



狄努股市分析

109550151 張昕蓊、110550044 吳沁霖、110550052 楊沁瑜、110550091 吳承瑀

1. An introduction of your application, including why you want to develop the application and the main functions of your application.

主題

這個程式的目的是分析股票走勢，給予使用者買入或賣出股票的參考。我們透過抓取台灣各公司的即時股票資訊，包括日期、開盤價、收盤價、最低及最高價、股價.....等基本資料來做進一步的股票走勢分析。

動機

我們的發想動機是，身為大學生，未來即將進入就業市場，對於投資市場未來勢必要有所接觸，因此我們想從股票切入作為投資的初步了解，藉此了解台灣企業的市場價值，對未來進入社會做一個初步了解。且我們都對於金融方面有

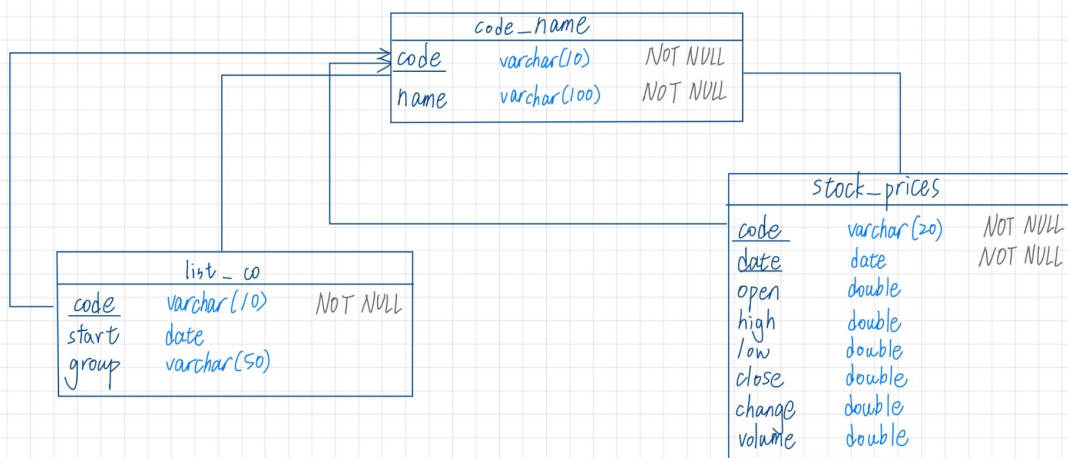
很濃厚的興趣，也可藉此機會了解金融相關知識。而在研究股票時，我們發現作為新手，有許多地方需要花很多時間去爬文，了解股票運作、如何分析，及哪公司的股票能買入賣出。因此，我們想建立一個讓新手更快速了解一間公司的股票走勢，或提供大家股票買出賣出建議的媒介。

運作方式

使用者可以選擇運作模式

1. 先尋找公司代碼，可採用下列方法：
 - 列出所有 970 間公司
 - 列出特定產業公司
 - 用公司名尋找特定公司
2. 輸入公司代碼或名稱，開始進行股票分析，可採用下列方法：
 - 即時
 - 一週
 - 一個月
 - 一年
 - 五年
3. 離開

2. Database design - describe the schema of all your tables in the database, including keys and index, if applicable (why you need the keys, or why you think that adding an index is or is not helpful).



3. Database design - describe the normal form of all your tables. If the tables are not in BCNF, please include the reason for it (performance trade-off, etc.).

我們的設計中總共有三個 relations，三個都符合 BCNF 的形式。

List of relations			
Schema	Name	Type	Owner
public	code_name	table	postgres
public	list_co	table	postgres
public	stock_prices	table	postgres
(3 rows)			

Table "public.list_co"				
Column	Type	Collation	Nullable	Default
_code	character varying(10)		not null	
_start	date			
_group	character varying(50)			
Indexes:				
"list_co_pkey" PRIMARY KEY, btree (_code)				
"list_co__group_idx" btree (_group)				

list_co (BCNF) {'code', 'start', 'group'}

pk = 'code'

index = 'code' (pk), 'group'

'code' → {'start', 'group'}

Table "public.code_name"				
Column	Type	Collation	Nullable	Default
_code	character varying(10)		not null	
_name	character varying(100)			
Indexes:				
"code_name_pkey" PRIMARY KEY, btree (_code)				
"code_name__name_idx" btree (_name)				

code_name (BCNF) {'code', 'name'}

pk = 'code'

index = 'code' (pk), 'name'

'code' → 'name'

Table "public.stock_prices"				
Column	Type	Collation	Nullable	Default
_code	character varying(10)		not null	
_date	date		not null	
_open	double precision			
_high	double precision			
_low	double precision			
_close	double precision			
_change	double precision			
_volume	double precision			
Indexes:				
"stock_prices_pkey" PRIMARY KEY, btree (_code, _date)				
"stock_prices__code_idx" btree (_code)				
"stock_prices__date_idx" btree (_date)				

stock_prices (BCNF) {'code', 'date', 'open', 'high', 'low', 'close', 'change', 'volume'}

pk = {'code', 'date'}

index = {'code', 'date'}(pk)

{'code', 'date'} → {'open', 'high', 'low', 'close', 'change', 'volume'}

4. From the data sources to the database - describe the data source and the original format.

Data source: 台灣證券交易所

Original format:

1. list_co: type(股票類型)、code(公司代碼)、name(公司名稱)、ISIN、start(上市時間)、group(產業類型)
2. stock_prices: date(日期)、open(開盤價)、high(最高價)、low(最低價)、close(收盤價)、change(漲跌)、change%(漲跌%)、volume(成交張數)

5. From the data sources to the database - describe the methods of importing the original data to your database and strategies for updating the data, if you have one.

