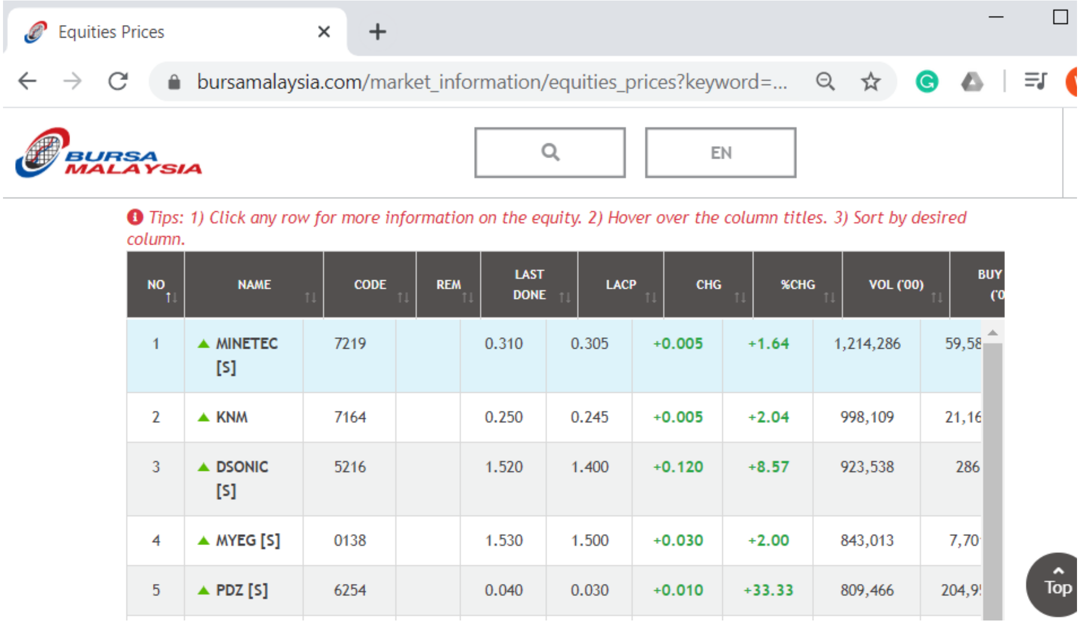
Name: Mohd Amirul Shafiq Bin Shafiee
Matric No: WQD180114
Milestone 1: Web Crawling Real Time Data by Using Python

Step 1 – Get List of Symbol for All Company in KLSE Main Market

- We will extract the data from Bursa Malaysia official website.
    - https://www.bursamalaysia.com/market_information/equities_prices?keyword=&top_stock=&board=MAIN-MKT&alphabetical=&sector=&sub_sector=&page=1
    - This website is chosen because based on random observation, other site such as Yahoo Finance does not list all the symbols.

