簡報日期：2025/06/04（三）

資料庫系統

E-Commerce
電商平台資料庫設計

第三組:

陳荔群 41043218

曾聖傑 41043220

蘇于驊 41043255

梁詠琳 41048110

指導老師:江季翰老師

目錄

題目介紹

本專題旨在設計一個資料庫，用於支援電商平台的基本與進階功能，包括商品管理、訂單處理、顧客評論與使用者角色分級等，並最終以 SQL 進行實作。

應用情境與使用案例

🛒 應用情境：

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：

- 顧客：註冊 / 瀏覽商品 / 下單 / 評價商品
- 賣家：上架商品 / 管理庫存 / 查看訂單
- 管理員：管理用戶 / 商品 / 訂單狀態

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

系統需求說明

Customer（買家）

欄位名稱	完整性限制說明
customer_id	主鍵。必須唯一且不可為空， 用於唯一識別每位買家。
name	不可為空。 應包含至少一個非空白字元。
email	必須唯一且不可為空。 格式需為有效 Email（包含一個 @ 字元及有效網域名稱，如 user@example.com）。 不得出現空白或無效格式。

系統需求說明

Seller（賣家）

欄位名稱	完整性限制說明
seller_id	主鍵。 必須唯一且不可為空。
name	不可為空， 需為具辨識度的名稱。
email	必須唯一且不可為空。 格式限制同 Customer.email。

系統需求說明

Admin（管理員）

欄位名稱	完整性限制說明
admin_id	主鍵。 必須唯一且不可為空。
username	必須唯一且不可為空。應限制為英數混合字元，不可包含空格或特殊符號。

系統需求說明

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。需具明確辨識意義。
price	必須為正數，且大於 0， 不可為負數或零。可為整數或兩位小數。
stock	必須為整數，且不可為負數。
category	可選欄位。若填寫， 應屬於預先定義的分類集合（如 "electronics"、"books" 等）， 避免任意輸入造成資料雜亂。
seller_id	外鍵。需參照 Seller.seller_id， 不可為空，且必須對應至現有賣家。

系統需求說明

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	必須為合理有效的日期。 格式為 YYYY-MM-DD， 日期不得晚於當前系統時間， 亦不得早於系統啟用日（避免系統錯誤）。
status	不可為空。限定值域為下列四種狀態之一： 「未付款」、「處理中」、「已出貨」、「已完成」。 可使用 ENUM 或 CHECK 條件強制限制。
customer_id	外鍵。需參照 Customer.customer_id， 不可為空，且必須對應至現有買家。

系統需求說明

OrderItem（訂單項目）

欄位名稱	完整性限制說明
order_id	外鍵。需存在於 Order.order_id 中。
product_id	外鍵。需存在於 Product.product_id 中。
(order_id, product_id)	複合主鍵。此組合必須唯一且不可為空。
quantity	必須為正整數（≥ 1）， 不得為負數或零， 且不得超過商品目前庫存數。
price_at_purchase	價格需大於 0， 保存購買當下的商品價格。 此價格不得為負數或零。

