# Hortonworks Tutorial 4

## Working with Pig II

In this tutorial, we will implement Pig Latin scripts to find out the highest points by a player for each year that we did in the "Hive II" tutorial. This tutorial enables you to see how the same query can be implemented/written and executed in Pig and Hive differently.

### Task 1. Load Data in Pig

1. Create a new Pig script. In Pig View, click the "+New Script" button in the upper-right corner, name the script "t4", then click the Create button.
2. Load the data using the "PigStorage" function:

   ```
   basketball_players = load '/tmp/FIT5148/basketball_players.csv'
   using PigStorage(',');
   ```

3. To filter out the first row of the data (the headings), we add the following line. "Filter" will return all the rows that meet the criteria after "BY" and will drop/remove the one that does not. What does $\$1 > 0$ mean here? (Hint: first find out which field $1 refers to and why the first row's value for $1 does not meet this condition)

   ```
   players_raw = FILTER basketball_players BY $1 > 0;
   ```

### Task 2. Implement a Script to Name the Fields

Elicit and name the required fields. We will use a "FOREACH" statement to generate data transformations on columns. Enter the code below into the edit, or choose "PIG helper → Relational Operators → FOREACH%DATA%GENERATE%NEW_DATA%" to see the template for this statement.

```
players = FOREACH players_raw GENERATE $0 as playerID,
$1 as year, $3 as tmID, $8 as points;
```

### Task 3. Use "GROUP BY" to Filter The Data (points for each year)

The next line of code is a "GROUP" statement that groups the players by year. You can always use the dump command to examine the output.

```
grp_by_year = GROUP players BY (year);
```

### Task 4. Compose a Script to Search for Max points Per Year

We now use the FOREACH statement to find the maximum points for each year. Use the code below:

```
max_year_points = FOREACH grp_by_year GENERATE group as year_grp,
MAX(players.points) as max_points;
DUMP max_year_points;
```

The result will be like Fig. 1

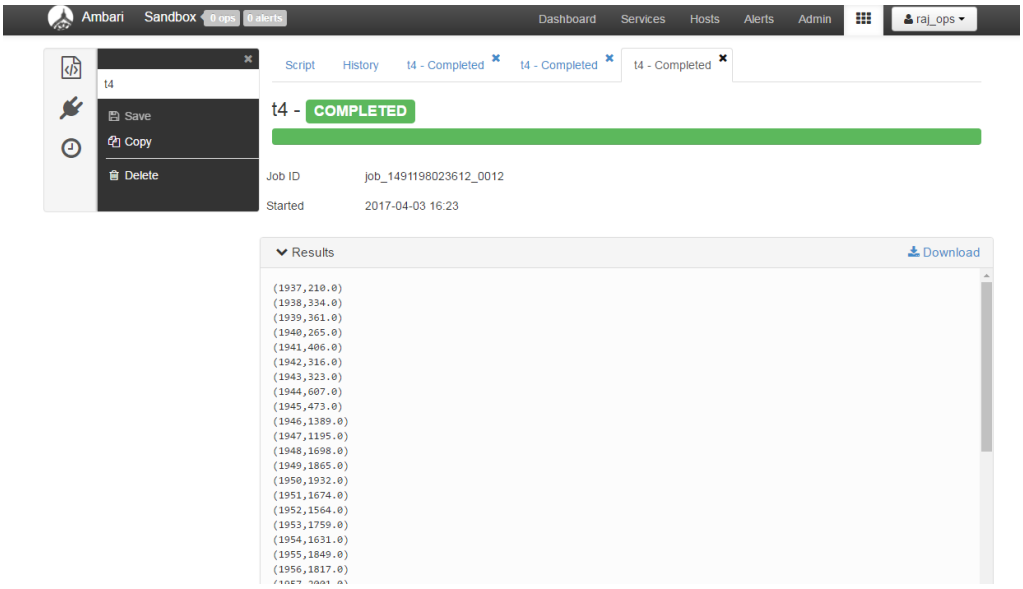


Figure 1: The max group point per year

### Task 5. Build a Script to join Year, PlayerID and Max Points

Now that we have the maximum points and year but we do not know which player or team won the highest point. Therefore, we need to join the results with the players so we can pick up the player ID and team ID. Please add the following code:

```
join_max_points = JOIN max_year_points by ($0, max_points),
players by (year, points);
```

If you dump the output you will see how the results of "max_year_points" and "players" are joined together based on common/matching points and year. You also see that that results now include the "player" and "team IDs" but it needs to be processed further to remove redundant fields. This is accomplished with the following code:

```
max_player_points = FOREACH join_max_points GENERATE $0 as year, $2
as playerID, $4 as tmID, $5 as points;
```

Use the dump command to examine the output. You will the results in the following format: "year", "playerID", tmID, "max_points" (see Fig. 2).

