校園生活用品交換與買賣平台

Group 22
B12705001 范嘉栩　B12705021 周建凱
B12705050 宋承陽
日期：November 1, 2025

November 1, 2025

Contents

1 系統分析（System Analysis）

1.1 主題與動機（System Purpose & Motivation）

在現今的線上交易環境中，使用者獲取商品的方式日趨多元。除了傳統的現金購買，以物易物、混合交易等形式也逐漸興起，臉書社團等平台已展現出這類多元交易的可能性。然而，現有的多數交易平台仍存在明顯侷限——大多僅支援單向的購買流程，缺乏完善的物品交換機制與協商功能，無法滿足使用者多樣化的交易需求。

本系統特別針對大學生族群設計。大學生普遍經濟能力有限，對於課本、二手用品、電子產品等生活與學習資源有高度需求。透過整合「購買」與「交換」功能，使用者能同時扮演買家與賣家的雙重角色，根據自身需求自由提出交易請求——無論是現金交易、物品交換或混合模式，都能在同一平台上順暢完成。此設計不僅降低取得資源的經濟門檻，更能達成節省支出與資源再利用的雙重目標。

在技術層面，本系統採用結構化的資料庫設計與明確的交易流程規範，有效紀錄商品資訊、使用者請求關係與協商結果，確保資料一致性、可追蹤性與可稽核性，並為後續的推薦機制、信譽評價與營運分析奠定堅實基礎。

1.2 系統功能（System Functions）

本系統的主要目標是提供一個整合「購買」與「交換」的校園二手交易平台，使使用者能以彈性的方式取得與交換所需物品。依照系統角色，功能可區分為「功能設定說明」、「一般使用者功能」與「平台管理者功能」三大部分。

1.2.1 功能設定說明（System Function Settings）

為了確保交易流程的安全性與一致性，本系統在多項操作上設有明確規則與限制，主要包括以下幾點：

1. **交易模式設定：**商品上架時，使用者必須選擇一種交易模式，包含「販售（Sale）」、「交換（Trade）」或「兩者皆可（Both）」。若選擇交換或混合模式，可輸入期望交換之物品描述（例如「電腦滑鼠」、「小說」、「宿舍電鍋」等）或者不限。
2. **商品狀態（Status）限制：**每項商品的狀態皆由系統自動維護，分為 *available*（可交易）、*reserved*（預留中）、*sold*（已售出）、*exchanged*（已交換）及 *removed*（已下架）。完成交易或被移除後，商品將無法再次被提出請求。
3. **請求與撮合流程：**本系統的交易流程涵蓋「販售」、「交換」及「開放式許願」三種互動型態：
    - 若使用者僅販售商品，其他人可直接提出購買請求（記錄於 TRADE_REQUEST，類型為 Purchase）。
    - 若使用者想以物易物但市場尚無對應物品，系統會建立一筆許願紀錄 TRADE_WISH，表示其開放式需求。
    - 若市場已有可交換商品，系統依情境區分：

  - 若對方明確表示「想換該使用者的物品」，則此請求屬於 Request Target
  - 若對方僅開放交換但未指定欲換之物，則此請求屬於 Request Offer
- 當雙方確認交換條件或完成購買時，系統自動生成交易紀錄 TRANSACTION。

所有請求狀態均會被即時更新，若任一方拒絕或撤回，請求狀態改為 Cancelled 或 Rejected，不影響其他待處理請求。

4. **交易紀錄保存：**所有已完成的交易紀錄皆永久保存在資料庫中，作為後續查詢、統計及信譽評價依據。此舉確保平台資料的可追蹤性與透明性。
5. **分類與搜尋機制：**所有商品皆須隸屬於一個明確的分類（如：Textbooks、Electronics、Daily Use 等），使用者可透過分類、關鍵字或條件搜尋商品，以提升查找效率。
6. **時間與狀態更新：**系統自動記錄商品上架日期 Post_date、請求建立時間 Created_at 及更新時間 Updated_at，以利後續分析與查詢。

使用情境範例：

- 小明想出售微積分課本，設定交易模式為「Sale」，價格 300 元
- 小華想用電子辭典交換英文課本，設定交易模式為「Trade」
- 小美可接受現金或物品交換，設定交易模式為「Both」

