國立虎尾科技大學資訊工程系

專題製作報告

電商平台資料庫設計

參與成員：

41043218陳荔群

41043220曾聖傑

41043255蘇于驊

41048110梁詠琳

指導教授：江季翰

中華民國 114 年 6 月 15 日

[**一、應用情境與使用案例（Use Case） 4**](#_1ehnrlxvidbm)

[應用情境說明 4](#_t8t5jaeb6fyt)

[使用者角色 4](#_iamr18bvbxyc)

[主要使用案例 Use Cases 4](#_c7yl5wze2em1)

[**二、系統需求說明 5**](#_x6bwr84a83wk)

[2.1 Customer（買家） 5](#_wh8d13dojovh)

[2.2 Seller（賣家） 6](#_ct3g3lsx9am)

[2.3 Admin（管理員） 7](#_l54hx836af8m)

[2.4 Product（商品） 8](#_5oa59vk77dxv)

[2.5 Order（訂單） 9](#_1j0xlb60itd3)

[2.6 OrderItem（訂單項目） 10](#_wekess2ubtyt)

[2.7 Review（商品評價） 11](#_8v2311erhtdw)

[**三、完整性限制（Integrity Constraints） 12**](#_2o8m33ca9238)

[**四、ER Diagram（實體關係圖） 14**](#_6in11bku8ekv)

[**五、ER Diagram 詳細說明（實體與關係） 15**](#_bslsqr6zhn09)

[5.1 Customer（買家） 15](#_4j3tbhevj4ga)

[5.2 Seller（賣家） 17](#_l17wfdfpnnug)

[5.3 Admin（管理員） 19](#_usayo2yxb74u)

[5.4 Product（商品） 20](#_vnp2icmaxjwh)

[5.5 Order（訂單） 22](#_lz5b2r8axgup)

[5.6 OrderItem（訂單項目） 24](#_l7bbudcai9e3)

[5.7 Review（商品評價） 26](#_hva4b1i1rm88)

[**六、完整DB Schema與說明 28**](#_wymi2awd8ulg)

[6.1 Customer（買家） 29](#_1tpk6hpzwpe)

[6.2 Seller（賣家） 31](#_1xfe7sb6i086)

[6.3 Admin（管理員） 33](#_kg77c8oegjml)

[6.4 Product（商品） 34](#_2olnre7zcla8)

[6.5 Order（訂單） 36](#_l6hd02h9mwog)

[6.6 OrderItem（訂單項目） 38](#_fx3lyakwvl9)

[6.7 Review（商品評價） 39](#_n2jp32cywfo2)

[**七、SQL語法涵概念層與View 41**](#_vey9lje4f0ct)

[7.1 CustomerPublicInfo View 41](#_y6cilc175dvd)

[7.2 SellerStats View 42](#_7q1wrq3ddq3y)

[7.3 HighPricedProducts View 43](#_jpvxof2zt7lu)

[7.4 OrderSummary View 44](#_csevsyc3hyhy)

[7.5 OrderItemDetails View 45](#_gj66ior5gehu)

[7.6 ProductReviewStats View 46](#_10mn0zfh3jw)

[7.7 LatestProductReview View 47](#_wbmecibp92m3)

[**八、系統實作成果 48**](#_vmpe63eiqlwm)

[8.1 資料表建立成功 48](#_74tuq5awq6ea)

[8.2 完整性限制應用成功 49](#_cp15bpwoc96h)

[8.3 範例資料成功插入 50](#_t8e8hu3nnizm)

[**九、團隊分工 55**](#_qsmeu1hkxn4t)

[**十、同學心得 56**](#_4p28tbyx92dz)

[**十一、參考資料 60**](#_3w7olp213uhg)

## 

## 一、應用情境與使用案例（Use Case）

### 應用情境說明

本系統模擬一個電商平台（如蝦皮），讓賣家能夠上架商品、管理訂單，買家能夠註冊帳號、瀏覽商品、下訂單與付款，並提供評價功能。系統亦包含後台管理員可監控整體平台狀況。

### 使用者角色

* 買家（Customer）
* 賣家（Seller）
* 管理員（Admin）

### 主要使用案例 Use Cases

| 使用者角色 | 使用案例 |
| --- | --- |
| 買家 | 註冊/登入、瀏覽商品、加入購物車、下單、付款、查看訂單、評價商品 |
| 賣家 | 註冊/登入、上架商品、修改商品資訊、查看訂單、出貨、查看評論 |
| 管理員 | 查看平台交易記錄、管理用戶帳號、下架不當商品 |

## 二、系統需求說明

### 2.1 Customer（買家）

| 欄位名稱 | 完整性限制說明 |
| --- | --- |
| customer\_id | 主鍵，唯一且不可為空。系統自動生成或由系統驗證唯一性。 |
| name | 僅可包含英文大小寫與數字（A-Z, a-z, 0-9），不得有空格或特殊字元，長度 3–20 字元。正規表示式：^[A-Za-z0-9]{3,20}$。 |
| email | 唯一且不可為空，需符合有效 Email 格式，例如 user@example.com，不可含空格。可使用正規表示式 ^[\w\.-]+@[\w\.-]+\.\w{2,}$ 驗證。 |
| phone\_number | 不可為空。必須為 10 位數字且開頭為 09。正規表示式：^09\d{8}$。 |
| gender | 不可為空。僅允許「男」、「女」、「其他」三種值。建議使用 ENUM 或 CHECK 限制。 |
| birthdate | 不可為空。格式須為 YYYY-MM-DD。必須為有效歷史日期，不得晚於系統當前日期。 |

### 

### 2.2 Seller（賣家）

| 欄位名稱 | 完整性限制說明 |
| --- | --- |
| seller\_id | 主鍵，唯一且不可為空。系統自動生成或由系統驗證唯一性。 |
| name | 限制同 Customer.name。 |
| email | 限制同 Customer.email，且必須在 Customer.email 中保持唯一性（跨表唯一）。 |
| phone\_number | 限制同 Customer.phone\_number。 |
| gender | 限制同 Customer.gender。 |
| birthdate | 限制同 Customer.birthdate。 |

### 

### 2.3 Admin（管理員）

| 欄位名稱 | 完整性限制說明 |
| --- | --- |
| admin\_id | 主鍵，唯一且不可為空。 |
| username | 唯一且不可為空，僅可使用英數字（A-Z, a-z, 0-9），長度建議為 4–20 字元。不得包含空白或特殊字元。正則式可為 ^[A-Za-z0-9]{4,20}$。 |

### 

### 2.4 Product（商品）

| 欄位名稱 | 完整性限制說明 |
| --- | --- |
| product\_id | 主鍵，唯一且不可為空。 |
| name | 不可為空，應具明確辨識意義，避免重複，建議使用唯一索引輔助搜尋。 |
| price | 必須為正數，> 0，允許最多兩位小數。建議使用 DECIMAL(10,2) 並加上 CHECK 條件。 |
| stock | 必須為 0 或正整數，建議使用 CHECK (stock >= 0)。 |
| category | 可為空，若填寫應屬於定義好的類別（例如 ENUM 或參照分類表），避免自由輸入造成資料混亂。 |
| seller\_id | 外鍵，參照 Seller.seller\_id，不可為空。必須對應現有賣家。違反時應拒絕寫入。 |

### 

### 2.5 Order（訂單）

| 欄位名稱 | 完整性限制說明 |
| --- | --- |
| order\_id | 主鍵，唯一且不可為空。 |
| order\_date | 不可為空，格式為 YYYY-MM-DD，不得早於系統啟用日或晚於系統當前時間。 |
| status | 不可為空，限定值：「未付款」、「處理中」、「已出貨」、「已完成」。建議使用 ENUM 或 CHECK 實作。 |
| customer\_id | 外鍵，參照 Customer.customer\_id，不可為空。 |

