Title: - Online Transaction Analysis

Project Author: - Sandeep Kumar Bhandoria

BACKGROUND: I am a Hadoop Analyst in my company OTA Pvt Ltd. My company give the suggestions to other company who shares their transaction data with us.

OBJECTIVE: A company that is planning to surprise their customers for the upcoming events for Christmas and New Year and they also wants some more suggestion so that they can make decision based on Online Transaction data.

My objective is to analyse the data and come up which some use case and solution for that.

DATA WE HAVE:

1. **Transaction's Data:**

File Used: txns-large.dat

TransactionID	T_date	UserId	Price	Product_Cat	Product
00000000	06-26-2015	4000003	040.33	Exercise & Fitness	Cardio Machine Accessories
00000001	06-01-2015	4009775	005.58	Outdoor Recreation	Archery

2. **Customer's Data:**

File Used: Customer.dat

UserID	FirstName	LastName	Age	Profession
4000001	Kristina	Chug	55	Pilot
4000002	Paige	Chen	74	Teacher
4000003	Sherri	Melton	34	Fire fighter
4000004	Karen	Puckett	74	Lawyer
4000005	Elsie	Hamilton	43	Pilot

TECHNOLOGY WE USED:

1) Apache Hadoop
2) Map Reduce programming in Java
3) HIVE
4) PIG

SOFTWARE WE USED:

1) Virtual Box
2) Eclipse
3) Ubuntu

PROJECT DESCRIPTION: Project Use the map reduce Hadoop programming to achieve the result of various different use case. We have used map side join as well as reduce side join for different tasks.

USE CASE 1

Scenario: - Heavy price based transactions that company have.

1) We find all the transaction or products based on the user defined prices.

   In the case we are expecting input from user for the value of amount on which we have to decide the transaction.

2) This can be used to find the transaction done on a specific price from where we can get products name that the users are interested in for a specific price.

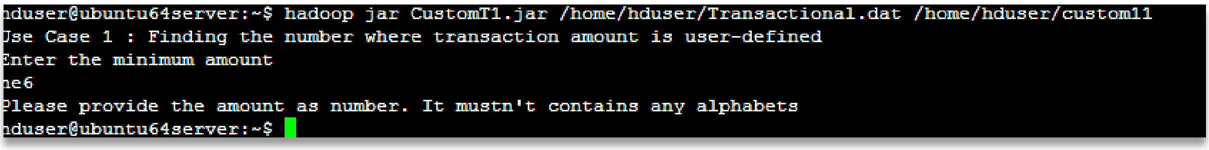
Validation:

We have done a check on the user input before processing it further.

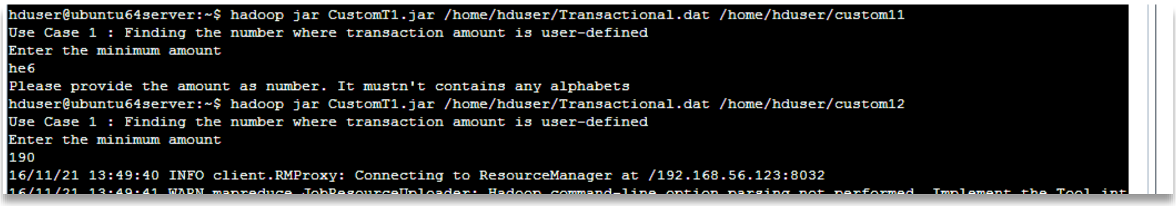
1) User can have to specify a minimum price and based on that price all transaction will be filter where price is greater than what user has specified.

2) ***If the user is passing String in place of number we will be displayed an error message*** to provide valid input and start the job again.

Output screenshot: -

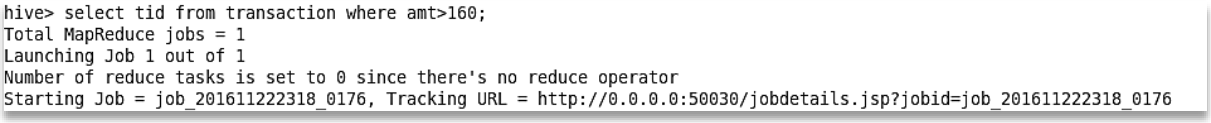


```
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom11
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
he6
Please provide the amount as number. It mustn't contains any alphabets
hduser@ubuntu64server:~$
```

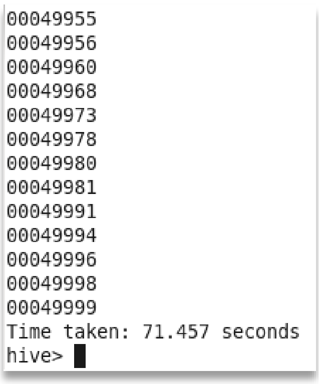


```
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom11
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
he6
Please provide the amount as number. It mustn't contains any alphabets
hduser@ubuntu64server:~$ hadoop jar CustomT1.jar /home/hduser/Transactional.dat /home/hduser/custom12
Use Case 1 : Finding the number where transaction amount is user-defined
Enter the minimum amount
190
16/11/21 13:49:40 INFO client.RMProxy: Connecting to ResourceManager at /192.168.56.123:8032
16/11/21 13:49:41 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool int
```

HIVE Output for same task

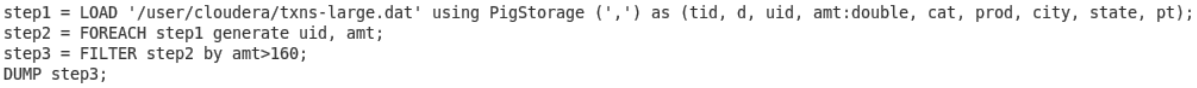


```
hive> select tid from transaction where amt>160;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201611222318_0176, Tracking URL = http://0.0.0.0:50030/jobdetails.jsp?jobid=job_201611222318_0176
```

