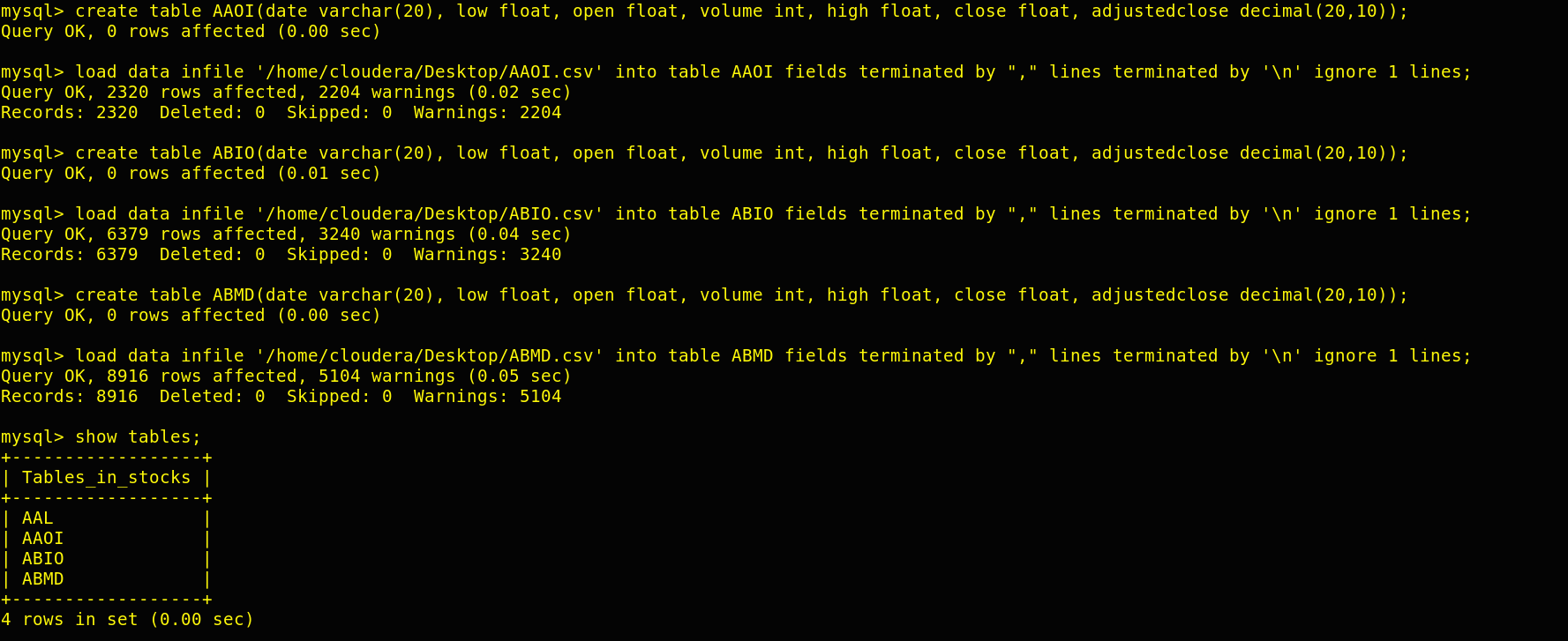
**PIPELINE FOR TRANSFERRING DATA FROM MYSQL TO HIVE AND THEN AFTER ANALYSIS EXPORTING DATA FROM EXTERNAL TABLE TO MYSQL**

**Step 1:** Transferred the csv file from C:\Users\Admin\Downloads to /home/cloudera/Desktop/

**Step 2:** Created a directory on Hadoop by using hadoop fs -mkdir wallmartproject

**Step 3:** Created a database called stocks then created all 4 tables and removed their headers.

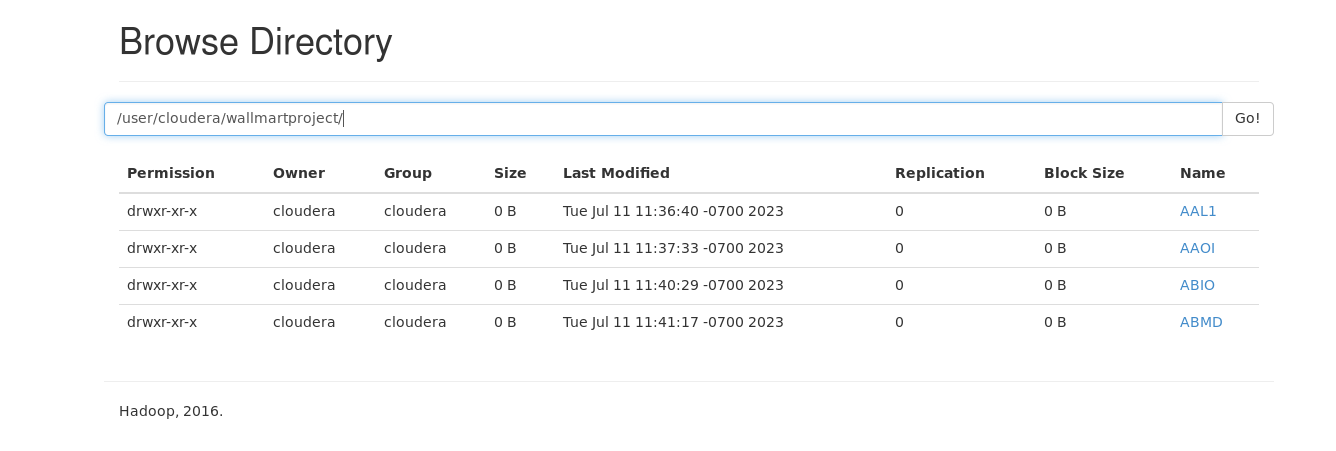
load data infile '/home/cloudera/Desktop/ABMD.csv' into table ABMD fields terminated by "," lines terminated by '\n' ignore 1 lines;



**Step 4:** I have transferred all these tables to Hadoop from mysql.

Command:

sqoop import-all-tables --connect jdbc:mysql://localhost:3306/stocks --username root -password cloudera --warehouse-dir /user/cloudera/wallmartproject/ -m 1



**Step 5:** Loading hdfs data into hive tables

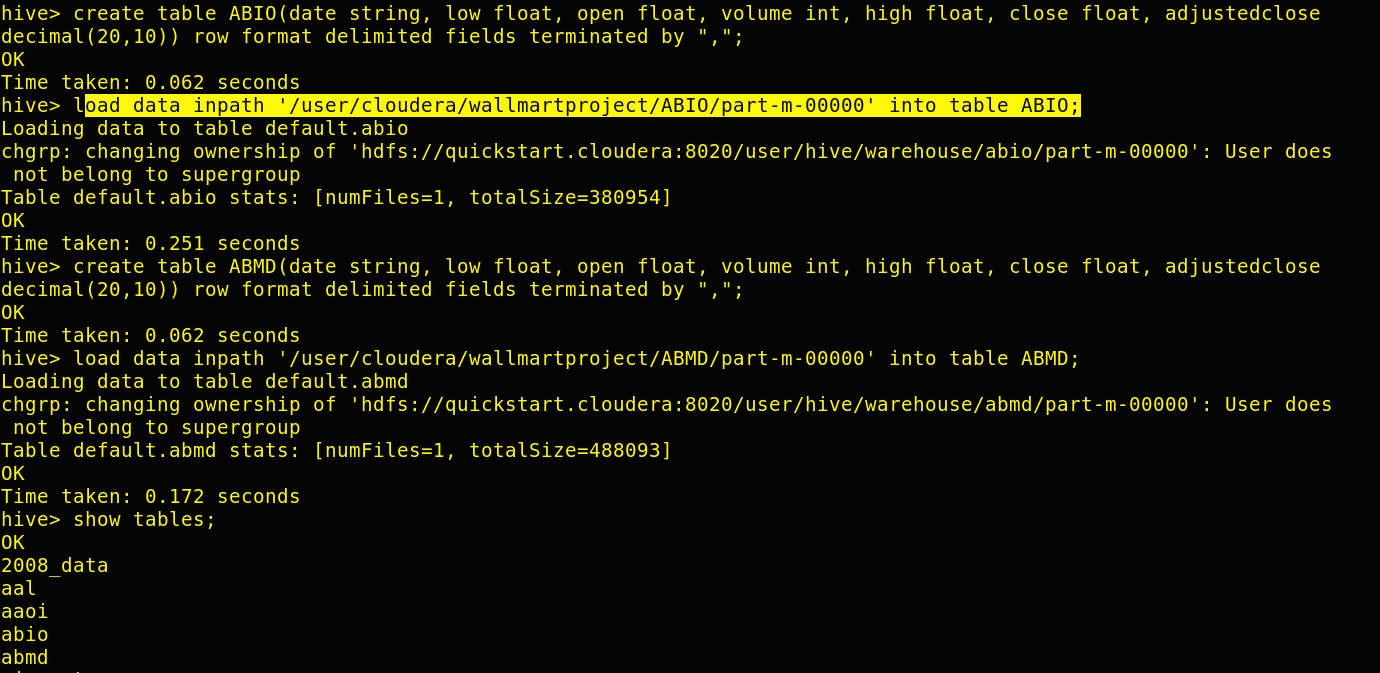
Here we have to create all the tables and load the data by using command:

