INFO834 – TP2 – BD NoSQL Orienté Colonne Parquet (Python)

Informations préalables

Apache Parquet fait partie de l'écosystème *Hadoop* (natif C++) mais vous pouvez également l'utiliser grâce à des API ou à travers des bibliothèques (Java, Python…). La bibliothèque Python s'appelle *PyArrow* (*https://arrow.apache.org/docs/python/*), vous en utiliserez notamment les fonctions statistiques *compute*. Pour cela ajoutez en début de programme la ligne « *import pyarrow.compute as pc* ». Vous allez utiliser également la bibliothèque *Pandas* (*http://www.python-simple.com/python-pandas/panda-intro.php*) qui fournit des structures de données faciles à manipuler et à visualiser. Vérifiez que la bibliothèque *Matplot* (https://matplotlib.org/) est bien installée dans votre environnement.

Rappel (cf. TD2)

Parquet étant fondé sur des fichiers binaires, le contenu de ceux-ci est lisible grâce à la bibliothèque *Pandas (structure DataFrame)*. PyArrow joue le rôle d'intermédiaire entre les 2 (structure *Table*). Quelques fonctions utiles sont données ci-après en italique.

Parquet (binaire sur disque)	<=>	Arrow (binaire mem. vive)	<=>	Pandas (structure mem. vive)
fichier	*write_table()* *read_table()* *write_to_dataset()* *ParquetDataset() et read()*	*Table*	*from_pandas()* *to_pandas()*	*DataFrame* *read_csv()*

Travail à faire :

Vous aurez besoin de 2 fichiers, à télécharger depuis votre espace Moodle : *villes_virgule.csv et academies_virgule.csv* (données ouvertes du site gouv.fr, le premier contient des informations sur les villes, le second contient les données géographiques des différentes académies.

Dans la suite, il vous est demandé de définir et tester des fonctions paramétrées et commentées.

1. Quatre fonctions qui permettent de convertir les données données en paramètre d'un format vers l'autre (en lecture et écriture) : dataframe < - > table, table < - > fichier parquet. Tester sur les données des villes et académies.

2. Une fonction qui permet d'afficher le schéma d'une table. Tester sur les données villes et académies.

3. Une fonction qui renvoie la colonne *col* d'une table donnée en paramètre (fonction *column() ou select()*).

4. En utilisant les fonctions *compute* de *pyarrow*, des fonctions qui renvoient des statistiques sur une colonne de table : *count*, *count_distinct*, *sum*, *min*, *max*. Tester sur des colonnes de la table des villes.

5. Filtrage et tri : des fonctions pour sélectionner les informations sur une ville (e.g. Annecy) et pour sélectionner les informations sur un département (e.g. Haute-Savoie) par ordre alphabétique des villes (fonction *filter* et *sort*).

6. Calculs sur plusieurs colonnes et agrégats : calcul du nombre moyen d'habitants en 2012 ; calcul du nombre moyen d'habitants par département ; afficher le résultat pour le département 74 (fonction *TableGroupBy*).

7. Opérations ensemblistes jointures : afficher les zones de vacances des villes ; les villes de la zone de vacances A ; les départements des zones de vacances A et B ; le nombre de villes par académie ;

8. Affichage avec *matplotlib* : e.g. histogramme de la distribution du nombre de villes par académie.