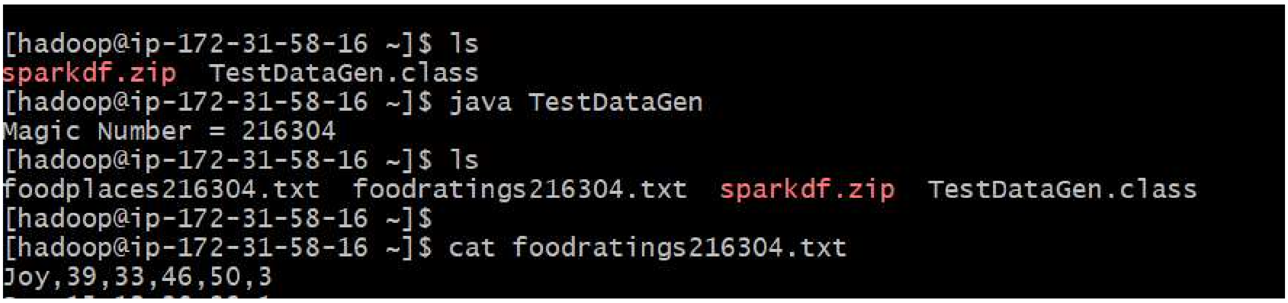
CWID: A20526972

CSP-554 Big Data Technologies

Bairi Rohith Reddy - Assignment #7

Exercise 1)

Magic Number : 216304

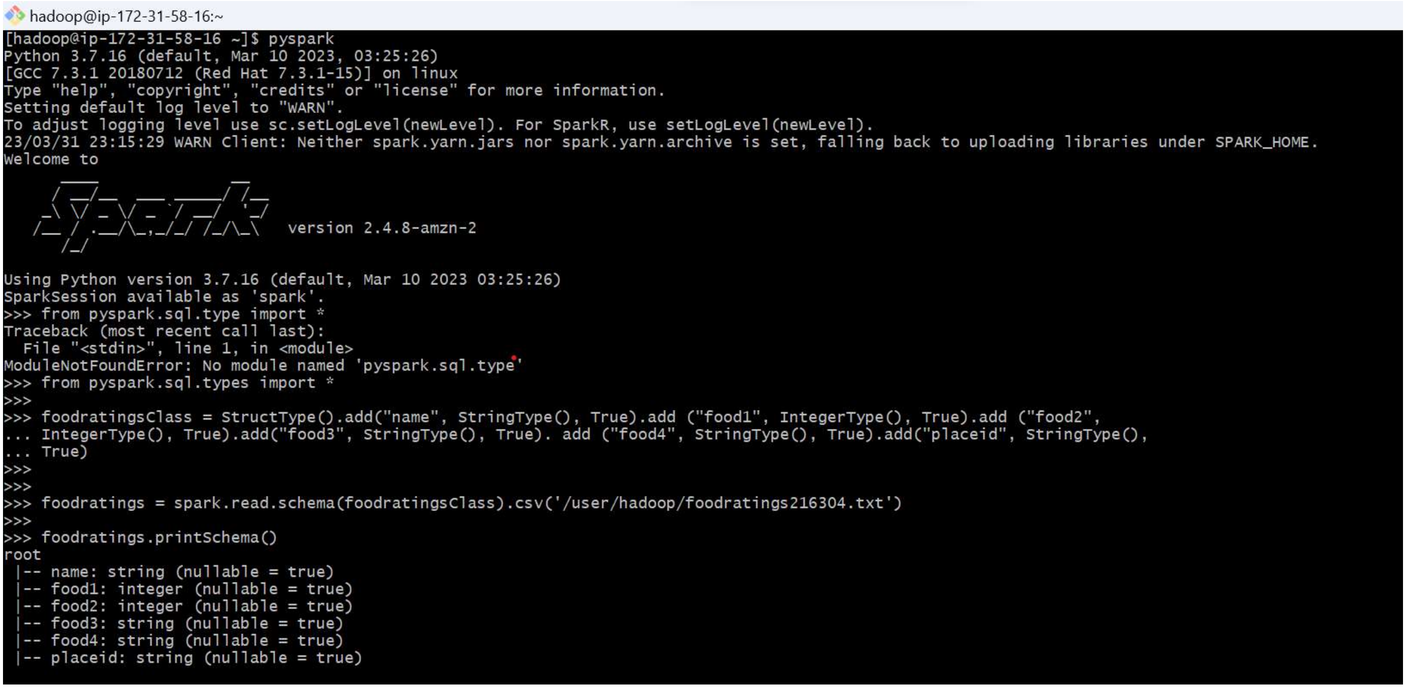


```
[hadoop@ip-172-31-58-16 ~]$ ls
sparkdf.zip  TestDataGen.class
[hadoop@ip-172-31-58-16 ~]$ java TestDataGen
Magic Number = 216304
[hadoop@ip-172-31-58-16 ~]$ ls
foodplaces216304.txt  foodratings216304.txt  sparkdf.zip  TestDataGen.class
[hadoop@ip-172-31-58-16 ~]$
[hadoop@ip-172-31-58-16 ~]$ cat foodratings216304.txt
Joy,39,33,46,50,3
```

