

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),volume int,high decimal (20,10),close decimal (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),volume int,high decimal (20,10),close decimal (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),volume int,high decimal (20,10),close decimal (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),volume int,high decimal (20,10),close decimal (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

```
mysql> Load data infile '/home/cloudera/Downloads/walmart_case_study_data/AAOI.csv' into table aaoi fields terminated by ',' ignore 1 lines;
Query OK, 2320 rows affected, 10805 warnings (0.03 sec)
Records: 2320 Deleted: 0 Skipped: 0 Warnings: 10805

mysql> select * from aaoi limit 5;
+-----+-----+-----+-----+-----+-----+-----+
| date       | low      | open      | volume  | high     | close     | adjusted_close |
+-----+-----+-----+-----+-----+-----+-----+
| 26-09-2013 | 9.3699998856 | 10.0000000000 | 946000 | 10.0900001526 | 9.9600000381 | 9.9600000381 |
| 27-09-2013 | 10.0000000000 | 10.4399995804 | 253300 | 10.4399995804 | 10.1000003815 | 10.1000003815 |
| 30-09-2013 | 9.7100000381 | 10.0000000000 | 84900  | 10.1800003052 | 10.0000000000 | 10.0000000000 |
| 01-10-2013 | 9.9200000763 | 9.9499998093 | 74500  | 10.0200004578 | 10.0000000000 | 10.0000000000 |
| 02-10-2013 | 9.8900003433 | 9.9899997711 | 94000  | 10.0000000000 | 9.9700002670 | 9.9700002670 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> Load data infile '/home/cloudera/Downloads/walmart_case_study_data/ABIO.csv' into table abio fields terminated by ',' ignore 1 lines;
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 ',' ignore 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 ',' ignore 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: 0  Warnings: 0
```

Cross-checking the data if it has come right:


```
mysql> select * from stocksdata limit 5;
+-----+-----+-----+-----+-----+-----+-----+-----+
| date      | low      | open      | volume  | high      | close      | adjusted_close | stock_name |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 27-09-2005 | 19.1000003815 | 21.04999992371 | 961200 | 21.39999996185 | 19.29999992371 | 18.1949100494 | aal        |
| 28-09-2005 | 19.2000007629 | 19.29999992371 | 5747900 | 20.5300006866 | 20.50000000000 | 19.3262042999 | aal        |
| 29-09-2005 | 20.1000003815 | 20.39999996185 | 1078200 | 20.5799999237 | 20.20999990845 | 19.0528049469 | aal        |
| 30-09-2005 | 20.1800003052 | 20.2600002289 | 3123300 | 21.04999992371 | 21.0100002289 | 19.8070011139 | aal        |
| 03-10-2005 | 20.89999996185 | 20.89999996185 | 1057900 | 21.75000000000 | 21.50000000000 | 20.2689399719 | aal        |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 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 --user name root --password cloudera --hive-import --hive-database stocksdb -m 1
```

CHECKING THE JOB STATUS:



Cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
3	0	0	3	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

User Metrics for dr.who

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Containers Pending	Containers Reserved	Memory Used	Memory Pending	Memory Reserved	VCores Used	VCores Pending	VCores Reserved
0	0	0	0	0	0	0	0 B	0 B	0 B	0	0	0

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Progress	Tracking UI
application_1689315550722_0003	cloudera	stocksdata.jar	MAPREDUCE	root.cloudera	Fri Jul 14 02:42:22 -0700	Fri Jul 14 02:43:07 -0700	FINISHED	SUCCEEDED	N/A	N/A	N/A		History



Cluster
About
Nodes
Applications
NEW
NEW SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools

Application Overview	
User:	cloudera
Name:	stocksdata.jar
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Fri Jul 14 02:42:22 -0700 2023
Elapsed:	45sec
Tracking URL:	History
Diagnostics:	

Application Metrics	
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>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	133326 MB-seconds, 78 vcore-seconds

ApplicationMaster			
Attempt Number	Start Time	Node	Logs
1	Fri Jul 14 02:42:22 -0700 2023	quickstart.cloudera:8042	logs

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;
OK
stocksdata
Time taken: 0.026 seconds, Fetched: 1 row(s)
hive> desc stocksdata;
OK
date                string
low                  double
open                 double
volume               int
high                 double
close                double
adjusted_close       double
stock_name           string
Time taken: 0.229 seconds, Fetched: 8 row(s)
```

