WEEK 2

ASSIGNMENT – SOLUTION

1.Login to your Gateway node & open a terminal

2.write a command to know what's your home directory in gateway node

3.There is a third party service which will drop a file named orders.csv in the landing folder under your home directory. Then you need to filter for all the orders where status is PENDING_PAYMENT & create a new file named orders_filtered.csv and put it to the staging folder. Then take this file and put it to hdfs in landing folder in your hdfs and do couple of more things…

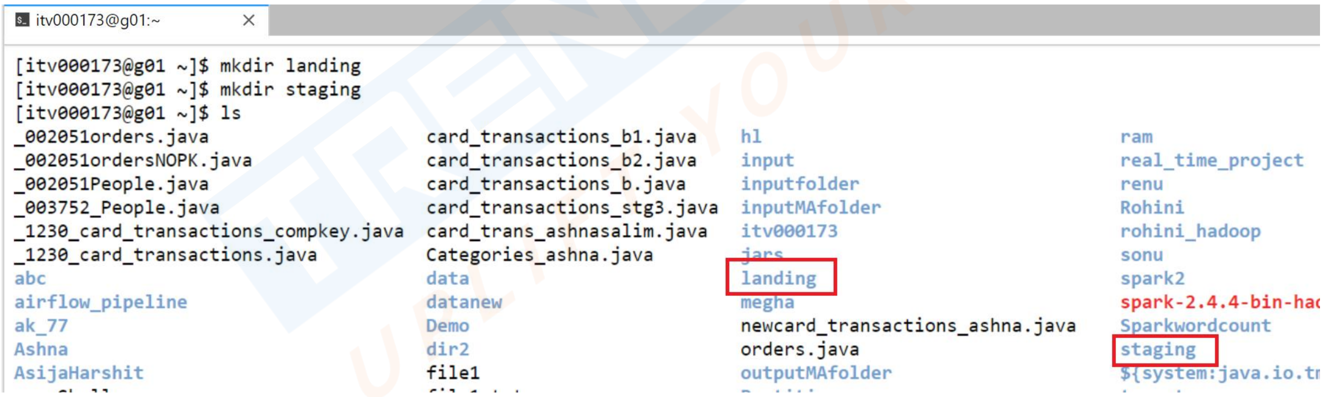
So to simulate this..

1.create two folders named landing and staging in your home directory.

cd ~

mkdir landing

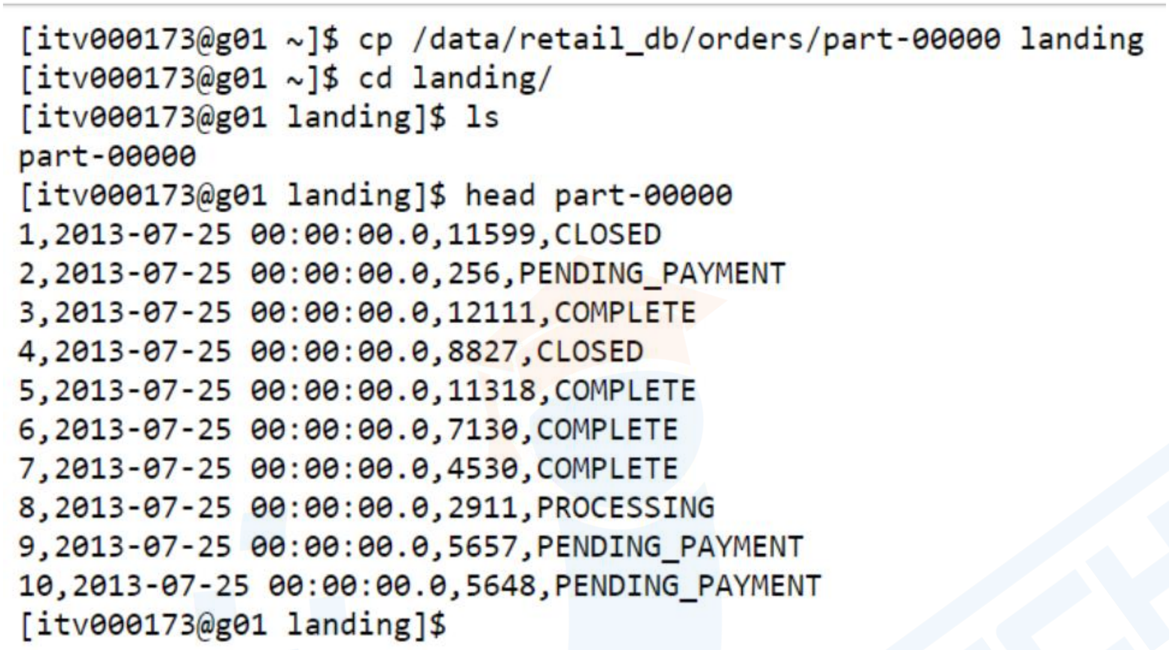
mkdir staging



```
[itv000173@g01 ~]$ mkdir landing
[itv000173@g01 ~]$ mkdir staging
[itv000173@g01 ~]$ ls
_002051orders.java                  card_transactions_b1.java    hl                              ram
_002051ordersNOPK.java              card_transactions_b2.java    input                           real_time_project
_002051People.java                  card_transactions_b.java     inputfolder                     renu
_003752_People.java                 card_transactions_stg3.java  inputMAfolder                   Rohini
_1230_card_transactions_compkey.java card_trans_ashnasalim.java  itv000173                       rohini_hadoop
_1230_card_transactions.java        Categories_ashna.java        jars                            sonu
abc                                 data                         landing                         spark2
airflow_pipeline                    datanew                      megha                           spark-2.4.4-bin-had
ak_77                               Demo                         newcard_transactions_ashna.java Sparkwordcount
Ashna                               dir2                         orders.java                     staging
AsijaHarshit                        file1                        outputMAfolder                  ${system:java.io.tr
```

