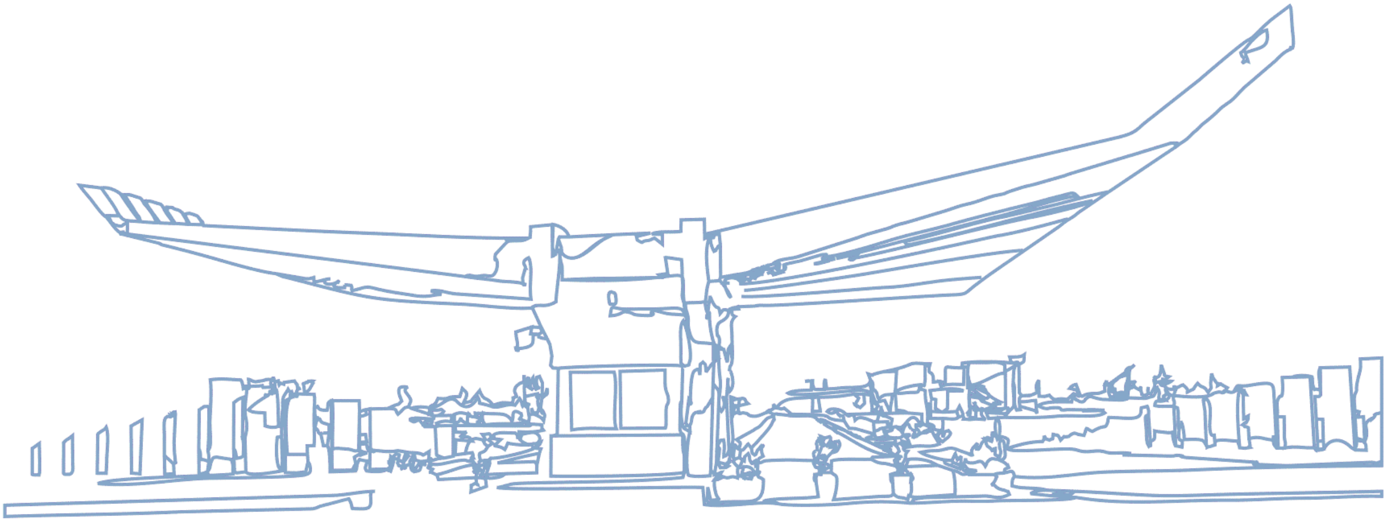
EPOKA UNIVERSITY | Computer Engineering Department

CEN 571 – Data Mining

Assignment 4: Pig and PigLatin



PREPARED:
Baftjar TABAKU

02.05.2020
Epoka University
Tirana, ALBANIA

ACCEPTED:
Prof.Dr. Arben Asllani

Perform the following tasks:

1. Upload the input file or files into the Hadoop cluster
2. Execute Pig Latin commands and display the results on the screen
3. Store final results in the Hadoop cluster

The deliverables for each task are:

- A script file with a list of Pig Latin commands; and
- Output results in a text file or as otherwise indicated in the exercise.

For this exercise, we use two files: ‘investor.txt' and ‘stockprice.txt'.

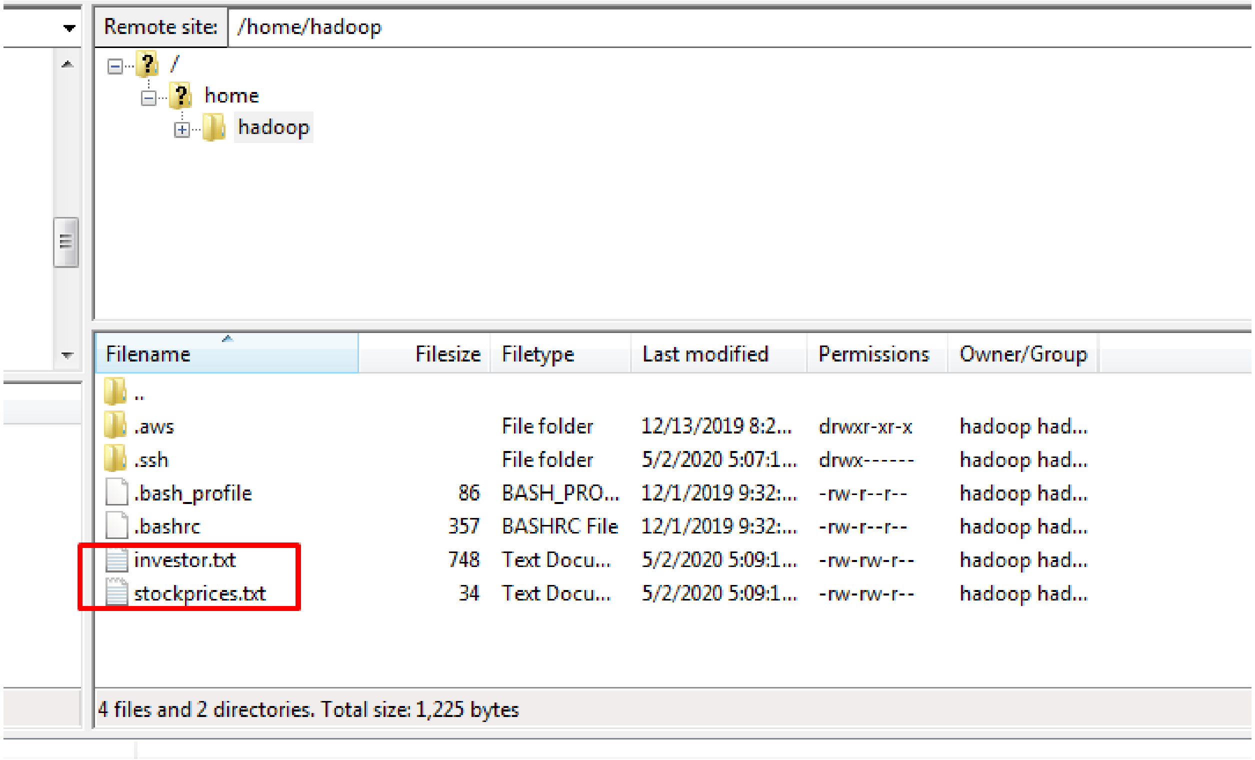
The investor file provides a list of investors and the number of shares that they have purchased in e given stock. The second file stores the stock prices.

