STOCK ANALYSIS

STOCK MARKET ANALYSIS: BATCH PROCESS WITH HIVE

Making separate database and then table schema(s) in the MySQL

```
mysql> use stocksdb;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> create table abio(date varchar(20),low decimal (20,10),open decimal (20,10),volumne int,high decimal (20,10),close deci
mal (20,10),adjusted_close decimal (20,10));
Query OK, 0 rows affected (0.02 sec)

mysql> create table aaoi(date varchar(20),low decimal (20,10),open decimal (20,10),volumne int,high decimal (20,10),close deci
mal (20,10),adjusted_close decimal (20,10));
Query OK, 0 rows affected (0.01 sec)

mysql> create table abmd(date varchar(20),low decimal (20,10),open decimal (20,10),volumne int,high decimal (20,10),close deci
mal (20,10),adjusted_close decimal (20,10));
Query OK, 0 rows affected (0.00 sec)

mysql> create table aal(date varchar(20),low decimal (20,10),open decimal (20,10),volumne int,high decimal (20,10),close decim
al (20,10),adjusted_close decimal (20,10));
Query OK, 0 rows affected (0.01 sec)
```

 Loading data of stocks like AAOI, ABIO, ABMD, AAL into their respective tables

```
nysql> Load data infile '/home/cloudera/Downloads/walmart_case_study_data/AAOI.csv' into table aaoi fields terminated by ','
gnore 1 lines;
Query OK, 2320 rows affected, 10805 warnings (0.03 sec)
Records: 2320 Deleted: 0 Skipped: 0 Warnings: 10805
nysql> select * from aaoi limit 5;
 date
             low
                                               volumne | high
                                                                        I close
                                                                                        | adjusted close
                               open
 26-09-2013
                9.3699998856
                               10.0000000000
                                                946000
                                                          10.0900001526
                                                                           9.9600000381
                                                                                            9.9600000381
 27-09-2013
               10.0000000000
                                                253300
                                                                          10.1000003815
                                                                                           10.1000003815
                               10.4399995804
                                                          10.4399995804
 30-09-2013
               9.7100000381
                               10.0000000000
                                                                          10.0000000000
                                                                                           10.0000000000
                                                 84900
                                                          10.1800003052
                                9.9499998093
 01-10-2013
                9.9200000763
                                                 74500
                                                          10.0200004578
                                                                          10.0000000000
                                                                                           10.0000000000
                9.8900003433
                                                                           9.9700002670
                                9.9899997711
                                                          10.0000000000
                                                                                            9.9700002670
 02-10-2013
                                                 94000
 rows in set (0.00 sec)
nysql> Load data infile '/home/cloudera/Downloads/walmart_case_study_data/ABIO.csv' into table_abio fields terminated by ',' i
Query OK, 6379 rows affected, 16004 warnings (0.05 sec)
Records: 6379 Deleted: 0 Skipped: 0 Warnings: 16004
```

```
mysql> Load data infile '/home/cloudera/Downloads/walmart_case_study_data/ABMD.csv' into table abmd fields terminated by ',' i
gnore 1 lines;
Query OK, 8916 rows affected, 25006 warnings (0.07 sec)
Records: 8916 Deleted: 0 Skipped: 0 Warnings: 25006
mysql> Load data infile '/home/cloudera/Downloads/walmart_case_study_data/AAL.csv' into table aal fields terminated by ',' ign
ore 1 lines;
Query OK, 4333 rows affected, 20438 warnings (0.04 sec)
Records: 4333 Deleted: 0 Skipped: 0 Warnings: 20438
```

then I have merged all the tables into one and added a column for their corresponding stock name

```
mysql> create table stocksdata as select * from aal union all select * from aaoi union all select * from abio union all select
* from abmd;
Query OK, 21948 rows affected (0.49 sec)
Records: 21948 Duplicates: O Warnings: O
```

Cross-checking the data if it has come right:

