Stock Exchange Data Analysis

by Sujit Sonar

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

Q1) 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

steps: MYSQL

1) Login to FTP and load the csv files
2) Login to sql shell by typing mysql
3) login to mysql using the sqoop mysql credentials:
Command: **mysql -h sqoopdb.slbdh.cloudlabs.com -u sujitsonargmail -p**

4) to list the existing/available databases
Command: **show databases;**

```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| sujitsonargmail    |
+--------------------+
2 rows in set (0.11 sec)
```

5) select the database to load the data from csv
Command: **Use sujitsonargmail;**

6) Create the table "STOCK_PRICES" using the below table description
Command: CREATE TABLE STOCK_PRICES (
Trading_date DATE NOT NULL,
Symbol VARCHAR(255) NOT NULL,
Open DOUBLE NOT NULL,
Close DOUBLE NOT NULL,
Low DOUBLE NOT NULL,
High DOUBLE NOT NULL,
Volume INT NOT NULL
);

7) check the table is created in mysql
Command: show tables;

```
MySQL [sujitsonargmail]> show tables;
+---------------------------+
| Tables_in_sujitsonargmail |
+---------------------------+
|                           |
| STOCK_PRICES              |
| dept                      |
| student                   |
+---------------------------+
```

6) load the data from csv file from local FTP into the newly created table STOCK_PRICES in mysql
Command: load data local infile 'StockPrices.csv' into table STOCK_PRICES fields terminated by ',' enclosed by '"' lines terminated by '\r\n' IGNORE 1 LINES;

7) Check the data is loaded correctly into the STOCK_PRICES table in mysql
Command: select * from STOCK_PRICES limit 5;

```
MySQL [sujitsonargmail]> select * from STOCK_PRICES limit 5;
+--------------+--------+------------+------------+------------+------------+---------+
| Trading_date | Symbol | Open       | Close      | Low        | High       | volume  |
+--------------+--------+------------+------------+------------+------------+---------+
| 2016-01-05   | WLTW   |     123.43 | 125.839996 | 122.309998 |     126.25 | 2163600 |
| 2016-01-06   | WLTW   | 125.239998 | 119.980003 | 119.940002 | 125.540001 | 2386400 |
| 2016-01-07   | WLTW   | 116.379997 | 114.949997 |     114.93 | 119.739998 | 2489500 |
| 2016-01-08   | WLTW   | 115.480003 | 116.620003 |      113.5 | 117.440002 | 2006300 |
| 2016-01-11   | WLTW   | 117.010002 | 114.970001 | 114.089996 | 117.330002 | 1408600 |
+--------------+--------+------------+------------+------------+------------+---------+
5 rows in set (0.00 sec)
```

8) Similarly create the table for STOCK_COMPANIES in mysql and load the data from csv file from FTP to my sql

Creating table STOCK_COMPANIES:
Command: CREATE TABLE STOCK_COMPANIES (
Symbol VARCHAR(255) NOT NULL,
Company_name VARCHAR(255) NOT NULL,
Sector VARCHAR(255) NOT NULL,
Sub_industry VARCHAR(255) NOT NULL,
Headquarter VARCHAR(255) NOT NULL
);

Checking the table STOCK_COMPANIES in mysql:
Command: show tables;

```
MySQL [sujitsonargmail]> show tables;
+----------------------------+
| Tables_in_sujitsonargmail  |
+----------------------------+
| STOCK_COMPANIES            |
| STOCK_PRICES               |
| dept                       |
| student                    |
+----------------------------+
```

Loading data into STOCK_COMPANIES:
Command: load data local infile 'Stockcompanies.csv' into table STOCK_COMPANIES fields terminated by ',' enclosed by '"' lines terminated by '\r\n'IGNORE 1 LINES;

