

第十一章 多表格資料庫設計

11-1 範例研討 – 客戶回應系統

11-1-1 需求分析與資料收集

『神氣飛昂網路行銷公司』期望在銷售網站上加入『客戶回應系統』，利用此它建立與客戶之間的溝通管道，也藉此探討客戶對公司的評價如何？期望系統的功能如下：

- 紀錄客戶反映訊息。
- 可查詢哪產品類別的反應訊息。
- 可依照日期查詢客戶訊息。
- 可依照客戶姓名、性別、年齡查詢回應訊息。
- 訊息包含滿意度調查：1~5分，5分最高。
- 可分類評估某一類別產品滿意度如何。
- 可依製造廠商評估產品滿意度。

另外，系統分析師也到該公司收集了下列資料：

- 客戶資料：姓名、電話、性別、年齡、電子郵件、地址。
- 商品資料：製造廠商、產品分類(衣服、褲子、包包、...)。

11-1-2 資料庫邏輯設計

我們由外往內拆解方法，來分析此系統的邏輯架構。

(A) 系統需求

此系統僅提供客戶登陸回應訊息的功能，如下圖所示。

設計

客戶回應系統



圖 11-1 客戶回應系統概況

(B) 系統 E-R 關係圖

此系統只要兩個實體，一者為客戶實體，另一者為訊息實體，我們將收集的資料依其屬性分別歸類於這兩實體上，如下圖所示：

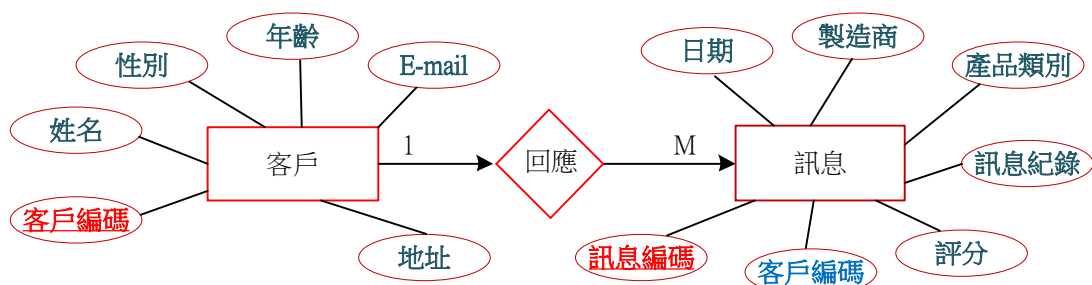


圖 11-2 歸類兩實體

建立 E-R 圖重點如下：

1. 客戶實體內沒有唯一識別值，產生一個屬性『客戶編碼』，並令為主鍵。
2. 訊息實體上也沒有唯一識別值，也產生『訊息編碼』屬性當為主鍵。
3. 為了讓兩實體之間產生關聯，在訊息實體上增加了一個外來鍵為『客戶編碼』，並參考到客戶實體的『客戶編碼』屬性。
4. 同一位客戶允許發送多筆訊息，但某一訊息只允許一位客戶發送，是『多對一』關係。

得到該系統的 E-R 關係圖如圖 11-3 所示。



設計

圖 11-3 系統 E-R 關係圖

11-1-3 E-R 圖轉換資料表

(A) 客戶實體轉換

客戶實體包含了 6 個屬性(Attribute)，且客戶編碼是唯一識別碼，並將它指定為主鍵(Primary key)。

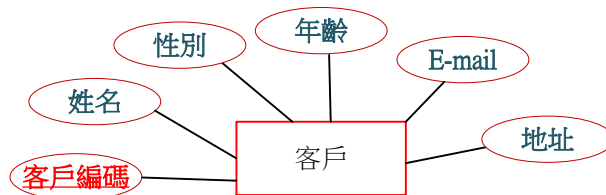


圖 11-3 客戶實體

並且將客戶實體定名為 Customers 資料表，各屬性之欄規劃如下：

<u>客戶編號</u>	姓名	性別	年齡	E-Mail	地址
<u>Cust_ID</u>	name	sex	age	Email	address
CHAR(20)	CHAR(20)	CHAR(20)	int	CHAR(20)	CHAR(50)

正規化分析：

1. 1NF：所有欄位都是最小單元(Atomic value)：符合。
2. 2NF：除了主鍵外，所有欄位都與主鍵相依：符合。
3. 3NF：除了主鍵外，所有欄位之間都沒有功能性相依：符合。

(A) 訊息實體轉換

客戶實體包含了 7 個屬性(Attribute)，且客戶編碼是唯一識別碼，並將它指定為主鍵(Primary key)。

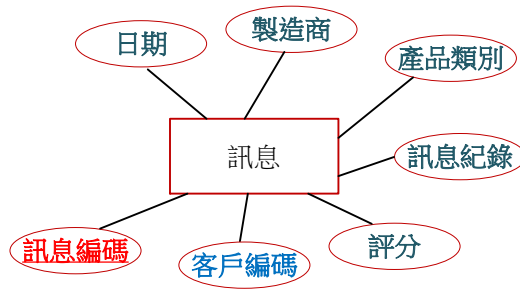
設計

圖 11-4 訊息實體

並且將訊息實體定名為 Messages 資料表，各屬性之欄規劃如下：

<u>訊息編號</u>	客戶編碼	日期	評分	製造商	產生類別	訊息紀錄
<u>Mess_ID</u>	Cust_ID	date	credit	manufacture	classify	record
int	CHAR(20)	DATE	int	CHAR(20)	CHAR(20)	CHAR(200)

正規化分析：

- 1NF：所有欄位都是最小單元(Atomic value)：符合。
- 2NF：除了主鍵外，所有欄位都與主鍵相依：符合。
- 3NF：除了主鍵外，所有欄位之間都沒有功能性相依：符合。