Create table command:

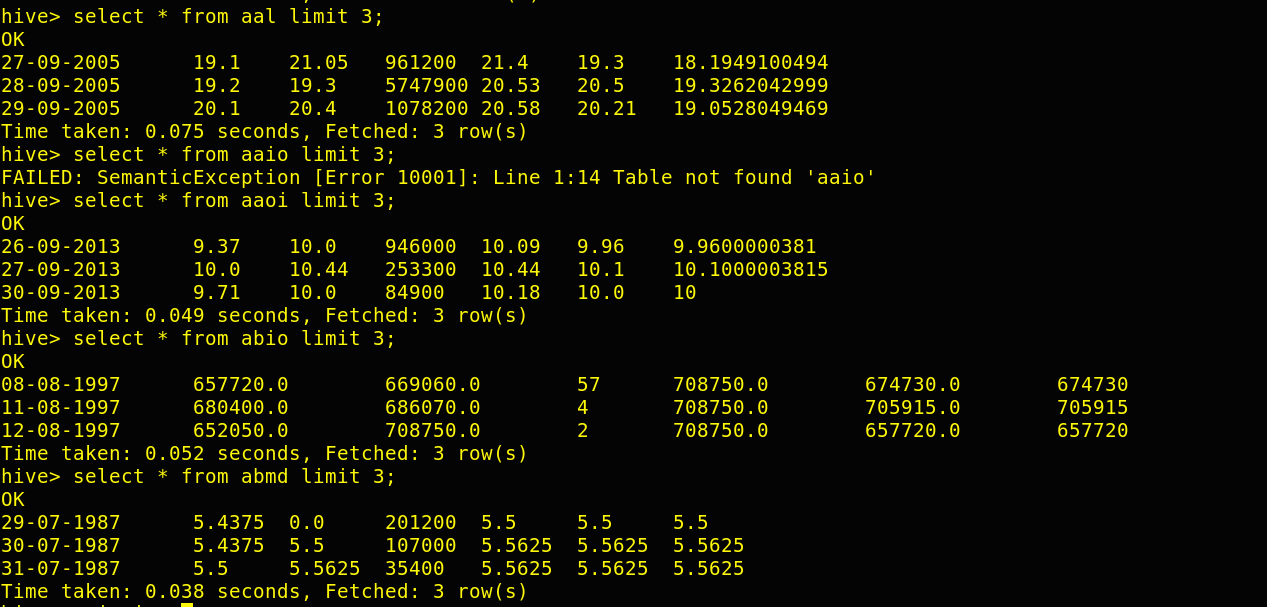
create table ABIO(date string, low float, open float, volume int, high float, close float, adjustedclose decimal(20,10)) row format delimited fields terminated by ",";

Loading data from hdfs to hive table command:

load data inpath '/user/cloudera/wallmartproject/ABIO/part-m-00000' into table ABIO;



**Displaying 3 records from each hive table:**



Step 6: After running the HIVE query I have made an external table in a new directory /user/cloudera/projectresult for each result and stored the result in that table. External table creation query and insert query is given below in each question

Step 7: After making external table and inserting data into that table, I have made same number of tables in mysql in a new database with name project\_result and exported the data to there by using sqoop query:

Eg: sqoop export --connect jdbc:mysql://localhost:3306/project\_result --username root --password cloudera --table scenario3\_ext\_abmd --export-dir /user/cloudera/projectresult/scenario3\_ext\_abmd/000000-0 --input-fields-terminated-by ','

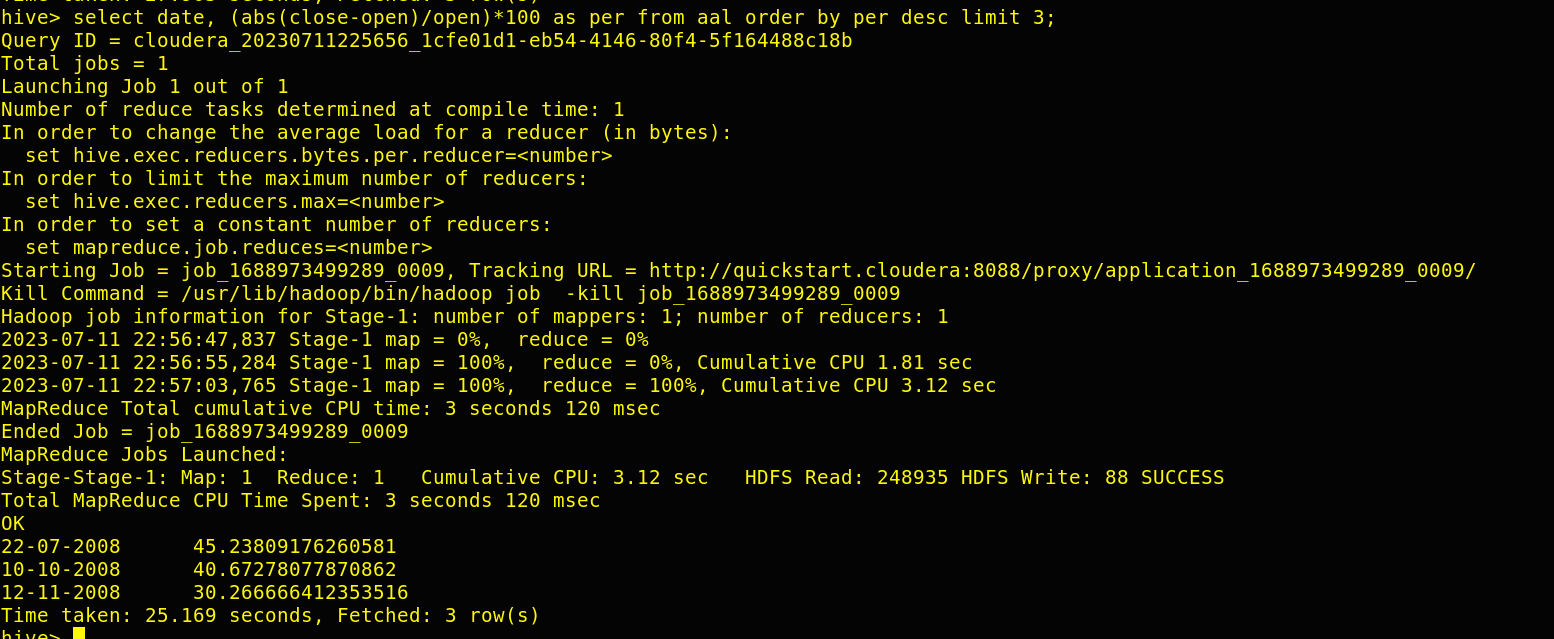
**ANALYSIS AND STORAGE HIVE QUERIES -> 7 SCENARIOS**

**Scenario 1 :**  Write a Hive query to identify the top three dates that experienced the largest percentage change in stock price (from open to close) for every stock.

Query:

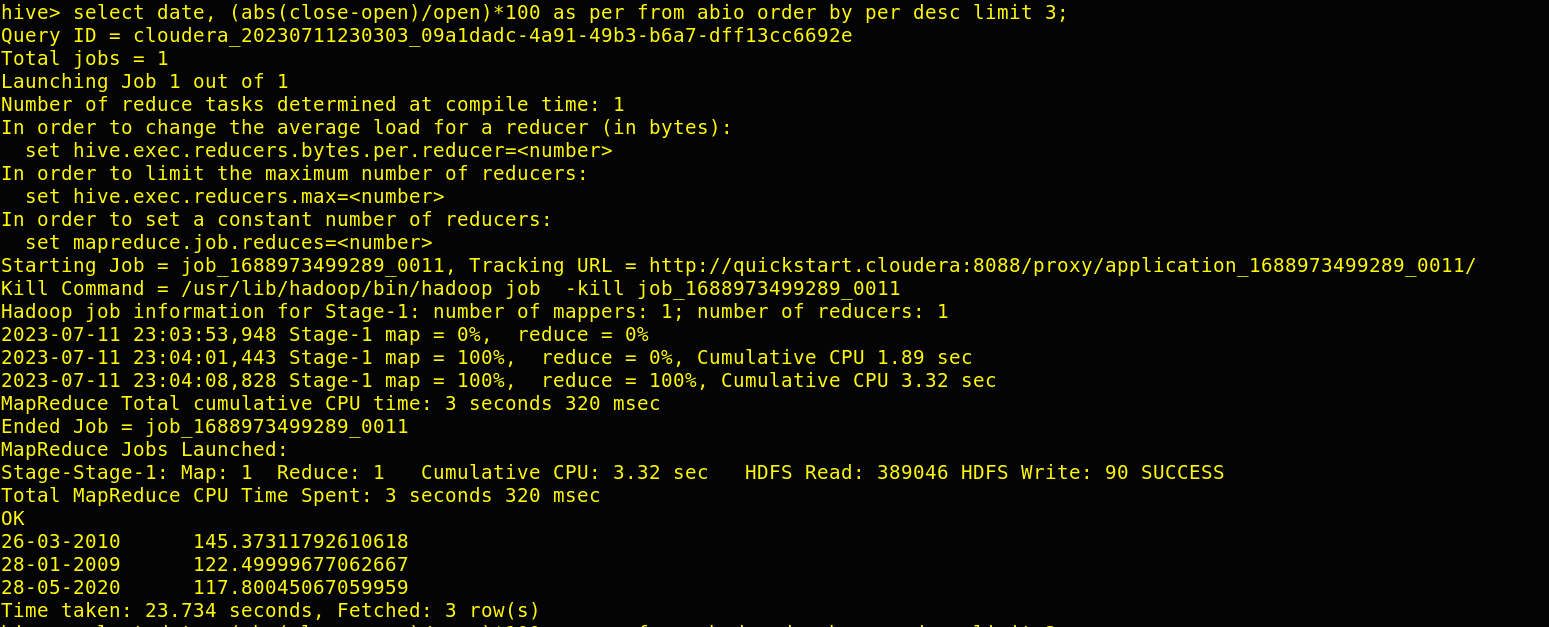
**For AAL:**

select date, (abs(close-open)/open)\*100 as per from aal order by per desc limit 3;