資料主要來源:

<https://invest.cnyes.com/twstock/TWS/1101/history?fbclid=IwAR0Krjdiw0d50bjc7weHkRYTI0m-Da0w5wi42mAY-HOAb6av8RI3Rvp6BR4#fixed>

我們從這個網站下載了台灣 970 家上市公司過去五年內的股票資料（各個時間點的即時價、開盤價、收盤價、日期），作為較久遠的股票資料查詢。

<https://www.twse.com.tw/zh/?fbclid=IwAR1cKKrxjdoyyYMHrDuW0-lwD4XYBB4KGODRnz5-zuPrL9jxyZICKDV6GVI>

而近期的股價資訊則透過 python 爬蟲台灣證券交易所的資料而來，製作出來的程式會以月為單位做出及時更新，而查詢的該公司也能查詢到最即時的股票資訊。

6. Application with database - explain why your application needs a database.

因為此股票分析系統需要大量股票資料，包含 970 間台灣公司近 5 年的歷史資料以及即時資料，所以當使用者需要的資料需要使用到超過一個月、一年、甚至是最多到五年的資料的時候，不需要靠爬蟲即時爬下如此巨量的資料，而可以快速從已經存在雲端的資料庫提取所需要的資料，增進效能。除了效能問題外，還有一點必須使用資料庫的原因是因為台灣證券交易所的網站有設定防駭客、機器人攻擊，因此有設定若五秒內存取超過三次的資料便會對該網域進行時長一個小時封鎖行動，導致沒有辦法再繼續存取資料。

7.Application with database - includes the queries that are performed by your application, how your application performed these queries (connections between application and database), and what is the cooperating functions for your application.

首先我們引用需要用來連結 RDS 的模組，「psycopg2」。

```
import psycopg2
from psycopg2.extras import RealDictCursor
```

並且設定連線參數。

```
host = "database-1.c4alhaecwxce.us-east-1.rds.amazonaws.com"
username = "postgres"
password = "postgres"
database = "dbhw1"

conn = psycopg2.connect(
    host = host,
    database = database,
    user = username,
    password = password
)
cur = conn.cursor(cursor_factory = RealDictCursor)
```

以下為我們的程式內有用到的 query 程式碼。

```
cur.execute("select _code, _name from code_name")
cur.execute("select _date, _close, _open, _high, _low \
            from stock_prices where _code = '" + company_code + "' \
            and _date < '" + oldest_day + "' \
            order by _date desc limit " + str(plus))
cur.execute("select _code, _name from code_name")
cur.execute("select distinct _group from list_co")
cur.execute("select code_name._name, code_name._code, list_co._group \
            from list_co, code_name where list_co._group like '%" \
            + group_name + "%' and code_name._code = list_co._code")
cur.execute("select _code, _name from code_name \
            where _name = '" + company_name + "'")
cur.execute("select _name from code_name \
            where _code = '" + company_code + "'")
cur.execute("select _name, _code from code_name \
            where _name like '%" + company_name + "%'")
```

並且在每一次進行 execute 後，使用以下的程式碼來進行擷取 RDS 回傳的結果。

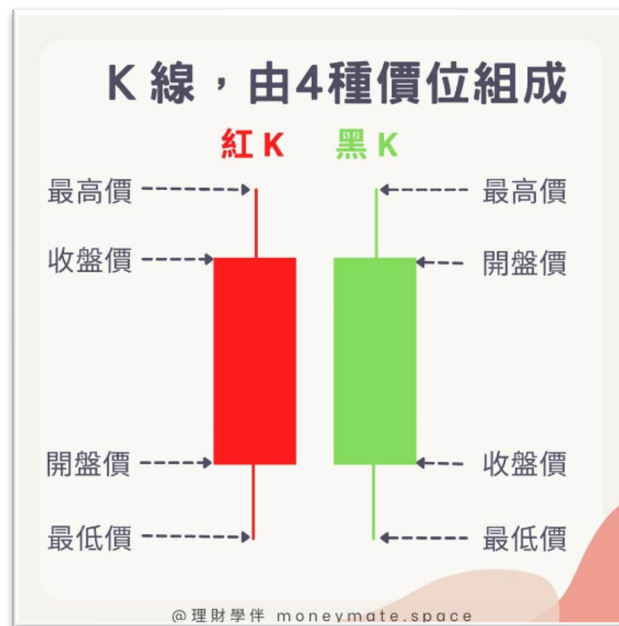
```
results = cur.fetchall()
```

8. All the other details of your application that you want us to know.

分析方式

K 線

簡化資訊，用視覺化的方式呈現股價的走勢，讓我們看出一段期間的開盤價、收盤價、最高價以及最低價，可分為日 K、周 K、月 K、年 K。



SMA N 日收盤價/ N (搭配 K 線圖)

移動平均線的目的，就是「透過取平均值讓價格變動平滑化」，這樣讓價格變動的走勢更為清楚。有短期、長期線。5MA 五日平均線、20MA 二十日平均線。

ex:

若股票現價向上穿越 MA 線，代表股市走強，為買進訊號。

若股票現價向下穿越 MA 線，代表股市走弱，為賣出訊號。

RSI $[\text{漲幅平均}/(\text{漲幅平均}+\text{跌幅平均})]*100$

RSI 指標是一個用來判斷近期股價相對強弱的技術指標。

ex:

RSI 大於 70 時，為超買訊號，市場過熱，要準備開始跌了。

RSI 小於 30 時，為超賣訊號，市場過冷，要準備開始漲了。