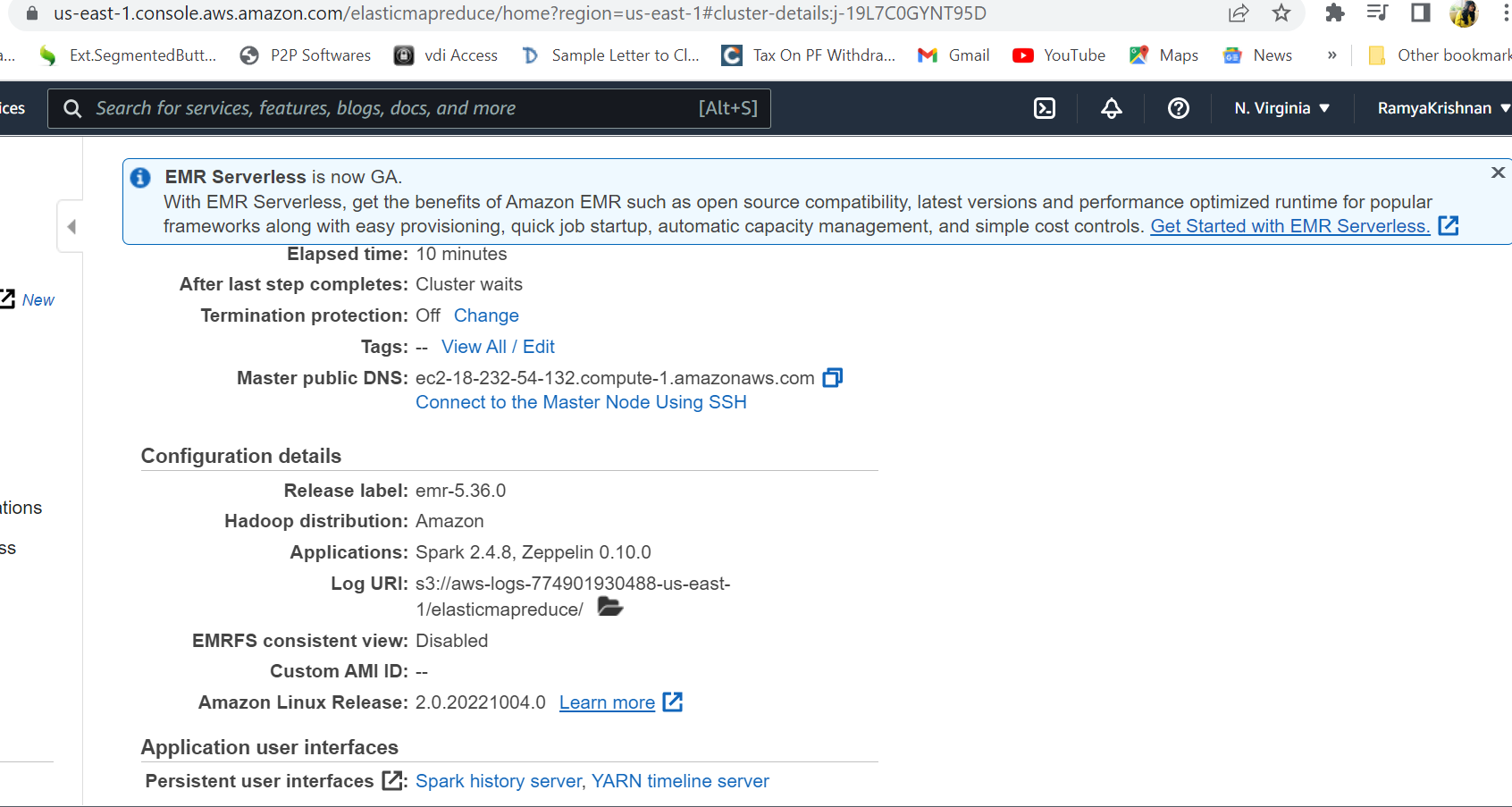
**Assignment #07**

**Exercise 1)**

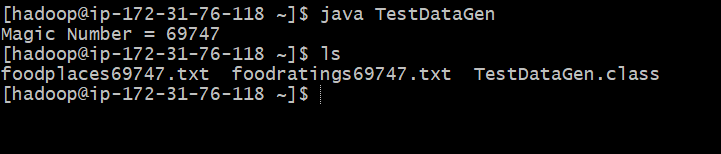
Step A

Start up a Hadoop cluster as previously, but instead of choosing the “Core Hadoop” configuration chose the “Spark” configuration (see below), otherwise proceed as before.

