

Assignment 2

JAYESH PARAB & GAGANDEEP SINGH BHUTANI

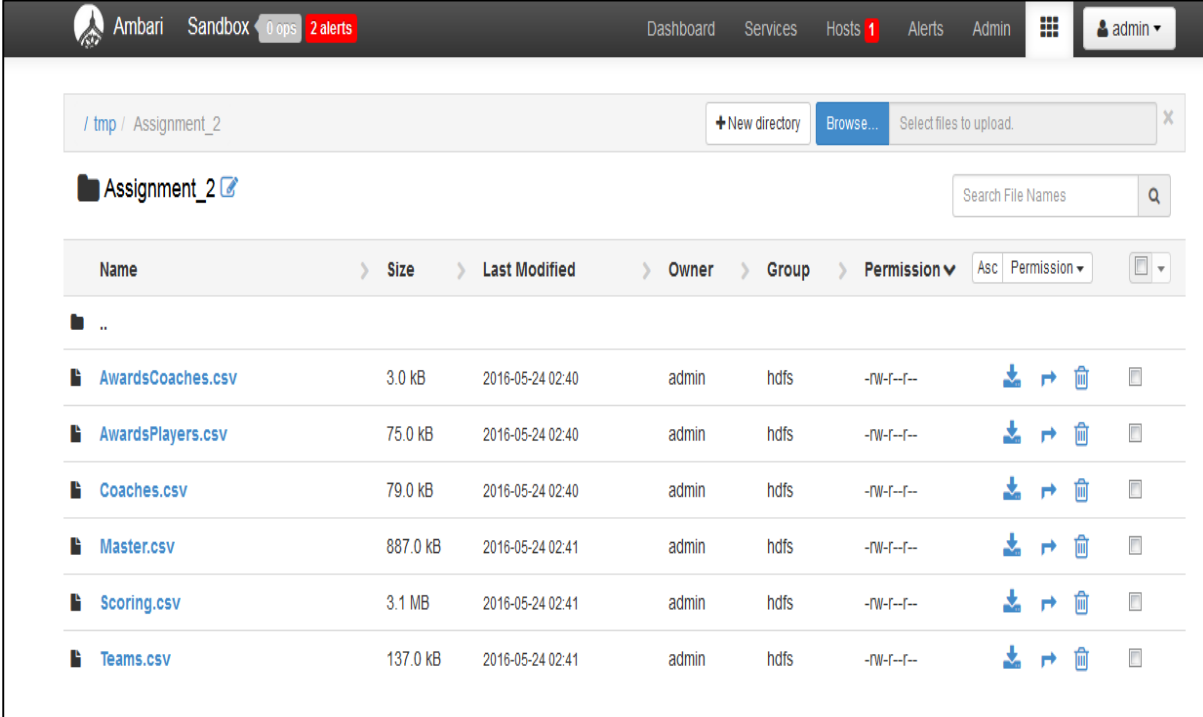
Table Of Content

Task 1 Creating Table	3
CSV Files uploaded to HDFS Files	3
Hive Tables	3
Create Masters Table	3
Create AwardsCoaches table	7
Create Award Player	10
Create Coaches Table	12
Create Scoring Table	15
Create Team Table	18
CSV Files Uploaded to HDFS Files	21
Pig Tables	22
Master table	22
Scoring Table	22
Team Table	23
Coach Table	23
AwardCoach Table	24
AwardPlayer Table	24
Hive and Pig Queries	25
Task 2	25
Hive Query 2A	25
Pig Query 2A	28
Comparison Table for 2A	29
Hive Query 2B	30
Pig Query 2B	32
Comparison table for 2B	34
Hive Query 2 C	35
Pig Query 2c	38
Comparison Table for 2c	41
Task 3	42
Hive Query 3 A	42
Pig Query 3A	44
Comparison table for 3A	45
Hive Query 3B	46
L	47
Pig Query 3B	48

Comparison Table for 3B.....	49
Hive Query 3C	50
Pig Query 3C.....	53
Comparison table for 3C	54
Task 4	55
Hive Query 4A	55
Pig Query 4A.....	58
Comparison Table for 4a.....	59
Hive Query 4 B	60
Pig Query 4B.....	64
Comparison Table For 4B.....	66
Task 5:	67
Big Data Report.....	79
Introduction	79
Current state of the art	79
Big Spatial Data Mining	79
Framework for categorizing and applying privacy and preservation techniques in Data Mining	80
Big Data is social media using data mining techniques	82
Reducing the search space for big data mining for interesting patterns from uncertain data	83
Conclusion.....	84
References	85

1 Task 1 Creating Table

1.1 CSV Files uploaded to HDFS Files

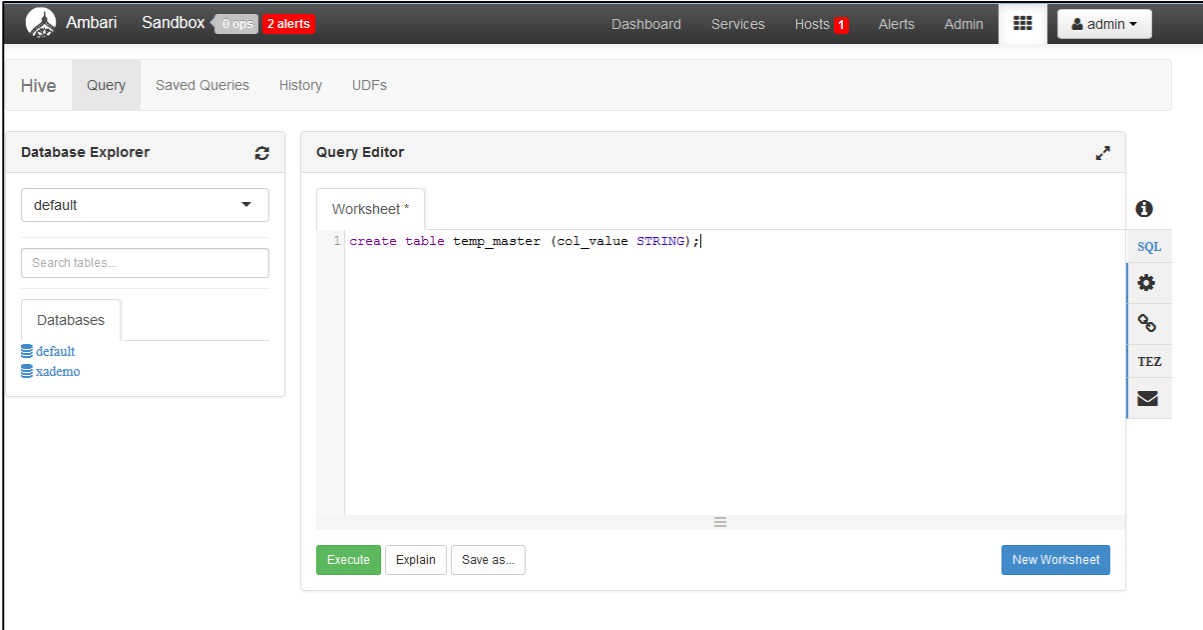


The screenshot shows the Ambari Sandbox interface. The top navigation bar includes 'Ambari', 'Sandbox', '0 ops', '2 alerts', 'Dashboard', 'Services', 'Hosts 1', 'Alerts', 'Admin', and a user profile 'admin'. The main content area displays the HDFS file system path '/tmp/Assignment_2'. A search bar for file names is present. Below the path, a table lists the files in the directory:

Name	Size	Last Modified	Owner	Group	Permission	Actions
..						
AwardsCoaches.csv	3.0 kB	2016-05-24 02:40	admin	hdfs	-rw-r--r--	Download, Upload, Delete, Copy
AwardsPlayers.csv	75.0 kB	2016-05-24 02:40	admin	hdfs	-rw-r--r--	Download, Upload, Delete, Copy
Coaches.csv	79.0 kB	2016-05-24 02:40	admin	hdfs	-rw-r--r--	Download, Upload, Delete, Copy
Master.csv	887.0 kB	2016-05-24 02:41	admin	hdfs	-rw-r--r--	Download, Upload, Delete, Copy
Scoring.csv	3.1 MB	2016-05-24 02:41	admin	hdfs	-rw-r--r--	Download, Upload, Delete, Copy
Teams.csv	137.0 kB	2016-05-24 02:41	admin	hdfs	-rw-r--r--	Download, Upload, Delete, Copy

1.2 Hive Tables

1.2.1 Create Masters Table



The screenshot shows the Ambari Sandbox interface with the 'Hive' tab selected. The 'Query Editor' is active, displaying a SQL statement in a worksheet:

```
1 create table temp_master (col_value STRING);
```

Below the query editor, there are buttons for 'Execute', 'Explain', 'Save as...', and 'New Worksheet'. On the left, the 'Database Explorer' shows a list of databases: 'default' and 'xademo'. On the right, there are icons for 'SQL', 'Settings', 'Refresh', 'TEZ', and 'Email'.

The screenshot shows the Ambari Query Editor interface. The top navigation bar includes 'Dashboard', 'Services', 'Hosts' (with 1 alert), 'Alerts', and 'Admin'. The left sidebar shows 'Database Explorer' with a dropdown set to 'default' and a search bar. Below the search bar, a list of databases is shown: 'default', 'temp_master', and 'xademo'. The main 'Query Editor' area has a 'Worksheet' tab with a single SQL query: `1 LOAD DATA INPATH '/tmp/Assignment_2/Master.csv' OVERWRITE INTO TABLE temp_master;`. Below the query editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. The 'Query Process Results' section shows a status of 'Succeeded' and a 'Save results...' dropdown. The 'Results' tab is active, displaying two log messages: 'INFO : Loading data to table default.temp_master from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/Master.csv' and 'INFO : Table default.temp_master stats: [numFiles=1, numRows=0, totalSize=908288, rawDataSize=0]'. A right-hand sidebar contains icons for 'SQL', 'Settings', 'Visuals', 'Connections', 'TEZ' (with 1 alert), and 'Email'.

The screenshot shows the Ambari Query Editor interface. The top navigation bar includes 'Dashboard', 'Services', 'Hosts' (with 1 alert), 'Alerts', and 'Admin'. The left sidebar shows 'Database Explorer' with a dropdown set to 'default' and a search bar. Below the search bar, a list of databases is shown: 'default', 'hockey_master', 'temp_master', and 'xademo'. The main 'Query Editor' area has a 'Worksheet' tab with a single SQL query: `1 create table hockey_master (playerID STRING, coachID STRING, hofID STRING, firstName STRING, 2 lastName STRING, nameNote STRING, nameGiven STRING, nameNick STRING, 3 height INT, weight INT, shootCatch STRING, legendsID INT, 4 iHdbID INT, hrefID STRING, firstNHL INT, lastNHL INT, 5 firstWHA INT, lastWHA INT, pos STRING, 6 birthYear INT, birthMon INT, birthDay INT, birthCountry STRING, 7 birthState STRING, birthCity STRING, deathYear INT, deathMon INT, 8 deathDay INT, deathCountry STRING, deathState STRING, 9 deathCity STRING);`. Below the query editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. The 'Query Process Results' section shows a status of 'Succeeded' and a 'Save results...' dropdown. The 'Results' tab is active, displaying a log message: 'INFO : Loading data to table default.hockey_master from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/Master.csv'. A right-hand sidebar contains icons for 'SQL', 'Settings', 'Visuals', 'Connections', 'TEZ' (with 6 alerts), and 'Email'.

The screenshot shows the Ambari web interface. At the top, there's a navigation bar with 'Ambari', 'Sandbox', '0 ops', '2 alerts', and tabs for 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. Below this is a secondary navigation bar with 'Hive', 'Query', 'Saved Queries', 'History', and 'UDFs'. The main content area is divided into two panels. The left panel, 'Database Explorer', shows a tree view of databases: 'default', 'hockey_master', 'temp_master', and 'xademo'. The right panel, 'Query Editor', contains a 'Worksheet' with a Hive SQL query. The query is an 'insert overwrite' statement that selects data from 'temp_master' and inserts it into 'hockey_master'. The select statement uses a series of 'regexp_extract' functions to parse a JSON-like string into individual fields. Below the query editor, there are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. At the bottom, a 'Query Process Results' section shows the status 'Succeeded' and a 'Save results...' dropdown. Below this are tabs for 'Logs' and 'Results'.

Query Editor

Worksheet *

```

1 insert overwrite table hockey_master
2 SELECT
3   regexp_extract(col_value, '^(?:[^\,]*\,)?(1)$', 1) playerID,
4   regexp_extract(col_value, '^(?:[^\,]*\,)?(2)$', 1) coachID,
5   regexp_extract(col_value, '^(?:[^\,]*\,)?(3)$', 1) hofID,
6   regexp_extract(col_value, '^(?:[^\,]*\,)?(4)$', 1) firstName,
7   regexp_extract(col_value, '^(?:[^\,]*\,)?(5)$', 1) lastName,
8   regexp_extract(col_value, '^(?:[^\,]*\,)?(6)$', 1) nameNote,
9   regexp_extract(col_value, '^(?:[^\,]*\,)?(7)$', 1) nameGiven,
10  regexp_extract(col_value, '^(?:[^\,]*\,)?(8)$', 1) nameNick,
11  regexp_extract(col_value, '^(?:[^\,]*\,)?(9)$', 1) height,
12  regexp_extract(col_value, '^(?:[^\,]*\,)?(10)$', 1) weight,
13  regexp_extract(col_value, '^(?:[^\,]*\,)?(11)$', 1) shootCatch,
14  regexp_extract(col_value, '^(?:[^\,]*\,)?(12)$', 1) legendsID,
15  regexp_extract(col_value, '^(?:[^\,]*\,)?(13)$', 1) ihdbID,
16  regexp_extract(col_value, '^(?:[^\,]*\,)?(14)$', 1) hrefID,
17  regexp_extract(col_value, '^(?:[^\,]*\,)?(15)$', 1) firstNHL,
18  regexp_extract(col_value, '^(?:[^\,]*\,)?(16)$', 1) lastNHL,
19  regexp_extract(col_value, '^(?:[^\,]*\,)?(17)$', 1) firstWHA,
20  regexp_extract(col_value, '^(?:[^\,]*\,)?(18)$', 1) lastWHA,
21  regexp_extract(col_value, '^(?:[^\,]*\,)?(19)$', 1) pos,
22  regexp_extract(col_value, '^(?:[^\,]*\,)?(20)$', 1) birthYear,
23  regexp_extract(col_value, '^(?:[^\,]*\,)?(21)$', 1) birthMon,
24  regexp_extract(col_value, '^(?:[^\,]*\,)?(22)$', 1) birthDay,
25  regexp_extract(col_value, '^(?:[^\,]*\,)?(23)$', 1) birthCountry,
26  regexp_extract(col_value, '^(?:[^\,]*\,)?(24)$', 1) birthState,
27  regexp_extract(col_value, '^(?:[^\,]*\,)?(25)$', 1) birthCity,
28  regexp_extract(col_value, '^(?:[^\,]*\,)?(26)$', 1) deathYear,
29  regexp_extract(col_value, '^(?:[^\,]*\,)?(27)$', 1) deathMon,
30  regexp_extract(col_value, '^(?:[^\,]*\,)?(28)$', 1) deathDay,
31  regexp_extract(col_value, '^(?:[^\,]*\,)?(29)$', 1) deathCountry,
32  regexp_extract(col_value, '^(?:[^\,]*\,)?(30)$', 1) deathState,
33  regexp_extract(col_value, '^(?:[^\,]*\,)?(31)$', 1) deathCity
34 from temp_master;

```

Query Process Results (Status: Succeeded) Save results...

Logs Results

The screenshot shows the 'Query Process Results' page for a successful query execution. The status is 'SUCCEEDED'. There are tabs for 'Logs' and 'Results'. The 'Logs' tab is selected, showing a series of informational messages from the Tez session and the Hive execution. The messages indicate that the session was already open, was closed and reopened, and then re-established. It also shows the status 'Running' and the execution details on the YARN cluster. Finally, it reports the loading of data to the 'hockey_master' table and the final statistics for the table.

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

INFO : Session is already open
 INFO : Tez session was closed. Reopening...
 INFO : Session re-established.
 INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464020964330_0002)

INFO : Map 1: -/-
 INFO : Map 1: 0/1
 INFO : Map 1: 0(+1)/1
 INFO : Map 1: 1/1
 INFO : Loading data to table default.hockey_master from hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/hockey_master/hive-staging_hive_2016-05-23_17-06-23_364_5602784475350367082-1/-ext-10000
 INFO : Table default.hockey_master stats: [numFiles=1, numRows=7770, totalSize=983699, rawDataSize=975929]

Ambari Sandbox 0 ops 2 alerts

Dashboard Services Hosts 1 Alerts Admin

admin

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_master
- temp_master
- xademo

Query Editor

Worksheet x hockey_master sample x

```
1 SELECT * FROM hockey_master LIMIT 100;
```

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded) Save results...

Logs Results

Filter columns... previous next

hockey_master.playerid	hockey_master.coachid	hockey_master.hofid	hockey_master.firstname	hockey_ma
playerID	coachID	hofID	firstName	lastName
aaitoan01			Antti	Aaito
abbeybr01			Bruce	Abbey
abbotge01			George	Abbott
abbotre01			Reg	Abbott
abdelju01			Justin	Abdelkader
abelci01			Clarence	Abel
abelge01			Gerry	Abel
abelsi01	abelsi01c	abelsi01h	Sid	Abel
abgrade01			Dennis	Abgrail
abidra01			Ramzi	Abid

1.2.2 Create AwardsCoaches table

The screenshot shows the Ambari web interface. The top navigation bar includes 'Ambari', 'Sandbox', '0 ops', '2 alerts', and links to 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The user is logged in as 'admin'. The main interface has tabs for 'Hive', 'Query', 'Saved Queries', 'History', and 'UDFs'. On the left, the 'Database Explorer' shows a tree of databases: 'default', 'hockey_master', 'temp_master', and 'xademo'. The 'Query Editor' is active, displaying a 'Worksheet' with the following SQL query:

```
1 create table temp_awardsCoaches (col_value STRING);
```

 Below the query editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. The 'Query Process Results' section shows a status of 'Succeeded'. It has tabs for 'Logs' and 'Results', a 'Filter columns...' input, and 'previous' and 'next' navigation buttons. On the right side of the Query Editor, there is a vertical toolbar with icons for 'SQL', 'Settings', 'Visualize', 'Refresh', 'TEZ', and 'Alerts'.

This screenshot shows the Ambari web interface after the table has been created. The 'Query Editor' now contains the following SQL query:

```
1 LOAD DATA INPATH '/tmp/Assignment_2/AwardsCoaches.csv' OVERWRITE INTO TABLE temp_awardscoaches;
```

 The 'Query Process Results' section still shows a status of 'Succeeded'. The rest of the interface, including the 'Database Explorer' and top navigation bar, remains the same as in the previous screenshot.

The screenshot shows the Ambari Query Editor interface. The top navigation bar includes 'Ambari', 'Sandbox', '0 ops', '2 alerts', 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user profile 'admin'. The main interface has tabs for 'Hive', 'Query', 'Saved Queries', 'History', and 'UDFs'. On the left, the 'Database Explorer' shows a 'default' database with tables 'hockey_master', 'temp_master', and 'xademo'. The 'Query Editor' contains a 'Worksheet' with the following SQL code:

```
1 create table hockey_award_coaches (coachID STRING, award STRING, year INT, lgID STRING,
2 );
```

Below the editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. The 'Query Process Results' section shows 'Status: Succeeded' and options for 'Logs' and 'Results'.

The screenshot shows the Ambari Query Editor interface with the same top navigation bar. The 'Database Explorer' on the left now includes 'hockey_award_coaches' and 'temp_awardscoaches'. The 'Query Editor' 'Worksheet' contains the following SQL code:

```
1 insert overwrite table hockey_award_coaches
2 SELECT
3 regexp_extract(col_value, '^(?:([^\,]*)\,)?(1)', 1) coachID,
4 regexp_extract(col_value, '^(?:([^\,]*)\,)?(2)', 1) award,
5 regexp_extract(col_value, '^(?:([^\,]*)\,)?(3)', 1) year,
6 regexp_extract(col_value, '^(?:([^\,]*)\,)?(4)', 1) lgID,
7 regexp_extract(col_value, '^(?:([^\,]*)\,)?(5)', 1) note
8 from temp_awardscoaches;
9
```

Below the editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'. A green progress bar indicates '100%' completion. The 'Query Process Results' section shows 'Status: SUCCEEDED' and options for 'Logs' and 'Results'.

