



# Contrôle terminal de BDBIO - session 1

UCBL - Département informatique (2023/2024)

Pour assurer l'anonymat, ne notez pas votre nom ou numéro étudiant sur la copie.  
Une feuille papier A4 autorisée. 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.

En visite chez vos parents, vous retrouvez les briques Legos de votre enfance et vous commencez à reconstruire les différents ensembles. Malheureusement, il manque quelques briques! Plusieurs sites web proposent des briques à vendre, et vous décidez d'utiliser vos compétences en intégration de données pour trouver efficacement celles qui vous manquent.

- BOITE (refB, nomB, prix)
- BRIQUE (idB, nomB, largeur, longueur, hauteur, forme, couleur)
- CONSTRUCTION (idC, nomC, thème, description, annéeSortie, dimensions, âge, nomCréateur, estLibre)
- CONTENU (#idB, #idC, quantité)
- DESCRIPTION (#idC, #idM)
- ÉTAPE (#idC, numéroE, instructions, image)
- FABRICATION (#idU, #idB, date, quantité)
- FIGURINE (idF, nomF)
- INCLUSION (#idF, #idC, quantité)
- INFORMATION (#idB, #idM)
- MOTCLÉ (idM, libellé)
- PHOTO (idP, titre, description, cheminFichier, #idB, #idC)
- SUBSTITUTION (#idB1, #idB2, #idT)
- TYPE SUB (idT, nomT, description)
- USINE (idU, ville, pays)
- VENTE (#idC, #refB)

Source 1 (schéma relationnel)

```
[
  {
    "color": "blue",
    "bricks": [
      {
        "_id": 127,
        "size": "2x2",
        "shape": "corner"
      },
      {
        "_id": 1325,
        "size": "2x2",
        "shape": "plate"
      }
    ]
  },
  {
    "color": "brown",
    "bricks": [
      {
        "_id": 2732,
        "size": "1x3x2",
        "shape": "arch with curved top"
      }
    ]
  }
]
```

Source 2 (JSON)

idB	nomB	largeur	longueur	hauteur	forme	couleur
1	Lego1	1	4	1		Rouge
2	Lego2	2	2	1		Bleu
3	Lego3	2	2	1	Coin	Vert
4	Lego4	2	3	2	Plat	Jaune
5	Lego5	2	8	2	Plat	Rouge
6	Lego6	2	2	4	Arche	Bleu
7	Lego7	4	4	1	Arche	Vert
8	Lego8	4	2	2	Arche	Marron

Source 1 (instances de la table Brique)

```
<briques>
  <brique id="1" forme="brique avec
  → grille">
    <dim>2x1x1</dim>
    <couleur nom="Rose" />
  </brique>
  <brique id="2" forme="brique 2x4">
    <couleur nom="Jaune"/>
    <couleur nom="Vert" />
  </brique>
</briques>
```

Source 3 (XML)

## 1 Compréhension des sources (7.5 points)

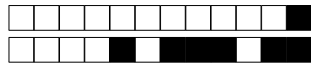
**Question 1** Dans le diagramme E/A qui produirait le schéma relationnel fourni, quelle est la représentation de CONTENU ?

Entité

Association binaire avec cardinalité max. en 1

Association binaire sans cardinalité max. en 1

Association n-aire réflexive



- |  |  |
|--|--|
| <input type="checkbox"/> Association n-aire avec cardinalité max. en 1 | <input type="checkbox"/> Association binaire réflexive |
| <input type="checkbox"/> Association n-aire sans cardinalité max. en 1 | <input type="checkbox"/> Entité faible                 |

**Question 2** Dans le diagramme E/A qui produirait le schéma relationnel fourni, une PHOTO est associée à...

- |   |   |
|---|---|
| <input type="checkbox"/> Une brique ou une construction   | <input type="checkbox"/> Une brique et une construction   |
| <input type="checkbox"/> Des briques et des constructions | <input checked="" type="checkbox"/> On ne peut pas savoir |
| <input type="checkbox"/> Des briques ou des constructions |   |

**Question 3** Dans le diagramme E/A qui produirait le schéma relationnel fourni, comment sont reliés CONSTRUCTION et ÉTAPE ?

- |   |   |  |                                      |
|---|---|--|--------------------------------------|
| <input type="checkbox"/> Spécialisation         | <input checked="" type="checkbox"/> Entité faible | <input type="checkbox"/> Association n-aire  | <input type="checkbox"/> Pas reliées |
| <input type="checkbox"/> Autre type de relation | <input type="checkbox"/> Association réflexive    | <input type="checkbox"/> Association binaire |                                      |

**Question 4** Dans le diagramme E/A qui produirait le schéma relationnel fourni, quelle est la représentation de SUBSTITUTION ?