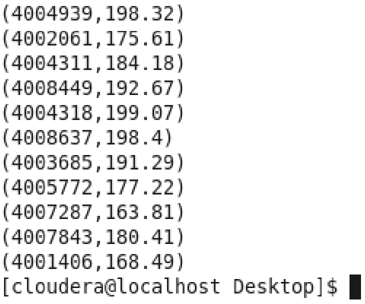


```
00049955
00049956
00049960
00049968
00049973
00049978
00049980
00049981
00049991
00049994
00049996
00049998
00049999
Time taken: 71.457 seconds
hive>
```

PIG Output for same task



```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate uid, amt;
step3 = FILTER step2 by amt>160;
DUMP step3;
```



```
(4004939,198.32)
(4002061,175.61)
(4004311,184.18)
(4008449,192.67)
(4004318,199.07)
(4008637,198.4)
(4003685,191.29)
(4005772,177.22)
(4007287,163.81)
(4007843,180.41)
(4001406,168.49)
[cloudera@localhost Desktop]$
```

USE CASE 2

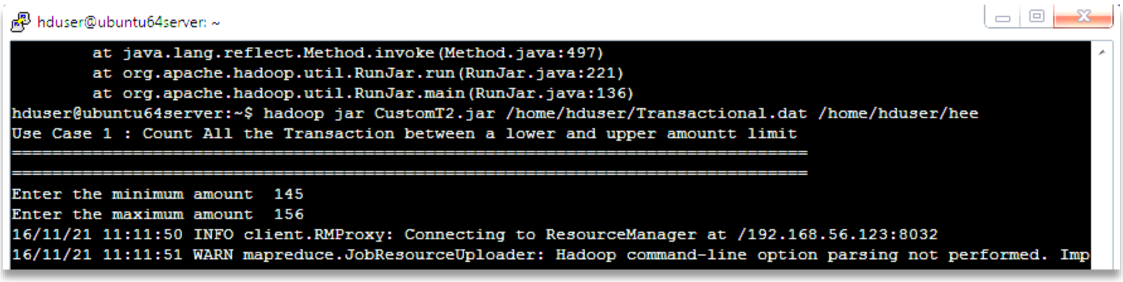
Scenario: - Price Range Based Products

1) We will find the number of products we have for a particular range of price.

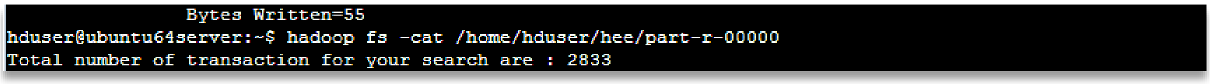
Validation:

1) We will be accepting user input for minimum and maximum limit for price.
2) ***The maximum price can't be less than minimum price. If attempted a message will be displayed for this.*** And also tells the user to run the task again with proper inputs.
3) ***Minimum amount can't be less than 0. Message will be displayed for the same.***
4) ***Maximum amount can't be less than 0. Message will be displayed for the same.***

Output screenshot: -

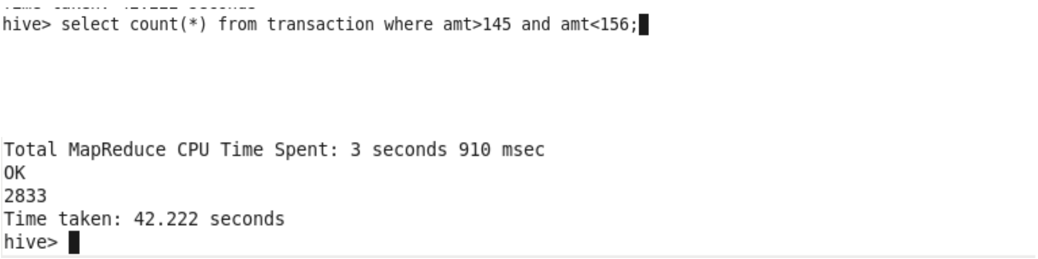


```
        at java.lang.reflect.Method.invoke(Method.java:497)
        at org.apache.hadoop.util.RunJar.run(RunJar.java:221)
        at org.apache.hadoop.util.RunJar.main(RunJar.java:136)
hduser@ubuntu64server:~$ hadoop jar CustomT2.jar /home/hduser/Transactional.dat /home/hduser/hee
Use Case 1 : Count All the Transaction between a lower and upper amountt limit
===============================================================================
===============================================================================
Enter the minimum amount  145
Enter the maximum amount  156
16/11/21 11:11:50 INFO client.RMProxy: Connecting to ResourceManager at /192.168.56.123:8032
16/11/21 11:11:51 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Imp
```



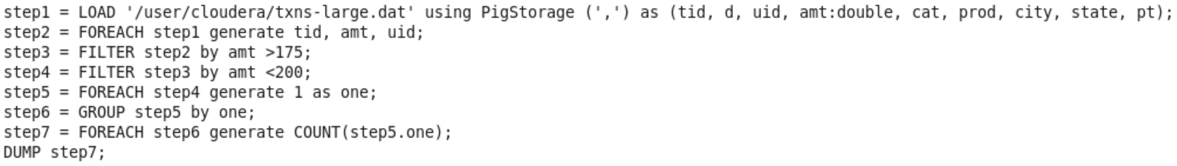
```
                  Bytes Written=55
hduser@ubuntu64server:~$ hadoop fs -cat /home/hduser/hee/part-r-00000
Total number of transaction for your search are : 2833
```

