



資料庫系統期末報告

虎尾美食推薦系統



Group 14

41043104 郭俞汎

41043146 張哲維

41143237 陳昱霖

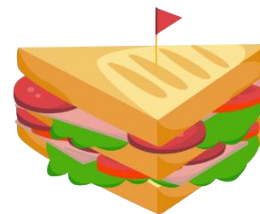
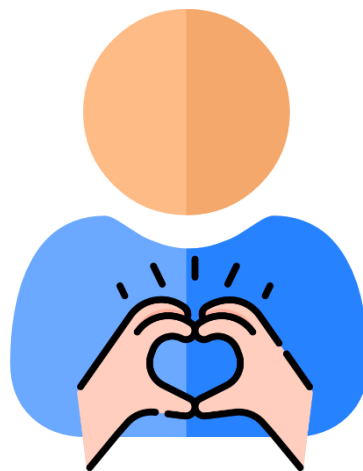
41143254 鄭名男



總覽



虎尾美食推薦系統旨在幫助使用者根據個人喜好尋找虎尾地區的合適餐廳。使用者可以註冊、登入、選擇喜好類別，並接收個人化的餐廳推薦。管理員可以管理餐廳資訊、喜好類別，並查看會員數量統計，但無法存取個人用戶資料。





虎尾美食推薦系統



目錄

1. 應用情境
2. 系統需求說明
3. 使用案例
4. 資料概念層模型
5. ER Diagram
6. 資料庫 SCHEMA
7. View SQL
8. 使用者權限SQL
9. 組員分工與心得

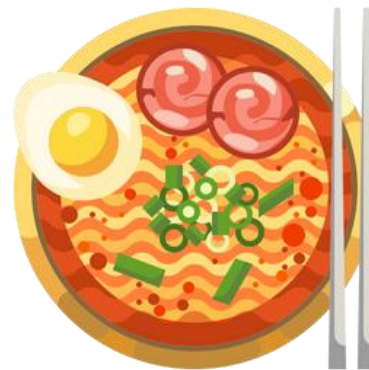


應用情境

初來乍到

情境描述

小明和他的三個朋友剛從台北開車到虎尾渡假，這是他們第一次來虎尾，對當地的美食店家一無所知。到了晚上，大家都開始肚子餓了，想要找間不錯的餐廳用餐。



系統應用

1. 小明打開虎尾美食推薦系統，進行註冊。
2. 登入後，小明勾選了「燒肉」、「丼飯」和「甜食」三個類別。
3. 系統根據喜好，產生一份推薦清單。
4. 透過打開推薦清單中的 Google 地圖連結，小明看到「色鼎燒肉」這家店的介紹：提供燒肉、丼飯和甜點等多種菜色，營業到晚上 11 點，離小明下榻的旅館也不太遠。
5. 小明點開看到具體位置和路線導航後，決定就帶著三個朋友前往這家餐廳用餐。

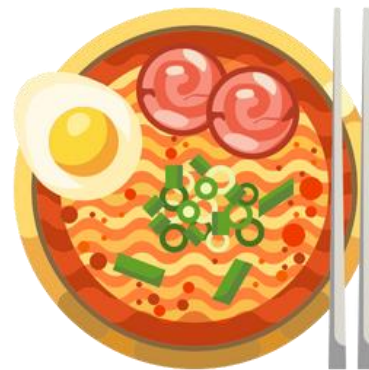


應用情境

熱門熟路

情境描述

老王是虎尾科大的學生，住在虎尾兩年了，他不喜歡每天都吃同樣的餐點，但卻想不到要吃什麼，詢問同學也得到了「不知道欸？你要吃什麼」的回答。



系統應用

1. 老王打開虎尾美食推薦系統，登入後，發現自己在上次選擇了「火鍋」這個類別。
2. 老王取消掉火鍋的選項，改選擇了「麵類」。
3. 系統根據老王的喜好，過濾和分析虎尾所有餐廳的菜色類別，很快產生一份推薦名單。
4. 在推薦清單中，老王看到「客家庄」、「熬鍋燒」等店家，他們都有販賣麵類，透過打開推薦清單中的 Google 地圖連結後發現「熬鍋燒」比較便宜，於是老王選擇了「熬鍋燒」。
5. 老王找了朋友們一起騎車去吃「熬鍋燒」。



系統需求說明

系統概述

本系統旨在幫助使用者找尋虎尾地區的美食店家，根據其個人喜好進行推薦。系統主要功能包括管理員登入、會員登入、美食店家資料管理、會員喜好類別輸入、推薦項目清單產生等。





系統需求說明



功能需求

會員管理

- 使用者會員註冊
- 使用者會員登入
- 管理員查看會員數量(無法查看會員個人資料)