- |   |   |
|---|---|
| <input type="checkbox"/> Spécialisation                                 | <input type="checkbox"/> Association n-aire avec cardinalité max. en 1  |
| <input checked="" type="checkbox"/> Association n-aire réflexive        | <input type="checkbox"/> Association binaire réflexive                  |
| <input type="checkbox"/> Entité faible                                  | <input type="checkbox"/> Association n-aire sans cardinalité max. en 1  |
| <input type="checkbox"/> Agrégation                                     | <input type="checkbox"/> Association binaire sans cardinalité max. en 1 |
| <input type="checkbox"/> Association binaire avec cardinalité max. en 1 |   |

**Question 5** ♣ Dans le diagramme E/A qui produirait le schéma relationnel fourni, quelles entités sont reliées à BRIQUE (par n'importe quel type de lien direct) ?

- |   |  |  |   |                                 |
|---|--|--|---|---------------------------------|
| <input checked="" type="checkbox"/> Brique  | <input checked="" type="checkbox"/> Construction | <input checked="" type="checkbox"/> Motclé | <input type="checkbox"/> Description      | <input type="checkbox"/> Étape  |
| <input type="checkbox"/> Substitution       | <input type="checkbox"/> Figurine                | <input type="checkbox"/> Fabrication       | <input type="checkbox"/> Boite            | <input type="checkbox"/> Aucune |
| <input checked="" type="checkbox"/> Typesub | <input type="checkbox"/> Vente                   | <input checked="" type="checkbox"/> Dates  | <input type="checkbox"/> Contenu          |                                 |
| <input checked="" type="checkbox"/> Usine   | <input type="checkbox"/> Information             | <input type="checkbox"/> Inclusion         | <input checked="" type="checkbox"/> Photo |                                 |

**Question 6** ♣ Dans le diagramme E/A qui produirait le schéma relationnel fourni, quelles entités sont reliées à CONSTRUCTION (par n'importe quel type de lien direct) ?

- |   |  |  |   |                                  |
|---|--|--|---|----------------------------------|
| <input type="checkbox"/> Substitution     | <input type="checkbox"/> Contenu           | <input type="checkbox"/> Description         | <input type="checkbox"/> Inclusion        | <input type="checkbox"/> Typesub |
| <input type="checkbox"/> Dates            | <input type="checkbox"/> Usine             | <input checked="" type="checkbox"/> Brique   | <input checked="" type="checkbox"/> Photo | <input type="checkbox"/> Aucune  |
| <input type="checkbox"/> Vente            | <input checked="" type="checkbox"/> Motclé | <input checked="" type="checkbox"/> Figurine | <input type="checkbox"/> Information      |                                  |
| <input checked="" type="checkbox"/> Boite | <input type="checkbox"/> Construction      | <input checked="" type="checkbox"/> Étape    | <input type="checkbox"/> Fabrication      |                                  |

**Question 7** ♣ Quelles lignes faut-il cocher pour construire une DTD qui valide le document XML ?

- |   |   |
|---|---|
| <input type="checkbox"/> <code>&lt;!ATTLIST couleur nom CDATA #IMPLIED&gt;</code> | <input checked="" type="checkbox"/> <code>&lt;!ATTLIST couleur nom CDATA #REQUIRED&gt;</code> |
| <input checked="" type="checkbox"/> <code>&lt;!ELEMENT couleur EMPTY&gt;</code>   | <input type="checkbox"/> <code>&lt;!ELEMENT brique (dim, couleur)&gt;</code>                  |
| <input type="checkbox"/> <code>&lt;!ATTLIST brique forme IMPLIED&gt;</code>       | <input checked="" type="checkbox"/> <code>&lt;!ELEMENT brique (dim?, couleur+)&gt;</code>     |



- |   |   |
|---|---|
| <input checked="" type="checkbox"/> <!ATTLIST brique forme CDATA> | <input checked="" type="checkbox"/> <!ELEMENT dim (#PCDATA)>          |
| <input type="checkbox"/> <!ELEMENT dim (#IMPLIED)>                | <input checked="" type="checkbox"/> <!ATTLIST brique id ID #REQUIRED> |
| <input type="checkbox"/> <!ELEMENT briques (brique)>              | <input checked="" type="checkbox"/> <!ELEMENT briques (brique*)>      |
| <input type="checkbox"/> <!ELEMENT brique (dim?, couleur)>        | <input type="checkbox"/> <!ELEMENT brique id ID>                      |
| <input type="checkbox"/> <!ELEMENT briques (brique?)>             | <input type="checkbox"/> <!ELEMENT couleur>                           |
| <input type="checkbox"/> <!ELEMENT dim (#CDATA)>                  | <input type="checkbox"/> <!ELEMENT brique (dim, couleur+)>            |

