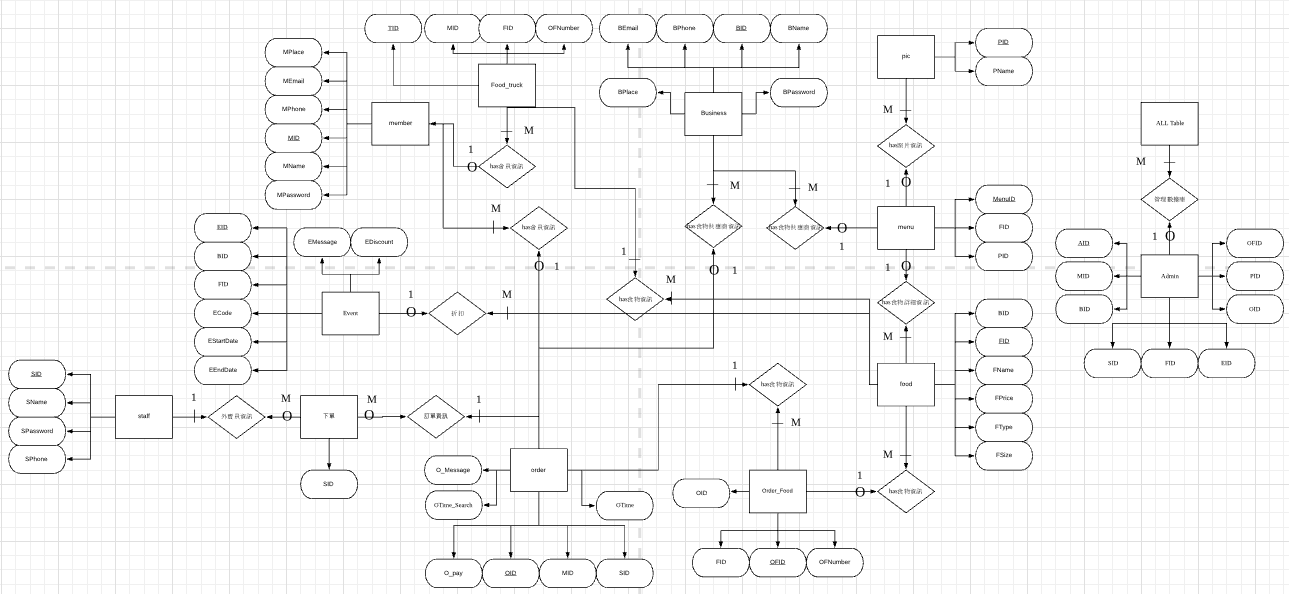
**資訊及通訊科技科校本評核**

**譽譽送**

**黃天佑**

**構思與應用**

CREATE DATABASE SBADB; 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| member |  |  |  |  |
| name | description | data type | length | key |
| MName | 名稱 | string | 10 |  |
| MID | 編號 | int | 10 | pk |
| MPassword | 密碼(使用加密系統) | string | 20 |  |
| MPhone | 電話 | int | 8 |  |
| MEmail | 電郵 | string | 50 |  |
| MPlace | 地址 | string | 100 |  |

CREATE Table member(

MName char(10),

MID int(10) NOT NULL UNIQUE,

MPassword char(20)NOT NULL,

MPhone int(8)NOT NULL,

MEmail char(50)NOT NULL UNIQUE,

MPlace char(100),

PRIMARY KEY(MID)

)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Business |  |  |  |  |
| name | description | data type | length | key |
| BName | 商家名稱 | string | 10 |  |
| BID | 商家編號 | int | 10 | pk |
| BPassword | 商家密碼(使用加密系統) | string | 20 |  |
| BPhone | 商家電話 | int | 8 |  |
| BPlace | 商家地址 | string | 100 |  |
| BEmail | 商家電郵 | string | 50 |  |