11-1-4 資料庫關聯圖

設計完成各個資料表之後，再依照 E-R 關聯圖繪出資料庫內所有資料表之間的關聯圖，如下：

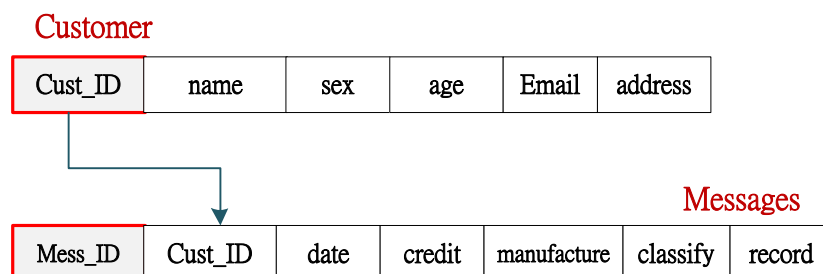


圖 11-5 資料庫關聯圖

11-1-5 實體建置

設計

我們將此資料庫定名為『**Response_db**』，並利用 AppServ – MySQL 建立，步驟如下：(備註：僅介紹用 SQL 命令產生，選單方式建立請自行參考)

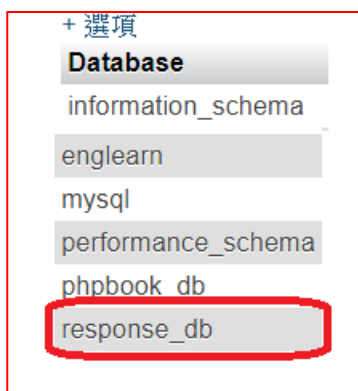
(A) 建立資料庫

吾人將此資料庫定名為：Response_db，則下列兩種方法產生，如下：

- 利用 SQL 命令為：

```
Create Database Response_db;
Show databases;
use Response_db;
```

執行結果：



(B) 建立 Customers 資料表

建立 Customers 命令如下：(選擇 response_db 資料庫)

```
Create Table Customers (
  Cust_ID CHAR(20) NOT NULL,
  name CHAR(20) NOT NULL,
  sex CHAR(20) NOT NULL,
  age int NOT NULL,
  Email CHAR(20),
  address CHAR(50),
  primary key (Cust_ID)
)
```

如果設定完成之後，可觀察 Customers 資料表的結構如下：

設計

#	名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊
1	Cust_ID 	char(20)	utf8_general_ci		否	無		
2	name	char(20)	utf8_general_ci		否	無		
3	sex	char(20)	utf8_general_ci		否	無		
4	age	int(11)			否	無		
5	Email	char(20)	utf8_general_ci		是	NULL		
6	address	char(50)	utf8_general_ci		是	NULL		

其特性說明如下：

1. Customers 資料表共有 6 個欄位，分別是：Cust_ID，name，sex，age，Email 與 address。
2. Cust_ID 欄位是主鍵，是系統自動產生的整數，也是個筆資料之間的唯一識別值(不允許有兩筆以上的紀錄，NO 欄位內容相同)。
3. Cust_ID、name、sex 與 age 等欄位不允許空值，一定必須填入內容，其他兩個欄位允許空值。

(B) 建立 Messages 資料表

建立 Messages 命令如下如下：(選擇 response_db 資料庫)

```

Create Table Messages (
  Mess_ID int auto_increment,
  Cust_ID CHAR(20) NOT NULL,
  date DATE ,
  credit int ,
  manufacture CHAR(20) ,
  classify CHAR(20),
  record CHAR(200),
  primary key (Mess_ID),
  Foreign Key (Cust_ID) References Customers(Cust_ID)
)

```

如果設定完成之後，可觀察 Messages 資料表結構，如下：

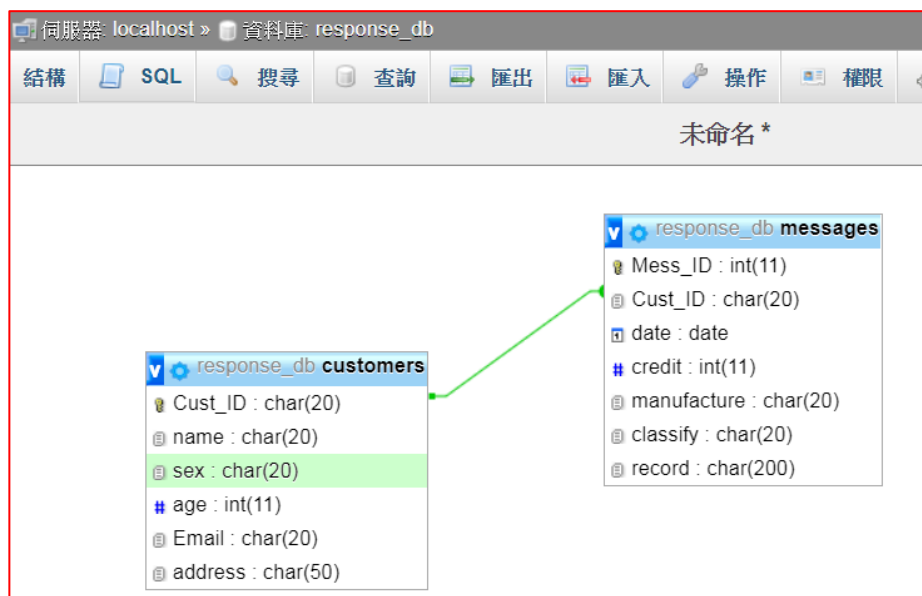
設計

#	名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊
1	Mess_ID	int(11)			否	無		AUTO_INCREMENT
2	Cust_ID	char(20)	utf8_general_ci		否	無		
3	date	date			是	NULL		
4	credit	int(11)			是	NULL		
5	manufacture	char(20)	utf8_general_ci		是	NULL		
6	classify	char(20)	utf8_general_ci		是	NULL		
7	record	char(200)	utf8_general_ci		是	NULL		

