

# FunCrew:基於揪團約會的管理系統

*By Brain 與他的快樂夥伴*

111753135\_鄞雋衡 (組長)\_資碩工一

108303067\_杜淑萍\_會計四

107202009\_劉子御\_企管四

111753121\_林侑萱\_資碩工一

111753214\_思沛淇\_資碩計一

111753228\_陳輝\_資計計一

# 題目簡介：以揪團為主題的約會管理系統



## Project: FunCrew

一個提供大家揪團去吃喝玩樂的平台，用戶註冊帳戶後即可在平台上發起活動或是發出貼文，吸引其他用戶來參加活動或在貼文上面做進一步的討論，享受互相聊天、交友的快樂！

### 三大系統要素

#### 活動

- 登記用戶發起及參與的活動內容，並維護活動的狀態跟相關資訊
- 活動會在平台上頁面展示供用戶挑選參加

#### 會員

- 平台上的登記用戶，會有各自的貼文跟參加的活動，以及各自的基本資料
- 會員可以透過瀏覽網站找到有興趣的活動參加，或是回應貼文

#### 貼文

- 會員可以在平台上發布貼文來討論各式主題，也可以觀看別人的貼文給予回應

# 需求分析-用戶模組、活動模組、貼文模組

## 用戶

- 用戶包含不同個人和登入資訊，例如：電郵(帳號)、密碼、暱稱、頭貼.....
- 用戶可以發佈和參加多個活動，也可以在活動底下討論
- 用戶可以發佈多個貼文和在多個貼文下留言

## 活動

- 活動包含不同活動資訊，例如：時間、地點、類型、人數、描述.....
- 用戶可以在活動結束後對發起人評分
- 討論和留言需要記錄內容和時間
- 一個活動可以有多個討論留言

## 貼文

- 貼文要包含標題、內容和觀看數
- 為防止刷觀看數，已設定每個人每次登入只能刷每個貼文的觀看數一次
- 一個貼文可以有多個留言

# 系統功能

## 用戶

- 註冊帳戶(C)
- 登入帳戶(R)
- 查看用戶資訊(R)
- 檢視發布過的貼文(R)
- 檢視用戶曾經發布及參加的活動(R)
- 修改個人資訊(U)

## 活動

- 活動篩選器:類型、報名中、準備進行、已完結(R)
- 發起活動(C)
- 修改活動(U)
- 刪除活動(D)
- 參加活動(C)
- 對活動發起人評分(C)(U)
- 查找並顯示活動的相關資訊(R)
- 新增評論(C)
- 修改評論(U)
- 刪除活動(D)
- 查看評論(R)

## 貼文

- 貼文篩選器:最新、最多觀看、討論熱烈(R)
- 以貼文標題發布者進行搜尋(R)
- 新增貼文(C)
- 刪除貼文(D)
- 更新貼文(U)
- 新增留言 (C)
- 檢視留言 (R)
- 更新留言 (U)
- 刪除留言 (D)

