BigData Technologies
Assignment-5

EMR Connection:

```
(base) sailavanyanarthu@Sailavanyas-MacBook-Air downloads % scp -i keyass5.pem TestDataGen.class hadoop@ec2-3-145-161-206.us-east-2.compute.amazonaws.com:/home/hadoop
TestDataGen.class                                                                                          100% 2189    61.0KB/s   00:00

(base) sailavanyanarthu@Sailavanyas-MacBook-Air downloads % scp -i keyass5.pem pigdemo.zip hadoop@ec2-3-145-161-206.us-east-2.compute.amazonaws.com:/home/hadoop
pigdemo.zip                                            100%  268KB   1.2MB/s   00:00
(base) sailavanyanarthu@Sailavanyas-MacBook-Air downloads %
```

```
(base) sailavanyanarthu@Sailavanyas-MacBook-Air ~ % cd downloads
(base) sailavanyanarthu@Sailavanyas-MacBook-Air downloads % chmod 400 keyass5.pem
(base) sailavanyanarthu@Sailavanyas-MacBook-Air downloads % ssh -i keyass5.pem hadoop@ec2-3-145-161-206.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-145-161-206.us-east-2.compute.amazonaws.com (3.145.161.206)' can't be established.
ED25519 key fingerprint is SHA256:I2sJYf3AstMsFsR0DuHk2rFvdMvmCRMwGhA6fPu8wso.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-145-161-206.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Tue Oct 17 03:30:12 2023

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR
```

Exercise 1

```
[hadoop@ip-172-31-14-128 ~]$ java TestDataGen
Error: Could not find or load main class TestDataGen
[hadoop@ip-172-31-14-128 ~]$ java TestDataGen
Error: Could not find or load main class TestDataGen
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$ java TestDataGen
Magic Number = 43721
[hadoop@ip-172-31-14-128 ~]$ ls
foodplaces43721.txt  foodratings43721.txt  TestDataGen.class
```

Magic Number : 43721

hadoop fs -put /home/hadoop/foodratings43721.txt /user/hadoop/
hadoop fs -put /home/hadoop/foodplaces43721.txt /user/hadoop/
pig

```
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$ hadoop fs -put /home/hadoop/foodratings43721.txt /user/hadoop/
[hadoop@ip-172-31-14-128 ~]$ hadoop fs -put /home/hadoop/foodplaces43721.txt /user/hadoop/
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$
[hadoop@ip-172-31-14-128 ~]$ pig
```

food_ratings = LOAD '/user/hadoop/foodratings43721.txt' USING PigStorage(',') AS (name:chararray,f1:int,f2:int,f3:int,f4:int,placeid:int);

describe food_ratings;

```
grunt> food_ratings = LOAD'foodratings43721.txt' USING PigStorage(',') AS (name:chararray,f1:int,f2:int,f3:int,f4:int,placeid:int);
2023-10-17 04:41:35,978 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt>
grunt> DESCRIBE food_ratings;
food_ratings: {name: chararray,f1: int,f2: int,f3: int,f4: int,placeid: int}
grunt>
```

Exercise 2

food_ratings_subset = FOREACH food_ratings GENERATE name, f4;
STORE food_ratings_subset INTO '/user/hadoop/fr-subset';

```
grunt>
grunt> food_ratings_subset = FOREACH food_ratings GENERATE name, f4;
grunt> STORE food_ratings_subset INTO '/user/hadoop/fr-subset';
```

food_ratings_subset_lim = LIMIT food_ratings_subset 6;
Dump food_ratings_subset_lim;

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


382611 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
2023-10-17 04:46:21,137 INFO util.MapRedUtil: Total input paths to process : 1
(Mel,26)
(Joy,20)
(Sam,28)
(Jill,25)
(Joy,2)
(Jill,33)
grunt>
```

Exercise3

food_ratings_group = GROUP food_ratings ALL;

food_ratings_profile = FOREACH (GROUP food_ratings ALL) 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);

```
grunt>
grunt> food_ratings_group = GROUP food_ratings ALL;
grunt> food_ratings_profile = FOREACH food_ratings_group 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);
grunt> DUMP food_ratings_profile;
```

DUMP food_ratings_profile;

```
Input(s):
Successfully read 1000 records (17437 bytes) from: "hdfs://ip-172-31-14-128.us-east-2.compute.internal:8020/user/hadoop/foodratings43721.txt"

Output(s):
Successfully stored 1 records (28 bytes) in: "hdfs://ip-172-31-14-128.us-east-2.compute.internal:8020/tmp/temp2138943321/tmp-1175449410"

