

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：

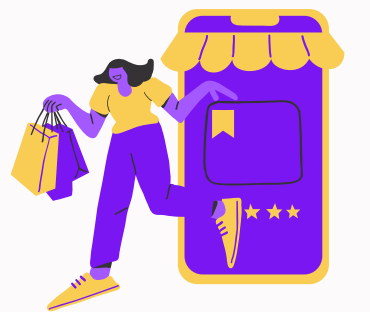
陳荔群 41043218

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
- 應用情境與使用案例
- 完整性限制
- ER Diagram與詳細說明
- 系統需求說明
- 完整DB Schema與說明
- SQL語法涵概念層與View
- 系統實作成果
- 團隊分工
- 參考資料



# 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

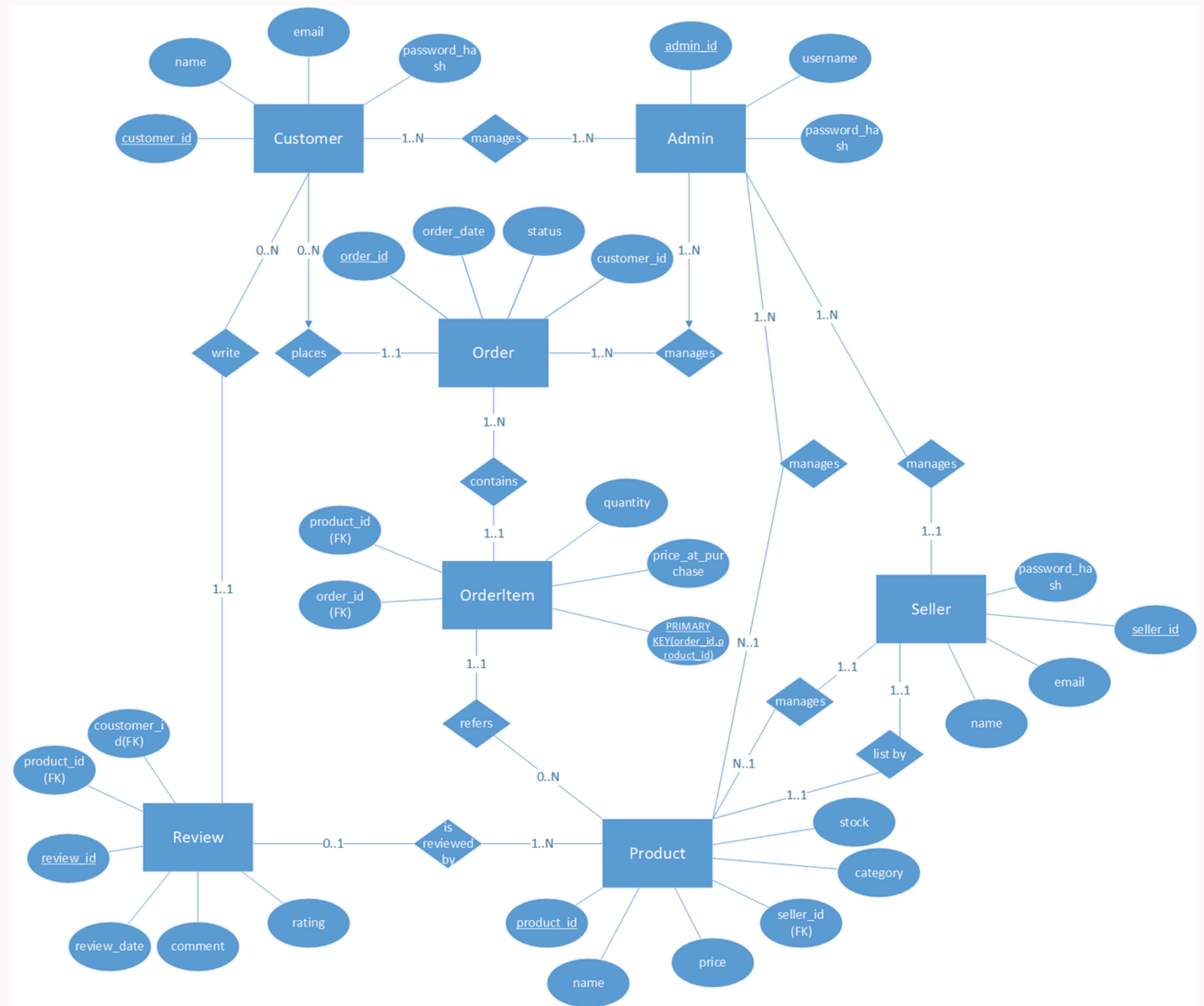


# 主要使用案例 Use Cases



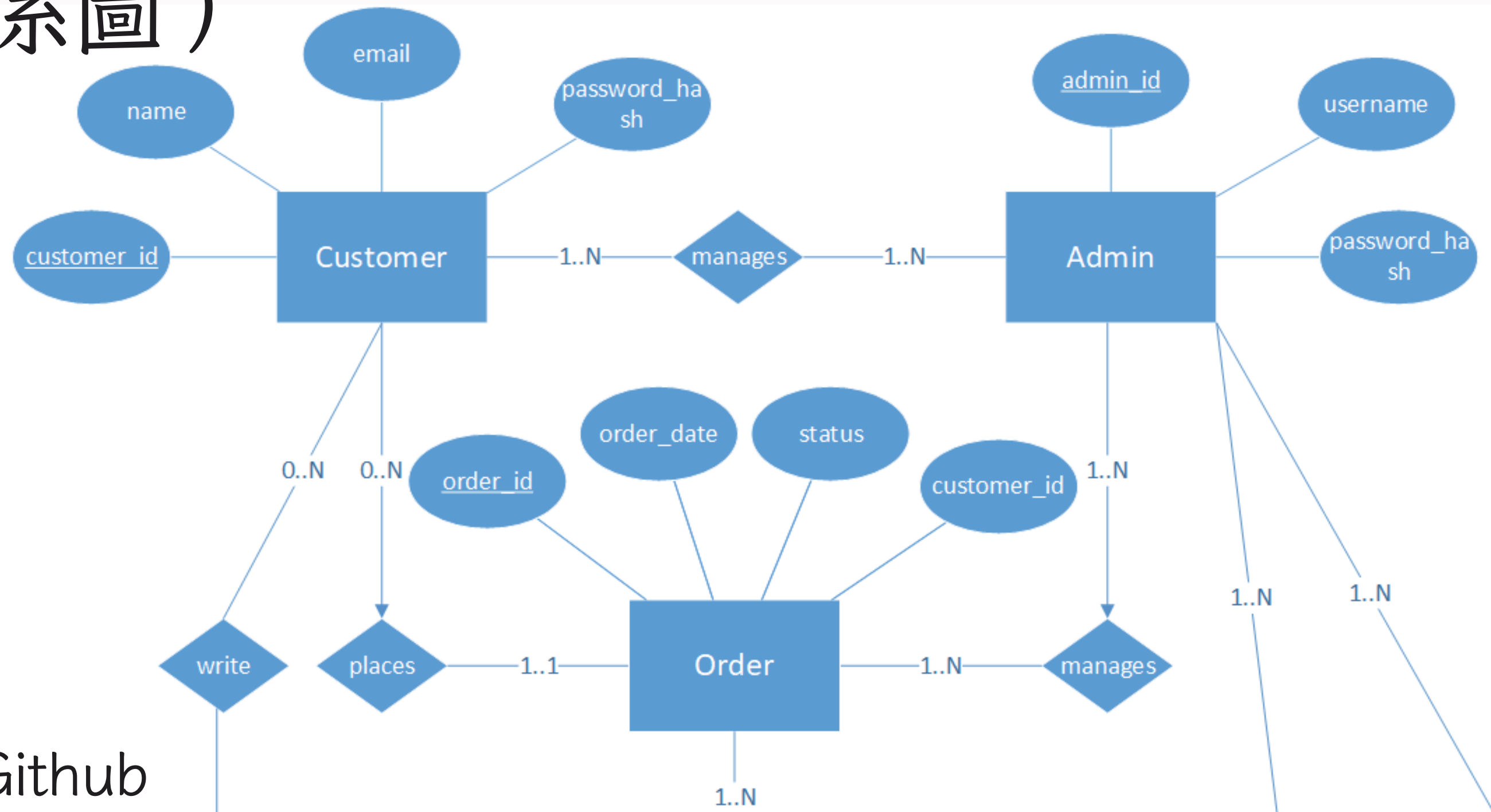
使用者 角色	使用案例
買家	註冊/登入、瀏覽商品、加入購物車、下單、付款、查看訂單、 評價商品
賣家	註冊/登入、上架商品、修改商品資訊、查看訂單、出貨、查看 評論
管理員	查看平台交易記錄、管理用戶帳號、下架不當商品

