

資料庫管理 期末專案計劃書

B11705044 江彥宏、B11705051 陳奕廷、B11705058 廖振翔

2024 年 10 月 29 日

1 系統分析

在當今科技化的時代，傳統的售票相較於線上售票有著許多限制，例如要到現場買票、買票時間受到限制、買票時的排隊因素等等，這些都是會影響顧客感受的因素，因此我們想要製作一個售票系統可以有效的解決上面遇到的困境。

這個系統分為兩個身份，分別是 User 和 Admin。一般的顧客會作為 User 的身份登入系統，User 可以做的事就是在系統上瀏覽自己有興趣的活動並且購買該場活動的票券，在瀏覽活動時使用者可以篩選自己有興趣的活動類別，並且從中挑選自己有興趣的活動，系統會列出所有符合使用者篩選過後的結果，並且列出各種活動的細節，例如活動名稱、場次、時間、主辦單位等等。若使用者對某個活動有興趣的話可以購買該場活動的票券用來參加活動，他可以選擇購買的活動、時間、場次等等。系統的經營者則是使用 Admin 的身份登入系統，主要是負責管理、新增、修改各項活動的資訊，並且可以查詢使用者的資訊以及他們的購票紀錄。

1.1 系統功能

1.1.1 關於售票系統的相關設定

售票系統上會顯示最近有哪些活動開放售票以及未來幾週會有什麼活動開放售票，使用者可以瀏覽這些活動並且查看活動的各種細節，使用者只能購買已開放售票活動的票券。當使用者在購買票券的時候他必須選擇活動名稱、場次、票種等等，當他選取完所有票券的細節時他還要付款完成才算完成訂票。

若使用者想要取消訂票的時候，他可以到系統上選擇他想取消的票券，他只需要按下取消即可，後續的操作系統會自動處理，包含退款及回收票券等等。

1.1.2 給 User 的功能

在本系統中，User 可以執行以下功能：

1. 查詢活動細節：使用者可以查看想要觀賞的表演活動的細節，例如：活動名稱、舉辦時間、舉辦地點等等。
2. 購買票券：使用者可以購買他想要參加的活動的票券，包含選取他想參加的活動、選擇票價、選擇場次等等。
3. 取消購買票券：若使用者因為某些因素不想參加他已購票的活動時，可以取消購買。
4. 付款：使用者可以直接在
5. 查看及編輯使用者資訊：使用者可以查看以及編輯自己的個人資訊，包含姓名、手機號碼、電子郵件、住址等等。
6. 查看購買紀錄：使用者可以查看自己過往的購票紀錄。

1.1.3 給 Admin 的功能

在本系統中，Admin 可以執行以下功能：

1. 發行票券：管理者可以幫活動新增不同區段的票價，標示各種票價的類型、金額。
2. 新增活動：管理者可以新增活動，包含活動的各種細節，例如：活動名稱、主辦方資訊、場地、活動類別等。
3. 查詢使用者資訊：管理者可以查詢使用者的個人資訊，例如姓名、電話、電子郵件、住址。
4. 查詢使用者購買紀錄：管理者可以查看使用者的購買紀錄，包含購票時間、金額等。