HIVE Output for same task

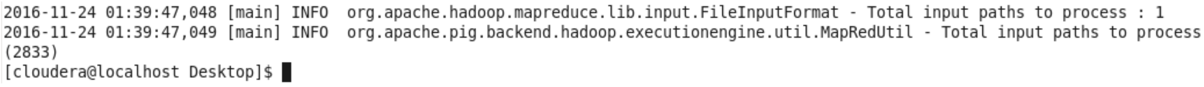


```
hive> select count(*) from transaction where amt>145 and amt<156;



Total MapReduce CPU Time Spent: 3 seconds 910 msec
OK
2833
Time taken: 42.222 seconds
hive>
```

PIG Output for same task



```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate tid, amt, uid;
step3 = FILTER step2 by amt >175;
step4 = FILTER step3 by amt <200;
step5 = FOREACH step4 generate 1 as one;
step6 = GROUP step5 by one;
step7 = FOREACH step6 generate COUNT(step5.one);
DUMP step7;
```



```
2016-11-24 01:39:47,048 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2016-11-24 01:39:47,049 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process
(2833)
[cloudera@localhost Desktop]$ 
```

USE CASE 3

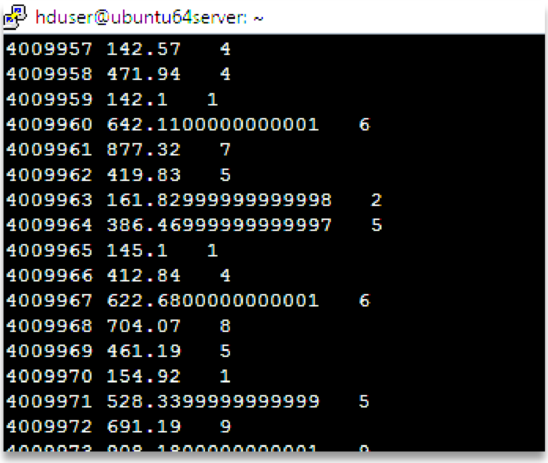
Scenario: - Customers wise transaction and purchase

1) The Company is planning for a scheme to give offers to customers based on

   a) Their past number of transactions.
   b) Total purchase they did.

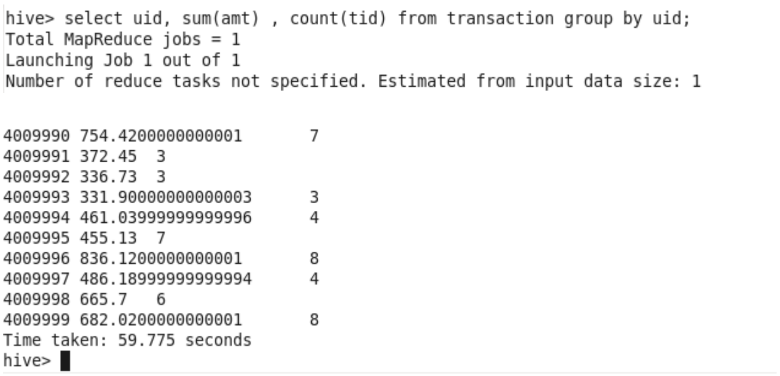
This requires an analysis to prepare a report per each user.

Output screenshot: -



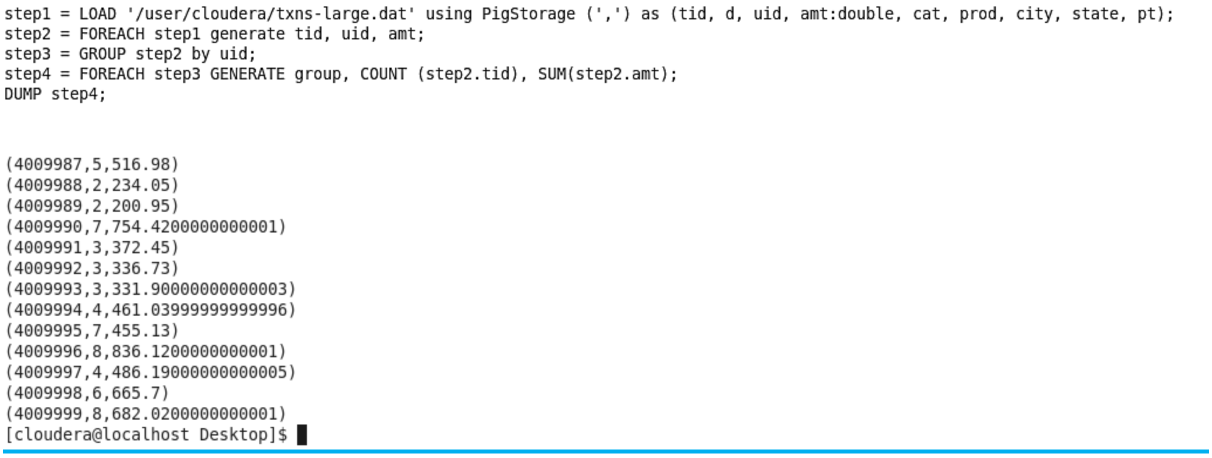
```
4009957 142.57    4
4009958 471.94    4
4009959 142.1   1
4009960 642.1100000000001    6
4009961 877.32    7
4009962 419.83    5
4009963 161.82999999999998    2
4009964 386.46999999999997    5
4009965 145.1   1
4009966 412.84    4
4009967 622.6800000000001    6
4009968 704.07    8
4009969 461.19    5
4009970 154.92    1
4009971 528.3399999999999    5
4009972 691.19    9
4009973 808.1800000000001    9
```

HIVE Output for same task



```
hive> select uid, sum(amt) , count(tid) from transaction group by uid;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1


4009990 754.4200000000001       7
4009991 372.45  3
4009992 336.73  3
4009993 331.90000000000003      3
4009994 461.03999999999996      4
4009995 455.13  7
4009996 836.1200000000001       8
4009997 486.18999999999994      4
4009998 665.7   6
4009999 682.0200000000001       8
Time taken: 59.775 seconds
hive> 
```

