Chirag Khandhar

A20438926 | CSP 554 - Big Data Technologies | Fall 2020 | Assignment 7

Magic Number = **61759**

Q 1. As the results of this exercise provide the magic number, the code you execute and screen shots of the following commands:

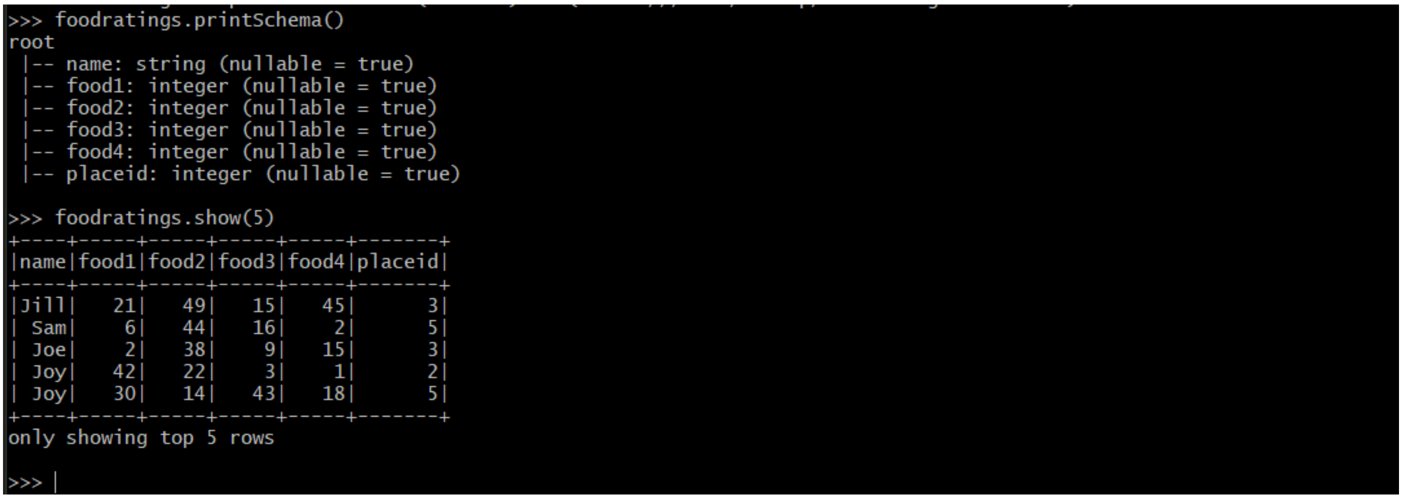
```
foodratings.printSchema()

foodratings.show(5)
```

```
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)

foodratings =
spark.read.schema(struct1).csv('hdfs:///user/hadoop/foodratings61759.csv')
```



```
>>> foodratings.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
|Jill|   21|   49|   15|   45|      3|
| Sam|    6|   44|   16|    2|      5|
| Joe|    2|   38|    9|   15|      3|
| Joy|   42|   22|    3|    1|      2|
| Joy|   30|   14|   43|   18|      5|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows

>>> |
```

