

# 資料庫管理

## 舞告 Match 期末專案專案書

第 2 組

資管三 b12705019 劉軒羽

資管三 b12705031 汪芷瑩

資管三 b12705039 蔡政穎

November 1, 2025

### 1 系統分析

想翻跳喜歡的 Kpop 舞曲卻找不到合拍的夥伴？練習時間總是喬不攏、舞風又怕不契合？別擔心，你不是一個人！「舞告 Match」幫你找到最和拍的舞伴！

「舞告 Match」是一個專為 Kpop 舞蹈翻跳愛好者打造的線上媒合平台，致力於協助使用者組建理想的舞蹈團隊。透過本平台，使用者可以從豐富的 Kpop 曲庫中挑選心儀歌曲，發起舞蹈翻跳計畫，並明確標示所需的團隊人數、練習地點與預期時程規劃。其他使用者則可以依據自身舞蹈實力與興趣，瀏覽並加入符合條件的翻跳計畫。

本平台所定義的「翻跳專案」(Dance Cover Project) 是一種以完成單一翻跳作品為目的的短期舞蹈合作活動。參與成員可能是曾經合作過的熟悉舞伴，也可能是透過平台初次相遇的新朋友。每當使用者想要完成一個新的翻跳作品，就必須在平台上建立一個全新的專案，每個專案都擁有獨立的識別編號以及完整的生命週期管理。因此，本平台中的「翻跳專案」代表的是一次性的舞蹈合作活動，而非長期固定的舞團組織。

依據功能權限與使用目的，「舞告 Match」的使用者分為兩種類型：**一般使用者 (User)** 與 **業務經營者 (Admin)**。一般使用者可以根據個人需求，選擇發起新的翻跳專案或報名參與現有專案。發起專案時，使用者需要從曲庫選擇翻跳歌曲、撰寫專案說明、設定所需團隊人數、規劃練習地點與時段等資訊。若選擇參與他人專案，使用者可以透過篩選條件（如歌曲名稱、團體名稱、練習地區、舞蹈難度等）瀏覽平台上的招募中專案，並提交報名申請。業務經營者主要職責包括維護平台的 Kpop 歌曲與團體資料庫、以及查詢使用者活動記錄與專案詳細資訊，藉此確保平台內容品質與使用者體驗。

## 1.1 系統功能

### 1.1.1 翻跳專案的規範與設定

平台提供完整的 Kpop 歌曲資料庫供使用者檢索與選擇。資料庫中的每首歌曲都包含詳盡的基本資訊，包括歌曲名稱（支援英文與韓文）、演唱團體或歌手、發行日期等。此外，系統也會標註歌曲的舞蹈難度指標，區分為初階、中階、高階三個等級，並提供原始 YouTube MV 與 Spotify 的外部連結。使用者在建立翻跳專案時，必須從此資料庫中指定一首目標歌曲。

系統要求使用者在發起專案時明確設定所需的團隊人數。使用者需要填寫目標成員總數，並可以選擇性地說明是否需要特定的舞蹈角色分配（如主舞、主唱等）。平台會即時更新並顯示目前已招募的成員人數以及剩餘的空缺名額。

使用者在發起專案時需要設定練習地點與拍攝地點，並可新增具體的練習時段安排，包含練習日期、開始時間與結束時間，讓成員清楚知道練習排程。系統會自動追蹤專案狀態，包含「招募中」、「進行中」、「已完成」、「已取消」等，使用者也可以手動調整專案狀態。

以實際案例說明：假設使用者小華想要發起一個翻跳 SEVENTEEN 《HOT》的專案。她首先在系統中搜尋並選擇歌曲《HOT》，接著輸入專案標題「徵求熱血 HOT 翻跳夥伴！」並撰寫專案理念說明。隨後她設定所需成員人數以及角色，並標註練習地點為「板橋火車站」，拍攝地點為「貴婦百貨」。之後她可以新增練習時段，例如第一次練習為「2025/01/19 14:00-17:00」。系統會將這些完整資訊儲存並公開在平台上，供其他使用者瀏覽與報名參與。

### 1.1.2 一般使用者功能

在本系統中，一般使用者可以執行以下功能：

1. **管理個人資料與作品集**使用者可以編輯個人資料（暱稱、地區、技能熟練度、經驗年資、社群媒體連結等），以及新增、修改、刪除作品集項目（包含作品標題、影片連結、描述等）。

