Parallel and Distributed Computing
Lab6
Advanced SparkSQL

In this lab you will read and write directly with DataFrames, perform joins, queries and partition while saving Parquet files.

The datafile

For this lab, we will be using the same baseball CSV files from Kaggle (https://www.kaggle.com/open-source-sports/baseball-databank/data) as in Lab5. You will need to load in Master.csv, Teams.csv and AllstarFull.csv. Please refer to the Kaggle site for more specific information on this dataset.

Creating the team allstars DataFrame

We want to create a new DataFrame that holds all first and last names of all the allstars, organized by team. Specifically, the DataFrame should have the following columns and only have that information for players who are allStars:
playerID
teamID
nameFirst
nameLast
teamName (from the name field in the Teams.csv)

To create this DataFrame, you should first load in the three CSV files. You can basically view the AllstarFull.csv file as a linking table to connect the other two tables together.

When loading, use the spark.read command to load the information directly into a DataFrame. You will not care about all the columns. In particular, you will only care about the playerID, nameFirst, and nameLast from Master; teamID and name from Teams; And playerID and teamID from AllstarFull. Since all these columns are Strings, we can just let the CSV reader infer the Schema as it will do it correctly in these cases.

Once your DataFrames are loaded, you should join them together appropriately. For this lab, I am requiring you to perform all your operations with DataFrame functions. That is, do not register a temporary table and use SQL. You will want to get rid of duplicate information (distinct()) is good for this task. You will also want to think about the order you do things to reduce the amount of data shuffled on joining.

Saving as Parquet

After you have created the new DataFrame of information, you are to save it out as a Parquet file so it can be read in later for queries. In particular, the types of queries we are expecting to happen on the data will be things like asking the names of all the allstars for a particular team. Thus, it would be a good idea to partition by teamName when writing out the Parquet file.

Read and Query

Lastly to test your Parquet file, in a separate command, read in the DataFrame you previously saved. And perform the query: give the first and last names of all the Colorado Rockies allstars. Note that you should specifically use the name "Colorado Rockies" rather than the internal teamID ("COL") since that is internal to the DataFrame and not supposed to be known for outside queries.

You should display both the number of all stars as well as showing the full list of them. Note there should be 24.

Submission

You are upload two documents as your submission. The first is your Python source code (PY). Make sure to have your name in a comment at the top of your source code. Seccond is a document that has a cut-paste of your results from running your code.