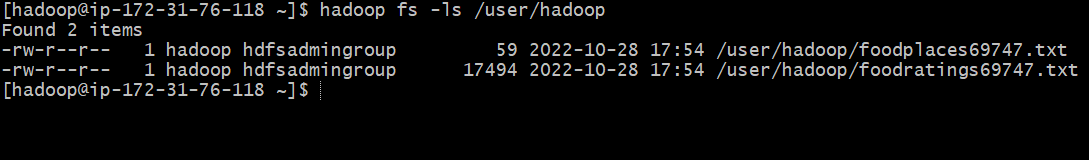


Step B

Use the TestDataGen program from previous assignments to generate new data files.



Copy both generated files to the HDFS directory “/user/hadoop”



Step C

Load the ‘foodratings’ file as a ‘csv’ file into a DataFrame called foodratings. When doing so specify a schema having fields of the following names and types:

|  |  |
| --- | --- |
| **Field Name** | **Field Type** |
| **name** | String |
| **food1** | Integer |
| **food2** | Integer |
| **food3** | Integer |
| **food4** | Integer |
| **placeid** | Integer |

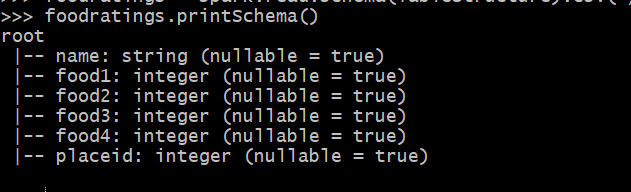
As the results of this exercise provide the magic number, *the code you execute* and screen shots of the following commands:

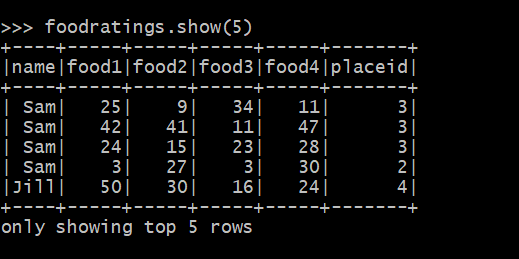
*from pyspark.sql.types import \**

*TableStructure = StructType().add("name", StringType(), True).add("food1",IntegerType(), True).add("food2",IntegerType(), True).add("food3",IntegerType(), True).add("food4",IntegerType(), True).add("placeid",IntegerType(), True)*

*foodratings = spark.read.schema(TableStructure).csv('/user/hadoop/foodratings69747.txt')*

*foodratings.printSchema()*



*foodratings.show(5)*

**Exercise 2)**

Load the ‘foodplaces’ file as a ‘csv’ file into a DataFrame called foodplaces. When doing so specify a schema having fields of the following names and types:

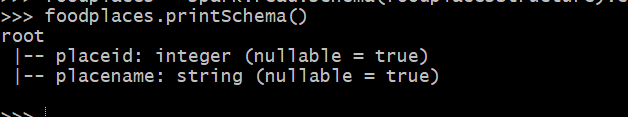
|  |
| --- |
| **Field Nampee Field Type** |
| **placeid** Integer |
| **placename** String |

As the results of this exercise provide *the code you execute* and screen shots of the following commands:

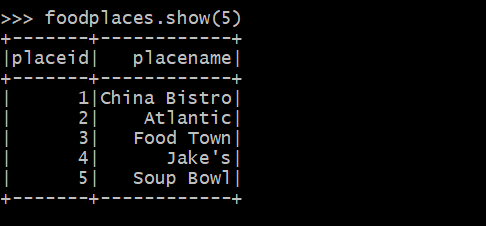
*foodplacesStructure = StructType().add("placeid",IntegerType(), True).add("placename", StringType(), True)*

*foodplaces = spark.read.schema(foodplacesStructure).csv('/user/hadoop/foodplaces69747.txt')*

*foodplaces.printSchema()*



*foodplaces.show(5)*



**Exercise 3)**

Step A

Register the DataFrames created in exercise 1 and 2 as tables called “foodratingsT” and “foodplacesT”

*foodratings.createOrReplaceTempView("foodratingsT") foodplaces.createOrReplaceTempView("foodplacesT")*

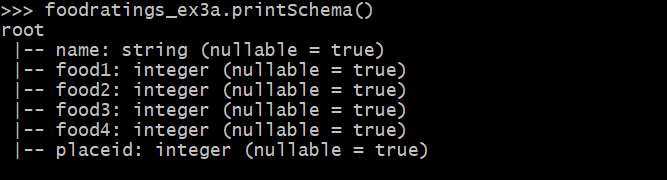
Step B

Use a SQL query on the table “foodratingsT” to create a new DataFrame called foodratings\_ex3a holding records which meet the following condition: food2 < 25 and food4 > 40. Remember, when defining conditions in your code use maximum parentheses.