2. **建立翻跳專案**

使用者可以發起翻跳專案，系統會自動分配唯一識別編號並將專案狀態設為「招募中」，公開在平台上供其他使用者瀏覽與申請加入。

3. **管理已發起的專案**

專案發起人可以審核加入申請，檢視申請者的個人資料、舞蹈經驗與作品集，並決定核准或婉拒。發起人也可以更新練習時程與地點，並追蹤專案進度。

#### 4. 申請加入專案

使用者可以瀏覽「招募中」的專案，透過歌曲名稱、團體、難度、地區等條件篩選。選定專案後填寫申請表單，系統會通知專案發起人進行審核。

#### 5. 查詢個人專案記錄

使用者可以查詢自己發起或參與過的所有專案，包含各種狀態（招募中、進行中、已完成、已取消等），系統會顯示專案基本資訊、狀態、成員名單與進度。

#### 6. 上傳完成作品

專案完成後，發起人可以上傳翻跳影片，填寫作品標題、描述、影片連結與製作團隊名單。完成作品會公開展示在平台上，供其他使用者欣賞與互動。

### 1.1.3 業務經營者功能

在本系統中，業務經營者可以執行以下功能：

#### 1. 管理 Kpop 團體與歌曲資料庫

管理員可以對團體與歌曲資訊進行新增、刪除、修改、查詢等操作。團體資料包含團體名稱（英文與韓文）、出道日期、經紀公司、團體類型、成員人數等。歌曲資料包含歌曲名稱（英文與韓文）、演唱團體、發行日期、專輯、音樂風格、舞蹈難度、MV 連結等。

#### 2. 查詢使用者資訊與活動紀錄

管理員可以查詢任何使用者的完整資料與活動紀錄，包括個人資料、發起與參與的專案、作品集、互動記錄等。

#### 3. 查詢專案完整資訊

管理員可以查詢任何專案的完整資訊，包括基本資料、成員名單與位置、申請記錄、進度與最終成果等。此功能用於了解專案運作或進行數據分析。

#### 4. 查詢平台營運統計數據

管理員可以查看平台營運數據，包括註冊與活躍使用者數、各狀態專案數量、熱門翻跳歌曲排行、地區分布、人數規模統計、專案完成率與完成時間等，作為內容規劃與功能優化的參考。

