

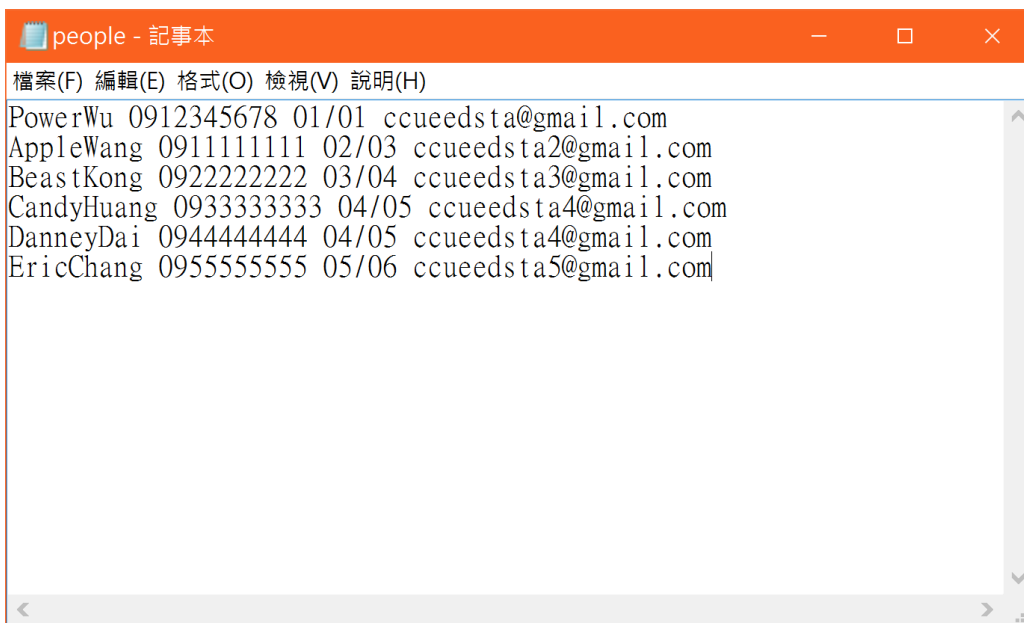
## Project 2 – Relational Database Management System

資料庫系統為資訊系統中非常重要的一環，現今三大資料庫系統 Oracle、MySQL、Microsoft SQL Server 皆是使用結構化查詢語言 (SQL) 來存取內部的資料，雖然我們並不知道 Oracle、MySQL 等資料庫底層檔案系統如何設計，但他們的查詢語言都是一致的，請依照 SQL 的查詢語言規則，利用 Linked List，來建構自己的資料庫系統。

關聯式資料庫 (Relational database management system) 是目前主流的資料庫型態，目的是方便查詢，與減少資料表的內容大小，其查詢方式可以合併表格，呈現方式常見的如 Entity Relationship Diagram (ERD) 等方式呈現。

