

# Contrôle terminal de BDBIO - session 1

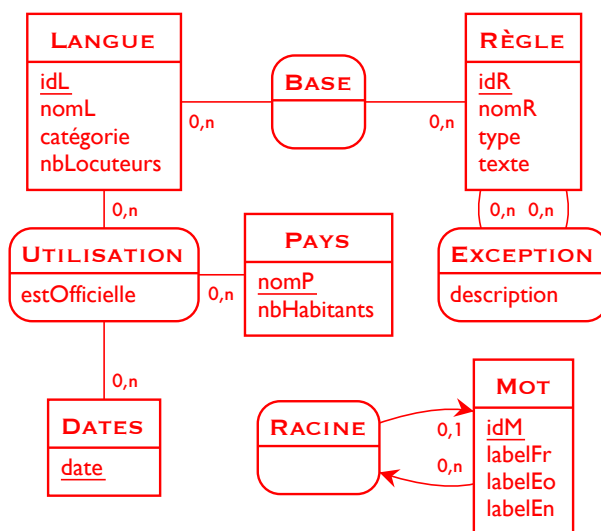
UCBL - Département informatique (2020/2021)

Documents papier autorisés. Anonymisez votre copie. Durée : 1h30.

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Remplissez les cases sur la dernière feuille, de préférence au stylo noir.

Vous rejoignez l'organisation Babelia spécialisée dans les traductions de texte, notamment en français (*fr*), anglais (*en*) et espéranto (*eo*). Plusieurs bases de données sont utilisées dans cette organisation, et votre mission est de faciliter les traductions en intégrant les données dans une même base.



Une première source de données est en relationnel, dont voici le schéma et quelques données :

LANGUE	( <u>idL</u> , <i>nomL</i> , <i>catégorie</i> , <i>nbLocuteurs</i> )
BASE	(# <u>idL</u> , # <u>idR</u> )
RÈGLE	( <u>idR</u> , <i>nomR</i> , <i>type</i> , <i>texte</i> )
UTILISATION	(# <u>nomP</u> , # <u>idL</u> , <i>date</i> , <i>estOfficielle</i> )
PAYS	( <u>nomP</u> , <i>nbHabitants</i> )
EXCEPTION	(# <u>idR1</u> , # <u>idR2</u> , <i>description</i> )
MOT	( <u>idM</u> , <i>labelFr</i> , <i>labelEo</i> , <i>labelEn</i> , # <u>idMracine</u> )

idM	labelFr	labelEo	labelEn	idMrac
1	pouvoir	povi	to can	
2	rien	nenio		
3	nulle part	nenie		
4		kato	cat	
5		kateto	little cat	
6		kataro	group of cats	
7	changer	ŝanĝi		

TABLE MOT

Une seconde source contient des données au format JSON (exemples de phrases traduites dans différentes langues) tandis que la troisième contient des données RDF sérialisées en XML (règles de conjugaison, de grammaire, etc.).

## CORRECTION

```
37 {
38   "exemples": [
39     {
40       "_id": 1,
41       "phrases": ["(eo) persono pensas pri la mondo",
42                  "(fr) une personne réfléchit sur le monde"],
43     },
44     {"_id": 2,
45      "phrases": ["(eo) mi dubetas",
46                  "(en) I doubt a little"],
47     },
48     {"_id": 3,
49      "phrases": ["(fr) où et quand ?",
50                  "(eo) kie kaj kiam ?"],
51     },
52     {"_id": 4,
53      "phrases": ["(fr) tu peux !",
54                  "(eo) vi povas !"],
55     }
56   ]
57 }
```

```
10 <rdf:RDF xmlns:rdf="http://www.w3.org/..." xmlns:bd="http://
    bd.bio/">
11 <rdf:Description rdf:about="http://bd.bio/regle1">
12   <bd:langue>espéranto</bd:langue>
13   <bd:categorie rdf:resource="http://bd.bio/conj"/>
14   <bd:texte>Terminaison 'as' pour présent</bd:texte>
15 </rdf:Description>
16 <rdf:Description rdf:about="http://bd.bio/regle2">
17   <bd:langue>espéranto</bd:langue>
18   <bd:categorie rdf:resource="http://bd.bio/conj"/>
19   <bd:texte>Terminaison 'i' pour infinitif</bd:texte>
20 </rdf:Description>
21 <rdf:Description rdf:about="http://bd.bio/regle3">
22   <bd:langue>espéranto</bd:langue>
23   <bd:texte>Terminaison 'u' pour impératif</bd:texte>
24 </rdf:Description>
25 <rdf:Description rdf:about="http://bd.bio/conj">
26   <bd:nom>conjugaison</bd:nom>
27 </rdf:Description>
28 <rdf:Description rdf:about="http://bd.bio/gramm">
29   <bd:nom>grammaire</bd:nom>
30 </rdf:Description>
31 </rdf:RDF>
```

