] = []
57   dict["vols_cuisine"].append({"objet": res['fromage'],
58     "type": "fromage",
59     "qté": res['qte']
60 })
61 json.vols.update(dict)
```

Question 17 : Question 18 : B C DQuestion 19 : A B C D