

GGMD : Gestion de Grandes Masses de Données

UCBL - Département Informatique de Lyon 1 – 2022

Contrôle terminal

1h30 - Documents autorisés

Chaque partie peut être traitée séparément.

Partie BD répartie (6 points)

Soit le schéma de la base de données « PaysDuMonde » suivant :

Pays (nomP, population, continent, nomHabitant, superficie, monnaie)

nomP identifie de manière unique un pays, *population* est le nombre d'habitants du pays exprimés en millions d'habitants, *continent* représente le continent sur lequel le pays se trouve (*continent* ∈ {'Afrique', 'Amérique', 'Asie', 'Europe', 'Océanie'}), *nomHabitant* est le nom porté par les habitants du pays, la *superficie* du pays est exprimée en milliers de km² et *monnaie* représente le nom de la devise du pays (e.g. euro, dollar, yen).

Frontiere (nomP1, nomP2, lgFrontiere)

nomP1 et *nomP2* référencent les noms de pays ayant une frontière commune. L'attribut *lgFrontiere* exprime en km la longueur de la frontière.

SurfaceMaritime (idSM, nomM, typeS, superficie)

idSM est un entier identifiant de manière unique une surface maritime, *nomM* est le nom de la surface maritime (e.g. 'Pacifique', 'Atlantique', 'Méditerranée'), *typeS* précise s'il s'agit d'une mer, d'un océan, d'une baie, d'un golfe ou d'un détroit. L'attribut *superficie* représente la superficie de la surface maritime. Elle est exprimée en milliers de km².

Littoral (nomP, idSM, lgFrontiere)

nomP référence le nom du pays ayant un littoral maritime avec la surface identifiée par *idSM*. L'attribut *lgFrontiere* exprime en km la longueur du littoral.

Nous supposons que la base de données « PaysDuMonde » est stockée dans une instance PostgreSQL sur le site de Londres.

Pour des raisons stratégiques, il a été décidé de répartir la base « PaysDuMonde » sur trois sites disposant chacun d'une instance PostgreSQL. Il s'agit des sites de New York, Paris et Sydney.

La réalisation de la répartition des données a été effectuée par l'exécution des scripts synthétisés dans l'Annexe A.

Q1. Après avoir pris connaissance de l'Annexe A, nommez les types de traitements qui ont été appliqués à chacune des 5 relations de la base « PaysDuMonde » et donnez les définitions algébriques correspondantes dans le cas où il s'agit de fragmentation.

Pour Pays : On créer une table distante entre notre base et ggmd_ct_london afin de pouvoir par la suite créer des fragments "paysX" qui correspondent aux données associés aux pays du site sur lequel on est.

Fragmentation Mixte: pays1 = $\Pi_{\text{nomP, population, continent, nomHabitant, superficie}} (\sigma_{\text{continent} = \text{"Amerique"}} (\text{Pays}))$

Pour Frontiere : On créer une table distante entre notre base et ggmd_ct_london afin de pouvoir par la suite créer des fragments "frontiereX" qui correspondent aux données associés aux frontières du site sur lequel on est.

Fragmentation Hor. dérivé : frontiere1 = $\text{Frontiere} \bowtie_{\text{nomP1} = \text{nomP or nomP2} = \text{nomP}} \text{pays1}$

Pour SurfaceMaritime : On créer une table distante entre notre base et ggmd_ct_london afin de pouvoir par la suite créer des fragments "surfaceMaritimeX" qui correspondent aux données associés à la surfaceMaritime du site sur lequel on est.

surfaceMaritime1 = $\text{london_surfaceMaritime} \bowtie_{\text{idsm}} ((\Pi_{\text{idsm}} (\text{littoral} \bowtie_{\text{nomP}} \text{pays1})))$

Pour Littoral : On créer une table distante entre notre base et ggmd_ct_london afin de pouvoir par la suite créer des fragments "littoralX" qui correspondent aux données associés à la surfaceMaritime du site sur lequel on est.

littoral1 = $\text{Littoral} \bowtie_{\text{nomP1} = \text{nomP}} \text{pays1}$

Q2. La fragmentation de la relation frontière est-elle correcte ? Justifiez !

Pour qu'une fragmentation soit correcte, on doit vérifier :

- Reconstruction,
- Disjointure,
- Complète.

