INF 551 – Fall 2019

Homework #5

Due: December 6, Friday
100 points

1. [60 points] Write a Hadoop MapReduce program, count.java, that takes the Sells table (stored as a comma-separated-value or CSV file) in the beers database, and computes the same results as the following SQL query. You can assume that there are no NULL prices in the table.

```
Select bar, count(*)
From Sells
Where beer like 'bud%'
Group by bar
Having max(price) <= 5
```

Example input file:

```
Joe's bar,bud,3
Mary's bar,bud light,4
…
```

Sample Output format: (Please use "\t" as delimiter between bar name and number)

```
Bob's bar	3
Joe's bar	3
…
```

Execution format:
hadoop jar count.jar count input/sells.csv output
(where sells.csv is the file storing the content of the sells table)
(please set the class name as **count**)

2. [40 points] For each of the following queries, write a Spark program in Python to implement the query. Assume that all tables in the beers database are stored in the CSV files.
   a. **Implement the same query as Question 1.**

      Execution format: spark-submit Qa.py input/sells.csv output.txt
      output.txt format: same as question 1.

   b. **For each bar, compute the average price of beers sold at the bar.**

      Execution format: spark-submit Qb.py input/sells.csv output.txt
      output.txt format: (Please use "\t" as delimiter between bar name and number)

```
Bar	Average_price
Bob's bar	3
```

c. Find all drinkers that frequent some bars but do not like any beers.

Execution format: spark-submit Qc.py input/frequents.csv input/likes.csv output.txt output.txt format:

Drinker
Steve

d. Find all drinker-beer pairs such that the drinker likes the beer and frequents a bar that sells the beer.

Execution format:
spark-submit Qd.py input/likes.csv input/frequents.csv input/sells.csv output.txt output.txt format: (Please use "\t" as delimiter between drinker name and beer name)

Drinker Beer
Steve Bud

Submissions

For q1: <firstname>_<lastname>_count.jar, <firstname>_<lastname>_count.java

For q2: <firstname>_<lastname>_Qa.py, <firstname>_<lastname>_Qb.py …and so on.

Then submit all files in a zip file named as <firstname>_<lastname>_hw5.zip

Grading Criteria

1. Please submit all the files in 1 zip file.
2. For q1, please do not use any library other than org.apache.hadoop.*, java.*.
3. You should implement Hadoop MapReduce for the task. The Hadoop version will be Hadoop-3.1.2.
4. For q2, please use python 3 and only libraries in Python Standard Library and pyspark are allowed.
5. For q2, you should implement the query in spark operations (RDD), no for loops and pyspark.sql allowed. The spark version will be spark-2.4.4-bin-hadoop2.7