

資料庫管理 (113-1)

期末專案計劃書

B11705028 胡翔瑞、B11705046 李昀熹、B11902168 李奕辰

1 系統分析

你是否擁有專業的 NBA 知識，卻不知道如何轉換成錢？那就快來上「Oin」下注運彩吧！

「Oin」是一款線上的 NBA 運彩下注平台，主要目的為讓用戶可以下注自己喜愛或相信的球隊，「Oin」會載入該 NBA 賽季的所有比賽，並提供各比賽所有下注的玩法，以及兩隊相對應的賠率。每位用戶可以有好幾筆下注紀錄，在比賽開始前，用戶都可以隨意更改下注金額或取消下注，而一場比賽也可以有數個使用者下注。

根據不同的功能及掌控權限，「Oin」系統的用戶可以分成兩種身分，分別是 User 及 Admin。User 代表一般的使用者，也就是在網站上下注的人，可以在將錢儲值進帳號後，用帳號的錢對即將發生的任何比賽場次做下注，而在該場比賽結束過後，使用者會收到該場比賽的結果，並可以在帳號查看這次的下注贏了或輸了多少錢。而 Admin 代表「Oin」的業務經營者，主要負責更新每場比賽各種玩法的賠率以及結果，並在比賽完成後更新對該場比賽有下注的所有使用者帳號的金錢流動，同時也可以查看使用者的每項下注紀錄，包括該使用者選擇什麼玩法、對哪支球隊下了多少錢。

1.1 系統功能

1.1.1 關於下注網站的相關設定

網站會提供現今 NBA 賽季的所有比賽，使用者可以對該賽季尚未舉行的比賽進行下注，每次下注的金額完全由使用者決定。下注後系統會從使用者帳號扣取金額，若是下注成功，系統會將使用者下注的錢乘上該玩法的賠率匯進用戶帳號；下注失敗的話則不會歸還。例如某使用者對明天勇士 (客場) 對決湖人 (主場) 的比賽下注 1000 元在**勇士獲勝**上，選擇的玩法為**不讓分**，勇士獲勝的賠率為 2.35，湖人獲勝的賠率為 1.7。系統會在隔天比賽開始的當下，從用戶的帳號扣取 1000 元，假設最後勇士獲勝，代表下注成功，系統會匯 2350 元到使用者的帳號裡頭；若是湖人獲勝，代表下注失敗，使用者會失去這 1000 元。

1.1.2 給 User 的功能

在本系統中，User 可以執行以下功能：

1. 儲值/提領現金到帳號裡：使用者可以自行選擇要儲值任意金額到帳號，也可以提領任意小於帳號金額的現金到自己的戶頭裡。
2. 下注比賽玩法：使用者能夠對尚未發生的比賽的任何玩法進行下注，但總下注的金額不能超過帳號的儲值金，例如某使用者帳號有 5000 元，那他所有進行中的下注金額總和不能超過 5000 元。
3. 更改/取消下注，使用者若是想更改某次下注的資訊，可以在比賽開始前任意修改，無論是修改金額或是取消下注都可以。
4. 查看隊伍資訊：使用者可以查詢任何隊伍現在的戰績、數據，例如幾勝幾敗、三分球命中率、近幾場戰績等等，以及某兩個隊伍的交戰數據。好讓使用者可以掌控所有隊伍的情況，做出對自己最有利的下注。
5. 查詢過去擁有類似賠率組合的玩法結果：假設有個玩法的賠率分別為 2.5 和 1.7，使用者可以查看過去所有類似此賠率組合的玩法結果，來協助下注判斷。

1.1.3 給 Admin 的功能

在本系統中，Admin 可以執行以下功能：

1. 增加/更改玩法及賠率：業務經營者可以修改玩法的內容，並決定每個玩法相對應的賠率。
2. 查看使用者資訊：業務經營者能夠查看使用者的資訊，包掛該使用者的下注紀錄、現金流動紀錄。
3. 查看/更改未來比賽：業務經營者可以查看將來未發生的比賽，若是賽季鍾某場比賽被取消或是有什麼資訊被更改，業務經營者也會將網站上德資訊一起更改。

