Name – Rishabh Jain
Mail – rjain35@hawk.iit.edu
Course – CSP 554

Assignment 7

Exercise 1)
Generating Magic Number

```
Desktop — hadoop@ip-172-31-83-226:~ —
[[hadoop@ip-172-31-83-226 ~]$ java TestDataGen
Magic Number = 145715
[[hadoop@ip-172-31-83-226 ~]$ ls
foodplaces145715.txt  foodratings145715.txt  TestDataGen.class
[hadoop@ip-172-31-83-226 ~]$
```

Step B
Use the TestDataGen program from previous assignments to generate new data files. Copy both generated files to the HDFS directory "/user/hadoop"

Sol)

```
[[hadoop@ip-172-31-83-226 ~]$ hdfs dfs -copyFromLocal foodratings145715.txt /user/hadoop/foodratings145715.csv
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/emr/emrfs/lib/slf4j-log4j12-1.7.12.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
```

Step C
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:

Sol)

```
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.8-amzn-0
      /_/

Using Python version 3.7.10 (default, Jun  3 2021 00:02:01)
SparkSession available as 'spark'.
[>>> from pyspark.sql.types import *
>>> struct1 = 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(struct1).csv('/user/hadoop/foodratings145715.csv')
>>> |
```

Name – Rishabh Jain
Mail – rjain35@hawk.iit.edu
Course – CSP 554

foodratings.printSchema()

```
>>> 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)

```
>>> foodratings.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Mel|    5|   26|   33|   25|      4|
| Joe|   42|   27|   18|   37|      2|
|Jill|   30|   32|    8|   46|      5|
| Joy|    5|    5|   44|   15|      2|
| Mel|   49|   37|   18|    1|      4|
+----+-----+-----+-----+-----+-------+
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:

```
>>> struct1 = StructType().add("placeid", IntegerType(),True).add("placename",StringType(), True)
>>> foodplaces = spark.read.schema(struct1).csv('/user/hadoop/foodplaces145715.csv')
```

foodplaces.printSchema()

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

foodplaces.show(5)

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

foodratings.printSchema()

```
>>> 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)
```

Name – Rishabh Jain
Mail – rjain35@hawk.iit.edu
Course – CSP 554
foodratings.show(5)

```
>>> foodratings.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Mel|    5|   26|   33|   25|      4|
| Joe|   42|   27|   18|   37|      2|
|Jill|   30|   32|    8|   46|      5|
| Joy|    5|    5|   44|   15|      2|
| Mel|   49|   37|   18|    1|      4|
+----+-----+-----+-----+-----+-------+
only showing top 5 rows
```

Exercise 3)

Step A

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

Sol)

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

```
>>> foodratings.createOrReplaceTempView("foodratingsT")
>>> foodplaces.createOrReplaceTempView("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.

Sol)

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

```
>>> foodratings_ex3a = spark.sql("SELECT * FROM foodratingsT WHERE food2 < 25 AND food4 > 40")
22/02/28 05:33:47 WARN ObjectStore: Version information not found in metastore. hive.metastore.schema.veri
22/02/28 05:33:47 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
22/02/28 05:33:48 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
```

foodratings_ex3a.printSchema()

```
>>> 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)
```

Name – Rishabh Jain
Mail – rjain35@hawk.iit.edu
Course – CSP 554
foodratings_ex3a.show(5)

```
>>> foodratings_ex3a.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Joy|   10|   20|   48|   45|      3|
| Sam|   33|   18|   31|   49|      3|
| Mel|   48|   19|   33|   49|      4|
| Joy|   27|   21|   23|   42|      5|
| Joe|   27|   16|   33|   41|      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

Sol)

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 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.

Sol)

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

Name – Rishabh Jain
Mail – rjain35@hawk.iit.edu
Course – CSP 554

foodratings_ex4.printSchema()

```
>>> 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)

```
>>> foodratings_ex4.show(5)
+----+-----+-----+-----+-----+-------+
|name|food1|food2|food3|food4|placeid|
+----+-----+-----+-----+-----+-------+
| Mel|   49|   37|   18|    1|      4|
| Mel|   36|   46|   17|   33|      1|
| Mel|   27|   42|   21|   13|      4|
| Mel|   11|   39|   17|   11|      4|
| Mel|   33|   32|   16|   15|      5|
+----+-----+-----+-----+-----+-------+
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'

Sol)

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: integer (nullable = true)
```

foodratings_ex5.show(5)

```
>>> foodratings_ex5.show(5)
+----+-------+
|name|placeid|
+----+-------+
| Mel|      4|
| Joe|      2|
|Jill|      5|
| Joy|      2|
| Mel|      4|
+----+-------+
only showing top 5 rows
```

Name – Rishabh Jain
Mail – rjain35@hawk.iit.edu
Course – CSP 554

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

Sol)

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

ex6.show(5)

```
>>> ex6.show(5)
+----+-----+-----+-----+-----+-------+-------+---------+
|name|food1|food2|food3|food4|placeid|placeid|placename|
+----+-----+-----+-----+-----+-------+-------+---------+
| Mel|    5|   26|   33|   25|      4|      4|   Jake's|
| Joe|   42|   27|   18|   37|      2|      2| Atlantic|
|Jill|   30|   32|    8|   46|      5|      5|Soup Bowl|
| Joy|    5|    5|   44|   15|      2|      2| Atlantic|
| Mel|   49|   37|   18|    1|      4|      4|   Jake's|
+----+-----+-----+-----+-----+-------+-------+---------+
only showing top 5 rows
```