Apache Hive (Part 5)

A Case Study Using Hive

Baseball Data Analysis

Some Baseball Data

We have two data files, Batting.csv (95195 lines of text) and Master.csv (17916 lines of text).

Batting.csv (partial)

```
aardsda01,2004,1,SFN,NL,11,11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,11
aardsda01,2006,1,CHN,NL,45,43,2,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,45
aardsda01,2007,1,CHA,AL,25,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2
aardsda01,2008,1,BOS,AL,47,5,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,5
aardsda01,2009,1,SEA,AL,73,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
aardsda01,2010,1,SEA,AL,53,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
aaronha01,1954,1,ML1,NL,122,122,468,58,131,27,6,13,69,2,2,28,39,,3,6,4,13,122
aaronha01,1955,1,ML1,NL,153,153,602,105,189,37,9,27,106,3,1,49,61,5,3,7,4,20,153
aaronha01,1956,1,ML1,NL,153,153,609,106,200,34,14,26,92,2,4,37,54,6,2,5,7,21,153
aaronha01,1957,1,ML1,NL,151,151,615,118,198,27,6,44,132,1,1,57,58,15,0,0,3,13,151
aaronha01,1958,1,ML1,NL,153,153,601,109,196,34,4,30,95,4,1,59,49,16,1,0,3,21,153
aaronha01,1959,1,ML1,NL,154,154,629,116,223,46,7,39,123,8,0,51,54,17,4,0,9,19,154
aaronha01,1960,1,ML1,NL,153,153,590,102,172,20,11,40,126,16,7,60,63,13,2,0,12,8,153
```

Master.csv (partial)

```
lahmanID,playerID,managerID,hofID,birthYear,birthMonth,birthDay,birthCountry,birthState,birthCity,deathYear
,deathMonth,deathDay,deathCountry,deathState,deathCity,nameFirst,nameLast,nameNote,nameGiven,nameNick,weigh
t,height,bats,throws,debut,finalGame,college,lahman40ID,lahman45ID,retroID,holtzID,bbrefID
1,aaronha01,,aaronha01h,1934,2,5,USA,AL,Mobile,,,,,,,Hank,Aaron,,Henry Louis,"Hammer,Hammerin' Hank,Bad
Henry",180,72,R,R,4/13/1954,10/3/1976,,aaronha01,aaronha01,aaroh101,aaronha01,aaronha01
2,aaronto01,,,1939,8,5,USA,AL,Mobile,1984,8,16,USA,GA,Atlanta,Tommie,Aaron,,Tommie Lee,,190,75,R,R,
4/10/1962,9/26/1971,,aaronto01,aaronto01,aarot101,aaronto01,aaronto01
3,aasedo01,,,1954,9,8,USA,CA,Orange,,,,,,,Don,Aase,,Donald William,,190,75,R,R,7/26/1977,10/3/1990,Cal St.
Fullerton,aasedo01,aasedo01,aased001,aasedo01,aasedo01
4,abadan01,,,1972,8,25,USA,FL,West Palm Beach,,,,,,,Andy,Abad,,,,184,73,L,L,9/10/2001,4/13/2006,Middle
Georgia JC,abadan01,abadan01,abada001,abadan01,abadan01
5,abadijo01,,,1854,11,4,USA,PA,Philadelphia,1905,5,17,USA,NJ,Pemberton,John,Abadie,,John,,192,72,R,R,
4/26/1875,6/10/1875,,abadijo01,abadijo01,abadj101,abadijo01,abadijo01
6,abbated01,,,1877,4,15,USA,PA,Latrobe,1957,1,6,USA,FL,Ft.Lauderdale,Ed,Abbaticchio,,Edward James,Batty,
170,71,R,R,9/4/1897,9/15/1910,,abbated01,abbated01,abbae101,abbated01,abbated01
```

Analysis Goals

- Find the player (ID only) with the highest runs for each year.
- What are the First and last names of the player who had the highest runs in each year? Display the names, year and the highest runs.
- Who (First name and last name) and in what year had the highest runs of all the years in the dataset?
- What is the run average of each year?
- What is the run average of all the years in the dataset?

