CSE 482: Big Data Analysis (Spring 2020) Homework 6

Due date: April 30, 2020 (before midnight)

1. Consider the following two data files (student.csv and transcript.csv). The content of these files are as follows:

```
grunt> cat student.csv
john,cs,senior
mary,cs,junior
lee,ece,junior
bob,ece,junior

grunt> cat transcript.csv
cse101,john,3.0
cse101,mary,3.5
cse231,john,3.5
cse231,mary,3.5
cse231,bob,2.0
cse480,john,3.0
ece200,bob,3.0
ece200,lee,4.0
ece335,lee,3.5
```

   State each of the following Pig Latin queries below in plain English and show the result obtained after processing the query.

   (a)
```
data = load 'student.csv' using PigStorage(',');
grp = group data by $2;
result = foreach grp generate group, COUNT(data);
dump result;
```

   (b)
```
data = load 'transcript.csv' using PigStorage(',');
grp = group data by $1;
tmp = foreach grp generate group, AVG(data.$2);
result = FILTER tmp by $1 >= 3.5;
dump result;
```

   (c)
```
s = load 'student.csv' using PigStorage(',');
t = load 'transcript.csv' using PigStorage(',');
data = join s by $0, t by $1;
tmp = foreach data generate $2,$3,$5;
tmp2 = filter tmp by $1 matches 'cse231';
grp = group tmp2 by $0;
result = foreach grp generate group, AVG(tmp2.$2);
dump result;
```

   (d)
```
s = load 'student.csv' using PigStorage(',');
n = filter s by not($1 matches 'cs');
t = load 'transcript.csv' using PigStorage(',');
```

```
t2 = filter t by $0 matches 'cs.*';
tmp = join t2 by $1, n by $0;
result = foreach tmp generate $1;
dump result;
```

(e)
```
t = load 'transcript.csv' using PigStorage(',');
c = forech t generate $0;
c2 = distinct c;
c3 = filter c2 by $0 matches 'cse.*';
gc = group c3 all;
maxc = foreach gc generate COUNT(c3);
tmp = join t by $0, c3 by $0;
grp = group tmp by $1;
tmp2 = foreach grp generate group, COUNT(tmp);
tmp3 = join tmp2 by $1, maxc by $0;
result = foreach tmp3 generate $0;
dump result;
```

2. For this question, you will be using the dataset from Exercise 13. First, download the data from `http://www.cse.msu.edu/~cse482/exercise13.tar`. After extracting the archived file, you will find 2 data files: patient.csv and visit.csv. The patient.csv file contains the following 4 comma-separated values: patient ID, name, gender, and age, while the visit.csv file contains the following 4 comma-separated values: visit ID, visitDate, patientID, and diagnosis. Write the Pig Latin scripts to process each query below. For each question below, you need to submit the corresponding script file as well as the query result. The source code should be written in a script file named q2*.pig. For example, the script for the first question is q2a.pig, the second question is q2b.pig, and so on. **The query results must also be saved in their corresponding directories named q2***. Create a zip or tar file to compress/archive all the script and result files into a single file named `question2.tar` or `question2.zip` and submit it to D2L.

   (a) Write a Pig Latin script that returns all the patients diagnosed with hypertension. The query result should contain only 2 columns (patient ID and patient name). Save the output into a directory named `q2a`.

   (b) Write a Pig Latin script that counts the number of visits to the healthcare provider by each patient. The query result must have only 3 columns: patient ID, patient name and number of visits. Save the output into a directory named `q2b`.

   (c) Write a Pig Latin script that returns the ID and names of patients who were diagnosed with both diabetes and hypertension. Save the output into a directory named `q2c`.

   (d) Write a Pig Latin script that returns the most frequent diagnosis for patients who are at least 40 years old. The query result should

contain one row and 1 column. Save the result into a directory named `q2d`.

3. For this question, you will use the same dataset as question 2. You should save the source code into a script file named q3*.sql (e.g., q3a.sql, q3b.sql, etc) and submit a compressed/archived version of the files (question3.zip or question3.tar). First, you need to upload the data files `patient.csv` and `visit.csv` to HDFS in the directories named `patient` and `visit`, respectively.

   (a) Write the corresponding HiveQL queries for creating the following two external tables: *Patient* and *Visit*. The schema for the tables are as follows:

   ```
   Patient(ID: int, Name: string, Gender: string, Age: int)
   Visit(VisitID: int, VDate: string, PatientID: int, Diagnosis: string)
   ```

   Store the HiveQL query in a script file named `q3a.sql`. Note that you can execute the script file in beeline by typing `source q3a.sql`.

   (b) Write the corresponding HiveQL query to find the ID and names of all patients diagnosed with hypertension. Store the query in a script file named `q3b.sql`.

   (c) Write the corresponding HiveQL query to count the numebr of visits to the provider by each patient. The query result must return only 3 columns: patient ID, name, and number of visits. Store the query in a script file named `q3c.sql`.

   (d) Write the HiveQL query to find the ID and names of patients who were diagnosed to have both diabetes and hypertension. The query result must return only 2 columns: patient ID and patient name. Store the query in a script file named `q3d.sql`.

   (e) Write the HiveQL query to find the most frequent diagnosis for patients who are at least 40 years old. The query result must return only 1 row and 2 columns: diagnosis and number of cases. Store the query in a script file named `q3e.sql`.