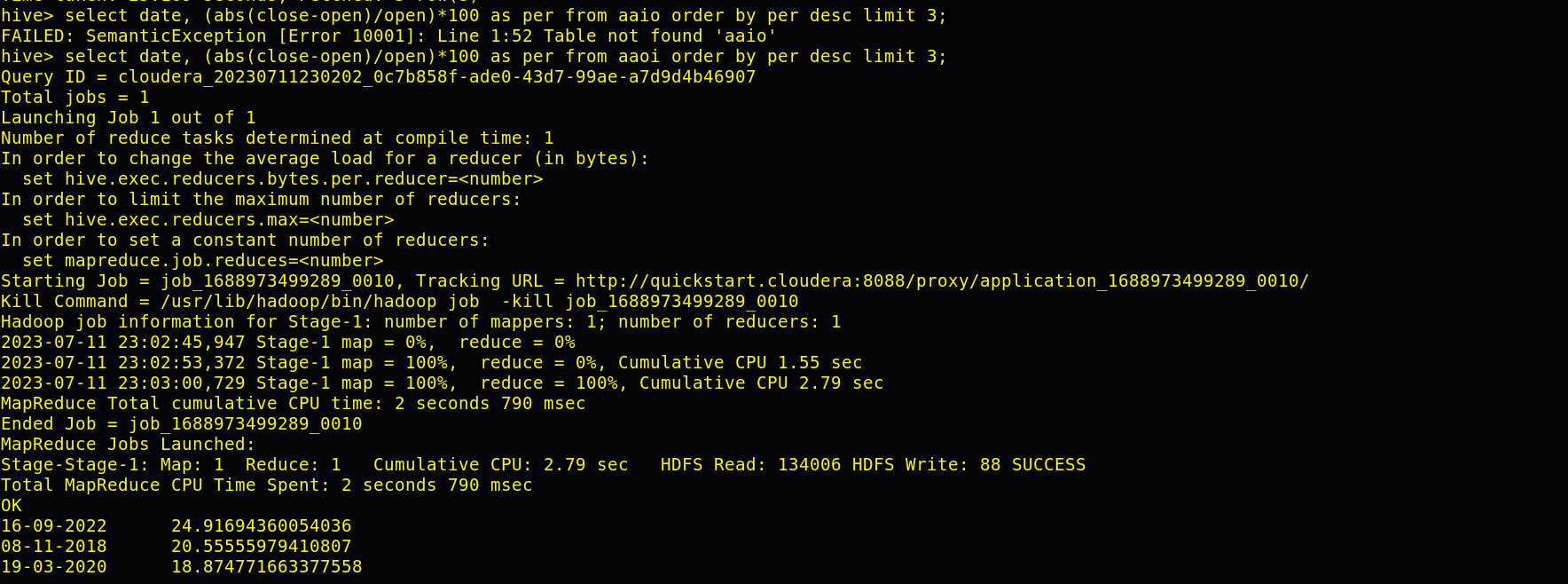
**For ABIO:**

select date, (abs(close-open)/open)\*100 as per from abio order by per desc limit 3;



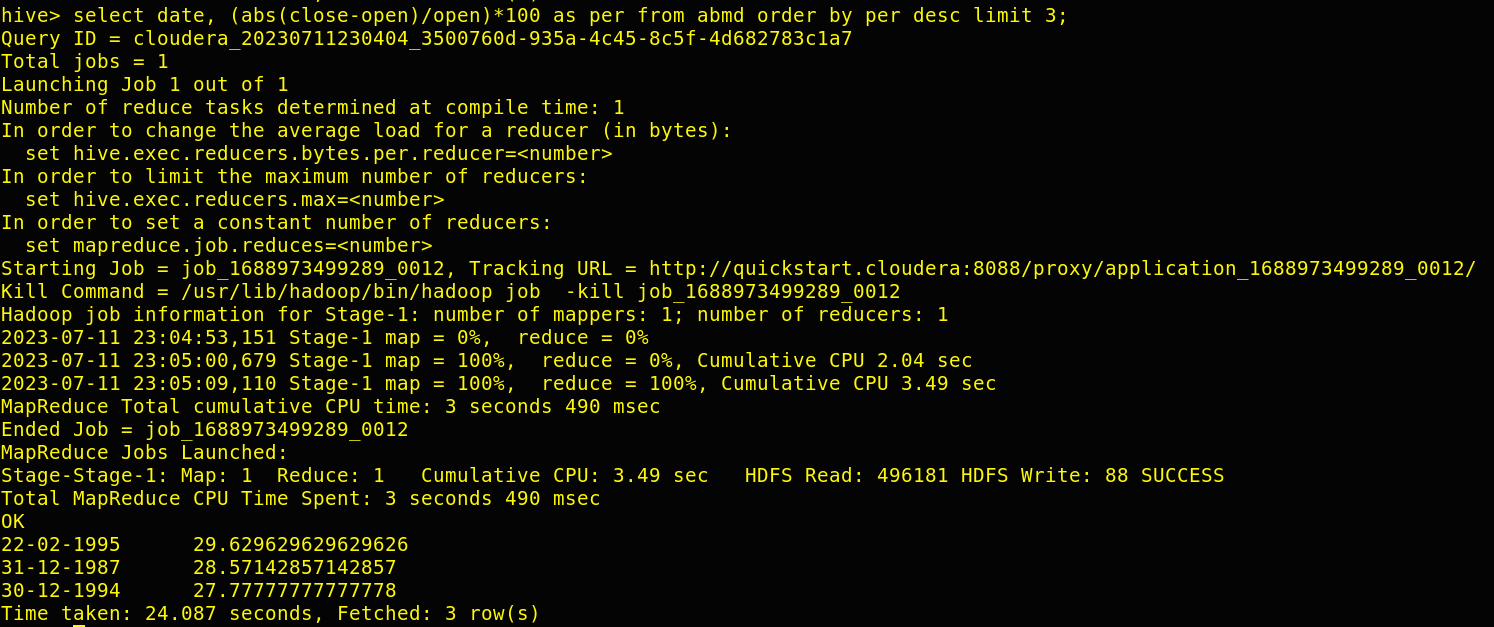
**For AAOI:**

select date, (abs(close-open)/open)\*100 as per from aaoi order by per desc limit 3;



**For ABMD:**

select date, (abs(close-open)/open)\*100 as per from abmd order by per desc limit 3;



For creating external table, I have used this query:

CREATE EXTERNAL TABLE per\_change\_AAL/ per\_change\_ABIO/ per\_change\_AAOI/per\_change\_ABMD(

Date string,

Percentage double

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/cloudera/projectresult/(AAL\_S1/ABIO\_S1/ AAOI\_S/ABMD\_S1)';

And I have used this query to insert result into my external table:

INSERT overwrite table per\_change\_AAL/ per\_change\_ABIO/ per\_change\_AAOI/per\_change\_ABMD

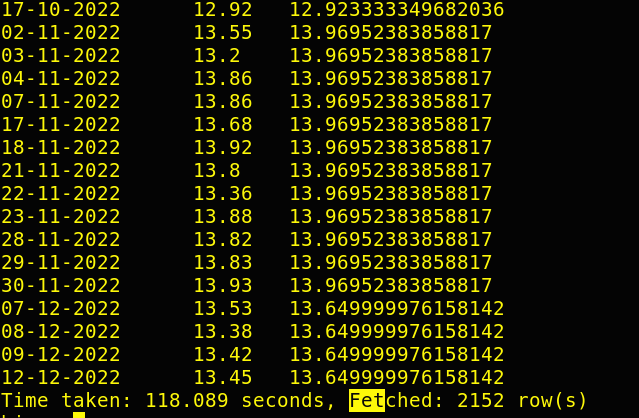
select date as Date, (abs(close-open)/open)\*100 as Percentage from aal/abio/aaoi/abmd order by Percentage desc limit 3;

**Scenario 2:**  Write a Hive query to identify the dates where Low is less than average month low for every stock.

Hive Query:

**For AAL :**

with cte as (SELECT date, low, AVG(low) OVER (PARTITION BY SUBSTRING(date, 4)) AS monthly\_avg\_low from AAL) select date, low, monthly\_avg\_low from cte where low<monthly\_avg\_low order by unix\_timestamp(date, 'dd-MM-yyyy');

