STSCI 5065 Final Project

(Assigned on: 5/2/2019; Due on: 5/13/2019 at 11:59 PM)

General instructions:

- **Important: Read and follow this whole document carefully!**
- **Do your own work. Any cheating behavior (for example, submitting code similar to that of other student(s), copying code from an internet source, etc.) will result in a serious consequence (e.g., getting zero points, failing the class, …). If you have a question, you should directly ask your instructor.**
- Your project report is a single PDF file, named **STSCI5065-Final-Project-LastName-FirstName.PDF**, including the questions themselves, code or commands used, and your answers to the questions. List the items in the order of the questions.
- All your code and commands should be highlighted with **blue** fonts, and they must be in the **text format** not a screenshot so that your grader can copy your code or command(s) and run it in his/her computing environment**.**
- You are asked to submit a screenshot of your result of each question; your screenshots in Ambari must contain the **job IDs** and f**ile names** if applicable.

What and how to turn in: Submit your STSCI5065-Final-Project-LastName-FirstName.PDF file electronically to the course website. Please note that **an overdue submission will not be graded.**

In this project, you will work on flight delay datasets. The data files are compressed into a file called **flightDelays.tar.gz**, which contains 12 separate files and can be downloaded from the course website under Final Project. You will load the data files into HDFS with different methods, manipulate and analyze the data based on the specifications in the following questions.

1. In the command line interface (CLI), create an HDFS folder, **/final_project/data**. Download **flightDelays.tar.gz** with your web browser and then load the file into your HDFS folder, /final_project/data, using the Files View in Ambari. Attach a screenshot of your Files View showing this file.

2. Create a directory **FP** (a non-HDFS directory) at the CentOS root. Use a command to copy flightDelays.tar.gz to /FP and use another command to decompress and untar flightDelays.tar.gz. Make sure that your /FP directory only contains the 12 newly created files all the time; delete any other files if any. Display the contents of the /FP directory (using the long listing and human readable format) and attach a screenshot.

3. Load the 12 files from the FP directory to the HDFS directory, /final_project/data. Use three different approaches to load these files. Describe each of your approach and attach a screenshot of the result of each method. Note that at the beginning of a new method, you need to delete the files loaded previously and get ready for the next method to load the same files.

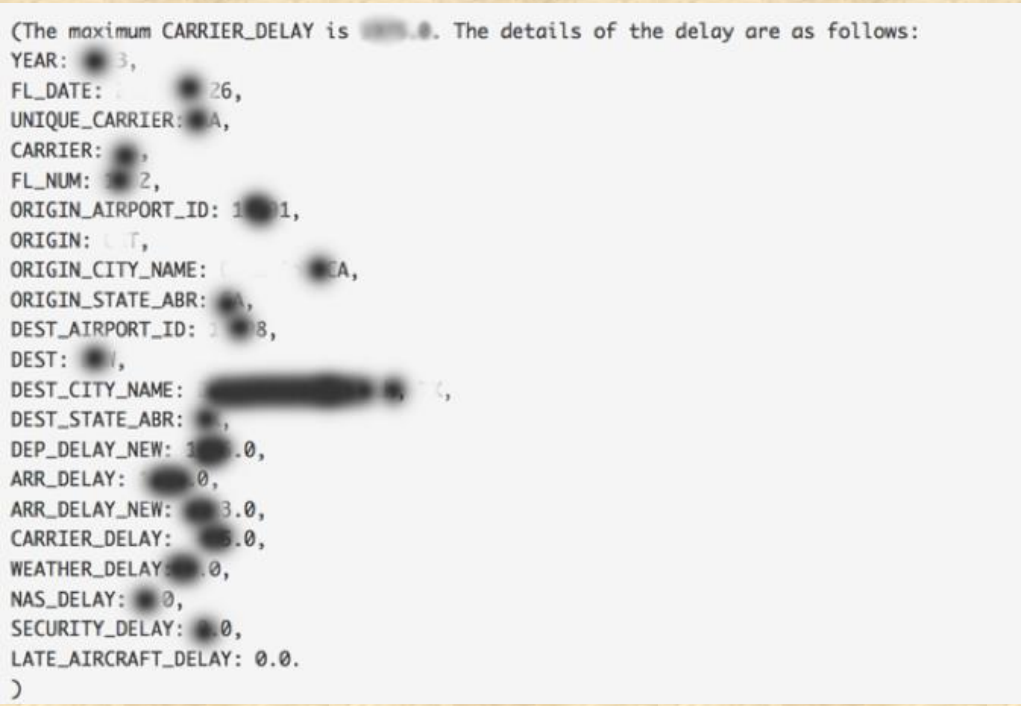
4. Use Hive CLI to create a database called **FPdb**, located in /final_project. Describe the database after its creation. Attach a screenshot.

5. Use a Pig script in Ambari to create a relation called **flightDelays** to load all the 12 files with a single LOAD statement. You use the following column names: YEAR, FL_DATE, UNIQUE_CARRIER, CARRIER, FL_NUM, ORIGIN_AIRPORT_ID, ORIGIN, ORIGIN_CITY_NAME, ORIGIN_STATE_ABR, DEST_AIRPORT_ID, DEST, DEST_CITY_NAME, DEST_STATE_ABR, DEP_DELAY_NEW, ARR_DELAY, ARR_DELAY_NEW, CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, and LATE_AIRCRAFT_DELAY. If a column name contains the word "DELAY," assign the **float** data type to it. Leave all other columns the **chararray** data type, except the YEAR column, which has an **int** data type. When you list the column names, please list one column name per line. Note that the data files contain a special data format: some values are enclosed in quotation marks but there is a comma within that value. Name your pig script flightDelays. Attach a screenshot showing that you successfully ran your pig script, including the logs.