Query Process Results (Status: SUCCEEDED)Save results... ▾

LogsResults

INFO : Tez session hasn't been created yet. Opening session

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464020964330_0003)

INFO : Map 1: -/-

INFO : Map 1: 0/1

INFO : Map 1: 0(+1)/1

INFO : Map 1: 1/1

INFO : Loading data to table default.hockey_award_coaches from hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/hockey_award_coaches/.hive-staging_hive_2016-05-23_17-13-51_529_7810236180340934160-1/-ext-10000

INFO : Table default.hockey_award_coaches stats: [numFiles=1, numRows=88, totalSize=3073, rawDataSize=2985]

Ambari

Sandbox 0 ops 2 alerts

Dashboard Services Hosts 1 Alerts Admin

admin ▾

HiveQuerySaved QueriesHistoryUDFs

Database Explorer

default ▾

Search tables...

Databases

default

hockey_award_coaches

hockey_master

temp_awardscoaches

temp_master

xademo

Query Editor

Worksheet * x hockey_award_coaches sample x

1 SELECT * FROM hockey_award_coaches LIMIT 100;

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded)Save results... ▾

LogsResults

Filter columns...

previousnext

hockey_award_coaches.coachid	hockey_award_coaches.award	hockey_award_coaches.year	hockey_award
coachID	award		lgID
patrile01c	First Team All-Star	1930	NHL
irvindi01c	Second Team All-Star	1930	NHL
patrile01c	First Team All-Star	1931	NHL
irvindi01c	Second Team All-Star	1931	NHL
patrile01c	First Team All-Star	1932	NHL
irvindi01c	Second Team All-Star	1932	NHL
patrile01c	First Team All-Star	1933	NHL
irvindi01c	Second Team All-Star	1933	NHL
patrile01c	First Team All-Star	1934	NHL
irvindi01c	Second Team All-Star	1934	NHL

9

1.2.3 Create Award Player

The screenshot shows the Ambari web interface. At the top, there's a navigation bar with 'Dashboard', 'Services', 'Hosts' (with a red alert icon), 'Alerts', and 'Admin'. The user is logged in as 'admin'. Below this is a tabbed interface with 'Hive', 'Query', 'Saved Queries', 'History', and 'UDFs'. The 'Query' tab is active.

On the left is the 'Database Explorer' panel. It shows a tree view of databases: 'default' (selected), 'hockey_award_coaches', 'hockey_master', 'temp_awardscoaches', 'temp_master', and 'xademo'. A search bar is present above the list.

The main area is the 'Query Editor'. It has a 'Worksheet' tab. The SQL query entered is: `1 create table temp_awardsPlayers (col_value STRING);`. Below the editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'.

Below the editor is the 'Query Process Results (Status: Succeeded)' section. It has tabs for 'Logs' and 'Results'. The 'Results' tab is active, but it's empty.

This screenshot shows the same Ambari interface, but the SQL query in the 'Query Editor' is now: `1 LOAD DATA INPATH '/tmp/Assignment_2/AwardsPlayers.csv' OVERWRITE INTO TABLE temp_awardeplayers;`. The 'Execute' button is highlighted in green.

The 'Query Process Results (Status: Succeeded)' section now shows data in the 'Results' tab. The output is as follows:

```
INFO : Loading data to table default.temp_awardsplayers from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/AwardsPlayers.csv
INFO : Table default.temp_awardsplayers stats: [numFiles=1, numRows=0, totalSize=76800, rawDataSize=0]
```

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_master
- temp_awardscoaches
- temp_master
- xademo

Query Editor

Worksheet

```

1 create table hockey_awards_players (playerID STRING, award STRING, year INT, lgID STRING,
2 note STRING, pos STRING
3 );
4

```

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded) Save results...

Logs Results

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_master
- temp_awardscoaches
- temp_awardsplayers
- temp_master
- xademo

Query Editor

Worksheet

```

1 insert overwrite table hockey_awards_players
2 SELECT
3 regexp_extract(col_value, '^(?:([*],*)\\,?) {1}', 1) playerID,
4 regexp_extract(col_value, '^(?:([*],*)\\,?) {2}', 1) award,
5 regexp_extract(col_value, '^(?:([*],*)\\,?) {3}', 1) year,
6 regexp_extract(col_value, '^(?:([*],*)\\,?) {4}', 1) lgID,
7 regexp_extract(col_value, '^(?:([*],*)\\,?) {5}', 1) note,
8 regexp_extract(col_value, '^(?:([*],*)\\,?) {6}', 1) pos
9 from temp_awardsplayers;

```

Execute Explain Save as... Kill Session New Worksheet

100%

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

INFO : Tez session hasn't been created yet. Opening session
INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464020964330_0004)

INFO : Map 1: -/-
INFO : Map 1: 0/1
INFO : Map 1: 0(+1)/1
INFO : Map 1: 1/1
INFO : Loading data to table default.hockey_awards_players from hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/hockey_awards_players/hive-staging_hive_2016-05-23_17-23-17_672_4973804989374124044-1/-ext-10000
INFO : Table default.hockey_awards_players stats: [numFiles=1, numRows=2093, totalSize=76810, rawDataSize=74717]

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_master
- temp_awardscoaches
- temp_awardsplayers
- temp_master
- xademo

Query Editor

Worksheet hockey_awards_players sample

```
1 SELECT * FROM hockey_awards_players LIMIT 100;
```

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded) Save results...

Logs Results

Filter columns... previous next

hockey_awards_players.playerid	hockey_awards_players.award	hockey_awards_players.year	hockey_award
playerID	award		lgID
malonjo01	Art Ross	1917	NHL
cleghod01	Art Ross	1918	NHL
malonjo01	Art Ross	1919	NHL
lalonne01	Art Ross	1920	NHL
broadpu01	Art Ross	1921	NHL
dyeba01	Art Ross	1922	NHL
dennecy01	Art Ross	1923	NHL

1.2.4 Create Coaches Table

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_master
- temp_awardscoaches
- temp_awardsplayers
- temp_master
- xademo

Query Editor

Worksheet

```
1 create table temp_Coaches (col_value STRING);
```

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded) Save results...

Logs Results

Filter columns... previous next

The screenshot displays the Hive Query Editor interface. On the left, the 'Database Explorer' shows a tree view of databases including 'default', 'hockey_award_coaches', 'hockey_awards_players', 'hockey_master', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_master', and 'xademo'. The 'Query Editor' on the right contains a single query: `1 LOAD DATA INPATH '/tmp/Assignment_2/Coaches.csv' OVERWRITE INTO TABLE temp_coaches;`. Below the query editor, the 'Query Process Results (Status: Succeeded)' section shows the execution log:
INFO : Loading data to table default.temp_coaches from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/Coaches.csv
INFO : Table default.temp_coaches stats: [numFiles=1, numRows=0, totalSize=80896, rawDataSize=0]

The screenshot displays the Hive Query Editor interface. On the left, the 'Database Explorer' shows a tree view of databases including 'default', 'hockey_award_coaches', 'hockey_awards_players', 'hockey_master', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_master', and 'xademo'. The 'Query Editor' on the right contains a query to create a table: `1 create table hockey_coaches (coachID STRING, year INT, tmID STRING, lgID STRING,
2 stint INT, notes STRING,
3 g INT, w INT, l INT, t INT, postg INT, postw INT,
4 postl INT, postt INT
5);`. Below the query editor, the 'Query Process Results (Status: Succeeded)' section shows the execution log:
INFO : Loading data to table default.hockey_coaches from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/Coaches.csv
INFO : Table default.hockey_coaches stats: [numFiles=1, numRows=0, totalSize=80896, rawDataSize=0]

Hive

Query

Saved Queries

History

UDFs

Database Explorer

default

Search tables...

Databases

default

hockey_award_coaches

hockey_awards_players

hockey_master

temp_awardscoaches

temp_awardsplayers

temp_master

xademo

Query Editor

Worksheet

```

1 insert overwrite table hockey_coaches
2 SELECT
3 regexp_extract(col_value, '^([^\,]*)\\.(?){1}', 1) coachID,
4 regexp_extract(col_value, '^([^\,]*)\\.(?){2}', 1) year,
5 regexp_extract(col_value, '^([^\,]*)\\.(?){3}', 1) tmID,
6 regexp_extract(col_value, '^([^\,]*)\\.(?){4}', 1) lgID,
7 regexp_extract(col_value, '^([^\,]*)\\.(?){5}', 1) stint,
8 regexp_extract(col_value, '^([^\,]*)\\.(?){6}', 1) notes,
9 regexp_extract(col_value, '^([^\,]*)\\.(?){7}', 1) g,
10 regexp_extract(col_value, '^([^\,]*)\\.(?){8}', 1) w,
11 regexp_extract(col_value, '^([^\,]*)\\.(?){9}', 1) l,
12 regexp_extract(col_value, '^([^\,]*)\\.(?){10}', 1) t,
13 regexp_extract(col_value, '^([^\,]*)\\.(?){11}', 1) postg,
14 regexp_extract(col_value, '^([^\,]*)\\.(?){12}', 1) postw,
15 regexp_extract(col_value, '^([^\,]*)\\.(?){13}', 1) postl,
16 regexp_extract(col_value, '^([^\,]*)\\.(?){14}', 1) postt
17 from temp_coaches;

```

Execute

Explain

Save as...

Kill Session

New Worksheet

100%

Query Process Results (Status: SUCCEEDED)

Save results...

Logs

Results

INFO : Session is already open

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464020964330_0004)

INFO : Map 1: 0/1

INFO : Map 1: 0(+1)/1

INFO : Map 1: 1/1

INFO : Loading data to table default.hockey_coaches from hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/hockey_coaches/hive-staging_hive_2016-05-23_17-27-21_450_4568274584518999282-1/-ext-10000

INFO : Table default.hockey_coaches stats: [numFiles=1, numRows=1824, totalSize=88058, rawDataSize=86234]

The screenshot shows the Query Editor interface with the following components:

- Database Explorer:** A sidebar on the left showing a list of databases including 'default', 'hockey_award_coaches', 'hockey_awards_players', 'hockey_coaches', 'hockey_master', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_coaches', 'temp_master', and 'xademo'.
- Query Editor:** The main workspace containing a worksheet with the SQL query: `1 SELECT * FROM hockey_coaches LIMIT 100;`. Below the editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'.
- Query Process Results (Status: Succeeded):** A section below the query editor showing the results of the query. It includes a 'Logs' tab and a 'Results' tab. The 'Results' tab displays a table with the following columns: 'hockey_coaches.coachid', 'hockey_coaches.year', 'hockey_coaches.tmid', 'hockey_coaches.lgid', and 'hockey_coach'. The table contains 8 rows of data.

hockey_coaches.coachid	hockey_coaches.year	hockey_coaches.tmid	hockey_coaches.lgid	hockey_coach
coachID		tmID	lgID	
abelsi01c	1952	CHI	NHL	1
abelsi01c	1953	CHI	NHL	1
abelsi01c	1957	DET	NHL	2
abelsi01c	1958	DET	NHL	1
abelsi01c	1959	DET	NHL	1
abelsi01c	1960	DET	NHL	1
abelsi01c	1961	DET	NHL	1

1.2.5 Create Scoring Table

The screenshot shows the Query Editor interface with the following components:

- Database Explorer:** A sidebar on the left showing a list of databases including 'default', 'hockey_award_coaches', 'hockey_awards_players', 'hockey_coaches', 'hockey_master', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_coaches', 'temp_master', and 'xademo'.
- Query Editor:** The main workspace containing a worksheet with the SQL query: `1 create table temp_Scoring (col_value STRING);`. Below the editor are buttons for 'Execute', 'Explain', 'Save as...', 'Kill Session', and 'New Worksheet'.
- Query Process Results (Status: Succeeded):** A section below the query editor showing the results of the query. It includes a 'Logs' tab and a 'Results' tab. The 'Results' tab is currently empty.

The screenshot displays the Hive Query Editor interface. On the left, the 'Database Explorer' shows the 'default' database selected, with a list of tables including 'hockey_award_coaches', 'hockey_awards_players', 'hockey_coaches', 'hockey_master', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_coaches', 'temp_master', and 'xademo'. The main 'Query Editor' area contains a single SQL query: `1 LOAD DATA INPATH '/tmp/Assignment_2/Scoring.csv' OVERWRITE INTO TABLE temp_scoring;`. Below the query editor, the 'Query Process Results (Status: Succeeded)' section shows the 'Results' tab with the following information:
INFO : Loading data to table default.temp_scoring from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/Scoring.csv
INFO : Table default.temp_scoring stats: [numFiles=1, numRows=0, totalSize=3272704, rawDataSize=0]

The screenshot displays the Hive Query Editor interface. On the left, the 'Database Explorer' shows the 'default' database selected, with a list of tables including 'hockey_award_coaches', 'hockey_awards_players', 'hockey_coaches', 'hockey_master', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_coaches', 'temp_master', and 'xademo'. The main 'Query Editor' area contains a single SQL query: `1 create table hockey_scoring (playerID STRING, year INT, stint INT, tmID STRING, lgID STRING, pos STRING, GP INT, G INT, A INT, Pts INT, PIM INT, PLUSMINUS INT, PPG INT, PPA INT, SHG INT, SHA INT, GWG INT, GTG INT, SOG INT, PostGP INT, PostG INT, PostA INT, PostPts INT, PostPIM INT, Post INT, PostPPG INT, PostPPA INT, PostSHG INT, PostSHA INT, PostGWG INT, PostSOG INT);`. Below the query editor, the 'Query Process Results (Status: Succeeded)' section shows the 'Results' tab with the following information:
INFO : Loading data to table default.hockey_scoring from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/Scoring.csv
INFO : Table default.hockey_scoring stats: [numFiles=1, numRows=0, totalSize=3272704, rawDataSize=0]

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_coaches
- hockey_master
- temp_awardscoaches
- temp_awardsplayers
- temp_coaches
- temp_master
- xademo

Query Editor

Worksheet

```

1 insert overwrite table hockey_scoring
2 SELECT
3 regexp_extract(col_value, '^([0-9]*)\\', 1) playerID,
4 regexp_extract(col_value, '^([0-9]*)\\', 2) year,
5 regexp_extract(col_value, '^([0-9]*)\\', 3) stint,
6 regexp_extract(col_value, '^([0-9]*)\\', 4) tmID,
7 regexp_extract(col_value, '^([0-9]*)\\', 5) lgID,
8 regexp_extract(col_value, '^([0-9]*)\\', 6) pos,
9 regexp_extract(col_value, '^([0-9]*)\\', 7) GP,
10 regexp_extract(col_value, '^([0-9]*)\\', 8) G,
11 regexp_extract(col_value, '^([0-9]*)\\', 9) A,
12 regexp_extract(col_value, '^([0-9]*)\\', 10) Pts,
13 regexp_extract(col_value, '^([0-9]*)\\', 11) PIM,
14 regexp_extract(col_value, '^([0-9]*)\\', 12) PLUSMINUS,
15 regexp_extract(col_value, '^([0-9]*)\\', 13) PPG,
16 regexp_extract(col_value, '^([0-9]*)\\', 14) PPA,
17 regexp_extract(col_value, '^([0-9]*)\\', 15) SHG,
18 regexp_extract(col_value, '^([0-9]*)\\', 16) SHA,
19 regexp_extract(col_value, '^([0-9]*)\\', 17) GWG,
20 regexp_extract(col_value, '^([0-9]*)\\', 18) SOG,
21 regexp_extract(col_value, '^([0-9]*)\\', 19) PostGP,
22 regexp_extract(col_value, '^([0-9]*)\\', 20) PostG,
23 regexp_extract(col_value, '^([0-9]*)\\', 21) PostA,
24 regexp_extract(col_value, '^([0-9]*)\\', 22) PostPts,
25 regexp_extract(col_value, '^([0-9]*)\\', 23) PostPIM,
26 regexp_extract(col_value, '^([0-9]*)\\', 24) Post,
27 regexp_extract(col_value, '^([0-9]*)\\', 25) PostPPG,
28 regexp_extract(col_value, '^([0-9]*)\\', 26) PostPPA,
29 regexp_extract(col_value, '^([0-9]*)\\', 27) PostSHG,
30 regexp_extract(col_value, '^([0-9]*)\\', 28) PostSHA,
31 regexp_extract(col_value, '^([0-9]*)\\', 29) PostSOG,
32 regexp_extract(col_value, '^([0-9]*)\\', 30)
33 regexp_extract(col_value, '^([0-9]*)\\', 31)
34 from temp_scoring;

```

Execute Explain Save as... Kill Session New Worksheet

100%

Query Process Results (Status: SUCCEEDED) Save results...

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

INFO : Tez session hasn't been created yet. Opening session
INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464020964330_0005)

INFO : Map 1: --
INFO : Map 1: 0/1
INFO : Map 1: 0(+1)/1
INFO : Map 1: 0(+1)/1
INFO : Map 1: 1/1

INFO : Loading data to table default.hockey_scoring from hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/hockey_scoring/.hive-staging_hive_2016-05-23_17-31-21_447_7225111650302003273-1-ext-10000

INFO : Table default.hockey_scoring stats: [numFiles=1, numRows=45975, totalSize=4237449, rawDataSize=4191474]

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_coaches
- hockey_master
- hockey_scoring
- temp_awardscoaches
- temp_awardsplayers
- temp_coaches
- temp_master
- temp_scoring
- xademo

Query Editor

Worksheet x hockey_scoring sample x

```
1 SELECT * FROM hockey_scoring LIMIT 100;
```

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded) Save results...

Logs Results

Filter columns... previous next

hockey_scoring.playerid	hockey_scoring.year	hockey_scoring.stint	hockey_scoring.tmid	hockey_scoring.lgid
playerID			tmID	lgID
aaltoan01	1997	1	ANA	NHL
aaltoan01	1998	1	ANA	NHL
aaltoan01	1999	1	ANA	NHL
aaltoan01	2000	1	ANA	NHL

1.2.6 Create Team Table

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_coaches
- hockey_master
- hockey_scoring
- temp_awardscoaches
- temp_awardsplayers
- temp_coaches
- temp_master
- temp_scoring
- xademo

Query Editor

Worksheet

```
1 create table temp_Team(col_value STRING);
```

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded) Save results...

Logs Results

Filter columns... previous next

The screenshot displays the Hive Query Editor interface. On the left, the 'Database Explorer' shows a list of databases including 'default', 'hockey_award_coaches', 'hockey_awards_players', 'hockey_coaches', 'hockey_master', 'hockey_scoring', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_coaches', 'temp_master', 'temp_scoring', and 'xademo'. The 'Query Editor' on the right contains a single query: `1 LOAD DATA INPATH '/tmp/Assignment_2/Teams.csv' OVERWRITE INTO TABLE temp_team;`. Below the query editor, the 'Query Process Results (Status: Succeeded)' section shows the 'Results' tab with the following information:
INFO : Loading data to table default.temp_team from hdfs://sandbox.hortonworks.com:8020/tmp/Assignment_2/Teams.csv
INFO : Table default.temp_team stats: [numFiles=1, numRows=0, totalSize=140288, rawDataSize=0]

The screenshot displays the Hive Query Editor interface. On the left, the 'Database Explorer' shows a list of databases including 'default', 'hockey_award_coaches', 'hockey_awards_players', 'hockey_coaches', 'hockey_master', 'hockey_scoring', 'temp_awardscoaches', 'temp_awardsplayers', 'temp_coaches', 'temp_master', 'temp_scoring', and 'xademo'. The 'Query Editor' on the right contains a query to create a table: `1 create table hockey_teams (year INT, lgID STRING, tmID STRING, franchID STRING, confID STRING, divID STRING, rank INT, playoff STRING, G INT, W INT, L INT, T INT, OIL INT, Pts INT, SoW INT, SoL INT, GF INT, GA INT, name STRING, FIM INT, BenchMinor INT, PPG INT, PPC INT, SHA INT, PKG INT, PKC INT, SHF INT);`. Below the query editor, the 'Query Process Results (Status: Succeeded)' section shows the 'Results' tab with a 'Filter columns...' input field and 'previous' and 'next' buttons.