Check the data is loaded correctly into the STOCK_COMPANIES table in mysql
Command: select * from STOCK_COMPANIES limit 5;

```
MySQL [sujitsonargmail]> select * from STOCK_COMPANIES  limit 5;
+--------+--------------------+------------------------+-------------------------------+--------------------------+
| Symbol | Company_name       | Sector                 | Sub_industry                  | Headquarter              |
+--------+--------------------+------------------------+-------------------------------+--------------------------+
| MMM    | 3M Company         | Industrials            | Industrial Conglomerates      | St. Paul; Minnesota      |
| ABT    | Abbott Laboratories | Health Care           | Health Care Equipment         | North Chicago; Illinois  |
| ABBV   | AbbVie             | Health Care            | Pharmaceuticals               | North Chicago; Illinois  |
| ACN    | Accenture plc      | Information Technology | IT Consulting & Other Services | Dublin; Ireland          |
| ATVI   | Activision Blizzard | Information Technology | Home Entertainment Software   | Santa Monica; California |
+--------+--------------------+------------------------+-------------------------------+--------------------------+
5 rows in set (0.00 sec)
```

steps: SQOOP

creating hive table using sqoop pipeline to load the data from mysql STOCK_PRICES and STOCK_COMPANIES table

1) checking the STOCK_PRICES data using sqoop and sql connection:

**sqoop eval **
**--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" **
**--username sujitsonargmail **
**--password sujitsonargmailcut66 **
--e "describe STOCK_PRICES";

```
[sujitsonargmail@ip-10-0-41-79 ~]$ sqoop eval \
> --connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" \
> --username sujitsonargmail \
> --password sujitsonargmailcut66 \
> --e "describe STOCK_PRICES";
Warning: /opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/bin/../lib/sqoop/../accumulo does not exist! A
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/slf4j-log4j12-1
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/log4j-slf4j-imp
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/03/19 14:15:44 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7-cdh6.3.2
22/03/19 14:15:44 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider u
22/03/19 14:15:44 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver
nual loading of the driver class is generally unnecessary.
+--------------------------------------------------------------------------------------------------+
| Field                | Type                 | Null | Key | Default              | Extra                |
+--------------------------------------------------------------------------------------------------+
| Trading_date         | date                 | NO   |     | (null)               |                      |
| Symbol               | varchar(255)         | NO   |     | (null)               |                      |
| Open                 | double               | NO   |     | (null)               |                      |
| Close                | double               | NO   |     | (null)               |                      |
| Low                  | double               | NO   |     | (null)               |                      |
| High                 | double               | NO   |     | (null)               |                      |
| volume               | int                  | NO   |     | (null)               |                      |
+--------------------------------------------------------------------------------------------------+
```

***sqoop eval ***
***--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" ***
***--username sujitsonargmail ***
***--password sujitsonargmailcut66 ***
--query "select * from STOCK_PRICES limit 5";

```
[sujitsonargmail@ip-10-0-41-79 ~]$ sqoop eval \
> --connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" \
> --username sujitsonargmail \
> --password sujitsonargmailcut66 \
> --query "select * from STOCK_PRICES limit 5";
Warning: /opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/bin/../lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerB
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/log4j-slf4j-impl-2.8.2.jar!/org/slf4j/impl/StaticLogge
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/03/19 14:20:19 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7-cdh6.3.2
22/03/19 14:20:19 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
22/03/19 14:20:19 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registe
nual loading of the driver class is generally unnecessary.
+-------------------------------------------------------------------------------------------------------------------------------+
| Trading_date | Symbol               | Open                 | Close                | Low                  | High                 | volume     |
+-------------------------------------------------------------------------------------------------------------------------------+
| 2016-01-05   | WLTW                 | 123.43               | 125.839996           | 122.309998           | 126.25               | 2163600    |
| 2016-01-06   | WLTW                 | 125.239998           | 119.980003           | 119.940002           | 125.540001           | 2386400    |
| 2016-01-07   | WLTW                 | 116.379997           | 114.949997           | 114.93               | 119.739998           | 2489500    |
| 2016-01-08   | WLTW                 | 115.480003           | 116.620003           | 113.5                | 117.440002           | 2006300    |
| 2016-01-11   | WLTW                 | 117.010002           | 114.970001           | 114.089996           | 117.330002           | 1408600    |
+-------------------------------------------------------------------------------------------------------------------------------+
```

2) checking the STOCK_COMPANIES data using sqoop and sql connection:

**sqoop eval **
**--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" **
**--username sujitsonargmail **
**--password sujitsonargmailcut66 **
--e "describe STOCK_COMPANIES"

```
[sujitsonargmail@ip-10-0-41-79 ~]$ sqoop eval \
> --connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" \
> --username sujitsonargmail \
> --password sujitsonargmailcut66 \
> --e "describe STOCK_COMPANIES"
Warning: /opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/bin/../lib/sqoop/../accumulo does not exist! Ac
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/slf4j-log4j12-1.7
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/log4j-slf4j-impl-
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/03/19 14:22:37 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7-cdh6.3.2
22/03/19 14:22:37 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider usi
22/03/19 14:22:37 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'.
nual loading of the driver class is generally unnecessary.
-----------------------------------------------------------------------------------------------------------
| Field                | Type                 | Null | Key | Default              | Extra                |
-----------------------------------------------------------------------------------------------------------
| Symbol               | varchar(255)         | NO   |     | (null)               |                      |
| Company_name         | varchar(255)         | NO   |     | (null)               |                      |
| Sector               | varchar(255)         | NO   |     | (null)               |                      |
| Sub_industry         | varchar(255)         | NO   |     | (null)               |                      |
| Headquarter          | varchar(255)         | NO   |     | (null)               |                      |
-----------------------------------------------------------------------------------------------------------
```

**sqoop eval **
**--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" **
**--username sujitsonargmail **
**--password sujitsonargmailcut66 **
--query "select * from STOCK_COMPANIES limit 5";

```
[sujitsonargmail@ip-10-0-41-79 ~]$ sqoop eval \
> --connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" \
> --username sujitsonargmail \
> --password sujitsonargmailcut66 \
> --query "select * from STOCK_COMPANIES limit 5";
Warning: /opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/bin/../lib/sqoop/../accumulo does not exist! Accumulo imports will
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/slf4j-log4j12-1.7.25.jar!/org/slf4j/i
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/CDH-6.3.2-1.cdh6.3.2.p0.1605554/jars/log4j-slf4j-impl-2.8.2.jar!/org/slf4j
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/03/19 14:23:18 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7-cdh6.3.2
22/03/19 14:23:18 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
22/03/19 14:23:18 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is autom
nual loading of the driver class is generally unnecessary.
-----------------------------------------------------------------------------------------------------------
| Symbol               | Company_name         | Sector               | Sub_industry         | Headquarter          |
-----------------------------------------------------------------------------------------------------------
| MMM                  | 3M Company           | Industrials          | Industrial Conglomerates | St. Paul; Minnesota  |
| ABT                  | Abbott Laboratories  | Health Care          | Health Care Equipment | North Chicago; Illinois |
| ABBV                 | AbbVie               | Health Care          | Pharmaceuticals      | North Chicago; Illinois |
| ACN                  | Accenture plc        | Information Technology | IT Consulting & Other Services | Dublin; Ireland      |
| ATVI                 | Activision Blizzard  | Information Technology | Home Entertainment Software | Santa Monica; California |
-----------------------------------------------------------------------------------------------------------
```

1) Login to hive and create a database “sujitsonarproject”
Create database sujitsonarproject;

2) creating STOCK_PRICES hive table under sujitsonarproject using sqoop
***sqoop create-hive-table ***
***--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" ***
***--username sujitsonargmail ***
***--password sujitsonargmailcut66 ***
***--table STOCK_PRICES ***
--hive-database sujitsonarproject;

4) Importing STOCK_PRICES data from mysql to hive table
***sqoop import ***
***--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" ***
***--username sujitsonargmail ***
***--password sujitsonargmailcut66 ***
***--table STOCK_PRICES ***
***--hive-import ***
***--hive-table sujitsonarproject.stock_prices ***
-m 1;

5) Changing the Trading_date datatype from String to Date:
ALTER TABLE stock_prices CHANGE trading_date trading_date_new DATE;

Checking the datatype is correctly changed from string to Date
Describe stock_prices;

```
hive> describe stock_prices;
OK
trading_date_new        date
symbol                  string
open                    double
close                   double
low                     double
high                    double
volume                  int
```

6) similarly creating the STOCK_COMPANIES hive table under sujitsonarproject using sqoop and importing the data from mysql to hive table

Creating STOCK_COMPANIES hive table:

***sqoop create-hive-table ***
***--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" ***
***--username sujitsonargmail ***
***--password sujitsonargmailcut66 ***
***--table STOCK_COMPANIES ***
--hive-database sujitsonarproject;