其特性說明如下：

1. Messages 資料表共有 7 個欄位，分別是：Mess_ID, Cust_ID, date, credit, Manufacture, classify 與 record。
2. Mess_ID 欄位是主鍵，是系統自動產生的整數，也是個筆資料之間的唯一識別值(不允許有兩筆以上的紀錄，NO 欄位內容相同)。
3. Cust_ID 是外來鍵，參考到 Customers 資料表的 Cust_ID 欄位。

(C) Response_db 系統關聯圖



11-1-6 建立測試資料

資料庫與資料表建立完成之後，需產生相關資料來驗證其運作是否能滿足需求。

設計

建立測試資料的步驟如下：

1. 依照資料表內個欄位的屬性，利用 Excel 相關資料，注意：Excel 欄位的順序要與資料表欄位順相同。
2. 將 Excel 建立資料後，再『另存新檔』，成『csv(逗點分隔)』格式。
3. 利用『記事本』將 csv 檔開啟後，再另存新檔，編碼選擇『UFT 8』格式。
4. 開啟資料庫，再匯入相關資料表內。

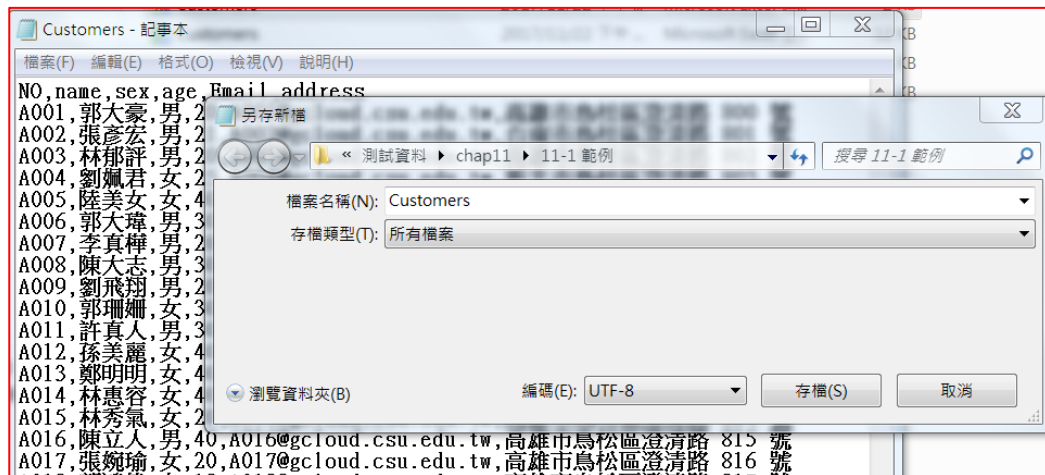
(A) 建立 Customers 資料並匯出 CSV 檔

首先開啟 Excel 並輸入相關資料(取名 Customers，第一行為欄位名稱)，另外需注意的是：『欄位順序要與資料表欄位順序相同』。如下圖所示：(可到教學網站下載)

A	B	C	D	E	F
NO	name	sex	age	Email	address
A001	郭大豪	男	20	A001@gcloud.csu.edu.tw	高雄市鳥松區澄清路 800 號
A002	張彥宏	男	21	A002@gcloud.csu.edu.tw	台南市鳥松區澄清路 801 號
A003	林郁評	男	20	A003@gcloud.csu.edu.tw	高雄市鳥松區澄清路 802 號
A004	劉佩君	女	25	A004@gcloud.csu.edu.tw	新北市鳥松區澄清路 803 號
A005	陸美女	女	40	A005@gcloud.csu.edu.tw	高雄市鳥松區澄清路 804 號
A006	郭大璋	男	30	A006@gcloud.csu.edu.tw	台中市鳥松區澄清路 805 號
A007	李真樺	男	20	A007@gcloud.csu.edu.tw	高雄市鳥松區澄清路 806 號
A008	陳大志	男	30	A008@gcloud.csu.edu.tw	高雄市鳥松區澄清路 807 號
A009	劉飛翔	男	20	A009@gcloud.csu.edu.tw	高雄市鳥松區澄清路 808 號
A010	郭珊瑚	女	30	A010@gcloud.csu.edu.tw	高雄市鳥松區澄清路 809 號
A011	許真人	男	30	A011@gcloud.csu.edu.tw	高雄市鳥松區澄清路 810 號
A012	孫美麗	女	40	A012@gcloud.csu.edu.tw	高雄市鳥松區澄清路 811 號
A013	鄭明明	女	40	A013@gcloud.csu.edu.tw	台南市鳥松區澄清路 812 號
A014	林惠容	女	40	A014@gcloud.csu.edu.tw	高雄市鳥松區澄清路 813 號
A015	林秀氣	女	20	A015@gcloud.csu.edu.tw	高雄市鳥松區澄清路 814 號
A016	陳立人	男	40	A016@gcloud.csu.edu.tw	高雄市鳥松區澄清路 815 號
A017	張婉瑜	女	20	A017@gcloud.csu.edu.tw	高雄市鳥松區澄清路 816 號
A018	郭鴻偉	女	18	A018@gcloud.csu.edu.tw	高雄市鳥松區澄清路 817 號
A019	林克強	男	40	A019@gcloud.csu.edu.tw	高雄市鳥松區澄清路 818 號

