
比賽數據管理系統

— Demo Date: 6/27 —

組員名單

組名: 國際籃球戰情室

- 資碩一 113753117 陳郁晴(隊長)
- 資訊三 110703011 李逸盛
- 資訊三 110703014 涂世豪
- 資訊三 110703040 陳怡然
- 資碩一 112753106 張曉雲
- 資碩一 112753108 黃翊瑄
- 資碩一 112753138 周宸宇
- [GitHub](#)

Outline

- 需求分析
- 系統分析
- ER Model
- Relational Schema
- 系統架構
- 心得、收穫與建議
- 分工

需求分析

此為一個籃球數據管理系統，用戶角色有：

- 1) 一般使用者(球迷): 追蹤每場比賽及球員的數據，為喜歡的球員和球隊喝采
 - 2) 管理者: 管理及呈現球員、球隊相關資訊，以及各場比賽結果
- **Player(球員):** PlayerID、Name[FName、LName]、BDate、Fitness[height、weight]、Position、Country
 - **Team(隊伍):** TeamID、Nickname、Abbreviation、Coach、YearFounded、City
 - **Game(比賽):** GameID、Date、Place
 - **GameRecord(球員單場比賽表現):** RID(PID+GID)、Assist(助攻次數)、FreeThrow(罰球命中次數)、Hit(投球命中次數)、Steal(抄截次數)、Score、Rebound(防守籃板次數)
 - **Season(賽季):** SeasonID、Year
 - 資料庫來源: <https://www.kaggle.com/datasets/wyattowalsh/basketball> (保留部分內容)

需求分析

1. 一支球隊有多名球員

- 每支球隊來自一個城市並由一位教練帶領
- 每支球隊擁有唯一的球隊編號及球隊名稱
- 記錄球隊成立的年份

2. 記錄球員的個人資訊

- 每位球員都有姓名及唯一的編號
- 記錄球員的國籍、在隊中的位置、生日、身高和體重

需求分析

3. 一場比賽由兩支球隊參與

- 每場比賽有唯一的編號，並記錄比賽日期、地點
- 我們記錄哪支球隊為主隊及客隊、各自得分、獲勝球隊

4. 一年會有多個賽季舉行比賽，因此也記錄比賽的賽季D和年份

5. 每位球員的單場比賽表現

- 記錄球員參與的所有比賽數據，包括助攻次數、投球命中次數、抄截次數、防守籃板次數、罰球命中次數和個人得分

系統分析

1. 球隊：

- 管理者新增球隊資訊，包含球隊編號、球隊名稱、教練姓名、成立年份、所在城市
- 若教練姓名異動，則管理者更新教練姓名
- 管理者及球迷皆能查看球隊資訊

系統分析

2. 球員：

- 管理者新增球員資訊，包含球員編號、球員姓名、生日、國籍、身高及體重
- 若球員的姓名、生日、國籍、身高、體重、隊中位置異動，
則管理者更新球員的姓名、生日、國籍、身高、體重、隊中位置
- 若球員退役，管理者刪除此球員的資訊
- 管理者及球迷查看球員個人資訊

