Big Data Hadoop and Spark Developer

Certification Project- Sheeba Varghese

**Stock Exchange Data Analysis**

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

Check the TABLE description: STOCK\_PRICES

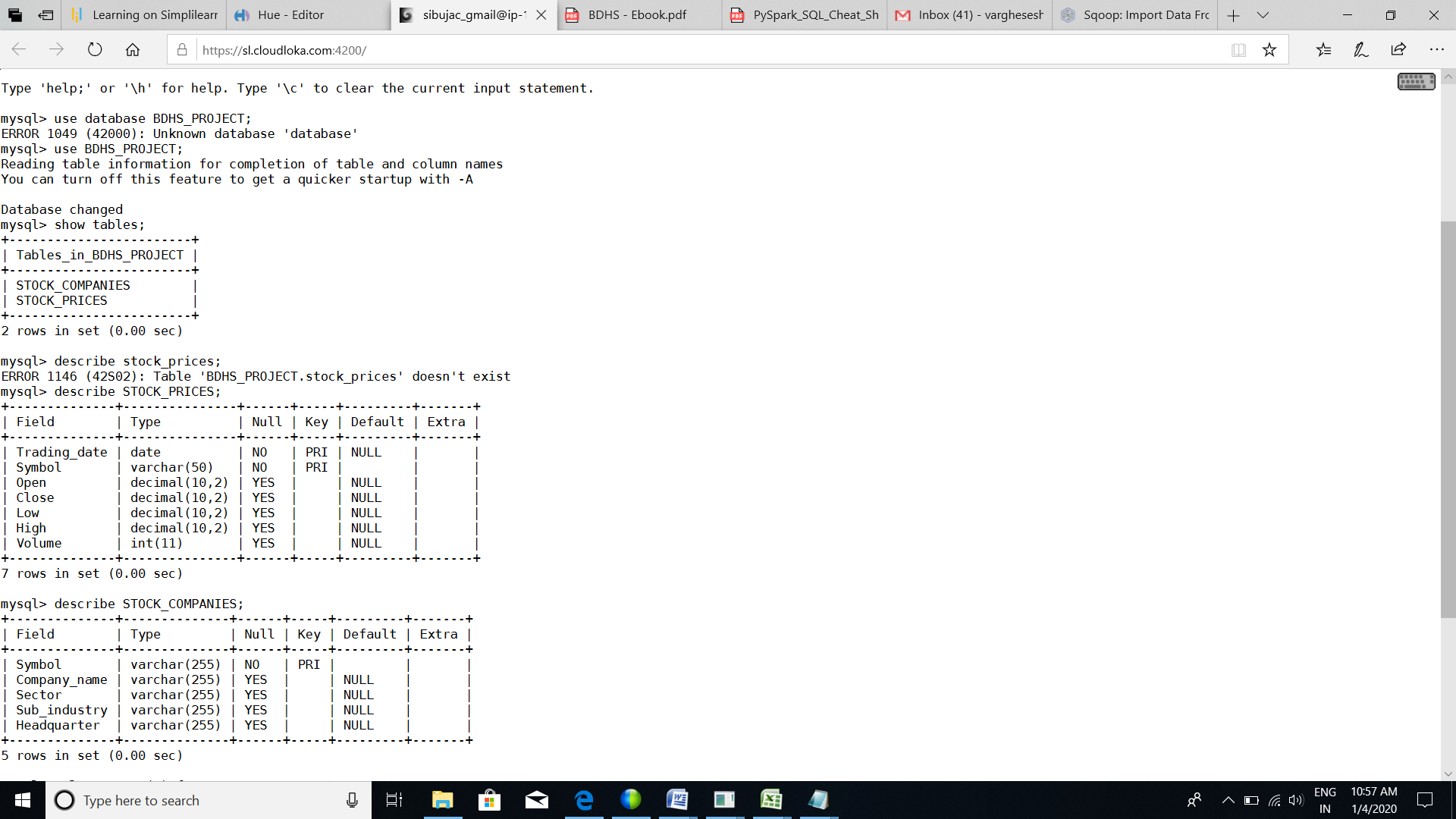
|  |  |
| --- | --- |
| Column Name | Datatype |
| Trading\_date | Date |
| Symbol | String |
| Open | double |
| Close | double |
| Low | double |
| High | double |
| Volume | int |