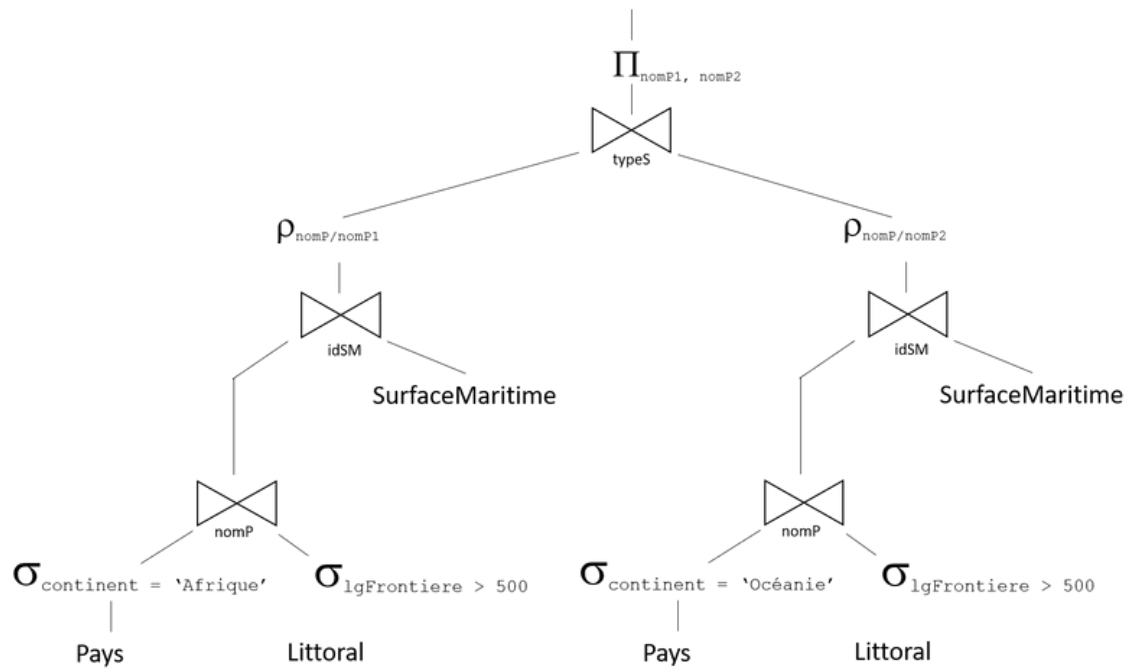
On suppose que le découpage de la table pays sur chaque base a bien été fait.

On réalise une jointure entre les nomP dans la table Frontière de la base distante et le nom des pays du fragment "Pays" que l'on a mtn sur notre site afin de ne récupérer que les frontières associés au pays de notre Site. Cependant, il peut exister des couples de pays qui font parties de 2 régions différentes et apparaître dans 2 sites différents. Il se peut que l'on ait alors de la redondance, annulant ainsi la propriété de Complétude.

Par ailleurs on récupère bien les deux clés primaires (Reconstruction).

Si le fragment Pays est correct alors les propr. "Complète" et "Reconstructible" seront nécessairement vraies puisque l'on se base sur les identifiants de ce fragment pour créer Frontière.

On considère l'expression algébrique E1 suivante portant sur le schéma initial :



Q3. Exprimer en français l'intitulé de la requête correspondant à l'expression algébrique E1.

Donner les noms des pays sur le continent "Afrique" et "Océanie" qui possèdent une surface maritime et dont la longueur de frontière littoral est supérieur à 500 km.

Q4. Exprimer l'expression algébrique E2 exprimant la même requête que E1 à partir de la base de décomposée (cf question 1 et annexe A)

🤔

$$\bigcup \left(\pi_{\text{nomP1, nomP2}} \left(\bigcap \left(\text{surfaceMaritime2} \right) \right) \right) \bigcap \left(\pi_{\text{nomP1, nomP2}} \left(\bigcap \left(\text{surfaceMaritime3} \right) \right) \right)$$

$\left(\sigma_{\text{littoral}} \right)_{\text{lgFrontiere} > 500}$

 $\left(\sigma_{\text{littoral}} \right)_{\text{lgFrontiere} \sim 500}$

Q5. L'utilisation de l'algorithme des semi-jointures présente-t-il un intérêt pour l'exécution de l'expression E2 ? Justifiez !