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

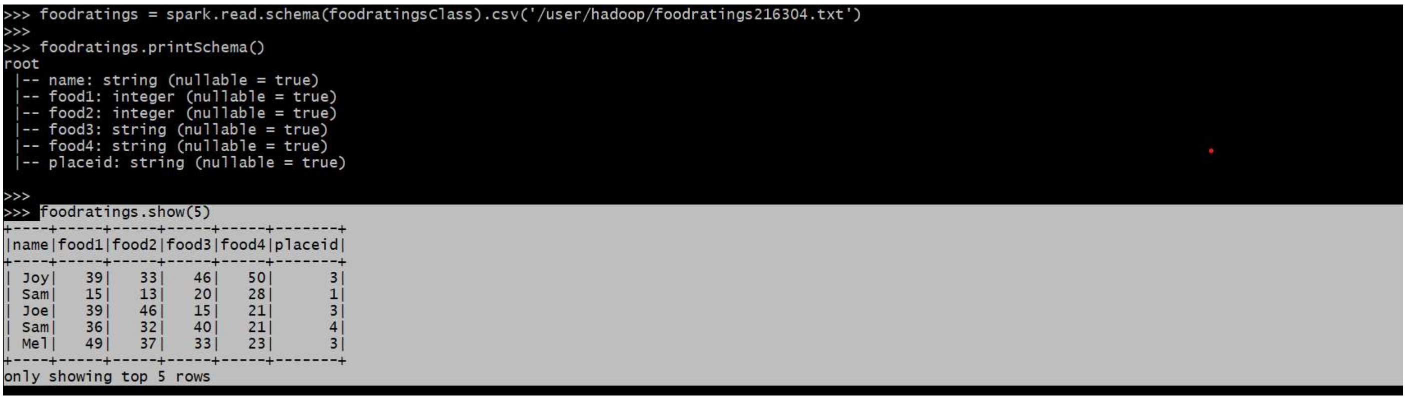
foodratings = spark.read.schema(foodratingsClass).csv('/user/hadoop/foodratings216304.txt')

foodratings.printSchema()



```
[hadoop@ip-172-31-58-16 ~]$ pyspark
Python 3.7.16 (default, Mar 10 2023, 03:25:26)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-15)] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/31 23:15:29 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.8-amzn-2
      /_/

Using Python version 3.7.16 (default, Mar 10 2023 03:25:26)
SparkSession available as 'spark'.
>>> from pyspark.sql.type import *
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ModuleNotFoundError: No module named 'pyspark.sql.type'
>>> from pyspark.sql.types import *
>>>
>>> foodratingsClass = StructType().add("name", StringType(), True).add ("food1", IntegerType(), True).add ("food2",
... IntegerType(), True).add("food3", StringType(), True). add ("food4", StringType(), True).add("placeid", StringType(),
... True)
>>>
>>>
>>> foodratings = spark.read.schema(foodratingsClass).csv('/user/hadoop/foodratings216304.txt')
>>>
>>> foodratings.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: string (nullable = true)
 |-- food4: string (nullable = true)
 |-- placeid: string (nullable = true)
```

foodratings.show(5)



```
>>> foodratings = spark.read.schema(foodratingsClass).csv('/user/hadoop/foodratings216304.txt')
>>>
>>> foodratings.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: string (nullable = true)
 |-- food4: string (nullable = true)
 |-- placeid: string (nullable = true)

>>>
>>> foodratings.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Joy|   39|   33|   46|   50|      3|
| Sam|   15|   13|   20|   28|      1|
| Joe|   39|   46|   15|   21|      3|
| Sam|   36|   32|   40|   21|      4|
| Mel|   49|   37|   33|   23|      3|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows
```

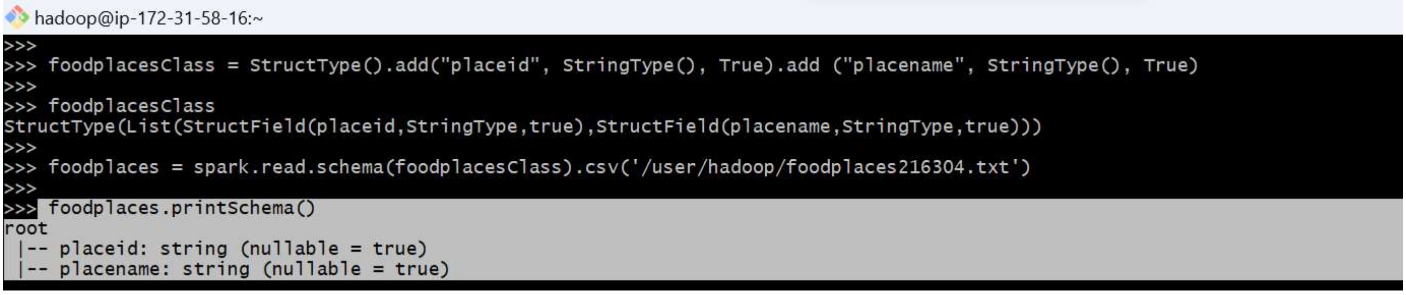
Exercise 2)