CREATE Table Business(

BName char(10)NOT NULL,

BID int(10) NOT NULL UNIQUE,

BPassword char(20)NOT NULL,

BPhone int(8)NOT NULL,

BPlace char(100),

BEmail char(50)NOT NULL UNIQUE,

PRIMARY KEY(BID)

)

商家擁有自己的登入介面以及食物上傳介面, 食物相傳介面中以business.BID需要等於food.BID才能成功上傳,同時商家需要自擬食物名稱以及價錢、大小選擇等.而FID則系統自動派發例如:

INSERT INTO food(

FName,FType,FPrice,

FSize,FID)

VALUES (蛋炒飯,飯類,39.9,中,329)

WHERE food.BID = business.BID;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| name | description | data type | length | key |
| SName | 外賣員名稱 | string | 10 |  |
| SID | 外賣員編號 | int | 10 | pk |
| SPassword | 外賣員密碼(使用加密系統) | string | 20 |  |
| SPhone | 外賣員電話 | int | 8 |  |
| SEmail | 外賣員電郵 | string | 50 |  |

CREATE Table staff(

SName char(10)NOT NULL,

SID int(10) NOT NULL UNIQUE,

SPassword char(20)NOT NULL,

SPhone char(8)NOT NULL,

SEmail char(50),

PRIMARY KEY(SID)

)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| food |  |  |  |  |
| name | description | data type | length | key |
| FName | 食物名稱 | string | 10 |  |
| FID | 食物編號 | int | 3 | pk |
| FType | 食物種類 | string | 3 |  |
| BID | 商家編號 | int | 10 | fk |
| FPrice | 食物價錢 | double | 10 |  |
| FSize | 食物大小 | sting | 1 |  |

CREATE Table food(

FName char(10)NOT NULL,

FID char(3)NOT NULL UNIQUE,

FType char(3)NOT NULL,

BID int(10),

FPrice double(10)NOT NULL,

FSize char(1)NOT NULL,

PRIMARY KEY(FID),

FOREIGN KEY (BID),

REFERENCES Business(BID)

)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pic |  |  |  |  |
| name | description | data type | length | key |
| PName | 相片名稱 | string | 10 |  |
| PID | 相片編號 | int | 3 | pk |

//相片ID為FID+BID,例如FID為001,BID為0000000001

PID=10000000001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 下單 |  |  |  |  |
| name | description | data type | length | key |
| SID | 外賣員編號 | int | 10 | fk |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| order |  |  |  |  |
| name | description | data type | length | key |
| OID | 訂單編號 | string | 100 | pk |
| MID | 會員編號 | int | 10 | fk |
| SID | 外賣員編號 | int | 10 | fk |
| OTime | 接單時間 | DATETIME | na |  |
| OTime\_Serch | 訂單日期 | DATE | na |  |
| O\_Message | 留言 | string | 1000 |  |
| O\_pay | 檢查是否已付款 | bool | na |  |

搜索功能:Select food.FName,food.FSize,food.FPrice,food

From food,pic

Where food.FName like “?(input)?”,

food.FType = (input\_FType),

pic.PID = ‘FID’+‘BID’ //參考PID註解

可以方便搜索自己喜歡的食物

飯類

炒飯

搜尋

[Input]

食物種類FType

submit按鈕



食物名稱:泡菜炒飯

食物大小:中

食物價錢:39.8HKD



食物名稱:揚州炒飯

食物大小:中

食物價錢:49.9HKD

1 2 3

下一頁

FName

FSize

FPrice

食物排行榜:Select food.FName,count(orderfood.FID) As Ranking,food.FSize,food.FPrice

From food,orderfood,pic

Group by food.FName

Where food.FType = (input\_FType)

pic.PID = ‘FID’+‘BID’ //參考PID註解

Oder by count(orderfood.FID) desc;