Oui,
On va d'abord réduire le nombre de tuple en filtrant les frontières sur les sites 2 et 3.
Puis, on va sélectionner seulement les tuples qui nous intéressent pour réduire la quantité de données à transférer vers la base qui demande les données.
Enfin, on réalise une intersection pour avoir seulement la paire de pays qui ont une longueur de littoral d'au moins 500 km et qui sont aussi bien sur la région Afrique et Océanie.

Partie Performance (3 points)

Dans cette partie, vous vous retrouvez dans le rôle d'un administrateur de Bases de Données. Le précédent DBA vous a laissé son carnet contenant les procédures nécessaires à la maintenance et la réparation des bases de données de l'entreprise.

En parcourant ce carnet, vous vous rendez compte qu'un des titres d'une des procédures est illisible, sûrement à la suite d'une manipulation hasardeuse d'une tasse de café au-dessus dudit carnet



Q 1 : Si vous deviez recoller une étiquette par-dessus la tâche de café, quel titre synthétique de procédure donneriez-vous ?

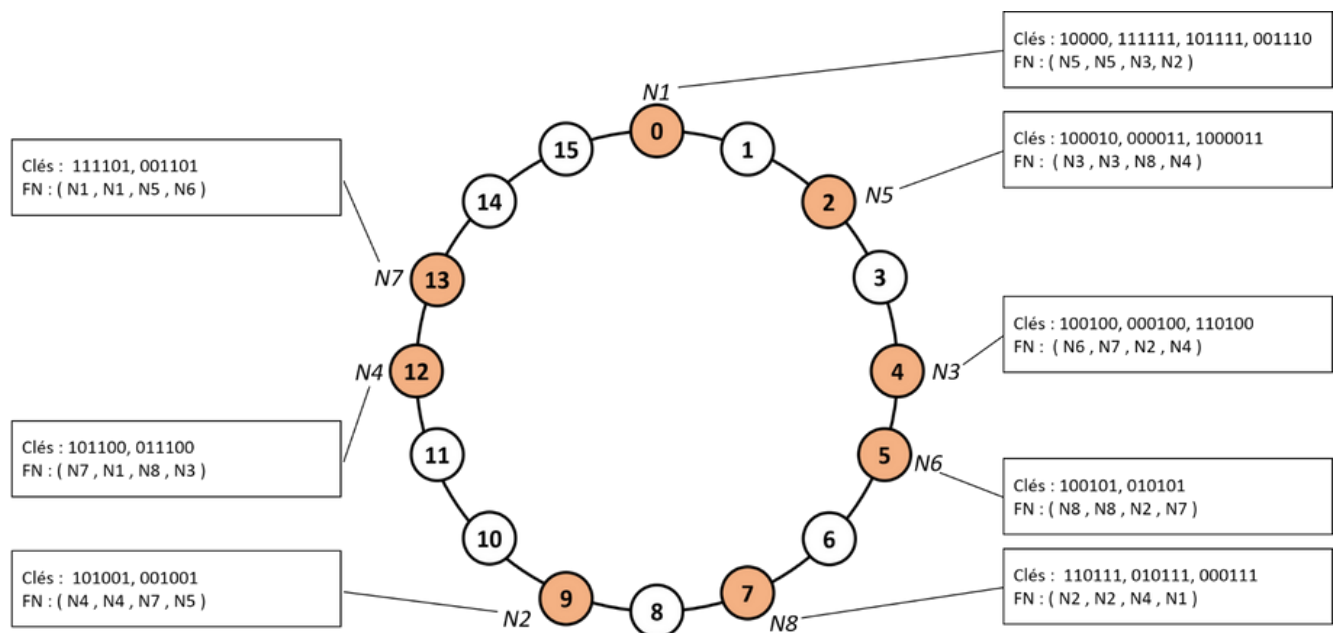
reparamétrage du chemin de données de PostgreSQL

Q2 : Expliquer brièvement pour quel type de problème peut être résolu par l'application de ce protocole.

On peut avoir besoin de cette procédure dans le cas où on souhaite changer de disque la base de données pour un nouveau avec plus de stockage ou un disque plus performant.

Partie P2P(3 points)

Soit un anneau Chord représenté par la figure ci-dessous. Les 8 nœuds connectés à la DHT sont répartis sur l'anneau (nœuds colorés). Pour chaque nœud, on dispose des clés stockées (clés) et des Finger Nodes (FN) utile dans le protocole de routage de Chord.



Q1 : Que garantit la phase de stabilisation d'une DHT de type Chord ?