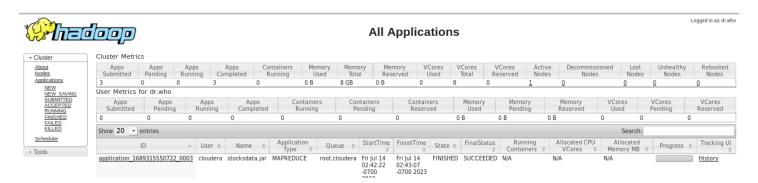
```
nysql> select * from stocksdata limit 5;
 date
                              open
                                              volumne | high
                                                                        | close
                                                                                          adjusted_close | stock_name
 27-09-2005
              19.1000003815
                              21.0499992371
                                               961200
                                                         21.3999996185
                                                                         19.2999992371
                                                                                           18.1949100494
                                                                                                           aal
                                                         20.5300006866
                                                                         20.5000000000
 28-09-2005
              19.2000007629
                              19.2999992371
                                               5747900
                                                                                           19.3262042999
              20.1000003815
                              20.3999996185
                                                                                           19.0528049469
 29-09-2005
                                               1078200
                                                         20.5799999237
                                                                         20.2099990845
                                                                                                           aal
 30-09-2005
              20.1800003052
                              20.2600002289
                                                         21.0499992371
                                               3123300
                                                                         21.0100002289
                                                                                           19.8070011139
                                                                                                           aal
 03-10-2005
              20.8999996185
                              20.8999996185
                                                         21.7500000000
                                                                         21.5000000000
                                                                                           20.2689399719
 rows in set (0.00 sec)
```

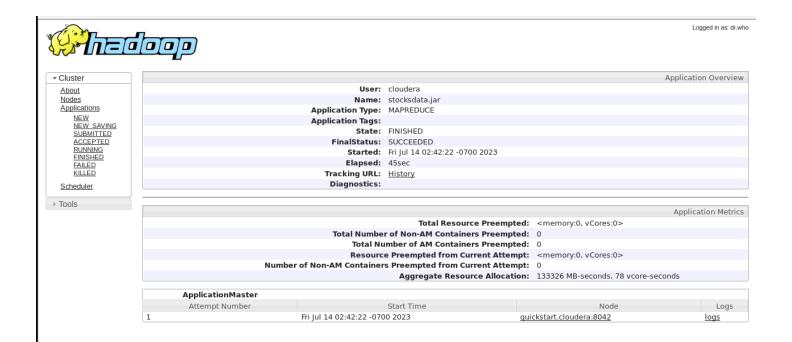
DATA INGESTION using SQOOP:

Using sqoop to import the table from MySQL to hive warehouse in a particular database (hence internal table is created)

[cloudera@quickstart walmart_case_study_data]\$ sqoop import-all-tables --connect jdbc:mysql://localhost:3306/stocksdb --userna me root --password cloudera --hive-import --hive-database stocksdb -m 1

CHECKING THE JOB STATUS:





All tables have been successfully imported to a database named stocksdb in my hive warehouse, in a table named stocks data.

Checking if the data has successfully transferred:

```
hive> show tables;
0K
stocksdata
Time taken: 0.026 seconds, Fetched: 1 row(s)
hive> desc stocksdata;
0K
date
                         string
                         double
low
                         double
open
volumne
                         int
high
                         double
close
                         double
adjusted close
                         double
stock name
                         string
 ime taken: 0.229 seconds, Fetched: 8 row(s)
```

```
nive> select * from stocksdata limit 5;
27 - 09 - 2005
                                21.0499992371
                                                                         19.29999923
                19.1000003815
                                                 961200 21.3999996185
       18.1949100494
28-09-2005
                19.2000007629
                                19.2999992371
                                                 5747900 20.5300006866
3262042999
                aal
29-09-2005
                20.1000003815
                                20.3999996185
                                                 1078200 20.5799999237
                                                                         20.20999908
        19.0528049469
45
                        aal
                                                 3123300 21.0499992371
30-09-2005
                20.1800003052
                                20.2600002289
                                                                         21.01000022
        19.8070011139
                        aal
                20.8999996185
03-10-2005
                                20.8999996185
                                                 1057900 21.75
                                                                 21.5
                                                                         20.26893997
        aal
           0.17 seconds. Fetched: 5 row(s)
```

CLIENT USERSTORIES:

I am creating partitioning as well as buckets on the table to improve optimization, I have created partition on stock_name and made a bucket on the date column with 4 buckets.

Cmd:

```
hive> create table stocksoptm(date string, low double,open double, volumne int, high double,close double,adjusted_
close double) partitioned by (stock_name string) clustered by (date) into 4 buckets row format delimited fields te
rminated by ',';
OK
Time taken: 0.093 seconds
```

```
10 create table stocksoptm(
11 date string, low double,open double,
12 volumne int, high double,close double,
13 adjusted_close double)
14 partitioned by (stock_name string)
15 clustered by (date) into 4 buckets |
16 row format delimited fields terminated by ',';
17
```

Activating the partitioning and bucketing in hive as its not activated by default using commands:

```
hive> set hive.enforce.bucketing=true;
hive> set hive.exec.dynamic.partition=true;
hive> set hive.exec.dynamic.partition.mode=nonstrict;;
hive>
```