```
hive> select * from stockdata limit 5;
OK
27-09-2005      19.1000003815    21.0499992371    961200    21.3999996185    19.29999923
71      18.1949100494    aal
28-09-2005      19.2000007629    19.2999992371    5747900    20.5300006866    20.5    19.
3262042999      aal
29-09-2005      20.1000003815    20.3999996185    1078200    20.5799999237    20.20999908
45      19.0528049469    aal
30-09-2005      20.1800003052    20.2600002289    3123300    21.0499992371    21.01000022
89      19.8070011139    aal
03-10-2005      20.8999996185    20.8999996185    1057900    21.75    21.5    20.26893997
19      aal
Time taken: 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, volume 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 volume 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_close,stock_name from stocksdata;
```

Checking if the data has come and got partitioned

```
hive> select * from stocksoptm limit 5;
OK
02-06-2006      45.25    46.2599983215    1219200 46.8100013733    45.8499984741    43.2246932983    aal
30-05-2006      43.8499984741    46.0      1009700 46.1199989319    44.6100006104    42.0556907654    aal
14-10-2020      12.2899999619    12.3500003815    37500000    12.5699996948    12.3599996567    12.3599996567    aal
23-05-2006      45.0900001526    47.0600013733    1062500 47.25    45.1899986267    42.602481842    aal
16-05-2006      47.5299987793    49.5      755300    49.5999984741    47.8499984741    45.1101760864    aal
Time taken: 0.068 seconds, Fetched: 5 row(s)
hive> show partitions stocksoptm;
OK
stock_name=aal
stock_name=aaol
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.

```
hive> with cte as
> (select stock_name,date,(open-close)*100/open as percentage_change from stocksoptm)
> select * from (
>     select stock_name,date,percentage_change,
>     rank() over(partition by stock_name order by percentage_change desc) as rankk
>     from cte) a
>     where rankk in (1,2,3);
```

Total MapReduce CPU Time Spent: 5 seconds 570 msec

OK

aal	12-11-2008	30.266666411999996	1
aal	23-06-2008	21.604939677392927	2
aal	21-05-2008	20.857992256746392	3
aaoi	20-09-2022	14.516127893773854	1
aaoi	05-11-2021	13.874188434203338	2
aaoi	04-08-2017	13.866668701199993	3
abio	20-02-2019	43.33333411796293	1
abio	26-02-2018	39.99999717407408	2
abio	14-03-2000	35.38461538461539	3
abmd	19-10-1987	24.63768115942029	1
abmd	08-01-2001	21.428571428571427	2
abmd	09-10-2002	21.404685714600493	3

Time taken: 33.764 seconds, Fetched: 12 row(s)



Logged in as: dr:who

Cluster
About
Nodes
Applications
NEW
NEW SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler

Tools

User: cloudera
Name: with cte as
(select stock_name,date...(1,2,3)(Stage-1)
Application Type: MAPREDUCE
Application Tags:
State: FINISHED
FinalStatus: SUCCEEDED
Started: Sun Jul 16 04:53:34 -0700 2023
Elapsed: 30sec
Tracking URL: History
Diagnostics:

Application Overview

Application Metrics
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>
Number of Non-AM Containers Preempted from Current Attempt: 0
Aggregate Resource Allocation: 94221 MB-seconds, 54 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Jul 16 04:53:34 -0700 2023	quickstart.cloudera:8042	logs

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/stocksd.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.hive1
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;
OK
aal      12-11-2008      30.266666412      1
aal      23-06-2008      21.6049396774     2
aal      21-05-2008      20.8579922567     3
aaoi     20-09-2022      14.5161278938     1
aaoi     05-11-2021      13.8741884342     2
aaoi     04-08-2017      13.8666687012     3
abio     20-02-2019      43.333334118      1
abio     26-02-2018      39.9999971741     2
abio     14-03-2000      35.3846153846     3
abmd     19-10-1987      24.6376811594     1
abmd     08-01-2001      21.4285714286     2
abmd     09-10-2002      21.4046857146     3
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.

```

1 select stock_name,date,low,avg_low
2 from
3     (select stock_name,date,low,high,
4         avg(low) over (partition by stock_name,year(date),month(date)) as avg_low from stocksoptm) sub
5 where low<avg_low order by date;