PIG Output for same task



```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate tid, uid, amt;
step3 = GROUP step2 by uid;
step4 = FOREACH step3 GENERATE group, COUNT (step2.tid), SUM(step2.amt);
DUMP step4;


(4009987,5,516.98)
(4009988,2,234.05)
(4009989,2,200.95)
(4009990,7,754.4200000000001)
(4009991,3,372.45)
(4009992,3,336.73)
(4009993,3,331.90000000000003)
(4009994,4,461.03999999999996)
(4009995,7,455.13)
(4009996,8,836.1200000000001)
(4009997,4,486.19000000000005)
(4009998,6,665.7)
(4009999,8,682.0200000000001)
[cloudera@localhost Desktop]$
```

USE CASE 4

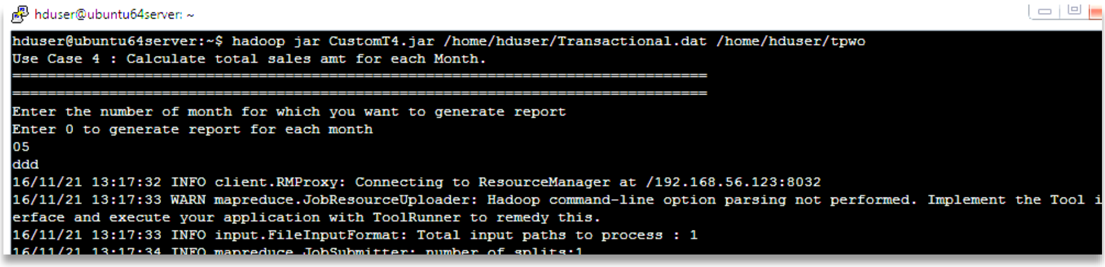
Scenario: - Monthly Wise Revenue

At the end of every year your company wants to do an analysis to know in which month people usually comes for shopping.

Validation:

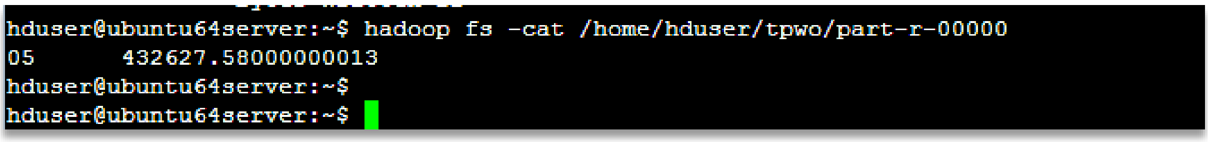
1) We will be accepting user input for month.
2) ***The month must be between 1 and 12 .Any number except these we display an error.*** And also tells the user to run the task again with proper inputs.
3) ***User can also input month in 3 character abbreviated form like feb, JAN etc.***

Output screenshot: -



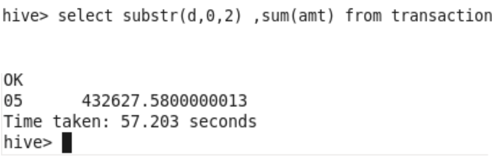
hduser@ubuntu64server: ~

```
hduser@ubuntu64server:~$ hadoop jar CustomT4.jar /home/hduser/Transactional.dat /home/hduser/tpwo
Use Case 4 : Calculate total sales amt for each Month.
===============================================================================
===============================================================================
Enter the number of month for which you want to generate report
Enter 0 to generate report for each month
05
ddd
16/11/21 13:17:32 INFO client.RMProxy: Connecting to ResourceManager at /192.168.56.123:8032
16/11/21 13:17:33 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool i
erface and execute your application with ToolRunner to remedy this.
16/11/21 13:17:33 INFO input.FileInputFormat: Total input paths to process : 1
16/11/21 13:17:34 INFO mapreduce.JobSubmitter: number of splits:1
```



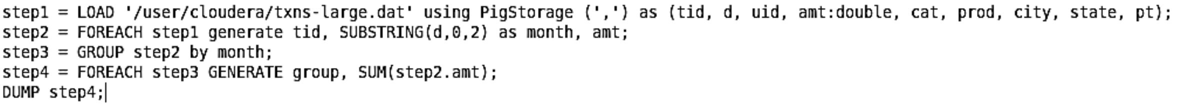
```
hduser@ubuntu64server:~$ hadoop fs -cat /home/hduser/tpwo/part-r-00000
05      432627.58000000013
hduser@ubuntu64server:~$
hduser@ubuntu64server:~$
```

HIVE Output for same task



```
hive> select substr(d,0,2) ,sum(amt) from transaction where substr(d,0,2)=='05' group by substr(d,0,2);

OK
05      432627.5800000013
Time taken: 57.203 seconds
hive>
```

PIG Output for same task



```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate tid, SUBSTRING(d,0,2) as month, amt;
step3 = GROUP step2 by month;
step4 = FOREACH step3 GENERATE group, SUM(step2.amt);
DUMP step4;
```