Using Pig Latin commands, perform the following operations:

1. Upload the two files in HDFS
2. Load files into as 'investors' and 'stock_prices'; Display both files to make sure they are loaded correctly
3. Display the structure of relation ‘investors’ and ‘stock_prices’
4. Join the two files ('investors' and 'stock_prices') by stock symbol; Display the joined file
5. Group the above file (joined file) by the 'lastname' of the investors; Display the results
6. Calculate the total shares (simply the sum of shares among all stocks); Display results
7. Calculate the total dollar amount that each investor has invested (shares per each stock multiplied by the stock price); Display the results.
8. Filter the top two investors that have invested the most; Display the results

1. Upload the two files in HDFS

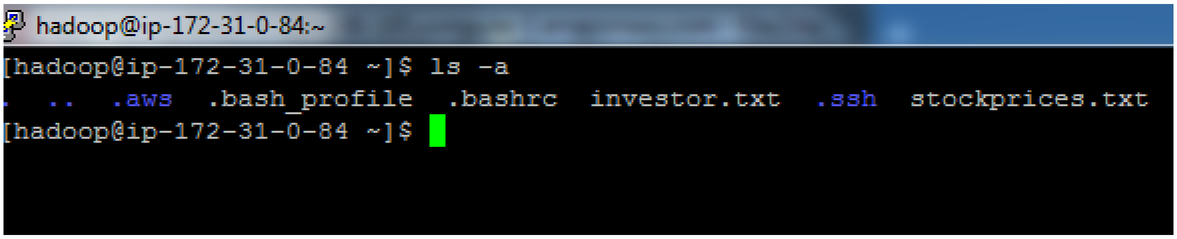
Uploading all the files into the Hadoop, and then creating the respective directories.



Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
.aws		File folder	12/13/2019 8:2...	drwxr-xr-x	hadoop had...
.ssh		File folder	5/2/2020 5:07:1...	drwx------	hadoop had...
.bash_profile	86	BASH_PRO...	12/1/2019 9:32:...	-rw-r--r--	hadoop had...
.bashrc	357	BASHRC File	12/1/2019 9:32:...	-rw-r--r--	hadoop had...
investor.txt	748	Text Docu...	5/2/2020 5:09:1...	-rw-rw-r--	hadoop had...
stockprices.txt	34	Text Docu...	5/2/2020 5:09:1...	-rw-rw-r--	hadoop had...

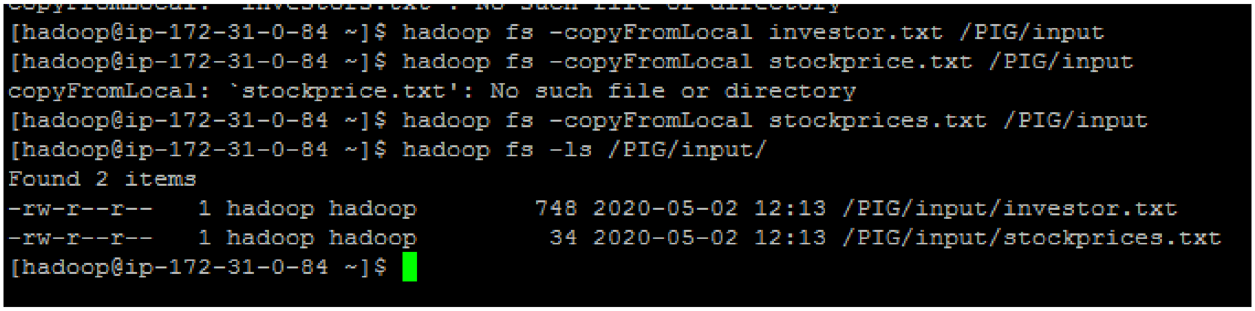
4 files and 2 directories. Total size: 1,225 bytes

Listing them all,



```
[hadoop@ip-172-31-0-84 ~]$ ls -a
.  ..  .aws  .bash_profile  .bashrc  investor.txt  .ssh  stockprices.txt
[hadoop@ip-172-31-0-84 ~]$ 
```

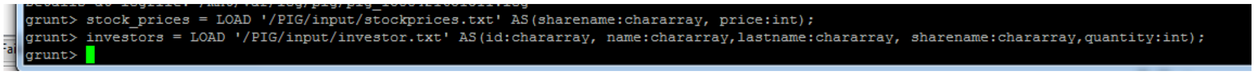
Then moving them to the respective folder as shown in the screenshot.



```
copyFromLocal: `investors.txt': No such file or directory
[hadoop@ip-172-31-0-84 ~]$ hadoop fs -copyFromLocal investor.txt /PIG/input
[hadoop@ip-172-31-0-84 ~]$ hadoop fs -copyFromLocal stockprice.txt /PIG/input
copyFromLocal: `stockprice.txt': No such file or directory
[hadoop@ip-172-31-0-84 ~]$ hadoop fs -copyFromLocal stockprices.txt /PIG/input
[hadoop@ip-172-31-0-84 ~]$ hadoop fs -ls /PIG/input/
Found 2 items
-rw-r--r--   1 hadoop hadoop        748 2020-05-02 12:13 /PIG/input/investor.txt
-rw-r--r--   1 hadoop hadoop         34 2020-05-02 12:13 /PIG/input/stockprices.txt
[hadoop@ip-172-31-0-84 ~]$ 
```

2. **Load files into as 'investors' and 'stock_prices'; Display both files to make sure they are loaded correctly**

