RDD- json file - 1 pd

Consider the following json file

{"id": 1, "fname": "Jeanette", "lname": "Penddreth", "gender": "Female" }

{"id": 2, "fname": "Giavani", "lname": "Frediani", "gender": "Male"}

{"id": 3, "fname": "Noell", "lname": "Bea", "gender": "Female"}

{"id": 4, "fname": "Willard", "lname": "Valek", "gender": "Male"}

1. Create RDD for above json file.

2. display the contents in tabular format.

3. Apply filter on id , gender to restrict output.

4. display contents in array format.

5. classify whole data into 2 files based on gender and save them as json files.

6. differentiate between show and collect when used with data frames.

7. sort the given data based out of gender and store in a file.

8. convert given json file into a text file.

Solution -

def main(args: Array[String]): Unit = {

//File Path, change it to where we save the file

val filePath = "C:\\Users\\H239006\\Desktop\\jsonFileTopgear.json"

val spark = SparkSession.builder.appName("jsonFiles").master("local[\*]").getOrCreate()

//1. Convert it to RDD

val jsonRdd = spark.read.json(filePath).rdd

//2. DataFrame and print it.

val jsonDF = spark.read.json(filePath)

jsonDF.select("id", "fname", "lname", "gender").show()

//3. Apply filter on id and gender

jsonDF.filter("id = 1 and gender = 'Female'").show()

//4. Display result in Array format

jsonDF.collect().foreach(println)

//5. Classify whole data into two files with partition

//The number of files generated is just 1 as we have used Coalesce

try {

jsonDF.coalesce(1).write.partitionBy("gender").format("json").save("C:\\Users\\H239006\\Desktop\\genderPartition.json")

} catch {

case e: Exception => e.printStackTrace()

}

//6. Collect and show()

jsonDF.collect()

jsonDF.show()

//7. Sort the data based on gender and store

jsonDF.sort("gender").show()

//8. Convert json to Text File

try{

jsonDF.coalesce(1).rdd.saveAsTextFile("C:\\Users\\H239006\\Desktop\\textFile")

} catch {

case e: Exception => e.printStackTrace()

}

}