

## 資料庫專題完整報告

# 全方位電商平台智慧型資料庫系統

指導老師：江季瀚

組別：第七組

41043229 楊峻朋

41043205 謝延偵

41143207 鍾文瑤

40843141 李維誠

### 團隊分工

成員姓名	負責領域
楊峻朋	商品管理、應用情境與使用案例、系統需求說明、完整性限制、SQL 語法、完整 DB Schema 與說明、View、部分介面、報告文件撰寫(PPT)
謝延偵	1 用戶管理、應用情境與使用案例、ER Diagram 及詳細說明、SQL 語法、部分介面、報告文件撰寫(PPT、PDF)
鍾文瑤	物流與倉儲、數據分析、SQL 語法、View、報告文件撰寫(PPT)
李維誠	訂單管理、SQL 語法

# 目錄

1. 專題內容概述.....	5
2. 應用情境與使用案例.....	5
2.1 主要應用情境.....	5
2.2 使用者角色分析.....	5
2.3 平台業務流程.....	6
3. 系統需求說明.....	6
3.1 使用者角色.....	6
3.2 帳號系統.....	6
3.3 商品上架與管理.....	7
3.4 購物流程.....	7
3.5 評價系統.....	7
3.6 使用者認證與權限驗證.....	7
3.7 價格走勢分析.....	7
3.8 安全性需求.....	8
4. ER 圖設計與說明 .....	8
4.1 ER 圖架構.....	8
4.2 四大核心實體 (Entity).....	9
4.3 弱實體 (Weak Entities).....	9
4.4 獨立實體 (Independent Entities).....	9
4.5 關係描述 (Relationship Descriptions).....	9
4.6 屬性關係 (Attribute Relationships).....	10
4.7 衍生資料 (Derived Data).....	10
5. 完整性限制.....	11

5.1	主鍵限制 (Primary Key Constraint)	11
5.2	外鍵限制 (Foreign Key Constraint)	11
5.3	唯一性限制 (Unique Constraint)	11
5.4	非空限制 (NOT NULL Constraint)	11
5.5	資料型別限制 (Data Type Constraint)	12
5.6	檢查限制 (CHECK Constraint)	12
5.7	業務邏輯限制 (Business Rule Constraints)	12
6.	資料庫 Schema 設計	14
6.1	分離式安全架構設計	14
6.2	完整資料庫關聯圖	15
6.3	資料格式規範與完整性需求	16
6.4	認證伺服器模組 Schema (楊峻朋)	17
6.5	用戶管理模組 Schema (謝延偵)	18
6.6	商品管理模組 Schema (楊峻朋)	22
6.7	物流倉儲模組 Schema (鍾文瑀)	24
6.8	數據分析模組 Schema (鍾文瑀)	27
6.9	訂單管理模組 Schema (李維誠)	30
7.	系統測試資料集	33
7.1	用戶管理 (謝延偵)	33
7.2	商品管理 (楊峻朋)	38
7.3	驗證伺服器(楊峻朋)	40
7.4	倉儲管理 (鍾文瑀)	42
7.5	數據分析 (鍾文瑀)	47
7.6	訂單管理 (楊峻朋)	50

8.	權限管理與 View.....	53
8.1	角色定義.....	53
8.2	權限表.....	53
8.3	查詢範例.....	63
9.	完成介面展示.....	75
9.1	未登入時的首頁.....	76
9.2	登入介面.....	77
9.3	買家個人檔案介面.....	77
9.4	會員等級與點數介面.....	78
10.	結論與心得.....	78
10.1	專題成果總結.....	78
10.2	個人心得.....	79
11.	參考資料.....	81

# 1. 專題內容概述

用戶管理：買家 & 賣家，會員等級、積分、優惠卷、買賣家溝通

商品管理：分類、庫存、商品價格、供應商資訊、評價評論

訂單管理：訂單狀態、支付、退貨退款

物流與倉儲：庫存分配、發貨時間預測

數據分析：熱銷商品、價格走勢

## 2. 應用情境與使用案例

### 2.1 主要應用情境

張小姐利用電腦訪問網站，並註冊一個新帳號。完成信箱驗證後，登入平台並開始瀏覽首頁上的商品。之後在搜尋欄輸入「保濕乳液」，系統根據關鍵字呈現多種商品，並顯示最近的熱銷排行榜。她點選一款產品，查看詳細介紹，並將商品加入購物車。結帳時，選擇儲存的地址並使用優惠券，完成線上付款。付款完成後，平台發送通知給張小姐與通知賣家處理訂單。賣家在看到新訂單通知，進行出貨作業，而系統隨後追蹤物流狀態並持續更新張小姐的訂單進度。

### 2.2 使用者角色分析

#### 2.2.1 買家（顧客）

- 主要功能：註冊、搜尋商品、下單、付款、評價
- 核心需求：搜尋商品、比較價格、安全付款、追蹤訂單

#### 2.2.2 賣家

- 主要功能：註冊、上架商品、管理訂單、回覆評價
- 核心需求：商品上架、訂單管理、庫存控制、營收分析

### 2.2.3 內部管理人員

- 系統管理員：審核賣家、處理糾紛、管理平台內容、用戶管理、數據分析
  - 倉儲人員：管理庫存、處理出入庫、物流配送、倉庫調撥、配送追蹤
  - 財務人員：處理付款、退款、財務報表分析、會計核算、風險控制
  - 行銷人員：分析銷售數據、客戶行為、市場趨勢、促銷活動規劃
  - 客服人員：協助客戶問題、處理客訴、評價管理、買賣家溝通協調
- 2.3 系統功能需求

## 2.3 平台業務流程

- 用戶註冊與認證流程
- 商品展示與搜尋體驗
- 交易與支付流程
- 售後服務與評價機制
- 平台管理與營運

## 3. 系統需求說明

### 3.1 使用者角色

- 買家：註冊、搜尋商品、下單、付款、評價
- 賣家：註冊、上架商品、管理訂單、回覆評價
- 管理員：審核賣家、處理糾紛、管理平台內容

### 3.2 帳號系統

- 使用者可註冊帳號並驗證電子郵件/手機號碼
- 使用者可設定角色（買家/賣家）
- 支援多層級會員制度與權限管理

### 3.3 商品上架與管理

- 賣家可以新增、編輯、刪除商品
- 商品需包含：名稱、描述、圖片、價格、庫存數量
- 支援商品分類管理與標籤系統

### 3.4 購物流程

- 買家可將商品加入購物車
- 結帳需選擇配送地址與付款方式
- 系統發送訂單通知給相關人員

### 3.5 評價系統

- 購買完成後可對賣家與商品評分
- 可留文字評論與圖片
- 評價真實性驗證機制

### 3.6 使用者認證與權限驗證

- 使用者登入後，伺服器回傳 JWT Token（含 Access 與 Refresh Token）
- 所有需要登入權限的操作（如新增商品、下單）均需在 Header 中帶入 Access Token
- Token 逾時後可透過 Refresh Token 自動更新

### 3.7 價格走勢分析

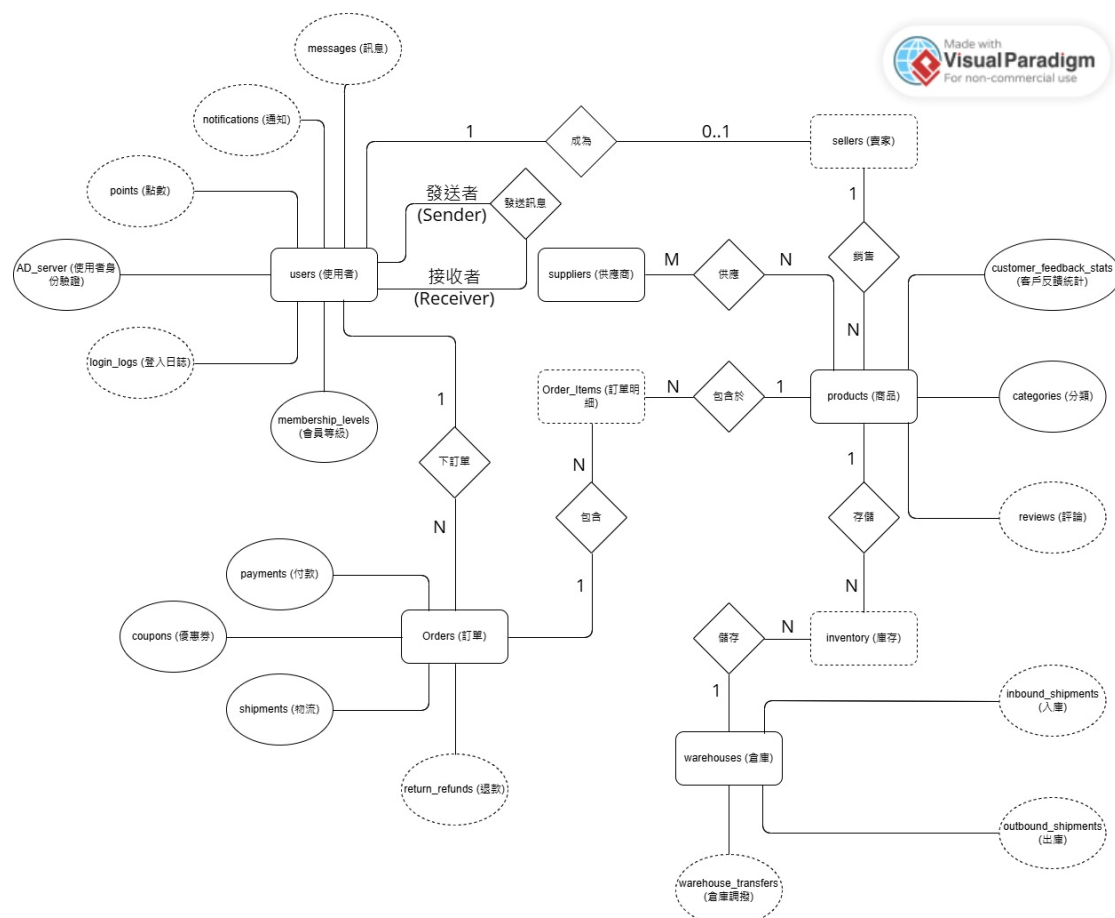
- 查看某商品歷史價格變化（可針對單一商品或類別分析）
- 支援比較同類型商品的價格趨勢

### 3.8 安全性需求

- 使用 HTTPS 通訊協定
- 密碼加密存儲（如 bcrypt）
- 防止 SQL injection、XSS 攻擊等
- JWT 認證機制：
  - Token 有效期限：Access Token 有效 3 小時
  - 防止 CSRF：Token 儲存於 HttpOnly cookie 時須實作 CSRF 保護策略
  - JWT 採 RSA 私鑰加密，公鑰對於竄改進行驗證

## 4. ER 圖設計與說明

### 4.1 ER 圖架構





## 4.2 四大核心實體 (Entity)

- users (使用者) 系統使用者的主要實體
- products (商品) 產品資訊的主要實體
- orders (訂單) 訂單資訊的主要實體
- warehouses (倉庫) 倉庫資訊的主要實體

## 4.3 弱實體 (Weak Entities)

- order\_items (訂單明細) 依賴於 orders 和 products 的弱實體
- inventory (庫存) 依賴於 products 和 warehouses 的弱實體
- sellers (賣家) 依賴於 users 的弱實體

## 4.4 獨立實體 (Independent Entities)

- suppliers (供應商) 保留為獨立實體，與 products 有供應關係

## 4.5 關係描述 (Relationship Descriptions)

- 一個 users (使用者) 可以發送多條訊息給其他使用者，而一個使用者也可以收到多條來自其他使用者的訊息：
  - 每位使用者作為「發送者」可以發送多則訊息(1 對 N)
  - 每位使用者作為「接收者」可以接收多則訊息(1 對 N)
  - 每則訊息只有一個「發送者」和一個「接收者」
- 一個 users (使用者) 可以成為零個或一個 sellers (賣家)，而每個賣家必定對應一個使用者
- 一個 users (使用者) 可以下多個 orders (訂單)，每個訂單只能由一個使用者下單
- 一個 orders (訂單) 可以包含多個 order\_items (訂單明細)，而每個訂單明細必須屬於一個特定訂單
- 一個 products (商品) 可以被包含於多個 order\_items (訂單明細) 中，

而每個訂單明細必定只包含一個商品

- 一個 suppliers (供應商) 可以供應多個 products (商品)，一個商品可以由多個供應商供應
- 一個 sellers (賣家) 可以銷售多個 products (商品)，一個商品只能由一個賣家銷售
- 一個 products (商品) 必須屬於一個 categories (分類)，一個分類可以包含多個商品
- 一個 products (商品) 可以存儲於多個 inventory (庫存) 記錄中，每筆庫存記錄只對應一個商品
- 一個 warehouses (倉庫) 可以儲存多筆 inventory (庫存) 記錄，每筆庫存記錄只能存放在一個倉庫中

## 4.6 屬性關係 (Attribute Relationships)

- 一個 users (使用者) 擁有一個 AD\_server (身份驗證) 屬性和一個 membership\_levels (會員等級) 屬性
- 一個 users (使用者) 可以有多筆 login\_logs (登入日誌)、points (點數) 和 notifications (通知) 多值屬性
- 一個 products (商品) 擁有一個 customer\_feedback\_stats (客戶反饋統計) 屬性和多個 reviews (評論) 多值屬性
- 一個 orders (訂單) 擁有 payments (付款)、shipments (物流) 和 coupons (優惠券) 屬性，以及多個 return\_refunds (退款) 多值屬性
- 一個 warehouses (倉庫) 擁有多個 inbound\_shipments (入庫)、outbound\_shipments (出庫) 和 warehouse\_transfers (調撥記錄) 多值屬性

## 4.7 衍生資料 (Derived Data)

系統中的統計分析資料被表示為 Derived Data：

- sales\_analysis (銷售分析)
- inventory\_analytics (庫存分析)

- order\_conversion\_stats (訂單轉換率)
- shipment\_performance (物流績效)