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_coaches
- hockey_master
- hockey_scoring
- temp_awardscoaches
- temp_awardsplayers
- temp_coaches
- temp_master
- temp_scoring
- xademo

Query Editor

Worksheet

```

1 insert overwrite table hockey_teams
2 SELECT
3 regexp_extract(col_value, '^(?:[^\,]*\\,)?(11)', 1) year,
4 regexp_extract(col_value, '^(?:[^\,]*\\,)?(2)', 1) lgID,
5 regexp_extract(col_value, '^(?:[^\,]*\\,)?(3)', 1) tmID,
6 regexp_extract(col_value, '^(?:[^\,]*\\,)?(4)', 1) franchID,
7 regexp_extract(col_value, '^(?:[^\,]*\\,)?(5)', 1) confID,
8 regexp_extract(col_value, '^(?:[^\,]*\\,)?(6)', 1) divID,
9 regexp_extract(col_value, '^(?:[^\,]*\\,)?(7)', 1) rank,
10 regexp_extract(col_value, '^(?:[^\,]*\\,)?(8)', 1) playoff,
11 regexp_extract(col_value, '^(?:[^\,]*\\,)?(9)', 1) G,
12 regexp_extract(col_value, '^(?:[^\,]*\\,)?(10)', 1) W,
13 regexp_extract(col_value, '^(?:[^\,]*\\,)?(11)', 1) L,
14 regexp_extract(col_value, '^(?:[^\,]*\\,)?(12)', 1) T,
15 regexp_extract(col_value, '^(?:[^\,]*\\,)?(13)', 1) OLT,
16 regexp_extract(col_value, '^(?:[^\,]*\\,)?(14)', 1) Pts,
17 regexp_extract(col_value, '^(?:[^\,]*\\,)?(15)', 1) SoW,
18 regexp_extract(col_value, '^(?:[^\,]*\\,)?(16)', 1) SoL,
19 regexp_extract(col_value, '^(?:[^\,]*\\,)?(17)', 1) GF,
20 regexp_extract(col_value, '^(?:[^\,]*\\,)?(18)', 1) GA,
21 regexp_extract(col_value, '^(?:[^\,]*\\,)?(19)', 1) name,
22 regexp_extract(col_value, '^(?:[^\,]*\\,)?(20)', 1) FIM,
23 regexp_extract(col_value, '^(?:[^\,]*\\,)?(21)', 1) BenchMin,
24 regexp_extract(col_value, '^(?:[^\,]*\\,)?(22)', 1) PPG,
25 regexp_extract(col_value, '^(?:[^\,]*\\,)?(23)', 1) PFC,
26 regexp_extract(col_value, '^(?:[^\,]*\\,)?(24)', 1) SHA,
27 regexp_extract(col_value, '^(?:[^\,]*\\,)?(25)', 1) PKG,
28 regexp_extract(col_value, '^(?:[^\,]*\\,)?(26)', 1) PKC,
29 regexp_extract(col_value, '^(?:[^\,]*\\,)?(27)', 1) SHF
30 from temp_team;

```

Stop execution Explain Save as... Kill Session New Worksheet

100%

Query Process Results (Status: RUNNING)

Logs Results

INFO : Tez session hasn't been created yet. Opening session

Query Process Results (Status: SUCCEEDED) Save results...

Logs Results

INFO : Tez session hasn't been created yet. Opening session

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464020964330_0006)

INFO : Map 1: -/-

INFO : Map 1: 0/1

INFO : Map 1: 0(+1)/1

INFO : Map 1: 0/1

INFO : Map 1: 1/1

INFO : Loading data to table default.hockey_teams from hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/hockey_teams/.hive-staging_hive_2016-05-23_17-35-11_267_7083463340662463749-1/-ext-10000

INFO : Table default.hockey_teams stats: [numFiles=1, numRows=1529, totalSize=154564, rawDataSize=153035]

Hive Query Saved Queries History UDFs

Database Explorer

default

Search tables...

Databases

- default
- hockey_award_coaches
- hockey_awards_players
- hockey_coaches
- hockey_master
- hockey_scoring
- hockey_teams
- temp_awardscoaches
- temp_awardsplayers
- temp_coaches
- temp_master
- temp_scoring
- temp_team
- xademo

Query Editor

Worksheet x hockey_teams sample x

```
1 SELECT * FROM hockey_teams LIMIT 100;
```

Execute Explain Save as... Kill Session New Worksheet

Query Process Results (Status: Succeeded) Save results...

Logs Results

Filter columns...

previous next

hockey_teams.year	hockey_teams.lgid	hockey_teams.tmid	hockey_teams.franchid	hockey_teams.confid
	lgID	tmID	franchID	confID
1909	NHA	COB	BKN	
1909	NHA	HAI	MTL	
1909	NHA	LES	TBS	
1909	NHA	MOS	MOS	
1909	NHA	MOW	MTW	
1909	NHA	OT1	STE	
1909	NHA	REN	REN	
1910	NHA	MOC	MTL	

1.3 CSV Files Uploaded to HDFS Files

/ tmp / Assignment_2

+ New directory Browse... Select files to upload.

Assignment_2

Search File Names

Name	Size	Last Modified	Owner	Group	Permission	Actions
..						
AwardsCoaches.csv	3.0 kB	2016-05-24 04:39	admin	hdfs	-rw-r--r--	Download Move Delete Copy
AwardsPlayers.csv	75.0 kB	2016-05-24 04:39	admin	hdfs	-rw-r--r--	Download Move Delete Copy
Coaches.csv	79.0 kB	2016-05-24 04:39	admin	hdfs	-rw-r--r--	Download Move Delete Copy
Master.csv	887.0 kB	2016-05-24 04:39	admin	hdfs	-rw-r--r--	Download Move Delete Copy
Scoring.csv	3.1 MB	2016-05-24 04:39	admin	hdfs	-rw-r--r--	Download Move Delete Copy
Teams.csv	137.0 kB	2016-05-24 04:39	admin	hdfs	-rw-r--r--	Download Move Delete Copy

1.4 Pig Tables

1.4.1 Master table

Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');

Dump Master;

The screenshot shows a Pig console window with a script named 't1 (copy)' that has been completed. The job ID is 'job_1464026143636_0004' and it started on '2016-05-24 04:46'. The results are displayed as a list of player records, each containing various attributes like playerID, coachID, hofID, firstName, lastName, nameNote, nameGiven, nameNick, height, weight, shootCatch, legendsID, and ihdbID. The records are truncated for display.

```

(playerID, coachID, hofID, firstName, lastName, nameNote, nameGiven, nameNick, height, weight, shootCatch, legendsID, ihdbID
(aaltoan01,,Antti,Aalto,,Antti,,73,210,L,14862,5928,aaltoan01,1997,2000,,C,1975,3,4,Finland,,Lappeenranta,,
(abbeybr01,,Bruce,Abbey,,Bruce,,73,185,L,11918,abbeybr01,,1975,1975,D,1951,8,18,Canada,ON,Toronto,,,,
(abbotge01,,George,Abbott,,George Henry,Preacher,67,153,L,18411,14591,abbotge01,1943,1943,,G,1911,8,3,Canada,
(abbotre01,,Reg,Abbott,,Reginald Stewart,,71,164,L,11801,11431,abbotre01,1952,1952,,C,1930,2,4,Canada,MB,Winn
(abdelju01,,Justin,Abdelkader,,73,195,L,21661,81002,abdelju01,2007,2011,,L,1987,2,25,USA,MI,Muskegon,,,,
(abelcl01,,Clarence,Abel,,Clarence John,Taffy,73,225,L,11802,18751,abelcl01,1926,1933,,D,1900,5,28,USA,MI,Sau
(abelge01,,Gerry,Abel,,Gerald Scott,,74,168,L,11803,10017,abelge01,1966,1966,,L,1944,12,25,USA,MI,Detroit,,
(abelsi01,abelsi01c,abelsi01h,Sid,Abel,,Sidney Gerald,Boot Nose,71,170,L,P196901,5931,abelsi01,1938,1953,,C,19
(abgrade01,,Dennis,Abgrall,,Dennis Harvey,,73,180,R,11805,6708,abgrade01,1975,1975,1976,1977,R,1953,4,24,Canad
(abidra01,,Ramzi,Abid,,74,210,L,16620,29408,abidra01,2002,2006,,L,1980,3,24,Canada,QC,Montreal,,,,
(abrahch01,,Christer,Abrahamsson,,Christer,,72,180,L,,5932,abrahch01,,1974,1976,G,1947,4,8,Sweden,,Umea,,,,
(abrahth01,,Thommy,Abrahamsson,,Thommy Ulf,,74,185,L,11806,5933,abrahth01,1980,1980,1974,1976,D,1947,4,12,Swed
(achtyge01,,Gene,Achtymichuk,,Eugene Edward,Acky,71,170,L,11807,11142,achtyge01,1951,1958,,C,1932,9,7,Canada,
(acombdo01,,Doug,Acomb,,Douglas Raymond,,71,165,L,11808,10475,acombdo01,1969,1969,,C,1949,5,15,Canada,ON,Toro
(actonke01,,Keith,Acton,,Keith Edward,Woody,68,170,L,10002,5935,actonke01,1979,1993,,C,1958,4,15,Canada,ON,St
(adairji01,,Jim,Adair,,James Albert John,,71,180,L,,8323,adairji01,,1973,1973,C,1948,9,29,Canada,ON,Brockvill
(adamdo01,,Douglas,Adam,,Douglas Patrick,,71,165,L,11809,11560,adamdo01,1949,1949,,L,1923,9,7,Canada,ON,Toron
(adamipe01,,Petr,Adamik,,,,,adamipe01,,1978,1978,,,,,)
(adamlu01,,Luke,Adam,,74,203,,adamlu01,2010,2011,,L,1990,6,18,Canada,NL,St. John's,,,,)
(adamru01,,Russ,Adam,,Russell Norman,,70,185,L,11810,8812,adamru01,1982,1982,,C,1961,5,5,Canada,ON,Windsor,,
(adamsbi01,,Bill,Adams,,William,,R,60149,,,,,R,,,,,)
(adamsbr01,,Bryan,Adams,,Bryan,,72,185,L,19147,22031,adamsbr01,1999,2000,,L,1977,3,20,Canada,BC,Fort St. Jame
(,adamsch01h,Charles,Adams,,Charles Francis,,,,,B196001,,,,,1876,10,18,USA,VT,Newport,1947,,,,)
(adamschr01,,Craig,Adams,,,,,72,200,R,10045,25067,adamschr01,2000,2011,,R,1977,4,26,Brunei Darussalam,,Seria,,,,
(adamsgr01,,Greg,Adams,,Gregory Charles,Eddie,73,190,L,11811,6236,adamsgr01,1980,1989,,L,1960,5,31,Canada,BC,
(adamsgr02,,Greg,Adams,,Gregory Daren,Gus,Hawk,75,195,L,10003,6237,adamsgr02,1984,2000,,L,1963,8,15,Canada,BC
(adamsja01,adamsja01c,adamsja01h,Jack,Adams,,John James,Jolly Jack,69,175,R,P195901,23806,adamsja01,1917,1926,,
(adamsjo01,,John,Adams,,John Ellis,Jack,70,163,L,11813,5940,adamsjo01,1940,1940,,L,1920,5,5,Canada,AB,Calgary

```

1.4.2 Scoring Table

Scoring = LOAD '/tmp/Assignment_2/Scoring.csv' using PigStorage(',');

Dump Scoring;

The screenshot shows a Pig console window with a script named 'g' that has been completed. The job ID is 'job_1464241803099_0114' and it started on '2016-05-26 17:11'. The results are displayed as a list of player records, each containing various attributes like playerID, year, stint, tmID, lgID, pos, GP, G, A, Pts, PIM, +/-, PPG, PPA, SHG, SHA, GWS, GTG, SOG, PostGP, PostG, PostA, PostPts, PostPIM. The records are truncated for display.

```

(playerID, year, stint, tmID, lgID, pos, GP, G, A, Pts, PIM, +/-, PPG, PPA, SHG, SHA, GWS, GTG, SOG, PostGP, PostG, PostA, PostPts, PostPIM
(aaltoan01,1997,1,ANA,NHL,C,3,0,0,0,0,-1,0,0,0,0,0,1,,,,,)
(aaltoan01,1998,1,ANA,NHL,C,73,3,5,8,24,-12,2,1,0,0,0,0,61,4,0,0,0,2,0,0,0,0,0,0)
(aaltoan01,1999,1,ANA,NHL,C,63,7,11,18,26,-13,1,0,0,0,1,0,102,,,,)
(aaltoan01,2000,1,ANA,NHL,C,12,1,1,2,2,1,0,0,0,0,0,0,18,,,,)
(abbeybr01,1975,1,CIN,NHL,D,17,1,0,1,15,-3,0,0,0,0,0,2,,,,)
(abbotge01,1943,1,BOS,NHL,D,1,0,0,0,0,,,,,)
(abbotre01,1952,1,MTL,NHL,C,3,0,0,0,0,,,,,)
(abdelju01,2007,1,DET,NHL,L,2,0,0,0,2,0,0,0,0,0,6,,,,)
(abdelju01,2008,1,DET,NHL,L,2,0,0,0,0,0,0,0,0,0,2,10,2,1,3,0,2,0,0,0,0,11)
(abdelju01,2009,1,DET,NHL,L,50,3,3,6,35,-11,0,0,0,0,0,79,11,1,1,2,36,1,0,0,0,0,12)
(abdelju01,2010,1,DET,NHL,L,74,7,12,19,61,15,0,0,0,1,1,129,11,0,0,22,-4,0,0,0,0,17)
(abdelju01,2011,1,DET,NHL,L,81,0,14,22,62,4,0,0,0,0,1,121,5,0,0,0,2,-5,0,0,0,0,7)
(abelcl01,1926,1,NYR,NHL,D,44,8,4,12,78,,,,,2,0,1,1,0,,,,)
(abelcl01,1927,1,NYR,NHL,D,23,0,1,1,28,,,,,0,1,0,1,14,,,,)
(abelcl01,1928,1,NYR,NHL,D,44,3,1,4,41,,,,,6,0,0,0,8,,,,)
(abelcl01,1929,1,CHI,NHL,D,38,3,3,6,42,,,,,2,0,0,0,10,,,,)
(abelcl01,1930,1,CHI,NHL,D,43,0,1,1,45,,,,,9,0,0,0,8,,,,)
(abelcl01,1931,1,CHI,NHL,D,48,4,3,7,34,,,,,2,0,0,0,2,,,,)
(abelcl01,1932,1,CHI,NHL,D,47,0,4,4,63,,,,,0,,,,)
(abelcl01,1933,1,CHI,NHL,D,46,2,1,3,28,,,,,0,0,0,0,8,,,,)
(abelge01,1966,1,DET,NHL,C,1,0,0,0,0,0,0,0,0,0,,,,)
(abelsi01,1938,1,DET,NHL,L,15,1,1,2,0,,,,,0,1,1,2,,,,)
(abelsi01,1939,1,DET,NHL,L,24,1,5,6,6,,,,,5,0,3,21,,,,)
(abelsi01,1940,1,DET,NHL,C,47,11,22,33,29,,,,,9,2,4,2,,,,)
(abelsi01,1941,1,DET,NHL,C,48,18,31,49,45,,,,,12,4,2,6,8,,,,)
(abelsi01,1942,1,DET,NHL,L,49,18,24,42,33,,,,,10,5,8,13,4,,,,)
(abelsi01,1945,1,DET,NHL,C,7,0,2,2,0,,,,,3,0,0,0,0,,,,)
(abelsi01,1946,1,DET,NHL,C,60,19,29,48,29,,,,,3,1,1,2,2,,,,)

```

1.4.3 Team Table

Team = LOAD '/tmp/Assignment_2/Teams.csv' using PigStorage(',');
Dump Team;

The screenshot shows the Pig script execution interface. The script is named 'g' and is in the 'COMPLETED' state. The job ID is 'job_1464241803099_0116' and it started on '2016-05-26 17:12'. The results are displayed in a scrollable list, showing the first few lines of the 'Team' table dump. The results are as follows:

```
(year, lgID, tmID, franchID, confID, divID, rank, playoff, G, W, L, T, OTL, Pts, Sol, Sol, GF, GA, name, PIN, BenchMInor, PPG, PPC, SHA, PKG)
(1909, NHA, COB, BKN, , , , 12, 4, 8, 0, , , , 79, 104, Cobalt Silver Kings, , , , , )
(1909, NHA, HAI, MTL, , , , 5, 12, 4, 8, 0, , , , 77, 83, Halleybury Hockey Club, , , , , )
(1909, NHA, LES, TBS, , , , 7, 12, 2, 10, 0, , , , 59, 100, Les Canadiens, , , , , )
(1909, NHA, MOS, MOS, , , , 6, 12, 3, 8, 1, , , , 52, 95, Montreal Shamrocks, , , , , )
(1909, NHA, MOW, MTH, , , , 1, 12, 11, 1, 0, , , , 22, 91, Montreal Wanderers, , , , , )
(1909, NHA, OTI, STE, , , , 2, 12, 9, 3, 0, , , , 89, 66, Ottawa Senators, , , , , )
(1909, NHA, REN, REN, , , , 3, 12, 8, 3, 1, , , , 96, 54, Renfrew Creamery Kings, , , , , )
(1910, NHA, MOC, MTL, , , , 2, 16, 8, 8, 0, , , , 66, 62, Montreal Canadiens, , , , , )
(1910, NHA, MOW, MTH, , , , 4, 16, 7, 9, 0, , , , 73, 88, Montreal Wanderers, , , , , )
(1910, NHA, OTI, STE, , , , 1, 16, 13, 3, 0, , , , 122, 69, Ottawa Senators, , , , , )
(1910, NHA, QUI, BKN, , , , 5, 16, 4, 12, 0, , , , 65, 97, Quebec Bulldogs, , , , , )
(1910, NHA, REN, REN, , , , 3, 16, 8, 8, 0, , , , 91, 101, Renfrew Creamery Kings, , , , , )
(1911, NHA, MOC, MTL, , , , 4, 18, 9, 10, 0, , , , 59, 66, Montreal Canadiens, , , , , )
(1911, NHA, MOW, MTH, , , , 3, 18, 9, 9, 0, , , , 95, 96, Montreal Wanderers, , , , , )
(1911, NHA, OTI, STE, , , , 2, 18, 9, 9, 0, , , , 99, 93, Ottawa Senators, , , , , )
(1911, NHA, QUI, BKN, , , , 1, 18, 10, 8, 0, , , , 81, 79, Quebec Bulldogs, , , , , )
(1911, PCHA, NWR, POI, , , , 1, 15, 9, 6, 0, , , , 78, 77, New Westminster Royals, , , , , )
(1911, PCHA, VAI, SPO, , , , 3, 16, 7, 9, 0, , , , 81, 90, Victoria Aristocrats, , , , , )
(1911, PCHA, VML, VWR, , , , 2, 15, 7, 8, 0, , , , 102, 94, Vancouver Millionaires, , , , , )
(1912, NHA, MOC, MTL, , , , 5, 20, 9, 11, 0, , , , 83, 81, Montreal Canadiens, , , , , )
(1912, NHA, MOW, MTH, , , , 2, 20, 10, 10, 0, , , , 93, 90, Montreal Wanderers, , , , , )
(1912, NHA, OTI, STE, , , , 3, 20, 9, 11, 0, , , , 87, 81, Ottawa Senators, , , , , )
(1912, NHA, QUI, BKN, , , , 1, 20, 16, 4, 0, , , , 112, 75, Quebec Bulldogs, , , , , )
(1912, NHA, TBS, TBS, , , , 4, 20, 9, 11, 0, , , , 86, 95, Toronto Blueshirts, , , , , )
(1912, NHA, TOI, TOO, , , , 6, 20, 7, 13, 0, , , , 59, 98, Toronto Tecumsehs, , , , , )
(1912, PCHA, NWR, POI, , , , 3, 13, 4, 9, 0, , , , 46, 61, New Westminster Royals, , , , , )
(1912, PCHA, VAI, SPO, , , , 1, 15, 10, 5, 0, , , , 68, 56, Victoria Aristocrats, , , , , )
(1912, PCHA, VML, VWR, , , , 2, 14, 7, 7, 0, , , , 71, 68, Vancouver Millionaires, , , , , )
```

1.4.4 Coach Table

The screenshot shows the Pig script execution interface. The script is named 'g' and is in the 'COMPLETED' state. The job ID is 'job_1464241803099_0118' and it started on '2016-05-26 17:14'. The results are displayed in a scrollable list, showing the first few lines of the 'Coach' table dump. The results are as follows:

```
(coachID, year, tmID, lgID, stint, notes, g, w, l, t, postg, postw, postl, postt)
(abelsi01c, 1952, CHI, NHL, 1, , 70, 27, 28, 15, 7, 3, 4, 0)
(abelsi01c, 1953, CHI, NHL, 1, , 70, 12, 51, 7, , , , )
(abelsi01c, 1957, DET, NHL, 2, , 33, 16, 12, 5, 4, 0, 4, 0)
(abelsi01c, 1958, DET, NHL, 1, , 70, 25, 37, 8, , , , )
(abelsi01c, 1959, DET, NHL, 1, , 70, 26, 29, 15, 6, 2, 4, 0)
(abelsi01c, 1960, DET, NHL, 1, , 70, 25, 29, 16, 11, 6, 5, 0)
(abelsi01c, 1961, DET, NHL, 1, , 70, 23, 33, 14, , , , )
(abelsi01c, 1962, DET, NHL, 1, , 70, 32, 25, 13, 11, 5, 6, 0)
(abelsi01c, 1963, DET, NHL, 1, , 70, 30, 29, 11, 14, 7, 7, 0)
(abelsi01c, 1964, DET, NHL, 1, , 70, 40, 23, 7, 7, 3, 4, 0)
(abelsi01c, 1965, DET, NHL, 1, , 70, 31, 27, 12, 12, 6, 6, 0)
(abelsi01c, 1966, DET, NHL, 1, , 70, 27, 39, 4, , , , )
(abelsi01c, 1967, DET, NHL, 1, , 74, 27, 35, 12, , , , )
(abelsi01c, 1969, DET, NHL, 2, , 74, 38, 21, 15, 4, 0, 4, 0)
(abelsi01c, 1971, STL, NHL, 1, , 10, 3, 6, 1, , , , )
(abelsi01c, 1975, KCS, NHL, 2, , 3, 0, 3, 0, , , , )
(adamsja01c, 1927, DTC, NHL, 1, , 44, 19, 19, 6, , , , )
(adamsja01c, 1928, DTC, NHL, 1, , 44, 19, 16, 9, 2, 0, 2, 0)
(adamsja01c, 1929, DTC, NHL, 1, , 44, 14, 24, 6, , , , )
(adamsja01c, 1930, DTF, NHL, 1, , 44, 16, 21, 7, , , , )
(adamsja01c, 1931, DTF, NHL, 1, , 48, 18, 20, 10, 2, 0, 1, 1)
(adamsja01c, 1932, DET, NHL, 1, , 48, 25, 15, 8, 4, 2, 2, 0)
(adamsja01c, 1933, DET, NHL, 1, , 48, 24, 14, 10, 9, 4, 5, 0)
(adamsja01c, 1934, DET, NHL, 1, , 48, 19, 22, 7, , , , )
(adamsja01c, 1935, DET, NHL, 1, , 48, 24, 16, 8, 7, 6, 1, 0)
(adamsja01c, 1936, DET, NHL, 1, , 48, 25, 14, 9, 10, 6, 4, 0)
(adamsja01c, 1937, DET, NHL, 1, , 48, 12, 25, 11, , , , )
(adamsja01c, 1938, DET, NHL, 1, , 48, 18, 24, 6, 6, 3, 3, 0)
```


1.4.5 AwardCoach Table

The screenshot shows a web interface for a script named 'g'. The script status is 'COMPLETED'. The job ID is 'job_1464241803099_0122' and it started on '2016-05-26 17:17'. The results are displayed in a scrollable area with a 'Download' button.

