BDLE – 5IN852 - Examen réparti du 7 février 2020

Exercice 1 : Données touristiques — 8 pts

On considère le schéma. Chaque table est un dataset.

Visite (photoID, personID, date, lat, lon) *on connait la (latitude, longitude) d'une photo*

Intérêt (POI, lat, lon, catégorie) *catégorie vaut hôtel, musée, restau, ...*
on connait la (latitude, longitude) d'un point d'intérêt POI.

Place (photoID, pays, ville) *il y a 50 pays et 20 villes par pays en moyenne*

Il y a 10 000 tuples dans Visite, 11 000 dans Place et 1000 dans Intérêt.

On suppose que tous les attributs sont indépendants.

La répartition initiale d'un dataset est aléatoire (*ie.*, ne dépend pas d'un attribut) sur **10** machines, avec **une** partition par machine, et le même nombre d'objets dans chaque partition.
L'ordonnancement du traitement en plusieurs étapes suit le principe map-reduce de spark : rassembler dans une étape (*stage*) toutes les opérations pouvant être faites avant un transfert (*shuffle*) qui répartit les données pour d'autres étapes. On quantifie, si possible, les transferts de données en nombre d'objets transférés.

Question 1. Dans Place, une photo peut être associée à plusieurs pays. Par exemple, on peut avoir les tuples (photo1, *England, London*) et (photo1, *United-Kingdom, London*) pour une photo prise à Londres. On considère la requête :

E1 = Place.groupBy("photoID", "ville").agg(collect_list("pays").as("listeP"))

Décrire l'exécution de E1 :

Question 2. On considère **Place1** (photoID, pays, ville) où photoID est **unique**. Expliquer brièvement comment obtenir Place1 à partir de Place. Vous pouvez répondre par une expression en syntaxe Dataframe.

Question 3. Soit la requête affichant le nombre de villes dans chaque pays :

E3 = Place1.groupBy("pays").agg(**countDistinct**("ville").as("nv")).orderBy("pays")

Proposer une exécution avec **un seul** transfert qui répartit les données par **intervalle** (et non par hachage). Décrire les étapes partielles et complètes du traitement.

Proposer une expression E3b donnant un résultat équivalent à celui de E3. E3b doit utiliser la fonction distinct() mais pas la fonction countDistinct().

Décrire l'exécution de E3b. Préciser les calculs (partiels/complets) et les transferts

Question 4 . Jointure entre Visite et Intérêt. On veut associer les photos avec les points d'intérêt ayant les mêmes coordonnées (lat, lon) :

E4 = Visite.join(Intérêt, ["lat", "lon"]).select("photoID", "POI", "catégorie").

Décrire l'exécution de E4 en effectuant une jointure par hachage parallèle.

Décrire l'exécution de E4 en effectuant une jointure par broadcast ou diffusion de la plus petite des deux tables

Question 5 On considère la fonction distance(lat1, lon1, lat2, lon2) qui retourne la distance entre les positions (lat1, lon1) et (lat2, lon2). On veut associer à chaque photo (dans Visite) les 10 POI (dans Intérêt) les plus proches et qui sont à moins de 500 mètres de la photo.

La structure du résultat est (PhotoID, POI, distance).

a) Proposer une solution simple mais éventuellement peu efficace pour exécuter ce traitement.

b) Proposer une solution efficace pour exécuter ce traitement en évitant de calculer la distance entre une photo et tous les POI. Indication, il est possible d'utiliser les informations de Place1 si vous pensez que cela est utile. Décrire chaque étape de manière très lisible.