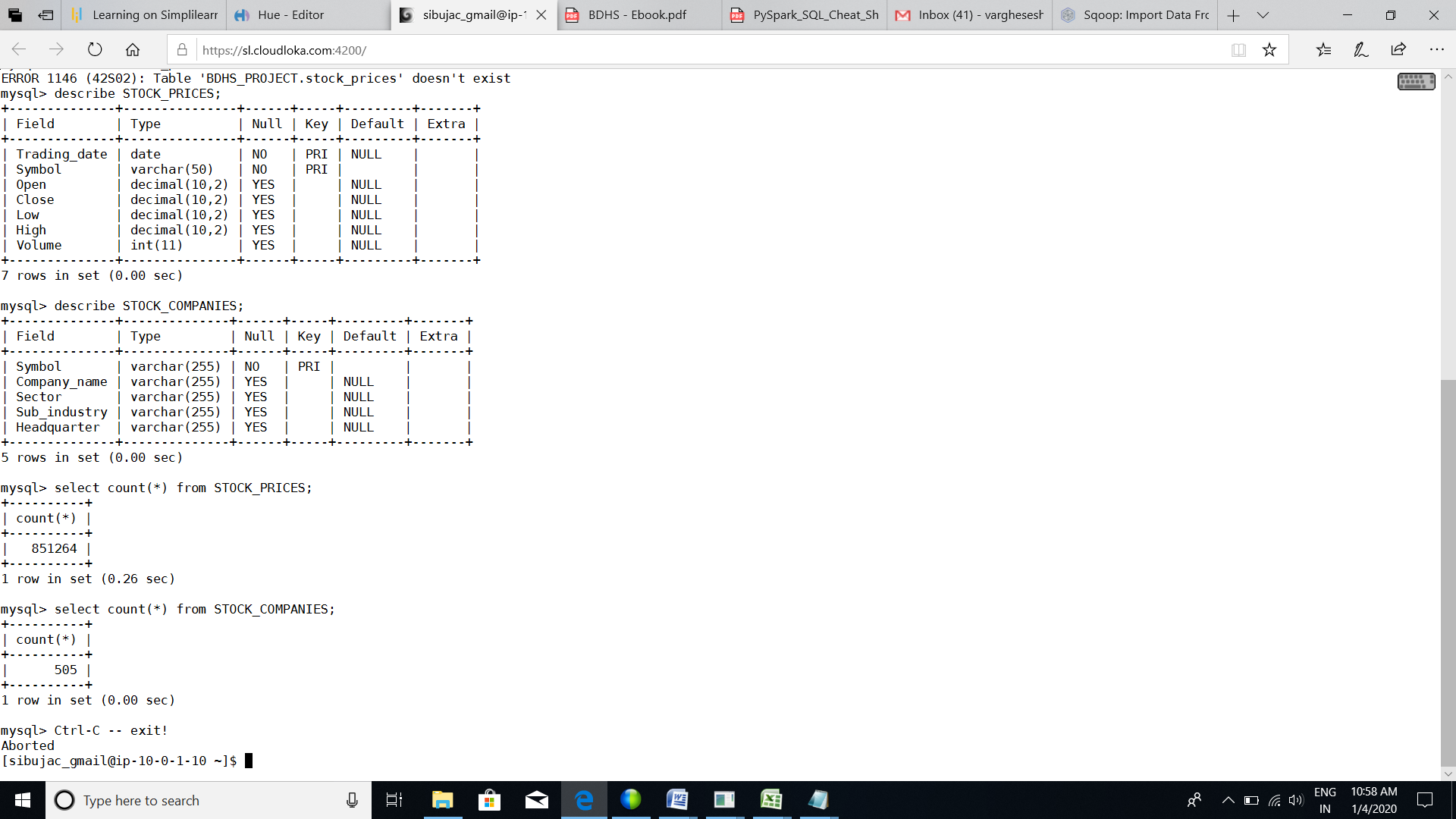
TABLE: STOCK\_COMPANIES

|  |  |
| --- | --- |
| Column Name | Datatype |
| Symbol | String |
| Company\_name | String |
| Sector | String |
| Sub\_industry | String |
| Headquarter | String |

Solution:-

Checking the mysql database, BDHS\_PROJECT, to check the table schema and row count of tables,





