

### Exercise 1)

Create new versions of the foodratings and foodplaces files by using TestDataGen (as described in assignment #4) and copy them to HDFS (say into /user/hadoop).

Write and execute a sequence of pig latin statements that loads the foodratings file as a relation. Call the relation 'food\_ratings'. The load command should associate a schema with this relation where the first attribute is referred to as 'name' and is of type chararray, the next attributes are referred to as 'f1' through 'f4' and are of type int, and the last field is referred to as 'placeid' and is also of type int.

Execute the describe command on this relation. Provide the magic number, the load command you wrote and the output of the describe command as the result of this exercise.

Answer-

The magic number is: 24283

```
[hadoop@ip-172-31-10-47 ~]$ java TestDataGen
Magic Number = 24283
```

The load command is as follows:

- **food\_ratings = LOAD '/user/hadoop/foodratings24283.txt' USING PigStorage(',') AS (name: chararray, f1:int, f2:int, f3:int, f4:int, placeid:int );**

The output of the DESCRIBE command:

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

### Exercise 2)

Now create another relation with two fields of the initial (food\_ratings) relation: 'name' and 'f4'. Call this relation 'food\_ratings\_subset'. Store this last relation, food\_ratings\_subset, back to HDFS (perhaps as the file /user/hadoop/fr\_subset) Also write 6 records of this relation out to the console. Submit the pig latin statements you used and the six records printed out to the console as the result of this exercise.

Answer-

The pig latin statements used are:

- **food\_ratings\_subset = FOREACH food\_ratings GENERATE name, f4;**
- **STORE food\_ratings\_subset INTO '/user/hadoop/fr\_subset' USING PigStorage('|');**
- **fr\_subset = LIMIT food\_ratings\_subset 6;**
- **DUMP fr\_subset;**

Output:

```
(ae1,12)
(oe,15)
(oe,7)
(oe1,7)
(1111,3)
(oe,28)
prints
```

### Exercise 3)

Now create another relation using the initial (food\_ratings) relation. Call this relation 'food\_ratings\_profile'. The new relation should only have one record. This record should hold the minimum, maximum and average values for the attributes 'f2' and 'f3'. (So this one record will have 6 fields). Write the record of this relation out to the console. Submit the pig latin statements you used and the record printed out to the console as the result of this exercise.

Answer-

The pig latin statements used are:

- **food\_ratings\_group = GROUP food\_ratings ALL;**
- **food\_ratings\_profile = FOREACH food\_ratings\_group GENERATE MIN(food\_ratings.f2) as f2\_min, MAX(food\_ratings.f2) as f2\_max, AVG(food\_ratings.f2) as f2\_avg, MIN(food\_ratings.f3) as f3\_min, MAX(food\_ratings.f3) as f3\_max, AVG(food\_ratings.f3) as f3\_avg;**
- **DUMP food\_ratings\_profile;**

Output:

```
Input(s):
Successfully read 1000 records (17511 bytes) from: "/user/hadoop/foodratings24283.txt"
Output(s):
Successfully stored 1 records (28 bytes) in: "hdfs://ip-172-31-10-47.ec2.internal:8020/tmp/temp1073244826/tmp595332222"
2023-10-19 17:29:18,014 INFO input.FileInputFormat: Total input files to process : 1
652358 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2023-10-19 17:29:18,014 INFO util.MapRedUtil: Total input paths to process : 1
(1,50,25,45,1,50,25,727)
grunt>
```

### Exercise 4)

Now create yet another relation from the initial (food\_ratings) relation. This new relation should only include tuples (records) where f1 < 20 and f3 > 5. Call this relation 'food\_ratings\_filtered'. Write 6 records of this relation out to the console. Submit the pig latin statements you used and the six records printed out to the console as the result of this exercise.

Answer-

The pig latin statements used are:

- **food\_ratings\_filtered = FILTER food\_ratings BY (f1 < 20) AND (f3 > 5);**
- **food\_ratings\_filtered\_6= LIMIT food\_ratings\_filtered 6;**
- **DUMP food\_ratings\_filtered\_6;**

Output:

```
Input(s):
Successfully read 12 records (17511 bytes) from: "/user/hadoop/foodratings24283.txt"
Output(s):
Successfully stored 6 records (120 bytes) in: "hdfs://ip-172-31-10-47.ec2.internal:8020/tmp/temp1073244826/tmp-100721373"
2023-10-19 17:32:34,649 INFO input.FileInputFormat: Total input files to process : 1
648993 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2023-10-19 17:32:34,649 INFO util.MapRedUtil: Total input paths to process : 1
(Joe,16,20,29,7,4)
(Jill,12,20,20,5,1)
(Jill,18,9,30,31,3)
(Joe,16,47,46,26,1)
(Mel,8,31,8,39,1)
(Jill,18,6,47,49,2)
grunt>
```

### Exercise 5)

Using the initial (food\_ratings) relation, write and execute a sequence of pig latin statements that creates another relation, call it 'food\_ratings\_2percent', holding a random selection of 2% of the records in the initial relation. Write 10 of the records out to the console. Submit the pig latin statements and the records printed out to the console.