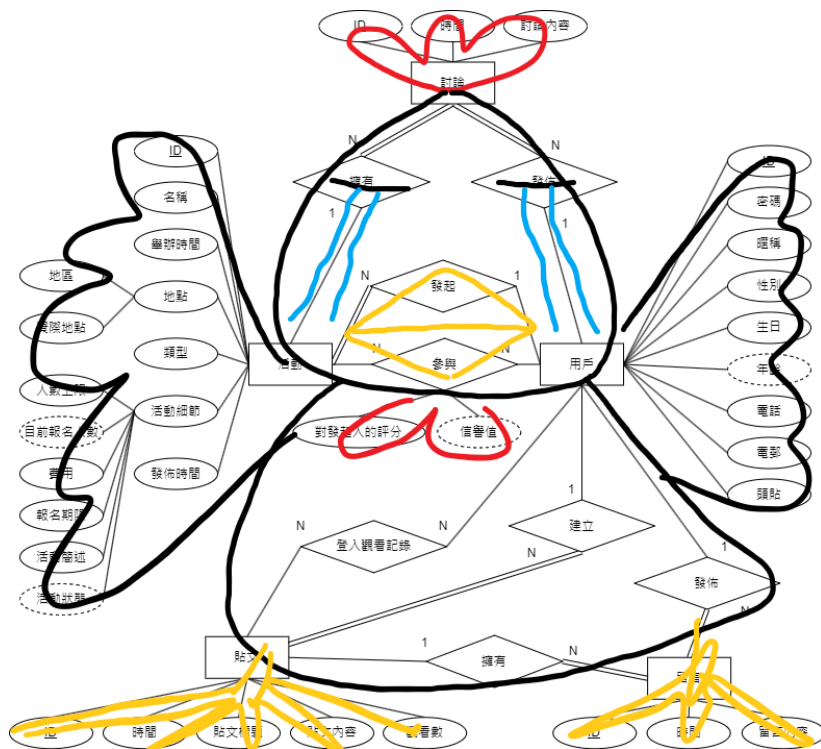
# ER Model

## ● 五個 Entities

- 活動:用戶舉辦的聚會活動事件
- 討論:活動頁面的討論紀錄
- 用戶:使用者的資訊
- 貼文:用戶發布的討論
- 留言:貼文下方的留言紀錄

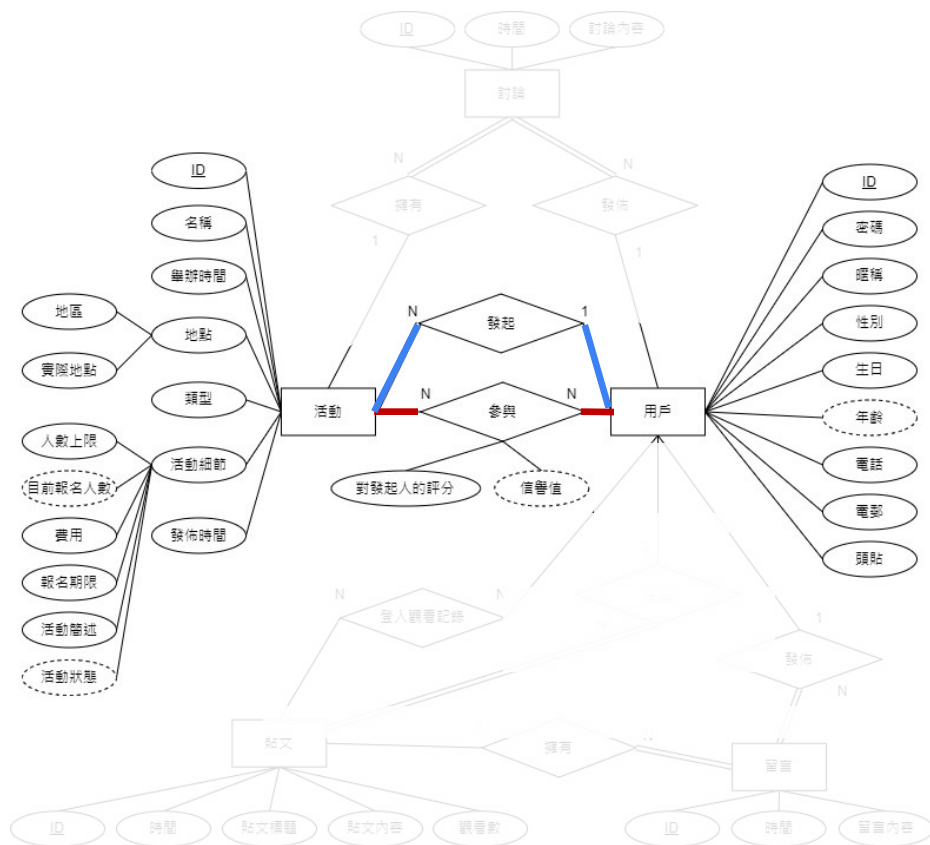
## ● 八個 Relationship

- 活動擁有討論
- 用戶發布討論
- 用戶發起活動
- 用戶參與活動
- 用戶建立貼文