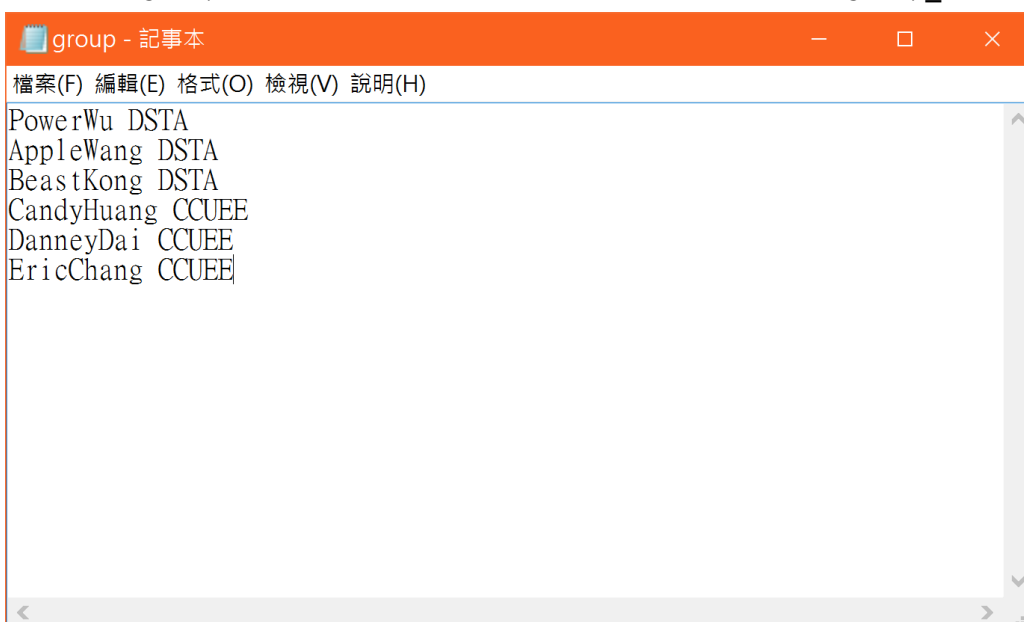
### ※問題描述：

目前有兩個資料庫，一個為 people.txt，用來儲存聯絡人的名字 (name [20])、電話 (phone [15])、生日 (birthday [10])、e-mail (email [50])。所有欄位的字中間沒有空格。



```
people - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
PowerWu 0912345678 01/01 ccueedsta@gmail.com
AppleWang 0911111111 02/03 ccueedsta2@gmail.com
BeastKong 0922222222 03/04 ccueedsta3@gmail.com
CandyHuang 0933333333 04/05 ccueedsta4@gmail.com
DanneyDai 0944444444 04/05 ccueedsta4@gmail.com
EricChang 0955555555 05/06 ccueedsta5@gmail.com
```

另一個為 group.txt，用來儲存名字 (name [20]) 與隊伍名稱 (group\_name [20])。



```
group - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
PowerWu DSTA
AppleWang DSTA
BeastKong DSTA
CandyHuang CCUEE
DanneyDai CCUEE
EricChang CCUEE
```

※功能需求 – Basic functions :

SQL : Insert (3%) , 增加資料庫中的資料, 有 people 與 group 兩個資料, 如果要在 people 資料表下增加一筆資料, 下達指令:

```
INSERT INTO people ( name , phone , birthday , email ) VALUES ( PowerWu ,  
09/2345678 , 01/01 , ccueedsta@gmail.com )
```

增加一筆 group 資料, 下達指令:

```
INSERT INTO group ( name , group_name ) VALUES ( PowerWu , DSTA )
```

SQL : Delete (3%) , 刪除一筆資料, 從 people 資料表或 group 資料表, 下達需要刪除的那筆資料的查詢值, 如刪除所有名字為 PowerWu 的資料, 下達指令:

```
DELETE FROM people WHERE name = PowerWu
```

從 people 資料表, 刪除所有手機為 09/2345678:

```
DELETE FROM people WHERE phone = 09/2345678
```

從 group 資料表, 刪除所有隊伍名稱為 DSTA 的資料:

```
DELETE FROM group WHERE group_name = DSTA
```

SQL : Update (4%) ,

從 people 資料表, 修改 PowerWu 的手機為 0900888888, 下達指令:

```
UPDATE people SET phone = 0900888888 WHERE name = PowerWu
```

從 group 資料表, 修改 PowerWu 的隊伍名稱為 DSTA, 下達指令:

```
UPDATE group SET group_name = DSTA WHERE name = PowerWu
```

SQL : SELECT 顯示欄位 (3%) ,

從 people 資料表, 顯示所有人的姓名、生日、電話, 下達:

```
SELECT name , birthday , phone FROM people
```

從 people 資料表, 顯示所有人的所有欄位, 下達以下兩種指令皆可:

```
SELECT name , phone , birthday , email FROM people
```

```
SELECT * FROM people
```

從 group 資料表, 顯示所有人的姓名, 下達:

```
SELECT name FROM group
```

從 group 資料表, 顯示所有人的所有欄位, 下達以下兩種指令皆可:

```
SELECT name , group_name FROM group
```

```
SELECT * FROM group
```

SQL : SELECT 查詢 (7%) ,

查詢生日為 03/25 的聯絡人, 並列出所有欄位, 下達:

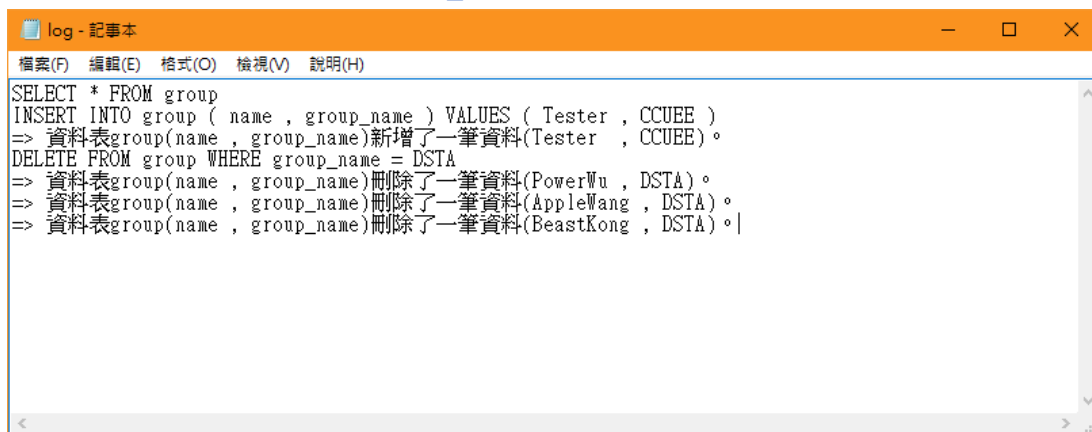
```
SELECT * FROM people WHERE birthday = 03/25
```