系統分析

3. 比賽數據

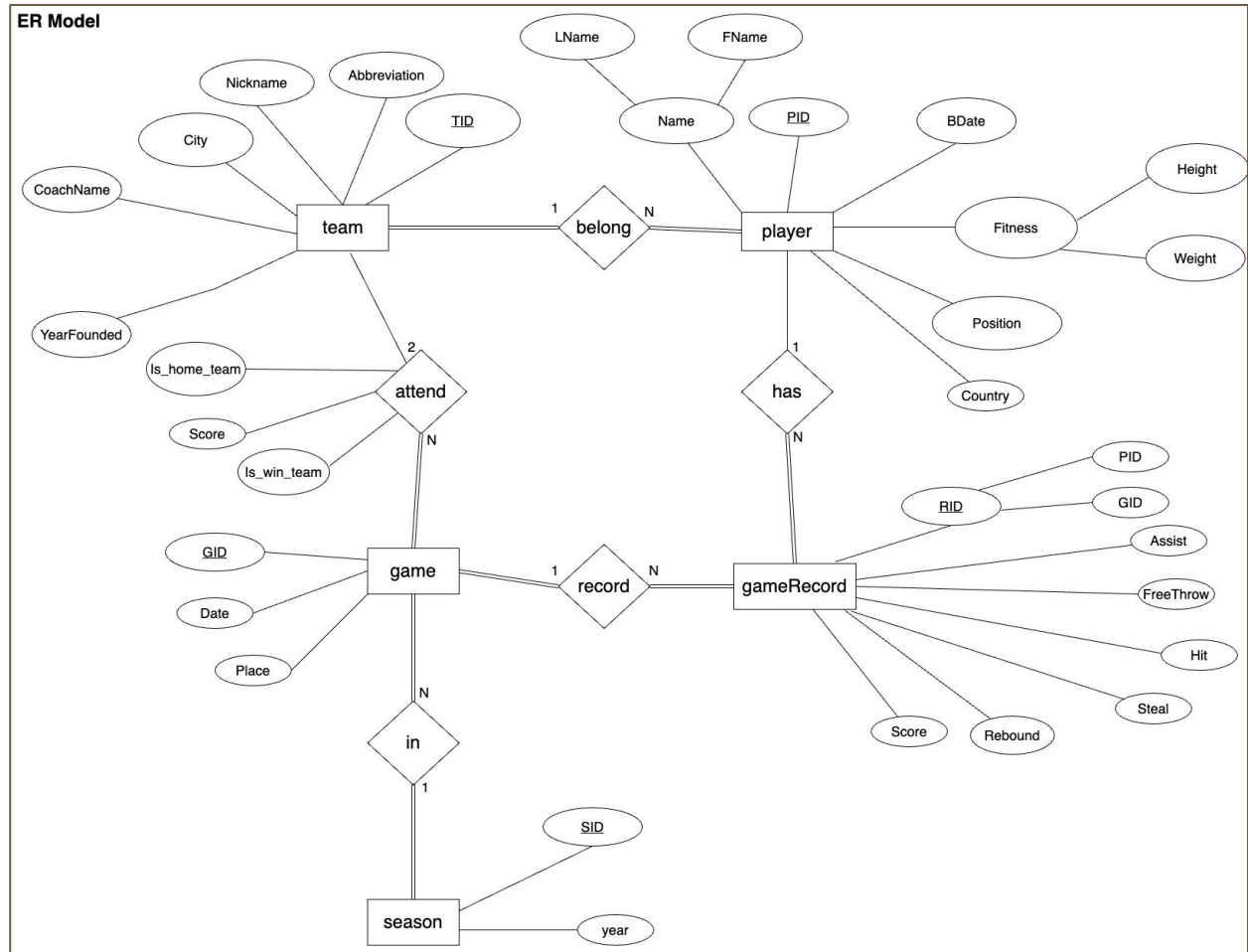
- 管理者新增比賽數據，包含比賽編號、時間、地點、參與球隊的編號及主客隊資訊、各隊得分數、獲勝球隊
- 若各隊得分數、獲勝球隊異動，則管理者更新各隊得分數、獲勝球隊
- 管理者及球迷查看比賽數據

系統分析

4. 球員單場比賽表現

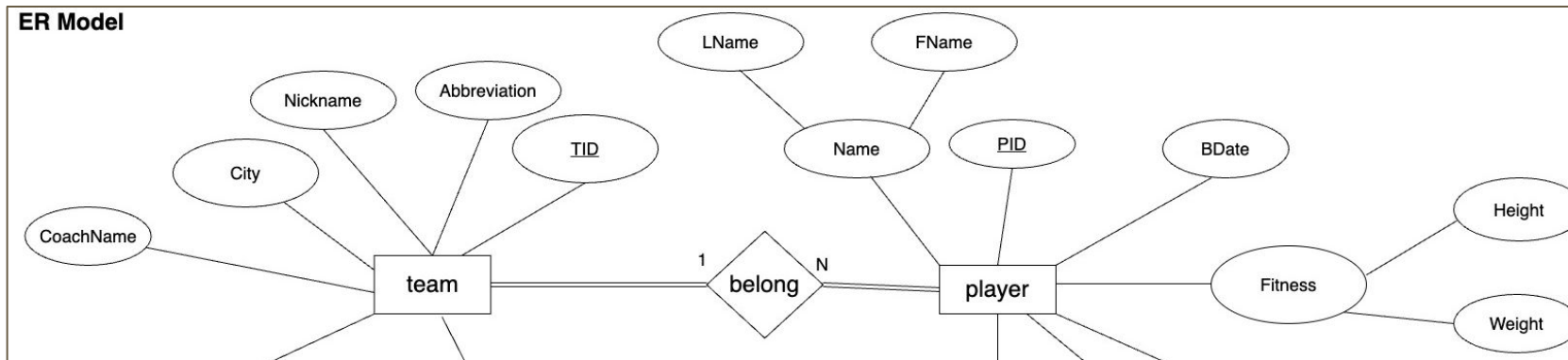
- 球員每打一場比賽，管理者新增個人單場比賽數據，包含助攻次數、投球命中次數、抄截次數、防守籃板次數、罰球命中次數和個人得分
- 若球員個人表現數據有異動，則管理者更新個人表現數據
- 管理者刪除此球員所有的單場比賽表現紀錄
- 管理者及球迷查看球員單場比賽表現數據