# ER Diagram (實體關係圖)



詳細請看Github

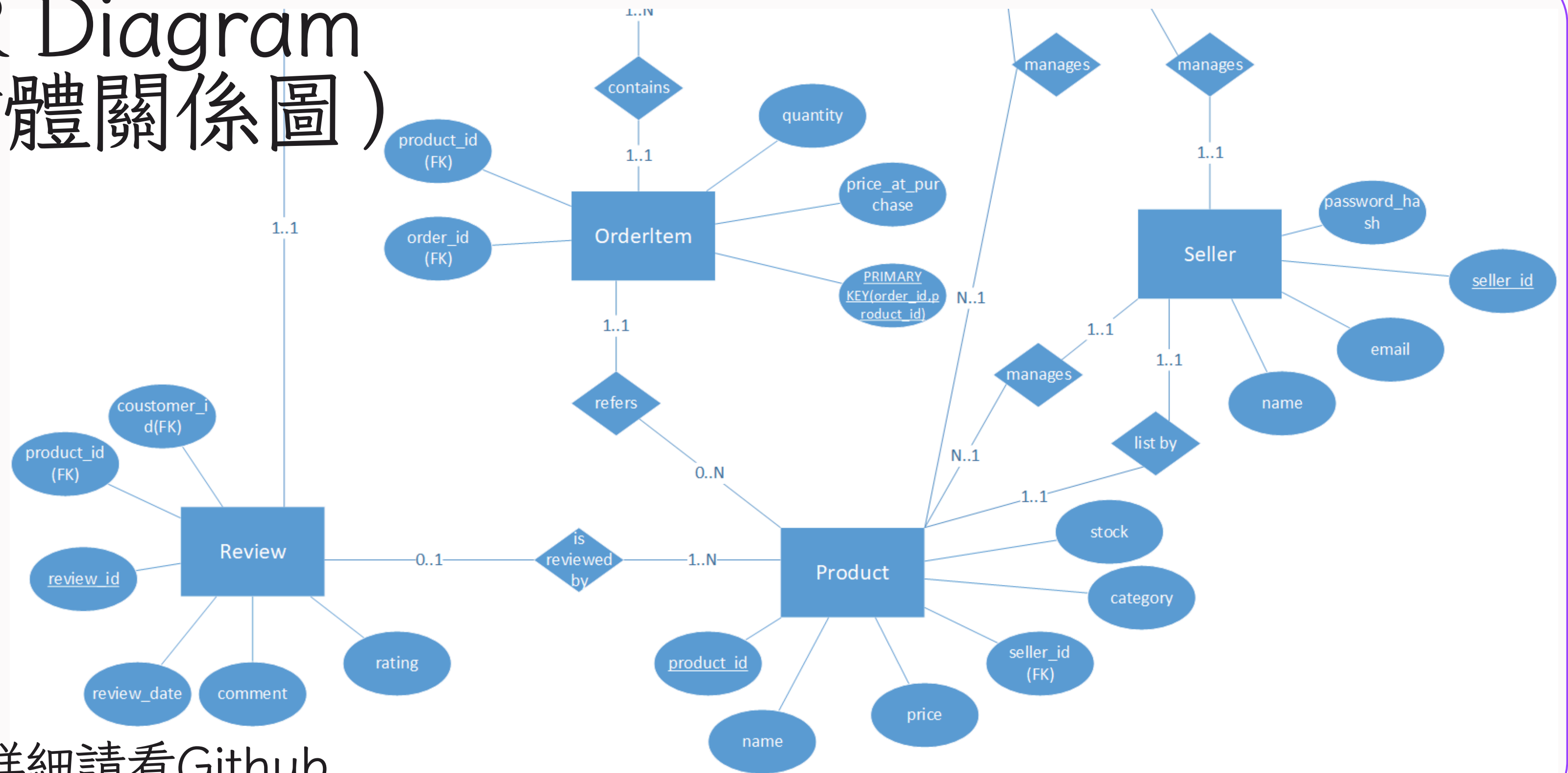
# ER Diagram (實體關係圖)



詳細請看Github



# ER Diagram (實體關係圖)



詳細請看Github



# 系統需求說明

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： <code>^[A-Za-z0-9]{3,20}\$</code> 。
email	唯一且不可為空，需符合有效 Email 格式，例如 <code>user@example.com</code> ，不可含空格。可使用正規表示式 <code>^[\w\.-]+@[\w\.-]+\.\w{2,}\$</code> 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： <code>^09\d{8}\$</code> 。
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。

# 系統需求說明

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。

# 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，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 <code>^[A-Za-z0-9]{4,20}\$</code> 。

# 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，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。

# 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	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。

# 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，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。

# 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，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。



# 完整性限制（Integrity Constraints）

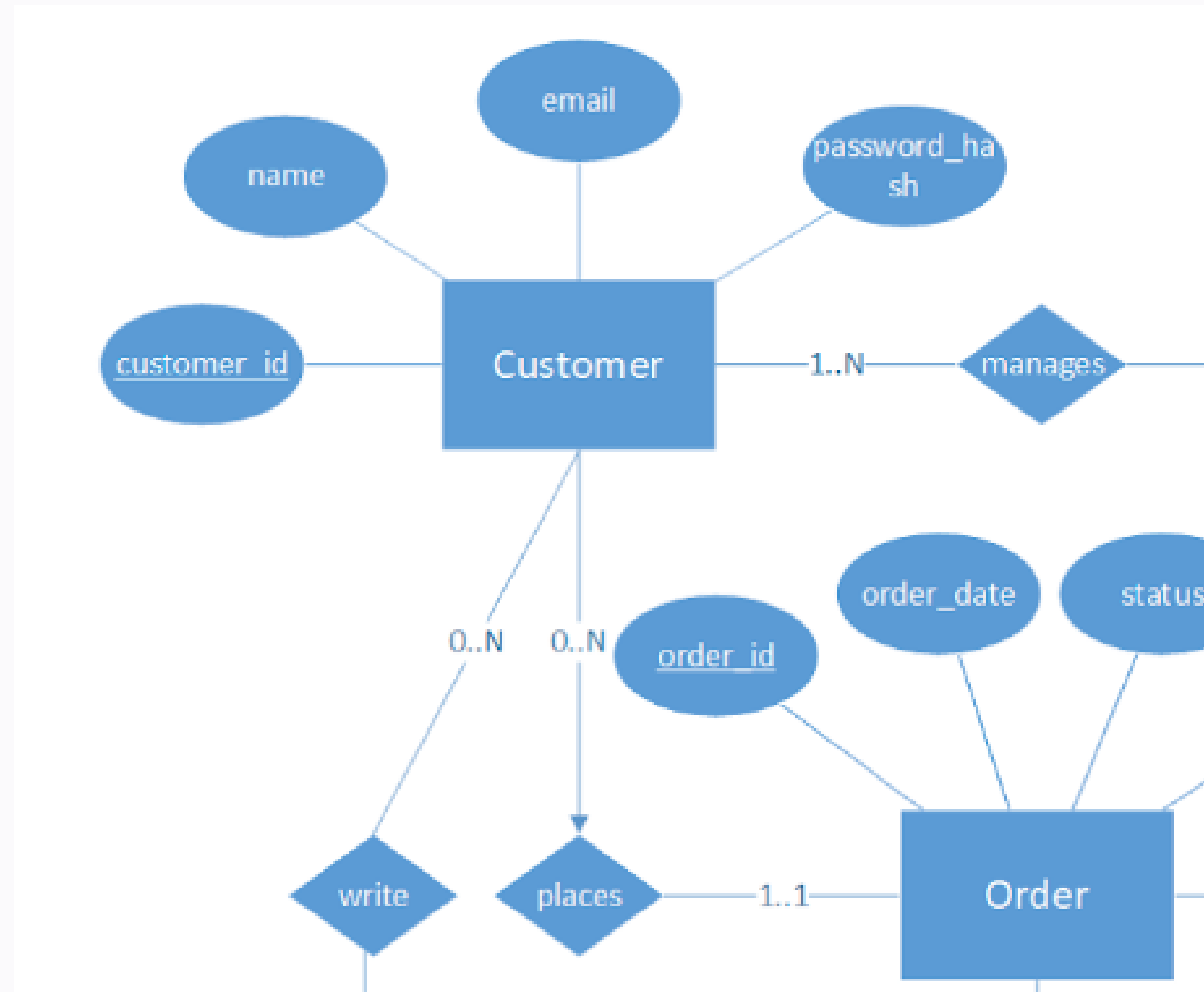
類別	詳細說明
1. 主鍵唯一性 (Primary Key Uniqueness)	<p>系統中每一筆資料應有唯一識別碼：</p> <ul style="list-style-type: none"><li>• <code>Customer.customer_id</code>、<code>Seller.seller_id</code>、<code>Admin.admin_id</code>、<code>Product.product_id</code>、<code>Order.order_id</code>、<code>Review.review_id</code> 等欄位皆設為主鍵。</li><li>• 此欄位應使用系統自動產生（如 UUID、AUTO_INCREMENT）或手動確保唯一性。<ul style="list-style-type: none"><li>• 不可為空（NOT NULL）。</li><li>• 不得重複，否則會違反資料一致性。</li></ul></li></ul>
2. 外鍵約束 (Foreign Key Constraints)	<p>確保資料之間的參照關係正確：</p> <ul style="list-style-type: none"><li>• <code>Product.seller_id</code> 必須參照存在於 <code>Seller.seller_id</code> 的資料，否則無法建立商品。</li><li>• <code>Order.customer_id</code> 必須對應至現有的 <code>Customer.customer_id</code>。</li><li>• <code>OrderItem.order_id</code> 與 <code>OrderItem.product_id</code> 必須參照 <code>Order</code> 與 <code>Product</code> 表中存在的資料。</li><li>• <code>Review.product_id</code> 與 <code>Review.customer_id</code> 也必須為有效外鍵。</li><li>• 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。</li></ul>
3. 庫存限制 (Stock Constraint)	<p>商品的 <code>stock</code> 欄位（庫存數量）必須為整數且 <math>\geq 0</math>：</p> <ul style="list-style-type: none"><li>• 禁止輸入負數，防止出現負庫存狀況。</li><li>• 可使用 <code>CHECK (stock &gt;= 0)</code> 約束實作。</li><li>• 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」。<ul style="list-style-type: none"><li>• 若商品已停售，其庫存應為 0，或透過「狀態欄位」表示停售。</li></ul></li></ul>

