Stock Exchange Data Analysis

Project 1

DESCRIPTION

Objective: To use hive features for data engineering or analysis and sharing the actionable insights

Problem Statement:
NewYork stock exchange data of seven years, between 2010 to 2016, is captured for 500+ listed companies. The data set comprises of intra-day prices and volume traded for each listed company. The data serves both for machine learning and exploratory analysis projects, to automate the trading process and to predict the next trading-day winners or losers.. The scope of this project is limited to exploratory data analysis.

Domain: BFSI

Analysis to be done: Exploratory analysis to understand how MoM or YoY companies from different sectors or industries and states have progressed in a period of 7 years

Content: This data set contains prices.csv and securities.csv files having the following features:

Prices.csv:

1. Date: Trading date
2. Symbol: Ticker code or listed company code on NY stock exchange
3. Open: Intra-day opening price for each listed company
4. Close: Intra-day closing price for each listed company
5. Low: Intra-day lowest price for each listed company
6. High: Intra-day highest price for each listed company
7. Volume: Number of shares traded per day per company

Securities.csv:

1. Ticker_Symbol: Country to which the customer belongs
2. Security: Legal name of the listed company
3. Sector: Business vertical of the listed company
4. Sub_Industry: Business domain of the listed company within a Sector.
5. Headquarter: Headquarters address

Steps to perform:

1) Create a data pipeline using sqoop to pull the data from the table below from MYSQL server into Hive.
a. MYSQL DATABASE NAME: BDHS_PROJECT
i. Stock_prices ii. Stock_companies

2) Create a new hive table with the following fields by joining the above two hive tables.
Please use appropriate Hive built-in functions for columns (a,b,e and h to l).

- Trading_year: Should contain YYYY for each record
- Trading_month: Should contain MM or MMM for each record
- Symbol: Ticker code
- CompanyName: Legal name of the listed company
- State: State to be extracted from headquarters value.
- Sector: Business vertical of the listed company
- Sub_Industry: Business domain of the listed company within a sector
- Open: Average of intra-day opening price by month and year for each listed company
- Close: Average of intra-day closing price by month and year for each listed company
- Low: Average of intra-day lowest price by month and year for each listed company
- High: Average of intra-day highest price by month and year for each listed company
- Volume: Average of number of shares traded by month and year for each listed company

DATA ANALYSIS USING HIVE

3) Find the top five companies that are good for investment
4) Show the best-growing industry by each state, having at least two or more industries mapped.
5) For each sector find the following.

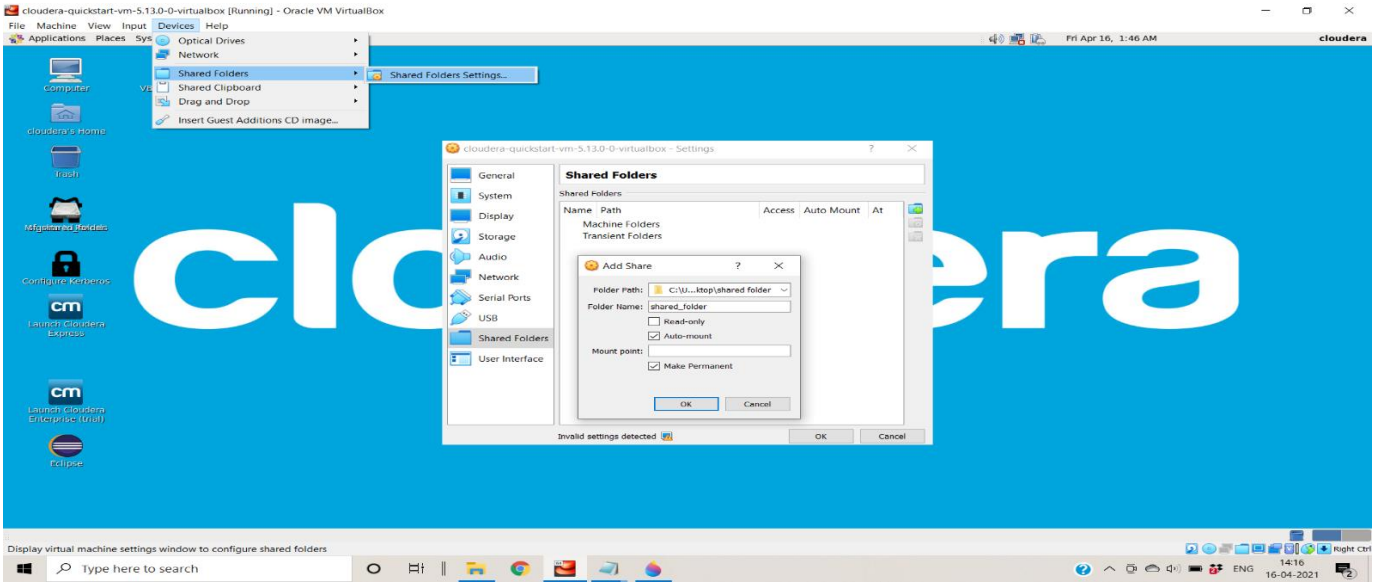
- Worst year
- b. Best year
- c. Stable year

==========

1) Create a data pipeline using sqoop to pull the data from the table below from MYSQL server into Hive.
a. MYSQL DATABASE NAME: BDHS_PROJECT

==========

- To download the dataset and pasting it to share folder.
- open cloudera vm--
- menu => devices => shared folder => shared folder settings => add new folder to machine folders option by selecting shared folder and keeping mounted and parmanent.

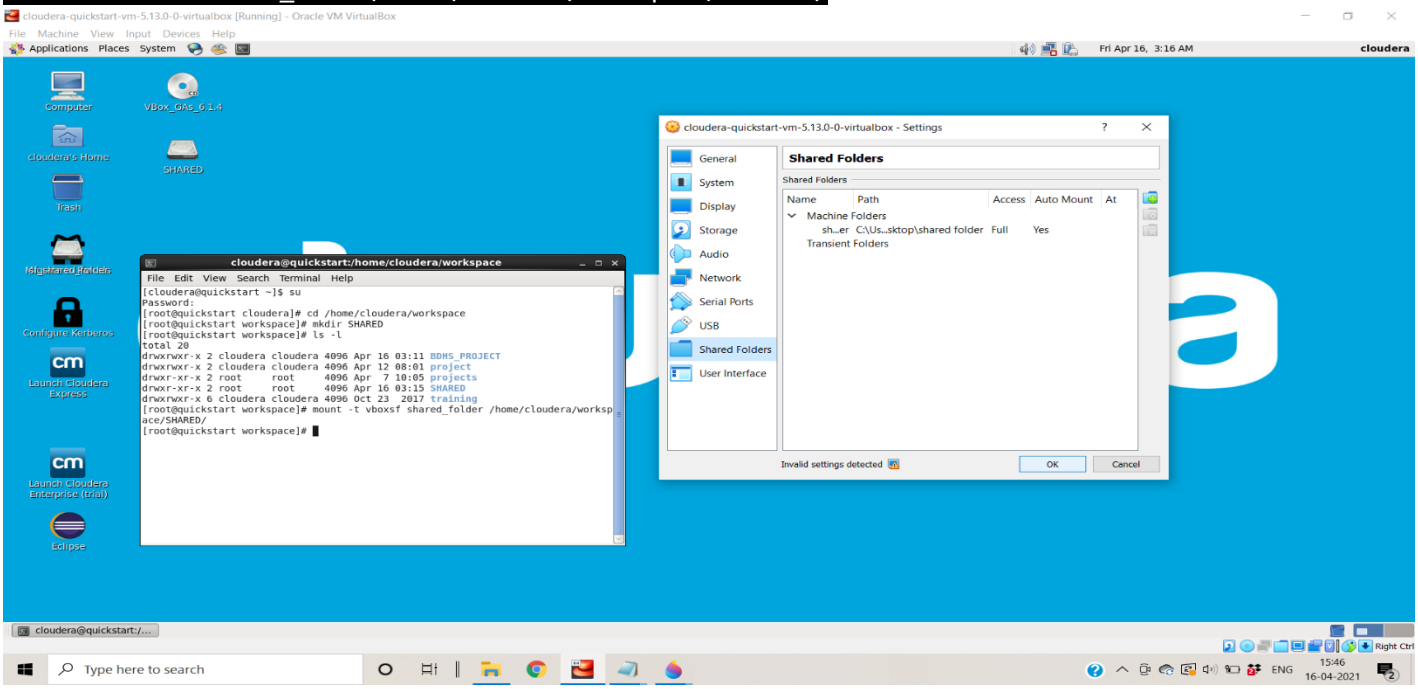


==========

cloudera @ quickstart terminal 1

==========

```
su
cloudera
cd /home/cloudera/workspace
 mkdir file
mount -t vboxsf shared_folder /home/cloudera/workspace/SHARED/
```



==========

cloudera @ quickstart terminal 2

==========

mysql -u root -p

cloudera

create database BDHS_PROJECT;

use BDHS_PROJECT;

