Homework 5: Hadoop and Spark

DSCI 551 – Fall 2024

Due: 11:59pm, November 26, 2024, Tuesday

Points: 100

1. [Hadoop MapReduce, 40 points] Using the provided data set: coffee_shop_sales_2023.csv, write a Hadoop MapReduce program to find answers to the following SQL query.

   ```
   SELECT store_location, sum(transaction_qty * unit_price)
   FROM coffee_shop_sales
   where product_category = 'Coffee'
   group by store_location
   having count(*) >= 20000
   ```

   Your program should use the provided template SQL2MR.java and fill in the code in the marked areas.

   Reminder on the step of compiling and running the program.

   ```
   hadoop com.sun.tools.javac.Main SQL2MR.java
    jar cf sql2mr.jar SQL2MR*.class
   hadoop jar sql2mr.jar SQL2MR input output
   ```

   Note that you need to remove the first row, the header row from the provided csv file, and store it under the input directory.

   Submission:
   - SQL2MR.java
   - sql2mr.jar
   - part-r-00000 file

2. [Spark DataFrame, 30 points] Write a Spark DataFrame script for each of the following questions. Do not use spark.sql(…). Json files, "employees.json", etc. are attached with this handout.

Please fill the missing part in the template provided for this question [q2_spark_dataframe.py]

Assume that the following statements have been executed:

```
import pyspark.sql.functions as fc

employees = spark.read.json('employees.json', multiLine=True)
locations = spark.read.json('locations.json', multiLine=True)
departments = spark.read.json('departments.json', multiLine=True)
projects = spark.read.json('projects.json', multiLine=True)
assignments = spark.read.json('assignments.json', multiLine=True)
```

(1) Find name and location of employees working in North America (region).
(2) Find names of employees who are not managers.
(3) Find the name and salary of employees who worked at the "Marketing" department and are located in "Berlin". You can assume no two departments (or locations) have the same name.
(4) Find names of employees who worked for at least 100 hours for projects. Return names of employees and the total number of hours they have worked. Assume that no two employees have the same name. Order the names alphabetically (ascending).
(5) Find projects which have been assigned with the largest number of employee hours. Note there may be multiple such projects. Return project names only.

3. [Spark RDD, 30 points] Write an RDD script for the same questions as in the previous question. Note your code should parallel the computation as much as possible.
   You can assume that the corresponding RDDs have been created for you.

Please fill the missing part in the template provided for this question [q3_spark_rdd.py]

```
employees_rdd = employees.rdd
departments_rdd = departments.rdd
locations_rdd = locations.rdd
projects_rdd = projects.rdd
assignments_rdd = assignments.rdd
```

(1) Find name and location of employees working in North America (region).
(2) Find names of employees who are not managers.
(3) Find the name and salary of employees who worked at the "Marketing" department and are located in "Berlin". You can assume that there is only one such department and only one such location.
(4) Find names of employees who worked for at least 100 hours for projects. Return names of employees and the total number of hours they have worked. Assume that no two employees have the same name. Order the names alphabetically (ascending).
(5) Find projects which have been assigned with the largest number of employee hours. Note there may be multiple such projects. Return project names only.

Submission Instructions:

1. Please submit Only 5 files [Q1 - SQL2MR.java, sql2mr.jar, part-r-00000 || Q2 - q2_spark_dataframe.py || Q3 - q3_spark_rdd.py]
2. Make sure to submit the updated files with solutions
3. Do not modify any contents in the template. Just fill the template by reading the comments.
4. Your code should work for different test cases.
5. [Q2, Q3] Also add the output for each query as a comment after the method as specified in the templates.
6. Please open all the folders to get access to the templates and datasets in attached **data_templates**.zip