Course Project. Hive.

Task: using baseball historical data, find the highest score for each year.

|  |  |
| --- | --- |
| **Assignment 1** | |
|  | Download two datasets from WebCampus: Batting.csv and Master.csv. |
|  | Connect to FDU VPN. |
|  | Login to Ambari by navigating to <http://132.238.7.193:8080> |
|  | Enter your user name and password. |
|  | Using Ambari, start the HDFS view. |
|  | In HDFS view, navigate to your user folder, which is in the form  **/usr/stXX**  where stXX is your user name. |
|  | Make sure that both Batting.csv and Master.csv are uploaded to your folder in HDFS. |
|  | Open the Hive View. |
|  | Create a new database, which should be named as  stXX\_FirstName  where stXX should be substituted with your user ID and FirstName should be your first name.  To do so, reuse this query but make changes as needed:  CREATE DATABASE stXX\_FirstName; |
|  | Make sure your database is currently active. To do so, you need to select your database in the dropdown list under the Database Explorer label. |
|  | Create a table to hold the data. Initially, it will be a table with a single column. Run this query:  create table temp\_batting (col\_value STRING); |
|  | Check if your database contains temp\_batting table. To do so, refresh the Database Explorer by clicking  . Then click on your database name.  If your query in Step 11 executed successfully but you don’t see the table, make your database active (see Step 10 above) and re-run the query. |
|  | Click on the icon next to the table name to see if it has any rows yet. |
|  | The table should still be empty. Now, load the data into the table by running the following query (do not forget to substitute stXX with your user name):  LOAD DATA INPATH '/user/**stXX**/Batting.csv' OVERWRITE INTO TABLE temp\_batting; |
|  | Check if table temp\_batting has data. |
|  | Answer the question: how many columns are there in table temp\_batting? |
|  | Create another table  create table batting (player\_id STRING, year INT, runs INT); |
|  | Make sure this table is in your database. |
|  | Populate table **batting** with data from table **temp\_batting**:  insert overwrite table batting  SELECT  regexp\_extract(col\_value, '^(?:([^,]\*),?){1}', 1) player\_id,  regexp\_extract(col\_value, '^(?:([^,]\*),?){2}', 1) year,  regexp\_extract(col\_value, '^(?:([^,]\*),?){9}', 1) run  from temp\_batting; |
|  | Make sure table **batting** is populated. |
|  | Answer the question: how many columns does the **batting** table have? |
|  | Now, find the largest number of runs per year:  SELECT year, max(runs) FROM batting GROUP BY year; |
|  | Now, join the largest runs per year back to the table in order to get player IDs.  SELECT a.year, a.player\_id, a.runs  FROM batting a  JOIN (SELECT year, max(runs) as maximumruns  FROM batting GROUP BY year ) b  ON (a.year = b.year AND a.runs = b.maximumruns); |
|  | Answer the question: what player had the largest number of runs in 1963? What is the largest number of runs for 1963? |

Assignment 2 (for extra credit)

|  |  |
| --- | --- |
| 1. | Modify the query in Step 23 to show the max number of runs for each year in 1970s. |

Assignment 3 (for extra credit)

|  |  |
| --- | --- |
| 1. | Download ManagersHalf.csv from WebCampus. |
| 2. | Similarly to the steps above do the following:   1. Import the file into a temporary table in your database with a single column. 2. Make a permanent table with 10 columns. 3. List ManagerIDs with largest rank. |