Assessment for DE Internship at DataGrokr

Thank you for your interest in the Data Engineering Internship at DataGrokr.

We anticipate the selected candidates to be working in Data Engineering and Cloud-related projects. As such for this given assignment, we'd like to test candidates' skills in those areas. Candidates who are already proficient in SQL, Python, and Spark will have an edge in this assignment but even if you don't know anything about any of these technologies you should be able to do this assignment by following along the instructions and studying the links provided.

Please note that this ability to learn new technologies while following instructions would really help you in your day-to-day activities at DataGrokr.

What you need to do:

The objective of the assignment is to test your proficiency in querying and data analysis. The assignment has 3 parts.

- **Section 1:** Setting up PySpark in Colab and loading some data sets.
- **Section 2:** Analyze the given dataset and answer the question using PySpark.
- **Section 3:** Save the results from Section 2 into your google drive.

Note:

- Please follow industry standards while writing the code. Preferred Programming language – Python

Section 1: Environment setup and data loading

1. Open this link to create a new Colab notebook (You need to sign in to your google account if not signed). Follow the below steps to setup spark in your notebook
    - Update currently installed packages in your Google Colab Notebook's runtime

        -
          ```
          !apt-get update -y
          ```
    - Spark is written in the Scala programming language and requires the Java Virtual Machine (JVM) to run. Therefore, our first task is to download Java.

        -
          ```
          !apt-get install openjdk-8-jdk-headless -qq > /dev/null
          ```
    - Next, we will download and unzip Apache Spark with Hadoop 2.7 to install it.

        -
          ```
          !wget -q https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop2.7.tgz
          ```

        -
          ```
          !tar xf spark-3.1.2-bin-hadoop2.7.tgz
          ```
    - Setup Environment variables for Java and Spark

        -
          ```
          import os
          os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
          os.environ["SPARK_HOME"] = "/content/spark-3.1.2-bin-hadoop2.7"
          ```
    - Then we need to install and import the 'findspark' library that will locate Spark on the system and import it as a regular library.

        -
          ```
          !pip install -q findspark
          import findspark
          findspark.init()
          ```
    - Now, import SparkSession from pyspark.sql and create a SparkSession, which will be the entry point to Spark.