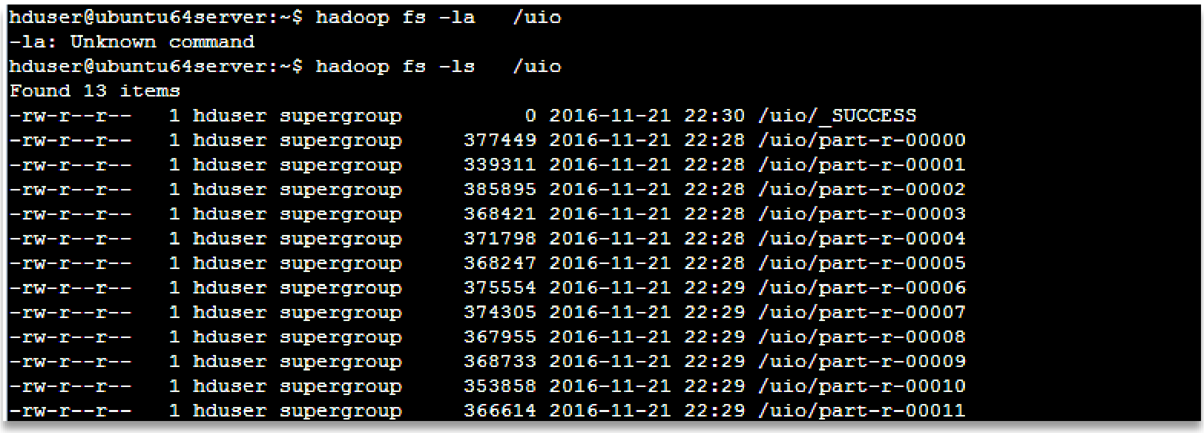
USE CASE 5

Scenario: - Monthly Wise Transaction Summary

Being a Hadoop Developer and Admin, You may need to partition your final data to make further processing easy.

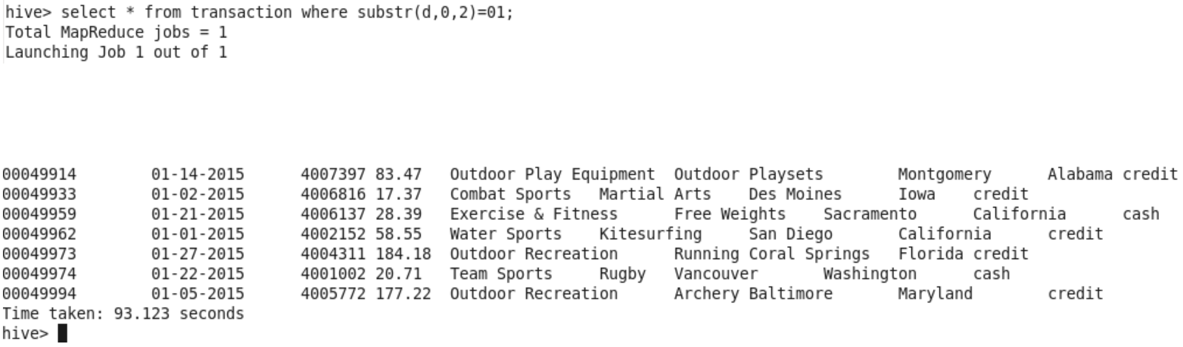
We have been asked to divide all the transaction based on the month and store each transaction according to the months. So12 files are created for this one for each month.

Output screenshot: -



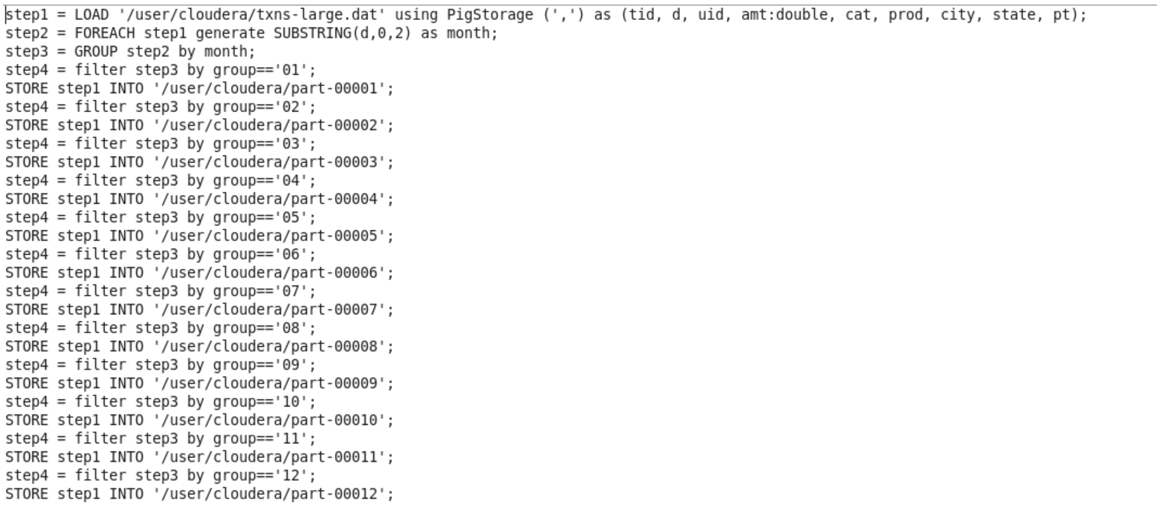
```
hduser@ubuntu64server:~$ hadoop fs -la   /uio
-la: Unknown command
hduser@ubuntu64server:~$ hadoop fs -ls   /uio
Found 13 items
-rw-r--r--   1 hduser supergroup          0 2016-11-21 22:30 /uio/_SUCCESS
-rw-r--r--   1 hduser supergroup     377449 2016-11-21 22:28 /uio/part-r-00000
-rw-r--r--   1 hduser supergroup     339311 2016-11-21 22:28 /uio/part-r-00001
-rw-r--r--   1 hduser supergroup     385895 2016-11-21 22:28 /uio/part-r-00002
-rw-r--r--   1 hduser supergroup     368421 2016-11-21 22:28 /uio/part-r-00003
-rw-r--r--   1 hduser supergroup     371798 2016-11-21 22:28 /uio/part-r-00004
-rw-r--r--   1 hduser supergroup     368247 2016-11-21 22:28 /uio/part-r-00005
-rw-r--r--   1 hduser supergroup     375554 2016-11-21 22:29 /uio/part-r-00006
-rw-r--r--   1 hduser supergroup     374305 2016-11-21 22:29 /uio/part-r-00007
-rw-r--r--   1 hduser supergroup     367955 2016-11-21 22:29 /uio/part-r-00008
-rw-r--r--   1 hduser supergroup     368733 2016-11-21 22:29 /uio/part-r-00009
-rw-r--r--   1 hduser supergroup     353858 2016-11-21 22:29 /uio/part-r-00010
-rw-r--r--   1 hduser supergroup     366614 2016-11-21 22:29 /uio/part-r-00011
```