### Task 6. Build a Script to get First Name and Last Name using "JOIN"

1. Before performing the join between the player, team and master relations, we need to perform the same steps (i.e. upload and filter) we did for players for the team and master relations. To filter the first row in the master relation, a different condition is used. The $0 refers to the first field/heading and "bioID" is its value. Why didn't we use the same condition ($> 0$) here? Add the following code:

   ```
   basketball_master = load '/tmp/FIT5148/basketball_master.csv'
   using PigStorage(',');
   master_raw = FILTER basketball_master BY $0 != 'bioID';
   basketball_teams = load '/tmp/FIT5148/basketball_teams.csv'
   using PigStorage(',');
   teams_raw = FILTER basketball_teams BY $0 > 0;
   ```

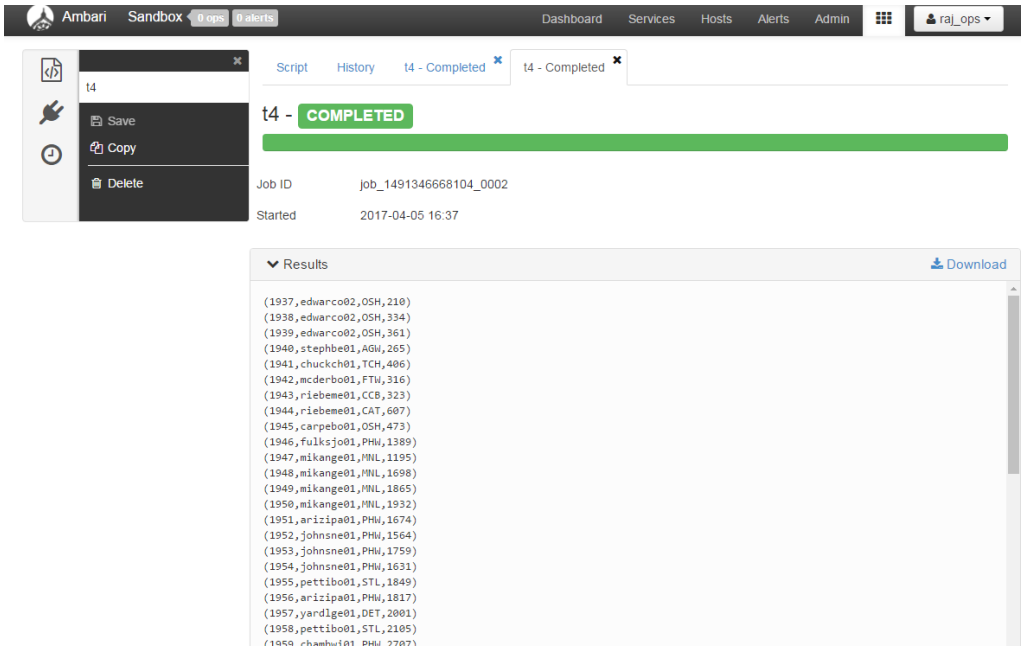


Figure 2: The max player point per year

2. We use FOREACH here again to generate data transformations based on columns of data that we are interested in.

```
master = FOREACH master_raw GENERATE $0 as playerID,
$2 as fname, $4 as lname;
teams = FOREACH teams_raw GENERATE $0 as year, $2 as tmID, $9 as name;
```

3. The next step is to join the result dataset from the previous step with the master relation to get the player's details including their first names and last names. Then we need to perform FOREACH to remove redundant fields and create the output with only the fields that we want:

```
join_player = JOIN max_player_points by playerID, master by playerID;
max_player_info = FOREACH join_player GENERATE $0 as year, $1 as playerID,
$5 as fname, $6 as lname, $2 as tmID, $3 as points;
```