ER Model



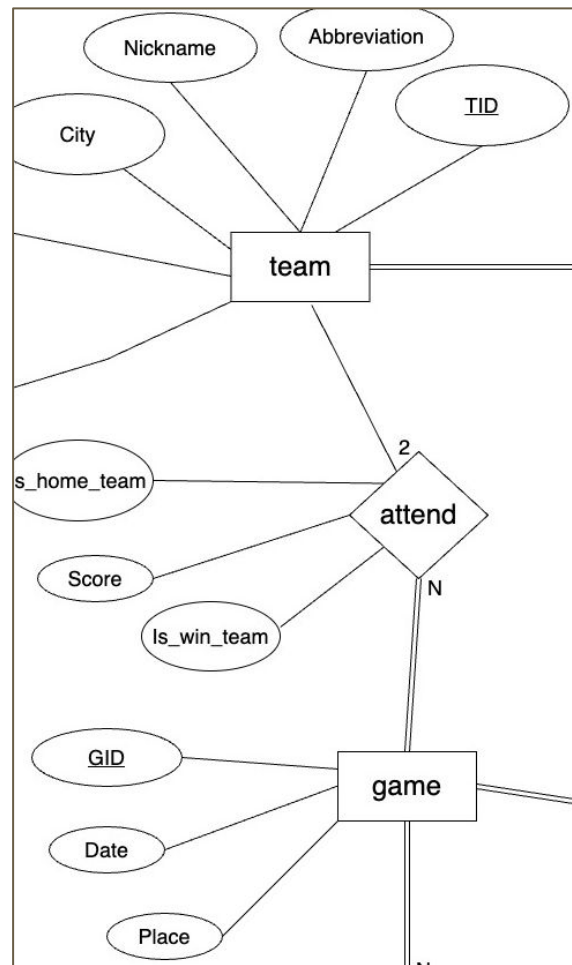
ER Model

- 每支球隊一定由多個球員組成，不允許沒有球員的狀況
- 每位球員一定只會隸屬於一支球隊
- 因此球隊與球員的關係為一對多，兩者在belong關係中皆為total participation



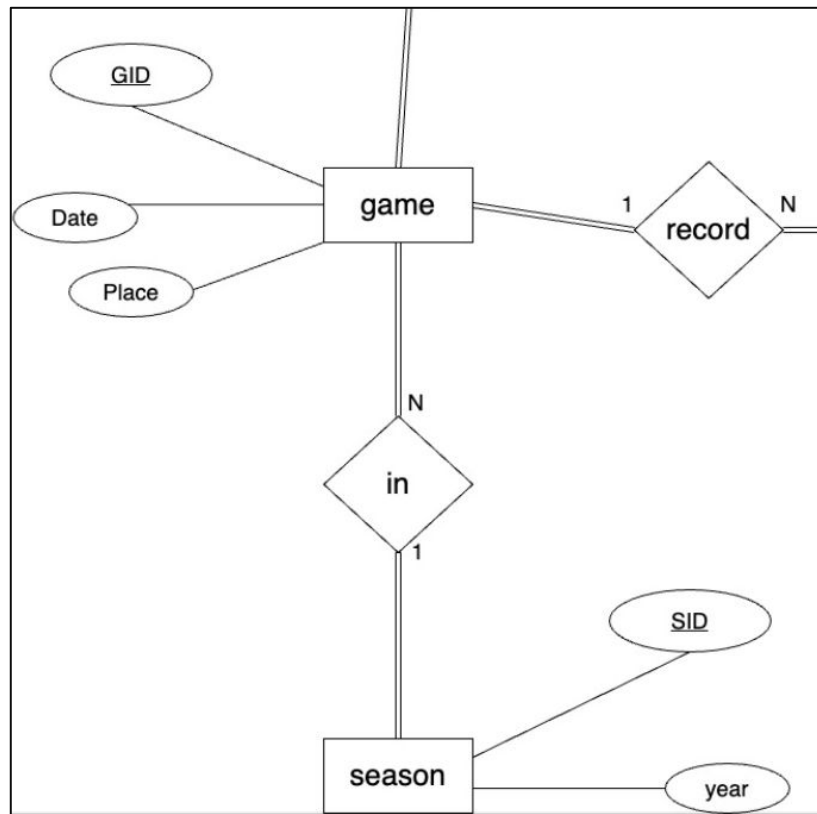
ER Model

- 每支球隊(team)會參與多場比賽
- 每場比賽(game)一定由兩支球隊參與
- 因此球隊與比賽的關係為二對多，比賽在 attend 關係中為 total participation
- Attend 關係中另外紀錄球隊是否為主場球隊、各球隊得分數、是否為獲勝球隊