Inserting data into the newly created partitioned table named stocksoptm.

```
hive> insert overwrite table stocksoptm partition(stock_name) select date,low,open,volume,high,close,adjusted_clos
e,stock_name from stocksdata;<mark>=</mark>
```

Checking if the data has come and got partitioned

```
      45.25
      46.2599983215
      1219200
      46.8100013733
      45.8499984741

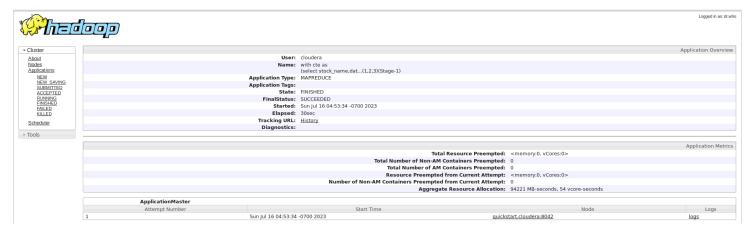
      43.8499984741
      46.0
      1009700
      46.1199989319
      44.6100006104

      12.2899999619
      12.3500003815
      37500000
      12.5699996948

                                                                                                                                   43.2246932983
                                                                                                                                  42.0556907654
12.3599996567
                                                                                                                                                              aal
                                                                                                                                                              12.3599996567
14-10-2020
23-05-2006
16-05-2006
                     45.090000152647.06000137331062500 47.2545.189998626747.529998779349.575530049.599998474147.8499984741
                                                                                                                                   42.602481842
                                                                                                                                                              aal
                                                                                                                                   45.1101760864
                                                                                                                                                              aal
Time taken: 0.068 seconds, Fetched: 5 row(s) hive> show partitions stocksoptm;
stock name=aal
stock_name=aaoi
 stock_name=abio
 stock name=abmd
Time taken: 0.086 seconds, Fetched: 4 row(s)
```

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.

```
Total MapReduce CPU Time Spent: 5 seconds 570 msec
0K
        12-11-2008
                         30.266666411999996
aal
aal
        23-06-2008
                          21.604939677392927
                                                    2
aal
        21-05-2008
                          20.857992256746392
                                                    3
        20-09-2022
                          14.516127893773854
aaoi
aaoi
        05-11-2021
                          13.874188434203338
                                                    2
aaoi
        04-08-2017
                         13.866668701199993
                                                    3
abio
        20-02-2019
                         43.33333411796293
abio
        26-02-2018
                         39.99999717407408
                                                    2
abio
        14-03-2000
                                                    3
                         35.38461538461539
abmd
        19-10-1987
                         24.63768115942029
                                                    1
abmd
        08-01-2001
                         21.428571428571427
                                                    2
        09-10-2002
                          21.404685714600493
                                                    3
abmd
     taken: 33.764 seconds, Fetched: 12 row(s)
```



Creating External Table:

```
hive> create external table hive1(stock_name string,date string, percentage_change Decimal (20,10),rank int) row format delimited fields terminated by ',' location '/user/hive/warehouse/stocksdb.db/stocksexternal/hive1';
OK
Time taken: 4.894 seconds
```

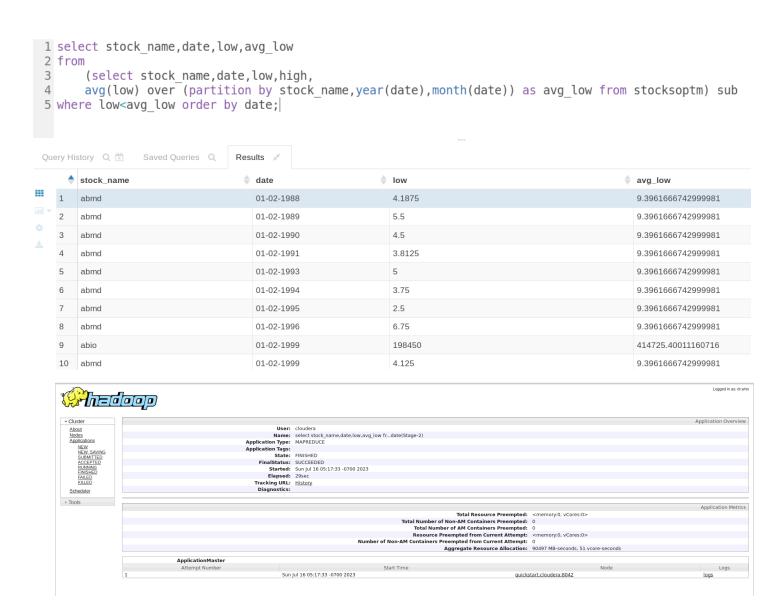
Inserting data into the External table:

```
1 with cte as
2 (select stock_name, date, (open-close)*100/open as percentage_change from stocksoptm)
3 insert overwrite table default.hivel
4 select * from (
5     select stock_name, date, percentage_change,
6     rank() over (partition by stock_name order by percentage_change desc) as rankk
7     from cte) a
8 where rankk in (1,2,3);
9
10 show tables;
11
```