# 完整性限制（Integrity Constraints）

4. 付款狀態驗證 (Order Status Verification)	<p>訂單狀態變更需依付款流程限制：</p> <ul style="list-style-type: none"><li>• 初始狀態為「未付款」。</li><li>• 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」。</li><li>• 後續狀態依序為「已出貨」→「已完成」，不可跳階段。</li><li>• 可使用狀態流轉邏輯（如狀態機）或資料庫觸發器加強限制。</li></ul>
5. 評價限制 (Review Constraint)	<p>使用者只能對已完成訂單中的商品進行一次評價：</p> <ul style="list-style-type: none"><li>• 僅當 <code>Order.status = '已完成'</code> 且商品為該筆訂單項目之一時，才可新增評價。</li><li>• 同一名買家對同一筆商品僅能評價一次，可使用 <code>(customer_id, product_id)</code> 組合建立唯一索引避免重複評價。</li><li>• 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。</li></ul>
6. Email 唯一性 (Email Uniqueness)	<p>所有使用者（Customer、Seller、Admin）的 Email 必須唯一，不可重複：</p> <ul style="list-style-type: none"><li>• 每個使用者帳號的 <code>email</code> 欄位設為 <code>UNIQUE</code>。</li><li>• 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。</li><li>• Email 欄位必須為有效格式，可使用正則表示式驗證： <code>^[\\w\\.-]+@[\\w\\.-]+\\.\\w{2,}\$</code>。</li></ul>

