CSP554—Big Data Technologies

Assignment #7

Exercise 1)

Magic Number Generation –

```
[[hadoop@ip-172-31-73-74 ~]$ java TestDataGen
 Magic Number = 193146
```

Magic Number – 193146

Copy the file to HDFS, say into the /user/hadoop directory

```
[[hadoop@ip-172-31-73-74 ~]$ hdfs dfs -copyFromLocal foodratings193146.txt /user/hadoop/foodratings193146.csv
[[hadoop@ip-172-31-73-74 ~]$ hdfs dfs -copyFromLocal foodplaces193146.txt /user/hadoop/foodplace193146.csv
```

```
[[hadoop@ip-172-31-73-74 ~]$ hdfs dfs -ls /user/hadoop/*193146*
-rw-r--r--   1 hadoop hdfsadmingroup         59 2023-11-18 23:05 /user/hadoop/foodplace193146.csv
-rw-r--r--   1 hadoop hdfsadmingroup      17489 2023-11-18 23:04 /user/hadoop/foodratings193146.csv
```

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)

Magic Number – 193146

Commands –

```
from pyspark.sql.types import *
struct_schema = StructType(
[
StructField("name", StringType(), True),
StructField("food1", IntegerType(), True),
StructField("food2", IntegerType(), True),
StructField("food3", IntegerType(), True),
StructField("food4", IntegerType(), True),
```

StructField("placeid", IntegerType(), True)
]
)
foodratings =
spark.read.schema(struct_schema).csv('/user/hadoop/foodratings193146.csv')
foodratings.printSchema()
foodratings.show(5)

```
[[hadoop@ip-172-31-73-74 ~]$ cat q1.py
from pyspark.sql.types import *

struct_schema = StructType(
        [
                StructField("name", StringType(), True),
                StructField("food1", IntegerType(), True),
                StructField("food2", IntegerType(), True),
                StructField("food3", IntegerType(), True),
                StructField("food4", IntegerType(), True),
                StructField("placeid", IntegerType(), True)
        ]

)

foodratings = spark.read.schema(struct_schema).csv('/user/hadoop/foodratings193146.csv')
foodratings.printSchema()
foodratings.show(5)[hadoop@ip-172-31-73-74 ~]$
```

```
>>> exec(open("/home/hadoop/q1.py").read())
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)

+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Joy|   29|   20|   43|    2|      1|
| Mel|   39|   23|    7|   23|      4|
|Jill|   18|    1|   17|   38|      1|
|Jill|   30|   43|   12|    4|      5|
| Sam|   32|   49|   38|   34|      5|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows
```

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)

Magic Number – 193146
Commands –
from pyspark.sql.types import *
struct_schema = StructType().add("placeid", IntegerType(), True).add("placename", StringType(), True)
foodplaces = spark.read.schema(struct_schema).csv('/user/hadoop/foodplaces193146.csv')
foodplaces.printSchema()
foodplaces.show(5)

```
[hadoop@ip-172-31-73-74 ~]$ cat q2.py
from pyspark.sql.types import *

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

foodplaces = spark.read.schema(struct_schema).csv('/user/hadoop/foodplaces193146.csv')
foodplaces.printSchema()
foodplaces.show(5)[hadoop@ip-172-31-73-74 ~]$ Traceback (most recent call last):
```

```
>>> exec(open("/home/hadoop/q2.py").read())
root
 |-- placeid: integer (nullable = true)
 |-- placename: string (nullable = true)

+-------+-----------+
|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”

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)

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)

ANS:

Magic Number – 193146

Commands –

```
from pyspark.sql.types import *
structfoodratings = StructType(
[
StructField("name", StringType(), True),
StructField("food1",IntegerType(), True),
StructField("food2",IntegerType(), True),
StructField("food3",IntegerType(), True),
StructField("food4",IntegerType(), True),
StructField("placeid",IntegerType(), True)
]
)
structfoodplaces = StructType().add("placeid", IntegerType(),
True).add("placename",StringType(), True)
foodratings =
spark.read.schema(structfoodratings).csv('/user/hadoop/foodratings193146.csv')
foodplaces =
spark.read.schema(structfoodplaces).csv('/user/hadoop/foodplaces193146.csv')
foodratings.createOrReplaceTempView("foodratingsT")
foodplaces.createOrReplaceTempView("foodplacesT")
foodratings_ex3a = spark.sql("SELECT * FROM foodratingsT WHERE food2 < 25 AND
food4 > 40")
foodratings_ex3a.printSchema()
foodratings_ex3a.show(5)
foodplaces_ex3b = spark.sql("SELECT * FROM foodplacesT WHERE placeid > 3")
foodplaces_ex3b.printSchema()
foodplaces_ex3b.show(5)
```

