

University Alumni Tracking System

113-1 資料庫管理 期末專案

B11705048 林杰

B11705022 李訓佑

B11705059 劉士銓





適用對象

- 校方行政人員
 - 可能是校友中心的員工
- 校友
 - 包括普通校友
 - 以及校友會的幹部
- 資料分析師
 - 可能是教務處的員工



系統目標與價值

1. 協助校方追蹤校友的以下資訊：
 - **捐款**資訊
 - 畢業後的**就業**資訊
 - 畢業後所得到的各種**獎項與成就**
2. 提供校友會一個發布訊息的平台，讓校友們Connect
 - 張貼**活動**訊息
 - 登記校友參與活動
 - 幹部**權限**交接
3. 提供資料分析師詳細的校友資料
 - 例如：教務處的**資料分析師**可以透過此系統分析畢業生薪水



系統功能：普通校友 (Alumni)

1. 查看**個人資訊**、更新個人資訊
2. 新增自己的**工作經歷與薪水**
3. 查看自己被記錄在系統中的**豐功偉業**
4. 查看過去自己的**捐款紀錄**
5. 查看所屬校友會相關資訊
 - 各校友會的**聯絡方式**
 - 校友會所舉辦的**活動**

```
=== Select the section you want to enter ===  
1. Profile  
2. Career  
3. Achievements  
4. Donation  
5. Alumni Association  
6. Exit  
===== Enter your choice: █
```

```
=== Alumni Association ===  
1. View your affiliated association  
2. View your association events  
3. View all alumni associations  
4. View all upcoming events  
5. Join an alumni association event  
6. Join an alumni association
```



系統功能：校友會幹部 (Alumni)

1. 查看自己所管理的校友會之會員名單
2. 對校友會名單新增／刪除成員
3. 新增／刪除校友會活動
4. 對於某個活動新增／刪除參加的校友
5. 指派另一個成員為幹部
6. 卸下幹部職位 (失去權限)

What do you want to do?

1. View Member List
2. Add a new member
3. Delete a member
4. Add an event
5. Delete an event
6. Add a participant to an event
7. Delete a participant from an event
8. Add another alumni the Position of Cadre
9. Exit Cadre Position

