Girish Rajani
A20503736
CSP-554 Big Data Technologies
Homework 5

After successfully connecting to the master node via SSH command, the TestDataGen.class from the previous assignment was uploaded to the /home/hadoop directory via scp.

```
giris@LAPTOP-18TD7ESB MINGW64 ~
$ scp -i d:/users/giris/downloads/emr-key-pair.pem d:/users/giris/downloads/Test
DataGen.class hadoop@ec2-100-24-238-210.compute-1.amazonaws.com:/home/hadoop
TestDataGen.class                             100% 2189    98.1KB/s   00:00
```

Figure 1 - Showing TestDataGen.class uploaded to /home/hadoop directory using scp

```
[hadoop@ip-172-31-48-207 ~]$ java TestDataGen
Magic Number = 89640
```

Figure 2 - Showing the file TestDataGen.class executed using “java TestDataGen” and generating a magic number, 89640

```
[hadoop@ip-172-31-48-207 ~]$ hadoop fs -put /home/hadoop/foodratings89640.txt /u
ser/hadoop
```

Figure 3 - Using -put command to copy foodratings89640.txt file from the /home/hadoop directory to HDFS /user/hadoop directory

```
[hadoop@ip-172-31-48-207 ~]$ hadoop fs -put /home/hadoop/foodplaces89640.txt /us
er/hadoop
[hadoop@ip-172-31-48-207 ~]$ hadoop fs -ls /user/hadoop
Found 2 items
-rw-r--r--   1 hadoop hdfsadmingroup         59 2022-10-05 16:12 /user/hadoop/fo
odplaces89640.txt
-rw-r--r--   1 hadoop hdfsadmingroup      17517 2022-10-05 16:11 /user/hadoop/fo
odratings89640.txt
```

Figure 4 - Using -put command to copy foodplaces89640.txt file from the /home/hadoop directory to HDFS /user/hadoop directory and using the ls command to confirm that both txt files were successfully copied to /user/hadoop directory

Exercise 1) 2 points

```
grunt> food_ratings = LOAD'/user/hadoop/foodratings89640.txt'
>> Using PigStorage(',')
>> AS (name:chararray, f1:int, f2:int, f3:int, f4:int, placeid:int);
```

Figure 5 - Showing Pig statement used to load the foodratings file as a relation

```
grunt> DESCRIBE food_ratings;
food_ratings: {name: chararray,f1: int,f2: int,f3: int,f4: int,placeid: int}
```

Figure 6 - Showing command 'DESCRIBE food_ratings;' executed to show the summary of food_ratings relation

Commands Used:
food_ratings = LOAD'/user/hadoop/foodratings89640.txt' USING PigStorage(',') AS (name:chararray, f1:int, f2:int, f3:int, f4:int, placeid:int);

DESCRIBE food_ratings;

Magic Number: **89640**

Exercise 2) 2 points

```
grunt> food_ratings_subset = FOREACH food_ratings GENERATE name, f4;
grunt> STORE food_ratings_subset INTO '/user/hadoop/fr_subset'
>> USING PigStorage(',');
```

Figure 7 - Showing creation of food_ratings_subset relation with two fields from the food_ratings relation and storing it back to HDFS in /user/hadoop/fr_subset directory

```
Input(s):
Successfully read 1000 records (17517 bytes) from: "/user/hadoop/foodratings8964
0.txt"

Output(s):
Successfully stored 1000 records (7013 bytes) in: "/user/hadoop/fr_subset"
```

Figure 8 - Showing confirmation of storing food_ratings_subset in /user/hadoop/fr_subset

```
grunt> food_ratings_subset_six_records = LIMIT food_ratings_subset 6;
grunt> DUMP food_ratings_subset_six_records;
```

Figure 9 - Commands used to write six records of the food_ratings_subset out to console

```
22/10/05 16:58:46 INFO util.MapRedUtil: Total input paths to process : 1
(Sam,14)
(Sam,41)
(Mel,28)
(Jill,13)
(Mel,14)
(Sam,40)
grunt>
```

Figure 10 - Showing the six records printed out to console

Commands Used:
food_ratings_subset = FOREACH food_ratings GENERATE name, f4;

STORE food_ratings_subset INTO '/user/hadoop/fr_subset'
>> USING PigStorage(',');

food_ratings_subset_six_records = LIMIT food_ratings_subset 6;

DUMP food_ratings_subset_six_records;

Exercise 3) 2 points

```
grunt> group_food_ratings = GROUP food_ratings ALL;
```

Figure 11 - Command used to group the food_ratings relation to be used in food_ratings_profile

```
grunt> food_ratings_profile = FOREACH group_food_ratings GENERATE MIN(food_ratin
gs.f2), MAX(food_ratings.f2), AVG(food_ratings.f2), MIN(food_ratings.f3), MAX(fo
od_ratings.f3), AVG(food_ratings.f3);
grunt> DUMP food_ratings_profile;
```

Figure 12 - Commands used to create food_ratings_profile relation holding min, max, and avg values for f2 and f3 attributes and also the dump command to output the records

```
22/10/05 17:25:32 INFO util.MapRedUtil: Total input paths to process : 1
(1,50,25.538,1,50,25.859)
```