foodplacesClass = StructType().add("placeid", StringType(), True).add ("placename", StringType(), True)

foodplacesClass

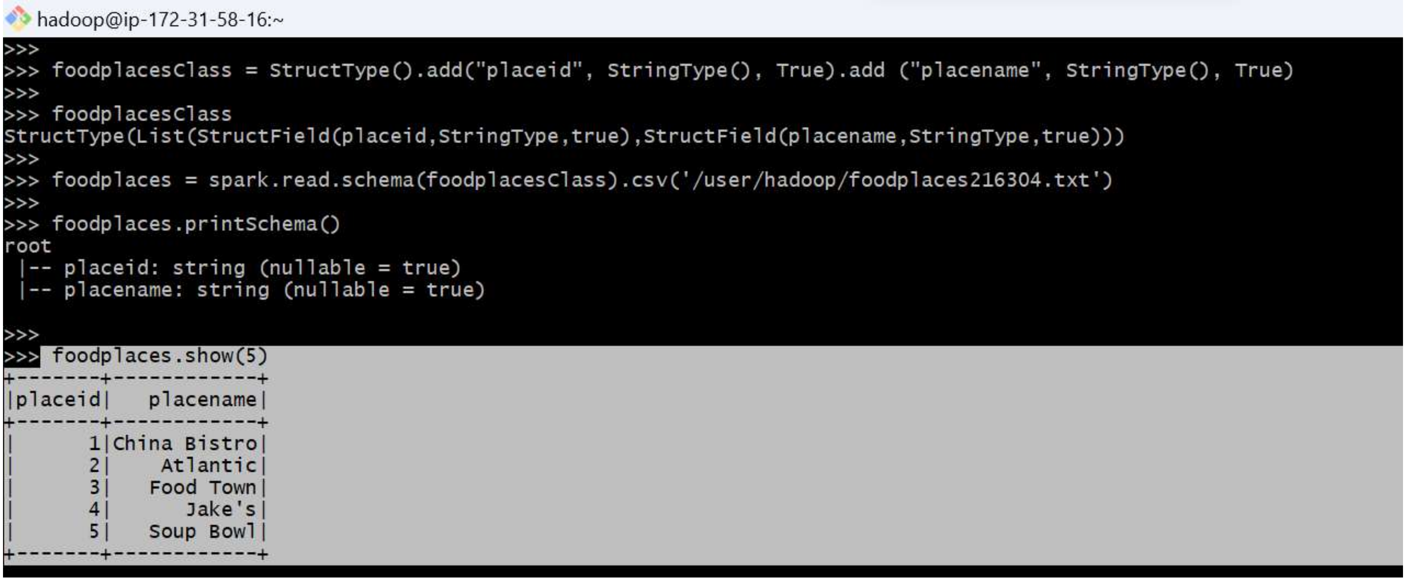
foodplaces = spark.read.schema(foodplacesClass).csv('/user/hadoop/foodplaces216304.txt')

foodplaces.printSchema()



```
>>>
>>> foodplacesClass = StructType().add("placeid", StringType(), True).add ("placename", StringType(), True)
>>>
>>> foodplacesClass
StructType(List(StructField(placeid,StringType,true),StructField(placename,StringType,true)))
>>>
>>> foodplaces = spark.read.schema(foodplacesClass).csv('/user/hadoop/foodplaces216304.txt')
>>>
>>> foodplaces.printSchema()
root
 |-- placeid: string (nullable = true)
 |-- placename: string (nullable = true)
```

foodplaces.show(5)



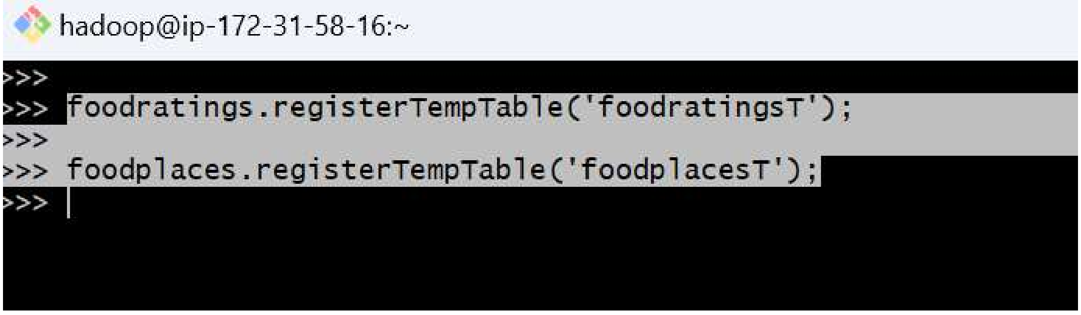
```
>>>
>>> foodplacesClass = StructType().add("placeid", StringType(), True).add ("placename", StringType(), True)
>>>
>>> foodplacesClass
StructType(List(StructField(placeid,StringType,true),StructField(placename,StringType,true)))
>>>
>>> foodplaces = spark.read.schema(foodplacesClass).csv('/user/hadoop/foodplaces216304.txt')
>>>
>>> foodplaces.printSchema()
root
 |-- placeid: string (nullable = true)
 |-- placename: string (nullable = true)

>>>
>>> foodplaces.show(5)
+-------+-----------+
|placeid|  placename|
+-------+-----------+
|      1|China Bistro|
|      2|    Atlantic|
|      3|   Food Town|
|      4|      Jake's|
|      5|   Soup Bowl|
+-------+------------+
```

