資料庫系統專題

第六組

期末報告

M974011034 陳政頤

M974012035 羅佳俊

N984020003 蘇永吉

目錄

[系統功能與需求分析 3](#_Toc250330262)

[Entity-Relational Diagram 4](#_Toc250330263)

[實體關連模式轉換關聯模式 5](#_Toc250330264)

[建置專案環境 7](#_Toc250330265)

[SQL 敘述 7](#_Toc250330266)

[各SQL敘述用 SQL\*PLUS 的執行結果 11](#_Toc250330267)

[系統展示 13](#_Toc250330268)

[課後心得報告 23](#_Toc250330269)

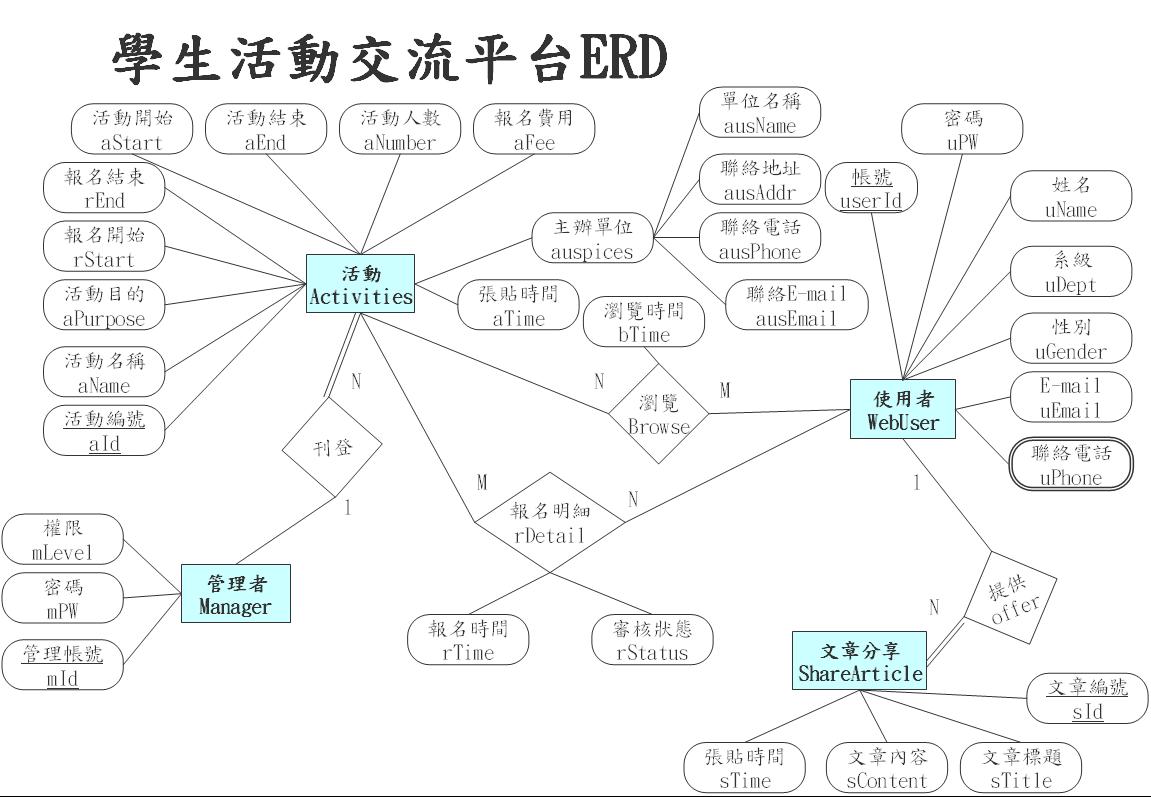
系統功能與需求分析

本系統在於實作有關學生活動交流平台的系統。此系統主要功能在於公告活動訊息、活動內容、活動文章訊息的交流並得以透過資料庫的功能做實作統計功能。系統中需由一名管理者進行網站的管理與維護，以確保網站運作和資料整合的可持續性。有關構建系統的資料需求設計如下：

