ASSIGNMENT - 7

**CSP - 554 BIG DATA TECHNOLOGIES**

Exercise 1)

Step A: Use the TestDataGen program from previous assignments to generate new data files. Copy the files to HDFS.

Step B: Load the ‘foodratings’ file as a ‘csv’ file into a DataFrame called ex1\_foodratings.

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

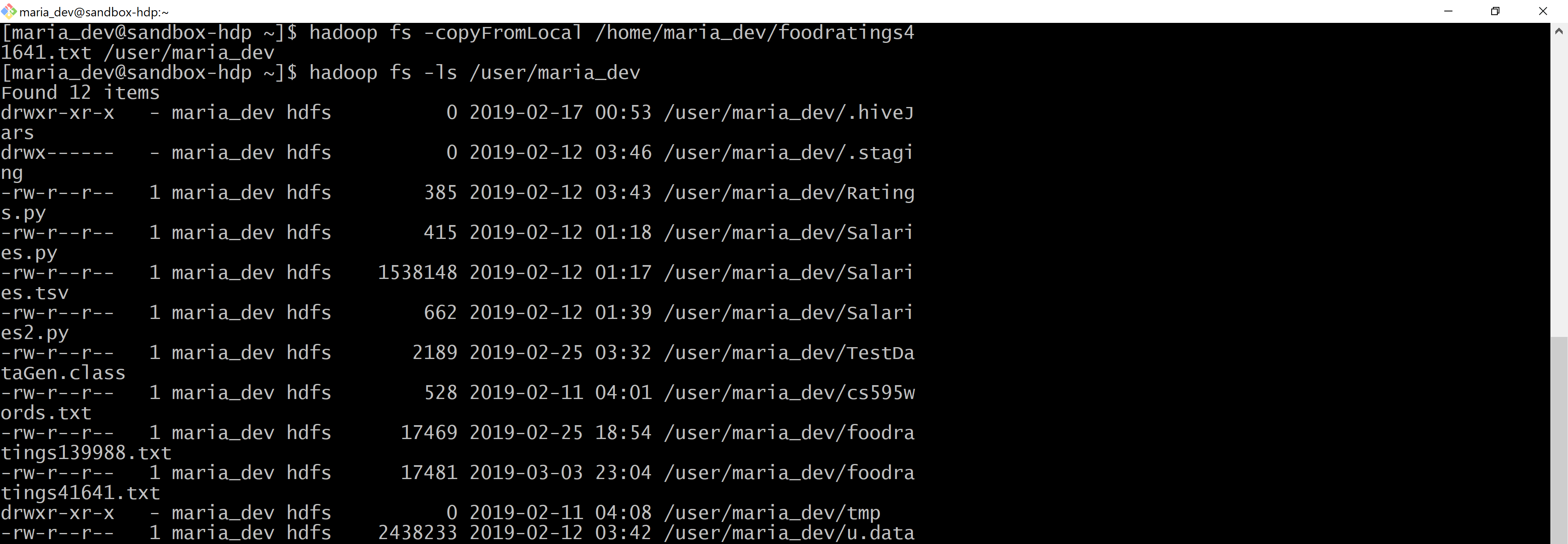
foodratings.printSchema()

foodratings.head(5)