美食店家管理

- 管理員新增、刪除、更新店家資訊
- 使用者瀏覽店家列表



系統需求說明



功能需求

喜好類別管理

- 管理員新增、刪除喜好類別
- 管理員為店家新增類別
- 使用者設定自己的喜好類別

推薦項目生成

- 系統根據會員的喜好類別，隨機產生符合類別的 2 個推薦店家

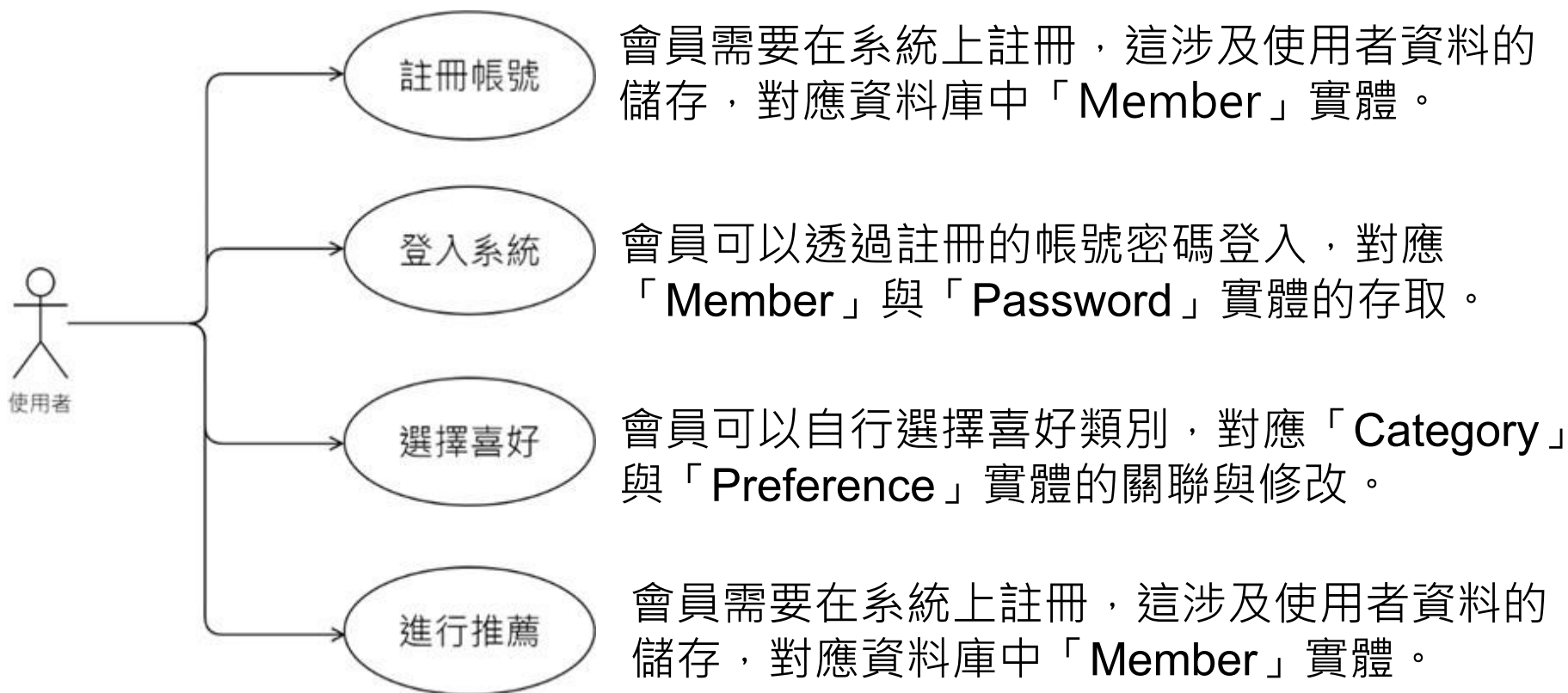
安全性

- 密碼獨立資料表儲存
- 一般管理員無法查看會員個人資料



使用案例

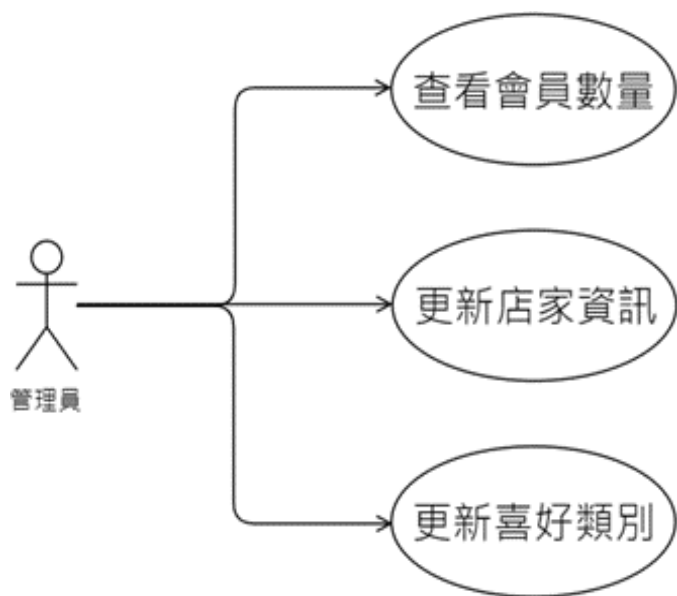
使用者角色(會員)





使用案例

管理員角色(餐廳資料管理者)



管理員可以在系統上查看會員數量，這涉及對「Member」實體數量的存取。

管理員可以在系統上更新店家的相關資訊，這涉及對「Restaurant」、「Hours」與「resCate」實體的修改。

管理員可以在系統上新增、刪除喜好類別，這涉及對「Category」實體的修改。

資料概念層模型

會員 (Member) 資料表

欄位名稱	說明	資料型態	是否為空	值域
mId	會員 ID	INT AUTO_INCREMENT	N	主鍵，自動遞增
mName	姓名	VARCHAR(12)	N	2~12 字元，限中文字或英文字母
mAccount	帳號	VARCHAR(20)	N	6~20 字元，限英文與數字組合
mEmail	電子郵件	VARCHAR(50)	N	Email 格式，'^[a-zA-Z0-9.] +@[a-zA-Z0-9.-] +.[a-zA-Z]{2,}\$'
mPhone	手機	CHAR(10)	Y	台灣手機格式，以 09 開頭後面接 8 個數字
mAddress	地址	VARCHAR(64)	Y	最長 64 字元的文字
mCreateDate	建立日期	DATE	N	格式為 yyyy-mm-dd，未設定時自動填入當前系統日期

資料概念層模型

喜好類別 (Category) 資料表

欄位名稱	說明	資料型態	是否為空	值域
cId	類別 ID	INT AUTO_INCREMENT	N	主鍵，自動遞增
cName	類別名稱	VARCHAR(16)	N	1~16 字元，限中文字或英文字母

資料概念層模型

營業時間 (Hours) 資料表

欄位名稱	說明	資料型態	是否為空	值域
rHoursId	營業時間 ID	INT	N	從 1 開始遞增的整數
day	星期	CHAR(3)	N	只能是固定的 Mon, Tue, Wed, Thu, Fri, Sat, Sun 這幾種
start_hr	開始小時	TINYINT	N	0~23 (24 小時制)，整數
start_min	開始分鐘	TINYINT	N	0~59，整數
end_hr	結束小時	TINYINT	N	0~23 (24 小時制)，整數
end_min	結束分鐘	TINYINT	N	0~59，整數

資料概念層模型

餐廳 (Restaurant) 資料表

欄位名稱	說明	資料型態	是否為空	值域
rId	店家 ID	INT AUTO_INCREMENT	N	主鍵，自動遞增
rName	店名	VARCHAR(30)	N	1~30 字元，中文或英文皆可
rAddress	地址	VARCHAR(64)	N	最長 64 字元的文字
rPhone	店家電話	VARCHAR(10)	Y	056 開頭後面 6 個數字的虎尾市話 或 09 開頭後面 8 個數字的行動電話
rHoursId	營業時間 ID	INT	Y	參考 hours.rHoursId
rLink	GoogleMap連結	VARCHAR(100)	Y	URL 格式，含 https://maps.app.goo.gl/ + 英文與數字的組合