Script: g - **COMPLETED**

Job ID: job_1464241803099_0122

Started: 2016-05-26 17:17

Results:

```
(coachID,award,year,lgID,note)
(patrule01c,First Team All-Star,1930,NHL,)
(irvindi01c,Second Team All-Star,1930,NHL,)
(patrule01c,First Team All-Star,1931,NHL,)
(irvindi01c,Second Team All-Star,1931,NHL,)
(patrule01c,First Team All-Star,1932,NHL,)
(irvindi01c,Second Team All-Star,1932,NHL,)
(patrule01c,First Team All-Star,1933,NHL,)
(irvindi01c,Second Team All-Star,1933,NHL,)
(patrule01c,First Team All-Star,1934,NHL,)
(irvindi01c,Second Team All-Star,1934,NHL,)
(patrule01c,First Team All-Star,1935,NHL,)
(gormato01c,Second Team All-Star,1935,NHL,)
(adamsja01c,First Team All-Star,1936,NHL,)
(hartce01c,Second Team All-Star,1936,NHL,)
(patrule01c,First Team All-Star,1937,NHL,)
(rossar01c,Second Team All-Star,1937,NHL,)
(rossar01c,First Team All-Star,1938,NHL,)
(duttore01c,Second Team All-Star,1938,NHL,)
(thomppa01c,First Team All-Star,1939,NHL,)
(bouchfr01c,Second Team All-Star,1939,NHL,)
(weilaco01c,First Team All-Star,1940,NHL,)
(irvindi01c,Second Team All-Star,1940,NHL,)
(bouchfr01c,First Team All-Star,1941,NHL,)
(thomppa01c,Second Team All-Star,1941,NHL,)
(adamsja01c,First Team All-Star,1942,NHL,)
(rossar01c,Second Team All-Star,1942,NHL,)
(irvindi01c,First Team All-Star,1943,NHL,)
(dayha01c,Second Team All-Star,1943,NHL,)
```

1.4.6 AwardPlayer Table

The screenshot shows a web interface for a script named 'g'. The script status is 'COMPLETED'. The job ID is 'job_1464241803099_0120' and it started on '2016-05-26 17:15'. The results are displayed in a scrollable area with a 'Download' button.

Script: g - **COMPLETED**

Job ID: job_1464241803099_0120

Started: 2016-05-26 17:15

Results:

```
(playerID,award,year,lgID,note,pos)
(malonjo01,Art Ross,1917,NHL,,)
(cleghod01,Art Ross,1918,NHL,,)
(malonjo01,Art Ross,1919,NHL,,)
(lalonne01,Art Ross,1920,NHL,,)
(broadpu01,Art Ross,1921,NHL,,)
(dyeba01,Art Ross,1922,NHL,,)
(dennecy01,Art Ross,1923,NHL,,)
(nighbfr01,Hart,1923,NHL,,)
(dyeba01,Art Ross,1924,NHL,,)
(burchbi01,Hart,1924,NHL,,)
(nighbfr01,Lady Byng,1924,NHL,,)
(stewane01,Art Ross,1925,NHL,,)
(stewane01,Hart,1925,NHL,,)
(nighbfr01,Lady Byng,1925,NHL,,)
(cookbi01,Art Ross,1926,NHL,,)
(gardihe01,Hart,1926,NHL,,)
(burchbi01,Lady Byng,1926,NHL,,)
(hainsge01,Vezina,1926,NHL,,)
(moreno01,Art Ross,1927,NHL,,)
(moreno01,Hart,1927,NHL,,)
(bouchfr01,Lady Byng,1927,NHL,,)
(hainsge01,Vezina,1927,NHL,,)
(baileac01,Art Ross,1928,NHL,,)
(wortero01,Hart,1928,NHL,,)
(bouchfr01,Lady Byng,1928,NHL,,)
(hainsge01,Vezina,1928,NHL,,)
(weilaco01,Art Ross,1929,NHL,,)
(stewane01,Hart,1929,NHL,,)
```

2 Hive and Pig Queries

2.1 Task 2

2.1.1 Hive Query 2A

```
SELECT s.year AS YEAR,
       m.firstname AS First_Name,
       m.lastname AS Last_Name,
       CONCAT(m.birthday,'/',m.birthmon,'/',m.BirthYear) AS Birth_Date,
       m.birthcountry AS Birth_Country,
       t.name AS Team_Name,
       s.Lgid AS League_ID,
       s.pos AS POSITION,
       s.gwg AS Game_Winning_Goals
FROM
  (SELECT playerid,
         gwg,
         YEAR,
         tmid,
         lgid,
         pos
  FROM hockey_scoring
 WHERE gwg IN
       (SELECT max(gwg)
        FROM hockey_scoring)) s
JOIN hockey_master m ON (s.playerid = m.playerid)
JOIN hockey_teams t ON (t.tmid = s.tmid
                        AND t.YEAR = s.YEAR);
```

2.1.1.1 Output

100%

Query Process Results (Status: SUCCEEDED) Save results... ▾

Logs Results

Filter columns... previous next

year	first_name	last_name	birth_date	birth_country	team_name	league_id	position	game_winning...
1970	Phil	Esposito	20/2/1942	Canada	Boston Bruins	NHL	C	16
1971	Phil	Esposito	20/2/1942	Canada	Boston Bruins	NHL	C	16
1983	Michel	Goulet	21/4/1960	Canada	Quebec Nordiques	NHL	L	16

2.1.1.2 Log

100%

Query Process Results (Status: SUCCEEDED) Save results... ▾



Logs Results

INFO : Session is already open
INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0132)

INFO : Map 1: -/- Map 2: 0/1 Map 4: -/- Map 5: 0/1 Reducer 3: 0/1
INFO : Map 1: 0/1 Map 2: 0/1 Map 4: 0/1 Map 5: 0/1 Reducer 3: 0/1
INFO : Map 1: 0(+1)/1 Map 2: 0(+1)/1 Map 4: 0(+1)/1 Map 5: 0/1 Reducer 3: 0/1
INFO : Map 1: 0(+1)/1 Map 2: 0(+1)/1 Map 4: 0(+1)/1 Map 5: 0(+1)/1 Reducer 3: 0/1

2.1.1.3 DAG Details

DAG Details	
 Download data	
Application Id	application_1464215212735_0132
Entity Id	dag_1464215212735_0132_2
User	hive
Status	 SUCCEEDED
Start Time	26 May 2016 14:26:30
End Time	26 May 2016 14:26:38
Duration	7 secs

2.1.1.4 Task Details

RegEx or Column1, Column2... :RegEx

Search

First1Last - 1

Rows25

Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration
02_000000	Reducer 3	<div><div></div>SUCCEEDED</div>	<div><div>100%</div></div>	26 May 2016 14:26:37	26 May 2016 14:26:37	195 ms
04_000000	Map 4	<div><div></div>SUCCEEDED</div>	<div><div>100%</div></div>	26 May 2016 14:26:31	26 May 2016 14:26:38	7 secs
03_000000	Map 1	<div><div></div>SUCCEEDED</div>	<div><div>100%</div></div>	26 May 2016 14:26:31	26 May 2016 14:26:38	7 secs
01_000000	Map 2	<div><div></div>SUCCEEDED</div>	<div><div>100%</div></div>	26 May 2016 14:26:31	26 May 2016 14:26:37	6 secs
00_000000	Map 5	<div><div></div>SUCCEEDED</div>	<div><div>100%</div></div>	26 May 2016 14:26:31	26 May 2016 14:26:37	5 secs

2.1.2 Pig Query 2A

```

Scorings = LOAD '/tmp/Assignment_2/Scoring.csv' using PigStorage(',');

Scorings_raw = FILTER Scorings BY $0 != 'playerID';

Scoring = FOREACH Scorings_raw GENERATE $0 as playerid, $1 as Year, $3 as tmID,$4 as lgID,
$5 as pos, $16 as gwg;

grp_by_year = GROUP Scoring BY (Year);

max_year_gwg = FOREACH grp_by_year GENERATE group as year_grp,
MAX(Scoring.gwg) as max_gwg;

order_max_gwg = ORDER max_year_gwg by max_gwg desc;

limit_gwg = LIMIT order_max_gwg 1;

join_Scoring = JOIN Scoring by gwg,limit_gwg by max_gwg;

sort_join_Scoring = FOREACH join_Scoring GENERATE $0 as playerID, $1 as Year,
$2 as tmID, $3 as lgID, $4 as pos, $5 as gwg;

Teams = LOAD '/tmp/Assignment_2/Teams.csv' using PigStorage(',');

Teams_raw = FILTER Teams BY $0 > 0;

Team = FOREACH Teams_raw GENERATE $0 As Year, $2 as tmID, $18 as name;

join_team = JOIN sort_join_Scoring by (Year,tmID), Team by (Year,tmID);

sort_join_team = FOREACH join_team GENERATE $0 as playerID, $1 as Year,
$3 as lgID, $4 as pos, $5 as gwg, $8 as teamName;

Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');

Masters_raw = FILTER Masters BY $0 != 'playerID';

Master = FOREACH Masters_raw GENERATE $0 as playerid, $1 as coachid, $3 as firstName, $4 as
lastName, $19 as birthYear, $20 as birthMon, $21 as birthDay, $22 as birthCountry;

join_Master = JOIN sort_join_team by playerID, Master by playerid;

sort_join_Master = FOREACH join_Master GENERATE $1 as Year,
$8 as firstname, $9 as lastname, $10 as birthyear, $11 as birthmonth, $12 as birthday,
$13 as birthcountry, $5 as teamname, $2 as LgID, $3 as Pos, $4 as GWG;

dump sort_join_Master;

```

2.1.2.1 Output

▼ Results

Download

(1971,Phil,Esposito,1942,2,20,Canada,Boston Bruins,NHL,C,16)

(1970,Phil,Esposito,1942,2,20,Canada,Boston Bruins,NHL,C,16)

(1983,Michel,Goulet,1960,4,21,Canada,Quebec Nordiques,NHL,L,16)

2.1.2.2 Log

```

HadoopVersion  PigVersion      UserId  StartedAt      FinishedAt      Features
2.7.1.2.3.2.0-2950  0.15.0.2.3.2.0-2950  yarn    2016-05-26 05:56:46  2016-05-26 05:59:27  HASH_JOIN,GR

Success!

Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime  MinMapTime  AvgMapTime  MedianMapTime  MaxReduceTime  MinReduceTime  Features
job_1464241803099_0002  1      1      5          5          5          5          3          3          Scoring,Scor:
job_1464241803099_0003  1      1      2          2          2          2          2          2          order_max_gw
job_1464241803099_0004  1      1      2          2          2          2          2          2          order_max_gw
job_1464241803099_0005  1      1      2          2          2          2          4          4          order_max_gw
job_1464241803099_0006  2      1      3          2          3          3          2          2          join_Scoring
job_1464241803099_0007  2      1      3          2          3          3          4          4          Team,Teams,Ti
job_1464241803099_0008  2      1      5          4          4          4          3          3          Master,Maste

Input(s):
Successfully read 45975 records (3273095 bytes) from: "/tmp/Assignment_2/Scoring.csv"
Successfully read 1529 records from: "/tmp/Assignment_2/Teams.csv"
Successfully read 7770 records from: "/tmp/Assignment_2/Master.csv"

Output(s):
Successfully stored 3 records (228 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp-83753670/tmp-1384434984"

```

```
2016-05-26 05:59:29,266 [main] INFO org.apache.pig.Main - Pig script completed in 2 minutes, 46 seconds and 742 mil
```

2.1.3 Comparison Table for 2A

	HIVE	PIG
No of Jobs	3	7
Maps	4	2
Reduces	1	1
Total Time	7 sec	2 min 46 sec

2.1.4 Hive Query 2B

```

SELECT mas.firstname AS First_Name,
       mas.lastname AS Last_Name,
       aw.maxAwards AS Total_Awards
FROM
  (SELECT max(ap.AwardCount) AS maxAwards,
         ap.playerid
   FROM
     ( SELECT count(playerid) AS AwardCount,
            playerid
      FROM hockey_awards_players
      GROUP BY playerid) ap
   WHERE ap.playerid IN
     (SELECT s.playerid
      FROM
        (SELECT playerid,
               gwg,
               YEAR,
               tmid,
               lgid,
               pos
        FROM hockey_scoring
        WHERE gwg IN
          (SELECT max(gwg)
           FROM hockey_scoring)) s
        JOIN hockey_master m ON (s.playerid = m.playerid))
   GROUP BY ap.playerid
   ORDER BY maxAwards DESC ) aw
JOIN
  (SELECT firstname,
         lastname,

```

playerid

FROM hockey_master) mas ON (aw.playerid = mas.PlayerID) LIMIT 1;

2.1.4.1 Output

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▾

Logs

Results

Filter columns...
previous next

first_name	last_name	total_awards
Phil	Esposito	17

2.1.4.2 Log

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▾

Logs

Results

INFO : Session is already open

INFO : Tez session was closed. Reopening...

INFO : Session re-established.

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0133)

INFO : Map 1: 0/1 Map 5: 0/1 Map 6: 0/1 Map 8: 0/1 Map 9: 0/1 Reducer 2: 0/1 Reducer 3: 0/1 Reducer 4: 0/1 Reducer 7: 0/1

2.1.4.3 DAG Details

DAG Details

Download data

Application Id	application_1464215212735_0133
Entity Id	dag_1464215212735_0133_1
User	hive
Status	SUCCEEDED
Start Time	26 May 2016 14:34:29
End Time	26 May 2016 14:34:39
Duration	9 secs

2.1.4.4 Task Details

RegEx or Column1, Column2...:RegEx Search First 1 Last - 1 Rows 25

Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration
08_000000	Reducer 4	✓ SUCCEEDED	100%	26 May 2016 14:34:38	26 May 2016 14:34:39	441 ms
07_000000	Reducer 3	✓ SUCCEEDED	100%	26 May 2016 14:34:38	26 May 2016 14:34:38	304 ms
03_000000	Reducer 7	✓ SUCCEEDED	100%	26 May 2016 14:34:37	26 May 2016 14:34:37	184 ms
06_000000	Reducer 2	✓ SUCCEEDED	100%	26 May 2016 14:34:37	26 May 2016 14:34:38	1 secs
04_000000	Map 5	✓ SUCCEEDED	100%	26 May 2016 14:34:30	26 May 2016 14:34:38	8 secs
00_000000	Map 8	✓ SUCCEEDED	100%	26 May 2016 14:34:30	26 May 2016 14:34:36	6 secs
02_000000	Map 6	✓ SUCCEEDED	100%	26 May 2016 14:34:30	26 May 2016 14:34:37	7 secs
05_000000	Map 1	✓ SUCCEEDED	100%	26 May 2016 14:34:30	26 May 2016 14:34:37	6 secs
01_000000	Map 9	✓ SUCCEEDED	100%	26 May 2016 14:34:30	26 May 2016 14:34:37	7 secs

2.1.5 Pig Query 2B

```
Scorings = LOAD '/tmp/Assignment_2/Scoring.csv' using PigStorage(',');
```

```
Scorings_raw = FILTER Scorings BY $0 != 'playerID';
```

```
Scoring = FOREACH Scorings_raw GENERATE $0 as playerid, $1 as Year, $3 as tmID,$4 as lgID,
$5 as pos, $16 as gwg;
```

```
grp_by_year = GROUP Scoring BY (Year);
```

```
max_year_gwg = FOREACH grp_by_year GENERATE group as year_grp,
```

```
MAX(Scoring.gwg) as max_gwg;
```

```
order_max_gwg = ORDER max_year_gwg by max_gwg desc;
```

```
limit_gwg = LIMIT order_max_gwg 1;
```

```
join_Scoring = JOIN Scoring by gwg,limit_gwg by max_gwg;
```

```
sort_join_Scoring = FOREACH join_Scoring GENERATE $0 as playerID, $1 as Year,
$2 as tmID, $3 as lgID, $4 as pos, $5 as gwg;
```

```
Teams = LOAD '/tmp/Assignment_2/Teams.csv' using PigStorage(',');
```

```
Teams_raw = FILTER Teams BY $0 > 0;
```

```
Team = FOREACH Teams_raw GENERATE $0 As Year, $2 as tmID, $18 as name;
```

```
join_team = JOIN sort_join_Scoring by (Year,tmID), Team by (Year,tmID);
```

```
sort_join_team = FOREACH join_team GENERATE $0 as playerID, $1 as Year,
$3 as lgID, $4 as pos, $5 as gwg, $8 as teamName;
```

```

Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');
Masters_raw = FILTER Masters BY $0 != 'playerID';
Master = FOREACH Masters_raw GENERATE $0 as playerid, $1 as coachid, $3 as firstName,
$4 as lastName, $19 as birthYear, $20 as birthMon, $21 as birthDay, $22 as birthCountry;
join_Master = JOIN sort_join_team by playerID, Master by playerid;
sort_join_Master = FOREACH join_Master GENERATE $0 as playerid,
$8 as firstname, $9 as lastname;
awardplayers = load '/tmp/Assignment_2/AwardsPlayers.csv' using PigStorage(',');
awardplayers_raw = FILTER awardplayers BY $0 != 'playerID';
awardplayer = FOREACH awardplayers_raw GENERATE $0 as playerID, $1 as award, $2 as year;
group_awardplayer = group awardplayer by (playerID);
count_award = FOREACH group_awardplayer generate group as grp_yr ,
COUNT(awardplayer.award) as no_of_award;
award_final = FOREACH count_award GENERATE $0 as playerid , $1 as No_of_awards;
join_awardplayer = JOIN sort_join_Master by (playerid), award_final by (playerid);
final_awardplayer = FOREACH join_awardplayer GENERATE $1 as firstname, $2 as lastname,
$4 as no_of_award;
order_final_awardplayer = ORDER final_awardplayer by no_of_award desc;
limit_final_awardplayer = LIMIT order_final_awardplayer 1;
dump limit_final_awardplayer;

```

2.1.5.1 Output

▼ Results	Download
(Phil,Esposito,17)	