   Loading the two files as shown below,



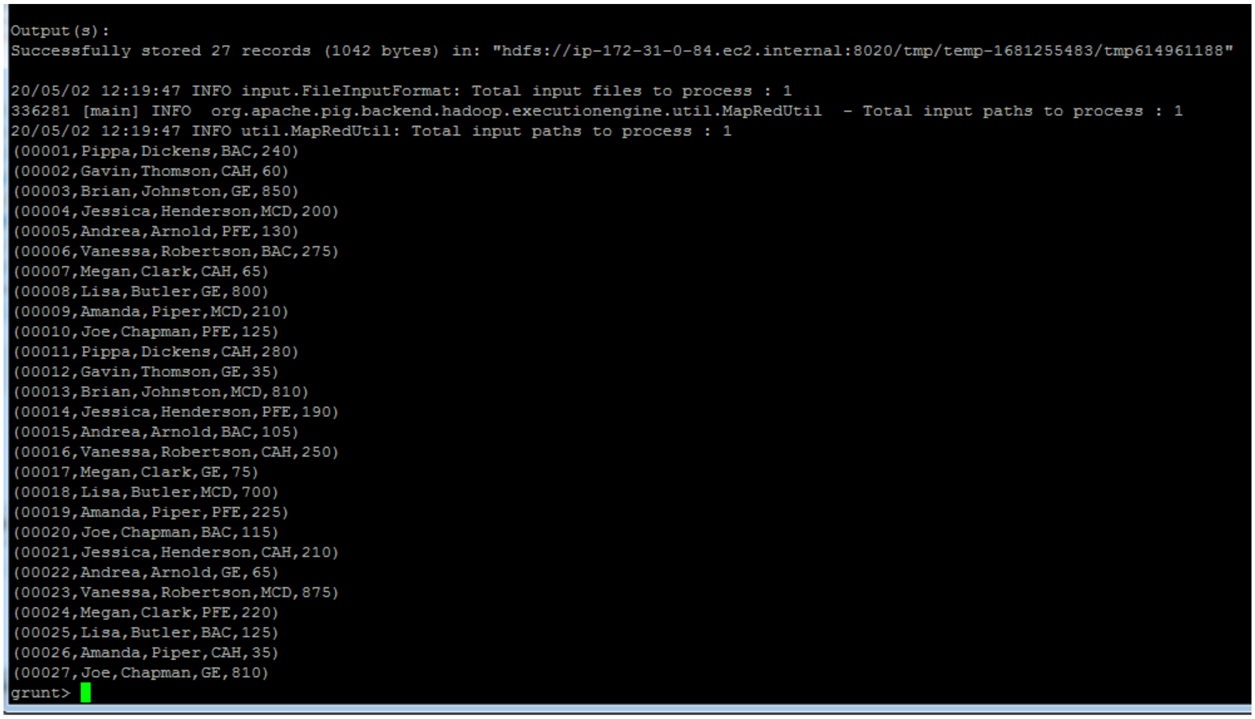
```
grunt> stock_prices = LOAD '/PIG/input/stockprices.txt' AS(sharename:chararray, price:int);
grunt> investors = LOAD '/PIG/input/investor.txt' AS(id:chararray, name:chararray,lastname:chararray, sharename:chararray,quantity:int);
grunt>
```

If the text is a bit smaller then these are the commands use for this:

```
investors = LOAD '/PIG/input/investor.txt' AS(id:chararray,
name:chararray,lastname:chararray, sharename:chararray,quantity:int);


stock_prices = LOAD '/PIG/input/stockprices.txt' AS(sharename:chararray,
price:int);
```

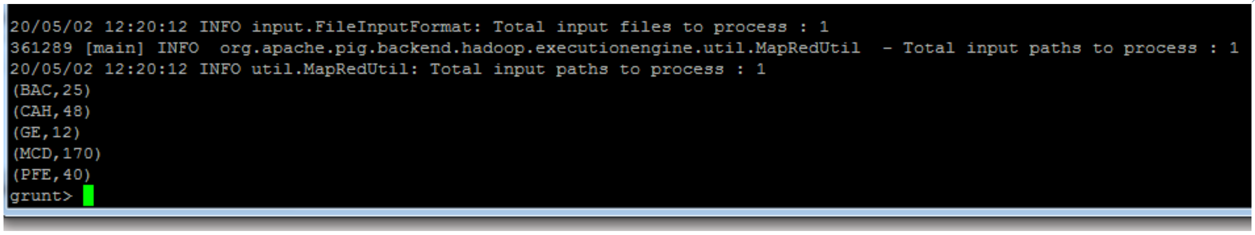
and displaying their data by dumping the new created tables as shown in the respective screenshots;

```
dump investors;
```



```
Output(s):
Successfully stored 27 records (1042 bytes) in: "hdfs://ip-172-31-0-84.ec2.internal:8020/tmp/temp-1681255483/tmp614961188"

20/05/02 12:19:47 INFO input.FileInputFormat: Total input files to process : 1
336281 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 12:19:47 INFO util.MapRedUtil: Total input paths to process : 1
(00001,Pippa,Dickens,BAC,240)
(00002,Gavin,Thomson,CAH,60)
(00003,Brian,Johnston,GE,850)
(00004,Jessica,Henderson,MCD,200)
(00005,Andrea,Arnold,PFE,130)
(00006,Vanessa,Robertson,BAC,275)
(00007,Megan,Clark,CAH,65)
(00008,Lisa,Butler,GE,800)
(00009,Amanda,Piper,MCD,210)
(00010,Joe,Chapman,PFE,125)
(00011,Pippa,Dickens,CAH,280)
(00012,Gavin,Thomson,GE,35)
(00013,Brian,Johnston,MCD,810)
(00014,Jessica,Henderson,PFE,190)
(00015,Andrea,Arnold,BAC,105)
(00016,Vanessa,Robertson,CAH,250)
(00017,Megan,Clark,GE,75)
(00018,Lisa,Butler,MCD,700)
(00019,Amanda,Piper,PFE,225)
(00020,Joe,Chapman,BAC,115)
(00021,Jessica,Henderson,CAH,210)
(00022,Andrea,Arnold,GE,65)
(00023,Vanessa,Robertson,MCD,875)
(00024,Megan,Clark,PFE,220)
(00025,Lisa,Butler,BAC,125)
(00026,Amanda,Piper,CAH,35)
(00027,Joe,Chapman,GE,810)
grunt>
```

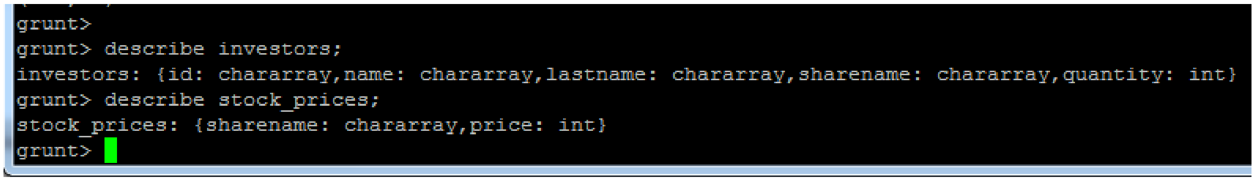
```
dump stock_prices;
```



```
20/05/02 12:20:12 INFO input.FileInputFormat: Total input files to process : 1
361289 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 12:20:12 INFO util.MapRedUtil: Total input paths to process : 1
(BAC,25)
(CAH,48)
(GE,12)
(MCD,170)
(PFE,40)
grunt>
```

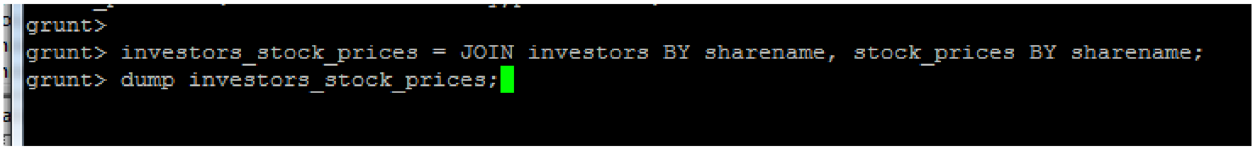
3. Display the structure of relation 'investors' and 'stock_prices'.