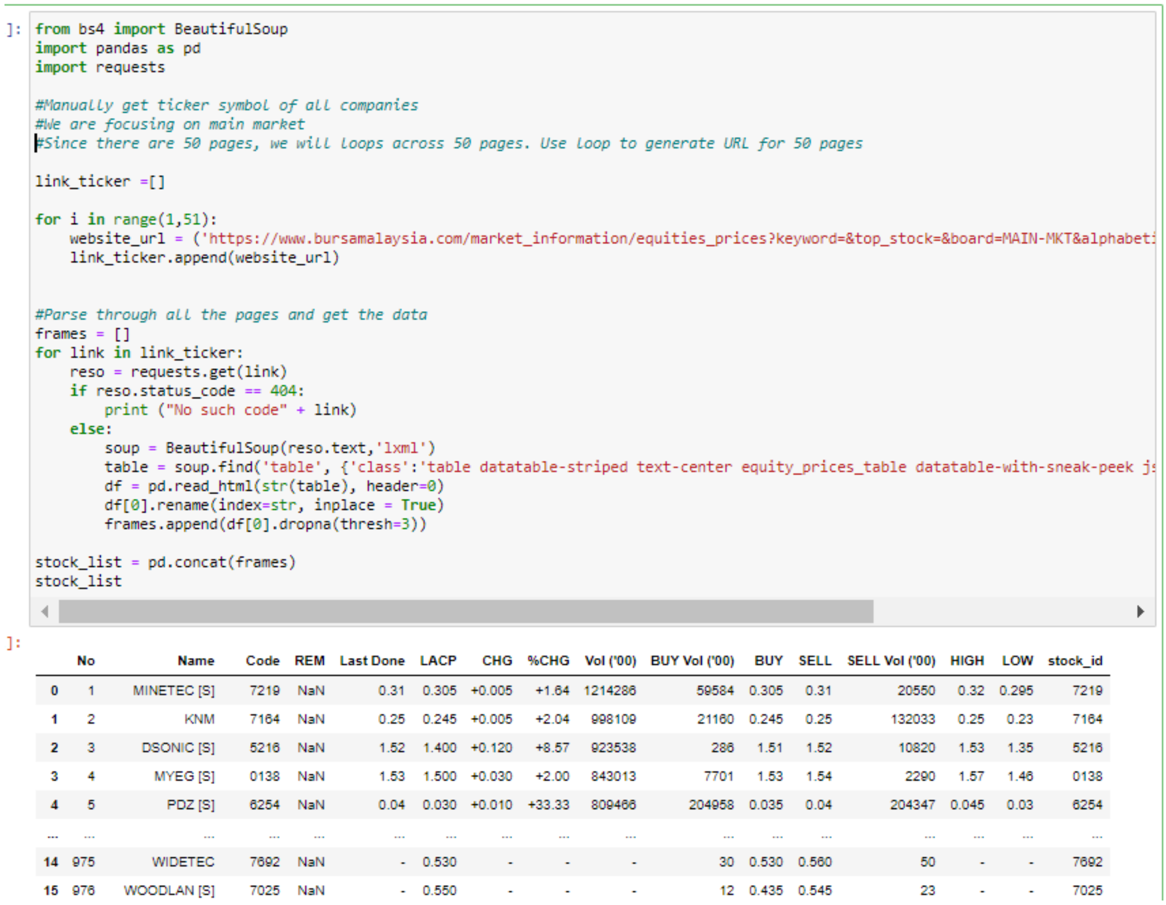

Equities Prices
bursamalaysia.com/market_information/equities_prices?keyword=...
BURSA MALAYSIA
EN
Top

Tips: 1) Click any row for more information on the equity. 2) Hover over the column titles. 3) Sort by desired column.

NO	NAME	CODE	REM	LAST DONE	LACP	CHG	%CHG	VOL ('00)	BUY ('0
1	▲ MINETEC [S]	7219		0.310	0.305	+0.005	+1.64	1,214,286	59,58
2	▲ KNM	7164		0.250	0.245	+0.005	+2.04	998,109	21,16
3	▲ DSONIC [S]	5216		1.520	1.400	+0.120	+8.57	923,538	286
4	▲ MYEG [S]	0138		1.530	1.500	+0.030	+2.00	843,013	7,70
5	▲ PDZ [S]	6254		0.040	0.030	+0.010	+33.33	809,466	204,9

- Import the necessary packages
    - pandas
    - Beautiful Soup

- As the data are in 50 pages, we will have to loop through all the pages.