Log : 紀錄所有動作 (10%) , 每一次輸入 SQL 指令後, 在 log.txt 紀錄輸入的指令, 以及資料表更改的內容。例如下達指令 INSERT INTO group ( name , group\_name ) VALUES

( PowerWu , DSTA ) , 系統除了記錄此指令還會記錄資料表的狀態:

```
INSERT INTO group ( name , group_name ) VALUES ( PowerWu , DSTA )
```

=> 資料表 group(name , group\_name) 新增了一筆資料 (PowerWu , DSTA) 。



```
log - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
SELECT * FROM group
INSERT INTO group ( name , group_name ) VALUES ( Tester , CCUEE )
=> 資料表group(name , group_name)新增了一筆資料(Tester , CCUEE)。
DELETE FROM group WHERE group_name = DSTA
=> 資料表group(name , group_name)刪除了一筆資料(PowerWu , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(AppleWang , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(BeastKong , DSTA)。
```

※功能需求 – Main functions：

Relational database management system(10%)，關聯式資料庫，可以連結兩個資料表，且查詢可以交互使用，例如查詢所有為 DSTA 組別的人的手機號碼，可以下達指令：

SELECT phone FROM people , group WHERE group\_name = DSTA 或

SELECT phone FROM group , people WHERE group\_name = DSTA

或是查詢 DSTA 組別的所有資料，下達指令：

SELECT \* FROM people , group WHERE group\_name = DSTA 或

SELECT \* FROM group , people WHERE group\_name = DSTA

必需考慮所有的資料表組合，也就是說 name, phone, birthday, email, group\_name 總共有 33 種出現組合與 N 總排列方式。測資也有可能出現：

SELECT group\_name , phone FROM group , people WHERE name = AppleWang

SELECT birthday , name FROM people , group WHERE group\_name = DSTA

.....

Undo (10%)，輸入「undo」指令，能復原上一個動作。例如：undo 一筆刪除的資料，除了要在原本的 txt 檔內復原外，log.txt 也需要紀錄。



```
log - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
INSERT INTO group ( name , group_name ) VALUES ( Tester , CCUEE )
=> 資料表group(name , group_name)新增了一筆資料(Tester , CCUEE)。
DELETE FROM group WHERE group_name = DSTA
=> 資料表group(name , group_name)刪除了一筆資料(PowerWu , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(AppleWang , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(BeastKong , DSTA)。
undo
=> 復原：資料表group(name , group_name)刪除了一筆資料(PowerWu , DSTA)。
=> 復原：資料表group(name , group_name)刪除了一筆資料(AppleWang , DSTA)。
=> 復原：資料表group(name , group_name)刪除了一筆資料(BeastKong , DSTA)。
undo
=> 復原：資料表group(name , group_name)新增了一筆資料(Tester , CCUEE)。
```

Redo (10%)，輸入「redo」指令，能重作「被復原」的上一個動作。



```
log - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
INSERT INTO group ( name , group_name ) VALUES ( Tester , CCUEE )
=> 資料表group(name , group_name)新增了一筆資料(Tester , CCUEE)。
DELETE FROM group WHERE group_name = DSTA
=> 資料表group(name , group_name)刪除了一筆資料(PowerWu , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(AppleWang , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(BeastKong , DSTA)。
undo
=> 復原：資料表group(name , group_name)刪除了一筆資料(PowerWu , DSTA)。
=> 復原：資料表group(name , group_name)刪除了一筆資料(AppleWang , DSTA)。
=> 復原：資料表group(name , group_name)刪除了一筆資料(BeastKong , DSTA)。
undo
=> 復原：資料表group(name , group_name)新增了一筆資料(Tester , CCUEE)。
redo
=> 資料表group(name , group_name)新增了一筆資料(Tester , CCUEE)。
redo
=> 資料表group(name , group_name)刪除了一筆資料(PowerWu , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(AppleWang , DSTA)。
=> 資料表group(name , group_name)刪除了一筆資料(BeastKong , DSTA)。
```

