Hadoop Map Reduce Assignment – Lengthwise Word Count Problem

Write a MapReduce program to calculate the size of each word (Number of alphabets in the word) and count the number of words of that size in a text file.

Example:

Input: *“Hello everyone this is a sample dataset. Calculate the word size and count the number of words of that size in this text file.”*

Output:

Word Size	**Word Count**
1	1 (As the word of size 1 is: a)
2	4 (As the words of size 2 are: is, of, of, in)
3	3 (As the words of size 3 are: the, and, the)
4	5 (As the words of size 4 are: this, word, size, that, size)

Hive Assignment – yellow Taxi Trip Analysis using HIVE

Problem statement:

In this case study, we are giving a real-world example of how to use HIVE on top of the HADOOP for different exploratory data analysis. In here, we have a predefined dataset (2018_Yellow_Taxi_Trip_Data.csv) having more than 15 columns and more than 100000 records in it. The dataset has different attributes like

1. vendor_id string,
2. pickup_datetime string,
3. dropoff_datetime string,
4. passenger_count int,
5. trip_distance DECIMAL(9,6),
6. pickup_longitude DECIMAL(9,6),
7. pickup_latitude DECIMAL(9,6),
8. rate_code int,
9. store_and_fwd_flag string,
10. dropoff_longitude DECIMAL(9,6),
11. dropoff_latitude DECIMAL(9,6),
12. payment_type string,
13. fare_amount DECIMAL(9,6),
14. extra DECIMAL(9,6),

15. mta_tax DECIMAL(9,6),
16. tip_amount DECIMAL(9,6),
17. tolls_amount DECIMAL(9,6),
18. total_amount DECIMAL(9,6),
19. trip_time_in_secs int

Perform taxi trip analysis by solving the questions below:

1. What is the total Number of trips (equal to the number of rows)?
2. What is the total revenue generated by all the trips? The fare is stored in the column total_amount.
3. What fraction of the total is paid for tolls? The toll is stored in tolls_amount.
4. What fraction of it is driver tips? The tip is stored in tip_amount.
5. What is the average trip amount?
6. What is the average distance of the trips? Distance is stored in the column trip_distance.
7. How many different payment types are used?
8. For each payment type, display the following details:
    - Average fare generated
    - Average tip
    - Average tax – tax is stored in column mta_tax
9. On an average which hour of the day generates the highest revenue?

Data information:

Yellow taxi trip analysis using Hive:

- You can get the complete data online through this link - https://data.cityofnewyork.us/Transportation/2018-Yellow-Taxi-Trip-Data/t29m-gskq
- The data dictionary is also available and you can refer to the attached document.

Preliminary Tasks related to taxi data:

1. Create a table named taxidata . Required ddl script is given below:

    CREATE TABLE IF NOT EXISTS taxidata (vendor_id string, pickup_datetime string, dropoff_datetime string, passenger_count int, trip_distance DECIMAL(9,6), pickup_longitude DECIMAL(9,6), pickup_latitude DECIMAL(9,6), rate_code int, store_and_fwd_flag string, dropoff_longitude DECIMAL(9,6), dropoff_latitude DECIMAL(9,6), payment_type string, fare_amount DECIMAL(9,6), extra DECIMAL(9,6), mta_tax DECIMAL(9,6), tip_amount DECIMAL(9,6), tolls_amount DECIMAL(9,6), total_amount DECIMAL(9,6), trip_time_in_secs int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED as TEXTFILE TBLPROPERTIES ("skip.header.line.count"="1")

2. Load data from the csv file - 2018_Yellow_Taxi_Trip_Data..csv
3. Run some basic queries to check the data is loaded properly.

Spark Assignment – Finding the most frequent words using Spark RDDs

Write code that calculates the most common words from Complete Works of William Shakespeare dataset.

Suggested steps for writing the word counting program:

- Create a base RDD from Complete_Shakespeare.txt file.
- Use RDD transformation to create a long list of words from each element of the base RDD.
- Remove stop words from your data.
- Create pair RDD where each element is a pair tuple of ('word', 1). Here 'word' is each word in the RDD
- Group the elements of the pair RDD by key ('word') and add up their values.
- Swap the keys (word) and values (counts) so that keys is count and value is the word.
- Finally, sort the RDD by descending order and print the 10 most frequent words and their frequencies.

Stop words are common words that are often uninteresting. For example, "I", "the", "a" etc., are stop words. Here is a set of stop words which you can use for this exercise, but you can remove many obvious stop words with a list of your own.

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 'can', 'will', 'just', 'don', 'should', 'now']

Points to note: The assignment must include appropriate explanation for each line of code. Screenshots of the successful execution along with the output should be included. The assignment must be your own work. If plagiarism is found, the student will be awarded zero for the assignment.