

Hortonworks Tutorial 3

Working with Hive II

In this tutorial we learn how to perform more advanced data processing with Hive.

Task 1. Data Analysis

Apache Hive provides a data warehouse function to the Hadoop cluster. Through the use of "HiveQL" you can view your data as a table and create queries just as you would in a database.

1. Use the Content Assist to build a query:

- Create a new SQL Worksheet.
- Start typing in the SELECT SQL command, but only enter the first two letters: "SE"
- Press "Ctrl+Space" to view the following content assist pop-up dialog window:

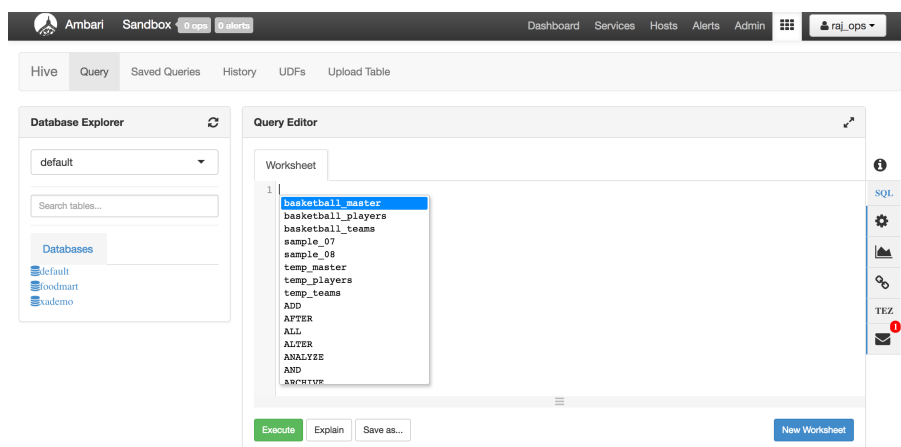


Figure 1: Pop-up dialog window in Hive View

- Notice content assist shows you some options that start with an "SE". Type in the following query to find out the number of players in each team. Execute the code.

```
SELECT tmid, COUNT (DISTINCT playerID) as count1 FROM  
basketball_players GROUP BY tmid;
```

2. Explore Explain Features of the Hive Query Editor:

- Explore the various explain features to better understand the execution of a query: "Text Explain", "Visual Explain", and "Tez Explain". Add the "Explain" command at the beginning of the query or click on the "Explain" button. The output displays the flow of the resulting job (see Fig. 2):
- To see the "Visual Explain" click on the "Visual Explain" icon on the right tabs (the button above the "TEZ"). This is a much more readable summary of the explain plan (see Fig. 3):

