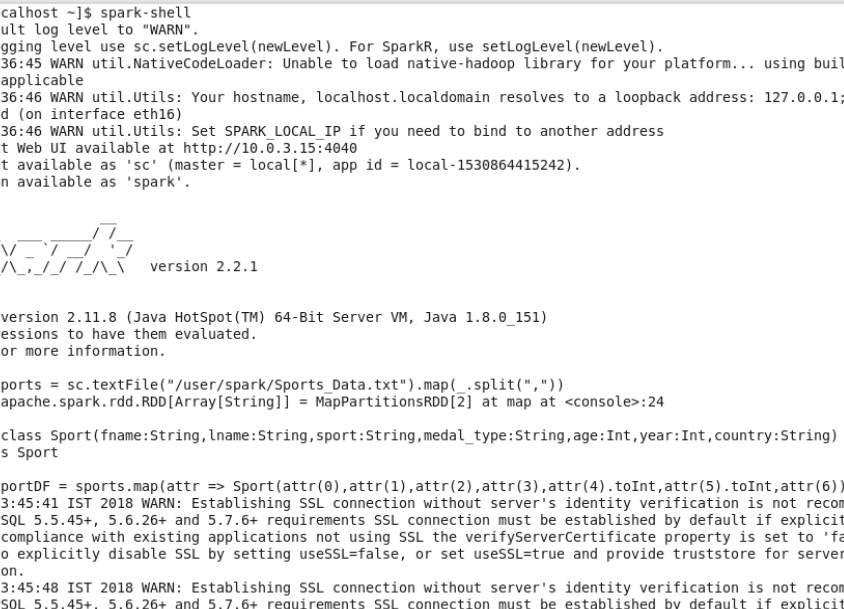


# SPARKSQL2 ASSIGNMENT



```
Applications Places Screenshot
acavgild@localhost:~
File Edit View Search Terminal Help
[acavgild@localhost ~]$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
18/07/06 13:36:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
18/07/06 13:36:46 WARN util.Utils: Your hostname, localhost.localdomain resolves to a loopback address: 127.0.0.1; using 10.0
.3.15 instead (on interface eth16)
18/07/06 13:36:46 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Spark context Web UI available at http://10.0.3.15:4040
Spark context available as 'sc' (master = local[*], app id = local-1530864415242).
Spark session available as 'spark'.
Welcome to

      / \
     / \
    / \
   / \
  / \
 / \
/ \

version 2.2.1

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151)
Type in expressions to have them evaluated.
Type :help for more information.

scala> val sports = sc.textFile("/user/spark/Sports Data.txt").map(_.split(","))
sports: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[2] at map at <console>:24

scala> case class Sport(fname:String,lname:String,sport:String,medal_type:String,age:Int,year:Int,country:String)
defined class Sport

scala> val sportDF = sports.map(attr => Sport(attr(0),attr(1),attr(2),attr(3),attr(4).toInt,attr(5).toInt,attr(6))).toDF
Fri Jul 06 13:45:41 IST 2018 WARN: Establishing SSL connection without server's identity verification is not recommended. Acc
ording to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option in
't set. For compliance with existing applications not using SSL the verifyServerCertificate property is set to 'false'. You n
eed either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificat
e verification.
Fri Jul 06 13:45:48 IST 2018 WARN: Establishing SSL connection without server's identity verification is not recommended. Acc
ording to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be established by default if explicit option in
't set. For compliance with existing applications not using SSL the verifyServerCertificate property is set to 'false'. You n
eed either to explicitly disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for server certificat
e verification.
```