資料概念層模型

店家類別對應表 (resCate)

欄位名稱	說明	資料型態	是否為空	值域
rld	店家 ID	INT	N	參考 restaurant.rld
cld	類別 ID	INT	N	參考 category.cld

會員喜好類別表 (Preference)

欄位名稱	說明	資料型態	是否為空	值域
mld	會員 ID	INT	N	參考 member.mld
cld	類別 ID	INT	N	參考 category.cld

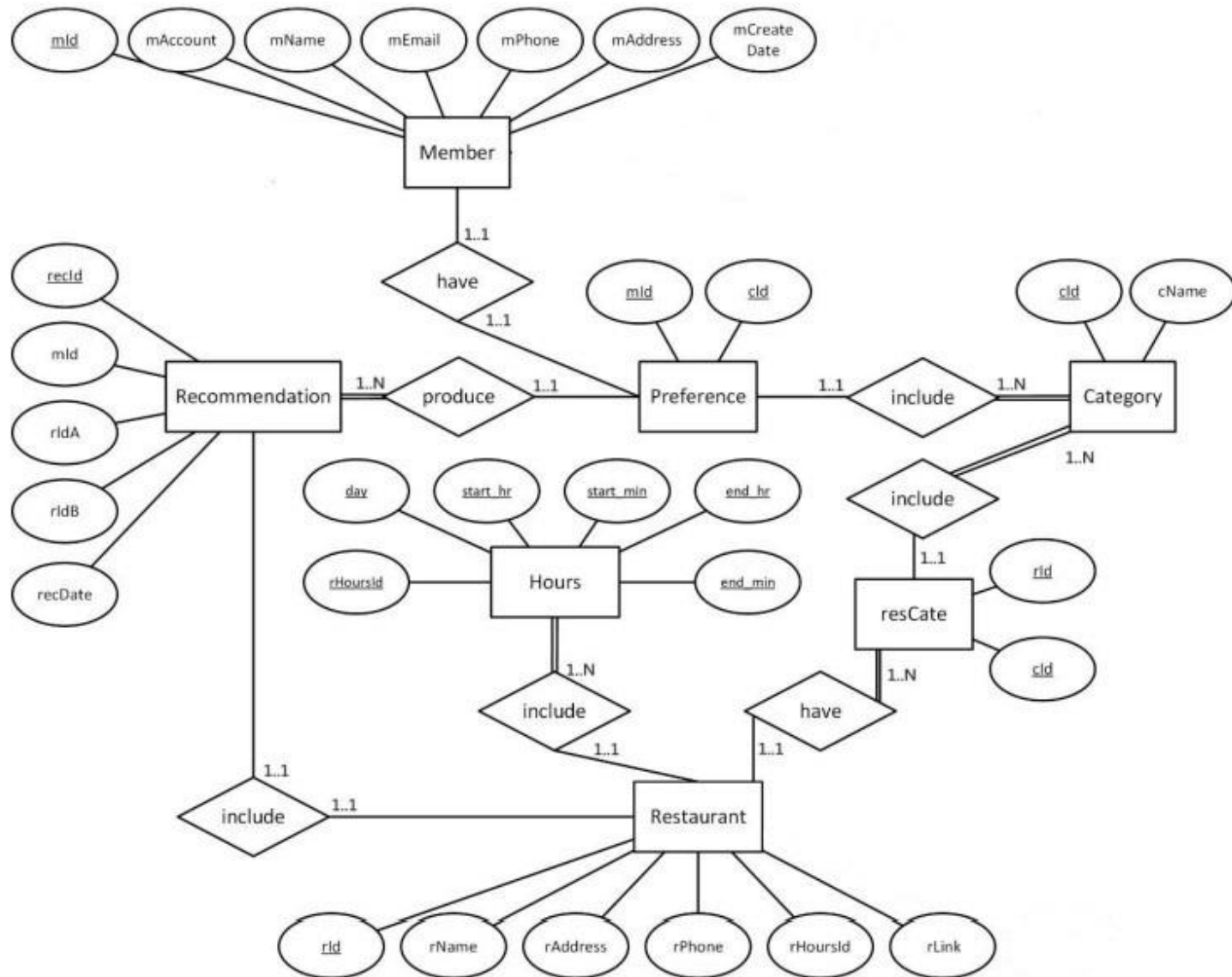
資料概念層模型

推薦清單表 (Recommendation)

欄位名稱	說明	資料型態	是否為空	值域
recId	推薦 ID	INT AUTO_INCREMENT	N	主鍵，自動遞增
mId	會員 ID	INT	N	參考 member.mId
rIdA	推薦店家 A ID	INT	N	參考 restaurant.rId
rIdB	推薦店家 B ID	INT	N	參考 restaurant.rId
recDate	推薦日期	DATE	N	格式為 yyyy-mm-dd，未設定時自動填入當前系統日期

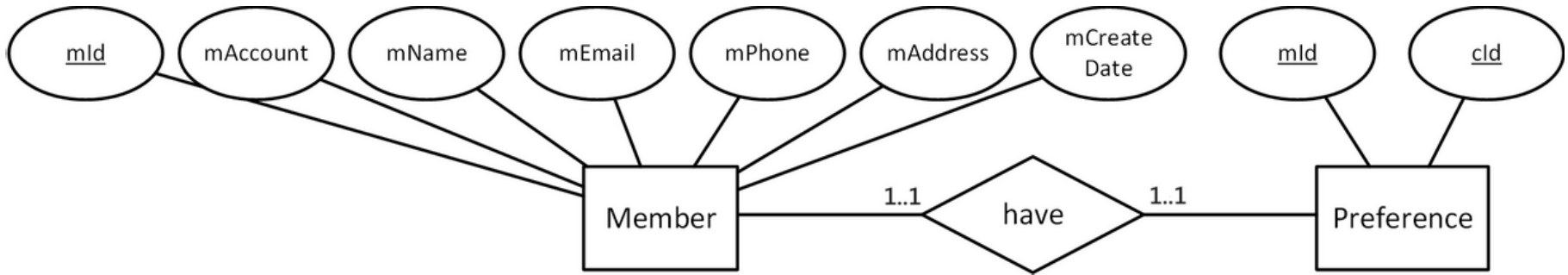


ER DIAGRAM





ER DIAGRAM



會員 (Member) 資料表屬性

- 會員ID (mId)
- 會員帳號 (mAccount)
- 姓名 (mName)
- 電子郵件 (mEmail)
- 連絡電話 (mPhone)
- 常用地址 (mAddress)
- 建立日期 (mCreateDate)