轉換成 CSV 格式如下：

設計

**(B) 建立 Messages 資料並匯出 CSV 檔**

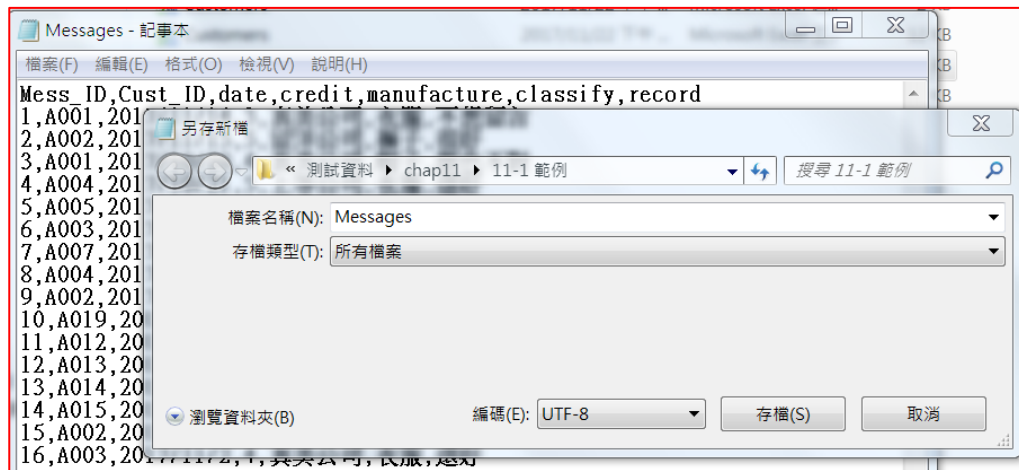
利用 Excel 並輸入相關資料(取名 Messages, 第一行為欄位名稱), 另外需注意的是 :

『欄位順序要與資料表欄位順序相同』。如下圖所示：(可到教學網站下載)

A	B	C	D	E	F	G
Mess_ID	Cust_ID	date	credit	manufacture	classify	record
1	A001	2017/11/14	3	真美公司	衣服	不想留言
2	A002	2017/11/15	5	留洋公司	褲子	很好
3	A001	2017/10/20	4	真美公司	帽子	顏色不對
4	A004	2017/10/21	5	上等公司	衣服	還好
5	A005	2017/10/22	3	真美公司	鞋子	還好
6	A003	2017/10/23	5	流尚公司	衣服	還好
7	A007	2017/10/24	2	時尚公司	外套	還好
8	A004	2017/10/25	4	真美公司	衣服	還好留言
9	A002	2017/10/26	5	會會公司	上衣	還好
10	A019	2017/10/27	5	流流公司	衣服	真好
11	A012	2017/10/28	4	有志公司	外套	還好
12	A013	2017/10/29	2	美崙公司	衣服	還好
13	A014	2017/10/30	4	真美公司	外套	我的留言
14	A015	2017/10/31	5	求美公司	鞋子	還好
15	A002	2017/11/1	5	真美公司	衣服	還好
16	A003	2017/11/2	4	真美公司	衣服	還好

轉換成 CSV 格式如下：

設計



11-1-7 匯入測試資料

匯入資料時必須考慮到資料表的外來鍵是否存在，不然會產生錯誤。由圖 11-7 Response_db 資料庫關聯看出來，匯入資料的順序是『**Customers -> Messages**』。

(A) 匯入 Customers 的 CSV 檔

由 Response_db 資料庫選擇 Customers 資料表並匯入 Customers CSV 檔，結果如下：

Cust_ID	name	sex	age	Email	address
A001	郭大豪	男	20	A001@gcloud.csu.edu.	高雄市鳥松區澄清路 800 號
A002	張彥宏	男	21	A002@gcloud.csu.edu.	台南市鳥松區澄清路 801 號
A003	林郁評	男	20	A003@gcloud.csu.edu.	高雄市鳥松區澄清路 802 號
A004	劉嫻君	女	25	A004@gcloud.csu.edu.	新北市鳥松區澄清路 803 號
A005	陸美女	女	40	A005@gcloud.csu.edu.	高雄市鳥松區澄清路 804 號
A006	郭大璋	男	30	A006@gcloud.csu.edu.	台中市鳥松區澄清路 805 號

(B) 匯入 Messages 的 CSV 檔

由 Response_db 資料庫選擇 Messages 資料表並匯入 Messages CSV 檔，結果如下：

設計

Mess_ID	Cust_ID	date	credit	manufacture	classify	record
1	A001	2017-11-14	3	真美公司	衣服	不想留言
2	A002	2017-11-15	5	留洋公司	褲子	很好
3	A001	2017-10-20	4	真美公司	帽子	顏色不對
4	A004	2017-10-21	5	上等公司	衣服	還好
5	A005	2017-10-22	3	真美公司	鞋子	還好
6	A003	2017-10-23	5	流尚公司	衣服	還好
7	A007	2017-10-24	2	時尚公司	外套	還好
8	A004	2017-10-25	4	真美公司	衣服	還好留言
9	A002	2017-10-26	5	會會公司	上衣	還好

11-1-8 測試資料庫功能

資料庫建構完成後，我們依照幾個系統需求，來驗證其功能是否滿足，如下：

(A) 驗證 1：查詢客戶回應訊息

查詢郭大豪先生的回應訊息如何，SQL 命令如下：

```
select *
from messages
where cust_ID = (SELECT Cust_ID
                 from customers
                 where name = "郭大豪");
```

執行結果：

Mess_ID	Cust_ID	date	credit	manufacture	classify	record
1	A001	2017-11-14	3	真美公司	衣服	不想留言
3	A001	2017-10-20	4	真美公司	帽子	顏色不對

(B) 驗證 2：查詢某家製造商的回應訊息

查詢有關真美公司的回應訊息如何，SQL 命令如下：

```
select *
from messages
where manufacture = "真美公司";
```

設計

執行結果：

Mess_ID	Cust_ID	date	credit	manufacture	classify	record
1	A001	2017-11-14	3	真美公司	衣服	不想留言
3	A001	2017-10-20	4	真美公司	帽子	顏色不對
5	A005	2017-10-22	3	真美公司	鞋子	還好
8	A004	2017-10-25	4	真美公司	衣服	還好留言
13	A014	2017-10-30	4	真美公司	外套	我的留言

(C) 驗證 3：查詢滿意度的回應訊息

查詢滿意度低於 3 的相關回應訊息如何，SQL 命令如下：

```
select *
from messages
where credit < 3;
```

執行結果：

Mess_ID	Cust_ID	date	credit	manufacture	classify	record
7	A007	2017-10-24	2	時尚公司	外套	還好
12	A013	2017-10-29	2	美崙公司	衣服	還好

