

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN CUỐI KÌ

***Môn: CÔNG NGHỆ MỚI TRONG PHÁT TRIỂN ỨNG DỤNG
CÔNG NGHỆ THÔNG TIN***

Đề tài: XÂY DỰNG HỆ THỐNG THÔNG TIN TÀI CHÍNH
(PHẦN CỔ PHIẾU)

Giảng viên hướng dẫn : Nguyễn Chí Kiên
Lớp học phần : DHKHDL15A

Thành viên

STT	Họ tên	MSSV
1	Nguyễn Tuấn Sinh	19477821
2	Trương Thị Cẩm Ly	19532211

Thành phố Hồ Chí Minh, ngày 25 tháng 12 năm 2022

PHẦN 1. GIỚI THIỆU

PHẦN 2. TỔNG QUAN ĐỀ TÀI

CHƯƠNG I. PHƯƠNG PHÁP TIẾP CẬN

CHƯƠNG II. THU THẬP VÀ XỬ LÝ DỮ LIỆU

CHƯƠNG III. THỰC HIỆN

1. Cài đặt Docker
2. Cài đặt Superset trên Docker
3. Cài đặt và kết nối PostgreSQL trên Docker
4. Tạo Table bằng SQL
5. Thêm dữ liệu vào database
6. Connect database với superset
7. Vẽ Dashboard

CHƯƠNG IV. KẾT LUẬN

CHƯƠNG V. HƯỚNG NGHIÊN CỨU TRONG TƯƠNG LAI

PHẦN 1. GIỚI THIỆU.

Phân tích chứng khoán là nhu cầu không thể thiếu trong đầu tư chứng khoán. Tùy theo khả năng trình độ, thời gian cũng như nhu cầu sử dụng, có thể có rất nhiều cách tiếp cận nghiên cứu, phân tích và ra các quyết định đầu tư khác nhau đối với từng chứng khoán riêng lẻ hoặc đối với cả danh mục đầu tư nói chung.

Cổ phiếu là một loại chứng khoán chủ yếu cần phân tích phục vụ đầu tư. Đến nay người ta đã tổng kết lại rằng có hai phương pháp phân tích đã được sử dụng một cách phổ biến tại hầu hết các thị trường chứng khoán (TTCK) trên thế giới, đó là phân tích cơ bản và phân tích kỹ thuật. Tuy nhiên, không ít nhà đầu tư còn nhầm lẫn về vai trò cũng như ứng dụng thực tiễn của hai phương pháp này. Vì vậy, việc hiểu rõ bản chất của hai phương pháp là cần thiết.

Phân tích cơ bản, phân tích kỹ thuật hay chỉ cần dựa trên những thông tin cập nhật về cổ phiếu được công bố nhà đầu tư đã có thể đạt được thành công lớn.

PHẦN 2. TỔNG QUAN ĐỀ TÀI

CHƯƠNG I. PHƯƠNG PHÁP TIẾP CẬN

1. Docker

a) Khái niệm về Docker

Docker là một nền tảng cho developers và system admin để develop, deploy và run application với container. Nó cho phép tạo các môi trường độc lập và tách biệt để khởi chạy và phát triển ứng dụng và môi trường này được gọi là container. Khi cần deploy lên bất kỳ server nào chỉ cần run container của Docker thì application của bạn sẽ được khởi chạy ngay lập tức.

b) Lợi ích của Docker

- Không như máy ảo, Docker start và stop chỉ trong vài giây
- Có thể chạy container trên mỗi hệ thống mình muốn
- Container có thể build và loại bỏ nhanh hơn máy ảo
- Dễ dàng thiết lập môi trường làm việc. Chỉ cần config 1 lần duy nhất và không bao giờ phải cài đặt lại các dependencies. Nếu bạn thay đổi máy hoặc có người mới tham gia vào project thì chỉ cần lấy config đó và đưa cho họ