Exercise 3)

Register the DataFrames

foodratings.registerTempTable('foodratingsT');

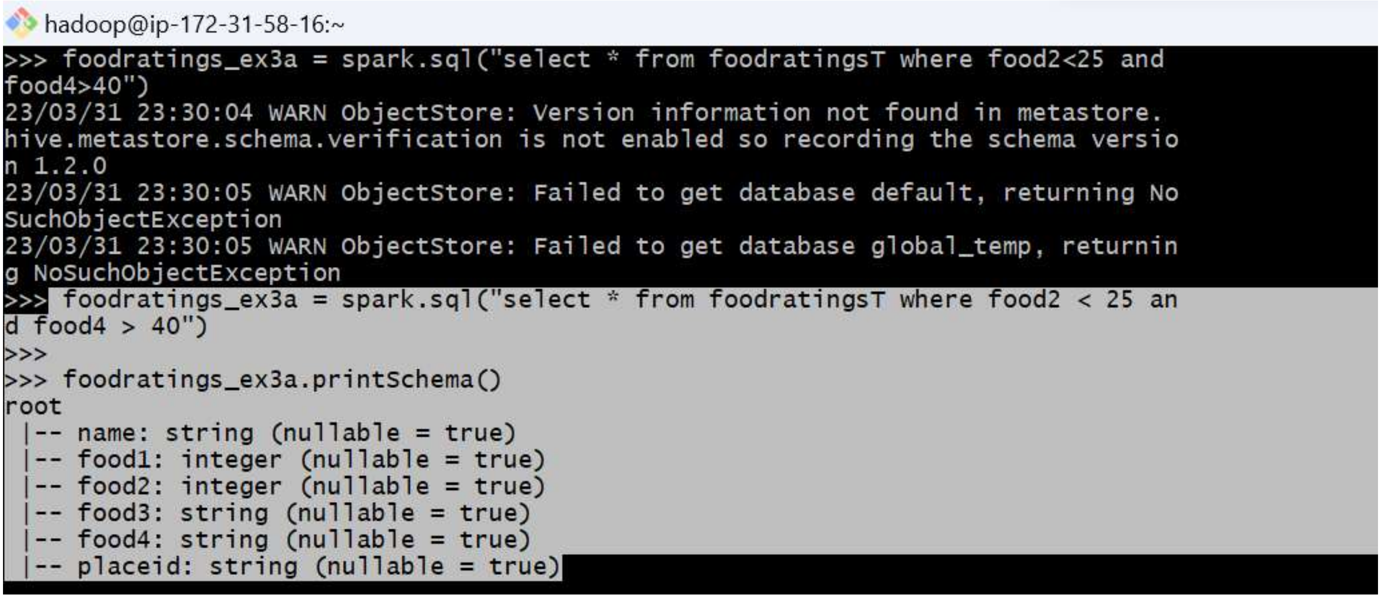
foodplaces.registerTempTable('foodplacesT');



```
>>>
>>> foodratings.registerTempTable('foodratingsT');
>>>
>>> foodplaces.registerTempTable('foodplacesT');
>>> |
```

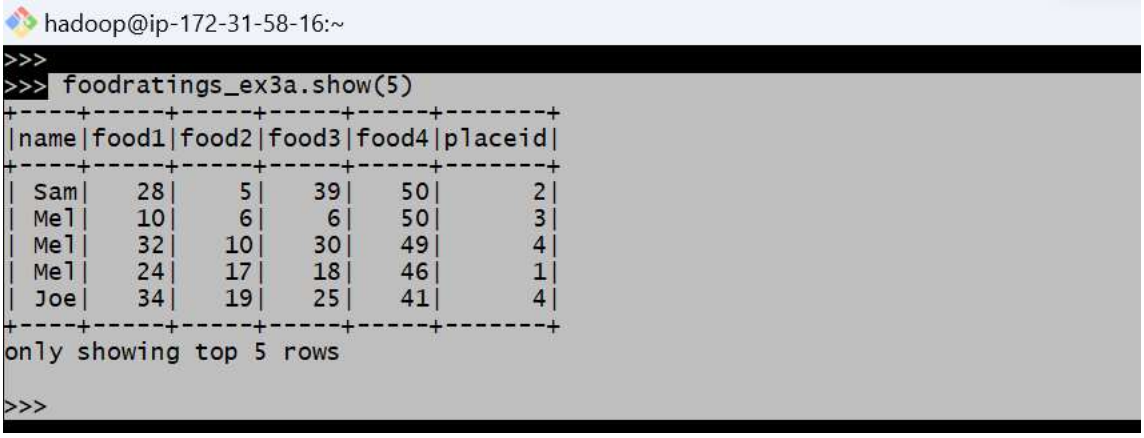
foodratings_ex3a = spark.sql("select * from foodratingsT where food2 < 25 and food4 > 40")

foodratings_ex3a.printSchema()