會員喜好類別 (Preference) 資料表屬性

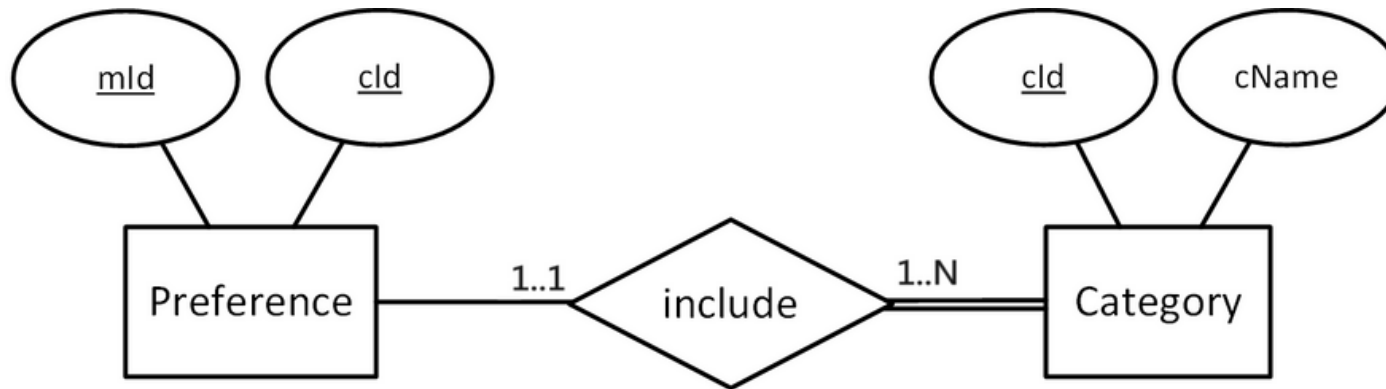
- 類別ID (cId)
- 類別名稱 (cName)

關聯

- 「Member」與「Preference」實體有一對一 (1..1) 的關係每個使用者對應到一整組喜好。



ER DIAGRAM



喜好類別 (Category) 資料表屬性

- 類別ID (cld)
- 類別名稱 (cName)

會員喜好類別 (Preference) 資料表屬性

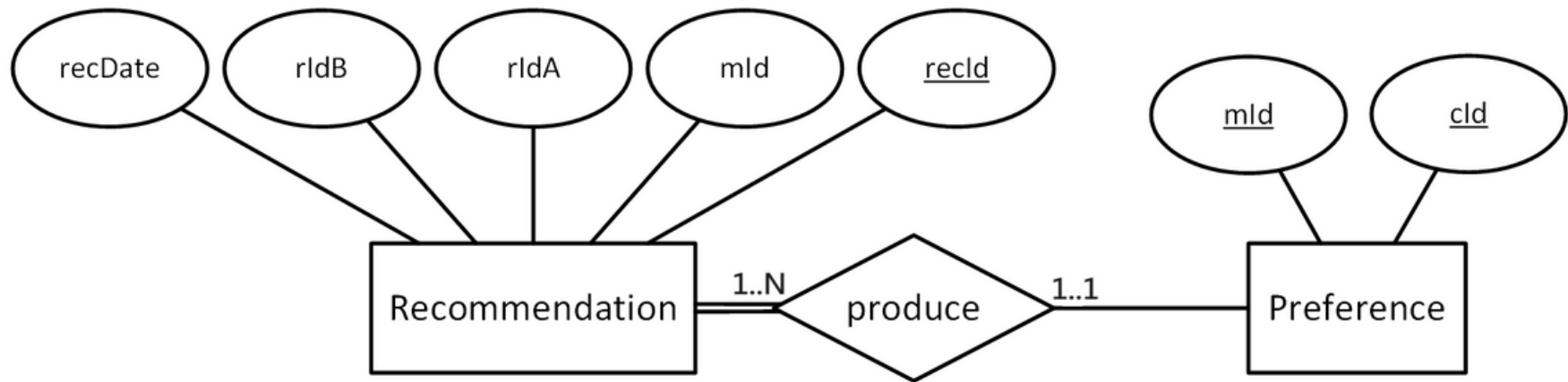
- 會員ID (mld)
- 類別ID (cld)

關聯

「Preference」與「Category」實體有一對多 (1..n) 的關係，每組使用者喜好對應到多個喜好類別。



ER DIAGRAM



推薦清單 (Recommendation) 資料表屬性

- 推薦ID (recId)
- 會員ID (mld)
- 店家1 ID (rldA)
- 店家2 ID (rldB)
- 推薦日期 (recDate)

會員喜好類別 (Preference) 資料表屬性

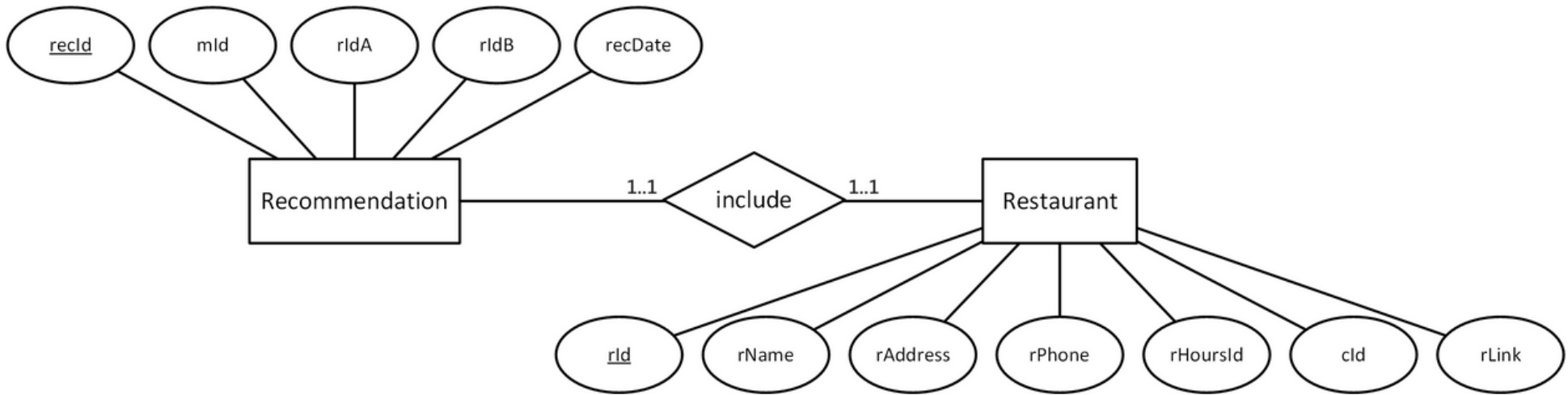
- 會員ID (mld)
- 類別ID (cld)

