**Stock Exchange Data Analysis**

* Download the dataset

Stockcompanies.csv and StockPrices.csv

* Login to web console via practice Labs;

hdfs dfs -mkdir NYSE

hdfs dfs -mkdir NYSE1

* Upload the data set

Stockcompanies.csv to NYSE folder via hue file browser.

StockPrices.csv to NYSE folder via hue file browser.

* Goto mysql by using following command

mysql -h sqoopdb.slbdh.cloudlabs.com -u heapvrgmail -p

heapvrgmailcq3p6

* Use database heapvrgmail
* Create table Stock\_Prices;

create table Stock\_Prices(Trading\_date date,Symbol varchar(100),Open double,Close double,Low double,High double,Volumn int);

* Create table Stock\_Companies;
* create table Stock\_Companies(Symbol varchar(100),Company\_name varchar(100),Sector varchar(100),Sub\_industry varchar(100),Headquarter varchar(100));
* Exit from mysql by pressing control+z and return back to webconsol;

**TASK : Create a data pipeline using sqoop to pull the data from the table below from MYSQL server into Hive.**

* Import data from hue file browser to mysql table via scoop ;
* sqoop export --connect jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/heapvrgmail --username heapvrgmail --password heapvrgmailcq3p6 --table Stock\_Companies --export-dir /user/heapvrgmail/NYSE --m 1;
* sqoop export --connect jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/heapvrgmail --username heapvrgmail --password heapvrgmailcq3p6 --table Stock\_Prices --export-dir /user/heapvrgmail/NYSE1 --m 1;
* Create table in hive using sqoop command via mysql
* sqoop import --connect jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/heapvrgmail --username heapvrgmail --password heapvrgmailcq3p6 --table Stock\_Prices --m 1 --hive-import --create-hive-table --hive-table hemal1.Stock\_Prices;
* sqoop import --connect jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/heapvrgmail --username heapvrgmail --password heapvrgmailcq3p6 --table Stock\_Companies --m 1 --hive-import --create-hive-table --hive-table hemal1.Stock\_Companies;

**TASK : 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 stock\_data\_analysis(trading\_year int,trading\_month int,Symbol string,companyname string, state string, sector string,sub\_industry string, open double, close double, low double, high double,volumn int);

INSERT INTO stock\_data\_analysis (trading\_year, trading\_month, symbol,companyname,state,sector,sub\_industry,open,close,low,high,volumn)

SELECT substr(b.trading\_date,0,4),substr(b.trading\_date,6,2),a.symbol,a.company\_name, substr(a.headquarter,locate(";",a.headquarter)+1,length(a.headquarter)-locate(";",a.headquarter)),a.sector,a.sub\_industry,round(avg(b.open)),round(avg(b.close)),round(avg(b.low)),round(avg(b.high)),round(avg(b.volumn))

FROM stock\_companies A

JOIN stock\_prices B ON A.symbol = B.symbol

GROUP BY A.headquarter,A.company\_name, A.sub\_industry,A.symbol,A.sector,substr(B.trading\_date,0,7),substr(B.trading\_date,0,4),substr(B.trading\_date,6,2)

**TASK : Find the top five companies that are good for investment**

**Below table is to get all the information about last closing price of each company**

create table Task\_3\_table1 as SELECT a.trading\_year as ending\_year,a.trading\_month as ending\_month,a.close as ending\_price,a.companyname

FROM

stock\_data\_analysis a JOIN

(

SELECT

max(trading\_year) as max\_year, max(trading\_month) as max\_month

FROM stock\_data\_analysis) maxyearmonth

ON a.trading\_year = maxyearmonth.max\_year and a.trading\_month=maxyearmonth.max\_month

ORDER BY a.companyname;

**Below table is to get all the information about initial closing price of each company**

create table Task\_3\_table2 as SELECT a.trading\_year as starting\_year,a.trading\_month as starting\_month,a.close as starting\_price,a.companyname

FROM

stock\_data\_analysis a JOIN

(

SELECT

min(trading\_year) as min\_year, min(trading\_month) as min\_month

FROM stock\_data\_analysis) minyearmonth

ON a.trading\_year = minyearmonth.min\_year and a.trading\_month=minyearmonth.min\_month

ORDER BY a.companyname;

**Below query will calculate and fetch best investment return given by top 5 company**

select round(((a.ending\_price-b.starting\_price)/b.starting\_price)\*100) as cg,a.companyname form from task\_3\_table1 a , task\_3\_table2 b where a.companyname=b.companyname

order by cg desc limit 6;