Q 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

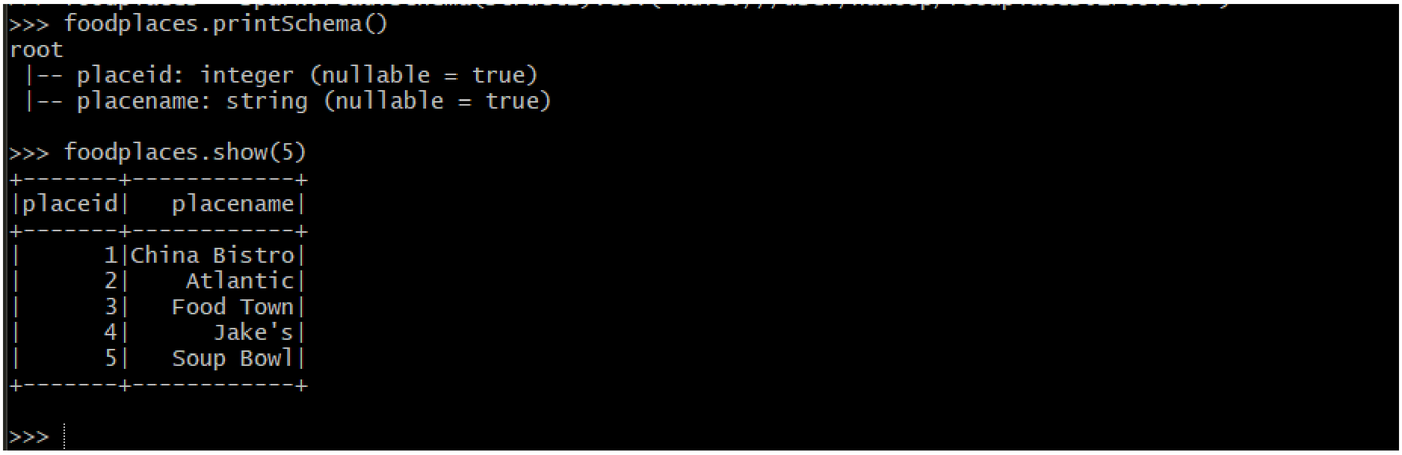
As the results of this exercise provide the code you execute and screen shots of the following commands:

```
foodratings.printSchema()

foodratings.show(5)
```

```
struct2 = StructType().add("placeid", IntegerType(), True).add("placename",
StringType(), True)

foodplaces =
spark.read.schema(struct2).csv('hdfs:///user/hadoop/foodplaces61759.csv’)
```



```
>>> foodplaces.printSchema()
root
 |-- placeid: integer (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|
+-------+------------+

>>>
```

Q 3. Step A

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

foodratings.registerTempTable('foodratingsT')

foodplaces.registerTempTable(' foodplaces T')

Step B

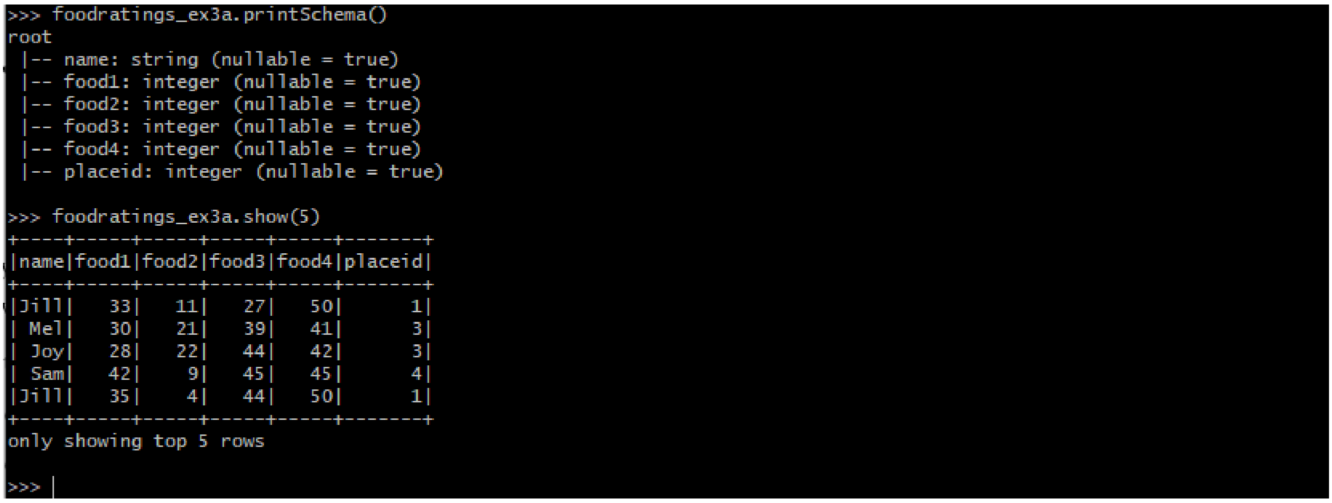
Use a SQL query on the table “foodratingsT” to create a new DataFrame called foodratings_ex3a holding records which meet the following condition: food2 < 25 and food4 > 40. Remember, when defining conditions in your code use maximum parentheses.