HIVE Output for same task



```
hive> select * from transaction where substr(d,0,2)=01;
Total MapReduce jobs = 1
Launching Job 1 out of 1



00049914        01-14-2015      4007397 83.47   Outdoor Play Equipment  Outdoor Playsets        Montgomery      Alabama credit
00049933        01-02-2015      4006816 17.37   Combat Sports   Martial Arts    Des Moines      Iowa    credit
00049959        01-21-2015      4006137 28.39   Exercise & Fitness      Free Weights    Sacramento      California      cash
00049962        01-01-2015      4002152 58.55   Water Sports    Kitesurfing     San Diego       California      credit
00049973        01-27-2015      4004311 184.18  Outdoor Recreation      Running Coral Springs   Florida credit
00049974        01-22-2015      4001002 20.71   Team Sports     Rugby   Vancouver       Washington      cash
00049994        01-05-2015      4005772 177.22  Outdoor Recreation      Archery Baltimore       Maryland        credit
Time taken: 93.123 seconds
hive> 
```

PIG Output for same task



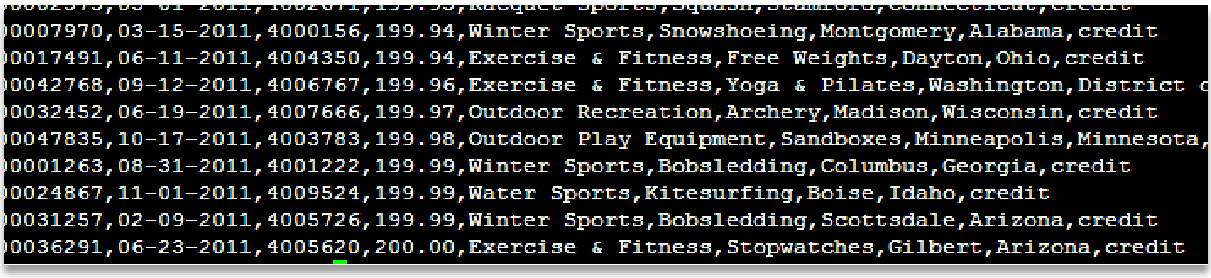
```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = FOREACH step1 generate SUBSTRING(d,0,2) as month;
step3 = GROUP step2 by month;
step4 = filter step3 by group=='01';
STORE step1 INTO '/user/cloudera/part-00001';
step4 = filter step3 by group=='02';
STORE step1 INTO '/user/cloudera/part-00002';
step4 = filter step3 by group=='03';
STORE step1 INTO '/user/cloudera/part-00003';
step4 = filter step3 by group=='04';
STORE step1 INTO '/user/cloudera/part-00004';
step4 = filter step3 by group=='05';
STORE step1 INTO '/user/cloudera/part-00005';
step4 = filter step3 by group=='06';
STORE step1 INTO '/user/cloudera/part-00006';
step4 = filter step3 by group=='07';
STORE step1 INTO '/user/cloudera/part-00007';
step4 = filter step3 by group=='08';
STORE step1 INTO '/user/cloudera/part-00008';
step4 = filter step3 by group=='09';
STORE step1 INTO '/user/cloudera/part-00009';
step4 = filter step3 by group=='10';
STORE step1 INTO '/user/cloudera/part-00010';
step4 = filter step3 by group=='11';
STORE step1 INTO '/user/cloudera/part-00011';
step4 = filter step3 by group=='12';
STORE step1 INTO '/user/cloudera/part-00012';
```

USE CASE 6

Scenario: -File sorting based on price

We have the transaction file and this file will be sorted based on the amounts available in each transaction.

Output screenshot: -



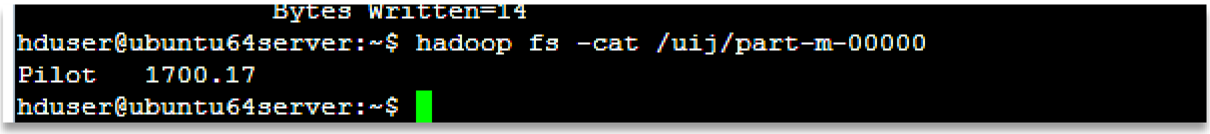
```
0002575,03-01-2011,4002071,199.93,Racquet Sports,Squash,Stamford,Connecticut,credit
0007970,03-15-2011,4000156,199.94,Winter Sports,Snowshoeing,Montgomery,Alabama,credit
0017491,06-11-2011,4004350,199.94,Exercise & Fitness,Free Weights,Dayton,Ohio,credit
0042768,09-12-2011,4006767,199.96,Exercise & Fitness,Yoga & Pilates,Washington,District c
0032452,06-19-2011,4007666,199.97,Outdoor Recreation,Archery,Madison,Wisconsin,credit
0047835,10-17-2011,4003783,199.98,Outdoor Play Equipment,Sandboxes,Minneapolis,Minnesota,
0001263,08-31-2011,4001222,199.99,Winter Sports,Bobsledding,Columbus,Georgia,credit
0024867,11-01-2011,4009524,199.99,Water Sports,Kitesurfing,Boise,Idaho,credit
0031257,02-09-2011,4005726,199.99,Winter Sports,Bobsledding,Scottsdale,Arizona,credit
0036291,06-23-2011,4005620,200.00,Exercise & Fitness,Stopwatches,Gilbert,Arizona,credit
```

USE CASE 7

Scenario: - Top profession who does shopping the most

1) Company wants to target the particular area where people are more interested in their products so we have analysed the top profession.