### 

### 2.6 OrderItem（訂單項目）

| 欄位名稱 | 完整性限制說明 |
| --- | --- |
| order\_id | 外鍵，參照 Order.order\_id，不可為空。 |
| product\_id | 外鍵，參照 Product.product\_id，不可為空。 |
| (order\_id, product\_id) | 複合主鍵，需唯一且不可為空，確保同一訂單中不會重複記錄相同商品。 |
| quantity | 必須為正整數，且不可為 0 或負值，應小於等於商品當下庫存。建議加上邏輯驗證。 |
| price\_at\_purchase | 必須為正數，紀錄下單當時的價格，不能為 0 或負數。與當前價格分開儲存以保留歷史紀錄。 |

### 

### 2.7 Review（商品評價）

| 欄位名稱 | 完整性限制說明 |
| --- | --- |
| review\_id | 主鍵，唯一且不可為空。 |
| product\_id | 外鍵，參照 Product.product\_id，不可為空。 |
| customer\_id | 外鍵，參照 Customer.customer\_id，不可為空。 |
| rating | 整數，限制為 1 至 5，使用 CHECK (rating BETWEEN 1 AND 5)。 |
| comment | 可為空，但若填寫，內容不可為空白或無意義字元。建議以正規表示式排除純空白。 |
| review\_date | 不可為空。格式為 YYYY-MM-DD，不得晚於系統當前時間，亦不得早於該筆商品訂單完成時間。 |

### 

## 三、完整性限制（Integrity Constraints）

| 類別 | 詳細說明 |
| --- | --- |
| 1. 主鍵唯一性  （Primary Key Uniqueness） | 系統中每一筆資料應有唯一識別碼：  • Customer.customer\_id、Seller.seller\_id、Admin.admin\_id、Product.product\_id、Order.order\_id、Review.review\_id 等欄位皆設為主鍵。  • 此欄位應使用系統自動產生（如 UUID、AUTO\_INCREMENT）或手動確保唯一性。  • 不可為空（NOT NULL）。  • 不得重複，否則會違反資料一致性。 |
| 2. 外鍵約束  （Foreign Key Constraints） | 確保資料之間的參照關係正確：  • Product.seller\_id 必須參照存在於 Seller.seller\_id 的資料，否則無法建立商品。  • Order.customer\_id 必須對應至現有的 Customer.customer\_id。  • OrderItem.order\_id 與 OrderItem.product\_id 必須參照 Order 與 Product 表中存在的資料。  • Review.product\_id 與 Review.customer\_id 也必須為有效外鍵。  • 外鍵違反時應拒絕新增或更新，或使用 ON DELETE CASCADE / SET NULL 做適當處理。 |
| 3. 庫存限制  （Stock Constraint） | 商品的 stock 欄位（庫存數量）必須為整數且 ≥ 0：  • 禁止輸入負數，防止出現負庫存狀況。  • 可使用 CHECK (stock >= 0) 約束實作。  • 下單時需驗證欲購買數量是否小於等於目前庫存，否則應提示「庫存不足」。  • 若商品已停售，其庫存應為 0，或透過「狀態欄位」表示停售。 |
| 4. 付款狀態驗證  （Order Status Verification） | 訂單狀態變更需依付款流程限制：  • 初始狀態為「未付款」。  • 僅當付款成功（系統內部記錄或金流 API 回傳成功）後，方可變更狀態為「處理中」。  • 後續狀態依序為「已出貨」→「已完成」，不可跳階段。  • 可使用狀態流轉邏輯（如狀態機）或資料庫觸發器加強限制。 |
| 5. 評價限制  （Review Constraint） | 使用者只能對已完成訂單中的商品進行一次評價：  • 僅當 Order.status = '已完成' 且商品為該筆訂單項目之一時，才可新增評價。  • 同一名買家對同一筆商品僅能評價一次，可使用 (customer\_id, product\_id) 組合建立唯一索引避免重複評價。  • 系統應自動比對評價商品是否曾被購買且訂單已完成，否則應提示錯誤訊息。 |
| 6. Email 唯一性（Email Uniqueness） | 所有使用者（Customer、Seller、Admin）的 Email 必須唯一，不可重複：  • 每個使用者帳號的 email 欄位設為 UNIQUE。  • 若使用者身份可重疊（如同時為賣家與買家），建議考慮統一 User 表設計並以角色欄區分，避免跨表重複。  • Email 欄位必須為有效格式，可使用正則表示式驗證：^[\w\.-]+@[\w\.-]+\.\w{2,}$。 |

## 四、ER Diagram（實體關係圖）

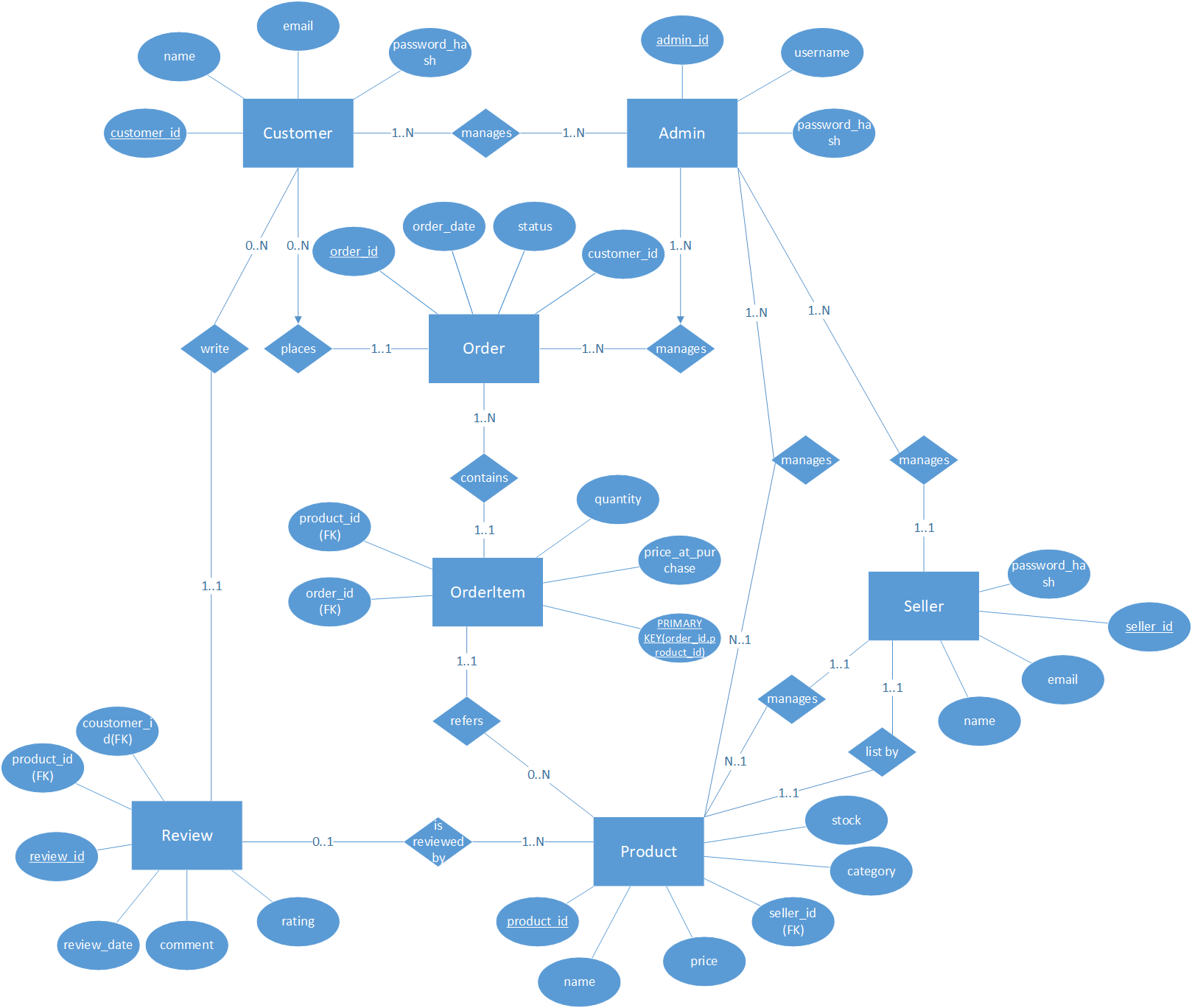


圖4.1 ER Diagram(實體關係圖)