## 2 系統設計

### 2.1 ER Diagram

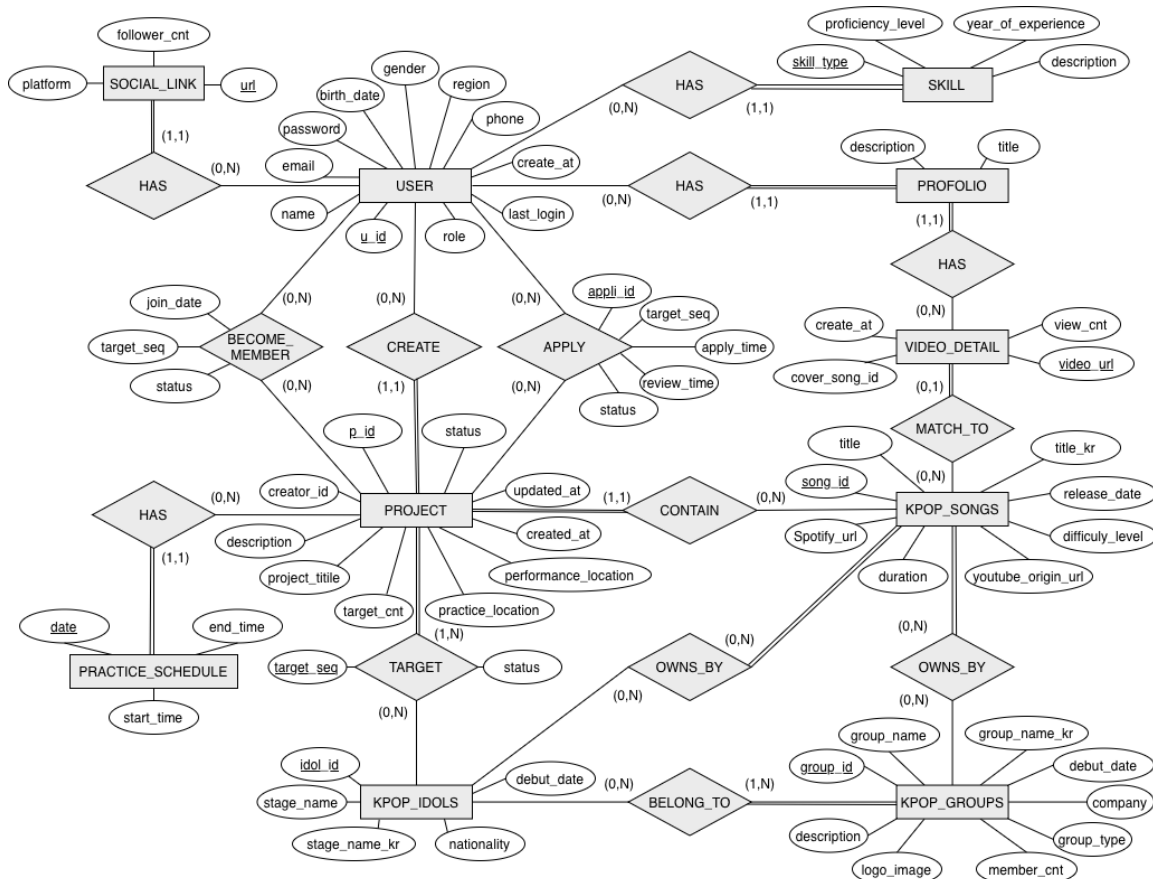


Figure 1: 「舞告 Match」的 ER Diagram

圖 1 是「舞告 Match」的 ER Diagram，共有十個實體 (entity)，分別是 USER、PROJECT、KPOP\_SONGS、KPOP\_GROUPS、KPOP\_IDOLS、PRACTICE\_SCHEDULE、SKILL、PORTFOLIO、VIDEO\_DETAIL、SOCIAL\_LINK，以及九個關係 (relationship)，包括 CREATE、APPLY、BECOME\_MEMBER、CONTAIN、MATCH\_TO、OWNS\_BY (兩個)、BELONG\_TO、TARGET、HAS (五個)。

**USER** 代表的是使用「舞告 Match」的任何人，任何人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 (name)、信箱 (email)、密碼 (password)，並可選擇性填寫出生日期 (birth\_date)、性別 (gender)、所在地區 (region) 及聯絡電話 (phone)。經註冊後便會產生一個專屬於該位使用者的使用者代號 (u\_id)，並記錄建立時間 (create\_at) 及最後登入時間 (last\_login)。經過後臺手動設定，可以將特定使用者設定為不同角色 (role)，例如一般使用者 (User) 或管理者 (Amdin)。

使用者可以擁有多項技能 (SKILL)，每項技能包含技能類型 (skill\_type)、熟練程度 (proficiency\_level)、經驗年數 (year\_of\_experience) 及技能描述 (description)。

這些技能資訊將幫助其他使用者評估該使用者是否適合特定的翻跳專案角色。此外，使用者也可以建立個人作品集 (PORTFOLIO)，展示過去的表演或練習影片。作品集包含標題 (title) 及描述 (description)，並透過 VIDEO\_DETAIL 來記錄影片的具體資訊，包括影片網址 (video\_url)、建立時間 (create\_at)、觀看次數 (view\_cnt) 及對應的歌曲代號 (cover\_song\_id)。

使用者可以透過社群媒體連結 (SOCIAL\_LINK) 分享自己的社群平台帳號，系統會記錄平台名稱 (platform)、個人頁面網址 (url) 及追蹤者數量 (follower\_cnt)，讓其他使用者能更了解其社群影響力及活躍度。

**PROJECT** 代表一個翻跳專案，由使用者發起 (CREATE 關係, 1:1)。發起專案時，系統會為該專案建立一個專案代號 (p\_id)，請使用者提供專案標題 (project\_title)、專案描述 (description)、練習地點 (practice\_location)、表演地點 (performance\_location)、目標人數 (target\_cnt)，以及建立時間 (created\_at) 和最後更新時間 (updated\_at)。系統會記錄專案的狀態 (status)，例如 A (可報名)、D (解散)、F (已徵滿)。

一個專案必定會對應到一首歌曲 (CONTAIN 關係, 1:1)，該歌曲由 **KPOP\_SONGS** 實體表示。**KPOP\_SONGS** 記錄歌曲的詳細資訊，包括歌曲代號 (song\_id)、歌曲標題 (title)、韓文標題 (title\_kr)、發行日期 (release\_date)、歌曲長度 (duration)、難度等級 (difficulty\_level)、Spotify 連結 (Spotify\_url) 及 YouTube 原曲連結 (youtube\_origin\_url)。**VIDEO\_DETAIL** 透過 **MATCH\_TO** 關係 (1:N) 連結到 **KPOP\_SONGS**，使得使用者可以將自己的 cover 影片與原曲進行對應。

每首歌曲可能由團體或 Solo 歌手演唱，透過 **OWNS\_BY** 關係連結到 **KPOP\_GROUPS** (0:N) 或 **KPOP\_IDOLS** (0:N)。**KPOP\_GROUPS** 記錄團體的基本資訊，包括團體代號 (group\_id)、團體名稱 (group\_name)、韓文名稱 (group\_name\_kr)、出道日期 (debut\_date)、經紀公司 (company)、團體類型 (group\_type)，例如 B (男團)、G (女團) 或 M (混合團體)、成員人數 (member\_cnt)、Logo 圖片 (logo\_image) 及團體描述 (description)。**KPOP\_IDOLS** 則記錄個別偶像的資訊，包括偶像代號 (idol\_id)、藝名 (stage\_name)、韓文藝名 (stage\_name\_kr)、國籍 (nationality) 及出道日期 (debut\_date)。