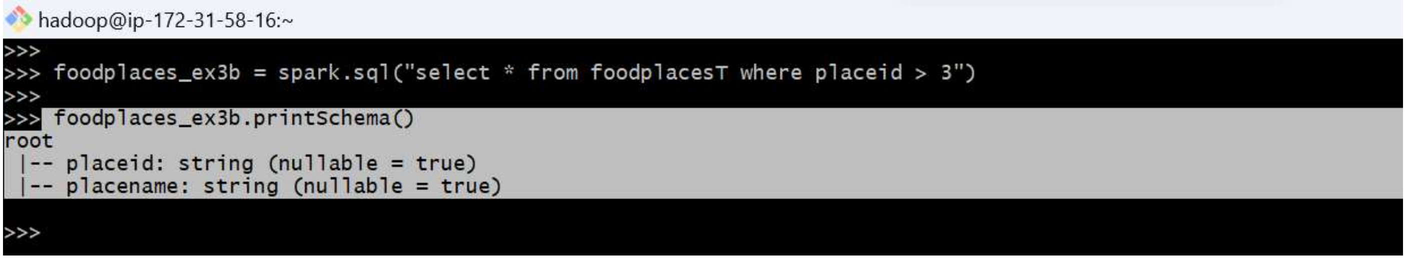
```
>>> foodratings_ex3a = spark.sql("select * from foodratingsT where food2<25 and
food4>40")
23/03/31 23:30:04 WARN ObjectStore: Version information not found in metastore.
hive.metastore.schema.verification is not enabled so recording the schema versio
n 1.2.0
23/03/31 23:30:05 WARN ObjectStore: Failed to get database default, returning No
SuchObjectException
23/03/31 23:30:05 WARN ObjectStore: Failed to get database global_temp, returnin
g NoSuchObjectException
>>> foodratings_ex3a = spark.sql("select * from foodratingsT where food2 < 25 an
d food4 > 40")
>>>
>>> foodratings_ex3a.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: string (nullable = true)
 |-- food4: string (nullable = true)
 |-- placeid: string (nullable = true)
```

foodratings_ex3a.show(5)



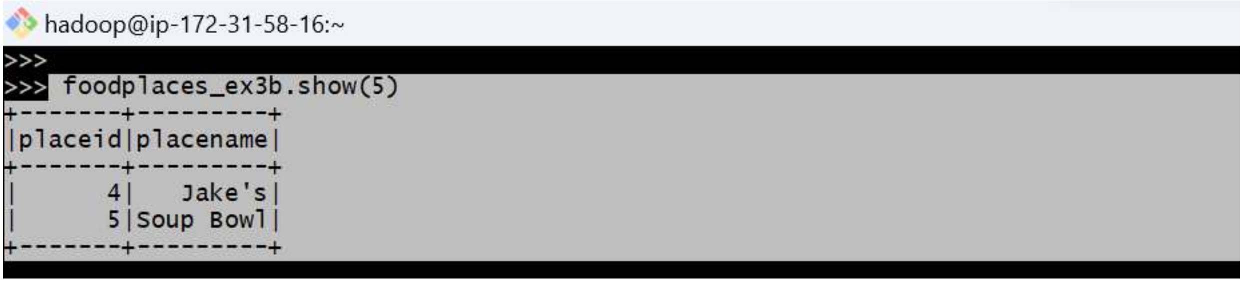
```
>>>
>>> foodratings_ex3a.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Sam|   28|    5|   39|   50|      2|
| Mel|   10|    6|    6|   50|      3|
| Mel|   32|   10|   30|   49|      4|
| Mel|   24|   17|   18|   46|      1|
| Joe|   34|   19|   25|   41|      4|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows

>>>
```

foodplaces_ex3b = spark.sql("select * from foodplacesT where placeid > 3") foodplaces_ex3b.printSchema()



```
>>>
>>> foodplaces_ex3b = spark.sql("select * from foodplacesT where placeid > 3")
>>>
>>> foodplaces_ex3b.printSchema()
root
 |-- placeid: string (nullable = true)
 |-- placename: string (nullable = true)

>>>
```

foodplaces_ex3b.show(5)