Magic Number = 41641



**Step A:**



**Step B:**

*Code:*

pyspark

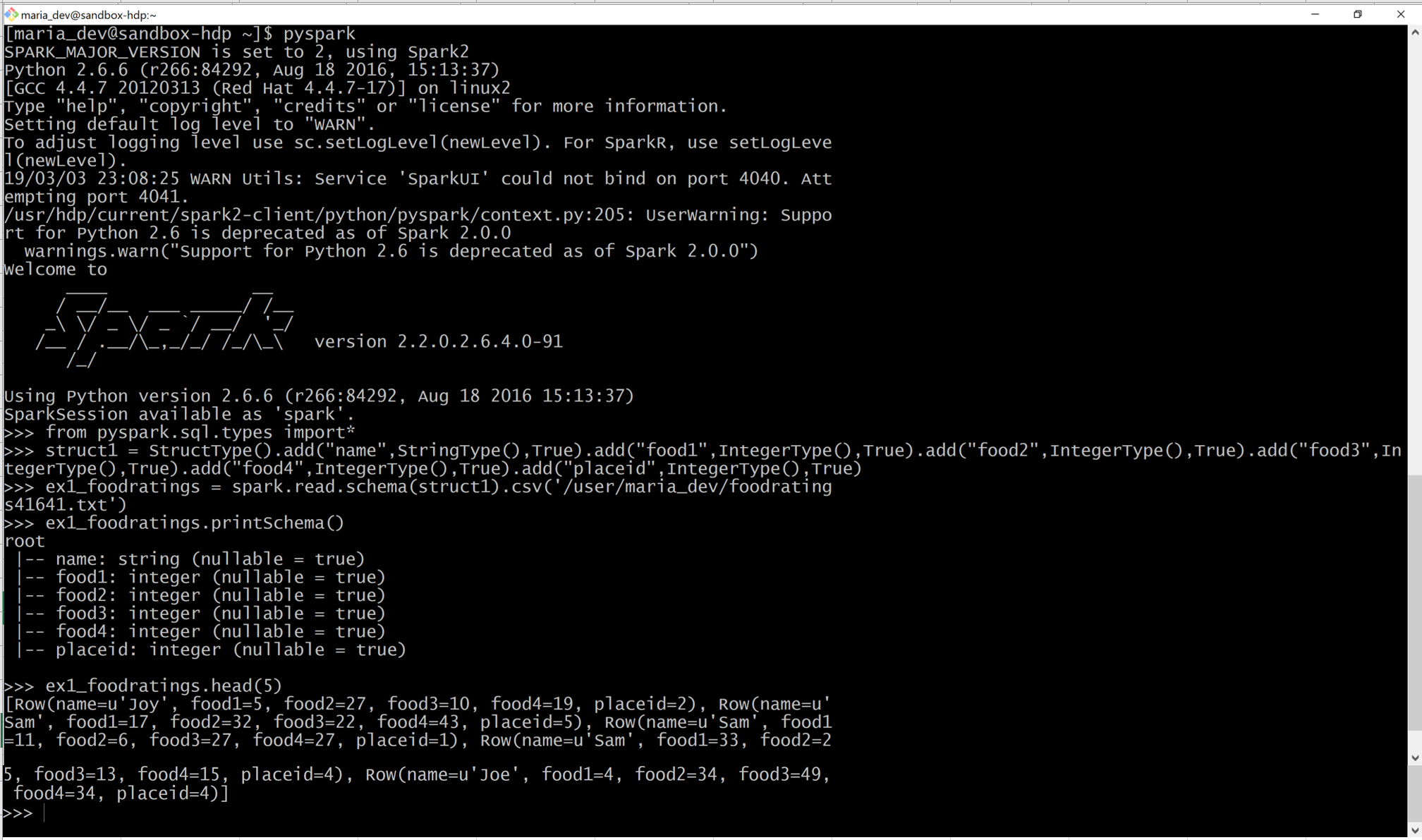
from pyspark.sql.types import \*

struct1 = 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)

ex1\_foodratings = spark.read.schema(struct1).csv('/user/maria\_dev/foodratings41641.txt')

ex1\_foodratings.printSchema()

ex1\_foodratings.head(5)



Exercise 2)

Load the ‘foodplaces’ file as a ‘csv’ file into a DataFrame called foodplaces.

