Big Data Technologies

2024/2025

Lab 6

➢ SparkSQL

The lecture and the webpage https://spark.apache.org/docs/latest/sql-programming-guide.html will provide you a programming guide to SparkSQL.
This exercise exploits the file “olympics.cvs”. The Summer Olympic Games, first held in 1896, is a major international multi-sport event normally held once every four years. The files “olympics.cvs” contains the list of all Olympic medalists since 1896, organized by each Olympic sport or discipline, and also by Olympiad.
Use the Spark contained and the SparkSQL guide to answer the following questions:

1. Build the image "spark:lab6" after downloading the Dockerfile from the github website in the directory "Lab6" .
   ```
   docker build -t spark:lab6 .
   ```

2. Run a container from the image "spark:lab6"
   ```
   docker run -it spark:lab6
   ```

3. Check the file “olympics.cvs” by running the following commands
   ```
   import scala.sys.process._
   "cat /opt/spark/work-dir/lab6/olympics.csv".!
   ```

4. The library “implicits” gives implicit conversions for converting Scala objects (incl. RDDs) into a Dataset, DataFrame, Columns or supporting such conversions. Import the “implicits” class into your ‘spark’ Session with the command
   ```
   import spark.implicits._
   ```

5. Read the file with the command “sc.textFile” into an object named “rddFile”. What is the type and the content of the resulting object?

6. Read the file with the command “spark.read.textFile” into an object named “dsFile”. What is the type and the content of the resulting object?

7. Read the file with the command “spark.read.text” into an object named “dfFile”. What is the type and the content of the resulting object?

8. Read the file with the command “spark.read.csv” into an object named “dfCSV”. What is the type and the content of the resulting object?

9. Print the schemas of all the previous objects.

10. Is it possible to convert “rddFile” into a dataframe? If so, how?

11. The library “import org.apache.spark.sql” contains useful functions to manipulate dataframes. Import this library with the commands

```
import org.apache.spark.sql.types._
import org.apache.spark.sql._
```

12. We want to create a dataframe whose schema precisely corresponds to the expected types of the fields contained in the CSV file. Create the explicit schema with “StructType” and “StructField”.

13. Read the input file with the explicit shema into the dataframe named “dfCSVschema”. You can use “spark.read” with the appropriate options.

14. We also want to load the input file into a dataset of objects named “dsRecord”. Each object corresponds to a record in the input file. Create a class “Record” which corresponds to a record in the input file.

15. Read the input file as a dataset of objects with type “Record”. You can still use the command “spark.read” with the appropriate options.

16. You have load the input file in several ways (rdd, dataset or dataframe, with or without explicit schema). What are the best objects to store the input file? Why?

17. Create a temporary view “olympics” of the dataframe “dfCSVschema”.

18. Create a SQL query on this temporary table whose goal is: find the total number of bronze medals won by each country in Football. Print the result with the command “show”. The output of the query is named “queryFootball”.

19. Create a dataframe based query (just use the dataframe “dfCSVschema” with dataset operations, do not use the temporary table) to find the total number of bronze medals won by each country in Football. The output of the query is named “queryFootballDF”. Print the result with the command “show”.

20. Create a SQL query on this temporary table whose goal is: find the number of Medals won by the USA grouped by sport. The output of the query is named “queryUSA”. Print the result with the command “show”.

21. Write the results of “queryUSA” into a JSON file named “olymp.json” stored in the directory “/opt/spark/work-dir/lab6”. What is the structure of the file “olymp.json” written by Spark?

    You can use the following command to see the content of “olymp.json”:

```
"ls -l /opt/spark/work-dir/lab6/olymp.json".!
```