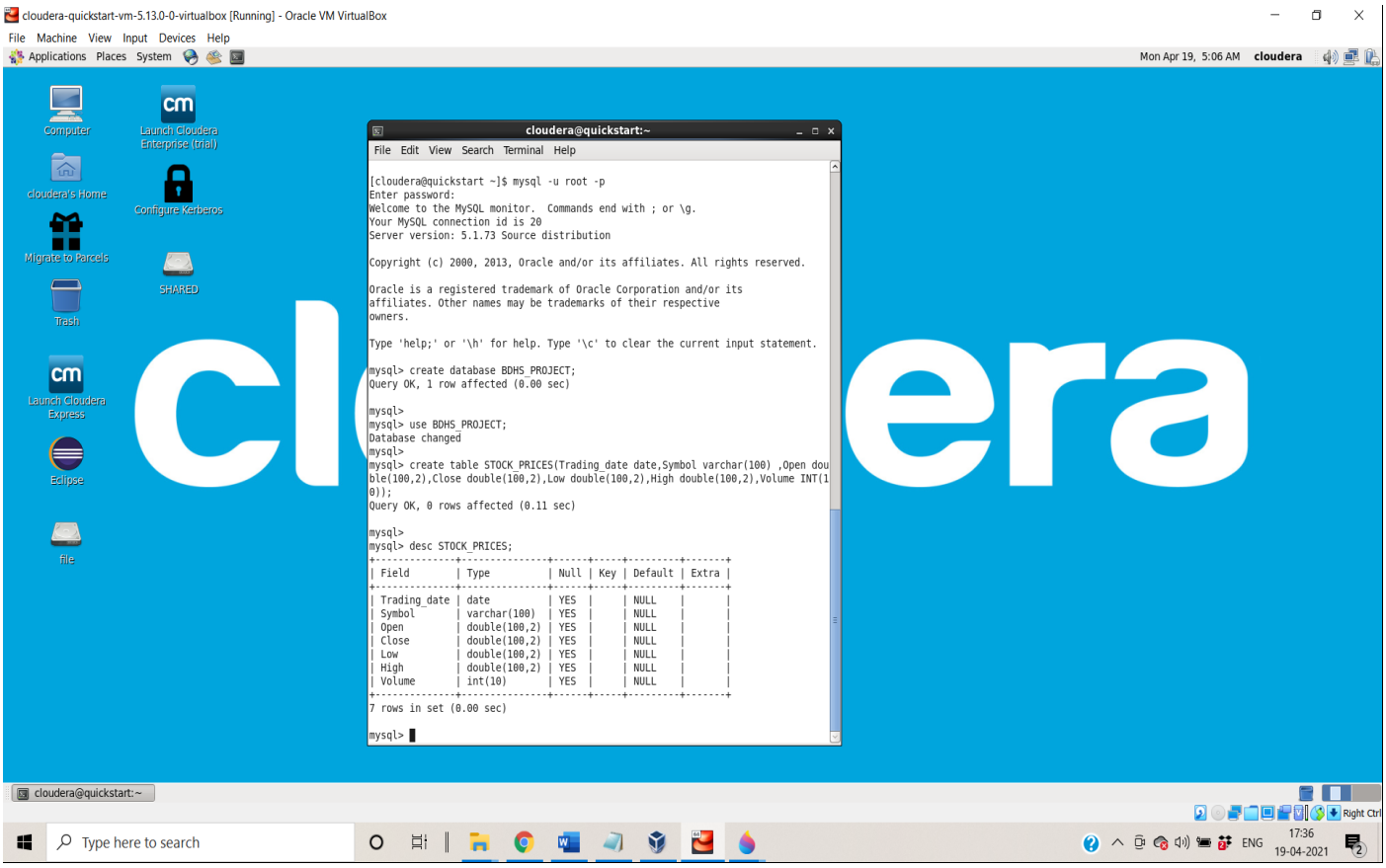
create table STOCK_PRICES(Trading_date date,Symbol varchar(100) ,Open double(100,2),Close double(100,2),Low double(100,2),High double(100,2),Volume INT(10));

create table STOCK_COMPANIES(Symbol varchar(100) ,Company_name varchar(100),Sector varchar(100),Sub_industry varchar(100),Headquarter varchar(100));

LOAD DATA INFILE '/home/cloudera/workspace/SHARED/Stockprices.csv' INTO TABLE STOCK_PRICES FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

LOAD DATA INFILE '/home/cloudera/workspace/SHARED/Stockcompanies.csv' INTO TABLE STOCK_COMPANIES FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;

desc STOCK_PRICES;



desc STOCK_COMPANIES;

select * from STOCK_PRICES limit 10;

```
mysql> create table STOCK_COMPANIES(Symbol varchar(100) ,Company_name varchar(100),Sector varchar(100),Sub_industry varchar(100),Headquarter varchar(100));
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> desc STOCK_COMPANIES;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| Symbol       | varchar(100) | YES  |     | NULL    |       |
| Company_name | varchar(100) | YES  |     | NULL    |       |
| Sector       | varchar(100) | YES  |     | NULL    |       |
| Sub_industry | varchar(100) | YES  |     | NULL    |       |
| Headquarter  | varchar(100) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> LOAD DATA INFILE '/home/cloudera/workspace/SHARED/Stockprices.csv' INTO TABLE STOCK_PRICES FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 851264 rows affected (5.66 sec)
Records: 851264  Deleted: 0  Skipped: 0  Warnings: 0

mysql> LOAD DATA INFILE '/home/cloudera/workspace/SHARED/Stockcompanies.csv' INTO TABLE STOCK_COMPANIES FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 LINES;
Query OK, 505 rows affected (0.09 sec)
Records: 505  Deleted: 0  Skipped: 0  Warnings: 0

mysql>
mysql> select * from STOCK_PRICES limit 10;
+--------------+--------+--------+--------+--------+--------+---------+
| Trading_date | Symbol | Open   | Close  | Low    | High   | Volume  |
+--------------+--------+--------+--------+--------+--------+---------+
| 2016-01-05   | WLTW   | 123.43 | 125.84 | 122.31 | 126.25 | 2163600 |
| 2016-01-06   | WLTW   | 125.24 | 119.98 | 119.94 | 125.54 | 2386400 |
| 2016-01-07   | WLTW   | 116.38 | 114.95 | 114.93 | 119.74 | 2489500 |
| 2016-01-08   | WLTW   | 115.48 | 116.62 | 113.50 | 117.44 | 2006300 |
| 2016-01-11   | WLTW   | 117.01 | 114.97 | 114.09 | 117.33 | 1408600 |
| 2016-01-12   | WLTW   | 115.51 | 115.55 | 114.50 | 116.06 | 1098000 |
| 2016-01-13   | WLTW   | 116.46 | 112.85 | 112.59 | 117.07 |  949600 |
| 2016-01-14   | WLTW   | 113.51 | 114.38 | 110.05 | 115.03 |  785300 |
| 2016-01-15   | WLTW   | 113.33 | 112.53 | 111.92 | 114.88 | 1093700 |
| 2016-01-19   | WLTW   | 113.66 | 110.38 | 109.87 | 115.87 | 1523500 |
+--------------+--------+--------+--------+--------+--------+---------+
10 rows in set (0.00 sec)

mysql>
```

select * from STOCK_COMPANIES limit 10;

```
mysql> select * from STOCK_COMPANIES limit 10;
+--------+--------------------+-----------------------+-----------------------------------------+-------------------------+
| Symbol | Company_name       | Sector                | Sub_industry                            | Headquarter             |
+--------+--------------------+-----------------------+-----------------------------------------+-------------------------+
      |  | 3M Company         | Industrials           | Industrial Conglomerates                | St. Paul; Minnesota
  |BT    | Abbott Laboratories | Health Care           | Health Care Equipment                   | North Chicago; Illinois
  |BBV   | AbbVie             | Health Care           | Pharmaceuticals                         | North Chicago; Illinois
        |Accenture plc       | Information Technology | IT Consulting & Other Services          | Dublin; Ireland
 |ATVI   | Activision Blizzard | Information Technology | Home Entertainment Software             | Santa Monica; California
       | Acuity Brands Inc  | Industrials           | Electrical Components & Equipment       | Atlanta; Georgia
     |   | Adobe Systems Inc  | Information Technology | Application Software                    | San Jose; California
       || Advance Auto Parts | Consumer Discretionary | Automotive Retail                       | Roanoke; Virginia
     |   | AES Corp           | Utilities             | Independent Power Producers & Energy Traders | Arlington; Virginia
   |     | Aetna Inc          | Health Care           | Managed Health Care                     | Hartford; Connecticut
+--------+--------------------+-----------------------+-----------------------------------------+-------------------------+
10 rows in set (0.00 sec)

mysql>
mysql>
mysql>
```

==========

cloudera @ quickstart terminal 3

==========

hive

show databases;

create database project;

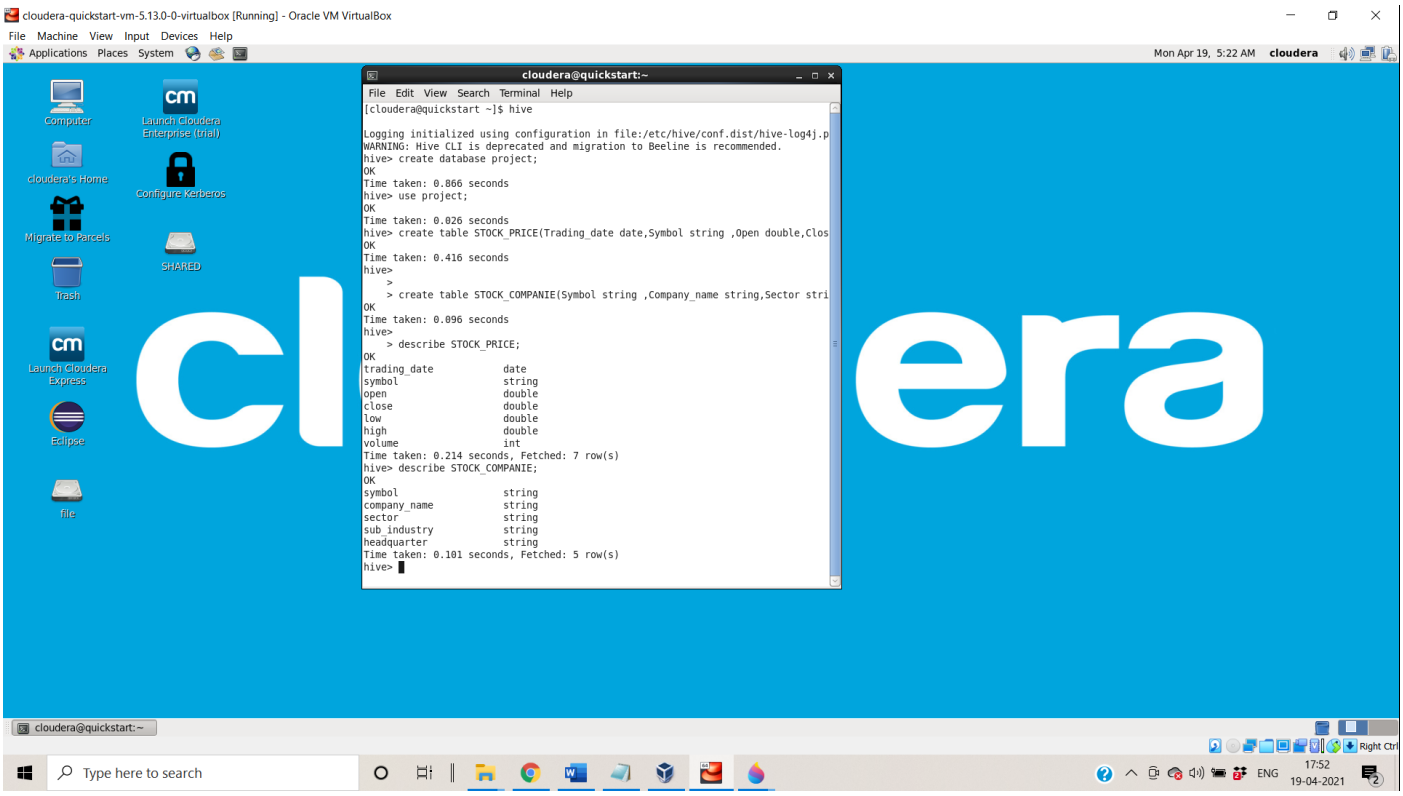
use project;

create table STOCK_PRICE(Trading_date date,Symbol string ,Open double,Close double,Low double,High double,Volume INT);

create table STOCK_COMPANIE(Symbol string ,Company_name string,Sector string,Sub_industry string,Headquarter string);

describe STOCK_PRICE;

describe STOCK_COMPANIE;

