Akash Tanwani A20448831

CSP554—Big Data Technologies

Assignment #07

Exercises

Exercise 1)

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:

foodratings.printSchema()

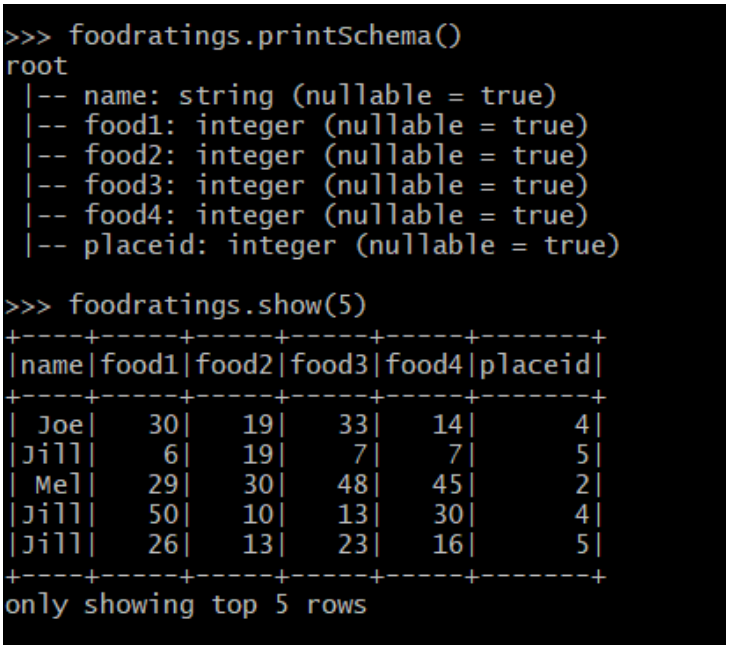
foodratings.show(5)

Ans:

Magic Number = 101264

Command:

```
>>> 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('/user/hadoop/foodratings101264.csv')
>>> foodratings.printSchema()
>>> foodratings.show(5)
```



Akash Tanwani **A20448831**

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:

foodratings.printSchema()

foodratings.show(5)

Command:

```
>>> struct2= StructType().add("placeid",IntegerType(),True).add("placename",StringType(),True)
>>> struct2
StructType(List(StructField(placeid,IntegerType,true),StructField(placename,StringType,true)))
>>> foodplaces = spark.read.schema(struct2).csv('/user/hadoop/foodplaces101264.csv')
>>> foodplaces.printSchema()
>>> foodplaces.show(5)
```

```
>>> 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|
+-------+------------+
```

Exercise 3)

Step A

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

Command:

```
>>> foodratings.registerTempTable('foodratingsT')
>>> foodplaces.registerTempTable('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.printSchema()

foodratings_ex3a.show(5)

Command:

```
>>> foodratings_ex3a=spark.sql("SELECT * FROM foodratingsT WHERE food2<25 AND food4>40")
>>> foodratings_ex3a.printSchema()
>>> foodratings_ex3a.show(5)
```

```
>>> 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|
+----+-----+-----+-----+-----+-------+
| Sam|   28|    3|   47|   41|      3|
| Sam|   10|   11|   19|   42|      3|
| Mel|   10|   15|   12|   47|      3|
| Joy|   18|    5|   18|   48|      2|
| Sam|   23|   15|   23|   46|      5|
+----+-----+-----+-----+-----+-------+
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)

Command:

```
>>> foodplaces_ex3b=spark.sql("SELECT * FROM foodplacesT WHERE placeid > 3")
>>> 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|
+-------+---------+
```

Exercise 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)

Command:

```
>>> foodratings_ex4= foodratings.filter( (foodratings['name']=='Mel') & (foodratings['food3']<25) )
>>> 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|   18|   50|   15|   24|      5|
| Mel|   10|   15|   12|   47|      3|
| Mel|   38|   24|    1|   24|      4|
| Mel|   21|    8|   21|   21|      2|
| Mel|   14|   42|   17|   38|      1|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows
```

Exercise 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)

Command:

```
>>> foodratings_ex5=foodratings.select(foodratings['name'], foodratings['placeid'])
>>> 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|
+----+-------+
| Joe|      4|
|Jill|      5|
| Mel|      2|
|Jill|      4|
|Jill|      5|
+----+-------+
only showing top 5 rows
```

Exercise 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)

Command:

```
>>> ex6 = foodratings.join(foodplaces,foodratings.placeid==foodplaces.placeid, 'inner')
>>> 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|
+----+-----+-----+-----+-----+-------+-------+---------+
| Joe|   30|   19|   33|   14|      4|      4|   Jake's|
|Jill|    6|   19|    7|    7|      5|      5|Soup Bowl|
| Mel|   29|   30|   48|   45|      2|      2| Atlantic|
|Jill|   50|   10|   13|   30|      4|      4|   Jake's|
|Jill|   26|   13|   23|   16|      5|      5|Soup Bowl|
+----+-----+-----+-----+-----+-------+-------+---------+
only showing top 5 rows
```