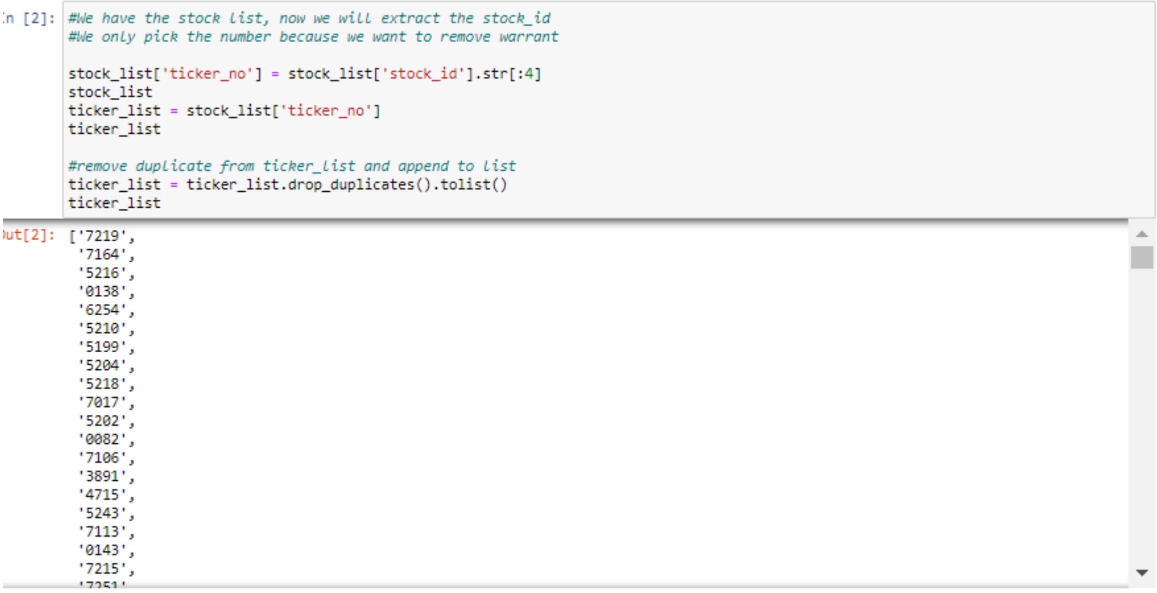
```
]: from bs4 import BeautifulSoup
   import pandas as pd
   import requests

   #Manually get ticker symbol of all companies
   #We are focusing on main market
   #Since there are 50 pages, we will loops across 50 pages. Use loop to generate URL for 50 pages

   link_ticker =[]

   for i in range(1,51):
       website_url = ('https://www.bursamalaysia.com/market_information/equities_prices?keyword=&top_stock=&board=MAIN-MKT&alphabet
       link_ticker.append(website_url)


   #Parse through all the pages and get the data
   frames = []
   for link in link_ticker:
       reso = requests.get(link)
       if reso.status_code == 404:
           print ("No such code" + link)
       else:
           soup = BeautifulSoup(reso.text,'lxml')
           table = soup.find('table', {'class':'table datatable-striped text-center equity_prices_table datatable-with-sneak-peek j
           df = pd.read_html(str(table), header=0)
           df[0].rename(index=str, inplace = True)
           frames.append(df[0].dropna(thresh=3))

   stock_list = pd.concat(frames)
   stock_list
```

]:

	No	Name	Code	REM	Last Done	LACP	CHG	%CHG	Vol ('00)	BUY Vol ('00)	BUY	SELL	SELL Vol ('00)	HIGH	LOW	stock_id
0	1	MINETEC [S]	7219	NaN	0.31	0.305	+0.005	+1.64	1214286	59584	0.305	0.31	20550	0.32	0.295	7219
1	2	KNM	7164	NaN	0.25	0.245	+0.005	+2.04	998109	21160	0.245	0.25	132033	0.25	0.23	7164
2	3	DSONIC [S]	5216	NaN	1.52	1.400	+0.120	+8.57	923538	286	1.51	1.52	10820	1.53	1.35	5216
3	4	MYEG [S]	0138	NaN	1.53	1.500	+0.030	+2.00	843013	7701	1.53	1.54	2290	1.57	1.46	0138
4	5	PDZ [S]	6254	NaN	0.04	0.030	+0.010	+33.33	809466	204958	0.035	0.04	204347	0.045	0.03	6254
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14	975	WIDETEC	7692	NaN	-	0.530	-	-	-	30	0.530	0.560	50	-	-	7692
15	976	WOODLAN [S]	7025	NaN	-	0.550	-	-	-	12	0.435	0.545	23	-	-	7025