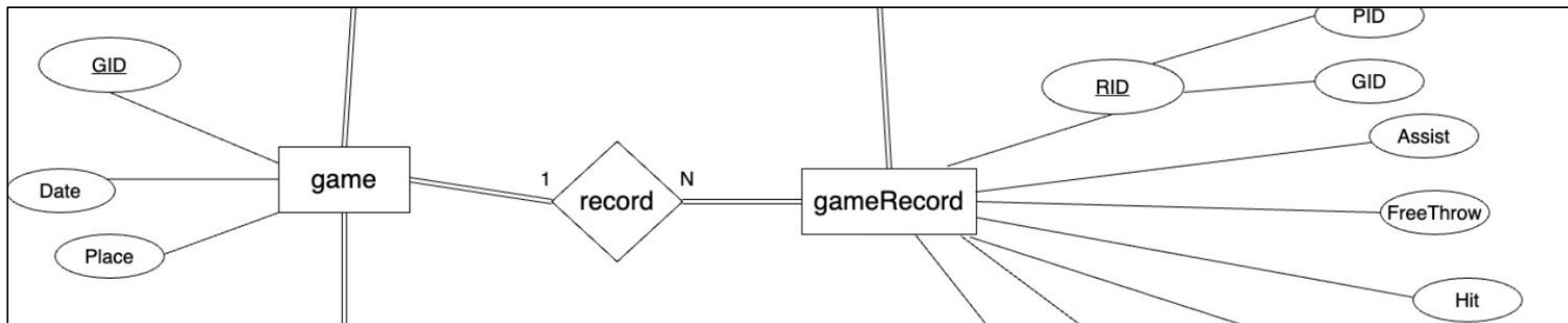
ER Model

- 每個賽季(season)一定會有 multiple 場比賽
- 每場比賽(game)一定只會屬於一個賽季
- 因此賽季與比賽的關係為一對多，兩者在 in 的關係中皆為 total participation



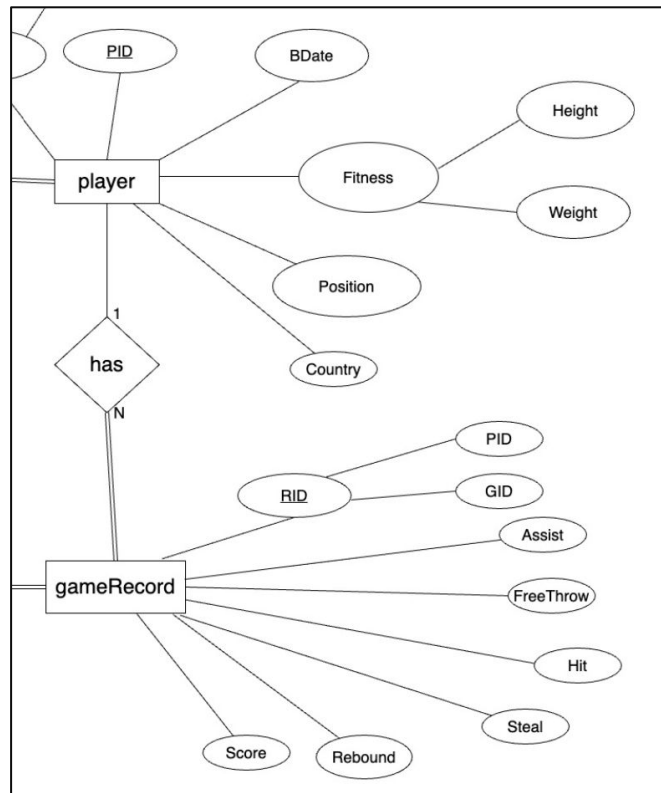
ER Model

- 每場比賽(game)一定會有多筆球員單場比賽表現(gameRecord)紀錄
- 每筆球員單場比賽表現(gameRecord)紀錄一定只對應到一場比賽
- 因此比賽與球員單場比賽表現的關係為一對多，兩者在ecord關係中皆為total participation