Importing STOCK_COMPANIES data from mysql to hive table:

***sqoop import ***
***--connect "jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/sujitsonargmail" ***
***--username sujitsonargmail ***
***--password sujitsonargmailcut66 ***
***--table STOCK_COMPANIES ***
***--hive-import ***
***--hive-table sujitsonarproject.stock_companies ***
-m 1;

7) Checking the newly created hive tables under sujitsonarproject database using hive

```
hive> show tables;
OK
stock_companies
stock_prices
Time taken: 0.054 seconds, Fetched: 9 row(s)
hive>
```

Querying data from these tables using hive command line

Stock_prices:

```
hive> select * from stock_prices limit 5;
OK
2016-01-05      WLTW    123.43  125.839996      122.309998      126.25  2163600
2016-01-06      WLTW    125.239998      119.980003      119.940002      125.540001      2386400
2016-01-07      WLTW    116.379997      114.949997      114.93  119.739998      2489500
2016-01-08      WLTW    115.480003      116.620003      113.5   117.440002      2006300
2016-01-11      WLTW    117.010002      114.970001      114.089996      117.330002      1408600
Time taken: 0.335 seconds, Fetched: 5 row(s)
hive>
```

Stock_companies:

```
hive> select * from stock_companies limit 5;
OK
MMM     3M Company      Industrials     Industrial Conglomerates        St. Paul; Minnesota
ABT     Abbott Laboratories     Health Care     Health Care Equipment   North Chicago; Illinois
ABBV    AbbVie  Health Care     Pharmaceuticals North Chicago; Illinois
ACN     Accenture plc   Information Technology  IT Consulting & Other Services  Dublin; Ireland
ATVI    Activision Blizzard     Information Technology  Home Entertainment Software     Santa Monica; California
Time taken: 0.1 seconds, Fetched: 5 row(s)
hive>
```

Querying data from these hive tables using HUE

stock_prices:

	stock_prices.trading_date_new	stock_prices.symbol	stock_prices.open	stock_prices.close	stock_prices.low	stock_prices.high	stock_prices.volume
1	2016-01-05	WLTW	123.43	125.839996	122.309998	126.25	2163600
2	2016-01-06	WLTW	125.239998	119.980003	119.940002	125.540001	2386400
3	2016-01-07	WLTW	116.379997	114.949997	114.93	119.739998	2489500
4	2016-01-08	WLTW	115.480003	116.620003	113.5	117.440002	2006300
5	2016-01-11	WLTW	117.010002	114.970001	114.089996	117.330002	1408600

Stock_companies:

	stock_companies.symbol	stock_companies.company_name	stock_companies.sector	stock_companies.sub_industry	stock_companies.headquarter
1	MMM	3M Company	Industrials	Industrial Conglomerates	St. Paul; Minnesota
2	ABT	Abbott Laboratories	Health Care	Health Care Equipment	North Chicago; Illinois
3	ABBV	AbbVie	Health Care	Pharmaceuticals	North Chicago; Illinois
4	ACN	Accenture plc	Information Technology	IT Consulting & Other Services	Dublin; Ireland
5	ATVI	Activision Blizzard	Information Technology	Home Entertainment Software	Santa Monica; California

Observations: These shows that the data from mysql is correctly loaded into hive tables and we now have the data available in hive for analysis.

Q2) 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

steps: HIVE

creating a new hive table by joining the stock_prices and stock_companies hive tables using hive sql commands.

1) creating a helper table:

```
create table NYSE_STOCK_ssonar as
select substr(p.trading_date_new,1,4) as year,substr(p.trading_date_new,6,2) as month,
p.symbol, c.company_name, substr(c.headquarter,instr(c.headquarter,';')+1) as state,
c.sector, c.sub_industry, p.open, p.close, p.low, p.high, p.volume
FROM sujitsonarproject.stock_prices p
LEFT OUTER JOIN sujitsonarproject.stock_companies c
ON p.symbol = c.symbol;
```

2) Applying hive built-in functions to the helper table to create the final NYSE_STOCK_ssonar_project table

```
create table NYSE_STOCK_ssonar_project as
select year as Trading_year, month as Trading_month, symbol, company_name, state, sector,
sub_industry,
avg(open) as open, avg(close) as close, avg(low) as low, avg(high) as high, avg(volume) as volume
FROM sujitsonarproject.NYSE_STOCK_ssonar
GROUP BY  sector,state, sub_industry,symbol,company_name,year,month
order by sector,state,sub_industry,symbol,company_name,year,month;
```

