Big Data Technologies

2024/2025

Solution of Lab 6

- **SparkSQL**

```
// For implicit conversions like converting RDDs to DataFrames
import spark.implicits._

// path of the CSV file
val path="/opt/spark/work-dir/lab6/olympics.csv"

// read the file as an RDD of String
val rddFile =  sc.textFile(path)

// read the file as a dataset of String
val dsFile =  spark.read.textFile(path)

// read the file as a dataframe of String
val dfFile =  spark.read.text(path)

// read the file as a dataframe with several columns
val dfCSV = spark.read.csv(path)

// print the schemas
// rddFile.printSchema // there is no schema for a RDD
dsFile.printSchema
dfFile.printSchema
dfCSV.printSchema

// convert the RDD into a dataframe
val dfFileFromRDD = rddFile.toDF
```

```
// Define the schema associated to the input file, including the types of each input field
import org.apache.spark.sql.types._
import org.apache.spark.sql._

val schema = StructType(
  List(
    StructField("year", IntegerType, true),
    StructField("city", StringType, true),
    StructField("sport", StringType, true),
   StructField("discipline", StringType, true),
   StructField("athlete", StringType, true),
   StructField("country", StringType, true),
   StructField("gender", StringType, true),
   StructField("event", StringType, true),
   StructField("medal", StringType, true)
  )
)

// Read the file with an explicit schema
val dfCSVschema = spark.read.schema(schema).csv(path)
// Read the file as a dataset of Record objects
case class Record(year : Int, city : String, sport : String, discipline : String, athlete :
String, country : String, gender : String, event : String, medal : String)
val dsRecord = spark.read.schema(schema).csv(path).as[Record]

// Create a temporary view
dfCSVschema.createOrReplaceTempView("olympics")

// First query as a SQL query
val queryFootball= spark.sql("select Country,count(Medal) as Medal from olympics
where Sport=='Football' and Medal=='Bronze' group by country")
queryFootball.show()

// First query with dataframe operations
val queryFootballDF =
dfCSVschema.filter($"medal"==="Bronze").filter($"discipline"==="Football").select(
$"country",$"medal").groupBy("country").count()
queryFootballDF.show()

// Second query as a SQL query
val queryUSA= spark.sql("select Count(Medal) as Medal,country,sport from olympics
where Country=='USA' group by Sport,country")
queryUSA.show()

// Write the results of the second query as a JSON file
queryUSA.repartition(1).write.mode(SaveMode.Overwrite).json("/opt/spark/work-
dir/lab6/olymp.json")
```