ps. undo, redo 僅限於 INSERT、UPDATE、DELETE。

#### ※功能需求 – Extra Bonus：

SQL：SELECT 邏輯查詢 (10%)，原本的 WHERE 查詢值，可以增加任意個 AND、OR 邏輯判斷子，以及判斷「等於 =」與「不等於 !=」，例如：下達查詢列出 (1.) group\_name 為 DSTA，且名字不是 PowerWu 的人 (2.) group\_name 為 CCUEE 的成員：

```
SELECT * FROM people , group WHERE group_name = DSTA AND name != PowerWu OR
group_name = CCUEE
```

ps. AND 的優先權 > OR

※配分 (110%) :

Basic functions (30%) :

- SQL : Insert (3%)
- SQL : Delete (3%)
- SQL : Update (4%)
- SQL : SELECT 顯示欄位 (3%)
- SQL : SELECT 查詢 (7%)
- Log : 紀錄所有動作 (10%)

Main functions (30%) :

- Relational database management system (10%)
- Undo (10%)
- Redo (10%)

Extra bonus (10%) :

- SQL : SELECT 邏輯查詢 (10%)

Documents + Interview (40%) :

- Design Document (flow chart, functions)
- User manual (how to use your project)
- Time log & team work (every team member)
- Interview (every team member)

※條件限制 :

1. 底層資料運算必須以 Linked list 表示。
2. 所有的聯絡人資料檔案必須存放在 people.txt，群組資料放在 group.txt。
3. 原本在 people.txt 和 group.txt 的檔案，開啟程式後必須先被讀取。例如：開始時輸入 SELECT \* FROM group，如果 group.txt 本身有東西就必須先讀取出來。
4. 下達新增、修改、刪除指令後，people.txt，group.txt 的檔案必須隨之更新。
5. 禁止使用「全域變數」。
6. 查詢資料必須顯示在螢幕上。
7. 必須使用 C 語言，請勿使用 C#、JAVA 等高階語言。
8. Log 紀錄放在 log.txt，需紀錄所有的 Input 指令與資料的變化。請排版以利評分。
9. 所有指令必須按照 SQL 語法下達，沒有自行設計的介面，例如輸入只有單純的數字來選取功能。

※重要 :

1. Deadline : 2016/11/17 (四) 上午 10 點，以 e-course 最後上傳的版本為主，不接受遲交。
2. Compile 不過，Project 0 分，上傳之前自己要對自己的 Project 負責，整合後務必自行測試 (做不出來的功能就先 Comment 起來，以免整個 Project 沒有分數)。
3. Interview deadline : 2016/11/24 (四) 下午 3 點前。
4. Interview 時間最晚必須提早前一天與助教約，助教答覆後才算約成。遲到者該次 Interview 5 折，遲到超過 30 分鐘或爽約該次 Interview 0 分。
5. 上次文件檔沒有完整的，這次請補齊，要有 flow chart、Time log、分工、功能完成度、

截圖、操作手冊.....。

6. 需符合以上條件限制，功能才算達成。

※提示：

1. Undo、Redo 可以使用 Stack。
2. 使用 string token 可以方便字串處理。

## 補充

※SQL 基本語法：

SELECT 欄位 , 欄位 , 欄位 FROM 表格

SELECT \* FROM 表格

SELECT \* FROM 表格 WHERE 欄位 = 查詢值

SELECT \* FROM 表格 WHERE 欄位 = 查詢值 AND 欄位 = 查詢值 AND 欄位 = 查詢值

SELECT \* FROM 表格 ORDERED BY 欄位

INSERT INTO 表格 ( 欄位 , 欄位 , 欄位 ) values ( 值 , 值 , 值 )

UPDATE 表格 SET 欄位 = 新的值 WHERE 欄位 = 查詢值

DELETE FROM 表格 WHERE 欄位 = 查詢值

※有興趣的同學可以安裝 Wampserver，實際操作資料庫系統：

<http://www.wampserver.com/en/>

需先安裝 VC // vc\_redist\_x64/86.exe

<http://www.microsoft.com/en-us/download/details.aspx?id=30679>

再下載 Wampserver，安裝完後開啟瀏覽器。

輸入「<http://localhost/phpmyadmin/>」即可進入 mySQL 後台管理系統。

初始帳號為「root」，密碼為空值。

可以增加資料庫與資料表，下 SQL 指令。

歡迎進入後端的世界~~~

工程師就是每天都要學一項新技術 QAQ

每天世界上多的新技術比每天能學的新技術還多上很多很多.....