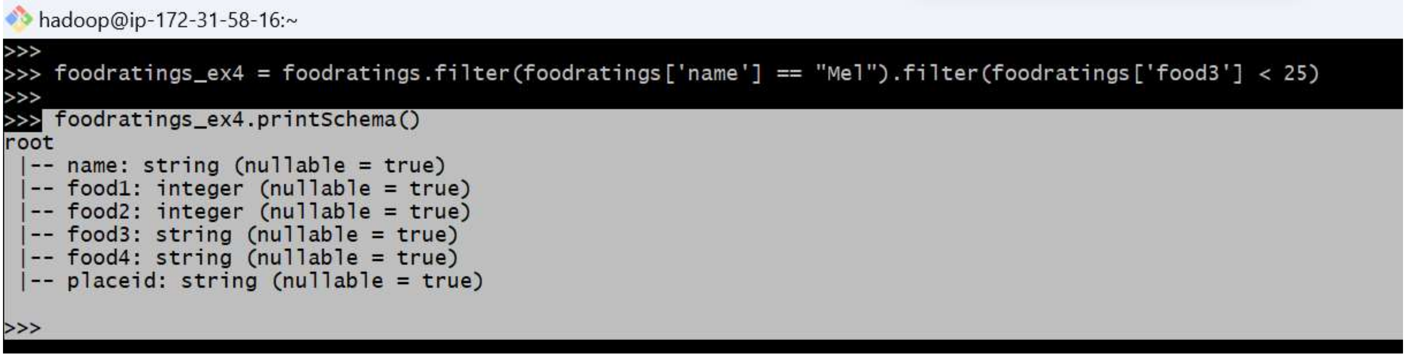


```
>>>
>>> foodplaces_ex3b.show(5)
+-------+---------+
|placeid|placename|
+-------+---------+
|      4|   Jake's|
|      5|Soup Bowl|
+-------+---------+
```

Exercise 4)

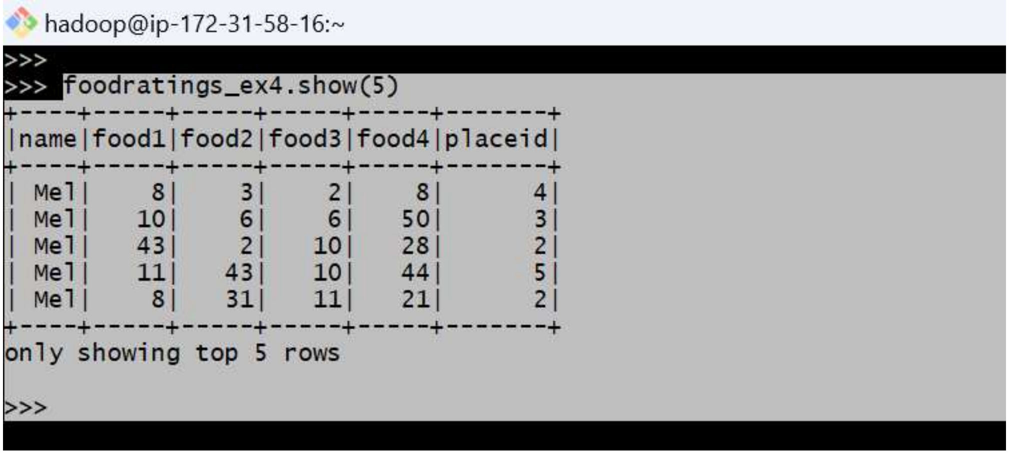
foodratings_ex4 = foodratings.filter(foodratings['name'] == "Mel").filter(foodratings['food3']<25)

foodratings_ex4.printSchema()



```
>>>
>>> foodratings_ex4 = foodratings.filter(foodratings['name'] == "Mel").filter(foodratings['food3'] < 25)
>>>
>>> foodratings_ex4.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: string (nullable = true)
 |-- food4: string (nullable = true)
 |-- placeid: string (nullable = true)

>>>
```

foodratings_ex4.show(5)

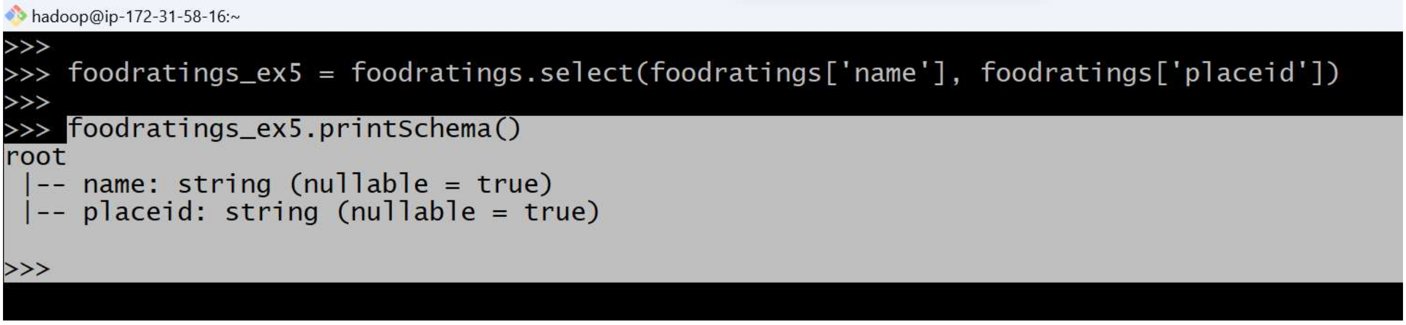


```
>>>
>>> foodratings_ex4.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Mel|    8|    3|    2|    8|      4|
| Mel|   10|    6|    6|   50|      3|
| Mel|   43|    2|   10|   28|      2|
| Mel|   11|   43|   10|   44|      5|
| Mel|    8|   31|   11|   21|      2|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows

>>>
```