Describing the structure of the tables,

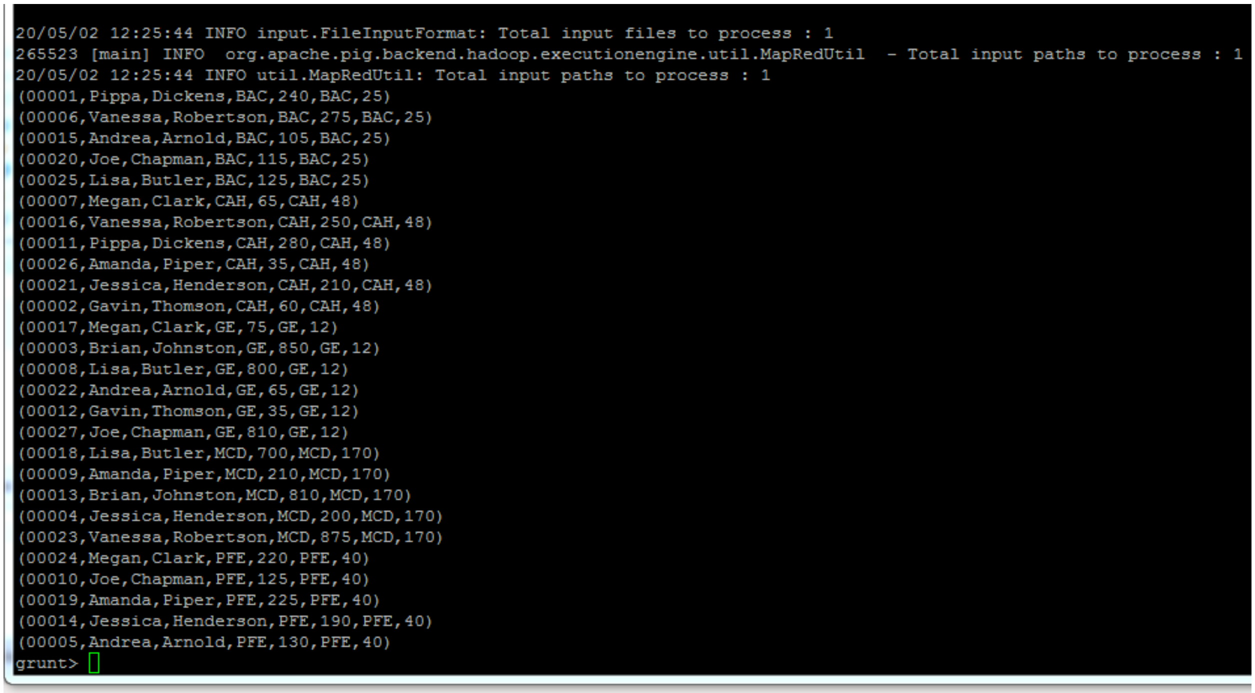


```
grunt>
grunt> describe investors;
investors: {id: chararray,name: chararray,lastname: chararray,sharename: chararray,quantity: int}
grunt> describe stock_prices;
stock_prices: {sharename: chararray,price: int}
grunt>
```

4. Join the two files ('investors' and 'stock_prices') by stock symbol; Display the joined file.



```
grunt>
grunt> investors_stock_prices = JOIN investors BY sharename, stock_prices BY sharename;
grunt> dump investors_stock_prices;
```

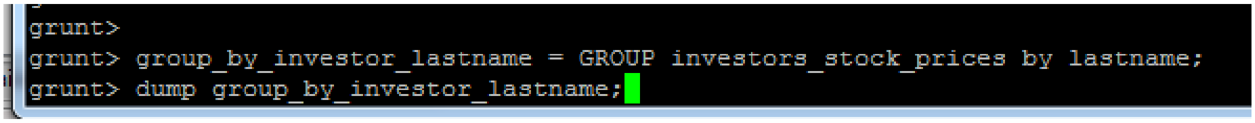


```
20/05/02 12:25:44 INFO input.FileInputFormat: Total input files to process : 1
265523 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 12:25:44 INFO util.MapRedUtil: Total input paths to process : 1
(00001,Pippa,Dickens,BAC,240,BAC,25)
(00006,Vanessa,Robertson,BAC,275,BAC,25)
(00015,Andrea,Arnold,BAC,105,BAC,25)
(00020,Joe,Chapman,BAC,115,BAC,25)
(00025,Lisa,Butler,BAC,125,BAC,25)
(00007,Megan,Clark,CAH,65,CAH,48)
(00016,Vanessa,Robertson,CAH,250,CAH,48)
(00011,Pippa,Dickens,CAH,280,CAH,48)
(00026,Amanda,Piper,CAH,35,CAH,48)
(00021,Jessica,Henderson,CAH,210,CAH,48)
(00002,Gavin,Thomson,CAH,60,CAH,48)
(00017,Megan,Clark,GE,75,GE,12)
(00003,Brian,Johnston,GE,850,GE,12)
(00008,Lisa,Butler,GE,800,GE,12)
(00022,Andrea,Arnold,GE,65,GE,12)
(00012,Gavin,Thomson,GE,35,GE,12)
(00027,Joe,Chapman,GE,810,GE,12)
(00018,Lisa,Butler,MCD,700,MCD,170)
(00009,Amanda,Piper,MCD,210,MCD,170)
(00013,Brian,Johnston,MCD,810,MCD,170)
(00004,Jessica,Henderson,MCD,200,MCD,170)
(00023,Vanessa,Robertson,MCD,875,MCD,170)
(00024,Megan,Clark,PFE,220,PFE,40)
(00010,Joe,Chapman,PFE,125,PFE,40)
(00019,Amanda,Piper,PFE,225,PFE,40)
(00014,Jessica,Henderson,PFE,190,PFE,40)
(00005,Andrea,Arnold,PFE,130,PFE,40)
grunt>
```