關聯

「Preference」與「Recommendation」實體有一對多 (1..n) 的關係，每組使用者喜好可以產生多個推薦清單。



ER DIAGRAM



推薦清單 (Recommendation) 資料表屬性

- 推薦ID (recId)
- 會員ID (mId)
- 店家1 ID (rIdA)
- 店家2 ID (rIdB)
- 推薦日期 (recDate)

關聯

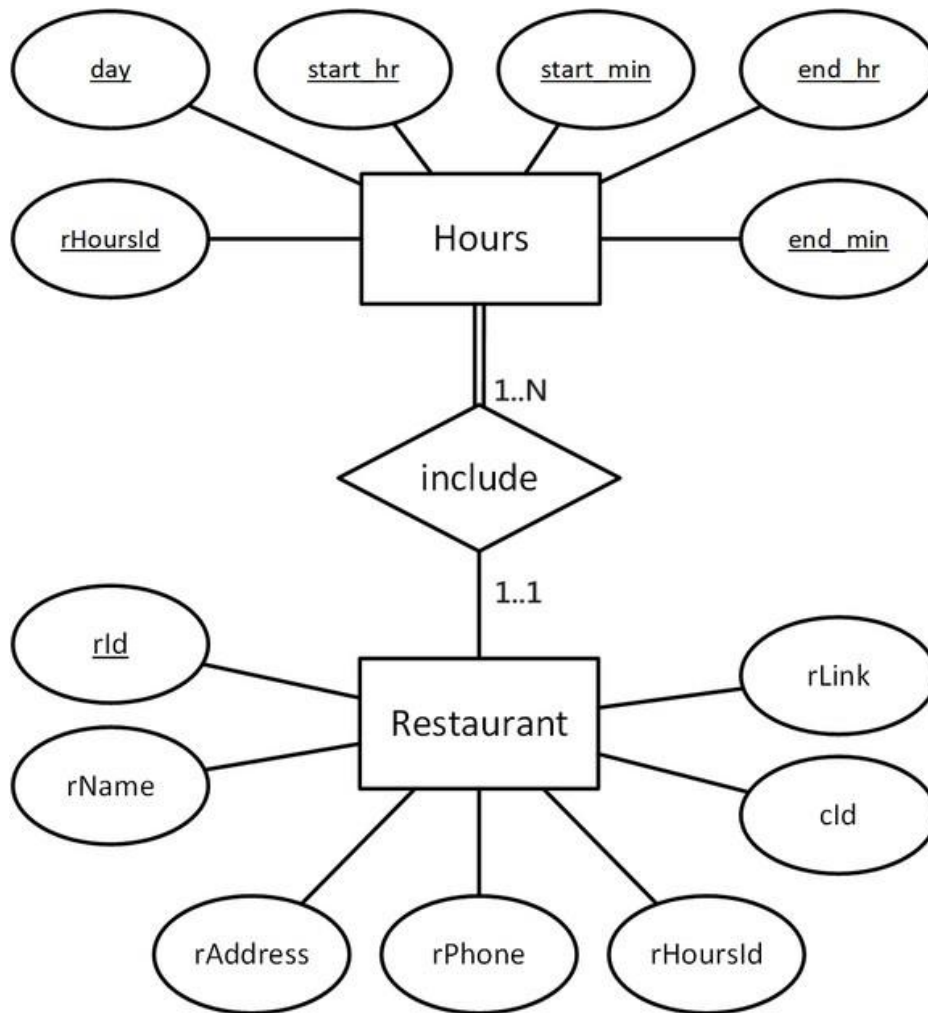
「Restaurant」與「Recommendation」實體有一對一 (1..1) 的關係，每個店家對應到一組推薦清單。

餐廳 (Restaurant) 資料表屬性

- 店家ID (rId)
- 店家名稱 (rName)
- 店家地址 (rAddress)
- 店家電話 (rPhone)
- 營業時間ID (rHoursId)
- 類別ID (cId)
- Google Map 連結 (rLink)



ER DIAGRAM



營業時間 (Hours) 資料表屬性

- 營業時間ID (rHoursId)
- 營業星期 (day)
- 營業開始時 (start_hr)
- 營業開始分 (start_min)
- 營業結束時 (end_hr)
- 營業結束分 (end_min)