The screenshot shows a terminal window with the following content:

```
scala> val lines = sc.textFile("/user/spark/Sports_Data.txt")
lines: org.apache.spark.rdd.RDD[String] = /user/spark/Sports_Data.txt MapPartitionsRDD[8] at textFile at <console>:24

scala> val headers = lines.first
headers: String = firstname,lastname,sports,medal_type,age,year,country

scala> val noheaders = lines.filter(_
    !=headers)
noheaders: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[9] at filter at <console>:28

scala> val sports = noheaders.map(_.split(","))
sports: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[10] at map at <console>:30

scala> sports.collect()
res4: Array[Array[String]] = Array(Array(lisa, cudrow, javellin, gold, 34, 2015, USA), Array(mathew, louis, javellin, gold, 3
4, 2015, RUS), Array(michael, phelps, swimming, silver, 32, 2016, USA), Array(usha, pt, running, silver, 30, 2016, IND), Arra
y(serena, williams, running, gold, 31, 2014, FRA), Array(roger, federer, tennis, silver, 32, 2016, CHN), Array(jenifer, cox,
swimming, silver, 32, 2014, IND), Array(fernando, johnson, swimming, silver, 32, 2016, CHN), Array(lisa, cudrow, javellin, go
ld, 34, 2017, USA), Array(mathew, louis, javellin, gold, 34, 2015, RUS), Array(michael, phelps, swimming, silver, 32, 2017, U
SA), Array(usha, pt, running, silver, 30, 2014, IND), Array(serena, williams, running, gold, 31, 2016, FRA), Array(roger, fed
erer, tennis, silver, 32, 2017, CHN), Array(jen...

scala> val sportDF = sports.map(attr => Sport(attr(0),attr(1),attr(2),attr(3),attr(4).toInt,attr(5).toInt,attr(6))).toDF
sportDF: org.apache.spark.sql.DataFrame = [fname: string, lname: string ... 5 more fields]

scala> sportDF.show()
+-----+
| fname | lname | sport | medal_type | age | year | country |
+-----+
| lisa   | cudrow | javellin | gold       | 34  | 2015 | USA      |
| mathew | louis  | javellin | gold       | 34  | 2015 | RUS      |
| michael | phelps | swimming | silver     | 32  | 2016 | USA      |
| usha   | pt     | running  | silver     | 30  | 2016 | IND      |
| serena | williams | running | gold       | 31  | 2014 | FRA      |
| roger  | federer | tennis  | silver     | 32  | 2016 | CHN      |
| jenifer | cox    | swimming | silver     | 32  | 2014 | IND      |
| fernando | johnson | swimming | silver     | 32  | 2016 | CHN      |
| lisa   | cudrow | javellin | gold       | 34  | 2017 | USA      |
| mathew | louis  | javellin | gold       | 34  | 2015 | RUS      |
| michael | phelps | swimming | silver     | 32  | 2017 | USA      |
+-----+
```

# SPARKSQL2 ASSIGNMENT

```
Applications Places System Fri Jul 6, 2:10 PM Acadgild
acadmild@localhost:~
File Edit View Search Terminal Help
+-----+
only showing top 20 rows

scala> sportDF.registerTempTable("SPORTS")
warning: there was one deprecation warning; re-run with -deprecation for details

scala> val sportData = spark.sql("select * from SPORTS")
sportData: org.apache.spark.sql.DataFrame = [fname: string, lname: string ... 5 more fields]

scala> sportData.show()
+-----+
| fname|  lname| sport|medal_type|age|year|country|
+-----+
| lisa| cudrow| javellin| gold| 34|2015| USA|
| mathew| louis| javellin| gold| 34|2015| RUS|
| michael| phelps| swimming| silver| 32|2016| USA|
| usha| pt| running| silver| 30|2016| IND|
| serena| williams| running| gold| 31|2014| FRA|
| roger| federer| tennis| silver| 32|2016| CHN|
| jenifer| cox| swimming| silver| 32|2014| IND|
| fernando| johnson| swimming| silver| 32|2016| CHN|
| lisa| cudrow| javellin| gold| 34|2017| USA|
| mathew| louis| javellin| gold| 34|2015| RUS|
| michael| phelps| swimming| silver| 32|2017| USA|
| usha| pt| running| silver| 30|2014| IND|
| serena| williams| running| gold| 31|2016| FRA|
| roger| federer| tennis| silver| 32|2017| CHN|
| jenifer| cox| swimming| silver| 32|2014| IND|
| fernando| johnson| swimming| silver| 32|2017| CHN|
| lisa| cudrow| javellin| gold| 34|2014| USA|
| mathew| louis| javellin| gold| 34|2014| RUS|
| michael| phelps| swimming| silver| 32|2017| USA|
| usha| pt| running| silver| 30|2014| IND|
+-----+
only showing top 20 rows

scala>
```

Task1: What are the total number of gold medal winners every year ?

```
val task1 = spark.sql("select year,count(medal_type) as Total from SPORTS where medal_type = 'gold' group by year order by year desc")
task1: org.apache.spark.sql.DataFrame = [year: int, Total: bigint]

scala> task1.show()
+-----+
|year|Total|
+-----+
|2017| 1|
|2016| 2|
|2015| 3|
|2014| 3|
+-----+
```

TASK2: How many silver medals have been won by USA in each sport ?

```
scala> val task1b = spark.sql("select sport,count(medal_type) from SPORTS where country = 'USA' group by sport order by sport desc")
task1b: org.apache.spark.sql.DataFrame = [sport: string, count(medal_type): bigint]

scala> task1b.show()
+-----+
| sport|count(medal_type)|
+-----+
|swimming| 3|
|javellin| 3|
+-----+
```