Part 2 – Extract the Symbols Into a list

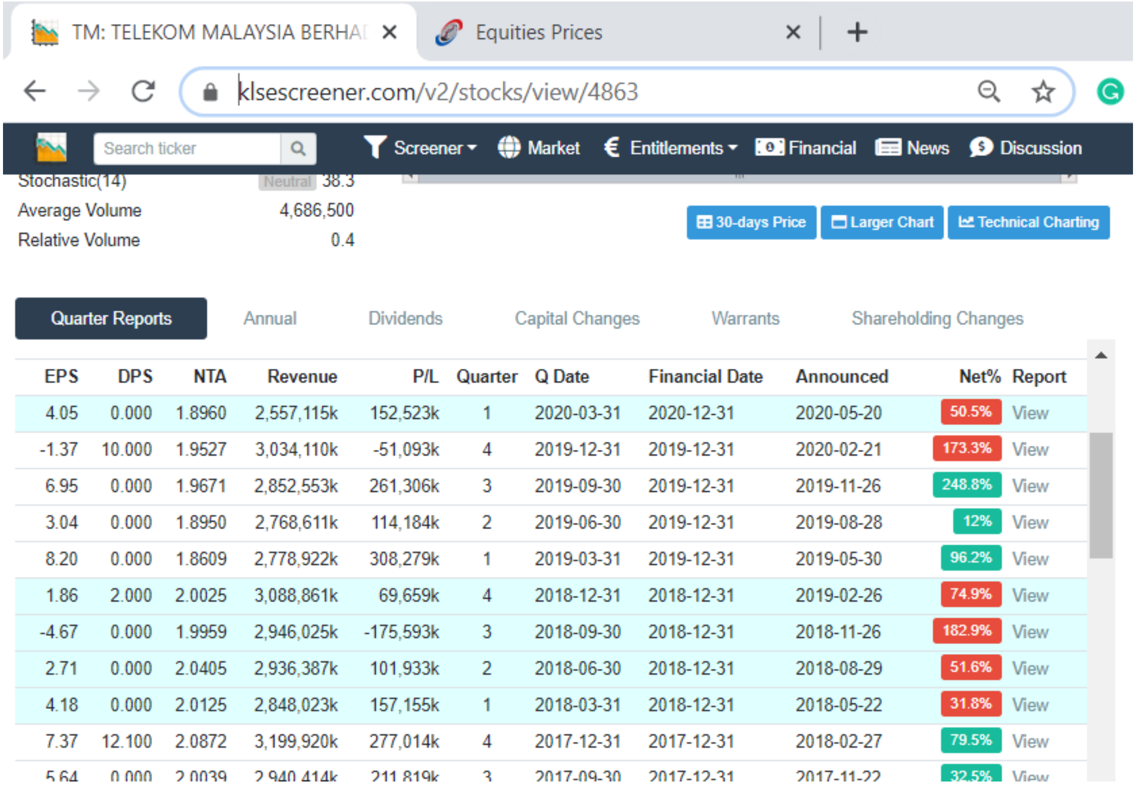
- Get all the symbols and put into a list



```
In [2]: #We have the stock list, now we will extract the stock_id
        #We only pick the number because we want to remove warrant

        stock_list['ticker_no'] = stock_list['stock_id'].str[:4]
        stock_list
        ticker_list = stock_list['ticker_no']
        ticker_list

        #remove duplicate from ticker_list and append to list
        ticker_list = ticker_list.drop_duplicates().tolist()
        ticker_list

Out[2]: ['7219',
         '7164',
         '5216',
         '0138',
         '6254',
         '5210',
         '5199',
         '5204',
         '5218',
         '7017',
         '5202',
         '0082',
         '7106',
         '3891',
         '4715',
         '5243',
         '7113',
         '0143',
         '7215',
         '7251'
```

Part 3 – Scrape the Financial Data for All Symbols

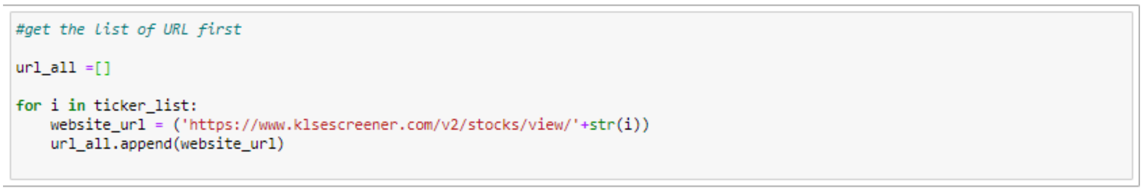
- For this part, we will data from this site:
  - https://www.klsescreener.com/v2/stocks/view/4863
  - We use this site as the official Bursa Malaysia did not table out all the company's performance.
  - Data for each symbol is different page.



