Case-Study – Stocks Data

Task 1:

List out closing price for the day along with the yesterday's closing price.

Ans:

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/closing_lag'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

SELECT Ticker,date_,Close,lag(Close,1)

over(partition by Ticker) as yesterday_price FROM stocks;

ScreenShot:

```
[acadgild@localhost hive]$ hive -f Stock_Data_Analysis.hql
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBi
nder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Stat
icLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/home/acadgild/install/hive/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/hive-log4j2.prop
erties Async: true
OK
Time taken: 13.86 seconds
OK
Time taken: 0.079 seconds
OK
Time taken: 0.667 seconds
Loading data to table custom.stocks
OK
Time taken: 4.085 seconds
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. sp
ark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180802111228_a9bca85d-7369-4557-a393-a7a630a7b581
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1533167278642_0033, Tracking URL = http://localhost:8088/proxy/application_1533167278642_0033/
Kill Command = /home/acadgild/install/hadoop/hadoop-2.6.5/bin/hadoop job  -kill job_1533167278642_0033
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
```

```
[acadgild@localhost closing_lag]$ less 000000_0
A       20100721        27.58   \N
A       20100722        28.72   27.58
A       20100723        29.3    28.72
A       20100726        29.64   29.3
A       20100727        28.87   29.64
A       20100728        28.78   28.87
A       20100729        28.15   28.78
A       20100730        27.93   28.15
A       20100802        28.82   27.93
A       20100803        27.84   28.82
A       20100804        28.29   27.84
A       20100805        28.46   28.29
A       20100806        28.73   28.46
A       20100809        29.82   28.73
A       20100810        29.46   29.82
A       20100811        28.22   29.46
A       20100812        27.53   28.22
A       20100813        27.35   27.53
A       20100816        27.16   27.35
A       20100817        29.28   27.16
A       20100819        28.54   29.28
A       20100820        28.56   28.54
A       20100820        28.56   28.56
A       20100819        28.54   28.56
A       20100817        29.28   28.54
A       20100816        27.16   29.28
A       20100813        27.35   27.16
A       20100812        27.53   27.35
A       20100811        28.22   27.53
A       20100810        29.46   28.22
A       20100809        29.82   29.46
A       20100806        28.73   29.82
A       20100805        28.46   28.73
A       20100804        28.29   28.46
A       20100803        27.84   28.29
A       20100802        28.82   27.84
```

Task 2:

--Find out whether the following day's closing price is higher or lesser than today's.

```
INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/closing_lead'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select Ticker,date_,Close,case(lead(Close,1) over(partition by Ticker)-Close)>0

when true then "higher" when false then "lesser" end as Changes from stocks;
```

ScreenShot:

```
[acadgild@localhost closing_lead]$ less 000000_0
A       20100721        27.58   higher
A       20100722        28.72   higher
A       20100723        29.3    higher
A       20100726        29.64   lesser
A       20100727        28.87   lesser
A       20100728        28.78   lesser
A       20100729        28.15   lesser
A       20100730        27.93   higher
A       20100802        28.82   lesser
A       20100803        27.84   higher
A       20100804        28.29   higher
A       20100805        28.46   higher
A       20100806        28.73   higher
A       20100809        29.82   lesser
A       20100810        29.46   lesser
A       20100811        28.22   lesser
A       20100812        27.53   lesser
A       20100813        27.35   lesser
A       20100816        27.16   higher
A       20100817        29.28   lesser
A       20100819        28.54   higher
A       20100820        28.56   lesser
A       20100820        28.56   lesser
A       20100819        28.54   higher
A       20100817        29.28   lesser
A       20100816        27.16   higher
A       20100813        27.35   higher
A       20100812        27.53   higher
A       20100811        28.22   higher
A       20100810        29.46   higher
A       20100809        29.82   lesser
A       20100806        28.73   lesser
A       20100805        28.46   lesser
A       20100804        28.29   lesser
A       20100803        27.84   higher
A       20100802        28.82   lesser
```

Task 3:

--Find the highest price of the ticker for all the days.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/first_high'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,date_,first_value(high) over(partition by ticker)

as first_high from stocks;