c) Các khái niệm khác về Docker

- **Docker Client:** là cách mà bạn tương tác với docker thông qua command trong terminal. Docker Client sẽ sử dụng API gửi lệnh tới Docker Daemon.
- **Docker Daemon:** là server Docker cho yêu cầu từ Docker API. Nó quản lý images, containers, networks và volume.
- **Docker Volumes:** là cách tốt nhất để lưu trữ dữ liệu liên tục cho việc sử dụng và tạo apps.
- **Docker Registry:** là nơi lưu trữ riêng của Docker Images. Images được push vào registry và client sẽ pull images từ registry. Có thể sử dụng registry của riêng bạn hoặc registry của nhà cung cấp như : AWS, Google Cloud, Microsoft Azure.
- **Docker Hub:** là Registry lớn nhất của Docker Images (mặc định). Có thể tìm thấy images và lưu trữ images của riêng bạn trên Docker Hub (miễn phí).
- **Docker Repository:** là tập hợp các Docker Images cùng tên nhưng khác tags. VD: golang:1.11-alpine.
- **Docker Networking:** cho phép kết nối các container lại với nhau. Kết nối này có thể trên 1 host hoặc nhiều host.
- **Docker Compose:** là công cụ cho phép run app với nhiều Docker containers 1 cách dễ dàng hơn. Docker Compose cho phép bạn config các command trong file docker-compose.yml để sử dụng lại. Có sẵn khi cài Docker.
- **Docker Swarm:** để phối hợp triển khai container.

- **Docker Services:** là các containers trong production. 1 service chỉ run 1 image nhưng nó mã hoá cách thức để run image — sử dụng port nào, bao nhiêu bản sao container run để service có hiệu năng cần thiết và ngay lập tức.

2. Postgre SQL

a) Khái niệm

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.

PostgreSQL được phát triển dựa trên POSTGRES 4.2 tại phòng khoa học máy tính Berkeley, Đại học California.

PostgreSQL được thiết kế để chạy trên các nền tảng tương tự UNIX. Tuy nhiên, PostgreSQL sau đó cũng được điều chỉnh linh động để có thể chạy được trên nhiều nền tảng khác nhau như Mac OS X, Solaris và Windows.

PostgreSQL là một **phần mềm mã nguồn mở miễn phí**. Mã nguồn của phần mềm khả dụng theo license của PostgreSQL, một license nguồn mở tự do. Theo đó, bạn sẽ được tự do sử dụng, sửa đổi và phân phối PostgreSQL dưới mọi hình thức.

PostgreSQL không yêu cầu quá nhiều công tác bảo trì bởi có tính ổn định cao. Do đó, nếu bạn phát triển các ứng dụng dựa trên PostgreSQL, chi phí sở hữu sẽ thấp hơn so với các hệ thống quản trị dữ liệu khác.

b) Ưu điểm

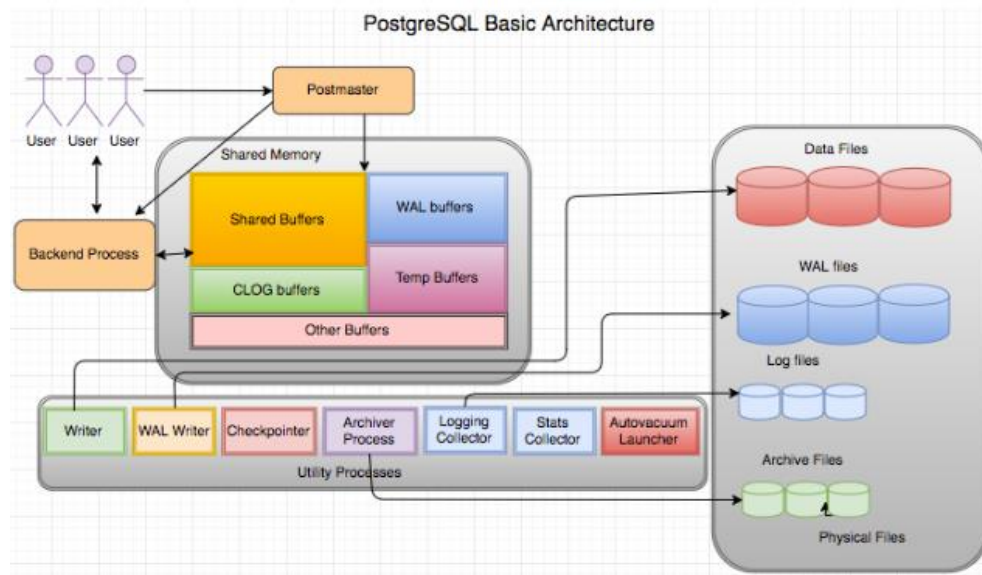
- Dễ sử dụng
- Cho phép chạy các trang web, ứng dụng web động với LAMP
- Lưu nhật ký đã viết trước đó để sử dụng như một cơ sở dữ liệu với khả năng sửa lỗi tốt
- Có sẵn mã nguồn miễn phí theo giấy phép nguồn mở, cho phép chỉnh sửa và triển khai tùy theo nhu cầu
- Có khả năng đối tượng hóa dữ liệu
- Được hỗ trợ bởi một cộng đồng đông đảo và nhiều kinh nghiệm
- Giảm thiểu việc bảo trì hệ thống