餐廳 (Restaurant) 資料表屬性

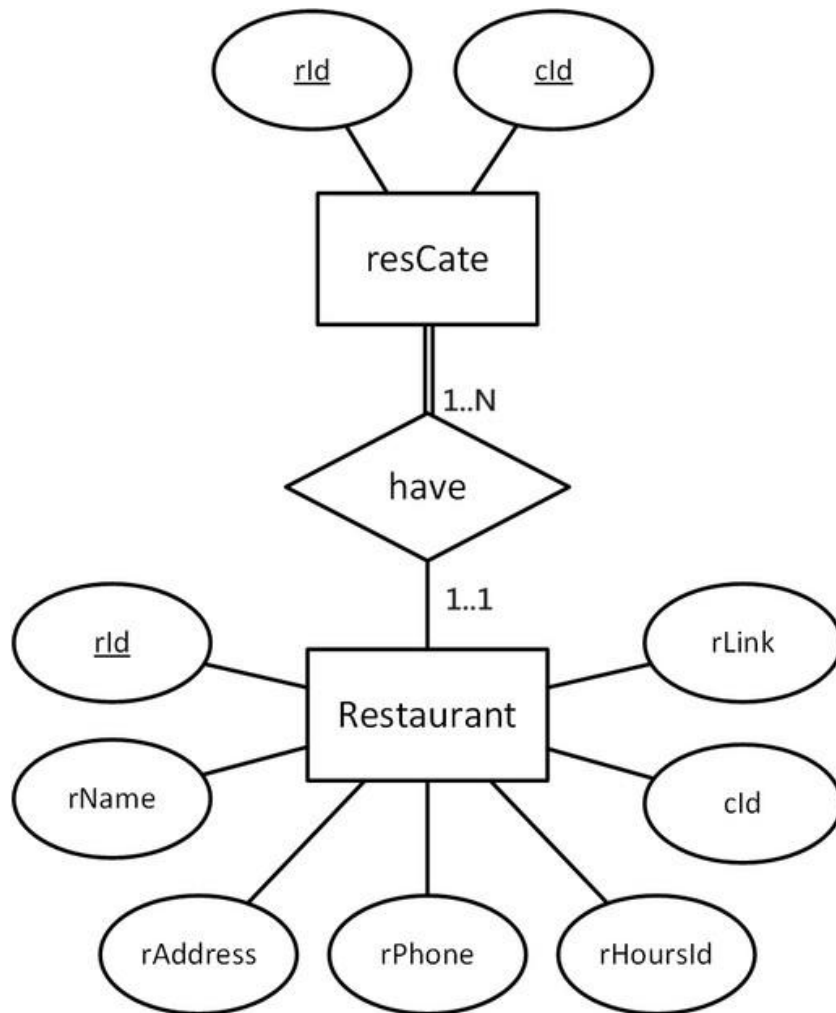
- 店家ID (rId)
- 店家名稱 (rName)
- 店家地址 (rAddress)
- 店家電話 (rPhone)
- 營業時間ID (rHoursId)
- 類別ID (cId)
- Google Map 連結 (rLink)

關聯

「Restaurant」與「Hours」實體有一對多 (1..n) 的關係，每個店家有不同的營業時段。



ER DIAGRAM



店家類別 (resCate) 資料表屬性

- 店家ID (rld)
- 類別ID (cld)

餐廳 (Restaurant) 資料表屬性

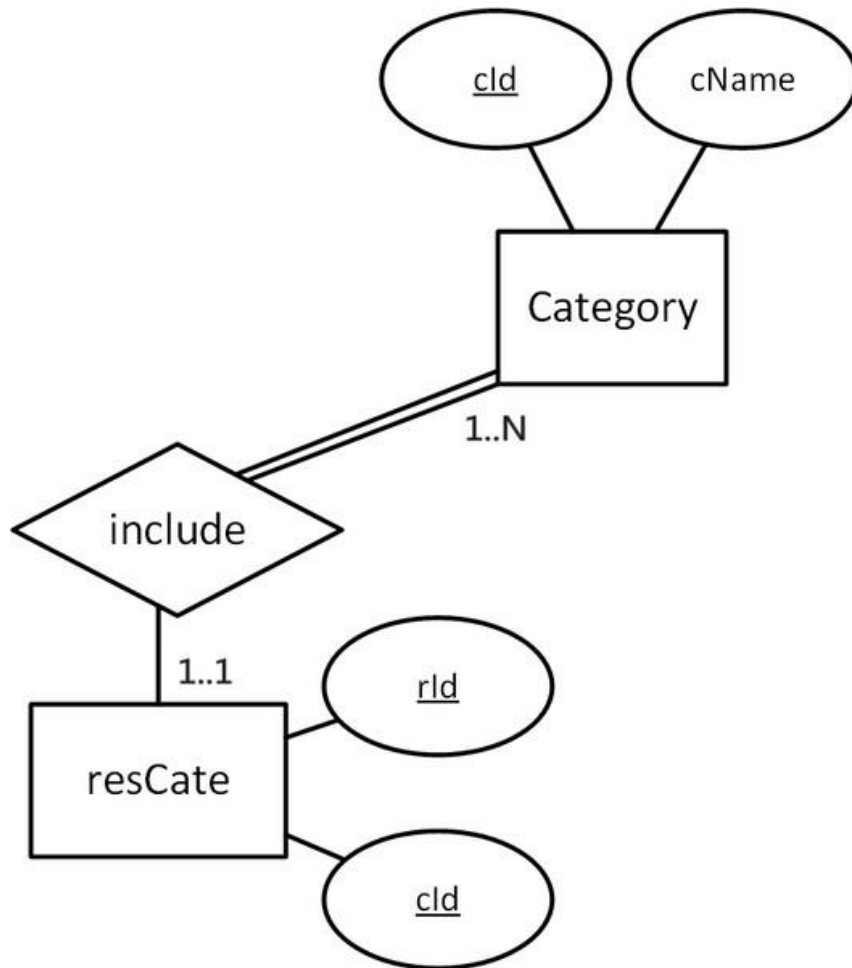
- 店家ID (rld)
- 店家名稱 (rName)
- 店家地址 (rAddress)
- 店家電話 (rPhone)
- 營業時間ID (rHoursId)
- 類別ID (cld)
- Google Map 連結 (rLink)

關聯

「Restaurant」與「resCate」實體有一對多 (1..n) 的關係，每個店家對應到多個類別。



ER DIAGRAM



喜好類別 (Category) 資料表屬性

- 類別ID (cld)
- 類別名稱 (cName)

店家類別 (resCate) 資料表屬性

- 店家ID (rld)
- 類別ID (cld)

關聯

「resCate」與「Category」實體有一對多 (1..n) 的關係，每個店家類別對應到多個喜好類別。



資料庫 SCHEMA

會員 (Member)資料表 SQL

```
CREATE TABLE Member (  
    mId INT NOT NULL AUTO_INCREMENT,  
    mName VARCHAR(12) NOT NULL,  
    mAccount VARCHAR(20) NOT NULL,  
    mEmail VARCHAR(50) NOT NULL,  
    mPhone CHAR(10),  
    mAddress VARCHAR(64),  
    mCreateDate DATE NOT NULL DEFAULT CURRENT_DATE,  
    CHECK (mAccount REGEXP '^[a-zA-Z0-9._]{6,20}$'),  
    CHECK (CHAR_LENGTH(mName) >= 2),  
    CHECK (mEmail REGEXP '^[a-zA-Z0-9._]+@[a-zA-Z0-9._]+\.[a-zA-Z]{2,}$'),  
    CHECK (mPhone IS NULL OR mPhone REGEXP '^09[0-9]{8}$'),  
    CHECK (mAddress IS NULL OR CHAR_LENGTH(mAddress) >= 1),  
    PRIMARY KEY (mId),  
    UNIQUE (mAccount),  
    UNIQUE (mEmail)  
);
```



