Stock Exchange Data Analysis

Analysis 1: Create a data pipeline using sqoop to pull the data from the table below from MYSQL server into Hive.

```
MySQL [ramkumarramachandran03gma]> show tables;
+-------------------------------------+
| Tables_in_ramkumarramachandran03gma |
+-------------------------------------+
| STOCK_COMPANIES                     |
| STOCK_PRICES                        |
+-------------------------------------+
2 rows in set (0.00 sec)

MySQL [ramkumarramachandran03gma]>
```

- sqoop import --connect jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/ramkumarramachandran03gma --username ramkumarramachandran03gma -P --table STOCK_COMPANIES -hive-import

```
MySQL [ramkumarramachandran03gma]> select * from STOCK_COMPANIES ;
+--------+---------------------------+---------------------+---------------------------------+--------------------------+
| Symbol | Company_Name              | Sector              | Sub_industry                    | Headquarter              |
+--------+---------------------------+---------------------+---------------------------------+--------------------------+
| CMG    | Chipotle Mexican Grill    | Consumer Discretionary | Restaurants                  | Denver; Colorado         |
| CB     | Chubb Limited             | Financials          | Property & Casualty Insurance   | Zurich; Switzerland      |
| CHD    | Church & Dwight           | Consumer Staples    | Household Products              | Ewing; New Jersey        |
| CI     | CIGNA Corp.               | Health Care         | Managed Health Care             | Philadelphia; Pennsylvania |
| XEC    | Cimarex Energy            | Energy              | Oil & Gas Exploration & Production | Denver; Colorado      |
```

- sqoop import --connect jdbc:mysql://sqoopdb.slbdh.cloudlabs.com/ramkumarramachandran03gma --username ramkumarramachandran03gma -P --table STOCK_PRICES -hive-import

```
MySQL [ramkumarramachandran03gma]> select * from STOCK_PRICES;
+--------------+--------+-------------+-------------+-------------+-------------+-----------+
| Trading_date | Symbol | Open        | Close       | Low         | High        | Volume    |
+--------------+--------+-------------+-------------+-------------+-------------+-----------+
| 2015-04-27   | ZTS    |       47.98 |   46.830002 |   46.740002 |   47.990002 |   2684100 |
| 2015-04-27   | AIV    |       38.77 |   39.220001 |       38.77 |       39.34 |   2007100 |
| 2015-04-28   | A      |   41.869999 |       42.18 |       41.57 |   42.310001 |   1803500 |
| 2015-04-28   | AAL    |   51.700001 |   51.189999 |   50.009998 |       51.82 |   9235700 |
| 2015-04-28   | AAP    |         145 |       144.5 |         143 |  145.229996 |    794300 |
| 2015-04-28   | AAPL   |  134.460007 |  130.559998 |  129.570007 |  134.539993 | 118924000 |
| 2015-04-28   | ABBV   |   64.599998 |   66.489998 |   64.199997 |   66.489998 |  13084900 |
```

Analysis 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.

```
CREATE TABLE BDHS_PROJECT_RAMKUMAR.CUSTOM_HIVETABLE as
SELECT
date_format(pr.trading_date,'yyyy') as Trading_year,
date_format(pr.trading_date,'MM') as Trading_month,
pr.symbol as Symbol,
cm.company_name as CompanyName,
split(cm.headquarter,'\\;')[1] as State,
cm.sector as Sector,
cm.sub_industry as Sub_Industry,
avg(pr.open) as Open,
avg(pr.close) as Close,
avg(pr.low) as Low,
avg(pr.high) as High,
avg(pr.volume) as Volume
FROM BDHS_PROJECT_RAMKUMAR.STOCK_PRICES pr
INNER JOIN BDHS_PROJECT_RAMKUMAR.STOCK_COMPANIES cm on (pr.symbol=cm.symbol) group by
date_format(pr.trading_date,'yyyy'),date_format(pr.trading_date,'MM'),cm.sub_industry,pr.symbol,cm.company_name,cm.sector,split(cm.head
```

```
INFO  : Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 26.44 sec   HDFS Read: 51306777 HDFS Write: 66 job_1601750042861_12555
INFO  : Total MapReduce CPU Time Spent: 26 seconds 440 msec
INFO  : Completed executing command(queryId=hive_20201205102200_438ac3ec-30d7-43ed-857a-e761b22ff117); Time taken: 33.444 seconds
INFO  : OK
```

```
Success.
```

Query used to create requested table,

CREATE TABLE CUSTOM_HIVETABLE as
SELECT
date_format(pr.trading_date,'yyyy') as Trading_year,
date_format(pr.trading_date,'MM') as Trading_month,
pr.symbol as Symbol,
cm.company_name as CompanyName,
split(cm.headquarter,'\\;')[1] as State,
cm.sector as Sector,
cm.sub_industry as Sub_Industry,
avg(pr.open) as Open,
avg(pr.close) as Close,
avg(pr.low) as Low,
avg(pr.high) as High,
avg(pr.volume) as Volume
FROM BDHS_PROJECT_RAMKUMAR.STOCK_PRICES pr
INNER JOIN BDHS_PROJECT_RAMKUMAR.STOCK_COMPANIES cm on (pr.symbol=cm.symbol) group by
date_format(pr.trading_date,'yyyy'),date_format(pr.trading_date,'MM'),cm.sub_industry,pr.symbol,cm.company_name,cm.sector,split(cm.headquarter,'\\;')[1];

Analysis 3: Find the top five companies that are good for investment

With the Calculation of Percentage Exchange for the STOCK we can get the required analysis via HIVE.

*select companyname, avg((close - open) / (open * 100)) as returns from CUSTOM_HIVETABLE*
group by companyname
order by returns desc limit 5;

Below are the top 5 companies for investment,

```
select companyname, avg((close - open) / (open * 100)) as returns
from BDHS_PROJECT_RAMKUMAR.CUSTOM_HIVETABLE
group by companyname
order by returns desc limit 5;
```

Query History | Saved Queries | Results (5)

	companyname	returns
1	Fortive Corp	1.180334491912715e-05
2	Accenture plc	1.1646649293984042e-05
3	Intuit Inc.	1.1607952010693232e-05
4	Illumina Inc	1.141030218826657e-05
5	Verisign Inc.	1.0891646385668341e-05

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

*select state,sector, avg((close - open) / (open * 100)) as returns from CUSTOM_HIVETABLE*
group by state, sector
having count(sector)>2
order by returns desc;

```
select state,sector, avg((close - open) / (open * 100)) as returns
from BDHS_PROJECT_RAMKUMAR.CUSTOM_HIVETABLE
group by state, sector
having count(sector)>2
order by returns desc;
```

Query History | Saved Queries | Results (208)

	state	sector	returns	ns
1	Ohio	Materials	1.0538914926610817e-05	
2	Missouri	Consumer Discretionary	1.048767925086381e-05	
3	Wisconsin	Information Technology	1.0249718680126459e-05	
4	Kentucky	Health Care	9.98283030903852 3e-06	
5	Illinois	Information Technology	9.938460558909571e-06	

1	te	string
2	tor	string
3	urns	double
4		
5		

Analysis 5: For each sector find the following.

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

*SELECT sector,trading_year, avg((close - open) / (open * 100)) as returns from CUSTOM_HIVETABLE*
group by sector,trading_year
order by returns desc;

Graphical view created in HIVE for differentiating the outcome of years for every sectors

```
select sector,trading_year, avg((close - open) / (open * 100)) as returns
from BDHS_PROJECT_RAMKUMAR.CUSTOM_HIVETABLE
group by sector,trading_year
order by returns desc;
```