**Question 8** Écrivez les triplets descriptifs (schéma) pour représenter la table BRIQUE sous forme de graphe RDF (sauf pour les attributs largeur et longueur). Vous disposez des classes `bd:Brick`, `bd:Shape` (forme), `sch:Text`, `sch:Number` et des prédicats `sch:name`, `sch:height`, `bd:forme` et `sch:color`. Voici des exemples de ressources pour la classe `bd:Shape` : `bd:Brick_1x1`, `bd:Brick_1x2_Hole`.

- 0  
 0.25  
 0.5  
 0.75  
 1  
 1.25  
 1.5  
 1.75  
 2 *Reservé*

```

bd:forme rdfs:domain bd:Brick .
bd:forme rdfs:range bd:Shape .
sch:name rdfs:domain bd:Brick .
sch:name rdfs:range sch:Text .
sch:height rdfs:domain bd:Brick .
sch:height rdfs:range sch:Number .
sch:color rdfs:domain bd:Brick .
sch:color rdfs:range sch:Text .

```

## 2 Interrogation des sources (4.5 points)

Pour les questions suivantes, cocher les cases des instances retournées par la requête (sur la source adéquate).

**Question 9** ♣ `SELECT hauteur FROM Brique EXCEPT SELECT longueur FROM Brique;`

- |                            |                            |                                       |                             |  |
|----------------------------|----------------------------|---------------------------------------|-----------------------------|--|
| <input type="checkbox"/> 7 | <input type="checkbox"/> 8 | <input type="checkbox"/> 6            | <input type="checkbox"/> 5  | <input type="checkbox"/> <i>Aucune</i> |
| <input type="checkbox"/> 2 | <input type="checkbox"/> 0 | <input checked="" type="checkbox"/> 1 | <input type="checkbox"/> 10 |  |

**Question 10** ♣ `SELECT longueur, COUNT(nomB) FROM Brique GROUP BY longueur;`

- |  |                               |                               |  |  |  |
|--|-------------------------------|-------------------------------|--|--|--|
| <input type="checkbox"/> 2, 0            | <input type="checkbox"/> 5, 0 | <input type="checkbox"/> 8, 2 | <input type="checkbox"/> 8, 0            | <input type="checkbox"/> 1, 0            | <input type="checkbox"/> 3, 2            |
| <input type="checkbox"/> 1, 1            | <input type="checkbox"/> 2, 3 | <input type="checkbox"/> 6, 0 | <input checked="" type="checkbox"/> 2, 4 | <input type="checkbox"/> 4, 0            | <input checked="" type="checkbox"/> 8, 1 |
| <input checked="" type="checkbox"/> 4, 2 | <input type="checkbox"/> 7, 0 | <input type="checkbox"/> 2, 5 | <input type="checkbox"/> 3, 4            | <input checked="" type="checkbox"/> 3, 1 | <input type="checkbox"/> <i>Aucune</i>   |

**Question 11** ♣ `SELECT couleur FROM Brique GROUP BY couleur HAVING SUM(idB) >= 8;`

- |                                |                                |  |  |  |  |
|--------------------------------|--------------------------------|--|--|--|--|
| <input type="checkbox"/> Jaune | <input type="checkbox"/> Rouge | <input checked="" type="checkbox"/> Marron | <input checked="" type="checkbox"/> Bleu | <input checked="" type="checkbox"/> Vert | <input type="checkbox"/> <i>Aucune</i> |
|--------------------------------|--------------------------------|--|--|--|--|