2.copy the file present under /data/retail_db/orders folder to the landing folder in your home directory.

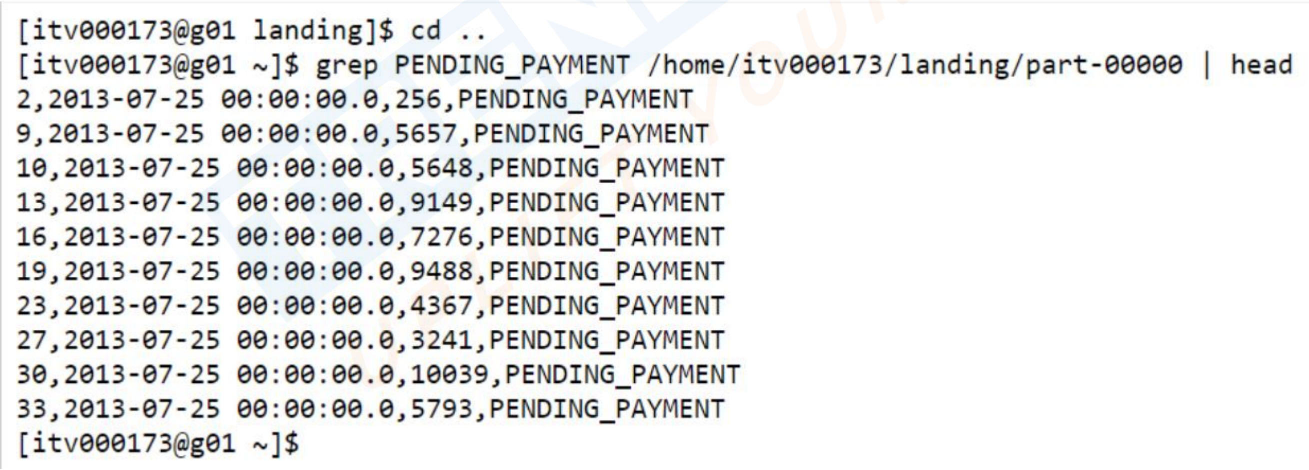
cp /data/retail_db/orders/part-00000 landing



```
[itv000173@g01 ~]$ cp /data/retail_db/orders/part-00000 landing
[itv000173@g01 ~]$ cd landing/
[itv000173@g01 landing]$ ls
part-00000
[itv000173@g01 landing]$ head part-00000
1,2013-07-25 00:00:00.0,11599,CLOSED
2,2013-07-25 00:00:00.0,256,PENDING_PAYMENT
3,2013-07-25 00:00:00.0,12111,COMPLETE
4,2013-07-25 00:00:00.0,8827,CLOSED
5,2013-07-25 00:00:00.0,11318,COMPLETE
6,2013-07-25 00:00:00.0,7130,COMPLETE
7,2013-07-25 00:00:00.0,4530,COMPLETE
8,2013-07-25 00:00:00.0,2911,PROCESSING
9,2013-07-25 00:00:00.0,5657,PENDING_PAYMENT
10,2013-07-25 00:00:00.0,5648,PENDING_PAYMENT
[itv000173@g01 landing]$
```