### Exercice 1 – Compréhension des sources (6 points)

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

**Question 1** Deux exceptions peuvent-elles avoir la même description ?

A Non  Oui

**Question 2** Est-ce qu'un pays peut avoir plusieurs langues officielles ?

A Non  Oui

**Question 3** Dans le diagramme E/A, comment sont reliées les entités dérivées des relations LANGUE et MOT ?

- |   |   |
|---|---|
| <input type="checkbox"/> A Entité faible                              | <input type="checkbox"/> D Spécialisation         |
| <input type="checkbox"/> B Association sans cardinalité maximale en 1 | <input type="checkbox"/> E Association réflexive  |
| <input type="checkbox"/> C Association avec cardinalité maximale en 1 | <input checked="" type="checkbox"/> Autre réponse |

**Question 4** Dans le diagramme E/A, comment est représentée UTILISATION ?

- |   |   |
|---|---|
| <input type="checkbox"/> A Association quaternaire sans propriété | <input type="checkbox"/> E Association ternaire sans propriété          |
| <input type="checkbox"/> B Association quaternaire avec propriété | <input checked="" type="checkbox"/> Association ternaire avec propriété |
| <input type="checkbox"/> C Entité                                 | <input type="checkbox"/> G Association binaire avec propriété           |
| <input type="checkbox"/> D Association binaire sans propriété     | <input type="checkbox"/> H Autre réponse                                |

CORRECTION

**Question 5** Dans le diagramme E/A, comment est représentée EXCEPTION ?

- |   |  |   |
|---|--|---|
| <input type="checkbox"/> A Entité                         | <input type="checkbox"/> D Entité faible | <input type="checkbox"/> G Spécialisation |
| <input type="checkbox"/> B Entité fille                   | <input type="checkbox"/> E Association   | <input type="checkbox"/> H Entité forte   |
| <input checked="" type="checkbox"/> Association réflexive | <input type="checkbox"/> F Agrégation    | <input type="checkbox"/> I Autre réponse  |

**Question 6** D'où provient la propriété *idMracine* de Mot ?

- |  |   |   |
|--|---|---|
| <input type="checkbox"/> A Association avec cardinalité maximale en 1                    | <input type="checkbox"/> C Association réflexive sans cardinalité maximale en 1 | <input type="checkbox"/> E Association sans cardinalité maximale en 1 |
| <input checked="" type="checkbox"/> Association réflexive avec cardinalité maximale en 1 | <input type="checkbox"/> D Entité   | <input type="checkbox"/> F Entité faible                              |
|  |   | <input type="checkbox"/> G Autre réponse                              |

**Question 7 ♣** Si l'on veut stocker en relationnel les données de la source JSON en respectant les formes normales, quelle(s) relation(s) doit-on ajouter au schéma relationnel ?