2.1.5.2 Log

```

HadoopVersion  PigVersion      UserId  StartedAt      FinishedAt      Features
2.7.1.2.3.2.0-2950      0.15.0.2.3.2.0-2950      yarn      2016-05-26 06:02:39      2016-05-26 06:06:29      HASH_JOIN,GR

Success!

Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime      MaxReduceTime      MinReduceTime
job_1464241803099_0010  1      1      3      3      3      3      2      2      2      2      award_final,i
job_1464241803099_0011  1      1      4      4      4      4      2      2      2      2      Scoring,Scor:
job_1464241803099_0012  1      1      2      2      2      2      2      2      2      2      order_max_gw
job_1464241803099_0013  1      1      2      2      2      2      2      2      2      2      order_max_gw
job_1464241803099_0014  1      1      2      2      2      2      2      2      2      2      order_max_gw
job_1464241803099_0015  2      1      3      2      2      3      3      3      3      3      join_Scoring
job_1464241803099_0016  2      1      3      3      3      3      2      2      2      2      Team,Teams,T
job_1464241803099_0017  2      1      3      3      3      3      2      2      2      2      Master,Maste
job_1464241803099_0018  2      1      4      3      3      3      3      3      3      3      final_awardp
job_1464241803099_0019  1      1      2      2      2      3      3      3      3      3      order_final_i
job_1464241803099_0020  1      1      2      2      2      2      2      2      2      2      order_final_i
job_1464241803099_0021  1      1      3      3      3      3      2      2      2      2      order_final_i

Input(s):
Successfully read 45975 records (3273095 bytes) from: "/tmp/Assignment_2/Scoring.csv"
Successfully read 1529 records from: "/tmp/Assignment_2/Teams.csv"
Successfully read 7770 records from: "/tmp/Assignment_2/Master.csv"
Successfully read 2093 records (77197 bytes) from: "/tmp/Assignment_2/AwardsPlayers.csv"

```

```

2016-05-26 06:06:31,601 [main] INFO org.apache.pig.Main - Pig script completed in 3 minutes, 54 seconds and 537 mil

```

2.1.6 Comparison table for 2B

	HIVE	PIG
No of Jobs	8	12
Maps	5	2
Reduces	4	1
Total Time	9 sec	3 min 54 sec

2.1.7 Hive Query 2 C

```

Select aw1.award as Award_Name,aw1.year as Year,s1.pts as Points
from hockey_scoring s1
join
(
select award,playerid,year
from hockey_awards_players
where playerid in (
select playerid
from hockey_master mas1
join
(
SELECT mas.firstname, mas.lastname, aw.maxAwards
from
(
select max(ap.AwardCount) as maxAwards,ap.playerid
from
(
select count(playerid) as AwardCount,playerid
from hockey_awards_players
group by playerid) ap
WHERE ap.playerid in
(SELECT s.playerid
FROM (
select playerid,gwg,year,tmid,lgid,pos
from hockey_scoring
where gwg in ( select max(gwg)
from hockey_scoring)) s
Join
hockey_master m
on ( s.playerid = m.playerid))
group by ap.playerid
order by maxAwards desc
) aw
JOIN
( SELECT firstname,lastname,playerid
from hockey_master) mas
ON (aw.playerid = mas.PlayerID)LIMIT 1) mas2
ON ( mas1.firstname = mas2.firstname and mas1.lastname = mas2.lastname))) aw1
ON ( s1.year = aw1.year and s1.playerid = aw1.playerid)
order by points desc;

```

2.1.7.1 Output

100%

Query Process Results (Status: SUCCEEDED)

Save results... ▾

Logs

Results

Filter columns...

previous

next

award_name	year	points
Art Ross	1970	152
Pearson	1970	152
First Team All-Star	1970	152
First Team All-Star	1973	145
Hart	1973	145
Art Ross	1973	145
Art Ross	1971	133
First Team All-Star	1971	133
Pearson	1972	130
First Team All-Star	1972	130
Art Ross	1972	130
Second Team All-Star	1974	127
First Team All-Star	1968	126
Hart	1968	126
Art Ross	1968	126
First Team All-Star	1969	99
Second Team All-Star	1967	84

2.1.7.2 Log

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▾

Logs

Results

INFO : Session is already open

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0133)

INFO : Map 1: -/- Map 10: -/- Map 11: 0/1 Map 12: -/- Map 13: -/- Map 2: 0/1 Map 7: -/- Map 8: -/- Reducer 14: 0/1 Reducer 3: 0/1 Reducer 4: 0/1 Reducer 5: 0/1 Reducer 6: 0/1 Reducer 9: 0/1

INFO : Map 1: 0/1 Map 10: 0/1 Map 11: 0/1 Map 12: 0/1 Map 13: 0/1 Map 2: 0/1 Map 7: 0/1 Map 8: 0/1 Reducer 14: 0/1 Reducer 3: 0/1 Reducer 4: 0/1 Reducer 5: 0/1 Reducer 6: 0/1 Reducer 9: 0/1

INFO : Map 1: 0/1 Map 10: 0/1 Map 11: 0/1 Map 12: 0/1 Map 13: 0/1 Map 2: 0/1 Map 7: 0/1 Map 8: 0/1 Reducer 14: 0/1 Reducer 3: 0/1 Reducer 4: 0/1 Reducer 5: 0/1 Reducer 6: 0/1 Reducer 9: 0/1

2.1.7.3 DAG Details

DAG Details	
<div style="background-color: #00bcd4; color: white; padding: 5px; display: inline-block;"> Download data </div>	
Application Id	application_1464215212735_0133
Entity Id	dag_1464215212735_0133_2
User	hive
Status	✓ SUCCEEDED
Start Time	26 May 2016 14:38:57
End Time	26 May 2016 14:39:12
Duration	14 secs

2.1.7.4 Task Details

[All DAGs](#) / DAG [hive_20160526043856_8998ec61-4418-4699-ae6f-909bf3ce6e7f:11]

[DAG Details](#)
[DAG Counters](#)
[Graphical View](#)
[All Vertices](#)
[All Tasks](#)
[All TaskAttempts](#)

Last refreshed at 26 May 2016 05:21:56
[Refresh](#)

First **1** Last - 1

Rows
25

Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration
13_000000	Reducer 14	✓ SUCCEEDED	100%	26 May 2016 14:39:11	26 May 2016 14:39:12	411 ms
09_000000	Reducer 6	✓ SUCCEEDED	100%	26 May 2016 14:39:10	26 May 2016 14:39:10	168 ms
08_000000	Reducer 5	✓ SUCCEEDED	100%	26 May 2016 14:39:10	26 May 2016 14:39:10	375 ms
06_000000	Reducer 4	✓ SUCCEEDED	100%	26 May 2016 14:39:10	26 May 2016 14:39:10	142 ms
01_000000	Reducer 9	✓ SUCCEEDED	100%	26 May 2016 14:39:08	26 May 2016 14:39:09	581 ms
05_000000	Reducer 3	✓ SUCCEEDED	100%	26 May 2016 14:39:08	26 May 2016 14:39:10	1 secs
12_000000	Map 13	✓ SUCCEEDED	100%	26 May 2016 14:38:58	26 May 2016 14:39:11	13 secs
11_000000	Map 1	✓ SUCCEEDED	100%	26 May 2016 14:38:58	26 May 2016 14:39:11	13 secs
10_000000	Map 12	✓ SUCCEEDED	100%	26 May 2016 14:38:58	26 May 2016 14:39:11	13 secs
03_000000	Map 7	✓ SUCCEEDED	100%	26 May 2016 14:38:58	26 May 2016 14:39:10	11 secs
02_000000	Map 10	✓ SUCCEEDED	100%	26 May 2016 14:38:58	26 May 2016 14:39:08	10 secs
04_000000	Map 2	✓ SUCCEEDED	100%	26 May 2016 14:38:57	26 May 2016 14:39:08	10 secs
07_000000	Map 11	✓ SUCCEEDED	100%	26 May 2016 14:38:57	26 May 2016 14:39:09	11 secs
00_000000	Map 8	✓ SUCCEEDED	100%	26 May 2016 14:38:57	26 May 2016 14:39:08	10 secs

2.1.8 Pig Query 2c

```

Scorings = LOAD '/tmp/Assignment_2/Scoring.csv' using PigStorage(',');
Scorings_raw = FILTER Scorings BY $0 != 'playerID';
Scoring = FOREACH Scorings_raw GENERATE $0 as playerid, $1 as Year, $3 as tmID,$4
as lgID,
$5 as pos, $16 as gwg;
grp_by_year = GROUP Scoring BY (Year);
max_year_gwg = FOREACH grp_by_year GENERATE group as year_grp,
MAX(Scoring.gwg) as max_gwg;
order_max_gwg = ORDER max_year_gwg by max_gwg desc;
limit_gwg = LIMIT order_max_gwg 1;
join_Scoring = JOIN Scoring by gwg,limit_gwg by max_gwg;
sort_join_Scoring = FOREACH join_Scoring GENERATE $0 as playerID, $1 as Year,
$2 as tmID, $3 as lgID, $4 as pos, $5 as gwg;
Teams = LOAD '/tmp/Assignment_2/Teams.csv' using PigStorage(',');
Teams_raw = FILTER Teams BY $0 > 0;
Team = FOREACH Teams_raw GENERATE $0 As Year, $2 as tmID, $18 as name;
join_team = JOIN sort_join_Scoring by (Year,tmID), Team by (Year,tmID);
sort_join_team = FOREACH join_team GENERATE $0 as playerID, $1 as Year,
$3 as lgID, $4 as pos, $5 as gwg, $8 as teamName;
Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');
Masters_raw = FILTER Masters BY $0 != 'playerID';
Master = FOREACH Masters_raw GENERATE $0 as playerid, $1 as coachid, $3 as

```

```

firstName,
$4 as lastName, $19 as birthYear, $20 as birthMon, $21 as birthDay, $22 as birthCountry;
join_Master = JOIN sort_join_team by playerID, Master by playerid;
sort_join_Master = FOREACH join_Master GENERATE $0 as playerid,
$8 as firstname, $9 as lastname;
awardplayers = load '/tmp/Assignment_2/AwardsPlayers.csv' using PigStorage(',');
awardplayers_raw = FILTER awardplayers BY $0 != 'playerID';
awardplayer = FOREACH awardplayers_raw GENERATE $0 as playerID, $1 as award, $2
as year;
group_awardplayer = group awardplayer by (playerID);
count_award = FOREACH group_awardplayer generate group as grp_yr ,
COUNT(awardplayer.award) as no_of_award;
award_final = FOREACH count_award GENERATE $0 as playerid , $1 as No_of_aways;
join_awardplayer = JOIN sort_join_Master by (playerid), award_final by (playerid);
final_awardplayer = FOREACH join_awardplayer GENERATE $0 as playerid, $1 as
firstname, $2 as lastname,
$4 as no_of_award;
order_final_awardplayer = ORDER final_awardplayer by no_of_award desc;
limit_final_awardplayer = LIMIT order_final_awardplayer 1;
final_solution = FOREACH limit_final_awardplayer GENERATE $0 as playerid , $1 as
firstname,
$2 as lastname;
Scoring_1 = FOREACH Scorings_raw GENERATE $0 as playerid, $1 as Year, $9 as Points;
Awardplayers = LOAD '/tmp/Assignment_2/AwardsPlayers.csv' using PigStorage(',');
Awardplayers_raw = FILTER Awardplayers BY $0 != 'playerID';
Awardplayer = FOREACH Awardplayers_raw GENERATE $0 as playerid, $1 as Award, $2
as Year;
join_scoring_awardplayer = JOIN Scoring_1 by (playerid,Year), Awardplayer by
(playerid,Year);
joint_Final = FOREACH join_scoring_awardplayer GENERATE $0 as playerid, $1 as Year,
$2 as Points,
$4 as Awardname;
join_final_answer = JOIN final_solution by (playerid),
joint_Final by (playerid);
format_join_final_1 = FOREACH join_final_answer GENERATE $6 as Awardname, $4 as
Year,
$5 as Points;
dump format_join_final_1;

```


2c - COMPLETED

Job ID job_1464220614001_0110

Started 2016-05-26 15:23

▼ Results

(Art Ross,1968,126)
(Second Team All-Star,1967,84)
(Hart,1968,126)
(First Team All-Star,1968,126)
(First Team All-Star,1969,99)
(Pearson,1970,152)
(Art Ross,1970,152)
(First Team All-Star,1970,152)
(First Team All-Star,1971,133)
(Art Ross,1971,133)
(First Team All-Star,1972,130)
(Art Ross,1972,130)
(Pearson,1972,130)
(Art Ross,1973,145)
(First Team All-Star,1973,145)
(Hart,1973,145)
(Second Team All-Star,1974,127)

2.1.8.2 Log

```
HadoopVersion  PigVersion      UserId StartedAt      FinishedAt      Features
2.7.1.2.3.2.0-2950  0.15.0.2.3.2.0-2950  yarn  2016-05-26 05:24:05  2016-05-26 05:29:01  HASH_JOIN,
```

Success!

Job Stats (time in seconds):

JobId	Maps	Reduces	MaxMapTime	MinMapTime	AvgMapTime	MedianMapTime	MaxReduceTime	MinReduceTime	MinReduceT
job_1464220614001_0112	1	1	7	7	7	7	2	2	group_by_y
job_1464220614001_0114	1	1	3	3	3	3	2	2	ap,ap_raw,
job_1464220614001_0117	1	1	2	2	2	2	3	3	order_max_
job_1464220614001_0118	2	1	4	3	3	3	3	3	ap1,final_
job_1464220614001_0120	1	1	2	2	2	2	2	2	order_max_
job_1464220614001_0122	1	1	2	2	2	2	3	3	order_max_
job_1464220614001_0124	2	1	3	3	3	3	3	3	final_s_li
job_1464220614001_0126	2	1	3	3	3	3	3	3	final_join
job_1464220614001_0128	2	1	5	5	5	5	3	3	join_s_lin
job_1464220614001_0130	2	1	3	3	3	2	2	2	final_join
job_1464220614001_0132	1	1	2	2	2	2	3	3	order_fina
job_1464220614001_0134	1	1	2	2	2	2	2	2	order_fina
job_1464220614001_0136	1	1	2	2	2	2	3	3	final_lim
job_1464220614001_0138	2	1	3	3	3	3	4	4	final_answ

Input(s):

Successfully read 45975 records (3273100 bytes) from: "/tmp/FIT5148assignment/Scoring.csv"
 Successfully read 2093 records (77202 bytes) from: "/tmp/FIT5148assignment/AwardsPlayers.csv"
 Successfully read 1529 records from: "/tmp/FIT5148assignment/Teams.csv"
 Successfully read 7770 records from: "/tmp/FIT5148assignment/Master.csv"

```

...
.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED
.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning ACCESSING_NON_EX
.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
.apache.pig.Main - Pig script completed in 4 minutes, 16 seconds and 467 milliseconds (256467 ms)

```

2.1.9 Comparison Table for 2c

	HIVE	PIG
No of Jobs	11	14
Maps	10	2
Reduces	6	1
Total Time	14sec	4 min 16 sec

2.2 Task 3

2.2.1 Hive Query 3 A

```

SELECT firstname AS First_Name,
       lastname AS Last_Name,
       concat(birthday,'/',birthmon,'/',birthyear) AS DOB,
       birthcountry AS Birth_Country,
       COUNT(award)AS Award_No
FROM hockey_award_coaches ac,
     hockey_master m
WHERE ac.coachid = m.coachid
GROUP BY firstname,
       lastname,
       concat(birthday,'/',birthmon,'/',birthyear),
       birthcountry
ORDER BY award_no DESC LIMIT 1;

```

2.2.1.1 Output

The screenshot shows a web-based interface for query results. At the top, a green progress bar indicates 100% completion. Below it, a header bar reads "Query Process Results (Status: SUCCEEDED)" with a "Save results..." dropdown on the right. There are two tabs: "Logs" and "Results", with "Results" being the active tab. Below the tabs is a "Filter columns..." input field. To the right of the filter are "previous" and "next" buttons. The main area displays a table with the following columns: first_name, last_name, dob, birth_country, and award_no. A single row of data is shown: Dick, Irvin, 19/7/1892, Canada, 9.

first_name	last_name	dob	birth_country	award_no
Dick	Irvin	19/7/1892	Canada	9

2.2.1.2 Log

100%

Query Process Results (Status: SUCCEEDED) Save results... ▼

Logs

Results

INFO : Session is already open

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0133)

INFO : Map 1: 0/1 Map 2: 0/1 Reducer 3: 0/1 Reducer 4: 0/1

INFO : Map 1: 0(+1)/1 Map 2: 0/1 Reducer 3: 0/1 Reducer 4: 0/1

INFO : Map 1: 0(+1)/1 Map 2: 0(+1)/1 Reducer 3: 0/1 Reducer 4: 0/1

INFO : Map 1: 1/1 Map 2: 0(+1)/1 Reducer 3: 0/1 Reducer 4: 0/1

INFO : Map 1: 1/1 Map 2: 1/1 Reducer 3: 0(+1)/1 Reducer 4: 0/1

INFO : Map 1: 1/1 Map 2: 1/1 Reducer 3: 1/1 Reducer 4: 0(+1)/1

INFO : Map 1: 1/1 Map 2: 1/1 Reducer 3: 1/1 Reducer 4: 1/1

2.2.1.3 DAG Details

DAG Details

Download data

Application Id	application_1464215212735_0133
Entity Id	dag_1464215212735_0133_4
User	hive
Status	✔ SUCCEEDED
Start Time	26 May 2016 14:44:47
End Time	26 May 2016 14:44:52
Duration	5 secs

2.2.1.4 Task Details

The screenshot shows the 'All Tasks' tab for a specific DAG. The table lists tasks with their indices, vertex names, statuses, progress, start/end times, and durations. All tasks shown are 'SUCCEEDED'.

Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration
03_000000	Reducer 4	SUCCEEDED	100%	26 May 2016 14:44:52	26 May 2016 14:44:52	307 ms
02_000000	Reducer 3	SUCCEEDED	100%	26 May 2016 14:44:52	26 May 2016 14:44:52	207 ms
01_000000	Map 2	SUCCEEDED	100%	26 May 2016 14:44:47	26 May 2016 14:44:52	4 secs
00_000000	Map 1	SUCCEEDED	100%	26 May 2016 14:44:47	26 May 2016 14:44:51	4 secs

2.2.2 Pig Query 3A

```
Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');
```

```
Masters_raw = FILTER Masters BY $0 != 'playerID';
```

```
Master = FOREACH Masters_raw GENERATE $1 as coachid, $3 as firstName, $4 as lastName, $19 as birthYear,
```

```
$20 as birthMon, $21 as birthDay, $22 as birthCountry;
```

```
AwardCoaches = LOAD '/tmp/Assignment_2/AwardsCoaches.csv' using PigStorage(',');
```

```
AwardCoaches_raw = FILTER AwardCoaches BY $0 != 'coachID';
```

```
AwardCoach = FOREACH AwardCoaches_raw GENERATE $0 as coachid, $1 as award;
```

```
grp_by_coachid = GROUP AwardCoach BY (coachid);
```

```
count_award = FOREACH grp_by_coachid GENERATE group as coachid_grp,
```

```
COUNT(AwardCoach.award) as award;
```

```
join_AwardCoach = JOIN count_award BY (coachid_grp), Master BY (coachid);
```

```
filter_join_AwardCoach = FOREACH join_AwardCoach GENERATE $3 as firstname,
```

```
$4 as lastname, $5 as birthyear, $6 as birthmonth, $7 as birthday, $8 as birthCountry,
```

```
$1 as awardcount;
```

```
order_answer = ORDER filter_join_AwardCoach by awardcount desc;
```

```
order_limit = LIMIT order_answer 1;
```

```
DUMP order_limit;
```

2.2.2.1 Output

▼ Results

Download

(Dick,Irvin,1892,7,19,Canada,9)

2.2.2.2 Log

```

2016-05-26 06:20:21,398 [main] INFO org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:

HadoopVersion PigVersion      UserId StartedAt      FinishedAt      Features
2.7.1.2.3.2.0-2950      0.15.0.2.3.2.0-2950      yarn      2016-05-26 06:18:43      2016-05-26 06:20:21      HASH_JOIN,GR

Success!

Job Stats (time in seconds):
JobId  Maps  Reduces MaxMapTime  MinMapTime  AvgMapTime  MedianMapTime  MaxReduceTime  MinReduceTime
job_1464241803099_0036  1      1      2      2      2      2      2      2      AwardCoach,A
job_1464241803099_0037  2      1      4      4      4      5      5      5      Master,Maste
job_1464241803099_0038  1      1      2      2      2      2      2      2      order_answer
job_1464241803099_0039  1      1      2      2      2      2      2      2      order_answer
job_1464241803099_0040  1      1      2      2      2      4      4      4      order_answer

Input(s):
Successfully read 88 records (3469 bytes) from: "/tmp/Assignment_2/AwardsCoaches.csv"
Successfully read 7770 records from: "/tmp/Assignment_2/Master.csv"

Output(s):
Successfully stored 1 records (40 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp1798016033/tmp-1221639234"

Counters:
Total records written : 1
Total bytes written : 40
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