2 系統設計

2.1 ER Diagram

圖 1 是「Oin(歐印)」的 ER-Diagram，在這個 ERD 中共有八個實體 (entity)，分別是 GAMBLER、CASHFLOW_RECORD、BET_ODDS_RECORD、BET_TYPE、GAME、STADIUM、

SEASON、TEAM，和一個弱實體（weak entity）GAMBLER_BETS，以及十一個關係（relationship），包括 MAKE、BET、IS FROM、SET ON、IN、PLAY ON、HAS、OF 等等。

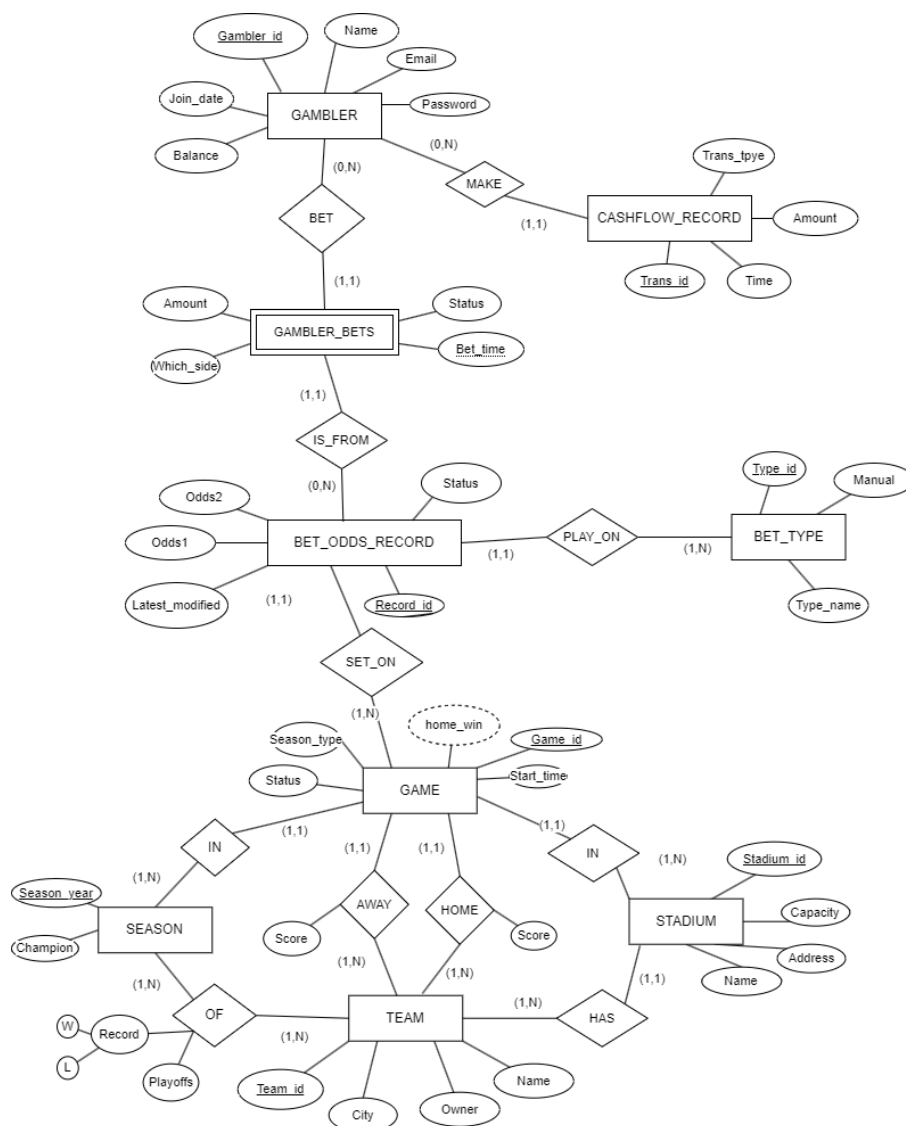


圖 1: 「Oin」的 ER Diagram

GAMBLER 代表的是使用 Oin 的一般使用者 (Gambler)，他們都須註冊才能開始使用。在註冊時，系統會要求使用者提供名稱、信箱、密碼，經註冊後便會產生一個專屬於該位使用者的 ID 及定義他在 Oin 中的身份為 Gambler。經過後臺手動設定，可以將特定使用者 (Oin 管理員) 也設定為 Admin，此時這個使用者就會同時有 Gambler 和 Admin 兩個身份。

一般使用者可以檢視查詢 BET_ODDS_RECORD 所代表的投注賠率項目紀錄，每一場的賠率組合有唯一的 Record_id，並記錄比賽雙方的賠率 (Odds1 和 Odds2)，而每一場比賽的詳細資訊會由 GAME 提供，但每場比賽可能有數種玩法，由 BET TYPE 提供玩法

種類，所以每一個紀錄即是一場比賽一種玩法的賠率紀錄。基本上此資料表僅管理員有權限對資料進行 CRUD，並且管理員會不斷地更新比賽的賠率，Latest_modified 可幫助管理員紀錄最近一次的修改時間，並以 Status 檢視此項目為「Expired (過期)」、「Processing (正在投注)」或「Not yet started (尚未開始)」。

同樣的，CASHFLOW_RECORD 管理所有 GAMBLER 在 Oin 的所有金錢流動，每一種都被可被分類在 Trans_type 中，包含「Earn」、「Loss」、「Deposit」和「Withdraw」，管理員可以查詢所有人的紀錄，而一般使用者僅能檢視自己的紀錄。