Stochastic(14)	Neutral	38.3
Average Volume		4,686,500
Relative Volume		0.4

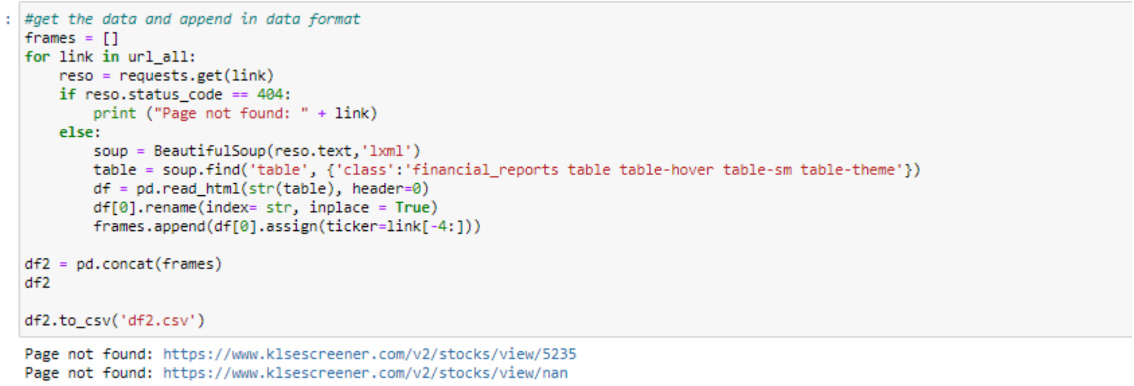
EPS	DPS	NTA	Revenue	P/L	Quarter	Q Date	Financial Date	Announced	Net%	Report
4.05	0.000	1.8960	2,557,115k	152,523k	1	2020-03-31	2020-12-31	2020-05-20	50.5%	View
-1.37	10.000	1.9527	3,034,110k	-51,093k	4	2019-12-31	2019-12-31	2020-02-21	173.3%	View
6.95	0.000	1.9671	2,852,553k	261,306k	3	2019-09-30	2019-12-31	2019-11-26	248.8%	View
3.04	0.000	1.8950	2,768,611k	114,184k	2	2019-06-30	2019-12-31	2019-08-28	12%	View
8.20	0.000	1.8609	2,778,922k	308,279k	1	2019-03-31	2019-12-31	2019-05-30	96.2%	View
1.86	2.000	2.0025	3,088,861k	69,659k	4	2018-12-31	2018-12-31	2019-02-26	74.9%	View
-4.67	0.000	1.9959	2,946,025k	-175,593k	3	2018-09-30	2018-12-31	2018-11-26	182.9%	View
2.71	0.000	2.0405	2,936,387k	101,933k	2	2018-06-30	2018-12-31	2018-08-29	51.6%	View
4.18	0.000	2.0125	2,848,023k	157,155k	1	2018-03-31	2018-12-31	2018-05-22	31.8%	View
7.37	12.100	2.0872	3,199,920k	277,014k	4	2017-12-31	2017-12-31	2018-02-27	79.5%	View
5.64	0.000	2.0039	2,940,414k	211,819k	3	2017-09-30	2017-12-31	2017-11-22	32.5%	View

- So, we have to create a list of URL to be crawled. We can do this by performing string operation and use the symbol we compiled in Part 2.

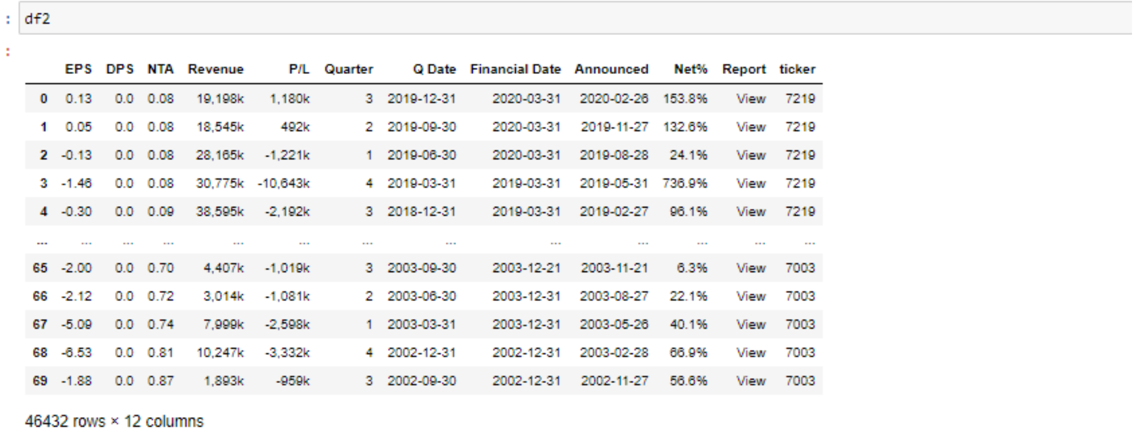


```
#get the list of URL first

url_all =[]

for i in ticker_list:
    website_url = ('https://www.klsescreener.com/v2/stocks/view/'+str(i))
    url_all.append(website_url)
```