   -
     ```
     from pyspark.sql import SparkSession

     spark = (SparkSession
       .builder
       .appName("<app_name>")
       .getOrCreate())
     ```

2. Google Drive link for the files needed to complete this assignment.
   - Authenticate Google Drive onto your colab notebook.
   - Write python function to download all the files from the above drive link into the content directory of colab notebook.
     - For Reference

3. Create dataframes for each of the datasets. Give proper column names and datatypes. Include basic schema validation. (Refer the schema provided with the data for reference) Check out spark.read.format from here to create spark dataframe.

4. Create Temporary Table (View) for All the required dataframe like this. So, you can apply SQL Queries on that dataframe(Don't need to worry If you are not comfortbale with pyspark Function. You just need to Write SQL queries and run that as mentioned below in on pyspark dataframe.)
   For Example: - Let's say you have Dataframe as df and you want to make Temporary Table as temp_table to apply SQL Query on Dataframe . You can do like this-

   df.createOrReplaceTempView("temp_table") – You have Temporary table name temp_table on which you can apply SQL Query.
   query= '''Select * from temp_table''' --- Query to Select Content of temp_table (Your Query will be here)
   df1 = spark.sql(query) --- Executing Query and storing Query result in Dataframe named df1
   df1.show(truncate=False) --- Show the Query result stored in df1

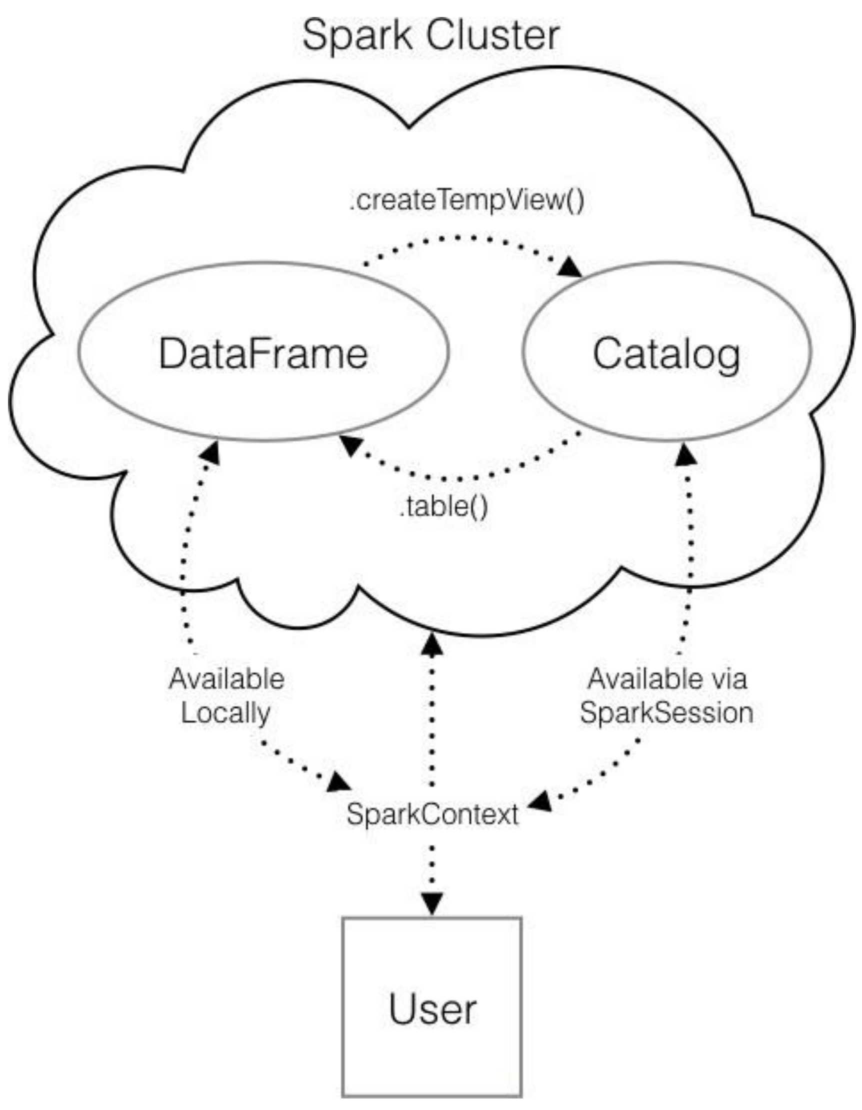
   Refer this video for more info.

References to learn about pyspark

1. Your SparkSession has an attribute called `catalog` which lists all the data inside the cluster. This attribute has a few methods for extracting different pieces of information.
   One of the most useful method is the .listTables() method, which return all table names in your cluster as a list.
   Register those dataframes as tables using .createOrReplaceTempView() , this method registers the DataFrame as a table in the catalog, but as this table is temporary, it can only be accessed from the specific SparkSession used to create the Spark DataFrame. To read more about it, refer to this link

2. Check out this pyspark documention if you are new to it.

3. Check out the diagram to see all the different ways your Spark data structures can interact with each other.



Section 2: Using pySpark for data analysis

Note: WCh knock out(Sub-String with k.o. and KO) are not included in main event. So, filter the event having Sub-String as k.o and KO in that specific Table Column.

1. List of Winners of Each World champions Trophy **Hint:** Total Result of all rounds of Tournament for that player is considered as that player's Score/Result.
   Result attributes: winner, tournament_name
2. List of Players with number of times they have won Tournament in descending order(Max to min).
   Result attributes: player_name, number_of_wins
3. Most and Least Popular eco move in world championship history.
   Result attributes: eco, eco_name, number_of_occurences
   Final result will have only two rows
4. Find the eco move with most winnings.
   Ps. Use this opening move in your next chess game 🙂
   Result attributes: eco, eco_name
5. Longest and shortest game ever played in a world championship in terms of move.
   Chess Funda: "move" is completed once both White and Black have played one turn. e.g If a game lasts 10 moves, both White and Black have played 10 moves)
   Result attributes: game_id, event, tournament_name, number_of_moves
   Final result will have only two rows
6. Shortest and Longest Draw game ever Played.
   Result attributes: game_id, event, tournament_name, number_of_moves
   Final result will have only two rows
7. Most and Least rated Player.
   Result attributes: player_name, elo
