Exercise 3

Pig Evaluation Functions

Contents

Lab 1 Pig Evaluation Functions

This exercise gives you the opportunity work with some of the Pig relational operators in order to use some Pig evaluation functions.

After completing this hands-on lab, you'll be able to:

Apply evaluation functions to fields in tuples

Invoke the FSShell from the Grunt shell

Allow 10 minutes to complete this lab.

This version of the lab was designed using the IBM BigInsights 4.1 Quick Start Edition, but has been tested on the 4.1.0.0 image. Throughout this lab you will be using the following account login information. If your passwords are different, please note the difference.

	Username	Password
RVM Login	root	password
Ambari	admin	admin

1.1 Evaluation functions

_ If Hadoop is not running, start it and its components using the icon on the desktop.

_ Make sure your shared folder is set-up.

_ You have the option of running your Pig commands from either the *Grunt shell* or from your Pig script. In either case you need to change to the *Pig bin* directory and start the shell running in local mode.

```
cd /usr/iop/4.1.0.0/pig/bin
```

_ If you are going to run from the *Grunt shell* locally, then execute;

```
./pig -x local
```

If you are going to use your *pig.script,* then you will execute the script and pass in your directory parameter by doing the following:

```
./pig -x local -param_file /home/biadmin/myparams /home/biadmin/pig.script
```

You can edit your *pig.script* by using *vi.* Also remember that in your script, you can comment out any commands by coding two dashes (--) before the command.

In the directions, the *DUMP* operator will only be specified for the final output. But that does not stop you from adding intermediate *DUMP* operators in order to see the effects of each operator.

Note: Remember that the labfiles folder is a shared folder, which can be found in the directory: /mnt/hgfs/

_ Read the */mnt/hgfs/labfiles/SampleData/books.csv* file.

```
books = load '$dir/books.csv' using PigStorage(',') as (bknum: int,
author:chararray, book:chararray, pubyear:int);
```

_ Read the */mnt/hgfs/labfiles/SampleData/reviews.csv* file.

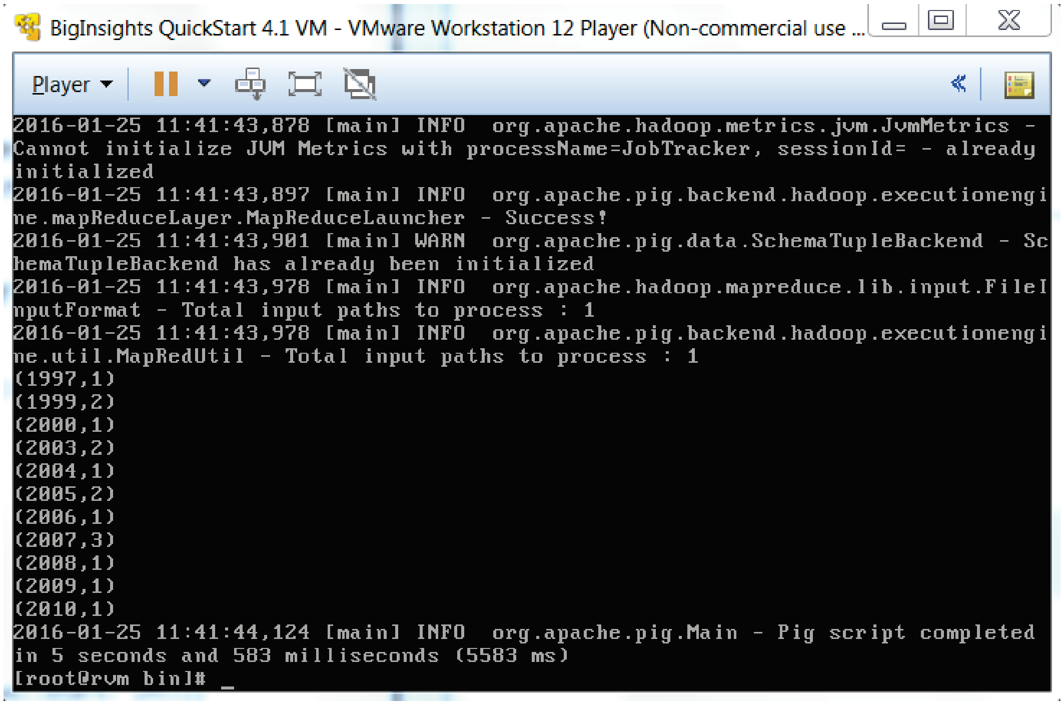
```
reviews = load '$dir/reviews.csv' using PigStorage(',') as (bknum:int,
reviewer:chararray, stars:int);
```