Create a Temp Table and Load Data Form the Dataset

- In Ambari Hive View, create a table, temp_batting, to hold the data :

  CREATE TABLE **temp_batting** (col_value STRING);

- Load the data file Batting.csv into temp_batting:

  LOAD DATA INPATH "/mydata/lahman591-csv/Batting.csv"
  OVERWRITE INTO TABLE **temp_batting**;

- Use Hive View to display some table contents (first 10 rows).

temp_batting.col_value
aardsda01,2004,1,SFN,NL,11,11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,11
aardsda01,2006,1,CHN,NL,45,43,2,0,0,0,0,0,0,0,0,0,0,0,1,0,0,45
aardsda01,2007,1,CHA,AL,25,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2
aardsda01,2008,1,BOS,AL,47,5,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,5
aardsda01,2009,1,SEA,AL,73,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
aardsda01,2010,1,SEA,AL,53,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
aaronha01,1954,1,ML1,NL,122,122,468,58,131,27,6,13,69,2,2,28,39,,3,6,4,13,122
aaronha01,1955,1,ML1,NL,153,153,602,105,189,37,9,27,106,3,1,49,61,5,3,7,4,20,153
aaronha01,1956,1,ML1,NL,153,153,609,106,200,34,14,26,92,2,4,37,54,6,2,5,7,21,153
aaronha01,1957,1,ML1,NL,151,151,615,118,198,27,6,44,132,1,1,57,58,15,0,0,3,13,151

Create a Hive Table and Load Data Form a Query

- Extract the data: create a Hive table called batting to hold player_id, year and the number of runs:

```
CREATE TABLE batting (player_id STRING, year INT, runs INT);
```

- Insert data extracted from temp_batting with three regular expression (regexp_extract) calls:

```
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) runs
FROM temp_batting;
```

Display the Contents of the *batting* table in Hive View

batting.player_id	batting.year	batting.runs
aardsda01	2004	0
aardsda01	2006	0
aardsda01	2007	0
aardsda01	2008	0
aardsda01	2009	0
aardsda01	2010	0
aaronha01	1954	58
aaronha01	1955	105
aaronha01	1956	106
aaronha01	1957	118
aaronha01	1958	109
aaronha01	1959	116
aaronha01	1960	102
aaronha01	1961	115

Query the Hive Table to Find the Yearly Max Runs

- Group the data by year to find the highest run score for each year.

```
SELECT year, max(runs) max_run
FROM batting
GROUP BY year;
```

year	max_run
1871	66
1872	94
1873	125
1874	91
1875	115
1876	126
1877	68
1878	60
1879	85
1880	91
1881	86
1882	99
1883	110
1884	160

Query the Hive Table Using a JOIN Operation

- Get the player_id(s) to know who the player(s) was, using a JOIN operation on the condition of matching each year's max number of runs:

```
SELECT a.year, a.player_id, a.runs
from batting a
JOIN
(SELECT year, max(runs) max_run
FROM batting
GROUP BY year ) b
ON (a.year = b.year AND a.runs = b.max_run) ;
```