一般使用者在做投注的時候，會將資料記錄在 GAMBLER_BETS，連結 GAMBLER、BET_ODDS_RECORD 及自身的 Bet_time 作為某一投注紀錄的唯一識別，同時也紀錄 Which_side (投注方) 和 Amount (投注金額)，基本上是可以實現對同個投注項目擁有多次投注紀錄，亦可撤銷投注，而這都由 Status 所紀錄，但是否能夠撤銷投注，則由 BET_ODDS_RECORD 的 STATUS 控管 (如 Expired 則無法撤銷投注)。

至於 STADIUM、SEASON、TEAM 的用途則很簡單，就是作為 GAME 的額外資料，實體之間的關聯設計也很直觀，如：每個比賽必定有一對主客場隊伍、一個體育場，並屬於一個賽季，剩餘的應該皆可從圖中判讀。新增成獨立的實體也是為了增加名稱改變或新增實例 (Instance) 時的彈性 (Flexibility)，基本上使用者僅會查詢比賽的比賽內容作為投注依據，考慮 STADIUM、SEASON 和 TEAM 裡的資訊算是比較冷門的偏方，但我們依舊提供這樣子的資料給使用者們。

2.2 Relational Database Schema Diagram

我們可以將圖 1 的 ER diagram 轉換成圖 2 的 Database Schema，一共由十個關聯 (relation) 組成，分別是 GAMBLER、CASHFLOW_RECORD、GAMBLER_BETS、BET_ODD_RECORD、BET_TYPE、GAME、SEASON、TEAM、TEAM_STANDING、STADIUM。

GAMBLER 這個關聯的主鍵 (Primary Key, PK) 是 Gambler_id。而 CASHFLOW_RECORD 這個關聯則是由 GAMBLER 的資金流動記錄延伸而來，因為其為多值屬性，故需獨立成為一個關聯，其主鍵為 Trans_id，並包含 Gambler_id 作為外部鍵 (Foreign Key, FK)，參考到 GAMBLER 的主鍵 Gambler_id。

