

SparkSQL - Transformations

<code>sqlContext.createDataFrame(rdd, schema)</code>	Créer un DataFrame à partir d'une collection Python (liste)
<code>sqlContext.createDataFrame(rdd, schema)</code>	Créer un DataFrame à partir d'une collection Python (liste)
<code>sqlContext.createDataFrame(rdd, schema)</code>	Créer un DataFrame à partir d'un fichier
<code>sqlContext.createDataFrame(rdd, schema)</code>	Créer un DataFrame à partir d'une colonne
<code>df.select('*')</code>	Sélectionner toutes les colonnes
<code>df.select('name', 'age')</code>	Sélectionner 1 ou plusieurs colonnes
<code>df.select(df.name, df.age + 10).alias('age')</code>	Sélectionner 2 colonnes, changer la valeur d'une colonne et renommer la colonne
<code>df.drop('age')</code>	Suppression d'une colonne.
<code>df.transform(lambda rdd: rdd)</code>	Retourne un nouveau DataFrame
<code>lambda a, b: a + b</code>	Fonction lambda ou nommée et type du retour
<code>slen = udf(lambda s: len(s), IntegerType())</code>	Retourne un DataFrame dont les lignes respectent la/les condition(s)
<code>inled sdf.filter(i sComment)</code>	Retourne un DataFrame dont les lignes respectent la/les condition(s)
<code>where (function)</code>	Retourne un DataFrame avec les lignes uniques
<code>df.distinct()</code>	Retourne un DataFrame dans l'ordre croissant ou décroissant
<code>order By(cols, *kw)</code>	Retourne un DataFrame dans l'ordre croissant ou décroissant
<code>df.sort(" age ", ascending = False)</code>	Chaque élément est dans une nouvelle ligne
<code>df.select(explode(df-4.interest).alias('anInterest'))</code>	
<code>df.groupBy(df.name).agg({'*': 'count'}).collect()</code>	
<code>df.groupBy(df.name).count()</code>	
<code>df.groupBy().avg().collect()</code>	
<code>df.groupBy('name').avg('age', 'grade').collect()</code>	

SparkSQL - Actions

<code>df.show(n, truncate)</code>	Affiche les n premières lignes du DataFrame
<code>df.take(n)</code>	Affiche les n premières lignes sous forme de une liste
<code>df.collect()</code>	Retourne les enregistrements sous forme de liste
<code>df.count()</code>	Retourne le nombre de lignes dans le DataFrame
<code>df.describe(*cols)</code>	Calcule les statistiques descriptives des colonnes numériques
<code>df.write.format('parquet').save()</code>	Enregistre le DataFrame dans le cache donc pas besoin de réexécuter toutes les transformations et actions. A utiliser si on réutilise souvent le DataFrame.
<code>df.unionAll(bdf)</code>	Concaténation de deux DataFrame
!! ATTENTION !! - S'assurer qu'on a assez d'espace dans le programme "driver" pour utiliser collect(). - Ne jamais utiliser collect() en production. Préférer take(n).	

