



# Contrôle terminal de BDBIO - session 1

UCBL - Département informatique (2025/2026)

Pour assurer l'anonymat, n'écrivez pas votre nom ou numéro étudiant sur la copie.  
Une feuille papier A4 autorisée. Durée : 1h30.

ASSEMBLAGES (#idS, #idE, *emplacement*)  
 ÉQUIPEMENTS (idE, *nomE*, *catégorieE*)  
 FABRICATIONS (#idE, #idPi)  
 OBSERVATIONS (#idS, #nomP, dateO, *description*)  
 PIÈCES (idPi, *nomPi*, *fournisseur*)  
 PLANÈTES (nomP, *gravité*, *vitesse*, *découvreur*)  
 SONDÉS (idS, *nomS*, *dateLancement*, *lieuLancement*, *masse*)

Source relationnelle

```

<!DOCTYPE planete [
<!ELEMENT planete (nom, observations)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT observations (observation*)>
<!ELEMENT observation (sonde, description?)>
<!ELEMENT sonde (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ATTLIST sonde id ID #REQUIRED>
<!ATTLIST observation id ID #REQUIRED>
<!ATTLIST observation date CDATA #REQUIRED>
]>

```

Source XML

```

bd:planete rdfs:domain bd:Observation .
bd:planete rdfs:range bd:Planete .
bd:sonde rdfs:domain bd:Observation .
bd:sonde rdfs:range bd:Sonde .
bd:date rdfs:domain bd:Observation .
bd:date rdfs:range xsd:date .
bd:description rdfs:domain bd:Observation .
bd:description rdfs:range xsd:string .
bd:Sonde rdfs:subclassOf bd:Object .
bd:Planete rdfs:subclassOf bd:Object .
bd:nom rdfs:domain bd:Object ;
rdfs:subPropertyOf rdfs:label .

```

Source RDF

```

[
  {
    "_id": <string>,
    "nom_sonde": <string>,
    "observations": [
      {
        "planete": <string>,
        "date": <date>,
        "descriptions": [<string>, <string>, ...]
      },
      ...
    ]
  },
  ...
]

```

Source JSON

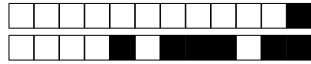
## 1 Modélisation (7 points)

Dans les questions suivantes, on souhaite compléter et améliorer le schéma relationnel. Ajoutez ou modifiez des relations pour répondre aux demandes (en respectant la normalisation).

**Question 1** Une pièce est en général fabriquée à partir d'autres pièces (plus simples).

0
  0.25
  0.5
  0.75
  1
 Réserve

**Question 2** Les sondes sont opérées par des entreprises et/ou par des laboratoires. Ces organisations possèdent un identifiant et un nom. L'entreprise a un numéro SIREN tandis que le laboratoire a un acronyme. Chaque opération est datée et minutée (durée). À terme, il est prévu de faire évoluer les laboratoires (e.g., ajout de nouvelles informations).



0  0.5  1  1.5  2 *Réservé*

**Question 3** Un consortium se compose d'une ou plusieurs organisations. En plus d'un identifiant et d'un nom, il possède une date de création. Chaque sonde est préparée par un unique consortium.

0  0.5  1  1.5  2 *Réservé*

**Question 4** La planète Zig (identifiant P19) a été observée le 01/01/2026 par la sonde Zag (identifiant S49), sans commentaire. Écrivez les triplets RDF pour ces données, qui utilisent le préfixe *ex*. Les URI sont construites à partir des identifiants.

0  0.25  0.5  0.75  1  1.25  1.5  1.75  2 *Réservé*

## 2 Interrogation (4 points)

**Question 5** Lister le nom des planètes qui ont été observées par une sonde utilisant un équipement de catégorie *lidar* et n'ayant jamais observé la planète *Mars* (SQL).



0  0.25  0.5  0.75  1  1.25  1.5  1.75  2 *Réservé*

**Question 6** Lister le nom des sondes avec leur nombre de planètes différentes observées, triés selon ce nombre dans l'ordre décroissant (XQuery).

0  0.25  0.5  0.75  1  1.25  1.5  1.75  2 *Réservé*

### 3 Intégration de données (9 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 python ou pseudo-langage (syntaxe libre, mais indenté et 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.