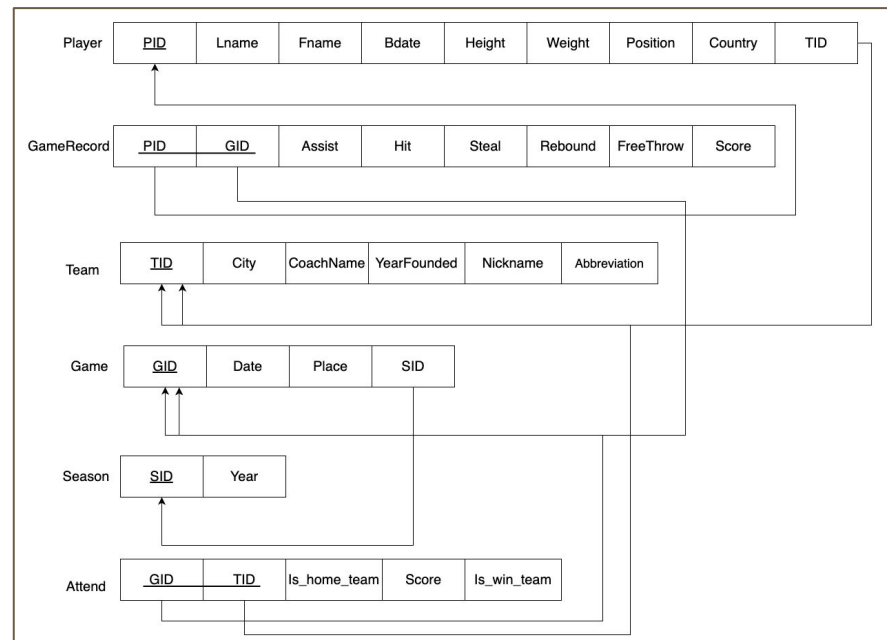
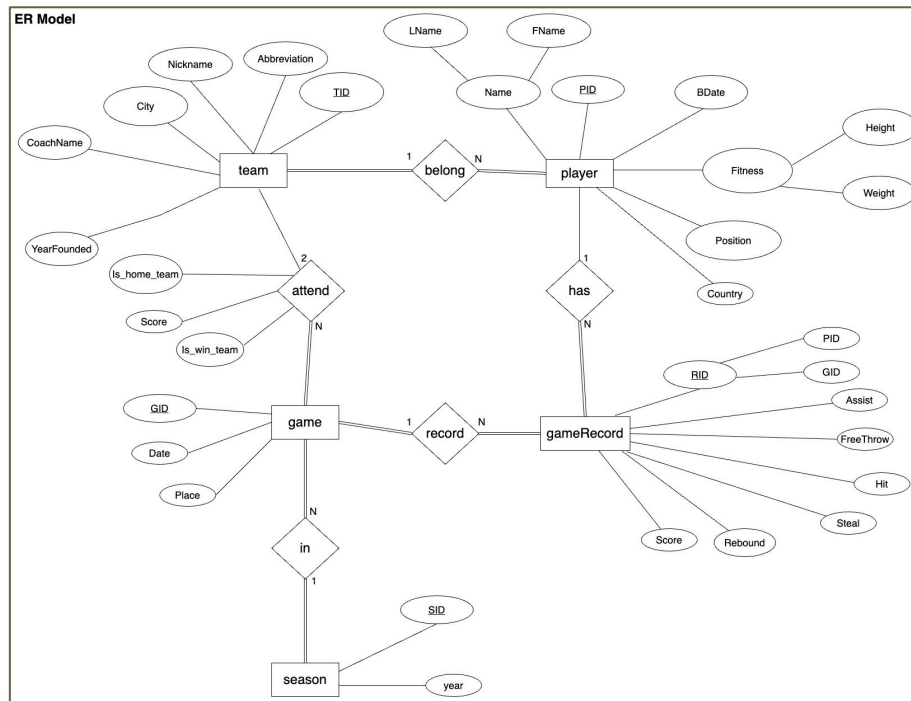
ER Model

- 每位球員(player)會有多筆球員單場比賽表現紀錄
- 每筆球員單場比賽表現(gameRecord)紀錄一定只會對應到一位球員
- 因此球員與球員單場比賽表現的關係為一對多，球員單場比賽表現在has的關係中為total participation



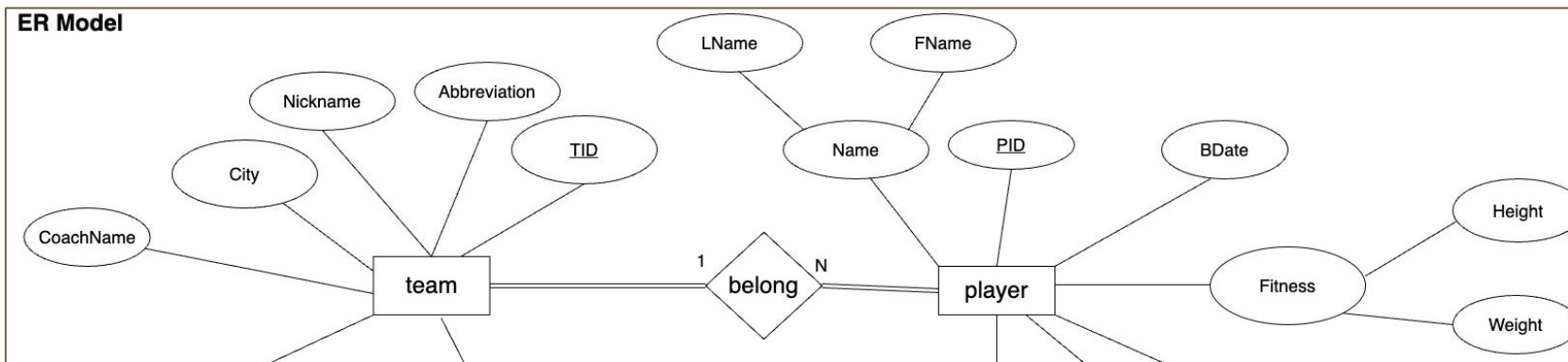
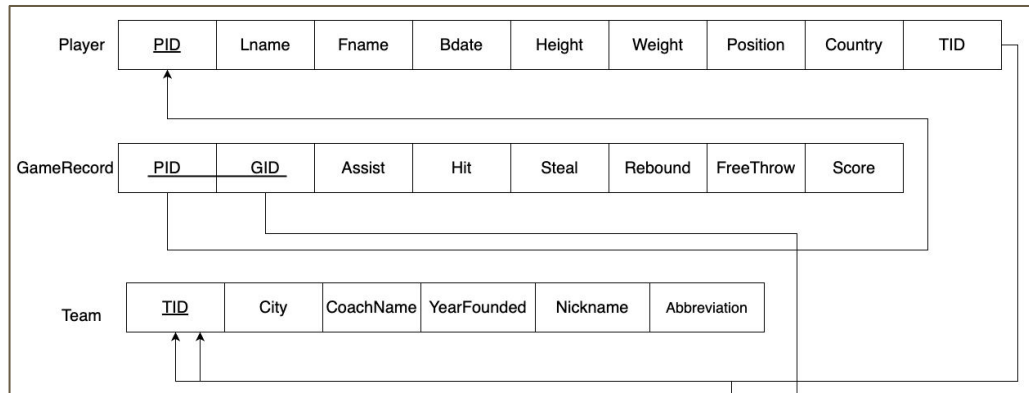
Relational Data Model