為食物做一個排行榜

飯類

Top

搜尋

食物種類FType

submit按鈕



食物名稱:泡菜炒飯

食物大小:中

食物價錢:39.8HKD



食物名稱:揚州炒飯

食物大小:中

食物價錢:49.9HKD

1 2 3

下一頁

FName

FSize

FPrice

Top1

Top2

Ranking

最喜爱食物:Select food.FName,count(orderfood.FID) As 次,food.FSize,food.FPrice

From member,order,food,pic

RIGHT JOIN orderfood

ON member.MID =order.MID,

order.OID = orderfood.OID,

pic.PID = ‘FID’+‘BID’ //參考PID註解

Group by food.FName  
 Oder by count(orderfood.FID) desc;

為顧客提供上次或者叫得最多的食物

最喜爱食物

搜尋

submit按鈕



食物名稱:泡菜炒飯

食物大小:中

食物價錢:39.8HKD



食物名稱:揚州炒飯

食物大小:中

食物價錢:49.9HKD

1 2 3

下一頁

FName

FSize

FPrice

Food1(4 次)

Food2(1 次)

count(orderfood.FID)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Food\_truck |  |  |  |  |
| name | description | data type | length | key |
| TID | 會員購物車編號 | int | 10 | pk |
| MID | 會員編號 | int | 10 | fk |
| FID | 食物編號 | int | 10 |  |
| FNumber | 食物数量 | int | 100 |  |

購物車:Select food.FName,Food\_truck.FNumber,food.FSize,food.FPrice

From member,Food\_truck,food

Where member.MID =Food\_truck.MID ,Food\_truck.FID = food.FID

Group by food.FName;

顧客獨有的購物車

FoodTruck



食物名稱:泡菜炒飯

食物大小:中

食物價錢:39.8HKD



1 2 3

下一頁

FName

FSize

FPrice

Food1

Food2

食物數量:2

OFNumber

食物名稱:揚州炒飯

食物大小:中

食物價錢:49.9HKD

食物數量:1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Admin |  |  |  |  |
| name | description | data type | length | key |
| AID | 管理員編號 | int | 3 | pk |
| MID | 會員編號 | int | 10 | fk |
| BID | 商家編號 | int | 10 | fk |
| SID | 外賣員編號 | int | 10 | fk |
| FID | 食物編號 | int | 3 | fk |
| EID | 活動編號 | int | 100 | fk |
| OID | 訂單編號 | string | 100 | fk |
| PID | 相片編號 | int | 3 | fk |

管理員表格重要用於資料探勘,例如收集顧客購買習慣的資料.運用關聯規則學習,可以幫助商家確定哪些食物經常一起買,並利用這些資訊幫助行銷.例如:  
Select food.FName,count(orderfood.FID) As 次數,food.FSize,food.FPrice,food,FType

From food,orderfood

Group by food.FName

Oder by count(orderfood.FID) desc;

可以分析顧客在食物的大小、價錢、種類方面的喜好,來發送給商家以幫助行銷.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| menu |  |  |  |  |
| name | description | data type | length | key |
| MenuID | 菜單編號(客制化) | int |  | pk |
| FID | 食物編號 | int | 3 | fk |
| PID | 相片編號 | int | 3 | fk |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OrderFood |  |  |  |  |
| name | description | data type | length | key |
| OFID | 訂單食物編號 | string | 100 | pk |
| OID | 訂單編號 | string | 100 | pk |
| FID | 食物編號 | int | 3 | fk |
| OFNumber | 食物数量 | int | 100 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Event |  |  |  |  |
| name | description | data type | length | key |
| BID | 商家編號 | int | 100 | pk |
| FID | 食物編號 | int | 3 | pk |
| EID | 活動編號 | int | 100 | pk |
| ECode | 活動代碼 | string | 100000 |  |
| EStartDate | 活動開始日期 | DATE | na | pk |
| EEndDate | 結束日期 | DATE | na |  |
| EMessage | 活動內容 | string | 100 |  |
| EDiscount | 活動折扣 | int | 2 |  |

