Session 4

Assignment 2

|  |  |
| --- | --- |
| **Prepared For:** | AcadGild |
|  |  |
| **Document Approval:** | **AcadGild** |
|  |  |
|  |  |
|  |  |
|  |  |
| **Project Title:** | Session 4 – Assignment 2 |
|  |  |
| **Prepared By:** | Duncan Burgess |
|  |  |
|  | dburgess@duncb.com |
|  |  |
| **Primary Engineer:** | Duncan Burgess |
|  |  |
| **Document Reference:** | **Session 4 – Assignment 2** |
|  |  |
| **Start Date:** | 15/09/2017 |
|  |  |
|  |  |

# 

# Contents

[Contents 2](#_Toc493234016)

[Change History 3](#_Toc493234017)

[1. Problem Statement 4](#_Toc493234018)

[2. Setting up the PIG environment 5](#_Toc493234019)

[3. Solutions 6](#_Toc493234020)

[3.1. Loading and viewing properties of datasets 6](#_Toc493234021)

[3.2. Join 7](#_Toc493234022)

[3.3. GroupBy 7](#_Toc493234023)

[3.4. SUM 8](#_Toc493234024)

[3.5. MIN 8](#_Toc493234025)

[3.6. MAX 8](#_Toc493234026)

[3.7. Average 8](#_Toc493234027)

[3.8. CONCAT 8](#_Toc493234028)

[3.9. STORE 9](#_Toc493234029)

[3.10. FLATTEN 9](#_Toc493234030)

[3.11. TOKENIZE 10](#_Toc493234031)

[3.12. DISTINCT 10](#_Toc493234032)

[3.13. LIMIT 11](#_Toc493234033)

[3.14. IsEmpty 11](#_Toc493234034)

# Change History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Document Revision** | **Date** | **Authored By** | **Authorised By** | **Sections Affected** | **Reason for Change** |
| Rev 01 | 15/09/2017 | Duncan Burgess |  | All | Initial release. |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Problem Statement

Create a sample dataset and implement the below Pig commands on the same dataset.

1) Concat

2) Tokenize

3) Sum

4) Min

5) Max

6) Limit

7) Store

8) Distinct

9) Flatten

10) IsEmpty

# Setting up the PIG environment

Steps taken to run PIG

cd /usr/local/hadoop-2.6.0/sbin

**Start services**

./start-all.sh

./mr-jobhistory-daemon.sh start historyserver

**Check services**

$ jps

3184 NodeManager

2930 SecondaryNameNode

13445 Jps

3080 ResourceManager

2632 NameNode

3562 JobHistoryServer

2762 DataNode

**Two datasets created to demonstrate Pig Commands.**

**optuminf.txt**

1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows

2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows

3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows

4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management

5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL

6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux

7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux

8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL

10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows

**optumpay.txt**

1,56000

2,34000

3,22000

4,62000

5,45000

6,39000

7,41000

8,47000

9,73000

10,46000

11, 29500

**Copy files to hdfs**

$ hadoop fs -copyFromLocal optuminf.txt

$ hadoop fs -copyFromLocal optumpay.txt

**Start Pig**

pig – *(mapreduce mode)*

2017-09-14 14:33:38,968 INFO [main] pig.ExecTypeProvider: Trying ExecType : MAPREDUCE

2017-09-14 14:33:38,969 INFO [main] pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType

pig -x local - *(local mode)*

2017-09-14 14:32:48,658 INFO [main] pig.ExecTypeProvider: Trying ExecType : LOCAL

2017-09-14 14:32:48,658 INFO [main] pig.ExecTypeProvider: Picked LOCAL as the ExecType

you will be presents with the pig shell: grunt>

We need to load the files now: ***Note PigStorage is cap sensitive.***

# Solutions

The following demonstrations show the use of Pig commands.

## Loading and viewing properties of datasets

We need to load the files now:

*optumstaff = load 'optuminf.txt' using PigStorage (',') AS (id:int,company:chararray, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray, tasks:int, skill:chararray);*

*staffpay = load 'optumpay.txt' using PigStorage (',') AS (id:int, pay:int);*

We now have 2 bags loaded **optumstaff** and **optumpay**.