3.Apply the grep command to filter for all orders with PENDING_PAYMENT status.

grep PENDING_PAYMENT /home/itv000173/landing/part-00000 | head

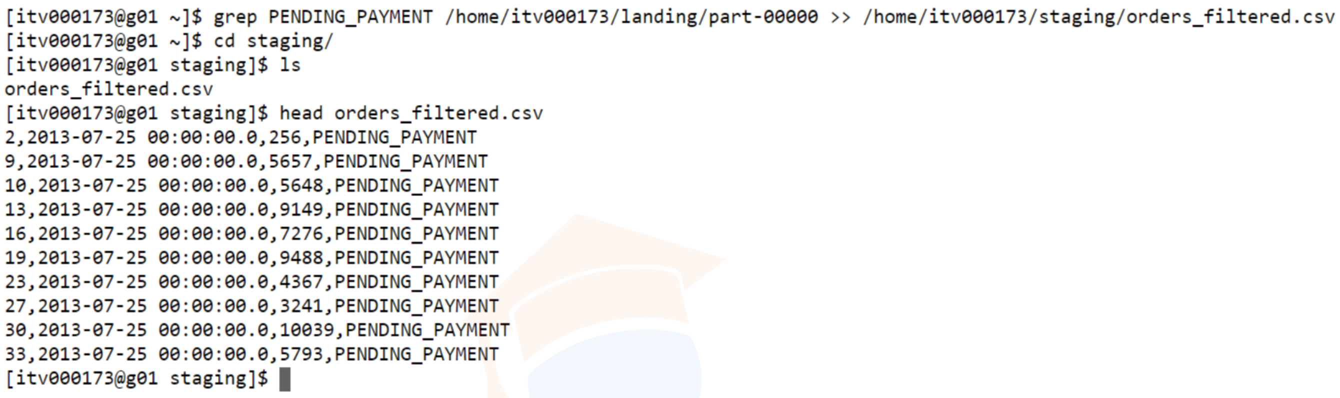


```
[itv000173@g01 landing]$ cd ..
[itv000173@g01 ~]$ grep PENDING_PAYMENT /home/itv000173/landing/part-00000 | head
2,2013-07-25 00:00:00.0,256,PENDING_PAYMENT
9,2013-07-25 00:00:00.0,5657,PENDING_PAYMENT
10,2013-07-25 00:00:00.0,5648,PENDING_PAYMENT
13,2013-07-25 00:00:00.0,9149,PENDING_PAYMENT
16,2013-07-25 00:00:00.0,7276,PENDING_PAYMENT
19,2013-07-25 00:00:00.0,9488,PENDING_PAYMENT
23,2013-07-25 00:00:00.0,4367,PENDING_PAYMENT
27,2013-07-25 00:00:00.0,3241,PENDING_PAYMENT
30,2013-07-25 00:00:00.0,10039,PENDING_PAYMENT
33,2013-07-25 00:00:00.0,5793,PENDING_PAYMENT
[itv000173@g01 ~]$
```

[Note: Here there are total 68881 records, hence only showcased first 10 records using head command]

4.create a new file named orders_filtered.csv under your staging folder with the filtered results.

