Exp Name: SPARK Framework UDF (User Defined Function) Experiment:

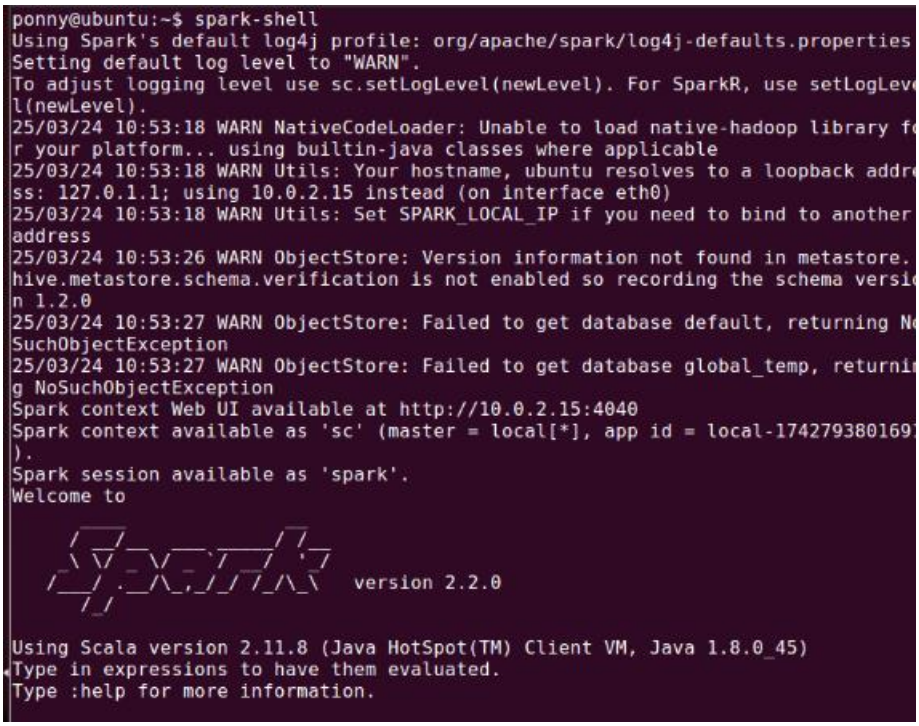
Name: Gouri Ramachandran

Aim

To implement and execute a **User Defined Function (UDF)** in the Apache Spark framework using Scala.

Algorithm/Procedure

1. **Environment Setup**
    - Open **Oracle VirtualBox**.
    - Start the **Hadoop** virtual machine.
    - Open the terminal and start Spark Shell by typing:
    - spark-shell
    - Confirm Spark version and wait for the Scala prompt to appear.



2. **Creating and Using a UDF in Spark**
    - In the Spark shell, follow these steps:
    - Import Spark SQL functions
    - Create a sample DataFrame
    - Define a UDF to add 5 years to age
    - Register the UDF and use it in DataFrame

- Show the result

```
scala> import spark.implicits._
import spark.implicits._
```

```
scala> val cols = Seq("sno","name")
cols: Seq[String] = List(sno, name)
```

```
scala> val data = Seq(("1","gowtham"),
     | ("2","nandhini"),
     | ("3","saravana")
     | )
data: Seq[(String, String)] = List((1,gowtham), (2,nandhini), (3,saravana))

scala> val df = data.toDF(cols:_*)
df: org.apache.spark.sql.DataFrame = [sno: string, name: string]

scala> df.show(false)
+---+--------+
|sno|name    |
+---+--------+
|1  |gowtham |
|2  |nandhini|
|3  |saravana|
+---+--------+


scala> val customUDF = udf(Ucase)
<console>:26: error: not found: value Ucase
       val customUDF = udf(Ucase)
                           ^
```

```
scala> val Ucase = (strQoute:String) => {
     | val dt = strQoute.split(" ")
     | dt.map(f=> f.substring(0,1).toUpperCase + f.substring(1,f.length)).mkString(" ")
     | }
Ucase: String => String = <function1>

scala> val customdf = udf(Ucase)
customdf: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>,StringType,Some(List(StringType)
))
```

```
scala> df.select(col("sno"), customdf(col("name")).as("name") ).show(false)
+---+--------+
|sno|name    |
+---+--------+
|1  |Gowtham |
|2  |Nandhini|
|3  |Saravana|
+---+--------+
```

Program (UDF in Spark using Scala)

import spark.implicits

val cols = Seq("sno", "name")

```
val data = Seq(("1", "gowtham"),
 | ("2","nandhini"),
| ("3", "saravana")  )
val df = data.toDF(cols:_*)
df.show(false)
val customUDF = udf (Ucase)
val Ucase (strQoute: String) => {
| val dt = strQoute.split(" ")
dt.map(f=> f.substring(0,1).toUpperCase + f.substring(1,f.length)).mkString("")
}
val customdf = udf (Ucase)
df.select(col("sno"), customdf (col("name")).as("name") ).show(false)
```

Output

The output in Spark shell will be:

sno	name
1	Gowtham
2	Nandhini
3	Saravana

```
scala> df.select(col("sno"), customdf(col("name")).as("name") ).show(false)
+---+--------+
|sno|name    |
+---+--------+
|1  |Gowtham |
|2  |Nandhini|
|3  |Saravana|
+---+--------+
```

Result

The User Defined Function (UDF) was successfully implemented and executed in the Spark framework. The program effectively added 5 years to each employee's age and displayed the updated values.