* 使用者：包括帳號(userId)、密碼(uPW)、姓名(uName)、系級(uDept)、性別(uGender)、Email(uEmail)、聯絡電話(uPhone)，其中使用者帳號為唯一，而使用者可能有多筆電話資料，且使用者在瀏覽活動時，瀏覽時間需加以記錄，另外，使用者在報名活動時，同時需紀錄其審核狀態與報名時間，且一個活動只允許同一個使用者報名一次。
* 管理者：包括管理帳號(mId)、管理密碼(mPW)、權限(mLevel) ，其中管理帳號為唯一，且活動由管理者刊登後，由該管理者負責審核管理。
* 活動：包括活動編號(aId)、活動名稱(aName)、活動目的(aPurpose)、報名開始(rStart)、報名結束(rEnd)、活動開始(aStart)、活動結束(aEnd)、活動人數(aNumber)、報名費用(aFee)、單位名稱(ausName)、聯絡地址(ausAddr)、聯絡電話(ausPhone)、連絡Email(ausEmail)、張貼時間(aTime)，其中活動編號為唯一。
* 文章分享：包括文章編號(sId)、文章標題(sTitle)、文章內容(sContent)、分享者(userId)及張貼時間(sTime) ，其中文章編號為唯一。

Entity-Relational Diagram

根據上述的系統需求資料描述，將之轉換成ER圖後如下圖所示：



此系統經實體關係模式分解系統需求後，共計有四個實體，分別是使用者、活動、管理者、文章分享，分別描述如下：

1.使用者：

(1).聯絡電話：複合屬性，包括行動電話、市話

(2).使用者帳號：唯一值

2.活動：

(1).活動編號：唯一值

(2).主辦單位：多值屬性，其包括單位名稱、聯絡地址、聯絡電話、聯絡E-mail。

3.管理者：

(1).管理帳號：唯一值

4.文章分享：

(1).文章編號：唯一值

實體關連模式轉換關聯模式

實體關係模轉換成關聯模式有幾種步驟：

1.一個實體型態產生一個關聯綱目

(1).主鍵：挑選一個關鍵屬性

(2).屬性：所有簡單和單位屬性加上所有複合屬性~~的~~展開成簡單屬性

2.多值屬性產生一個關聯綱目

(1).主鍵：所有屬性

(2).屬性：所有組成屬性和一個外部鍵

3.弱實體型態產生一個關聯綱目

(1).主鍵：部份鍵屬性和所有外部鍵的組合

(2).屬性：所有組成屬性和外部鍵(參考到每一個主實體型態的關聯綱目主鍵)