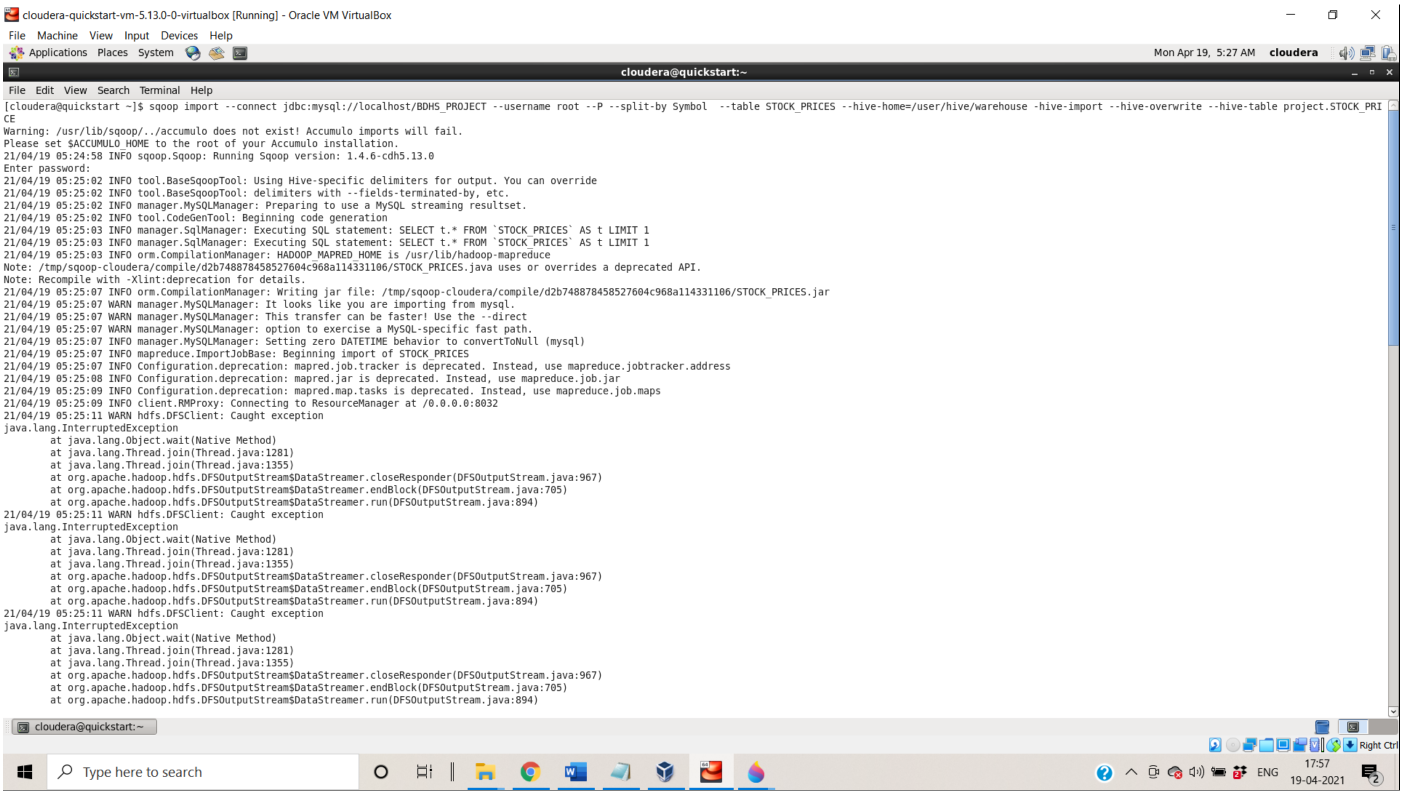


==

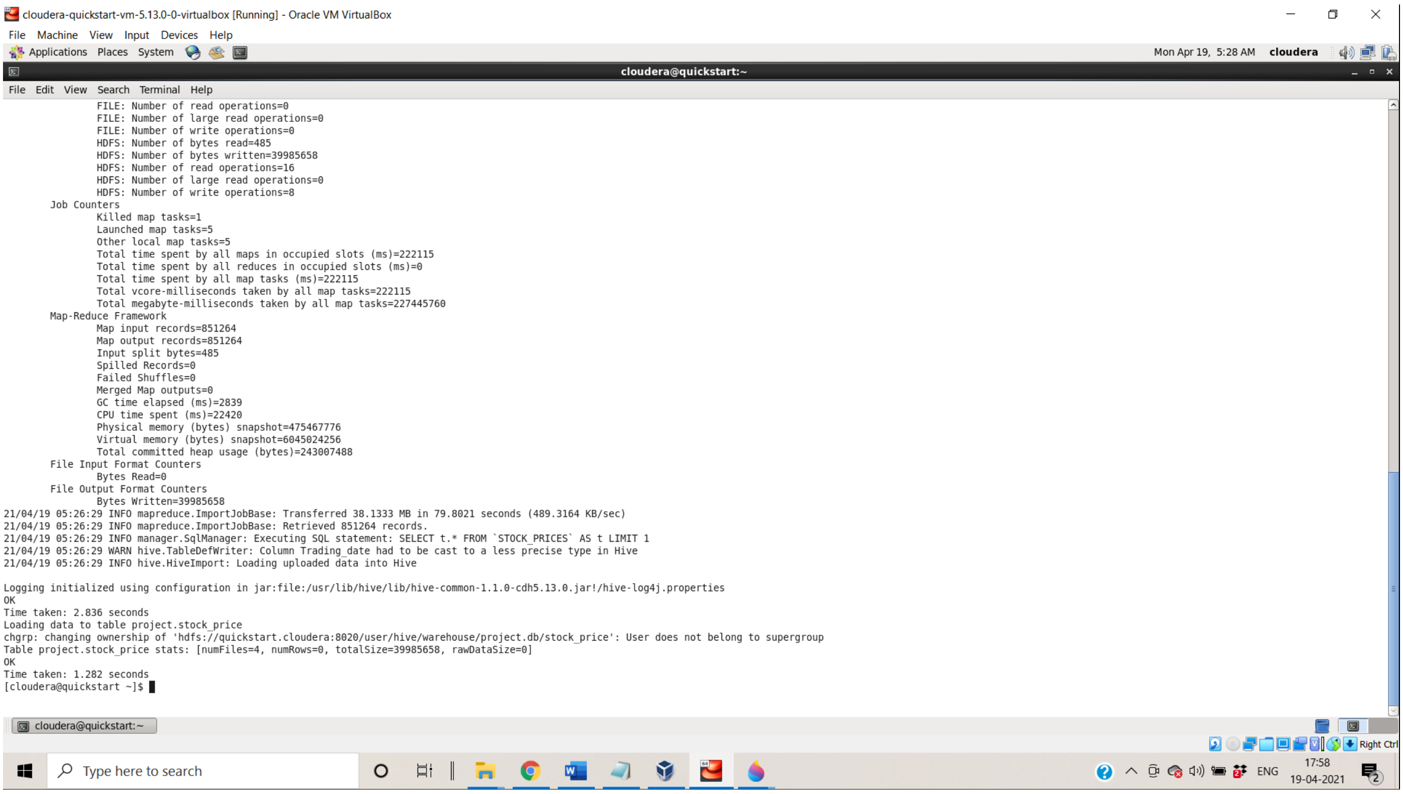
cloudera @ quickstart terminal 4

==

```
sqoop import --connect jdbc:mysql://localhost/BDHS_PROJECT --username root --P --split-by Symbol  --table STOCK_PRICES --hive-home=/user/hive/warehouse -hive-import --hive-overwrite --hive-table project.STOCK_PRICE
```