Exercise 5)

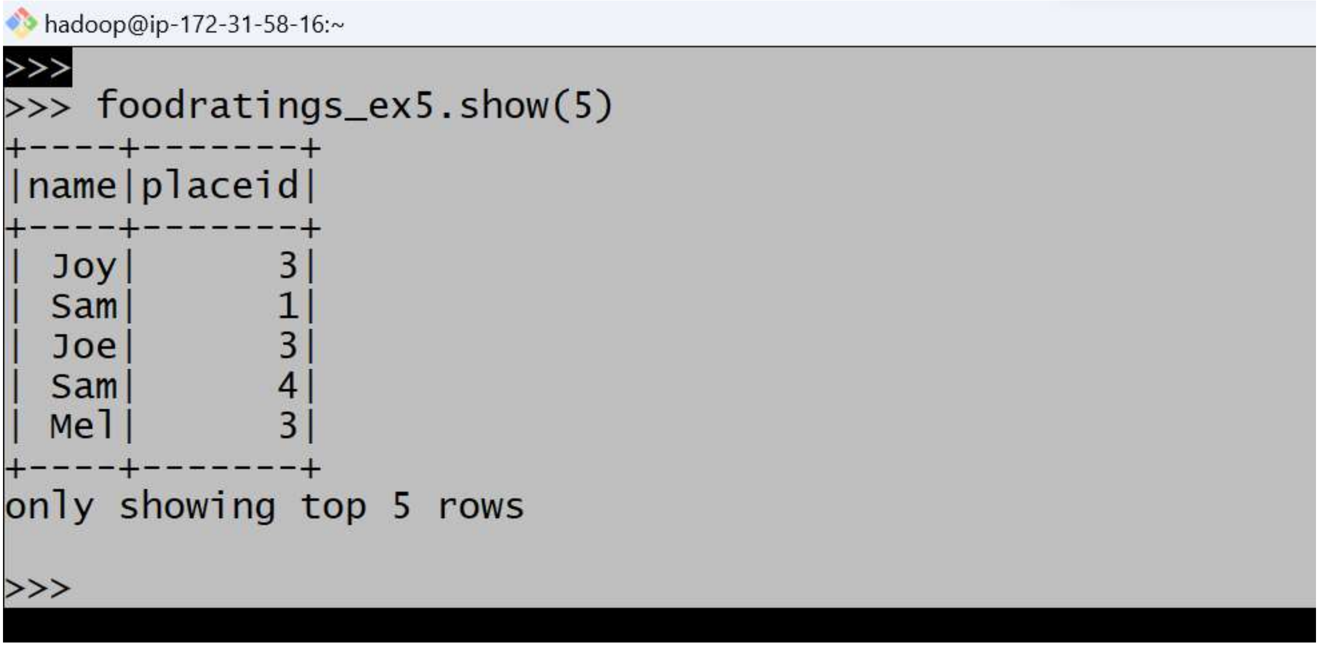
foodratings_ex5 = foodratings.select(foodratings['name'], foodratings['placeid'])

foodratings_ex5.printSchema()



```
>>>
>>> foodratings_ex5 = foodratings.select(foodratings['name'], foodratings['placeid'])
>>>
>>> foodratings_ex5.printSchema()
root
 |-- name: string (nullable = true)
 |-- placeid: string (nullable = true)

>>>
```

foodratings_ex4.show(5)