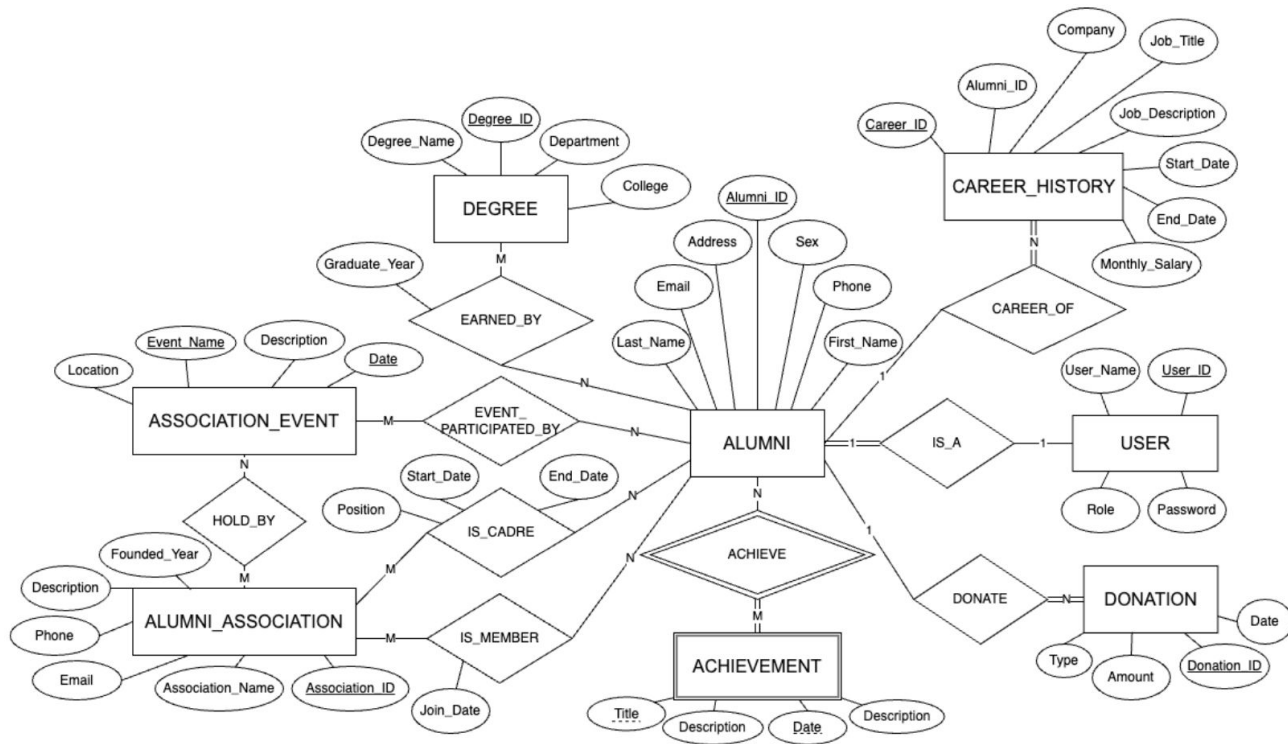


系統功能：學校行政人員 (Admin)

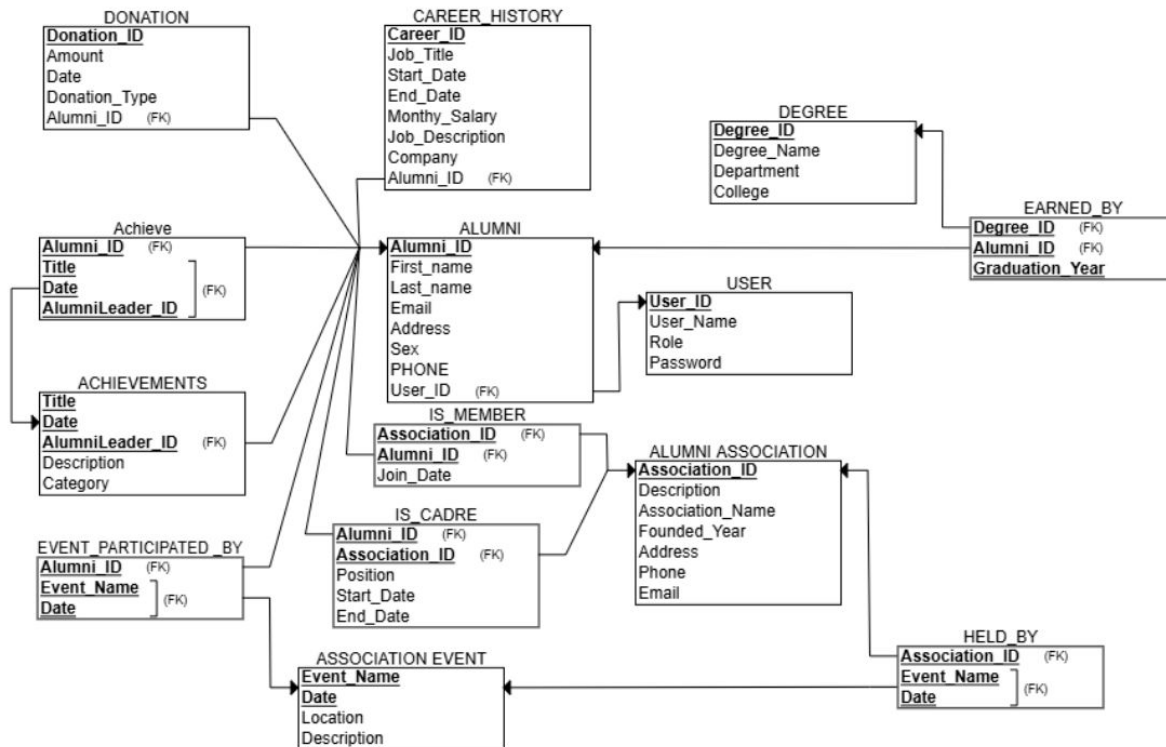
1. 新增／刪除校友的捐款資訊
 - 我們假設有另外的捐款系統，此系統僅作為紀錄使用
2. 新增校友帳號
 - 針對每一屆畢業的畢業生，Admin 應該要為每個人新增帳號
 - 必要時也可以進行帳號刪除、資料修正
3. 新增／刪除校友成就
 - 當得知校友的獲獎訊息，將其記錄到系統中

```
=== Select the section number you want to enter ===  
1: Donation Information  
2: Alumni Account  
3. Achievement  
=====
```