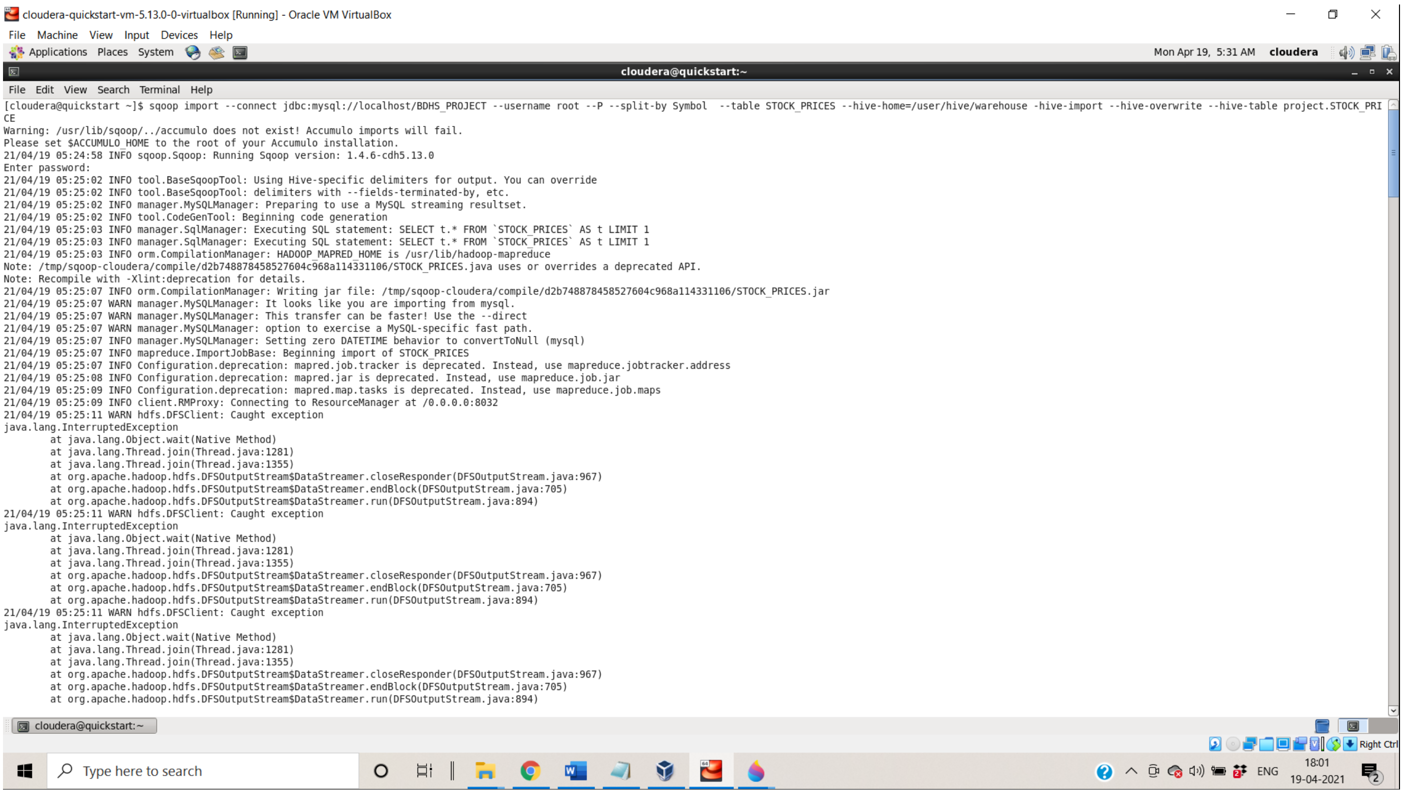


```
[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://localhost/BDHS_PROJECT --username root --P --split-by Symbol  --table STOCK_PRICES --hive-home=/user/hive/warehouse -hive-import --hive-overwrite --hive-table project.STOCK_PRI
CE
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/04/19 05:24:58 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
21/04/19 05:25:02 INFO tool.BaseSqoopTool: Using Hive-specific delimiters for output. You can override
21/04/19 05:25:02 INFO tool.BaseSqoopTool: delimiters with --fields-terminated-by, etc.
21/04/19 05:25:02 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
21/04/19 05:25:02 INFO tool.CodeGenTool: Beginning code generation
21/04/19 05:25:03 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `STOCK_PRICES` AS t LIMIT 1
21/04/19 05:25:03 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `STOCK_PRICES` AS t LIMIT 1
21/04/19 05:25:03 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/d2b748878458527604c968a114331106/STOCK_PRICES.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
21/04/19 05:25:07 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/d2b748878458527604c968a114331106/STOCK_PRICES.jar
21/04/19 05:25:07 WARN manager.MySQLManager: It looks like you are importing from mysql.
21/04/19 05:25:07 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
21/04/19 05:25:07 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
21/04/19 05:25:07 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
21/04/19 05:25:07 INFO mapreduce.ImportJobBase: Beginning import of STOCK_PRICES
21/04/19 05:25:07 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
21/04/19 05:25:08 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
21/04/19 05:25:09 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
21/04/19 05:25:09 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/04/19 05:25:11 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/04/19 05:25:11 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/04/19 05:25:11 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
```

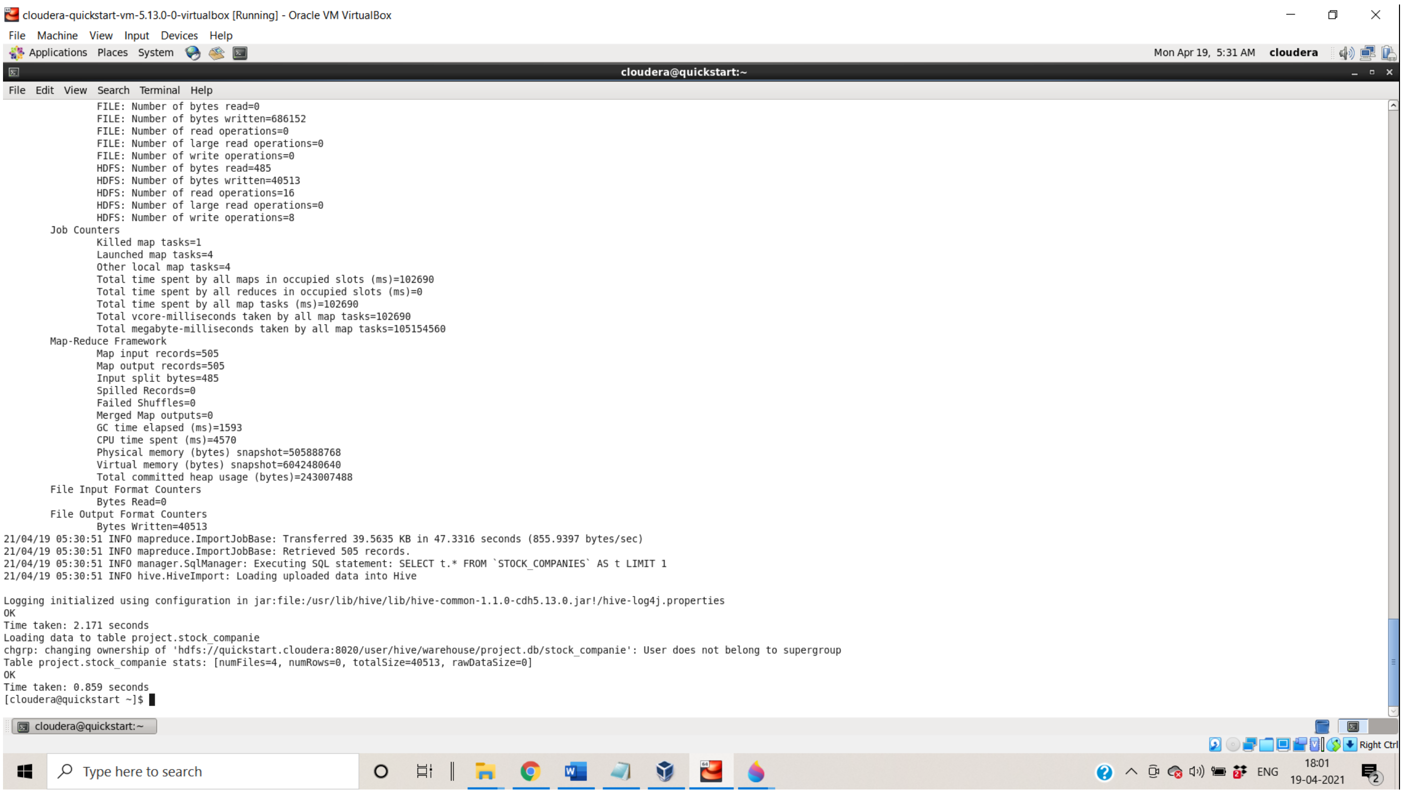


```
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=485
                HDFS: Number of bytes written=39985658
                HDFS: Number of read operations=16
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=8
        Job Counters
                Killed map tasks=1
                Launched map tasks=5
                Other local map tasks=5
                Total time spent by all maps in occupied slots (ms)=222115
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=222115
                Total vcore-milliseconds taken by all map tasks=222115
                Total megabyte-milliseconds taken by all map tasks=227445760
        Map-Reduce Framework
                Map input records=851264
                Map output records=851264
                Input split bytes=485
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=2839
                CPU time spent (ms)=22420
                Physical memory (bytes) snapshot=475467776
                Virtual memory (bytes) snapshot=6045024256
                Total committed heap usage (bytes)=243007488
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=39985658
21/04/19 05:26:29 INFO mapreduce.ImportJobBase: Transferred 38.1333 MB in 79.8021 seconds (489.3164 KB/sec)
21/04/19 05:26:29 INFO mapreduce.ImportJobBase: Retrieved 851264 records.
21/04/19 05:26:29 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `STOCK_PRICES` AS t LIMIT 1
21/04/19 05:26:29 WARN hive.TableDefWriter: Column Trading_date had to be cast to a less precise type in Hive
21/04/19 05:26:29 INFO hive.HiveImport: Loading uploaded data into Hive

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-1.1.0-cdh5.13.0.jar!/hive-log4j.properties
OK
Time taken: 2.836 seconds
Loading data to table project.stock_price
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/project.db/stock_price': User does not belong to supergroup
Table project.stock_price stats: [numFiles=4, numRows=0, totalSize=39985658, rawDataSize=0]
OK
Time taken: 1.282 seconds
[cloudera@quickstart ~]$ 
```

```
sqoop import --connect jdbc:mysql://localhost/BDHS_PROJECT --username root --P --split-by Symbol  --table STOCK_COMPANIES --hive-home=/user/hive/warehouse -hive-import --hive-overwrite --hive-table project.STOCK_COMPANIE
```



```
[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://localhost/BDHS_PROJECT --username root --P --split-by Symbol  --table STOCK_PRICES --hive-home=/user/hive/warehouse -hive-import --hive-overwrite --hive-table project.STOCK_PRI
CE
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
21/04/19 05:24:58 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
21/04/19 05:25:02 INFO tool.BaseSqoopTool: Using Hive-specific delimiters for output. You can override
21/04/19 05:25:02 INFO tool.BaseSqoopTool: delimiters with --fields-terminated-by, etc.
21/04/19 05:25:02 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
21/04/19 05:25:02 INFO tool.CodeGenTool: Beginning code generation
21/04/19 05:25:03 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `STOCK_PRICES` AS t LIMIT 1
21/04/19 05:25:03 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `STOCK_PRICES` AS t LIMIT 1
21/04/19 05:25:03 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/d2b748878458527604c968a114331106/STOCK_PRICES.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
21/04/19 05:25:07 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/d2b748878458527604c968a114331106/STOCK_PRICES.jar
21/04/19 05:25:07 WARN manager.MySQLManager: It looks like you are importing from mysql.
21/04/19 05:25:07 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
21/04/19 05:25:07 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
21/04/19 05:25:07 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
21/04/19 05:25:07 INFO mapreduce.ImportJobBase: Beginning import of STOCK_PRICES
21/04/19 05:25:07 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
21/04/19 05:25:08 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
21/04/19 05:25:09 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
21/04/19 05:25:09 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/04/19 05:25:11 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/04/19 05:25:11 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/04/19 05:25:11 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
```