GAMBLER_BETS 這個關聯的主鍵為 Bet_time、Gambler_id 和 Record_id，並包含兩個外部鍵：Gambler_id 參考到 GAMBLER 的主鍵，Record_id 則參考到 BETS_ODD_RECORD 的主鍵。

BET_TYPE 這個關聯的主鍵是 Type_id。BETS_ODD_RECORD 這個關聯的主鍵是 Record_id，並包含外部鍵 Game_id 參考到 GAME 的主鍵，Type_id 參考到 BET_TYPE 的主鍵。

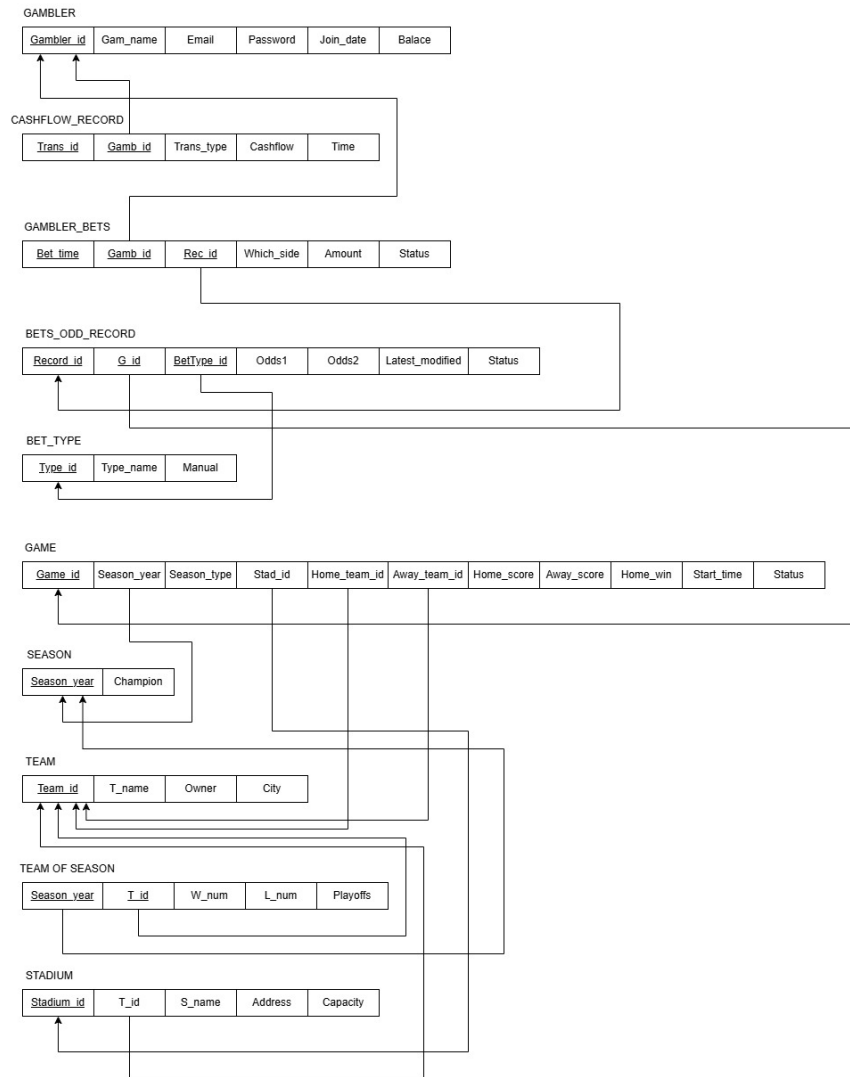


圖 2: 「Oin」的 Relational Database Schema Diagram

GAME 這個關聯的主鍵是 Game_id，外部鍵包括 Season_year 參考到 SEASON 的主鍵，Stadium_id 參考到 STADIUM 的主鍵，以及 Home_team_id 和 Away_team_id 參考到 TEAM 的主鍵。而 STADIUM 這個關聯的主鍵是 Stadium_id。