## 

## 五、ER Diagram 詳細說明（實體與關係）

### 5.1 Customer（買家）

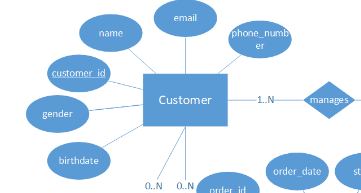


圖5.1 Customer（買家）

| 欄位名稱 | 資料型別 | 鍵類型 | 約束條件與驗證 | 是否可為空 | 說明 |
| --- | --- | --- | --- | --- | --- |
| customer\_id | CHAR(10) | 主鍵 | UNIQUE、NOT NULL | 否 | 客戶編號，系統生成或手動指定 |
| name | VARCHAR(20) |  | ^[A-Za-z0-9]{3,20}$ 格式驗證 | 否 | 僅允許英文、數字，3~20 字元 |
| email | VARCHAR(100) | 唯一 | UNIQUE, Email 格式驗證 | 否 | 使用者唯一電子郵件帳號 |
| phone\_number | CHAR(10) |  | ^09\d{8}$ 格式驗證 | 否 | 台灣手機格式（09 開頭） |
| gender | ENUM |  | 限定值：「男」、「女」、「其他」 | 否 | 可用 ENUM 或 CHECK 約束 |
| birthdate | DATE |  | 不得晚於當前系統時間 | 否 | 格式：DD/MM/YYYY |

### 

### 5.2 Seller（賣家）

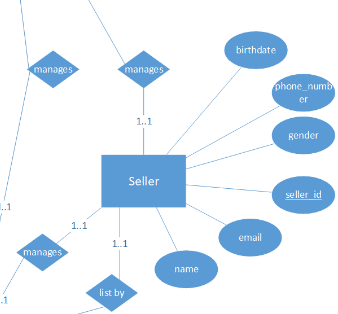


圖5.2 Seller（賣家）

| 欄位名稱 | 資料型別 | 鍵類型 | 約束條件與驗證 | 是否可為空 | 說明 |
| --- | --- | --- | --- | --- | --- |
| seller\_id | CHAR(10) | 主鍵 | UNIQUE、NOT NULL | 否 | 賣家編號 |
| name | VARCHAR(20) |  | ^[A-Za-z0-9]{3,20}$ 格式驗證 | 否 | 僅允許英文、數字 |
| email | VARCHAR(100) | 唯一 | UNIQUE, Email 格式驗證 | 否 | 賣家電子信箱 |
| phone\_number | CHAR(10) |  | ^09\d{8}$ 格式驗證 | 否 | 台灣手機格式（09 開頭） |
| gender | ENUM |  | 限定值：「男」、「女」、「其他」 | 否 | 性別欄位 |
| birthdate | DATE |  | 不得晚於當前系統時間 | 否 | 格式：DD/MM/YYYY |

### 

### 

### 5.3 Admin（管理員）

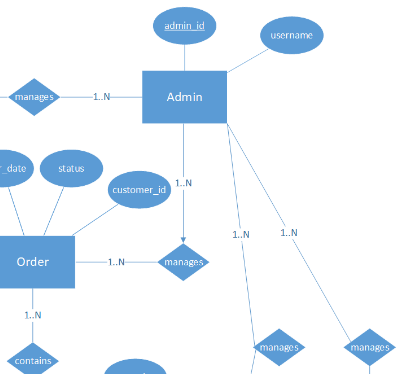


圖5.3 Admin（管理員）

| 欄位名稱 | 資料型別 | 鍵類型 | 約束條件與驗證 | 是否可為空 | 說明 |
| --- | --- | --- | --- | --- | --- |
| admin\_id | CHAR(10) | 主鍵 | UNIQUE、NOT NULL | 否 | 管理員編號 |
| username | VARCHAR(20) | 唯一 | ^[A-Za-z0-9]{3,20}$ 格式驗證 | 否 | 僅允許英文、數字，3~20 字元 |

### 

### 5.4 Product（商品）

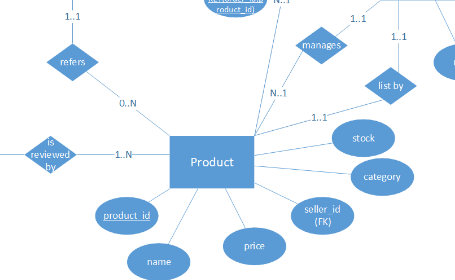


圖5.4 Product（商品）

| 欄位名稱 | 資料型別 | 鍵類型 | 約束條件與驗證 | 是否可為空 | 說明 |
| --- | --- | --- | --- | --- | --- |
| product\_id | CHAR(10) | 主鍵 | UNIQUE、NOT NULL | 否 | 商品編號 |
| name | VARCHAR(50) |  | NOT NULL | 否 | 商品名稱 |
| price | DECIMAL(10,2) |  | > 0 | 否 | 必須大於 0，可含兩位小數 |
| stock | INT |  | >= 0 | 否 | 商品庫存數量，不得為負 |
| category | VARCHAR(30) |  | 可選欄位，若填寫應屬於預先定義類別集合 | 是 | 如 electronics、books 等 |
| seller\_id | CHAR(10) | 外鍵 | FOREIGN KEY 參照 Seller.seller\_id，NOT NULL | 否 | 商品所屬賣家 |

### 

### 5.5 Order（訂單）

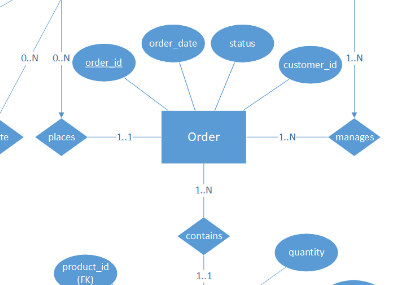


圖5.5 Order（訂單）

| 欄位名稱 | 資料型別 | 鍵類型 | 約束條件與驗證 | 是否可為空 | 說明 |
| --- | --- | --- | --- | --- | --- |
| order\_id | CHAR(10) | 主鍵 | UNIQUE、NOT NULL | 否 | 訂單編號 |
| order\_date | DATE |  | 格式為 YYYY-MM-DD，不得晚於當前系統時間，亦不得早於系統啟用日 | 否 | 訂單日期 |
| status | ENUM |  | ENUM: 未付款、處理中、已出貨、已完成 | 否 | 訂單狀態，限制於四種固定狀態 |
| customer\_id | CHAR(10) | 外鍵 | FOREIGN KEY 參照 Customer.customer\_id，NOT NULL | 否 | 所屬客戶 |