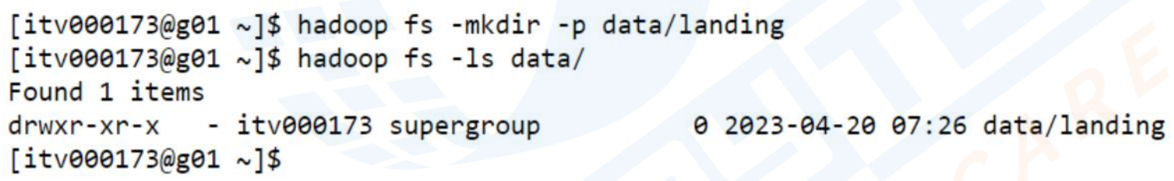
grep PENDING_PAYMENT /home/itv000173/landing/part-00000 >> /home/itv000173/staging/orders_filtered.csv



```
[itv000173@g01 ~]$ grep PENDING_PAYMENT /home/itv000173/landing/part-00000 >> /home/itv000173/staging/orders_filtered.csv
[itv000173@g01 ~]$ cd staging/
[itv000173@g01 staging]$ ls
orders_filtered.csv
[itv000173@g01 staging]$ head orders_filtered.csv
2,2013-07-25 00:00:00.0,256,PENDING_PAYMENT
9,2013-07-25 00:00:00.0,5657,PENDING_PAYMENT
10,2013-07-25 00:00:00.0,5648,PENDING_PAYMENT
13,2013-07-25 00:00:00.0,9149,PENDING_PAYMENT
16,2013-07-25 00:00:00.0,7276,PENDING_PAYMENT
19,2013-07-25 00:00:00.0,9488,PENDING_PAYMENT
23,2013-07-25 00:00:00.0,4367,PENDING_PAYMENT
27,2013-07-25 00:00:00.0,3241,PENDING_PAYMENT
30,2013-07-25 00:00:00.0,10039,PENDING_PAYMENT
33,2013-07-25 00:00:00.0,5793,PENDING_PAYMENT
[itv000173@g01 staging]$ █
```

5.create a folder hierarchy in your hdfs home named data/landing

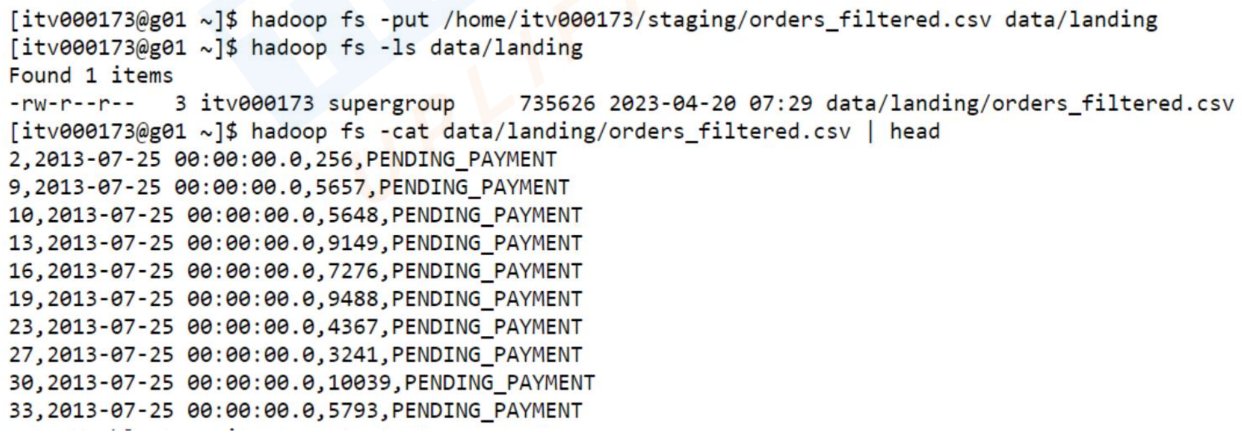
hadoop fs -mkdir -p data/landing



```
[itv000173@g01 ~]$ hadoop fs -mkdir -p data/landing
[itv000173@g01 ~]$ hadoop fs -ls data/
Found 1 items
drwxr-xr-x   - itv000173 supergroup          0 2023-04-20 07:26 data/landing
[itv000173@g01 ~]$
```

6.copy this orders_filtered.csv file from your staging folder in local to data/landing folder in your hdfs.