4. After this step, we successfully translate a player id field into the first and last names. See Fig. 3.

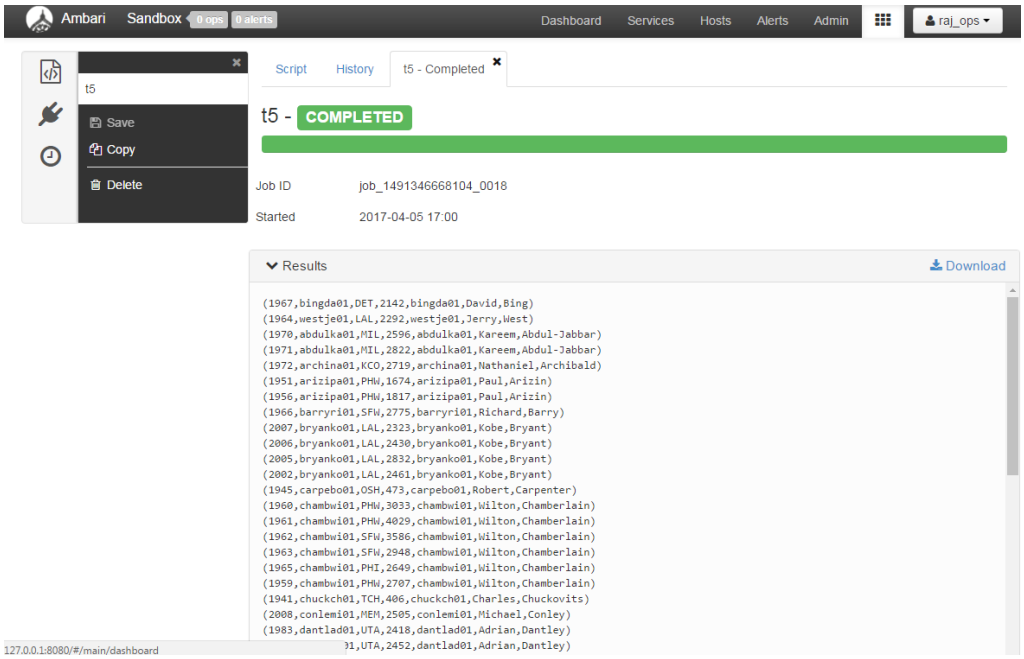


Figure 3: The players' names

### Task 7. Excecise: Join by tmID to get team name

Let's focus on an exercise task. Join the result dataset from the previous task with the teams relation so we can pick up the team name. The result will be a dataset with year, player ID, player name, team name and max points. At the end we sort the result by year and DUMP the data to the output.

The result will be seen as Fig. 4.

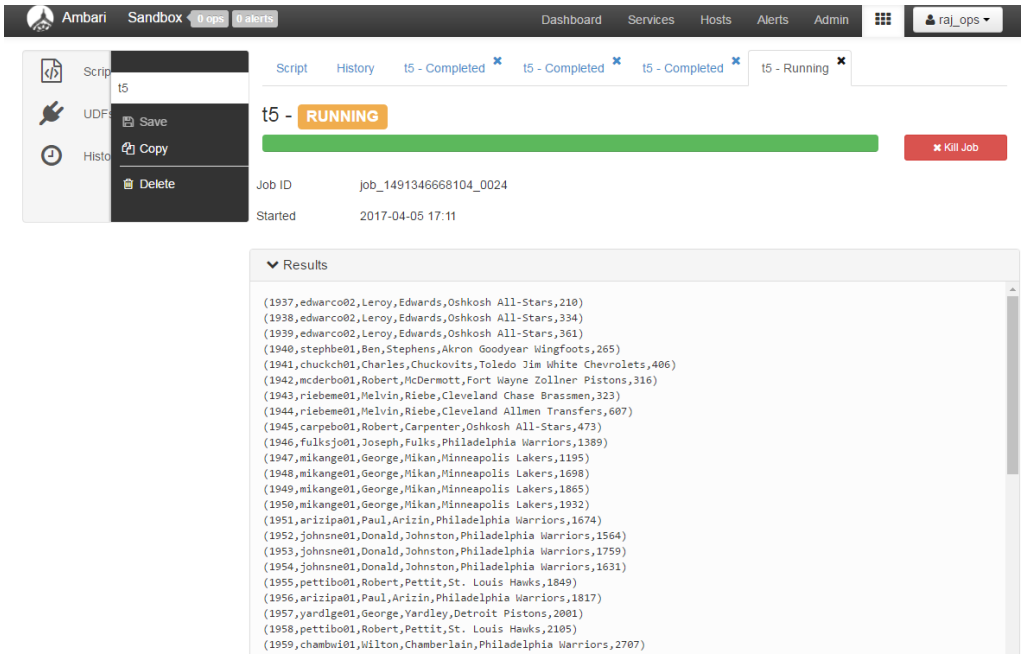


Figure 4: The players and teams

### Task 8. Code Recap

- We used "Pig script" to upload our data from three files and created 3 relations. We filtered the first row (that included column names) according to the type of the fields in each relation.
- Then we pulled out the required fields from each row in all the three relations using FOREACH.
- We then grouped players by year with one statement, GROUP.
- Then we found the maximum points for each year.
- This was then mapped to the players (using a join) so that we could access player and team IDs.
- Finally, we joined three relations to get the other details, i.e. player names and team names.

**References** If you need more practice or more information, use the following links:

- `http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-pig/`
- `http://hortonworks.com/hadoop-tutorial/hello-world-an-introduction-to-hadoop-hcatalog-hive-and-pig/#section_5`
- `http://hortonworks.com/hadoop-tutorial/how-to-use-basic-pig-commands/`
- `https://pig.apache.org/docs/r0.12.0/basic.html#store`