SEASON 這個關聯的主鍵是 Season_name。TEAM 這個關聯的主鍵是 Team_id。由於 TEAM 和 SEASON 存在多對多的關係「OF」，因此特別將其獨立出來形成 TEAM_OF_SEASON 關聯。該關聯的主鍵由 Team_id 和 Season_year 組成，並作為外部鍵分別參考到 TEAM 的 Team_id 和 SEASON 的 Season_year。

2.3 Data Dictionary

「Oin」的資料表共有圖 2 所示的十個，各個資料表的欄位相關資訊依序呈現在表 1 到表 10。

Column Name	Meaning	Data Type	Key	Constraint	Domain
Gambler_id	投注者代號	bigint	PK	Not Null	
Gam_name	投注者名稱	varchar(30)		Not Null, Unique	
Email	電子郵件	varchar(50)		Not Null	
Password	密碼	varchar(50)		Not Null	
Join_date	加入日期	date		Not Null	
Balance	餘額	decimal		Not Null	

表 1: 資料表 GAMBLER 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Trans_id	交易代號	bigint	PK	Not Null	
Gamb_id	玩家代號	bigint	PK, FK: GAMBLER(Gambler_id)	Not Null	
Trans_type	交易類型	varchar(20)		Not Null	
Cashflow	現金流量	decimal		Not Null	
Time	交易時間	datetime		Not Null	
Referential triggers		On Delete	On Update		
Gamb_id: GAMBLER(Gambler_id)		Cascade	Cascade		

表 2: 資料表 CASHFLOW_RECORD 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Bet_time	下注時間	datetime	PK	Not Null	
Gamb_id	投注者代號	bigint	PK, FK: GAMBLER(Gambler_id)	Not Null	
Rec_id	賠率記錄代號	bigint	PK, FK: BETS_ODD_RECORD(Record_id)	Not Null	
Which_side	下注區選擇	varchar(10)		Not Null	
Amount	金額	int		Not Null	
Status	狀態	varchar(20)		Not Null	Completed, Cancelled, Pending
Referential triggers		On Delete	On Update		
Gamb_id: GAMBLER(Gambler_id)		Cascade	Cascade		
Rec_id: BETS_ODD_RECORD(Record_id)		Cascade	Cascade		

表 3: 資料表 GAMBLER_BETS 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Record_id	賠率記錄代號	bigint	PK	Not Null	
Game_id	比賽代號	bigint	PK, FK: GAME(Game_id)	Not Null	
Type_id	下注類型代號	bigint	PK, FK: BET_TYPE(Type_id)	Not Null	
Odd_1	賠率 1	decimal		Not Null	
Odd_2	賠率 2	decimal		Not Null	
Latest_modified	最後修改時間	datetime		Not Null	
Status	狀態	varchar(20)		Not Null	Expired, Processing , Not yet started
Referential triggers		On Delete	On Update		
Game_id: GAME(Game_id)		Cascade	Cascade		
Type_id: BET_TYPE(Type_id)		Set null	Cascade		

表 4: 資料表 BETS_ODD_RECORD 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Type_id	下注類型代號	bigint	PK	Not Null	
Type_name	類型名稱	varchar(15)		Not Null	
Manual	玩法說明	Varchar(500)		Not Null	