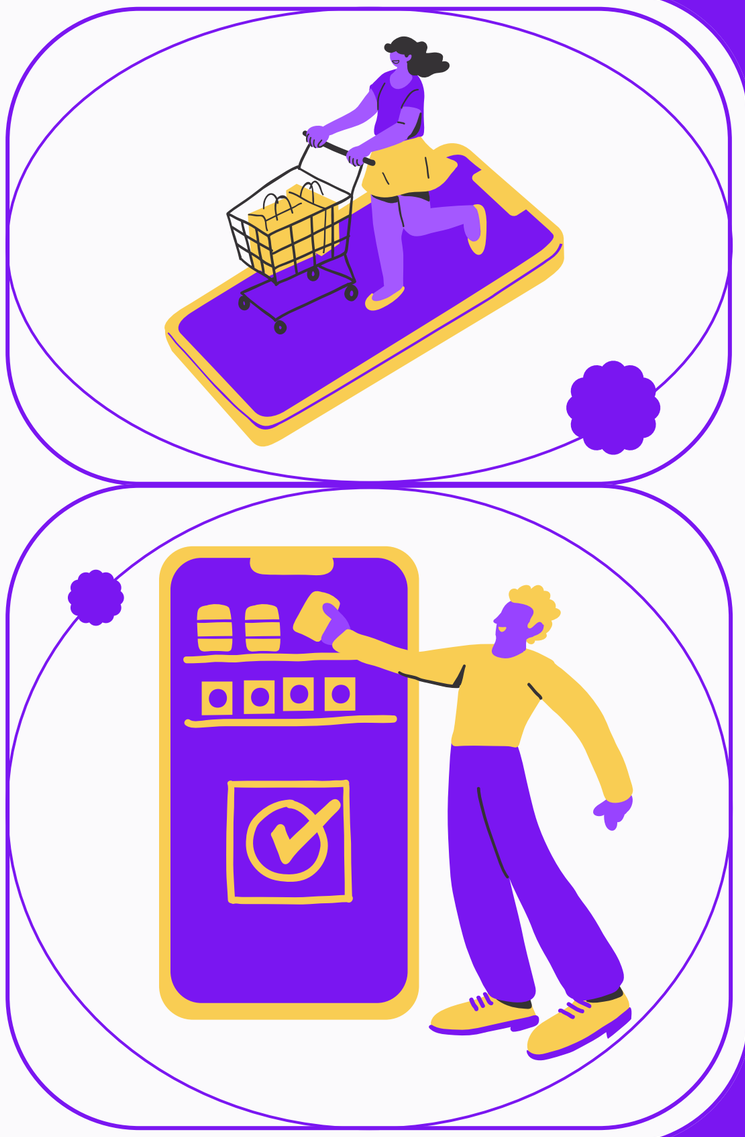
系統需求說明

Review（商品評價）

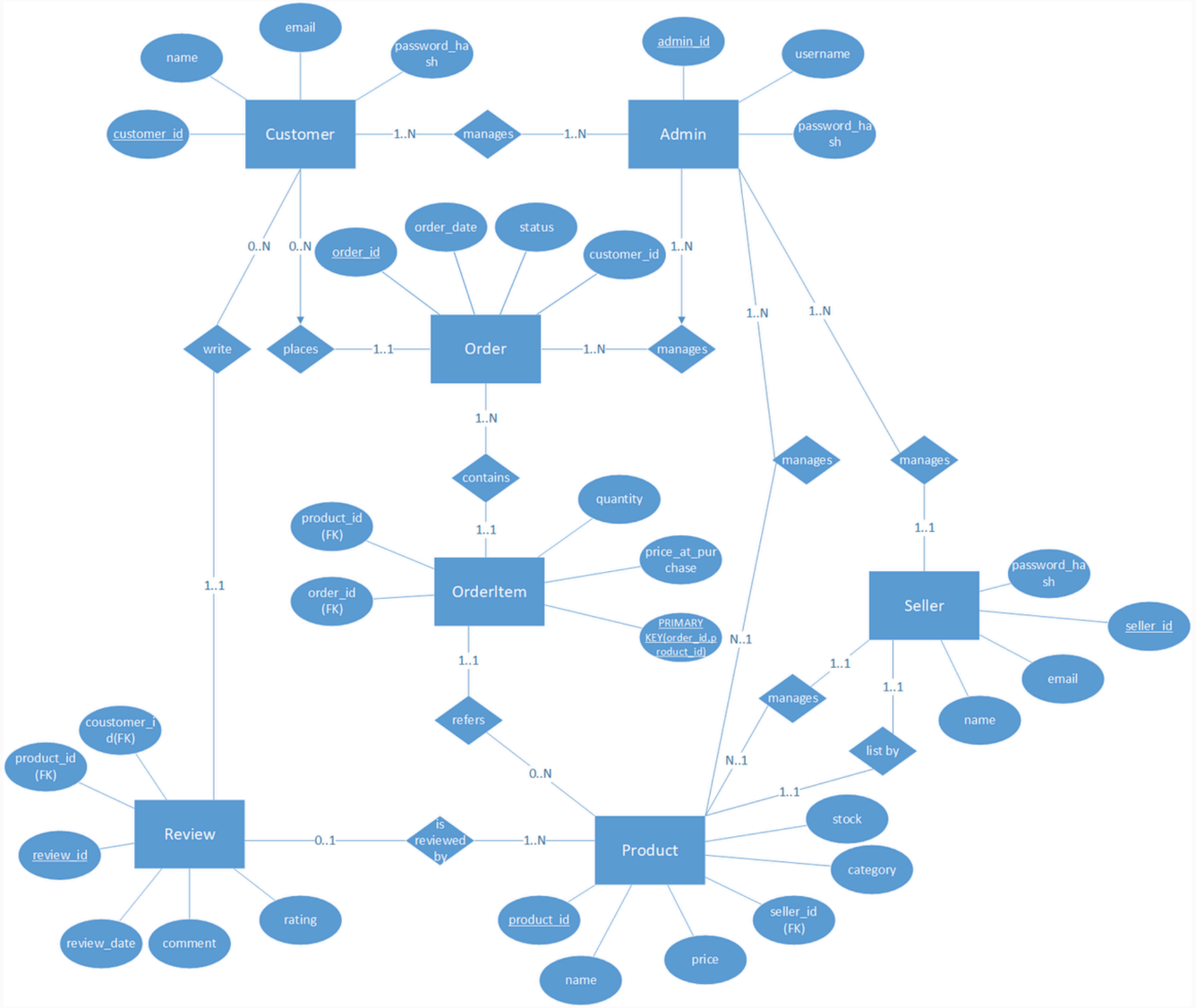
欄位名稱	完整性限制說明
review_id	主鍵。唯一且不可為空。
product_id	外鍵。需參照 Product.product_id，且必須存在。
customer_id	外鍵。需參照 Customer.customer_id，且必須存在。
rating	整數，值域限定為 1 至 5。可使用 CHECK (rating BETWEEN 1 AND 5) 強制限制評分等級。
comment	可為空，但若填寫，須具備有效文字內容 （不得為純空白或無意義字元）。
review_date	必須為合理有效日期 （格式為 YYYY-MM-DD），不得晚於當前系統時間， 亦不得早於該商品的訂單完成時間。

完整性限制（Integrity Constraints）

1. 主鍵唯一性：每筆商品、訂單、使用者皆有唯一 ID。
2. 外鍵約束：
    - 商品屬於某賣家（Product.seller_id -> Seller.id）
    - 訂單屬於某買家（Order.customer_id -> Customer.id）
3. 庫存限制：商品庫存數量不得為負。
4. 付款狀態驗證：僅當付款成功後，訂單狀態才能從「未付款」轉為「處理中」。