系統設計: ER Diagram



系統設計: Relational Database Schema Diagram





系統 Demo



正規化分析

- **1NF**: Schema 中每個關係的屬性都是簡單且單一值的。
 - 例如: 我們考慮到一個 User 可能從一間學校得到複數個學位, 所以利用 **EARNED_BY** 這張表紀錄學位與 User 的關係。
- **2 NF**: 如果每個關聯中的非主屬性都沒有部分功能相依於任何候選鍵, 則滿足第二正規形式。
- **3 NF**: 如果每個關聯中的非主屬性都沒有遞移相依於任何候選鍵, 則滿足第三正規形式。我們的模式也符合該標準。
- **BCNF 和 4NF**: 如果每個關聯中的每一個功能相依的左側部分都是超鍵, 且不存在多值相依, 則滿足這些正規形式。我們的模式也符合這些標準。






資料庫與系統建置方式

- 資料庫：
 - 我們使用 PostgreSQL 作為資料庫系統。
 - 提供穩定的交易管理與併行控制。
- 系統框架:Flask Framework
- 系統運作流程與上課所提及之類似：
 1. **Client** 發送請求:通過前端與使用者交互。
 2. **Server** 處理請求:與資料庫互動, 執行查詢或數據處理。
 3. **資料庫**:回傳所需資料。



資料來源說明

- 校內提供：
 - 由台大教務處提供「本校學生學號各碼代表之意義」的資料。
 - 選取了 64 個科系建立 **DEGREE** 表。
- 資料生成：
 - 因為校友資訊涉及個人資料，所以我們使用假資料代替。
 - 使用 Python 的 **Faker** 套件生成模擬數據。

degree_id [PK] character varying (5) 	degree_name character varying (150) 	department character varying (150) 	college character varying (100)
1010	Bachelor in Chinese Literature	Chinese Literature	College of Liberal Arts
1011	Bachelor in Chinese Literature	Bachelor's Degree Program in Chinese Literature for Internat...	College of Liberal Arts
1020	Bachelor in Foreign Languages and Lit...	Foreign Languages and Literatures	College of Liberal Arts
1030	Bachelor in History	History	College of Liberal Arts



資料生成: 範例

- 校友會資料與捐款紀錄資料:

association_id [PK] integer	association_name character varying (50)	address character varying (100)	phone character (10)	email character varying (50)	founded_year integer	description character varying (500)
1	IM Alumni Association	No. 123, ZhongZheng Rd., Daan Dist...	0912737646	imalumni@ntu.edu.tw	2020	IM Alumni Association is a non-profit organizati
2	CS Alumni Association	No. 456, Ren'ai Rd., Taichung City 4...	0922123456	csalumni@ntu.edu.tw	2019	CS Alumni Association is a non-profit organizati
3	Taipei Alumni Association	No. 789, Xinyi Rd., Xinyi District, Tai...	0933456789	taipeialumni@ntu.edu.tw	2018	Taipei Alumni Association is a non-profit organiz
4	Lawrencefort Alumni Association	84110 Kristina Alley	6568069255	anthonyking@example....	2006	Present people structure cold gun process work.
5	South Keithstad Alumni Association	Unit 7896 Box 5290	209.872.35	ronald16@example.net	1978	Eat other face though sing mind possible. Area a