表 5: 資料表 BET_TYPE 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Game_id	比賽代號	bigint	PK	Not Null	
Season_year	賽季年份	int	FK: SEASON(Season_year)	Not Null	
Season_type	賽季類型	varchar(10)		Not Null	
Stadium_id	球場代號	bigint	FK: STADIUM(Stadium_id)	Not Null	
Home_team_id	主隊代號	bigint	FK: TEAM(Team_id)	Not Null	
Away_team_id	客隊代號	bigint	FK: TEAM(Team_id)	Not Null	
Home_score	主隊得分	int			
Away_score	客隊得分	int			
Home_win	主隊勝利	char(1)			W, L
Start_time	開賽時間	datetime		Not Null	
Status	狀態	varchar(20)		Not Null	Ongoing, Ended, Not yet started
Referential triggers		On Delete	On Update		
Season_year: SEASON(Season_year)		Set NULL	Cascade		
Stadium_id: STADIUM(Stadium_id)		Set NULL	Cascade		
Home_team_id: TEAM(Team_id)		Cascade	Cascade		
Away_team_id: TEAM(Team_id)		Cascade	Cascade		

表 6: 資料表 GAME 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Season__year	賽季名稱	varchar(10)	PK	Not Null	
Champion	冠軍隊伍	bigint			

表 7: 資料表 SEASON 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Team__id	隊伍代號	bigint	PK	Not Null	
T__name	隊伍名稱	varchar(30)		Not Null	
Owner	擁有者	varchar(30)		Not Null	
City	所在城市	varchar(50)		Not Null	

表 8: 資料表 TEAM 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Season__year	賽季年份	int	PK, FK: GAME(Season__year)	Not Null	
Team__id	隊伍代號	bigint	PK, FK: TEAM(Team__id)	Not Null	
W__num	勝場數	int			
L__num	敗場數	int			
Playoffs	是否進入季後賽	boolean			1, 0
Referential triggers		On Delete	On Update		
Team__id: TEAM(Team__id)		Set NULL	Cascade		
Season__year: SEASON(Season__year)		Set NULL	Cascade		

表 9: 資料表 TEAM_OF_SEASON 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Stadium_id	球場代號	bigint	PK	Not Null	
T_id	隸屬隊伍代號	bigint	FK: TEAM(Team_id)		
S_name	球場名稱	varchar(20)		Not Null	
Address	地址	varchar(30)		Not Null	
Capacity	容量	int			
Referential triggers		On Delete	On Update		
T_id: TEAM(Team_id)		Set Null	Cascade		

表 10: 資料表 STADIUM 的欄位資訊

2.4 正規化分析

當設計關聯式資料庫時，我們可以檢視資料庫綱目（database schema）是否滿足正規化（normalization）條件，因此我們將依序從第一正規式（1NF）到第四正規式（4NF）來說明「Oin」的關聯是如何滿足這些規則。

在 1NF 方面，如果每個關聯的屬性都是 simple 且 single-valued，換句話說，在關聯中沒有任何一個屬性是 composite 或 multi-valued，則滿足 1NF。在所有 relation 裡，每一個 attributes 的每個欄位都只會有單一個值，不會有一個欄位儲存多個值，故我們的 schema 滿足 1 NF。

在 2NF 方面，如果關聯中的所有非鍵屬性（non-prime attribute）都完全功能相依（fully functional dependency）於任一候選鍵（candidate key），也就是沒有出現部分功能相依性（partial functional dependency），且此關聯滿足 1NF，則滿足 2NF。仔細觀察我們的 relation schema 就可發現，所有 PK 都不是多餘的，並且在每個 table，都不會有 partial dependency 的情形產生，因此本 schema 也符合 2NF。

在 3NF 方面，如果一個關聯中的非鍵屬性都沒有遞移相依（transitively dependency）於主鍵，則滿足 3NF。因此同樣檢視一下設計的關聯，的確有符合 3NF。在比 3NF 更嚴謹的 BCNF 方面，要求關聯中的每一個功能相依的箭頭左方都要是超級鍵（superkey），也就是要確保 $X \rightarrow Y$ 的 X 一定是超級鍵。而我們的 schema 的 PK，都足以產生所有的 non prime attribute，並不會有 transitive dependency 的情形產生，故也符合 3NF。

最後是 4NF，由於「Oin」的所有關聯都不存在多值相依（multi-valued dependency），因此滿足 4NF 的條件。