Checking Data from External Table:

```
hive> select * from hive1;
0K
aal
        12-11-2008
                         30.266666412
                                          1
aal
        23-06-2008
                         21.6049396774
                                          2
aal
                                          3
        21-05-2008
                         20.8579922567
                                          1
aaoi
        20-09-2022
                         14.5161278938
                                          2
aaoi
        05-11-2021
                         13.8741884342
                         13.8666687012
                                          3
aaoi
        04-08-2017
                                          1
abio
        20-02-2019
                         43.333334118
                                          2
abio
        26-02-2018
                         39.9999971741
                                          3
abio
        14-03-2000
                         35.3846153846
                                          1
abmd
        19-10-1987
                         24.6376811594
abmd
        08-01-2001
                         21.4285714286
                                          2
        09-10-2002
                         21.4046857146
abmd
Time taken: 1.511 seconds, Fetched: 12 row(s)
```

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



Creating External Table:

```
hive> create external table hive2(stock_name string,date string,low decimal (20,10),avg_low decimal (20,15)) row format delimited fields terminated by ',' location '/user/hive/warehouse/stocksdb.db/stocksexternal/hive2';
OK
Time taken: 0.369 seconds
```

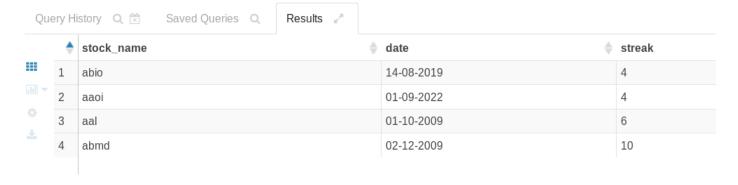
Inserting data into the External table:

Checking Data from External Table:

```
hive> select * from hive2 limit 10;
0K
abmd
        01-02-1988
                        4.1875
abmd
        01-02-1989
                        5.5
                        4.5
abmd
        01-02-1990
abmd
        01-02-1991
                        3.8125
abmd
       01-02-1993
                        5
abmd
        01-02-1994
                        3.75
       01-02-1995
abmd
                        2.5
abmd
        01-02-1996
                        6.75
abio
       01-02-1999
                        198450
abmd
        01-02-1999
                        4.125
Time taken: 0.161 seconds, Fetched: 10 row(s)
```

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

```
1 WITH ctel AS(
 2 SELECT stock name, date, close,
           LEAD(close) OVER(PARTITION BY stock name ORDER BY date) AS next,
           ROW NUMBER() OVER(PARTITION BY stock name ORDER BY date) AS rn
 5 FROM stocksoptm
6 ), cte2 AS(
7 SELECT *,
           rn - ROW NUMBER() OVER(PARTITION BY stock name ORDER BY date) AS rnk
9 FROM ctel
10 WHERE close < next
11 ), cte3 AS(
12 SELECT stock name, date,
           COUNT(*) OVER(PARTITION BY stock name, rnk) AS streak
14 FROM cte2
15 ), cte4 AS(
16 SELECT stock name,
         MIN(date) AS date,
18
          MAX(streak) AS streak,
          RANK() OVER(PARTITION BY stock name ORDER BY streak desc) as rn
19
20 FROM cte3
21 GROUP BY stock name, streak
22 ORDER BY streak
23 )
24 SELECT STOCK NAME, DATE, STREAK
25 FROM cte4 WHERE rn=1;
```



