**CSP 554 – Assignment #5**

Name: Adarsh Mathad Vijayakumar

CWID: A20424847

Email ID: avijayakumar@hawk.iit.edu

Exercise 1)

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

Command: java TestDataGen

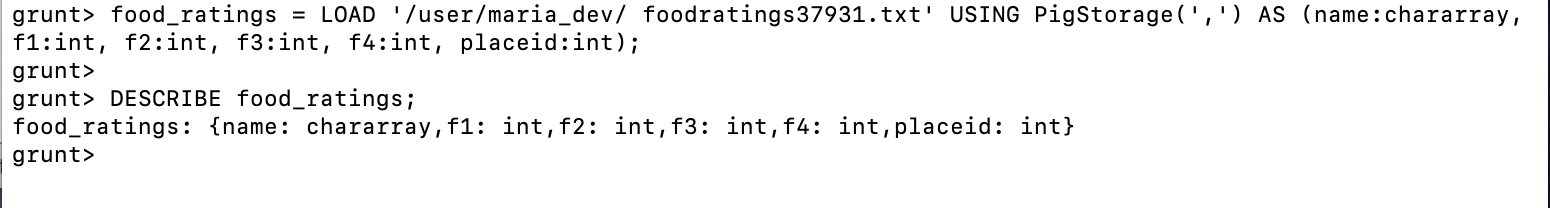
Output: Magic Number = 37931



Write and execute a sequence of pig latin statements that loads the foodratings file as a relation. Call the relation ‘food\_ratings’.

Commands:

1. food\_ratings = LOAD '/user/maria\_dev/foodratings37931.txt' USING PigStorage(',') AS (name:chararray, f1:int, f2:int, f3:int, f4:int, placeid:int);
2. DESCRIBE food\_ratings;



Exercise 2)

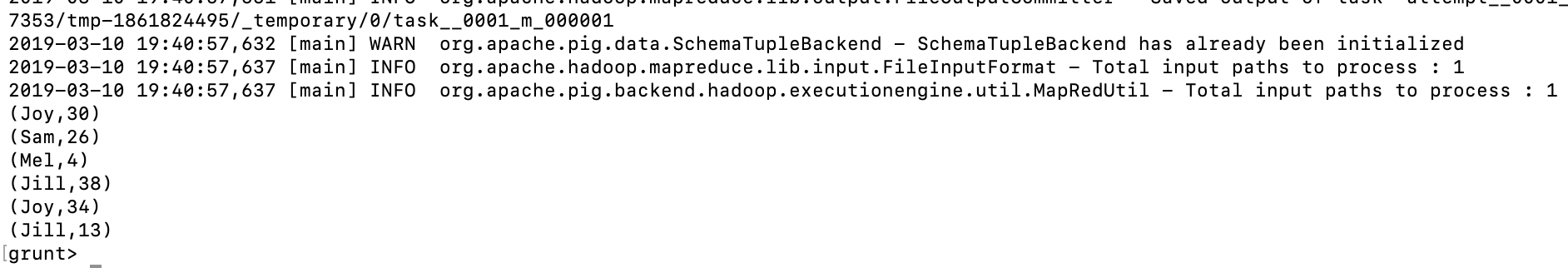
Now create another relation with two fields of the initial (food\_ratings) relation: ‘name’ and ‘f4’. Call this relation ‘food\_ratings\_subset’.

Store this last relation back to HDFS.

Also write 6 records of this relation out to the console.

Commands:

1. food\_ratings\_subset = FOREACH food\_ratings GENERATE name, f4;
2. STORE food\_ratings\_subset INTO '/user/maria\_dev/food\_ratings\_subset' USING PigStorage(',');
3. Top6\_food\_ratings\_subset = LIMIT food\_ratings\_subset 6;
4. DUMP Top6\_food\_ratings\_subset;

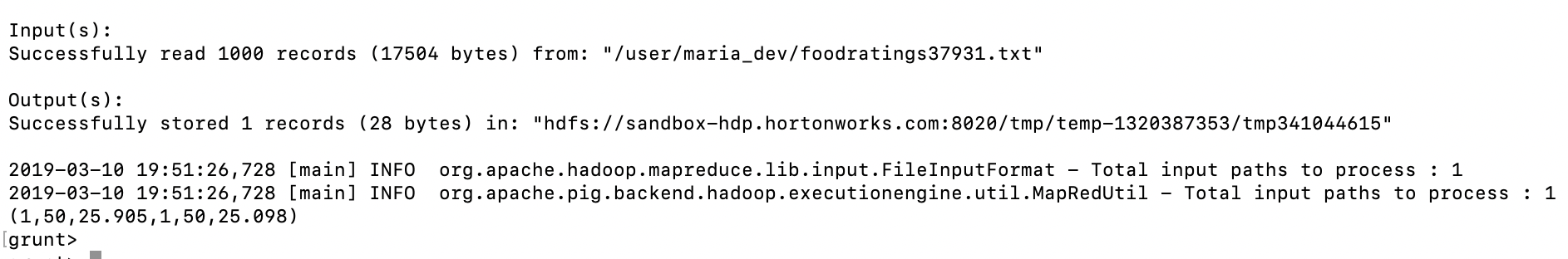


Exercise 3)

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 fileds).

Commands:

1. 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);
2. DUMP food\_ratings\_profile;



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.