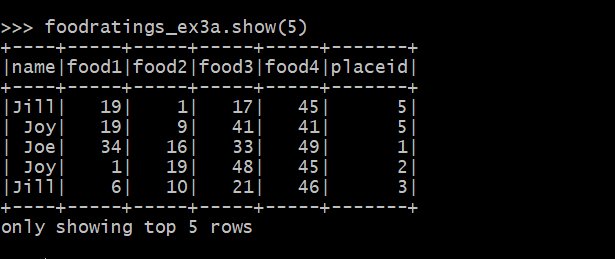
As the results of this step *provide the code you execute* and screen shots of the following commands:

*foodratings\_ex3a = spark.sql("SELECT \* from foodratingsT WHERE food2 < 25 AND food4 > 40")*

*foodratings\_ex3a.printSchema()*



*foodratings\_ex3a.show(5)*



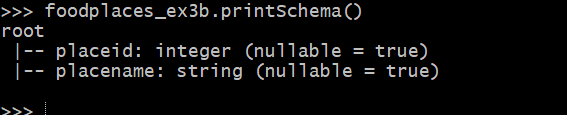
Step C

Use a SQL query on the table “foodplacesT” to create a new DataFrame called foodplaces\_ex3b holding records which meet the following condition: placeid > 3

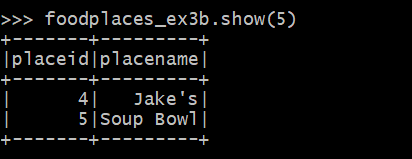
As the results of this step *provide the code you execute* and screen shots of the following commands:

*foodplaces\_ex3b = spark.sql("SELECT \* from foodplacesT WHERE placeid>3")*

*foodplaces\_ex3b.printSchema()*



*foodplaces\_ex3b.show(5)*



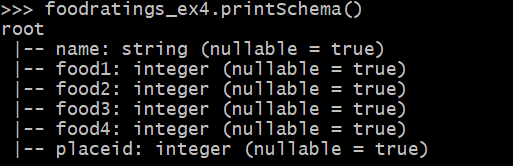
**Exercise 4)**

Use a transformation (not a SparkSQL query) on the DataFrame ‘foodratings’ created in exercise 1 to create a new DataFrame called foodratings\_ex4 that includes only those records (rows) where the ‘name’ field is “Mel” and food3 < 25.

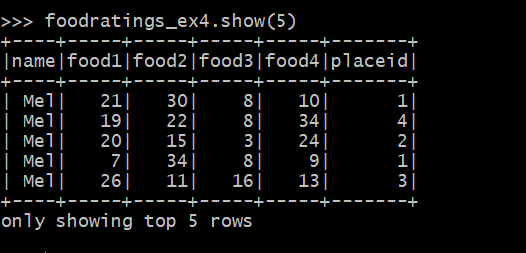
As the results of this step provide the code you execute and screen shots of the following commands:

*foodratings\_ex4 = foodratings.filter(foodratings.name == "Mel").filter(foodratings.food3 < 25)*

*foodratings\_ex4.printSchema()*



*foodratings\_ex4.show(5)*



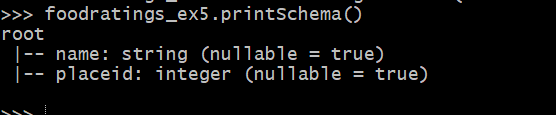
**Exercise 5)**

Use a transformation (**not a SparkSQL query**) on the DataFrame ‘foodratings’ created in exercise 1 to create a new DataFrame called foodratings\_ex5 that includes only the columns (fields) ‘name’ and ‘placeid’

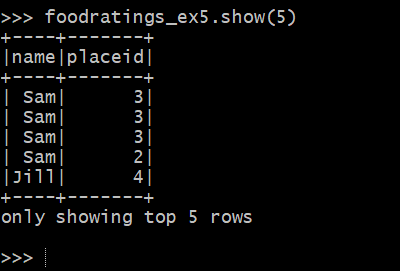
As the results of this step provide the code you execute and screen shots of the following commands:

*foodratings\_ex5 = foodratings.select(foodratings.name, foodratings.placeid)*

*foodratings\_ex5.printSchema()*



*foodratings\_ex5.show(5)*



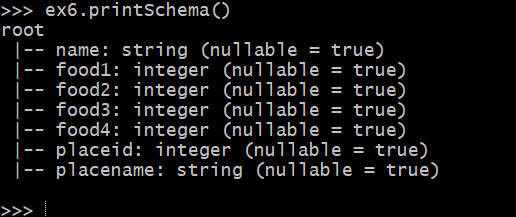
Exercise 6)

Use a transformation (**not a SparkSQL query**) to create a new DataFrame called ex6 which is the inner join, on placeid, of the DataFrames ‘foodratings’ and ‘foodplaces’ created in exercises 1 and 2

As the results of this step provide the code you execute and screen shots of the following commands:

*ex6 = foodratings.join(foodplaces, foodratings.placeid == foodplaces.placeid, "inner").drop(foodratings.placeid)*

*ex6.printSchema()*



*ex6.show(5)*