上述設定確保平台運作邏輯一致、資料可追蹤，同時避免交易衝突與狀態錯誤。

1.2.2 一般使用者功能（User Functions）

一般使用者（User）為平台的主要參與者，具備以下功能：

1. **註冊與登入：**使用者可於平台上註冊帳號、登入系統，並設定基本資料（姓名、電子郵件、聯絡方式等）。
2. **商品上架：**使用者可新增商品資訊，包括商品名稱、分類、狀況、價格、描述、交易方式（販售／交換／混合）及圖片，完成上架。
3. **瀏覽與搜尋：**使用者可依照分類、名稱或關鍵字搜尋商品，並瀏覽詳細資訊。
4. **提出請求（購買／交換）：**使用者可對他人商品提出購買請求，或提供欲交換之物品並發送交換邀請。
5. **管理交易請求：**使用者可查看自己提出與收到的請求，選擇接受、拒絕或取消。
6. **查看交易紀錄：**使用者可查詢已完成或進行中的交易紀錄，包含交易對象、商品名稱、金額或交換內容。
7. **修改與刪除商品：**若商品尚未成交，使用者可修改或下架該商品。
8. **帳號管理：**使用者可更新個人資料、修改密碼或停用帳號。

透過上述功能，使用者能以彈性方式發起與參與交易，並藉由結構化的交易流程確保安全性與透明度。

1.2.3 平台管理者功能（Admin Functions）

平台管理者（Admin）負責維護整體系統的穩定運作與資訊正確性，具備以下功能：

1. **用戶管理：**管理者可檢視、停權或恢復使用者帳號，以維護平台秩序。
2. **商品管理：**管理者可審核或刪除不當商品（例如違規、重複或含不當內容的上架物品）。
3. **分類管理：**管理者可新增、修改或刪除商品分類，使平台分類結構保持清晰。
4. **交易監控：**管理者可查詢所有交易紀錄，包括交易狀態與雙方帳號資訊，以便處理爭議或異常情況。
5. **系統維護與分析：**管理者可進行資料備份、檢視統計報告，分析平台使用狀況（如熱門分類、交易量等）。