w 20 🔻 entr	ries										Search
ID	¥	User 💠	Name \$	Application Type \$	Queue \$	StartTime	FinishTime	State \$	FinalStatus	Running Containers	Allocate CPU VCores
lication_168958	32433728_0026	cloudera	WITH ctel AS(SELECT stock_name, datrn=1(Stage-6)	MAPREDUCE	root.cloudera	Mon Jul 17 21:02:14 -0700 2023	Mon Jul 17 21:02:41 -0700 2023	FINISHED	SUCCEEDED	N/A	N/A
lication_168958	32433728_0025	cloudera	WITH ctel AS(SELECT stock_name, datrn=1(Stage-5)	MAPREDUCE	root.cloudera	Mon Jul 17 21:01:23 -0700 2023	Mon Jul 17 21:02:12 -0700 2023	FINISHED	SUCCEEDED	N/A	N/A
lication_168958	32433728_0024	cloudera	WITH ctel AS(SELECT stock_name, datrn=1(Stage-4)	MAPREDUCE	root.cloudera	Mon Jul 17 21:00:03 -0700 2023	Mon Jul 17 21:01:20 -0700 2023	FINISHED	SUCCEEDED	N/A	N/A
lication_168958	32433728_0023	cloudera	WITH ctel AS(SELECT stock_name, datrn=1(Stage-3)	MAPREDUCE	root.cloudera	Mon Jul 17 20:58:32 -0700 2023	Mon Jul 17 21:00:01 -0700 2023	FINISHED	SUCCEEDED	N/A	N/A
lication_168958	32433728_0022	cloudera	WITH ctel AS(SELECT stock_name, datrn=1(Stage-2)	MAPREDUCE	root.cloudera	Mon Jul 17 20:57:03 -0700 2023	Mon Jul 17 20:58:29 -0700 2023	FINISHED	SUCCEEDED	N/A	N/A
lication_168958	32433728_0021	cloudera	WITH cte1 AS(SELECT stock_name, datrn=1(Stage-1)	MAPREDUCE	root.cloudera	Mon Jul 17 20:55:32 -0700 2023	Mon Jul 17 20:57:00 -0700 2023	FINISHED	SUCCEEDED	N/A	N/A

Creating External table:

hive> create external table hive4(date string) row format delimited fields terminated by ',' location '/user/hive/warehouse/stocksdb.db/stocksexternal/hive4';
OK
Time taken: 0.114 seconds

Inserting data into the External table:

```
2 SELECT stock name, date, close,
          LEAD(close) OVER(PARTITION BY stock name ORDER BY date) AS next,
          ROW NUMBER() OVER(PARTITION BY stock name ORDER BY date) AS rn
5 FROM stocksoptm
6 ), cte2 AS(
7 SELECT *,
          rn - ROW NUMBER() OVER(PARTITION BY stock name ORDER BY date) AS rnk
9 FROM ctel
10 WHERE close < next
11 ), cte3 AS(
12 SELECT stock_name, date,
13
          COUNT(*) OVER(PARTITION BY stock name, rnk) AS streak
14 FROM cte2
15 ), cte4 AS(
16 SELECT stock_name,
17
          MIN(date) AS date,
          MAX(streak) AS streak,
19
          RANK() OVER(PARTITION BY stock name ORDER BY streak desc) as rn
20 FROM cte3
21 GROUP BY stock_name, streak
22 ORDER BY streak
24 insert overwrite table default.hive3
25 SELECT STOCK NAME, DATE, STREAK
26 FROM cte4 WHERE rn=1;
```

Checking the data of external table:

```
hive> select * from hive3;

OK

abio 14-08-2019 4

aaoi 01-09-2022 4

aal 01-10-2009 6

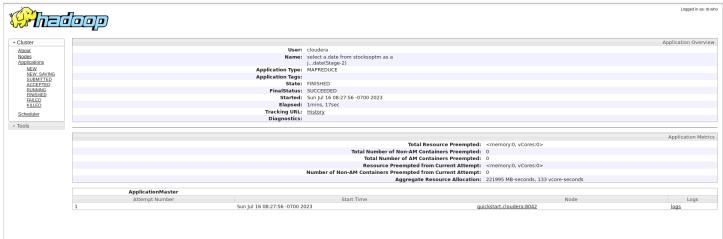
abmd 02-12-2009 10

Time taken: 0.051 seconds, Fetched: 4 row(s)
```

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 optimized way).

```
1 select a.date from stocksoptm as a
2 join stocksoptm as b on a.date=b.date join stocksoptm as c on b.date=c.date
3 where a.stock_name='aal' and b.stock_name='aaoi'
4          and c.stock_name='abmd'
5          and (a.open>b.open or a.volume>c.volume)
6 order by date;
7
```





Creating External table:

```
hive> create external table hive4(date string) row format delimited fields terminated by ',' location '/user/hive/warehouse/stocksdb.db/stocksexternal/hive4';
OK
Time taken: 0.114 seconds
```