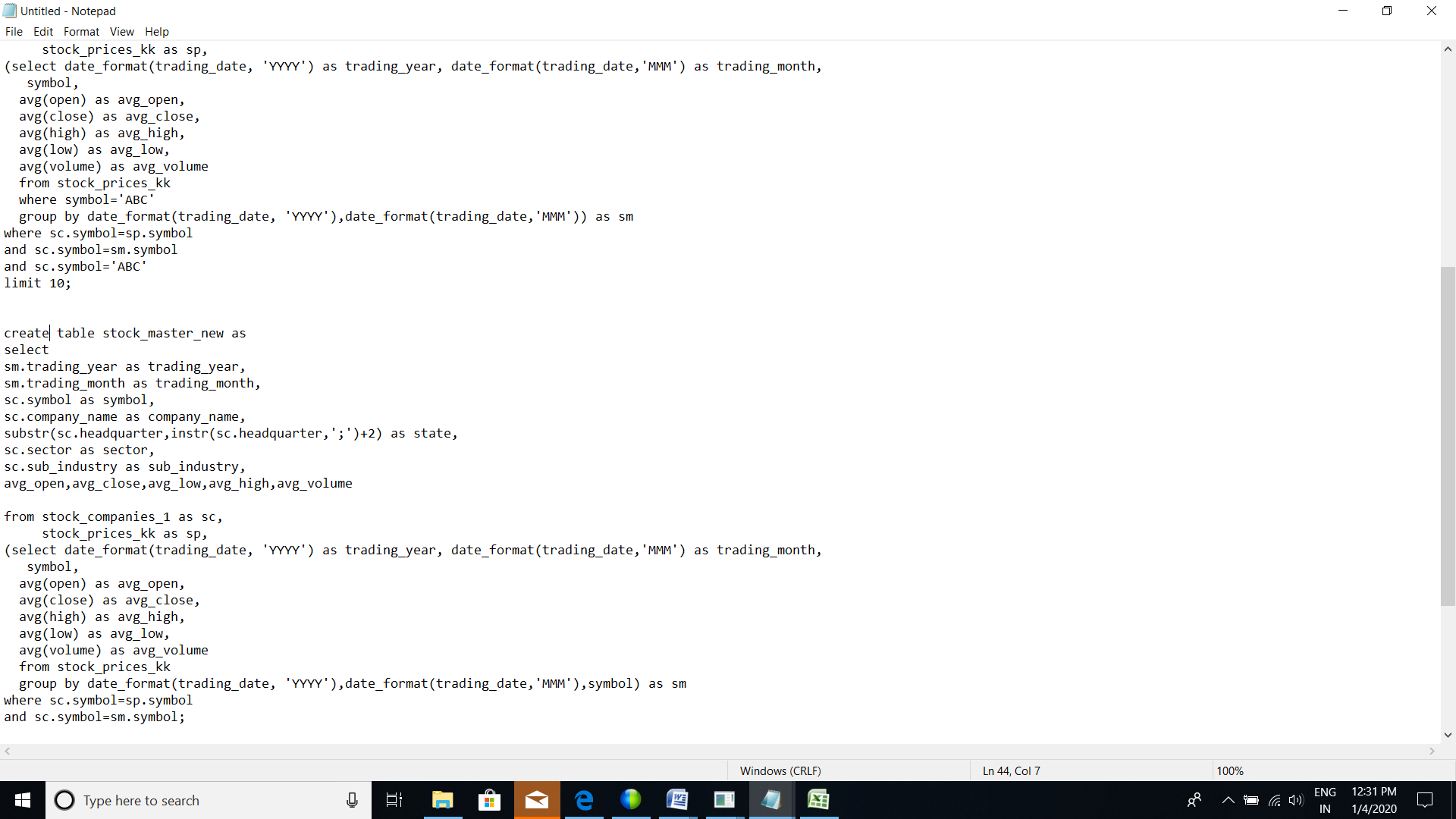
sqoop import --connect jdbc:mysql://ip-10-0-1-10.ec2.internal/BDHS\_PROJECT --username labuser --password simplilearn --table STOCK\_PRICES --hive-import --create-hive-table –hive-table sheeba\_123.stock\_prices --m 1