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

foodplaces.printSchema()

foodplaces.head(5)

Code :

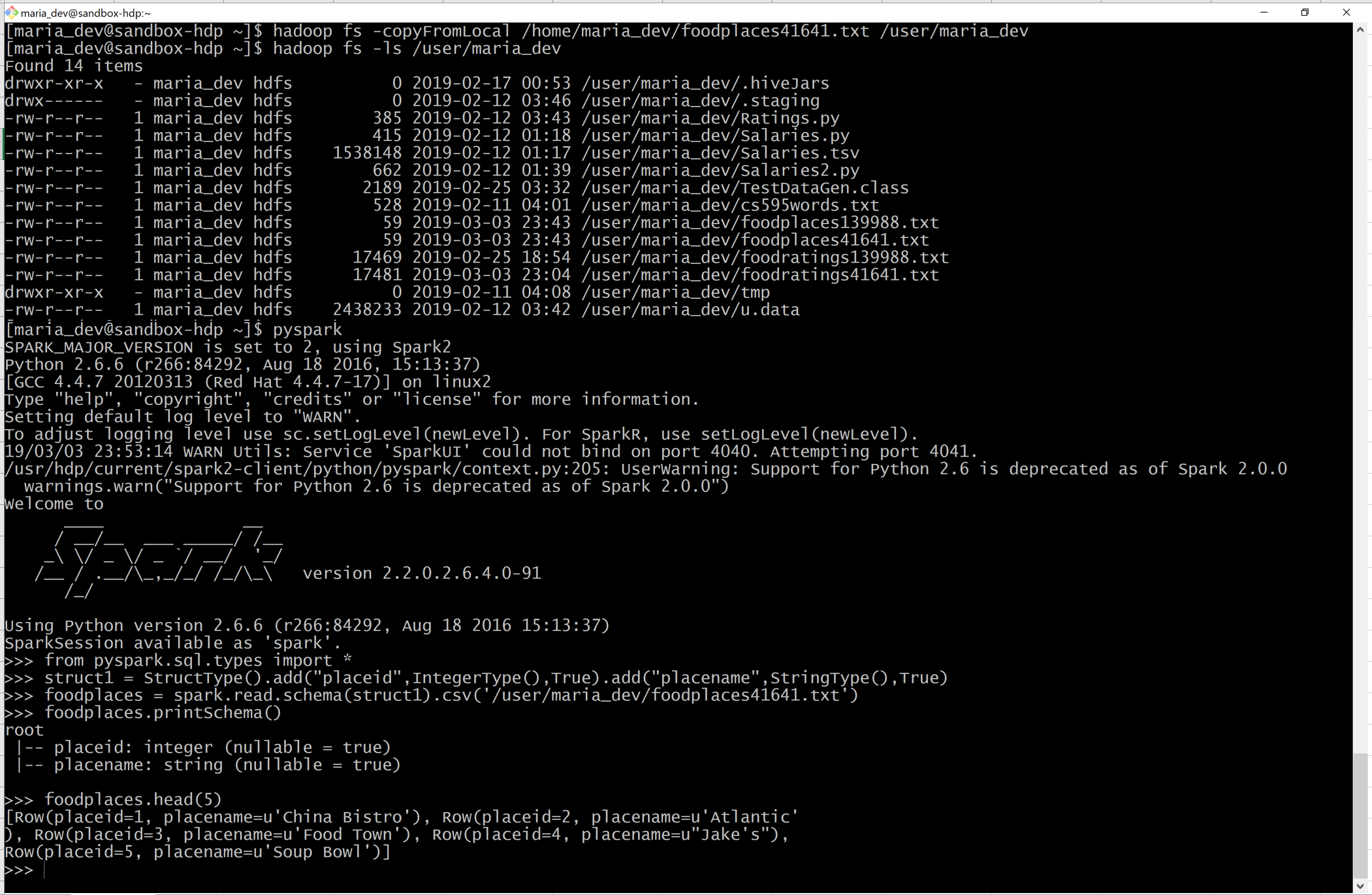
pyspark

from pyspark.sql.types import \*

struct1 = StructType().add("placeid", IntegerType(), True).add("placename", StringType(), True) foodplaces = spark.read.schema(struct1).csv('/user/maria\_dev/foodplaces41641.txt')

foodplaces.printSchema()

foodplaces.head(5)