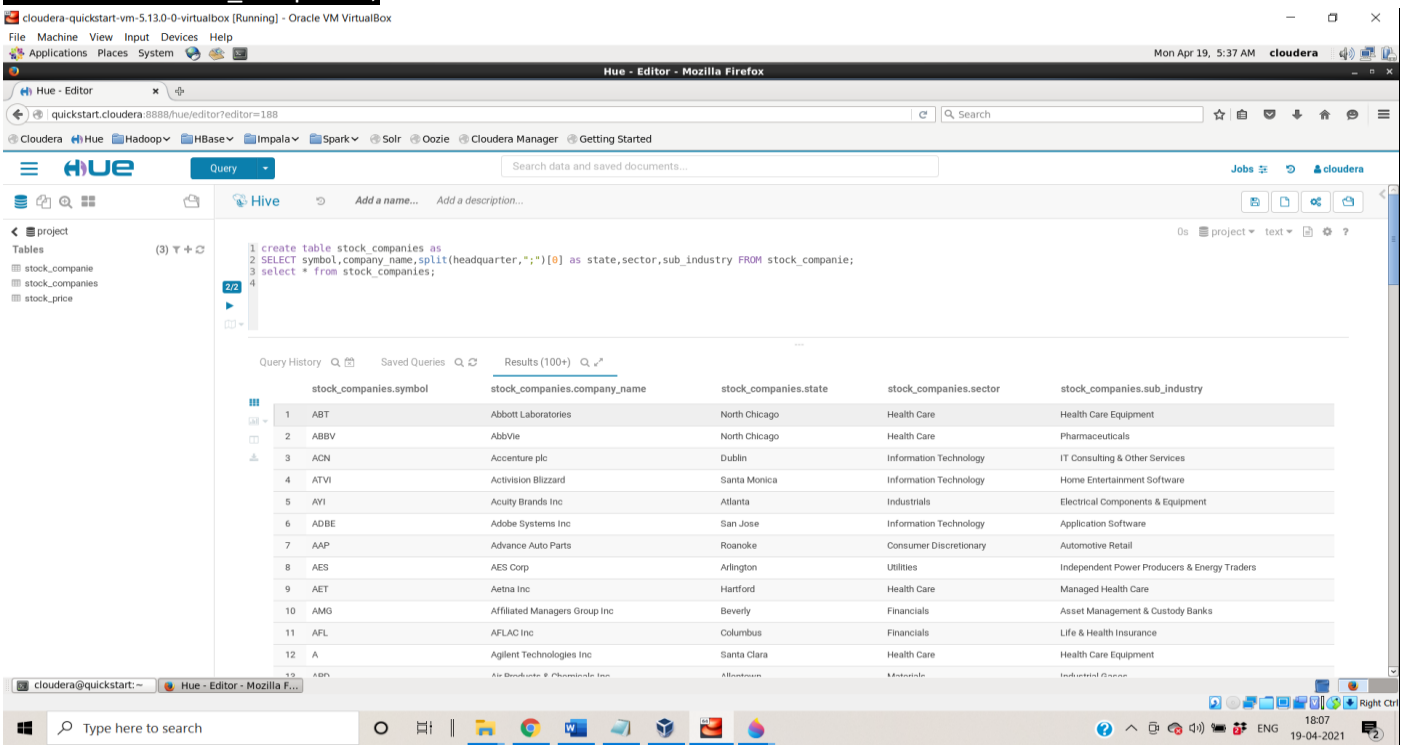
```
                FILE: Number of bytes read=0
                FILE: Number of bytes written=686152
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=485
                HDFS: Number of bytes written=40513
                HDFS: Number of read operations=16
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=8
        Job Counters
                Killed map tasks=1
                Launched map tasks=4
                Other local map tasks=4
                Total time spent by all maps in occupied slots (ms)=102690
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=102690
                Total vcore-milliseconds taken by all map tasks=102690
                Total megabyte-milliseconds taken by all map tasks=105154560
        Map-Reduce Framework
                Map input records=505
                Map output records=505
                Input split bytes=485
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=1593
                CPU time spent (ms)=4570
                Physical memory (bytes) snapshot=505888768
                Virtual memory (bytes) snapshot=6042480640
                Total committed heap usage (bytes)=243007488
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=40513
21/04/19 05:30:51 INFO mapreduce.ImportJobBase: Transferred 39.5635 KB in 47.3316 seconds (855.9397 bytes/sec)
21/04/19 05:30:51 INFO mapreduce.ImportJobBase: Retrieved 505 records.
21/04/19 05:30:51 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `STOCK_COMPANIES` AS t LIMIT 1
21/04/19 05:30:51 INFO hive.HiveImport: Loading uploaded data into Hive

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-1.1.0-cdh5.13.0.jar!/hive-log4j.properties
OK
Time taken: 2.171 seconds
Loading data to table project.stock_companie
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/project.db/stock_companie': User does not belong to supergroup
Table project.stock_companie stats: [numFiles=4, numRows=0, totalSize=40513, rawDataSize=0]
OK
Time taken: 0.859 seconds
[cloudera@quickstart ~]$ 
```

==

hue hive editor

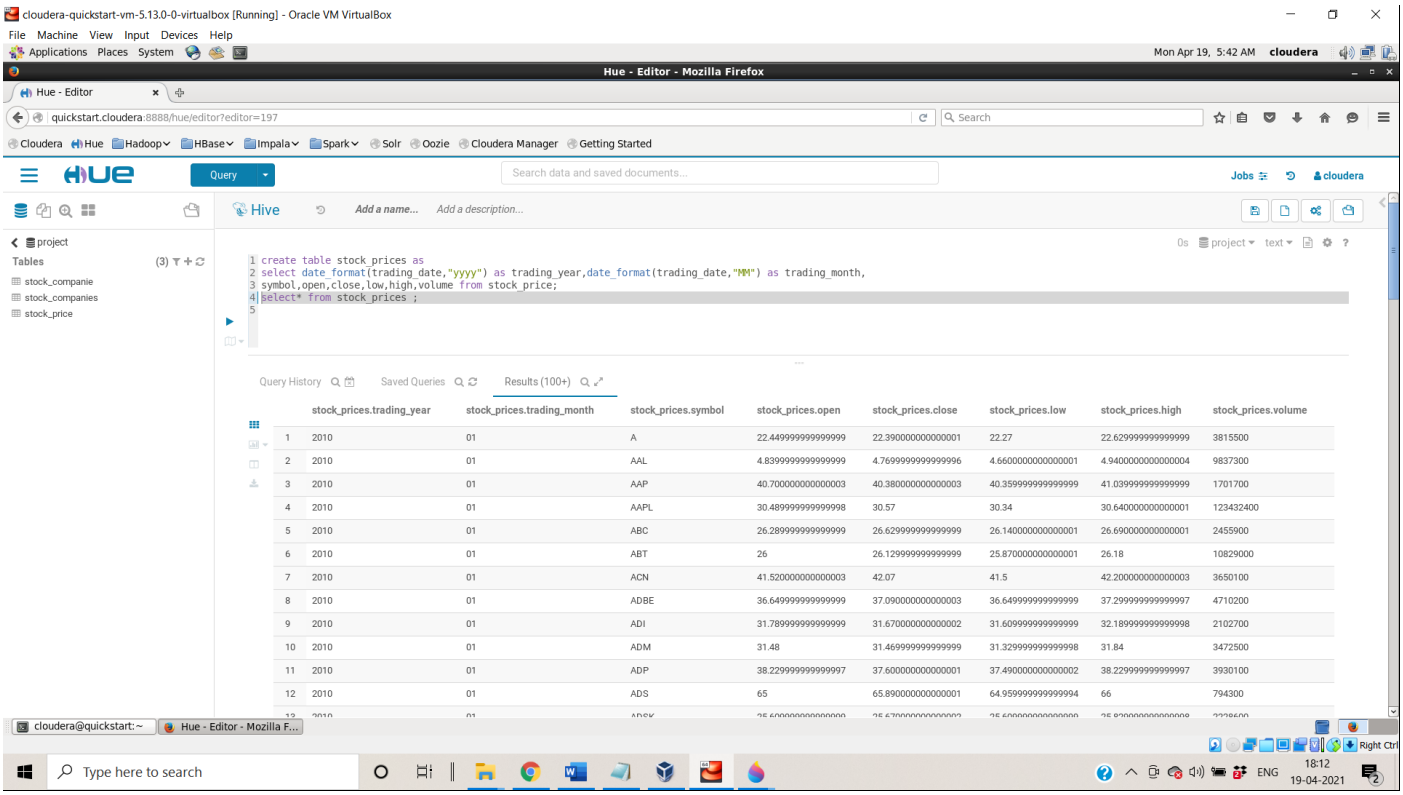
==

```
create table stock_companies as
SELECT symbol,company_name,split(headquarter,";")[0] as state,sector,sub_industry FROM stock_companie;
select * from stock_companies;
```



	stock_companies.symbol	stock_companies.company_name	stock_companies.state	stock_companies.sector	stock_companies.sub_industry
1	ABT	Abbott Laboratories	North Chicago	Health Care	Health Care Equipment
2	ABBV	AbbVie	North Chicago	Health Care	Pharmaceuticals
3	ACN	Accenture plc	Dublin	Information Technology	IT Consulting & Other Services
4	ATVI	Activision Blizzard	Santa Monica	Information Technology	Home Entertainment Software
5	AYI	Acuity Brands Inc	Atlanta	Industrials	Electrical Components & Equipment
6	ADBE	Adobe Systems Inc	San Jose	Information Technology	Application Software
7	AAP	Advance Auto Parts	Roanoke	Consumer Discretionary	Automotive Retail
8	AES	AES Corp	Arlington	Utilities	Independent Power Producers & Energy Traders
9	AET	Aetna Inc	Hartford	Health Care	Managed Health Care
10	AMG	Affiliated Managers Group Inc	Beverly	Financials	Asset Management & Custody Banks
11	AFL	AFLAC Inc	Columbus	Financials	Life & Health Insurance
12	A	Agilent Technologies Inc	Santa Clara	Health Care	Health Care Equipment

```
create table stock_prices as
select date_format(trading_date,"yyyy") as trading_year, date_format(trading_date,"MM") as trading_month,
symbol,open,close,low,high,volume from stock_price;

select* from stock_prices ;
```