```
[hadoop@ip-172-31-73-74 ~]$ cat q3.py
from pyspark.sql.types import *

structfoodratings = StructType(
       [
              StructField("name", StringType(), True),
              StructField("food1",IntegerType(), True),
              StructField("food2",IntegerType(), True),
              StructField("food3",IntegerType(), True),
              StructField("food4",IntegerType(), True),
              StructField("placeid",IntegerType(), True)
       ]
)

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


foodratings = spark.read.schema(structfoodratings).csv('/user/hadoop/foodratings193146.csv')
foodplaces = spark.read.schema(structfoodplaces).csv('/user/hadoop/foodplaces193146.csv')

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

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

foodplaces_ex3b = spark.sql("SELECT * FROM foodplacesT WHERE placeid > 3")
foodplaces_ex3b.printSchema()
foodplaces_ex3b.show(5)
```

```
>>> exec(open("/home/hadoop/q3.py").read())
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)

+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Joy|    5|   20|    6|   43|      4|
| Joe|   11|   10|   20|   48|      5|
| Joy|   15|   21|   20|   50|      5|
|Jill|   14|   20|   11|   48|      5|
| Joe|   37|   19|   45|   43|      3|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows

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

+-------+---------+
|placeid|placename|
+-------+---------+
|      4|   Jake's|
|      5|Soup Bowl|
+-------+---------+
```

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:

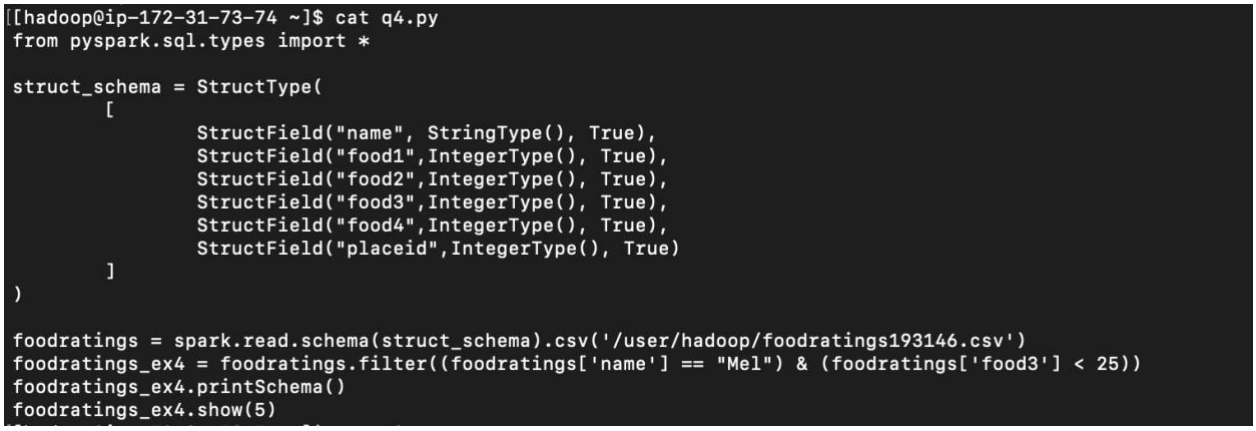
foodratings_ex4.printSchema() foodratings_ex4.show(5)

ANS:

Magic Number – 193146

Commands –

```
from pyspark.sql.types import *
struct_schema = StructType(
[
StructField("name", StringType(), True),
StructField("food1",IntegerType(), True),
StructField("food2",IntegerType(), True),
StructField("food3",IntegerType(), True),
StructField("food4",IntegerType(), True),
StructField("placeid",IntegerType(), True)
]
)
foodratings =
spark.read.schema(struct_schema).csv('/user/hadoop/foodratings193146.csv')
foodratings_ex4 = foodratings.filter((foodratings['name'] == "Mel") & (foodratings['food3']
< 25))
foodratings_ex4.printSchema()
foodratings_ex4.show(5)
```



```
>>> exec(open("/home/hadoop/q4.py").read())
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)