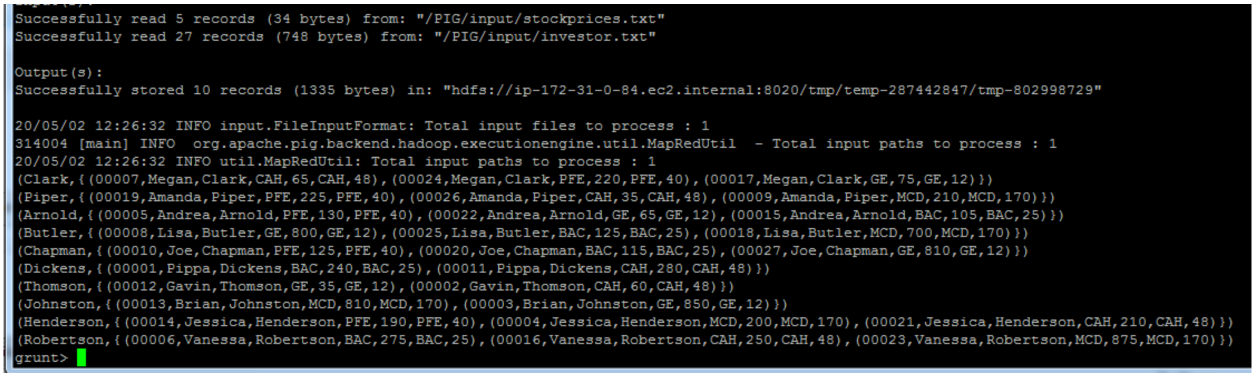
Joining the two tables as the below screenshot shows.

5. A Group the above file (joined file) by the 'lastname' of the investors; Display the results.

Grouping by 'lastname'



```
grunt>
grunt> group_by_investor_lastname = GROUP investors_stock_prices by lastname;
grunt> dump group_by_investor_lastname;
```



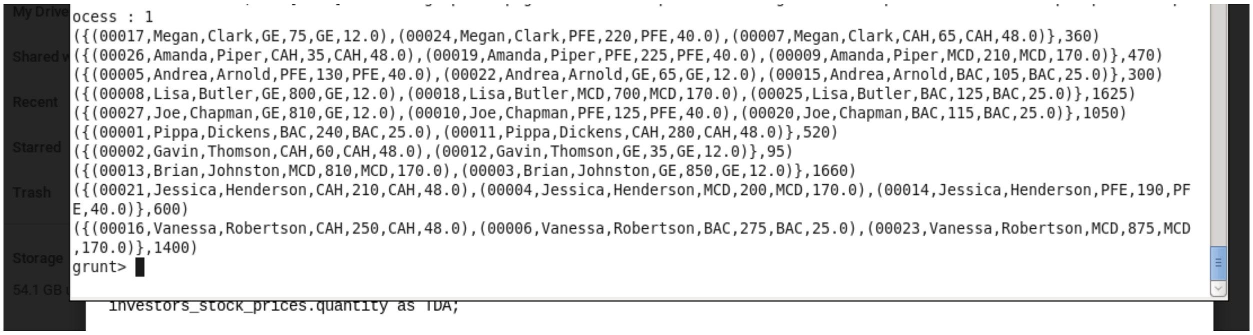
```
Successfully read 5 records (34 bytes) from: "/PIG/input/stockprices.txt"
Successfully read 27 records (748 bytes) from: "/PIG/input/investor.txt"

Output(s):
Successfully stored 10 records (1335 bytes) in: "hdfs://ip-172-31-0-84.ec2.internal:8020/tmp/temp-287442847/tmp-802998729"

20/05/02 12:26:32 INFO input.FileInputFormat: Total input files to process : 1
314004 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 12:26:32 INFO util.MapRedUtil: Total input paths to process : 1
(Clark,{(00007,Megan,Clark,CAH,65,CAH,48),(00024,Megan,Clark,PFE,220,PFE,40),(00017,Megan,Clark,GE,75,GE,12)})
(Piper,{(00019,Amanda,Piper,PFE,225,PFE,40),(00026,Amanda,Piper,CAH,35,CAH,48),(00009,Amanda,Piper,MCD,210,MCD,170)})
(Arnold,{(00005,Andrea,Arnold,PFE,130,PFE,40),(00022,Andrea,Arnold,GE,65,GE,12),(00015,Andrea,Arnold,BAC,105,BAC,25)})
(Butler,{(00008,Lisa,Butler,GE,800,GE,12),(00025,Lisa,Butler,BAC,125,BAC,25),(00018,Lisa,Butler,MCD,700,MCD,170)})
(Chapman,{(00010,Joe,Chapman,PFE,125,PFE,40),(00020,Joe,Chapman,BAC,115,BAC,25),(00027,Joe,Chapman,GE,810,GE,12)})
(Dickens,{(00001,Pippa,Dickens,BAC,240,BAC,25),(00011,Pippa,Dickens,CAH,280,CAH,48)})
(Thomson,{(00012,Gavin,Thomson,GE,35,GE,12),(00002,Gavin,Thomson,CAH,60,CAH,48)})
(Johnston,{(00013,Brian,Johnston,MCD,810,MCD,170),(00003,Brian,Johnston,GE,850,GE,12)})
(Henderson,{(00014,Jessica,Henderson,PFE,190,PFE,40),(00004,Jessica,Henderson,MCD,200,MCD,170),(00021,Jessica,Henderson,CAH,210,CAH,48)})
(Robertson,{(00006,Vanessa,Robertson,BAC,275,BAC,25),(00016,Vanessa,Robertson,CAH,250,CAH,48),(00023,Vanessa,Robertson,MCD,875,MCD,170)})
grunt>
```

6. Calculate the total shares (simply the sum of shares among all stocks); Display results.

On this point I have provided two codes, based on how I understood this part, I had doubts so I made two codes,

```
total_shares = FOREACH group_by_investor_lastname GENERATE
investors_stock_prices, SUM(investors_stock_prices.quantity);
```