團體與偶像之間透過 **BELONG\_TO** 關係 (0:N 對 1:N) 建立連結，因為一個偶像可能同時屬於多個團體 (例如小分隊或聯合團體)，而一個團體必須由多位偶像組成。

專案會透過 **TARGET** 關係 (1:N) 定義需要招募的角色，招募的有可能是某一個 Idol 的位置，也有可能是伴舞。每個角色會記錄目標序號 (target\_seq)，代表該角色是這個專案第幾個需招募的角色，以及狀態 (status，例如 F (已徵到)、I (正在徵) 或 D (被刪除))。若該職位是要 cover 特定偶像的角色，則會透過 idol\_id 連結到 **KPOP\_IDOLS** 實體；若是招伴舞，則會設定 idol\_id = NULL。

使用者可以透過 **APPLY** 關係 (0:N) 申請加入專案的某個角色。申請時會建立一個申請代號方便查閱 (appli\_id)、目標角色序號 (target\_seq)、申請時間 (apply\_time)、

審核時間 (review\_time) 及申請狀態 (status, 例如 A(通過)、R (拒絕)、W (待回覆)、C (取消))。若申請被接受, 使用者便會透過 **BECOME\_MEMBER** 關係 (0:N) 正式加入專案, 成為專案成員。成為成員時會記錄加入日期 (join\_date)、目標職位 (target\_id) 及成員狀態 (status, 例如 Y(還在團隊)、N (已退出))。

專案會有多個排練時間安排, 透過 **PRACTICE\_SCHEDULE** 實體 (HAS 關係, 1:N) 來管理。每個排練包含日期 (date)、開始時間 (start\_time) 及結束時間 (end\_time)。