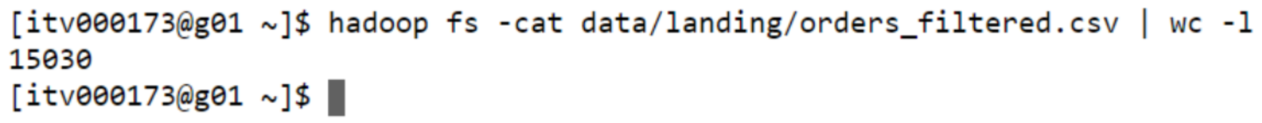
hadoop fs -put /home/itv000173/staging/orders_filtered.csv data/landing



```
[itv000173@g01 ~]$ hadoop fs -put /home/itv000173/staging/orders_filtered.csv data/landing
[itv000173@g01 ~]$ hadoop fs -ls data/landing
Found 1 items
-rw-r--r--   3 itv000173 supergroup     735626 2023-04-20 07:29 data/landing/orders_filtered.csv
[itv000173@g01 ~]$ hadoop fs -cat data/landing/orders_filtered.csv | head
2,2013-07-25 00:00:00.0,256,PENDING_PAYMENT
9,2013-07-25 00:00:00.0,5657,PENDING_PAYMENT
10,2013-07-25 00:00:00.0,5648,PENDING_PAYMENT
13,2013-07-25 00:00:00.0,9149,PENDING_PAYMENT
16,2013-07-25 00:00:00.0,7276,PENDING_PAYMENT
19,2013-07-25 00:00:00.0,9488,PENDING_PAYMENT
23,2013-07-25 00:00:00.0,4367,PENDING_PAYMENT
27,2013-07-25 00:00:00.0,3241,PENDING_PAYMENT
30,2013-07-25 00:00:00.0,10039,PENDING_PAYMENT
33,2013-07-25 00:00:00.0,5793,PENDING_PAYMENT
```

7.Run a command to check number of records in orders_filtered.csv file under data/landing folder.

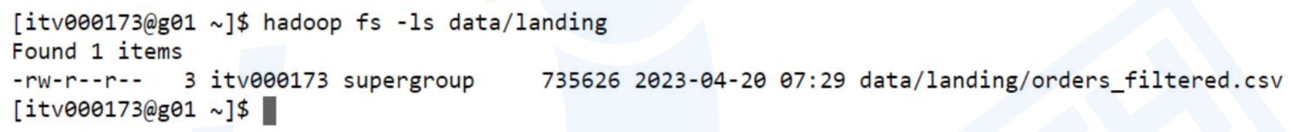
hadoop fs -cat data/landing/orders_filtered.csv | wc -l



```
[itv000173@g01 ~]$ hadoop fs -cat data/landing/orders_filtered.csv | wc -l
15030
[itv000173@g01 ~]$ █
```

8.Write a command to list the files in the data/landing folder of hdfs.

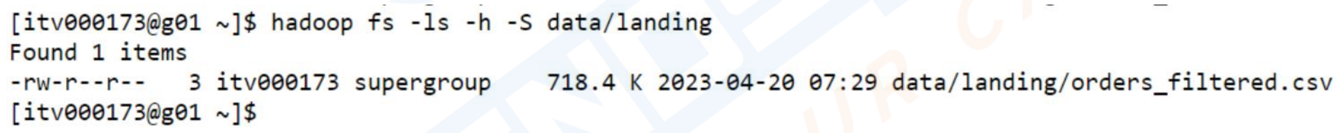
hadoop fs -ls data/landing



```
[itv000173@g01 ~]$ hadoop fs -ls data/landing
Found 1 items
-rw-r--r--   3 itv000173 supergroup     735626 2023-04-20 07:29 data/landing/orders_filtered.csv
[itv000173@g01 ~]$ █
```