ScreenShot:

```
[acadgild@localhost first_high]$ less 000000_0
A       20100721        28.2
A       20100722        28.2
A       20100723        28.2
A       20100726        28.2
A       20100727        28.2
A       20100728        28.2
A       20100729        28.2
A       20100730        28.2
A       20100802        28.2
A       20100803        28.2
A       20100804        28.2
A       20100805        28.2
A       20100806        28.2
A       20100809        28.2
A       20100810        28.2
A       20100811        28.2
A       20100812        28.2
A       20100813        28.2
A       20100816        28.2
A       20100817        28.2
A       20100819        28.2
A       20100820        28.2
A       20100820        28.2
A       20100819        28.2
A       20100817        28.2
A       20100816        28.2
A       20100813        28.2
A       20100812        28.2
A       20100811        28.2
A       20100810        28.2
A       20100809        28.2
A       20100806        28.2
A       20100805        28.2
A       20100804        28.2
A       20100803        28.2
A       20100802        28.2
```

Task 4:

--Find the last row high price value of the ticker for all the days.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/last_high'

ROW FORMAT DELIMITED

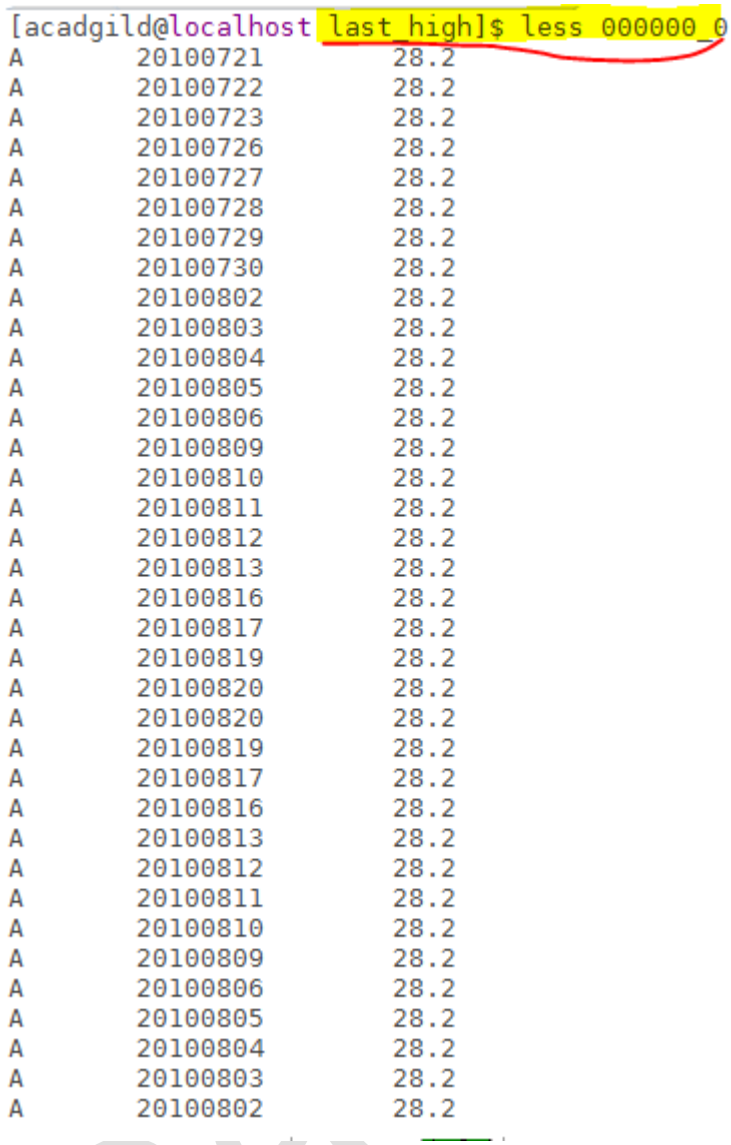
FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,date_,last_value(high) over(partition by ticker)

as last_high from stocks;