c) Nhược điểm

- Do không thuộc quyền quản lý của bất kỳ tổ chức nào nên gây khó khăn cho người dùng khi muốn tiếp cận với đầy đủ tính năng
- Tập trung vào khả năng tương thích nên khi thực hiện thay đổi để cải thiện tốc độ nên sẽ yêu cầu nhiều công việc hơn so với MySQL
- Một số ứng dụng nguồn mở không hỗ trợ PostgreSQL dù có hỗ trợ MySQL
- Hiệu suất chậm hơn so với MySQL

d) Kiến trúc PostgreSQL

Nó bao gồm một bộ nhớ chung, một vài tiến trình nền và các file dữ liệu.



- Bộ nhớ chung

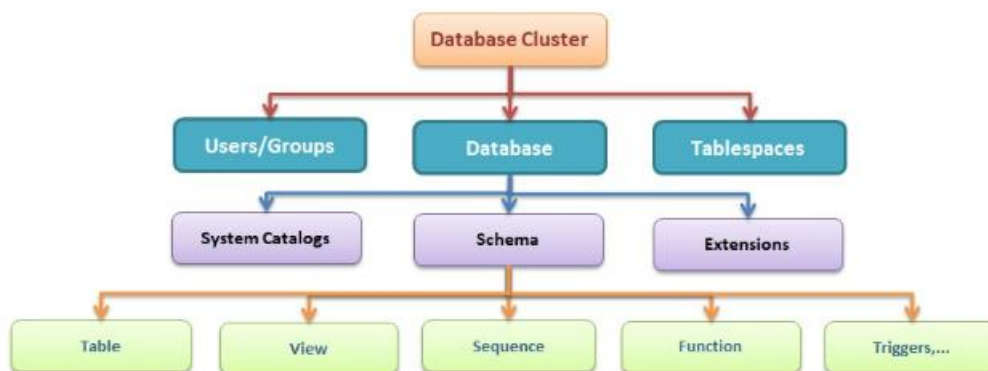
Bộ nhớ chung (Shared Memory) là bộ nhớ dành riêng cho lưu trữ cơ sở dữ liệu và nhật ký giao dịch. Các thành phần quan trọng của bộ nhớ chung có thể kể đến là Shared Buffer, WAL Buffer, CLOG Buffer, Temp Buffer, Work Memory và Vacuum Buffer

e) Loại tiến trình PostgreSQL

PostgreSQL có bốn loại tiến trình là:

- Postmaster (Daemon) Process
- Background Process
- Backend Process
- Client Process

f) Cấu trúc database



3. Superset

a) Khái niệm

- Superset là nền tảng mã nguồn mở dùng để khám phá và trực quan hóa dữ liệu. Superset được code bằng flask và react.