## 5. 完整性限制

### 5.1 主鍵限制 (Primary Key Constraint)

- 每一個表格都應有唯一的主鍵 (如 user\_id、product\_id)
- 主鍵不可為 NULL，不可重複

### 5.2 外鍵限制 (Foreign Key Constraint)

表格之間的關聯應透過外鍵實現，例如：

- orders.user\_id 必須參照 users.user\_id
- product\_reviews.product\_id 必須參照 products.product\_id
- 刪除或修改參照資料時應定義對應行為 (如：CASCADE、RESTRICT)

### 5.3 唯一性限制 (Unique Constraint)

某些欄位必須全系統唯一，例如：

- 使用者的 email、帳號名稱不可重複
- 商品 SKU 編號必須唯一

### 5.4 非空限制 (NOT NULL Constraint)

必填欄位不得為 NULL，例如：

- 商品名稱、價格、數量、訂單狀態

## 5.5 資料型別限制 (Data Type Constraint)

- 價格欄位應為 decimal，數量欄位應為正整數
- 日期欄位應限定為有效格式 (ex: ISO 8601)

## 5.6 檢查限制 (CHECK Constraint)

某些欄位的值需落在特定範圍：

- 商品價格必須大於 0
- 評價分數為 1~5 分之間
- 訂單狀態為  
ENUM( 'pending' , ' shipped' , ' delivered' , ' cancelled' , ' return  
' )
- 對於 JWT token 金鑰過期以及竄改檢測

## 5.7 業務邏輯限制 (Business Rule Constraints)

### 5.7.1 評價限制

- 同一位買家只能對同一筆訂單留一次評價
- 未完成訂單不能回評

### 5.7.2 下單限制

- 不得對已下架商品下單
- 商品庫存不足時不可結帳
- 單一訂單金額不得超過 NT\$100,000

### 5.7.3 出貨限制

- 超過 7 天未出貨自動通知賣家
- 無法修改已出貨之訂單內容

#### 5.7.4 優惠限制

- 優惠券不得與其他折扣疊加
- 優惠券使用次數限制

#### 5.7.5 上架限制

- 商品名稱不得超過 100 字元
- 商品描述不得含有禁用關鍵字

#### 5.7.6 帳號限制

- 連續登入錯誤 5 次帳號自動鎖定 30 分鐘
- 被檢舉超過 5 次的賣家將暫停上架權限
- 停用帳號不得進行任何操作

#### 5.7.7 通知機制

- 買家下單後系統須即時發送通知給賣家
- 訂單領貨後 3 天未點收，系統自動標示「已送達」

## 6. 資料庫 Schema 設計

### 6.1 分離式安全架構設計

本系統採用**分離式安全架構**，將認證資料與業務資料完全隔離，此設計符合老師要求的安全性考量：

#### AD 認證資料庫（驗證伺服器）：

- 專門處理登入認證
- 只存放帳號、密碼雜湊、登入狀態等敏感資訊
- 與業務資料完全隔離

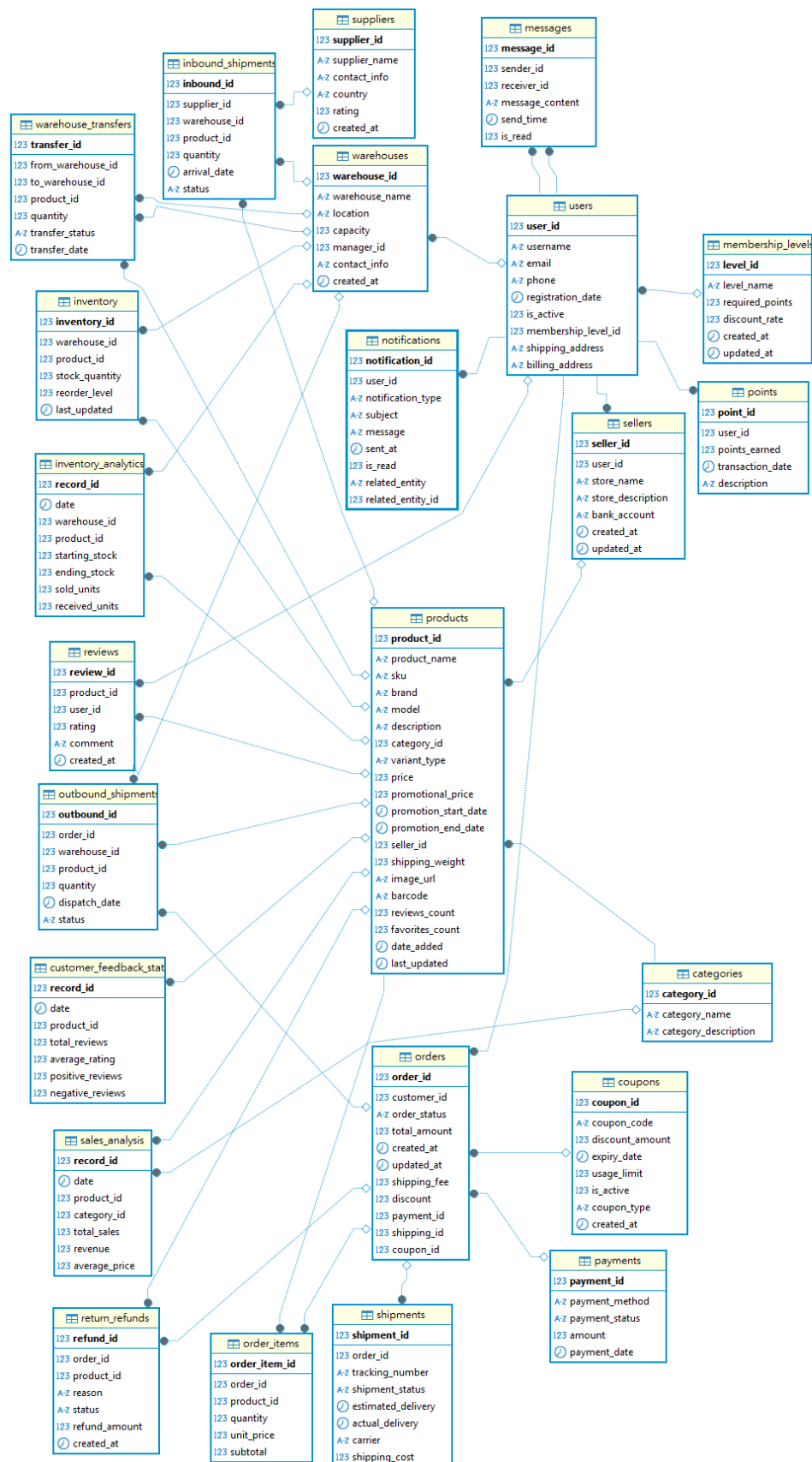
#### 主要電商資料庫（業務伺服器）：

- 存放用戶的業務資料
- 包含個人資料、會員等級、地址等非敏感資訊
- 不包含任何密碼資訊

#### 安全優勢：

- 資料隔離：即使業務資料庫被攻擊，密碼資料依然安全
- 權限分離：不同系統管理員負責不同資料庫
- 降低風險：減少敏感資料暴露面
- 符合最佳實務：遵循「最小權限原則」

## 6.2 完整資料庫關聯圖



## 6.3 資料格式規範與完整性需求

### 6.3.1 基本格式支援

- TIMESTAMP 支援格式
  - 2024-04-10T18:30:00
  - 2024-04-10T00:00:00
  
- Username、商店名稱支援符號
  - 0~9, a~z, A~Z, \_, -, @
  - 允許長度 5~20
  
- Email 支援符號
  - 0~9, a~z, A~Z
  - !#\$%&'\*+,-/=/?^\_{|}~‘，並且不可為首尾字母
  
- 密碼規格
  - 0~9, a~z, A~Z, # % \$ @
  - 密碼設定長度至少為 8 個字元
  - 無法使用重複、過於簡單且易於猜測或與帳號相同的字母或數字
  - 必須包含一個特殊字元
  
- Phone 支援格式
  - 0~9，格式：09\*\*\*\*\*
  - 手機號碼開頭不可為 0 以外的數字
  
- 地址支援符號
  - 0~9、中文字符
  - 長度不可超過 255



### 6.3.2 資料完整性需求與 Database Schema Diagram

由於篇幅過長，請至 github 進行參考：

<https://github.com/EZ-Super/Database-System-Course-Final-Project/blob/main/File/DataIntegrityRequirements.md>

## 6.4 認證伺服器模組 Schema（楊峻朋）

- AD 認證資料庫 獨立於主要電商資料庫

```
CREATE TABLE users (  
    user_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    password_hash VARCHAR(255) NOT NULL,  
    account_enable BOOLEAN NOT NULL DEFAULT TRUE,  
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    failed_login_attempts INT NOT NULL DEFAULT 0 CHECK  
(failed_login_attempts <= 5),  
    is_verified BOOLEAN NOT NULL,  
    updated_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;  
  
CREATE TABLE login_logs (  
    log_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    user_id INT NOT NULL,  
    email VARCHAR(255) NOT NULL,
```

```

login_time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,

ip_address VARCHAR(45) NOT NULL,

user_agent VARCHAR(255),

success BOOLEAN NOT NULL,

failure_reason VARCHAR(255)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

## 6.5 用戶管理模組 Schema（謝延偵）

### - 會員等級表

```

CREATE TABLE membership_levels (

    level_id INT AUTO_INCREMENT PRIMARY KEY,

    level_name VARCHAR(255) NOT NULL,

    required_points INTEGER NOT NULL,

    discount_rate DECIMAL(5,2) NOT NULL,

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

### - 業務用戶表（主要電商資料庫）

```

CREATE TABLE users (

    user_id INT AUTO_INCREMENT PRIMARY KEY,

    username VARCHAR(255) NOT NULL UNIQUE,

    email VARCHAR(255) NOT NULL UNIQUE,

    phone VARCHAR(20),

```

```

    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    is_active BOOLEAN DEFAULT TRUE,

    membership_level_id INTEGER,

    shipping_address VARCHAR(255),

    billing_address VARCHAR(255),

    FOREIGN KEY (membership_level_id) REFERENCES
membership_levels(level_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

– 賣家表

```

CREATE TABLE sellers (

    seller_id INT AUTO_INCREMENT PRIMARY KEY,

    user_id INTEGER NOT NULL UNIQUE,

    store_name VARCHAR(255) NOT NULL,

    store_description TEXT,

    bank_account VARCHAR(255),

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

    FOREIGN KEY (user_id) REFERENCES users(user_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

– 點數表

```

CREATE TABLE points (

    point_id INT AUTO_INCREMENT PRIMARY KEY,

```

```

user_id INTEGER NOT NULL,

points_earned INTEGER,

transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

description VARCHAR(255),

FOREIGN KEY (user_id) REFERENCES users(user_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

#### - 優惠券表

```

CREATE TABLE coupons (

    coupon_id INT AUTO_INCREMENT PRIMARY KEY,

    coupon_code VARCHAR(255) NOT NULL UNIQUE,

    discount_amount DECIMAL(10,2) NOT NULL,

    expiry_date DATE,

    usage_limit INTEGER,

    is_active BOOLEAN DEFAULT TRUE,

    coupon_type ENUM('single_use', 'multi_use_limited') NOT NULL
    DEFAULT 'single_use',

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

#### - 訊息表

```

CREATE TABLE messages (

    message_id INT AUTO_INCREMENT PRIMARY KEY,

    sender_id INTEGER NOT NULL,

```

```

receiver_id INTEGER NOT NULL,

message_content TEXT NOT NULL,

send_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

is_read BOOLEAN DEFAULT FALSE,

FOREIGN KEY (sender_id) REFERENCES users(user_id),

FOREIGN KEY (receiver_id) REFERENCES users(user_id),

INDEX idx_sender (sender_id),

INDEX idx_receiver (receiver_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

- 通知表

```

CREATE TABLE notifications (

    notification_id INT AUTO_INCREMENT PRIMARY KEY,

    user_id INTEGER NOT NULL,

    notification_type VARCHAR(50) NOT NULL,

    subject VARCHAR(255) NOT NULL,

    message TEXT NOT NULL,

    sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    is_read BOOLEAN DEFAULT FALSE,

    related_entity VARCHAR(50),

    related_entity_id INTEGER,

    FOREIGN KEY (user_id) REFERENCES users(user_id),

    INDEX idx_user_read (user_id, is_read)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

## 6.6 商品管理模組 Schema (楊峻朋)

- 商品分類表

```
CREATE TABLE categories (  
    category_id INT PRIMARY KEY,  
    category_name VARCHAR(255) NOT NULL UNIQUE,  
    category_description TEXT  
);
```

- 商品表

```
CREATE TABLE products (  
    product_id INT PRIMARY KEY,  
    product_name VARCHAR(255) NOT NULL,  
    sku VARCHAR(100) NOT NULL UNIQUE,  
    brand VARCHAR(100),  
    model VARCHAR(100),  
    description TEXT,  
    category_id INT NOT NULL,  
    variant_type VARCHAR(255),  
    price DECIMAL(10,2) NOT NULL,  
    promotional_price DECIMAL(10,2),  
    promotion_start_date TIMESTAMP,  
    promotion_end_date TIMESTAMP,  
    seller_id INT,  
    shipping_weight DECIMAL(10,2) DEFAULT 0,
```

```

image_url VARCHAR(255),

barcode VARCHAR(50),

reviews_count INT DEFAULT 0,

favorites_count INT DEFAULT 0,

date_added TIMESTAMP DEFAULT NOW(),

last_updated TIMESTAMP DEFAULT NOW() ON UPDATE NOW(),


FOREIGN KEY (category_id) REFERENCES categories(category_id),

FOREIGN KEY (seller_id) REFERENCES sellers(seller_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

#### - 評論表

```

CREATE TABLE reviews (

    review_id INT PRIMARY KEY,

    product_id INT,

    user_id INT,

    rating INT NOT NULL,

    comment TEXT NOT NULL,

    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,


    FOREIGN KEY (product_id) REFERENCES products(product_id),

    FOREIGN KEY (user_id) REFERENCES users(user_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