La phase de stabilisation permet de maintenir la cohérence des nœuds au sein de l'anneau afin que l'on puisse toujours parcourir l'anneau en $\log_2 n$.

Q2 : Expliquer le déroulement de la localisation de la clé 001100 depuis le nœud N1. Vous justifierez à chaque fois le routage choisi.

table N1

valeur	resp	Noeud		valeur	resp	Noeud
$0 \cdot 2^0 = 1$	$[1;2[$	N5	$12 \in [8;0[$ go N2 →	$1 \cdot 2^0 = 10$	$[10;11[$	N4
$0 \cdot 2^1 = 2$	$[2;4[$	N5		$1 \cdot 2^1 = 11$	$[11;13[$	N4
$0 \cdot 2^2 = 4$	$[4;8[$	N3		$1 \cdot 2^2 = 13$	$[13;1[$	N7
$0 \cdot 2^3 = 8$	$[8;0[$	N2		$1 \cdot 2^3 = 1$	$[1;9[$	N5

found !

Clés : 101100, 011100
FN : (N7, N1, N8, N3)

Q3 : L'implémentation de la DHT a été faite mais la primitive gérant la phase de stabilisation n'a pas été testée. Il semblerait que la recherche de la clé 100010 depuis le nœud N3 pose problème. Pourquoi ?

Q4 : Quel correctif, en précisant le nœud et la nouvelle table de FingerNode, proposez-vous pour résoudre la recherche de la clé 100010 depuis le nœud N3 ?

Ajouter un nouveau noeud et recalibrer l'anneau ?

Partie Spark(2 points)

Dans cette partie on considère des requêtes sur le jeu de données d'observations astronomiques du TP tutoriel Spark. Le code épuré de la deuxième application (comptage d'observations par objet) est disponible en annexe C.

Q1. On souhaite modifier cette application de façon à compter également les observations qui n'ont pas d'OBJECT_ID associé. Indiquer quelle modification apporter au script.

```
.filter(lambda l: l[S_OBJECT_ID] != "NULL")
```

Il suffit de supprimer cette ligne qui enlève les objectId vides.

Q2. L'attribut movingObjectId correspond à la colonne n°4. Donner une fonction sur le modèle de countByObjectId (i.e. qui prend en argument inputFilename et sc) qui renverra un rdd contenant les valeurs de object_id pour lesquels il existe plusieurs valeurs de movingObjectId. Indice : on peut dans un premier temps récupérer le plus grand et le plus petit movingObjectId.

```
def doStuff(inputFilename, sc):
    valObjectIdWithMultiplesMovingObjectId = inputFilename \
        .map(lambda l: l.strip(), l.split(";") ) \
        .map(lambda l: (l[S_OBJECT_ID], l[4]) ) \
        .reduceByKey(lambda acc, obj: {
            MAX: max(acc[1],obj[1]),
            MIN : min(acc[1], obj[1])
        }) \
        .collectAsMap() \
        # Sinon on se retrouve avec une structure à reparser
        .filter(lambda obj: obj[MIN] != obj[MAX])

    return valObjectIdWithMultiplesMovingObjectId
```


Partie Flux(2 points)

Dans cette partie, nous considérons le flux des tortues tournant sur une piste circulaire composée de 254 cellules. Pour rappel, les données issues du flux ont pour schéma (*id*, *top*, *position*, *nbDevant*, *nbTour*) avec *id* un entier correspondant au dossard de la tortue qui l'identifie, *top* un entier qui indique le numéro d'observation des tortues sur la piste, *position* un entier qui correspond à la cellule courante où se trouve la tortue (Attention, la position ne permet pas de déterminer le classement de la tortue, car la piste est circulaire et qu'une tortue peut avoir au moins un tour d'avance), *nbDevant* un entier qui indique le nombre de tortues se trouvant devant la tortue dans le classement et *nbTour* indique le nombre de tours de piste effectué par la tortue.

Vous disposez en Annexe B d'un programme python permettant d'effectuer un traitement sur le flux de données décrit précédemment.

Q1. Exprimer en français l'intitulé de la requête à laquelle répond le script de l'annexe B.

Construit un tableau d'objet JSON dont les clés sont *en*, *to*, *tre* et correspondent au classement des tortues dans la course.