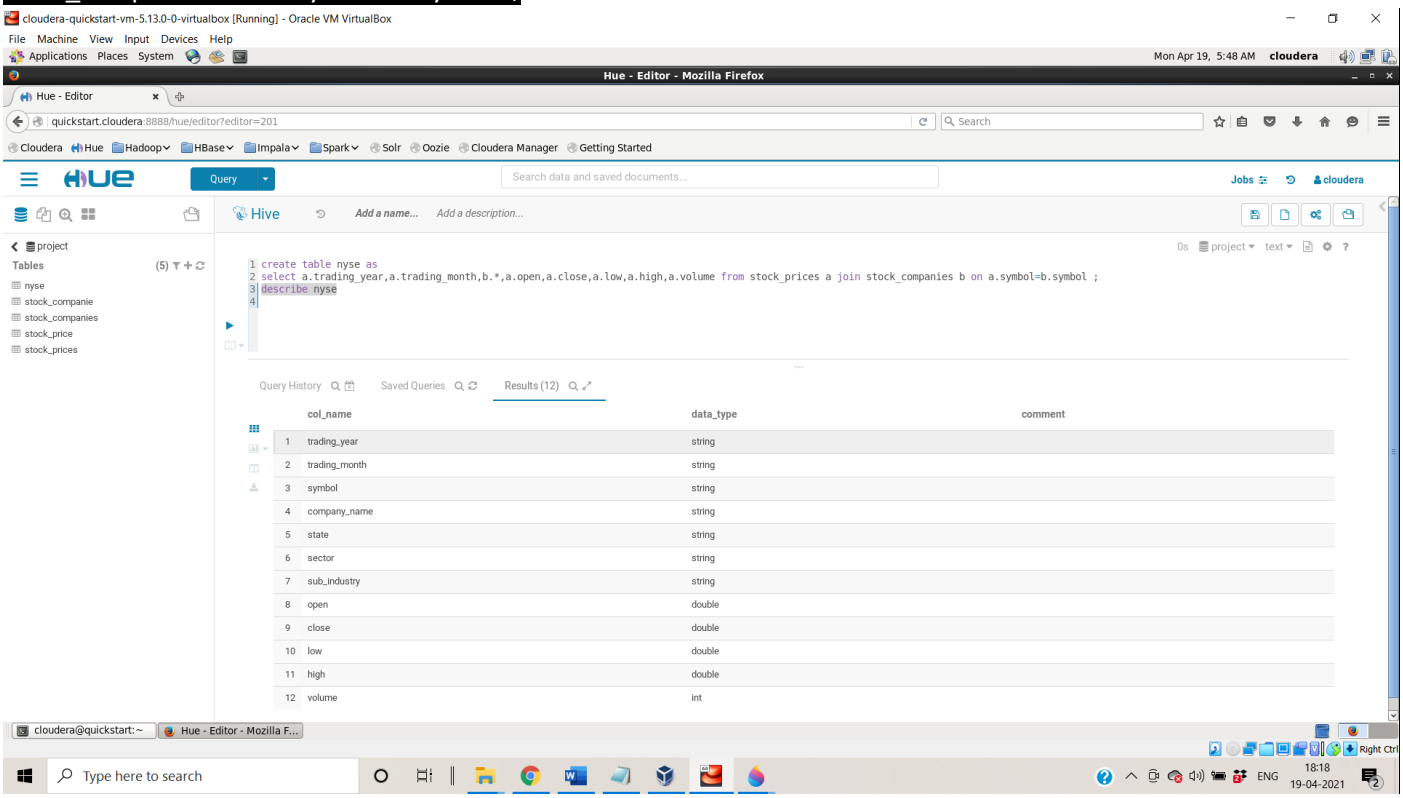
	stock_prices.trading_year	stock_prices.trading_month	stock_prices.symbol	stock_prices.open	stock_prices.close	stock_prices.low	stock_prices.high	stock_prices.volume
1	2010	01	A	22.449999999999999	22.390000000000001	22.27	22.629999999999999	3815500
2	2010	01	AAL	4.839999999999999	4.769999999999996	4.660000000000001	4.940000000000004	9837300
3	2010	01	AAP	40.700000000000003	40.380000000000003	40.359999999999999	41.039999999999999	1701700
4	2010	01	AAPL	30.489999999999998	30.57	30.34	30.640000000000001	123432400
5	2010	01	ABC	26.289999999999999	26.629999999999999	26.140000000000001	26.690000000000001	2455900
6	2010	01	ABT	26	26.129999999999999	25.870000000000001	26.18	10829000
7	2010	01	ACN	41.520000000000003	42.07	41.5	42.200000000000003	3650100
8	2010	01	ADBE	36.649999999999999	37.090000000000003	36.649999999999999	37.299999999999997	4710200
9	2010	01	ADI	31.789999999999999	31.670000000000002	31.609999999999999	32.189999999999998	2102700
10	2010	01	ADM	31.48	31.469999999999999	31.329999999999998	31.84	3472500
11	2010	01	ADP	38.229999999999997	37.600000000000001	37.490000000000002	38.229999999999997	3930100
12	2010	01	ADS	65	65.890000000000001	64.959999999999994	66	794300

==========

2) Create a new hive table with the following fields by joining the above two hive tables.Please use appropriate Hive built-in functions for columns (a,b,e and h to l).

==========

create table nyse as
select a.trading_year,a.trading_month,b.*,a.open,a.close,a.low,a.high,a.volume from stock_prices a join stock_companies b on a.symbol=b.symbol ;



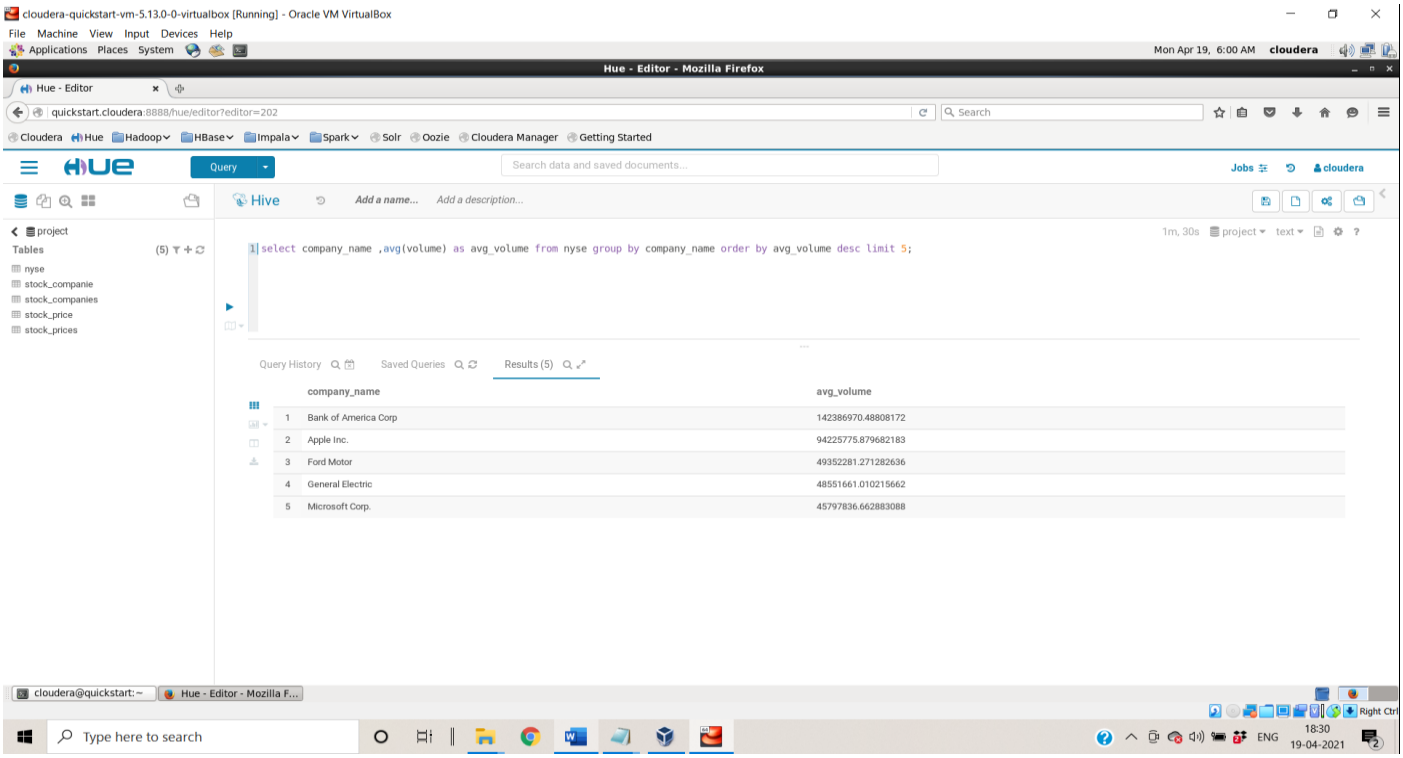
==========

#########DATA ANALYSIS USING HIVE############

3) Find the top five companies that are good for investment

==========

select company_name ,avg(volume) as avg_volume from nyse group by company_name order by avg_volume desc limit 5;