## 2.2 Relational Database Schema Diagram

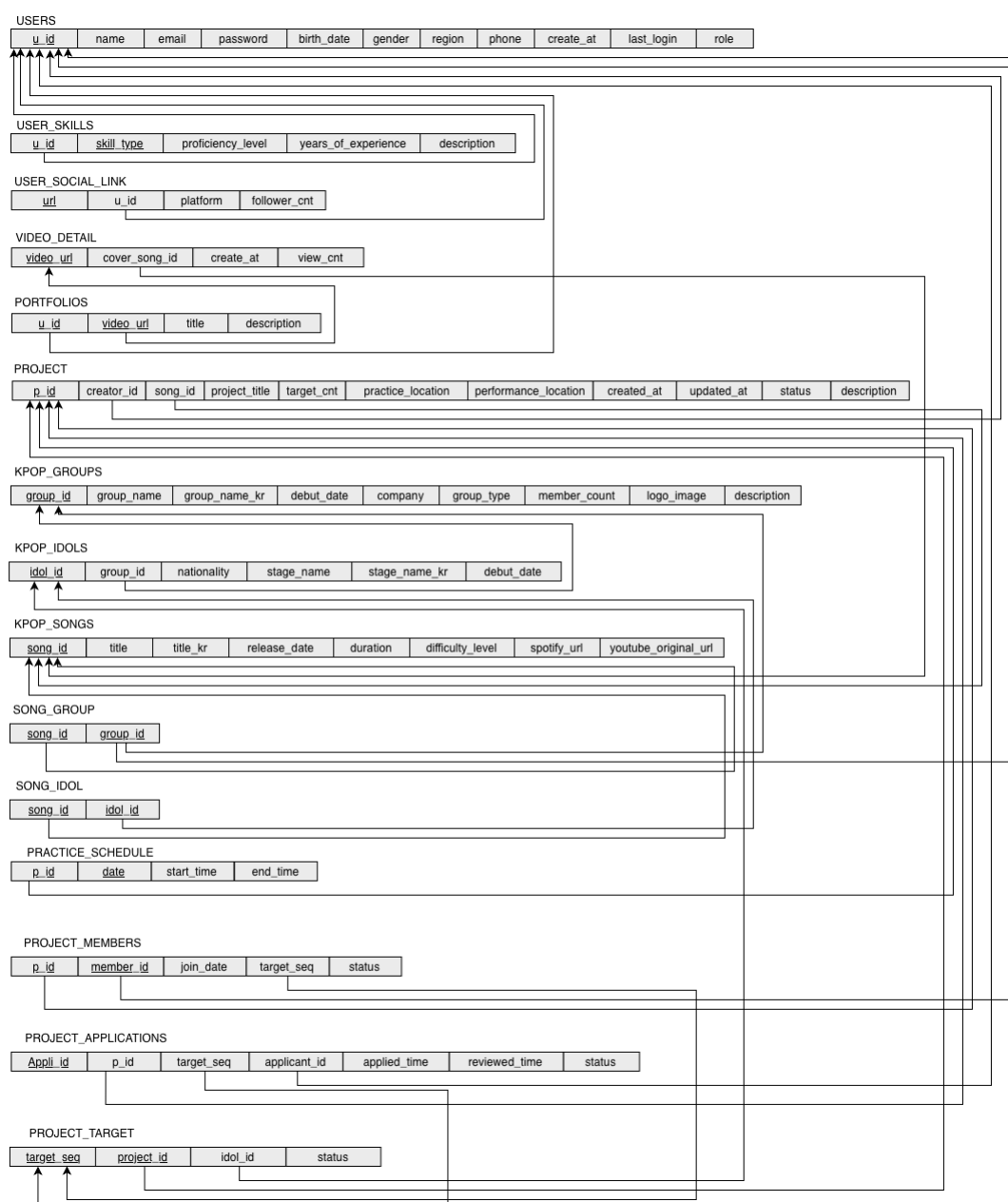


Figure 2: 「舞告 Match」的 Relational Database Schema Diagram

我們將 ER Diagram 轉換成圖中的 Database Schema, 一共由**十五個關聯 (relation)** 組成, 以下將逐一說明各關聯的設計。

**USERS** 這個關聯的主鍵 (Primary key, PK) 是 **u\_id**。

**USER\_SKILLS** 關聯是由 **USER** 的技能延伸而來, 因為一個使用者可以擁有多項技能 (一對多關係), 所以必須獨立出來成為一個關聯。該關聯的主鍵由 **u\_id** 和 **skill\_type** 組合而成, 其中 **u\_id** 同時也作為外鍵 (Foreign key, FK) 參考到 **USERS** 的主鍵 **u\_id**。

**USER\_SOCIAL\_LINK** 關聯同樣源自於使用者的多值屬性 (社群連結), 該關聯的主鍵為 **url**, 且 **u\_id** 作為外鍵參考到 **USERS** 的主鍵。

**VIDEO\_DETAIL** 關聯的主鍵是 **video\_url**, 外鍵包括參考到 **KPOP\_SONGS** 主鍵的 **cover\_song\_id**。

**PORTFOLIOS** 關聯的主鍵由 **u\_id** 和 **video\_url** 組合而成, 這兩個欄位同時也都作為外鍵, 分別參考到 **USERS** 的主鍵 **u\_id** 以及 **VIDEO\_DETAIL** 的主鍵 **video\_url**, 代表使用者的作品集與影片之間的關聯。

**PROJECT** 這個關聯的主鍵是 **p\_id**, 外鍵則包括參考到 **USERS** 主鍵的 **creator\_id** (專案發起人), 以及參考到 **KPOP\_SONGS** 主鍵的 **song\_id** (專案對應的歌曲)。

**KPOP\_GROUPS** 關聯的主鍵是 **group\_id**, 記錄 K-POP 團體的基本資訊。

**KPOP\_IDOLS** 關聯的主鍵是 **idol\_id**, 外鍵包括參考到 **KPOP\_GROUPS** 主鍵的 **group\_id**, 代表該偶像所屬的主要團體。

**KPOP\_SONGS** 關聯的主鍵是 **song\_id**, 記錄歌曲的詳細資訊。

**SONG\_GROUP** 和 **SONG\_IDOL** 這兩個關聯是由多對多關係產生而來。**SONG\_GROUP** 的主鍵由 **song\_id** 和 **group\_id** 組合而成, 這兩個欄位同時也作為外鍵, 分別參考到 **KPOP\_SONGS** 和 **KPOP\_GROUPS** 的主鍵。**SONG\_IDOL** 的結構類似, 主鍵由 **song\_id** 和 **idol\_id** 組成, 分別參考到 **KPOP\_SONGS** 和 **KPOP\_IDOLS** 的主鍵, 代表歌曲可以由團體或個人偶像演唱。

**PRACTICE\_SCHEDULE** 關聯的主鍵由 **p\_id**、**date** 組合而成, 其中 **p\_id** 作為外鍵參考到 **PROJECT** 的主鍵, 代表每個專案的排練時間安排。

**PROJECT\_MEMBERS** 關聯是由 **BECOME\_MEMBER** 關係產生而來, 該關聯的主鍵由 **p\_id** 和 **member\_id** 組合而成, 這兩個欄位同時也作為外鍵, 分別參考到 **PROJECT** 和 **USERS** 的主鍵。此外, **target\_seq** 作為外鍵參考到 **PROJECT\_TARGET**, 代表成員所擔任的角色。