## 6.7 物流倉儲模組 Schema (鍾文瑤)

### - 倉庫管理

```
CREATE TABLE warehouses (  
    warehouse_id INT PRIMARY KEY AUTO_INCREMENT,  
    warehouse_name VARCHAR(100) NOT NULL,  
    location VARCHAR(255) NOT NULL,  
    capacity INT NOT NULL,  
    manager_id INT,  
    contact_info VARCHAR(255),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (manager_id) REFERENCES users(user_id)  
);
```

### - 庫存管理

```
CREATE TABLE inventory (  
    inventory_id INT PRIMARY KEY AUTO_INCREMENT,  
    warehouse_id INT,  
    product_id INT,  
    stock_quantity INT DEFAULT 0,  
    reorder_level INT DEFAULT 10,  
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(warehouse_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```



- 供應商管理

```
CREATE TABLE suppliers (  
    supplier_id INT PRIMARY KEY AUTO_INCREMENT,  
    supplier_name VARCHAR(100) NOT NULL,  
    contact_info VARCHAR(255),  
    country VARCHAR(50),  
    rating INT CHECK (rating BETWEEN 1 AND 5),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

- 入庫記錄

```
CREATE TABLE inbound_shipments (  
    inbound_id INT PRIMARY KEY AUTO_INCREMENT,  
    supplier_id INT,  
    warehouse_id INT,  
    product_id INT,  
    quantity INT NOT NULL,  
    arrival_date DATE NOT NULL,  
    status ENUM('pending', 'received', 'damaged') DEFAULT 'pending',  
    FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id),  
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(warehouse_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

- 出庫記錄

```
CREATE TABLE outbound_shipments (  
    outbound_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT,  
    warehouse_id INT,  
    product_id INT,  
    quantity INT NOT NULL,  
    dispatch_date DATE NOT NULL,  
    status ENUM('preparing', 'shipped', 'delivered') DEFAULT  
'preparing',  
    FOREIGN KEY (order_id) REFERENCES orders(order_id),  
    FOREIGN KEY (warehouse_id) REFERENCES warehouses(warehouse_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

- 物流配送

```
CREATE TABLE shipments (  
    shipment_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT,  
    tracking_number VARCHAR(50) UNIQUE,  
    shipment_status ENUM('preparing', 'shipped', 'delivered') NOT  
NULL,  
    estimated_delivery DATE,  
    actual_delivery DATE,
```

```

    carrier VARCHAR(50),

    shipping_cost DECIMAL(10,2) DEFAULT 0.00,

    FOREIGN KEY (order_id) REFERENCES orders(order_id)
);

- 倉庫內調撥

CREATE TABLE warehouse_transfers (

    transfer_id INT PRIMARY KEY AUTO_INCREMENT,

    from_warehouse_id INT,

    to_warehouse_id INT,

    product_id INT,

    quantity INT NOT NULL,

    transfer_status ENUM('pending', 'in_transit', 'completed')
DEFAULT 'pending',

    transfer_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (from_warehouse_id) REFERENCES
warehouses(warehouse_id),

    FOREIGN KEY (to_warehouse_id) REFERENCES
warehouses(warehouse_id),

    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

```

## 6.8 數據分析模組 Schema (鍾文瑤)

- 銷售分析

```

CREATE TABLE sales_analysis (

```

```

record_id INT PRIMARY KEY AUTO_INCREMENT,

date DATE NOT NULL,

product_id INT,

category_id INT,

total_sales INT NOT NULL,

revenue DECIMAL(10,2) NOT NULL,

average_price DECIMAL(10,2) NOT NULL,

FOREIGN KEY (product_id) REFERENCES products(product_id),

FOREIGN KEY (category_id) REFERENCES categories(category_id)

);

```

#### - 庫存趨勢

```

CREATE TABLE inventory_analytics (

    record_id INT PRIMARY KEY AUTO_INCREMENT,

    date DATE NOT NULL,

    warehouse_id INT,

    product_id INT,

    starting_stock INT NOT NULL,

    ending_stock INT NOT NULL,

    sold_units INT NOT NULL,

    received_units INT NOT NULL,

    FOREIGN KEY (warehouse_id) REFERENCES warehouses(warehouse_id),

    FOREIGN KEY (product_id) REFERENCES products(product_id)

);

```

- 訂單轉換率

```
CREATE TABLE order_conversion_stats (  
    record_id INT PRIMARY KEY AUTO_INCREMENT,  
    date DATE NOT NULL,  
    total_visits INT NOT NULL,  
    total_orders INT NOT NULL,  
    conversion_rate DECIMAL(5,2) NOT NULL  
);
```

- 物流配送表現

```
CREATE TABLE shipment_performance (  
    record_id INT PRIMARY KEY AUTO_INCREMENT,  
    date DATE NOT NULL,  
    carrier VARCHAR(50) NOT NULL,  
    total_shipments INT NOT NULL,  
    on_time_deliveries INT NOT NULL,  
    late_deliveries INT NOT NULL,  
    average_delivery_time INT NOT NULL - 單位: 小時  
);
```

- 顧客滿意度分析

```
CREATE TABLE customer_feedback_stats (  
    record_id INT PRIMARY KEY AUTO_INCREMENT,
```

```

    date DATE NOT NULL,

    product_id INT,

    total_reviews INT NOT NULL,

    average_rating DECIMAL(3,2) NOT NULL,

    positive_reviews INT NOT NULL,

    negative_reviews INT NOT NULL,

    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

```

## 6.9 訂單管理模組 Schema (李維誠)

- orders 訂單表

```

CREATE TABLE orders (

    order_id      INT AUTO_INCREMENT PRIMARY KEY,

    customer_id   INT NOT NULL,

    order_status  ENUM('pending', 'shipped', 'delivered',
'cancelled', 'return') NOT NULL,

    total_amount  NUMERIC(10,2) NOT NULL,

    created_at    TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    updated_at    TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,

    shipping_fee  NUMERIC(10,2) DEFAULT 0,

    discount      NUMERIC(10,2) DEFAULT 0,

    payment_id    INT,

    shipping_id   INT,

    coupon_id     INT,

```

```

FOREIGN KEY (customer_id) REFERENCES users(user_id),
FOREIGN KEY (payment_id) REFERENCES payments(payment_id),
FOREIGN KEY (coupon_id) REFERENCES coupons(coupon_id)
);

- order_items 訂單品項表
CREATE TABLE order_items (
    order_item_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id      INT,
    product_id    INT NOT NULL,
    quantity      INT NOT NULL,
    unit_price    NUMERIC(10,2) NOT NULL,
    subtotal      NUMERIC(10,2) AS (quantity * unit_price) STORED,

    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);

- payments 付款表
CREATE TABLE payments (
    payment_id      INT AUTO_INCREMENT PRIMARY KEY,
    payment_method  ENUM('credit_card', 'paypal', 'bank_transfer',
'cash_on_delivery') NOT NULL,

```

```
payment_status    ENUM('pending', 'completed', 'failed') NOT NULL,  
amount            NUMERIC(10,2) NOT NULL,  
payment_date      TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

- return\_refunds 退貨退款表

```
CREATE TABLE return_refunds (  
    refund_id      INT AUTO_INCREMENT PRIMARY KEY,  
    order_id       INT,  
    product_id     INT,  
    reason         TEXT,  
    status         ENUM('requested', 'approved', 'rejected',  
'processed') DEFAULT 'requested',  
    refund_amount  NUMERIC(10,2),  
    created_at     TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  
    FOREIGN KEY (order_id) REFERENCES orders(order_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```



## 7. 系統測試資料集

### 7.1 用戶管理（謝延偵）

- membership\_levels

```
INSERT INTO membership_levels (level_id, level_name, required_points,
discount_rate) VALUES
(1, '銅魚', 0, 2.00),
(2, '銀魚', 1000, 5.00),
(3, '金魚', 3000, 10.00),
(4, '銅魚', 0, 2.00),
(5, '銀魚', 1000, 5.00),
(6, '金魚', 3000, 10.00),
(7, '銅魚', 0, 2.00),
(8, '銀魚', 1000, 5.00),
(9, '金魚', 3000, 10.00),
(10, '銅魚', 0, 2.00);
```

- users

```
INSERT INTO users (user_id, username, email, phone,
registration_date, is_active, membership_level_id, shipping_address,
billing_address) VALUES
(1, '陳小明', 'chen.xiaoming@gmail.com', '0912345678', '2023-01-15
10:30:00', TRUE, 1, '台北市大安區忠孝東路四段 112 號', '台北市大安區忠
孝東路四段 112 號'),
(2, '林雅婷', 'lin.yating@yahoo.com.tw', '0923456789', '2023-02-20
14:22:15', TRUE, 2, '新北市板橋區文化路二段 88 號', '新北市板橋區文化
路二段 88 號'),
```

(3, '王大偉', 'david.wang@hotmail.com', '0934567890', '2023-03-08 09:45:30', TRUE, 3, '台中市西區台灣大道二段 285 號', '台中市西區台灣大道二段 285 號'),

(4, '張美玲', 'meiling.chang@outlook.com', '0945678901', '2023-04-12 16:18:45', FALSE, 4, '高雄市前金區中正四路 105 號', '高雄市前金區中正四路 105 號'),

(5, '李志豪', 'lee.zhihao@gmail.com', '0956789012', '2023-05-25 11:33:20', TRUE, 5, '台南市東區長榮路三段 66 號', '台南市東區長榮路三段 66 號'),

(6, '黃淑芬', 'huang.shufen@yahoo.com.tw', '0967890123', '2023-06-30 13:55:10', TRUE, 6, '桃園市中壢區中正路 199 號', '桃園市中壢區中正路 199 號'),

(7, '劉建國', 'liu.jianguo@gmail.com', '0978901234', '2023-07-14 08:27:35', TRUE, 7, '新竹市東區光復路二段 321 號', '新竹市東區光復路二段 321 號'),

(8, '蔡雨晴', 'tsai.yuqing@hotmail.com', '0989012345', '2023-08-19 15:42:18', TRUE, 8, '彰化市中山路一段 177 號', '彰化市中山路一段 177 號'),

(9, '楊文華', 'yang.wenhua@outlook.com', '0990123456', '2023-09-22 12:08:42', TRUE, 9, '嘉義市西區文化路 255 號', '嘉義市西區文化路 255 號'),

(10, '吳玉芳', 'wu.yufang@gmail.com', '0901234567', '2023-10-05 17:15:55', TRUE, 10, '宜蘭市中山路五段 133 號', '宜蘭市中山路五段 133 號');

## ● sellers

```
INSERT INTO sellers (seller_id, user_id, store_name,
store_description, bank_account) VALUES
```

```
(1, 3, '大偉 3C 數位館', '專售手機、平板、電腦週邊商品，品質保證，價格實惠', '700-0001234567890'),
```

(2, 5, '志豪文創小舖', '手作文創商品、客製化禮品、文具用品專賣店', '822-0009876543210'),

(3, 7, '建國運動用品', '各式運動器材、健身用品、戶外用品專業販售', '012-3456789012345'),

(4, 1, '小明美食天地', '精選台灣在地美食、伴手禮、有機農產品', '808-0002468013579'),

(5, 10, '玉芳居家生活', '居家裝飾、收納用品、廚房用具一應俱全', '703-0001357924680'),

(6, 2, '雅婷時尚服飾', '女性服飾、配件、包包, 追求時尚與品質並重', '700-0009630741852'),

(7, 8, '雨晴寵物天堂', '寵物用品、飼料、玩具, 寵物健康照護專家', '822-0007410852963'),

(8, 4, '美玲花藝工作室', '鮮花花束、盆栽、園藝用品, 美化您的生活空間', '012-9876543210987'),

(9, 6, '淑芬手工烘焙', '手工餅乾、蛋糕、麵包, 天然食材無添加', '808-0001472583690'),

(10, 9, '文華書香閣', '二手書籍、文學作品、學習教材, 知識的寶庫', '703-0008529637410');

● points

```
INSERT INTO points (point_id, user_id, points_earned,
transaction_date, description) VALUES
```

(1, 1, 150, '2024-01-10 14:30:25', '購物'),

(2, 2, 80, '2024-01-15 09:45:12', '活動'),

(3, 3, 320, '2024-01-20 16:22:38', '購物'),

(4, 4, 50, '2024-01-25 11:15:45', '簽到'),

(5, 5, 250, '2024-02-01 13:55:20', '活動'),

(6, 6, 120, '2024-02-08 10:33:15', '購物'),

```
(7, 7, -200, '2024-02-14 15:18:30', '折抵消費'),
(8, 8, 90, '2024-02-20 08:42:55', '簽到'),
(9, 9, 180, '2024-02-28 17:25:10', '活動'),
(10, 10, -100, '2024-03-05 12:38:42', '折抵消費');
```

#### ● coupons

```
INSERT INTO coupons (coupon_id, coupon_code, discount_amount,
expiry_date, usage_limit, is_active, coupon_type) VALUES
(1, 'WELCOME100', 100.00, '2024-12-31', NULL, TRUE, 'single_use'),
(2, 'SPRING2024', 50.00, '2024-06-30', 1000, TRUE,
'multi_use_limited'),
(3, 'VIP200', 200.00, '2024-10-31', NULL, TRUE, 'single_use'),
(4, 'SUMMER20', 20.00, '2024-08-31', 500, TRUE, 'multi_use_limited'),
(5, 'BIRTHDAY150', 150.00, '2024-12-31', NULL, TRUE, 'single_use'),
(6, 'FRIEND80', 80.00, '2024-09-30', 300, TRUE, 'multi_use_limited'),
(7, 'WEEKEND30', 30.00, '2024-07-31', 200, FALSE,
'multi_use_limited'),
(8, 'NEWUSER50', 50.00, '2024-12-31', NULL, TRUE, 'single_use'),
(9, 'GOLD300', 300.00, '2024-11-30', NULL, TRUE, 'single_use'),
(10, 'FLASH25', 25.00, '2024-06-15', 100, TRUE, 'multi_use_limited');
```