# ER Diagram 詳細說明（實體與關係）

## 1. Customer（買家）



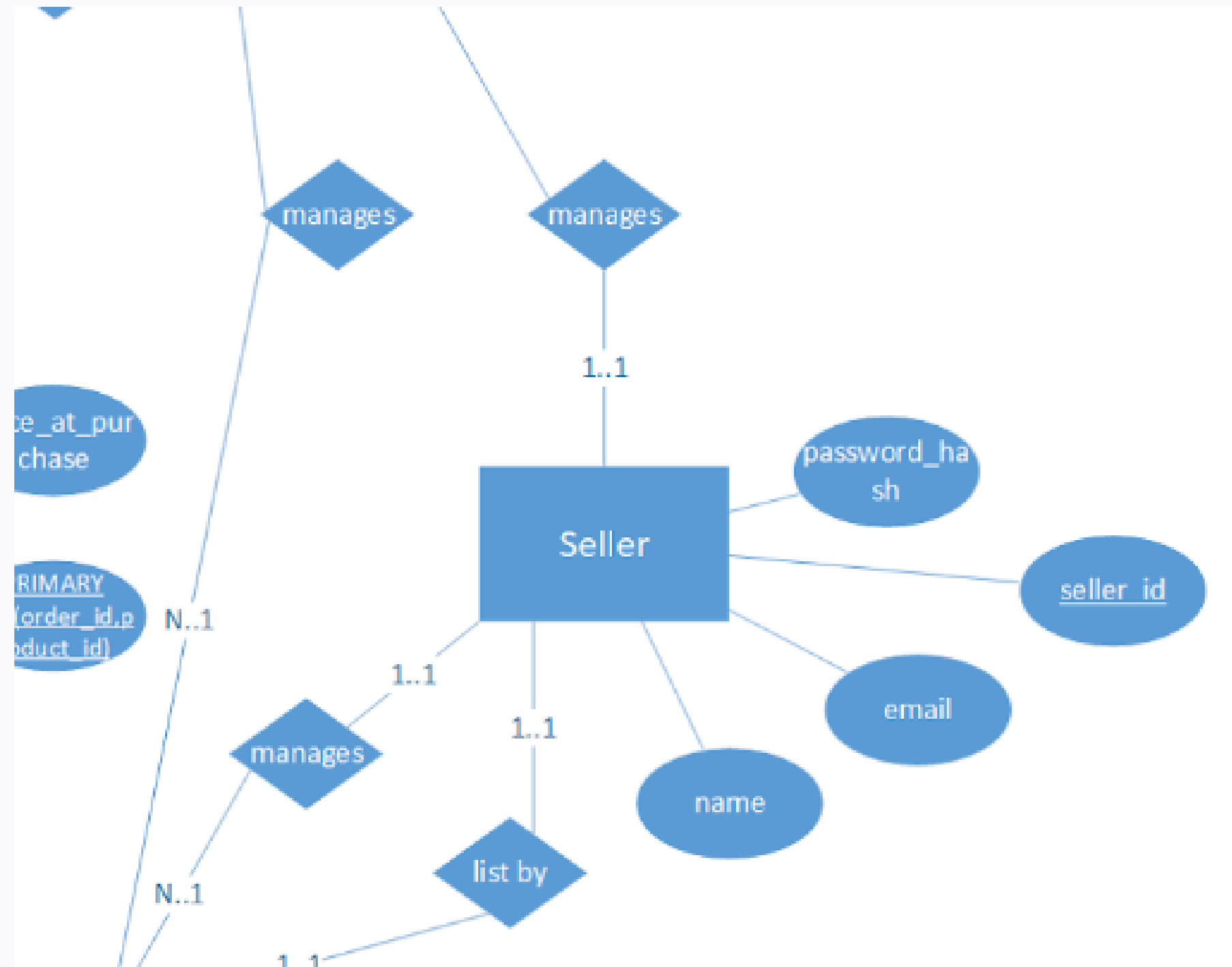
# ER Diagram 詳細說明（實體與關係）

## 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

# ER Diagram 詳細說明（實體與關係）

## 2. Seller（賣家）



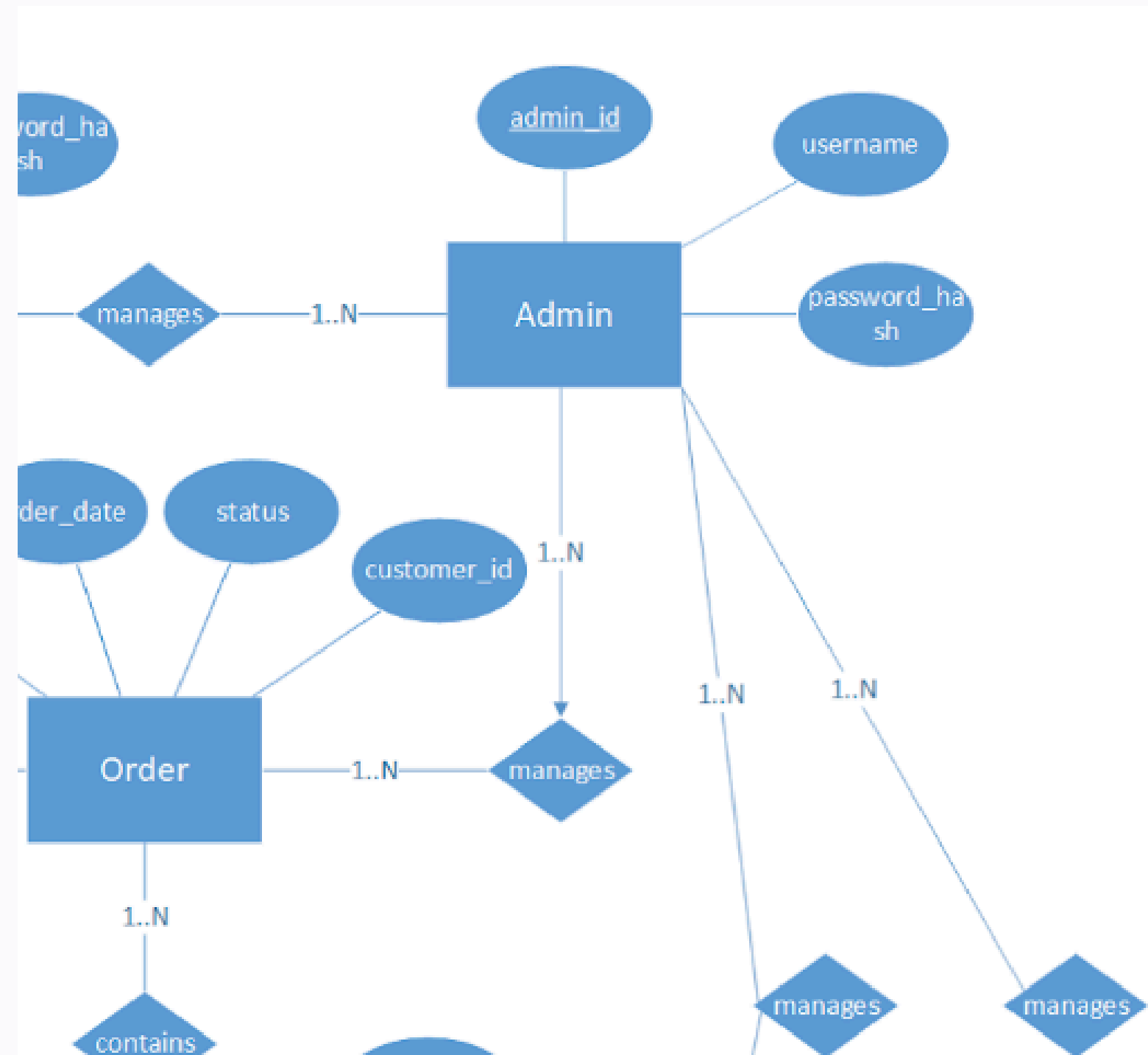
# ER Diagram 詳細說明（實體與關係）

## 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

# ER Diagram 詳細說明（實體與關係）

## 3.Admin（管理員）





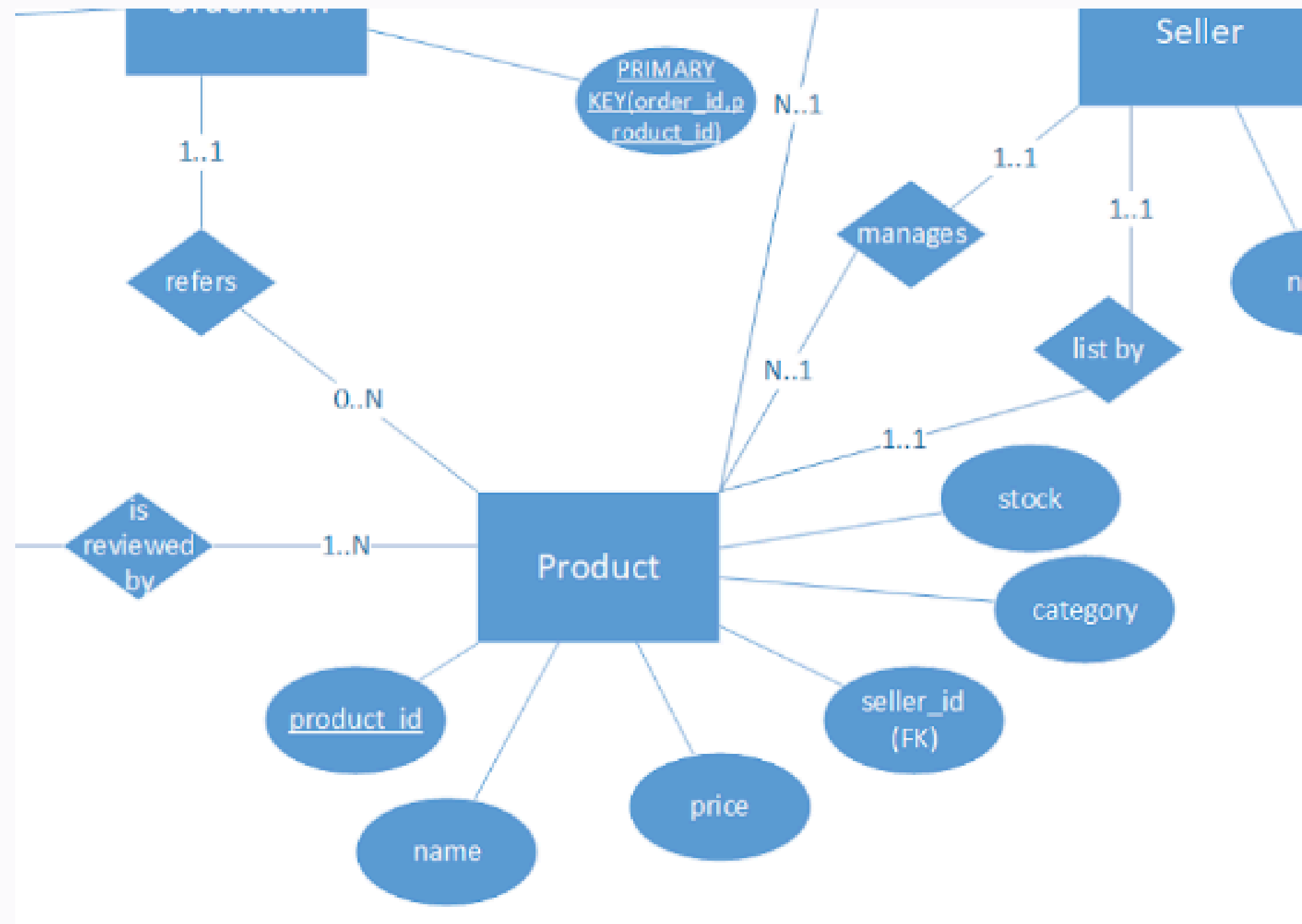
# ER Diagram 詳細說明（實體與關係）

## 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

# ER Diagram 詳細說明（實體與關係）

## 4. Product（商品）



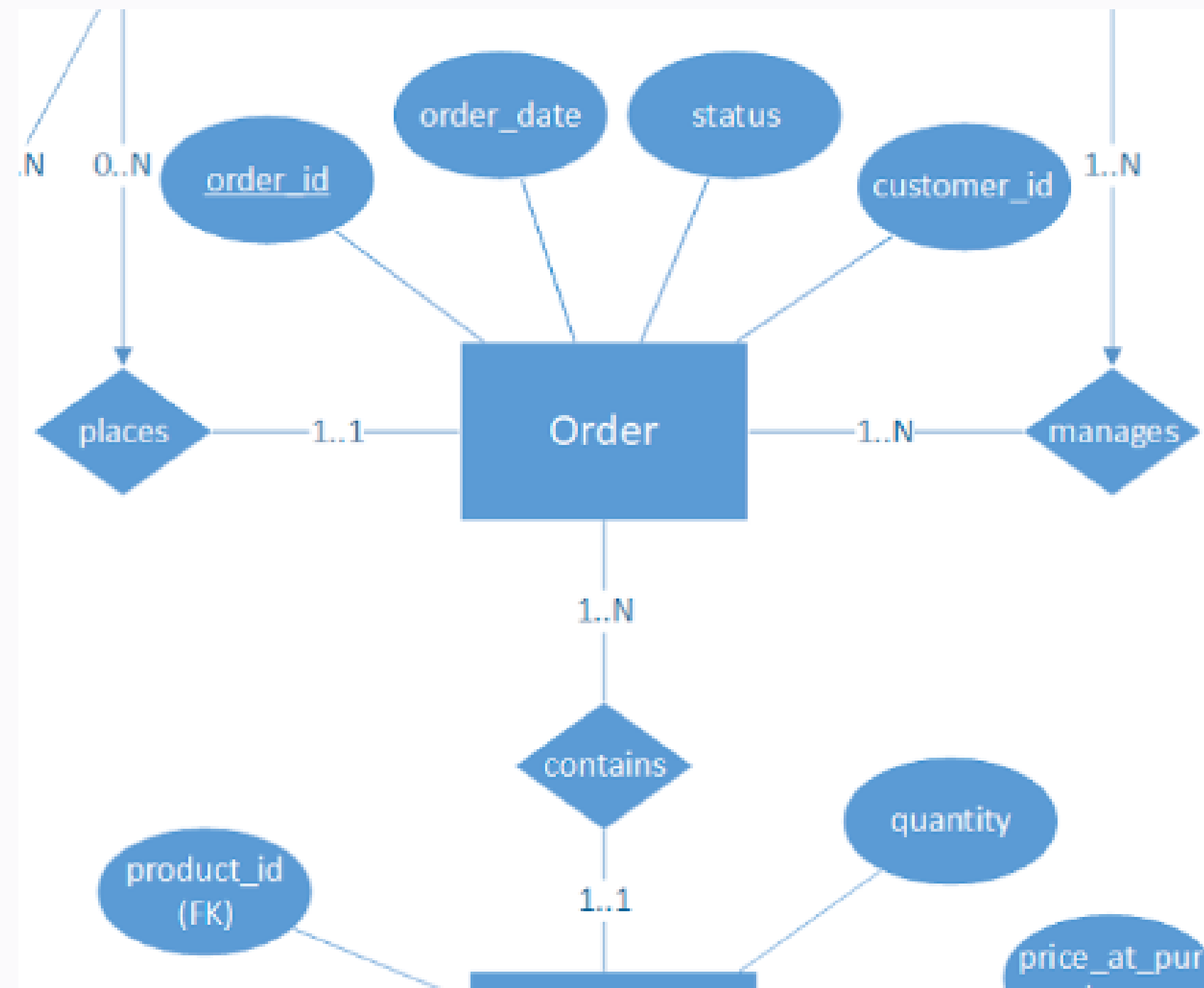