Q2. Quelles modifications apporteriez-vous pour que le calcul porte sur les 10 dernières secondes et ce fasse toutes les 30 secondes

Pour que le calcul se fasse toutes les 30 secondes on va changer la window qui est à 10 pour 30.
Enfin pour que le calcul porte sur les 10 dernières secondes on va devoir filtrer les rdd selon leur timestamp afin de n'avoir que les 10 dernières secondes par rapport à notre window de 30s.

Partie BIG DATA : (4points)

Exercice

1/ Comment les clés et les valeurs sont-elles transmises aux reducers durant une phase shuffle et sort standard de MapReduce?

- ☐ Les clés sont transmises au reducer dans un ordre aléatoire; les valeurs pour une clé donnée ne sont pas triées.
- ☒ Les clés sont transmises au reducer dans un ordre trié, les valeurs pour une clé donnée ne sont pas triées.
- ☐ Les clés sont transmises au reducer dans un ordre trié, les valeurs pour une clé donnée sont triées dans un ordre décroissant.
- ☐ Les clés sont transmises au reducer dans un ordre aléatoire; les valeurs pour une clé donnée sont triées dans un ordre croissant.
- ☐ Aucune de ces propositions

Dans MapReduce, durant la phase de shuffle et sort, les clés sont triées avant d'être transmises aux reducers. Cela permet au reducer de traiter toutes les valeurs associées à une même clé en une seule fois. Les valeurs associées à une clé spécifique ne sont pas triées; elles sont regroupées mais l'ordre au sein de ce groupe n'est pas défini.

2/ Que se passe-t-il si le nombre de reducers est de 0 (e.g., dans Hadoop)?

- ☐ Seulement la tâche Reduce se déroule
- ☒ Seulement la tâche Map se déroule

Dans Hadoop, si le nombre de reducers est fixé à 0, cela signifie qu'il n'y aura pas de phase de réduction. Seule la phase de map sera exécutée, et la sortie de cette phase sera la sortie finale du job MapReduce.

- Ⓐ La sortie du Reducer sera la sortie finale
- Ⓐ Aucune de ces trois possibilités

3/ Parmi les opérations suivantes, laquelle ne peut pas utiliser Reducer comme Combiner?

- Ⓐ Groupe par Maximum
 - Ⓐ Groupe par Minimum
 - Ⓐ Groupe par Count
 - Ⓐ Groupe par Moyenne (Average)
- Un combiner peut être vu comme un mini-reducer qui effectue une réduction locale sur les données d'un même nœud avant de les envoyer au reducer. Pour les opérations comme le maximum, le minimum, et le comptage, un combiner peut être utilisé efficacement. Cependant, pour calculer une moyenne, la combinaison des moyennes partielles ne donne pas la moyenne globale, donc un combiner n'est pas approprié.

4/ Le fichier en entrée est divisé en 300 blocs et nous avons à notre disposition 30 clusters de Mapping.

- ☐ Il y aura 100 blocs qui seront donnés à chaque cluster, car dans le MapReduce les tâches sont parallélisées.

On ne connaît pas la répartition en nombre de blocs sur les clusters de Mapping.

Dès qu'un cluster a fini son traitement, un autre bloc lui est alloué.

- ☒ Il faudra 10 vagues de Map pour finir le mapping des données.

Il faudra 300 vagues de Map pour finir le mapping des données.

Dans MapReduce, les tâches sont assignées aux clusters de manière dynamique. Lorsqu'un cluster finit de traiter un bloc, un autre lui est attribué jusqu'à ce que tous les blocs soient traités. Avec 30 clusters et 300 blocs, chaque cluster devrait en moyenne traiter 10 blocs, donc il faudrait 10 vagues de Map pour traiter tous les blocs.

Exercice 2

Définissez les opérations ROLL-UP et DRILL-DOWN et les illustrer sur un exemple de votre choix.

ROLL-UP :

L'opération de roll-up est utilisée dans le contexte de. Dans Hadoop, si le nombre de reducers est fixé à 0, cela signifie qu'il n'y aura pas de phase de réduction. Seule la phase de map sera exécutée, et la sortie de cette phase sera la sortie finale du job MapReduce. Elle consiste à résumer ou à agréger des données à un niveau de granularité plus élevé. Par exemple, passer d'une analyse quotidienne à une analyse mensuelle est un roll-up.

DRILL-DOWN :

À l'inverse du roll-up, le drill-down est le processus de descente dans la hiérarchie de granularité des données. Il implique de regarder les données à un niveau de détail plus fin. Par exemple, passer d'une analyse annuelle à une analyse mensuelle ou quotidienne serait un drill-down.