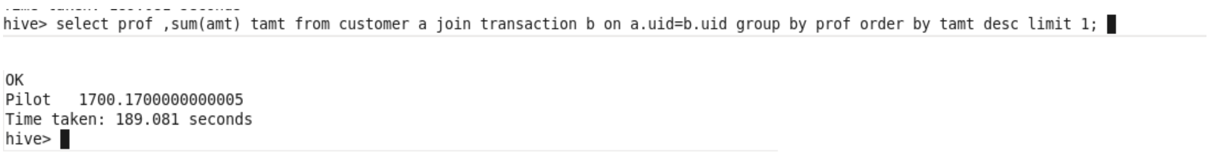
Output screenshot: -



```
                Bytes Written=14
hduser@ubuntu64server:~$ hadoop fs -cat /uij/part-m-00000
Pilot   1700.17
hduser@ubuntu64server:~$
```

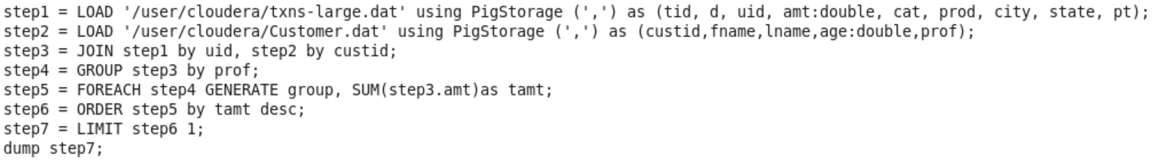
The customers who are pilot are doing more transactions.

HIVE Output for same task

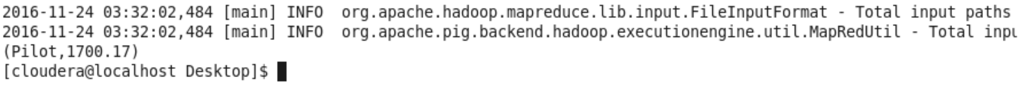


```
hive> select prof ,sum(amt) tamt from customer a join transaction b on a.uid=b.uid group by prof order by tamt desc limit 1;

OK
Pilot   1700.1700000000005
Time taken: 189.081 seconds
hive>
```

PIG Output for same task



```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = LOAD '/user/cloudera/Customer.dat' using PigStorage (',') as (custid,fname,lname,age:double,prof);
step3 = JOIN step1 by uid, step2 by custid;
step4 = GROUP step3 by prof;
step5 = FOREACH step4 GENERATE group, SUM(step3.amt)as tamt;
step6 = ORDER step5 by tamt desc;
step7 = LIMIT step6 1;
dump step7;
```



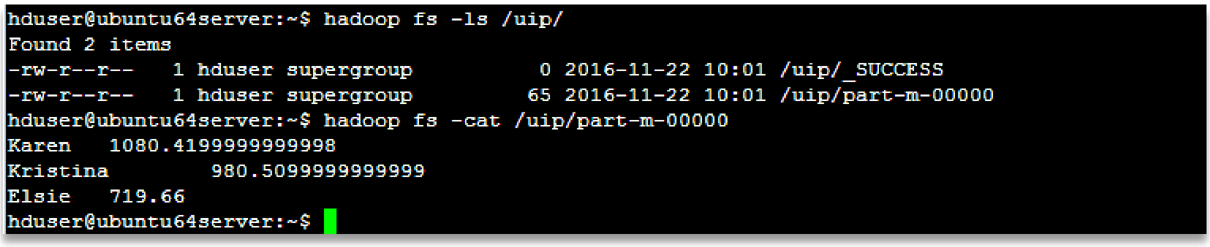
```
2016-11-24 03:32:02,484 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths
2016-11-24 03:32:02,484 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total inpu
(Pilot,1700.17)
[cloudera@localhost Desktop]$
```

USE CASE 8

Scenario: -Analyze Top 3 customers to give additional rewards.

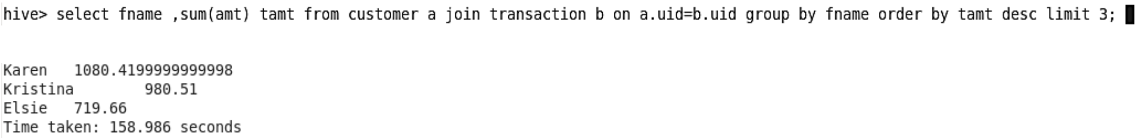
1) Our online shopping website wants to give rewards to some top 3 customers.

Output screenshot: -



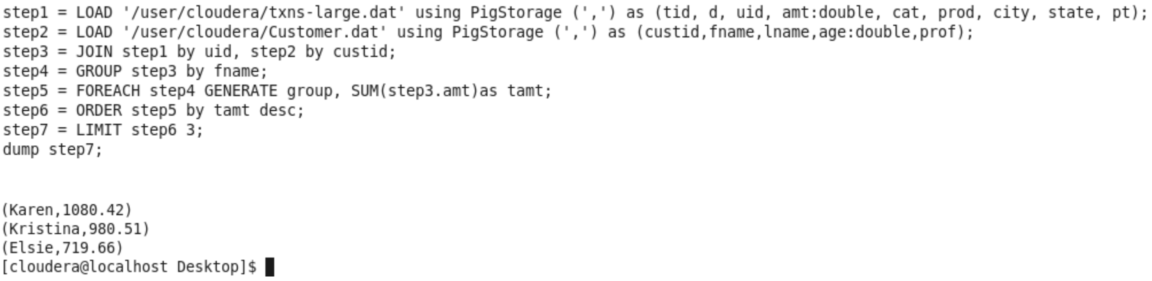
```
hduser@ubuntu64server:~$ hadoop fs -ls /uip/
Found 2 items
-rw-r--r--   1 hduser supergroup          0 2016-11-22 10:01 /uip/_SUCCESS
-rw-r--r--   1 hduser supergroup         65 2016-11-22 10:01 /uip/part-m-00000
hduser@ubuntu64server:~$ hadoop fs -cat /uip/part-m-00000
Karen   1080.4199999999998
Kristina        980.5099999999999
Elsie   719.66
hduser@ubuntu64server:~$
```