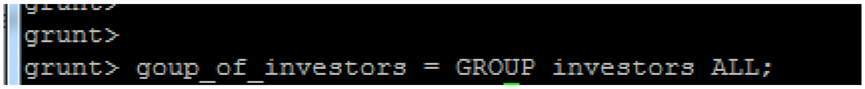
This code will generate a sum for each investor, and testing it on cloudera shows the following output:



```
ocess : 1
({(00017,Megan,Clark,GE,75,GE,12.0),(00024,Megan,Clark,PFE,220,PFE,40.0),(00007,Megan,Clark,CAH,65,CAH,48.0)},360)
({(00026,Amanda,Piper,CAH,35,CAH,48.0),(00019,Amanda,Piper,PFE,225,PFE,40.0),(00009,Amanda,Piper,MCD,210,MCD,170.0)},470)
({(00005,Andrea,Arnold,PFE,130,PFE,40.0),(00022,Andrea,Arnold,GE,65,GE,12.0),(00015,Andrea,Arnold,BAC,105,BAC,25.0)},300)
({(00008,Lisa,Butler,GE,800,GE,12.0),(00018,Lisa,Butler,MCD,700,MCD,170.0),(00025,Lisa,Butler,BAC,125,BAC,25.0)},1625)
({(00027,Joe,Chapman,GE,810,GE,12.0),(00010,Joe,Chapman,PFE,125,PFE,40.0),(00020,Joe,Chapman,BAC,115,BAC,25.0)},1050)
({(00001,Pippa,Dickens,BAC,240,BAC,25.0),(00011,Pippa,Dickens,CAH,280,CAH,48.0)},520)
({(00002,Gavin,Thomson,CAH,60,CAH,48.0),(00012,Gavin,Thomson,GE,35,GE,12.0)},95)
({(00013,Brian,Johnston,MCD,810,MCD,170.0),(00003,Brian,Johnston,GE,850,GE,12.0)},1660)
({(00021,Jessica,Henderson,CAH,210,CAH,48.0),(00004,Jessica,Henderson,MCD,200,MCD,170.0),(00014,Jessica,Henderson,PFE,190,PF
E,40.0)},600)
({(00016,Vanessa,Robertson,CAH,250,CAH,48.0),(00006,Vanessa,Robertson,BAC,275,BAC,25.0),(00023,Vanessa,Robertson,MCD,875,MCD
,170.0)},1400)
grunt>
```

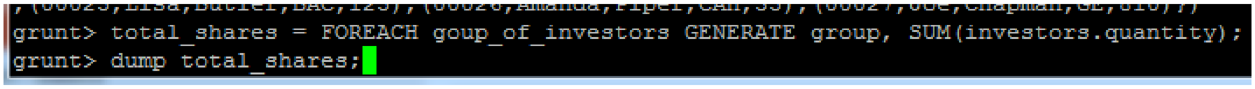
```
investors_stock_prices.quantity as TDA;
```

And the other version which I think is the right one is just to get a total sum of all the investors shares quantity, so first we group the investors table,

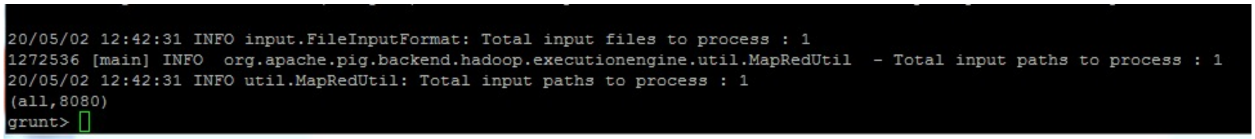


```
grunt>
grunt> goup_of_investors = GROUP investors ALL;
```

Then for each grouped investor in the created group we calculate the amount of shares in total as shown below;



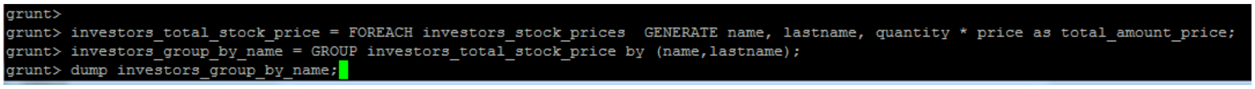
```
grunt> total_shares = FOREACH goup_of_investors GENERATE group, SUM(investors.quantity);
grunt> dump total_shares;
```



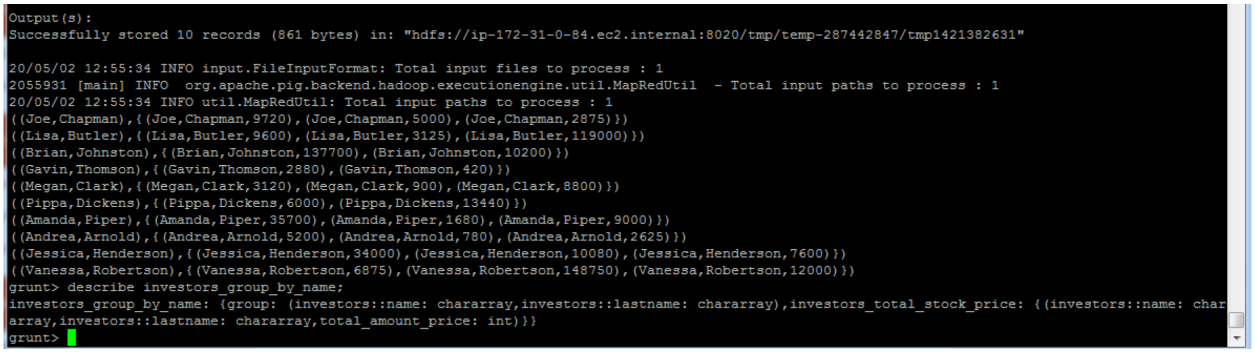
```
20/05/02 12:42:31 INFO input.FileInputFormat: Total input files to process : 1
1272536 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 12:42:31 INFO util.MapRedUtil: Total input paths to process : 1
(all,8080)
grunt>
```

7. **Calculate the total dollar amount that each investor has invested (shares per each stock multiplied by the stock price); Display the results.**

To do this we generate a new table from 'investors_stock_prices' from exercise 4 and we include the quantity of shares * the quantity as an extra column as shown below, and also, we group the investors on the second command according to the first table.