We can see the properties using

***DESCIBE***

***ILLUSTRATE***

***EXPLAIN***

grunt> describe optumstaff;

optumstaff: {id: int,company: chararray,firstname: chararray,lastname: chararray,age: int,phone: chararray,city: chararray,tasks: int,skill: chararray}

grunt> describe staffpay;

staffpay: {id: int,pay: int}

grunt> illustrate optumstaff;

----------------------------------------------------------------------

| optumstaff | id:int | company:chararray | firstname:chararray | lastname:chararray | age:int | phone:chararray | city:chararray | tasks:int | skill:chararray |

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

| | 2 | Optumuk | Mike | Tartaglia | 32 | 9848022338 | Burton Latimer | 78 | windows |

--------------------------------------------------------------------------------------------------------------------------

grunt> illustrate staffpay;

-----------------------------------------

| staffpay | id:int | pay:int |

-----------------------------------------

| | 1 | 56000 |

explain staffpay;

#--------------------------------------------------

# Map Reduce Plan

#--------------------------------------------------

MapReduce node scope-29

Map Plan

staffpay: Store(fakefile:org.apache.pig.builtin.PigStorage) - scope-28

|

|---staffpay: New For Each(false,false)[bag] - scope-27

| |

| Cast[int] - scope-22

| |

| |---Project[bytearray][0] - scope-21

| |

| Cast[int] - scope-25

| |

| |---Project[bytearray][1] - scope-24

|

|---staffpay: Load(hdfs://localhost:9000/user/acadgild/optumpay.txt:PigStorage(',')) - scope-20--------

Global sort: false

## Join

To join the files on common schema key;

optum\_combined = JOIN optumstaff by id, staffpay by id;

dump optum\_combined;

(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows,1,56000)

(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows,2,34000)

(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows,3,22000)

(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management,4,62000)

(5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL,5,45000)

(6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux,6,39000)

(7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux,7,41000)

(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL,8,47000)

(10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows,10,46000)

grunt> describe optum\_combined;

optum\_combined: {optumstaff::id: int,optumstaff::company: chararray,optumstaff::firstname: chararray,optumstaff::lastname: chararray,optumstaff::age: int,optumstaff::phone: chararray,optumstaff::city: chararray,optumstaff::tasks: int,optumstaff::skill: chararray,staffpay::id: int,staffpay::pay: int}

includes ***pay.***

## GroupBy

To perform aggregate function’s you need to use GROUPBY.

grunt> optum\_group = group optum\_combined by company;

(Optumuk,{(10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows,10,46000),(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL,8,47000),(7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux,7,41000),(6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux,6,39000),(5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL,5,45000),(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management,4,62000),(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows,3,22000),(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows,2,34000),(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows,1,56000)})

## SUM

To find the total cost of pay:

totalpay = foreach optum\_group generate group as grp, SUM(optum\_combined.pay);

dump totalpay;

(Optumuk,392000) this is all the combined pay for all optum staff.

## MIN

To Find out the lowest pay and show name of employee using MIN.

minpay1 = FOREACH optum\_group GENERATE MIN(optum\_combined.pay) as P;

minpay = FILTER optum\_combined BY pay == (int)minpay1.P;

*(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows,3,22000)*

## MAX

To Find out the highest pay and show name of employee using Max.

maxpay1 = FOREACH optum\_group GENERATE MAX(optum\_combined.pay) as M;

maxpay = FILTER optum\_combined BY pay == (int)maxpay1.M;

*(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management,4,62000)*

## Average

To find the average pay and round it to 2 decimal places.

*averagepay = foreach optum\_group generate group as grp, ROUND\_TO( ((double)AVG(optum\_combined.pay)),2);*

(Optumuk,43555.56)

## CONCAT

To view firstname and secondname as 1 using CONCAT.