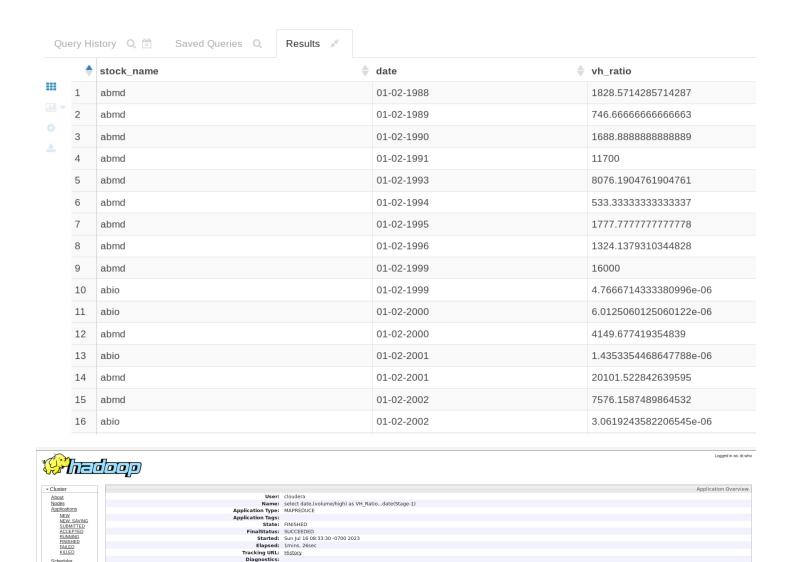
Inserting data into the External table:

Checking Data from External Table:

```
hive> select * from hive4 limit 15;
0K
01-02-2016
01-02-2017
01-02-2018
01-02-2019
01-02-2021
01-02-2022
01-03-2016
01-03-2017
01-03-2018
01-03-2019
01-03-2021
01-03-2022
01-04-2014
01-04-2015
01-04-2016
Time taken: 0.117 seconds, Fetched: 15 row(s)
```

5. write a Hive query to calculate VH ratio(volume to high ratio).

```
1 select stock_name, date, (volume/high) as VH_Ratio
2 from stocksoptm
3 order by date;
```



Creating External table:

ApplicationMaster

hive> create external table hive5(stock_name string,date string,vh_ratio decimal(20,10)) row format delimited fields terminated by ',' location '/user/hive/warehouse/stocksdb.db/stocksexternal/hive5';
OK
Time taken: 0.135 seconds

Total Resource Preempted: <memory:0, vCores:0>
Total Number of Non-AM Containers Preempted: 0
Total Number of AM Containers Preempted: 0
Resource Preempted from Current Attempt: <memory.0, vCores:0>
r of Non-AM Containers Preempted from Current Attempt: 0
Aggregate Resource Allocation: 249154 M8-seconds, 150 vcore-second

guickstart.cloudera:8042

Inserting data into the External table:

- 1 insert into default.hive5
- 2 select stock name, date, (volume/high) as VH_Ratio

Sun Jul 16 08:33:30 -0700 2023

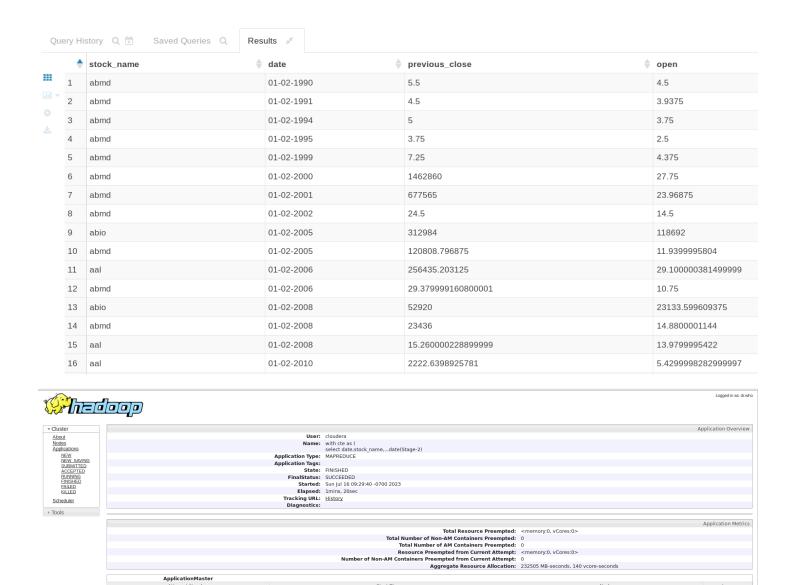
- 3 from stocksoptm
- 4 order by date;