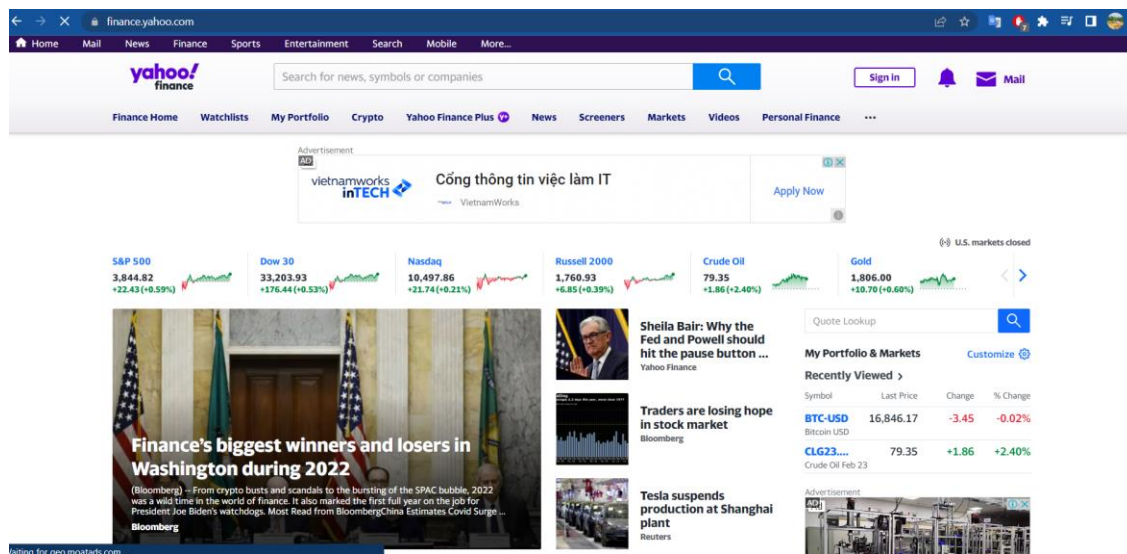
b) Ưu điểm

- Mạnh mẽ và dễ sử dụng
- Có tích hợp các cơ sở dữ liệu hiện đại, có upload file data
- Có số lượng biểu đồ phong phú
- Có thể sửa code để bổ sung những tính năng mới

CHƯƠNG II. THU THẬP VÀ XỬ LÝ DỮ LIỆU

1. Thu thập dữ liệu

Để có thể xây dựng được hệ thống thông tin tài chính, chúng tôi đã lấy dữ liệu giá cổ phiếu trên web yahoo finance.



Cách lấy dữ liệu: Sử dụng thư viện của python để lấy dữ liệu của 13 loại cổ phiếu: 'NKE', 'BB', 'FDX', 'CCL', 'BB.TO', 'LASE', 'URNM', 'VERO', 'TSLA', 'FTNT', 'CEI', 'OP', 'ADCT'. Câu lệnh python:

```
[1] !pip install yfinance
import yfinance as yf
msft = yf.Ticker('name_finance')
hist = msft.history(period='max')
```

- Dùng period='max' để lấy được toàn bộ dữ liệu của cổ phiếu đó

Dữ liệu thu được có dạng:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
1	2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2	2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
3	2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
4	2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0

2. Xử lý dữ liệu

Chúng tôi tiến hành sử dụng code python để thêm một số trường dữ liệu cần thiết vào dữ liệu:

```
def finance(name_finance):
    msft = yf.Ticker(name_finance)
    hist = msft.history(period='max')
    hist['finance'] = name_finance
    hist.reset_index(inplace=True)
    hist['Year'] = hist['Date'].apply(lambda x: x.year)
    hist['Month'] = hist['Date'].apply(lambda x: x.month)
    hist['WeekNum'] = hist['Date'].apply(lambda x: x.isocalendar()[1])
    hist['Day'] = hist['Date'].apply(lambda x: datetime.datetime.strptime(str(x).split(' ')[0], '%Y-%m-%d'))
    return hist[['Date', 'Open', 'High', 'Low', 'Close', 'Volume', 'Dividends',
                  'Stock Splits', 'finance', 'Year', 'Month', 'WeekNum', 'Day']]
```