sqoop import --connect jdbc:mysql://ip-10-0-1-10.ec2.internal/BDHS\_PROJECT --username labuser --password simplilearn --table STOCK\_COMPANIES --hive-import --create-hive-table –hive-table sheeba\_123.stock\_companies --m 1

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

Solution:- ( Execute in hive)



create table stock\_master\_new as

select sm.trading\_year as trading\_year, sm.trading\_month as trading\_month,

sc.symbol as symbol,sc.company\_name as company\_name,

substr(sc.headquarter,instr(sc.headquarter,';')+2) as state,

sc.sector as sector,sc.sub\_industry as sub\_industry,

avg\_open,avg\_close,avg\_low,avg\_high,avg\_volume

from stock\_companies\_1 as sc, stock\_prices\_kk as sp,

(select date\_format(trading\_date, 'YYYY') as trading\_year, date\_format(trading\_date,'MMM') as trading\_month, symbol,avg(open) as avg\_open,avg(close) as avg\_close,avg(high) as avg\_high,

avg(low) as avg\_low,avg(volume) as avg\_volume

from stock\_prices\_kk

group by date\_format(trading\_date, 'YYYY'),date\_format(trading\_date,'MMM'),symbol) as sm

where sc.symbol=sp.symbol and sc.symbol=sm.symbol;