9.Reframe this command so that you can see the file size in kb's

hadoop fs -ls -h -S data/landing

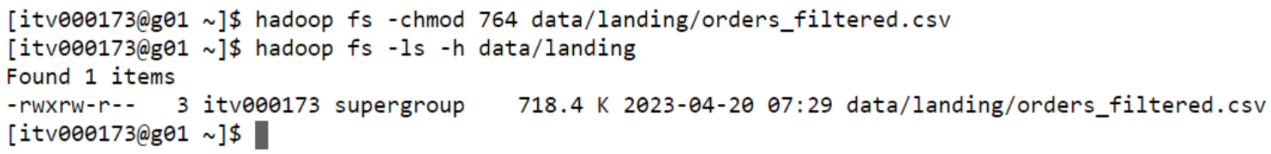


```
[itv000173@g01 ~]$ hadoop fs -ls -h -S data/landing
Found 1 items
-rw-r--r--   3 itv000173 supergroup    718.4 K 2023-04-20 07:29 data/landing/orders_filtered.csv
[itv000173@g01 ~]$
```

10.Change the permission of this file:

- give read, write and execute to the owner
- read and write to the group
- read to others

hadoop fs -chmod 764 data/landing/orders_filtered.csv

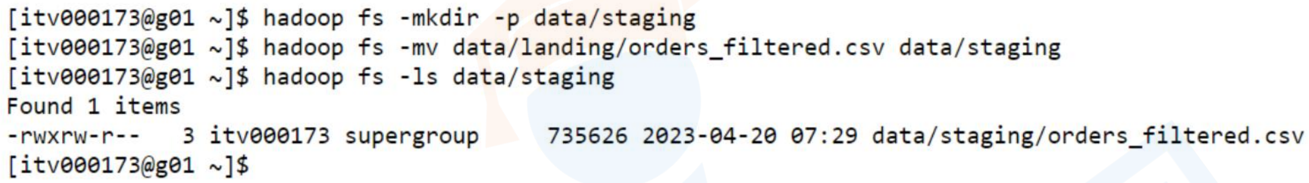


```
[itv000173@g01 ~]$ hadoop fs -chmod 764 data/landing/orders_filtered.csv
[itv000173@g01 ~]$ hadoop fs -ls -h data/landing
Found 1 items
-rwxrw-r--   3 itv000173 supergroup    718.4 K 2023-04-20 07:29 data/landing/orders_filtered.csv
[itv000173@g01 ~]$ 
```

11.Create a new folder data/staging in your hdfs and move orders_filtered.csv from data/landing to data/staging

hadoop fs -mkdir -p data/staging

hadoop fs -mv data/landing/orders_filtered.csv data/staging



```
[itv000173@g01 ~]$ hadoop fs -mkdir -p data/staging
[itv000173@g01 ~]$ hadoop fs -mv data/landing/orders_filtered.csv data/staging
[itv000173@g01 ~]$ hadoop fs -ls data/staging
Found 1 items
-rwxrw-r--   3 itv000173 supergroup     735626 2023-04-20 07:29 data/staging/orders_filtered.csv
[itv000173@g01 ~]$
```

12.Now let's assume a spark program would have run on your staging folder to do some processing and let's say the processed results gives you just 2 lines as ouput

3617,2013-08-1500:00:00.0,8889,PENDING_PAYMENT

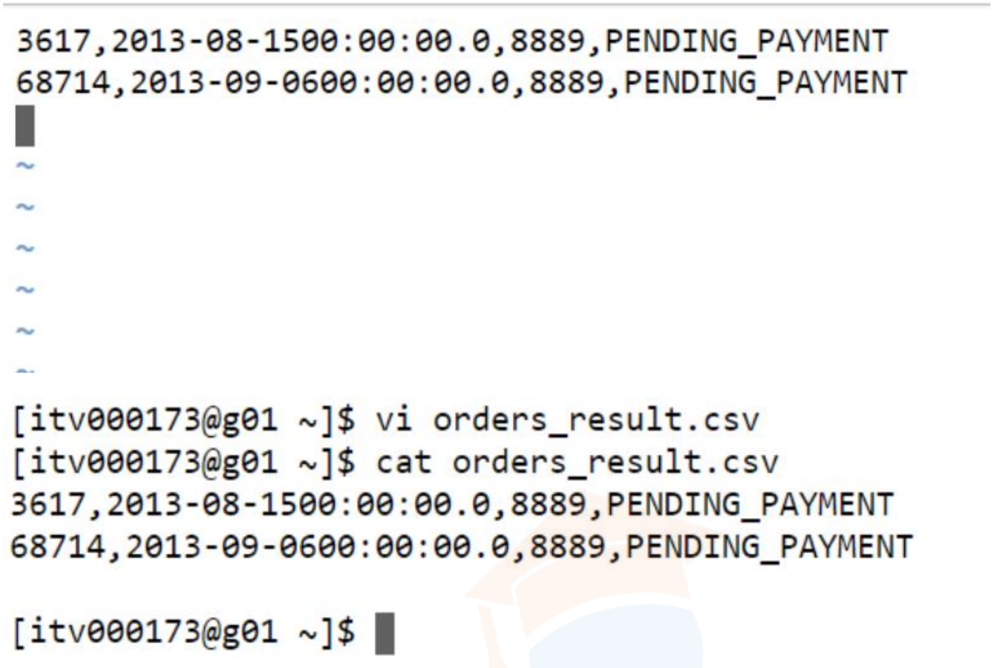
68714,2013-09-0600:00:00.0,8889,PENDING_PAYMENT