綜合而言，本系統透過明確的權限分工與資料庫設計，建立一個兼顧使用者體驗與系統安全性的校園二手交易生態環境。

2 系統設計（System Design）

2.1 實體關聯圖（ER Diagram）

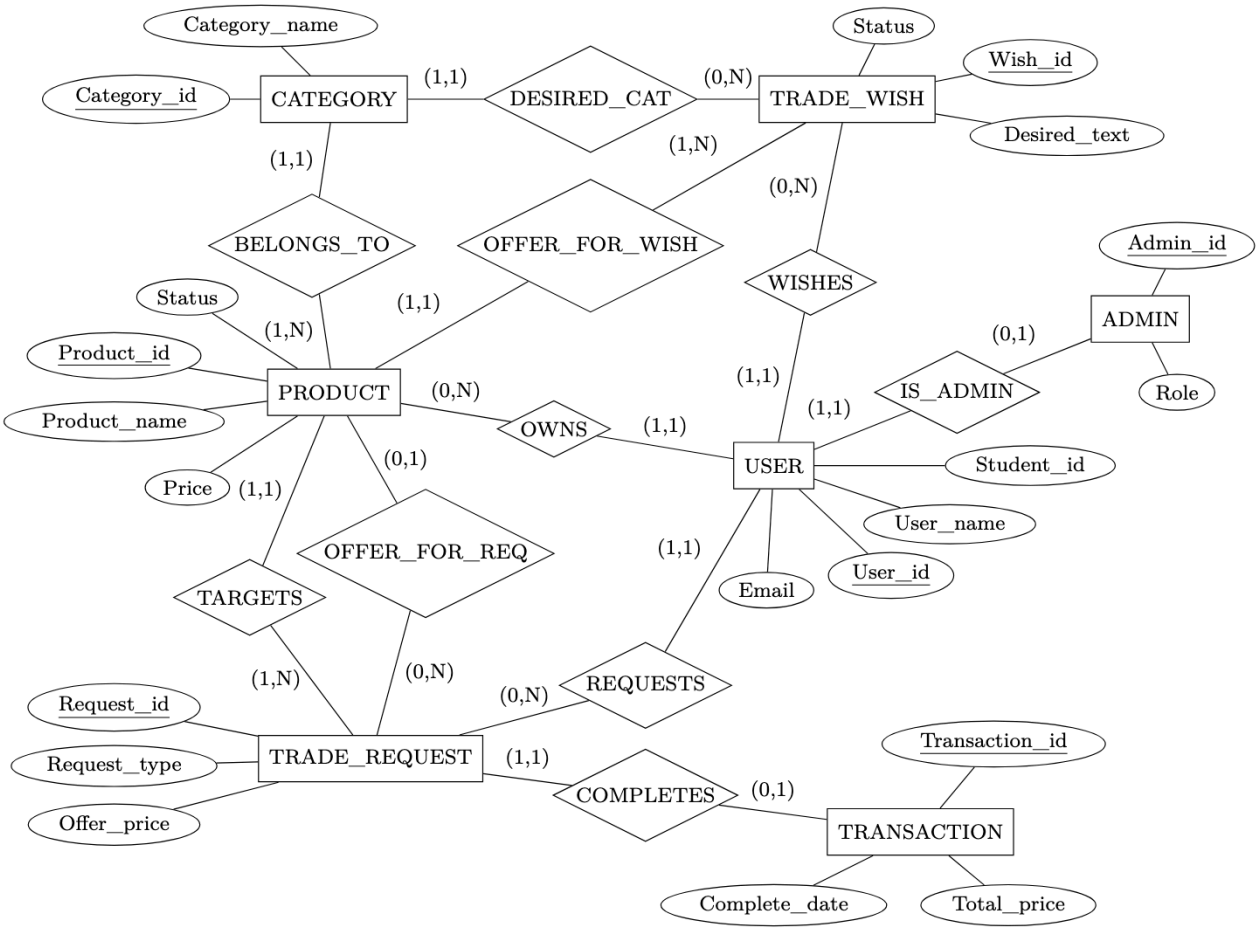

Figure 1: 本系統之 ER Diagram（實體關聯圖）

在此 ER Diagram 中，共有六個主要實體（Entity），分別是 USER、PRODUCT、CATEGORY、TRADE_REQUEST、TRANSACTION 以及 TRADE_WISH，同時也包含一個特殊角色實體 ADMIN。這些實體之間透過多個關係（Relationship）彼此連結，例如使用者上架商品（OWN）、發送交易請求（REQUEST）、完成交易（COMPLETE）及發起許願（WISH）等。

USER 代表的是使用「校園生活用品交換與買賣平台」的任何人。任何使用者在進入系統前，皆須先註冊帳號，提供姓名、學號、電子郵件及密碼等基本資訊。註冊完成後，系統會自動為每位使用者分配唯一的 `User_id`，作為後續所有交易與操作的識別依據。若該使用者同時負責管理系統或審核商品，則會在後端被設定為 ADMIN，此時該使用者便同時具有一般使用者（User）與平台管理員（Admin）兩種身份。

PRODUCT 代表使用者所上架的商品。每一項商品皆由一位使用者所擁有（Owner_id

參考 USER），並歸屬於特定分類（Category_id 參考 CATEGORY）。商品包含名稱、價格、狀況、圖片、上架日期與交易方式（販售、交換或混合）等屬性。若商品開放交換，使用者可以輸入期望交換的物品描述（Trade_item），作為後續撮合的依據。

CATEGORY 則用來記錄商品的類別資訊，例如「教科書」、「電子產品」、「生活用品」等。每項商品必須歸屬於一個明確分類，以便後續搜尋與統計分析。

TRADE_REQUEST 為交易請求的關聯實體，用以記錄使用者之間的互動。當某位使用者想購買或交換他人商品時，系統會產生一筆交易請求紀錄，其中包含請求者（Requester_id）、目標商品（Target_product_id）及請求類型（Request_type）等欄位。若為交換請求，還會紀錄該使用者提供的交換商品（Offered_product_id）。請求狀態會依交易流程變化，從 Pending、Accepted、Completed 至 Cancelled。

TRANSACTION 則對應於實際完成的交易紀錄。當請求雙方確認條件並完成交易後，系統會自動產生交易紀錄，其中包含完成日期、交易金額與付款狀態等欄位。此關聯可視為 TRADE_REQUEST 的延伸，兩者呈現一對一關係。

TRADE_WISH 表示使用者對尚未出現的商品提出的開放式交換需求。使用者可描述欲交換之物品名稱、類別與條件，系統會將該許願資訊公開，並在有相符商品上架時提供通知。此設計使交換流程不僅限於即時配對，也能支持延遲撮合的情境。