- |   |  |
|---|--|
| <input type="checkbox"/> A PHRASELANGUE( <u>#idP</u> , #idL, phrase)    | <input type="checkbox"/> H PHRASEPHRASE( <u>#idP1</u> , #idP2, #idL) |
| <input checked="" type="checkbox"/> PHRASE( <u>idP</u> , phrase, #idL)  | <input type="checkbox"/> I PHRASE( <u>idP</u> , langue)              |
| <input type="checkbox"/> C PHRASELANGUE( <u>#idP</u> , #idL)            | <input type="checkbox"/> J PHRASE( <u>idP</u> , langue, phrase)      |
| <input checked="" type="checkbox"/> PHRASEPHRASE( <u>#idP1</u> , #idP2) | <input type="checkbox"/> K PHRASEMOT( <u>#idP</u> , #idM)            |
| <input type="checkbox"/> E PHRASEMOT( <u>#idP</u> , #idM, langue)       | <input type="checkbox"/> L PHRASE( <u>idP</u> , phrase)              |
| <input type="checkbox"/> F PHRASELANGUE( <u>#idP</u> , #idL, phrase)    | <input type="checkbox"/> M PHRASEPHRASE( <u>#idP1</u> , #idP2, #idL) |
| <input type="checkbox"/> G PHRASE( <u>idP</u> , langue, phrase, #idL)   | <input type="checkbox"/> N PHRASE( <u>idP</u> , langue, #idL)        |
|   | <input type="checkbox"/> O Aucune de ces réponses n'est correcte.    |

**Question 8 ♣** Quelles contraintes sont valides sur le document RDF/XML ?

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> bd:nom rdfs:domain rdfs:Class | <input checked="" type="checkbox"/> bd:categorie rdf:type rdf:Property |
| <input type="checkbox"/> B rdf:about rdf:type rdf:Property        | <input checked="" type="checkbox"/> bd:nom rdfs:range rdfs:Literal     |
| <input type="checkbox"/> C bd:langue rdfs:range rdfs:Class        | <input type="checkbox"/> F Aucune de ces réponses n'est correcte       |

**Question 9 ♣** Quelles contraintes sont valides sur le document RDF/XML ?

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> <!ATTLIST categorie resource IDREF #REQUIRED> | <input type="checkbox"/> C <!ATTLIST Description about CDATA>    |
| <input type="checkbox"/> B <!ELEMENT Description (langue, categorie?, texte)>     | <input type="checkbox"/> D <!ELEMENT categorie (#NULL)>          |
|   | <input checked="" type="checkbox"/> <!ELEMENT langue (#PCDATA)>  |
|   | <input type="checkbox"/> F Aucune de ces réponses n'est correcte |

## CORRECTION

### Exercice 2 – Interrogation des sources (6 points)

Écrivez les requêtes suivantes dans le langage demandé.

**Question 10** Les informations sur les langues qui ne sont jamais utilisées comme langue officielle ou qui ont moins de 1000 locutrices (SQL). Le résultat sera trié sur le nom par ordre alphabétique.

**Question 11** Pour chaque langue officielle utilisée par au moins 3 pays, affichez l'identifiant et le nom de la langue ainsi que le ratio entre nombre de locuteurs et nombre d'habitants des pays utilisant cette langue (SQL).

**Question 12** Les paires de règles (nom et langue) de même langue et de même catégorie (SPARQL).

**Question 13** Pour chaque catégorie, son nombre de règles (XQuery). Le résultat sera de la forme `<cat nb="2">conjugaison</cat>` et trié par nombre de règles décroissant.

### Exercice 3 – Intégration de données (8 points)

Pour faciliter l'utilisation des données, vous proposez de les intégrer dans une base de données centralisée. Cette dernière utilise le schéma relationnel existant comme schéma globale. 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 14** Écrivez tout d'abord un programme informatique pour intégrer certaines données de la source JSON. L'objectif est de parcourir chacune des phrases et d'en extraire les mots de plus de 3 lettres, puis d'ajouter chacun de ces mots dans la BD centralisée quand ils ne sont pas déjà présents. Les phrases ne seront pas stockées dans la BD.

**Question 15** Écrivez ensuite un programme informatique pour intégrer les données de la source RDF/XML. Toutes les informations de cette source doivent être ajoutées dans la BD centralisée.

### Exercice 4 – Bonus (1 point)

**Question 16** Traduisez en français la phrase espérantiste "Neniam dubu, ke pripensema kaj engaĝita personareto povas ŝanĝi la mondon".