#### ● messages

```
INSERT INTO messages (message_id, sender_id, receiver_id,
message_content, send_time, is_read) VALUES
(1, 1, 3, '你好，想詢問這款手機還有現貨嗎?', '2024-03-01 10:15:30',
TRUE),
(2, 3, 1, '有的，目前還有3台黑色的，需要的話可以直接下單', '2024-03-
```

```

01 10:22:45', TRUE),
(3, 2, 6, '這件洋裝有其他顏色嗎?', '2024-03-02 14:33:20', FALSE),
(4, 5, 8, '請問寵物飼料有提供試吃包嗎?', '2024-03-03 09:45:15',
TRUE),
(5, 8, 5, '當然有，我們有小包裝的試吃包，可以先試試看', '2024-03-03
09:52:30', TRUE),
(6, 4, 9, '想買幾本程式設計的書，有推薦的嗎?', '2024-03-04
16:18:45', TRUE),
(7, 9, 4, '我推薦 Python 入門和 JavaScript 實戰這兩本', '2024-03-04
16:25:12', FALSE),
(8, 7, 10, '這個收納盒的尺寸是多少?', '2024-03-05 11:40:55', TRUE),
(9, 10, 7, '長 30cm、寬 20cm、高 15cm，很適合放書桌上', '2024-03-05
11:48:20', TRUE),
(10, 6, 1, '謝謝你的蛋糕食譜分享，做出來很成功!', '2024-03-06
13:25:38', FALSE);

```

#### ● notifications

```

INSERT INTO notifications (
    notification_id, user_id, notification_type, subject, message,
    sent_at, is_read, related_entity, related_entity_id
) VALUES
(1, 1, 'order', '訂單確認通知', '您的訂單 #12345 已確認，預計 3-5 個工
作天送達', '2024-03-01 11:30:15', TRUE, 'order', 12345),
(2, 2, 'promotion', '限時優惠活動', '春季服飾特賣會開始了！全館 8 折起
', '2024-03-02 09:00:00', FALSE, 'promotion', 101),
(3, 3, 'message', '新訊息通知', '您收到了來自陳小明的新訊息', '2024-
03-01 10:22:50', TRUE, 'message', 2),
(4, 4, 'system', '密碼修改通知', '您的密碼已成功修改，若非本人操作請

```

```

    聯繫客服', '2024-03-03 15:45:20', TRUE, 'user', 4),
    (5, 5, 'points', '點數獲得通知', '恭喜您獲得 250 點數！繼續加油！',
    '2024-02-01 13:55:25', TRUE, 'points', 5),
    (6, 6, 'order', '配送通知', '您的訂單正在配送中，預計今日下午送達',
    '2024-03-04 08:30:45', FALSE, 'order', 67890),
    (7, 7, 'review', '評價提醒', '別忘了為您購買的商品留下評價喔！',
    '2024-03-05 10:15:30', FALSE, 'order', 11111),
    (8, 8, 'coupon', '優惠券到期提醒', '您的優惠券將於明天到期，記得使用
    喔！', '2024-03-06 12:00:00', TRUE, 'coupon', 7),
    (9, 9, 'membership', '會員升級通知', '恭喜您升級為銀魚會員！享受更多
    專屬優惠', '2024-03-07 14:22:18', TRUE, 'user', 9),
    (10, 10, 'promotion', '生日優惠', '生日快樂！專屬生日優惠券已送到您的
    帳戶', '2024-03-08 00:00:00', FALSE, 'coupon', 5);

```

## 7.2 商品管理（楊峻朋）

### ● products

```

INSERT INTO products (product_id, product_name, sku, brand, model,
price, promotional_price, category_id, seller_id, reviews_count,
favorites_count, date_added, last_updated) VALUES
(1, '無線滑鼠', 'WM-001', 'Logitech', 'M185', 599.00, 499.00, 1, 1,
23, 15, '2025-05-30 10:00:00', '2025-05-30 10:00:00'),
(2, '機械式鍵盤', 'KB-002', 'Keychron', 'K2 V2', 2890.00, 2590.00, 1,
2, 120, 78, '2025-05-30 10:01:00', '2025-05-30 10:01:00'),
(3, '藍牙耳機', 'BT-003', 'Sony', 'WF-1000XM4', 7990.00, NULL, 2, 1,
55, 40, '2025-05-30 10:02:00', '2025-05-30 10:02:00'),
(4, '曲面電競螢幕', 'MON-004', 'Samsung', 'C27F398', 6990.00,
6490.00, 3, 3, 10, 7, '2025-05-30 10:03:00', '2025-05-30 10:03:00'),
(5, '筆電支架', 'ST-005', 'MOFT', 'Z Sit-Stand', 1380.00, NULL, 4, 2,

```

```

8, 12, '2025-05-30 10:04:00', '2025-05-30 10:04:00'),
(6, '外接固態硬碟', 'SSD-006', 'SanDisk', 'Extreme V2', 3290.00,
2990.00, 5, 4, 34, 22, '2025-05-30 10:05:00', '2025-05-30 10:05:00'),
(7, 'Type-C 集線器', 'HUB-007', 'UGREEN', '9-in-1', 990.00, 890.00,
6, 1, 19, 18, '2025-05-30 10:06:00', '2025-05-30 10:06:00'),
(8, '攝影腳架', 'TRIP-008', 'Manfrotto', 'MK290DUA3', 4690.00, NULL,
7, 5, 4, 6, '2025-05-30 10:07:00', '2025-05-30 10:07:00'),
(9, '行動電源', 'PB-009', 'Anker', 'PowerCore', 1190.00, 990.00, 8,
1, 45, 51, '2025-05-30 10:08:00', '2025-05-30 10:08:00'),
(10, '智慧手錶', 'WT-010', 'Apple', 'Watch SE', 8990.00, 8490.00, 9,
2, 65, 58, '2025-05-30 10:09:00', '2025-05-30 10:09:00');

```

#### ● categories

```

INSERT INTO categories (category_id, category_name,
category_description) VALUES
(1, '電腦週邊', '滑鼠、鍵盤、耳機等常用配件'),
(2, '音訊設備', '各類耳機、喇叭、麥克風等聲音相關設備'),
(3, '顯示器', '各種尺寸與功能的顯示器與螢幕'),
(4, '筆電配件', '筆電支架、散熱墊、包包等相關產品'),
(5, '儲存設備', 'SSD、隨身碟、記憶卡與備份設備'),
(6, '轉接與集線設備', 'USB Hub、轉接頭與連接線等'),
(7, '攝影器材', '腳架、雲台、燈光與攝影周邊'),
(8, '行動裝置配件', '行動電源、充電線與手機架'),
(9, '穿戴裝置', '智慧手錶、手環與健康監控設備'),
(10, '智慧家電', '家用物聯網設備、智慧插座、溫濕度感測器');

```

#### ● reviews

```
INSERT INTO reviews (review_id, product_id, user_id, rating, comment,
created_at) VALUES
```

```
(1, 1, 1, 5, '滑鼠非常靈敏，握感舒適，超值！', '2025-05-30
09:01:00'),
```

```
(2, 2, 2, 4, '鍵盤手感不錯，燈效很炫，但有點重', '2025-05-30
09:02:00'),
```

```
(3, 3, 3, 5, '耳機降噪效果超棒，聽音樂很有沉浸感', '2025-05-30
09:03:00'),
```

```
(4, 4, 4, 3, '螢幕解析度不錯，但底座不太穩', '2025-05-30 09:04:00'),
```

```
(5, 5, 5, 4, '筆電支架輕巧又方便，角度調整剛好', '2025-05-30
09:05:00'),
```

```
(6, 6, 6, 5, '傳輸速度超快！非常適合備份資料', '2025-05-30
09:06:00'),
```

```
(7, 7, 7, 4, '集線器支援功能多，但插孔間距有點擠', '2025-05-30
09:07:00'),
```

```
(8, 8, 8, 5, '腳架很穩，出外拍照必備', '2025-05-30 09:08:00'),
```

```
(9, 9, 9, 2, '行動電源容量不如預期，使用兩次就沒電了', '2025-05-30
09:09:00'),
```

```
(10, 10, 3, 5, 'Apple Watch 很實用，健康追蹤功能很準', '2025-05-30
09:10:00');
```

### 7.3 驗證伺服器(楊峻朋)

- users

```
INSERT INTO users (user_id, email, password_hash, account_enable,
created_at, failed_login_attempts, is_verified, updated_at) VALUES
```

```
(101, 'alice@example.com', '$2b$12$aliceHash', TRUE, '2025-05-01
09:00:00', 0, TRUE, '2025-05-01 09:00:00'),
```

```
(102, 'bob@example.com', '$2b$12$bobHash', TRUE, '2025-05-02
```



```

09:30:00', 2, TRUE, '2025-05-28 13:00:00' ),
(103, 'charlie@example.com', '$2b$12$charlieHash', FALSE, '2025-05-03
08:50:00', 5, TRUE, '2025-05-29 10:00:00' ),
(104, 'diana@example.com', '$2b$12$dianaHash', FALSE, '2025-05-04
11:20:00', 0, FALSE, '2025-05-30 08:00:00' ),
(105, 'ethan@example.com', '$2b$12$ethanHash', TRUE, '2025-05-05
07:45:00', 1, TRUE, '2025-05-30 07:45:00' ),
(106, 'fiona@example.com', '$2b$12$fionaHash', TRUE, '2025-05-06
12:15:00', 0, TRUE, '2025-05-06 12:15:00' ),
(107, 'george@example.com', '$2b$12$georgeHash', TRUE, '2025-05-07
10:10:00', 0, FALSE, '2025-05-30 08:17:00' ),
(108, 'hannah@example.com', '$2b$12$hannahHash', TRUE, '2025-05-08
09:55:00', 0, TRUE, '2025-05-08 09:55:00' ),
(109, 'ian@example.com', '$2b$12$ianHash', TRUE, '2025-05-09
14:20:00', 4, TRUE, '2025-05-29 18:00:00' ),
(110, 'jess@example.com', '$2b$12$jessHash', TRUE, '2025-05-10
16:40:00', 0, FALSE, '2025-05-10 16:40:00' );

```

- login\_logs

```

INSERT INTO login_logs (log_id, user_id, email, login_time,
ip_address, user_agent, success, failure_reason) VALUES
(1, 101, 'alice@example.com', '2025-05-30 08:00:01', '192.168.1.10',
'Mozilla/5.0 (Windows NT 10.0)', TRUE, NULL),
(2, 102, 'bob@example.com', '2025-05-30 08:03:17', '172.16.0.5',
'Chrome/90.0 (Mac OS X 10.15)', FALSE, 'Wrong password'),
(3, 103, 'charlie@example.com', '2025-05-30 08:05:22',
'203.0.113.12', 'Safari/13.1 (iPhone; iOS 14.4)', TRUE, NULL),
(4, 104, 'diana@example.com', '2025-05-30 08:07:10', '10.0.0.8',
'Firefox/88.0 (Ubuntu 20.04)', FALSE, 'Account locked'),

```

```
(5, 105, 'ethan@example.com', '2025-05-30 08:09:35', '192.168.0.33',
'Edge/92.0 (Windows 10)', TRUE, NULL),

(6, 101, 'alice@example.com', '2025-05-30 08:11:50', '192.168.1.10',
'Mozilla/5.0 (Windows NT 10.0)', FALSE, 'Wrong password'),

(7, 106, 'fiona@example.com', '2025-05-30 08:15:02', '198.51.100.25',
'Chrome/91.0 (Linux x86_64)', TRUE, NULL),

(8, 107, 'george@example.com', '2025-05-30 08:17:47', '203.0.113.33',
'Safari/14.0 (MacOS)', FALSE, 'Invalid email format'),

(9, 108, 'hannah@example.com', '2025-05-30 08:19:12', '10.10.0.11',
'Mozilla/5.0 (Android 11; Mobile)', TRUE, NULL),

(10, 109, 'ian@example.com', '2025-05-30 08:21:30', '172.16.100.50',
'Edge/93.0 (Windows 11)', FALSE, 'Too many attempts');
```

## 7.4 倉儲管理（鍾文瑤）

- warehouses

```
INSERT INTO 'warehouses' ('warehouse_id', 'warehouse_name',
'location', 'capacity', 'manager_id', 'contact_info', 'created_at')
VALUES

(1, '倉庫 1', '台灣 1 號', 1815, 1, 'manager1@warehouse.com', '2025-
05-07 00:00:00'),

(2, '倉庫 2', '台灣 2 號', 4737, 2, 'manager2@warehouse.com', '2025-
01-23 00:00:00'),

(3, '倉庫 3', '台灣 3 號', 2565, 3, 'manager3@warehouse.com', '2025-
01-31 00:00:00'),

(4, '倉庫 4', '台灣 4 號', 1967, 4, 'manager4@warehouse.com', '2025-
05-22 00:00:00'),

(5, '倉庫 5', '台灣 5 號', 2088, 5, 'manager5@warehouse.com', '2025-
03-26 00:00:00'),
```

```
(6, '倉庫 6', '台灣 6 號', 3755, 6, 'manager6@warehouse.com', '2025-01-02 00:00:00'),
(7, '倉庫 7', '台灣 7 號', 1688, 7, 'manager7@warehouse.com', '2025-02-13 00:00:00'),
(8, '倉庫 8', '台灣 8 號', 1248, 8, 'manager8@warehouse.com', '2025-02-10 00:00:00'),
(9, '倉庫 9', '台灣 9 號', 1491, 9, 'manager9@warehouse.com', '2025-01-17 00:00:00'),
(10, '倉庫 10', '台灣 10 號', 3980, 10, 'manager10@warehouse.com', '2025-05-27 00:00:00');
```

- inventory

```
INSERT INTO 'inventory' ('inventory_id', 'warehouse_id',
'product_id', 'stock_quantity', 'reorder_level', 'last_updated')
VALUES
(1, 4, 1, 383, 8, '2025-05-04 00:00:00'),
(2, 6, 4, 326, 10, '2025-04-25 00:00:00'),
(3, 6, 3, 207, 10, '2025-03-16 00:00:00'),
(4, 4, 7, 258, 13, '2025-04-02 00:00:00'),
(5, 8, 10, 298, 6, '2025-05-04 00:00:00'),
(6, 2, 6, 283, 9, '2025-05-07 00:00:00'),
(7, 4, 5, 424, 7, '2025-01-23 00:00:00'),
(8, 8, 3, 28, 6, '2025-02-22 00:00:00'),
(9, 1, 3, 393, 18, '2025-05-25 00:00:00'),
(10, 1, 3, 134, 15, '2025-04-21 00:00:00');
```