ScreenShot:



```
[acadgild@localhost last_high]$ less 000000_0
A       20100721        28.2
A       20100722        28.2
A       20100723        28.2
A       20100726        28.2
A       20100727        28.2
A       20100728        28.2
A       20100729        28.2
A       20100730        28.2
A       20100802        28.2
A       20100803        28.2
A       20100804        28.2
A       20100805        28.2
A       20100806        28.2
A       20100809        28.2
A       20100810        28.2
A       20100811        28.2
A       20100812        28.2
A       20100813        28.2
A       20100816        28.2
A       20100817        28.2
A       20100819        28.2
A       20100820        28.2
A       20100820        28.2
A       20100819        28.2
A       20100817        28.2
A       20100816        28.2
A       20100813        28.2
A       20100812        28.2
A       20100811        28.2
A       20100810        28.2
A       20100809        28.2
A       20100806        28.2
A       20100805        28.2
A       20100804        28.2
A       20100803        28.2
A       20100802        28.2
```

Task 5:

--Find the number of rows present for each ticker.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/count_ticker'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,count(ticker) over(partition by ticker)

as count_ticker from stocks;

ScreenShot:

```
[acadgild@localhost count_ticker]$ less 000000_0
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
A       44
```

Task 6:

--Find the sum of all the closing stock prices for that particular ticker.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/sum_closing'

ROW FORMAT DELIMITED

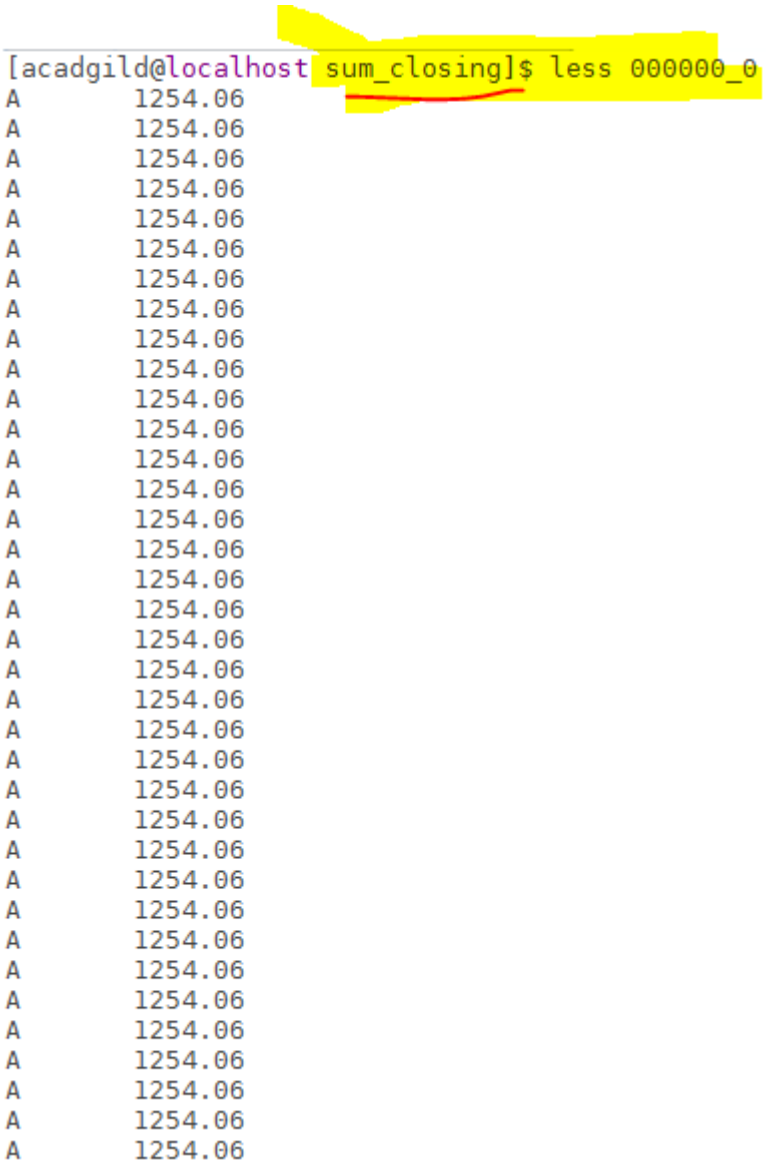
FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,sum(close) over(partition by ticker)

as total from stocks;

ScreenShot:



Task 7:

--Find the running total of the volume_for_the_day for all the days for every ticker

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/total_Vol_day'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