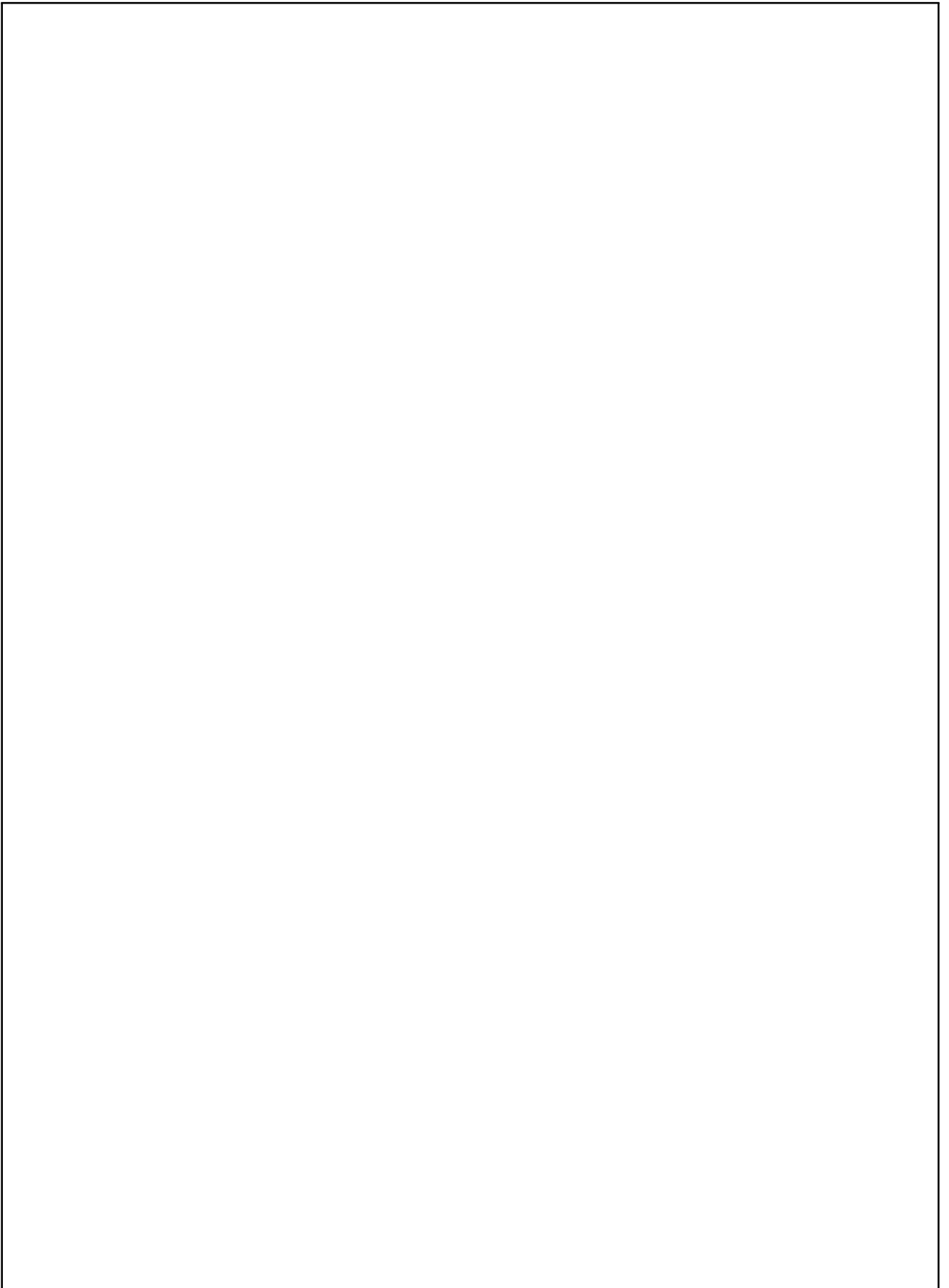
On souhaite afficher les sondes qui ont observé le plus de planètes différentes. On retournera le nom des sondes et le nombre de planètes différentes observées, avec un tri décroissant selon ce nombre. Pour répondre à cette requête, nous allons migrer les données JSON dans la BD relationnelle (car il peut y avoir des doublons sur les observations entre ces 2 sources), puis on interrogera et on fusionnera les résultat des 3 autres sources.

**Question 7** 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 SONDES est auto-incrémentée. Vous vérifierez que vous n'insérez pas une sonde ou une observation déjà existante. Les planètes existent déjà toutes dans la BD relationnelle.



+1/4/57+

0  0.5  1  1.5  2  2.5  3  3.5  4  4.5  5 *Réservé*





**Question 8** Écrivez un programme pour interroger la base de données relationnelle et celle en RDF afin de répondre à la question initiale. Les résultats sur la BD XML (obtenus avec la requête de la question 6) sont stockés dans une liste (de tuples ou de dictionnaires) nommée `resultats_xml` et doivent aussi être agrégés. Il n'y a pas d'observation en doublon entre les 3 sources. Vous retournerez un dictionnaire trié comme résultat de cette interrogation distribuée.

0  0.5  1  1.5  2  2.5  3  3.5  4 *Réservé*



+1/6/55+