**PROJECT\_APPLICATIONS** 關聯是由 **APPLY** 關係產生而來, 該關聯的主

鍵是 Appli\_id, 外鍵包括參考到 **PROJECT** 主鍵的 p\_id、參考到 **USERS** 主鍵的 applicant\_id, 以及參考到 **PROJECT\_TARGET** 的 target\_seq, 記錄使用者申請加入專案特定角色的資訊。

**PROJECT\_TARGET** 關聯是由 **TARGET** 關係產生而來, 該關聯的主鍵由 target\_seq 和 project\_id 組合而成, 其中 project\_id 作為外鍵參考到 **PROJECT** 的主鍵, idol\_id 作為外鍵參考到 **KPOP\_IDOLS** 的主鍵 (若為 NULL 則代表招募伴舞), 定義專案需要招募的角色。

## 2.3 Data Dictionary

「舞告 Match」的資料表共有 Figure 2 所示的十五個, 各個資料表的欄位相關資訊依序呈現在 Table 1 到 Table 15。

Column	Meaning	Data Type	Key	Constraint	Domain
u_id	使用者代號	bigint	PK	Not Null	
name	使用者名稱	varchar(15)		Not Null, Unique	
email	使用者信箱	varchar(30)		Not Null, Unique	
password	使用者密碼	varchar(15)		Not Null	
birthdate	使用者生日	date		Not Null	
gender	使用者性別	char		Not Null	{B, G}
region	所在地區	varchar(20)		Not Null	
phone	註冊電話	char(10)		Not Null, Unique	
status	可用情況	char		Not Null	{A, N}
create_at	註冊時間	timestamp		Not Null	
last_login	上次登入	timestamp		Not Null	
role	角色	char		NotNull	{U, A}

Table 1: 資料表 USERS 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
u_id	使用者代號	bigint	PK, FK : USERS(u_id)	Not Null	
skill_type	技能名稱	varchar(15)	PK	Not Null	
proficiency_level	熟悉程度	int		Not Null	[0, 100]
years_of_experience	經驗 (年)	int		Not Null	
discription	簡單說明	varchar(200)			
<b>Referential triggers</b>		<b>On Delete</b>	<b>On Update</b>		
u_id: USERS(u_id)	Cascade	Cascade			

Table 2: 資料表 USER\_SKILLS 的欄位資訊



Column	Meaning	Data Type	Key	Constraint	Domain
url	社群連結	varchar(100)	PK	Not Null	
u_id	使用者代號	bigint	FK : USERS(u_id)	Not Null	
platform	社群平台	varchar(20)		Not Null	
follower_cnt	關注人數	int		Not Null	
<b>Referential triggers</b>		<b>On Delete</b>	<b>On Update</b>		
u_id: USERS(u_id)		Cascade	Cascade		

Table 3: 資料表 USER\_SOCIAL\_LINK 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
video_url	影片連結	varchar(100)	PK	Not Null	
cover_song_id	Cover 的歌曲代號	bigint	FK : KPOP_SONGS(song_id)	Not Null	
created_at	創建時間	timestamp		Not Null	
view_cnt	觀看次數	int		Not Null	
<b>Referential triggers</b>		<b>On Delete</b>	<b>On Update</b>		
cover_song_id: KPOP_SONGS(song_id)		Set Null	Cascade		

Table 4: 資料表 VIDEO\_DETAIL 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
u_id	使用者代號	bigint	PK, FK : USERS(u_id)	Not Null	
video_url	Cover 影片連結	varchar(100)	PK, FK : VIDEO_DETAIL(video_url)	Not Null	
title	作品輯	varchar(20)		Not Null	
discription	作品簡介	varchar(500)			
<b>Referential triggers</b>		<b>On Delete</b>	<b>On Update</b>		
u_id: USERS(u_id)		Cascade	Cascade		
video_url: VIDEO_DETAIL(video_url)		Set Null	Cascade		

Table 5: 資料表 PORTFOLIOS 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
p_id	專案代號	bigint	PK	Not Null	
creator_id	發起人代號	bigint	FK : USERS(u_id)	Not Null, Default('00000000')	
song_id	表演歌曲代號	bigint	FK : KPOP_SONGS(song_id)	Not Null	
porject_title	專案名稱	varchar(50)		Not Null	
target_cnt	目前缺的人數	int		Not Null	
practice_location	練習地點	varchar(50)		Not Null	
performance_location	表演地點	varchar(50)		Not Null	
create_at	建立時間	timestamp		Not Null	
update_at	最近更新時間	timestamp		Not Null	
status	狀態	char		Not Null	{A(可報名), D(解散), F(已徵滿)}
discription	團體簡介或要求	varchar(500)			
<b>Referential triggers</b>		<b>On Delete</b>	<b>On Update</b>		
creator_id: USERS(u_id)		Set Default	Cascade		
song_id: KPOP_SONGS(song_id)		Set Null	Cascade		