2 系統設計

2.1 ER Diagram

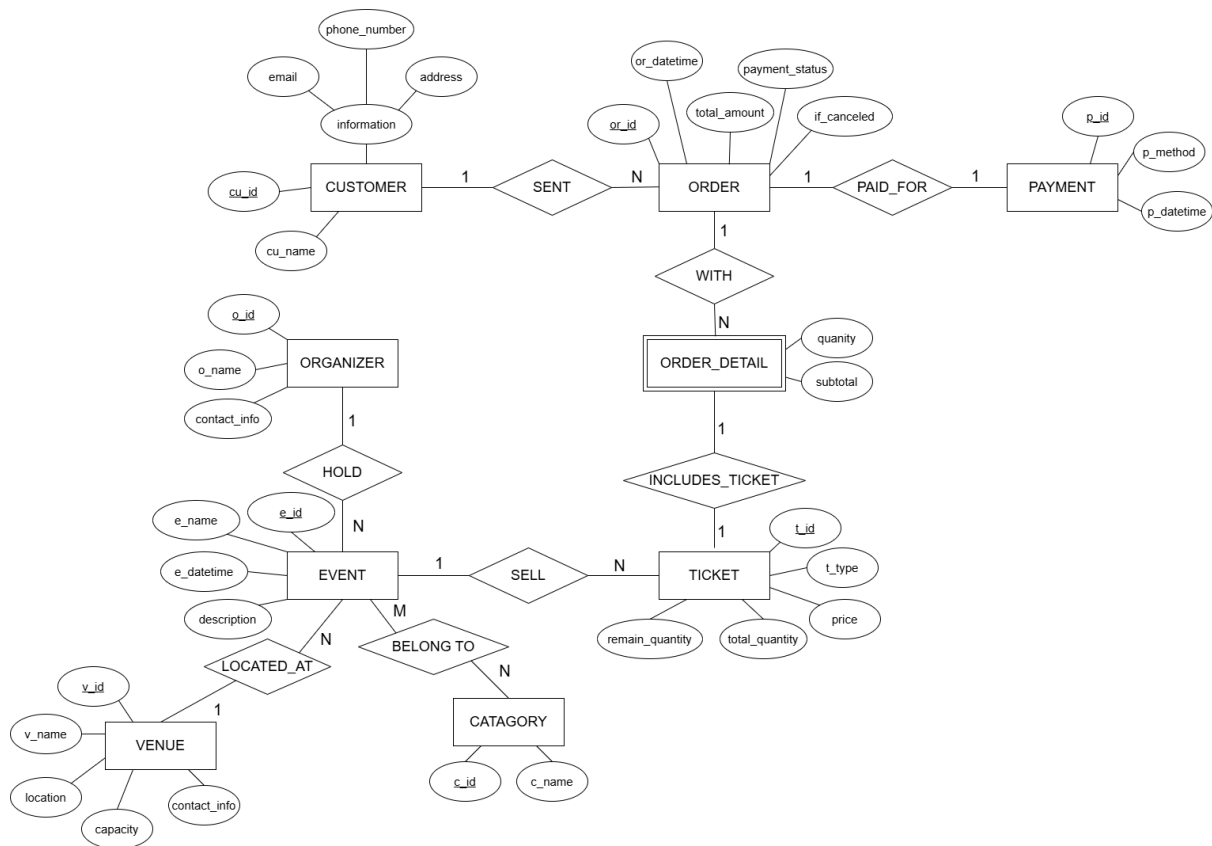


圖 1: ER Diagram

圖 1 是本售票系統的 ER Diagram，在這個 ERD 中共有九個實體 (entity)，分別是 CUSTOMER、ORDER、ORDER_DETAIL、PAYMENT、TICKET、EVENT、ORGANIZER、VENUE、CATAGORY。此外，系統中存在多個關係 (relationship)，包括 SENT、PAID_FOR、WITH、INCLUDES_TICKET、SELL、LOCATED_AT、HOLD 以及 BELONG_TO。

以下為實體 (Entities) 的說明：

1. CUSTOMER

- *cu_id*：顧客代號 (PK)
- *cu_name*：顧客名稱
- *email*：電子郵件
- *phone_number*：電話號碼

- address：地址

2. ORDER

- or_id：訂單代號（PK）
- or_datetime：訂單日期時間
- total_amount：總金額
- payment_status：支付狀態
- if_canceled：是否取消訂單

3. ORDER_DETAIL(Weak Entity 依賴於 ORDER)

- quantity：數量
- subtotal：小計

4. PAYMENT

- p_id：支付代號（PK）
- payment_method：支付方式
- p_datetime：支付日期時間
- amount：金額

5. TICKET

- t_id：票種代號（PK）
- t_type：票種類型
- price：價格
- total_quantity：總票數
- remaining_quantity：剩餘票數

6. EVENT

- e_id：活動代號（PK）
- e_name：活動名稱
- e_datetime：活動日期時間
- description：活動描述

7. ORGANIZER

- o_id：主辦方代號（PK）
- o_name：主辦方名稱

- contact_info：聯絡資訊

8. VENUE

- v_id：場地代號 (PK)
- v_name：場地名稱
- location：地址
- capacity：場地容納人數
- contact_info：聯絡資訊