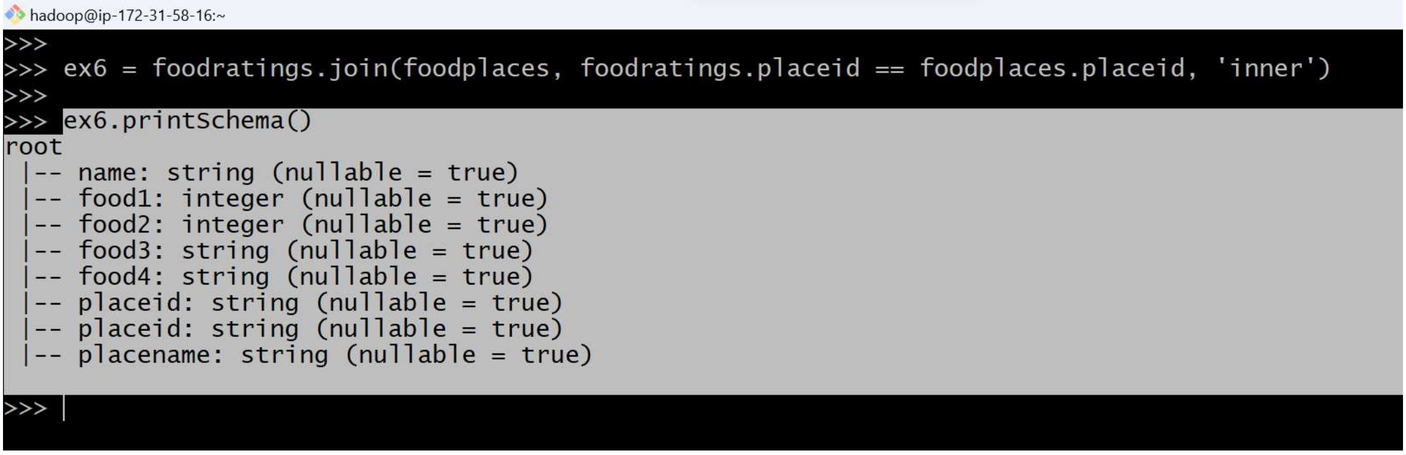


```
>>>
>>> foodratings_ex5.show(5)
+----+-------+
|name|placeid|
+----+-------+
| Joy|      3|
| Sam|      1|
| Joe|      3|
| Sam|      4|
| Mel|      3|
+----+-------+
only showing top 5 rows

>>>
```

Exercise 6)

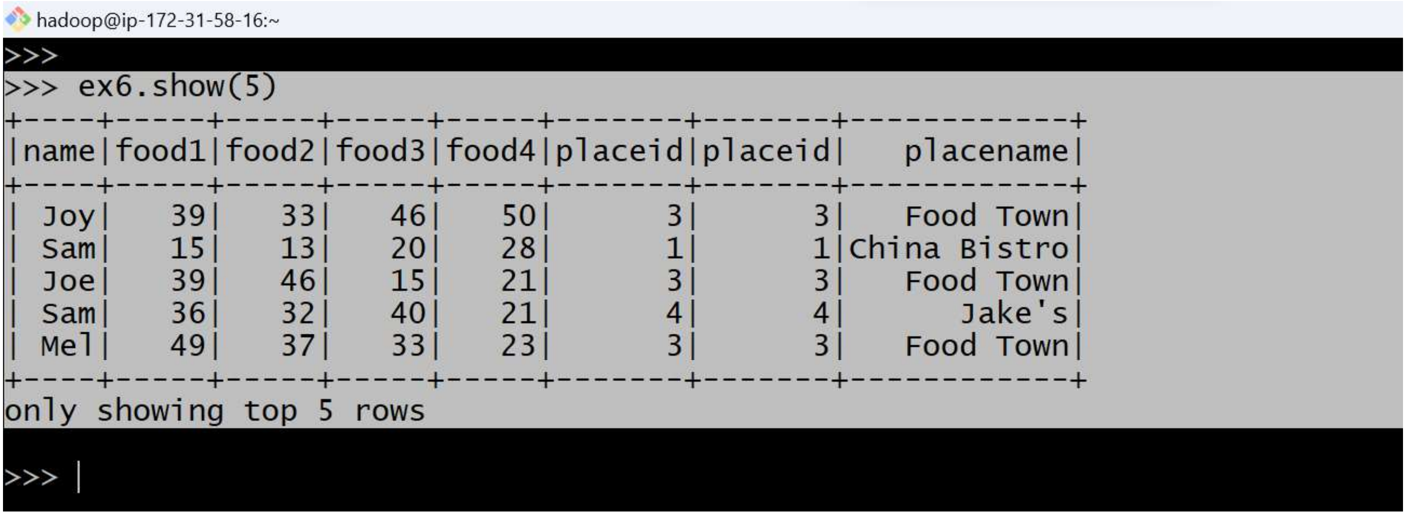
ex6 = foodratings.join(foodplaces, foodratings.placeid == foodplaces.placeid, 'inner')

ex6.printSchema()



```
>>>
>>> ex6 = foodratings.join(foodplaces, foodratings.placeid == foodplaces.placeid, 'inner')
>>>
>>> ex6.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: string (nullable = true)
 |-- food4: string (nullable = true)
 |-- placeid: string (nullable = true)
 |-- placeid: string (nullable = true)
 |-- placename: string (nullable = true)

>>> |
```

ex6.show(5)



```
>>>
>>> ex6.show(5)
+----+-----+-----+-----+-----+-------+-------+------------+
|name|food1|food2|food3|food4|placeid|placeid|   placename|
+----+-----+-----+-----+-----+-------+-------+------------+
| Joy|   39|   33|   46|   50|      3|      3|   Food Town|
| Sam|   15|   13|   20|   28|      1|      1|China Bistro|
| Joe|   39|   46|   15|   21|      3|      3|   Food Town|
| Sam|   36|   32|   40|   21|      4|      4|      Jake's|
| Mel|   49|   37|   33|   23|      3|      3|   Food Town|
+----+-----+-----+-----+-----+-------+-------+------------+
only showing top 5 rows

>>> |
```