### 

### 5.6 OrderItem（訂單項目）

### 

圖5.6 OrderItem（訂單項目）

| 欄位名稱 | 資料型別 | 鍵類型 | 約束條件與驗證 | 是否可為空 | 說明 |
| --- | --- | --- | --- | --- | --- |
| order\_id | CHAR(10) | 複合主鍵 | FOREIGN KEY 參照 Order.order\_id，NOT NULL | 否 | 訂單編號 |
| product\_id | CHAR(10) | 複合主鍵 | FOREIGN KEY 參照 Product.product\_id，NOT NULL | 否 | 商品編號 |
| quantity | INT |  | >= 1，且不得超過商品目前庫存 | 否 | 購買數量 |
| price\_at\_purchase | DECIMAL(10,2) |  | > 0 | 否 | 訂單當下商品價格，固定紀錄 |

### 

### 5.7 Review（商品評價）

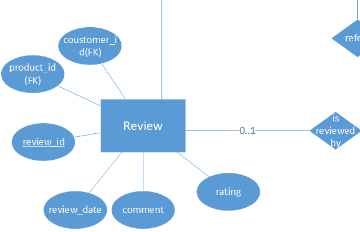


圖5.7 Review（商品評價）

| 欄位名稱 | 資料型別 | 鍵類型 | 約束條件與驗證 | 是否可為空 | 說明 |
| --- | --- | --- | --- | --- | --- |
| review\_id | CHAR(10) | 主鍵 | UNIQUE、NOT NULL | 否 | 評價編號 |
| product\_id | CHAR(10) | 外鍵 | FOREIGN KEY 參照 Product.product\_id，NOT NULL | 否 | 所評價的商品 |
| customer\_id | CHAR(10) | 外鍵 | FOREIGN KEY 參照 Customer.customer\_id，NOT NULL | 否 | 評價人（買家） |
| rating | INT |  | CHECK (rating BETWEEN 1 AND 5) | 否 | 評分範圍 1~5 |
| comment | TEXT |  | 若填寫須為有效內容，不得為純空白或無意義字元 | 是 | 可留評論，需具意義 |
| review\_date | DATE |  | 格式為 YYYY-MM-DD，且不得早於該商品之訂單完成時間，不得晚於今日 | 否 | 評價時間 |

## 

## 六、完整DB Schema與說明

圖5.1 DB Schema

### 

### 6.1 Customer（買家）

| 欄位名稱 | 資料型別 | 說明 |
| --- | --- | --- |
| customer\_id | INT (PK) | 買家主鍵，自動編號 |
| name | VARCHAR(100) | 買家姓名 |
| email | VARCHAR(255) | 電子郵件，唯一值 |
| phone | VARCHAR(20) | 電話號碼 |
| birthdate | DATETIME | 買家生日，不可為空。格式須為YYYY-MM-DD，必須為有效歷史日期，不得晚於系統當前日期。 |

SQL:

CREATE TABLE Customer (

customer\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(20) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

phone\_number VARCHAR(10) NOT NULL,

gender ENUM('男', '女', '其他') NOT NULL,

birthdate DATE NOT NULL,

CONSTRAINT chk\_name\_format CHECK (name REGEXP '^[A-Za-z0-9]{3,20}$'),

CONSTRAINT chk\_email\_format CHECK (email REGEXP '^[\\w\\.-]+@[\\w\\.-]+\\.\\w{2,}$'),

CONSTRAINT chk\_phone\_format CHECK (phone\_number REGEXP '^09\\d{8}$')

);

範例

INSERT INTO Customer (name, email, phone\_number, gender, birthdate) VALUES

('Alice123', 'alice123@example.com', '0912345678', '女', '1990-05-10'),

('BobChen01', 'bobchen01@example.com', '0987654321', '男', '1985-12-01'),

('CathyWang9', 'cathyw9@example.com', '0911122233', '其他', '1992-07-23'),

('DavidLiu88', 'davidliu88@example.com', '0922334455', '男', '1988-03-14'),

('EmmaWu77', 'emmawu77@example.com', '0933445566', '女', '1995-10-30'),

('FrankHuang', 'frankh@example.com', '0944556677', '男', '1979-08-22'),

('GraceTsai1', 'gracetsai1@example.com', '0955667788', '女', '1983-01-18'),

('HenryYeh22', 'henryyeh22@example.com', '0966778899', '男', '1991-06-05'),

('IreneKuo3', 'irenekuo3@example.com', '0977889900', '女', '1996-11-11'),

('JackyHsu44', 'jackyhsu44@example.com', '0988990011', '男', '1980-07-27');

### 

### 6.2 Seller（賣家）

| 欄位名稱 | 資料型別 | 完整性限制說明 |
| --- | --- | --- |
| seller\_id | INT (PK) | 主鍵，唯一且不可為空。系統自動生成或由系統驗證唯一性。 |
| name | VARCHAR(100) | 僅可包含英文大小寫與數字（A-Z, a-z, 0-9），不得有空格或特殊字元，長度 3–20 字元。正規表示式：^[A-Za-z0-9]{3,20}$。 |
| email | VARCHAR(255) | 唯一且不可為空，需符合有效 Email 格式，例如 user@example.com，不可含空格。可使用正規表示式 ^[\w\.-]+@[\w\.-]+\.\w{2,}$ 驗證。 |
| phone\_number | VARCHAR(20) | 不可為空。必須為 10 位數字且開頭為 09。正規表示式：^09\d{8}$。 |
| gender | ENUM('男', '女', '其他') / VARCHAR(10) | 不可為空。僅允許「男」、「女」、「其他」三種值。建議使用 ENUM 或 CHECK 限制。 |
| birthdate | DATE | 賣家生日， 不可為空。格式須為YYYY-MM-DD，必須為有效歷史日期，不得晚於系統當前日期。 |

SQL:

CREATE TABLE Seller (

seller\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(20) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

phone\_number VARCHAR(10) NOT NULL,

gender ENUM('男', '女', '其他') NOT NULL,

birthdate DATE NOT NULL,

CONSTRAINT chk\_name\_format\_seller CHECK (name REGEXP '^[A-Za-z0-9]{3,20}$'),

CONSTRAINT chk\_email\_format\_seller CHECK (email REGEXP '^[\\w\\.-]+@[\\w\\.-]+\\.\\w{2,}$'),

CONSTRAINT chk\_phone\_format\_seller CHECK (phone\_number REGEXP '^09\\d{8}$')

);

範例

INSERT INTO Seller (name, email, phone\_number, gender, birthdate) VALUES

('ShopA123', 'shopa123@example.com', '0912345678', '男', '1980-05-10'),

('ShopB456', 'shopb456@example.com', '0922334455', '女', '1985-12-01'),

('ShopC789', 'shopc789@example.com', '0933445566', '其他', '1990-07-23'),

('GadgetWorld', 'gadgetworld@example.com', '0944556677', '男', '1975-03-14'),

('BookHaven1', 'bookhaven1@example.com', '0955667788', '女', '1988-10-30'),

('FashionZone', 'fashionzone@example.com', '0966778899', '男', '1995-08-22'),

('TechHub123', 'techhub123@example.com', '0977889900', '女', '1992-01-18'),

('KitchenPro', 'kitchenpro@example.com', '0988990011', '其他', '1983-06-05'),

('ToyPlanetX', 'toyplanetx@example.com', '0911222333', '男', '1991-11-11'),

('GreenGarden', 'greengarden@example.com', '0922333444', '女', '1979-07-27');

### 6.3 Admin（管理員）

| 欄位名稱 | 資料型別 | 說明 |
| --- | --- | --- |
| admin\_id | INT (PK) | 管理員主鍵，自動編號 |
| username | VARCHAR(50) | 使用者名稱，唯一 |

SQL:

CREATE TABLE Admin (

admin\_id INT PRIMARY KEY AUTO\_INCREMENT,

username VARCHAR(20) NOT NULL UNIQUE,

CONSTRAINT chk\_username\_format\_admin CHECK (username REGEXP '^[A-Za-z0-9]{3,20}$')

);