Graphical user interface, text, application, email

Description automatically generated

**TASK : Show the best-growing industry by each state, having at least two or more industries mapped.**

**Below table is to get all the information about last closing price of each company with State and sub\_industry,**

create table Task\_5\_table1 as SELECT a.trading\_year as ending\_year,a.trading\_month as ending\_month,a.close as ending\_price,a.companyname,a.sector,a.sub\_industry,a.state

FROM

stock\_data\_analysis a JOIN

(

SELECT

max(trading\_year) as max\_year, max(trading\_month) as max\_month

FROM stock\_data\_analysis) maxyearmonth

ON a.trading\_year = maxyearmonth.max\_year and a.trading\_month=maxyearmonth.max\_month

ORDER BY a.companyname;

**Below table is to get all the information about initial closing price of each company with State and sub\_industry,**

create table Task\_5\_table2 as SELECT a.trading\_year as starting\_year,a.trading\_month as starting\_month,a.close as starting\_price,a.companyname,a.sector,a.sub\_industry,a.state

FROM

stock\_data\_analysis a JOIN

(

SELECT

min(trading\_year) as min\_year, min(trading\_month) as min\_month

FROM stock\_data\_analysis) minyearmonth

ON a.trading\_year = minyearmonth.min\_year and a.trading\_month=minyearmonth.min\_month

ORDER BY a.companyname;

**Below table calculate the growth of each company by sector and sub\_industry wise.**

create table Task\_5\_table3 as select round(((a.ending\_price-b.starting\_price)/(b.starting\_price))\*100) as growth,a.sector as sector,a.sub\_industry as sub\_industry,a.state as state from task\_5\_table1 a,task\_5\_table2 b where a.companyname=b.companyname

order by growth;

**below table calculate the total growth of the sector having more then 1 sub\_industry**

create table task\_5\_table4 as select state,sub\_industry,round(avg(growth),2) as growth from task\_5\_table3 group by state,sub\_industry having count(sub\_industry)>2

order by growth desc;

**below query will calculate and fetch the industry which has maximum growth by each state**

select a.state , sub\_industry, growth from task\_5\_table4 a, (select state,max(growth) max\_g from task\_5\_table4 group by state) max\_t1 where max\_t1.state = a.state and

a.growth = max\_t1.max\_g order by growth desc;

**Graphical user interface, text, application, email

Description automatically generated**

**TASK : For each sector find the following.**

* **Worst year**
* **Best year**
* **Stable year**

**below table calculate and fetch the closing price of each company on starting of the each year wise**

create table Task\_4\_table1 as SELECT a.trading\_year as ending\_year,a.trading\_month as ending\_month,a.close as ending\_price,a.sector,a.companyname

FROM

stock\_data\_analysis a JOIN

(

SELECT

max(trading\_month) as max\_month

FROM stock\_data\_analysis) maxyearmonth

ON a.trading\_month=maxyearmonth.max\_month

order by a.companyname;

**below table calculate and fetch the closing price of each company ending of the year wise**

create table Task\_4\_table2 as SELECT a.trading\_year as starting\_year,a.trading\_month as starting\_month,a.close as starting\_price,a.sector,a.companyname

FROM

stock\_data\_analysis a JOIN

(

SELECT

min(trading\_month) as min\_month

FROM stock\_data\_analysis) minyearmonth

ON a.trading\_month=minyearmonth.min\_month

order by a.companyname;

**below table calculate and fetch the total closing price of each company sector by year end**

create table Task\_4\_table3 as SELECT a.sector as sector,sum(a.ending\_price) as totalendprice, a.ending\_year as endingyear FROM task\_4\_table1 a GROUP BY a.sector,a.ending\_year;

**below table calculate and fetch the total closing price of each company sector by starting of each year**

create table Task\_4\_table4 as SELECT a.sector as sector,sum(a.starting\_price) as totalstarting\_price, a.starting\_year as starting\_year FROM task\_4\_table2 a GROUP BY a.sector,a.starting\_year;

**below query calculate and fetch information about**

* **Best year**
* **Worst year**
* **Stable year**

**For each sector.**

create table task\_4\_table5 as select a.endingyear as years,round(((a.totalendprice-b.totalstarting\_price)/b.totalstarting\_price)\*100) as growth,a.sector from task\_4\_table3 a , task\_4\_table4 b where a.sector=b.sector and a.endingyear=b.starting\_year

order by growth desc;

select distinct(sector),years,growth from task\_4\_table5;

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**