5. 評價限制：使用者只能針對已完成的訂單商品進行評價一次。
6. Email 唯一性：每個使用者帳號的 Email 必須唯一。



ER Diagram（實體關係圖）



詳細請看Github

ER Diagram 詳細說明（實體與關係）

1. Customer（買家）

欄位名稱	欄位說明	鍵類型	參照說明	是否可為空
customer_id	客戶編號	主鍵		否
name	姓名			否
email	電子郵件	唯一鍵		否

ER Diagram 詳細說明（實體與關係）

2. Seller（賣家）

欄位名稱	欄位說明	鍵類型	參照說明	是否可為空
seller_id	賣家編號	主鍵		否
name	姓名			否
email	電子郵件	唯一鍵		否

ER Diagram 詳細說明（實體與關係）

3.Admin（管理員）

欄位名稱	欄位說明	鍵類型	參照說明	是否可為空
admin_id	管理員編號	主鍵		否
username	使用者名稱	唯一鍵		否

ER Diagram 詳細說明（實體與關係）

4. Product（商品）

欄位名稱	欄位說明	鍵類型	參照說明	是否可為空
product_id	商品編號	主鍵		否
name	商品名稱			否
price	價格			否
stock	庫存數量			否
category	商品類別			是
seller_id	賣家編號	外鍵	參照 Seller.seller_id	否