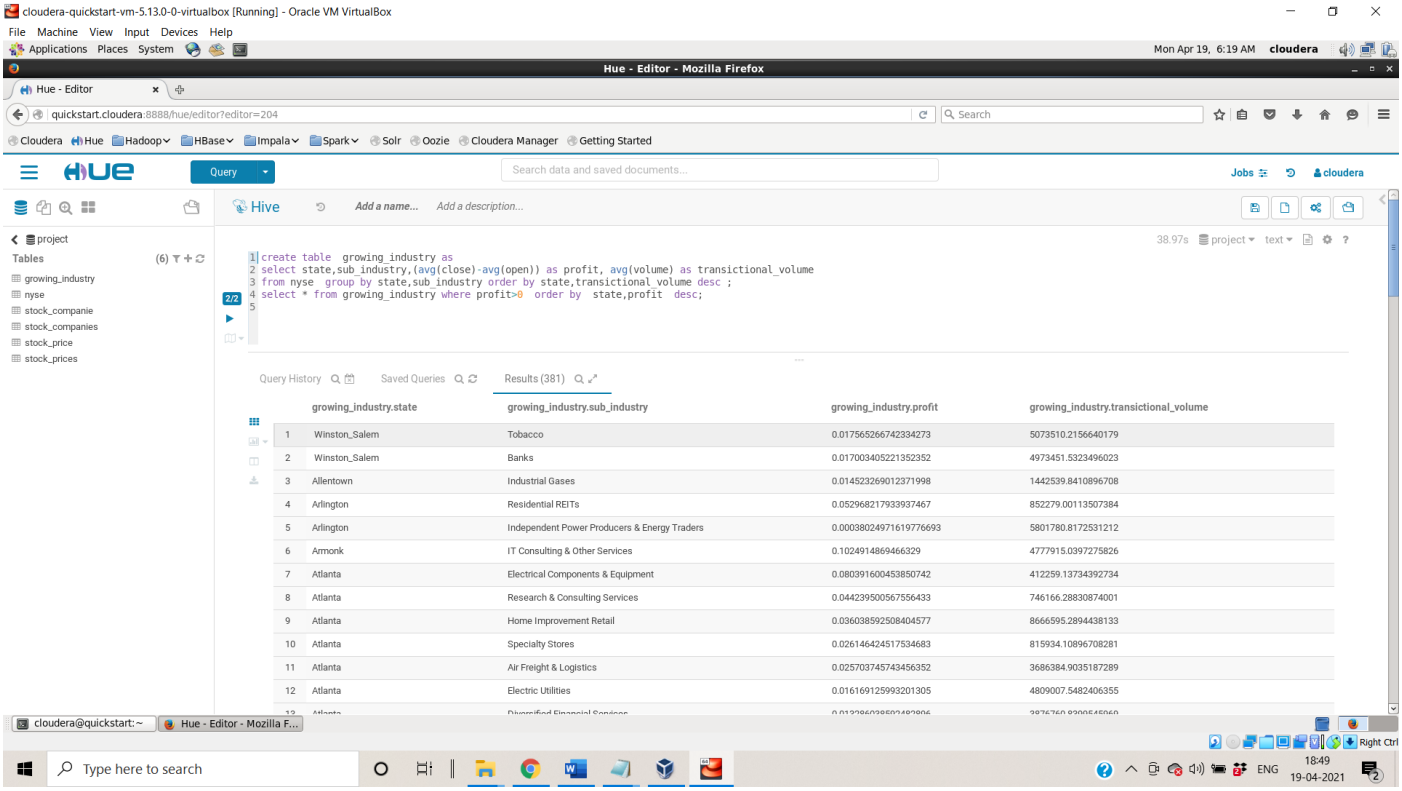
==========

#########DATA ANALYSIS USING HIVE############

4) Show the best-growing industry by each state, having at least two or more industries mapped.

==========

```
create table  growing_industry as
select state,sub_industry,(avg(close)-avg(open)) as profit, avg(volume) as transictional_volume
from nyse  group by state,sub_industry order by state,transictional_volume desc ;

select * from growing_industry where profit>0  order by  state,profit  desc;
```



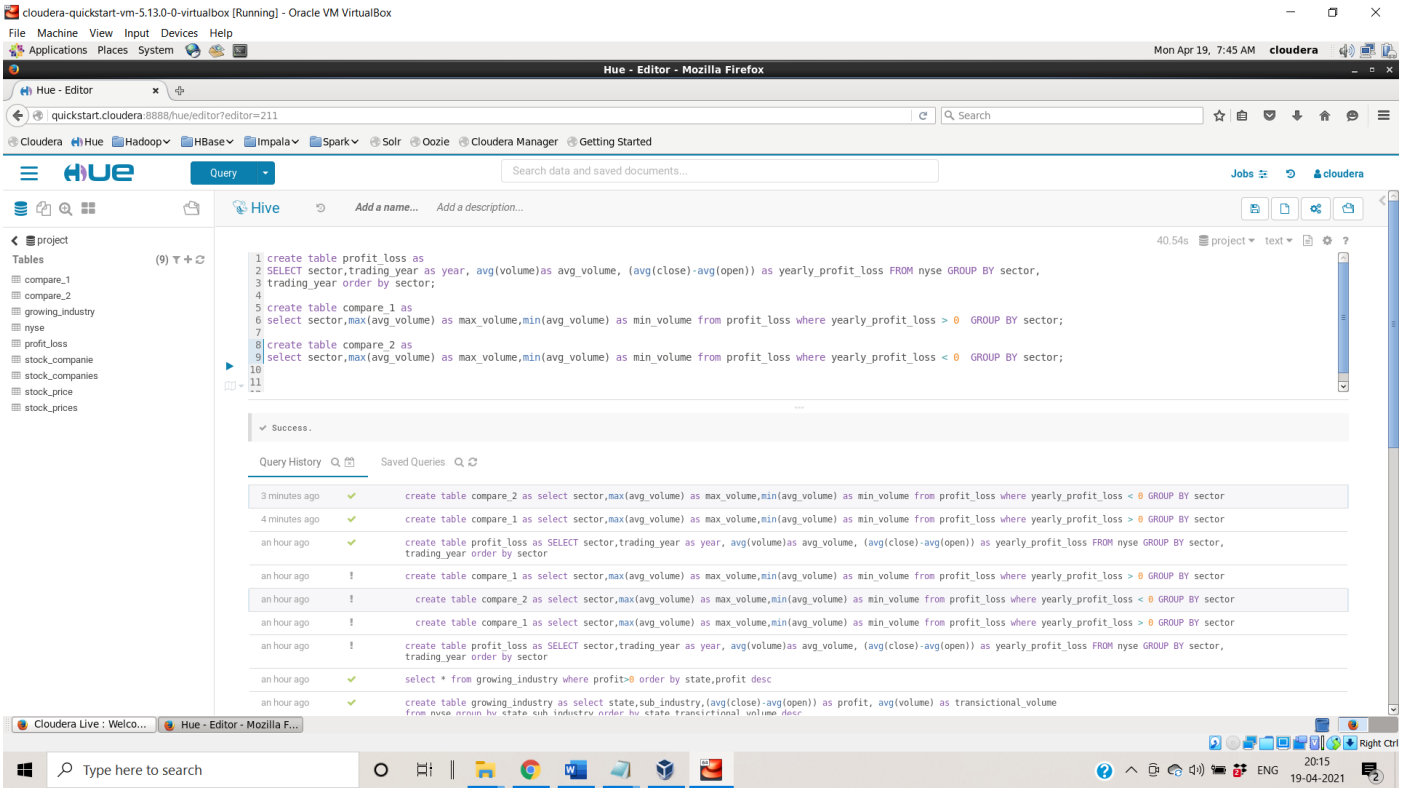
==========

#########DATA ANALYSIS USING HIVE############

5) For each sector find the following.

==========

```
create table profit_loss as
SELECT sector,trading_year as year, avg(volume)as avg_volume, (avg(close)-avg(open)) as yearly_profit_loss FROM
nyse GROUP BY sector,
trading_year order by sector;


create table compare_1 as;
select sector,max(avg_volume) as max_volume,min(avg_volume) as min_volume from profit_loss where
yearly_profit_loss > 0  GROUP BY sector;


create table compare_2 as;
select sector,max(avg_volume) as max_volume,min(avg_volume) as min_volume from profit_loss where
yearly_profit_loss < 0  GROUP BY sector;
```

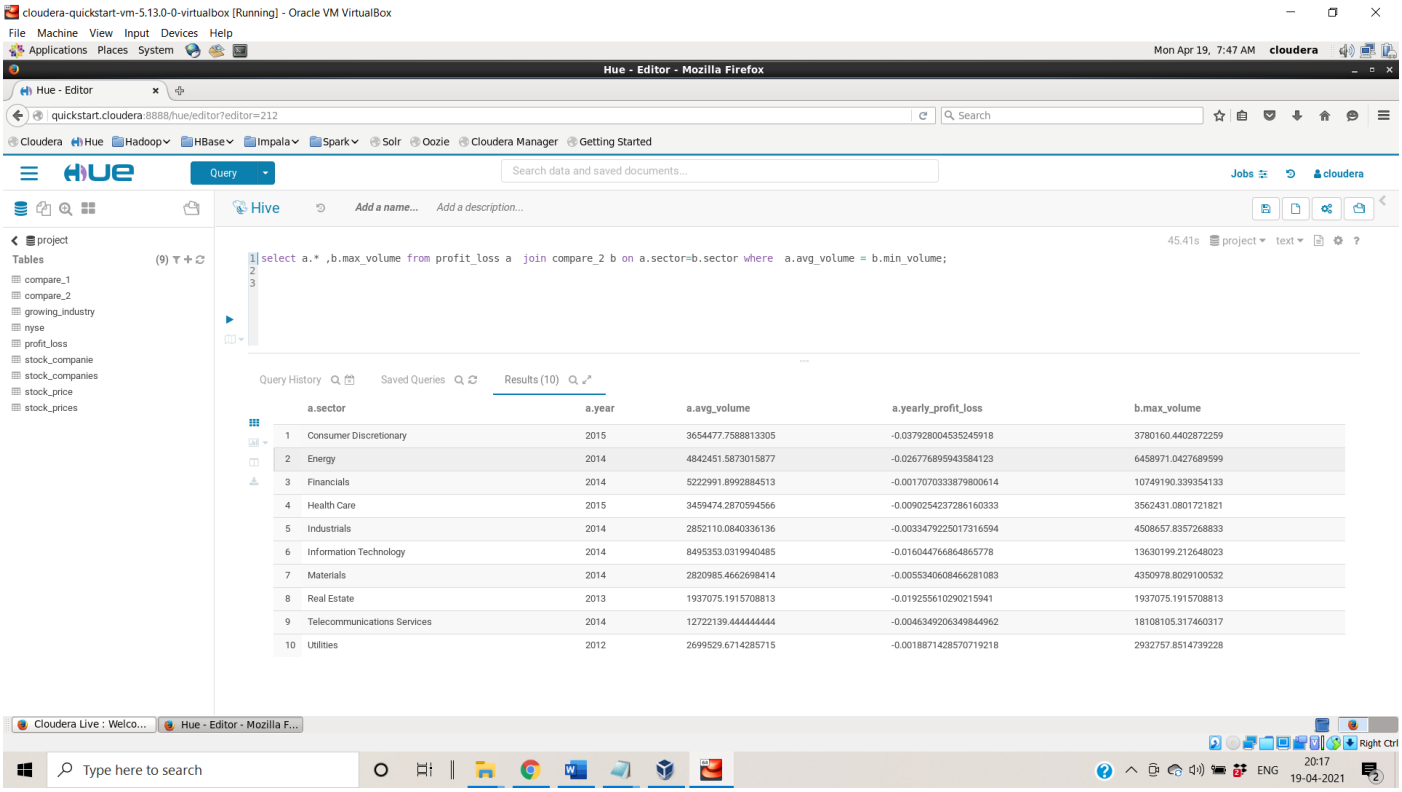


==========

a. Worst year

==========

select a.* ,b.max_volume from profit_loss a join compare_2 b on a.sector=b.sector where a.avg_volume = b.min_volume;