```

Query History Saved Queries Results

	stock_name	date	low	avg_low
1	abmd	01-02-1988	4.1875	9.3961666742999981
2	abmd	01-02-1989	5.5	9.3961666742999981
3	abmd	01-02-1990	4.5	9.3961666742999981
4	abmd	01-02-1991	3.8125	9.3961666742999981
5	abmd	01-02-1993	5	9.3961666742999981
6	abmd	01-02-1994	3.75	9.3961666742999981
7	abmd	01-02-1995	2.5	9.3961666742999981
8	abmd	01-02-1996	6.75	9.3961666742999981
9	abio	01-02-1999	198450	414725.40011160716
10	abmd	01-02-1999	4.125	9.3961666742999981



Logged in as: dr:who

- Cluster
- About
- Nodes
- Applications
 - NEW
 - NEW SAVING
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler
- Tools

Application Overview	
User:	cloudera
Name:	select stock_name,date,low,avg_low fr...date(Stage-2)
Application Type:	MAPREDUCE
Application Tags:	
State:	FINISHED
FinalStatus:	SUCCEEDED
Started:	Sun Jul 16 05:17:33 -0700 2023
Elapsed:	29sec
Tracking URL:	History
Diagnostics:	

Application Metrics	
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>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	90497 MB-seconds, 51 vcore-seconds

ApplicationMaster	Attempt Number	Start Time	Node	Logs
1		Sun Jul 16 05:17:33 -0700 2023	quickstart.cloudera:8042	logs

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/stocksd.db/stocksexternal/hive2';
OK
Time taken: 0.369 seconds

```

Inserting data into the External table:

```

1 insert into table default.hive2
2 select stock_name, date, low
3 from
4     (select stock_name,date,low,high,
5         avg(low) over (partition by substr(date, 3,9) order by cast(substr(date, 6, 9) as int))
6         as avg_low from stocksoptm
7     ) s
8 where low<avg_low;
9
10
11

```

Checking Data from External Table:


```
hive> select * from hive2 limit 10;
OK
abmd      01-02-1988      4.1875
abmd      01-02-1989      5.5
abmd      01-02-1990      4.5
abmd      01-02-1991      3.8125
abmd      01-02-1993      5
abmd      01-02-1994      3.75
abmd      01-02-1995      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 cte1 AS(
2 SELECT stock_name, date, close,
3        LEAD(close) OVER(PARTITION BY stock_name ORDER BY date) AS next,
4        ROW_NUMBER() OVER(PARTITION BY stock_name ORDER BY date) AS rn
5 FROM stocksoptm
6 ), cte2 AS(
7 SELECT *,
8        rn - ROW_NUMBER() OVER(PARTITION BY stock_name ORDER BY date) AS rnk
9 FROM cte1
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,
18        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
23 )
24 SELECT STOCK_NAME, DATE, STREAK
25 FROM cte4 WHERE rn=1;
```

Query History Saved Queries Results			
	stock_name	date	streak
1	abio	14-08-2019	4
2	aaai	01-09-2022	4
3	aal	01-10-2009	6
4	abmd	02-12-2009	10

Show 20 entries											Search:
ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	
application_1689582433728_0026	cloudera	WITH cte1 AS(SELECT stock_name, dat...rn=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	N
application_1689582433728_0025	cloudera	WITH cte1 AS(SELECT stock_name, dat...rn=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	N
application_1689582433728_0024	cloudera	WITH cte1 AS(SELECT stock_name, dat...rn=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	N
application_1689582433728_0023	cloudera	WITH cte1 AS(SELECT stock_name, dat...rn=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	N
application_1689582433728_0022	cloudera	WITH cte1 AS(SELECT stock_name, dat...rn=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	N
application_1689582433728_0021	cloudera	WITH cte1 AS(SELECT stock_name, dat...rn=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	N

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:

```

1 WITH cte1 AS(
2 SELECT stock_name, date, close,
3        LEAD(close) OVER(PARTITION BY stock_name ORDER BY date) AS next,
4        ROW_NUMBER() OVER(PARTITION BY stock_name ORDER BY date) AS rn
5 FROM stocksoptm
6 ), cte2 AS(
7 SELECT *,
8        rn - ROW_NUMBER() OVER(PARTITION BY stock_name ORDER BY date) AS rnk
9 FROM cte1
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,
18        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
23 )
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

```

	a.date
1	01-02-2016
2	01-02-2017
3	01-02-2018
4	01-02-2019
5	01-02-2021
6	01-02-2022
7	01-03-2016
8	01-03-2017
9	01-03-2018
10	01-03-2019
11	01-03-2021
12	01-03-2022
13	01-04-2014
14	01-04-2015
15	01-04-2016
16	01-04-2019



Cluster

[About](#)
[Nodes](#)
[Applications](#)
[NEW](#)
[NEW SAVING](#)
[SUBMITTED](#)
[ACCEPTED](#)
[RUNNING](#)
[FINISHED](#)
[FAILED](#)
[KILLED](#)
[Scheduler](#)

Tools

Application Overview

User: cloudera

Name: select a date from stocksptm as a j--date(Stage-2)

Application Type: MAPREDUCE

Application Tags:

State: FINISHED

FinalStatus: SUCCEEDED

Started: Sun Jul 16 08:27:56 -0700 2023

Elapsed: 1mins, 17sec

Tracking URL: [History](#)

Diagnostics:

Application Metrics

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

Number of Non-AM Containers Preempted from Current Attempt: 0

Aggregate Resource Allocation: 221995 MB-seconds, 133 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Jul 16 08:27:56 -0700 2023	quickstart.cloudera-8042	logs

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:

```

1 insert into table default.hive4
2 select a.date from stocksoptm as a
3 join stocksoptm as b on a.date=b.date join stocksoptm as c on b.date=c.date
4 where a.stock_name='aal' and b. stock_name='aaoi'
5     and c.stock_name='abmd'
6     and (a.open>b.open or a.volume>c.volume)
7 order by date;
8
9 |
10

```

Checking Data from External Table:

```

hive> select * from hive4 limit 15;
OK
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;


```

Query History

Saved Queries

Results

	stock_name	date	vh_ratio
1	abmd	01-02-1988	1828.5714285714287
2	abmd	01-02-1989	746.66666666666663
3	abmd	01-02-1990	1688.8888888888889
4	abmd	01-02-1991	11700
5	abmd	01-02-1993	8076.1904761904761
6	abmd	01-02-1994	533.33333333333337
7	abmd	01-02-1995	1777.7777777777778
8	abmd	01-02-1996	1324.1379310344828
9	abmd	01-02-1999	16000
10	abio	01-02-1999	4.7666714333380996e-06
11	abio	01-02-2000	6.0125060125060122e-06
12	abmd	01-02-2000	4149.677419354839
13	abio	01-02-2001	1.4353354468647788e-06
14	abmd	01-02-2001	20101.522842639595
15	abmd	01-02-2002	7576.1587489864532
16	abio	01-02-2002	3.0619243582206545e-06


Logged in as: dr:who

Cluster

- About
- Nodes
- Applications
 - NEW
 - NEW SAVING
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler

Tools

User: cloudera

Name: select date,(volume/high) as VH_Ratio,...date(Stage-1)

Application Type: MAPREDUCE

Application Tags:

State: FINISHED

FinalStatus: SUCCEEDED

Started: Sun Jul 16 08:33:30 -0700 2023

Elapsed: 1mins, 26sec

Tracking URL: History

Diagnostics:

Application Overview

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>

Number of Non-AM Containers Preempted from Current Attempt: 0

Aggregate Resource Allocation: 249154 MB-seconds, 150 vcore-seconds

Application Metrics

ApplicationMaster

Attempt Number

Start Time

Node

Logs

1

Sun Jul 16 08:33:30 -0700 2023

quickstart.cloudera:8042

logs

Creating External table:

```
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/stocksd.db/stocksexternal/hive5';
OK
Time taken: 0.135 seconds
```

Inserting data into the External table:

```
1 insert into default.hive5
2 select stock_name, date, (volume/high) as VH_Ratio
3 from stocksoptm
4 order by date;
```

Checking Data from External Table:

```
hive> select * from hive5 limit 15;
OK
abmd      01-02-1988      1828.5714285714
abmd      01-02-1989      746.6666666667
abmd      01-02-1990      1688.8888888889
abmd      01-02-1991      11700
abmd      01-02-1993      8076.1904761905
abmd      01-02-1994      533.3333333333
abmd      01-02-1995      1777.7777777778
abmd      01-02-1996      1324.1379310345
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
abmd      01-02-2001      20101.5228426396
abmd      01-02-2002      7576.1587489865
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 ;
```

Query History Saved Queries Results				
	stock_name	date	previous_close	open
1	abmd	01-02-1990	5.5	4.5
2	abmd	01-02-1991	4.5	3.9375
3	abmd	01-02-1994	5	3.75
4	abmd	01-02-1995	3.75	2.5
5	abmd	01-02-1999	7.25	4.375
6	abmd	01-02-2000	1462860	27.75
7	abmd	01-02-2001	677565	23.96875
8	abmd	01-02-2002	24.5	14.5
9	abio	01-02-2005	312984	118692
10	abmd	01-02-2005	120808.796875	11.9399995804
11	aal	01-02-2006	256435.203125	29.100000381499999
12	abmd	01-02-2006	29.379999160800001	10.75
13	abio	01-02-2008	52920	23133.599609375
14	abmd	01-02-2008	23436	14.8800001144
15	aal	01-02-2008	15.260000228899999	13.9799995422
16	aal	01-02-2010	2222.6398925781	5.4299998282999997



Cluster

- About
- Nodes
- Applications
 - NEW
 - NEW SAVING
 - SUBMITTED
 - ACCEPTED
 - PENDING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler

Tools

User: cloudera

Name: with cte as (select date,stock_name,...date(Stage-2))

Application Type: MAPREDUCE

Application Tags:

State: FINISHED

FinalStatus: SUCCEEDED

Started: Sun Jul 16 09:29:40 -0700 2023

Elapsed: 1mins, 20sec

Tracking URL: History

Diagnostics:

Application Overview

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>

Number of Non-AM Containers Preempted from Current Attempt: 0

Aggregate Resource Allocation: 232505 MB-seconds, 140 vcore-seconds

Application Metrics

ApplicationMaster

Attempt Number

Start Time

Node

Logs

1		Sun Jul 16 09:29:40 -0700 2023	quickstart.cloudera:8042	logs
---	--	--------------------------------	--------------------------	------

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:







```
1
2 with cte as (
3 select date,stock_name,open,close,lag(close) over (order by date) as previous_close from stocksoptm
4 insert into default.hive6
5 select stock_name,date,previous_close,open
6 from cte
7 where (previous_close - open)>0
8 order by stock_name;
9
```


Checking Data from External Table:


```
hive> select * from hive6 limit 15;
OK
aal      11-06-2009      2903.0400390625  2.69000000572
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
aal      11-06-2008      8467.2001953125  3.7400000095
aal      06-06-2012      362.8800048828  11.8199996948
aal      06-06-2011      1443.9599609375  8.9099998474
aal      20-11-2008      20109.599609375  4.8299999237
aal      06-06-2008      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.

```
5
6 SELECT stock_name,
7        percentile_approx(volume, 0.5) AS median_volume
8 FROM stocksoptm
9 where stock_name='abio'
10 group by stock_name;
11
```

Query History  		Saved Queries 		Results 
 stock_name				 median_volume
1	abio			61.1875


Logged in as: dr:who

- Cluster
- About
- Nodes
- Applications
 - NEW
 - NEW SAVING
 - SUBMITTED
 - ACCEPTED
 - BURNING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler
- Tools

Application Overview

User: cloudera

Name: -- with cte as(
-- select *, ro...stock_name(Stage-1)

Application Type: MAPREDUCE

Application Tags:

State: FINISHED

FinalStatus: SUCCEEDED

Started: Sun Jul 16 11:01:47 -0700 2023

Elapsed: 1mins, 10sec

Tracking URL: History

Diagnostics:

Application Metrics

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>

Number of Non-AM Containers Preempted from Current Attempt: 0

Aggregate Resource Allocation: 201490 MB-seconds, 119 vcore-seconds

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sun Jul 16 11:01:47 -0700 2023	quickstart.cloudera:8042	logs

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:

```
1 insert into default.hive7
2 select stock_name,
3     percentile_approx(volume,0.5) as median_volume
4 From stocksoptm
5 where stock_name ='abio'
6 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_change 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 decimal(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 hive1 --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/hive6/000000_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;
Query 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)
```