3) Querying the first few records from the newly created hive table **NYSE_STOCK_ssonar_project**

Hive command line:

```
hive> select * from NYSE_STOCK_ssonar_project limit 5;
OK
2010    01      SIG     Signet Jewelers  Bermuda        Consumer Discretionary  Specialty Stores        27.31894715789474       27.667368526315787      27.0531579473684
2       27.821052421052627      521305.2631578947
2010    02      SIG     Signet Jewelers  Bermuda        Consumer Discretionary  Specialty Stores        27.901578999999998      28.10894721052632       27.6384212105263
14      28.29947384210526       297542.1052631579
2010    03      SIG     Signet Jewelers  Bermuda        Consumer Discretionary  Specialty Stores        29.413478347826093      29.635652       29.118695782608697      2
9.846521565217397       420786.95652173914
2010    04      SIG     Signet Jewelers  Bermuda        Consumer Discretionary  Specialty Stores        33.21047638095238       33.29809542857142       32.9057142857142
8       33.55523847619048       339376.1904761905
2010    05      SIG     Signet Jewelers  Bermuda        Consumer Discretionary  Specialty Stores        30.934500099999998      31.287000049999996      30.3769999999999
92      31.733000300000004      858560.0
Time taken: 0.095 seconds, Fetched: 5 row(s)
hive>
```

Hive in HUE:

	nyse_stock_ssonar_project.trading_year	nyse_stock_ssonar_project.trading_month	symbol	open	close	ock_ssonar_project.low	nyse_stock_ssonar_project.high	nyse_stock_ssonar_project.volum
1	2010	01	SIG	27.31894715789474	27.667368526315787	794736842	27.821052421052627	521305.2631578947
2	2010	02	SIG	27.901578999999998	28.10894721052632	1210526314	28.29947384210526	297542.1052631579
3	2010	03	SIG	29.413478347826093	29.635652	5782608697	29.846521565217397	420786.95652173914
4	2010	04	SIG	33.21047638095238	33.29809542857142	428571428	33.55523847619048	339376.1904761905
5	2010	05	SIG	30.934500099999998	31.287000049999996	9999999992	31.733000300000004	858560

Observations: These shows that the newly combined hive table with all the columns is correctly created, and we have the data available in hive for analysis.

steps: DATA ANALYSIS USING HIVE

querying data from the nyse_stock_ssonar_project table for for data analysis.

Q3) Find the top five companies that are good for investment

1)creating a helper table

```
create table stock_data1 as
select company_name, min(trading_year) min_year, max(trading_year) max_year,
min(trading_month) min_month, max(trading_month) max_month
from sujitsonarproject.nyse_stock_ssonar_project
group by company_name;
```

2) creating the final table to query the top five companies that are good for investment.
p.s: The metric considered are open and close data.

```
create table stock_growth_table as
select table1.company_name, table1.symbol, table1.state, table1.sector, table1.sub_industry,
table1.min_year,table1.min_month,table2.max_year,table2.max_month,
table1.open,table2.close, round(((table2.close-table1.open)/table1.open)*100,2) as
growth_percent
from (select t1.company_name, t2.symbol, t2.state, t2.sector, t2.sub_industry,
t1.min_year, t1.min_month, t2.open
FROM sujitsonarproject.stock_data1 t1, sujitsonarproject.nyse_stock_ssonar_project t2
where t1.company_name = t2.company_name
and t1.min_year = t2.trading_year
and t1.min_month = t2.trading_month) as table1,
(select t1.company_name, t2.symbol, t2.state, t2.sector, t2.sub_industry,
t1.max_year, t1.max_month, t2.close
FROM sujitsonarproject.stock_data1 t1, sujitsonarproject.nyse_stock_ssonar_project t2
where t1.company_name = t2.company_name
and t1.max_year = t2.trading_year
and t1.max_month = t2.trading_month) as table2
where table1.company_name = table2.company_name
sort by growth_percent desc;
```