2023-10-17 04:53:03,892 INFO input.FileInputFormat: Total input files to process : 1
785366 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
2023-10-17 04:53:03,892 INFO util.MapRedUtil: Total input paths to process : 1
(1,50,25.091,1,50,25.818)
grunt>
```

Exercise4

food_ratings_filtered = FILTER food_ratings BY (f1<20) AND (f3<5);
food_ratings_filtered_result = LIMIT food_ratings_filtered 6;
DUMP food_ratings_filtered_result;

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

2023-10-17 04:56:15,544 INFO input.FileInputFormat: Total input files to process : 1
977018 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
2023-10-17 04:56:15,544 INFO util.MapRedUtil: Total input paths to process : 1
(Joy,30,36,23,20,1)
(Sam,39,35,24,28,4)
(Jill,48,43,28,25,3)
(Joy,37,44,21,2,5)
(Jill,31,26,31,33,4)
(Sam,49,29,23,29,1)
grunt>
```

Exercise5

food_ratings_2percent = SAMPLE food_ratings 0.02;
food_ratings_2percent_result = LIMIT food_ratings_2percent 10;
DUMP food_ratings_2percent_lim;

```
grunt>
grunt> food_ratings_2percent = SAMPLE food_ratings 0.02;
grunt> food_ratings_2percent_lim = LIMIT food_ratings_2percent 10;
grunt> DUMP food_ratings_2percent_lim;
```

```
2023-10-17 05:00:24,710 INFO input.FileInputFormat: Total input files to process : 1
1226184 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
2023-10-17 05:00:24,710 INFO util.MapRedUtil: Total input paths to process : 1
(Sam,7,28,49,34,4)
(Jill,7,30,24,43,4)
(Joy,20,23,8,13,1)
(Joe,28,18,49,16,1)
(Mel,25,46,12,50,3)
(Joy,12,22,50,33,5)
(Mel,30,37,22,24,4)
(Sam,18,35,47,41,4)
(Jill,10,25,7,23,4)
(Sam,40,36,7,38,1)
grunt>
```

Exercise6

food_places= LOAD 'foodplaces43721.txt' USING PigStorage(',') AS (placeid:int, placename:chararray);

```
grunt> food_places = LOAD 'foodplaces43721.txt' USING PigStorage(',') AS (placeid:int,placename:chararray);
2023-10-17 05:02:26,778 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt> {placeid: int,placename: chararray};
```

food_ratings_w_place_names= JOIN food_ratings BY placeid,food_places BY placeid;
food_ratings_w_place_names_lim = LIMIT food_ratings_w_place_names 6;
DUMP food_ratings_w_place_names_lim;

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

```
2023-10-17 05:10:33,804 INFO input.FileInputFormat: Total input files to process : 1
1835278 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
2023-10-17 05:10:33,804 INFO util.MapRedUtil: Total input paths to process : 1
(Joy,19,27,30,48,1,1,China Bistro)
(Sam,45,49,37,6,1,1,China Bistro)
(Sam,39,6,12,20,1,1,China Bistro)
(Sam,1,17,26,27,1,1,China Bistro)
(Sam,24,29,46,26,1,1,China Bistro)
(Sam,20,3,40,13,1,1,China Bistro)
```

Exercise7

I. Which keyword is used to select a certain number of rows from a relation when forming a new relation? **Answer: A**
Choices: A. LIMIT B. DISTINCT C. UNIQUE D. SAMPLE

II. Which keyword returns only unique rows for a relation when forming a new relation?
Choices: **Answer: C**
A. SAMPLE B. FILTER C. DISTINCT D. SPLIT

III. Assume you have an HDFS file with a large number of records similar to the examples below • Mel, 1, 2, 3 • Jill, 3, 4, 5 Which of the following would NOT be a correct pig schema for such a file?
Choices: **Answer: C**

A. (f1: CHARARRY, f2: INT, f3: INT, f4: INT) B. (f1: STRING, f2: INT, f3: INT, f4: INT) C. (f1, f2, f3, f4) D. (f1: BYTEARRAY, f2: INT, f3: BYTEARRAY, f4: INT)

IV. Which one of the following statements would create a relation (relB) with two columns from a relation (relA) with 4 columns? Assume the pig schema for relA is as follows: (f1: INT, f2, f3, f4: FLOAT) **Answer: B**
Choices:
A. relB = GROUP relA GENERATE f1, f3; B. relB = FOREACH relA GENERATE $0, f3; C. relB = FOREACH relA GENERATE f1, f5; D. relB = FOREACH relA SELECT f1, f3;

V. Pig Latin is a **dataflow** language. Select the best choice to fill in the blank.
Answer: B
Choices: A. functional B. data flow C. procedural D. declarative

VI. Given a relation (relA) with 4 columns and pig schema as follows: (f1: INT, f2, f3, f4: FLOAT) which one statement will create a relation (relB) having records all of whose first field is less than 20
Answer: A
Choices: A. relB = FILTER relA by $0 < 20 B. relB = GROUP relA by f1 < 20 C. relB = FILTER relA by $1 < 20 D. relB = FOREACH relA GENERATE f1 < 20

Submitted by:
Sailavanya Narthu
A20516764