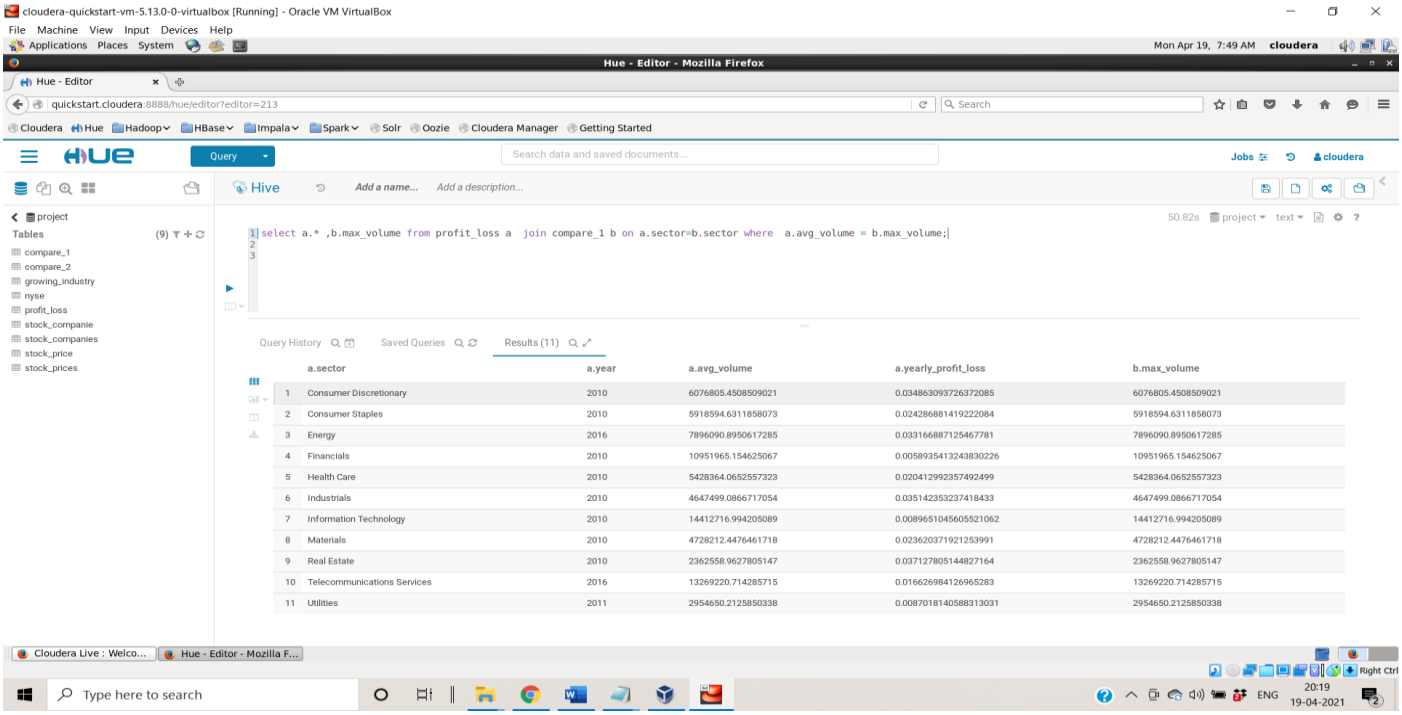
	a.sector	a.year	a.avg_volume	a.yearly_profit_loss	b.max_volume
1	Consumer Discretionary	2015	3654477.7588813305	-0.037928004535245918	3780160.4402872259
2	Energy	2014	4842451.5873015877	-0.026776895943584123	6458971.0427689599
3	Financials	2014	5222991.8992884513	-0.0017070333879800614	10749190.339354133
4	Health Care	2015	3459474.2870594566	-0.009025423728616033 3	3562431.0801721821
5	Industrials	2014	2852110.0840336136	-0.0033479225017316594	4508657.8357268833
6	Information Technology	2014	8495353.0319940485	-0.016044766864865778	13630199.212648023
7	Materials	2014	2820985.4662698414	-0.005534060846628108 3	4350978.8029100532
8	Real Estate	2013	1937075.1915708813	-0.019255610290215941	1937075.1915708813
9	Telecommunications Services	2014	12722139.444444444	-0.004634920634984496 2	18108105.317460317
10	Utilities	2012	2699529.6714285715	-0.0018871428570719218	2932757.8514739228

==========

b. Best year

==========

select a.* ,b.max_volume from profit_loss a join compare_1 b on a.sector=b.sector where a.avg_volume = b.max_volume;



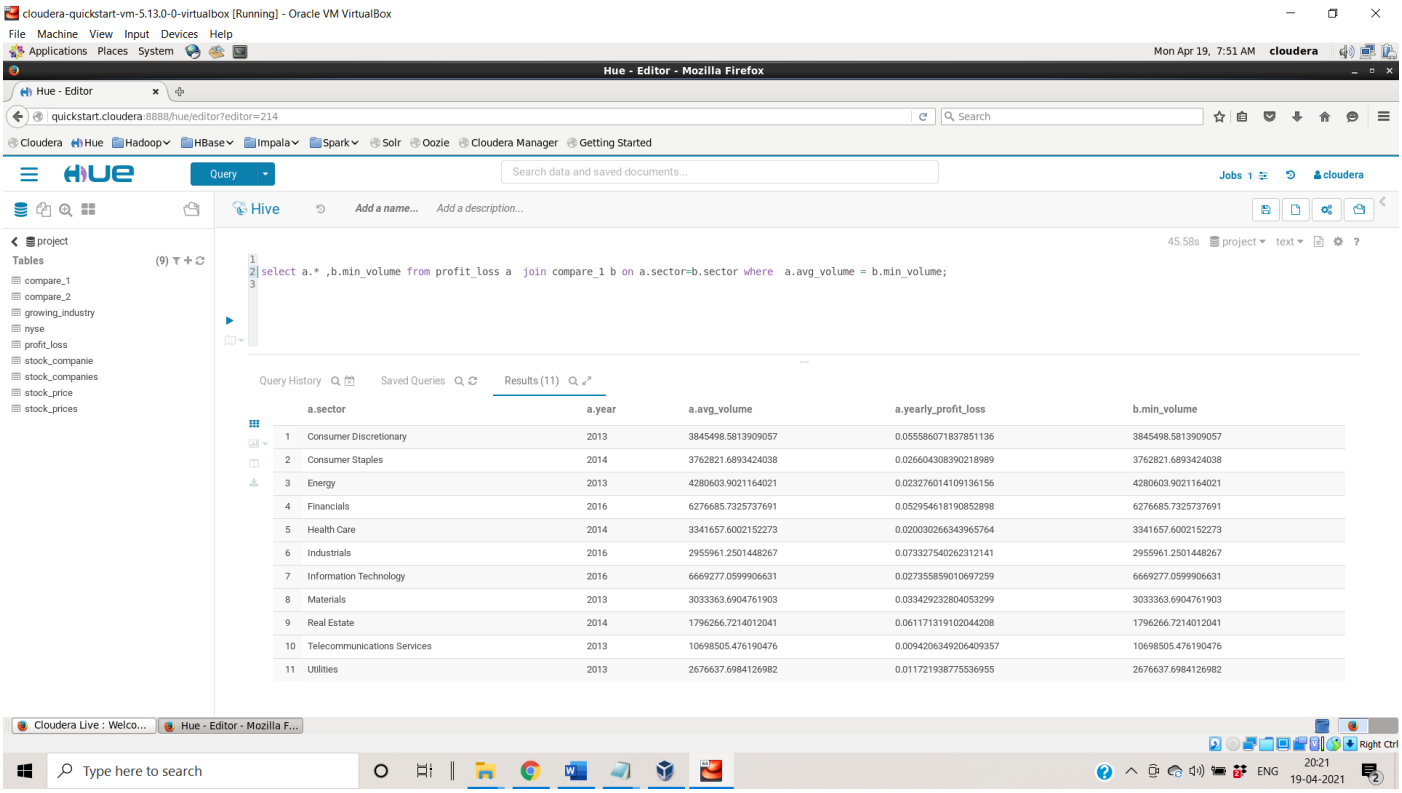
	a.sector	a.year	a.avg_volume	a.yearly_profit_loss	b.max_volume
1	Consumer Discretionary	2010	6076805.4508509021	0.034863093726372085	6076805.4508509021
2	Consumer Staples	2010	5918594.6311858073	0.024286881419222084	5918594.6311858073
3	Energy	2016	7896090.8950617285	0.033166887125467781	7896090.8950617285
4	Financials	2010	10951965.154625067	0.005893541324383022 6	10951965.154625067
5	Health Care	2010	5428364.0652557323	0.020412992357492499	5428364.0652557323
6	Industrials	2010	4647499.0866717054	0.035142353237418433	4647499.0866717054
7	Information Technology	2010	14412716.994205089	0.008965104560552106 2	14412716.994205089
8	Materials	2010	4728212.4476461718	0.023620371921253991	4728212.4476461718
9	Real Estate	2010	2362558.9627805147	0.037127805144827164	2362558.9627805147
10	Telecommunications Services	2016	13269220.714285715	0.016626984126965283	13269220.714285715
11	Utilities	2011	2954650.2125850338	0.008701814058831303 1	2954650.2125850338

==========

c. Stable year

==========

select a.* ,b.min_volume from profit_loss a join compare_1 b on a.sector=b.sector where a.avg_volume = b.min_volume;



	a.sector	a.year	a.avg_volume	a.yearly_profit_loss	b.min_volume
1	Consumer Discretionary	2013	3845498.5813909057	0.055586071837851136	3845498.5813909057
2	Consumer Staples	2014	3762821.6893424038	0.026604308390218989	3762821.6893424038
3	Energy	2013	4280603.9021164021	0.023276014109136156	4280603.9021164021
4	Financials	2016	6276685.7325737691	0.052954618190852898	6276685.7325737691
5	Health Care	2014	3341657.6002152273	0.020030266343965764	3341657.6002152273
6	Industrials	2016	2955961.2501448267	0.073327540262312141	2955961.2501448267
7	Information Technology	2016	6669277.0599906631	0.027355859010697259	6669277.0599906631
8	Materials	2013	3033363.6904761903	0.033429232804053299	3033363.6904761903
9	Real Estate	2014	1796266.7214012041	0.061171319102044208	1796266.7214012041
10	Telecommunications Services	2013	10698505.476190476	0.009420634920640935 7	10698505.476190476
11	Utilities	2013	2676637.6984126982	0.011721938775536955	2676637.6984126982