ADMIN 為平台管理員，屬於特殊使用者。管理員可審核上架商品、刪除違規內容、管理分類、公告規則與查看平台統計數據，以維護整體系統的安全與秩序。

整體而言，本系統的 ER Diagram 以 USER 為核心，透過多對多與一對多的關係串接其他實體，完整呈現使用者從上架、瀏覽、請求到完成交易的全流程互動。此結構亦為後續關聯式資料庫設計提供明確依據。

2.2 關聯式資料庫綱目圖（Relational Database Schema Diagram）

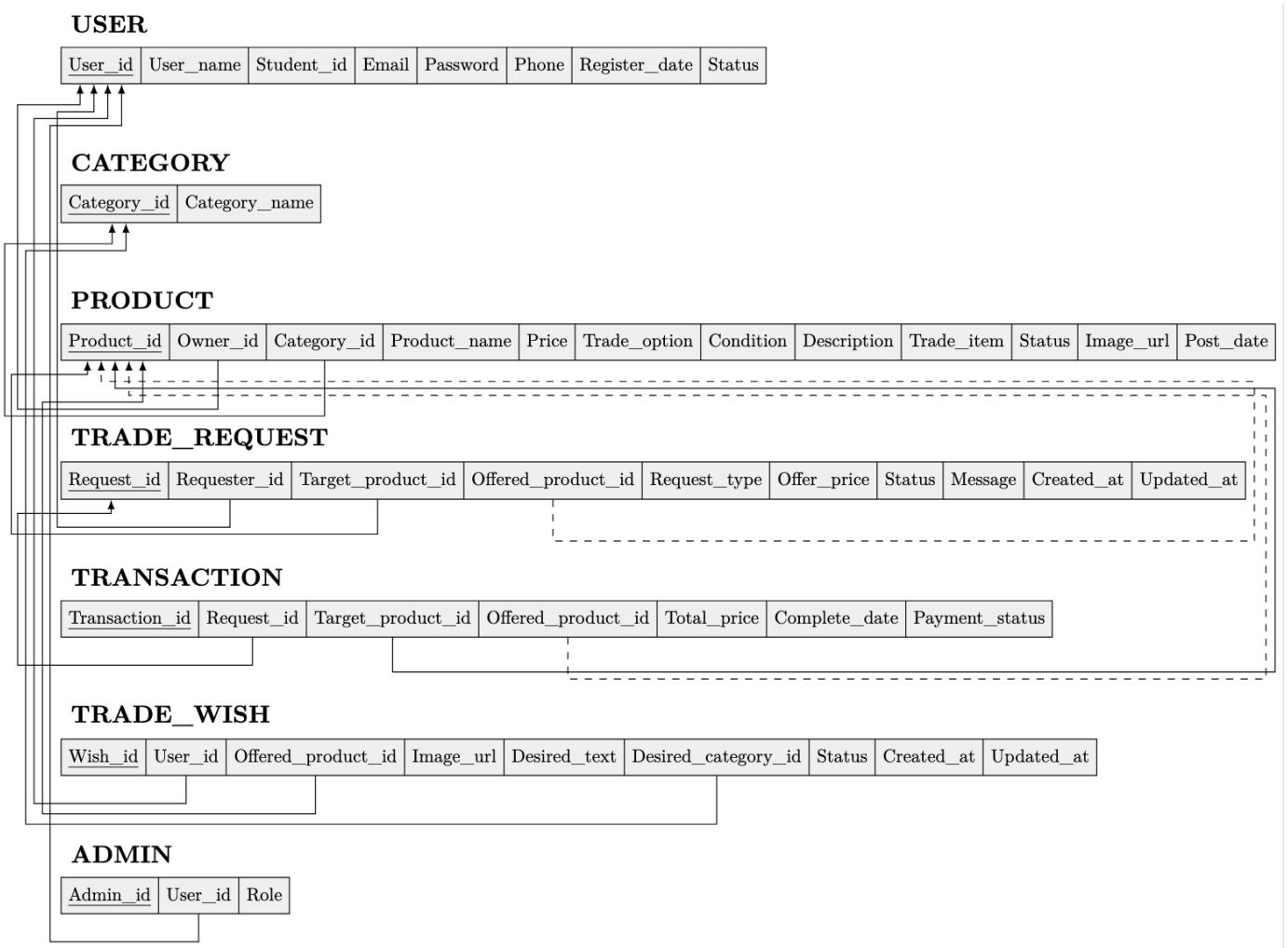

Figure 2: 本系統之 Relational Database Schema

整體 Schema 架構共包含六張主要資料表：USER、CATEGORY、PRODUCT、TRADE_REQUEST、TRANSACTION、TRADE_WISH，以及一張平台管理權限表 ADMIN。