範例

INSERT INTO Admin (username) VALUES

('admin1'),

('admin2'),

('juliaadmin'),

('superadmin'),

('systemmgr'),

('useradmin'),

('johndoe'),

('susanadmin'),

('admintest'),

('webmaster');

### 

### 6.4 Product（商品）

| 欄位名稱 | 資料型別 | 說明 |
| --- | --- | --- |
| product\_id | INT (PK) | 商品主鍵，自動編號，唯一且不可為空 |
| name | VARCHAR(50) | 商品名稱，不可為空，長度 3-50，允許中英文、數字及空格，禁止特殊符號 |
| price | DECIMAL(10,2) | 商品價格，不可為空，必須大於 0，最多兩位小數 |
| stock | INT | 商品庫存數量，不可為空，整數且不可為負 |
| category | VARCHAR(50) | 商品分類，可空，長度不超過 50，允許中英文、數字及空格 |
| seller\_id | INT (FK) | 賣家外鍵，連結 Seller.seller\_id，不可為空 |

SQL:

CREATE TABLE Product (

product\_id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50) NOT NULL CHECK (name REGEXP '^[A-Za-z0-9 ]{3,50}$'),

price DECIMAL(10, 2) NOT NULL CHECK (price > 0),

stock INT NOT NULL CHECK (stock >= 0),

category VARCHAR(50) CHECK (name REGEXP '^[A-Za-z0-9 ]{3,50}$'),

seller\_id INT NOT NULL,

FOREIGN KEY (seller\_id) REFERENCES Seller(seller\_id)

);

範例

INSERT INTO Product (name, price, stock, category, seller\_id) VALUES

('Wireless Mouse', 599.00, 100, 'Electronics', 1),

('Bluetooth Keyboard', 899.00, 80, 'Electronics', 1),

('USBC Charger', 299.00, 150, 'Accessories', 2),

('Laptop Stand', 799.00, 50, 'Office', 3),

('Notebook', 59.00, 200, 'Stationery', 5),

('TShirt', 399.00, 120, 'Clothing', 6),

('Coffee Maker', 1299.00, 30, 'Kitchen', 8),

('Action Figure', 499.00, 75, 'Toys', 9),

('Garden Shovel', 199.00, 60, 'Garden', 10),

('LED Desk Lamp', 699.00, 40, 'Lighting', 4);

### 

### 6.5 Order（訂單）

| 欄位名稱 | 資料型別 | 說明 |
| --- | --- | --- |
| order\_id | INT | 主鍵，自動編號，唯一且不可為空 |
| order\_date | DATETIME | 建立日期，不可為空，格式為 YYYY-MM-DD HH:mm:ss，預設為系統當前時間 |
| status | ENUM | 訂單狀態，不可為空，限定值：未付款、處理中、已出貨、已完成 |
| customer\_id | INT | 買家外鍵，不可為空，參照 Customer.customer\_id |

SQL:

CREATE TABLE `Order` (

order\_id INT PRIMARY KEY AUTO\_INCREMENT,

order\_date DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

status ENUM('未付款', '處理中', '已出貨', '已完成') NOT NULL,

customer\_id INT NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

範例

INSERT INTO `Order` (status, customer\_id) VALUES

('處理中', 1),

('已完成', 2),

('已出貨', 3),

('未付款', 4),

('處理中', 5),

('已完成', 6),

('已出貨', 7),

('處理中', 8),

('未付款', 9),

('處理中', 10);

### 

### 6.6 OrderItem（訂單項目）

| 欄位名稱 | 資料型別 | 說明 |
| --- | --- | --- |
| order\_id | INT (PK, FK) | 訂單編號，複合主鍵，不可為空，參照 Order.order\_id |
| product\_id | INT (PK, FK) | 商品編號，複合主鍵，不可為空，參照 Product.product\_id |
| quantity | INT | 購買數量，不可為空，必須大於 0 |
| price\_at\_purchase | DECIMAL(10,2) | 購買時的單價，不可為空，價格不得為負 |

SQL:

CREATE TABLE OrderItem (

order\_id INT,

product\_id INT,

quantity INT NOT NULL,

price\_at\_purchase DECIMAL(10, 2) NOT NULL,

PRIMARY KEY (order\_id, product\_id),

FOREIGN KEY (order\_id) REFERENCES `Order`(order\_id),

FOREIGN KEY (product\_id) REFERENCES Product(product\_id)

);

範例

INSERT INTO OrderItem (order\_id, product\_id, quantity, price\_at\_purchase) VALUES

(1, 1, 2, 599.00),

(1, 2, 1, 899.00),

(2, 3, 3, 299.00),

(3, 4, 1, 799.00),

(4, 5, 5, 59.00),

(5, 6, 2, 399.00),

(6, 7, 1, 1299.00),

(7, 8, 4, 499.00),

(8, 9, 2, 199.00),

(9, 10, 1, 699.00);

### 6.7 Review（商品評價）

| 欄位名稱 | 資料型別 | 說明 |
| --- | --- | --- |
| review\_id | INT (PK) | 評價主鍵，自動編號，不可為空 |
| product\_id | INT (FK) | 商品外鍵，不可為空，參照 Product.product\_id |
| customer\_id | INT (FK) | 買家外鍵，不可為空，參照 Customer.customer\_id |
| rating | INT | 評分，不可為空，限制為 1 至 5 之間的整數 |
| comment | TEXT | 留言內容，可為空 |
| review\_date | DATETIME | 評價時間，預設為目前時間，不可為空 |

SQL:

CREATE TABLE Review (

review\_id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT NOT NULL,

customer\_id INT NOT NULL,

rating INT CHECK (rating BETWEEN 1 AND 5),

comment TEXT,

review\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (product\_id) REFERENCES Product(product\_id),

FOREIGN KEY (customer\_id) REFERENCES Customer(customer\_id)

);

範例

INSERT INTO Review (product\_id, customer\_id, rating, comment) VALUES

(1, 1, 5, '非常好用，品質也很棒'),

(2, 2, 4, '打字手感不錯，就是有點重'),

(3, 3, 3, '充電頭普通，不過夠用'),

(4, 4, 5, '很穩定的支架，推薦'),

(5, 5, 4, '紙質不錯，適合書寫'),

(6, 6, 2, '材質一般，容易皺'),

(7, 7, 5, '煮咖啡很方便，滿意'),

(8, 8, 4, '小朋友很喜歡'),

(9, 9, 3, '有點小貴，但好用'),

(10, 10, 5, '燈光柔和，很適合夜讀');

## 七、SQL語法涵概念層與View

## 

### 7.1 CustomerPublicInfo View

顯示顧客基本資料:

CREATE VIEW CustomerPublicInfo AS

SELECT

customer\_id,

name

FROM Customer;

範例

SELECT \* FROM CustomerPublicInfo;

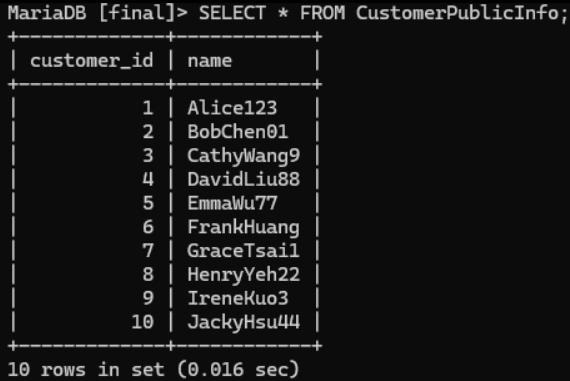
結果範例

圖7.1 買家資料表view

### 

### 7.2 SellerStats View

統計每位賣家商品數量與平均評價：

CREATE VIEW SellerStats AS

SELECT

s.seller\_id,

s.name AS seller\_name,

COUNT(p.product\_id) AS total\_products,

ROUND(AVG(r.rating), 2) AS average\_rating

FROM Seller s

LEFT JOIN Product p ON s.seller\_id = p.seller\_id

LEFT JOIN Review r ON p.product\_id = r.product\_id

GROUP BY s.seller\_id, s.name;