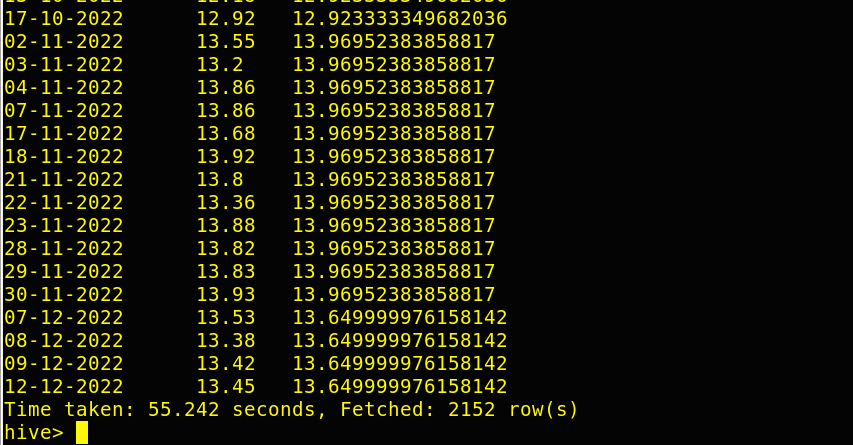


**With Indexing:** Creating index on date column and running the same query then time taken is reduced.

create index date\_index on table aal(date) as

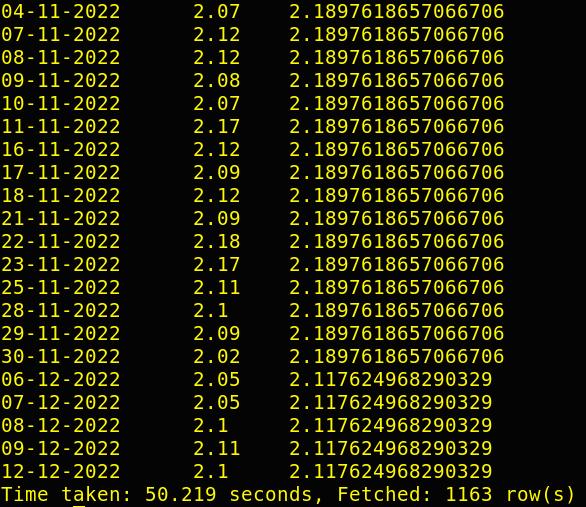
'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'

WITH DEFERRED REBUILD;



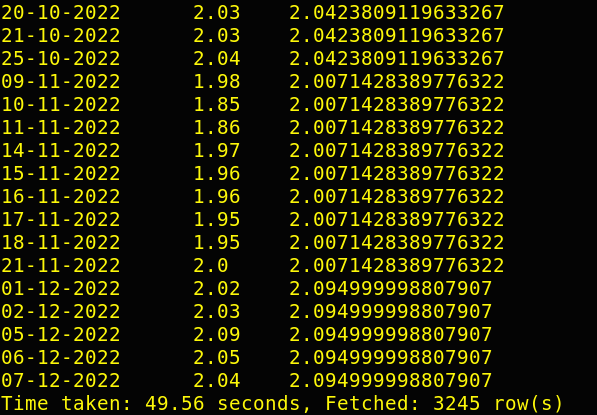
**For AAOI :**

with cte as (SELECT date, low, AVG(low) OVER (PARTITION BY SUBSTRING(date, 4)) AS monthly\_avg\_low from AAOI) select date,low,monthly\_avg\_low from cte where low<monthly\_avg\_low order by unix\_timestamp(date, 'dd-MM-yyyy');



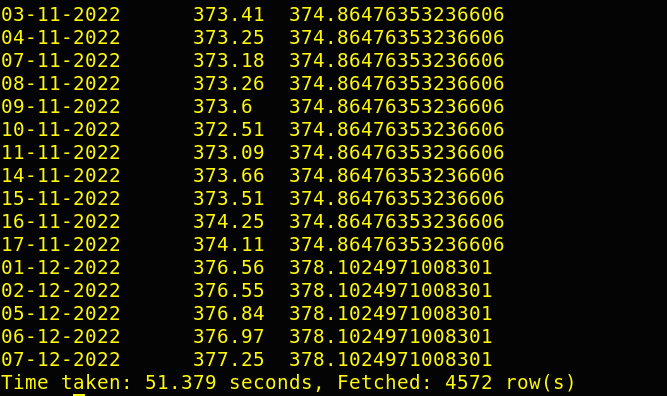
**For ABIO :**

with cte as (SELECT date, low, AVG(low) OVER (PARTITION BY SUBSTRING(date, 4)) AS monthly\_avg\_low from ABIO) select date,low,monthly\_avg\_low from cte where low<monthly\_avg\_low order by unix\_timestamp(date, 'dd-MM-yyyy');



**For ABMD :**

with cte as (SELECT date, low, AVG(low) OVER (PARTITION BY SUBSTRING(date, 4)) AS monthly\_avg\_low from ABIO) select date,low,monthly\_avg\_low from cte where low<monthly\_avg\_low order by unix\_timestamp(date, 'dd-MM-yyyy');



For creating external table, I have used this query:

CREATE EXTERNAL TABLE scenario2\_aal/ scenario2\_aaoi/ scenario2\_abio\_main/

scenario2\_abmd (date string, low float, avg\_low float)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/cloudera/projectresult/(scenario2\_aal/ scenario2\_aaoi/ scenario2\_abio\_main/

scenario2\_abmd) ';

And I have used this query to insert result into my external table:

with cte as (SELECT date, low, AVG(low) OVER (PARTITION BY SUBSTRING(date, 4)) AS monthly\_avg\_low from AAL / AAOI / ABIO/ ABMD) insert overwrite table scenario2\_aal/ scenario2\_aaoi/ scenario2\_abio\_main/

scenario2\_abmd select date,low,monthly\_avg\_low from cte where low<monthly\_avg\_low order by unix\_timestamp(date, 'dd-MM-yyyy');

**Scenario 3:** Write a Hive query to find the date with the longest consecutive streak of increasing closing prices for every stock.

Hive Query:

with cte as (SELECT Date,Close,lag(Close) over(order by unix\_timestamp(Date, 'dd-MM-yyyy') ) as prev FROM aal/aaoi/abio/abmd),

cte1 as (select Date,Close,if(Close < prev, 0, 1) as is\_inc from cte),

cte2 as (select Date,Close, is\_inc, lead(is\_inc) over(order by unix\_timestamp(Date, 'dd-MM-yyyy')) as next\_val from cte1),

cte3 as (select Date,Close, is\_inc, if( is\_inc = 0 and is\_inc != next\_val,1,0) as is\_chg from cte2),

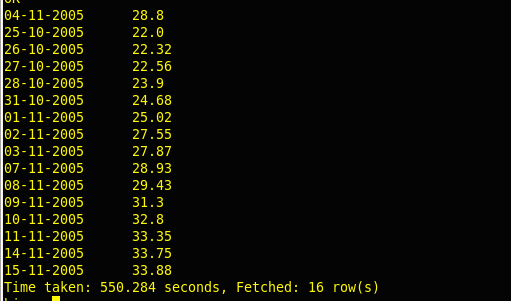
cte4 as (select Date,Close, is\_inc, is\_chg,sum(is\_chg) over( order by unix\_timestamp(Date, 'dd-MM-yyyy')) as bucket from cte3 where not (is\_inc = 0

and is\_chg = 0)),

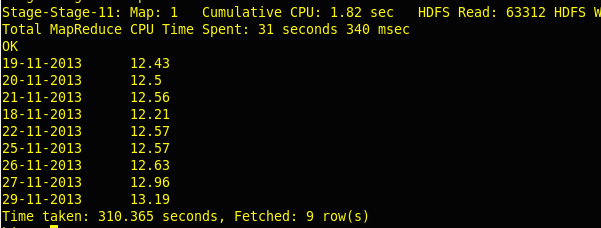
cte5 as (select Date, Close,is\_inc,is\_chg,bucket, count(\*) over(partition by bucket) as cnt from cte4)

select Date,close from cte5 cross join (select max(cnt) as max\_cnt from cte5) cte6 where cte5.cnt = cte6.max\_cnt;