USER 關聯的主鍵（Primary Key, PK）為 `User_id`，所有使用者的基本資料（姓名、學號、信箱、密碼等）皆依此識別。由於管理者身分為一般使用者的延伸，因此 ADMIN 關聯中同樣以 `User_id` 作為外鍵（Foreign Key, FK），並以 `Role` 欄位標示管理權限層級（例如 PlatformAdmin、ClubAdmin）。

PRODUCT 關聯的主鍵為 `Product_id`，外鍵包括 `Owner_id`（參考 USER）與 `Category_id`（參考 CATEGORY）。商品相關資訊如價格、描述、交易狀況、圖片網址等皆以屬性形式儲存。若該商品允許交換，`Trade_item` 欄位會記錄期望交換之物品描述。商品下架或交易完成後，其狀態（Status）會由系統自動更新為 sold, removed 或 exchanged。

CATEGORY 關聯則為靜態資料表，用以保存商品分類，主鍵為 Category_id，每項分類皆對應多筆商品。

TRADE_REQUEST 關聯是由使用者之間的 Request 關係轉換而來，主鍵為 Request_id。其外鍵包括 Requester_id（參考 USER）、Target_product_id（參考 PRODUCT）與 Offered_product_id（參考 PRODUCT）。此設計允許同時支援購買與交換兩種請求形式。當交易完成時，對應的請求會被標示為 Completed，並生成 TRANSACTION 紀錄。

TRANSACTION 關聯由 TRADE_REQUEST 延伸而來，主鍵為 Transaction_id，外鍵參考 Request_id。此表用於記錄最終成交的交易，包括完成日期與付款狀態（Paid ／ Unpaid）。此外，為確保資料完整性，交易刪除或修改時將觸發對應的參照更新規則（Cascade ／ Set Null）。

TRADE_WISH 關聯則由 TRADE_WISH 實體轉換而來，用以紀錄使用者開放式的交換願望。主鍵為 Wish_id，外鍵包括 User_id（參考 USER）、Offered_product_id（參考 PRODUCT）與 Desired_category_id（參考 CATEGORY）。此設計支援使用者提出交換需求，即使尚未出現對應商品，系統仍可追蹤並在後續進行配對。

整體而言，本系統之關聯式資料庫結構已將 ER Diagram 的多對多與層級關係完整轉換為實際可實作的關聯模型。各表的主鍵、外鍵與觸發規則均清楚定義，能確保交易資料的一致性、可追蹤性與高正規化設計品質。

2.3 資料字典（Data Dictionary）

以下為本系統之資料字典：在交易邏輯上，本系統同時支援「許願交換」、「目標交換」、「開放交換」與「直接購買」等多種互動情境。透過 TRADE_WISH、TRANSACTION、TRADE_REQUEST 三張表的設計，可完整覆蓋以下四種典型案例：

- **情況 1**：使用者想販售商品 → 建立 PRODUCT（Trade_option = 'sale'）。
- **情況 2**：想交換但市場尚無對應物品 → 建立 TRADE_WISH。
- **情況 3**：想交換且市場已有對應商品 → 建立 TRADE_REQUEST。若對方指定想換此物 → Request_Target；若對方僅開放交換 → Request_Offer。
- **情況 4**：直接購買現有商品 → 建立 TRADE_REQUEST（Request_type = 'Purchase'），完成後生成 TRANSACTION。

Column Name	Meaning	Data Type	Key	Constraint	Domain
User_id	使用者代號	bigint	PK	Not Null, Unique	
User_name	使用者名稱	varchar(20)		Not Null	
Student_id	學號	varchar(15)		Not Null, Unique	
Email	電子郵件	varchar(50)		Not Null, Unique	
Password	密碼	varchar(15)		Not Null	
Phone	聯絡電話	varchar(15)		Nullable	
Register_date	註冊日期	date		Not Null	
Status	帳號狀態	varchar(10)		Default 'active'	{active, suspended}