**DATA ANALYSIS USING HIVE**

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

Solution:-

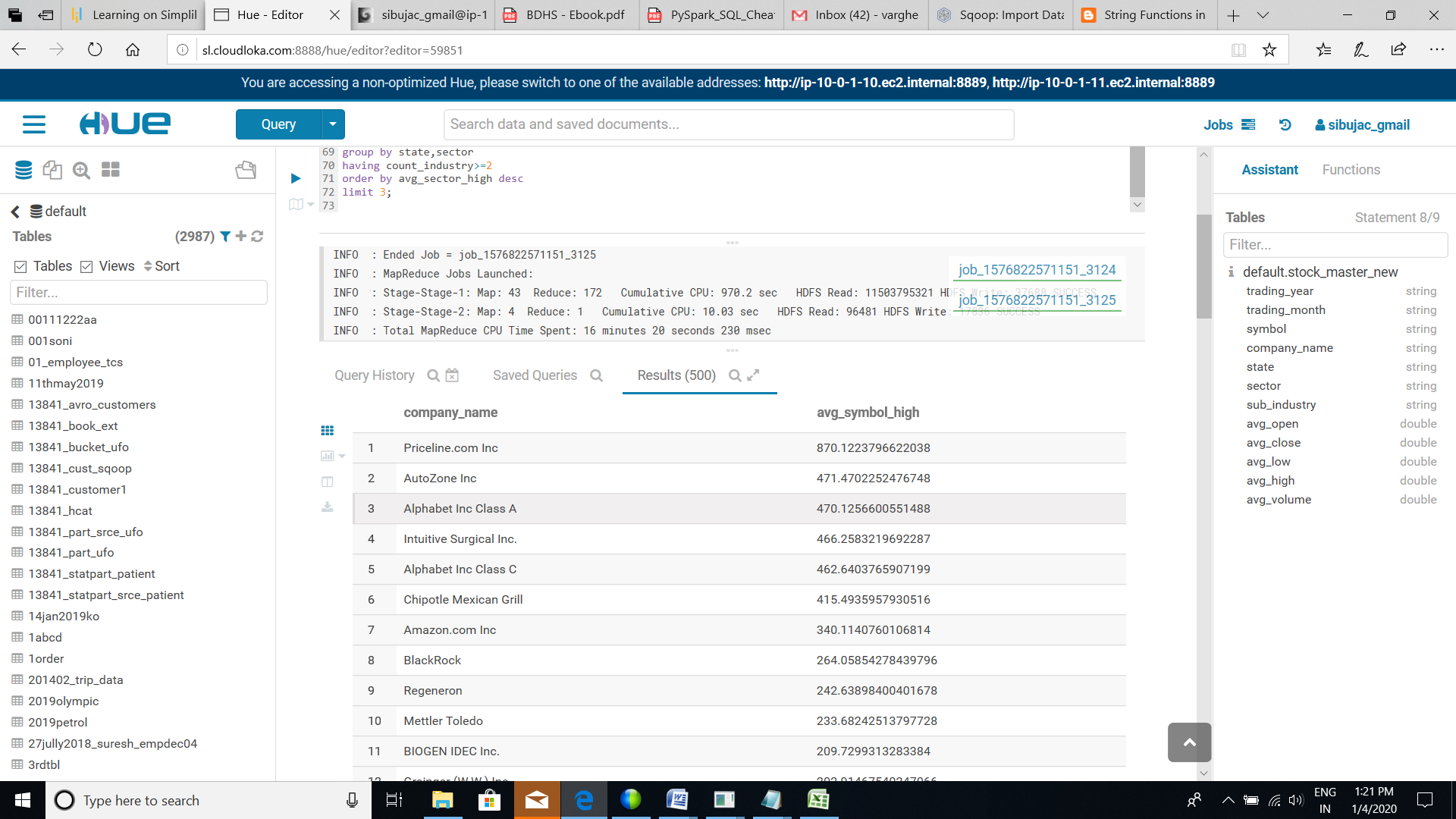
Select company\_name, avg(avg\_high) as avg\_symbol\_high

from stock\_master\_new

group by company\_name

order by avg\_symbol\_high desc

limit 5;