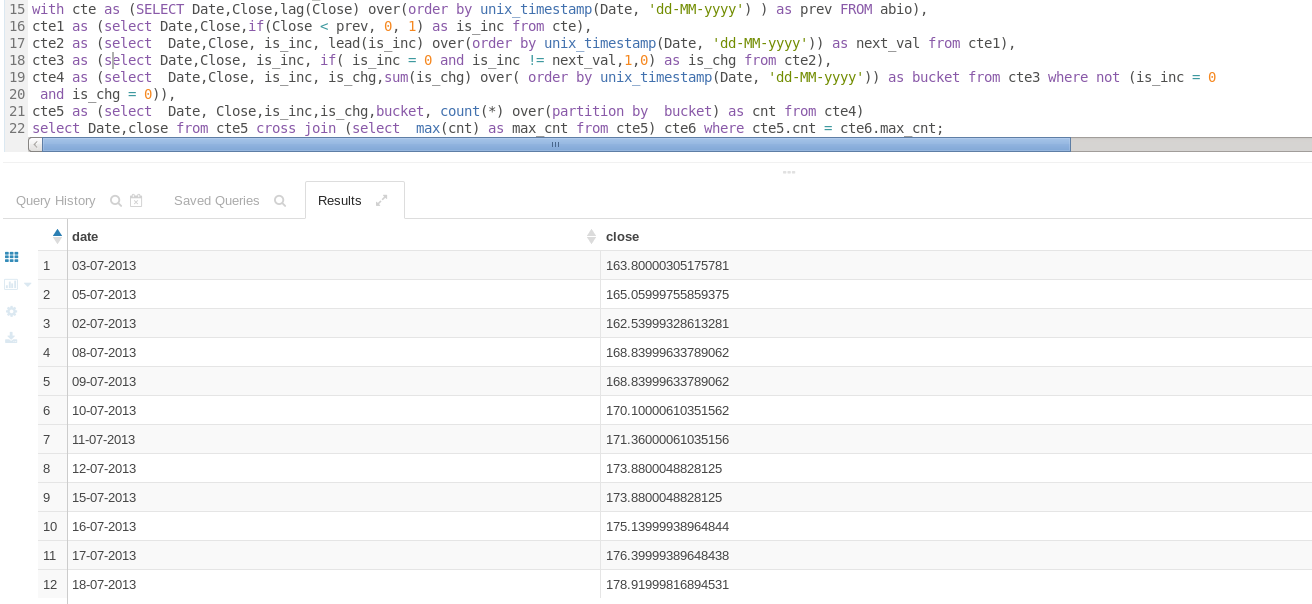
For AAL:



For AAOI:



For ABIO:



For ABMD:



For creating external table, I have used this query:

CREATE EXTERNAL TABLE scenario3\_ext\_aal/ scenario3\_ext\_ aaoi/ scenario3\_ext\_abio/ scenario3\_ext\_abmd (

date string, close float)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE

LOCATION '/user/cloudera/projectresult/scenario3\_ext\_aal/scenario3\_ext\_ aaoi/scenario3\_ext\_abio/scenario3\_ext\_abmd’;

And I have used this query to insert result into my external table:

with cte as (SELECT Date,Close,lag(Close) over(order by unix\_timestamp(Date, 'dd-MM-yyyy') ) as prev FROM aal/aaoi/abio/abmd),

cte1 as (select Date,Close,if(Close < prev, 0, 1) as is\_inc from cte),

cte2 as (select Date,Close, is\_inc, lead(is\_inc) over(order by unix\_timestamp(Date, 'dd-MM-yyyy')) as next\_val from cte1),

cte3 as (select Date,Close, is\_inc, if( is\_inc = 0 and is\_inc != next\_val,1,0) as is\_chg from cte2),

cte4 as (select Date,Close, is\_inc, is\_chg,sum(is\_chg) over( order by unix\_timestamp(Date, 'dd-MM-yyyy')) as bucket from cte3 where not (is\_inc = 0

and is\_chg = 0)),

cte5 as (select Date, Close,is\_inc,is\_chg,bucket, count(\*) over(partition by bucket) as cnt from cte4)

insert overwrite table select Date,close from cte5 cross join (select max(cnt) as max\_cnt from cte5) cte6 where cte5.cnt = cte6.max\_cnt;

**Scenario 4:** Write a Hive query to find the dates where AAL open price is higher than AAOI open price OR AAL volume greater than AMBD (write your query in an optimised way).

For this I have done bucketing on 4 tables:

For AAL:

create table stock\_bucket\_aal(date string, low float, open float, volume int, high float, close float, adjusted close float) clustered by (date) into 3 buckets row format delimited fields terminated by ',';

For AAOI:

create table stock\_bucket\_aaoi(date string, low float, open float, volume int, high float, close float, adjusted close float) clustered by (date) into 3 buckets row format delimited fields terminated by ',';

For ABIO:

create table stock\_bucket\_abio(date string, low float, open float, volume int, high float, close float, adjusted close float) clustered by (date) into 3 buckets row format delimited fields terminated by ',';

For ABMD:

create table stock\_bucket\_abmd(date string, low float, open float, volume int, high float, close float, adjusted close float) clustered by (date) into 3 buckets row format delimited fields terminated by ',';

Then perform this query: set hive.enforce.bucketing = true;

Then I have insert data from hive tables that are aal, aaoi, abmd, abio to this bucket table which I have created here:

Insert overwrite table stock\_bucket\_aal/abio/abmd/aaoi select \* from aal/abio/abmd/aaoi;

Hive Query:

select stock\_bucket\_aal.date, stock\_bucket\_aal.open as aal\_open\_price, stock\_bucket\_aaoi.open aaoi\_open\_price, stock\_bucket\_aal.volume aal\_volume\_price, stock\_bucket\_abmd.volume abmd\_volume\_price

from

stock\_bucket\_aal

join

stock\_bucket\_aaoi

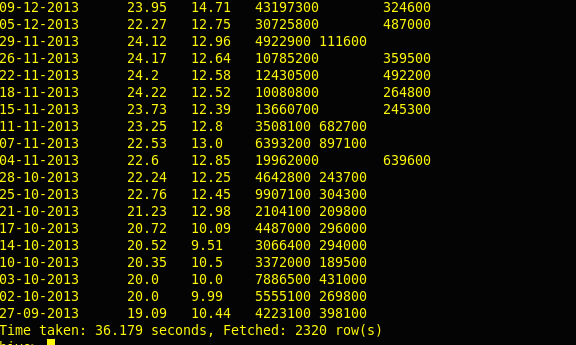
on stock\_bucket\_aal.date = stock\_bucket\_aaoi.date

join

stock\_bucket\_abmd

on stock\_bucket\_aal.date = stock\_bucket\_abmd.date

where stock\_bucket\_aal.open > stock\_bucket\_aaoi.open or stock\_bucket\_aal.volume > stock\_bucket\_abmd.volume



For creating external table, I have used this query:

CREATE EXTERNAL TABLE scenario4\_external(

date string, aal\_open\_price float, aaoi\_open\_price float, aal\_volume\_price int , abmd\_volume\_price int

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/cloudera/projectresult/scenario4\_external';

And I have used this query to insert result into my external table:

insert overwrite table scenario4\_external select stock\_bucket\_aal.date, stock\_bucket\_aal.open as aal\_open\_price, stock\_bucket\_aaoi.open aaoi\_open\_price, stock\_bucket\_aal.volume aal\_volume\_price, stock\_bucket\_abmd.volume abmd\_volume\_price

from

stock\_bucket\_aal

join

stock\_bucket\_aaoi

on stock\_bucket\_aal.date = stock\_bucket\_aaoi.date

join

stock\_bucket\_abmd

on stock\_bucket\_aal.date = stock\_bucket\_abmd.date

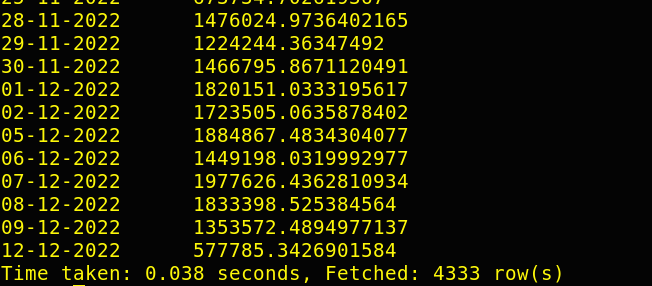
where stock\_bucket\_aal.open > stock\_bucket\_aaoi.open or stock\_bucket\_aal.volume > stock\_bucket\_abmd.volume;

**Scenario 5:** Write a Hive query to calculate VH ratio(volume to high ratio).

**Hive query:**

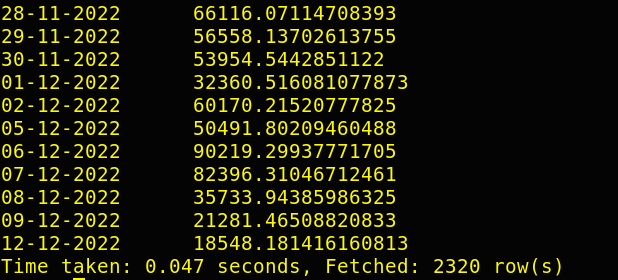
**For AAL :**

select DATE,(volume/high) as volume\_to\_high\_ratio from aal ;



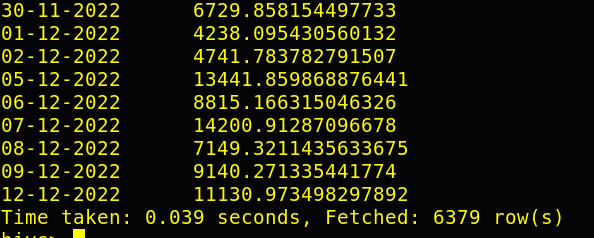
**For AAOI :**

select DATE,(volume/high) as volume\_to\_high\_ratio from aaoi ;



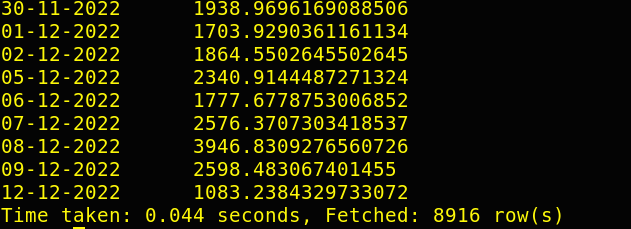
**For ABIO :**

select DATE,(volume/high) as volume\_to\_high\_ratio from abio;



