ex49

August 14, 2022

[ ]:

```
from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession

conf = SparkConf().setAppName("ex49")
sc = SparkContext(conf=conf)
ssql = SparkSession.builder.getOrCreate()
```

[3]:

```
inputPath = "data/Ex49/data/"
outputPath = "out49/"
```

[4]:

```
df = ssql.read.load(
    inputPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[ ]:

```
#definisco una UDF per implementare il mapping richiesto di age
ssql.udf.register("newAge", lambda age: "["+str((age//10)*10)+"-"+str((age//
 ↪10)*10+9)+"]")
```

[6]:

```
final_df = df.selectExpr("name", "surname", "newAge(age) as AgeCategory").write.
 ↪csv(outputPath, header=True)
```

[ ]:

```
#posso fare la stessa cosa in SQL dopo aver definito la nuova UDF
ssql.createOrReplaceTempView("people")
df_sql = ssql.sql("""
SELECT name, surname, newAge(age) as AgeCategory
FROM people
""")
```