- Etc



# ER Model

1. 用戶發起活動 (1-N)
2. 用戶參與活動 (N-N)
3. 用戶參與後可以對發起人評分

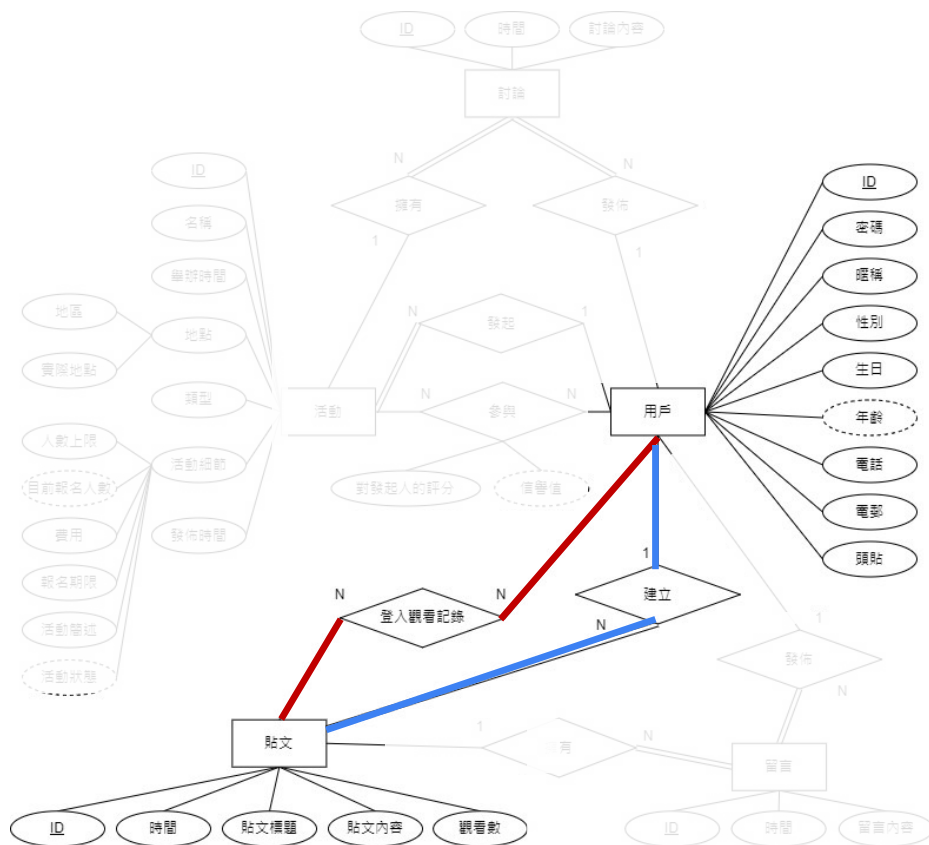


1. 用戶發佈討論 (1-N)
2. 活動擁有討論 (1-N)



# ER Model

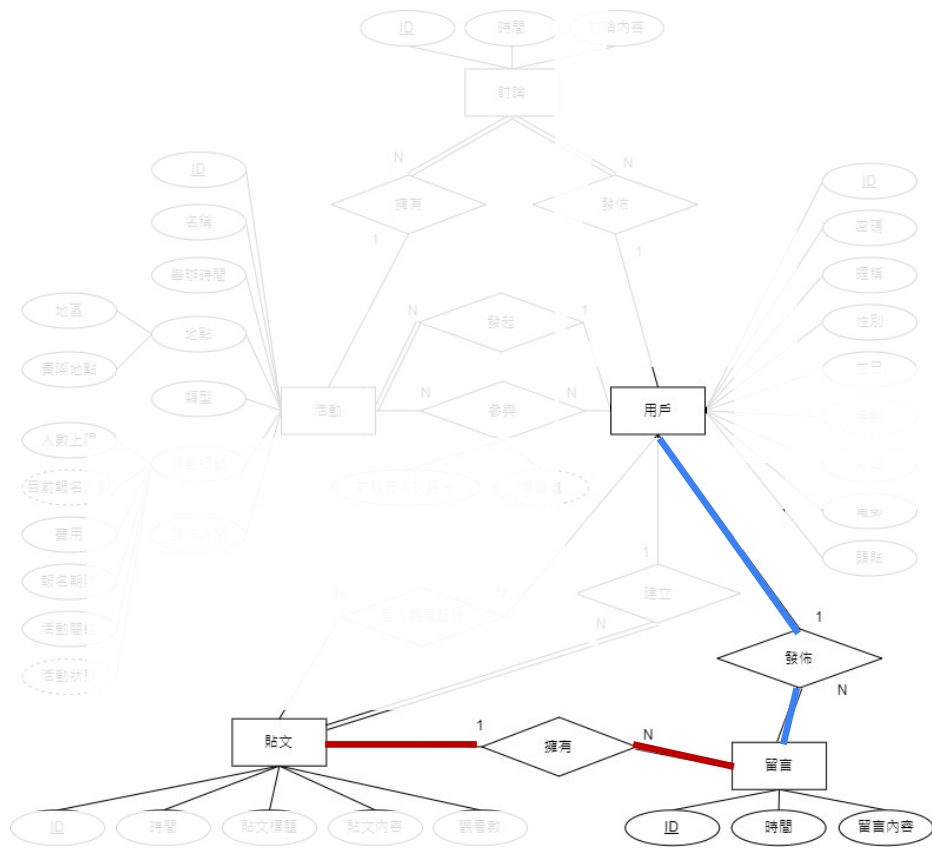
1. 用戶建立貼文 (1-N)
2. 用戶會記錄登入觀看的貼文(N-N)