# ER Diagram 詳細說明（實體與關係）

## 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

# ER Diagram 詳細說明（實體與關係）

## 5. Order（訂單）



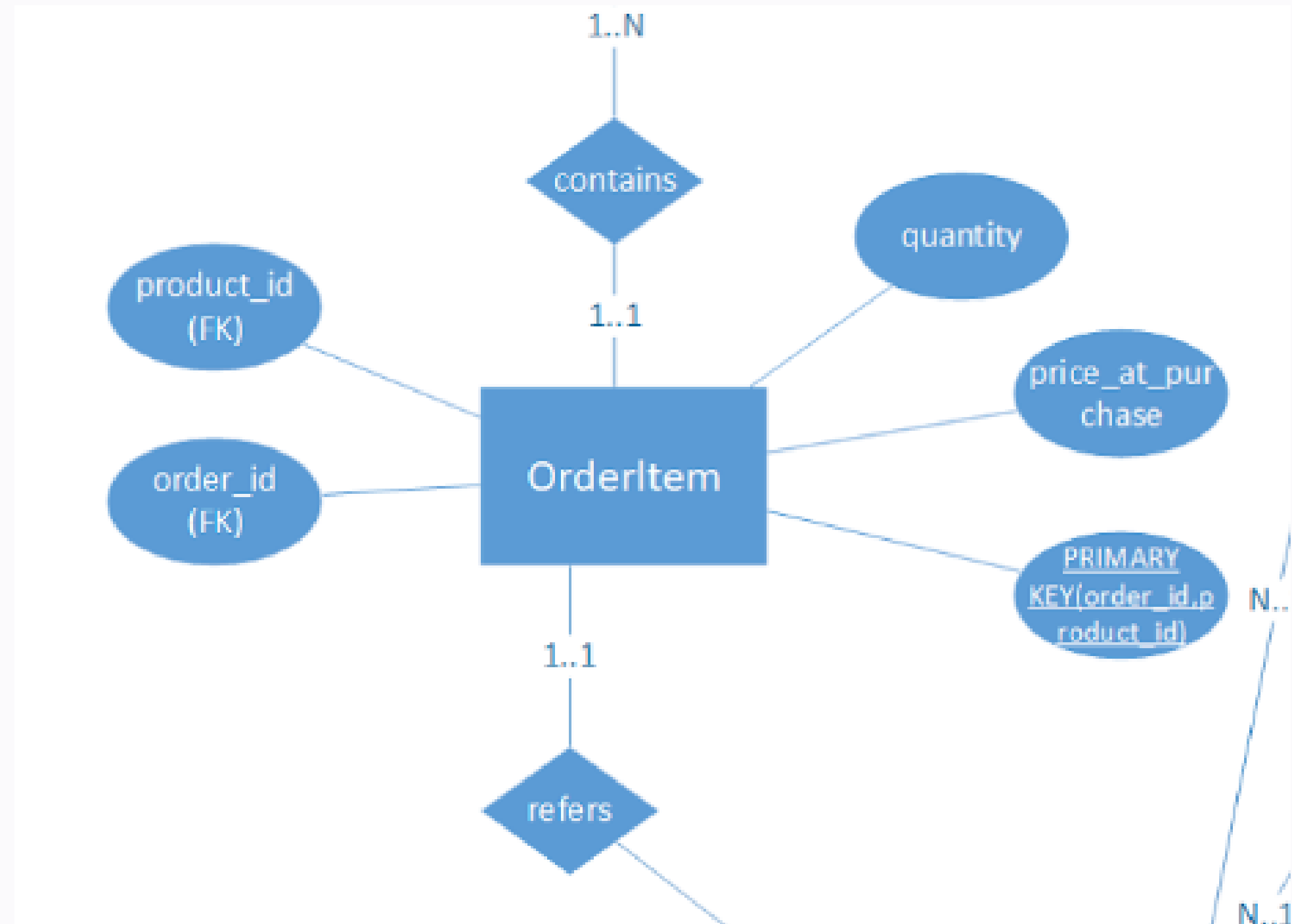
# ER Diagram 詳細說明（實體與關係）

## 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

# ER Diagram 詳細說明（實體與關係）

## 6. OrderItem（訂單項目）



# ER Diagram 詳細說明（實體與關係）

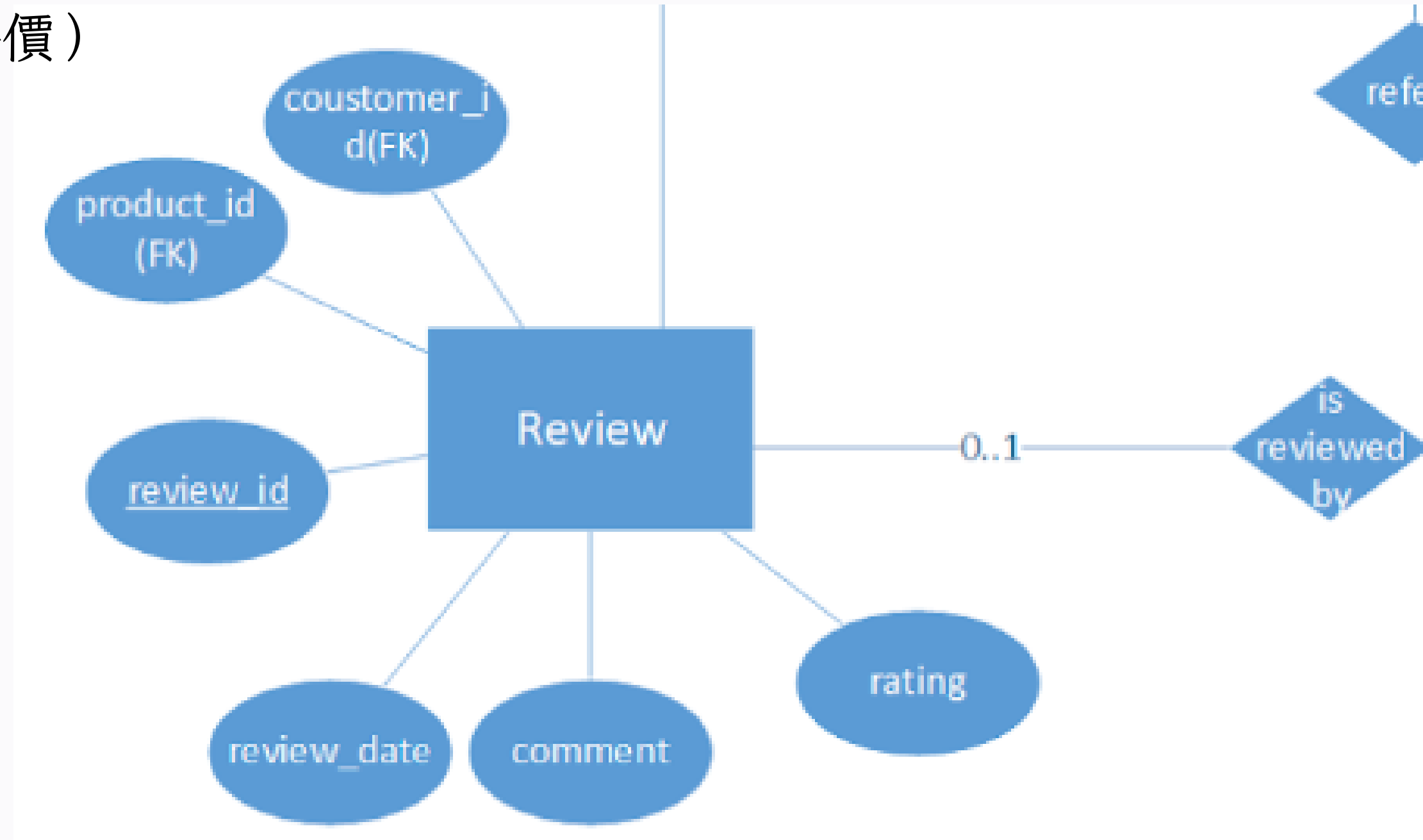
## 6. OrderItem（訂單項目）

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		$\geq 1$ ，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