   Chess Funda: elo is the rating of the player in chess tournament.
   Final result will have only two rows
8. 3rd Last Player with most Loss.
   Result attributes: player_name
   Final result will have only one row
9. How many times players with low rating won matches with their total win Count.
   Result attributes: player_name, win_count
10. Move Sequence for Each Player in a Match.
    Result attributes: game_id, player_name, move_sequence, move_count
11. Total Number of games where losing player has more Captured score than Winning player.
    Hint: Captured score is cumulative, i.e., for 3rd capture it will have score for 1, 2, and 3rd.
    Result attributes: total_number_of_games Final result will have only one row
12. List All Perfect Tournament with Winner Name.
    Chess Funda: Perfect Tournament means a player has won all the matches excluding draw matches. e.g Player A has won 5 matches out of 7 Matches in tournament where 2 matches are draw and player B has won 0 matches)
    Result attributes: winner_name, tournament_name
13. Player with highest winning ratio.
    Hint: Winning ratio: (Number of rounds won)/(Number of rounds played)
    Result attributes: player_name
    Final result will have only one row
14. Player who had given checkmate with Pawn.
    Note: Consider all events for this query
    Result attributes: player_name
    Final result will have only one row
15. List games where player has won game without queen.
    Result attributes: game_id, event, player_name

Section 3: Save results from Section 2 into google drive

Create a python function to do these tasks:

1. Function will accept list of spark dataframe as argument.

2. Mount your Google Drive if not mounted onto your colab notebook.

3. Create a directory in your google drive if not created with name: **DE_SOLUTION_*FirstName_LastName*/results/**
   - Create directory using **os**, python's built-in library.
   - Current working directory is **'/content/'**, so to create the above directory you need to pass **'gdrive/MyDrive/DE_SOLUTION_*FirstName_LastName*/results/'** as your path parameter.

4. Convert spark dataframe to pandas dataframe.
   - **Note:** We are converting it to pandas dataframe to avoid having multiple files in parts saved for a single csv file.

5. Convert the pandas dataframe to csv and save it to goggle drive under **DE_SOLUTION_*FirstName_LastName*/results/**.

6. File name should be df1.csv, df2.csv, and so on... And they should be inside **results folder**.

Deliverables:

1. A single colab notebook where you have developed the code for the Section 1, Section 2 and Section 3.
   - Move the colab notebook to **DE_SOLUTION_*FirstName_LastName*/** folder in your drive with name **Assignment_Solution_*FirstName_LastName*.ipynb**
   - Upload your up-to-date resume to **DE_SOLUTION_*FirstName_LastName*/** folder in your drive with name ***FirstName_LastName*.pdf**
   - Now email us your **google drive folder link** of **DE_SOLUTION_*FirstName_LastName*/**
     - **NOTE: Make sure to share the google drive link after choosing: *Anyone with the link* and *Viewer Mode***
   - We will run the colab notebook on our end and correct your submissions.

2. Your code will be evaluated not just on the basis of final results but also on code quality. Here are few tips:
   - Follow coding standards (PEP-8)
   - Appropriate error/exception handling
   - Modular function design
   - Schema and data validation

3. Your final submission are due to us by end of day 31st October 2022 and subject should follow following pattern <Source e.g.: Internshala, College name> Data Engineer:

4. Please include your up-to-date resume, named as Firstname_Lastname.pdf

If you have any questions during the assignment, send your questions to dataengineering@datagrokr.com

Good luck and we hope you learn something new in this process!