範例

SELECT \* FROM SellerStats WHERE average\_rating >= 4.5;

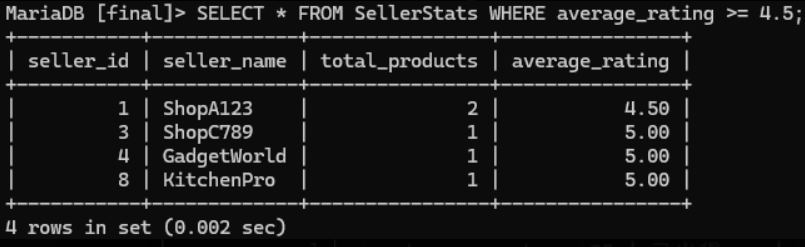
結果範例

圖7.2 賣家資料表view

### 

### 7.3 HighPricedProducts View

列出目前庫存中之商品資訊：

CREATE VIEW HighPricedProducts AS

SELECT

product\_id,

name,

price,

stock,

category,

seller\_id

FROM Product

WHERE stock > 0

ORDER BY price DESC;

範例

SELECT \* FROM HighPricedProducts LIMIT 2;

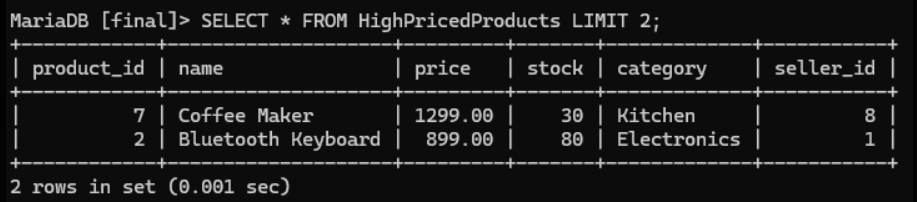
結果範例

圖7.3 商品資訊view

### 

### 7.4 OrderSummary View

呈現訂單基本資料與顧客資訊：

CREATE VIEW OrderSummary AS

SELECT

o.order\_id,

o.order\_date,

o.status,

o.customer\_id,

c.name AS customer\_name

FROM `Order` o

JOIN Customer c ON o.customer\_id = c.customer\_id;

範例

SELECT \* FROM OrderSummary WHERE status = '已出貨';

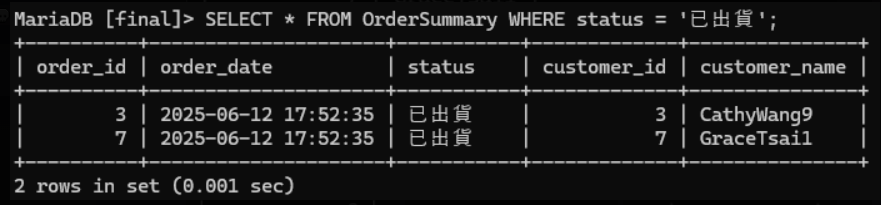
結果範例

圖7.4 訂單主檔view

### 

### 7.5 OrderItemDetails View

查詢每筆訂單中商品詳細資訊：

CREATE VIEW OrderItemDetails AS

SELECT

oi.order\_id,

oi.product\_id,

p.name AS product\_name,

oi.quantity,

oi.price\_at\_purchase

FROM OrderItem oi

JOIN Product p ON oi.product\_id = p.product\_id;

範例

SELECT \* FROM OrderItemDetails;

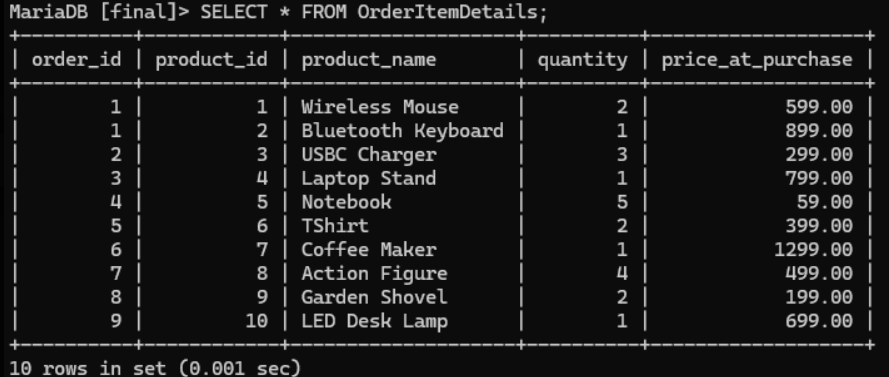
結果範例

圖7.5 訂單明細view

### 

### 7.6 ProductReviewStats View

呈現商品之平均評價與最近評價時間：

CREATE VIEW ProductReviewStats AS

SELECT

p.product\_id,

p.name AS product\_name,

ROUND(AVG(r.rating), 2) AS average\_rating,

MAX(r.review\_date) AS latest\_review\_time

FROM Product p

JOIN Review r ON p.product\_id = r.product\_id

GROUP BY p.product\_id, p.name;

範例

SELECT \* FROM ProductReviewStats;

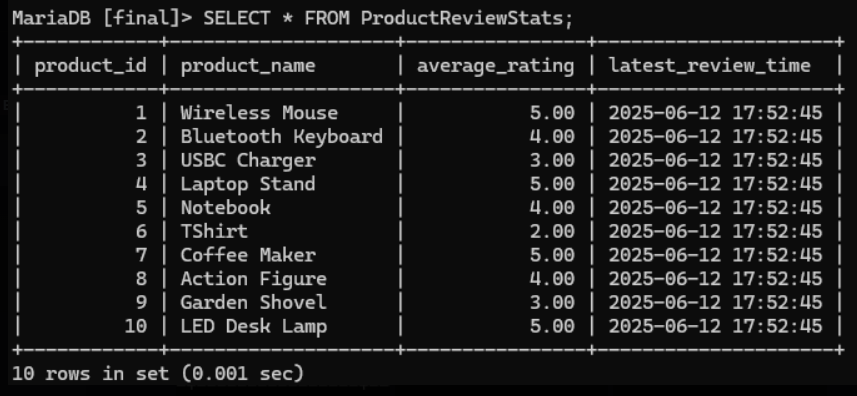
結果範例

圖7.6 商品資訊view

### 

### 7.7 LatestProductReview View

查詢每個商品的最新一筆評價：

CREATE VIEW LatestProductReview AS

SELECT r.\*

FROM Review r

JOIN (

SELECT product\_id, MAX(review\_date) AS latest\_date

FROM Review

GROUP BY product\_id

) latest ON r.product\_id = latest.product\_id AND r.review\_date = latest.latest\_date;

範例

SELECT \* FROM LatestProductReview;

結果範例

圖7.7 商品評價view

## 八、系統實作成果

本專案以 MariaDB 作為資料庫管理系統，依據設計需求成功建構完整的資料表，並導入測試資料進行驗證。實作成果如下：

### 8.1 資料表建立成功

共建置 7 張資料表，涵蓋使用者（買家、賣家、管理員）、商品、訂單、訂單項目與評價等主要資料實體。 每張表皆設定必要的 主鍵（PRIMARY KEY）、外鍵（FOREIGN KEY） 與 完整性限制（CHECK、UNIQUE、NOT NULL），以確保資料一致性與正確性。

| 資料表名稱 | 說明 |
| --- | --- |
| Customer | 買家資料表，含個人資訊、驗證條件 |
| Seller | 賣家資料表，與商品關聯 |
| Admin | 管理員帳號資料，限制名稱格式 |
| Product | 商品資訊，含價格、庫存、所屬賣家 |
| Order | 訂單主檔，與買家關聯 |
| OrderItem | 訂單明細，每筆對應訂單與商品 |
| Review | 商品評價，含評分、評論、顧客與商品資訊 |