- Then, we will crawl the data. This will take times as the script will have to crawl over 900 pages.



```
:   #get the data and append in data format
    frames = []
    for link in url_all:
        reso = requests.get(link)
        if reso.status_code == 404:
            print ("Page not found: " + link)
        else:
            soup = BeautifulSoup(reso.text,'lxml')
            table = soup.find('table', {'class':'financial_reports table table-hover table-sm table-theme'})
            df = pd.read_html(str(table), header=0)
            df[0].rename(index= str, inplace = True)
            frames.append(df[0].assign(ticker=link[-4:]))

    df2 = pd.concat(frames)
    df2

    df2.to_csv('df2.csv')

    Page not found: https://www.klsescreener.com/v2/stocks/view/5235
    Page not found: https://www.klsescreener.com/v2/stocks/view/nan
```



```
:   df2
```

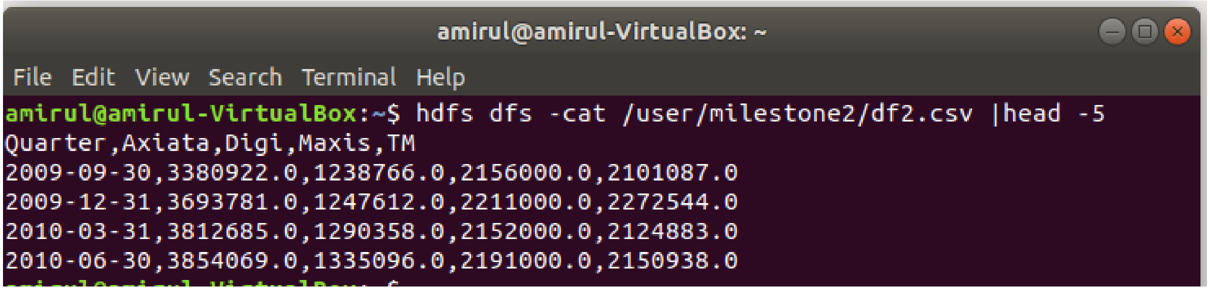
	EPS	DPS	NTA	Revenue	P/L	Quarter	Q Date	Financial Date	Announced	Net%	Report	ticker
0	0.13	0.0	0.08	19,198k	1,180k	3	2019-12-31	2020-03-31	2020-02-26	153.8%	View	7219
1	0.05	0.0	0.08	18,545k	492k	2	2019-09-30	2020-03-31	2019-11-27	132.6%	View	7219
2	-0.13	0.0	0.08	28,165k	-1,221k	1	2019-06-30	2020-03-31	2019-08-28	24.1%	View	7219
3	-1.46	0.0	0.08	30,775k	-10,643k	4	2019-03-31	2019-03-31	2019-05-31	736.9%	View	7219
4	-0.30	0.0	0.09	38,595k	-2,192k	3	2018-12-31	2019-03-31	2019-02-27	96.1%	View	7219
...	...	...	...	...	...	...	...	...	...	...	...	...
65	-2.00	0.0	0.70	4,407k	-1,019k	3	2003-09-30	2003-12-21	2003-11-21	6.3%	View	7003
66	-2.12	0.0	0.72	3,014k	-1,081k	2	2003-06-30	2003-12-31	2003-08-27	22.1%	View	7003
67	-5.09	0.0	0.74	7,999k	-2,598k	1	2003-03-31	2003-12-31	2003-05-26	40.1%	View	7003
68	-6.53	0.0	0.81	10,247k	-3,332k	4	2002-12-31	2002-12-31	2003-02-28	66.9%	View	7003
69	-1.88	0.0	0.87	1,893k	-959k	3	2002-09-30	2002-12-31	2002-11-27	56.6%	View	7003

46432 rows × 12 columns

- Then, we will store the data into a .csv file for next step.