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

foodratings_ex3a.printSchema()

foodratings_ex3a.show(5)

foodratings_ex3a = spark.sql('SELECT * FROM foodratingsT WHERE food2 < 25 AND food4 > 40')



```
>>> foodratings_ex3a.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex3a.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
|Jill|   33|   11|   27|   50|      1|
| Mel|   30|   21|   39|   41|      3|
| Joy|   28|   22|   44|   42|      3|
| Sam|   42|    9|   45|   45|      4|
|Jill|   35|    4|   44|   50|      1|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows

>>> |
```

Step C

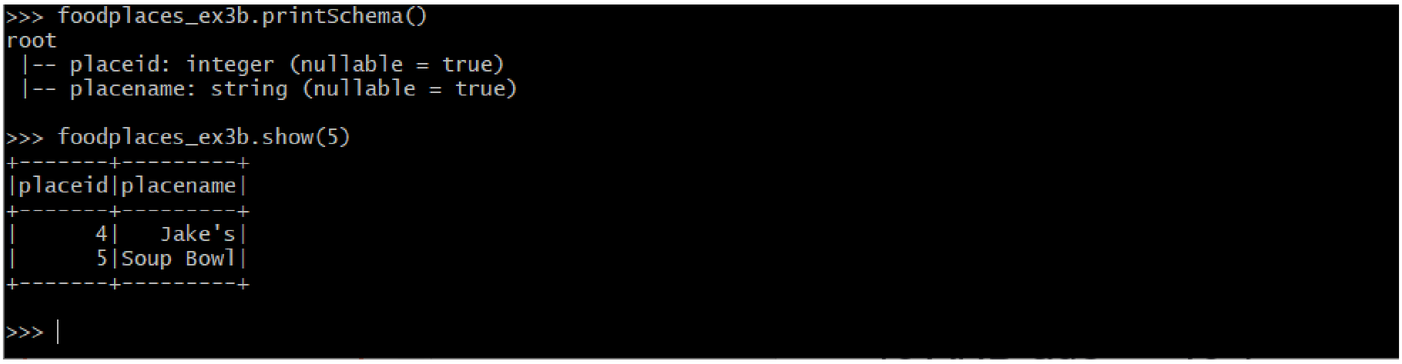
Use a SQL query on the table “foodplacesT” to create a new DataFrame called foodplaces_ex3b holding records which meet the following condition: placeid > 3

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

```
foodplaces_ex3b.printSchema()

foodplaces_ex3b.show(5)
```

foodplaces_ex3b = spark.sql('SELECT * FROM foodplacesT WHERE placeid > 3)



```
>>> foodplaces_ex3b.printSchema()
root
 |-- placeid: integer (nullable = true)
 |-- placename: string (nullable = true)

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

>>> |
```

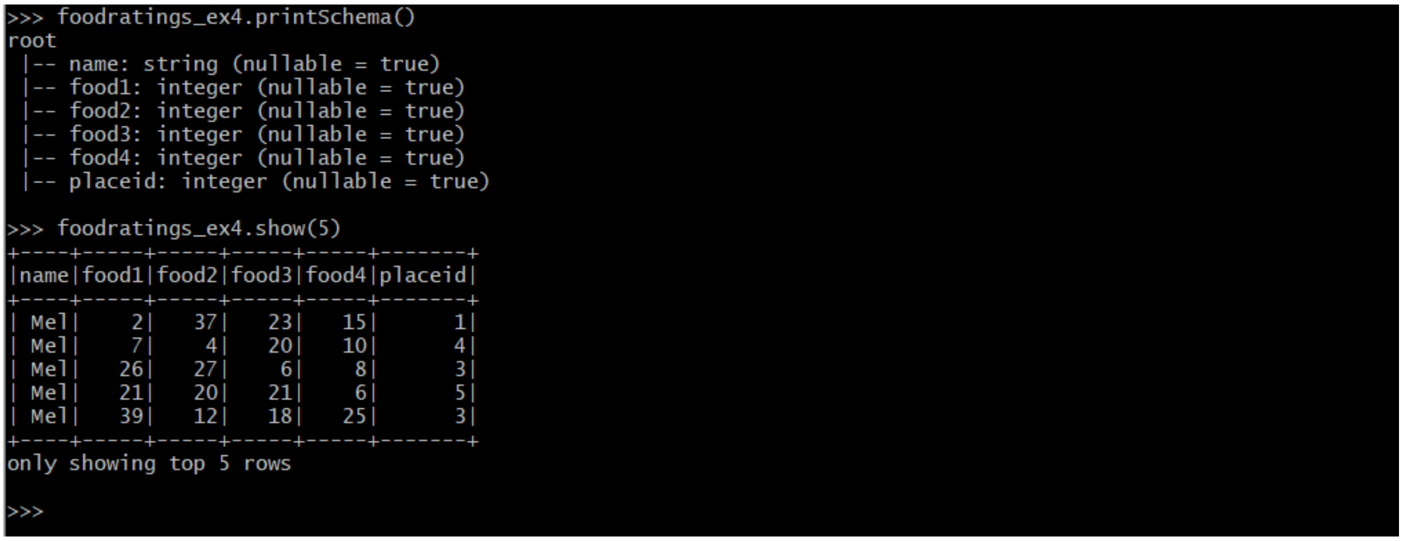
Q 4. Use a transformation (not an SQL query) on the DataFrame 'foodratings' created in exercise 1 to create a new DataFrame called foodratings_ex4 that includes only those records (rows) where the 'name' field is "Mel" and food3 < 25.