_ Group the *books* relations by *pubyear.*

```
booksInYear = group books by pubyear;
```

_ Calculate the number of books published in each year.

```
booksPerYear = foreach booksInYear generate group, COUNT($1);
dump booksPerYear;
```


BigInsights QuickStart 4.1 VM - VMware Workstation 12 Player (Non-commercial use ...
Player

```
2016-01-25 11:41:43,878 [main] INFO  org.apache.hadoop.metrics.jvm.JvmMetrics -
Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - already
initialized
2016-01-25 11:41:43,897 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
2016-01-25 11:41:43,901 [main] WARN  org.apache.pig.data.SchemaTupleBackend - Sc
hemaTupleBackend has already been initialized
2016-01-25 11:41:43,978 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileI
nputFormat - Total input paths to process : 1
2016-01-25 11:41:43,978 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.util.MapRedUtil - Total input paths to process : 1
(1997,1)
(1999,2)
(2000,1)
(2003,2)
(2004,1)
(2005,2)
(2006,1)
(2007,3)
(2008,1)
(2009,1)
(2010,1)
2016-01-25 11:41:44,124 [main] INFO  org.apache.pig.Main - Pig script completed
in 5 seconds and 583 milliseconds (5583 ms)
[root@rvm bin]# _
```

_ This is going to be a bit more interesting. Calculate the average number of stars for each book. Start by joining the *books* relation and the *reviews* relation on *bknum.*

```
bookssAndReviews = join books by bknum, reviews by bknum;
```

_ Next project a new relation so that you only are working with the title of the book and the number of stars for each reviewer.

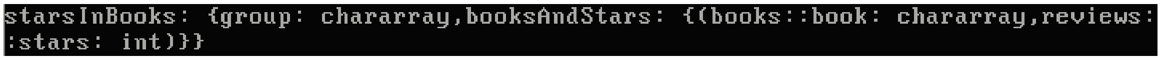
```
booksAndStars = foreach booksAndReviews generate book, stars;
```

_ Group the *booksAndStars* relation by book title.

```
starsInBooks = group booksAndStars by book;
```

_ Now it might be a good idea to look at the schema for *starsInBooks.*

```
describe starsInBooks;
```



```
starsInBooks: {group: chararray,booksAndStars: {(books::book: chararray,reviews:
:stars: int)}}
```

_ Next you are going to have to use the *FOREACH* operator. Getting access to the book title is easy since that how the records are grouped. But what about the number of stars? Which relation was used in the first grouping? It was *booksAndStars.* So you will have to dereference *stars* using *booksAndStars.*

```
avgStars = foreach starsInBooks generate group, AVG(booksAndStars.stars);
dump avgStars;
```

_ Having the value for *stars* as a double may not be what you want. You cannot give a half of a star or a third of a star so when calculating the average, you might want it to stay as an integer. To do this, cast the average value to *int.*

```
avgStars = foreach starsInBooks generate group,
(int)AVG(booksAndStars.stars);
dump avgStars;
```

_ What if you wanted to eliminate any ratings that were less than four stars. (I know that is may not make sense, but who says that a learning opportunity has to make sense?) You could use a *FOREACH* operator with a nested block.

```
bogusAvgStars = foreach starsInBooks {filteredStars = filter
                                      booksAndStars by stars > 3;
                                      numStars = filteredStars.stars;
                                      generate group, (int)AVG(numStars);}
dump bogusAvgStars;
```

_ Using the *EXPLAIN* operator, it is possible to get and understanding as to how Pig is going to attack a particular MapReduce problem.

```
explain bogusAvgStars;
```

_ A quick look at running hdfs commands from the *Grunt shell.* If you do not have the *Grunt shell* open in local mode, execute the following:

```
./pig -x local
```

_ Next list the current directory using the *FSShell* command from the *Grunt shell.*

```
fs -ls /
```

What is listed? It is the / directory. Interesting. Why did it not list the directories and files in hdfs? You are running in local mode.

_ Exit from Pig local mode.

```
quit;
```

_ Invoke the *Grunt shell* in MapReduce mode. Remember it is the default mode.

```
./pig
```

_ Again execute the *FSShell* and do a directory listing.

```
fs -ls /
```

This time it listed the data in hdfs.

_ You can quit from pig. You can stop Hadoop. This is the end of the exercise.

End of exercise

NOTES

NOTES

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

Please Recycle