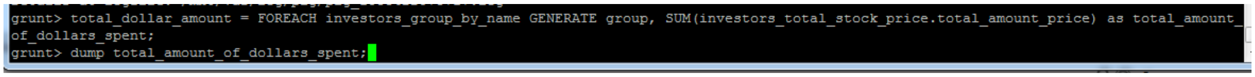


```
grunt>
grunt> investors_total_stock_price = FOREACH investors_stock_prices  GENERATE name, lastname, quantity * price as total_amount_price;
grunt> investors_group_by_name = GROUP investors_total_stock_price by (name,lastname);
grunt> dump investors_group_by_name;
```

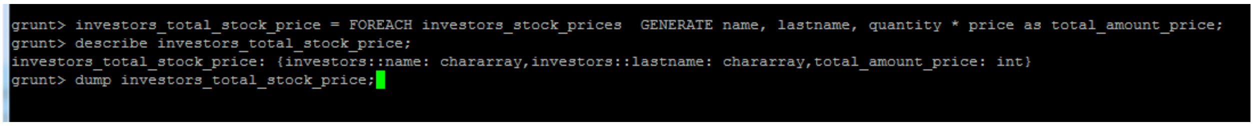


```
Output(s):
Successfully stored 10 records (861 bytes) in: "hdfs://ip-172-31-0-84.ec2.internal:8020/tmp/temp-287442847/tmp1421382631"

20/05/02 12:55:34 INFO input.FileInputFormat: Total input files to process : 1
2055931 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 12:55:34 INFO util.MapRedUtil: Total input paths to process : 1
((Joe,Chapman),{(Joe,Chapman,9720),(Joe,Chapman,5000),(Joe,Chapman,2875)})
((Lisa,Butler),{(Lisa,Butler,9600),(Lisa,Butler,3125),(Lisa,Butler,119000)})
((Brian,Johnston),{(Brian,Johnston,137700),(Brian,Johnston,10200)})
((Gavin,Thomson),{(Gavin,Thomson,2880),(Gavin,Thomson,420)})
((Megan,Clark),{(Megan,Clark,3120),(Megan,Clark,900),(Megan,Clark,8800)})
((Pippa,Dickens),{(Pippa,Dickens,6000),(Pippa,Dickens,13440)})
((Amanda,Piper),{(Amanda,Piper,35700),(Amanda,Piper,1680),(Amanda,Piper,9000)})
((Andrea,Arnold),{(Andrea,Arnold,5200),(Andrea,Arnold,780),(Andrea,Arnold,2625)})
((Jessica,Henderson),{(Jessica,Henderson,34000),(Jessica,Henderson,10080),(Jessica,Henderson,7600)})
((Vanessa,Robertson),{(Vanessa,Robertson,6875),(Vanessa,Robertson,148750),(Vanessa,Robertson,12000)})
grunt> describe investors_group_by_name;
investors_group_by_name: {group: (investors::name: chararray,investors::lastname: chararray),investors_total_stock_price: {(investors::name: char
array,investors::lastname: chararray,total_amount_price: int)}}
grunt>
```

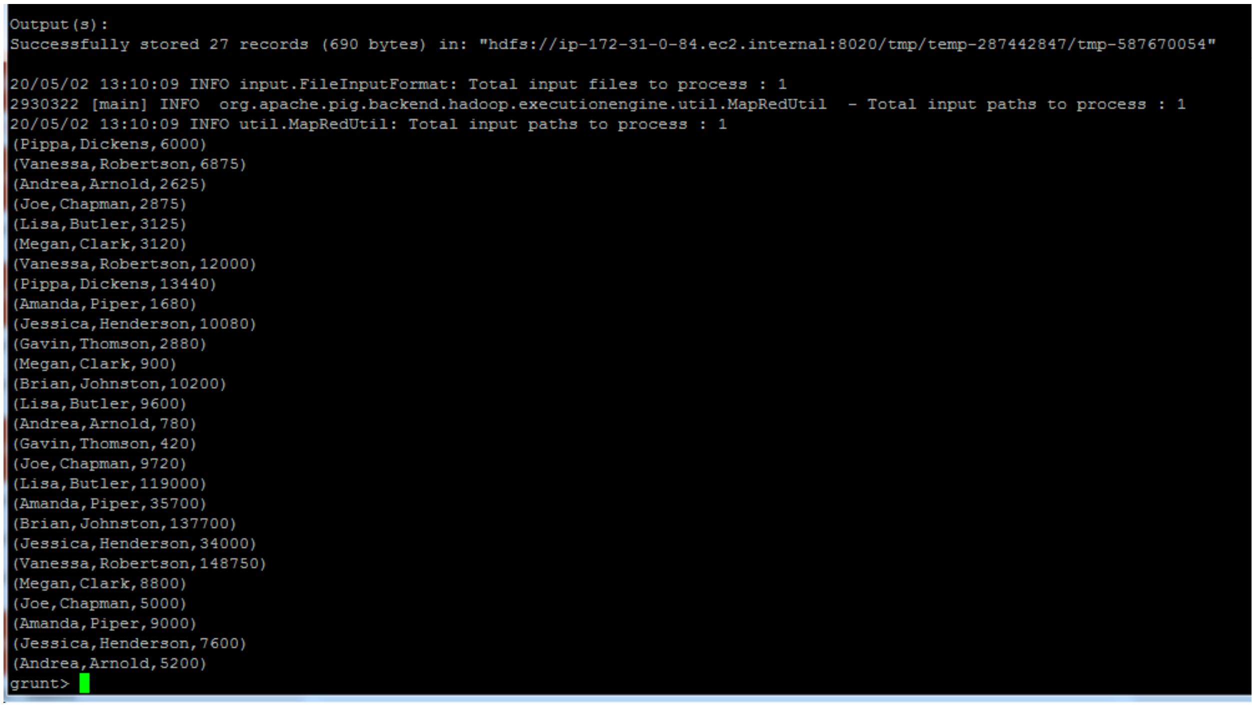
And then we get the total amount of dollars according to the two previous created tables.



```
grunt> total_dollar_amount = FOREACH investors_group_by_name GENERATE group, SUM(investors_total_stock_price.total_amount_price) as total_amount_
of_dollars_spent;
grunt> dump total_amount_of_dollars_spent;
```