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

```
foodratings_ex4.printSchema()

foodratings_ex4.show(5)
```

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



```
>>> foodratings_ex4.printSchema()
root
 |-- name: string (nullable = true)
 |-- food1: integer (nullable = true)
 |-- food2: integer (nullable = true)
 |-- food3: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex4.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Mel|    2|   37|   23|   15|      1|
| Mel|    7|    4|   20|   10|      4|
| Mel|   26|   27|    6|    8|      3|
| Mel|   21|   20|   21|    6|      5|
| Mel|   39|   12|   18|   25|      3|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows

>>>
```

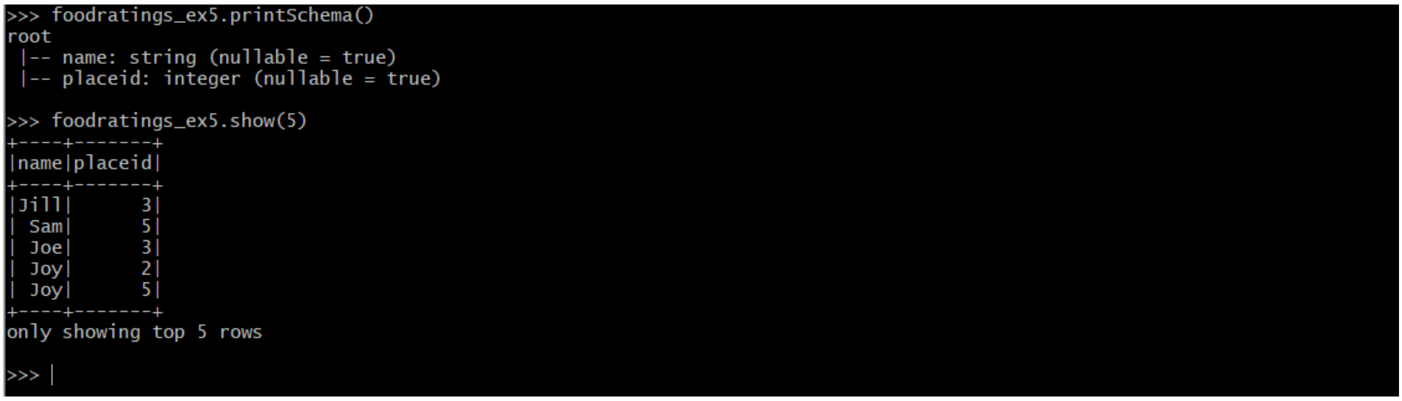
Q 5. Use a transformation (not an SQL query) on the DataFrame 'foodratings' created in exercise 1 to create a new DataFrame called foodratings_ex5 that includes only the columns (fields) 'name' and 'placeid'

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

```
foodratings_ex5.printSchema()

foodratings_ex5.show(5)
```