Event這個表格用以給予商家舉辦一些活動,例如聖誕節優惠,可以:

Update Food SET FPrice = FPrice \* ‘[EDiscount]%’

Where food.BID=Event.BID ,food.FID = Event.FID;

註冊

身份:顧客

登入名稱:689

密碼:218500callme

名字:黃蛋佑

Email:egg@gmail.com

電話:2185000

完成

資料儲入不同

的Database

不同DataBase:

Member/

Business/

Staff

所對應的:  
ID

Password  
Name  
Email  
Phone

submit按鈕

下拉式選單

Sql:

INSERT INTO *member*(

M*ID*,*Mpassword*,*MName*,

MEmail,MPhone)

VALUES (*689*,*218500callme*,黃

蛋佑,egg@gmai l.com,218500);

更新密码:

UPDATE Member

SET password = [newpassword];

注銷帳號:

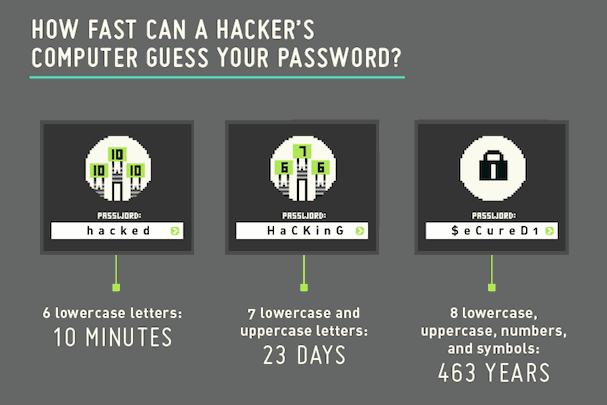
DELETE FROM Member WHERE MID = [ID],Mpassword = [password];

以上欄位經過管理員測試,符合第一範式-主鍵(無重複元組),沒有重複群,欄位原子性(元組只有一個值)-,第二範式-沒有部分函式依賴(值依賴於每個主鍵這一整體)-,第三範式-沒有傳遞函式依賴(值僅依賴於候選鍵)-,因此可以減少資料庫中資料冗餘,增進資料的一致性.

安全措施

密碼安全:

密碼應該是具有安全性的,因此用戶在輸入密碼時我們加入了一些限制,例如使用特殊符號,字母和數字的混合,不使用簡單的單詞等,並且使用類型檢查,長度檢查以及欄位完整性檢查進行限制.

但是,如果密碼以不安全的方式存儲,所有這些限制都是無效的.在SQL中,密碼通常存儲在數據庫中,因此我們應該非常小心地將密碼存儲.可是以純文本形式將密碼存儲在表中非常容易受到攻擊,因為如果攻擊者訪問數據庫,則他可以竊取用戶甚至管理員的密碼.所以數據庫中的密碼應該被加密,並使其盡可能不被非法解密.

例如SQL injection:

登入驗證的SQL查詢代碼為

SELECT \* FROM member

WHERE (name = '" username "')

and (password = '" password "');

但是當惡意填入帳號名稱和密碼:

username = "1' OR '1'='1";

password = "1' OR '1'='1";

會將導致原本的SQL字串被填為

SELECT \* FROM member

WHERE (name = '1' OR '1'='1')