Exercise 3)

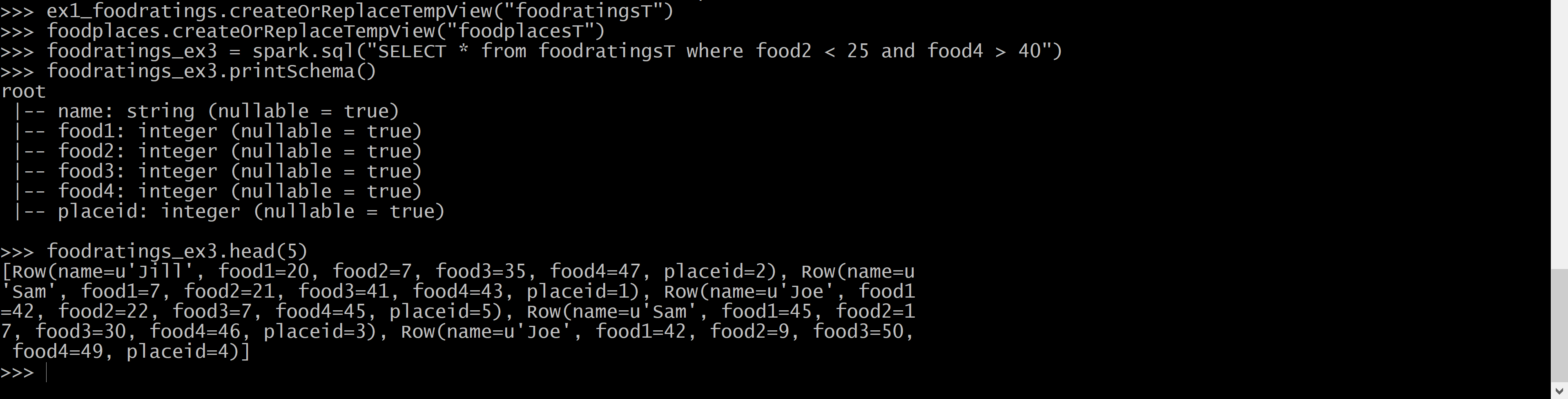
Step A) Register the DataFrames created in exercise 1 and 2 as tables called “foodratingsT” and “foodplacesT”

ex1\_foodratings.createOrReplaceTempView("foodratingsT") foodplaces.createOrReplaceTempView("foodplacesT")

Step B) Use a SQL query on the table “foodratingsT” to create a new DataFrame called foodratings\_ex3 holding records which meet the following condition: food2 < 25 and food4 > 40

Code:

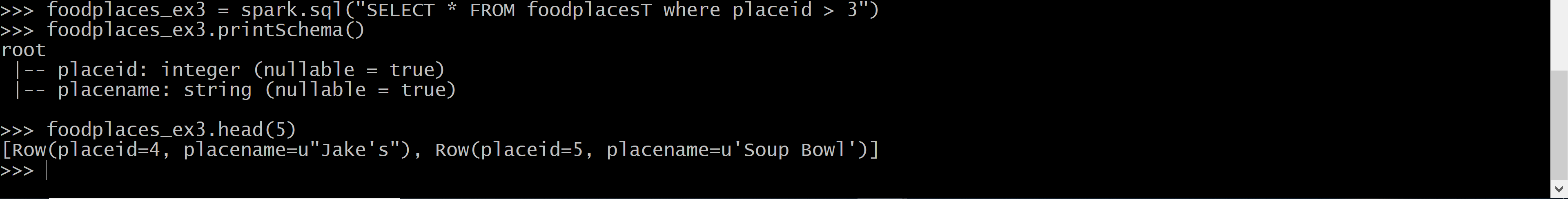
foodratings\_ex3 = spark.sql("SELECT \* from foodratingsT where food2 < 25 and food4 > 40")