- 將ER Model轉為Relational Data Model共有6張table



Relational Data Model

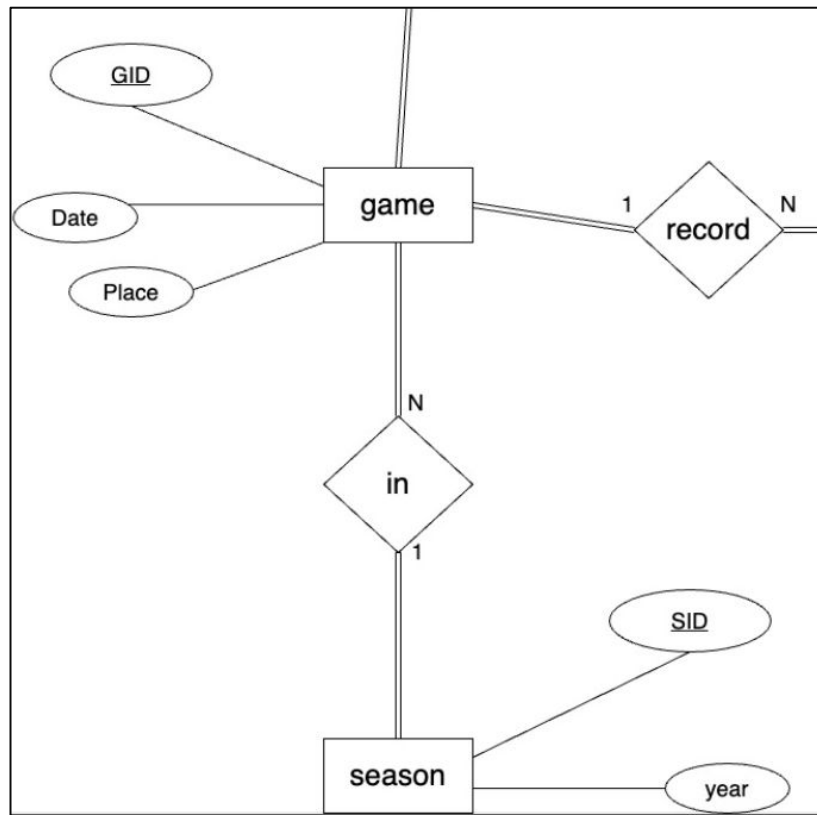
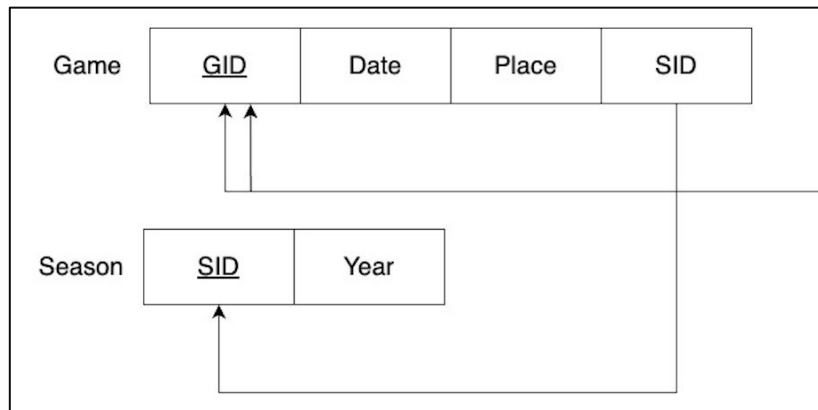
- Player: PID為primary key
- Team: TID為primary key
- Team與Player的關係為一對多, 因此將Team的primary key (TID) 加入至Player table



