TP Spark

Pour apprendre à manipuler Spark, nous partirons d'un jeu de données open data hébergé sur le site : data.gouv.fr

L'URL pour le téléchargement du fichier CSV : https://www.data.gouv.fr/fr/datasets/frequentation-des-musees-franciliens-entre-2006-et-2010-idf/

Il s'agit d'une liste de fréquentation au cours du temps de différents musées franciliens.

Premières étapes :

1) Récupérer le fichier sur le portail data.gouv.fr

2) Renommer le fichier en « freq_musees.csv »

3) Transférer le fichier vers la sandbox dans « /root »

4) Pousser le fichier dans HDFS dans le répertoire /data (si le répertoire n'existe pas, créez-le)

5) Ouvrer le terminal Spark

6) Charger le fichier freq_musees.csv (dans hdfs) dans un dataframe nommé « df »

7) Afficher un extrait du dataframe

8) Afficher les 5 premières lignes du dataframe

9) Afficher seulement les colonnes « etablissements » et « localite » à l'écran (texte complet)

10) Compter le nombre de musées disponibles dans ce dataframe

11) Nous ne nous intéresserons pas à certaines des colonnes du dataframe, autant les enlever. Enlever les colonnes « grat_06 », « grat_07 », « grat_08 », « commentaires », « wgs84 » en créant un nouveau dataframe « df_clean »

12) Créer un nouveau dataframe « df_test » d'après « df_clean » contenant uniquement les colonnes « num_ref », « etablissements » et « total_06 ».

13) Vérifier le type de la colonne « total_06 » de « df_test ». Si ce n'est pas le bon, créez un nouveau dataframe « df_test2 » à partir du « df_test ». Que constatez-vous ? Faites-en part à vos encadrants.

14) Le format du fichier csv de départ n'est pas très propre (par exemple, le nombre 288179 est écrit avec un espace dans le fichier : 288 179). Au moment du cast, Spark considère ce nombre comme un string et le passe à null, ce que nous voulons éviter. Nous devons donc créer une fonction pour enlever ces espaces. La fonction suivante enlève tout caractère qui n'est pas un chiffre :

```
def remove_all_except_numbers(col):
    return F.regexp_replace(col, "[^0-9]+", "")
```

15) Recréééz un nouveau dataframe « df_test3 » à partir de « df_test ». Après cela, appliquez la fonction « remove_all_except_numbers » pour enlever tous les espaces sur la colonne « total_06 ». Enfin, passez le type de « total_06 » à Integer, le problème devrait être réglé.

16) Maintenant que vous avez compris comment faire le cast proprement sur une colonne, créez un nouveau dataframe « df_clean_and_cast » en appliquant le cast sur toutes les colonnes le nécessitant à partir de « df_clean » (qui est complet).
Liste des colonnes à caster : « total_06 », « total_07 », « total_08 », « total_09 », « total_10 », « evolution_07_06_en », « evolution_08_07_en », « evolution_09_08_en », « evolution_10_09_en »

17) Trouver combien il y a de musées parisiens

18) Afficher par ordre décroissant la liste des musées les plus fréquentés pendant l'année 2010

19) Afficher par ordre décroissant la liste des musées parisiens les plus fréquentés pendant l'année 2010

20) Afficher la liste des musées ayant une fréquentation supérieure à 1000000 millions de personnes en 2010

21) Afficher la liste des musées ayant une fréquentation entre 500000 et 1000000 millions de visiteurs en 2010

22) Créez un nouveau dataframe « df_enrich » à partir de « df_clean_and_cast » contenant une colonne supplémentaire : la somme de toutes les fréquentations de l'année 2006 à 2010 (somme_freq)

23) Créez un nouveau dataframe « df_enrich2 » à partir de « df_enrich » écartant tous les musées dont la fréquentation n'a pas été remontée au moins une année (de sorte à écarter tous les « null »). Notez ensuite le nombre de musées suite au filtre appliqué.

24) Calculez la moyenne de fréquentation sur les 5 années pour chacun des musées et ajouter une nouvelle colonne « moy_freq » au dataframe « df_enrich2 »

25) Créez un nouveau dataframe « df_enrich3 » à partir de « df_enrich2 » ne gardant que les musées qui ont une fréquentation qui a augmenté d'année en année de 2006 à 2010. Affichez les ainsi que leur nombre.

26) A partir du « df_enrich3 », afficher à l'écran l'ensemble des valeurs moyenne, min, max pour chaque colonne « total_XX » du dataframe

27) Afficher à l'écran la colonne « localite » du « df_enrich3 » en « ville »

28) A partir du « df_enrich2 », créer un nouveau dataframe « df_enrich4 » et calculez l'évolution de fréquentation des musées entre les années 2006 et 2010 dans une nouvelle colonne « evolution_10_06_en »

29) Créez un nouveau dataframe « df_enrich5 » à partir de « df_enrich4 » contenant une nouvelle colonne « écart » et affichant l'écart de fréquentation entre les musées (à partir du musée le plus fréquenté jusqu'au moins fréquenté).

30) Créez un nouveau dataframe « df_enrich6 » à partir de « df_enrich5 ». Créez trois nouvelles colonnes à partir de la colonne « num_ref ». Une colonne « departement_id » contenant les deux premiers chiffres, une colonne « code_insee » contenant les 5 premiers et une colonne « num_musee_insee » contenant les deux derniers chiffres.

31) Télécharger le fichier suivant sur data.gouv.fr : https://www.data.gouv.fr/fr/datasets/correspondance-code-insee-code-postal/ Il s'agit de la liste des communes de France. Transférer le fichier sur la sandbox puis sur hdfs dans /data/ (en le renommant « communes.csv »)

32) Créez un dataframe « df_communes » à partir du fichier communes.csv et ne gardez que les champs « code_insee », « commune », « département », « region », « population »

33) Faire un join entre le dataframe « df_communes » et le « df_enrich6 » sur la colonne « code_insee » dans un nouveau dataframe « df_join ».

34) Faire un dataframe « df_group_by_code_insee » contenant la somme des fréquentations des musées par code_insee, le minimum de fréquentation, le maximum de fréquentation et le nombre de musée dans le code_insee

35) Enregistrer le dataframe « df_join » dans une table hive (nommé « frequentations_musees »)

36) Requêter la table hive une fois enregistré

37) Enregistrer le dataframe « df_group_by_code_insee » dans un fichier csv