Table 6: 資料表 PROJECT 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
group_id	韓團代號	bigint	PK	Not Null	
group_name	韓團名稱	varchar(20)		Not Null	
group_namekr	韓團韓文名稱	varchar(20)			
debut_date	出道日期	date		Not Null	
company	經紀公司名稱	varchar(20)		Not Null	
group_type	韓團種類 (男女)	char		Not Null	{B, G, M}
member_count	成員人數	int		Not Null	
logo_image	logo 儲存位置	varchar(100)			
discription	韓團簡介	char(500)			

Table 7: 資料表 KPOP\_GROUPS 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
idol_id	偶像代號	bigint	PK	Not Null	
group_id	所屬韓團	bigint	FK : KPOP_GROUPS(group_id)		
nationality	偶像國籍	varchar(20)		Not Null	
stage_name	偶像藝名	varchar(30)		Not Null	
stage_name_kr	韓文藝名	varchar(30)		Not Null	
debut_date	出道日期	date		Not Null	
Referential triggers		On Delete	On Update		
group_id: KPOP_GROUPS(group_id)		Set Null	Cascade		

Table 8: 資料表 KPOP\_IDOLS 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
song_id	歌曲代號	bigint	PK	Not Null	
title	歌曲名稱	varchar(50)		Not Null	
title_kr	歌曲韓文名稱	varchar(50)		Not Null	
release_date	發行日期	date		Not Null	
duration	歌曲長度 (秒)	int		Not Null	
difficulty_level	歌曲難度	int		Not Null	[0, 10]
spotify_url	歌曲 spotify 網址	varchar(100)			
youtube_original_url	歌曲 YT 網址	varchar(100)		Not Null	

Table 9: 資料表 KPOP\_SONGS 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
song_id	歌曲代號	bigint	PK, FK : KPOP_SONGS(song_id)	Not Null	
group_id	所屬團體代號	bigint	PK, FK : KPOP_GROUPS(group_id)	Not Null	
Referential triggers		On Delete	On Update		
song_id: KPOP_SONGS(song_id)		Cascade	Cascade		
group_id: KPOP_GROUPS(group_id)		Cascade	Cascade		

Table 10: 資料表 SONG\_GROUP 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
song_id	歌曲代號	bigint	PK, FK : KPOP_SONGS(song_id)	Not Null	
idol_id	所屬歌手代號	bigint	PK, FK : KPOP_IDOLS(idol_id)	Not Null	
Referential triggers		On Delete	On Update		
song_id: KPOP_SONGS(song_id)		Cascade	Cascade		
idol_id: KPOP_IDOLS(idol_id)		Cascade	Cascade		

Table 11: 資料表 SONG\_IDOL 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
p_id	專案代號	bigint	PK, FK : PROJECT(p_id)	Not Null	
date	練習日期	date	PK	Not Null	
start_time	開始時間	time		Not Null	
end_time	結束時間	time		Not Null	
Referential triggers		On Delete	On Update		
p_id: PROJECT(p_id)		Cascade	Cascade		

Table 12: 資料表 PRACTICE\_SCHEDULE 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
p_id	專案代號	bigint	PK, FK : PROJECT(p_id)	Not Null	
member_id	成員代號	bigint	PK, FK : USERS(u_id)	Not Null	
join_date	加入日期	date		Not Null	
target_seq	參與徵求序號	bigint	FK : PROJECT_TARGET(target_id)		
status	是否仍在團隊	char		Not Null	Y(還在團隊), N(已退出)
Referential triggers		On Delete	On Update		
p_id: PROJECT(p_id)		Cascade	Cascade		
member_id: USERS(u_id)		Set Null	Cascade		
target_seq: PROJECT_TARGET(target_seq)		Set Null	Cascade		

Table 13: 資料表 PROJECT\_MEMBERS 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
Appli_id	申請代號	bigint	PK	Not Null	
p_id	專案代號	bigint	PK, FK : PROJECT(p_id)	Not Null	
target_seq	徵求角色序號	bigint	FK : PROJECT_TARGET(target_seq)	Not Null	
applicant_id	申請人代號	bigint	FK : USERS(u_id)	Not Null	
applied_time	申請時間	timestamp		Not Null	
reviewed_time	回覆時間	timestamp			
status	申請狀況	char		Not Null	{A(通過), R(拒絕), W(待回覆), C(取消)}
Referential triggers		On Delete	On Update		
p_id: PROJECT(p_id)		Cascade	Cascade		
target_seq: PROJECT_TARGET(target_seq)		Cascade	Cascade		
applicant_id: USERS(u_id)		Cascade	Cascade		