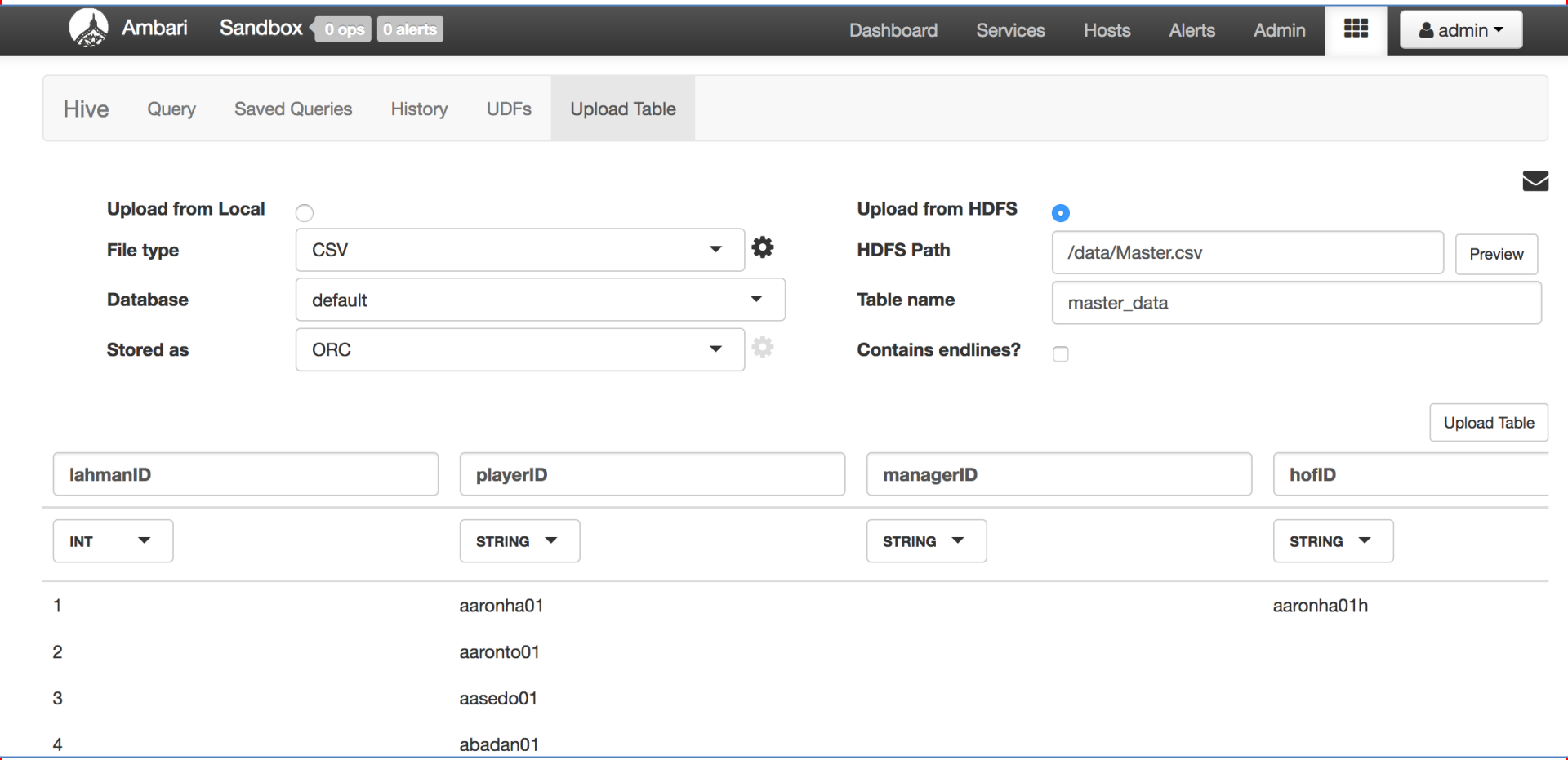
The JOIN Result

a.year	a.player_id	a.runs
1871	barnero01	66
1872	eggleda01	94
1873	barnero01	125
1874	mcveyca01	91
1875	barnero01	115
1876	barnero01	126
1877	orourji01	68
1878	highadi01	60
1879	jonesch01	85
1880	dalryab01	91
1881	gorege01	86
1882	gorege01	99
1883	stoveha01	110
1884	dunlafr01	160
1885	stoveha01	130

...

1996	burksel01	142
1997	biggicr01	146
1998	sosasa01	134
1999	bagweje01	143
2000	bagweje01	152
2001	sosasa01	146
2002	soriaal01	128
2003	pujolal01	137
2004	pujolal01	133
2005	pujolal01	129
2006	sizemgr01	134
2007	rodrial01	143
2008	ramirha01	125
2009	pujolal01	124
2010	pujolal01	115
2011	grandcu01	136

Create Another Table *master_data* Using Ambrari



Create a Hive View

```
CREATE VIEW year_max_runs as
  SELECT a.year, a.player_id, a.runs
  FROM batting a
  JOIN
  (SELECT year, max(runs) max_run
  FROM batting
  GROUP BY year ) b
  ON (a.year = b.year AND a.runs = b.max_run) ;
```

Query: Who Had Highest Runs in Each Year

```
SELECT namefirst, namelast, m.playerid,
    year, runs max_runs
FROM year_max_runs y, master_data m
WHERE y.player_id = m.playerid;
```

Query Result

namefirst	namelast	m.playerid	year	max_runs
Ross	Barnes	barnero01	1871	66
Dave	Eggler	eggleda01	1872	94
Ross	Barnes	barnero01	1873	125
Cal	McVey	mcveyca01	1874	91
Ross	Barnes	barnero01	1875	115
Ross	Barnes	barnero01	1876	126
Jim	O'Rourke	orourji01	1877	68
Dick	Higham	highadi01	1878	60
Charley	Jones	jonesch01	1879	85
Abner	Dalrymple	dalryab01	1880	91
George	Gore	gorege01	1881	86