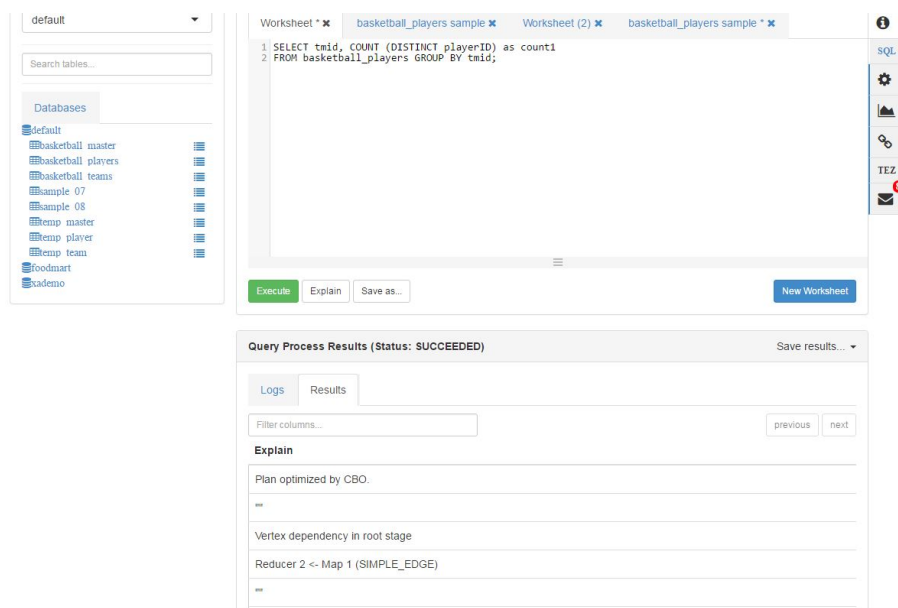


Figure 2: The Explain output in Hive View

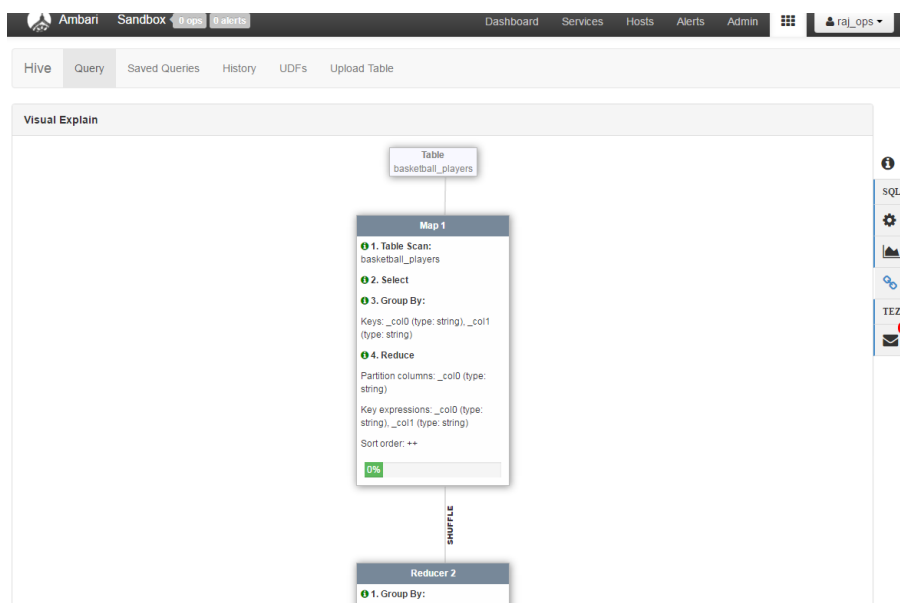


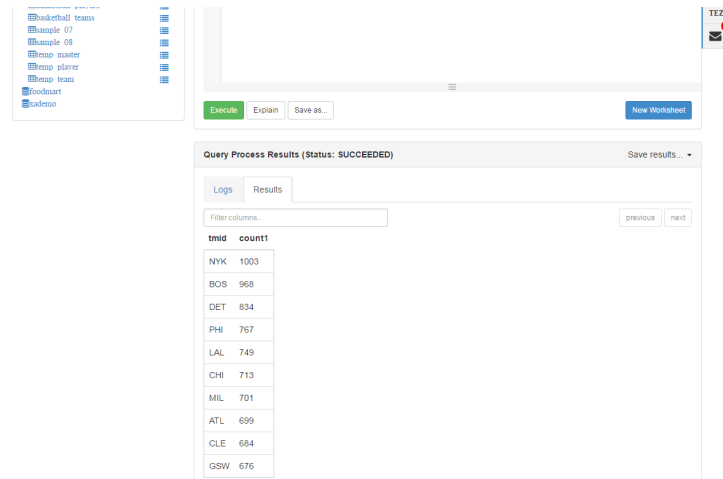
Figure 3: A Visual Explain sample in Hive View

3. Query the table using "COUNT", "GROUP BY" and "ORDER BY":

- You can query tables in Hive using the same way you do with other database queries. In the previous query, we found out the number of players per team. Now let's find top-10 teams with the highest number of players using ORDER BY to sort the results in a descending order and limit the results to the top 10.

```
SELECT tmid, COUNT(playerID) as count1 FROM basketball_players
GROUP BY tmid
ORDER BY count1 desc limit 10;
```

- Press Execute and the following results will be shown (see Fig. 4):