4.1:1的關係型態

選擇一個實體型態，在其關聯綱目中加上一個參考到另一關聯綱目主鍵的外部鍵。在決定哪一個實體型態加入外部鍵時，以完全參與此關係型態的實體為優先。

5.1:N的關係型態

選定基數為N的實體型態，並在該關聯綱目中加上一個參考到另一關聯綱目主鍵的外部鍵。

6.M:N的關係型態為產生一個關聯綱目

(1).主鍵：二個外部鍵的組合

(2).屬性：所有關係型態的屬性和二個外部鍵（其分別參考到二個關聯綱目的主鍵）

7.n元關係型態為產生一個關聯綱目

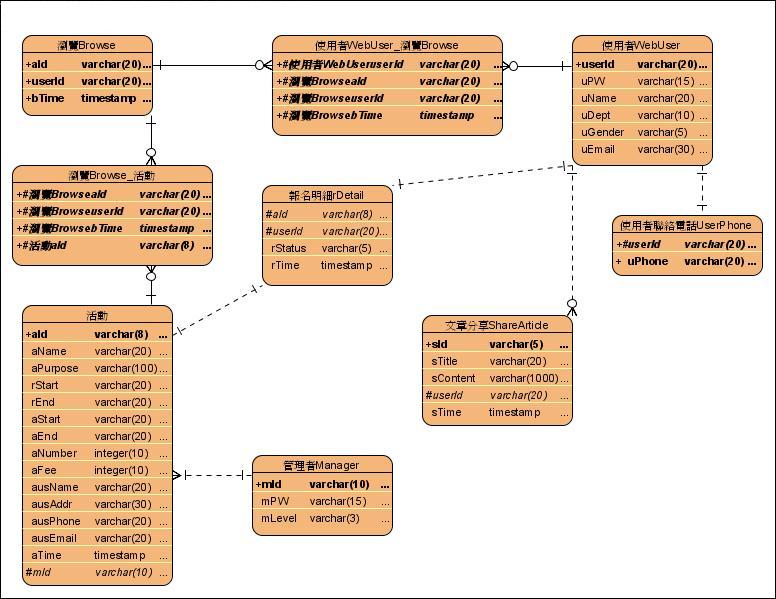
(1).主鍵：若各實體型態上的基數比全為N，則n個外部鍵組合成主鍵。若各實體型態上的基數比不全為N，則選定一個基數為1的實體型態，其他n-1個實體型態外部鍵給合成主鍵。

(2).屬性：所有關係型態上的屬性和n個外部鍵分別參考到所有參與實體型態之關聯綱目的主鍵。

經由以上實體關係圖轉成關聯模式資料庫綱目的原則，我們將學生活動交流平台的ERD轉成如下的資料庫綱目，屬性有加底線為該綱目的主鍵，箭頭所指屬性為外部鍵：



將關聯資料綱目轉換成關聯圖



建置專案環境

在本次專案的工具選擇上，本組擬採用開源碼方式進行，所有工具除了資料庫採用商用授權版本外，開發工具和關聯圖繪製均採用開放原始碼的工具進行。使用的工具和版本如下：

作業系統：Windows XP SP2 or later

開發工具：PHP 5.2.0

網頁伺服器：Apache 2.2.14

資料庫：Oracle 10.2.0.1 + Oracle 10.2 Client

運用工具：Visual Paradigm For UML 7.1

SQL 敘述

在完成關聯綱目後，我們將以SQL語法實作資料庫的資料表。根據關聯綱目的屬性，我們總共分析出七個資料表，分別為Activity、Browse、rDetail、Manager、ShareArticle、WebUser、UserPhone。底下分別使用SQL敘述進行資料表格建表工作。

1. 建立資料表

(1).Activity(活動)

CREATE TABLE Activities (

aId varchar(8) NOT NULL,

aName varchar(20) NOT NULL,

aPurpose varchar(100) NOT NULL,

rStart varchar(20) NOT NULL,

rEnd varchar(20) NOT NULL,

aStart varchar(20) NOT NULL,

aEnd varchar(20) NOT NULL,

aNumber INT NOT NULL,

aFee INT NOT NULL,

ausName varchar(20) NOT NULL,

ausAddr varchar(30) NOT NULL,

ausPhone varchar(20) NOT NULL,

ausEmail varchar(20) NOT NULL,

aTime TIMESTAMP NOT NULL,

mId varchar(10) NOT NULL,

PRIMARY KEY (aId),

FOREIGN KEY (mId) REFERENCES Manager (mId)

);

(2).Browse(瀏覽)

CREATE TABLE Browse (

aId varchar(20) NOT NULL,

userId varchar(20) NOT NULL,

bTime timestamp NOT NULL,

PRIMARY KEY (aId,userId,bTime),

FOREIGN KEY (aId) REFERENCES Activities(aId),

FOREIGN KEY (userId) REFERENCES WebUser(userId)

);

(3).rDetail(報名明細)

CREATE TABLE rDetail (

aId varchar(8) NOT NULL,

userId varchar(20) NOT NULL,

rStatus varchar(5) NOT NULL,

rTime timestamp NOT NULL,

PRIMARY KEY (aId,userId),

FOREIGN KEY (userId) REFERENCES WebUser (userId),

FOREIGN KEY (aId) REFERENCES Activities (aId)