- suppliers

```
INSERT INTO 'suppliers' ('supplier_id', 'supplier_name',
```

```

'contact_info', 'country', 'rating', 'created_at') VALUES
(1, '供應商 1', 'contact1@supplier.com', '台灣', 4, '2025-01-23
00:00:00'),
(2, '供應商 2', 'contact2@supplier.com', '台灣', 2, '2025-02-02
00:00:00'),
(3, '供應商 3', 'contact3@supplier.com', '台灣', 2, '2025-03-19
00:00:00'),
(4, '供應商 4', 'contact4@supplier.com', '台灣', 4, '2025-04-07
00:00:00'),
(5, '供應商 5', 'contact5@supplier.com', '台灣', 1, '2025-03-03
00:00:00'),
(6, '供應商 6', 'contact6@supplier.com', '台灣', 5, '2025-04-26
00:00:00'),
(7, '供應商 7', 'contact7@supplier.com', '台灣', 2, '2025-05-07
00:00:00'),
(8, '供應商 8', 'contact8@supplier.com', '台灣', 4, '2025-04-09
00:00:00'),
(9, '供應商 9', 'contact9@supplier.com', '台灣', 2, '2025-05-18
00:00:00'),
(10, '供應商 10', 'contact10@supplier.com', '台灣', 3, '2025-01-28
00:00:00');

```

#### ● inbound\_Shipments

```

INSERT INTO 'inbound_Shipments' ('inbound_id', 'supplier_id',
'warehouse_id', 'product_id', 'quantity', 'arrival_date', 'status')
VALUES
(1, 8, 6, 10, 73, '2025-04-15', 'damaged'),
(2, 4, 10, 5, 78, '2025-04-14', 'pending'),

```

```
(3, 10, 2, 4, 91, '2025-01-06', 'damaged'),
(4, 8, 3, 5, 29, '2025-03-05', 'received'),
(5, 2, 4, 1, 75, '2025-05-21', 'damaged'),
(6, 1, 10, 3, 53, '2025-02-07', 'pending'),
(7, 10, 8, 10, 54, '2025-05-02', 'received'),
(8, 10, 8, 7, 89, '2025-03-03', 'damaged'),
(9, 7, 2, 6, 17, '2025-04-23', 'damaged'),
(10, 4, 6, 3, 84, '2025-04-12', 'damaged');
```

- outboundShipments

```
INSERT INTO 'outbound_shipments' ('outbound_id', 'order_id',
'warehouse_id', 'product_id', 'quantity', 'dispatch_date', 'status')
VALUES
(1, 8, 3, 6, 3, '2025-01-01', 'preparing'),
(2, 8, 2, 3, 4, '2025-05-05', 'delivered'),
(3, 8, 5, 2, 1, '2025-02-28', 'shipped'),
(4, 10, 6, 7, 10, '2025-01-17', 'shipped'),
(5, 6, 1, 7, 3, '2025-04-21', 'preparing'),
(6, 6, 1, 10, 9, '2025-02-17', 'preparing'),
(7, 10, 9, 9, 7, '2025-05-25', 'shipped'),
(8, 2, 6, 2, 9, '2025-02-24', 'shipped'),
(9, 3, 9, 6, 8, '2025-01-23', 'shipped'),
(10, 1, 3, 2, 10, '2025-02-19', 'delivered');
```

- shipments

```
INSERT INTO 'shipments' ('shipment_id', 'order_id',
'tracking_number', 'shipment_status', 'estimated_delivery',
```

```

'actual_delivery', 'carrier', 'shipping_cost') VALUES
(1, 1, 'TN000001', 'shipped', '2025-04-25', '2025-04-13', '快递公司
1', 473.92),
(2, 5, 'TN000002', 'shipped', '2025-02-21', '2025-04-14', '快递公司
2', 399.36),
(3, 1, 'TN000003', 'shipped', '2025-03-31', '2025-02-04', '快递公司
3', 490.37),
(4, 10, 'TN000004', 'delivered', '2025-02-22', '2025-05-14', '快递公
司 4', 289.43),
(5, 9, 'TN000005', 'delivered', '2025-03-19', '2025-03-29', '快递公司
5', 288.88),
(6, 3, 'TN000006', 'preparing', '2025-01-18', '2025-03-17', '快递公司
6', 193.95),
(7, 6, 'TN000007', 'shipped', '2025-03-24', '2025-02-21', '快递公司
7', 280.09),
(8, 4, 'TN000008', 'preparing', '2025-02-18', '2025-02-21', '快递公司
8', 125.88),
(9, 4, 'TN000009', 'preparing', '2025-03-13', '2025-03-13', '快递公司
9', 469.71),
(10, 4, 'TN000010', 'delivered', '2025-03-30', '2025-02-28', '快递公
司 10', 468.08);

```

#### ● warehouse\_Transfers

```

INSERT INTO 'warehouse_Transfers' ('transfer_id',
'from_warehouse_id', 'to_warehouse_id', 'product_id', 'quantity',
'transfer_status', 'transfer_date') VALUES
(1, 1, 10, 9, 30, 'completed', '2025-04-20 00:00:00'),
(2, 2, 9, 9, 13, 'completed', '2025-04-15 00:00:00'),

```

```
(3, 3, 8, 3, 26, 'pending', '2025-04-22 00:00:00'),
(4, 4, 7, 5, 15, 'completed', '2025-02-26 00:00:00'),
(5, 5, 6, 5, 21, 'pending', '2025-05-26 00:00:00'),
(6, 6, 5, 7, 20, 'in_transit', '2025-02-19 00:00:00'),
(7, 7, 4, 3, 7, 'pending', '2025-04-10 00:00:00'),
(8, 8, 3, 7, 15, 'completed', '2025-05-23 00:00:00'),
(9, 9, 2, 1, 25, 'pending', '2025-03-14 00:00:00'),
(10, 10, 1, 6, 20, 'pending', '2025-03-16 00:00:00');
```

## 7.5 數據分析（鍾文瑤）

- sales\_analysis

```
INSERT INTO 'sales_analysis' ('record_id', 'date', 'product_id',
'category_id', 'total_sales', 'revenue', 'average_price') VALUES
(1, '2025-05-30', 3, 1, 65, 7566.33, 579.02),
(2, '2025-05-29', 5, 5, 93, 1503.09, 470.06),
(3, '2025-05-28', 5, 5, 43, 9887.02, 386.1),
(4, '2025-05-27', 10, 2, 76, 5378.29, 237.54),
(5, '2025-05-26', 1, 4, 100, 2974.4, 765.54),
(6, '2025-05-25', 7, 3, 84, 8289.15, 700.52),
(7, '2025-05-24', 9, 4, 55, 7335.28, 661.48),
(8, '2025-05-23', 6, 5, 33, 4495.3, 270.57),
(9, '2025-05-22', 7, 5, 64, 9313.62, 455.99),
(10, '2025-05-21', 2, 1, 31, 4570.76, 760.73);
```

- inventory\_analytics

```
INSERT INTO 'inventory_analytics' ('record_id', 'date',
```

```

'warehouse_id', 'product_id', 'starting_stock', 'ending_stock',
'sold_units', 'received_units') VALUES
(1, '2025-05-30', 5, 4, 459, 405, 54, 27),
(2, '2025-05-29', 2, 8, 316, 225, 91, 6),
(3, '2025-05-28', 5, 2, 379, 368, 11, 46),
(4, '2025-05-27', 2, 4, 434, 377, 57, 2),
(5, '2025-05-26', 1, 8, 481, 467, 14, 5),
(6, '2025-05-25', 1, 5, 110, 46, 64, 16),
(7, '2025-05-24', 4, 6, 104, 59, 45, 21),
(8, '2025-05-23', 3, 10, 303, 221, 82, 27),
(9, '2025-05-22', 5, 5, 128, 79, 49, 20),
(10, '2025-05-21', 3, 5, 425, 360, 65, 1);

```

- order\_conversion\_stats

```

INSERT INTO 'order_conversion_stats' ('record_id', 'date',
'total_visits', 'total_orders', 'conversion_rate') VALUES
(1, '2025-05-30', 577, 23, 3.99),
(2, '2025-05-29', 184, 17, 9.24),
(3, '2025-05-28', 814, 52, 6.39),
(4, '2025-05-27', 614, 47, 7.65),
(5, '2025-05-26', 321, 23, 7.17),
(6, '2025-05-25', 721, 60, 8.32),
(7, '2025-05-24', 194, 10, 5.15),
(8, '2025-05-23', 360, 28, 7.78),
(9, '2025-05-22', 465, 38, 8.17),

```



```
(10, '2025-05-21', 163, 11, 6.75);
```

- shipment\_performance

```
INSERT INTO 'shipment_performance' ('record_id', 'date', 'carrier',  
'total_shipments', 'on_time_deliveries', 'late_deliveries',  
'average_delivery_time') VALUES
```

```
(1, '2025-05-30', '快递公司 1', 83, 58, 25, 52),
```

```
(2, '2025-05-29', '快递公司 2', 50, 31, 19, 60),
```

```
(3, '2025-05-28', '快递公司 3', 23, 21, 2, 38),
```

```
(4, '2025-05-27', '快递公司 4', 59, 15, 44, 38),
```

```
(5, '2025-05-26', '快递公司 5', 67, 33, 34, 36),
```

```
(6, '2025-05-25', '快递公司 6', 45, 22, 23, 31),
```

```
(7, '2025-05-24', '快递公司 7', 31, 25, 6, 72),
```

```
(8, '2025-05-23', '快递公司 8', 82, 26, 56, 28),
```

```
(9, '2025-05-22', '快递公司 9', 95, 36, 59, 22),
```

```
(10, '2025-05-21', '快递公司 10', 83, 71, 12, 38);
```

- customer\_feedback\_stats

```
INSERT INTO 'customer_feedback_stats' ('record_id', 'date',  
'product_id', 'total_reviews', 'average_rating', 'positive_reviews',  
'negative_reviews') VALUES
```

```
(1, '2025-05-30', 9, 122, 3.9, 122, 0),
```

```
(2, '2025-05-29', 8, 77, 4.78, 42, 35),
```

```
(3, '2025-05-28', 8, 91, 3.52, 52, 39),
```

```
(4, '2025-05-27', 2, 158, 4.74, 82, 76),
```

```
(5, '2025-05-26', 7, 53, 4.8, 23, 30),
```

```
(6, '2025-05-25', 8, 47, 4.87, 16, 31),
```

```
(7, '2025-05-24', 10, 138, 4.64, 133, 5),  
(8, '2025-05-23', 1, 111, 4.6, 33, 78),  
(9, '2025-05-22', 2, 88, 4.81, 44, 44),  
(10, '2025-05-21', 1, 97, 3.01, 26, 71);
```

## 7.6 訂單管理（楊峻朋）

- orders

```
INSERT INTO 'orders' ('order_id', 'customer_id', 'order_status',  
'total_amount', 'created_at', 'updated_at', 'shipping_fee',  
'discount', 'payment_id', 'shipping_id', 'coupon_id') VALUES  
(1, 1, 'cancelled', 954.71, '2025-05-30', '2025-05-30', 109.44,  
163.74, 1, 1, NULL),  
(2, 4, 'cancelled', 907.94, '2025-05-29', '2025-05-29', 73.14, 41.07,  
2, 2, NULL),  
(3, 1, 'cancelled', 1081.47, '2025-05-28', '2025-05-28', 110.21,  
133.98, 3, 3, NULL),  
(4, 7, 'shipped', 1307.22, '2025-05-27', '2025-05-27', 73.11, 191.48,  
4, 4, NULL),  
(5, 1, 'cancelled', 2449.96, '2025-05-26', '2025-05-26', 145.2,  
141.51, 5, 5, NULL),  
(6, 1, 'delivered', 1932.85, '2025-05-25', '2025-05-25', 24.55,  
184.24, 6, 6, NULL),  
(7, 2, 'shipped', 1500.12, '2025-05-24', '2025-05-24', 37.35, 154.11,  
7, 7, NULL),  
(8, 9, 'cancelled', 2939.06, '2025-05-23', '2025-05-23', 89.62,  
110.93, 8, 8, NULL),  
(9, 10, 'pending', 2915.59, '2025-05-22', '2025-05-22', 79.98, 56.87,  
9, 9, NULL),
```

```
(10, 2, 'pending', 2228.16, '2025-05-21', '2025-05-21', 13.22,  
179.38, 10, 10, NULL);
```

- order\_items

```
INSERT INTO 'order_items' ('order_item_id', 'order_id', 'product_id',  
'quantity', 'unit_price') VALUES  
(1.0, 1.0, 3.0, 3.0, 696.66),  
(2.0, 2.0, 6.0, 5.0, 141.18),  
(3.0, 3.0, 10.0, 3.0, 544.23),  
(4.0, 4.0, 8.0, 1.0, 337.1),  
(5.0, 5.0, 2.0, 3.0, 141.09),  
(6.0, 6.0, 3.0, 4.0, 480.92),  
(7.0, 7.0, 5.0, 3.0, 709.42),  
(8.0, 8.0, 8.0, 4.0, 337.28),  
(9.0, 9.0, 5.0, 4.0, 368.56),  
(10.0, 10.0, 2.0, 5.0, 786.05);
```

- payments

```
INSERT INTO 'payments' ('payment_id', 'payment_method',  
'payment_status', 'amount', 'payment_date') VALUES  
(1, 'bank_transfer', 'pending', 1358.29, '2025-05-30'),  
(2, 'credit_card', 'completed', 2814.99, '2025-05-29'),  
(3, 'paypal', 'pending', 1728.38, '2025-05-28'),  
(4, 'paypal', 'pending', 917.84, '2025-05-27'),  
(5, 'cash_on_delivery', 'pending', 809.96, '2025-05-26'),  
(6, 'cash_on_delivery', 'completed', 1843.58, '2025-05-25'),  
(7, 'paypal', 'completed', 2597.12, '2025-05-24'),
```

```
(8, 'cash_on_delivery', 'completed', 1932.53, '2025-05-23'),  
(9, 'credit_card', 'failed', 2589.92, '2025-05-22'),  
(10, 'cash_on_delivery', 'completed', 2579.3, '2025-05-21');
```