**For ABMD :**

select DATE,(volume/high) as volume\_to\_high\_ratio from abmd ;



For creating external table, I have used this query:

CREATE EXTERNAL TABLE scenario5\_ext\_aal/ scenario5\_ext\_ aaoi/ scenario5\_ext\_abio/ scenario5\_ext\_abmd (

date string, volume\_to\_high\_ratio float)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE

LOCATION '/user/cloudera/projectresult/ scenario5\_ext\_aal/ scenario5\_ext\_ aaoi/ scenario5\_ext\_abio/ scenario5\_ext\_abmd’;

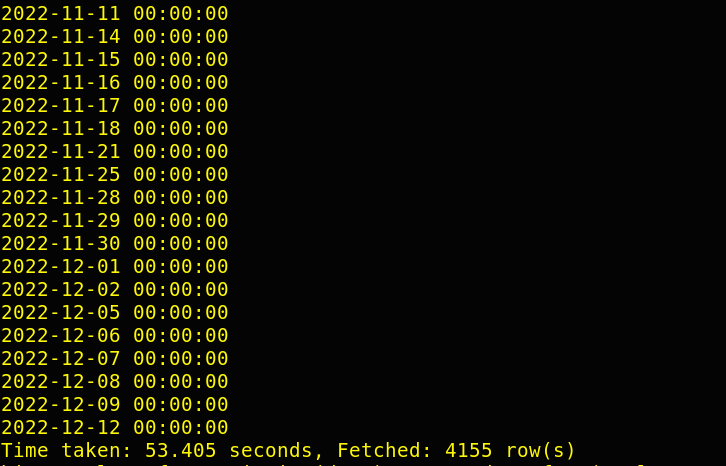
And I have used this query to insert result into my external table:

Insert overwrite table scenario5\_ext\_aal/scenario5\_ext\_ aaoi/scenario5\_ext\_abio/scenario5\_ext\_abmd select DATE,(volume/high) as volume\_to\_high\_ratio from aal/aaoi/abio/abmd;

**Scenario 6:** Write a Hive query to find the dates where previous day close and current day open difference is greater than 0 for each stock.

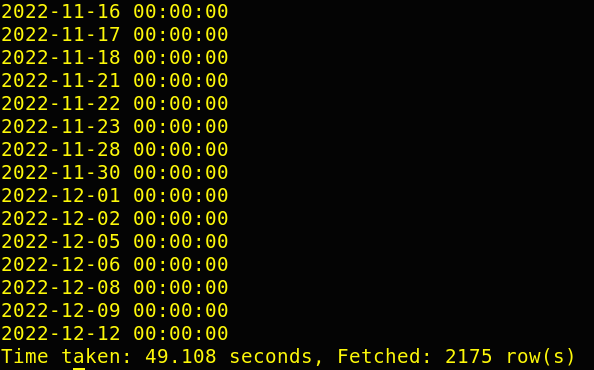
**For AAL :**

select from\_unixtime(date) as req\_date from( select unix\_timestamp(date, 'dd-MM-yyyy') as date,lag(Close) over (order by unix\_timestamp(date, 'dd-MM-yyyy')) as previous\_close, Open as current\_open from aal) subquery where abs(current\_open - previous\_close )> 0 order by req\_date;

****

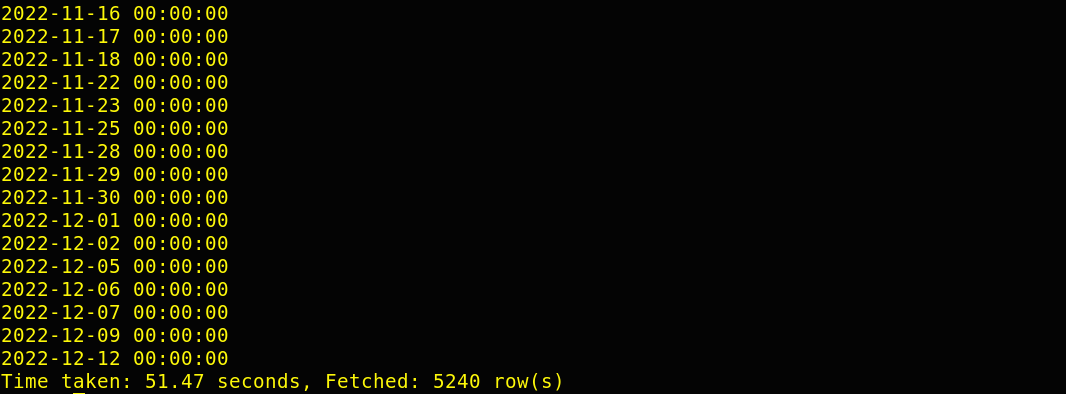
**For AAOI :**

select from\_unixtime(date) as req\_date from( select unix\_timestamp(date, 'dd-MM-yyyy') as date,lag(Close) over (order by unix\_timestamp(date, 'dd-MM-yyyy')) as previous\_close, Open as current\_open from aaoi) subquery where abs(current\_open - previous\_close )> 0 order by req\_date;

****

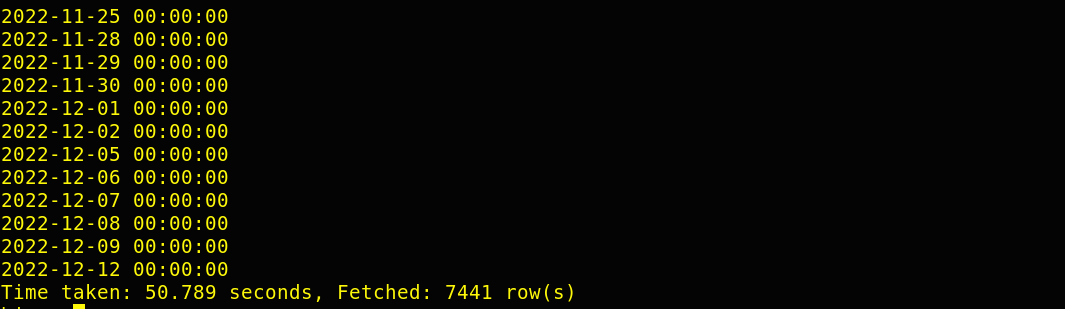
**For ABIO :**

select from\_unixtime(date) as req\_date from( select unix\_timestamp(date, 'dd-MM-yyyy') as date,lag(Close) over (order by unix\_timestamp(date, 'dd-MM-yyyy')) as previous\_close, Open as current\_open from abio) subquery where abs(current\_open - previous\_close )> 0 order by req\_date;

****

**For ABMD :**

select from\_unixtime(date) as req\_date from( select unix\_timestamp(date, 'dd-MM-yyyy') as date,lag(Close) over (order by unix\_timestamp(date, 'dd-MM-yyyy')) as previous\_close, Open as current\_open from abmd) subquery where abs(current\_open - previous\_close )> 0 order by req\_date;

****

For creating external table, I have used this query:

CREATE EXTERNAL TABLE scenario6\_ext\_aal/ scenario6\_ext\_ aaoi/ scenario6\_ext\_abio/ scenario6\_ext\_abmd (

from\_unixtime string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE

LOCATION '/user/cloudera/projectresult/ scenario6\_ext\_aal/ scenario6\_ext\_ aaoi/ scenario6\_ext\_abio/ scenario6\_ext\_abmd’;

And I have used this query to insert result into my external table:

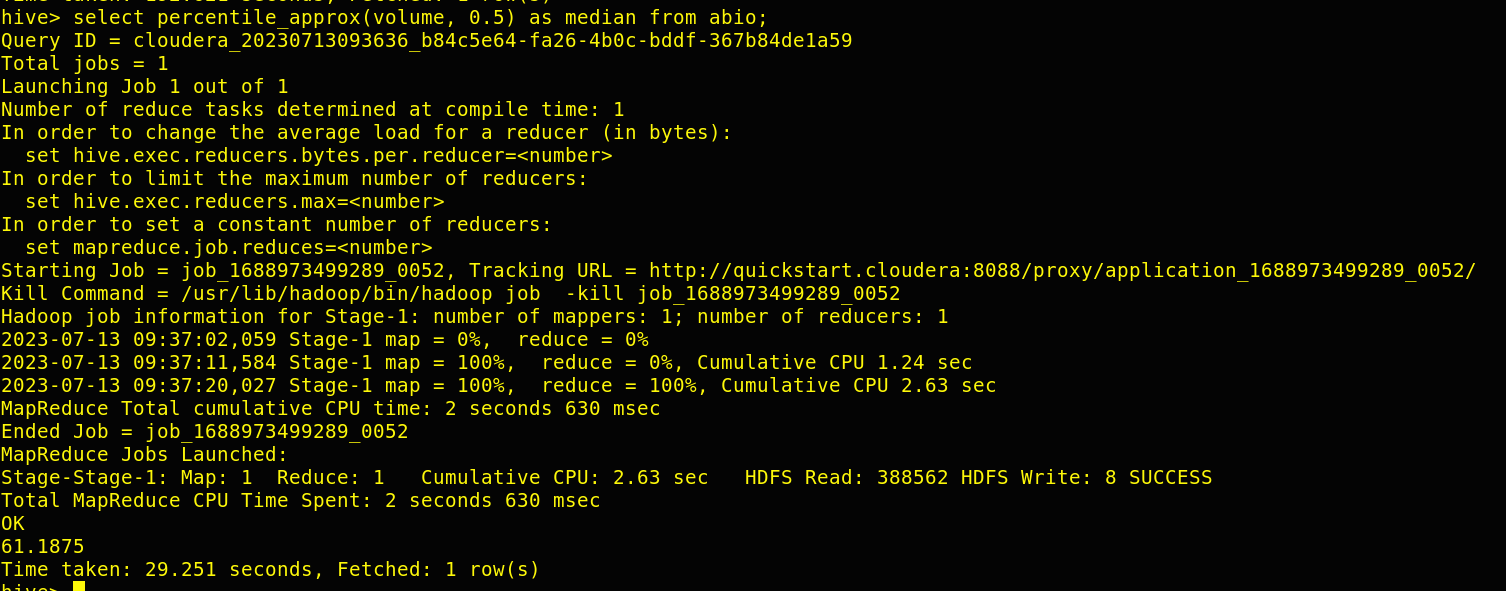
Insert overwrite table scenario6\_ext\_aal/scenario6\_ext\_ aaoi/scenario6\_ext\_abio/scenario6\_ext\_abmd