amount integer	date date	donation_type character varying (50)	alumni_id character (9)	donation_id [PK] integer
58391	1985-07-27	Research	B98505005	109018
72	2024-06-11	Others	B95610055	109019
33429	1980-08-06	Scholarship	B91302126	109020
58786	1970-10-20	Research	B08704043	109021
97208	2021-10-08	Others	B97408127	109022



SQL 指令效能優化與索引建立分析

- 我們所建立的 Index:
 - 在 **DONATION** 中加入 Alumni_ID + Date 的索引。
 - 期望提升查詢效率，特別針對捐款校友與日期範圍查詢。
- 進行 Query 速度的實驗
 - 查詢在某個時間前
 - 捐款總額前十高的校友



索引對於 Query 的影響

- 未使用索引：
 - 平均查詢時間：約 0.1638 秒。
 - 標準差：約 0.0482 秒。
- 使用索引：
 - 平均查詢時間：約 0.0784 秒。
 - 標準差：約 0.0184 秒。
- 查詢時間降低 52.14%，且結果更穩定。

```
SELECT
    d.Alumni_ID,
    SUM(d.Amount) AS total_donations
FROM
    DONATION d
WHERE
    d.Date <= '2024-01-01'
GROUP BY
    d.Alumni_ID
ORDER BY
    total_donations DESC
LIMIT 10;
```



交易管理:範例

情境:將活動資訊插入資料庫的兩個相關表中: ASSOCIATION_EVENT 和 HELD_BY

1. 開始交易

- a. 將活動基本資料插入 ASSOCIATION_EVENT 表(活動名稱、日期等)。
- b. 使用 SQL INSERT 確保資料插入完整性。

2. 建立活動與校友會的關聯

- a. 將活動與與參與的校友會插入 HELD_BY 表。一個活動可能有多個校友會共同舉辦。
- b. 建立 ASSOCIATION_ID 關聯, 確保數據間關聯性完整。

3. 提交交易

- a. 所有步驟成功後, 執行 `conn.commit()` 永久保存變更。

4. 錯誤處理與回滾

- a. 若異常發生, 系統執行 `conn.rollback()` 回滾操作, 取消先前所有更改。



並行控制: 範例

當管理員新增校友會新成員時, 可能出現多名管理員同時進行新增操作, 導致資料不一致或重複新增。

- 利用 Python 套件進行並行控制:
 - 使用 `threading.Lock`:
 - 確保同一時間僅允許一個新增帳號的請求進行資料庫插入。
 - 程序中, 使用 `lock.acquire()` 獲取鎖並檢查校友帳號是否已存在。
 1. 是: 直接釋放鎖並回傳錯誤訊息。
 2. 否: 執行新增操作並提交變更。
 - 新增操作完成後, 釋放鎖(`lock.release()`), 允許其他請求進行。



期末專案整體心得 - A 同學

- **團隊合作經驗：**

- 第一次與多人合作開發系統，過程中感受到溝通的重要性。
- 體會到硬實力固然重要，但有效的溝通方式與技巧更關鍵，能確保正確的人負責正確的任務。

- **技術挑戰：**

- 雖然技術本身(查詢、輸入、刪除 function)不算複雜，但系統的龐大規模與資料庫的多層關聯性常造成思維混亂。
- 從挑戰中學到如何應對大量資料互相牽連的系統設計問題。



期末專案整體心得 - B 同學

- 團隊反思：
 - 首次進行前後端整合專案，雖然工作量與時間壓力大，但過程充滿成就感。
 - 組員表現令人欽佩，學到許多值得效仿的特質，如效率、程式概念，以及使用工具(如 ChatGPT)的技巧，激勵自己成為更好的程式開發者。