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



```
>>> foodratings_ex5.printSchema()
root
 |-- name: string (nullable = true)
 |-- placeid: integer (nullable = true)

>>> foodratings_ex5.show(5)
+----+-------+
|name|placeid|
+----+-------+
|Jill|      3|
| Sam|      5|
| Joe|      3|
| Joy|      2|
| Joy|      5|
+----+-------+
only showing top 5 rows

>>> |
```

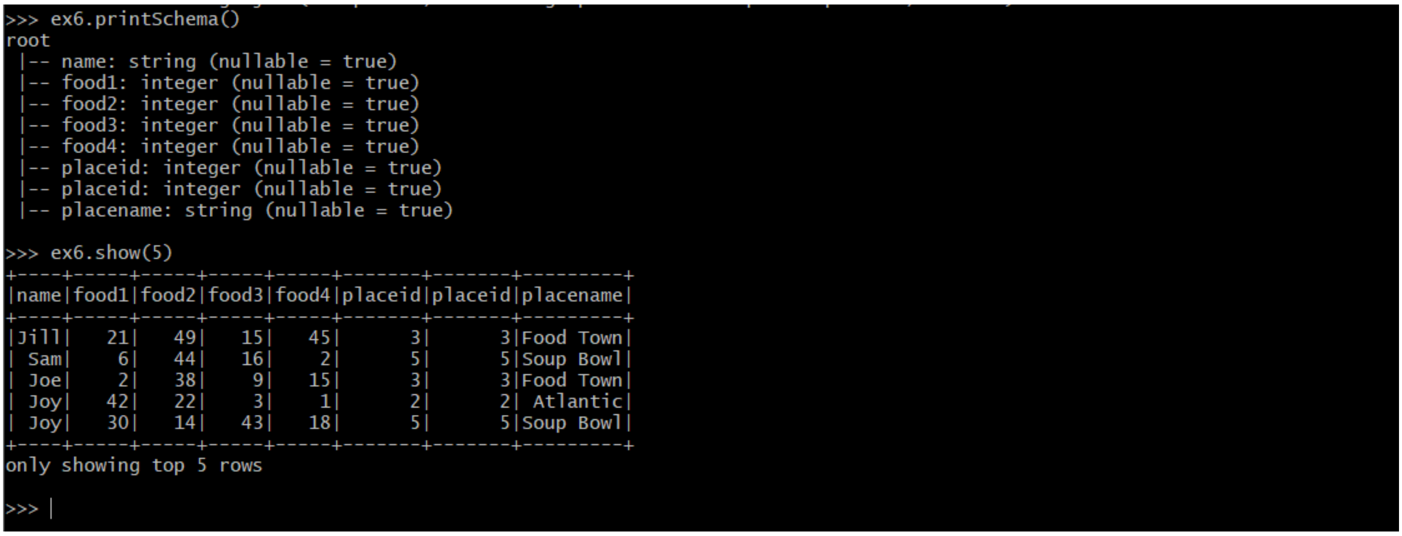
Q 6. Use a transformation (not an SQL query) to create a new DataFrame called ex6 which is the inner join, on placeid, of the DataFrames ‘foodratings; and ‘foodplaces’ created in exercises 1 and 2

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

```
ex6.printSchema()

ex6.show(5)
```

```
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: integer (nullable = true)
 |-- food4: integer (nullable = true)
 |-- placeid: integer (nullable = true)
 |-- placeid: integer (nullable = true)
 |-- placename: string (nullable = true)

>>> ex6.show(5)
+----+-----+-----+-----+-----+-------+-------+---------+
|name|food1|food2|food3|food4|placeid|placeid|placename|
+----+-----+-----+-----+-----+-------+-------+---------+
|Jill|   21|   49|   15|   45|      3|      3|Food Town|
| Sam|    6|   44|   16|    2|      5|      5|Soup Bowl|
| Joe|    2|   38|    9|   15|      3|      3|Food Town|
| Joy|   42|   22|    3|    1|      2|      2| Atlantic|
| Joy|   30|   14|   43|   18|      5|      5|Soup Bowl|
+----+-----+-----+-----+-----+-------+-------+---------+
only showing top 5 rows

>>> |
```