

Spark SQL - Exercise

Exercise #47

■ Input:

- A CSV file containing a list of user profiles
 - Header
 - name,age,gender
 - Each line of the file contains the information about one user

■ Output:

- Select male users (gender="male"), increase by one their age, and store in the output folder name and age of these users sorted by decreasing age and ascending name (if the age value is the same)
- The output does not contain the header line

Exercise #47

- Example of input data:

name,age,gender

Paul,40,male

John,40,male

David,15,male

Susan,40,female

Karen,34,female

- Example of expected output:

John,41

Paul,41

David,16

Exercise #47

- Implement two different solutions for this exercise
 - A solution based only on DataFrames
 - A solution based on SQL like queries executed on a temporary table associated with the input data

Exercise #48

■ Input:

- A CSV file containing a list of user profiles
 - Header
 - name,age,gender
 - Each line of the file contains the information about one user

■ Output:

- Select the names occurring at least two times and store in the output folder name and average(age) of the selected names
- The output does not contain the header line

Exercise #48

- Example of input data:

name,age,gender

Paul,40,male

Paul,38,male

David,15,male

Susan,40,female

Susan,34,female

- Example of expected output:

Paul,39

Susan,37

Exercise #48

- Implement two different solutions for this exercise
 - A solution based only on DataFrames
 - A solution based on SQL like queries executed on a temporary table associated with the input data

Exercise #49

■ Input:

- A csv file containing a list of profiles
 - Header: name,surname,age
 - Each line of the file contains one profile
 - name,surname,age

■ Output:

- A csv file containing one line for each profile. The original age attribute is substituted with a new attributed called rangeage of type String
 - rangeage = "[" + (age/10)*10 + "-" + (age/10)*10 + 9"]"

Exercise #49

- Input:

name,surname,age

Paolo,Garza,42

Luca,Boccia,41

Maura,Bianchi,16

- Expected output:

name,surname,rangeage

Paolo,Garza,[40-49]

Luca,Boccia,[40-49]

Maura,Bianchi,[10-19]

Exercise #50

■ Input:

- A csv file containing a list of profiles
 - Header: name,surname,age
 - Each line of the file contains one profile
 - name,surname,age

■ Output:

- A csv file containing one single column called "name_surname" of type String
 - name_surname = name+" "+surname

Exercise #50

- Input:

name,surname,age

Paolo,Garza,42

Luca,Boccia,41

Maura,Bianchi,16

- Expected output:

name_surname

Paolo Garza

Luca Boccia

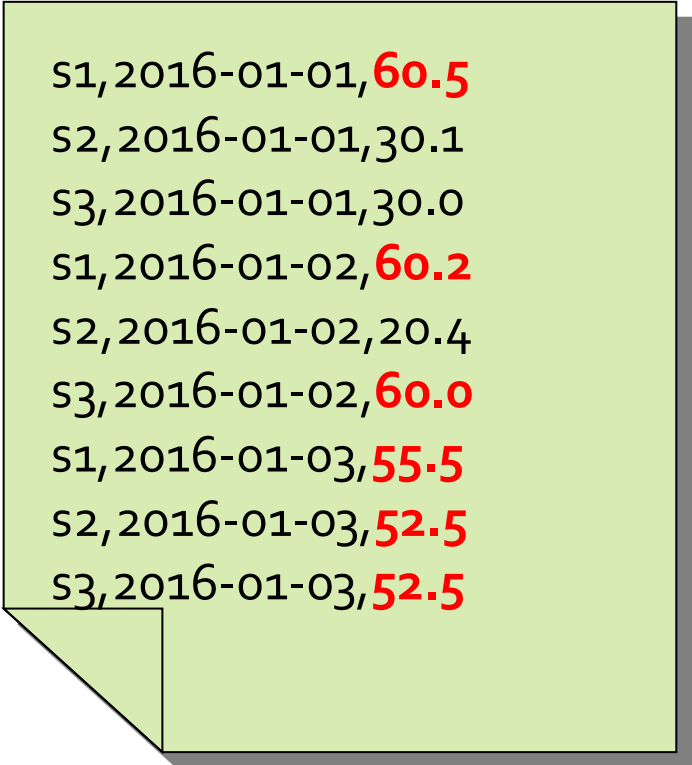
Maura Bianchi

Exercise #50_new

- More critical than normal values
- Input: a textual csv file containing the daily value of PM10 for a set of sensors
 - Each line of the files has the following format
sensorId,date,PM10 value ($\mu\text{g}/\text{m}^3$)\n
- Output: The ids of the sensors with a number of critical dates ($\text{PM}_{10} > 50$) greater than the number of normal dates
 - Store the result in the output folder

Exercise #50_new

- Input file



```
s1,2016-01-01,60.5  
s2,2016-01-01,30.1  
s3,2016-01-01,30.0  
s1,2016-01-02,60.2  
s2,2016-01-02,20.4  
s3,2016-01-02,60.0  
s1,2016-01-03,55.5  
s2,2016-01-03,52.5  
s3,2016-01-03,52.5
```

- Output

s1

s3