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

Solution:-

Select state,sector,

avg(avg\_high) avg\_sector\_high ,

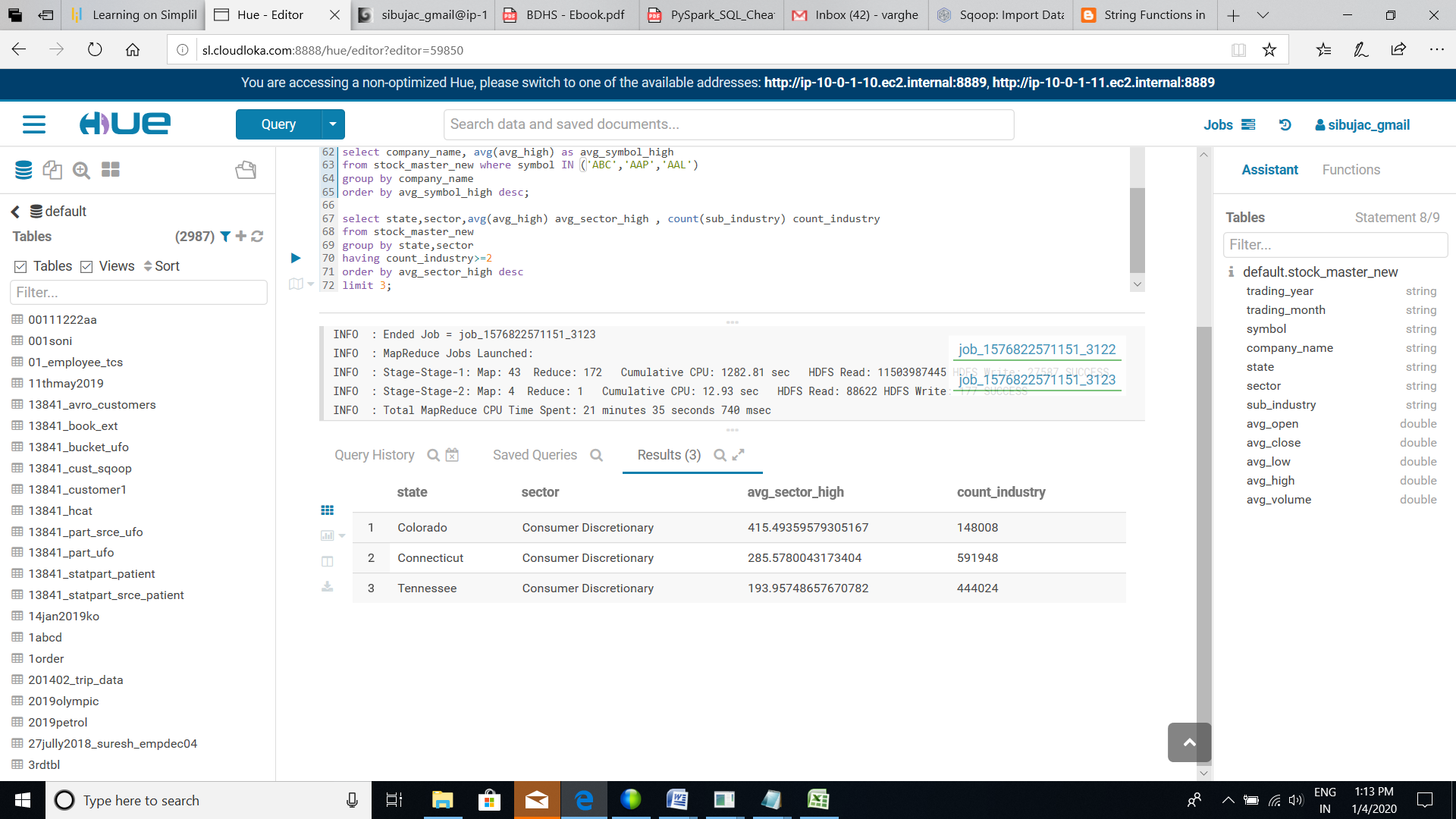
count(sub\_industry) count\_industry

from stock\_master\_new

group by state,sector

having count\_industry>=2

order by avg\_sector\_high desc;



5) For each sector find the following.