(D) 驗證 4：依客戶年紀查詢

查詢年紀在 20 ~ 30 歲之間客戶的相關回應訊息如何，SQL 命令如下：

```
Select *
From messages
Where Cust_ID in (Select Cust_ID
                  From customers
                  Where age < 30
                  AND age > 20);
```

執行結果：

Mess_ID	Cust_ID	date	credit	manufacture	classify	record
2	A002	2017-11-15	5	留洋公司	褲子	很好
9	A002	2017-10-26	5	會會公司	上衣	還好
15	A002	2017-11-01	5	真美公司	衣服	還好
4	A004	2017-10-21	5	上家公司	衣服	還好
8	A004	2017-10-25	4	真美公司	衣服	還好留言

設計

(E) 驗證 5：依產品類別查詢

查詢客戶對本公司銷售『衣服』的滿意度低於 4 的相關回應訊息，SQL 命令如下：

```
select *  
from messages  
where classify = "衣服"  
and credit < 4;
```

執行結果：

Mess_ID	Cust_ID	date	credit	manufacture	classify	record
1	A001	2017-11-14	3	真美公司	衣服	不想留言
12	A013	2017-10-29	2	美崙公司	衣服	還好

11-2 範例研討 – 圖書館管理系統

11-2-1 需求分析與資料收集

起初萬聖宮圖書室建立時，收藏圖書還不是很多，村民借書/還書都還不是很困難。但經過幾年之後，圖書室功能漸漸顯著，村民閱讀風氣業逐漸打開，除了各地捐輸之外，萬聖宮也採購許多書籍以供村民閱讀。如此一來，圖書室亦發展成規模不小的圖書館。萬聖宮總幹事有感於此，原來圖書管理系統已不符所需，期望建立一套較完整的圖書館管理系統，期望功能如下：

- 所有圖書皆登錄於該系統內。
- 可瀏覽圖書館內所有書籍。
- 可經由作者姓名、出版商或書名，查詢到那些書籍。
- 圖書可以任意增加或刪除。
- 每一本書都有固定位置，借書或還書都可以由固定區域存取。
- 可登錄村民借用那些書籍。
- 可催收逾期借用書籍。
- 可由書籍名稱查詢出哪位村民借用，以及借書日期。

設計

另外，總幹事也希望系統能登錄下列相關資料：

- 書籍資料：書名、作者、出版社、放置位置、書籍分類、書籍來源、購買金額。
- 借書者：姓名、性別、電話、地址、電話。

圖書館也準備了借書卡，以供村民申請借書、還書之依據。

11-2-2 資料庫邏輯設計

此系統有比較複雜一點，我們由外往內拆解方法，來分析此系統的邏輯架構。

(A) 第一層分析：系統需求

吾人可以發現，此管理系統提供圖書登錄與村民閱讀的功能，如下圖所示。

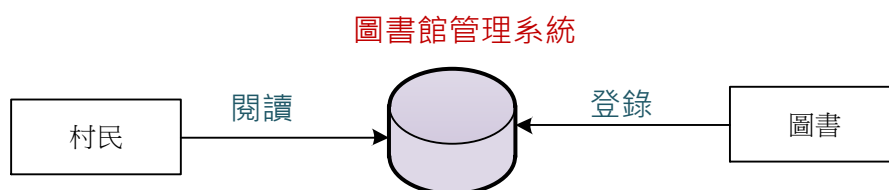


圖 11-6 圖書館管理系統概況

(B) 第二層分析：圖書分類

首先，我們先拆解圖書出版狀況，每一本書大多由某一家出版社發行，分析如下：

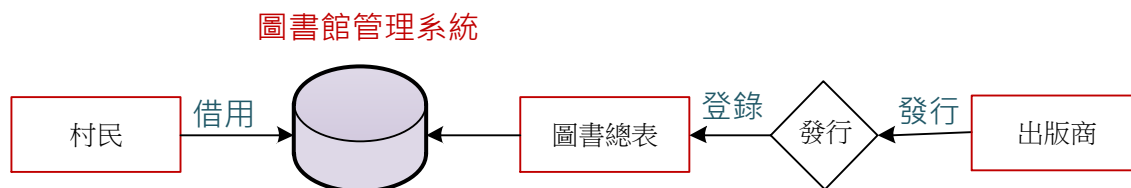


圖 11-7 第二層分析

(C) 第三層分析：村民借用圖書閱讀

村民借用圖書閱讀之關聯。我們假設一位村民可以借用多本圖書，則村民與圖書總表之間的關係是一對多的關係。另外，一家出版商也可能發行多本圖書，之間也是一對多的關係。每位村民也可以多次借書，村民與借書紀錄之間也是一對多的關係。如下

