

資料庫管理 (114-1)

期末專案

在這個期末專案中，每組將開發一個包含簡單前端使用者介面的資料庫應用程式，主題自選。助教們會附上一份寫好的範例供大家參考，但當然那是最低限度的範例，我們期待各組都做得比範例更完整、更好。由於這是一個專案而非作業，我們不會限定大家的方向或主題，只要是「一個在真實世界會有夠多人想要用的資訊系統」並且「這個資訊系統很自然地需要一個資料庫」就好。我們期待看到的是(1) 大家能設定盡量貼近現實世界的題目，使用盡量真實的資料（或者如果不可能，那也要創造盡量真實、合理的資料），並盡力實現相關功能，(2) 大家能分析問題並且制定出好的系統架構和分工，(3) 大家能針對這個專案設計合理、恰當的資料庫及相關程式，(4) 大家能良好地團隊合作（這很重要！），(5) 大家能清楚生動地展示你們的想法和成果。

希望大家都能享受做專案的樂趣。祝大家好運！

1 任務概述與重要日期

所有與期末專案相關的日期都如表 1 所示。每組要在學期末的截止時間 C1 繳交一個完整的資訊系統（包含程式碼和資料庫備份檔）、書面報告，以及展示系統功能與設計的展示影片。此外，每位同學也應該在截止時間 C2 紿交其他組和自己的組員回饋。為了減輕大家在期末的負擔，並且讓授課團隊可以及時給予大家回饋，在截止時間 B 會讓大家先交一部分成果（主要是關於系統功能與資料庫的設計），收到回饋後可以做調整再繼續進行下一階段。也因此，每一階段繳交的東西可以拿之前繳交的來複製貼上（如果大家沒打算改），也可以修改甚至重做之前階段繳交的東西¹。

任務	時間	截止時間編號
分組	10/8 週三 08:00:00	A
專案諮詢 (1)	10/23 週四 12:30–13:30	–
專案計畫書繳交	10/29 週三 08:00:00	B
專案諮詢 (2)	12/5 週五 12:30–13:30	–
完整報告、系統、展示影片繳交	12/10 週三 08:00:00	C1
展示影片欣賞與回饋	12/24 週三 08:00:00	C2

表 1: 重要日期

¹兩個助教諮詢時間中，第一個原本在課程大綱上寫 10/24，但 10/24 是假日，因此提前一天。

2 分組

請大家自行分組，一組應該有恰好三位修課的學生（旁聽生不計）。一個同學不能參加兩組。因為專案的目的是讓大家練習團隊合作（或至少感覺到它的困難），因此有最低人數限制；因為也要讓每個同學都有合理的付出，所以有最高人數限制。

如果你想徵求組員，可以上 NTU COOL 在一篇討論「期末專案分組揪團區」底下回文。如果你有組員但你們組還有名額可以容納其他同學，請在繳交組員名單時註明是否願意被隨機指派其他同學當組員；如果你找不到組，請在截止前報名「我要請助教幫我隨機分組」。在截止前沒跟著組報名或沒報名隨機分組的人都會被隨機分組；但在截止前什麼都沒講的，或者沒報名又不接受隨機分組的，會以**期末專案總成績扣 10 分**做為懲罰。

3 專案任務：一個包含前後端的資訊系統

各組應該完成一個有前端系統（使用者操作介面）、後端系統（後臺演算）和資料庫的完整資訊系統，來實現他們的專案內容。

3.1 使用者與功能

一般而言，會跟系統或資料庫互動的有「一般使用者」、「業務經營者」、「資料分析師」、「系統管理員」這幾種角色，而大家的系統在專案中要涵蓋到的包含一般使用者和業務經營者。以下我們簡要說明這些角色。