Step C) Use a SQL query on the table “foodplacesT” to create a new DataFrame called foodplaces\_ex3 holding records which meet the following condition: placeid > 3

Code:

foodplaces\_ex3 = spark.sql("SELECT \* from foodplacesT where placeid > 3")



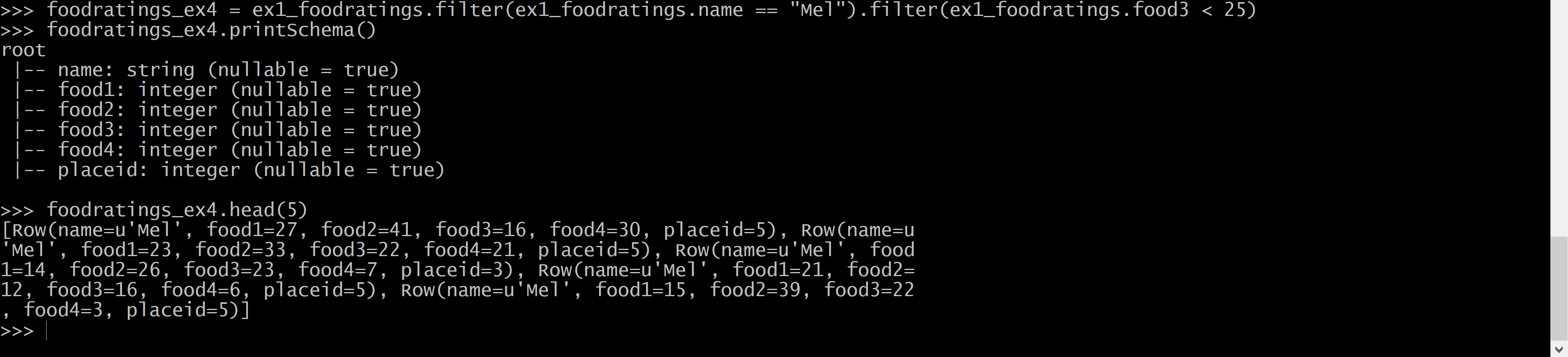
Exercise 4)

Use an operation (not a SQL query) on the DataFrame ‘foodratings’ create 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

Code:

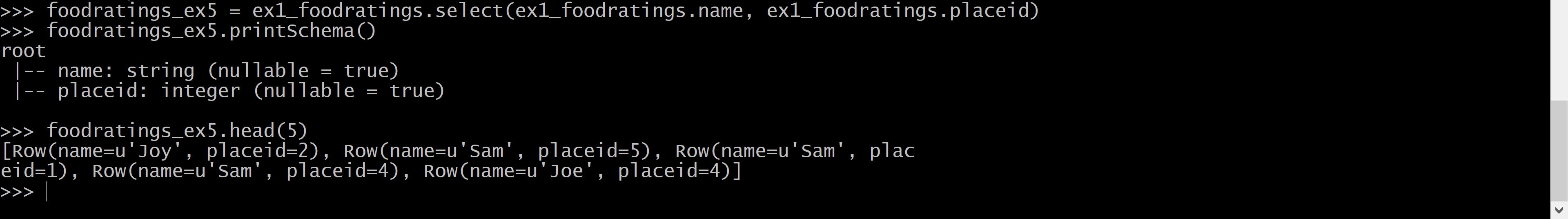
foodratings\_ex4 = ex1\_foodratings.filter(ex1\_foodratings.name == "Mel").filter(ex1\_foodratings.food3 < 25)



Exercise 5) Use an operation (not a SQL query) on the DataFrame ‘foodratings’ create 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

Code:

foodratings\_ex5 = ex1\_foodratings.select(ex1\_foodratings.name, ex1\_foodratings.placeid)



Exercise 6) Use an operation on the DataFrame ‘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.

Code:

ex6 = ex1\_foodratings.join(foodplaces, ex1\_foodratings.placeid == foodplaces.placeid, "inner").drop(ex1\_foodratings.placeid)