```

```

2016-05-26 06:20:22,578 [main] INFO org.apache.pig.Main - Pig script completed in 1 minute, 41 seconds and 927 mill:

```

2.2.3 Comparison table for 3A

	HIVE	PIG
No of Jobs	2	5
Maps	2	2
Reduces	2	1
Total Time	5sec	1min 41 sec

2.2.4 Hive Query 3B

```

SELECT firstname AS First_Name,
       lastname AS Last_Name,
       c.year AS YEAR,
       c.g AS Games,
       c.w AS Wins,
       c.l AS losses,
       c.t AS Ties
FROM hockey_coaches c,
     hockey_master m
WHERE c.coachid=m.coachid
AND c.w IN
      (SELECT max(w)
       FROM hockey_coaches);

```

2.2.4.1 Output

100%						
Query Process Results (Status: SUCCEEDED)						
Save results... ▼						
<div> <div>Logs</div> <div>Results</div> </div>						
<div> <div>Filter columns...</div> <div>previous</div> <div>next</div> </div>						
first_name	last_name	year	games	wins	losses	ties
Scotty	Bowman	1995	82	62	13	7

2.2.4.2 Log

100%

Query Process Results (Status: SUCCEEDED) Save results... ▼

Logs

Results

INFO : Session is already open

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0133)

INFO : Map 1: 0/1 Map 2: 0/1 Map 3: 0/1 Reducer 4: 0/1

INFO : Map 1: 0(+1)/1 Map 2: 0/1 Map 3: 0/1 Reducer 4: 0/1

INFO : Map 1: 0(+1)/1 Map 2: 0/1 Map 3: 0(+1)/1 Reducer 4: 0/1

INFO : Map 1: 0(+1)/1 Map 2: 0(+1)/1 Map 3: 0(+1)/1 Reducer 4: 0/1

INFO : Map 1: 1/1 Map 2: 0(+1)/1 Map 3: 0(+1)/1 Reducer 4: 0/1

INFO : Map 1: 1/1 Map 2: 0(+1)/1 Map 3: 1/1 Reducer 4: 0/1

INFO : Map 1: 1/1 Map 2: 0(+1)/1 Map 3: 1/1 Reducer 4: 0(+1)/1

INFO : Map 1: 1/1 Map 2: 0(+1)/1 Map 3: 1/1 Reducer 4: 1/1

INFO : Map 1: 1/1 Map 2: 1/1 Map 3: 1/1 Reducer 4: 1/1

2.2.4.3 DAG Detail

DAG Details

Download data

Application Id	application_1464215212735_0133
Entity Id	dag_1464215212735_0133_5
User	hive
Status	SUCCEEDED
Start Time	26 May 2016 14:47:13
End Time	26 May 2016 14:47:21
Duration	7 secs

2.2.4.4 Task Details

Home All DAGs / DAG [hive_20160526044713_ffcab237-aa1d-499c-9636-df1a7a30a26f:14]

DAG Details DAG Counters Graphical View All Vertices **All Tasks** All TaskAttempts

Last refreshed at 26 May 2016 05:28:46 Refresh

RegEx or Column1, Column2...:RegEx Search First 1 Last - 1 Rows 25

Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration
01_000000	Reducer 4	✓ SUCCEEDED	100%	26 May 2016 14:47:20	26 May 2016 14:47:20	295 ms
03_000000	Map 2	✓ SUCCEEDED	100%	26 May 2016 14:47:13	26 May 2016 14:47:21	7 secs
00_000000	Map 3	✓ SUCCEEDED	100%	26 May 2016 14:47:13	26 May 2016 14:47:20	6 secs
02_000000	Map 1	✓ SUCCEEDED	100%	26 May 2016 14:47:13	26 May 2016 14:47:19	5 secs

2.2.5 Pig Query 3B

```
Coaches = LOAD '/tmp/Assignment_2/Coaches.csv' using PigStorage(',');
Coaches_raw = FILTER Coaches BY $0 != 'coachID';
Coach = FOREACH Coaches_raw GENERATE $0 as coachid, $1 as Year, $6 as Games, $7 as Wins,
$8 as Losses, $9 as Ties;
grp_by_year = GROUP Coach BY (Year);
max_points = FOREACH grp_by_year GENERATE group as year_grp,
MAX(Coach.Wins) as max_Wins;
join_max_points = JOIN max_points by (year_grp,max_Wins), Coach by (Year, Wins);
max_player_points = FOREACH join_max_points GENERATE $0 as Year, $1 as Wins,
$2 as coachid, $4 as Games, $6 as Losses, $7 as Ties;
Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');
Masters_raw = FILTER Masters BY $1 != 'coachID';
Master = FOREACH Masters_raw GENERATE $1 as coachid, $3 as firstName, $4 as lastName;
join_coach_master = JOIN max_player_points by coachid, Master by coachid;
max_player_info = FOREACH join_coach_master GENERATE $0 as year, $7 as firstname,
$8 as lastname, $3 as Games, $1 as Wins, $4 as Loses, $5 as Ties;
order_scoring = ORDER max_player_info by Wins desc;
order_limit = LIMIT order_scoring 1;
dump order_limit;
```

2.2.5.1 Output

▼ Results
[Download](#)

(1995,Scotty,Bowman,82,62.0,13,7)

2.2.5.2 Log

```

HadoopVersion  PigVersion  UserId  StartedAt  FinishedAt  Features
2.7.1.2.3.2.0-2950  0.15.0.2.3.2.0-2950  yarn  2016-05-26 06:24:58  2016-05-26 06:27:02  HASH_JOIN,GR

Success!

Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime  MinMapTime  AvgMapTime  MedianMapTime  MaxReduceTime  MinReduceTime
job_1464241803099_0043  1  1  3  3  3  3  3  3
job_1464241803099_0044  2  1  3  3  3  2  2  2
job_1464241803099_0045  2  1  3  3  3  3  3  3
job_1464241803099_0046  1  1  3  3  3  3  3  3
job_1464241803099_0047  1  1  2  2  2  2  2  2
job_1464241803099_0048  1  1  2  2  2  2  2  2

Input(s):
Successfully read 1824 records (81287 bytes) from: "/tmp/Assignment_2/Coaches.csv"
Successfully read 7770 records from: "/tmp/Assignment_2/Master.csv"

Output(s):
Successfully stored 1 records (46 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp330812894/tmp2112797200"

Counters:
Total records written : 1
Total bytes written : 46
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

```

```

2016-05-26 06:27:03,640 [main] INFO  org.apache.pig.Main - Pig script completed in 2 minutes, 8 seconds and 263 mill:

```

2.2.6 Comparison Table for 3B

	HIVE	PIG
No of Jobs	3	6
Maps	3	1
Reduces	1	2
Total Time	7sec	2min 8 sec

2.2.7 Hive Query 3C

```
SELECT firstname AS First_Name,  
        lastname AS Last_Name,  
        c.year AS YEAR,  
        c.g AS Games,  
        c.w AS Wins,  
        c.l AS losses,  
        c.t AS Ties,  
        count(ac.award) AS COUNT_AWARDS  
FROM hockey_coaches c,  
        hockey_master m,  
        hockey_award_coaches ac  
WHERE c.coachid=m.coachid  
        AND c.coachid=ac.coachid  
        AND c.w IN  
        (SELECT max(w)  
        FROM hockey_coaches)  
GROUP BY firstname,  
        lastname,  
        c.year,  
        c.g,  
        c.w,  
        c.l,  
        c.t;
```

2.2.7.1 Output

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▼

Logs

Results

previous
next

first_name	last_name	year	games	wins	losses	ties	count_awards
Scotty	Bowman	1995	82	62	13	7	2

2.2.7.2 Log

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▼

Logs

Results

INFO : Session is already open
INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0133)

INFO : Map 1: 0/1 Map 2: 0/1 Map 4: 0/1 Map 5: 0/1 Reducer 3: 0/1 Reducer 6: 0/1
INFO : Map 1: 0(+1)/1 Map 2: 0/1 Map 4: 0/1 Map 5: 0/1 Reducer 3: 0/1 Reducer 6: 0/1
INFO : Map 1: 0(+1)/1 Map 2: 0/1 Map 4: 0(+1)/1 Map 5: 0/1 Reducer 3: 0/1 Reducer 6: 0/1
INFO : Map 1: 0(+1)/1 Map 2: 0(+1)/1 Map 4: 0(+1)/1 Map 5: 0/1 Reducer 3: 0/1 Reducer 6: 0/1
INFO : Map 1: 0(+1)/1 Map 2: 0(+1)/1 Map 4: 0(+1)/1 Map 5: 0(+1)/1 Reducer 3: 0/1 Reducer 6: 0/1

2.2.7.3 DAG Details

DAG Details	
Download data	
Application Id	application_1464215212735_0133
Entity Id	dag_1464215212735_0133_6
User	hive
Status	✓ SUCCEEDED
Start Time	26 May 2016 14:50:17
End Time	26 May 2016 14:50:26
Duration	9 secs

2.2.7.4 Task Details

All DAGs / DAG [hive_20160526045016_3579d709-88c3-4571-ae01-7298b5c56189:15]								
DAG Details DAG Counters Graphical View All Vertices All Tasks All TaskAttempts								
Last refreshed at 26 May 2016 05:31:55 Refresh								
RegEx or Column1, Column2...:RegEx <input type="text"/> Search								
First 1 Last - 1 Rows 25								
Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration		
05_000000	Reducer 3	✓ SUCCEEDED	100%	26 May 2016 14:50:26	26 May 2016 14:50:26	432 ms		
02_000000	Reducer 6	✓ SUCCEEDED	100%	26 May 2016 14:50:24	26 May 2016 14:50:24	361 ms		
04_000000	Map 2	✓ SUCCEEDED	100%	26 May 2016 14:50:17	26 May 2016 14:50:26	8 secs		
01_000000	Map 5	✓ SUCCEEDED	100%	26 May 2016 14:50:17	26 May 2016 14:50:24	6 secs		
00_000000	Map 4	✓ SUCCEEDED	100%	26 May 2016 14:50:17	26 May 2016 14:50:22	5 secs		
03_000000	Map 1	✓ SUCCEEDED	100%	26 May 2016 14:50:17	26 May 2016 14:50:22	4 secs		

2.2.8 Pig Query 3C

```

Coaches = LOAD '/tmp/Assignment_2/Coaches.csv' using PigStorage(',');

Coaches_raw = FILTER Coaches BY $0 != 'coachID';

Coach = FOREACH Coaches_raw GENERATE $0 as coachid, $1 as Year, $6 as Games, (int) $7 as Wins,
$8 as Losses, $9 as Ties;

grp_by_year = GROUP Coach BY (Year);

max_points = FOREACH grp_by_year GENERATE group as year_grp,
MAX(Coach.Wins) as max_Wins;

join_max_points = JOIN max_points by (year_grp,max_Wins), Coach by (Year, Wins);

max_player_points = FOREACH join_max_points GENERATE $0 as Year, $1 as Wins,
$2 as coachid, $4 as Games, $6 as Losses, $7 as Ties;

Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');

Masters_raw = FILTER Masters BY $1 != 'coachID';

Master = FOREACH Masters_raw GENERATE $1 as coachid, $3 as firstName, $4 as lastName;

join_coach_master = JOIN max_player_points by coachid, Master by coachid;

max_player_info = FOREACH join_coach_master GENERATE $0 as year, $2 as coachid, $7 as
firstname,
$8 as lastname, $3 as Games, $1 as Wins, $4 as Loses, $5 as Ties;

AwardCoaches = LOAD '/tmp/Assignment_2/AwardsCoaches.csv' using PigStorage(',');

AwardCoaches_raw = FILTER AwardCoaches BY $0 != 'coachID';

AwardCoach = FOREACH AwardCoaches_raw GENERATE $0 as coachid;

grp_by_AwardCoach = GROUP AwardCoach BY (coachid);

Grp_count_award = FOREACH grp_by_AwardCoach GENERATE group as coachid_grp,
COUNT(AwardCoach.coachid) as Count_Award;

join_coach_master_award = JOIN max_player_info by coachid, Grp_count_award by coachid_grp;

join_coach_master_award_1 = FOREACH join_coach_master_award GENERATE $0 as year,
$2 as firstname, $3 as lastname, $4 as Games, $5 as Wins, $6 as Loses, $7 as Ties, $9 as Awards;

order_join_coach_master_award = ORDER join_coach_master_award_1 by Wins desc;

order_limit = LIMIT order_join_coach_master_award 1;

Dump order_limit;

```

2.2.8.1 Output

▼ Results
[Download](#)

(1995,Scotty,Bowman,82,62,13,7,2)

2.2.8.2 Log

```

HadoopVersion  PigVersion      UserId  StartedAt      FinishedAt      Features
2.7.1.2.3.2.0-2950  0.15.0.2.3.2.0-2950  yarn    2016-05-26 06:29:50  2016-05-26 06:32:22  HASH_JOIN,GRI

Success!

Job Stats (time in seconds):
JobId  Maps    Reduces  MaxMapTime    MinMapTime    AvgMapTime    MedianMapTime    MaxReduceTime    MinReduceTime
job_1464241803099_0050  1      1        3      3        3      3        3      3      AwardCoach,A
job_1464241803099_0051  1      1        4      4        4      4        3      3      Coach,Coache:
job_1464241803099_0052  2      1        3      3        3      3        2      2      join_max_poi
job_1464241803099_0053  2      1        3      2        3      3        2      2      Master,Mastei
job_1464241803099_0054  2      1        3      2        3      3        4      4      join_coach_m
job_1464241803099_0055  1      1        2      2        2      2        2      2      order_join_c
job_1464241803099_0056  1      1        2      2        2      2        2      2      order_join_c
job_1464241803099_0057  1      1        2      2        2      2        3      3      order_join_c

Input(s):
Successfully read 88 records (3469 bytes) from: "/tmp/Assignment_2/AwardsCoaches.csv"
Successfully read 1824 records (81287 bytes) from: "/tmp/Assignment_2/Coaches.csv"
Successfully read 7770 records from: "/tmp/Assignment_2/Master.csv"

Output(s):
Successfully stored 1 records (41 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp748822897/tmp-421113721"

```

```
2016-05-26 06:32:24,132 [main] INFO org.apache.pig.Main - Pig script completed in 2 minutes, 36 seconds and 168 mil
```

2.2.9 Comparison table for 3C

	HIVE	PIG
No of Jobs	4	8
Maps	4	1
Reduces	2	2
Total Time	9 sec	2min 36 sec

2.3 Task 4

2.3.1 Hive Query 4A

```
SELECT t.name AS Teamname,
```

```
       sum(s.pts) AS Points,
```

```
       sum(s.g) AS Goals,
```

```
       sum(s.a) AS assists
```

```
FROM hockey_scoring s
```

```
JOIN hockey_teams t ON (s.tmid = t.tmid
```

```
       AND s.year = t.year
```

```
       AND s.lgid = t.lgid)
```

```
GROUP BY t.name
```

```
ORDER BY Points DESC;
```

2.3.1.1 Output

100%			
Query Process Results (Status: SUCCEEDED)			
Save results... ▾			
Logs	Results		
Filter columns...		previous	next
teamname	points	goals	assists
Montreal Canadiens	50813	20447	30366
Boston Bruins	48677	19067	29610
New York Rangers	46805	18031	28774
Toronto Maple Leafs	46515	18154	28361
Detroit Red Wings	46253	17798	28455
Pittsburgh Penguins	32257	11969	20288
Philadelphia Flyers	32012	12020	19992

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▼

Logs

Results

Filter columns...

previous

next

teamname	points	goals	assists
Les Canadiens	59	59	0
Toronto Tecumsehs	59	59	0
Montreal Shamrocks	52	52	0
Finland	9	4	5

2.3.1.2 Log

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▼

Logs

Results

INFO : Session is already open

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0133)

INFO : Map 1: 0/1 Map 4: 0/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 0(+1)/1 Map 4: 0/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 0(+1)/1 Map 4: 0(+1)/1 Reducer 2: 0/1 Reducer 3: 0/1


INFO : Map 1: 0(+1)/1 Map 4: 1/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 1/1 Map 4: 1/1 Reducer 2: 0(+1)/1 Reducer 3: 0/1






INFO : Map 1: 1/1 Map 4: 1/1 Reducer 2: 1/1 Reducer 3: 0(+1)/1

INFO : Map 1: 1/1 Map 4: 1/1 Reducer 2: 1/1 Reducer 3: 1/1

2.3.1.3 DAG Details

DAG Details	
Download data	
Application Id	application_1464215212735_0133
Entity Id	dag_1464215212735_0133_7
User	hive
Status	 SUCCEEDED
Start Time	26 May 2016 14:52:46
End Time	26 May 2016 14:52:51
Duration	5 secs

2.3.1.4 Task Details

All DAGs / DAG [hive_20160526045245_a29ca826-e00e-4116-a865-9c51eabea80f:16]							
DAG Details		DAG Counters	Graphical View	All Vertices	All Tasks	All TaskAttempts	
		Last refreshed at 26 May 2016 05:34:48				Refresh	
<input type="text" value="RegEx or Column1, Column2...:RegEx"/>		Search		First	1	Last - 1	Rows 25
Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration	
03_000000	Reducer 3	 SUCCEEDED	100%	26 May 2016 14:52:51	26 May 2016 14:52:51	352 ms	
02_000000	Reducer 2	 SUCCEEDED	100%	26 May 2016 14:52:51	26 May 2016 14:52:51	205 ms	
01_000000	Map 1	 SUCCEEDED	100%	26 May 2016 14:52:46	26 May 2016 14:52:51	4 secs	
00_000000	Map 4	 SUCCEEDED	100%	26 May 2016 14:52:46	26 May 2016 14:52:50	3 secs	

2.3.2 Pig Query 4A

```

Scorings = LOAD '/tmp/Assignment_2/Scoring.csv' using PigStorage(',');

Scorings_raw = FILTER Scorings BY $0 != 'playerID';

Scoring = FOREACH Scorings_raw GENERATE $1 as Year, $3 as tmID,$4 as lgID, $7 as g, $8 as a, $9 as Pts;

Teams = LOAD '/tmp/Assignment_2/Teams.csv' using PigStorage(',');

Teams_raw = FILTER Teams BY $0 > 0;

Team = FOREACH Teams_raw GENERATE $0 as Year, $1 as lgID, $2 as tmID, $18 as name;

team_scoring = JOIN Scoring BY (lgID,tmID,Year), Team BY (lgID,tmID,Year);

team_final = FOREACH team_scoring GENERATE $3 as G,$4 as A,$5 as Pts, $9 as team_name;

grp_team_final = GROUP team_final BY (team_name);

total_scoring = FOREACH grp_team_final GENERATE group as team_name, SUM(team_final.Pts) as Pts,

SUM(team_final.G) as G, SUM(team_final.A) as A;

order_total_scoring = ORDER total_scoring by Pts desc;

dump order_total_scoring;

```

2.3.2.1 Output

Results	Download
(Montreal Canadiens,50813.0,20447.0,30366.0)	
(Boston Bruins,48677.0,19067.0,29610.0)	
(New York Rangers,46805.0,18031.0,28774.0)	
(Toronto Maple Leafs,46515.0,18154.0,28361.0)	
(Detroit Red Wings,46253.0,17798.0,28455.0)	
(Pittsburgh Penguins,32257.0,11969.0,20288.0)	
(Philadelphia Flyers,32012.0,12020.0,19992.0)	
(Los Angeles Kings,30148.0,11260.0,18888.0)	
(St. Louis Blues,29500.0,11048.0,18452.0)	
(Chicago Black Hawks,29495.0,11473.0,18022.0)	
(Buffalo Sabres,28723.0,10825.0,17898.0)	
(Edmonton Oilers,28347.0,10587.0,17760.0)	
(Vancouver Canucks,27573.0,10344.0,17229.0)	
(New York Islanders,26846.0,10072.0,16774.0)	
(Washington Capitals,24982.0,9377.0,15605.0)	
(Calgary Flames,22221.0,8323.0,13898.0)	
(Winnipeg Jets,19330.0,7253.0,12077.0)	
(New Jersey Devils,19130.0,7077.0,12053.0)	
(Quebec Nordiques,18576.0,6899.0,11677.0)	
(Minnesota North Stars,17745.0,6690.0,11055.0)	
(Chicago Blackhawks,16246.0,6023.0,10223.0)	
(Ottawa Senators,14962.0,6567.0,8395.0)	
(San Jose Sharks,12813.0,4745.0,8068.0)	
(Hartford Whalers,12515.0,4704.0,7811.0)	
(Tampa Bay Lightning,11489.0,4262.0,7227.0)	
(Dallas Stars,11085.0,4072.0,7013.0)	
(Colorado Avalanche,10659.0,3893.0,6766.0)	
(Florida Panthers,10029.0,3737.0,6292.0)	
(Phoenix Coyotes,9433.0,3459.0,5974.0)	