資料庫 SCHEMA

喜好類別 (Category) 資料表 SQL

```
-- 喜好類別 (Category) 資料表  
CREATE TABLE Category (  
    cId INT NOT NULL AUTO_INCREMENT,  
    cName VARCHAR(16) NOT NULL,  
    PRIMARY KEY (cId),  
    CHECK (CHAR_LENGTH(cName) >= 1)  
    UNIQUE (cName)  
);
```



資料庫 SCHEMA

營業時間 (Hours) 資料表 SQL

```
-- 營業時間 (Hours) 資料表
CREATE TABLE Hours (
    rHoursId INT NOT NULL,
    day CHAR(3) NOT NULL,
    start_hr NUMERIC(2) CHECK (start_hr >= 0 AND start_hr <= 24),
    start_min NUMERIC(2) CHECK (start_min >= 0 AND start_min < 60),
    end_hr NUMERIC(2) CHECK (end_hr >= 0 AND end_hr <= 24),
    end_min NUMERIC(2) CHECK (end_min >= 0 AND end_min < 60),
    PRIMARY KEY (rHoursId, day, start_hr, start_min, end_hr, end_min),
    CHECK (day IN ('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'))
);
```



資料庫 SCHEMA

餐廳 (Restaurant) 資料表 SQL

```
-- 餐廳 (Restaurant) 資料表
CREATE TABLE Restaurant (
  rId INT NOT NULL AUTO_INCREMENT,
  rName VARCHAR(30) NOT NULL,
  rAddress VARCHAR(64) NOT NULL,
  rPhone VARCHAR(10),
  rHoursId INT,
  rLink VARCHAR(100),
  PRIMARY KEY (rId),
  FOREIGN KEY (rHoursId) REFERENCES Hours(rHoursId) ON DELETE SET NULL,
  CHECK (CHAR_LENGTH(rName) >= 1),
  CHECK (CHAR_LENGTH(rAddress) >= 1),
  CHECK (rLink IS NULL OR rLink REGEXP '^https://maps\.app\.goo\.gl/.*$'),
  CHECK (rPhone IS NULL OR rPhone REGEXP '^056[0-9]{6}|09[0-9]{8}$')
);
```



資料庫 SCHEMA

店家類別對應表 (resCate) SQL

```
-- 店家類別 (resCate) 資料表
CREATE TABLE resCate (
    rId INT NOT NULL,
    cId INT NOT NULL,
    PRIMARY KEY (rId, cId),
    FOREIGN KEY (rId) REFERENCES Restaurant(rId) ON DELETE CASCADE,
    FOREIGN KEY (cId) REFERENCES Category(cId) ON DELETE CASCADE
);
```



資料庫 SCHEMA

會員喜好類別表 (Preference) SQL

```
-- 會員喜好類別 (Preference) 資料表
CREATE TABLE Preference (
    mId INT NOT NULL,
    cId INT NOT NULL,
    PRIMARY KEY (mId, cId),
    FOREIGN KEY (mId) REFERENCES Member(mId) ON DELETE CASCADE,
    FOREIGN KEY (cId) REFERENCES Category(cId) ON DELETE CASCADE
);
```



資料庫 SCHEMA

推薦清單表(Recommendation) SQL

```
-- 推薦清單 (Recommendation) 資料表
CREATE TABLE Recommendation (
    recId INT NOT NULL AUTO_INCREMENT,
    mId INT NOT NULL,
    rIdA INT NOT NULL,
    rIdB INT NOT NULL,
    recDate DATE NOT NULL DEFAULT CURRENT_DATE,
    PRIMARY KEY (recId),
    FOREIGN KEY (mId) REFERENCES Member(mId) ON DELETE CASCADE,
    FOREIGN KEY (rIdA) REFERENCES Restaurant(rId) ON DELETE CASCADE,
    FOREIGN KEY (rIdB) REFERENCES Restaurant(rId) ON DELETE CASCADE
);
```



VIEW SQL

會員喜好 View : user_preferences
用途：彙整會員的喜好類別

```
SELECT
    m.mID,
    m.mName,
    m.mAccount,
    GROUP_CONCAT(c.cName SEPARATOR ', ') AS preferences
FROM member m
LEFT JOIN preference p ON m.mID = p.mID
LEFT JOIN category c ON p.cID = c.cID
GROUP BY m.mID, m.mName, m.mAccount;
```




VIEW SQL

營業中餐廳 View : open_restaurants

用途：查詢目前時間仍在營業的餐廳

```
SELECT DISTINCT
    r.rID,
    r.rName,
    r.rAddress,
    r.rPhone,
    r.rLink
FROM restaurant r
JOIN hours h ON r.rHoursID = h.rHoursID
WHERE h.day = LEFT(DAYNAME(NOW()), 3)
AND (
    (h.start_hr < HOUR(NOW()) OR
     (h.start_hr = HOUR(NOW()) AND h.start_min <= MINUTE(NOW())))
AND
    (h.end_hr > HOUR(NOW()) OR
     (h.end_hr = HOUR(NOW()) AND h.end_min >= MINUTE(NOW())))
);
```



VIEW SQL

推薦候選餐廳 View : recommendation_candidates

**用途：計算會員與餐廳間的喜好配對數量
(供推薦邏輯使用)**

```
SELECT
    p.mID,
    r.rID,
    r.rName,
    r.rAddress,
    r.rPhone,
    r.rLink,
    COUNT(DISTINCT rc.cID) AS matching_categories
FROM preference p
JOIN resCate rc ON p.cID = rc.cID
JOIN restaurant r ON rc.rID = r.rID
GROUP BY p.mID, r.rID, r.rName, r.rAddress, r.rPhone, r.rLink
ORDER BY p.mID, matching_categories DESC;
```



