WQD7005 Data Mining

Assignment – Milestone 2

Name: Ng Kang Wei

Student ID: WQD170068

Video presentation: https://youtu.be/veRR8RlvpMQ (Please turn on caption or subtitle)

In milestone 1, I have showed the scraping of daily stock prices data from the star website. Besides that, my group members and I also scraped other related information from other sources.

The star website does provide some extra information other than the daily stock prices. We collected the information as well, namely company income, financial results of the company, share buyback conducted by the company, the dividend paid by the company.

The information from other sources include currency rate, as we think currency exchange rate might relate to the stock price changes in some way. We get this from Bank Negara Malaysia website. Besides structured data, we also crawled some unstructured data such as news and twitter.

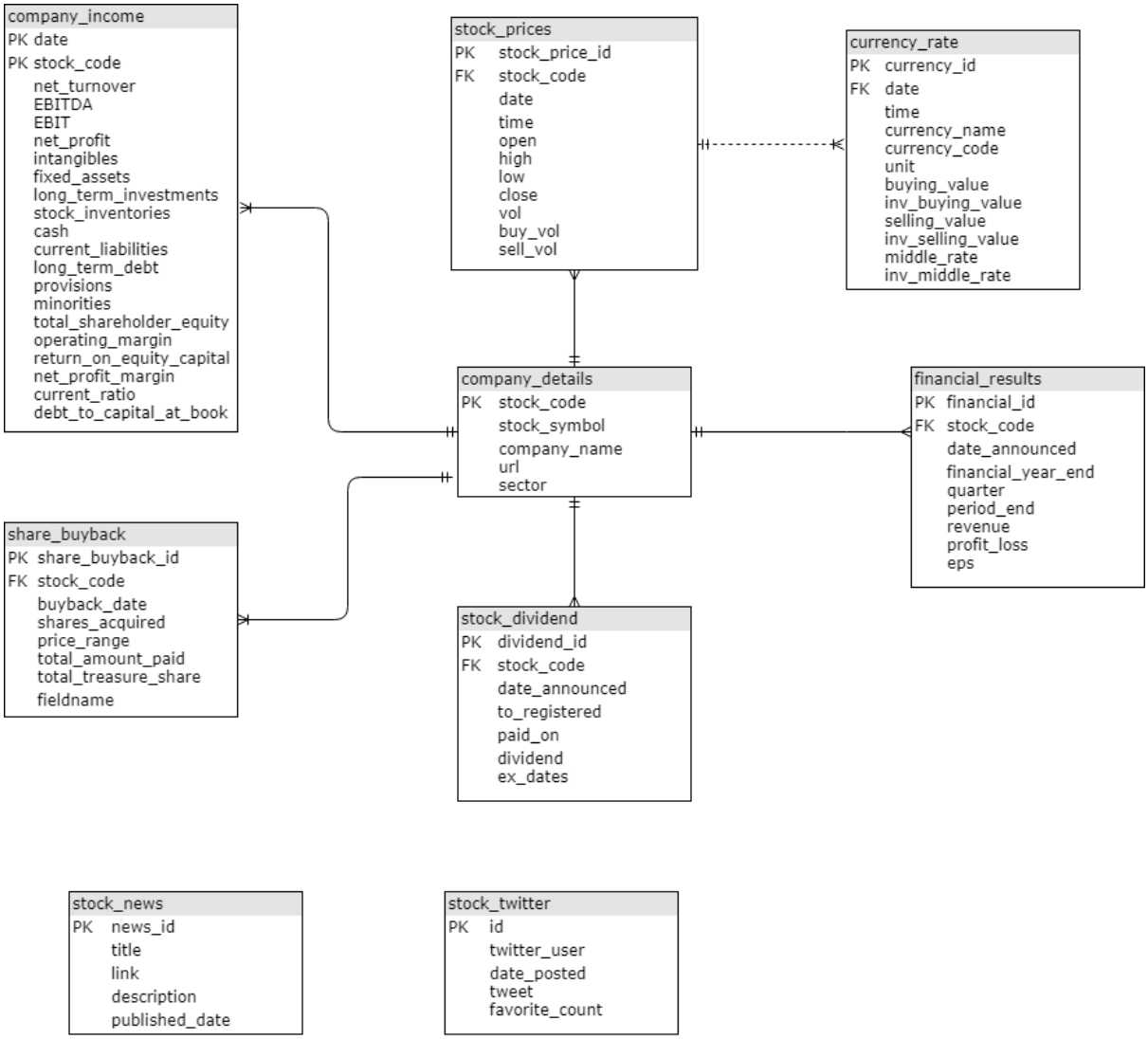
Database tables

1. company_details
2. stock_prices
3. company_income
4. currency_rate
5. financial_results
6. share_buyback
7. stock_dividend
8. stock_news