2.3.2.2 Log

```

HadoopVersion  PigVersion      UserId  StartedAt      FinishedAt      Features
2.7.1.2.3.2.0-2950  0.15.0.2.3.2.0-2950  yarn    2016-05-26 06:34:28  2016-05-26 06:35:49  HASH_JOIN,GRU

Success!

Job Stats (time in seconds):
JobId  Maps  Reduces  MaxMapTime  MinMapTime  AvgMapTime  MedianMapTime  MaxReduceTime  MinReduceTime
job_1464241803099_0059  2      1        3          3          3          3          3          3          Scoring,Scor:
job_1464241803099_0060  1      1        3          3          3          2          2          2          grp_team_fini
job_1464241803099_0061  1      1        2          2          2          2          2          2          order_total_:
job_1464241803099_0062  1      1        2          2          2          2          2          2          order_total_:

Input(s):
Successfully read 1529 records from: "/tmp/Assignment_2/Teams.csv"
Successfully read 45975 records from: "/tmp/Assignment_2/Scoring.csv"

Output(s):
Successfully stored 104 records (5255 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp-1973846067/tmp-682756:

Counters:
Total records written : 104
Total bytes written : 5255
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

```

```

2016-05-26 06:35:50,615 [main] INFO org.apache.pig.Main - Pig script completed in 1 minute, 24 seconds and 701 mill:

```

2.3.3 Comparison Table for 4a

	HIVE	PIG
No of Jobs	2	4
Maps	2	1
Reduces	2	2
Total Time	5sec	1min 24 sec

2.3.4 Hive Query 4 B

```
SELECT t.name AS Team_name,  
       m.firstname AS First_Name,  
       m.lastname AS Last_Name,  
       s.year AS YEAR,  
       s.pts AS Points,  
       s.g AS Goals,  
       s.a AS Assist  
FROM hockey_scoring s  
JOIN  
(SELECT tmid,  
       max(pts) pts  
FROM hockey_scoring  
GROUP BY tmid) b ON (s.tmid = b.tmid  
                     AND s.pts = b.pts)  
JOIN hockey_master m ON (m.playerid = s.playerid)  
JOIN hockey_teams t ON (t.tmid = s.tmid  
                       AND t.year = s.year)  
ORDER BY Team_name;
```

2.3.4.1 Output

100%						
Query Process Results (Status: SUCCEEDED)						
Save results... ▾						
<div> <div>Logs</div> <div>Results</div> </div>						
<div> <div>Filter columns...</div> <div>previous</div> <div>next</div> </div>						
team_name	first_name	last_name	year	points	goals	assist
228th Battalion	Eddie	Oatman	1916	22	17	5
Alberta Oilers	Jim	Harrison	1972	86	39	47
Anaheim Ducks	Corey	Perry	2010	98	50	48
Atlanta Flames	Bob	MacMillan	1978	108	37	71
Atlanta Thrashers	Marian	Hossa	2006	100	43	57
Birmingham Bulls	Mark	Napier	1976	96	60	36
Boston Bruins	Phil	Esposito	1970	152	76	76
Brooklyn Americans	Tom	Anderson	1941	41	12	29
Buffalo Sabres	Pat	LaFontaine	1992	148	53	95
Calgary Cowboys	Danny	Lawson	1975	96	44	52
Calgary Flames	Kent	Nilsson	1980	131	49	82
Calgary Tigers	Harry	Oliver	1923	34	22	12
California Golden Seals	Joey	Johnston	1973	67	27	40
Carolina Hurricanes	Eric	Staal	2005	100	45	55

Toronto Blueshirts	Jack	Walker	1913	36	20	16
Toronto Maple Leafs	Doug	Gilmour	1992	127	32	95
Toronto Ontarios	Jack	McDonald	1913	35	27	8
Toronto St. Patricks	Babe	Dye	1924	46	38	8
Toronto Tecumsehs	Harry	Smith	1912	14	14	0
Toronto Toros	Vaclav	Nedomansky	1975	98	56	42
Vancouver Blazers	Bryan	Campbell	1973	89	27	62
Vancouver Canucks	Henrik	Sedin	2009	112	29	83
Vancouver Maroons	Mickey	MacKay	1922	40	28	12
Vancouver Maroons	Mickey	MacKay	1924	33	27	6
Vancouver Millionaires	Gord	Roberts	1916	53	43	10
Victoria Aristocrats	Albert	Kerr	1913	31	20	11
Victoria Aristocrats	Tommy	Dunderdale	1919	33	26	7
Victoria Cougars	Frank	Fredrickson	1924	30	22	8
Victoria Cougars	Frank	Fredrickson	1922	55	39	16
Washington Capitals	Dennis	Maruk	1981	136	60	76
Winnipeg Jets	Bobby	Hull	1974	142	77	65
Winnipeg Jets	Teemu	Selanne	1992	132	76	56
Winnipeg Jets	Blake	Wheeler	2011	64	17	47

2.3.4.2 Log

100%

Query Process Results (Status: SUCCEEDED)
Save results... ▾

Logs

Results

INFO : Session is already open

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464215212735_0133)

INFO : Map 1: 0/1 Map 4: 0/1 Map 5: 0/1 Map 6: 0/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 0/1 Map 4: 0/1 Map 5: 0/1 Map 6: 0(+1)/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 0(+1)/1 Map 4: 0(+1)/1 Map 5: 0(+1)/1 Map 6: 0(+1)/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 0(+1)/1 Map 4: 0(+1)/1 Map 5: 0(+1)/1 Map 6: 1/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 0(+1)/1 Map 4: 0(+1)/1 Map 5: 1/1 Map 6: 1/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 1/1 Map 4: 1/1 Map 5: 1/1 Map 6: 1/1 Reducer 2: 0/1 Reducer 3: 0/1

INFO : Map 1: 1/1 Map 4: 1/1 Map 5: 1/1 Map 6: 1/1 Reducer 2: 0(+1)/1 Reducer 3: 0/1

INFO : Map 1: 1/1 Map 4: 1/1 Map 5: 1/1 Map 6: 1/1 Reducer 2: 1/1 Reducer 3: 0/1

INFO : Map 1: 1/1 Map 4: 1/1 Map 5: 1/1 Map 6: 1/1 Reducer 2: 1/1 Reducer 3: 0(+1)/1

INFO : Map 1: 1/1 Map 4: 1/1 Map 5: 1/1 Map 6: 1/1 Reducer 2: 1/1 Reducer 3: 1/1

2.3.4.3 DAG Details

DAG Details	
<div style="background-color: #00bcd4; color: white; padding: 5px; display: inline-block;"> Download data </div>	
Application Id	application_1464215212735_0133
Entity Id	dag_1464215212735_0133_8
User	hive
Status	✔ SUCCEEDED
Start Time	26 May 2016 14:55:39
End Time	26 May 2016 14:55:47
Duration	7 secs

2.3.4.4 Task Detail

Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration
05_000000	Reducer 3	✓ SUCCEEDED	100%	26 May 2016 14:55:46	26 May 2016 14:55:47	516 ms
04_000000	Reducer 2	✓ SUCCEEDED	100%	26 May 2016 14:55:46	26 May 2016 14:55:46	753 ms
01_000000	Map 4	✓ SUCCEEDED	100%	26 May 2016 14:55:40	26 May 2016 14:55:46	6 secs
02_000000	Map 5	✓ SUCCEEDED	100%	26 May 2016 14:55:40	26 May 2016 14:55:45	5 secs
03_000000	Map 1	✓ SUCCEEDED	100%	26 May 2016 14:55:40	26 May 2016 14:55:46	6 secs
00_000000	Map 6	✓ SUCCEEDED	100%	26 May 2016 14:55:40	26 May 2016 14:55:45	5 secs

2.3.5 Pig Query 4B

```
Scorings = LOAD '/tmp/Assignment_2/Scoring.csv' using PigStorage(',');
```

```
Scorings_raw = FILTER Scorings BY $0 != 'playerID';
```

```
Scoring = FOREACH Scorings_raw GENERATE $0 as playerid, $1 as Year, $3 as tmID, $7 as g,
$8 as a, $9 as Pts;
```

```
grp_by_tmID = GROUP Scoring BY (tmID);
```

```
sum_points = FOREACH grp_by_tmID GENERATE group as tmID_grp,
```

```
MAX(Scoring.Pts) as max_points;
```

```
join_max_points = JOIN sum_points by (tmID_grp, max_points), Scoring by (tmID, Pts);
```

```
max_player_points = FOREACH join_max_points GENERATE $0 as tmID, $1 as Points,
$2 as playerid, $3 as Year, $5 as Goals, $6 as Assists;
```

```
Teams = LOAD '/tmp/Assignment_2/Teams.csv' using PigStorage(',');
```

```
Teams_raw = FILTER Teams BY $0 > 0;
```

```
Team = FOREACH Teams_raw GENERATE $0 As Year, $2 as tmID, $18 as name;
```

```
join_player = JOIN max_player_points by (Year,tmID), Team by (Year,tmID);
```

```
join_team_Scoring = FOREACH join_player GENERATE $0 as tmID, $1 as Points,
```

```
$2 as playerid, $3 as Year, $4 as Goals, $5 as Assists, $8 as TeamName;
```

```
Masters = LOAD '/tmp/Assignment_2/Master.csv' using PigStorage(',');
```

```
Masters_raw = FILTER Masters BY $0 != 'playerID';
```

```
Master = FOREACH Masters_raw GENERATE $0 as playerid, $3 as firstName, $4 as lastName;
```

Final_answer = JOIN join_team_Scoring by playerid, Master by playerid;

Formatted_Final_Answer = FOREACH Final_answer GENERATE \$6 as TeamName, \$8 as FirstName, \$9 as LastName,

\$3 as Year, \$1 as Points, \$4 as Goals, \$5 as Assists;

order_Formatted_Final_Answer = ORDER Formatted_Final_Answer by TeamName;

Dump order_Formatted_Final_Answer;

2.3.5.1 Output

Results	Download
(228th Battalion,Eddie,Oatman,1916,22.0,17,5) (Alberta Oilers,Jim,Harrison,1972,86.0,39,47) (Anaheim Ducks,Corey,Perry,2010,98.0,50,48) (Atlanta Flames,Bob,MacMillan,1978,108.0,37,71) (Atlanta Thrashers,Marian,Hossa,2006,100.0,43,57) (Birmingham Bulls,Mark,Napier,1976,96.0,60,36) (Boston Bruins,Phil,Esposito,1970,152.0,76,76) (Brooklyn Americans,Tom,Anderson,1941,41.0,12,29) (Buffalo Sabres,Pat,LaFontaine,1992,148.0,53,95) (Calgary Cowboys,Danny,Lawson,1975,96.0,44,52) (Calgary Flames,Kent,Nilsson,1980,131.0,49,82) (Calgary Tigers,Harry,Oliver,1923,34.0,22,12) (California Golden Seals,Joey,Johnston,1973,67.0,27,40) (Carolina Hurricanes,Eric,Staal,2005,100.0,45,55) (Chicago Blackhawks,Denis,Savard,1987,131.0,44,87) (Chicago Cougars,Bob,Sicinski,1972,88.0,25,63) (Cincinnati Stingers,Robbie,Ftorek,1978,116.0,39,77) (Cleveland Barons,Dennis,Maruk,1976,78.0,28,50) (Cleveland Crusaders,Ron,Ward,1975,82.0,32,50) (Cobalt Silver Kings,Harry,Smith,1909,28.0,28,0) (Colorado Avalanche,Joe,Sakic,1995,120.0,51,69) (Colorado Rockies,Wilf,Paement,1977,87.0,31,56) (Columbus Blue Jackets,Rick,Nash,2008,79.0,40,39) (Czechoslovakia,Vladislav,Vlcek,1977,9.0,2,7) (Dallas Stars,Mike,Modano,1993,93.0,50,43) (Denver Spurs/Ottawa Civics,Ralph,Backstrom,1975,50.0,21,29) (Detroit Cougars,Carson,Cooper,1929,36.0,18,18) (Detroit Falcons,Ebbie,Goodfellow,1930,48.0,25,23) (Detroit Red Wings,Steve,Yzerman,1988,155.0,65,90)	

2.3.5.2

2.3.5.3 Log

HadoopVersion	PigVersion	UserId	StartedAt	FinishedAt	Features				
2.7.1.2.3.2.0-2950	0.15.0.2.3.2.0-2950	yarn	2016-05-26 06:38:14	2016-05-26 06:40:25	HASH_JOIN,GR				
Success!									
Job Stats (time in seconds):									
JobId	Maps	Reduces	MaxMapTime	MinMapTime	AvgMapTime	MedianMapTime	MaxReduceTime	MinReduceTime	
job_1464241803099_0064	1	1	3	3	3	2	2	2	Scoring,Scor:
job_1464241803099_0065	2	1	3	2	3	3	3	3	join_max_poi
job_1464241803099_0066	2	1	4	3	3	4	4	4	Team,Teams,Ti
job_1464241803099_0067	2	1	4	3	3	3	3	3	Final_answer
job_1464241803099_0068	1	1	2	2	2	2	2	2	order_Format
job_1464241803099_0069	1	1	3	3	3	2	2	2	order_Format
Input(s):									
Successfully read 45975 records (3273095 bytes) from: "/tmp/Assignment_2/Scoring.csv"									
Successfully read 1529 records from: "/tmp/Assignment_2/Teams.csv"									
Successfully read 7770 records from: "/tmp/Assignment_2/Master.csv"									
Output(s):									
Successfully stored 123 records (7522 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp1840634538/tmp-1065571"									

```
2016-05-26 06:40:26,744 [main] INFO org.apache.pig.Main - Pig script completed in 2 minutes, 16 seconds and 236 mil
```

2.3.6 Comparison Table For 4B

	HIVE	PIG
No of Jobs	2	6
Maps	4	2
Reduces	2	1
Total Time	7sec	2min 16 sec

3 Task 5:

3.1.1.1 Hortonworks shell

```
EgzTFUSS0U -O Query.pig
--2016-05-26 11:36:20-- https://drive.google.com/drive/folders/0B1mL7jdRfAbaZF9aeEgzTFUSS0U
Resolving drive.google.com... 216.58.220.110, 2404:6800:4006:801::200e
Connecting to drive.google.com|216.58.220.110|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://accounts.google.com/ServiceLogin?service=wise&passive=1209600&continue=https://drive.google.com/drive/folders/0B1mL7jdRfAbaZF9aeEgzTFUSS0U&followup=https://drive.google.com/drive/folders/0B1mL7jdRfAbaZF9aeEgzTFUSS0U [following]
--2016-05-26 11:36:20-- https://accounts.google.com/ServiceLogin?service=wise&passive=1209600&continue=https://drive.google.com/drive/folders/0B1mL7jdRfAbaZF9aeEgzTFUSS0U&followup=https://drive.google.com/drive/folders/0B1mL7jdRfAbaZF9aeEgzTFUSS0U
Resolving accounts.google.com... 216.58.220.109, 2404:6800:4006:801::200d
Connecting to accounts.google.com|216.58.220.109|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: "Query.pig"

  [ <=>          ] 55,479      --.-K/s   in 0.03s

2016-05-26 11:36:21 (2.11 MB/s) - "Query.pig" saved [55479]

[hdfs@sandbox ~]$_
```

3.1.1.1.1 Without Tez

```
[ <=>          ] 55,479      --.-K/s   in 0.03s

2016-05-26 11:36:21 (2.11 MB/s) - "Query.pig" saved [55479]

[hdfs@sandbox ~]$_ pig Query.pig
WARNING: Use "yarn jar" to launch YARN applications.
16/05/26 11:37:47 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
16/05/26 11:37:47 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
16/05/26 11:37:47 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2016-05-26 11:37:47,666 [main]INFO org.apache.pig.Main - Apache Pig version 0.15.0.2.3.2.0-2950 (rexported) compiled Sep 30 2015, 19:39:20
2016-05-26 11:37:47,667 [main]INFO org.apache.pig.Main - Logging error message
s to: /home/hdfs/pig_1464262667665.log
2016-05-26 11:37:48,217 [main]INFO org.apache.pig.impl.util.Utils - Default bootstrap file /home/hdfs/.pigbootstrap not found
2016-05-26 11:37:48,317 [main]ERROR org.apache.pig.impl.PigContext - Undefined parameter : p
2016-05-26 11:37:48,331 [main]ERROR org.apache.pig.Main - ERROR 2997: Encountered IOException. org.apache.pig.tools.parameters.ParameterSubstitutionException: Undefined parameter : p
Details at logfile: /home/hdfs/pig_1464262667665.log
2016-05-26 11:37:48,347 [main]INFO org.apache.pig.Main - Pig script completed in 799 milliseconds (799 ms)
[hdfs@sandbox ~]$_
```

With Tez

```

Details at logfile: /home/hdfs/pig_1464262667665.log
2016-05-26 11:37:48,347 [main] INFO  org.apache.pig.Main - Pig script completed
in 799 milliseconds (799 ms)
lhdfs@sandbox ~1$ pig -x tez Query.pig
WARNING: Use "yarn jar" to launch YARN applications.
16/05/26 11:38:49 INFO  pig.ExecTypeProvider: Trying ExecType : LOCAL
16/05/26 11:38:49 INFO  pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
16/05/26 11:38:49 INFO  pig.ExecTypeProvider: Trying ExecType : TEZ_LOCAL
16/05/26 11:38:49 INFO  pig.ExecTypeProvider: Trying ExecType : TEZ
16/05/26 11:38:49 INFO  pig.ExecTypeProvider: Picked TEZ as the ExecType
2016-05-26 11:38:50,009 [main] INFO  org.apache.pig.Main - Apache Pig version 0.
15.0.2.3.2.0-2950 (reexported) compiled Sep 30 2015, 19:39:20
2016-05-26 11:38:50,010 [main] INFO  org.apache.pig.Main - Logging error message
s to: /home/hdfs/pig_1464262730008.log
2016-05-26 11:38:50,543 [main] INFO  org.apache.pig.impl.util.Utils - Default bo
otup file /home/hdfs/.pigbootup not found
2016-05-26 11:38:50,666 [main] ERROR org.apache.pig.impl.PigContext - Undefined
parameter : p
2016-05-26 11:38:50,683 [main] ERROR org.apache.pig.Main - ERROR 2997: Encounter
ed IOException. org.apache.pig.tools.parameters.ParameterSubstitutionException:
Undefined parameter : p
Details at logfile: /home/hdfs/pig_1464262730008.log
2016-05-26 11:38:50,699 [main] INFO  org.apache.pig.Main - Pig script completed
in 813 milliseconds (813 ms)
lhdfs@sandbox ~1$ _

```

3.1.1.2 Ambari For Hive

Map Reduce:

Query Editor

Worksheet *

+ Add

- Remove All

+ Save Default Settings

hive.execution.engine

mr

SQL

TEZ

Query Editor

Worksheet *

```
1 SELECT firstname AS First_Name,
2         lastname AS Last_Name,
3         concat(birthday,'/',birthmon,'/',birthyear) AS DOB,
4         birthcountry AS Birth_Country,
5         COUNT(award)AS Award_No
6 FROM hockey_award_coaches ac,
7      hockey_master m
8 WHERE ac.coachid = m.coachid
9 GROUP BY firstname,
10         lastname,
11         concat(birthday,'/',birthmon,'/',birthyear),
12         birthcountry
13 ORDER BY award_no DESC LIMIT 1;
```

Execute

Explain

Save as...