Using $2 and $3 to represent 2nd and 3rd columns (1st is represented as $0)

*concatfile = FOREACH optum\_combined GENERATE CONCAT($2, ' ', $3);*

*(Duncan Burgess)*

*(Mike Tartaglia)*

*(Ben Ludford)*

*(Tracey Baker)*

*(Chris Ginn)*

*(Chris Reid)*

*(Stuart Ford)*

*(Shaun Turner)*

*(Rosario Marino)*

## STORE

*STORE optum\_combined INTO 'output/optum\_combined' USING PigStorage(',');*

*[main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!*

**Check results**

$ hadoop fs -ls output/optum\_combined

*17/09/14 22:03:29 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable*

*Found 2 items*

*-rw-r--r-- 1 acadgild supergroup 0 2017-09-14 22:01 output/optum\_combined/\_SUCCESS*

*-rw-r--r-- 1 acadgild supergroup 581 2017-09-14 22:01 output/optum\_combined/part-r-00000*

$ hadoop fs -cat output/optum\_combined/part-r-00000

*17/09/14 22:03:54 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable*

*1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows,1,56000*

*2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows,2,34000*

*3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows,3,22000*

*4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management,4,62000*

*5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL,5,45000*

*6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux,6,39000*

*7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux,7,41000*

*8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL,8,47000*

*10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows,10,46000*

## FLATTEN

*Create a grouped file*

*grunt> names = GROUP optumstaff by firstname;*

**Resulting file {} = group () = Tuple**

*(Ben,{(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows)})*

*(Mike,{(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows)})*

*(****Chris,{(6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux),(5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL)})***

*(Shaun,{(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL)})*

*(Duncan,{(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows)})*

*(Stuart,{(7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux)})*

*(Tracey,{(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management)})*

*(Rosario,{(10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows)})*

*grunt> flatten\_file = foreach names FLATTEN();*

**Resulting file**

*(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows)*

*(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows)*

***(6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux)***

***(5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL)***

*(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL)*

*(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows)*

*(7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux)*

*(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management)*

*(10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows)*

## TOKENIZE

To split a string (which contains a group of words) in a single tuple and returns a bag which contains the output of the split operation

grunt> optum\_tokenize = foreach optumstaff Generate TOKENIZE(firstname);

**Resulting file**

*({(Duncan)})*

*({(Mike)})*

*({(Ben)})*

*({(Tracey)})*

*({(Chris)})*

*({(Chris)})*

*({(Stuart)})*

*({(Shaun)})*

*({(Rosario)})*

## DISTINCT

To remove duplicate tuples from a relation.

File loaded optumstaffwithduplicate

(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows)

(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows)

(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows)

(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows)

(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management)

(5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL)

(6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux)

(7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux)

(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL)

(10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows)

grunt> distinct\_file = DISTINCT optumstaffwithduplicate;

**Results**

*(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows)*

*(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows)*

*(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows)*

*(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management)*

*(5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL)*

*(6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux)*

*(7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux)*

*(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL)*

*(10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows)*

Duplicate name Duncan Burgess removed.

## LIMIT

*To sort the staff by age then to limit the results to top 3;*

*grunt> age = ORDER optumstaff by age DESC, lastname ASC;*

*(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows)*

*(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL)*

*(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows)*

*(7,Optumuk,Stuart,Ford,24,9848022334,Edingborough,83,linux)*

*(10,Optumuk,Rosario,Marino,24,9848022333,Oxford,72,windows)*

*(5,Optumuk,Chris,Ginn,23,9848022336,Cardif,14,SQL)*

*(6,Optumuk,Chris,Reid,23,9848022335,Leek,87,linux)*

*(3,Optumuk,Ben,Ludford,22,9848022339,Swindon,90,windows)*

*(4,Optumuk,Tracey,Baker,21,9848022330,Birmingham,93,management)*

Top the oldest 4 staff

grunt> oldest = LIMIT age 3;

**Resulting file**

*(1,Optumuk,Duncan,Burgess,52,9848022337,Mawsley,89,windows)*

*(8,Optumuk,Shaun,Turner,38,9848022388,Leicester,67,SQL)*

*(2,Optumuk,Mike,Tartaglia,32,9848022338,Burton Latimer,78,windows)*

## IsEmpty

**IsEmpty**

To check if a bag or map is empty.

Create a cogroup from optumstaff and staffpay.

grunt> optum\_cogroup = COGROUP optumstaff BY id, staffpay BY id;

grunt> optum\_isempty = filter optum\_cogroup by IsEmpty(optumstaff);

**Results**

*(9,{},{(9,73000)})*

*(11,{},{(11,29500)})*