## CORRECTION

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

Question 4 :  A  B  C  D  E  G  H

Question 5 :  A  B  D  E  F  G  H  I

Question 6 :  A  C  D  E  F  G

Question 7 :  A  C  E  F  G  H  I  J  K  L  M  N

O

Question 8 :  B  C  F

Question 9 :  B  C  D  F

Question 10 :

0  0.25  0.5  0.75  1  1.25

```
SELECT l.* FROM LANGUES l NATURAL JOIN UTILISATION
WHERE idL NOT IN (SELECT idL FROM UTILISATION
AND estOfficielle = 1)
UNION
SELECT l.* FROM LANGUES l
WHERE nbLocuteurs < 1000
ORDER BY nomL DESC;
```

Question 11 :

0  0.25  0.5  0.75  1  1.25

```
SELECT l.idL, nomL, (nbLocuteurs / SUM(nbHabitants)) AS ratio
FROM LANGUES l NATURAL JOIN UTILISATION u NATURAL JOIN PAYS p
WHERE estOfficielle = 1
GROUP BY l.idL, nomL
HAVING COUNT(DISTINCT nomP) > 3;
```

CORRECTION

Question 12 :

0

0.25

0.5

0.75

1

1.25



```
SELECT ?n1 ?l1 ?n2
WHERE { ?n1 bd :langue ?l1 .
        ?n2 bd :langue ?l2 .
        ?n1 bd :categorie ?c .
        ?n2 bd :categorie ?c .
        FILTER(?l1 = ?l2 && ?n1 > ?n2)
}
```

Question 13 :

0

0.25

0.5

0.75

1

1.25



```
for $n in //nom
  let $uri := $nom/../@about
  let $nb := fn :count(//categorie[@resource=$uri])
  let $nom := $n/text()
  group by $uri
  order by $nb descending
  return <cat>{@$nb} {$nom}</cat>
```

CORRECTION

Question 14 :

0

0.25

0.5

0.75

1

1.25

1.5

1.75

2

2.25

2.5

2.75



```
3 conn1 = connexion("source1.json")
4 conn_sg = connexion("schema global")
5
6 /* intégration source JSON pour peupler SG.Mot avec les mots de plus de 3 lettres */
7
8 dict_source1 = conn1.get_data()
9 pour doc de dict_source1['exemples']:
10   pour phrase de doc['phrases']:
11     lang = phrase.substring(1,2)
12     attrib = select_attrib(lang) # la table SG.Mot est mal modélisée, la fonction retourne le bon label (labelEo,
13       labelFr, labelEn)
14     pour mot de phrase:
15       si longueur(mot) > 3: # mot de plus de 3 lettres
16         res = conn_sg.query("select * from Mot where {attrib}='{mot}'")
17         if res == false:
18           conn_sg.query("insert into Mot(idM, {attrib}) VALUES(NULL, '{mot}')")
```

## CORRECTION

Question 15 :

0

0.5

1

1.5

2

2.5

3

3.5

4

4.5

```

22 conn2 = connexion("source2.rdf")
23 conn_sg = connexion("schema global")
24
25 req = "SELECT ?r ?l ?cat ?txt
26 WHERE { ?r bd:langue ?l .
27         ?r bd:categorie ?c .
28         ?c bd:nom ?cat .
29         ?t bd:texte ?txt .
30 }"
31 req = "SELECT idL FROM Regles WHERE texte = {?txt}"
32 res = conn_sg.executer(req)
33 if !res: # le texte de la règle n'existe pas déjà, donc insertion
34     nom_r = substring(?r, '/', -1) # extraction du nom de regle après le dernier / de l'URI
35     req = "SELECT idL FROM Langues WHERE nomL = {?1}"
36     res = conn_sg.executer(req)
37     id_lang = res[0]['idL']
38     conn_sg.query("insert into Regle VALUES(NULL, '{nom_r}', '{?cat}', '{?txt}')"
39     last_id = conn_sg.get_last_inserted_id() # récupère le dernier auto-incrément inséré
40     conn_sg.query("insert into Base VALUES({id_lang}, {last_id})")

```

**instances\_rdf = conn2.executer(req) # execute la requete SPARQL  
pour chaque resultat de instances\_rdf :  
# lignes 31 - 40 dans la boucle**



CORRECTION

Question 16 :

0

0.25

0.5

0.75



”Ne doutez jamais qu’un petit groupe de gens réfléchis et engagés puisse changer le monde.  
En fait, c’est toujours comme cela que ça s’est passé.”  
(Margaret Mead, anthropologue, 1901 - 1978).

CORRECTION