9. stock_twitter

Entity Relation Diagram (ERD)

The tables are related to each other in the ERD diagram shown below:



Company details would be the entity at the center. It relates to all the tables via stock_code primary key, except for stock_news and stock_twitter.

The information collected in stock_news and stock_twitter is not related to a stock for now, so there is no a foreign key stock_code in the tables.

Currency_rate relate to the stock_prices table via the date. As the pattern of the fluctuations of the currency exchange rate might have an effect on the movement or fluctuations of the stock prices.

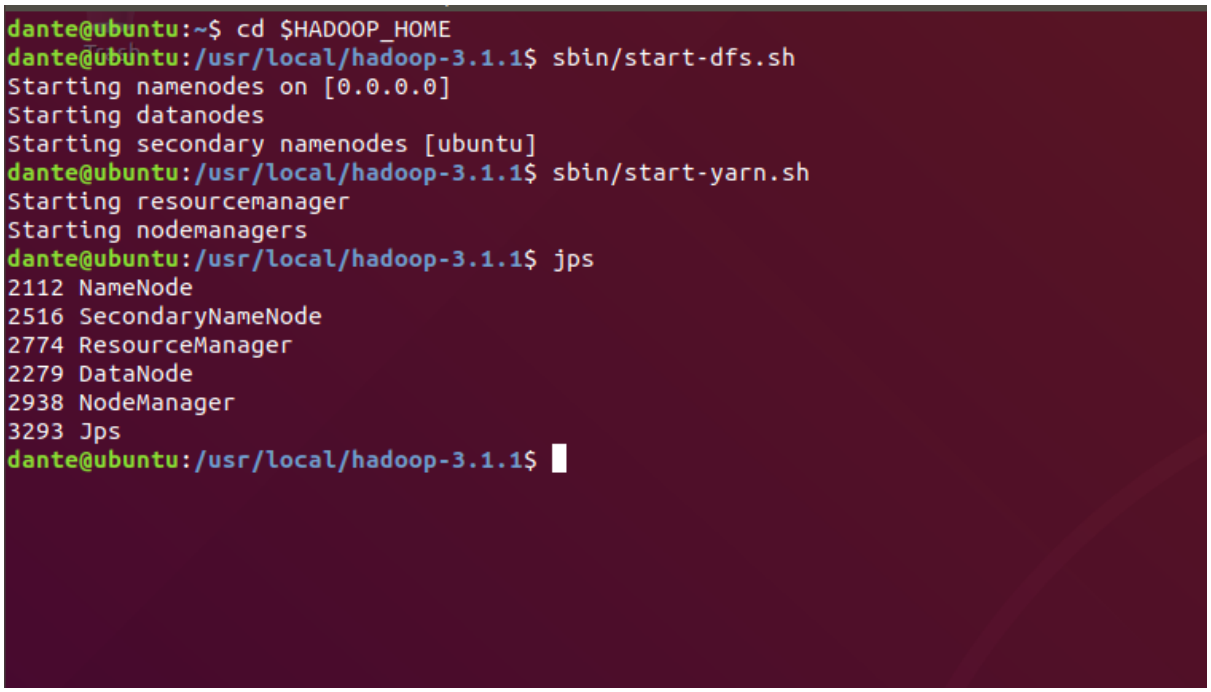
Data import into Apache HIVE

Apache Hadoop and Apache HIVE need to be installed and working properly before proceeding to this part.