# ER Diagram 詳細說明（實體與關係）

## 7. Review（商品評價）



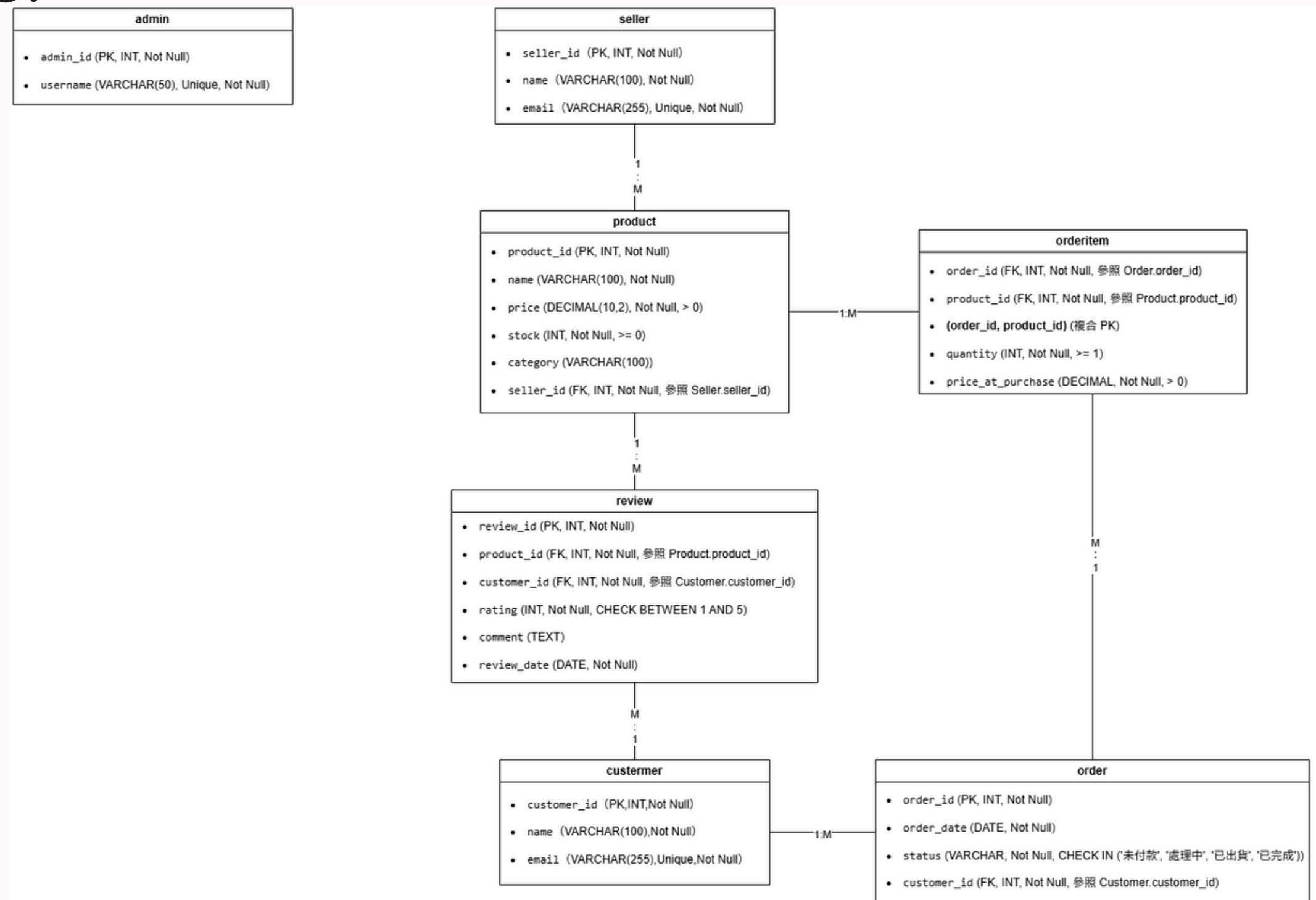
# ER Diagram 詳細說明（實體與關係）

## 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

# 完整DB Schema 與說明

詳細請看Github



# 完整DB Schema 與說明

## 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。

# 完整DB Schema 與說明

## 6.1 Customer (買家)

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');
```

# 完整DB Schema 與說明

## 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： <code>^[A-Za-z0-9]{3,20}\$</code> 。
email	VARCHAR(255)	唯一且不可為空，需符合有效 Email 格式，例如 <code>user@example.com</code> ，不可含空格。可使用正規表示式 <code>^[\\w\\.-]+@[\\w\\.-]+\\.\\w{2,}\$</code> 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： <code>^09\\d{8}\$</code> 。
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 <code>YYYY-MM-DD</code> ，必須為有效歷史日期，不得晚於系統當前日期。

# 完整DB Schema 與說明

## 6.2 Seller (賣家)

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');
```



# 完整DB Schema 與說明

## 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

# 完整DB Schema 與說明

## 6.3 Admin ( 管理員 )

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' );
```

# 完整DB Schema 與說明

## 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

# 完整DB Schema 與說明

## 6.4 Product (商品)

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);
```

# 完整DB Schema 與說明

## 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

# 完整DB Schema 與說明

## 6.5 Order (訂單)

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);
```

# 完整DB Schema 與說明

## 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 <a href="#">Order.order_id</a>
product_id	INT (PK, FK)	商品編號，複合主鍵，不可為空，參照 <a href="#">Product.product_id</a>
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

# 完整DB Schema 與說明

## 6.6 OrderItem (訂單項目)

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);
```



# 完整DB Schema 與說明

## 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

# 完整DB Schema 與說明

## 6.7 Review (商品評價)

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');
```

# SQL語法涵概念層與View

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;
```

```
MariaDB [final]> SELECT * FROM CustomerPublicInfo;
+-----+-----+
| customer_id | name      |
+-----+-----+
| 1           | Alice123  |
| 2           | BobChen01 |
| 3           | CathyWang9|
| 4           | DavidLiu88|
| 5           | EmmaWu77  |
| 6           | FrankHuang|
| 7           | GraceTsai1|
| 8           | HenryYeh22|
| 9           | IreneKuo3 |
| 10          | JackyHsu44|
+-----+-----+
10 rows in set (0.016 sec)
```

圖7.1 買家資料表view

# SQL語法涵概念層與View

## 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;
```

```
MariaDB [final]> SELECT * FROM SellerStats WHERE average_rating >= 4.5;
+-----+-----+-----+-----+
| seller_id | seller_name | total_products | average_rating |
+-----+-----+-----+-----+
|          1 | ShopA123    |          2    |          4.50 |
|          3 | ShopC789    |          1    |          5.00 |
|          4 | GadgetWorld |          1    |          5.00 |
|          8 | KitchenPro  |          1    |          5.00 |
+-----+-----+-----+-----+
4 rows in set (0.002 sec)
```

圖7.2 賣家資料表view

## SQL語法涵概念層與View

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;
```

```
MariaDB [final]> SELECT * FROM HighPricedProducts LIMIT 2;
```

product_id	name	price	stock	category	seller_id
7	Coffee Maker	1299.00	30	Kitchen	8
2	Bluetooth Keyboard	899.00	80	Electronics	1

```
2 rows in set (0.001 sec)
```

圖7.3 商品資訊view

# SQL語法涵概念層與View

## 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';
```

```
MariaDB [final]> SELECT * FROM OrderSummary WHERE status = '已出貨';
+-----+-----+-----+-----+-----+
| order_id | order_date       | status | customer_id | customer_name |
+-----+-----+-----+-----+-----+
|      3   | 2025-06-12 17:52:35 | 已出貨 |      3      | CathyWang9    |
|      7   | 2025-06-12 17:52:35 | 已出貨 |      7      | GraceTsai1    |
+-----+-----+-----+-----+-----+
2 rows in set (0.001 sec)
```

圖7.4 訂單主檔view

## OrderItemDetails View

查詢每筆訂單中商品詳細資訊：

## SQL語法涵概念層與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
```

```
MariaDB [final]> SELECT * FROM OrderItemDetails;
+-----+-----+-----+-----+-----+
| order_id | product_id | product_name | quantity | price_at_purchase |
+-----+-----+-----+-----+-----+
| 1 | 1 | Wireless Mouse | 2 | 599.00 |
| 1 | 2 | Bluetooth Keyboard | 1 | 899.00 |
| 2 | 3 | USBC Charger | 3 | 299.00 |
| 3 | 4 | Laptop Stand | 1 | 799.00 |
| 4 | 5 | Notebook | 5 | 59.00 |
| 5 | 6 | TShirt | 2 | 399.00 |
| 6 | 7 | Coffee Maker | 1 | 1299.00 |
| 7 | 8 | Action Figure | 4 | 499.00 |
| 8 | 9 | Garden Shovel | 2 | 199.00 |
| 9 | 10 | LED Desk Lamp | 1 | 699.00 |
+-----+-----+-----+-----+-----+
10 rows in set (0.001 sec)
```

範例

```
SELECT * FROM OrderItemDetails;
```

圖7.5 訂單明細view

# SQL語法涵概念層與View

## 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;
```

```
MariaDB [final]> SELECT * FROM ProductReviewStats;
+-----+-----+-----+-----+
| product_id | product_name | average_rating | latest_review_time |
+-----+-----+-----+-----+
| 1 | Wireless Mouse | 5.00 | 2025-06-12 17:52:45 |
| 2 | Bluetooth Keyboard | 4.00 | 2025-06-12 17:52:45 |
| 3 | USBC Charger | 3.00 | 2025-06-12 17:52:45 |
| 4 | Laptop Stand | 5.00 | 2025-06-12 17:52:45 |
| 5 | Notebook | 4.00 | 2025-06-12 17:52:45 |
| 6 | TShirt | 2.00 | 2025-06-12 17:52:45 |
| 7 | Coffee Maker | 5.00 | 2025-06-12 17:52:45 |
| 8 | Action Figure | 4.00 | 2025-06-12 17:52:45 |
| 9 | Garden Shovel | 3.00 | 2025-06-12 17:52:45 |
| 10 | LED Desk Lamp | 5.00 | 2025-06-12 17:52:45 |
+-----+-----+-----+-----+
10 rows in set (0.001 sec)
```

圖7.6 商品資訊view



# LatestProductReview View

查詢每個商品的最新一筆評價：

## SQL語法涵概念層與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;
```

圖7.7 商品評價view

範例

```
SELECT * FROM LatestProductReview;
```

MariaDB [final]> SELECT \* FROM LatestProductReview;

review_id	product_id	customer_id	rating	comment	review_date
11	1	1	5	非常好用，品質也很棒	2025-06-12 17:52:45
12	2	2	4	打字手感不錯，就是有點重	2025-06-12 17:52:45
13	3	3	3	充電頭普通，不過夠用	2025-06-12 17:52:45
14	4	4	5	很穩定的支架，推薦	2025-06-12 17:52:45
15	5	5	4	紙質不錯，適合書寫	2025-06-12 17:52:45
16	6	6	2	材質一般，容易皺	2025-06-12 17:52:45
17	7	7	5	煮咖啡很方便，滿意	2025-06-12 17:52:45
18	8	8	4	小朋友很喜歡	2025-06-12 17:52:45
19	9	9	3	有點小貴，但好用	2025-06-12 17:52:45
20	10	10	5	燈光柔和，很適合夜讀	2025-06-12 17:52:45

10 rows in set (0.001 sec)