一般使用者是這個系統最主要的使用者，他們除了可以做基本的查詢，也可以新增修改刪除屬於自己的資料和交易紀錄；業務經營者是另一種使用者，他們可以透過這個系統經營這項業務，主要包含維護在這個系統上供一般使用者存取的資料；資料分析師並非系統在業務邏輯上的使用者，但他們可以分析資料庫中的資料，以幫助業務經營者得到有用的、可以幫助提升營業成效的資訊與洞見；系統管理員是資訊人員，擁有所有的權限，可以做所有的事，甚至下架這個系統。舉例來說，如果主題是臺大選課系統，那一般使用者主要是學生（可以查詢課程、選課），業務經營者包含教師（可以新增修改刪除課程資訊，但不能新增修改刪除課程）、系辦職員（可以新增修改刪除課程、做一些設定）、教務處職員（可以新增修改刪除課程、做很多設定）等等，資料分析師可能是教務處的一些同仁，可以分析學生都在什麼時間點選課、選上午還是下午的課、哪些課被僑外生選得最多等等，系統管理者則可能是學校的計算機中心或教務處的資訊人員。如果主題是高鐵訂票系統，那一般使用者主要是乘客，業務經

營者主要是高鐵票務部職員，資料分析師是高鐵分析部門的職員，系統管理者則主要是高鐵資訊部的職員。

在本專案中，請假設資料分析師會獲得完整的資料庫讀取權限，他們會直接對資料庫做分析，因此在系統上不需要讓資料分析師可以登入與操作。簡言之，資料分析師「不會」使用這個系統，你也不用為他們設計功能。同樣地，你也可以假設系統管理者「不會」使用這個系統。但你的系統應該涵蓋一般使用者和業務經營者，並且為他們規劃和開發他們需要的功能。

3.2 系統架構

關於前端，原則上我們只期待各組做純文字介面（主控台模式，又稱為 terminal 或 console），各組只要有功能完整、正確、夠好的純文字使用者介面，這一項就可以得到滿分。如果有同學真的很想要自己的系統有圖形介面（graphical user interface），他們可以選擇做網頁、手機應用程式或視窗應用程式，也不限語言，但請務必留意本課程的重點在資料庫，所以花時間把前端做好主要是為了自我挑戰與實現，對成績幾乎沒有幫助。

雖然在前端只需要純文字介面，但這個系統應該要遵循 client-server 架構，亦即後端系統和資料庫應該在一臺伺服器上，而前端系統可以在另一臺機器上被執行。考慮到大家不一定都有真的伺服器可以用，最終展示系統時大家可以在單一臺機器上執行前端（亦即讓前端系統去連 localhost）、後端和資料庫，但必須要開至少兩個前端程式去連同一個後端系統，並且展示出正確的資料同步與併行控制。

這個系統應該要有一定的規模和複雜度，並且包含至少兩個資料庫，一個是負責儲存交易資料的關聯式資料庫，另一個是負責儲存行為資料的 NoSQL 資料庫。舉例來說，如果是臺大選課系統，我們應該會把課程、系所、教師、學生、選課紀錄、退選紀錄等等儲存在關聯式資料庫中，而把查詢紀錄、檢視課程資訊的紀錄等等儲存在 NoSQL 資料庫中。

3.2.1 關聯式資料庫

我們不限定要使用哪一款資料庫，只要是一般常見的關聯式資料庫即可；如果不確定你們組要用的是否可以，可以先問授課團隊。資料庫中應該要包含資料。如果你們的專案中的某些資料表可以被放入真實資料（例如臺大選課系統中的系所、課程），我們會期待大家這麼做；對於無從取得真實資料的資料表（例如學生的選課紀錄），我們則期待大家自行生成盡量接近真實的合理資料。

由於要展示大家不僅會開發系統，也有能力處理大量資料，因此大家的關聯式資

料庫中應該要有至少一兩個資料表內很自然地會有大量資料（例如學生們的選課紀錄），並且該系統中的某些功能會需要處理這些大量資料（例如教務處同仁可以查詢每門課被多少個外生選）、某些查詢會需要合併數張資料表。在一般的情況下，我們期待你的關聯式資料庫中至少有十張以上的資料表，以及至少有一張資料表中有上萬筆資料。

我們期待看到大家的關聯式資料庫綱要（relational DB schema）有良好的設計，也期待看到大家在正規化方面的考量與取捨。我們也期待看到大家幫資料表建立索引（index），並且在系統中良好地處理交易管理和併行控制。我們當然也期待看到大家有能力寫 SQL 處理資料，進而做出有一定難度和意義的分析。不管怎樣，大家應該評估上課教過的重要設計原則，例如要不要在某些資料表上設計「status」這種欄位來表達「已刪除」並且避免使用 `Delete`；不是絕對不能用 `Delete`，而是大家不論要做什麼設計，都應該確保那會讓系統的行為符合大家的期待²。請大家將這些期待納入考量吧。