## SPARKSQL2 ASSIGNMENT

### TASK2: Using udfs on dataframe

Change firstname, lastname columns into Mr.first\_two\_letters\_of\_firstname<space>lastname

```
Applications Places System acadgild@localhost:~
File Edit View Search Terminal Help

scala> val namechange = (s:String) =>{
  | "Mr." + s.take(2)}
namechange: String => String = <function1>

scala> val sqlContext = new org.apache.spark.sql.SQLContext(sc)
warning: there was one deprecation warning; re-run with -deprecation for details
sqlContext: org.apache.spark.sql.SQLContext = org.apache.spark.sql.SQLContext@28d8ac67

scala> sqlContext.udf.register("namechange",namechange)
res18: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(StringType)))
```

```
Applications Places System acadgild@localhost:~
File Edit View Search Terminal Help

scala> val task2a = spark.sql("select namechange(fname),lname from SPORTS")
task2a: org.apache.spark.sql.DataFrame = [UDF:namechange(fname): string, lname: string]

scala> task2a.show()
+-----+-----+
|UDF:namechange(fname)|  lname|
+-----+-----+
|                Mr.li| cudrow|
|                Mr.ma|  louis|
|                Mr.mi| phelps|
|                Mr.us|    pt|
|                Mr.se|williams|
|                Mr.ro| federer|
|                Mr.je|   cox|
|                Mr.fe| johnson|
|                Mr.li| cudrow|
|                Mr.ma|  louis|
|                Mr.mi| phelps|
|                Mr.us|    pt|
|                Mr.se|williams|
|                Mr.ro| federer|
|                Mr.je|   cox|
|                Mr.fe| johnson|
|                Mr.li| cudrow|
|                Mr.ma|  louis|
|                Mr.mi| phelps|
|                Mr.us|    pt|
+-----+-----+
only showing top 20 rows
```

task2a.registerTempTable("TASK2")

## SPARKSQL2 ASSIGNMENT

```
scala> val result = spark.sql("select CONCAT(newfname, ' ', lname) as newname from TASK2 order by newname desc")
result: org.apache.spark.sql.DataFrame = [newname: string]
```

```
scala> result.show()
```

```
+-----+
|      newname|
+-----+
|    Mr.us pt|
|    Mr.us pt|
|    Mr.us pt|
|Mr.se williams|
|Mr.se williams|
|Mr.se williams|
|  Mr.ro federer|
|  Mr.ro federer|
|  Mr.ro federer|
|  Mr.mi phelps|
|  Mr.mi phelps|
|  Mr.mi phelps|
|  Mr.ma louis|
|  Mr.ma louis|
|  Mr.ma louis|
|  Mr.li cudrow|
|  Mr.li cudrow|
|  Mr.li cudrow|
|    Mr.je cox|
|    Mr.je cox|
+-----+
```

```
only showing top 20 rows
```

```
scala> █
```

[acadgild@localhos... acadgild@localhost:~ acadgild Downloads Sports\_Data.txt (~/...