HIVE Output for same task



```
hive> select fname ,sum(amt) tamt from customer a join transaction b on a.uid=b.uid group by fname order by tamt desc limit 3;

Karen   1080.4199999999998
Kristina        980.51
Elsie   719.66
Time taken: 158.986 seconds
```

PIG Output for same task



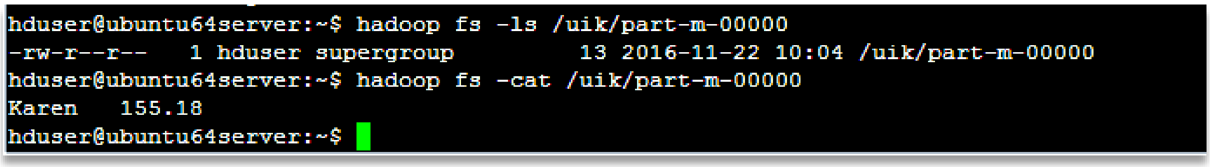
```
step1 = LOAD '/user/cloudera/txns-large.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = LOAD '/user/cloudera/Customer.dat' using PigStorage (',') as (custid,fname,lname,age:double,prof);
step3 = JOIN step1 by uid, step2 by custid;
step4 = GROUP step3 by fname;
step5 = FOREACH step4 GENERATE group, SUM(step3.amt)as tamt;
step6 = ORDER step5 by tamt desc;
step7 = LIMIT step6 3;
dump step7;


(Karen,1080.42)
(Kristina,980.51)
(Elsie,719.66)
[cloudera@localhost Desktop]$
```

USE CASE 9

Scenario: - Month Wise top customer

1) We have analysed the data to get the top customer for a specific month July.

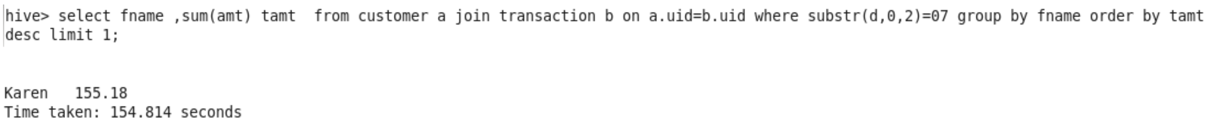
Output screenshot: -



```
hduser@ubuntu64server:~$ hadoop fs -ls /uik/part-m-00000
-rw-r--r--   1 hduser supergroup         13 2016-11-22 10:04 /uik/part-m-00000
hduser@ubuntu64server:~$ hadoop fs -cat /uik/part-m-00000
Karen   155.18
hduser@ubuntu64server:~$
```

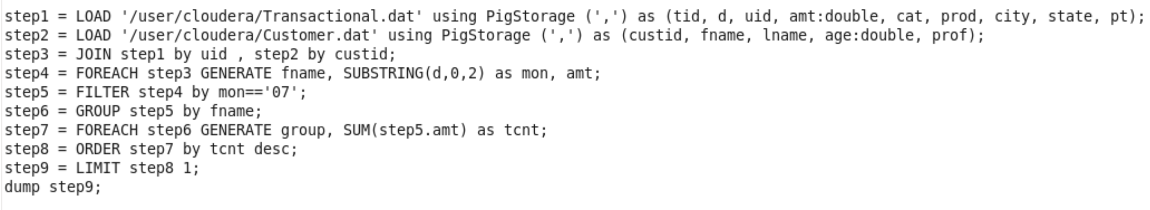
Karen is the top customer who spent the most for online shipping.

HIVE Output for same task

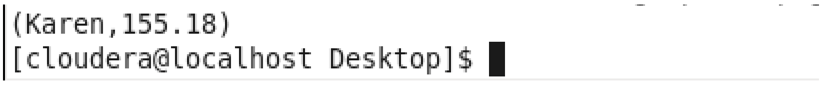


```
hive> select fname ,sum(amt) tamt  from customer a join transaction b on a.uid=b.uid where substr(d,0,2)=07 group by fname order by tamt
desc limit 1;


Karen   155.18
Time taken: 154.814 seconds
```

PIG Output for same task



```
step1 = LOAD '/user/cloudera/Transactional.dat' using PigStorage (',') as (tid, d, uid, amt:double, cat, prod, city, state, pt);
step2 = LOAD '/user/cloudera/Customer.dat' using PigStorage (',') as (custid, fname, lname, age:double, prof);
step3 = JOIN step1 by uid , step2 by custid;
step4 = FOREACH step3 GENERATE fname, SUBSTRING(d,0,2) as mon, amt;
step5 = FILTER step4 by mon=='07';
step6 = GROUP step5 by fname;
step7 = FOREACH step6 GENERATE group, SUM(step5.amt) as tcnt;
step8 = ORDER step7 by tcnt desc;
step9 = LIMIT step8 1;
dump step9;
```



```
(Karen,155.18)
[cloudera@localhost Desktop]$
```

CONCLUSION - Above Data Analysis shows that we can get various information using map reduce Hadoop processing to make better decision in E-commerce Industry which will help the website owner in providing better service for their customers.