* Worst year

Solution:-

select sector, trading\_year worst\_year

from

(select sector, trading\_year,avg\_sector\_volume\_per\_year,

rank() over(partition by sector order by avg\_sector\_volume\_per\_year) as rn

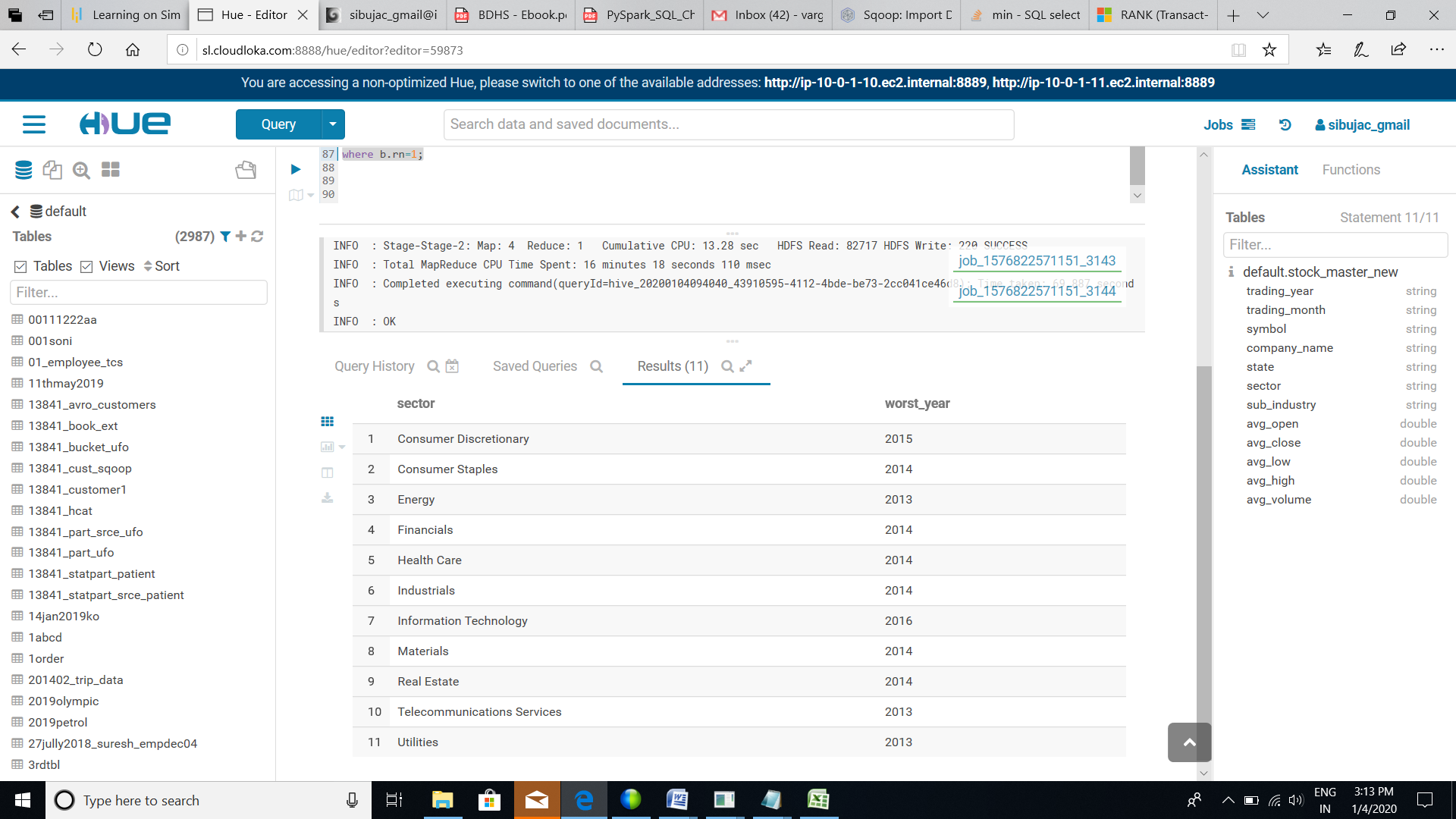
from

(select sector, trading\_year, avg(avg\_volume) avg\_sector\_volume\_per\_year

from stock\_master\_new

group by sector,trading\_year ) a) b

where b.rn=1;



* Best year

Solution:-

select sector, trading\_year best\_year

from

(select sector, trading\_year,avg\_sector\_volume\_per\_year,

rank() over(partition by sector order by avg\_sector\_volume\_per\_year desc) as rn