Table 1: 資料表 USER 的欄位資訊（使用者基本資料）

Column Name	Meaning	Data Type	Key	Constraint	Domain
Category_id	商品分類代號	int	PK	Auto Increment, Not Null	
Category_name	商品分類名稱	varchar(30)		Not Null	{Textbooks, Electronics, Clothing, Stationery, Daily_Use, Others}

Table 2: 資料表 CATEGORY 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Product_id	商品代號	bigint	PK	Not Null, Unique	
Owner_id	上架者代號	bigint	FK	Not Null	USER(User_id)
Category_id	商品分類代號	int	FK	Not Null	CATEGORY(Category_id)
Product_name	商品名稱	varchar(50)		Not Null	
Price	商品價格	int		Nullable	≥ 0；*僅販售時使用*
Trade_option	交易方式	varchar(10)		Default 'sale'	`{sale, trade, both}`
Condition	商品狀況	varchar(20)		Not Null	`{Brand_New, Good, Used}`
Description	商品描述	varchar(100)			
Trade_item	欲交換之物品	varchar(100)		Nullable	*當 Trade_option 為 trade/both 時填寫*
Status	銷售狀態	varchar(20)		Default 'available'	`{available, reserved, sold, exchanged, removed}`
Image_url	圖片網址	varchar(255)			
Post_date	上架日期	datetime		Not Null	
Referential triggers				**On Delete**	**On Update**
Owner_id: USER(User_id)				Cascade	Cascade
Category_id: CATEGORY(Category_id)				Restrict	Cascade

Table 3: 資料表 PRODUCT 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
Transaction_id	交易紀錄代號	bigint	PK	Not Null, Unique	
Request_id	對應請求代號	bigint	FK	Not Null	TRADE_REQUEST(Request_id)
Target_product_id	目標商品代號	bigint	FK	Not Null	PRODUCT(Product_id)
Offered_product_id	提供商品代號	bigint	FK	Nullable	PRODUCT(Product_id)；若為購買則為 NULL
Total_price	交易金額	int		Nullable	僅購買類型使用
Complete_date	完成日期	datetime		Not Null	
Payment_status	付款狀態	varchar(10)		Default 'Unpaid'	`{Paid, Unpaid, NA}`
Referential triggers				**On Delete**	**On Update**
Request_id: TRADE_REQUEST(Request_id)				Cascade	Cascade
Target_product_id: PRODUCT(Product_id)				Cascade	Cascade
Offered_product_id: PRODUCT(Product_id)				Set Null	Cascade

Table 4: 資料表 TRANSACTION 的欄位資訊（記錄完成的交易或購買）

Column Name	Meaning	Data Type	Key	Constraint	Domain
Wish_id	許願代號	bigint	PK	Not Null, Unique	
User_id	使用者代號	bigint	FK	Not Null	USER(User_id)
Offered_product_id	提供交換商品代號	bigint	FK	Not Null	PRODUCT(Product_id)；屬於該 User
Image_url	交換商品圖片網址	varchar(255)			
Desired_text	欲交換之物品描述	varchar(100)		Not Null	文字敘述，如「鉛筆」或「遊戲機」
Desired_category_id	欲交換物品分類代號	int	FK	Nullable	CATEGORY(Category_id)
Status	許願狀態	varchar(15)		Default 'Active'	{Active, Matched, Cancelled}
Created_at	建立時間	datetime		Not Null	
Updated_at	更新時間	datetime		Not Null	
Referential triggers				**On Delete**	**On Update**
User_id: USER(User_id)				Cascade	Cascade
Offered_product_id: PRODUCT(Product_id)				Cascade	Cascade
Desired_category_id: CATEGORY(Category_id)				Set Null	Cascade