# 實作結果

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。



# 實作結果

## 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。



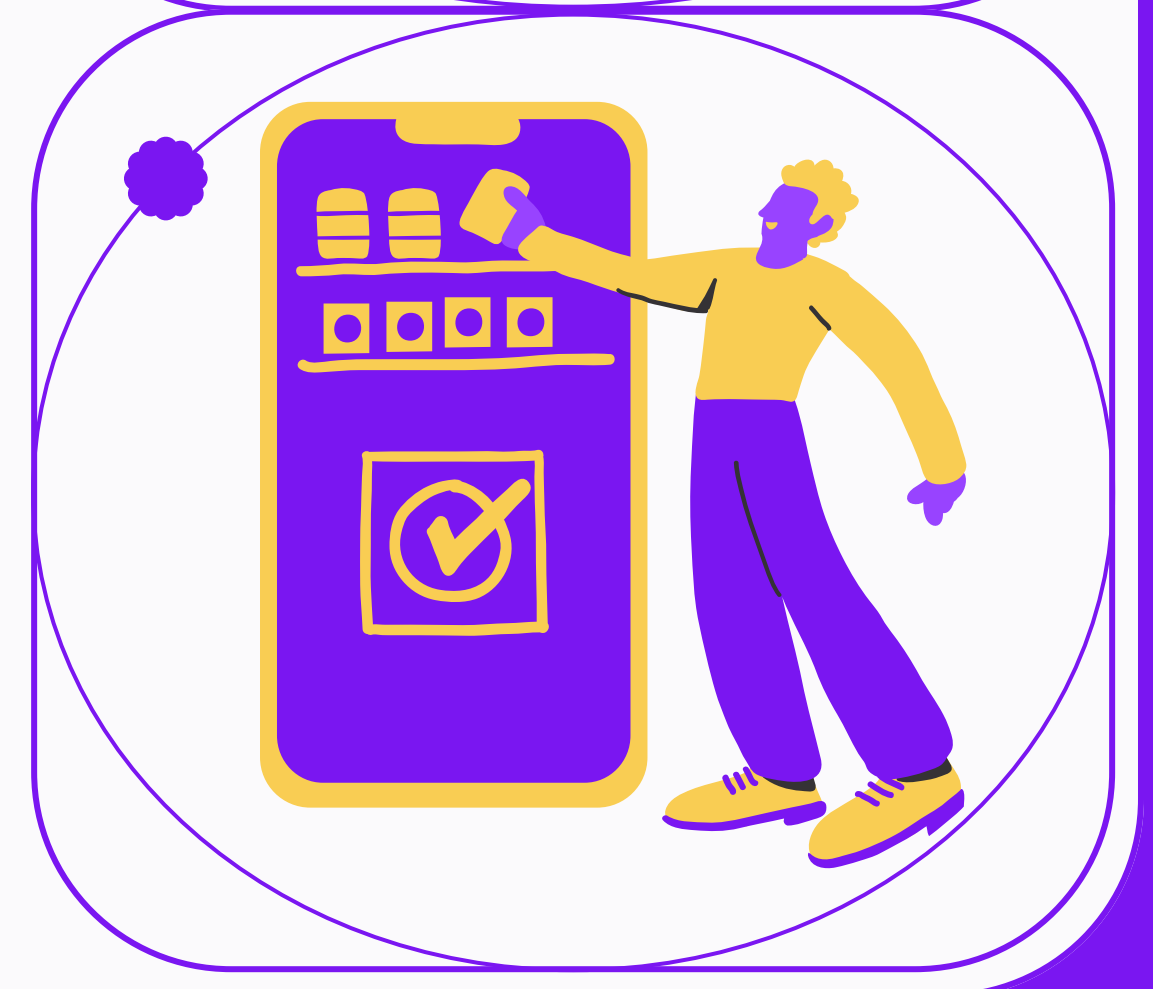
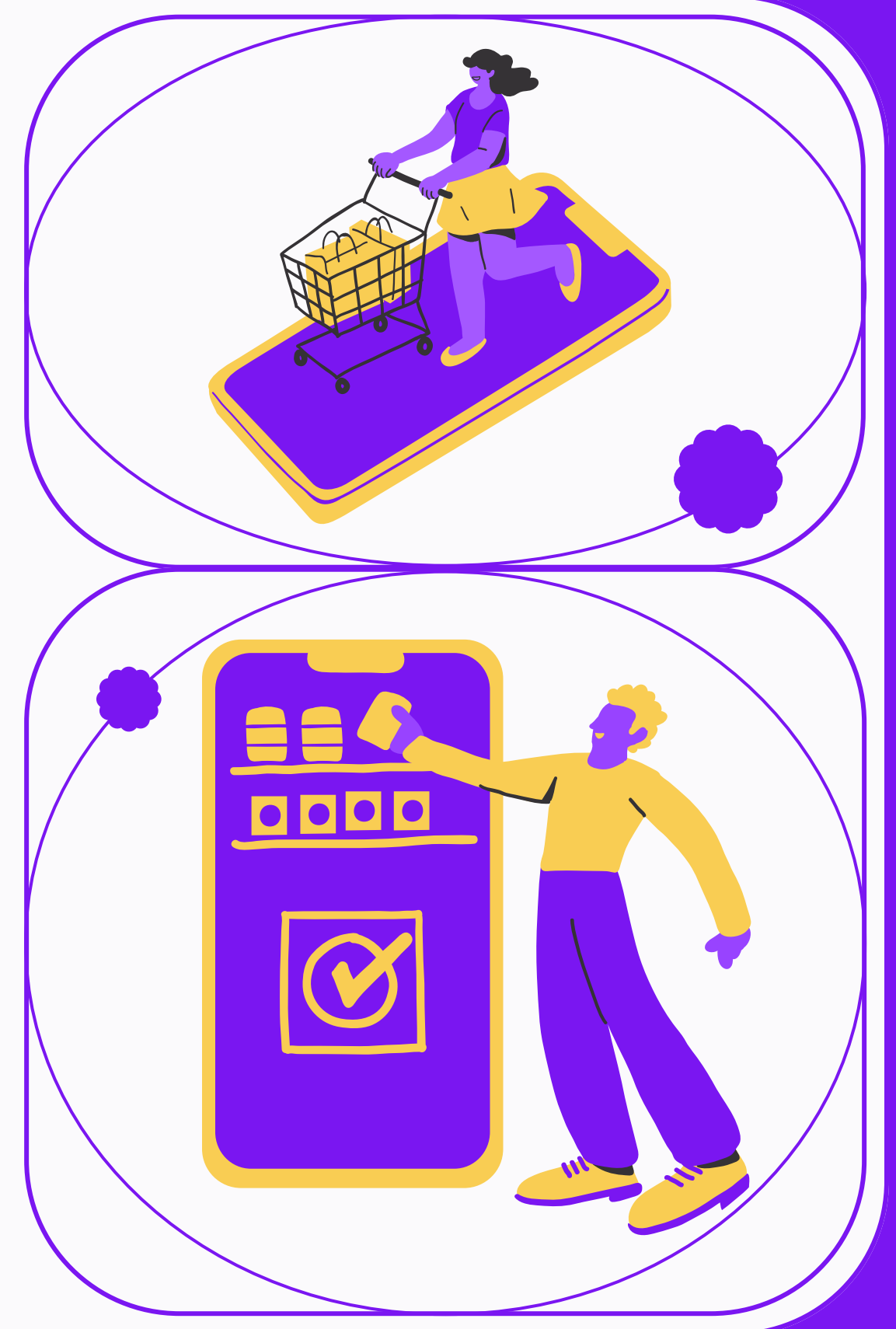
# 實作結果