9. CATEGORY

- c_id：類別代號 (PK)
- c_name：類別名稱

以下為關係 (Relationships) 的說明：

1. SENT (CUSTOMER 與 ORDER)

- 類型：一對多 (1:N)
- 描述：一個顧客可以下多個訂單，一個訂單只能由一個顧客下達。

2. PAID_FOR (ORDER 與 PAYMENT)

- 類型：一對一 (1:1)
- 描述：一個訂單對應一個支付記錄。

3. WITH (ORDER 與 ORDER_DETAIL)

- 類型：一對多 (1:N)
- 描述：一個訂單包含多個訂單詳情，一個訂單詳情只能屬於一個訂單。

4. INCLUDES_TICKET (ORDER_DETAIL 與 TICKET)

- 類型：多對一 (1:1)
- 描述：每個訂單詳情只有同一種票種

5. SELL (EVENT 與 TICKET)

- 類型：一對多 (1:N)
- 描述：一個活動包含多種票種，一種票種只能屬於一個活動。

6. LOCATED_AT (EVENT 與 VENUE)

- 類型：多對多 (M:N) (透過 EV 實體實現)
- 描述：一個活動可以在多個場地舉行，一個場地可以承辦多個活動。

7. HOLD (ORGANIZER 與 EVENT)

- 類型：一對多 (1:N)
- 描述：一個承辦商可以承辦多個活動，每個活動只能被一個承辦商承辦。

8. BELONG_TO (CATEGORY 與 EVENT)

- 類型：一對多 (1:N)
- 描述：一個類別包含多個活動，一個活動屬於一個類別。

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 CUSTOMER、ORDER、ORDER_DETAIL、PAYMENT、TICKET、ORGANIZER、EVENT、CATEGORY、VENUE、EVENT_VENUE。