Answer-

The pig latin statements used are:

- **food\_ratings\_2percent = SAMPLE food\_ratings 0.02;**
- **DESCRIBE food\_ratings\_2percent;**
- **food\_ratings\_2percent\_10= LIMIT food\_ratings\_2percent 10;**
- **DESCRIBE food\_ratings\_2percent\_10;**
- **DUMP food\_ratings\_2percent\_10;**

Output:

```
Input(s):
Successfully read 518 records (17511 bytes) from: "/user/hadoop/foodratings24283.txt"

Output(s):
Successfully stored 10 records (198 bytes) in: "hdfs://ip-172-31-10-47.ec2.internal:8020/tmp/temp1073244826/tmp-1662919934"

2023-10-19 17:35:49,307 INFO input.FileInputFormat: Total input files to process : 1
1043652 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2023-10-19 17:35:49,308 INFO util.MapRedUtil: Total input paths to process : 1
(Mel,8,31,8,39,1)
(Joy,10,15,23,29,5)
(Joy,1,50,40,38,4)
(Joy,42,4,13,2,3)
(Joe,18,20,48,5,3)
(Mel,45,42,44,10,5)
(Sam,12,12,16,42,4)
(Joy,22,46,48,19,4)
(Sam,34,22,29,31,3)
(Joe,13,42,43,47,2)
grunt>
```

### Exercise 6)

Write and execute a sequence of pig latin statements that loads the foodplaces file as a relation. Call the relation 'food\_places'. The load command should associate a schema with this relation where the first attribute is referred to as 'placeid' and is of type int and the second attribute is referred to as 'placename' and is of type chararray. Execute the describe command on this relation. Now perform a join between the initial place\_ratings relation and the food\_places relation on the placeid attributes to create a new relation called 'food\_ratings\_w\_place\_names'. This new relation should have all the attributes (columns) of both relations. The new relation will allow us to work with place ratings and place names together. Write 6 records of this relation out to the console. Submit the pig latin statements you used and the six records printed out to the console as the result of this exercise.

Answer-

The pig latin statements that loads the foodplaces file as a relation:

- **food\_places = LOAD '/user/hadoop/foodplaces24283.txt' USING PigStorage(',') AS (placeid:int, placename:chararray);**

Output of the DESCRIBE command :

```
grunt> food_places = LOAD '/user/hadoop/foodplaces24283.txt' USING PigStorage(',') AS (placeid:int, placename:chararray);
2023-10-19 17:42:32,39 [INFO] Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
grunt> DESCRIBE food_places;
food_places: (placeid: int,placename: chararray)
grunt>
```

The pig latin statements used are:

- **food\_ratings\_w\_place\_names = JOIN food\_ratings BY placeid, food\_places BY placeid;**
- **DESCRIBE food\_ratings\_w\_place\_names;**
- **food\_ratings\_w\_place\_names\_6 = LIMIT food\_ratings\_w\_place\_names 6;**
- **DESCRIBE food\_ratings\_w\_place\_names\_6;**
- **DUMP food\_ratings\_w\_place\_names\_6;**

Output:

```
Input(s):
Successfully read 5 records (59 bytes) from: "/user/hadoop/foodplaces24283.txt"
Successfully read 1000 records (17511 bytes) from: "/user/hadoop/foodratings24283.txt"

Output(s):
Successfully stored 6 records (211 bytes) in: "hdfs://ip-172-31-10-47.ec2.internal:8020/tmp/temp1073244826/tmp-1310236312"

2023-10-19 17:45:40,401 INFO input.FileInputFormat: Total input files to process : 1
1634745 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2023-10-19 17:45:40,401 INFO util.MapRedUtil: Total input paths to process : 1
(Joe,22,22,26,43,1,1,China Bistro)
(Joe,43,10,22,15,1,1,China Bistro)
(Jill,13,20,20,5,1,1,China Bistro)
(Joy,34,48,39,38,1,1,China Bistro)
(Joe,16,47,46,26,1,1,China Bistro)
(Mel,8,31,8,39,1,1,China Bistro)
grunt>
```

Exercise 7)

Identify the one correct answer for each the following questions.

I. Which keyword is used to select a certain number of rows from a relation when forming a new relation?

Answer: \_\_\_\_\_

Choices:

- A. LIMIT
- B. DISTINCT
- C. UNIQUE
- D. SAMPLE

II. Which keyword returns only unique rows for a relation when forming a new relation?

Answer: \_\_\_\_\_

Choices:

- 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?

Answer: \_\_\_\_\_

Choices:

- A. (f1: CHARARRAY, 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: \_\_\_\_\_

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 \_\_\_\_\_ language. Select the best choice to fill in the blank.

Answer: \_\_\_\_\_

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: \_\_\_\_\_

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