6. Based on the flightDelays relation, calculate the **average delays** (rounded to two decimal points) of the following delay categories: CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, and LATE_AIRCRAFT_DELAY. Name your pig script averageDelays. Attach a screenshot showing that you successfully ran your pig script, including the logs.

7. Based on the flightDelays relation, calculate the **longest delays** of the following delay categories: CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, and LATE_AIRCRAFT_DELAY. Name your pig script longestDelays. Attach a screenshot showing that you successfully ran your pig script, including the logs.

8. Find out all the information (e.g., the flight number, date, and airports, etc.) that was related to the longest CARRIER_DELAY, 1975.0 minutes. Write a Python UDF to go with your Pig script to produce an output like the following screenshot (use the same language and layout). Name your Python file **flight_delay_udf.py** and save it in the /FP directory. Name your UDF **get_max**. You are required to send the values from the Pig script to the Python UDF with a tuple of tuples. List your Python code and Pig script. Name your pig script flight_delays_udf. Attach a screenshot showing that you successfully ran your pig script, including the logs.



9. Create a new relation named **allTheDelays** based on the flightDelays relation by only including the following columns: FL_DATE, FL_NUM, CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY and LATE_AIRCRAFT_DELAY, and then created another relation called **theDelays** based on the allTheDelays relation by only including the rows whose columns of CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY and LATE_AIRCRAFT_DELAY are not null. Store theDelays to the following HDFS directory: **/final_project/theDelays**. Name your pig script allTheDelays. Attach a screenshot showing that you successfully ran your pig script, including the logs.

   After that, you should see eight files in the above directory. Navigate to the directory with the Files View in Ambari, and delete the file named _**SUCCESS**. Attach a screenshot of the contents of your /final_project/theDelays directory.

10. With the seven files in the /final_project/theDelays directory, create seven Hive tables, **fd1_t** to **fd7_t** in the **FPdb** database (if your computer produces a different number of files other than 7, create the first three files, i.e., fd1_t to fd3_t, exactly as specified below in steps a and b; for creating the rest of your tables, use the method in step c):

    a. In Hive View, create the first two tables fd1_t and fd2_t by loading **part-m-00000** and **part-m-00001** respectively. Note that the field values are separated by tabs. Use the same column names as defined in Pig, and set the data types of the first two columns (FL_DATE and FL_NUM) to **string** and the other five columns to

double. Attach a screenshot of the screen when you create fd1_t, right before you are ready to click the "Upload Table" button.

Open the tables in Hive View and attach a screenshot of the first 10 rows of each table.

b. With the Query Editor in Hive View, define the third table, fd3_t, with the same column names and data types as in fd1_t. Load part-m-00002 into fd3_t. Do a query to display the first 10 rows of fd3_t. Attach a screenshot.

c. In Hive View, create the rest of tables (fd4_t to fd7_t) based on the table definition of fd3_t and load the rest of data files (**part-m-00003 to part-m-00006**) respectively. Do a query to display the first 10 rows of fd7_t. Attach a screenshot.

11. In Hive View, use HiveQL to create a new table called **fd_t** by combining the seven Hive tables created above. Use the "describe formatted" command in Hive CLI to describe fd_t and attach a screenshot of your result.

12. Query your fd_t table to find out the **longest delays** and the **average delays** (round to 2 decimal points) of each delay category. Assign an alias to each result with the max_ or mean_ prefix to the column names, e.g., max_carrier_delay. You should see the same results as you got previously in Pig scripting.

13. Create a Hive view called **averageDelays_v**, which can produce the average delay results of Question 12. Use **vi** to create a Python **UDF** for Hive to find out the delay category that has the longest average delay time. Name the Python file as **FindMaxAverageDelayType.py**. List your Python code. Use your UDF in Hive CLI to run your Hive script. Attach a screenshot containing your Hive script (your text version of this script must also be included in your submission since we will run your script), Hive system log and the output of your Hive script. Your output should say that "The delay category with the longest average delay is XX; the average delay time is YY minutes." Here, XX and YY are replaced by actual values in your report.

14. It will be interesting to know if these delay categories are correlated or not. If they are, and then to what degree are they correlated? A common measure for this is the Pearson Correlation Coefficient (r), which can be calculated with the following formula:

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\, n\Sigma x^2 - (\Sigma x)^2 \,][\, n\Sigma y^2 - (\Sigma y)^2 \,]}}$$

or with a Hive built-in function, CORR(col1, col2), where col1 and col2 are two numeric Hive table columns. Query the fd_t table to calculated the r value (round to 4 decimal points, the same below) of the weather_delay and the carrier_delay categories with both methods. You should get the same results with these two methods. Assign an alias to each result. For example, use **w_c** as an alias for the **w**eather_delay and **c**arrier_delay pair. After that, use the CORR() function to calculate all the r values of the 10 possible pairs of delay categories, in a single query; use aliases for all your pairs. Comment on your results, and attach screenshots of your results. (Hint: you will need to use several Hive built-in functions to calculate the r values with the formula.)