To simulate this, create a new file called orders_result.csv in the home directory of your local gateway node using vi editor and have the above 2 records..

cd ~

vi orders_result.csv

[you can insert the records by clicking “i”, then for saving file give – esc key “:wq”

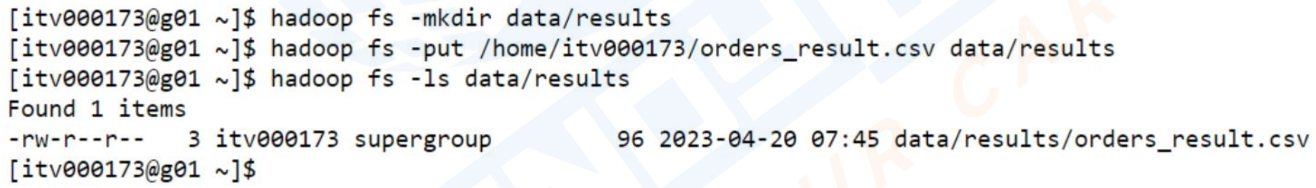


```
3617,2013-08-1500:00:00.0,8889,PENDING_PAYMENT
68714,2013-09-0600:00:00.0,8889,PENDING_PAYMENT

~
~
~
~
~
~
[itv000173@g01 ~]$ vi orders_result.csv
[itv000173@g01 ~]$ cat orders_result.csv
3617,2013-08-1500:00:00.0,8889,PENDING_PAYMENT
68714,2013-09-0600:00:00.0,8889,PENDING_PAYMENT

[itv000173@g01 ~]$
```

13.move orders_result.csv from local to hdfs under a new directory called data/results (think as if spark program has run and has created this file)

hadoop fs -mkdir data/results

hadoop fs -put /home/itv000173/orders_result.csv data/results

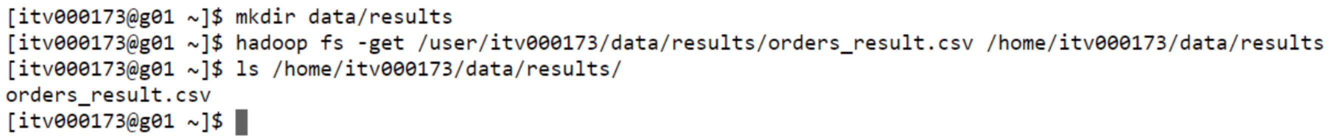


```
[itv000173@g01 ~]$ hadoop fs -mkdir data/results
[itv000173@g01 ~]$ hadoop fs -put /home/itv000173/orders_result.csv data/results
[itv000173@g01 ~]$ hadoop fs -ls data/results
Found 1 items
-rw-r--r--   3 itv000173 supergroup         96 2023-04-20 07:45 data/results/orders_result.csv
[itv000173@g01 ~]$
```

14.Now the processed results we want to bring back to local under a folder data/results in your local. So run a command to bring the file from hdfs to local.