);

(4).Manager(管理者)

CREATE TABLE Manager (

mId varchar(10) NOT NULL,

mPW varchar(15) NOT NULL,

mLevel varchar(3) NOT NULL,

PRIMARY KEY (mId)

);

(5).ShareArticle(文章分享)

CREATE TABLE ShareArticle (

sId varchar(5) NOT NULL,

sTitle varchar(20) NOT NULL,

sContent varchar(1000) NOT NULL,

userId varchar(20) NOT NULL,

sTime timestamp NOT NULL,

PRIMARY KEY (sId),

FOREIGN KEY (userId) REFERENCES WebUser (userId)

);

(6).WebUser(使用者)

CREATE TABLE WebUser (

userId varchar(10) NOT NULL,

uPW varchar(15) NOT NULL,

uName varchar(20) NOT NULL,

uDept varchar(10) NOT NULL,

uGender varchar(5) NOT NULL,

uEmail varchar(30) NOT NULL,

PRIMARY KEY (userId)

);

(7).UserPhone(聯絡電話)

CREATE TABLE UserPhone (

userId varchar(20) NOT NULL,

uPhone varchar(20) NOT NULL,

PRIMARY KEY (userId,uPhone),

FOREIGN KEY (userId) REFERENCES WebUser (userId)

);

1. 功能SQL

(1).新增功能：

SQL敘述：

在Browse的table中，新增一筆使用者ID「M984032121」在’ 2009-11-14 14:34:53’時瀏覽活動「a220007」的記錄。

SQL語法：

INSERT INTO browse (aId, userId, bTime) VALUES

('a220007', 'M984032121', '2009-11-14 14:34:53');

(2).刪除功能：

SQL敘述：

刪除「李四」在2009年的所有瀏覽記錄

SQL語法：

DELETE BROWSE

WHERE to\_char(bTime,'yyyy')=2009 AND userId IN(

SELECT userId

FROM WebUser

WHERE uName='李四');

(3).修改功能：

SQL敘述：

「聯絡電話」數量超過一項的使用者，修改其曾報名過的活動人數為100人。

SQL語法：

UPDATE Activities

SET aNumber=100

WHERE aId IN(

SELECT aId FROM Activities

WHERE aId IN (

SELECT aId FROM rDetail

WHERE userId IN (

SELECT userId FROM WebUser NATURAL JOIN UserPhone

GROUP BY userId

HAVING COUNT(uPhone)>1)));

(4).查詢功能：

SQL敘述：

那些被「張三」瀏覽過的活動，列出現在報名該活動的使用者姓名與系級

SQL語法：

SELECT DISTINCT userId,uName,uDept

FROM WebUser

WHERE userId IN (

SELECT userId

FROM WebUser NATURAL JOIN rDetail

WHERE aId IN (

SELECT aId

FROM WebUser NATURAL JOIN Browse

WHERE uName='張三'));

(5).統計功能：

SQL敘述：

曾報名活動人數超過60人的使用者，列出其姓名與其分享文章的數目(沒分享者也要保留)

SQL語法：

SELECT DISTINCT uName,COUNT(sId)

FROM WebUser U LEFT OUTER JOIN ShareArticle S ON U.userId=S.userId

WHERE U.userId IN (

SELECT userId

FROM rDetail NATURAL JOIN Activities

WHERE aNumber > 60)

GROUP BY uName;