Step 3 – Check the Data

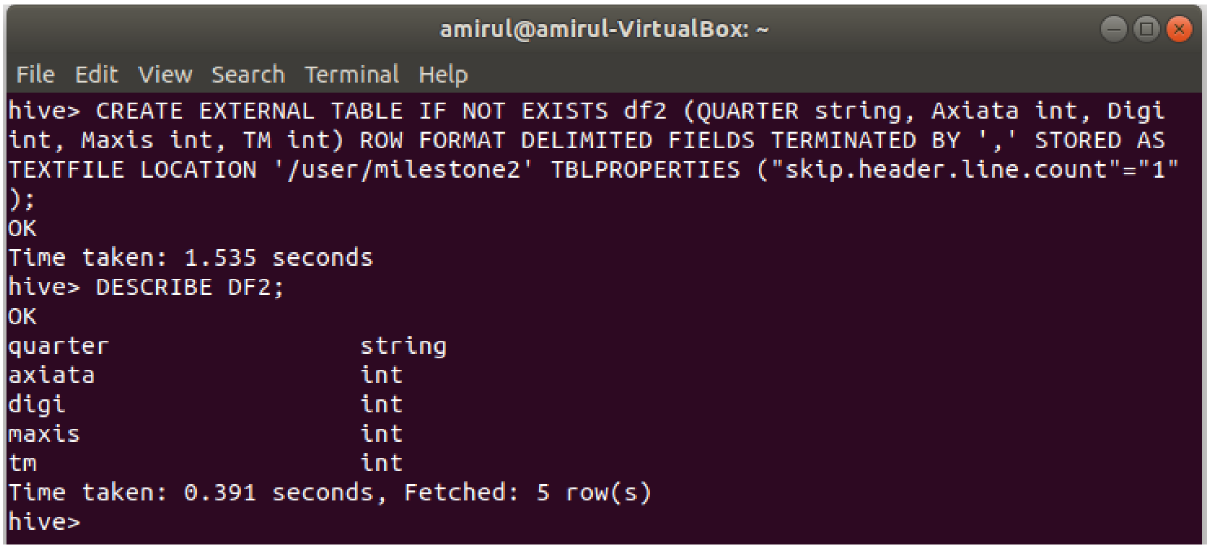
- Read the head of the data


amirul@amirul-VirtualBox: ~
File Edit View Search Terminal Help

```
amirul@amirul-VirtualBox:~$ hdfs dfs -cat /user/milestone2/df2.csv |head -5
Quarter,Axiata,Digi,Maxis,TM
2009-09-30,3380922.0,1238766.0,2156000.0,2101087.0
2009-12-31,3693781.0,1247612.0,2211000.0,2272544.0
2010-03-31,3812685.0,1290358.0,2152000.0,2124883.0
2010-06-30,3854069.0,1335096.0,2191000.0,2150938.0
amirul@amirul-VirtualBox:~$
```