### 8.2 完整性限制應用成功

使用正規表示式（REGEXP）、CHECK、ENUM 等方式，針對如下欄位進行驗證：

| 欄位 | 完整性限制 |
| --- | --- |
| 姓名 | 限制為英文或數字，長度 3–20 字 |
| Email | 符合 email 格式，並保證唯一 |
| 電話 | 必須為台灣手機號碼格式（如 09 開頭，共 10 碼） |
| 出生日期 | 不可為未來時間 |
| 商品價格與庫存 | 不得為負數或零 |
| 評價分數 | 限定為 1–5 分 |

詳細請看 完整性限制（Integrity Constraints）

### 

### 8.3 範例資料成功插入

每張資料表均已導入 10 筆測試資料。

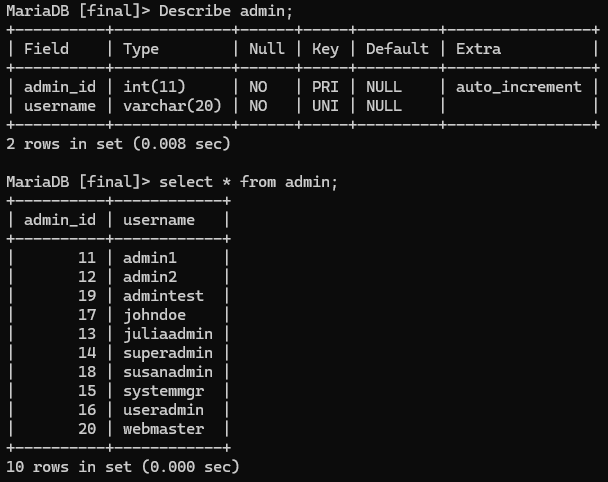


圖8.1 管理員之資料表

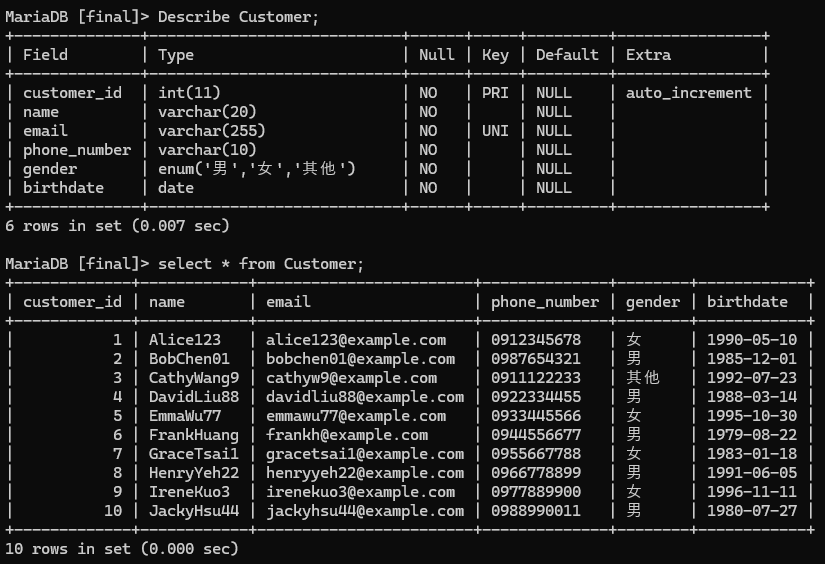
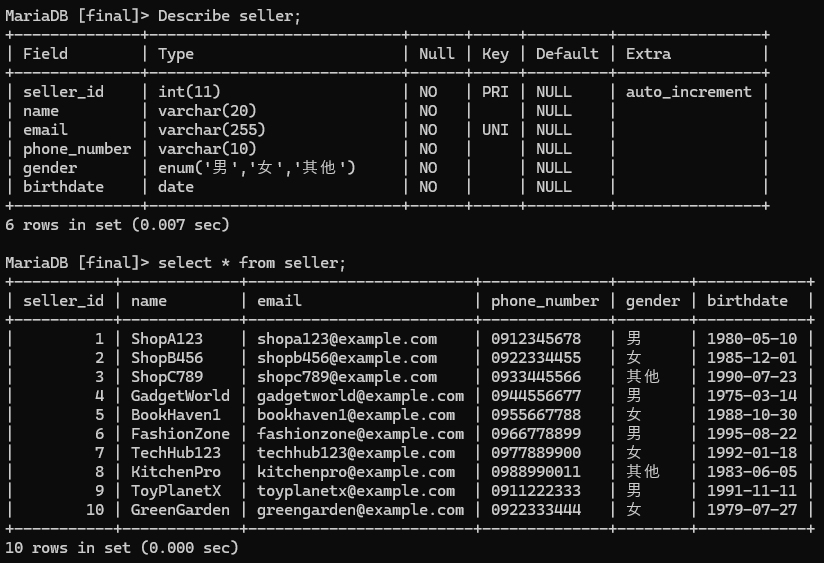
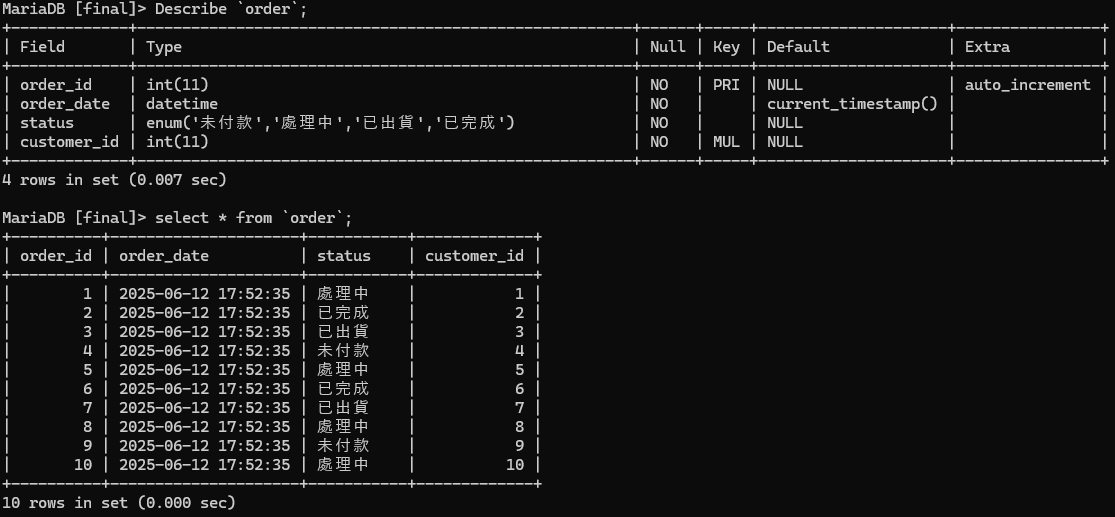
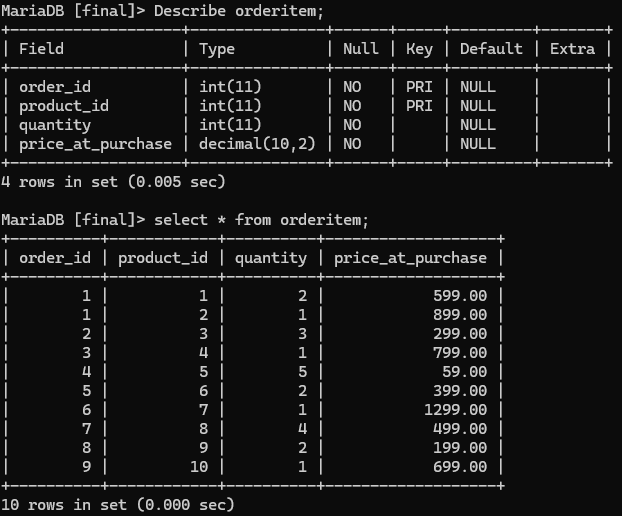