select from\_unixtime(date) as req\_date from( select unix\_timestamp(date, 'dd-MM-yyyy') as date,lag(Close) over (order by unix\_timestamp(date, 'dd-MM-yyyy')) as previous\_close, Open as current\_open from aal/aaoi/abio/abmd) subquery where abs(current\_open - previous\_close )> 0 order by req\_date;

**Scenario 7:**  Find median of volume for ABIO.

Query:

select percentile\_approx(volume, 0.5) as median from abio;

****

For creating external table, I have used this query:

CREATE EXTERNAL TABLE scenario7\_external (

from\_unixtime string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ',' STORED AS TEXTFILE

LOCATION '/user/cloudera/projectresult/ scenario7\_external;

And I have used this query to insert result into my external table:

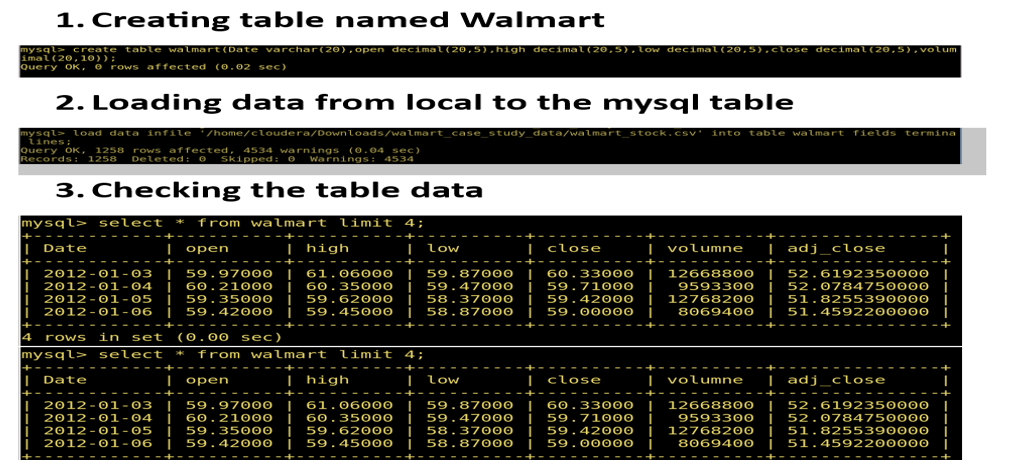
Insert overwrite table scenario7\_external select percentile\_approx(volume, 0.5) as median from abio;

**FILE: 2** 🡪 **Walmart Stock**

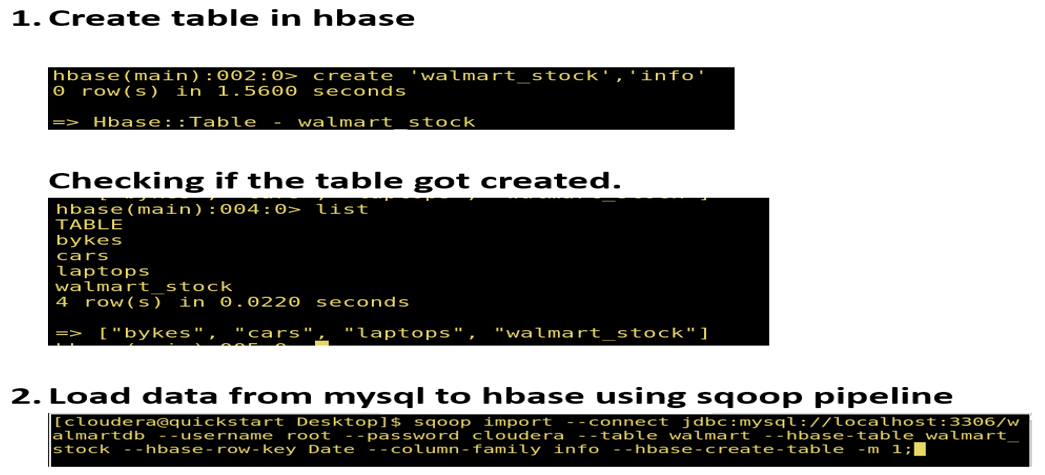
I have downloaded the Walmart Stock csv file directly to my google collab from where I am accessing the data and running my commands get the result of various scenario.

**Scenario 1:** Print out first 5 columns.

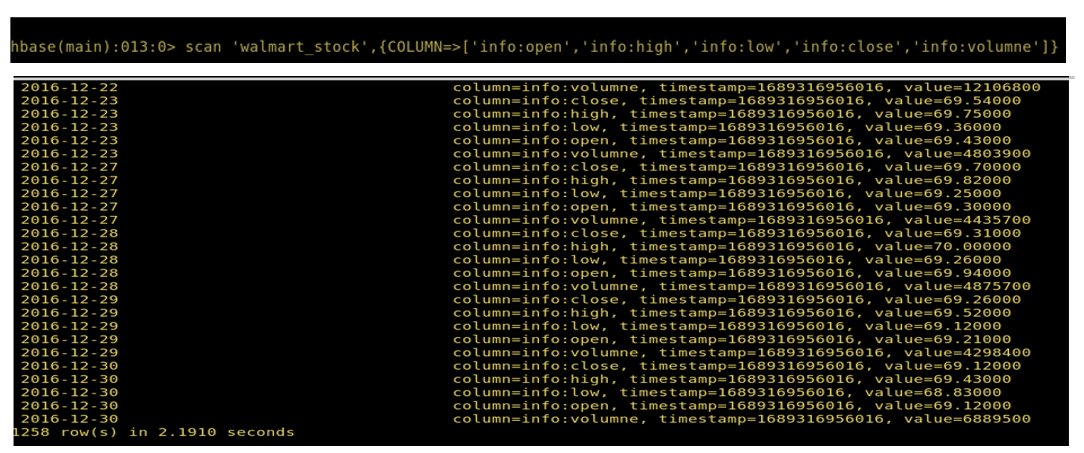
Loading the data from local system to MySQL-