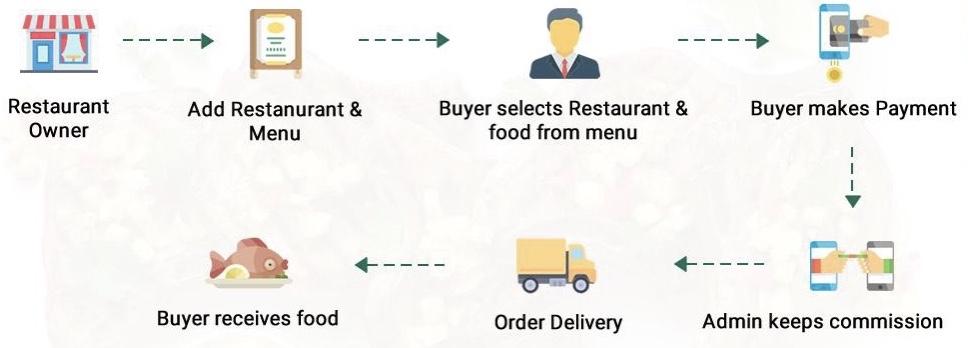
and (password = '1' OR '1'='1') ;

也就是實際上執行的SQL命令會變成下面這樣的

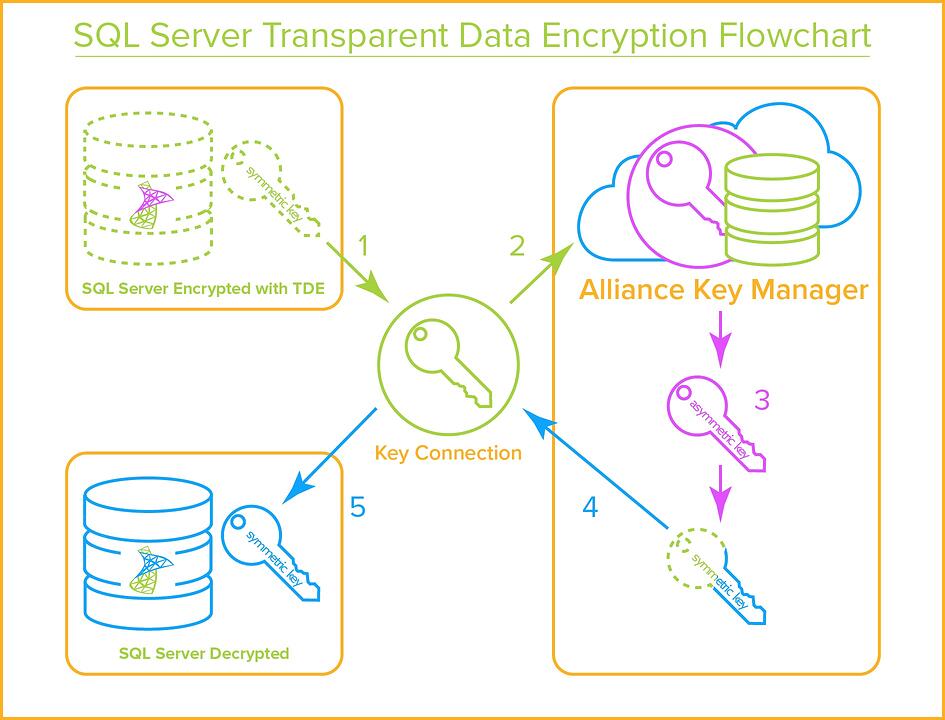
SELECT \* FROM member;

因此達到無帳號密碼,亦可登入.

可能造成的資料表中的資料外洩,例如企業及個人機密資料,帳戶資料,密碼等,或者資料結構被駭客探知,得以做進一步攻擊(SELECT \* FROM Member)例如竄改系統管理員帳戶(ALTER LOGIN admin WITH PASSWORD='xxxxxx'),取得系統較高權限後,有可能得以在應用程式加入惡意連結、惡意代碼等.



用戶在過程中需要輸入密碼進行付款,而付款進行中則需要進行加密

1. SQL服務器請求數據加密密鑰Data Encryption Key(DEK)由密鑰加密密鑰Key Encryption Key(KEK)解密.
2. 鍵連接發送解密請求Alliance Key Manager(AKM).
3. AKM解密與KEK的DEK.
4. AKM發送解密DEK到Key connection.
5. Key connection發送解密的DEK到SQL Server,以便可以對數據庫進行解密.