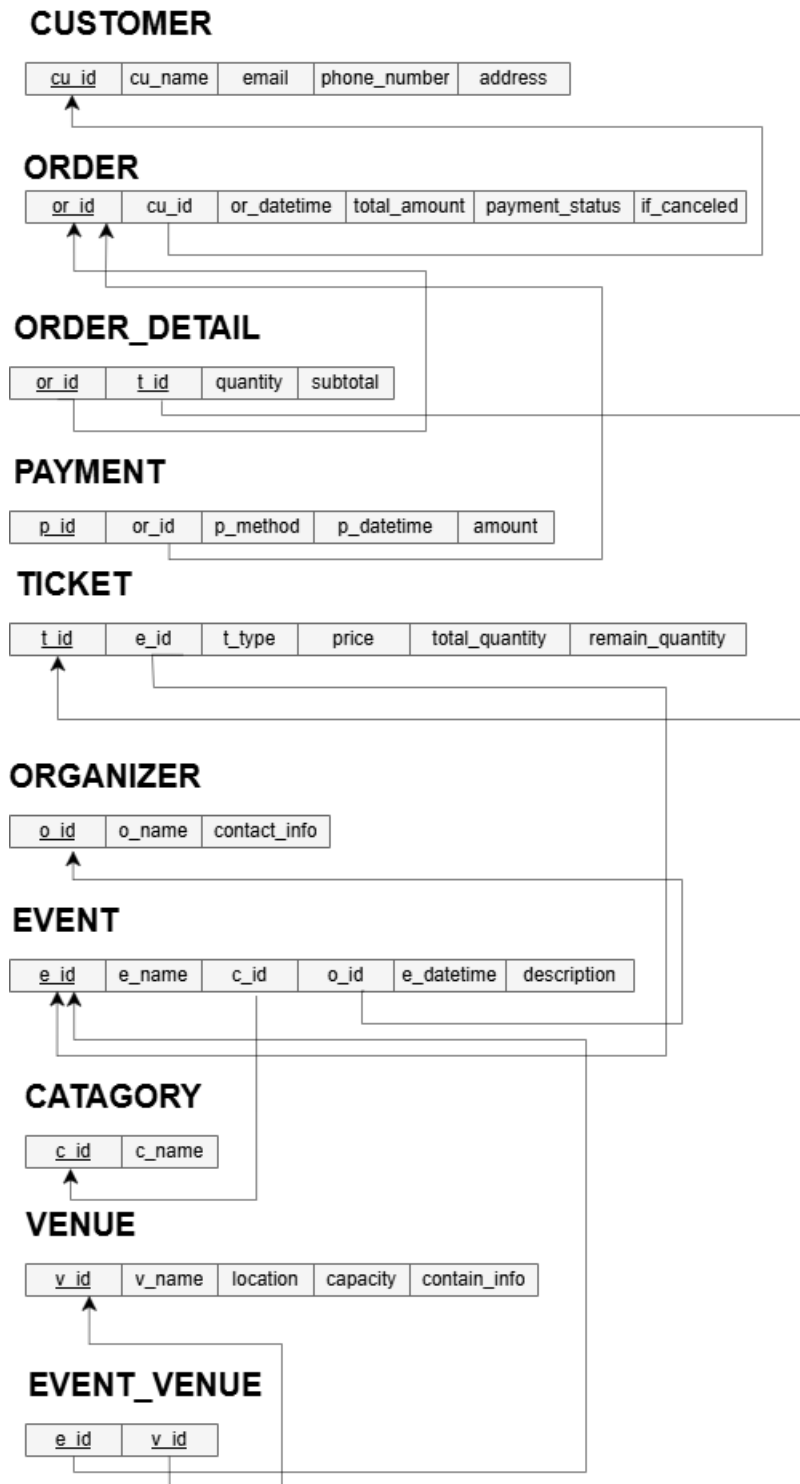


圖 2: Relational Database Schema Diagram

CUSTOMER 這個關聯的主鍵是 cu_id。為了避免 ORDER 中，quantity 和 subtotal 造成查詢上的 Redundancy，因此我們將它分為兩個不同的關聯，以 o_id (order id) 和 t_id (ticket id) 當作 foreign key，參考到 ORDER 跟 TICKET 這兩個關聯當中。

VENUE 的主鍵是 `v_id`，EVENT 的主鍵是 `e_id`。EVENT_VENUE 則是 LOCATED_AT 的多對多關係產生的，EVENT_VENUE 的主鍵由兩個外部鍵組合而成，分別參考了 EVENT 的 `e_id` 和 VENUE 的 `v_id`。

2.3 Data Dictionary

「售票系統」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
<code>c_id</code>	類別 ID	INT	PK	Not Null, Unique	整數
<code>c_name</code>	類別名稱	VARCHAR(15)		Not Null	字串 (名稱)

表 1: 資料表 CATEGORY 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
<code>e_id</code>	活動 ID	INT	PK	Not Null, Unique	整數
<code>e_name</code>	活動名稱	VARCHAR(15)		Not Null	字串 (名稱)
<code>c_id</code>	類別 ID	INT	FK: Category(<code>c_id</code>)	Not Null	整數
<code>o_id</code>	主辦方 ID	INT	FK: Organizer(<code>o_id</code>)	Not Null	整數
<code>e_datetime</code>	活動日期及時間	DATETIME		Not Null	日期時間格式
<code>e_location</code>	活動地點	VARCHAR(15)		Not Null	字串 (地點)
<code>description</code>	活動描述	TEXT			字串 (描述)
Referential triggers		On Delete	On Update		
<code>c_id: Category(<code>c_id</code>)</code>		Cascade	Cascade		
<code>o_id: Organizer(<code>o_id</code>)</code>		Cascade	Cascade		

表 2: 資料表 EVENT 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
<code>o_id</code>	主辦方 ID	INT	PK	Not Null, Unique	整數
<code>o_name</code>	主辦方名稱	VARCHAR(15)		Not Null	字串 (名稱)
<code>contact_info</code>	主辦方聯絡資訊	VARCHAR(15)			字串 (聯絡資訊)

表 3: 資料表 ORGANIZER 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
t_id	票 ID	INT	PK	Not Null, Unique	整數
e_id	活動 ID	INT	FK: Event(e_id)	Not Null	整數
t_type	票種	VARCHAR(10)			字串 (票種)
price	票價	DECIMAL(10,2)		Not Null	正數
total_quantity	總票數	INT		Not Null	正整數
remain_quantity	剩餘票數	INT		Not Null	正整數
Referential triggers		On Delete	On Update		
e_id: Event(e_id)		Cascade	Cascade		