# ER Model

1. 用戶發佈留言 (1-N)
2. 貼文擁有留言 (1-N)



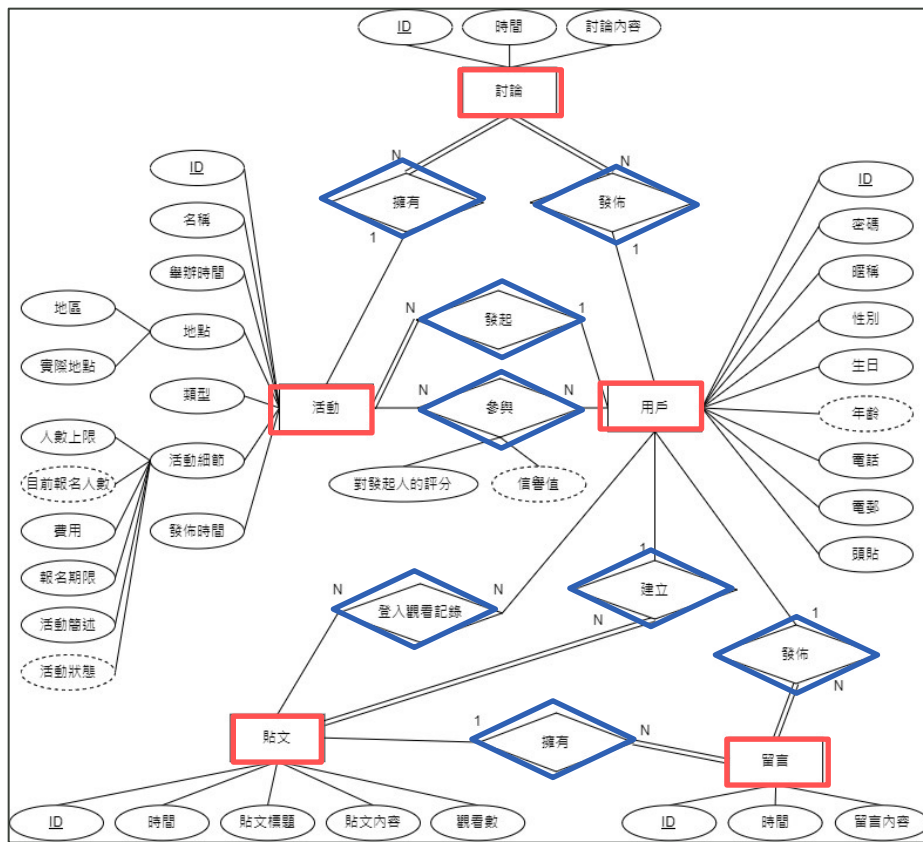
# ER Model

## ● 五個 Entities

- 活動:用戶舉辦的聚會活動事件
- 討論:活動頁面的討論紀錄
- 用戶:使用者的資訊
- 貼文:用戶發布的討論
- 留言:貼文下方的留言紀錄

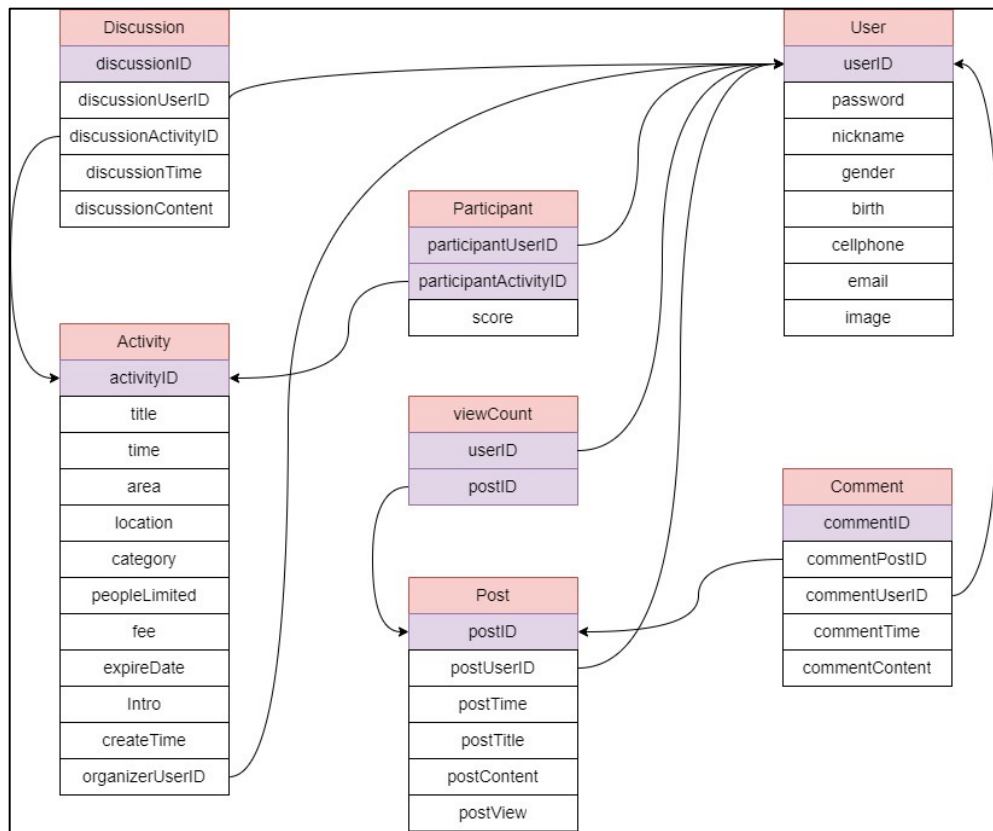
## ● 八個 Relationship

- 活動擁有討論
- 用戶發布討論
- 用戶發起活動
- 用戶參與活動
- 用戶建立貼文
- 貼文擁有留言
- Etc



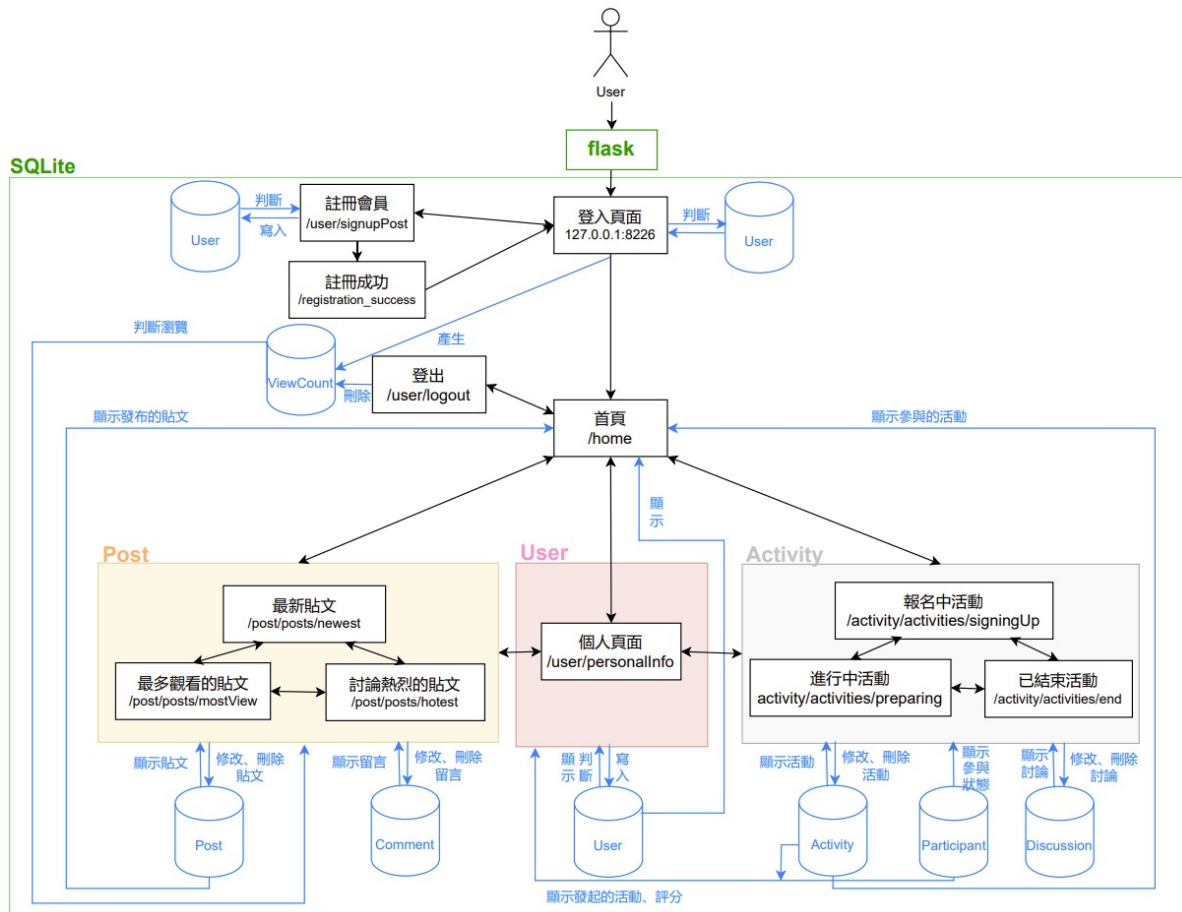
# Relational Schema

- 7 個 Table
- 紅色是table Name
- 紫色是primary key
- 箭頭是foreign key



# 系統架構

- Flask + SQLite
- Frontend