3.2.2 NoSQL 資料庫

由於課程對 NoSQL 資料庫的著墨較少，因此我們對大家在系統中納入 NoSQL 資料庫的要求也較少。大家只要設計一個的確適合 NoSQL 資料庫的場景，把那些存進 NoSQL 資料庫比關聯式資料庫更適合的資料確實存進 NoSQL 資料庫，並且設計和實作相對應的資料庫架構和分析功能即可。具體來說，它可以只有一張表，儲存一種紀錄（例如搜尋課程或檢視課程的紀錄），並且被一項分析工作使用到。

4 期末要繳交的完整成果

在本節中，我們會先敘述截止時間 C1 大家要繳交的完整內容，包含完整的系統、書面報告，以及展示影片。後續我們會接著說明在截止時間 B 要繳交的計畫書中要包含完整內容中的哪些部分，最後說明截止時間 C2 前要完成的展示影片回饋。

4.1 完整的資訊系統

各組應於 GitHub 上傳小組專案的所有程式碼，以及專案的資料庫備份檔。GitHub 上也應該要有一個 `README.md` 檔案，該檔案應該包含專案簡介、安裝說明和一個簡短的

²以往年為例，許多組沒有設計「status」這種欄位，又把所有 Foreign Key 的 on-delete referential trigger 都設計為「cascade」，又讓使用者要刪掉資料時是真的執行 `Delete` 語法，結果就是導致大量不該刪掉的資料被刪掉（例如某個使用者訂票後刪除自己的帳號，他的訂票紀錄也都被刪掉了），而這很可能並不是系統期待的。

示例或教學。助教會依照你的指示進行安裝或運行，如果助教在安裝或運行過程中遇到問題，視情況嚴重的程度，你的小組會失去專案成績最多 10 分。

4.2 書面報告

在書面報告中，需要包含系統分析、系統設計、系統實作、分工資訊，以及專案心得。書面報告可以用中文或英文撰寫，最多二十面 A4 紙。在報告的第一頁，請寫上所有組員的學號、姓名、在繳交計畫書後被指定的組別編號、GitHub 專案連結、展示影片的 YouTube 連結。書面報告應該被以 PDF 檔的形式上傳到 NTU COOL。

4.2.1 系統分析：系統有哪些功能

在這一部分，大家應該說明系統中有哪些使用者、分別扮演什麼角色，以及他們可以在系統上執行的操作。

實務上系統分析常常是系統開發的整個流程中最關鍵、最耗費人力和時間的階段，要決定哪些人可以用這個系統、可以怎麼用、使用的流程和步驟順序、一件事如何在數種角色的數個人之間流動、系統有哪些功能等等。但因為（至少資管系的）大家之後還有「系統分析與設計」這門課，在那邊會更嚴謹、仔細地訓練大家如何做系統分析，所以在這個專案中，同學們只要簡單地做這一塊即可。具體來說，大家原則上只要用文字敘述即可，不需要照業界慣例去畫流程圖、使用者案例圖等等。如果真的想畫圖幫助說明，用任何方式畫圖均可。

在說明系統功能時，對於正常系統都應該要有的功能，包含創建帳號、使用者登入、修改密碼等等，在文件上只要一筆帶過即可（但如果大家有很特別的設計，還是可以介紹）。換言之，在說明系統功能時，只要著重在說明「你若不說，別人不一定會知道」的功能即可。

4.2.2 系統設計：如何實現前述功能

在關聯式資料庫方面，應該要有 ER diagram 及相關文字說明、Relational schema diagram 及相關文字說明，以及資料字典 (data dictionary)，透過這些來讓讀者瞭解你們的資料庫長怎樣、如何支持系統運作。大家也應該有一些段落說明自己怎麼做資料庫正規化 (normalization)、怎麼取捨要做到哪個程度；如果實在不知道該寫什麼，就對自己準備使用的 schema 做分析，說明自己的 schema 是在哪一種正規型式。

