Pig Use Case: Baseball Stats

**Exercise Objectives:**

* You will be given a fairly large set of game statistics (over 97,800 rows)
* Stats for each American baseball player by year from 1871-2013
* Identify players who scored highest runs for each year in ascending order
* Determine First and Last name for the each player by joining 2 data sets
* You should try to accomplish this with minimal instruction

**Review and understand Baseball statistics data files**

1. In your VM, create a new directory in your home directory to use for sample data.

2. Via a browser, get the latest baseball stats file from Sean Lahman’s baseball stats [web site](http://www.apple.com) (the file version may change.) On the page http:// www.seanlahman.com/baseball- archive/statistics you’ll see the link for downloading some comma-delimited CSV files in a zipped archive. Get that archive (easiest using your VM browser), expand it and do a put of the files needed into HDFS.

NOTE: you can also achieve this from a Linux shell like this:

**$** wgethttp://seanlahman.com/files/database/ baseballdatabank-master\_2016-03-02.zip

3.Unzip the file in your new directory. Many statistics files will unpack from the file.

4. Examine the following file:

**Batting.csv**

| # | NAME | DESCRIPTION |  |
| --- | --- | --- | --- |
| 0 | playerID | A unique code assigned to each player. The playerID links the data in this file with records in other files. | |
| 1 | yearID | Year | |
| 2 | stint | Player’s Stint (Order of appearence in season.) | |
| 3 | teamID | Team | |
| 4 | lgID | League | |
| 5 | G | Games | |
| 6 | AB | At Bats | |
| 7 | R | Runs | |
| 8 | H | Hits | |
| 9 | 2B | Doubles | |
| 10 | 3B | Triples | |
| 11 | HR | Homeruns | |
| 12 | RBI | Runs Batted In | |
| 13 | SB | Stolen Bases | |
| 14 | CS | Caught Stealing | |
| 15 | BB | Base on Balls | |
| 16 | SO | Strikeouts | |
| 17 | IBB | Intentional Walks | |
| 18 | HBP | Hit By Pitch | |
| 19 | SH | Sacrifice Hits | |
| 20 | SF | Sacrifice Flies | |
| 21 | GIDP | Grounded Into Double Plays | |

5. Consider following fields/columns:

Column # 0 (Player ID)

Column # 1 (Year)

Column # 7 (Runs)

6. Put the file Batting.csv into HDFS under an appropriate directory, such as

batting/input. You may wish to rename the file in HDFS. Use hdfs dfs

–cat to verify the put if needed.

**Review and understand Baseball statistics data files**

1. Examine following files. Here is the column definition for the join file:

**Master.csv**

| # | NAME | DESCRIPTION |  |
| --- | --- | --- | --- |
| 0 | playerID | Unique ID for Player | |
| 1 | birthYear | Year Player was Born | |
| 2 | birthMonth | Month Player was Born | |
| 3 | birthday | Day Player was Born | |
| 4 | birthCountry | Country Where Player was Born | |
| 5 | birthState | State Where Player was Born | |
| 6 | birthCity | City Where Player was Born | |
| 7 | deathYear | Year Player Died | |
| 8 | deathMonth | Month Player Died | |
| 9 | deathDay | Day Player Dies | |
| 10 | deathCountry | Country Where Player Died | |
| 11 | deathState | State Where Player was Died | |
| 12 | deathCity | City Where Player was Died | |
| 13 | nameFirst | Player’s First Name | |
| 14 | nameLast | Player's Last Name | |
| 15 | nameGiven | Player's Given Name (Typically First & Middle) | |
| 16 | weight | Player's Weight in Pounds | |
| 17 | height | Player's Height in Inches | |
| 18 | bats | Player's Batting Hand (Left, Right, or Both) | |
| 19 | throws | Player's Throwing Arm (Left or Right) | |
| 20 | debut | Date that Player Made First Major League Appearence | |
| 21 | finalGame | Date Player Made Last Major League Appearence | |
| 22 | more fields… |  | |

For this lab, you should consider following fields/columns:

**Batting.csv**

* + - * Column # 0 (Player ID)
      * Column # 1 (Year)
      * Column # 7 (Runs)

**Master.csv**

* + - * Column # 0 (Player ID)
      * Column # 13 (First Name)
      * Column # 14 (Last Name)

**Identify players who scored highest runs for each year**

1. If you haven’t yet, create a new directory, such as baseball/input in HDFS and

put statistic files Batting.csv & Master.csv into this directory from your local machine.

You may rename them in HDFS as you wish.

2.Start the Pig shell

$ pig

3.Load batting data into Pig using the PigStorage()function.

NOTE: The default delimiter in Pig is a TAB (\t). A CSV file has comma- separated data in each line, so we need to inform Pig explicitly about that field delimiter. See here for the correct function syntax.

4.Read relevant fields from the loaded data. In this case we are interested in 1st , 2nd

and 9th fields for each record. Use FOREACH-GENERATE statements to accomplish

this task. Look [here](http://pig.apache.org/docs/r0.10.0/basic.html%2523foreach) for correct FOREACH syntax.

HINT: In Pig Latin $0 can represent the 1st field, $1 the 2nd field and so on. In the

GENERATE statement, you can use the “$X AS (alias:type)” syntax to create text

aliases for positional fields. More on this here.

5. Group runs from step 4 by year. See the syntax for GROUP here.

HINT: use DUMP and DESCRIBE to validate your assumptions along the way.

6. Use FOREACH-GENERATE, GROUP and MAX functions in Step 5 data to get max runs for each year.

7. Now, join Step 4 and Step 6 data based on the ‘year’ and ‘runs’ fields. Inner and outer joins syntax is shown here.

8. To identify the playerID who scored the highest one for each year, create a new bag with Year, PlayerID and Max Run data using FOREACH, GENERATE on Step7 data.

9. Check the output of the above exercise using a DUMP command.

**Determine First and Last name for the each player**

1. Load master data to Pig using the PigStorage() function.

2. Read relevant fields from the file. In this case we are interested in 13th & 14th fields for each record using FOREACH and generate command:

3. Join PLAYERS dataset with the result dataset from previous task (Step 9) based on the common field ‘playerID’:

4. Create a new dataset having Year, Player’s First and Last Name and the MAX from Step #3 using FOREACH and GENERATE commands.

5. Make sure that data in Step 4 is sorted on Year in ascending order.

6. Get the output of Step 5 using DUMP command.

**[OPTIONAL] Run a similar job using Pig Streaming**

The syntax for Pig streaming is [here](http://pig.apache.org/docs/r0.10.0/basic.html%2523stream).

1. Write a simple script to take stdin and massage to stdout

2. In Pig, LOAD data in using PigStorage() syntax.

3. Use DEFINE to reference your script in Pig.

4. Try to come up with similar results as before.

**END**