Relational Data Model

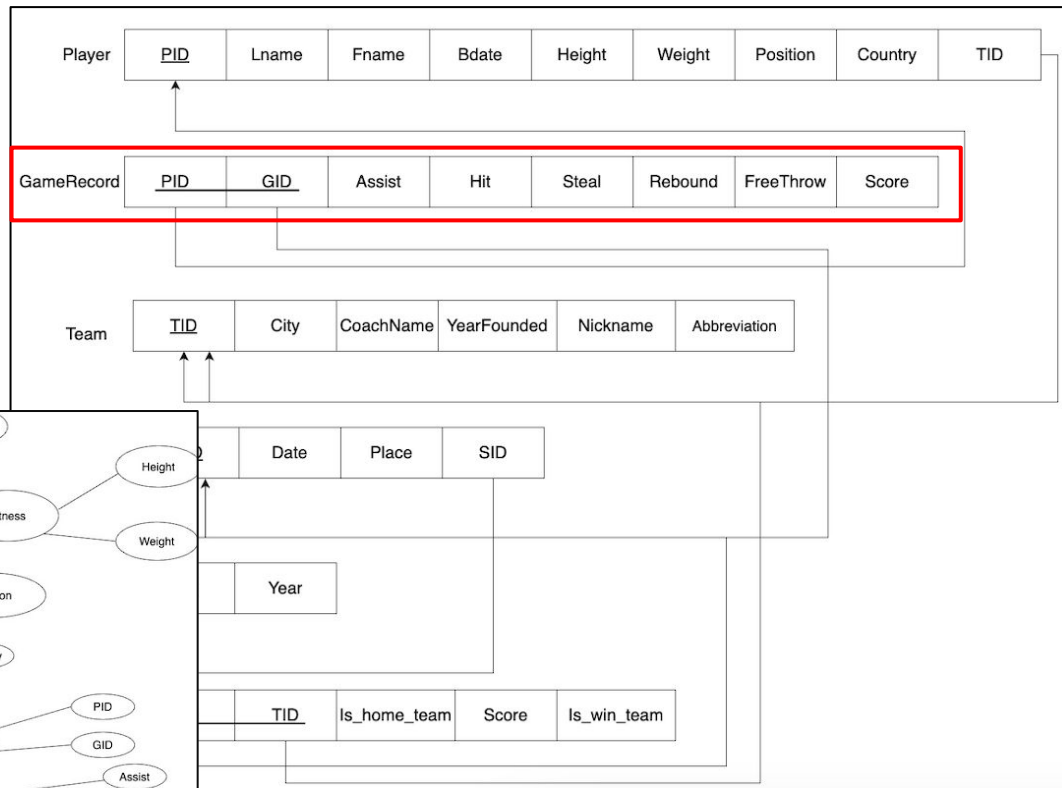
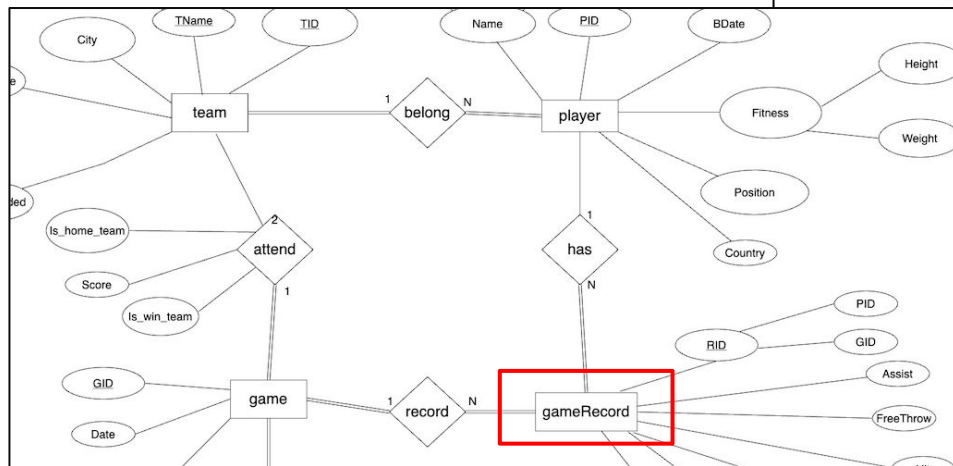
- Game: GID為primary key
- Season: SID為primary key
- Season與Game的關係為一對多,