在前端方面，通常此時業界人士會針對主要部分畫出使用者介面的設計，據此說明使用者可以或應該如何操作系統，結果又會如何被呈現；如果前端如本專案期待的

是個純文字介面，就不會「畫出」使用者介面的設計，但還是會說明使用者可以在哪些場景輸入什麼然後得到什麼。由於本課程著重在後端，在前端方面我們就不做特定規範，大家用自己喜歡的方式簡要地說明系統提供哪些功能給每種角色，而每種角色又該依序執行哪些步驟以使用這些功能即可。如果寫得夠清楚，純粹使用文字說明也是可以的。

在前後端整合方面，請大家說明自己的整個系統是如何實現 client-server 架構，具體來說就是說明系統的哪些部分是規劃要被放在伺服器上，哪些部分則被規劃要放在使用者的前端上，前端與後端之間又是藉由什麼樣的方式溝通、交換資料、傳遞資訊。雖然有多種使用者，但大家完成的系統中應該只有一個前端系統，不同的使用者登入後依照各自的權限，可以做不同的事。

最後，在 NoSQL 資料庫方面，只要有簡單的文字說明即可，形式不拘。

4.2.3 系統實作：實際開發時的考量

系統實作階段，其實就是把系統開發出來。即使如此，在文件方面也要請大家針對開發階段的考量做相對應的說明，包含資料建置方式、資料處理與分析、效能提升方面的做法，以及交易管理和併行控制。

首先，大家應該說明資料庫中所存之資料的資料建置方式，也就是你們的資料是從何處取得或基於什麼原則被建立的、各資料表中各有多少筆資料。

其次，請大家針對系統中至少十個重要的功能，在文件中寫下該功能要被完成的（一或數個）SQL 或 NoSQL 指令，並且說明該指令被執行後系統會發生什麼事。這裡的功能其中至少五個要是一般使用者或業務經營者能操作的功能（例如系辦同仁新增課程、學生選課），它們應該要是這個系統中最常被用到的幾個功能；至少五個則是系統管理員或資料分析師會去執行的分析功能（例如看看哪些英語授課課程其實沒有外籍生），它們應該要有一定的複雜度和輔助營運決策的價值，且至少有一個牽涉到你的 NoSQL 資料庫。請注意即使大家真正實作的時候並沒有寫 SQL（例如有些同學可能用了 ORM 架構），在這裡也還是要把各功能對應的 SQL 指令寫出來。

接著要請大家說明為了讓前述 SQL 指令中最複雜、需要最長執行時間的指令，你們做了什麼分析與優化（例如比較了不同 SQL 寫法的效能），以及幫哪些資料表用怎樣的語法建立了怎樣的索引（index）、這些索引又帶來了多大程度的效能提升。如果要能好好地寫好這個部分，應該會需要在資料庫裡面找一些資料表放入大量（例如幾百萬筆）資料。

最後，請說明你們的系統中哪裡需要做交易管理和併行控制，並說明你們如何處理。

4.2.4 分工資訊與專案心得

報告中應簡明清楚地描述每一位組員負責專案中的哪些項目。最後，每個組員都應個別寫下一段簡單的心得感想，去描述自己在專案中學到的東西、體驗到的困難，以及其他任何想說的話。

4.3 展示影片

請錄製一個展示影片，將影片傳到 YouTube 之後，將影片連結放在書面報告的第一頁。你可以設定你的影片為有連結的人才看得到（在 YouTube 這叫做 unlisted）。繳交的影片連結將被在 NTU COOL 上公開給全班同學欣賞、互相觀摩。除了影片連結，各組還必須附上最多 100 字的專案摘要、一張系統執行螢幕截圖，以及製作影片時使用的投影片 PDF 檔，供班上同學快速瞭解影片內容。

每組的影片長度以十到十五分鐘為原則，稍微超過沒關係，超過太多者將被酌予扣分。請先口頭說明你們要做什麼、為什麼它被需要、被誰需要、有什麼功能，接著執行你們的系統給大家看、盡量地展示它的功能（包含前面說的執行兩個以上的前端程式）。然後請詳細展示你們的系統設計，包括資料庫設計、正規化方面的考量或分析等等。再來請說明你們在蒐集資料、資料庫效能調校、交易管理和併行控制等方面的工作。最後，請說說你們的心得感想與未來展望。