各SQL敘述用 SQL\*PLUS 的執行結果

(1).新增功能



(2).修改功能



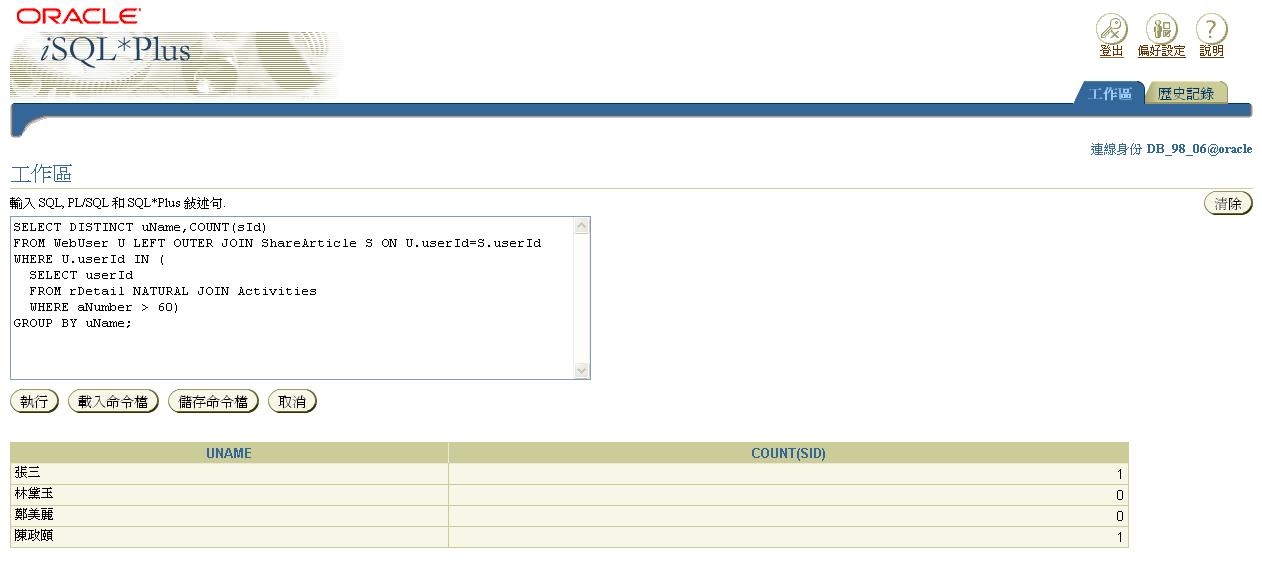
(3).刪除功能



(4).查詢功能



(5).統計功能



系統展示

實作完系統的資料庫規劃、設計與轉置到系統面後，運用開發工具的實作，我們將整個系統預先規劃方式和預期開發後的結果，完整實做出可執行系統。由於系統執行前需要先進行系統的環境建置，此部份不在撰述範圍，可參考各軟體的安裝方式建置系統環境。設定需注意事項有幾點：

1. 系統預設傳回資料採用的語系為UTF-8
2. 假設使用者是全部應用軟體都安裝在本機，則要另外注意Apache預設的啟動目錄
3. Apache 和PHP 的組態設定，此二個組態設定檔請自行參考手冊設定