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

# 實作結果

## 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：



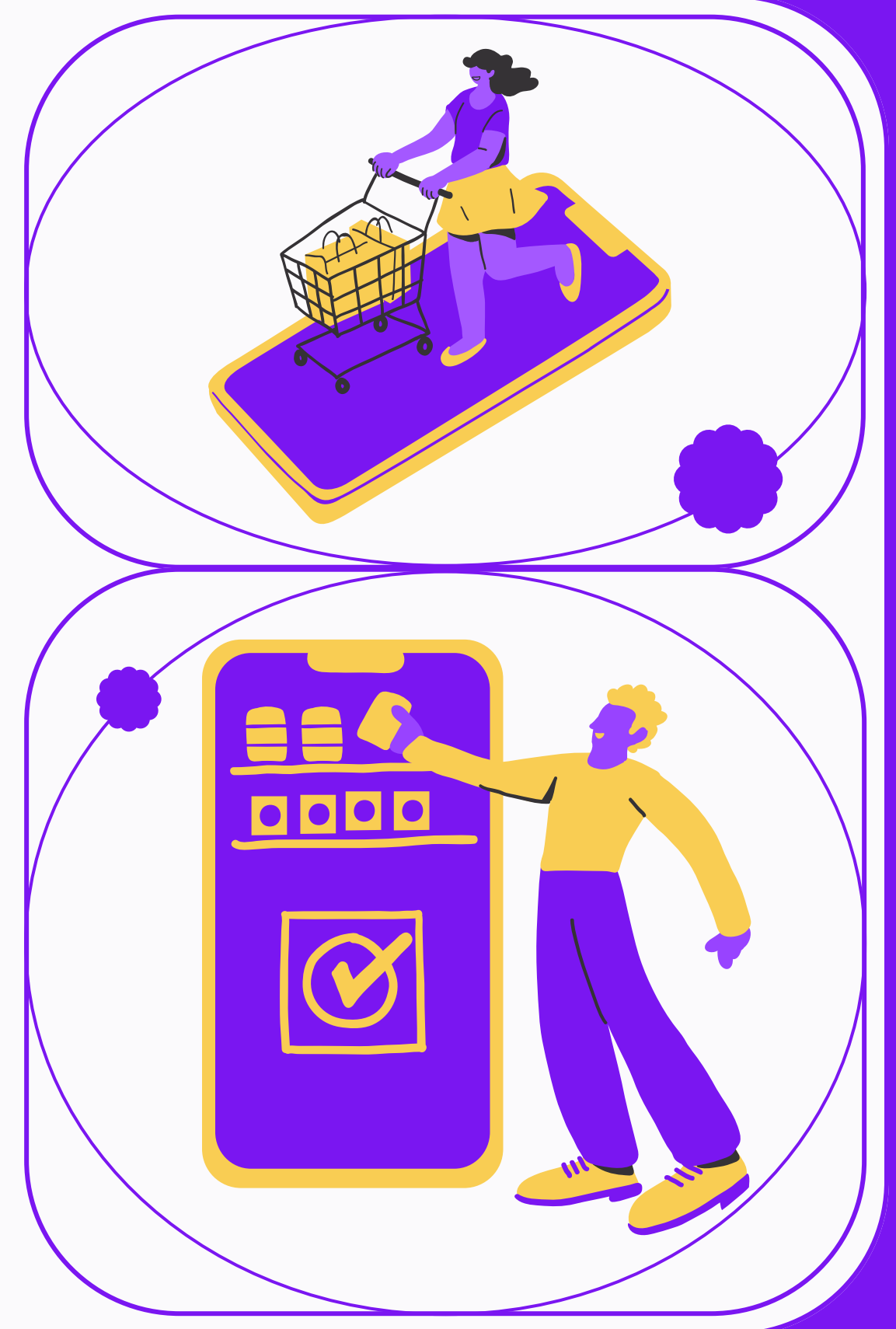
# 實作結果

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

# 實作結果

## 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。



## 實作結果

```
MariaDB [final]> Describe admin;
```

Field	Type	Null	Key	Default	Extra
admin_id	int(11)	NO	PRI	NULL	auto_increment
username	varchar(20)	NO	UNI	NULL	

```
2 rows in set (0.008 sec)
```

```
MariaDB [final]> select * from admin;
```

admin_id	username
11	admin1
12	admin2
19	admintest
17	johndoe
13	juliaadmin
14	superadmin
18	susanadmin
15	systemmgr
16	useradmin
20	webmaster

```
10 rows in set (0.000 sec)
```

管理員之資料表



## 實作結果

```
MariaDB [final]> Describe Customer;
```

Field	Type	Null	Key	Default	Extra
customer_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(20)	NO		NULL	
email	varchar(255)	NO	UNI	NULL	
phone_number	varchar(10)	NO		NULL	
gender	enum('男','女','其他')	NO		NULL	
birthdate	date	NO		NULL	

```
6 rows in set (0.007 sec)
```

```
MariaDB [final]> select * from Customer;
```

customer_id	name	email	phone_number	gender	birthdate
1	Alice123	alice123@example.com	0912345678	女	1990-05-10
2	BobChen01	bobchen01@example.com	0987654321	男	1985-12-01
3	CathyWang9	cathyw9@example.com	0911122233	其他	1992-07-23
4	DavidLiu88	davidliu88@example.com	0922334455	男	1988-03-14
5	EmmaWu77	emmawu77@example.com	0933445566	女	1995-10-30
6	FrankHuang	frankh@example.com	0944556677	男	1979-08-22
7	GraceTsail	gracetsail@example.com	0955667788	女	1983-01-18
8	HenryYeh22	henryyeh22@example.com	0966778899	男	1991-06-05
9	IreneKuo3	irenekuo3@example.com	0977889900	女	1996-11-11
10	JackyHsu44	jackyhsu44@example.com	0988990011	男	1980-07-27

```
10 rows in set (0.000 sec)
```

買家之資料表

## 實作結果

```
MariaDB [final]> Describe seller;
```

Field	Type	Null	Key	Default	Extra
seller_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(20)	NO		NULL	
email	varchar(255)	NO	UNI	NULL	
phone_number	varchar(10)	NO		NULL	
gender	enum('男','女','其他')	NO		NULL	
birthdate	date	NO		NULL	

```
6 rows in set (0.007 sec)
```

```
MariaDB [final]> select * from seller;
```

seller_id	name	email	phone_number	gender	birthdate
1	ShopA123	shopa123@example.com	0912345678	男	1980-05-10
2	ShopB456	shopb456@example.com	0922334455	女	1985-12-01
3	ShopC789	shopc789@example.com	0933445566	其他	1990-07-23
4	GadgetWorld	gadgetworld@example.com	0944556677	男	1975-03-14
5	BookHaven1	bookhaven1@example.com	0955667788	女	1988-10-30
6	FashionZone	fashionzone@example.com	0966778899	男	1995-08-22
7	TechHub123	techhub123@example.com	0977889900	女	1992-01-18
8	KitchenPro	kitchenpro@example.com	0988990011	其他	1983-06-05
9	ToyPlanetX	toyplanetx@example.com	0911222333	男	1991-11-11
10	GreenGarden	greengarden@example.com	0922333444	女	1979-07-27

```
10 rows in set (0.000 sec)
```

賣家之資料表

## 實作結果

```
MariaDB [final]> Describe `order`;
```

Field	Type	Null	Key	Default	Extra
order_id	int(11)	NO	PRI	NULL	auto_increment
order_date	datetime	NO		current_timestamp()	
status	enum('未付款','處理中','已出貨','已完成')	NO		NULL	
customer_id	int(11)	NO	MUL	NULL	

```
4 rows in set (0.007 sec)
```

```
MariaDB [final]> select * from `order`;
```

order_id	order_date	status	customer_id
1	2025-06-12 17:52:35	處理中	1
2	2025-06-12 17:52:35	已完成	2
3	2025-06-12 17:52:35	已出貨	3
4	2025-06-12 17:52:35	未付款	4
5	2025-06-12 17:52:35	處理中	5
6	2025-06-12 17:52:35	已完成	6
7	2025-06-12 17:52:35	已出貨	7
8	2025-06-12 17:52:35	處理中	8
9	2025-06-12 17:52:35	未付款	9
10	2025-06-12 17:52:35	處理中	10

```
10 rows in set (0.000 sec)
```

訂單之資料表

## 實作結果

```
MariaDB [final]> Describe orderitem;
```

Field	Type	Null	Key	Default	Extra
order_id	int(11)	NO	PRI	NULL	
product_id	int(11)	NO	PRI	NULL	
quantity	int(11)	NO		NULL	
price_at_purchase	decimal(10,2)	NO		NULL	

```
4 rows in set (0.005 sec)
```

```
MariaDB [final]> select * from orderitem;
```

order_id	product_id	quantity	price_at_purchase
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

```
10 rows in set (0.000 sec)
```

訂單項目之資料表

## 實作結果

```
MariaDB [final]> Describe product;
```

Field	Type	Null	Key	Default	Extra
product_id	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	NO		NULL	
price	decimal(10,2)	NO		NULL	
stock	int(11)	NO		NULL	
category	varchar(50)	YES		NULL	
seller_id	int(11)	NO	MUL	NULL	

```
6 rows in set (0.007 sec)
```

```
MariaDB [final]> select * from product;
```

product_id	name	price	stock	category	seller_id
1	Wireless Mouse	599.00	100	Electronics	1
2	Bluetooth Keyboard	899.00	80	Electronics	1
3	USBC Charger	299.00	150	Accessories	2
4	Laptop Stand	799.00	50	Office	3
5	Notebook	59.00	200	Stationery	5
6	TShirt	399.00	120	Clothing	6
7	Coffee Maker	1299.00	30	Kitchen	8
8	Action Figure	499.00	75	Toys	9
9	Garden Shovel	199.00	60	Garden	10
10	LED Desk Lamp	699.00	40	Lighting	4

```
10 rows in set (0.000 sec)
```

商品之資料表

## 實作結果

```
MariaDB [final]> Describe Review;
```

Field	Type	Null	Key	Default	Extra
review_id	int(11)	NO	PRI	NULL	auto_increment
product_id	int(11)	NO	MUL	NULL	
customer_id	int(11)	NO	MUL	NULL	
rating	int(11)	YES		NULL	
comment	text	YES		NULL	
review_date	datetime	YES		current_timestamp()	

```
6 rows in set (0.007 sec)
```

```
MariaDB [final]> select * from Review;
```

review_id	product_id	customer_id	rating	comment	review_date
11	1	1	5	非常好用，品質也很棒	2025-06-12 17:52:45
12	2	2	4	打字手感不錯，就是有點重	2025-06-12 17:52:45
13	3	3	3	充電頭普通，不過夠用	2025-06-12 17:52:45
14	4	4	5	很穩定的支架，推薦	2025-06-12 17:52:45
15	5	5	4	紙質不錯，適合書寫	2025-06-12 17:52:45
16	6	6	2	材質一般，容易皺	2025-06-12 17:52:45
17	7	7	5	煮咖啡很方便，滿意	2025-06-12 17:52:45
18	8	8	4	小朋友很喜歡	2025-06-12 17:52:45
19	9	9	3	有點小貴，但好用	2025-06-12 17:52:45
20	10	10	5	燈光柔和，很適合夜讀	2025-06-12 17:52:45

```
10 rows in set (0.000 sec)
```

商品評價之資料表

# 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





Thank  
You