Kill Session

New Worksheet

SQL

TEZ

Query Process Results (Status: Succeeded)
Save results... ▼

Logs

Results

INFO : Execution completed successfully

INFO : MapredLocal task succeeded

INFO : Number of reduce tasks not specified. Estimated from input data size: 1

INFO : In order to change the average load for a reducer (in bytes):

INFO : set hive.exec.reducers.bytes.per.reducer=<number>

INFO : In order to limit the maximum number of reducers:

INFO : set hive.exec.reducers.max=<number>

INFO : In order to set a constant number of reducers:

INFO : set mapreduce.job.reduces=<number>

INFO : number of splits:1

INFO : Submitting tokens for job: job_1464249118967_0001

INFO : The url to track the job: http://sandbox.hortonworks.com:8088/proxy/application_1464249118967_0001/

INFO : Starting Job = job_1464249118967_0001, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1464249118967_0001/

INFO : Kill Command = /usr/hdp/2.3.2.0-2950/hadoop/bin/hadoop job -kill job_1464249118967_0001

INFO : Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

INFO : 2016-05-26 08:14:00,123 Stage-2 map = 0%, reduce = 0%

INFO : 2016-05-26 08:14:06,324 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 1.77 sec

INFO : 2016-05-26 08:14:12,503 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 3.11 sec

INFO : MapReduce Total cumulative CPU time: 3 seconds 110 msec

INFO : Ended Job = job_1464249118967_0001

INFO : Number of reduce tasks determined at compile time: 1

INFO : In order to change the average load for a reducer (in bytes):

INFO : set hive.exec.reducers.bytes.per.reducer=<number>

INFO : In order to limit the maximum number of reducers:

INFO : set hive.exec.reducers.max=<number>

INFO : In order to set a constant number of reducers:

INFO : set mapreduce.job.reduces=<number>

INFO : number of splits:1

INFO : Submitting tokens for job: job_1464249118967_0002

INFO : The url to track the job: http://sandbox.hortonworks.com:8088/proxy/application_1464249118967_0002/

INFO : Starting Job = job_1464249118967_0002, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1464249118967_0002/

INFO : Kill Command = /usr/hdp/2.3.2.0-2950/hadoop/bin/hadoop job -kill job_1464249118967_0002

INFO : Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1

INFO : 2016-05-26 08:14:19,042 Stage-3 map = 0%, reduce = 0%

INFO : 2016-05-26 08:14:24,172 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 0.8 sec

Query Editor

Worksheet *

[+ Add](#) [- Remove All](#) [+ Save Default Settings](#)

hive.execution.engine ▼ tez ▼ ✕

SQL

TEZ

3.1.1.2.1 Enabling TEZ

Query Editor

Worksheet *

```

1 SELECT firstname AS First_Name,
2        lastname AS Last_Name,
3        concat(birthday,'/',birthmon,'/',birthyear) AS DOB,
4        birthcountry AS Birth_Country,
5        COUNT(award)AS Award_No
6 FROM   hockey_award_coaches ac,
7        hockey_master m
8 WHERE  ac.coachid = m.coachid
9 GROUP BY  firstname,
10         lastname,
11         concat(birthday,'/',birthmon,'/',birthyear),
12         birthcountry
13 ORDER BY award_no DESC LIMIT 1;

```

[Execute](#) [Explain](#) [Save as...](#) [Kill Session](#) [New Worksheet](#)

100%

Query Process Results (Status: SUCCEEDED) [Save results...](#)

Logs Results

INFO : Tez session hasn't been created yet. Opening session
INFO :



INFO : Status: Running (Executing on YARN cluster with App id application_1464249118967_0003)

INFO : Map 1: -/- Map 2: -/- Reducer 3: 0/1 Reducer 4: 0/1
INFO : Map 1: 0/1 Map 2: 0/1 Reducer 3: 0/1 Reducer 4: 0/1

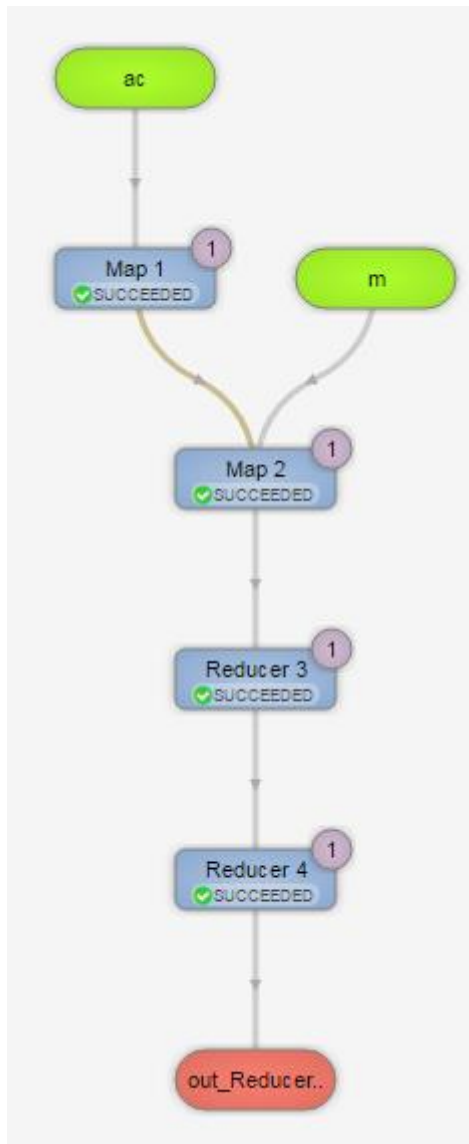
3.1.1.2.2 DAG Details

DAG Details	
Download data	
Application Id	application_1464249118967_0003
Entity Id	dag_1464249118967_0003_1
User	hive
Status	✓ SUCCEEDED
Start Time	26 May 2016 18:18:53
End Time	26 May 2016 18:19:07
Duration	14 secs

3.1.1.2.3 Time for individual Map Job

All DAGs / DAG [hive_20160526081809_a403bad3-244a-4874-8162-50530007fd9b:1]							
		DAG Details	DAG Counters	Graphical View	All Vertices	All Tasks	All TaskAttempts
		Last refreshed at 26 May 2016 09:06:01					Refresh
RegEx or Column1, Column2...:RegEx		Search		First	1	Last - 1	Rows 25 
Task Index	Vertex Name	Status	Progress	Start Time	End Time	Duration	
03_000000	Reducer 4	✓ SUCCEEDED	100%	26 May 2016 18:19:06	26 May 2016 18:19:07	945 ms	
02_000000	Reducer 3	✓ SUCCEEDED	100%	26 May 2016 18:19:06	26 May 2016 18:19:06	390 ms	
01_000000	Map 2	✓ SUCCEEDED	100%	26 May 2016 18:18:59	26 May 2016 18:19:06	6 secs	
00_000000	Map 1	✓ SUCCEEDED	100%	26 May 2016 18:18:59	26 May 2016 18:19:05	6 secs	

3.1.1.2.4 Graphical View



3.1.1.3 Cost Based Optimization

Query Editor

Worksheet

```
1 SELECT firstname AS First_Name,
2         lastname AS Last_Name,
3         concat(birthday,'/',birthmon,'/',birthyear) AS DOB,
4         birthcountry AS Birth_Country,
5         COUNT(award)AS Award_No
6 FROM hockey_award_coaches ac,
7      hockey_master m
8 WHERE ac.coachid = m.coachid
9 GROUP BY firstname,
10         lastname,
11         concat(birthday,'/',birthmon,'/',birthyear),
12         birthcountry
13 ORDER BY award_no DESC LIMIT 1;
```

Execute




Explain


Save as...

Kill Session

New Worksheet

SQL



TEZ 

100%

Query Process Results (Status: SUCCEEDED)

Save results... ▾

Logs

Results

Filter columns...

previous

next

first_name	last_name	dob	birth_country	award_no
Dick	Irvin	19/7/1892	Canada	9

Query Process Results (Status: Succeeded)

Explain

- Plan not optimized by CBO.
- Vertex dependency in root stage
- Map 2 <- Map 1 (BROADCAST_EDGE)
- Reducer 3 <- Map 2 (SIMPLE_EDGE)
- Reducer 4 <- Reducer 3 (SIMPLE_EDGE)
- Stage-0

Query Editor

Worksheet

```
1 ANALYZE TABLE hockey_master COMPUTE STATISTICS;
```

SQL

Settings

Charts

Links

TEZ

2

Execute Explain Save as... Kill Session New Worksheet

100%

Query Process Results (Status: SUCCEEDED) Save results... ▾

Logs Results

Filter columns...




previous next

Query Editor


Worksheet

```
1 ANALYZE TABLE hockey_master COMPUTE STATISTICS FOR COLUMNS coachid, firstname, lastname, birthyear,
2 birthmon, birthcountry;
```

SQL



TEZ

3

Execute

Explain

Save as...

Kill Session

New Worksheet

100%

Query Process Results (Status: SUCCEEDED)

Save results... ▾

Logs

Results

INFO : Session is already open
INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464249118967_0005)




INFO : Map 1: -/- Reducer 2: 0/1
INFO : Map 1: 0/1 Reducer 2: 0/1
INFO : Map 1: 0(+1)/1 Reducer 2: 0/1
INFO : Map 1: 1/1 Reducer 2: 0/1
INFO : Map 1: 1/1 Reducer 2: 0(+1)/1
INFO : Map 1: 1/1 Reducer 2: 1/1

Query Editor


Worksheet

1 ANALYZE TABLE hockey_award_coaches COMPUTE STATISTICS;

SQL



TEZ

4

Execute

Explain

Save as...

Kill Session

New Worksheet

100%

Query Process Results (Status: SUCCEEDED)

Save results... ▾

Logs

Results

INFO : Session is already open

INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464249118967_0005)

INFO : Map 1: 0/1

INFO : Map 1: 0(+1)/1

INFO : Map 1: 1/1


INFO : Table default.hockey_award_coaches stats: [numFiles=1, numRows=88, totalSize=3073, rawDataSize=2985]


Query Editor


Worksheet

1 ANALYZE TABLE hockey_award_coaches COMPUTE STATISTICS FOR COLUMNS coachid, award;


SQL







TEZ

7

Execute

Explain

Save as...

Kill Session

New Worksheet

100%

Query Process Results (Status: SUCCEEDED)

Save results... ▾

Logs

Results

INFO : Session is already open
INFO :

INFO : Status: Running (Executing on YARN cluster with App id application_1464249118967_0005)

INFO : Map 1: 0/1 Reducer 2: 0/1
INFO : Map 1: 0(+1)/1 Reducer 2: 0/1
INFO : Map 1: 1/1 Reducer 2: 0(+1)/1
INFO : Map 1: 1/1 Reducer 2: 1/1

4 Big Data Report

4.1 Introduction

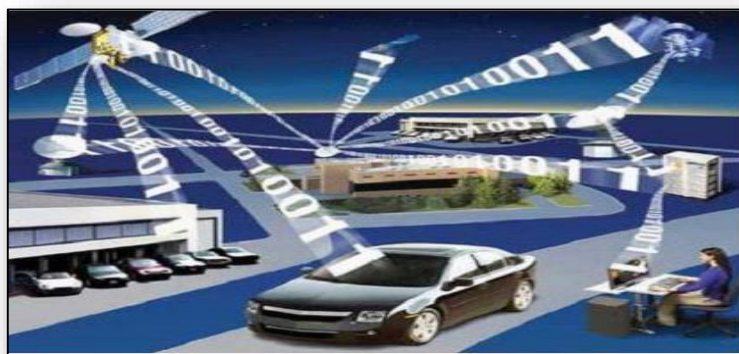
This paper describes big data mining concepts that facilitate the extraction of valuable information from a collection of big data from various dimensions, particularly spatial data (in this case). Big data can be seen as a complex set of data that revolves around four 'V' Viz. Volume, Variety, Veracity and Velocity (Shuliang, Gangyi & Ming, 2014). Big data mining often leads to privacy issues because data miners have access to data of various individuals which makes them in charge of all the information at hand. This paper also discusses privacy frameworks which allow increased data privacy by treating data miners and sensitive data as two different entities and only allowing limited data access.

4.2 Current state of the art

4.2.1 Big Spatial Data Mining

Spatial data is seen as a basis for big data. This data relates to space which accounts for 80% of the total data. It is through spatial data that we can describe a specific geographic orientation of any object in this world (Shuliang, Gangyi & Ming, 2014). Spatial data is used to monitor the earthly activities and used by data intelligence applications to predict things. For instance, the weather updates and likelihood of any natural disasters are inferred from the research in trends of spatial data over the time and and any anomalies indicate such outcomes. Thus, importance of monitoring spatial data is crucial for the living conditions. The satellites send enormous amount of data to earth every minute which contains information about our geography and natural happenings, traffic, people, etc.

(Fig: Satellite data and services)



This data allows us to gain valuable information about geospatial objects and utilize this knowledge in positioning and locating things from anywhere around the world. This

knowledge is then transcended into data intelligence to make our lives safer and easier. For instance, the data about vehicles transiting road each day allows data scientists to come up with trends that allow them to focus on means to reduce traffic congestions and offer alternative solutions which in turn help us live safer.

The current issues that exist with spatial data are:

1. Garbage: Majority of this data is junk that cannot be used to make any conclusion. This data is further filtered down to find some conclusive data.
2. Contamination: Almost 95% of the data is inconsistent and hence contaminated. This is because of the inconsistencies, repetitions, errors and incomplete data that is collected.
3. Difficult to use: Having such a large volume of data with 95% contamination makes it really difficult to use for data intelligence but data cleansing algorithms are further applied to it to make some valuable inferences.

All of these issues associated with Big Spatial Data can be resolved use one of the following techniques:

1. Basic Big Data Technology
2. Spatial Data Technology
3. Extraction Data Intelligence

This paper presents a good insight on the importance of spatial data in its applications in the real while explaining the issues related to it and possible solutions. Since Big data is still a relatively new concept, this paper does not offer much depth into the solutions as one cannot possible infer all the information from 95% contaminated data. Thus, some more research is needed to decrease the level of contamination and hence use majority of data at hand rather than discarding majority of data that we do now.

[Framework for categorizing and applying privacy and preservation techniques in Data Mining](#)

Data mining or knowledge discovery from data (KDD) is the process of extracting valuable information from data that lies around us. This process can lead to privacy threat as data miners have access to secret information of various individuals (Xu et al, 2016).

Data privacy in some decreases data utility. For instance, using privacy preserving data mining (PPDM) leads to privacy issues in some KDD stages (Xu et al, 2016). Data privacy can be maintained using different approaches. One such example is demonstrated in the figure below.

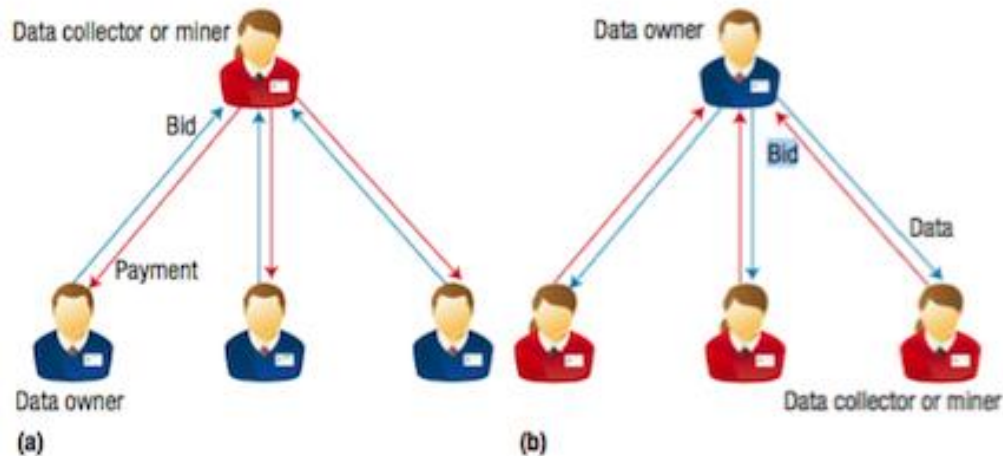


Fig: Auction models

- a) Data owner sells data to data miner via bid (Blue arrow) and receives payment (Red arrow)**
- b) Data owner offers data for sale (Blue arrow) to receive bids from multiple miners (Red arrow)**

This figure clearly demonstrates that the best way to keep privacy of data intact is by preventing others from accessing data. In the first case, the data is only offered to one bidder whereas in second case, all the bidders had access to data. In case b, the data privacy is breached.

In order to maintain data privacy and data utility, this research highlights Rampart Framework as the possible solution for KDD. This can be achieved by the following 5 practices (Xu et al, 2016):

1. k-Anonymity: Anonymization technique is used to modify the set of values so that they are indistinguishable from other tuples to obtain privacy preferences that are personalized.
2. Modification: Even after anonymization, data contains sensitive information that can be

identified by the data miners. In order to avoid possible privacy breaches, geometric transformations such as scaling and rotation are applied to data.

3. Provenance: Decision makers are supported by the research of data miners and if data does not come directly from data miners, they must know the credibility of data and thus provenance allows them to evaluate data and reveals ancestral information and also the transformations that were applied to data.

4. Restriction: Restrictions are applied to data based on the privacy rules of a country.

5. Agreement & Trade: This paper highlights the trade agreement issues and privacy breaches and this is further managed using game theory.

Overall, this paper expresses a good insight on preserving the data privacy using Rampart's framework but some loopholes still exist that can be seen in trade agreements where data is made public for auctions and people can access it. And moreover, restrictions differ from country to country which makes it difficult for data miners and decision makers to verify the credibility of data if it comes from various sources. Some more stringent restrictions need to be applied on data availability to ensure its privacy in the future.

4.2.2 Big Data is social media using data mining techniques

Social media is the greatest source of big data and individual practices. This paper highlights that every individual on average spends two and a half hours on social media (Gole & Tidke, 2015) which means we generate Trillions of Terabytes of data every single day that can be used to analyze individual preferences and hence predict the trends. Due to the volume of this data, it is currently not possible to directly infer conclusive information from this data.

Hence, clustering algorithms are used to find some relationship and dependencies in this data which is further analyzed to make decisions.

Data mining algorithms such as HACE and RMKMC are quite popular for identifying data patterns using clustered matrices. Hadoop Map Reduce and NoSQL is used to further process the data and gain valuable insight from the whole unstructured set of Big Data. The following table presents with a view of the challenges that such data presents and which techniques can be applied to comprehend them based on the characteristics of data.

Characteristics	Challenge	Technique
Volume	Storage/Scale	Distributed File System
Velocity	Fast Processing	Parallel Programming
Variety	Heterogeneity	NOSQL Databases
Value	Knowledge Discovery, Semantics, Analytics	Data Mining Algorithms
Data Accessibility/Availability	Privacy/Security	WINE platform & <u>BotCloud</u>

As seen in the table, Big Data is primarily driven by volume, velocity, variety and value. Each of these characteristics have different impact on the accessibility of data and thus different technique needs to be used for processing different set of data. Big Data Mining using Hadoop and clustering technique offers good speed of data processing as described in this paper. The main problems that Hadoop caters to can be regarded as cost effectiveness, big clusters, parallel processing, big storage, failovers, data distribution and map reduce.

This paper presents a great analysis on the current issues and mitigation technologies that can be used by data miners to process data effectively and efficiently.

Reducing the search space for big data mining for interesting patterns from uncertain data

This paper highlights the evident fact that search space for big data mining is not only uncertain but huge in its aspect. Mining algorithms without a focus point would not return valuable information from such a vast search space (Leung et al, 2014). Thus, in order to reduce the search space, the authors propose Map Reduce function to satisfy the user needs which can be seen as succinct anti-monotone. The authors present two basic algorithms as mining frequent singletons and mining frequent non-singletons. Both of these algorithms focus on expressing the result with minimum support for each constraint. This allows in reducing the overall search space. The algorithm works by scanning and identifying the uncertain data in order to compute the support required by the items in the set. This scan is again performed but this time in the database to ensure that the insertion of all the transactions are completed in the form a tree structure.

The outcome of this algorithm is the set of pattern that interest the users. This certainty of data set thus helps save time using Map Reduce function for constraint checks. The authors present their result sets in an experiment set which looks promising and the algorithm seems to work effectively. This sort of reducing in a n uncertain data search space marks the beginning of a new era in data mining.

Overall, the authors have done a great job is providing experimental analysis and results that can be seen to provide a great insight on dealing with uncertain data sets that prevail in big data environment. This paper presents a good view of structuring the veracity of data by following singleton approach and achieving the interested user goals.

Conclusion

All the research papers present with a single notion of big data mining that deals with a great volume of big data. All the authors have presented their arguments and research providing various algorithm to drill through the unstructured data that lingers around the sophisticated field of IT. These papers provide a good insight on how to deal with different sets of Big data that revolve majorly around velocity, veracity, volume and value. Each of these characteristics present a different challenge that is being identified in these research papers and presented with possible solutions. Overall, these researches prove to be beneficial for big data mining as they offer various approaches for various problems being identified and some further research in this field can offer the gateway to a robust application design that deals with all problems at once rather than deploying different algorithms for different problems.

4.3 References

1. Gole, S. & Tidke, B. (2015). A survey of Big Data in social media using data mining techniques. In *International Conference on Advanced Computing and Communication Systems (ICACCS -2015)*.
2. Leung, C., MacKinnon, R., & Jiang, F. (2014). Reducing the Search Space for Big Data Mining for Interesting Patterns from Uncertain Data. In *IEEE International Congress on Big Data*.
3. Shuliang, W., Gangyi, D., & Ming, Z. (2014). Big Spatial Data Mining. In *IEEE International Conference on Big Data*.
4. Xu, L., Jiang, C., Chen, Y., Wang, J., & Ren, Y. (2016). A Framework for Categorizing and Applying Privacy-Preservation Techniques in Big Data Mining. *Computer*, 49(2), 54-62. <http://dx.doi.org/10.1109/mc.2016.43>