mkdir data/results

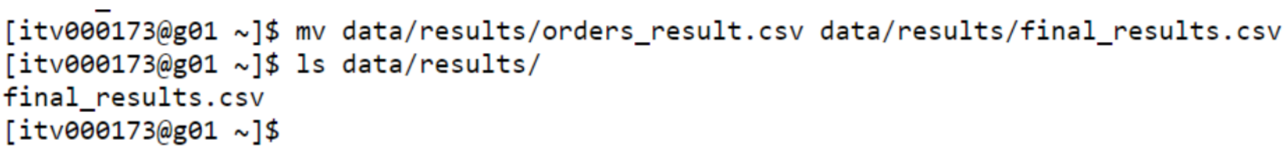
hadoop fs -get /user/itv000173/data/results/orders_result.csv /home/itv000173/data/results



```
[itv000173@g01 ~]$ mkdir data/results
[itv000173@g01 ~]$ hadoop fs -get /user/itv000173/data/results/orders_result.csv /home/itv000173/data/results
[itv000173@g01 ~]$ ls /home/itv000173/data/results/
orders_result.csv
[itv000173@g01 ~]$
```

15.Rename the file orders_result.csv under data/results folder in your local to final_results.csv.

mv data/results/orders_result.csv data/results/final_results.csv



```
[itv000173@g01 ~]$ mv data/results/orders_result.csv data/results/final_results.csv
[itv000173@g01 ~]$ ls data/results/
final_results.csv
[itv000173@g01 ~]$
```

16.Now we are done.. So delete all the directories that you have created in your local as well as hdfs.

rm -R landing

rm -R staging

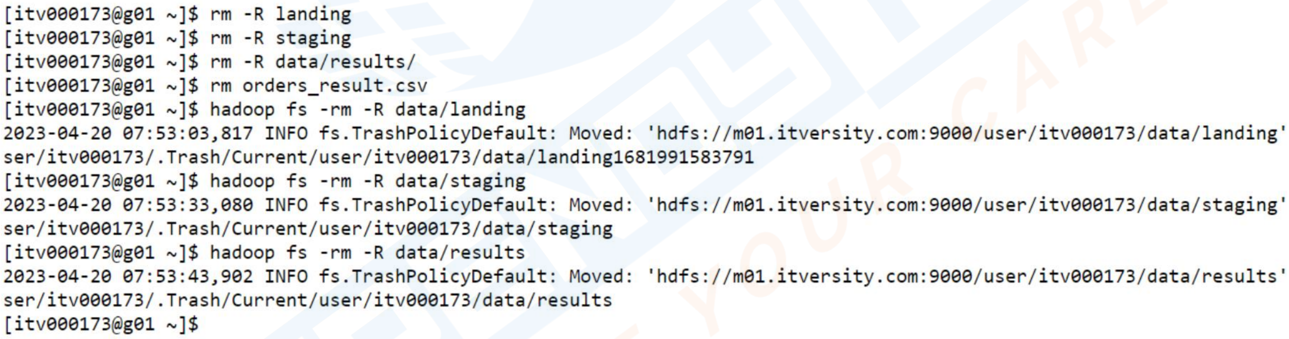
rm -R data/results

rm orders_result.csv

hadoop fs -rm -R data/landing

hadoop fs -rm -R data/staging

hadoop fs -rm -R data/results



```
[itv000173@g01 ~]$ rm -R landing
[itv000173@g01 ~]$ rm -R staging
[itv000173@g01 ~]$ rm -R data/results/
[itv000173@g01 ~]$ rm orders_result.csv
[itv000173@g01 ~]$ hadoop fs -rm -R data/landing
2023-04-20 07:53:03,817 INFO fs.TrashPolicyDefault: Moved: 'hdfs://m01.itversity.com:9000/user/itv000173/data/landing'
ser/itv000173/.Trash/Current/user/itv000173/data/landing1681991583791
[itv000173@g01 ~]$ hadoop fs -rm -R data/staging
2023-04-20 07:53:33,080 INFO fs.TrashPolicyDefault: Moved: 'hdfs://m01.itversity.com:9000/user/itv000173/data/staging'
ser/itv000173/.Trash/Current/user/itv000173/data/staging
[itv000173@g01 ~]$ hadoop fs -rm -R data/results
2023-04-20 07:53:43,902 INFO fs.TrashPolicyDefault: Moved: 'hdfs://m01.itversity.com:9000/user/itv000173/data/results'
ser/itv000173/.Trash/Current/user/itv000173/data/results
[itv000173@g01 ~]$
```

==