下面將對系統做操作說明。

管理者帳號、密碼：admin / admin

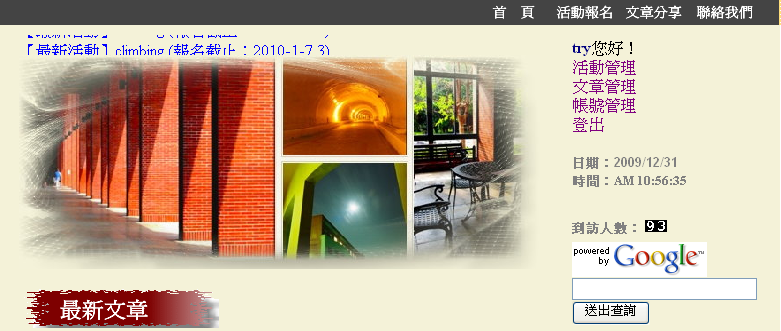
使用者帳號、密碼：try / try

1. 系統首頁

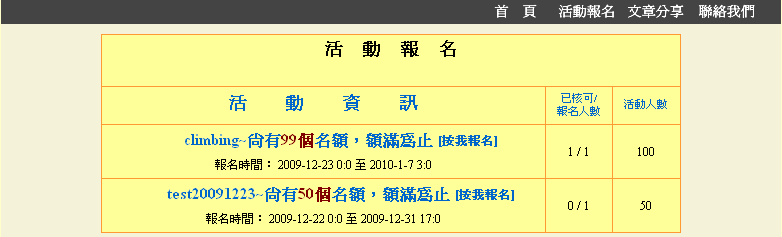
使用者透過瀏覽器，執行系統的 index.php即可開啟首頁。



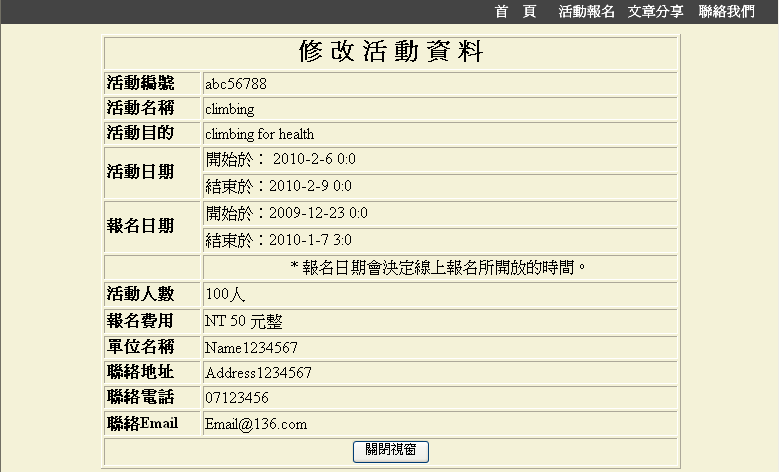
1. 使用者登入畫面



1. [活動報名]畫面



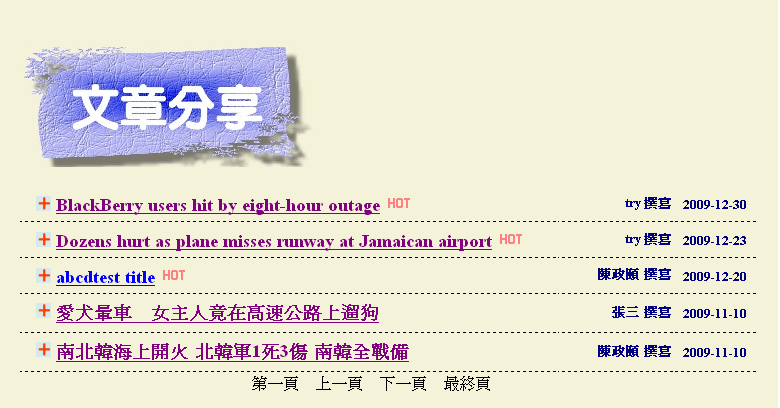
1. 點擊[活動尚有名額]



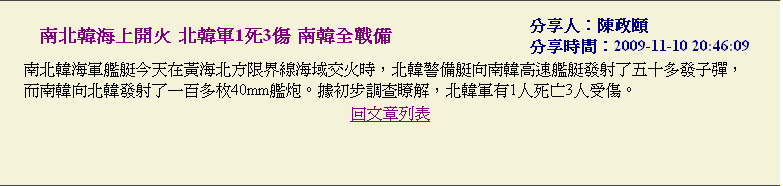
1. 點擊[按我報名]