The screenshot shows a web-based query execution interface. On the left, there is a sidebar with a list of tables: 'basketball_teams', 'sample_07', 'sample_08', 'temp_player', 'temp_teams', 'foodmart', and 'sakdemo'. The main area displays the 'Query Process Results (Status: SUCCEEDED)' for a query. The results are shown in a table with two columns: 'tmid' and 'count1'. The table lists 12 teams and their corresponding counts, ordered by team ID. The interface includes buttons for 'Execute', 'Explain', 'Save as...', and 'New Worksheet'. There are also tabs for 'Logs' and 'Results', and a 'Filter columns...' input field. Navigation buttons 'previous' and 'next' are visible on the right side of the results table.

tmid	count1
NYK	1003
BOS	968
DET	834
PHI	767
LAL	749
CHI	713
MIL	701
ATL	699
CLE	684
GSW	676

Figure 4: A query result using Count, GroupBy, and Orderby

4. Query tables using "JOIN ON":

- You can make a join with other tables in Hive using the same way you do with other database queries.
- Let's find out the details of players (their full names, and how many points they won per year) by making a join "basketball_master" and "basketball_players" tables. Enter the following code into the query editor and execute it:

```
SELECT a.year, a.playerID, b.fname, b.lname, a.points
FROM basketball_players a
JOIN basketball_master b ON (a.playerID = b.bioID);
```

- You can see the job running in the log (see Fig. 5). When the job completes, you can see the results.

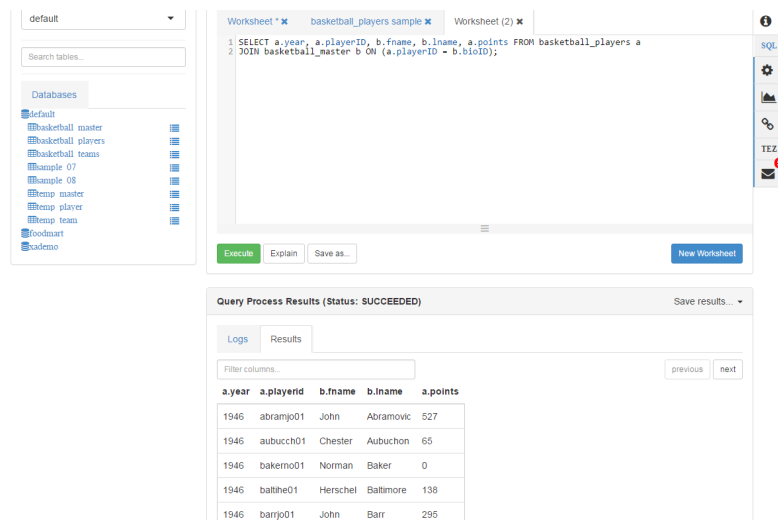


Figure 5: A query result using Count, GroupBy, and Orderby

Task 2. Data Processing Task

In this task, we will find out the player for each year who had the highest points. The query should include the following details: the player first name, last name, points and the team name (some players might change teams over time). This complex query will include the join between 3 tables. We build the query step by step.

1. **Create query to get the highest points for each year:** Group the data by year so we can find the highest score for each year. This query first groups all the records by year and then selects the player with the highest points from each year:

```
SELECT year, max(points) FROM basketball_players GROUP BY year;
```

The results of the query look like Fig. 6.

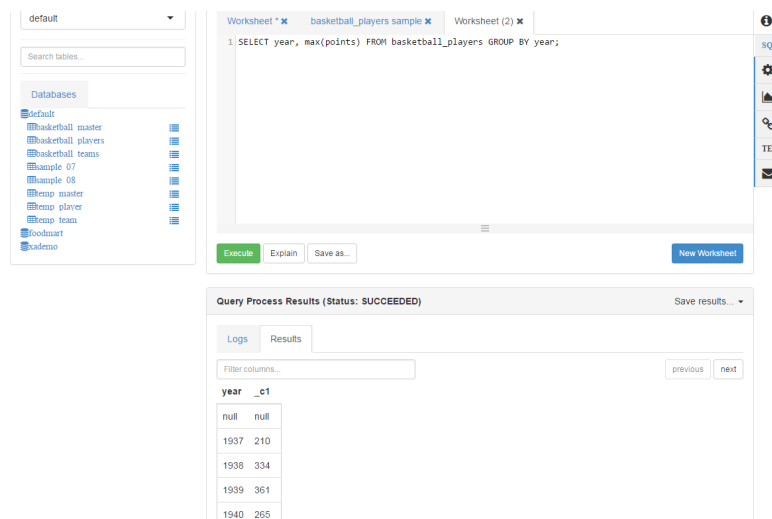


Figure 6: Find the highest points for each year

2. **Create query to filter playerID:** Go back and get the playerID(s) so we know who the