ER Diagram 詳細說明（實體與關係）

5. Order（訂單）

欄位名稱	欄位說明	鍵類型	參照說明	是否可為空
order_id	訂單編號	主鍵		否
order_date	訂單日期			否
status	訂單狀態			否
customer_id	客戶編號	外鍵	參照 Customer.customer_id	否

ER Diagram 詳細說明（實體與關係）

6. OrderItem（訂單項目）

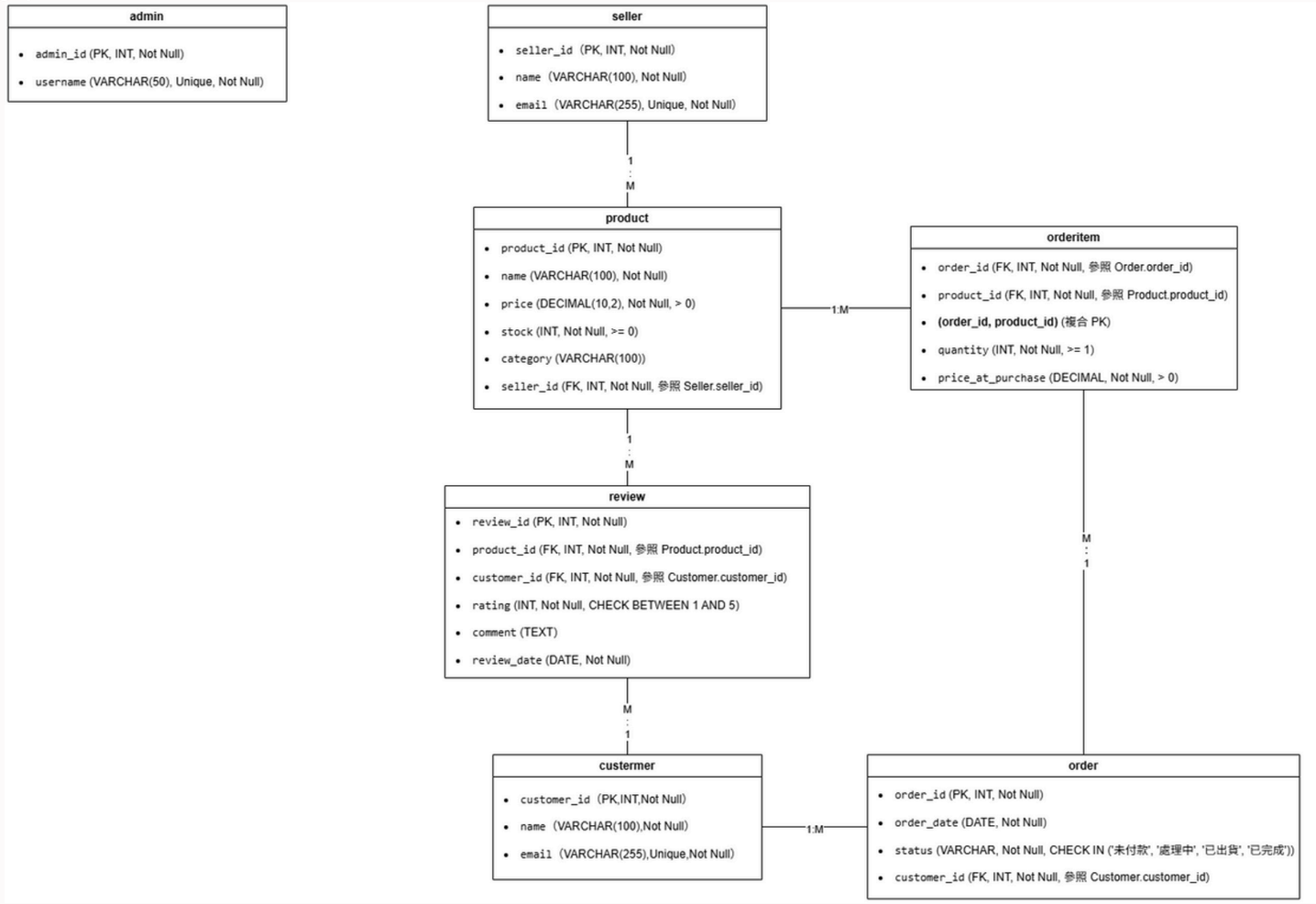
欄位名稱	欄位說明	鍵類型	參照說明	是否可為空
order_id	訂單編號	複合主鍵	參照 Order.order_id	否
product_id	商品編號	複合主鍵	參照 Product.product_id	否
quantity	購買數量			否
price_at_purchase	購買時單價			否