CHƯƠNG III. THỰC HIỆN

1. Cài đặt Docker

Bước 1. Truy cập link: [docker](#) sau đó tải file Docker Desktop for Windows

Bước 2. Docker run -d -p 80:80 docker/getting-started

2. Cài đặt Superset trên Docker

Bước 1. Mở cmd và chạy câu lệnh: git clone

<https://github.com/apache/superset.git>

Bước 2. Chạy câu lệnh: `docker-compose -f docker-compose-non-dev.yml pull`

Bước 3. Chạy câu lệnh: `docker-compose -f docker-compose-non-dev.yml up`

Bước 4. Truy cập vào đường dẫn: <http://localhost:8088/login/> để login vào tài khoản superset

3. Cài đặt và kết nối PostgreSQL trên Docker

Bước 1. Tạo một file `docker-compose.yml` trên máy

```
docker-compose.yml
1  version: '3'
2
3  services:
4    postgres-db:
5      image: postgres:latest
6      container_name: database
7      restart: always
8      environment:
9        - POSTGRES_DB=finance
10       - POSTGRES_USER=postgres
11       - POSTGRES_PASSWORD=postgres
12      volumes:
13        - postgresql_bin:/usr/lib/postgresql
14        - postgresql_data:/var/lib/postgresql/data
15        - ./data/data.sql:/docker-entrypoint-initdb.d/data.sql
16      ports:
17        - 2001:5432
18
19  volumes:
20    postgresql_bin:
21      driver: local
22    postgresql_data:
23      driver: local
24
```

- **Bước 2.** Chạy câu lệnh: `docker-compose up` để tạo container chứa database PostgreSQL

4. Thêm dữ liệu và database

- Connect to database

```
31  # Connect to existing database
32  conn = psycopg2.connect(
33      database="finance",
34      user="postgres",
35      password="postgres",
36      host="host.docker.internal",
37      port="2001"
38  )
```

- Open Cursor to perform database operation

```
40 # Open cursor to perform database operation
41 cur = conn.cursor()
42
43 ✓ if cur is not None:
44     print('có database nè ')
45 ✓ else:
46     print('no')
47 #
```

- Create table

```
48 # #create table
49 cur.execute("create table finance (dtme timestamp, open_price float, \
50     high_price float, low_price float, close_price float, volume integer, dividends float,\
51     stock_splits float, finance varchar(20), year integer, month integer, week_num integer, day timestamp);")
52
```

- Insert data

```
53 #insert data
54 ✓ for i in df.values:
55     print(i)
56 ✓ cur.execute("INSERT INTO finance (dtme , open_price, high_price, low_price, close_price, volume, dividends, \
57     stock_splits, finance, year, month, week_num, day) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)", i)
58 conn.commit()
59
```

- Query the database

```
60 # Query the database
61 cur.execute("SELECT * FROM finance")
62 rows = cur.fetchall()
63 for row in rows:
64     print(row)
65 cur.close()
66 conn.close()
67
```

- Mở Terminal và chạy câu lệnh: python db_connect.py

5. Connect database với superset

Bước 1. Vào Superset → chọn Setting → chọn Database Connected → chọn Thêm Database → PostgreSQL

Bước 2. Nhập vào các ô trống như sau:

Connect a database

STEP 2 OF 3

Enter the required PostgreSQL credentials

Need help? Learn more about [connecting to PostgreSQL](#).

HOST * ? PORT *

host.docker.internal 2001

DATABASE NAME *

finance

Copy the name of the database you are trying to connect to.

USERNAME *

postgres

PASSWORD

postgres

DISPLAY NAME *

PostgreSQL

Pick a nickname for how the database will display in Superset.