Figure 13 - Showing the min, max, and avg values for f2 and f3 attributes output in console

Commands Used:
group_food_ratings = GROUP food_ratings ALL;

food_ratings_profile = FOREACH group_food_ratings GENERATE MIN(food_ratings.f2), MAX(food_ratings.f2), AVG(food_ratings.f2), MIN(food_ratings.f3), MAX(food_ratings.f3), AVG(food_ratings.f3);

DUMP food_ratings_profile;

Exercise 4) 2 points

```
grunt> food_ratings_filtered = FILTER food_ratings BY (f1<20) AND (f3>5);
grunt> six_food_ratings_filtered = LIMIT food_ratings_filtered 6;
grunt> DUMP six_food_ratings_filtered;
```

Figure 14 - Showing commands used to create a new filtered relation where f1 < 20 and f3 > 5 while limiting it to 6 records. The relation was then displayed on console using DUMP

```
22/10/05 19:07:43 INFO util.MapRedUtil: Total input paths to process : 1
(Mel,18,4,47,28,2)
(Mel,6,30,24,14,2)
(Sam,15,29,39,40,5)
(Joe,14,28,40,32,2)
(Joe,2,18,32,37,4)
(Mel,6,35,27,5,5)
```

Figure 15 - Showing the six records output of the relation created

Commands Used:
food_ratings_filtered = FILTER food_ratings BY (f1<20) AND (f3>5);

six_food_ratings_filtered = LIMIT food_ratings_filtered 6;

DUMP six_food_ratings_filtered;

Exercise 5) 2 points

```
grunt> food_ratings_2percent = SAMPLE food_ratings 0.02;
grunt> food_ratings_2percent_ten_records = LIMIT food_ratings_2percent 10;
grunt> DUMP food_ratings_2percent_ten_records;
```

Figure 16 - Showing commands used to create a new relation holding a random selection of 2% of the records in the initial relation while limiting the relation to 10 records and displaying it on console using DUMP

```
22/10/05 19:11:27 INFO util.MapRedUtil: Total input paths to process : 1
(Mel,33,3,42,35,2)
(Joe,14,31,41,37,3)
(Joy,37,46,7,16,3)
(Sam,49,42,22,39,1)
(Sam,36,34,21,49,5)
(Joe,43,21,43,49,3)
(Sam,15,17,47,2,3)
(Jill,41,35,49,39,1)
(Jill,34,8,3,41,3)
(Sam,24,38,13,33,1)
```

Figure 17 - Showing the ten records output of the relation created

Commands Used:
food _ratings_2percent = SAMPLE food_ratings 0.02;

food_ratings_2percent_ten_records = LIMIT food _ratings_2percent 10;

DUMP food_ratings_2percent_ten_records;

Exercise 6) 2 points

```
grunt> food_places = LOAD'/user/hadoop/foodplaces89640.txt'
>> USING PigStorage(',')
>> AS (placeid:int, placename:chararray);
22/10/05 19:16:14 INFO Configuration.deprecation: yarn.resourcemanager.system-me
trics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publishe
r.enabled
grunt> DESCRIBE food_places;
food_places: {placeid: int,placename: chararray}
```

Figure 18 - Showing Pig statement used to load the foodplaces file as a relation and running the 'DESCRIBE food_places;' command to show the summary of the created relation

```
grunt> food_ratings_w_place_names = JOIN food_ratings BY placeid,
>> food_places BY placeid;
grunt> food_ratings_w_place_names_six_records = LIMIT food_ratings_w_place_names
 6;
grunt> DUMP food_ratings_w_place_names_six_records;
```

Figure 19 - Showing code used to perform a join between the initial place_ratings relation and the food_places relation while limiting the created joined relation to 6 records and using the DUMP command to display it on the console

```
22/10/05 19:20:36 INFO util.MapRedUtil: Total input paths to process : 1
(Joe,31,20,24,3,1,1,China Bistro)
(Sam,4,27,45,30,1,1,China Bistro)
(Mel,3,48,49,9,1,1,China Bistro)
(Joy,24,41,20,16,1,1,China Bistro)
(Joe,26,31,44,1,1,1,China Bistro)
(Sam,37,29,7,7,1,1,China Bistro)
```

Figure 20 - Showing the six records output of the joined relation created

Commands Used:
food_places = LOAD'/user/hadoop/foodplaces89640.txt' USING PigStorage(',') AS (placeid:int, placename:chararray);

DESCRIBE food_places;

food_ratings_w_place_names = JOIN food_ratings BY placeid, food_places BY placeid;

food_ratings_w_place_names_six_records = LIMIT food_ratings_w_place_names 6;

DUMP food_ratings_w_place_names_six_records;

Exercise 7) 3 points

1. A - LIMIT
2. C - DISTINCT
3. B - (f1: STRING, f2: INT, f3: INT, f4: INT)
4. B - relB = FOREACH relA GENERATE $0, f3;
5. B - data flow
6. A - relB = FILTER relA by $0 < 20