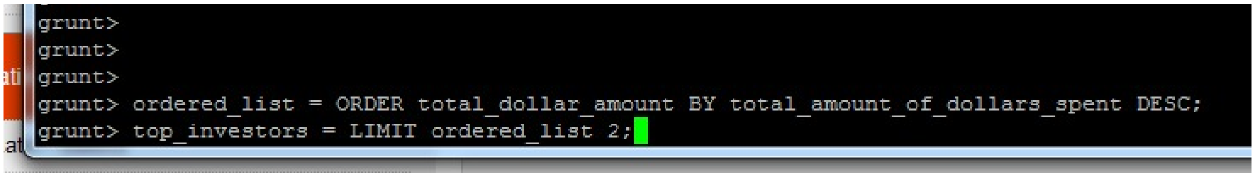
```
grunt> investors_total_stock_price = FOREACH investors_stock_prices  GENERATE name, lastname, quantity * price as total_amount_price;
grunt> describe investors_total_stock_price;
investors_total_stock_price: {investors::name: chararray,investors::lastname: chararray,total_amount_price: int}
grunt> dump investors_total_stock_price;
```



```
Output(s):
Successfully stored 27 records (690 bytes) in: "hdfs://ip-172-31-0-84.ec2.internal:8020/tmp/temp-287442847/tmp-587670054"

20/05/02 13:10:09 INFO input.FileInputFormat: Total input files to process : 1
2930322 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 13:10:09 INFO util.MapRedUtil: Total input paths to process : 1
(Pippa,Dickens,6000)
(Vanessa,Robertson,6875)
(Andrea,Arnold,2625)
(Joe,Chapman,2875)
(Lisa,Butler,3125)
(Megan,Clark,3120)
(Vanessa,Robertson,12000)
(Pippa,Dickens,13440)
(Amanda,Piper,1680)
(Jessica,Henderson,10080)
(Gavin,Thomson,2880)
(Megan,Clark,900)
(Brian,Johnston,10200)
(Lisa,Butler,9600)
(Andrea,Arnold,780)
(Gavin,Thomson,420)
(Joe,Chapman,9720)
(Lisa,Butler,119000)
(Amanda,Piper,35700)
(Brian,Johnston,137700)
(Jessica,Henderson,34000)
(Vanessa,Robertson,148750)
(Megan,Clark,8800)
(Joe,Chapman,5000)
(Amanda,Piper,9000)
(Jessica,Henderson,7600)
(Andrea,Arnold,5200)
grunt>
```

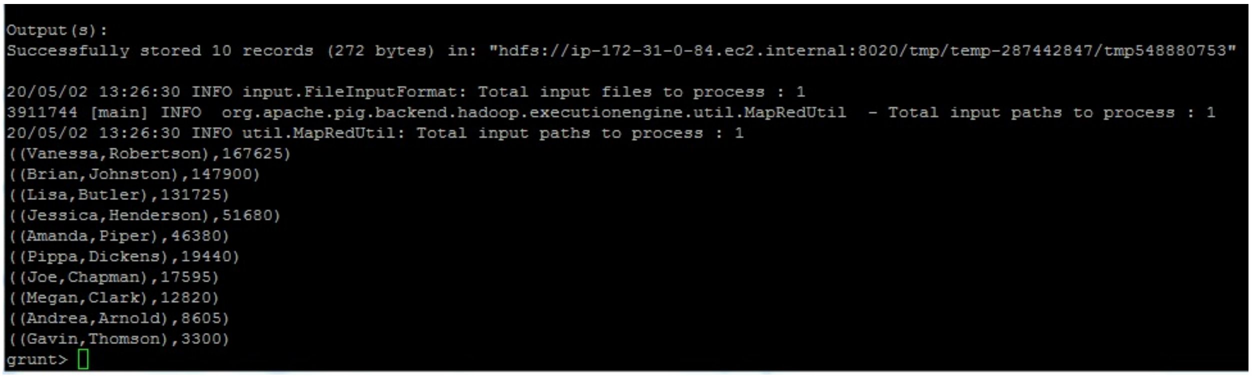
8. Filter the top two investors that have invested the most; Display the results.

Ordering the list in descending order as shown below;



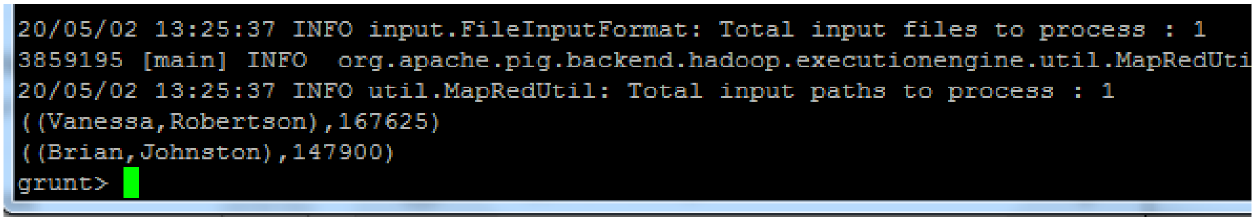
```
grunt>
grunt>
grunt>
grunt> ordered_list = ORDER total_dollar_amount BY total_amount_of_dollars_spent DESC;
grunt> top_investors = LIMIT ordered_list 2;
```

The dumped ordered list would be:



```
Output(s):
Successfully stored 10 records (272 bytes) in: "hdfs://ip-172-31-0-84.ec2.internal:8020/tmp/temp-287442847/tmp548880753"

20/05/02 13:26:30 INFO input.FileInputFormat: Total input files to process : 1
3911744 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil  - Total input paths to process : 1
20/05/02 13:26:30 INFO util.MapRedUtil: Total input paths to process : 1
((Vanessa,Robertson),167625)
((Brian,Johnston),147900)
((Lisa,Butler),131725)
((Jessica,Henderson),51680)
((Amanda,Piper),46380)
((Pippa,Dickens),19440)
((Joe,Chapman),17595)
((Megan,Clark),12820)
((Andrea,Arnold),8605)
((Gavin,Thomson),3300)
grunt>
```

And then displaying tow top investors;



```
20/05/02 13:25:37 INFO input.FileInputFormat: Total input files to process : 1
3859195 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUti
20/05/02 13:25:37 INFO util.MapRedUtil: Total input paths to process : 1
((Vanessa,Robertson),167625)
((Brian,Johnston),147900)
grunt> 
```

Note: This homework was worked in a virtual machine and using windows 7, the exercises we worked in parallel using the Cloudera on Cent OS and AWS, the information shown is based on AWS services. Before starting the homework, using a text editor for simplicity like notepad++ to not mixing the names and variables and the commands 'Homework4_PIG_COMMANDS .txt" and the total log.txt of PuTTy terminal is also included, even if in some cases there might be wrong commands which are fixed after attempting sometimes, everything worked perfectly.