設計

分析：

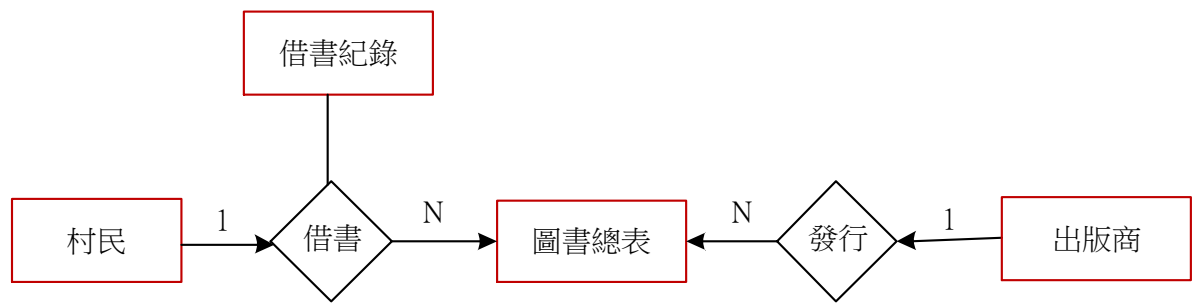
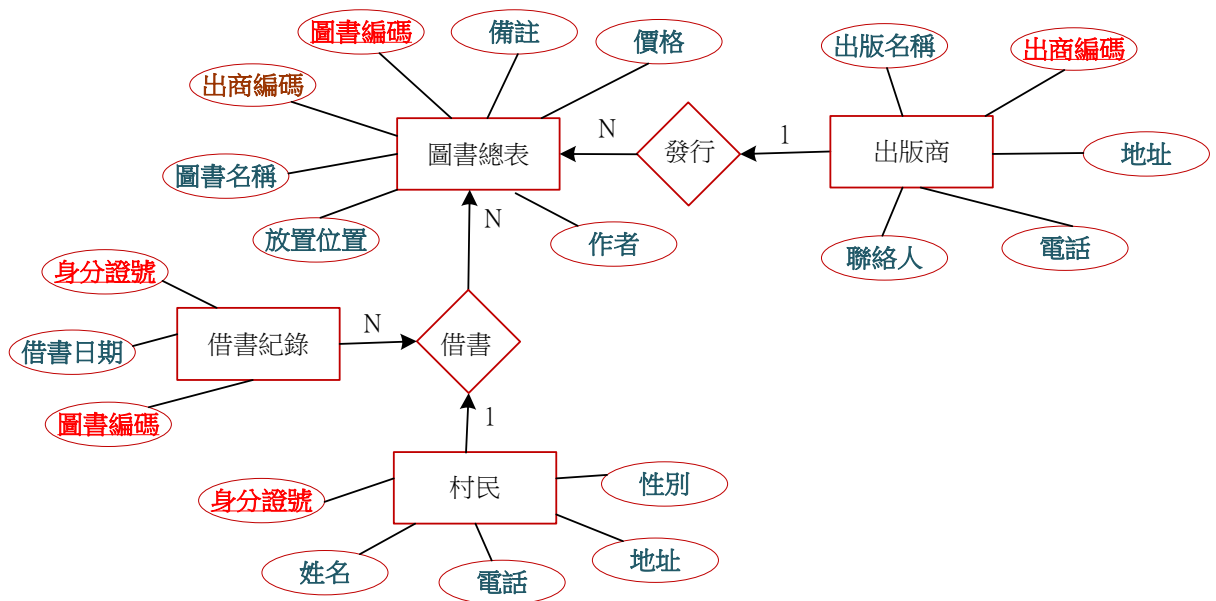


圖 11-8 第三層分析

(D) 填入各實體的屬性

接著我們依照系統所需的資料填入各個實體上，如下圖：

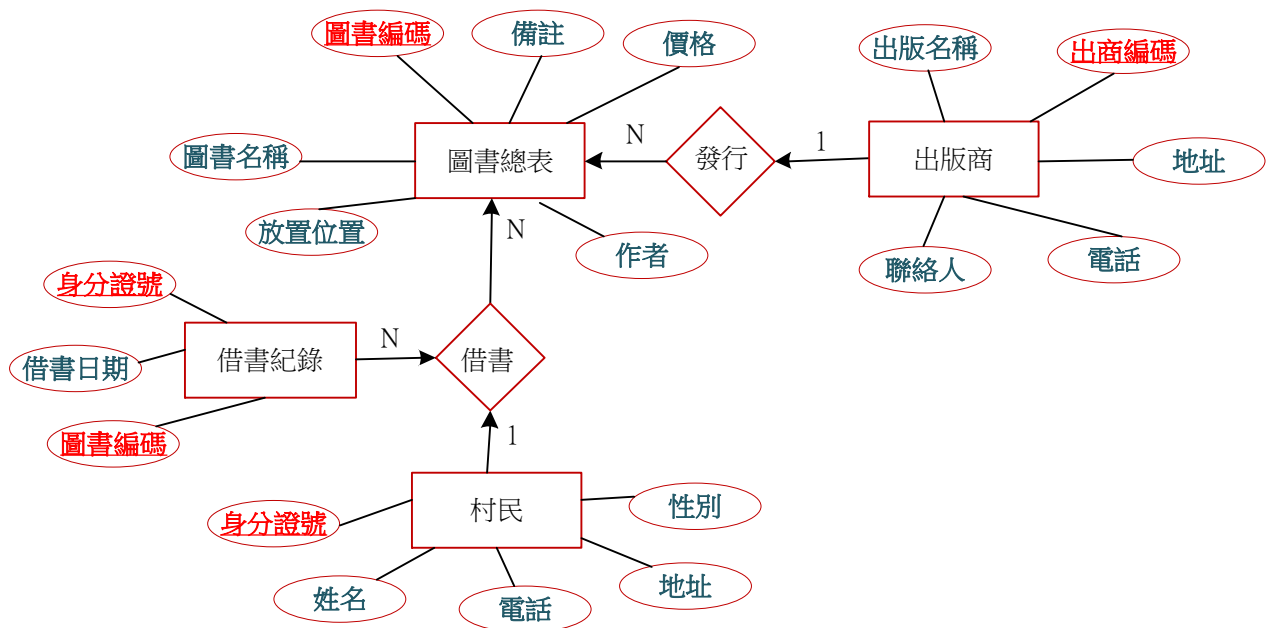
(E) 選定各實體的主鍵

對各實體的主鍵選定步驟如下：

- (1) 圖書總表：將所有屬性填入後，找不到唯一識別值，增加『圖書編碼』屬性，並指定為主鍵。
- (2) 出版商：增列『出商編碼』屬性，並指定為主鍵。
- (3) 村民：增列『身分證字號』屬性，並指定為主鍵。

設計

- (4) 借書紀錄：因每位村民可以同時借用多本書，增列 {身分證號、圖書編碼} 兩屬性，並指定為主鍵。

**(F) 建立外來鍵連結 – 完整系統 E-R 關聯圖**

- (1) 在圖書總表增加『出版編碼』為外來鍵，並參考到出版商的『出商編碼』，建立兩實體之間關係。
- (2) 且將借書紀錄的『身分證號』設定為外來鍵，並參考到村民實體的『身分證號』，建立借書紀錄與村民實體之間的關係。
- (3) 且將借書紀錄的『圖書編碼』設定為外來鍵，並參考到圖書總表實體的『圖書編碼』，建立借書紀錄與圖書總表之間的關係。