ER Diagram 詳細說明（實體與關係）

7. Review（商品評價）

欄位名稱	欄位說明	鍵類型	參照說明	是否可為空
review_id	評價編號	主鍵		否
product_id	商品編號	外鍵	參照 Product.product_id	否
customer_id	客戶編號	外鍵	參照 Customer.customer_id	否
rating	評分			否
comment	評論內容			是
review_date	評論日期			否

完整DB Schema 與說明



詳細請看Github

SQL語法涵概念層與View

CustomerPublicInfo View

```
CREATE VIEW CustomerPublicInfo AS
SELECT
    customer_id,
    name
FROM Customer;
```

範例

```
SELECT * FROM CustomerPublicInfo;
```

結果範例

customer_id	name
1	Alice Lin
2	Bob Lee

SQL語法涵概念層與View

範例

```
SELECT * FROM SellerStats WHERE average_rating >= 4.5;
```

SellerStats View

```
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;
```

結果範例

seller_id	seller_name	total_products	average_rating
1	Jane Chen	12	4.67

SQL語法涵概念層與View

範例

```
SELECT * FROM HighPricedProducts LIMIT 2;
```

HighPricedProducts View

```
CREATE VIEW HighPricedProducts AS
SELECT
    product_id,
    name,
    price,
    stock,
    category,
    seller_id
FROM Product
WHERE stock > 0
ORDER BY price DESC;
```

結果範例

product_id	name	price	stock	category	seller_id
101	Luxury Watch	8999	5	Watches	2
102	Diamond Ring	7999	3	Jewelry	3

SQL語法涵概念層與View

範例

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

OrderSummary View

```
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;
```

結果範例

order_id	order_date	status	customer_name	customer_email
1	2024-06-01 12:00:00	已出貨	Alice Lin	alice@example.com

SQL語法涵概念層與View

範例

```
SELECT * FROM OrderItemDetails;
```

OrderItemDetails View

```
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;
```

結果範例

order_id	product_id	product_name	quantity	price_at_purchase
401	302	iPhone 15 Pro	1	45900
401	303	AirPods Pro	2	7490

SQL語法涵概念層與View

範例

```
SELECT * FROM ProductReviewStats;
```

ProductReviewStats View

```
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;
```

結果範例

product_id	product_name	average_rating	latest_review_time
302	iPhone 15 Pro	4.75	2024-05-30 14:25:00
303	AirPods Pro	4.6	2024-05-29 18:00:00

SQL語法涵概念層與View

範例

```
SELECT * FROM LatestProductReview;
```

LatestProductReview View

```
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;
```

結果範例

review_id	product_id	rating	review_text	review_date
601	302	5	Excellent sound quality!	2024-05-30 14:25:00
602	303	4	Great value for price.	2024-05-29 18:00:00

Thank You