  - HTML CSS JavaScript
- Backend
  - Python
- DBMS
  - SQLite



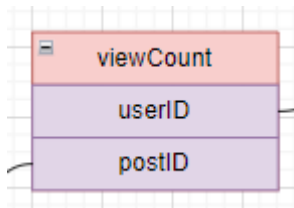
# 任務分工

工人	分工	貢獻百分比
鄧雋衡	系統初期架構建立、資料庫建立	12
杜淑萍	UI設計、部分前端畫面、後端功能 ( 活動模組、討論模組 )	18
劉子御	UI設計、初期系統架構、系統需求文件、簡報與報告	12
林侑萱	系統架構圖整合、後端功能(使用者相關模組:修改、頭像、連結活動、貼文)	18
思沛淇	User相關模組、系統整合(連結處理)	20
陳輝	ER Model和Schema整合、後端功能 (貼文模組、留言模組)、輔助系統整合	20

## 心得、收穫與建議

- 在設計資料庫時需要想得更長期，且規模要盡量做大，方便後期調整 (Partition, Distributed Database)
- 若完成後還需要修改資料庫真的曠日廢時，會嚴重影響 Table, Integrity constrain等
- 前台及後台在完成任務時可以雙重保障或是協調
- 完成專案中負責整合的人會需要花很多心力，更加了解實務上的開發運作

## 心得、收穫與建議\_Partition



viewCount
userID
postID

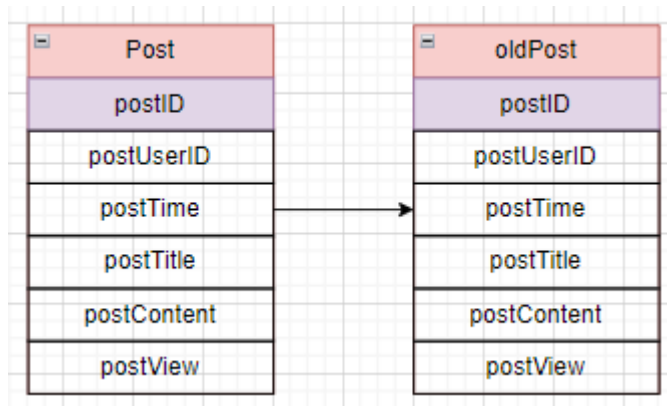


Table: temp5ViewCount

	postID
	過濾
1	2
2	5
3	62

- Table名稱會是變數
- Attribute會不同 (存postID就可以，userID成為Table名稱的一部份)
- \*甚至連foreign key也不用設 (以登入來計算，登出就會drop掉)

# 心得、收穫與建議\_Partition + Index



前設:很很很常有新貼文新增

所以不適合建Index，但用完Partition就可能適合建Index

## Pros

- 舊貼文一段時間才更新，可以建Index加速搜尋
- 貼文t資料量可以減少，查詢 / 排序 可以更快

## Cons

- 有機會浪費空間，過時貼文跟本就不常被用戶搜尋(但內部人員做資料分析會常用到過往資料，所以使用條件很重要)