整個過程可以確保付款安全,防止被干擾或惡性利用.

**測試與評估**

**登入測試**

登入

登入名稱:542

密碼:5432186

MID

MPassword

測試1:MID輸入為542  
 Mpassword輸入為5432186 預期結果為 登入成功

實際結果為 登入成功

測試2:MID輸入為N423  
 Mpassword輸入為abc222  
 預期結果為 登入名稱或密碼錯誤

實際結果為 登入名稱或密碼錯誤

測試3:MID輸入為542  
 Mpassword輸入為abc222  
 預期結果為 登入名稱或密碼錯誤

實際結果為 登入名稱或密碼錯誤

測試3:MID輸入為N423  
 Mpassword輸入為5432186  
 預期結果為 登入名稱或密碼錯誤

實際結果為 登入名稱或密碼錯誤

註冊

身份:顧客

登入名稱:689

密碼:218500callme

名字:黃蛋佑

Email:egg@gmail.com

電話:2185000

完成

資料儲入不同

的Database

不同DataBase:

Member/

Business/

Staff

所對應的:  
ID

Password  
Name  
Email  
Phone

submit按鈕

下拉式選單

**註冊測試**

測試1: 輸入 測試2:輸入

身份:顧客 身份:顧客

登入名稱:689 登入名稱:689

密碼:218500callme 密碼:218500callmeyoyocheckitout

名字:黃蛋佑 名字:黃蛋佑

Email:[egg@gmail.com](mailto:egg@gmail.com) Email:[egg@gmail.com](mailto:egg@gmail.com)

電話:2185000 電話:2185000

預期結果為 成功以顧客身份註冊 預期結果為 密碼過長,超過20個字

實際結果為 成功以顧客身份註冊 實際結果為 密碼過長,超過20個字

測試3: 輸入 測試4:輸入

身份:顧客 身份:顧客

登入名稱:689 登入名稱:689

密碼:218500callme 密碼:218500callme

名字:黃蛋佑 名字:黃蛋佑

Email:[egg](mailto:egg@gmail.com)yoyo@--yoyoyo Email:[egg@gmail.com](mailto:egg@gmail.com)

電話:2185000 電話:99965498612

預期結果為 電郵格式錯誤 預期結果為 無效電話,必須為8位數字

實際結果為 電郵格式錯誤 實際結果為 無效電話,必須為8位數字

測試3: 輸入 測試5:輸入

身份:顧客 身份:外賣員

登入名稱:689 登入名稱:689

密碼: 密碼:218500callme

名字: 名字:黃蛋佑

Email: Email:[egg@gmail.com](mailto:egg@gmail.com)

電話: 電話:2185000

預期結果為 錯誤,所有選項為必填 預期結果為 成功以外賣員身份註冊

實際結果為 錯誤,所有選項為必填 實際結果為 成功以外賣員身份註冊

**搜索功能測試**

飯類

炒飯

搜尋

[Input]