Observations: Querying the top five companies that are good for investment based on open and close price:

```
select company_name,sector,growth_percent from stock_growth_table limit 5;
```

	company_name	sector	growth_percent
1	Netflix Inc.	Information Technology	1536.85
2	Regeneron	Health Care	1382.28
3	Ulta Salon Cosmetics & Fragrance Inc	Consumer Discretionary	1174.92
4	United Rentals; Inc.	Industrials	1064.19
5	Alaska Air Group Inc	Industrials	879.08

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

1) Creating a helper table to filter states , having at least two or more industries mapped

```
create table industry_growth as
select state, sub_industry, count(concat(state,sub_industry)) as ind_count, avg(growth_percent) as industry_growth
from sujitsonarproject.stock_growth_table
group by state, sub_industry
having count(concat(state,sub_industry)) >=2
sort by industry_growth desc ;
```

Final table:

```
create table state_industry_growth as
select t3.state as state,t3.sub_industry as sub_industry, t3.ind_count as ind_count,
round(t3.industry_growth,2) as max_ind_growth
from sujitsonarproject.industry_growth as t3,
(select state, max(industry_growth) as max_ind_growth
from sujitsonarproject.industry_growth
group by state) as t4
where t3.state = t4.state
and t3.industry_growth = t4.max_ind_growth
sort by max_ind_growth desc;
```

Observations: Querying the best-growing industry by each state, having at least two or more industries mapped

```
select state,sub_industry,max_ind_growth from sujitsonarproject.state_industry_growth;
```

	state	sub_industry	max_ind_growth
1	Texas	Airlines	571.09
2	California	Internet Software & Services	336.64
3	Washington	Internet & Direct Marketing Retail	323.85
4	Massachusetts	Semiconductors	285.15
5	Virginia	Aerospace & Defense	255.38
6	New York	Diversified Financial Services	244.28
7	North Carolina	Apparel; Accessories & Luxury Goods	234.57
8	Florida	Industrial Conglomerates	177.39
9	Ohio	Banks	171.57
10	Minnesota	Packaged Foods & Meats	166.67
11	Pennsylvania	Diversified Chemicals	163.19
12	Ireland	Pharmaceuticals	155.86
13	Michigan	MultiUtilities	143.77
14	Illinois	Industrial Machinery	136.02
15	Wisconsin	Electric Utilities	131.92
16	New Jersey	Health Care Equipment	120.89
17	United Kingdom	Insurance Brokers	101.05
18	Maryland	Cable & Satellite	85.84
19	Missouri	Industrial Conglomerates	85.57
20	Connecticut	Industrial Conglomerates	75.07
21	Oklahoma	Oil & Gas Exploration & Production	74.72
22	Arizona	Semiconductors	29.19

Q5) For each sector find the following.

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

1) creating table to find the Worst, Best and Stable year for each sector

```
create table sector_growth_table as
select open_table.sector as sector, open_table.trading_year as trading_year, open_table.open as open,
close_table.close as close, (close_table.close - open_table.open) as growth,
round(((close_table.close - open_table.open)/open_table.open)*100,2) as percent_growth
from
(select t5.sector as sector, t5.trading_year as trading_year,avg(t5.open) as open
from sujitsonarproject.nyse_stock_ssonar_project as t5,
(select trading_year, sector, min(trading_month) as min_month , max(trading_month) as max_month
from sujitsonarproject.nyse_stock_ssonar_project
group by sector, trading_year) as min_max_month
where t5.sector = min_max_month.sector
and t5.trading_year = min_max_month.trading_year
and t5.trading_month = min_max_month.min_month
group by t5.sector, t5.trading_year) as open_table
join (select t5.sector as sector, t5.trading_year as trading_year,avg(t5.close) as close
from sujitsonarproject.nyse_stock_ssonar_project as t5,
(select trading_year, sector, min(trading_month) as min_month , max(trading_month) as max_month
from sujitsonarproject.nyse_stock_ssonar_project
group by sector, trading_year) as min_max_month
where t5.sector = min_max_month.sector
and t5.trading_year = min_max_month.trading_year
and t5.trading_month = min_max_month.max_month
group by t5.sector, t5.trading_year) as close_table
where open_table.sector = close_table.sector
and open_table.trading_year = close_table.trading_year;
```