表 4: 資料表 TICKET 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
cu_id	客戶 ID	INT	PK	Not Null, Unique	整數
cu_name	客戶名稱	VARCHAR(15)		Not Null	字串 (名稱)
email	客戶電子郵件	VARCHAR(15)			字串 (電子郵件)
phone_number	客戶電話	VARCHAR(15)			字串 (電話號碼)
address	客戶地址	TEXT			字串 (地址)

表 5: 資料表 CUSTOMER 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
or_id	訂單 ID	INT	PK	Not Null, Unique	整數
cu_id	客戶 ID	INT	FK: Customer(cu_id)	Not Null	整數
or_date	訂單日期	DATE		Not Null	日期格式
total_amount	總訂單金額	DECIMAL(10,2)			正數
payment_status	付款狀態	VARCHAR(10)			字串 (狀態)
is_canceled	是否取消訂單	BOOLEAN		NOT NULL DEFAULT FALSE	TRUE 或 FALSE
Referential triggers		On Delete	On Update		
cu_id: Customer(cu_id)		Cascade	Cascade		

表 6: 資料表 ORDER 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
or_id	訂單 ID	INT	FK: Order(or_id)	Not Null	整數
t_id	票 ID	INT	FK: Ticket(t_id)	Not Null	整數
quantity	票數	INT		Not Null	正整數
subtotal	小計	DECIMAL(10,2)		Not Null	正數
Referential triggers		On Delete	On Update		
or_id: Order(or_id)		Cascade	Cascade		
t_id: Ticket(t_id)		Cascade	Cascade		

表 7: 資料表 ORDER_DETAIL 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
p_id	付款 ID	INT	PK	Not Null, Unique	整數
or_id	訂單 ID	INT	FK: Order(or_id)	Not Null	整數
payment_method	付款方式	VARCHAR(10)		Not Null	字串 (支付方式)
payment_datetime	付款日期及時間	DATETIME		Not Null	日期時間格式
amount	付款金額	DECIMAL(10,2)		Not Null	正數
Referential triggers		On Delete	On Update		
or_id: Order(or_id)		Cascade	Cascade		

表 8: 資料表 PAYMENT 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
v_id	場地 ID	INT	PK	Not Null, Unique	整數
v_name	場地名稱	VARCHAR(15)		Not Null	字串 (名稱)
address	場地地址	TEXT			字串 (地址)
capacity	容量	INT		Not Null	正整數
contact_info	場地聯絡資訊	VARCHAR(15)			字串 (聯絡資訊)

表 9: 資料表 VENUE 的欄位資訊

Column Name	Meaning	Data Type	Key	Constraint	Domain
e_id	活動 ID	INT	FK: Event(e_id)	Not Null	整數
v_id	場地 ID	INT	FK: Venue(v_id)	Not Null	整數
arrangement	場地配置	TEXT			字串 (配置)
Referential triggers		On Delete	On Update		
e_id: Event(e_id)		Cascade	Cascade		
v_id: Venue(v_id)		Cascade	Cascade		

表 10: 資料表 EVENT_VENUE 的欄位資訊

2.4 正規化分析

以下是針對資料庫設計是否符合 1NF、2NF 和 3NF 的說明：

- **1NF** 要求所有欄位的值都不能是 multi-valued，即每個欄位的值必須是不可再分的單一值，且不能包含重複組合或子集合。在此資料字典中，每張表的欄位均為單一值，因此符合 1NF 的要求。
- **2NF** 要求所有非主鍵欄位完全依賴於主鍵，且不存在部分依賴。在此資料庫設計中，每張表的非主鍵欄位（例如 EVENT 表中的 e_name 和 description）皆完全依賴其主鍵（如 e_id），符合 2NF 的要求。

- **3NF** 要求每個非主鍵欄位必須直接依賴於主鍵，且不能依賴其他非主鍵欄位。在這個設計中，幾乎所有非主鍵欄位都直接依賴於主鍵，並且不存在傳遞依賴的情況，因此大部分符合 3NF 的要求。然而在 TICKET 表中，t_id 和 e_id 與 t_type 之間可能存在傳遞依賴 (transitively depends)。由於 Ticket 表主要紀錄票的狀態，而每個 EVENT 的 t_type 也不多，選擇不拆開也不太會影響資料的冗餘性，而拆解可能會降低資料庫的可讀性，因此選擇保留現有設計。