+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Mel|   39|   23|    7|   23|      4|
| Mel|   16|   45|   18|   27|      1|
| Mel|   25|   30|    8|   23|      4|
| Mel|   12|    8|   21|   19|      5|
| Mel|   24|    5|    6|   14|      3|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows
```

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.printSchema() foodratings_ex5.show(5)

ANS:

Magic Number – 193146
Commands –

```
from pyspark.sql.types import *
struct_schema = StructType(
[
StructField("name", StringType(), True),
StructField("food1",IntegerType(), True),
StructField("food2",IntegerType(), True),
StructField("food3",IntegerType(), True),
StructField("food4",IntegerType(), True),
```

StructField("placeid",IntegerType(), True)
]
)
foodratings = spark.read.schema(struct_schema).csv('/user/hadoop/foodratings193146.csv')
foodratings_ex5 = foodratings.select(foodratings['name'],foodratings['placeid'])
foodratings_ex5.printSchema()
foodratings_ex5.show(5)

```
[hadoop@ip-172-31-73-74 ~]$ cat q5.py
from pyspark.sql.types import *

struct_schema = StructType(
        [
                StructField("name", StringType(), True),
                StructField("food1",IntegerType(), True),
                StructField("food2",IntegerType(), True),
                StructField("food3",IntegerType(), True),
                StructField("food4",IntegerType(), True),
                StructField("placeid",IntegerType(), True)
        ]
)

foodratings = spark.read.schema(struct_schema).csv('/user/hadoop/foodratings193146.csv')
foodratings_ex5 = foodratings.select(foodratings['name'],foodratings['placeid'])
foodratings_ex5.printSchema()
foodratings_ex5.show(5)
```

```
>>> exec(open("/home/hadoop/q5.py").read())
root
 |-- name: string (nullable = true)
 |-- placeid: integer (nullable = true)

+----+-------+
|name|placeid|
+----+-------+
| Joy|      1|
| Mel|      4|
|Jill|      1|
|Jill|      5|
| Sam|      5|
+----+-------+
only showing top 5 rows
```

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.printSchema() ex6.show(5)

ANS:

Magic Number – 193146
Commands –
from pyspark.sql.types import *
structfoodratings = StructType(
[
StructField("name", StringType(), True),
StructField("food1",IntegerType(), True),
StructField("food2",IntegerType(), True),
StructField("food3",IntegerType(), True),
StructField("food4",IntegerType(), True),
StructField("placeid",IntegerType(), True)
]
)
structfoodplaces = StructType().add("placeid", IntegerType(),
True).add("placename",StringType(), True)
foodratings = spark.read.schema(structfoodratings).csv('/user/hadoop/foodratings193146.csv')
foodplaces = spark.read.schema(structfoodplaces).csv('/user/hadoop/foodplaces193146.csv')
ex6 = foodratings.join(foodplaces, foodratings.placeid == foodplaces.placeid, 'inner')
ex6.printSchema()
ex6.show(5)

```
[[hadoop@ip-172-31-73-74 ~]$ cat q6.py
from pyspark.sql.types import *

structfoodratings = StructType(
        [
                StructField("name", StringType(), True),
                StructField("food1",IntegerType(), True),
                StructField("food2",IntegerType(), True),
                StructField("food3",IntegerType(), True),
                StructField("food4",IntegerType(), True),
                StructField("placeid",IntegerType(), True)
        ]
)

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


foodratings = spark.read.schema(structfoodratings).csv('/user/hadoop/foodratings193146.csv')
foodplaces = spark.read.schema(structfoodplaces).csv('/user/hadoop/foodplaces193146.csv')

ex6 = foodratings.join(foodplaces, foodratings.placeid == foodplaces.placeid, 'inner')
ex6.printSchema()
ex6.show(5)
```

```
>>> exec(open("/home/hadoop/q6.py").read())
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)

+----+-----+-----+-----+-----+-------+-------+------------+
|name|food1|food2|food3|food4|placeid|placeid|   placename|
+----+-----+-----+-----+-----+-------+-------+------------+
| Joy|   29|   20|   43|    2|      1|      1|China Bistro|
| Mel|   39|   23|    7|   23|      4|      4|      Jake's|
|Jill|   18|    1|   17|   38|      1|      1|China Bistro|
|Jill|   30|   43|   12|    4|      5|      5|   Soup Bowl|
| Sam|   32|   49|   38|   34|      5|      5|   Soup Bowl|
+----+-----+-----+-----+-----+-------+-------+------------+
only showing top 5 rows
```