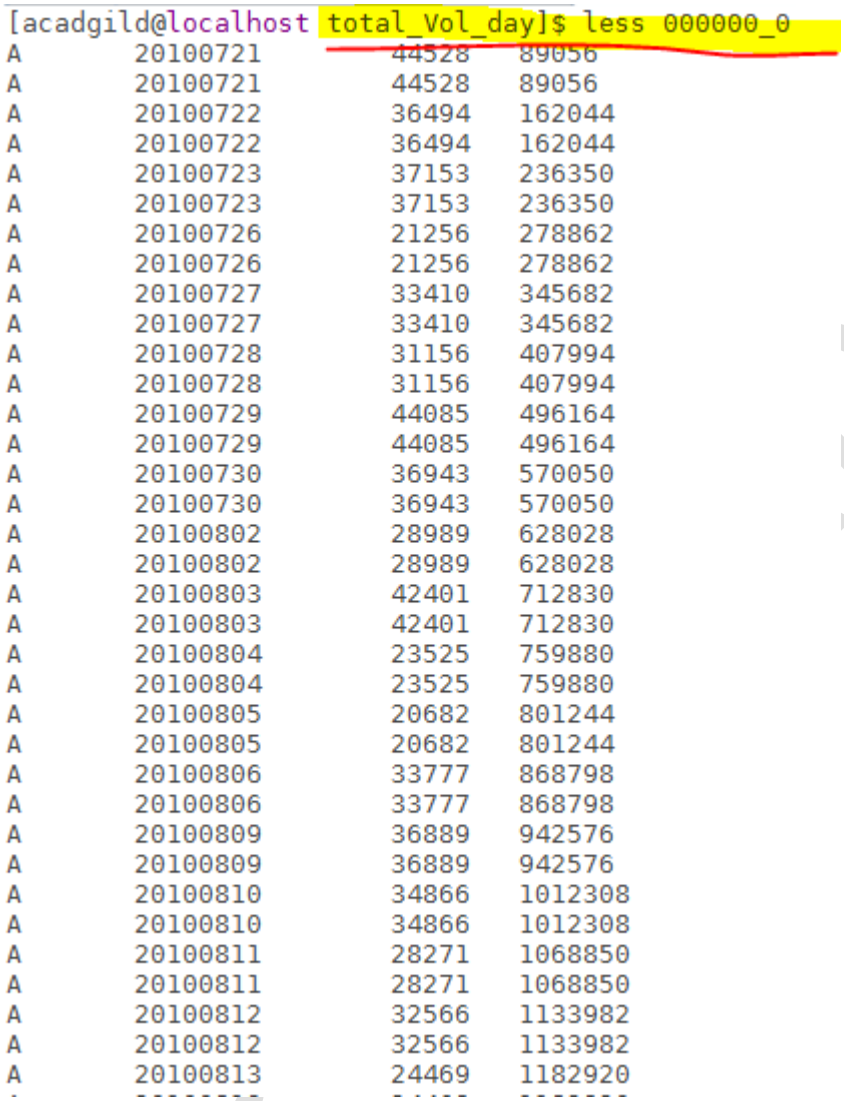
STORED AS TEXTFILE

select ticker,date_,volume_for_the_day,sum(volume_for_the_day)

over(partition by ticker order by date_)

as running_total from stocks;

ScreenShot:



```
[acadgild@localhost total_Vol_day]$ less 000000_0
A       20100721        44528   89056
A       20100721        44528   89056
A       20100722        36494   162044
A       20100722        36494   162044
A       20100723        37153   236350
A       20100723        37153   236350
A       20100726        21256   278862
A       20100726        21256   278862
A       20100727        33410   345682
A       20100727        33410   345682
A       20100728        31156   407994
A       20100728        31156   407994
A       20100729        44085   496164
A       20100729        44085   496164
A       20100730        36943   570050
A       20100730        36943   570050
A       20100802        28989   628028
A       20100802        28989   628028
A       20100803        42401   712830
A       20100803        42401   712830
A       20100804        23525   759880
A       20100804        23525   759880
A       20100805        20682   801244
A       20100805        20682   801244
A       20100806        33777   868798
A       20100806        33777   868798
A       20100809        36889   942576
A       20100809        36889   942576
A       20100810        34866   1012308
A       20100810        34866   1012308
A       20100811        28271   1068850
A       20100811        28271   1068850
A       20100812        32566   1133982
A       20100812        32566   1133982
A       20100813        24469   1182920
```

Task 8:

--Find the percentage of the volume_for_the_day on the total volumes for that particular ticker

```
INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/percent_total_Vol_day'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select
ticker,date_,volume_for_the_day,(volume_for_the_day*100/(sum(volume_for_th
e_day)

over(partition by ticker))) from stocks;
```

ScreenShot:

```
[acadgild@localhost percent_total_Vol_day]$ less 000000_0
A       20100721        44528   2.9091441257214385
A       20100722        36494   2.3842594709863048
A       20100723        37153   2.4273138632529783
A       20100726        21256   1.3887164825802845
A       20100727        33410   2.1827727551283074
A       20100728        31156   2.0355123603345566
A       20100729        44085   2.8802016435148587
A       20100730        36943   2.4135939506945543
A       20100802        28989   1.8939359293150104
A       20100803        42401   2.7701810113796874
A       20100804        23525   1.536956871128208
A       20100805        20682   1.3512153882539255
A       20100806        33777   2.206749935647077
A       20100809        36889   2.410065973179531
A       20100810        34866   2.277897482200047
A       20100811        28271   1.8470268949485895
A       20100812        32566   2.127631773226832
A       20100813        24469   1.598631144724171
A       20100816        33231   2.1710781629951743
A       20100817        73050   4.772569582823192
A       20100819        33832   2.210343246079045
A       20100820        33738   2.2042019518862266
A       20100820        33738   2.2042019518862266
A       20100819        33832   2.210343246079045
A       20100817        73050   4.772569582823192
A       20100816        33231   2.1710781629951743
A       20100813        24469   1.598631144724171
A       20100812        32566   2.127631773226832
A       20100811        28271   1.8470268949485895
A       20100810        34866   2.277897482200047
A       20100809        36889   2.410065973179531
A       20100806        33777   2.206749935647077
A       20100805        20682   1.3512153882539255
A       20100804        23525   1.536956871128208
A       20100803        42401   2.7701810113796874
A       20100802        28989   1.8939359293150104
```

Task 9:

--Find the minimum closing stock price for each particular ticker.

INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/Desktop/TestHadoop/hive/output/min_ticker'

ROW FORMAT DELIMITED

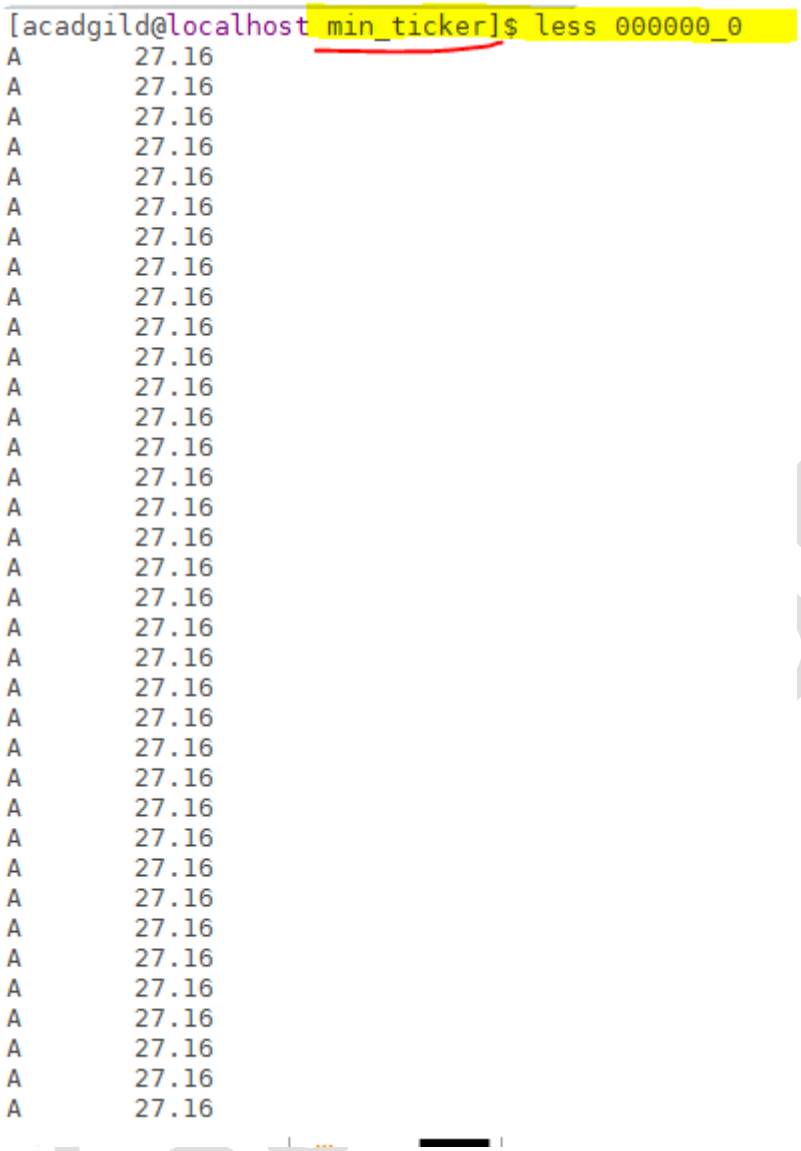
FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker, min(close) over(partition by ticker) as minimum