1. 文章分享



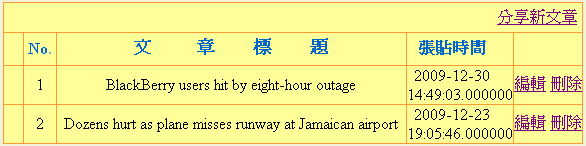
1. 點擊其中一篇文章



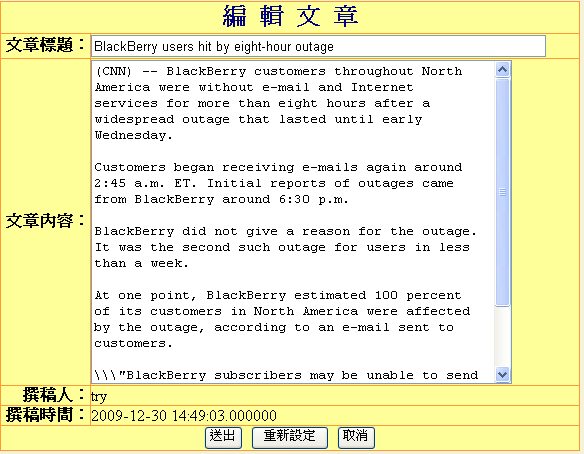
1. [活動管理]🡪沒有權限



1. [文章管理]



1. [編輯文章]



1. [新增文章]



1. [帳號管理]



1. [管理者]登入畫面



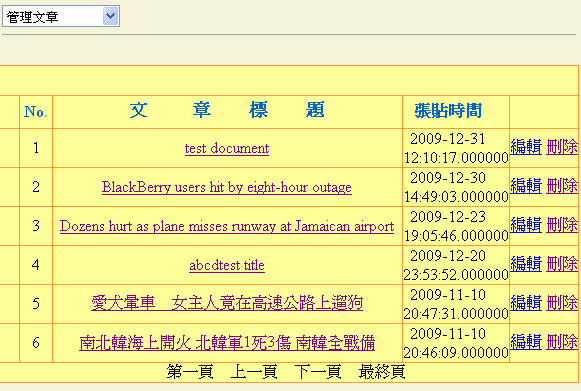
1. 管理者進入[管理頁面]



1. 管理者使用[管理者帳號管理]



1. 管理者使用[管理文章]



1. 管理者使用[管理活動]



1. 管理者[新增活動]



1. 活動名單[審核報名]



1. 修改[活動資料]



1. [核可名單]



1. [刪除名單]



課後心得報告

企研二 M974011034 陳政頤

隨著課堂進入尾聲，也讓我回想著整個過程的心情轉折。

還記得剛開學的前兩週，聽過老師課程的簡介與詳細看過Syllabus上洋洋灑灑的眾多規定後，內心想著：「這課的loading會不會太重了點？」，甚至再加退選截止的前夕，還一度猶豫著要不要乾脆把這門課給退掉，改修一門輕鬆一點的課。

不過這時內心也有另一股聲音不斷的鼓勵著自己，原因也正因為課綱上詳列了這麼多內容，表示老師在這門課上的用心，如果我能在這門課中持續堅持學習下去，肯定能從中獲得良多，「同樣是每週花三個小時到課堂上課，與其選一門輕輕鬆鬆但沒有收穫的課，為什麼我不選一門能讓我終身受益的課呢？」最後這樣的想法讓我堅持到現在，而我也正因為這樣的想法，讓我有機會把以前相當薄弱的資料庫概念加以強化，進而提升。

還記得幾年前，在快退伍前，曾去報考過高普考，當時考資訊類，雖然大學時是讀資管系，但在資料庫的科目上，卻考的慘不忍睹，印象中我連題目中一個簡單JOIN的SQL語法都寫不出來，而如今經過一個學期的學習下來，開始訝異自己當年怎麼會連這麼簡單的題目都答不出來？

課堂中，雖然期中期末考、Homework、習題作業甚至加上Project，份量可謂不輕，但隨著課堂的安排，一個接著一個密集的練習，反而讓觀念上能不斷的複習強化，如今回頭，反而開始感謝有這些練習與作業，不然很難想像一個學期過去，我還能留下多少印象。

而對於期末的Project，雖然主題可以各自發揮，而且在學期初就已宣布，不過整個系統建構下來，還是花了不少心力與時間，即便之前就已寫過PHP與MySQL的網頁程式，但在Project要求使用Oracle的前提下，也著實讓我吃了不少苦頭，但也如同老師跟我說的一樣：「趁著還在學，有機會多練習業界最常用的Oracle，對你會很有幫助的。」確實，隨著程式一點一點建構完成，不但一步一步將書本的理論實踐，而且隨著程式完成的比率，內心就越感到充實，有人說「克服困難是最好的學習」，我想大概就是指這樣的情況吧。