Commands:

1. food\_ratings\_filtered = FILTER food\_ratings BY (f1 < 20) AND (f3 > 5);
2. Top6\_food\_ratings\_filtered = LIMIT food\_ratings\_filtered 6;
3. DUMP Top6\_food\_ratings\_filtered;



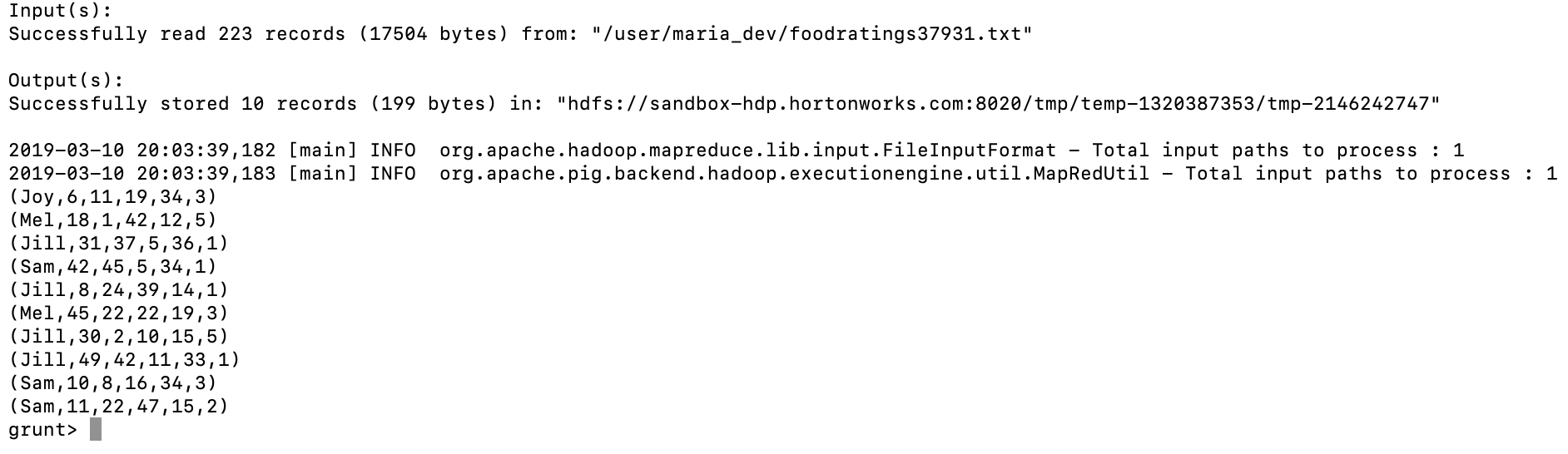
Exercise 5)

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.

Commands:

1. food\_ratings\_2percent = SAMPLE food\_ratings 0.02;
2. Top10\_food\_ratings\_2percent = LIMIT food\_ratings\_2percent 10;
3. DUMP Top10\_food\_ratings\_2percent;



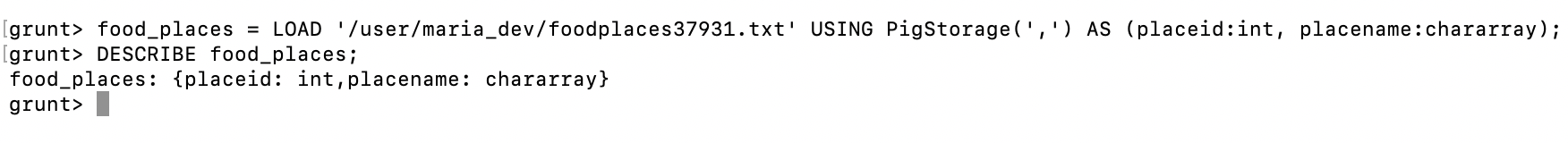
Exercise 6)

Write and execute a sequence of pig latin statements that loads the foodplaces file as a relation. Call the relation ‘food\_places’.

Execute the describe command on this relation.

Commands:

1. food\_places = LOAD '/user/maria\_dev/foodplaces37931.txt' USING PigStorage(',') AS (placeid:int, placename:chararray);
2. DESCRIBE food\_places;



Now perform a join between the initial food\_ratings relation and the food\_places relation on the placeid attributes to create a new relation called ‘food\_ratings\_w\_place\_names’.

Write 6 records of this relation out to the console.

Commands:

1. food\_ratings\_w\_place\_names = JOIN food\_ratings BY placeid, food\_places BY placeid;
2. Top6\_food\_ratings\_w\_place\_names = LIMIT food\_ratings\_w\_place\_names 6;
3. DUMP Top6\_food\_ratings\_w\_place\_names;



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

Answer: **Option A. LIMIT**

Choices:

1. LIMIT
2. DISTINCT
3. UNIQUE
4. SAMPLE
5. Which keyword returns only unique rows for a relation when forming a new relation?

**Answer: C. Distinct**

1. SAMPLE
2. FILTER
3. DISTINCT
4. SPLIT
5. 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: B. (f1: STRING, f2: INT, f3: INT, f4: INT)**

1. (f1: CHARARRY, f2: INT, f3: INT, f4: INT)
2. (f1: STRING, f2: INT, f3: INT, f4: INT)
3. (f1, f2, f3, f4)
4. (f1: BYTEARRAY, f2: INT, f3: BYTEARRAY, f4: INT)
5. 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: option B**

Choices:

1. relB = GROUP relA GENERATE f1, f3;
2. relB = FOREACH relA GENERATE $0, f3;
3. relB = FOREACH relA GENERATE f1, f5;
4. relB = FOREACH relA SELECT f1, f3;
5. Pig Latin is a \_\_\_\_\_\_\_ language. Select the best choice to fill in the blank.

**Answer:** **Option B – Data Flow**

Choices:

1. functional
2. data flow
3. procedural
4. declarative
5. 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:** **Option A**

Choices:

1. relB = FILTER relA by $0 < 20
2. relB = GROUP relA by f1 < 20
3. relB = FILTER relA by $1 < 20
4. relB = FOREACH relA GENERATE f1 < 20