**Question 12** ♣ `SELECT x.longueur, y.hauteur FROM Brique x LEFT OUTER JOIN Brique y ON x.forme = y.largeur;`



- |                               |                               |   |   |   |                                  |
|-------------------------------|-------------------------------|---|---|---|----------------------------------|
| <input type="checkbox"/> 1, 4 | <input type="checkbox"/> 2, 2 | <input checked="" type="checkbox"/> 8, NULL | <input checked="" type="checkbox"/> 3, NULL | <input checked="" type="checkbox"/> 2, NULL | <input type="checkbox"/> 1, 1    |
| <input type="checkbox"/> 2, 1 | <input type="checkbox"/> 8, 1 | <input type="checkbox"/> 8, 2               | <input type="checkbox"/> 4, 1               | <input type="checkbox"/> 1, 2               | <input type="checkbox"/> 1, NULL |
| <input type="checkbox"/> 2, 4 | <input type="checkbox"/> 4, 4 | <input checked="" type="checkbox"/> 4, NULL | <input type="checkbox"/> 8, 4               | <input type="checkbox"/> 4, 2               | <input type="checkbox"/> Aucune  |

**Question 13 ♣** Sur la source 2, quelle(s) valeur(s) sont retournées par la requête suivante (seule la valeur est demandée, sans considérer la structure du document) ?

```
db.coll.find({ bricks: { $size: 2 } }, { _id: 0, color: true });
```

- |                               |  |                               |                                |                              |                                 |
|-------------------------------|--|-------------------------------|--------------------------------|------------------------------|---------------------------------|
| <input type="checkbox"/> 2732 | <input checked="" type="checkbox"/> blue | <input type="checkbox"/> 1325 | <input type="checkbox"/> brown | <input type="checkbox"/> 127 | <input type="checkbox"/> Aucune |
|-------------------------------|--|-------------------------------|--------------------------------|------------------------------|---------------------------------|

### 3 Intégration de données (8 points)

Pour répondre aux questions suivantes, vous utiliserez les langages de requête appropriés en fonction des sources. Le reste du programme informatique 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 `tab_sorted = 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 pour migrer les données JSON dans la BD relationnelle. Considérez que vous avez un objet Python pour la source de données JSON. La clé primaire de la table `BRIQUE` est auto-incrémentée. Pour le nom de la brique (`nomB`), vous y stockerez l'identifiant (`_id`) et vous vérifierez grâce à cet identifiant que vous n'insérez pas une brique déjà existante dans la table.



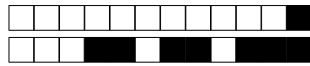
```
bdrel = connexion(source1)
json = connexion(source2.json)

for doc_col in json: # pour chaque doc de couleur
    couleur = doc_col['color']
    for doc_b in doc_col['bricks']:
        id = doc_b['_id']
        dim = doc_b['size'].split('x') # dim[0]=long, dim[1]=larg et dim[2]=haut
        haut = dim[2] if len(dim) >= 3 else 'NULL' # hauteur = dim[2] ou NULL
        forme = doc_b['shape']

        # vérification si brique déjà existante
        req = f"select * from Brique where nomB = '{id}'"
        res = bdrel.executer(req)

    if not res: # brique inexistante
        # requête d'insertion de la brique (idB auto-incrémenté, nomB stocke _id)
        req = f"insert into Briques values (NULL, '{id}', {dim[1]}, {dim[0]}, {haut}, '{forme}',
        ↵ '{couleur}')"
        bdrel.executer(req)
```

**Question 15** Écrivez ensuite un programme pour interroger les données relationnelles et le document XML. L'objectif est de retrouver les briques vertes ou celles avec une grille. Le script affichera le nombre de résultats pour chaque couleur. La fonction XQuery `data(attribut)` permet d'extraire et d'utiliser la valeur d'un attribut.



```
bdrel = connexion(source1)
bdxml = connexion(source3)

# récupération données relationnelles
req = "select couleur, count(*) as nb from Brique where couleur like '%vert%' or forme like '%grille%'
      ↪ group by couleur"
res1 = bdrel.executer(req)

# récupération des données XML (résultat [<couleur>{'Vert':2}</couleur>, ...])
req = """for $c in //couleur
      let $b := $c/..
      let $nb := fn:count(fn:distinct-values($b/@id)
                    where contains($b/@forme, "grille") or contains($c/@nom, "Vert")
                    group by $c/@nom)
      return <couleur>{{'data($c/@nom)':{$nb}}}</couleur>
      """
res2 = bdrdf.executer(req)
res2.remove_tags(res2) # suppression des balises, résultat [{'Vert':2},...]
res2 = {k: v for doc in res2 for k, v in doc.items()} # transformation en 1 seul dict

# union des résultats relationnels et XML
for r in res1:
    if r['couleur'] in res2:
        r['nb'] += res2['couleur']
return res1
```

## 4 Bonus (0.5 point)

Question 16 ♣ Dans quel(s) pays sont fabriquées les pièces Legos ?

France

Chili

Hongrie

Chine

Danemark

Maroc