a. Worst year

select x.sector,x.trading_year, x.growth from sujitsonarproject.sector_growth_table as x,
(select sector,min(growth) as min_growth from sujitsonarproject.sector_growth_table
group by sector) as y
where x.sector = y.sector
and x.growth = y.min_growth
order by x.trading_year;

	x.sector	x.trading_year	x.growth
1	Materials	2011	-3.9670180038293523
2	Information Technology	2011	-2.903352390444063
3	Financials	2011	-6.859744235256144
4	Consumer Discretionary	2011	4.860477526637112
5	Real Estate	2013	-4.463394495796393
6	Utilities	2015	-6.474239674240252
7	Telecommunications Services	2015	-2.2943546409090914
8	Industrials	2015	-2.640463503133674
9	Energy	2015	-10.098865639520199
10	Health Care	2016	2.0804205425002777
11	Consumer Staples	2016	3.1815668437405975

Observations: 2011 and 2015 are years where most of sectors are most hit

b. Best year

select x.sector,x.trading_year, x.growth from sujitsonarproject.sector_growth_table as x,
(select sector,max(growth) as max_growth from sujitsonarproject.sector_growth_table
group by sector) as y
where x.sector = y.sector
and x.growth = y.max_growth
order by x.trading_year;

	x.sector	x.trading_year	x.growth
1	Information Technology	2013	15.538204765520831
2	Consumer Discretionary	2013	24.392649212454756
3	Utilities	2014	10.267945201691397
4	Telecommunications Services	2014	5.06370545238094
5	Real Estate	2014	18.892443342722785
6	Health Care	2014	24.325416317192747
7	Consumer Staples	2014	9.323455196317902
8	Materials	2016	19.037147210416038
9	Industrials	2016	19.804276198605308
10	Financials	2016	16.32740404883174
11	Energy	2016	18.641447432470073

Observations: 2013,3014 and 2016 are the best years for most of the sectors

c. Stable year

```
SELECT sector,
COLLECT_SET(`2010`)[0] AS `2010`,
COLLECT_SET(`2011`)[0] AS `2011`,
COLLECT_SET(`2012`)[0] AS `2012`,
COLLECT_SET(`2013`)[0] AS `2013`,
COLLECT_SET(`2014`)[0] AS `2014`,
COLLECT_SET(`2015`)[0] AS `2015`,
COLLECT_SET(`2016`)[0] AS `2016`
FROM (
SELECT sector,
CASE WHEN trading_year=2010 THEN round(growth,2) END AS `2010`,
CASE WHEN trading_year=2011 THEN round(growth,2) END AS `2011`,
CASE WHEN trading_year=2012 THEN round(growth,2) END AS `2012`,
CASE WHEN trading_year=2013 THEN round(growth,2) END AS `2013`,
CASE WHEN trading_year=2014 THEN round(growth,2) END AS `2014`,
CASE WHEN trading_year=2015 THEN round(growth,2) END AS `2015`,
CASE WHEN trading_year=2016 THEN round(growth,2) END AS `2016`
FROM sujitsonarproject.sector_growth_table)tbl1
GROUP BY sector;
```

	sector	2010	2011	2012	2013	2014	2015	2016
1	Consumer Discretionary	14.4	4.86	7.73	24.39	9.6	6.2	14.68
2	Consumer Staples	4.18	4.16	4.54	8.82	9.32	4.89	3.18
3	Energy	7.71	-3.59	-1.95	12.58	-6.26	-10.1	18.64
4	Financials	2.99	-6.86	5.6	15.3	5.81	0.21	16.33
5	Health Care	4.55	4.64	9.58	17.32	24.33	4.42	2.08
6	Industrials	9.4	-2.55	3.22	16.46	8.57	-2.64	19.8
7	Information Technology	6.5	-2.9	4.06	15.54	6.65	8.97	14.41
8	Materials	6.39	-3.97	7.42	8.63	6.7	-2.82	19.04
9	Real Estate	8.5	4.44	8.41	-4.46	18.89	1.74	1.23
10	Telecommunications Services	2.27	-1.64	2.75	0.92	5.06	-2.29	3.94
11	Utilities	0.58	3.18	1.38	2.41	10.27	-6.47	6.11

Observations: by looking at the table, 2012 appears to be the stable year for most of the sectors.