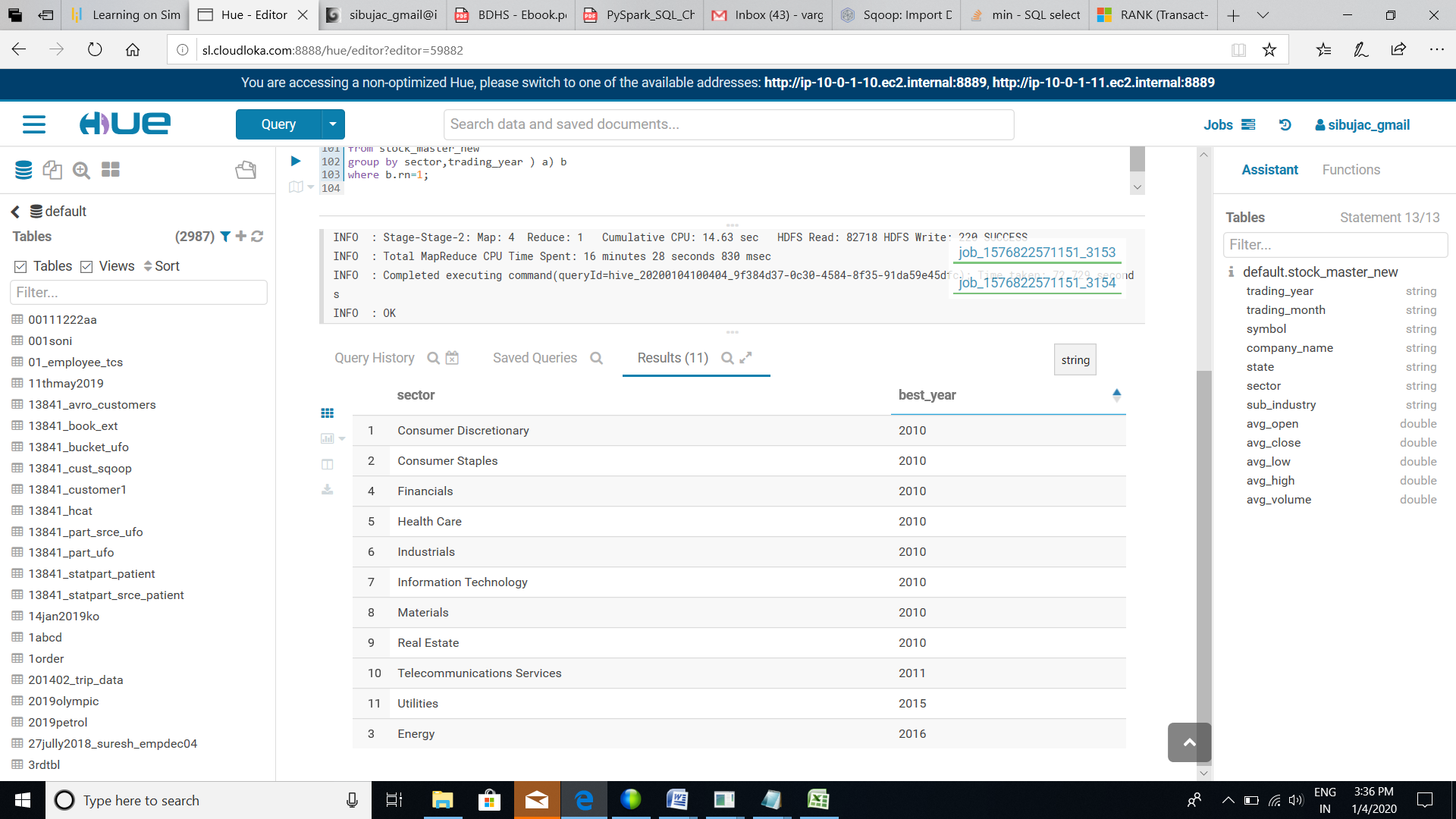
from

(select sector, trading\_year, avg(avg\_volume) avg\_sector\_volume\_per\_year

from stock\_master\_new

group by sector,trading\_year ) a) b

where b.rn=1;



* Stable year

Solution:-

Select sector, trading\_year stable\_year

from

(select sector, trading\_year,avg\_sector\_stability,

rank() over(partition by sector order by avg\_sector\_stability) as rn

from

(select sector, trading\_year, avg(avg\_high-avg\_low) avg\_sector\_stability

from stock\_master\_new

group by sector,trading\_year ) a) b

where b.rn=1;