Checking Data from External Table:

```
hive> select * from hive5 limit 15;
0K
abmd
        01-02-1988
                         1828.5714285714
abmd
        01-02-1989
                         746,666666667
abmd
        01-02-1990
                         1688.888888889
abmd
        01-02-1991
                         11700
abmd
        01-02-1993
                         8076.1904761905
abmd
        01-02-1994
                         533.333333333
abmd
        01-02-1995
                         1777.77777778
                         1324.1379310345
abmd
        01-02-1996
abmd
        01-02-1999
                         16000
abio
        01-02-1999
                         0.0000047667
abio
        01-02-2000
                         0.0000060125
abmd
        01-02-2000
                         4149.6774193548
abio
        01-02-2001
                         0.0000014353
        01-02-2001
abmd
                         20101.5228426396
        01-02-2002
                         7576.1587489865
abmd
Time taken: 0.1 seconds, Fetched: 15 row(s)
```

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.

```
1 with cte as (
2 select date, stock_name, open, close, lag(close) over (order by date) as previous_close from stocksoptm)
3 select stock_name, date, previous_close, open
4 from cte
5 where (previous_close - open)>0
6 order by stock_name;
```



Creating external table:

```
hive> create external table hive6(stock_name string,date string,previous_close decimal(20,10),open decimal (20,10)) row format delimited fields terminated by ',' location '/user/hive/warehouse/stocksdb.db/stocksexternal/hive6';
OK
Time taken: 0.179 seconds
```

Inserting data into the External table:

```
with cte as (
select date, stock_name, open, close, lag(close) over (order by date) as previous_close from stocksoptm
insert into default.hive6
select stock_name, date, previous_close, open
from cte
where (previous_close - open)>0
order by stock_name;
```

Checking Data from External Table:

```
hive> select * from hive6 limit 15;
0K
aal
        11-06-2009
                         2903.0400390625 2.6900000572
aal
        06-06-2017
                         73.4100036621
                                         49.4700012207
aal
        06-06-2016
                         101.8899993896
                                         30.7700004578
aal
        12-04-2017
                         96.1800003052
                                         44.7900009155
aal
        25-01-2006
                         232394.40625
                                         31.1499996185
aal
        20-11-2009
                         11.3599996567
                                         3.0999999046
                        8467.2001953125 3.7400000095
aal
        11-06-2008
aal
        06-06-2012
                         362.8800048828
                                         11.8199996948
aal
        06-06-2011
                         1443.9599609375 8.9099998474
aal
        20-11-2008
                         20109.599609375 4.8299999237
        06-06-2008
aal
                         17.9300003052
                                         4.1399998665
aal
        22-12-2009
                        8.5500001907
                                         4.6199998856
aal
        29-09-2017
                         64.6699981689
                                         47.4900016785
aal
        06-06-2006
                         250387.203125
                                         45.0200004578
aal
        29-09-2016
                         128.4199981689
                                         35.6500015259
Time taken: 0.129 seconds, Fetched: 15 row(s)
```

7. Find median of volume for ABIO.

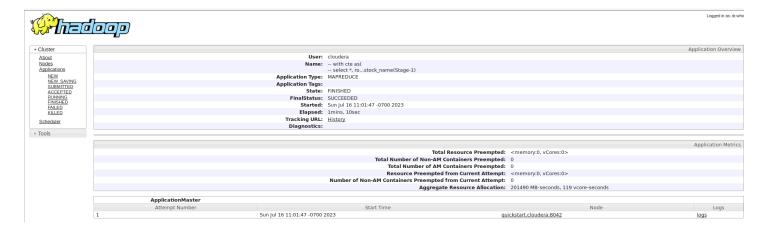
```
SELECT stock_name,
percentile_approx(volume, 0.5) AS median_volume

FROM stocksoptm
where stock_name='abio'
group by stock_name;

Query History Q Saved Queries Q Results 
stock_name

stock_name

abio
61.1875
```



Creating External Table:

```
hive> create external table hive7(stock_name string, median_value decimal(20,10)) row format delimited fields terminated by ','
location '/user/hive/warehouse/stocksdb.db/stocksexternal/hive7';
OK
Time taken: 0.174 seconds
```

Inserting data into the External table:

```
insert into default.hive7
select stock_name,
percentile_approx(volume,0.5) as median_volume
from stocksoptm
where stock_name = 'abio'
group by stock_name;
```

Checking Data from External Table:

```
hive> select * from hive7;
OK
abio 61.1875
Time taken: 0.144 seconds, Fetched: 1 row(s)
```

Exporting external tables to MySQL:

1. Creating Tables in MySQL

```
mysql> create table hive1(stock_name varchar(100),date varchar(100), percentage_cha
nge Decimal (20,10),rank int);
Query OK, 0 rows affected (0.02 sec)

mysql> create table hive2(stock_name varchar(100),date varchar(100),low decimal (20
,10));
Query OK, 0 rows affected (0.01 sec)

mysql> create table hive3(stock_name varchar(100),date varchar(100),streak int);
Query OK, 0 rows affected (0.01 sec)

mysql> create table hive4(date varchar(100));
Query OK, 0 rows affected (0.00 sec)

mysql> create table hive5(stock_name varchar(100),date varchar(100),vh_ratio decima
l(20,10));
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> create table hive6(stock_name varchar(100),date varchar(100),previous_close
decimal(20,10),open decimal (20,10));
Query OK, 0 rows affected (0.01 sec)

mysql> create table hive7(stock_name varchar(100), median_value decimal(20,10));
Query OK, 0 rows affected (0.01 sec)
```

2. Exporting Hive Tables to MySQL using sqoop

a. Q1

```
[cloudera@quickstart Desktop]$ sqoop export --connect jdbc:mysql://localhost:3306/stocksdb --use rname root --password cloudera --table hivel --export-dir /user/hive/warehouse/stocksdb.db/stock sexternal/hive1/000000_0 --input-fields-terminated-by ','
```

b. Q2

[cloudera@quickstart Desktop]\$ sqoop export --connect jdbc:mysql://localhost:3306/stocksdb --use rname root --password cloudera --table hive2 --export-dir /user/hive/warehouse/stocksdb.db/stock sexternal/hive2/000000 0 --input-fields-terminated-by ','

c. Q3

[cloudera@quickstart Desktop]\$ sqoop export --connect jdbc:mysql://localhost:3306/stocksdb --use rname root --password cloudera --table hive3 --export-dir /user/hive/warehouse/stocksdb.db/stock sexternal/hive3/000000_0 --input-fields-terminated-by ','

d. Q4

[cloudera@quickstart Desktop]\$ sqoop export --connect jdbc:mysql://localhost:3306/stocksdb --use rname root --password cloudera --table hive4 --export-dir /user/hive/warehouse/stocksdb.db/stock sexternal/hive4/000000_0 --input-fields-terminated-by ','

e. Q5

[cloudera@quickstart Desktop]\$ sqoop export --connect jdbc:mysql://localhost:3306/stocksdb --use
rname root --password cloudera --table hive5 --export-dir /user/hive/warehouse/stocksdb.db/stock
sexternal/hive5/000000_0 --input-fields-terminated-by ','

f. Q6

[cloudera@quickstart Desktop]\$ sqoop export --connect jdbc:mysql://localhost:3306/stocksdb --use rname root --password cloudera --table hive6 --export-dir /user/hive/warehouse/stocksdb.db/stock sexternal/hive6/0000000_0 --input-fields-terminated-by ','

g. Q7

[cloudera@quickstart Desktop]\$ sqoop export --connect jdbc:mysql://localhost:3306/stocksdb --use rname root --password cloudera --table hive7 --export-dir /user/hive/warehouse/stocksdb.db/stock sexternal/hive7/000000 0 --input-fields-terminated-by ','

EXPORTING DATA FROM MYSQL TO LOCAL SYSTEM:

```
mysql> SELECT * INTO OUTFILE '/home/cloudera/Downloads/walmart_case_study_data/resu
lts/hiver1.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERM
INATED BY '\n' FROM hive1;
Query OK, 12 rows affected (0.00 sec)
mysql> SELECT * INTO OUTFILE '/home/cloudera/Downloads/walmart case study data/resu
lts/hiver2.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERM
INATED BY '\n' FROM hive2;
Query OK, 15401 rows affected (0.02 sec)
mysql> SELECT * INTO OUTFILE '/home/cloudera/Downloads/walmart case study data/resu
lts/hiver3.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERM
INATED BY '\n' FROM hive3;
Ouery OK, 4 rows affected (0.00 sec)
mysql> SELECT * INTO OUTFILE '/home/cloudera/Downloads/walmart case study data/resu
lts/hiver4.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERM
INATED BY '\n' FROM hive4;
Query OK, 2320 rows affected (0.00 sec)
```

```
mysql> SELECT * INTO OUTFILE '/home/cloudera/Downloads/walmart_case_study_data/results/hiver5.cs
v' FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERMINATED BY '\n' FROM hive5;

Query OK, 21948 rows affected (0.03 sec)

mysql> SELECT * INTO OUTFILE '/home/cloudera/Downloads/walmart_case_study_data/results/hiver6.cs
v' FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERMINATED BY '\n' FROM hive6;

Query OK, 11101 rows affected (0.02 sec)

mysql> SELECT * INTO OUTFILE '/home/cloudera/Downloads/walmart_case_study_data/results/hiver7.cs
v' FIELDS TERMINATED BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERMINATED BY '\n' FROM hive7;

Query OK, 1 row affected (0.00 sec)
```