因此將Season的primary key (SID)加入至 Game table



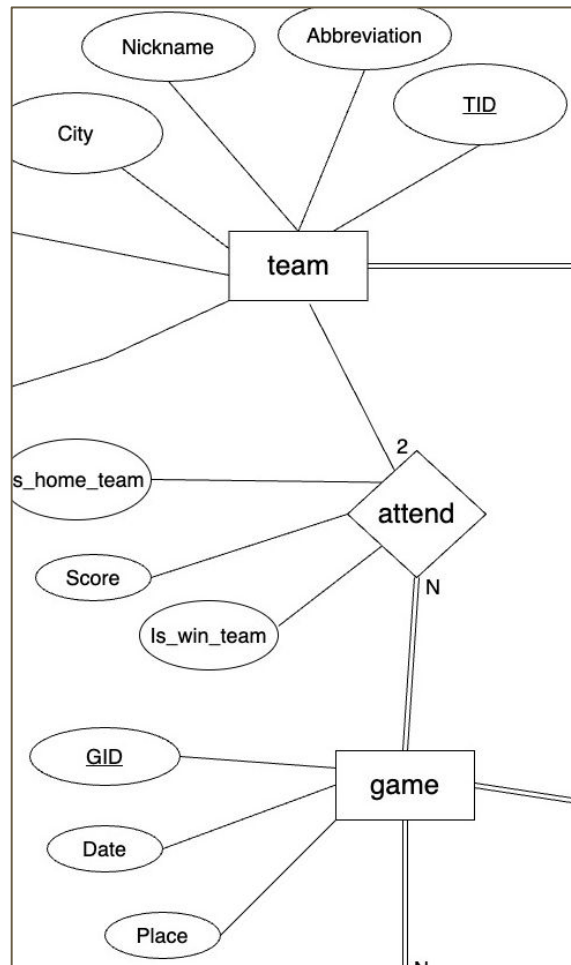
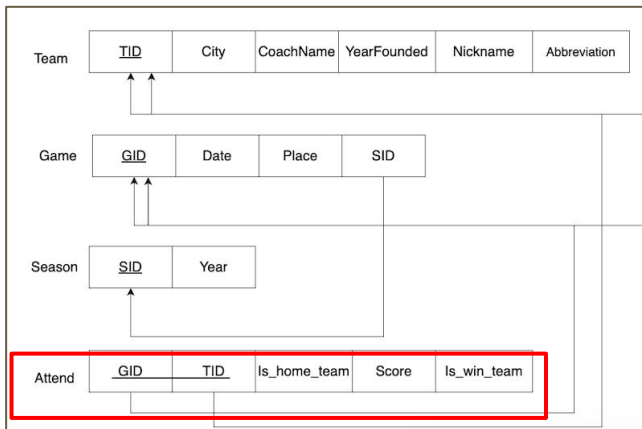
Relational Data Model

- GameRecord: 球員單場比賽表現
可由PID+GID決定, 因此為
primary key



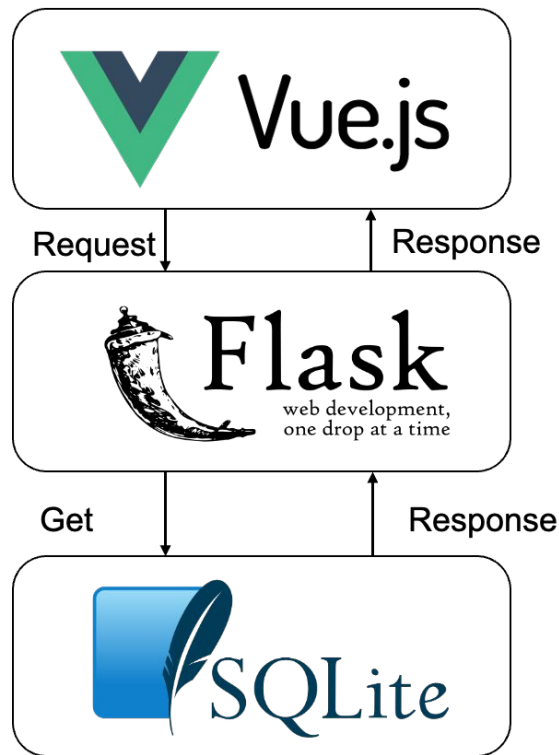
Relational Data Model

- Attend: 球隊參與比賽紀錄可由GID+TID決定, 因此為primary key
- 球隊是否為主場隊伍以及獲勝隊伍使用boolean值記錄
- 同時記錄球隊的得分數



系統架構

- 前端: Vue.js 3 + Sass + TypeScript
 - Libraries: Pinia
 - Tools: Vite
- 後端: Flask (Python)
- 資料庫: SQLite



心得與收穫

- 設計：
 - 使用Figma將所預想的畫面實際畫出
 - 必須考慮schema與constraint設計網頁
- 資料庫：
 - 因為我們用的是引用別人的資料庫但我們只要其中的部分內容，所以需要考慮filter，也必須確保data constraints
 - 加深資料庫操作理解

心得與收穫

- 前端：
 - 以 Vue 實踐 RWA 並理解現代 Web Dev 觀念
 - 以 OpenAPI Swagger tools 建立符合需求設計的 RESTful API
 - 理解需求，並積極參與、問問題很重要，需要合理評估時間的能力
- 後端：
 - 熟悉 Flask 框架的基本結構，學習到 Flask 路由、模板渲染等功能
 - 學會怎麼看 API 規格文件及使用 Swagger 測試，針對前端的 request 需求修改 response
 - 了解資料庫 schema 設計對後端邏輯實踐的高度關聯性及重要性

分工

系級	學號	姓名	工作內容	貢獻百分比
資碩一	113753117	陳郁晴 (隊長)	後端、PPT	14 %
資訊三	110703011	李逸盛	DB Schema 設計	8 %
資訊三	110703014	涂世豪	資料庫、後端	14 %
資訊三	110703040	陳怡然	前端、API 設計	22 %
資碩一	112753106	張曉雲	後端	14 %
資碩一	112753108	黃翊瑄	UI 繪製、SQL 設計	14 %
資碩一	112753138	周宸宇	後端	14 %

Demo Time

Thanks for Listening!