最後很感謝老師對這門課如此的堅持與用心，才能讓我們收穫良多，我想這兩年的研究所，資料庫將會是最讓我印象深刻的課堂之ㄧ。

資管碩班 N984020003 蘇永吉

以前學過資料庫，也常在用資料庫寫程式，不過在上完課後覺得一般常用的資料庫使用方式都有問題。當然在一般寫程式的Coding 上，大部份程式設計師只專注處理演算法就好，不會去特別考慮資料庫的問題，因為那是資料庫管理員的事。所以很多時候就會使程式在設計開發階段就影響很多方面的系統問題。資料庫管理員也認為不需要太過著墨在程式設計上（也許是SQL的設計程式），因為平常的資料庫維護工作就已經佔據太多時間，所以若未稍加注意資料庫的維護與調校，那麼在長時間的用戶需求變更後，產生的資料庫問題會愈來愈影響效能的展現。在這門課中和以往使用資料庫的方式和學理背後的差距肯定是有的，不過在學習上因為是有相關經驗的關係，反而對資料庫的焦點集中在資料庫能怎麼用、該怎麼用；它能帶給我在原本對資料庫的認知上有什麼不同的體會。

在這學期課程後，覺得真的在學理上的探索需要再加強。雖然資料庫在很多書籍上都可以看到，但要有比較深入論述的教科書卻很少，工具的使用若僅僅是必要之好，那麼基礎的建立，顯示是必要之惡。很多在業界的使用上，為因應市場的需求改變，而必須快速回應系統的彈性時，常常面對的是永無止境的痛苦與無奈的系統調整與轉變，但是如果有較好的基礎和正確觀念，那麼在轉變的過程中，它似乎不再是專注的焦點，反而對於現象的觀察與正確理解才能讓工作在變化無常的需求變動上，更快更正確的達到目的，同時又能讓往後的擴展上取得良好的基礎。

企研99級乙班 M974012035 羅佳俊

本身學歷一路是企管出身，由於本身對於資訊的興趣以及原本任職公司有培訓福利的號召之下投身資訊業，也因為對於基礎知識的缺乏因此工作時常遇上瓶頸，於是趁著目前還有學生身份來學習相關課程。資料庫課程設計相當繁重，舉凡個人作業、小組作業甚至專案報告，皆是必須將課本熟讀才有辦法完成，當然考試更不用說了。每每個人作業公佈初始，我都必須花好幾天將課本熟讀之後才有辦法進行作業，邊寫邊修正答案，而且對於答案也不具有百分百的信心，本課程相當具有難度。原本工作時以為會下SQL命令就代表懂得基本資料庫觀念，甚至原先公司表格也很少作到第二正規化及以上(不過實務上某些情境做正規化會引起更多麻煩)，上課之後深深覺得自己不僅不懂資料庫，連SQL命令都不算懂得使用。隨著課程的進展也讓我碰觸到許多以往不曾考慮過的面向，例如交易處理、查詢成本概念等，許許多多參照老鳥作法理所當然的方式皆有了更進一步的瞭解，實在受惠良多。

在學習過程中辛苦是免不了的，每每繳交作業時，我都必須花上數天熟讀課本以及練習作業，但許多基礎觀念是以前從未接觸過。課程初始時老師宣佈了課堂作業分組以及專案作業分組的教學方式，當時對於這種分組方式相當疑惑，隨著課程逐漸艱深，越來越多時後必須仰賴同組本科系出身同學的幫助才有辦法進行作業，在此也要特別感謝正頤與琇鈞。印象最深刻是在查詢最佳化的課程時，琇鈞花了數小時教我B+Tree的畫法，只因為我沒有學過資料結構，可以說若是沒有政頤與琇鈞的幫助，許多作業可能我都無法完成。而很多次的錯誤答案也在與小組成員包括永毅的討論下大家協助修正。