● return\_refunds

```
INSERT INTO 'return_refunds' ('refund_id', 'order_id', 'product_id',  
'reason', 'status', 'refund_amount', 'created_at') VALUES  
(1, 1, 5, '原因 1', 'rejected', 589.6, '2025-05-30'),  
(2, 2, 9, '原因 2', 'processed', 657.86, '2025-05-29'),  
(3, 3, 9, '原因 3', 'requested', 256.1, '2025-05-28'),  
(4, 4, 8, '原因 4', 'approved', 917.24, '2025-05-27'),  
(5, 5, 9, '原因 5', 'approved', 570.47, '2025-05-26'),  
(6, 6, 3, '原因 6', 'processed', 236.95, '2025-05-25'),  
(7, 7, 4, '原因 7', 'rejected', 770.41, '2025-05-24'),  
(8, 8, 5, '原因 8', 'rejected', 255.84, '2025-05-23'),  
(9, 9, 5, '原因 9', 'requested', 459.53, '2025-05-22'),  
(10, 10, 6, '原因 10', 'requested', 981.95, '2025-05-21');
```

## 8. 權限管理與 View

本系統實作了詳細的權限管理機制，針對七種不同角色提供相應的資料存取權限。完整的 View SQL 語法、截圖示例與詳細說明請參考：

<https://github.com/EZ-Super/Database-System-Course-Final-Project/blob/main/File/View.md>

### 8.1 角色定義

- Admin (系統管理員)：擁有完整系統權限
- Seller (賣家)：管理自己的商品與訂單
- Customer (顧客)：查看個人資料與購買記錄
- Warehouse (倉儲人員)：管理庫存與物流
- Finance (財務人員)：處理付款與財務報表
- Marketing (行銷人員)：分析銷售數據與客戶行為
- Support (客服人員)：協助客戶問題處理

### 8.2 權限表

- membership\_levels

**membership\_levels** 欄位可視權限表

欄位名稱	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
level_id	✓	✓	✓	X	X	✓	✓
level_name	✓	✓	✓	X	✓	✓	✓
required_points	✓	✓	✓	X	X	✓	✓
discount_rate	✓	✓	✓	X	✓	✓	✓
created_at	✓	X	X	X	X	✓	X
updated_at	✓	X	X	X	X	✓	X

- users(使用者)

**users** 欄位可視權限表

欄位名稱	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
user_id	✓	✓	X	X	✓	✓	✓
username	✓	✓	✓	X	✓	✓	✓
email	✓	✓	✓	X	✓	✓	✓
phone	✓	✓	✓	X	✓	X	✓
registration_date	✓	X	✓	X	X	✓	✓
is_active	✓	✓	X	X	X	X	✓
membership_level_id	✓	X	✓	X	✓	✓	✓
shipping_address	✓	✓	✓	X	X	X	X
billing_address	✓	X	✓	X	✓	X	X

- sellers

**sellers** 欄位可視權限表

欄位名稱	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
seller_id	✓	✓	X	X	✓	✓	X
user_id	✓	✓	X	X	✓	✓	X
store_name	✓	✓	✓	X	X	✓	✓
store_description	✓	✓	✓	X	X	✓	✓
bank_account	✓	✓	X	X	✓	X	X
created_at	✓	✓	X	X	✓	✓	X
updated_at	✓	✓	X	X	✓	✓	X

- points

**points** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
point_id	✓	X	✓	X	✓	X	✓
user_id	✓	X	✓	X	✓	X	✓
points_earned	✓	X	✓	X	✓	X	✓
transaction_date	✓	X	✓	X	✓	X	✓
description	✓	X	✓	X	X	X	✓

- coupons

**coupons** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
coupon_id	✓	X	X	X	X	✓	X
coupon_code	✓	X	✓	X	X	✓	X
discount_amount	✓	X	✓	X	✓	✓	X
expiry_date	✓	X	✓	X	X	✓	X
usage_limit	✓	X	X	X	X	✓	X
is_active	✓	X	X	X	X	✓	X
coupon_type	✓	X	X	X	X	✓	X
created_at	✓	X	X	X	X	✓	X

- messages

**messages** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
message_id	✓	✓	✓	X	X	X	✓
sender_id	✓	✓	✓	X	X	X	✓
receiver_id	✓	✓	✓	X	X	X	✓
message_content	✓	✓	✓	X	X	X	✓
send_time	✓	✓	✓	X	X	X	✓
is_read	✓	✓	✓	X	X	X	✓

- notifications

**notifications** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
notification_id	✓	X	✓	X	X	✓	✓
user_id	✓	X	✓	X	X	✓	✓
notification_type	✓	X	✓	X	X	✓	✓
subject	✓	X	✓	X	X	✓	✓
message	✓	X	✓	X	X	✓	✓
sent_at	✓	X	✓	X	X	✓	✓
is_read	✓	X	✓	X	X	X	✓
related_entity	✓	X	✓	X	X	✓	✓
related_entity_id	✓	X	✓	X	X	✓	✓

- warehouses

**warehouses** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
warehouse_id	✓	X	X	✓	X	X	X
warehouse_name	✓	X	X	✓	X	X	X
location	✓	X	X	✓	X	X	X
capacity	✓	X	X	✓	X	X	X
manager_id	✓	X	X	✓	X	X	X
contact_info	✓	X	X	✓	X	X	X
created_at	✓	X	X	✓	X	X	X

- inventory

**inventory** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
inventory_id	✓	✓	X	✓	X	X	X
warehouse_id	✓	✓	X	✓	X	X	X
product_id	✓	✓	X	✓	X	X	X
stock_quantity	✓	✓	X	✓	X	X	X
reorder_level	✓	✓	X	✓	X	X	X
last_updated	✓	✓	X	✓	X	X	X

- suppliers

**suppliers** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
supplier_id	✓	✓	X	✓	X	X	X
supplier_name	✓	✓	X	✓	X	X	X
contact_info	✓	✓	X	✓	X	X	X
country	✓	✓	X	✓	X	X	X
rating	✓	✓	X	✓	X	X	X
created_at	✓	✓	X	✓	X	X	X



- inboundShipments

**inboundShipments** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
inbound_id	✓	✓	X	✓	X	X	X
supplier_id	✓	✓	X	✓	X	X	X
warehouse_id	✓	✓	X	✓	X	X	X
product_id	✓	✓	X	✓	X	X	X
quantity	✓	✓	X	✓	X	X	X
arrival_date	✓	✓	X	✓	X	X	X
status	✓	✓	X	✓	X	X	X

- outboundShipments

**outboundShipments** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
outbound_id	✓	✓	X	✓	X	X	X
order_id	✓	✓	✓	✓	✓	X	✓
warehouse_id	✓	✓	X	✓	X	X	X
product_id	✓	✓	X	✓	X	X	X
quantity	✓	✓	X	✓	X	X	X
dispatch_date	✓	✓	X	✓	X	X	X
status	✓	✓	X	✓	X	X	X

- shipments

**shipments** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
shipment_id	✓	✓	✓	✓	X	X	✓
order_id	✓	✓	✓	✓	X	X	✓
tracking_number	✓	✓	✓	✓	X	X	✓
shipment_status	✓	✓	✓	✓	X	X	✓
estimated_delivery	✓	✓	✓	✓	X	X	✓
actual_delivery	✓	✓	✓	✓	X	X	✓
carrier	✓	✓	✓	✓	X	X	✓
shipping_cost	✓	✓	X	✓	✓	X	✓

- warehouse\_Transfers

**warehouse\_Transfers** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
transfer_id	✓	X	X	✓	X	X	X
from_warehouse_id	✓	X	X	✓	X	X	X
to_warehouse_id	✓	X	X	✓	X	X	X
product_id	✓	X	X	✓	X	X	X
quantity	✓	X	X	✓	X	X	X
transfer_status	✓	X	X	✓	X	X	X
transfer_date	✓	X	X	✓	X	X	X

- sales\_Analysis

**sales\_Analysis** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
record_id	✓	✓	X	X	X	✓	X
date	✓	✓	X	X	X	✓	X
product_id	✓	✓	X	X	X	✓	X
category_id	✓	✓	X	X	X	✓	X
total_sales	✓	✓	X	X	X	✓	X
revenue	✓	✓	X	X	X	✓	X
average_price	✓	✓	X	X	X	✓	X

- inventory\_Analytics

**inventory\_Analytics** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
record_id	✓	✓	X	✓	X	✓	X
date	✓	✓	X	✓	X	✓	X
warehouse_id	✓	✓	X	✓	X	✓	X
product_id	✓	✓	X	✓	X	✓	X
starting_stock	✓	✓	X	✓	X	✓	X
ending_stock	✓	✓	X	✓	X	✓	X
sold_units	✓	✓	X	✓	X	✓	X
received_units	✓	✓	X	✓	X	✓	X

- order\_Conversion\_Stats

**order\_Conversion\_Stats** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
record_id	✓	X	X	X	X	✓	X
date	✓	X	X	X	X	✓	X
total_visits	✓	X	X	X	X	✓	X
total_orders	✓	X	X	X	X	✓	X
conversion_rate	✓	X	X	X	X	✓	X

- shipment\_Performance

**shipment\_Performance** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
record_id	✓	X	X	✓	X	✓	X
date	✓	X	X	✓	X	✓	X
carrier	✓	X	X	✓	X	✓	X
total_shipments	✓	X	X	✓	X	✓	X
on_time_deliveries	✓	X	X	✓	X	✓	X
late_deliveries	✓	X	X	✓	X	✓	X
average_delivery_time	✓	X	X	✓	X	✓	X

- coustomer\_Feedback\_Stats

**customer\_Feedback\_Stats** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
record_id	✓	✓	X	X	X	✓	✓
date	✓	✓	X	X	X	✓	✓
product_id	✓	✓	X	X	X	✓	✓
total_reviews	✓	✓	X	X	X	✓	✓
average_rating	✓	✓	X	X	X	✓	✓
positive_reviews	✓	✓	X	X	X	✓	✓
negative_reviews	✓	✓	X	X	X	✓	✓

- products

**products** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
product_id	✓	✓	✓	✓	X	✓	✓
product_name	✓	✓	✓	✓	X	✓	✓
sku	✓	✓	X	✓	X	✓	✓
brand	✓	✓	✓	✓	X	✓	✓
model	✓	✓	✓	✓	X	✓	✓
description	✓	✓	✓	✓	X	✓	✓
category_id	✓	✓	✓	✓	X	✓	✓
variant_type	✓	✓	✓	✓	X	✓	✓
price	✓	✓	✓	✓	✓	✓	✓
promotional_price	✓	✓	✓	✓	✓	✓	✓
promotion_start_date	✓	✓	✓	✓	X	✓	✓
promotion_end_date	✓	✓	✓	✓	X	✓	✓
seller_id	✓	✓	X	X	X	✓	✓
shipping_weight	✓	✓	✓	✓	X	X	X
image_url	✓	✓	✓	✓	X	✓	✓
barcode	✓	✓	X	✓	X	X	X
reviews_count	✓	✓	✓	X	X	✓	✓
favorites_count	✓	✓	✓	X	X	✓	✓
date_added	✓	✓	✓	✓	X	✓	✓
last_updated	✓	✓	✓	✓	X	✓	✓

- reviews

**reviews** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
review_id	✓	✓	✓	X	X	✓	✓
product_id	✓	✓	✓	X	X	✓	✓
user_id	✓	X	✓	X	X	X	✓
rating	✓	✓	✓	X	X	✓	✓
comment	✓	✓	✓	X	X	✓	✓
created_at	✓	✓	✓	X	X	✓	✓

- categories

**categories** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
category_id	✓	✓	✓	✓	X	✓	✓
category_name	✓	✓	✓	✓	X	✓	✓
category_description	✓	✓	✓	✓	X	✓	✓

- login\_logs

**login\_logs** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
log_id	✓	X	X	X	X	X	X
user_id	✓	X	X	X	X	X	X
email	✓	X	X	X	X	X	X
login_time	✓	X	X	X	X	X	X
ip_address	✓	X	X	X	X	X	X
user_agent	✓	X	X	X	X	X	X
success	✓	X	X	X	X	X	X
failure_reason	✓	X	X	X	X	X	X

- users (AD Server)

**users** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
user_id	✓	X	X	X	X	X	X
email	✓	X	X	X	X	X	✓
password_hash	✓	X	X	X	X	X	X
account_enable	✓	X	X	X	X	X	✓
created_at	✓	X	X	X	X	X	✓
failed_login_attempts	✓	X	X	X	X	X	✓
is_verified	✓	X	X	X	X	X	✓
updated_at	✓	X	X	X	X	X	✓

- orders

**orders** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
order_id	✓	✓	✓	✓	✓	X	✓
customer_id	✓	✓	✓	X	✓	X	✓
order_status	✓	✓	✓	✓	✓	X	✓
total_amount	✓	✓	✓	✓	✓	X	✓
created_at	✓	✓	✓	X	✓	X	✓
updated_at	✓	✓	✓	X	✓	X	✓
shipping_fee	✓	✓	✓	✓	✓	X	✓
discount	✓	✓	✓	X	✓	X	✓
payment_id	✓	X	✓	X	✓	X	✓
shipping_id	✓	✓	✓	✓	X	X	✓
coupon_id	✓	✓	✓	X	✓	✓	✓

- order\_items

**order\_items** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
order_item_id	✓	✓	✓	X	✓	X	✓
order_id	✓	✓	✓	X	✓	X	✓
product_id	✓	✓	✓	✓	X	✓	✓
quantity	✓	✓	✓	✓	✓	✓	✓
unit_price	✓	✓	✓	X	✓	✓	✓
subtotal	✓	✓	✓	X	✓	✓	✓

- payments

**payments** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
payment_id	✓	X	✓	X	✓	X	✓
payment_method	✓	X	✓	X	✓	X	✓
payment_status	✓	X	✓	X	✓	X	✓
amount	✓	X	✓	X	✓	X	✓
payment_date	✓	X	✓	X	✓	X	✓