player(s) was. We know that for a given year we can use the points to find the player(s) for that year. So we can take the previous query and join it with the "basketball_players" records to get the interim table:

```
SELECT a.year, a.playerID, a.points from basketball_players a
JOIN (SELECT year, max(points) points FROM basketball_players
GROUP BY year) b
ON (a.year = b.year AND a.points = b.points);
```

The results of the query look like Fig. 7.

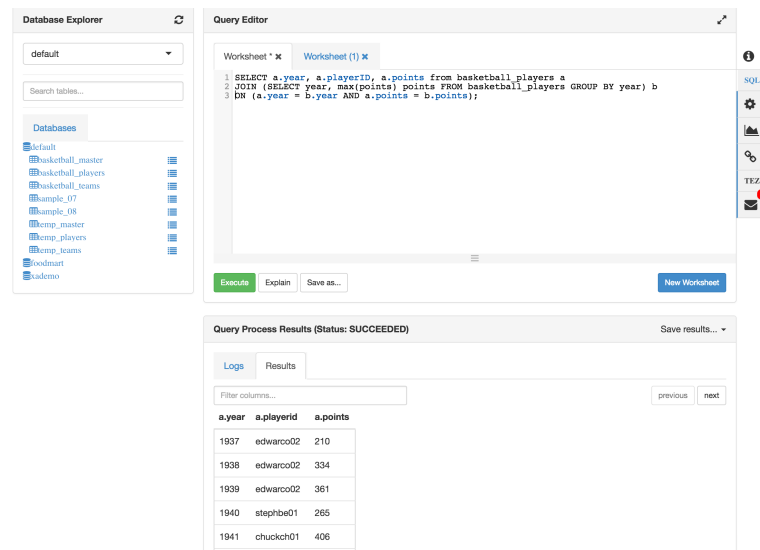


Figure 7: Find the highest points with playerIDs for each year

- Create Query to Get the Player Name:** Now that we have the playerID(s) for each year with the highest points. We can use this information to obtain the full name of the player(s) by performing a join with the basketball_master that has data fields "fname" and "lname".

```
SELECT a.year, a.playerID, c.fname, c.lname, a.points
FROM basketball_players a
JOIN (SELECT year, max(points) points FROM basketball_players
GROUP BY year) b
ON (a.year = b.year AND a.points = b.points) JOIN basketball_master c
ON (c.bioID = a.playerID);
```

The results of the query look like Fig. 8.

The screenshot shows the Hive Query Editor interface. On the left is the 'Database Explorer' with a tree view of databases including 'default', 'basketball_master', 'basketball_players', 'basketball_teams', 'sample_07', 'sample_08', 'temp_master', 'temp_players', 'temp_teams', 'foodmart', and 'xademo'. The 'Query Editor' on the right contains a SQL query in 'Worksheet (1)'. The query is:

```
1 SELECT a.year, a.playerID, c.fname, c.lname, a.points FROM basketball_players a
2 JOIN (SELECT year, max(points) points FROM basketball_players GROUP BY year) b
3 ON (a.year = b.year AND a.points = b.points) JOIN basketball_master c
4 ON (c.bioID = a.playerID);
```

Below the query editor, the 'Query Process Results (Status: SUCCEEDED)' section shows the 'Results' tab. It displays a table with the following data:

a.year	a.playerid	c.fname	c.lname	a.points
1937	edwarco02	Leroy	Edwards	210
1938	edwarco02	Leroy	Edwards	334
1939	edwarco02	Leroy	Edwards	361

Figure 8: Find the highest points with the player names for each year

Task 3. Exercise: Create Query to Get the Team Name

From the previous steps, we got the player's details with the highest points for each year. As an exercise, in addition to these details, perform another join with `basketball_teams` to get the team name as well.

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

- http://hortonworks.com/hadoop-tutorial/hello-world-an-introduction-to-hadoop-hcatalog-hive-and-pig/#section_4
- <http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-hive/>