食物種類

FType

submit按鈕



食物名稱:泡菜炒飯

食物大小:中

食物價錢:39.8HKD



食物名稱:揚州炒飯

食物大小:中

食物價錢:49.9HKD

1 2 3

下一頁

FName

FSize

FPrice

測試1:FType輸入為 飯類

Input輸入為 炒飯  
 預期結果為 顯示飯類的炒飯

實際結果為 顯示飯類的炒飯

測試2:FType輸入為 包類

Input輸入為 炒飯

預期結果為 沒有搜索結果

實際結果為 沒有搜索結果

測試2:FType輸入為 飯類

Input輸入為 漢堡

預期結果為 沒有搜索結果

實際結果為 沒有搜索結果

飯類

Top

搜尋

食物種類

FType

submit按鈕



食物名稱:泡菜炒飯

食物大小:中

食物價錢:39.8HKD



食物名稱:揚州炒飯

食物大小:中

食物價錢:49.9HKD

1 2 3

下一頁

FName

FSize

FPrice

Top1

Top2

**食物排行榜測試**

測試1:FType輸入為 飯類  
 預期結果為 顯示飯類的食物

實際結果為 顯示飯類的食物

測試2:FType輸入為 包類

預期結果為 顯示包類的食物

實際結果為 顯示包類的食物

測試2:FType輸入為 麵類

預期結果為 顯示麵類的食物

實際結果為 顯示麵類的食物

最喜爱食物

搜尋

submit按鈕



食物名稱:泡菜炒飯

食物大小:中

食物價錢:39.8HKD



食物名稱:揚州炒飯

食物大小:中

食物價錢:49.9HKD

1 2 3

下一頁

FName

FSize

FPrice

Food1

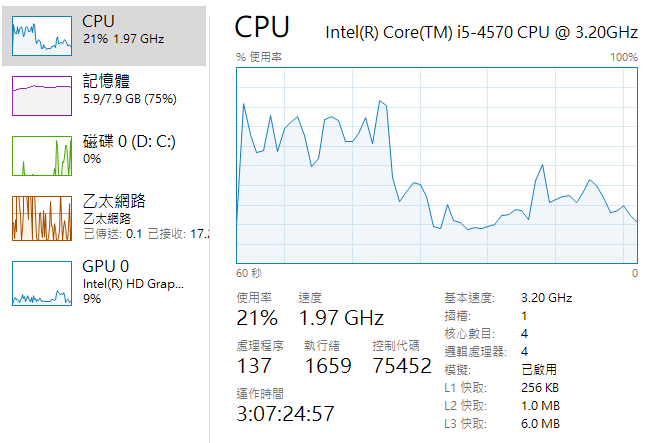
Food2

**最喜爱食物測試**

測試1:搜尋  
 預期結果為 最喜爱食物

實際結果為 最喜爱食物

第一次測試,邀請400名用戶測試伺服器穩定度,結果發現伺服器CPU滿載,透過更新伺服器硬件,希望改善問題,成果如下:



從圖片可見，伺服器一開始雖然達到80%，但是仍然足夠應付400人同時登入以及運作，因此和預期一樣測試成功。

第二次測試,邀請200名用戶測試登入介面,登入名稱為MID,測試後,用戶平均登入時間超過五分鐘,因此以email取代MID作登入名稱,用戶平均登入時間縮短至30秒.

第三次測試,邀請50名員工和50名商家及200名用戶測試下單介面,測試後,用戶平均送餐時間為1至2小時,同時外賣員反映沒有商家及用戶地址使溝通時間增加,導致過長的送餐時間,因此商家及用戶地址轉為必填,送餐時間縮短至30分鐘至1.5小時.Sql如下:

ALTER TABLE Member,staff

MODIFY MPlace char(100) NOT NULL,

BPlace char(100) NOT NULL;

第四次測試,邀請200名用戶測試購物車界面是否存在異常,測試後,用戶反映登出時購物車不會被清空同時食物的選擇也沒有受到限制,這是因為數據庫已經擁有購物車表格以存儲存儲購物車內容,因此和預期一樣測試成功.

第五次測試,當自行生成編號，讀取Max值後加一時，發現這種方法存在一個效率問題，如果記錄非常大的話，那麽Max()也會影響排序演算法的效率,更嚴重的是同步問題，如果同時有兩人同時註冊時,會讀到相同的Max，加一後插入的ID值會重復,導致數據庫的唯一欄位發生重複的錯誤，令註冊速度變慢。

因此更改為採用數據庫自動編號，使50名員工和50名商家及200名用戶的註冊測試中,由原本的5分鐘減少至至只需要30秒，而且由於編號具有有順序性，可以加快檢索速度，減輕伺服器負擔，同時如果透過非系統增加記錄，例如手動錄入時，不用擔心主鍵重復問題。