- return\_refunds

**return\_refunds** 欄位可視權限表

欄位	Admin	Seller	Customer	Warehouse	Finance	Marketing	Support
refund_id	✓	✓	✓	X	✓	X	✓
order_id	✓	✓	✓	X	✓	X	✓
product_id	✓	✓	✓	X	X	X	✓
reason	✓	✓	✓	X	X	X	✓
status	✓	✓	✓	X	✓	X	✓
refund_amount	✓	✓	✓	X	✓	X	✓
created_at	✓	✓	✓	X	✓	X	✓

## 8.3 查詢範例

查詢結果截圖不清可至 github 上查看：

<https://github.com/EZ-Super/Database-System-Course-Final-Project/blob/main/File/UsageExample.md>

### 8.3.1 系統管理員查找帳號

- SQL 語法

```
SELECT * FROM admin_user_view WHERE user_id = 1;
```

- 查詢結果

```
MariaDB [shop2]> select * from admin_user_view where user_id = 1;
```

user_id	username	email	phone	registration_date	is_active	membership_level_id	shipping_address	billing_address
1	陳小明	chen.xiaoming@gmail.com	0912345678	2023-01-15 10:30:00	1	1	台北市大安區世尊路四段112號	台北市大安區世尊路四段112號

1 row in set (0.207 sec)

### 8.3.2 使用者查詢賣場

- SQL 語法

```
SELECT store_name, store_description, seller_id, username
FROM customer_seller_view c
JOIN users u ON u.user_id = c.seller_id
WHERE seller_id = 1;
```

- 查詢結果

```
MariaDB [shop2]> select store_name,store_description,seller_id,username from customer_seller_view c join users u on u.user_id = c.seller_id where seller_id = 1;
```

store_name	store_description	seller_id	username
大偉3C數位館	專售手機、平板、電腦週邊商品，品質保證，價格實惠	1	陳小明

### 8.3.3 依賣場名稱查詢

- SQL 語法

```
SELECT store_name, store_description, seller_id, username
FROM customer_seller_view c
JOIN users u ON u.user_id = c.seller_id
WHERE c.store_name = "大偉 3C 數位館";
```

- 查詢結果

```
MariaDB [shop2]> select store_name,store_description,seller_id,username from customer_seller_view c join users u on u.user_id = c.seller_id where c.store_name = "大偉3C數位館";
```

store_name	store_description	seller_id	username
大偉3C數位館	專售手機、平板、電腦週邊商品，品質保證，價格實惠	1	陳小明



### 8.3.4 賣家查詢訂單明細

- SQL 語法

```
SELECT * FROM seller_orders_view WHERE seller_id = 1;
```

- 查詢結果

```
MariaDB [shop2]> select * from seller_orders_view where seller_id = 1;
```

order_id	customer_id	order_status	total_amount	created_at	updated_at	shipping_fee	discount	coupon_id	seller_id
1	1	cancelled	954.71	2025-05-30 00:00:00	2025-05-30 00:00:00	109.44	163.74	NULL	1
6	1	delivered	1932.85	2025-05-25 00:00:00	2025-05-25 00:00:00	24.55	184.24	NULL	1

### 8.3.5 賣家查詢自己賣場銷售前三名

- SQL 語法

```
SELECT  
  
    s.seller_id,  
  
    s.store_name,  
  
    ranked_products.product_id,  
  
    ranked_products.product_name,  
  
    ranked_products.total_sales,  
  
    ranked_products.revenue,  
  
    ranked_products.average_price  
  
FROM (  
  
    SELECT  
  
        p.seller_id,  
  
        p.product_id,  
  
        p.product_name,  
  
        sa.total_sales,  
  
        sa.revenue,
```

```

        sa.average_price,

        ROW_NUMBER() OVER (PARTITION BY p.seller_id ORDER BY
sa.total_sales DESC) AS sales_rank

    FROM seller_products_view AS p

    JOIN sales_analysis AS sa ON p.product_id = sa.product_id AND
p.seller_id = 2

    WHERE sa.total_sales > 0

) AS ranked_products

JOIN sellers AS s ON s.seller_id = ranked_products.seller_id

WHERE ranked_products.sales_rank <= 3;

```

## ● 查詢結果

```

MariaDB [shop2]> SELECT
-> s.seller_id,
-> s.store_name,
-> ranked_products.product_id,
-> ranked_products.product_name,
-> ranked_products.total_sales,
-> ranked_products.revenue,
-> ranked_products.average_price
-> FROM (
-> SELECT
-> p.seller_id,
-> p.product_id,
-> p.product_name,
-> sa.total_sales,
-> sa.revenue,
-> sa.average_price,
-> ROW_NUMBER() OVER (PARTITION BY p.seller_id ORDER BY sa.total_sales DESC) AS sales_rank
-> FROM seller_products_view AS p
-> JOIN sales_analysis AS sa ON p.product_id = sa.product_id AND p.seller_id = 2
-> WHERE sa.total_sales > 0
-> ) AS ranked_products
-> JOIN sellers AS s ON s.seller_id = ranked_products.seller_id
-> WHERE ranked_products.sales_rank <= 3;

```

seller_id	store_name	product_id	product_name	total_sales	revenue	average_price
2	志豪文創小舖	5	筆電支架	93	1503.09	470.06
2	志豪文創小舖	10	智慧手錶	76	5378.29	237.54
2	志豪文創小舖	2	機械式鍵盤	31	4570.76	760.73

### 8.3.6 賣家依照賣場銷售平均價格前三名排序

- SQL 語法

```
SELECT

    s.seller_id,

    s.store_name,

    ranked_products.product_id,

    ranked_products.product_name,

    ranked_products.total_sales,

    ranked_products.revenue,

    ranked_products.average_price

FROM (

    SELECT

        p.seller_id,

        p.product_id,

        p.product_name,

        sa.total_sales,

        sa.revenue,

        sa.average_price,

        ROW_NUMBER() OVER (PARTITION BY p.seller_id ORDER BY
sa.total_sales DESC) AS sales_rank

    FROM seller_products_view AS p

    JOIN sales_analysis AS sa ON p.product_id = sa.product_id AND
p.seller_id = 2

    WHERE sa.total_sales > 0

) AS ranked_products
```

```
JOIN sellers AS s ON s.seller_id = ranked_products.seller_id  
WHERE ranked_products.sales_rank <= 3;
```

● 查詢結果

```
MariaDB [shop2]>  
MariaDB [shop2]> SELECT  
-> s.seller_id,  
-> s.store_name,  
-> ranked_products.product_id,  
-> ranked_products.product_name,  
-> ranked_products.total_sales,  
-> ranked_products.revenue,  
-> ranked_products.average_price  
-> FROM (  
-> SELECT  
-> p.seller_id,  
-> p.product_id,  
-> p.product_name,  
-> sa.total_sales,  
-> sa.revenue,  
-> sa.average_price,  
-> ROW_NUMBER() OVER (PARTITION BY p.seller_id ORDER BY sa.average_price DESC) AS sales_rank  
-> FROM seller_products_view AS p  
-> JOIN sales_analysis AS sa ON p.product_id = sa.product_id AND p.seller_id = 2  
-> WHERE sa.total_sales > 0  
-> ) AS ranked_products  
-> JOIN sellers AS s ON s.seller_id = ranked_products.seller_id  
-> WHERE ranked_products.sales_rank <= 3;
```

seller_id	store_name	product_id	product_name	total_sales	revenue	average_price
2	志豪文創小舖	2	機械式鍵盤	31	4570.76	760.73
2	志豪文創小舖	5	筆電支架	93	1503.09	470.06
2	志豪文創小舖	10	智慧手錶	76	5378.29	237.54

### 8.3.7 賣家查找自己庫存小於3的商品

- SQL 語法

```
SELECT

    p.product_id,

    p.product_name,

    i.ending_stock,

    p.seller_id

FROM seller_inventory_analytics_view AS i

JOIN seller_products_view AS p ON i.product_id = p.product_id

WHERE p.seller_id = 4 AND

    i.ending_stock < 3

ORDER BY i.ending_stock ASC;
```

- 查詢結果

```
MariaDB [shop2]> SELECT
->     p.product_id,
->     p.product_name,
->     i.ending_stock,
->     p.seller_id
-> FROM seller_inventory_analytics_view  AS i
-> JOIN seller_products_view  AS p ON i.product_id = p.product_id
-> WHERE  p.seller_id = 4 AND
-> i.ending_stock < 3
-> ORDER BY i.ending_stock ASC;
```

product_id	product_name	ending_stock	seller_id
6	外接固態硬碟	2	4

### 8.3.8 賣家查詢某時段自己某商品的銷售數量

- SQL 語法

```
SELECT

    i.record_id,

    i.date,

    s.seller_id,

    ss.store_name,

    i.sold_units

FROM inventory_analytics AS i

JOIN seller_products_view AS s ON i.product_id = s.product_id

JOIN seller_seller_view AS ss ON s.seller_id = ss.seller_id

WHERE i.date BETWEEN '2025-05-01' AND '2025-05-30' AND i.product_id = 4;
```

- 查詢結果

```
MariaDB [shop2]> SELECT
->     i.record_id,
->     i.date,
->     s.seller_id,
->     ss.store_name,
->     i.sold_units
->
-> FROM inventory_analytics AS i
-> JOIN seller_products_view AS s ON i.product_id = s.product_id
-> JOIN seller_seller_view AS ss ON s.seller_id = ss.seller_id
-> WHERE i.date BETWEEN '2025-05-01' AND '2025-05-30' AND i.product_id = 4;
```

record_id	date	seller_id	store_name	sold_units
1	2025-05-30	3	建國運動用品	54
4	2025-05-27	3	建國運動用品	57

### 8.3.9 倉儲人員查詢訂單

- SQL 語法

```
SELECT * FROM warehouse_orders_view WHERE order_id = 1;
```

- 查詢結果

```
MariaDB [shop2]> select * from warehouse_orders_view where order_id = 1;
```

order_id	order_status	total_amount	shipping_fee	shipping_id
1	cancelled	954.71	109.44	1

### 8.3.10 財務人員查看退款的訂單

- SQL 語法

```
SELECT r.refund_id, r.order_id, r.product_id, r.status,  
r.refund_amount, r.created_at, p.product_name, p.seller_id,  
u.username  
FROM finance_return_refunds_view r  
JOIN products p ON p.product_id = r.product_id  
JOIN users u ON u.user_id = r.seller_id  
WHERE r.order_id = 1;
```

- 查詢結果

```
MariaDB [shop2]> select f.refund_id,f.order_id,f.product_id,f.status,f.refund_amount,f.created_at,p.product_name,p.seller_id,u.username from finance_return_refunds_view f join products p ON p.product_id = f.pr  
duct_id join users u ON u.user_id = p.seller_id where f.order_id = 1;
```

refund_id	order_id	product_id	status	refund_amount	created_at	product_name	seller_id	username
1	1	5	rejected	589.68	2025-05-30 00:00:00	華電支所	2	林雅婷

1 row in set (0.286 sec)

### 8.3.11 行銷人員查詢銷售分析

- SQL 語法

```
SELECT * FROM marketing_products_view WHERE product_id = 1;
```

- 查詢結果

```
MariaDB [shop2]> select * from marketing_products_view where product_id = 1;
```

product_id	product_name	brand	category_id	price	promotional_price	reviews_count	favorites_count	date_added
1	無線滑鼠	Logitech	1	599.00	499.00	23	15	2025-05-30 10:00:00

### 8.3.12 客服人員查詢評價

- SQL 語法

```
SELECT s.record_id, s.date, s.product_id, s.total_reviews,  
s.average_rating, p.product_name, p.seller_id, u.username  
FROM support_feedback_stats_view s  
JOIN products p ON p.product_id = s.product_id  
JOIN users u ON u.user_id = p.seller_id  
WHERE s.product_id = 1;
```

- 查詢結果

```
MariaDB [shop2]> select s.record_id,s.date,s.product_id,s.total_reviews,s.average_rating,p.product_name ,p.seller_id,u.username from support_feedback_stats_view s join products p on s.product_id = p.product_id  
join users u on u.user_id = p.seller_id where s.product_id = 8;
```

record_id	date	product_id	total_reviews	average_rating	product_name	seller_id	username
2	2025-05-29	8	77	4.78	攝影腳架	5	李志豪
3	2025-05-28	8	91	3.52	攝影腳架	5	李志豪
6	2025-05-25	8	47	4.07	攝影腳架	5	李志豪



### 8.3.13 查詢某快遞公司的運送準時率

- SQL 語法

```
SELECT

    w.carrier,

    w.total_shipments,

    w.on_time_deliveries,

    w.late_deliveries,

    ROUND(w.on_time_deliveries / w.total_shipments * 100, 2) AS
on_time_rate_percent,

    ROUND(w.late_deliveries / w.total_shipments * 100, 2) AS
late_rate_percent

FROM warehouse_shipment_performance_view AS w

WHERE w.carrier = '快遞公司1';
```

- 查詢結果

```
MariaDB [shop2]> SELECT
-> w.carrier,
-> w.total_shipments,
-> w.on_time_deliveries,
-> w.late_deliveries,
-> ROUND(w.on_time_deliveries / w.total_shipments * 100, 2) AS on_time_rate_percent,
-> ROUND(w.late_deliveries / w.total_shipments * 100, 2) AS late_rate_percent
-> FROM warehouse_shipment_performance_view AS w
-> WHERE w.carrier = '快遞公司1';
```

carrier	total_shipments	on_time_deliveries	late_deliveries	on_time_rate_percent	late_rate_percent
快遞公司1	83	58	25	69.88	30.12

### 8.3.14 查詢某賣家好評高於某數值的商品

- SQL 語法

```
SELECT

    s.product_id,

    p.seller_id,

    s.total_reviews,

    s.average_rating,

    s.positive_reviews,

    s.negative_reviews,

    ROUND(s.positive_reviews / s.total_reviews * 100, 2) AS
positive_review_rate,

    ROUND(s.negative_reviews / s.total_reviews * 100, 2) AS
negative_review_rate

FROM seller_feedback_stats_view AS s

JOIN seller_products_view AS p ON s.product_id = p.product_id

WHERE p.seller_id = 1 AND (s.positive_reviews / s.total_reviews *
100) > 40;
```

- 查詢結果

```
MariaDB [shop2]> SELECT
->   s.product_id,
->   p.seller_id,
->   s.total_reviews,
->   s.average_rating,
->   s.positive_reviews,
->   s.negative_reviews,
->   ROUND(s.positive_reviews / s.total_reviews * 100, 2) AS positive_review_rate,
->   ROUND(s.negative_reviews / s.total_reviews * 100, 2) AS negative_review_rate
-> FROM seller_feedback_stats_view AS s
-> JOIN seller_products_view AS p ON s.product_id = p.product_id
-> WHERE p.seller_id = 1 AND (s.positive_reviews / s.total_reviews * 100) > 40;
```

product_id	seller_id	total_reviews	average_rating	positive_reviews	negative_reviews	positive_review_rate	negative_review_rate
7	1	53	4.80	23	30	43.40	56.60
9	1	122	3.90	122	0	100.00	0.00

## 9. 完成介面展示

- 部分完成介面成果展示：<https://yupi.ethidra.org/>

- 可驗證的帳戶：

帳號：test@example.com

密碼：@Test894523

- 介面分工

介面	前端	後端
未登入時的首頁	楊峻朋	楊峻朋
登入介面	楊峻朋	楊峻朋
買家個人檔案介面	謝延偵	楊峻朋
會員等級與點數介面	謝延偵	楊峻朋

## 9.1 未登入時的首頁

 魚皮購物

Home 登入



### 歡迎來到魚皮購物

全館 0 元購，限時搶購!!!