Find the Highest Runs of All the Years

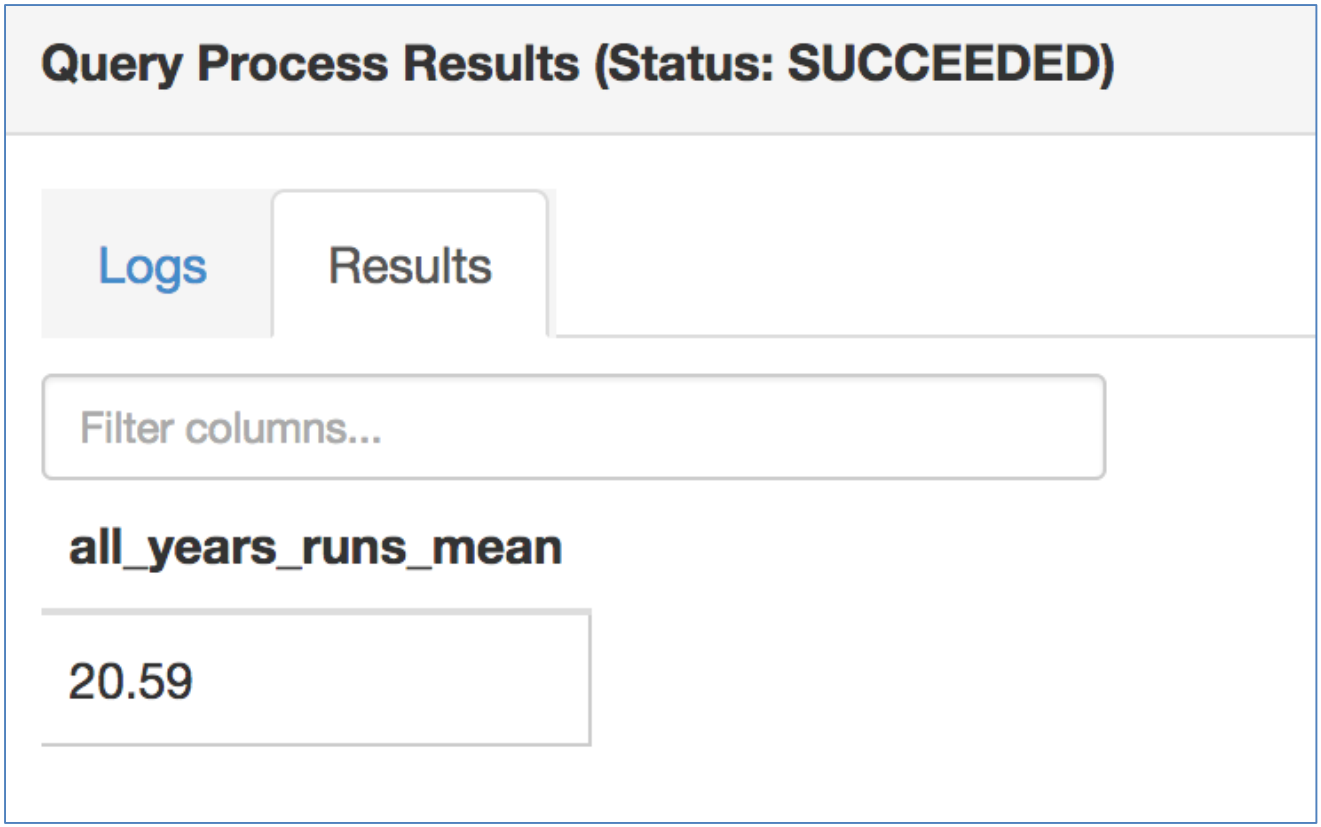
```
SELECT namefirst, namelast, m.playerid,
    year, runs max_runs
FROM (SELECT max(runs) r FROM batting) y,
   master_data m, batting b
WHERE b.player_id = m.playerid
   AND y.r=b.runs;
```

namefirst	namelast	m.playerid	year	max_runs
Billy	Hamilton	hamilbi01	1894	192

Find the Average Runs of Each Year

year	mean_runs
1871	23.12
1872	21.73
1873	28.64
1874	28.21
1875	19.42
1876	24.73
1877	21.03
1878	23.8
1879	26.84

Find the Average Runs of All the Players in All the Years



Using Regular Expression in Hive

^	Beginning of string
?: ()	Grouping the value to return
([^,]*)	Capture chars for buffer 1: a string of many non-comma chars (getting all the charts before you hit a comma).
\,?	Zero or one instance of comma character
{1}	Repeat group exactly 1 (or N) times