VIEW SQL

推薦記錄 View : recent_recommendations

用途：顯示推薦歷史記錄與對應餐廳名稱

```
SELECT
    rec.recID,
    m.mName,
    m.mAccount,
    r1.rName AS restaurant_1,
    r2.rName AS restaurant_2,
    rec.recDate
FROM recommendation rec
JOIN member m ON rec.mID = m.mID
JOIN restaurant r1 ON rec.rIDA = r1.rID
JOIN restaurant r2 ON rec.rIDB = r2.rID
ORDER BY rec.recDate DESC, rec.recID DESC;
```



VIEW SQL

系統統計 View : admin_statistics

用途：顯示會員總數、餐廳數、類別數與推薦記錄數

```
SELECT
    (SELECT COUNT(*) FROM member) AS total_members,
    (SELECT COUNT(*) FROM restaurant) AS total_restaurants,
    (SELECT COUNT(*) FROM category) AS total_categories,
    (SELECT COUNT(*) FROM recommendation) AS total_recommendations;
```



使用者權限SQL

```
-- 1. 建立會員使用者帳號
CREATE USER 'user_general'@'%' IDENTIFIED BY 'userpassword123';

-- 2. 會員使用者對 huwei_food 資料庫中的操作權限
GRANT SELECT, INSERT, UPDATE, DELETE ON huwei_food.Preference TO 'user_general'@'%';
GRANT SELECT ON huwei_food.recommendation TO 'user_general'@'%';

-- 3. 授權會員使用者查詢view
GRANT SELECT ON huwei_food.user_preferences TO 'user_general'@'%';
GRANT SELECT ON huwei_food.open_restaurants TO 'user_general'@'%';
GRANT SELECT ON huwei_food.recommendation_candidates TO 'user_general'@'%';
GRANT SELECT ON huwei_food.recent_recommendations TO 'user_general'@'%';
GRANT SELECT ON huwei_food.restaurant_details TO 'user_general'@'%';

-- 4. 建立餐廳資料管理者帳號
CREATE USER 'user_admin'@'%' IDENTIFIED BY 'userpassword789';

-- 5. 餐廳資料管理者對 huwei_food 資料庫中的操作權限
GRANT SELECT, INSERT, UPDATE, DELETE ON huwei_food.category TO 'user_admin'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON huwei_food.restaurant TO 'user_admin'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON huwei_food.rescate TO 'user_admin'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON huwei_food.hours TO 'user_admin'@'%';

-- 6. 授權餐廳資料管理者查詢view
GRANT SELECT ON huwei_food.admin_statistics TO 'user_admin'@'%';

FLUSH PRIVILEGES;
```



組員 分工

學號與姓名	負責工作
41043104 郭俞汎	資料概念層模型規劃 功能需求規劃 ER Diagram 規劃與實作 店家資料彙整 資料庫 Schema 實作 資料庫 View 實作 GITHUB撰寫 書面報告整理、簡報優化
41043146 張哲維	應用情境與使用案例規劃 ER Diagram實作 店家資料蒐集 資料庫 Schema 實作 資料庫 View 實作 簡報製作
41143237 陳昱霖	功能需求規劃 值域規劃 資料庫 Schema 實作 資料庫 View 實作 GITHUB撰寫
41143254 鄭名男	資料概念層模型規劃 店家資料彙整 資料庫 Schema 實作 資料庫 View 實作 前端實作



組員心得



組員心得

41043104 郭俞汎

這次的資料庫期末專題是我第一次實作資料庫系統，挑戰性很高。我們選擇了「虎尾美食推薦系統」作為題目，主題貼近日常生活，讓我們在設計與開發過程中充滿動力。從資料蒐集、資料表設計到功能實作，過程中遇到許多困難，如資訊即時更新、資料表結構需重建等，但也因此更熟悉資料庫邏輯與語法。我們還運用了日期判斷函式，避免推薦到公休店家，提升系統實用性與使用者體驗。這次專題不僅讓我熟悉資料庫基礎，也學到許多課堂上未提及的進階語法與實務技巧，是一次充滿挑戰也收穫滿滿的學習經驗。



組員心得



組員心得

41043146 張哲維

通過這次的資料庫專題，我們學到了很多的知識，尤其是在建立和管理資料庫方面。我們的專題是虎尾美食推薦系統，首先，我們需要一個一個地查找虎尾的店家資料，這是一個耗時的過程，這也讓我們學會了如何有效地收集和整理大量資料。在完成資料收集後，我們還需要考慮系統的完整性，這包括設計資料庫結構、設定資料表之間的關聯、確保資料的一致性等，通過團隊的共同努力和討論，我們最後成功地克服了這些困難。這次專題讓我們學會了如何建立和管理資料庫，我們對資料庫系統有了更深入的理解。



組員心得



組員心得

41143237 陳昱霖

在實作資料庫的完整性限制時，我學到如何用 **CHECK**、**FOREIGN KEY** 和 **UNIQUE** 來確保資料的正確性，像是在 **Restaurant** 表中用 **CHECK** 限制 **rName** 不能為空，還有用正規表達式檢查電話格式。**FOREIGN KEY** 讓表之間的關聯更穩固，比如 **Recommendation** 表參考 **Member** 和 **Restaurant**，刪除時能自動處理。觸發器也很實用，像是在 **Restaurant** 表把空電話轉成 **NULL**，減少錯誤。這些限制讓資料庫更可靠，確保資料不易出錯。這次實作讓我更懂怎麼設計一個穩健的系統。



組員心得



組員心得

41143254 鄭名男

這次專題讓我學會了Schema規劃、輸入限制與API實作等技能。起初規劃的八個資料表，實作時卻發現 restaurant 表的cid與rescate表重複，導致資料冗餘；rHoursID的外鍵也使新增餐廳失敗，經過多次重新修正才規劃出目前較為完整的資料庫系統。後端開發也遇到很多問題：在 port 5000 執行時出現 404 和 500 錯誤，debug後發現是多個Flask實例衝突，改用 port 5001才排除問題、輸入驗證不足導致造成資料寫入失敗、店家電話格式處理、推薦功能中，當僅有一間餐廳匹配時，rIDB會重複rIDA，後來加上匹配數檢查並允許 rIDB 為 NULL，使結果更合理等。這些問題很大程度提升了我的實作與問題解決能力。



Thanks For Listening