from stocks;

ScreenShot:



Task 10:

--Find the maximum closing stock price for each particular ticker.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/max_ticker'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker, max(close) over(partition by ticker) as maximum

from stocks;

ScreenShot:

```
[acadgild@localhost max_ticker]$ less 000000_0
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
A       29.82
```

Task 11:

--Find the average closing stock price for each particular ticker.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/avg_ticker'

ROW FORMAT DELIMITED

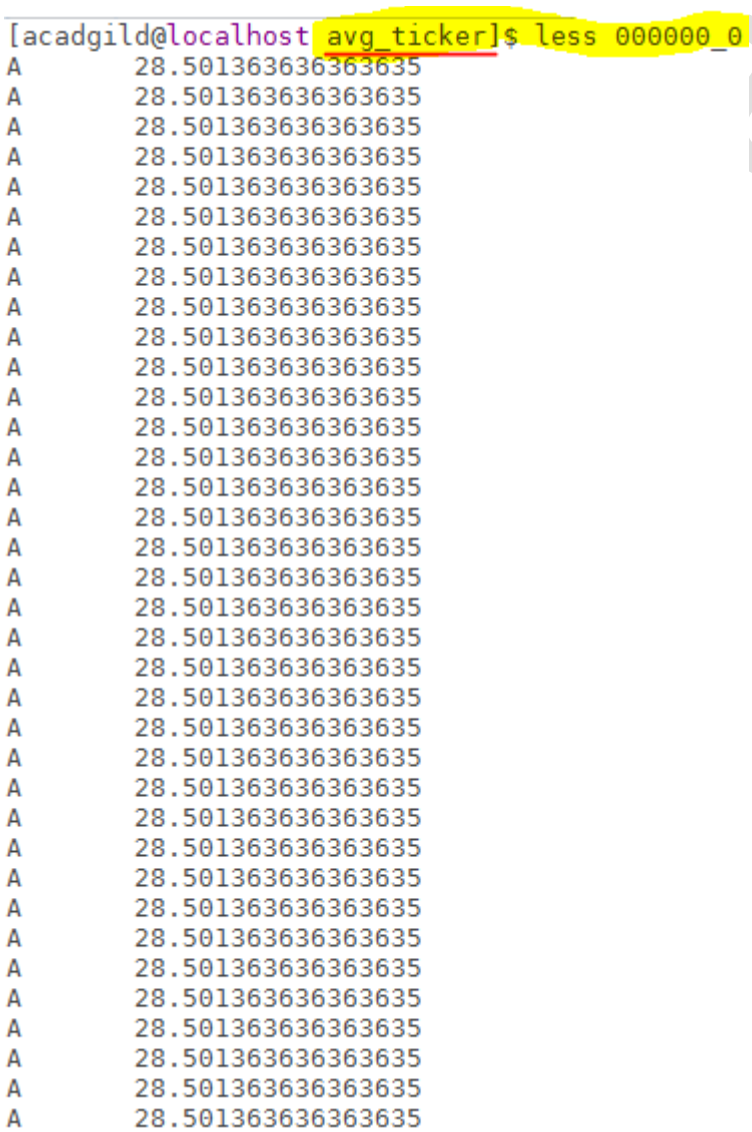
FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker, avg(close) over(partition by ticker) as average

from stocks;

ScreenShot:



Task 12:

--How to rank the closing prices of the stock for each ticker.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/rank_ticker'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,close,rank() over(partition by ticker order by close) as closing

from stocks;

ScreenShot:

```
[acadgild@localhost rank_ticker]$ less 000000_0
A       27.16   1
A       27.16   1
A       27.35   3
A       27.35   3
A       27.53   5
A       27.53   5
A       27.58   7
A       27.58   7
A       27.84   9
A       27.84   9
A       27.93   11
A       27.93   11
A       28.15   13
A       28.15   13
A       28.22   15
A       28.22   15
A       28.29   17
A       28.29   17
A       28.46   19
A       28.46   19
A       28.54   21
A       28.54   21
A       28.56   23
A       28.56   23
A       28.72   25
A       28.72   25
A       28.73   27
A       28.73   27
A       28.78   29
A       28.78   29
A       28.82   31
A       28.82   31
A       28.87   33
A       28.87   33
A       29.28   35
A       29.28   35
```

Task 13:

--How to get the ticker, closing price and its row number for each ticker.

```
INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/rowNum_ticker'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
select ticker,close,row_number() over(partition by ticker order by close)
as num from stocks;
```

ScreenShot:

```
[acadgild@localhost rowNum_ticker]$ less 000000_0
A       27.16   1
A       27.16   2
A       27.35   3
A       27.35   4
A       27.53   5
A       27.53   6
A       27.58   7
A       27.58   8
A       27.84   9
A       27.84   10
A       27.93   11
A       27.93   12
A       28.15   13
A       28.15   14
A       28.22   15
A       28.22   16
A       28.29   17
A       28.29   18
A       28.46   19
A       28.46   20
A       28.54   21
A       28.54   22
A       28.56   23
A       28.56   24
A       28.72   25
A       28.72   26
A       28.73   27
A       28.73   28
A       28.78   29
A       28.78   30
A       28.82   31
A       28.82   32
A       28.87   33
A       28.87   34
A       29.28   35
A       29.28   36
```

Task14:

--How would you rank the closing prices of the stock for each ticker.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/denseRank_ticker'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,close,dense_rank() over(partition by ticker order by close) as

closing from stocks;

ScreenShot:

```
[acadgild@localhost denseRank_ticker]$ less 000000_0
A       27.16   1
A       27.16   1
A       27.35   2
A       27.35   2
A       27.53   3
A       27.53   3
A       27.58   4
A       27.58   4
A       27.84   5
A       27.84   5
A       27.93   6
A       27.93   6
A       28.15   7
A       28.15   7
A       28.22   8
A       28.22   8
A       28.29   9
A       28.29   9
A       28.46   10
A       28.46   10
A       28.54   11
A       28.54   11
A       28.56   12
A       28.56   12
A       28.72   13
A       28.72   13
A       28.73   14
A       28.73   14
A       28.78   15
A       28.78   15
A       28.82   16
A       28.82   16
A       28.87   17
A       28.87   17
A       29.28   18
A       29.28   18
```

Task 15:

--Find the cumulative of each record for every ticker.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/cummu_ticker'