ADDITIONAL PARAMETERS

e.g. param1=value1¶m2=value2

BACK CONNECT

Bước 3. Chọn CONNECT

Bước 4. Chọn mục SQL → SQL Lab → Chọn DATABASE muốn connect → Chọn Schema public → Chọn table finance → Query bằng câu lệnh:

SELECT * FROM finance

Sau đó lưu dữ liệu

Save or Overwrite Dataset

Save as new

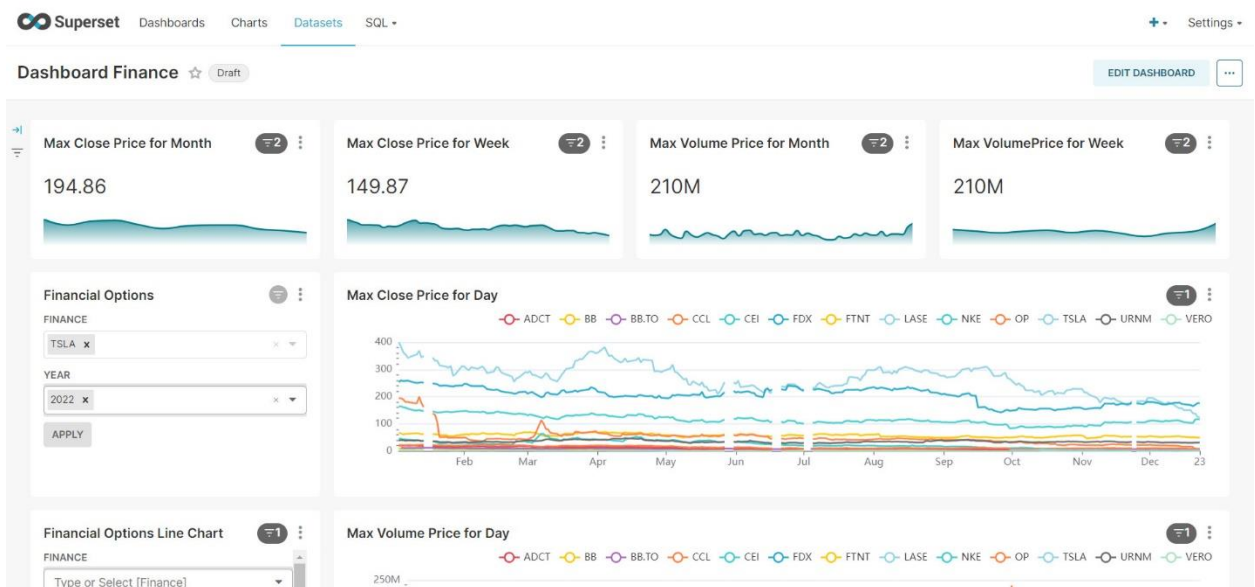
Overwrite existing

data_finance

Select or type dataset name

SAVE & EXPLORE

6. Vě dashboard



CHƯƠNG IV. KẾT LUẬN

Sau một thời gian nghiên cứu và thực hiện nhóm đã hoàn thành đề tài. Đề tài đã đạt được một số kết quả sau:

Hiểu lý thuyết và hiện thực các công nghệ liên quan như:

+ Docker

+ MySQL

Nhóm cũng đã phân tích được một số kiến thức về tài chính liên quan, các loại giá (giá đóng cửa, giá mở cửa, giá lớn nhất trong ngày, ...) của giá của các loại cổ phiếu thay đổi theo thời gian.

CHƯƠNG V. HƯỚNG NGHIÊN CỨU TRONG TƯƠNG LAI

Đề tài này hiện tại chỉ có thể thực hiện trên local. Chúng tôi sẽ cải tiến đề tài này bằng cách deploy lên webservice để bất kỳ ai quan tâm đến trị giá cổ phiếu đều có thể dễ dàng theo dõi. Tiến hành làm thêm phần front-end và back-end nữa để có thể dễ dàng sử dụng.

Không chỉ dừng lại ở những loại cổ phiếu này, chúng tôi có thể áp dụng cho nhiều loại cổ phiếu hơn nữa để có thể bắt kịp các xu thế khác trên thế giới.