Table 14: 資料表 PROJECT\_APPLICATIONS 的欄位資訊

Column	Meaning	Data Type	Key	Constraint	Domain
target_seq	徵求角色序號 (每個專案從 1 開始)	int	PK	Not Null	
project_id	專案代號	bigint	PK, FK : PROJECT(p_id)	Not Null	
idol_id	該角色對應的 IDOL (伴舞為 NULL)	bigint	FK : KPOP_IDOLS(idol_id)		
status	狀態	char			{F (已徵到), I (正在徵), D (被刪除)}
Referential triggers		On Delete	On Update		
project_id: PROJECT(p_id)		Set Null	Cascade		
role_id: KPOP_IDOLS(idol_id)		Set Null	Cascade		

Table 15: 資料表 PROJECT\_TARGET 的欄位資訊

## 2.4 正規化分析

在設計關聯式資料庫時，我們需要檢視資料庫綱目是否滿足正規化條件。本系統的資料庫設計從第一正規式（1NF）至第四正規式（4NF）皆符合正規化要求，以下分別說明：

**第一正規式（1NF）：** 要求關聯中的每個屬性都不可包含複合屬性或多值屬性。本系統透過關聯分解達成此要求：使用者的多項技能被獨立為 **USER\_SKILLS** 關聯、社群連結資訊獨立為 **USER\_SOCIAL\_LINK** 關聯、排練時段獨立為 **PRACTICE\_SCHEDULE** 關聯。此外，多對多關係也透過關聯表處理，例如歌曲與團體的關係透過 **SONG\_GROUP**、歌曲與偶像的關係透過 **SONG\_IDOL** 來表達，確保所有屬性值都是單一且不可再分割的。

**第二正規式（2NF）：** 在滿足 1NF 的基礎上，要求所有非鍵屬性必須完全功能相依於主鍵，不允許部分相依。觀察本系統的關聯設計，具有組合鍵的關聯如 **USER\_SKILLS**，其非鍵屬性 *proficiency\_level*、*years\_of\_experience*、*description* 必須同時依賴 *u\_id* 與 *skill\_type* 才能決定，無法僅由其中一個欄位決定。同理，**PRACTICE\_SCHEDULE** 的 *end\_time* 需要完整的 (*p\_id*, *date*, *start\_time*) 才能確定。其他關聯同理。因此本系統不存在部分功能相依的問題。

**第三正規式（3NF）：** 在滿足 2NF 的條件下，進一步要求非鍵屬性之間不存在遞移相依關係。檢視本系統各關聯，**USERS** 中的 *name*、*email*、*region* 等屬性皆直接相依於 *u\_id*，彼此之間不存在功能相依；**KPOP\_IDOLS** 中的 *stage\_name*、*nationality*、*debut\_date* 也都直接相依於 *idol\_id*。所有關聯的非鍵屬性都直接相依於主鍵，未透過其他非鍵屬性間接相依，符合 3NF 規範。

**Boyce-Codd 正規式（BCNF）：** 這是比 3NF 更嚴格的正規化要求，規定關聯中每個功能相依  $X \rightarrow Y$  的決定因子  $X$  都必須是超級鍵。本系統的設計中，**PROJECT** 關聯的所有功能相依如  $p\_id \rightarrow project\_title$ 、 $p\_id \rightarrow status$  等，其決定因子  $p\_id$  即為主鍵；**PORTFOLIOS** 關聯中，組合鍵  $(u\_id, video\_url) \rightarrow title, description$  的決定因子也是主鍵。每個功能相依的左側都是該關聯的超級鍵，滿足 BCNF 要求。

**第四正規式（4NF）：** 要求關聯中不存在非平凡的多值相依，或所有多值相依  $X \twoheadrightarrow Y$  的  $X$  必須是超級鍵。本系統在設計階段即已識別並處理所有多值相依情況：一位使用者擁有多項技能與多個社群連結這兩個獨立的多值屬性，已分別獨立為 **USER\_SKILLS** 與 **USER\_SOCIAL\_LINK**；一首歌曲可能由團體演唱也可能由個人演唱的多值關係，透過 **SONG\_GROUP** 與 **SONG\_IDOL** 分開處理。這些分解確保了每個關聯中不存在會造成資料冗余的多值相依問題，符合 4NF 規範。