Table 5: 資料表 TRADE_WISH 的欄位資訊（使用者開放式徵求交換）

Column Name	Meaning	Data Type	Key	Constraint	Domain
Request_id	請求代號	bigint	PK	Not Null, Unique	
Requester_id	提案者代號	bigint	FK	Not Null	USER(User_id)
Target_product_id	目標商品代號	bigint	FK	Not Null	PRODUCT(Product_id)；trade_option='trade'或'both'
Offered_product_id	提供交換之商品代號	bigint	FK	Nullable	PRODUCT(Product_id)；僅當 Request_type='Trade'時有值
Request_type	請求類型	varchar(10)		Not Null	{Purchase, Trade}
Offer_price	出價金額	int		Nullable	僅當 Request_type='Purchase'時使用
Status	請求狀態	varchar(15)		Default 'Pending'	{Pending, Accepted, Rejected, Completed, Cancelled}
Message	提案訊息	varchar(255)		Nullable	可選文字補充說明
Created_at	建立時間	datetime		Not Null	
Updated_at	更新時間	datetime		Not Null	
Referential triggers				**On Delete**	**On Update**
Requester_id: USER(User_id)				Cascade	Cascade
Target_product_id: PRODUCT(Product_id)				Cascade	Cascade
Offered_product_id: PRODUCT(Product_id)				Set Null	Cascade

Table 6: 資料表 TRADE_REQUEST 的欄位資訊（使用者之間的交易／購買提案）

Column Name	Meaning	Data Type	Key	Constraint	Domain
Admin_id	管理者代號	bigint	PK	Not Null	
User_id	使用者代號	bigint	FK	Not Null	USER(User_id)
Role	權限分類	varchar(20)		Not Null	{PlatformAdmin, ClubAdmin}
Referential triggers				**On Delete**	**On Update**
User_id: USER(User_id)				Cascade	Cascade

Table 7: 資料表 ADMIN 的欄位資訊（平台管理權限）

2.4 正規化分析（Normalization Analysis）

在設計本系統的關聯式資料庫時，我們檢視整體綱目（database schema）是否符合正規化（Normalization）的設計原則。以下依序從第一正規式（1NF）至第四正規式（4NF）說明本系統之關聯如何滿足相關規則，並舉例說明轉換過程。

第一正規式（1NF） 若關聯中的所有屬性皆為「單一且不可再分」的基本值（simple and single-valued），則該關聯滿足 1NF。在本系統中，所有資料表的欄位皆為原子屬性（atomic attributes），不存在複合屬性（composite attributes）或多值屬性（multi-valued attributes）。

具體例子：例如，使用者的多角色（Admin ／ User）已以獨立的 ADMIN 資料表描述，而商品的多分類、交易狀態或交換項目也以單一屬性欄位呈現。若將使用者的多個角色儲存在同一欄位中（如"Admin,User"），將違反 1NF，因此我們將其獨立成 ADMIN 表。

第二正規式（2NF） 若關聯已符合 1NF，且所有非鍵屬性（non-prime attributes）皆完全功能相依（fully functional dependency）於候選鍵（candidate key），則滿足 2NF。本系統的各關聯中，主鍵多為單一欄位（如 User_id、Product_id、Request_id 等），因此所有非鍵屬性均完全依賴於該主鍵，並無部分依賴情形。

具體例子：在 PRODUCT 表中，所有屬性（Product_name、Price、Condition 等）都完全依賴於主鍵 Product_id，不存在部分依賴的情況。

第三正規式（3NF） 若關聯已符合 2NF，且所有非鍵屬性皆不會遞移相依（transitive dependency）於主鍵，則該關聯滿足 3NF。在本系統設計中，所有非鍵屬性均直接依賴其主鍵，不存在「屬性 → 屬性」的間接依賴關係。

具體例子：例如，Product_name 與 Category_id 皆直接依賴於 Product_id，而不互相依賴；User_name 與 Email 也直接依賴 User_id。若將商品分類名稱直接儲存於 PRODUCT 表中，當分類名稱需要更新時，必須更新所有相關商品，將違反 3NF，因此我們透過 Category_id 外鍵參考獨立的 CATEGORY 表。

BCNF（Boyce–Codd Normal Form） BCNF 的條件比 3NF 更嚴謹：若關聯中所有的函數相依（functional dependency）都滿足「左側屬性集合為超級鍵（superkey）」的條件，則該關聯滿足 BCNF。經檢視各關聯的主鍵與外鍵依賴關係，所有函數相依關係之決定因素皆為該關聯的超級鍵，因此本系統亦符合 BCNF。

第四正規式（4NF） 若一個關聯不存在多值相依（multi-valued dependency），則滿足第四正規式。由於本系統所有多值特性（例如使用者的多角色、商品的多分類）皆已獨立成不同關聯（如 ADMIN 表或分類表 CATEGORY），各表中亦無重複或多值集合存在，因此可確定所有關聯皆符合 4NF。