2. Add a new column called ranking using udfs on dataframe, where :

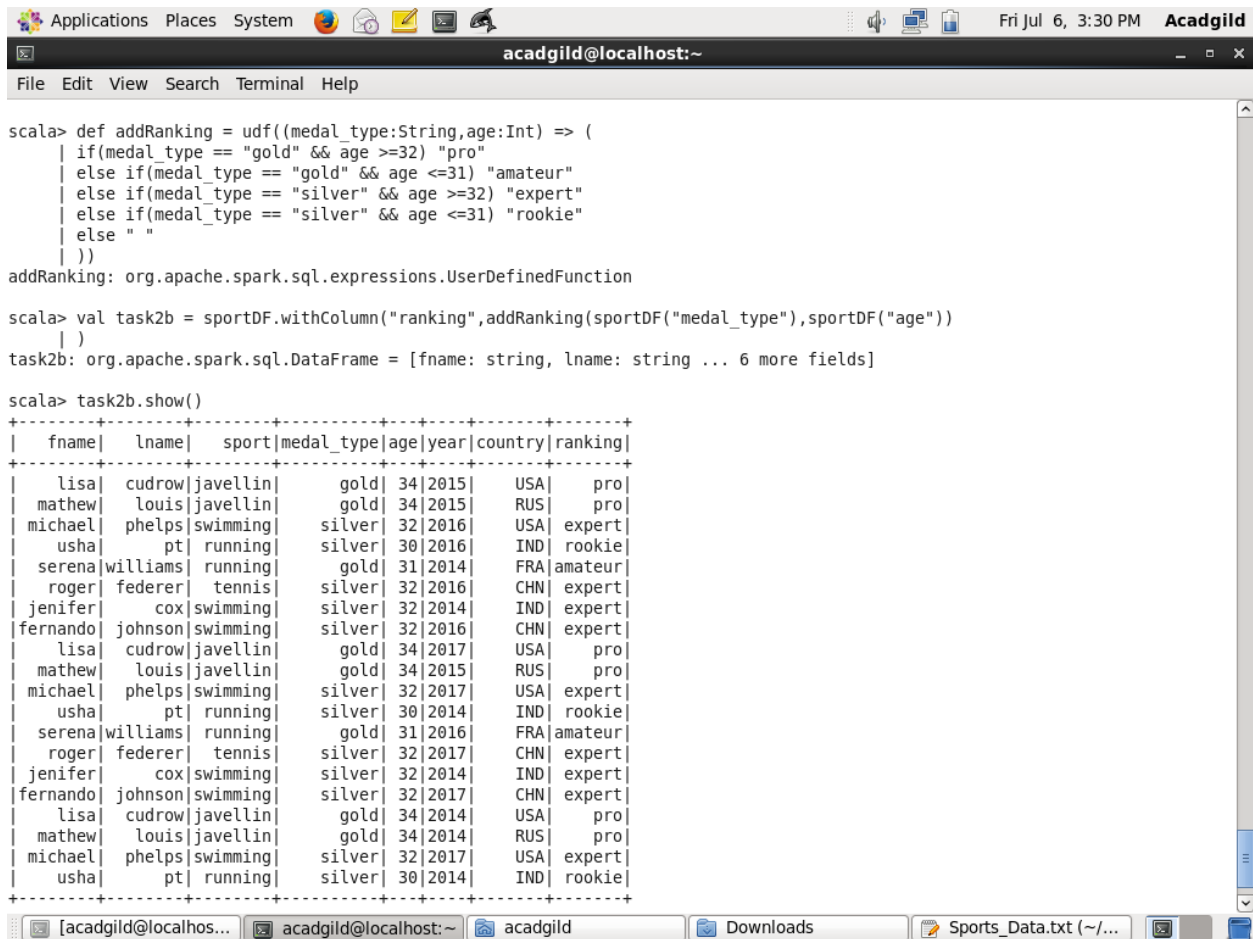
gold medalist, with age  $\geq 32$  are ranked as pro

gold medalists, with age  $\leq 31$  are ranked amateur

silver medalist, with age  $\geq 32$  are ranked as expert

silver medalists, with age  $\leq 31$  are ranked rookie

# SPARKSQL2 ASSIGNMENT



The screenshot shows a terminal window titled 'acadgild@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the following code and output:

```
scala> def addRanking = udf((medal_type:String,age:Int) => (
  | if(medal_type == "gold" && age >=32) "pro"
  | else if(medal_type == "gold" && age <=31) "amateur"
  | else if(medal_type == "silver" && age >=32) "expert"
  | else if(medal_type == "silver" && age <=31) "rookie"
  | else " "
  | ))
addRanking: org.apache.spark.sql.expressions.UserDefinedFunction

scala> val task2b = sportDF.withColumn("ranking",addRanking(sportDF("medal_type"),sportDF("age")))
task2b: org.apache.spark.sql.DataFrame = [fname: string, lname: string ... 6 more fields]

scala> task2b.show()
```

fname	lname	sport	medal_type	age	year	country	ranking
lisa	cudrow	javellin	gold	34	2015	USA	pro
mathew	louis	javellin	gold	34	2015	RUS	pro
michael	phelps	swimming	silver	32	2016	USA	expert
usha	pt	running	silver	30	2016	IND	rookie
serena	williams	running	gold	31	2014	FRA	amateur
roger	federer	tennis	silver	32	2016	CHN	expert
jenifer	cox	swimming	silver	32	2014	IND	expert
fernando	johnson	swimming	silver	32	2016	CHN	expert
lisa	cudrow	javellin	gold	34	2017	USA	pro
mathew	louis	javellin	gold	34	2015	RUS	pro
michael	phelps	swimming	silver	32	2017	USA	expert
usha	pt	running	silver	30	2014	IND	rookie
serena	williams	running	gold	31	2016	FRA	amateur
roger	federer	tennis	silver	32	2017	CHN	expert
jenifer	cox	swimming	silver	32	2014	IND	expert
fernando	johnson	swimming	silver	32	2017	CHN	expert
lisa	cudrow	javellin	gold	34	2014	USA	pro
mathew	louis	javellin	gold	34	2014	RUS	pro
michael	phelps	swimming	silver	32	2017	USA	expert
usha	pt	running	silver	30	2014	IND	rookie

The terminal window has a taskbar at the bottom with several open applications: [acadgild@localhos...], acadgild@localhost:~, acadgild, Downloads, and Sports\_Data.txt (~/...