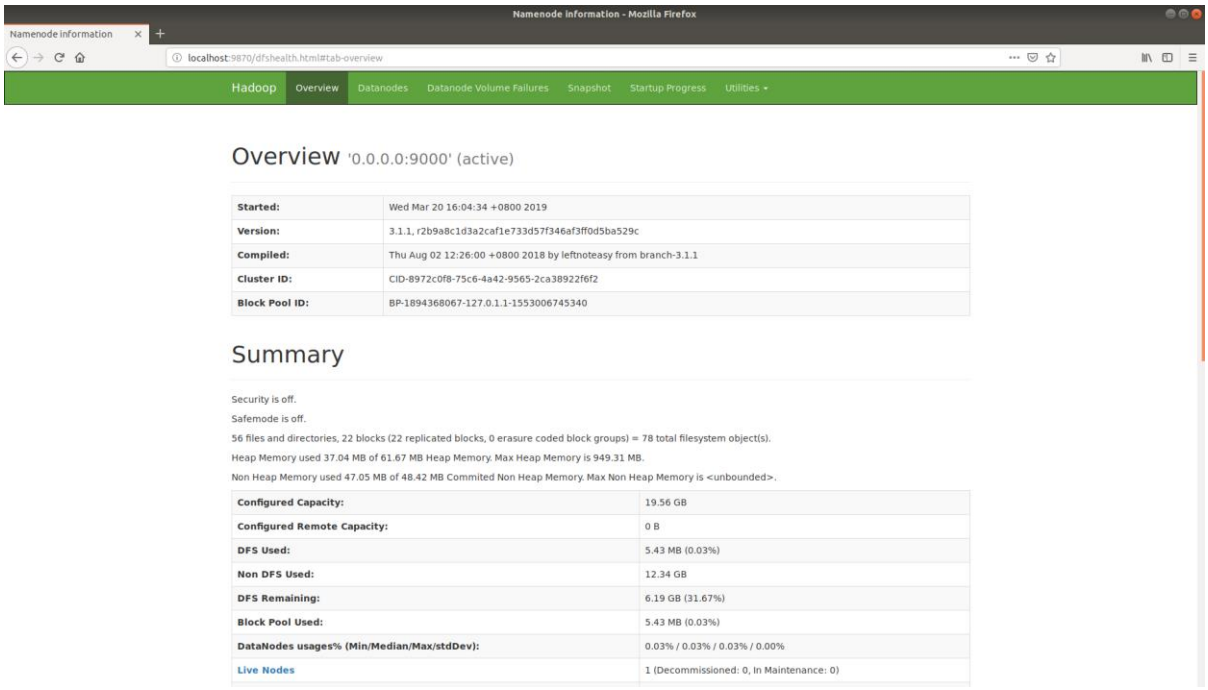
I am running my Apache Hadoop and Apache HIVE on Ubuntu Linux in a virtual machine.

First, the data in PostgreSQL database must be exported to CSV files. Then, a new database is created in Apache HIVE. After that, I create the tables in Apache HIVE and import the data into HIVE. Now I can view and query the data in Apache HIVE with SQL-like statements.

Before running hive, Hadoop HDFS and YARN need to be started.



Hadoop HDFS is verified to be started and running via the dashboard.



After Hadoop HDFS and YARN is started and running. We can run the hive command.

```
dante@ubuntu:/usr/local/hadoop-3.1.1$ cd
dante@ubuntu:~$ cd development/
dante@ubuntu:~/development$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.0/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.1/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = ed6950f3-77f0-4c77-a3fe-439418fb1113

Logging initialized using configuration in jar:file:/usr/local/apache-hive-3.1.0/lib/hive-common-3.1.0.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive Session ID = 09fbc0e0-5e21-4854-ab71-87779bcfb601
hive>
```

Create the table and import the data from csv into the table in HIVE

```
hive> DROP TABLE IF EXISTS company_details;
OK
Time taken: 1.55 seconds
hive>     CREATE TABLE company_details (
    >         stock_code string,
    >         stock_symbol string,
    >         company_name string,
    >         url string,
    >         sector string
    >     ) row format delimited fields terminated by ',';
OK
Time taken: 0.513 seconds
hive> LOAD DATA LOCAL INPATH '/home/dante/development/stock_data/company_details.csv' OVERWRITE INTO TABLE company_details;
Loading data to table datamining.company_details
OK
Time taken: 1.18 seconds
hive>
```

All the data is imported into Apache HIVE