ROW FORMAT DELIMITED

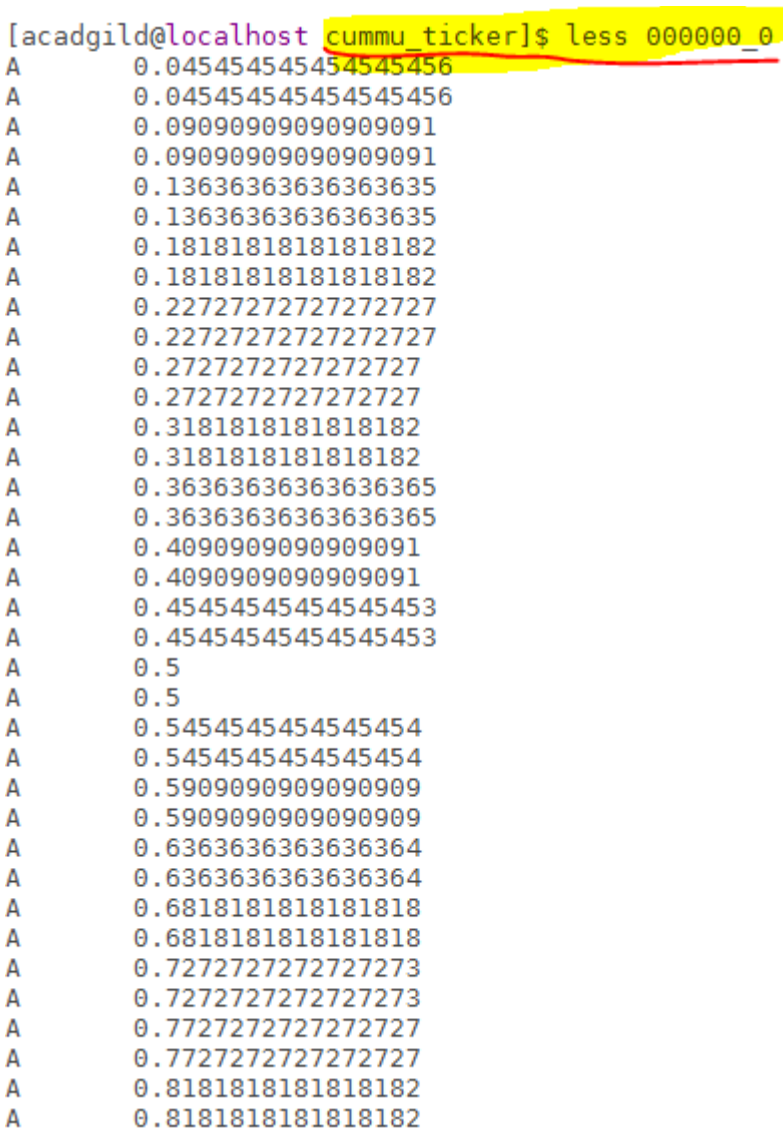
FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,cume_dist() over(partition by ticker order by close)

as cummulative from stocks;

ScreenShot:



Task 16:

--Calculate the percent_rank for every row in each partition.

INSERT OVERWRITE LOCAL DIRECTORY
'/home/acadgild/Desktop/TestHadoop/hive/output/percentRank_ticker'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,close,percent_rank() over(partition by ticker order by close)

as closing from stocks;

ScreenShot:

```
[acadgild@localhost percentRank_ticker]$ less 000000_0
A       27.16   0.0
A       27.16   0.0
A       27.35   0.046511627906976744
A       27.35   0.046511627906976744
A       27.53   0.09302325581395349
A       27.53   0.09302325581395349
A       27.58   0.13953488372093023
A       27.58   0.13953488372093023
A       27.84   0.18604651162790697
A       27.84   0.18604651162790697
A       27.93   0.23255813953488372
A       27.93   0.23255813953488372
A       28.15   0.27906976744186046
A       28.15   0.27906976744186046
A       28.22   0.32558139534883723
A       28.22   0.32558139534883723
A       28.29   0.37209302325581395
A       28.29   0.37209302325581395
A       28.46   0.4186046511627907
A       28.46   0.4186046511627907
A       28.54   0.46511627906976744
A       28.54   0.46511627906976744
A       28.56   0.5116279069767442
A       28.56   0.5116279069767442
A       28.72   0.5581395348837209
A       28.72   0.5581395348837209
A       28.73   0.6046511627906976
A       28.73   0.6046511627906976
A       28.78   0.6511627906976745
A       28.78   0.6511627906976745
A       28.82   0.6976744186046512
A       28.82   0.6976744186046512
A       28.87   0.7441860465116279
A       28.87   0.7441860465116279
A       29.28   0.7906976744186046
A       29.28   0.7906976744186046
```

Task 17:

--Create 5 buckets for every ticker such that, the first 20% records for every ticker will be in the 1st bucket and so on.

INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/Desktop/TestHadoop/hive/output/bucket_ticker'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

select ticker,ntile(5) over(partition by ticker order by close) as

bucket from stocks;

ScreenShot:

```
[acadgild@localhost bucket_ticker]$ less 000000_0
A       1
A       1
A       1
A       1
A       1
A       1
A       1
A       1
A       1
A       2
A       2
A       2
A       2
A       2
A       2
A       2
A       2
A       2
A       2
A       3
A       3
A       3
A       3
A       3
A       3
A       3
A       3
A       3
A       4
A       4
A       4
A       4
A       4
A       4
```

********** **End** **********