完成後結果，如下圖所示：

設計

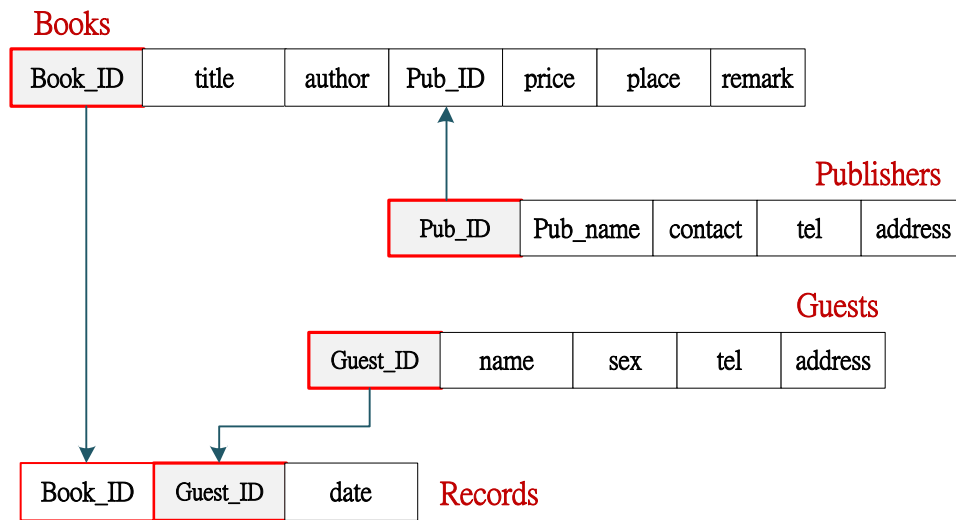


圖 11-9 E-R 關聯圖

11-2-3 E-R 圖轉換資料表

(A) 圖書總表關聯

下圖為圖書總表的關聯(Relation)，它包含了 6 個屬性(Attribute)，我們找出圖書編碼可以當作每一筆紀錄的唯一識別碼，並將它指定為主鍵(Primary key)。

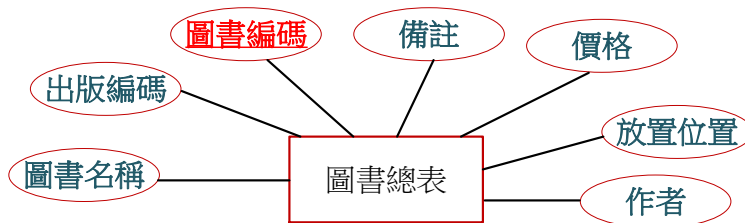


圖 11-10 圖書總表關聯

並且將此關聯定名為 Books 資料表，各屬性之欄規劃如下：

<u>圖書編號</u>	圖書名稱	作者	出版商編碼	備註	價格	放置位置
<u>Book_ID</u>	title	author	Pub_ID	remark	price	place
int	CHAR(50)	CHAR(20)	CHAR(20)	CHAR(50)	int	CHAR(20)

其中，備註欄位記錄書籍來源(採購、捐贈)或其他訊息。價格該本書購買金額或捐贈人認為價值多少錢。

正規化分析：

設計

1. **1NF**：所有欄位都是最小單元(Atomic value)：符合。
2. **2NF**：除了主鍵外，所有欄位都與主鍵相依：符合。
3. **3NF**：除了主鍵外，所有欄位之間都沒有功能性相依：符合。

(B) 出版商關聯

出版商關聯(Relation)包含了 5 個屬性(Attribute)，我們找出版商編碼可以當作每一筆紀錄的唯一識別碼，並將它指定為主鍵(Primary key)。

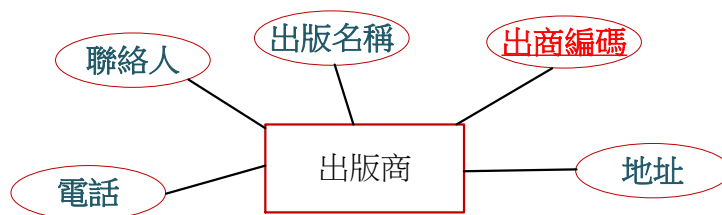


圖 11-11 出版商關聯

並且將此關聯定名為 Publishers 資料表，各屬性之欄規劃如下：

<u>出版商編碼</u>	出版商名稱	聯絡人	電話	地址
<u>Pub_ID</u>	pub_name	contact	tel	address
int	CHAR(20)	CHAR(20)	CHAR(20)	CHAR(50)

其中，書籍來源有採購與捐贈兩種可能。價格該本書購買金額或捐贈人認為價值多少錢。

正規化分析：

1. **1NF**：所有欄位都是最小單元(Atomic value)：符合。
2. **2NF**：除了主鍵外，所有欄位都與主鍵相依：符合。
3. **3NF**：除了主鍵外，所有欄位之間都沒有功能性相依：符合。

(C) 村民關聯

村民關聯(Relation)包含了 5 個屬性(Attribute)，很容易地可以看出，選用身分證字

設計

號當為主鍵(Primary key)。

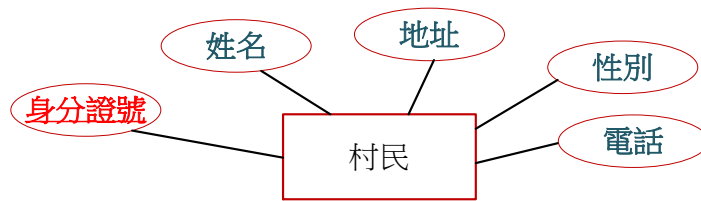


圖 11-12 村民關聯

並且將此關聯定名為 Guests 資料表，各屬性之欄規劃如下：

<u>身分證字號</u>	姓名	性別	電話	地址
<u>Guest_ID</u>	name	sex	tel	address
CHAR(20)	CHAR(20)	CHAR(20)	CHAR(20)	CHAR(50)