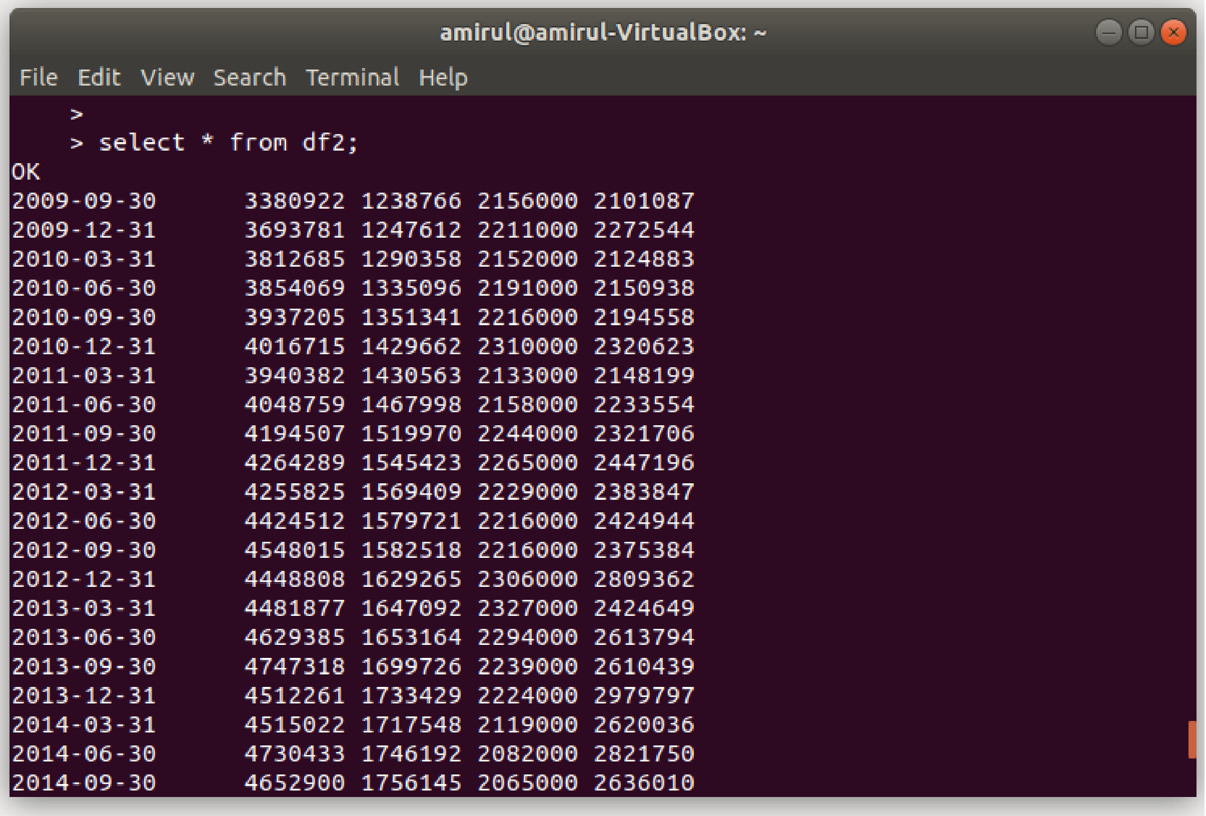
Step 4 – Create table in Hive

- Use query to create table with 5 columns and define the data type
- Remove first row as it is the column names


amirul@amirul-VirtualBox: ~
File Edit View Search Terminal Help

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS df2 (QUARTER string, Axiata int, Digi
int, Maxis int, TM int) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
TEXTFILE LOCATION '/user/milestone2' TBLPROPERTIES ("skip.header.line.count"="1"
);
OK
Time taken: 1.535 seconds
hive> DESCRIBE DF2;
OK
quarter                 string
axiata                  int
digi                    int
maxis                   int
tm                      int
Time taken: 0.391 seconds, Fetched: 5 row(s)
hive>
```

Step 5 – Select all data in DF2 in HIVE



amirul@amirul-VirtualBox: ~
File Edit View Search Terminal Help

```
    >
    > select * from df2;
OK
2009-09-30      3380922 1238766 2156000 2101087
2009-12-31      3693781 1247612 2211000 2272544
2010-03-31      3812685 1290358 2152000 2124883
2010-06-30      3854069 1335096 2191000 2150938
2010-09-30      3937205 1351341 2216000 2194558
2010-12-31      4016715 1429662 2310000 2320623
2011-03-31      3940382 1430563 2133000 2148199
2011-06-30      4048759 1467998 2158000 2233554
2011-09-30      4194507 1519970 2244000 2321706
2011-12-31      4264289 1545423 2265000 2447196
2012-03-31      4255825 1569409 2229000 2383847
2012-06-30      4424512 1579721 2216000 2424944
2012-09-30      4548015 1582518 2216000 2375384
2012-12-31      4448808 1629265 2306000 2809362
2013-03-31      4481877 1647092 2327000 2424649
2013-06-30      4629385 1653164 2294000 2613794
2013-09-30      4747318 1699726 2239000 2610439
2013-12-31      4512261 1733429 2224000 2979797
2014-03-31      4515022 1717548 2119000 2620036
2014-06-30      4730433 1746192 2082000 2821750
2014-09-30      4652900 1756145 2065000 2636010
```