各組可以自行決定由幾位同學製作、口頭報告，也可以自行決定報告者的臉是否要出現，但如果報告的聲音聽起來太像 AI 發出的，分數不會太高。不管怎樣，請努力讓這個簡報顯得有趣且專業吧！

5 學期中要繳交的計畫書

在截止時間 B 前，各組應該將專案計畫書 (proposal) 的書面報告 PDF 檔繳交到 NTU COOL。計畫書內容應包含 (1) 系統分析 (2) 系統設計的關聯式資料庫部分（不包含 NoSQL 的部分）。可以用中文或英文撰寫，最多**十二面 A4 紙**，在報告的第一頁包含所有組員的學號、姓名，以及在繳交分組表單後，統一公告的組別編號。

6 展示影片回饋

每一位修課同學都必須挑選至少 10 組其他同學的作品做欣賞，並在看完影片之後填寫表單提供評分與回饋。給每一組的文字回饋需要至少有 50 字。每一位修課同學也都必

須針對組內所有其他同學的貢獻度給予評分與回饋，給每一位組員的文字回饋需要至少有 10 字。

上述評分與回饋應在截止時間 C2 之前透過指定的表單上傳。這些評分與文字回饋都會被匿名地提供給各組做參考。授課團隊保證各組和每位同學收到來自其他同學的回饋時，不會看到給該評分和撰寫該回饋的同學的個人資訊。

7 評分標準

這個專案的評分標準如下：

1. 專案計畫書（20 分）：

- (a) 主題的有趣、創意、實用性、必要性（5 分）。
- (b) 專案範疇、資料複雜性、功能完整性（5 分）：這個系統的範疇是否夠大（通常夠大的系統中的資料庫會有不少資料表）、功能是否合適且完整。
- (c) 資料庫設計（10 分）：這個系統的關聯式資料庫是否能確實支持我們想要實現的功能，並且在彈性、擴充性、容易理解的程度、避免資料冗餘與不一致等面向表現良好。

2. 專案文件、系統、展示影片（80 分）：

- (a) 主題的有趣、創意、實用性、必要性（5 分）。
- (b) 專案範疇、資料複雜性、功能完整性、資料量與資料真實性（10 分）：這個系統的範疇是否夠大（通常夠大的系統中的資料庫會有不少資料表）、功能是否合適且完整，以及資料庫中是否有大量真實或接近真實的資料。
- (c) 資料庫設計（20 分）：這個系統的關聯式資料庫是否能確實支持我們想要實現的功能，並且在彈性、擴充性、容易理解的程度、避免資料冗餘與不一致等面向表現良好，以及這個系統的 NoSQL 資料庫是否被賦予合適的任務，並且能確實支持我們想要實現的功能。
- (d) 資料庫效能與實作（20 分）：這個系統的關聯式資料庫是否在建立索引、關鍵的 SQL 指令的優化與分析、交易管理和併行控制等面向表現良好。
- (e) 使用者介面以及前後端系統架構（5 分）：在本專案中前端系統並不是重點，client-server 架構也不是重點，但如果良好的使用者介面和合適的前後端系統架構，還是會高分一點。

- (f) 組間回饋（10 分或更多，最多 20 分）：一位同學若能確實完成指定數量的組間回饋（有給分數、字數足夠），可以得到 5 分，反之則為 0 分。超過指定數量後每多給一組回饋再加 0.3 分。若回饋亂寫被授課團隊發現，會被酌予扣分。另外 5 分則為其他組給這位同學所屬之組的分數平均。給予其它組回饋和分數時，請不要考慮他們是否有做圖形化的前端介面。
- (g) 組內回饋（10 分）：一位同學若能確實給所有組員回饋（有給分數、字數足夠），可以得到 5 分，反之則為 0 分。若回饋亂寫被授課團隊發現，會被酌予扣分。另外 5 分則為其他組員給這位同學的分數平均。

以上所有評分項目中有繳交期限的，遲交 6 小時內該項成績打九折，6 至 24 小時該項成績打五折，超過 24 小時則該項成績以零分計算。