項目	數值
已成交單數	12,345
目前線上人數	678
總成交金額	12,345

#### 每月銷售統計

銷售額 (萬元)



月份	銷售額 (萬元)
一月	10
二月	18
三月	8
四月	15
五月	12

#### 每月登入統計

登入總數 (萬)



月份	登入總數 (萬)
一月	10
二月	20
三月	8
四月	15
五月	12
六月	10
七月	20
八月	8
九月	15
十月	12
十一月	10
十二月	20

#### 客服中心

- 幫助中心
- 購物教學
- 付款方式
- 運送服務
- 退貨政策
- 聯絡我們

#### 關於我們

- 關於魚皮
- 加入我們
- 隱私權政策
- 賣家中心

#### 付款

全平台免付費

#### 物流合作

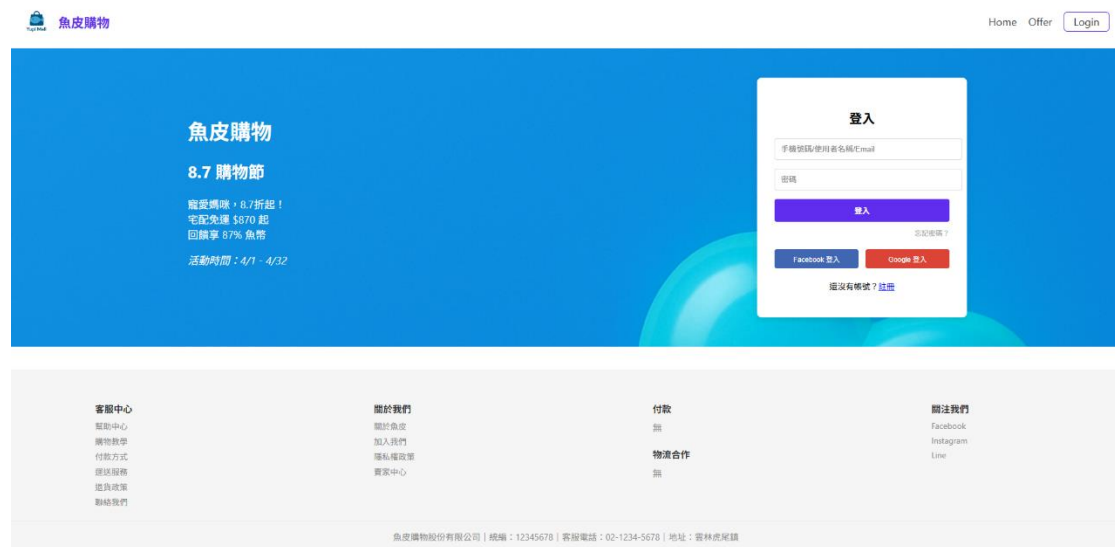
黑狗宅急便

#### 關注我們

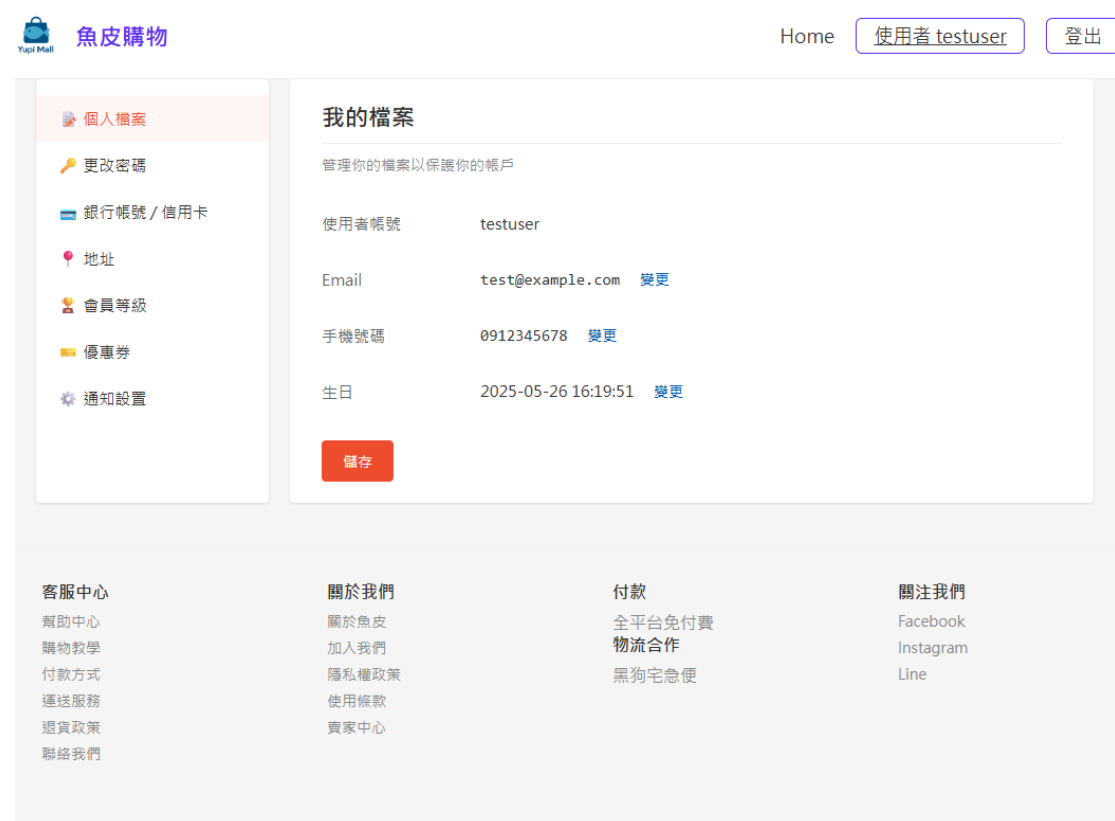
- Facebook
- Instagram
- Line

© 2025 魚皮購物股份有限公司 版權所有 | 統一編號：12345678


## 9.2 登入介面



## 9.3 買家個人檔案介面



## 9.4 會員等級與點數介面

 魚皮購物

Home 使用者 testuser 登出

個人檔案

更改密碼

銀行帳號 / 信用卡

地址

**會員等級**

優惠券

通知設置

會員等級

銅魚

銅魚會員

等級說明

距離下一等級還需：\$89  
本期累積消費：\$11/6000  
完成進度以解鎖下一等級！  
2025.06.30 重新計算，依前 6 個月消費予以對應會員等級

點數歷程

日期	類型	描述	點數變動	點數餘額
2025-05-26 17:05:01	購物	hello	+10	10
2025-05-26 17:42:26	購物	Hello2	+1	1

客服中心  
幫助中心  
購物教學  
付款方式  
運送服務  
退貨政策  
聯絡我們

關於我們  
關於魚皮  
加入我們  
隱私權政策  
使用條款  
賣家中心

付款  
全平台免付費  
物流合作  
黑狗宅急便

關注我們  
Facebook  
Instagram  
Line

## 10. 結論與心得

### 10.1 專題成果總結

本專題完成了一個全方位電商平台資料庫系統的設計與規劃，涵蓋了現代電商平台所需的核​​心功能模組。透過完整的需求分析、ER 圖設計、資料庫 Schema 建置以及詳細的權限管理機制設計，完成了一個結構完整且具備實用性的資料庫系統架構。

主要完成項目：

- 完整的資料庫設計：包含用戶管理、商品管理、訂單處理、物流倉儲、數據分析等五大核心模組的資料表設計
- 分離式安全架構設計：設計認證資料與業務資料隔離的架構方案
- 權限管理機制：規劃七種不同角色的權限體系與 View 設計
- 測試資料建置：建立完整的測試資料集驗證系統設計可行性

本專題展示了完整的資料庫系統設計流程，從需求分析、ER 圖設計到 Schema 實作，每個階段都遵循資料庫設計的理論與實務。所設計的資料庫架構具有良好的擴展性和維護性，可作為實際電商平台開發的參考基礎。雖然僅止於資料庫層面的設計，但為後續的系統開發提供了堅實的資料基礎架構。

## 10.2 個人心得

### 10.2.1 楊峻朋

這次的專題學習到如何規劃一個購物平台的資料庫系統，以及所有欄位的完整性限制，還有權限跟 view 的規劃。目的是為了實現電子商務網站背後的資料管理機制，從資料表設計到資料操作、完整性限制、實際資料視圖，每一個階段都是一次學習經驗。一開始，我們從需求分析著手，釐清購物平台中應該具備哪些功能，如：會員註冊與登入、商品上架與分類、購物車管理、訂單系統、付款與出貨流程等，更要站在系統整體運作的角度思考資料的關聯。在資料表設計上，我們特別注重「正規化」原則，避免資料重複與異常，並利用外來鍵（FOREIGN KEY）來維護資料的參照完整性，也透過限制輸入來避免 xss、sql 注入等漏洞。

### 10.2.2 謝延偵

這次專題我們是按照模組來分工的，一開始覺得很簡單，大家各做各的就好了，沒想到後來要整合的時候才發現有多麻煩。當開始彙整各模組的設計時，才發現資料統一是一件非常困難的事情，花費了大量的時間進行協調。

例如欄位的命名方式，每個人的習慣都不一樣：有人偏好使用大寫命名，有人習慣底線命名法，甚至在資料型態的選擇上，每個人和各自使用的 AI 工具都有不同的見解和建議。這讓我們真的體會到，如果一開始沒有訂好規則，後面整合和使用上會有多痛苦。

在 ER 圖設計階段，剛開始對於如何適當地分類各種資料物件感到困惑，畫出來的圖會變得非常龐大。雖然上課有學到相關概念，但在實際應用時，並不是一看到某個東西就能立刻判斷它應該如何分類。透過 AI 的協助和指導，才學會如

何在實際情境中正確地進行這些判斷，讓整個 ER 圖設計變得更加清晰和合理。

在用戶管理模組的開發過程中，我學會了如何將抽象的業務需求轉換為具體的資料模型。特別是在設計會員等級系統時，需要仔細考慮如何在系統彈性與資料一致性之間取得平衡。雖然權限管理 View 的設計不是我直接負責的部分，但透過團隊協作，我也理解了資料安全的重要性，以及如何透過資料庫層級的權限控制來保護敏感資訊。

這次專題最大的收穫是學會了團隊協作的重要性。每個人負責不同的模組，但最終需要整合成一個完整的系統，這個過程讓我們深刻理解溝通協調的重要性，以及建立統一標準的必要性。

### 10.2.3 鍾文瑤

經過這次的專題，我對 SQL 指令的應用有了更深入的認識。以往僅從課堂上學習語法，直到實際設計資料庫時才發現，資料表之間的關聯、正規化處理、權限控管與查詢效率等問題遠比想像中複雜。在專題中，我們以角色分工方式規劃資料庫，並設計對應的 SQL View，讓每種使用者只存取他們所需的資訊，這不僅提升了系統安全性，也改善了資料使用效率。我負責了與倉儲與庫存有關的模組，從中了解了如何管理不同倉庫的庫存、追蹤商品流向與處理入出庫紀錄，對資料一致性與實務流程設計有更深體會。這次經驗讓我了解，建構一個完整的系統資料庫不只是技術堆疊，更是邏輯與溝通的結合。資料庫是整個系統運作的核心，未來若要做出更大型或更穩定的應用，這些基礎都將成為重要養分。

### 10.2.4 李維誠

做了這份期末報告，讓我更了解了資料庫系統這門課，也讓我知道了可以應用在哪些東西上，原本以為會跟我們的生活沒有關，但做了才知道我們生活中的許多地方都能去使用到，例如飲料商家、網路購物和一些生活周遭的地方都有蹤跡，也聽了許多組的報告讓我去了解一些我沒有想到的地方也有它的蹤跡，讓我能夠更好地去理解與使用。



專題完成日期：2024 年 6 月 15 日

團隊成員：楊峻朋、謝延偵、鍾文瑤、李維誠

課程名稱：資料庫系統

## 11. 參考資料

- 專題 github 連結：

<https://github.com/EZ-Super/Database-System-Course-Final-Project/tree/main>

- 參考網站與平台：

- PChome：<https://24h.pchome.com.tw/>
- 蝦皮購物：<https://shopee.tw/>

- AI 工具：

- <https://claude.ai/>
- <https://chatgpt.com/>

- ER 圖繪製工具：

<https://online.visual-paradigm.com/tw/diagrams/features/erd-tool/>

- 其餘細節部分的參考資料請至專題 github 上查看。