圖8.2 買家之資料表

圖8.3 賣家之資料表

圖8.4 訂單之資料表

圖8.5 訂單項目之資料表

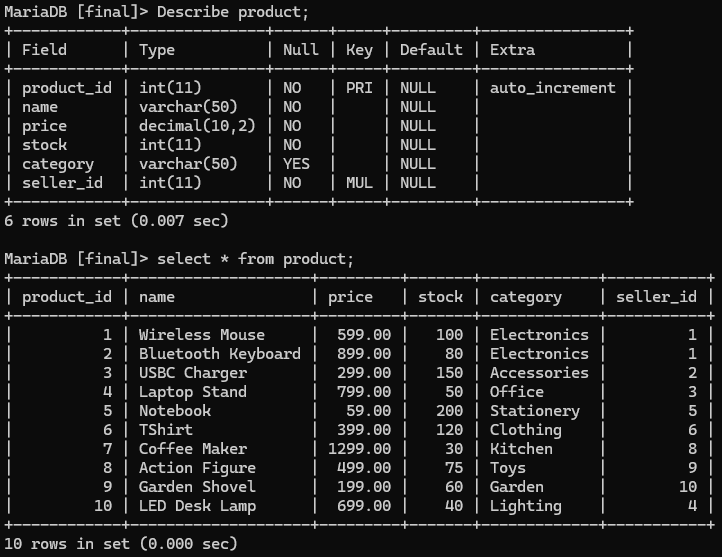


圖8.6 商品之資料表

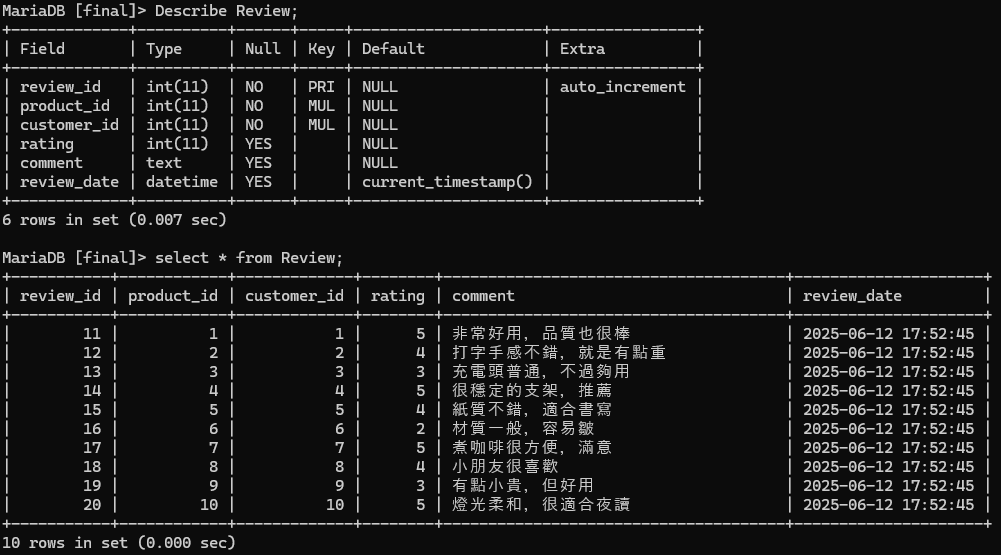


圖8.7 商品評價之資料表

## 九、團隊分工

| **職位** | **姓名** | **分工項目** |
| --- | --- | --- |
| 組長 | 曾聖傑 | 所有(舉凡Github專案管理、撰寫README.md、所有組員做的內容、資料庫實作等) |
| 組員 | 陳立群 | 自我介紹、ER Diagram繪製、view實作 |
| 組員 | 蘇于驊 | 自我介紹、DB Schema資料表、SQL、繪製 |
| 組員 | 梁詠琳 | 自我介紹、系統需求、簡報及完整報告製作 |

## 

## 

## 十、同學心得

41043218 陳荔群:

**經過這學期的報告製作，我發現如果沒有 ChatGPT 的幫助，我可能需要花費更多的時間和心力，報告的成果也可能不如現在這麼好。雖然如此，我也覺得自己對 AI 的依賴越來越深，導致我變得越來越懶得親自動手做，甚至在沒有 AI 協助的情況下，會覺得難以下手。我有種預感，自己有一天會花錢去買更高級的 AI，只希望那時的我還沒有被 AI 取代。**

**十、同學心得**

41043220曾聖傑:

**在本次專題中，我學到了很多我以前沒有學過的東西，從寫MarkDown語法到實作一個資料庫。我們以電商平台為主題，從設計 ER 圖、建立資料表到撰寫查詢語句，過程中學會了如何處理多對多關係、設計合適的完整性限制，並實際運用 SQL 語法操作資料。**

**透過這個專題，我不只更熟悉 MariaDB 的使用，也體會到資料結構設計對整體系統運作的重要性。除錯的過程，也讓我在實務經驗與技術能力上都有明顯提升。**

**十、同學心得**

41043255蘇于驊:

**在這個期末電商平台資料庫專題中，我學習到了資料庫的從無到有、從大架構到小細節，每一部分都是重要的環節、環環相扣缺一不可。**

**我這次主要負責的是DB Schema的製作，不管是資料表還是sql對我都算是很新鮮的事物，就算資料庫系統是常見於大家都在使用的app裡面，但也不容易有機會去學習到，除了前台大家所看到的，後台的資料庫也是不可或缺的，所以這是一個寶貴的學習經驗，而在跟組員討論交流的過程中也了解到資料庫各個部分所負責的作用，知道要完善一個資料庫不是一個隨便就能建構出來的。**

**完成本次的專題後更加的了解了整個資料庫的建構，如果以後有建構資料庫的需求，我也能更容易的去跟我的夥伴們架構起來，老師也在上課報告提供了我們很多沒注意到的細節，幫助了我們能更加完善我們的專題，總之，讀萬卷書不如行萬里路，讀再多的理論也比不過一次到實作。**

**十、同學心得**

41048110梁詠琳:

**在這次的電商平台資料庫設計專題裡，我主要幫忙整理資料、寫應用情境與使用案例、系統需求說明，還有初步的 ER 圖構思與修改，另外也協助報告和簡報內容的製作。**

**實作部分比較不是我負責的，不過在整理資料的過程中，還是學到了不少。 以前用網購平台的時候，從來沒有想過背後的運作其實要靠很多資料表來支撐。這次因為要寫使用案例，開始試著把使用者的行為轉成具體的資料欄位和關聯，例如「顧客只能針對已完成的訂單留下評論」，就要設計在資料表裡做限制。**

**這讓我第一次意識到：我們平常看不到的那些資料表，其實藏著平台很多邏輯和規則。 雖然我不是主要在寫 SQL 的人，但透過組員的實作，加上自己整理，也讓我對資料庫的架構、欄位設計、甚至 View 的應用有了基本認識。整體來說，是一個不錯的經驗，也讓我更有感覺資料庫這門課其實離生活蠻近的，只是以前沒注意到。**

## 十一、參考資料

[MarkDown語法大全](https://hackmd.io/@mrcoding/ryZE7k8cN#H1)

[Day 23 — 關聯式資料庫: 從設計講起](https://ithelp.ithome.com.tw/articles/10242588)