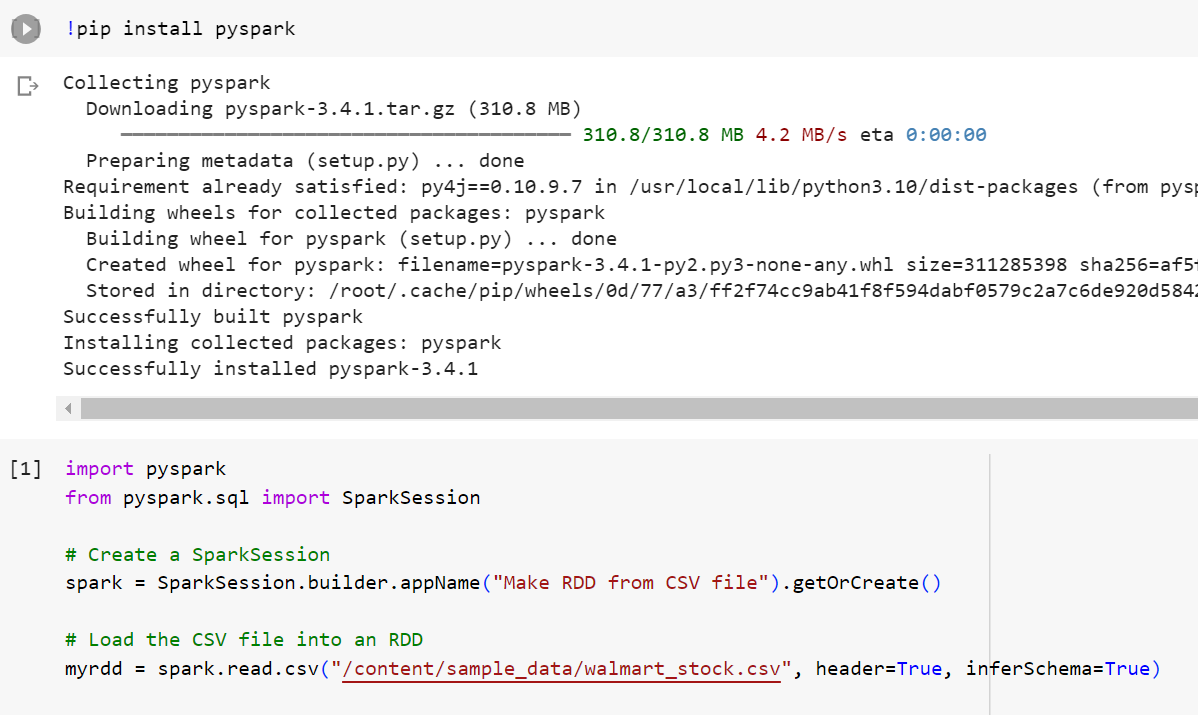
Loading data into HBASE-

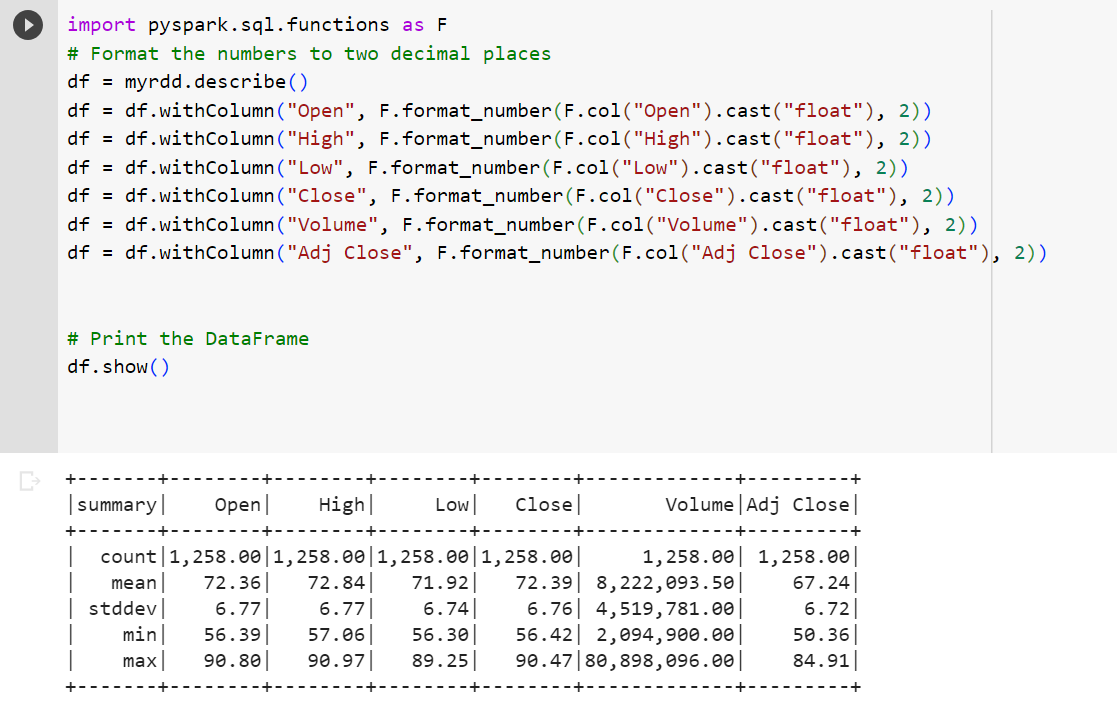


Printing out first 5 columns-

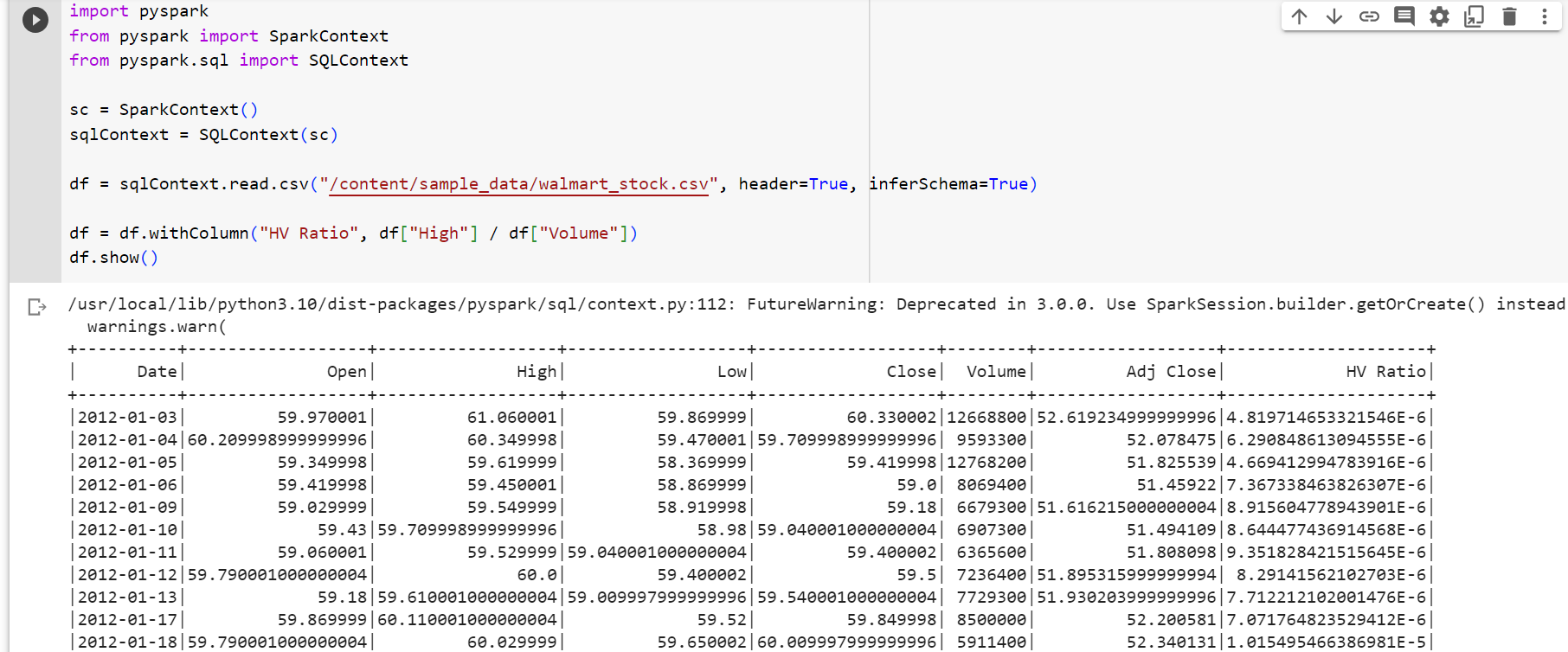


**Scenario 2:** There are too many decimal places for mean and stddev in the describe() dataframe. Format the numbers to just show up to two decimal places.





**Scenario 3:** Create a new dataframe with a column called HV Ratio that is the ratio of the High Price versus volume of stock traded for a day.



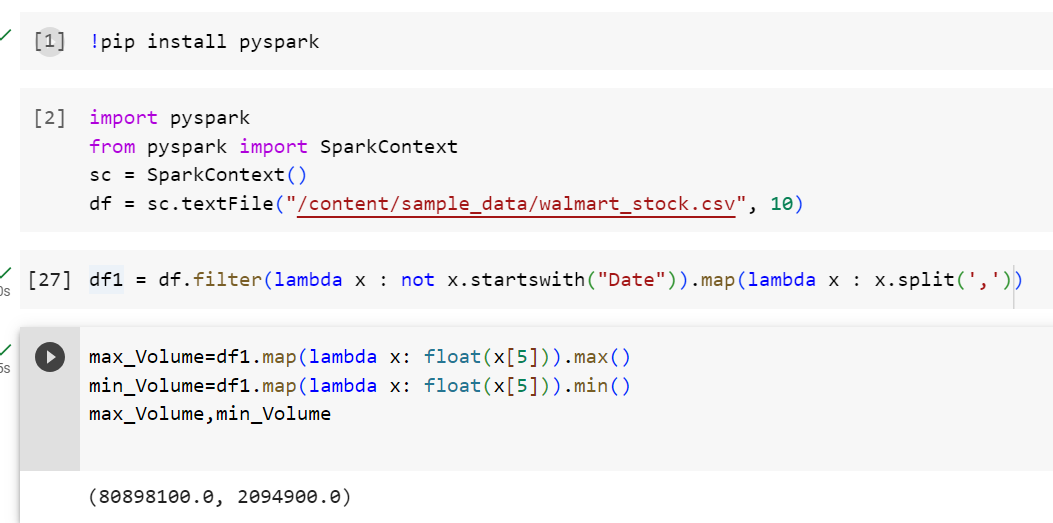
**Scenario 4:** What day had the Peak High in Price?

****

**Scenario 5:** What is the mean of the Close column??

****

**Scenario 6:** What is the max and min of the Volume column?

****

**Scenario 7:** How many days was the Close lower than 60 dollars?

****

**Scenario 8:** What percentage of the time was the High greater than 80 dollars?

****

**Scenario 9:** What is the max High per year?

****