其中，書籍來源有採購與捐贈兩種可能。價格該本書購買金額或捐贈人認為價值多少錢。

正規化分析：

1. **1NF**：所有欄位都是最小單元(Atomic value)：符合。
2. **2NF**：除了主鍵外，所有欄位都與主鍵相依：符合。
3. **3NF**：除了主鍵外，所有欄位之間都沒有功能性相依：符合。

(C) 借書紀錄關聯

借書紀錄關聯(Relation)包含了 3 個屬性(Attribute)。系統允許每位村民同時可以借用多本書籍，因此不能只用身分證字號當主鍵，必須 {身分證字號, 書籍編碼} 兩欄位當為主鍵(Primary key)。

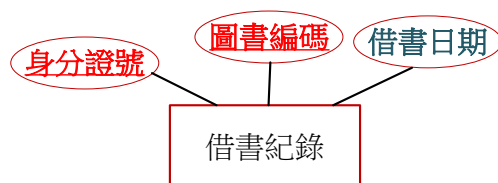


圖 11-13 借書紀錄關聯

設計

並且將此關聯定名為 Records 資料表，各屬性之欄規劃如下：

<u>身分證字號</u>	<u>圖書編號</u>	借書日期
<u>Guest_ID</u>	Book_ID	date
CHAR(20)	CHAR(20)	DATE

正規化分析：

1. **1NF**：所有欄位都是最小單元(Atomic value)：符合。
2. **2NF**：除了主鍵外，所有欄位都與主鍵相依：符合。
3. **3NF**：除了主鍵外，所有欄位之間都沒有功能性相依：符合。

11-2-4 資料庫關聯圖

設計完成各個資料表之後，再依照 E-R 關聯圖繪出資料庫內所有資料表之間的關聯圖，如下：

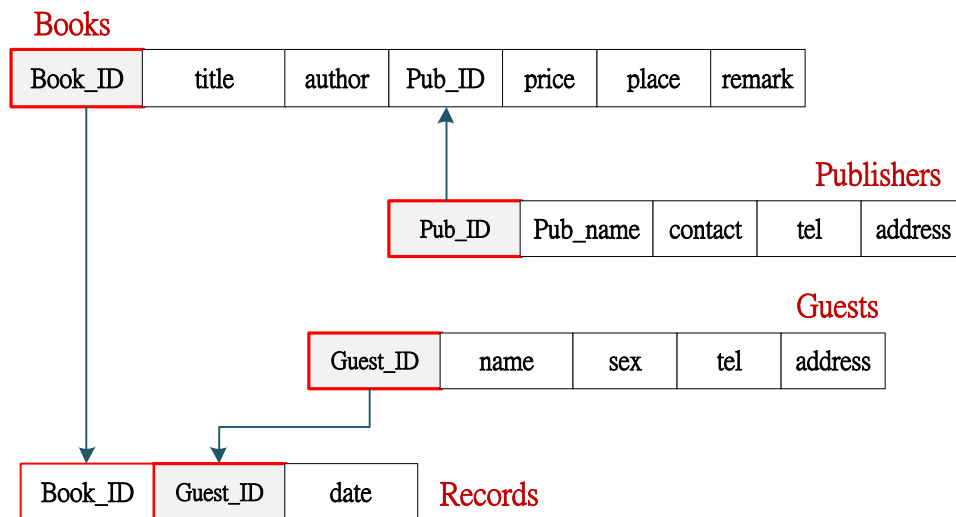


圖 11-14 資料庫關聯圖

11-2-5 實體建置

我們將此資料庫定名為『**temple_library**』，並利用 AppServ – MySQL 建立，步驟如下：(備註：僅介紹用 SQL 命令產生，選單方式建立請自行參考)

設計

(A) 建立資料庫

吾人將此資料庫定名為：temp_library，則下列兩種方法產生，如下：

- 利用 SQL 命令為：

```
Create Database temp_library;
```

- 利用選單方式：

(B) 建立 Books 資料表

建立 Books 命令如下如下：(選擇 temp_library 資料庫)

```
Create Table Books (  
    Book_ID int auto_increment,  
    title CHAR(50) NOT NULL,  
    author CHAR(20) NOT NULL,  
    Pub_ID int NOT NULL,  
    price int  
    place CHAR(20),  
    remark CHAR(50),  
    primary key (Book_ID)  
)
```

如果設定完成之後，可觀察 Books 資料表的結構如下：

設計

名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊
Book_ID 	int(11)			否	無		AUTO_INCREMENT
title	char(50)	utf8_unicode_ci		否	無		
author	char(20)	utf8_unicode_ci		否	無		
Pub_ID	int(11)			否	無		
price	int(11)			是	NULL		
place	char(20)	utf8_unicode_ci		是	NULL		
remark	char(50)	utf8_unicode_ci		是	NULL		

其特性說明如下：

1. Books 資料表共有 7 個欄位，分別是：Book_ID、author、Pub_ID、title、place、price 與 remark。
2. Book_ID 欄位是主鍵，是系統自動產生的整數，也是個筆資料之間的唯一識別值(不允許有兩筆以上的紀錄，NO 欄位內容相同)。
3. Book_ID、author、Pub_ID 與 title 等欄位不允許空值，一定必須填入內容，其他三個欄位允許空值。

(B) 建立 Publishers 資料表

建立 publishers 命令如下如下：(選擇 **temple_library** 資料庫)

```
Create Table publishers (
  Pub_ID int auto_increment,
  Pub_name CHAR(50) NOT NULL,
  contact CHAR(20),
  tel CHAR(20),
  address CHAR(50),
  primary key (Pub_ID)
)
```

如果設定完成之後，可觀察 Publishers 資料表結構，如下：

設計

#	名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊
1	Pub_ID 	int(11)			否	無		AUTO_INCREMENT
2	Pub_name	char(50)	utf8_unicode_ci		否	無		
3	contact	char(20)	utf8_unicode_ci		是	NULL		
4	tel	char(20)	utf8_unicode_ci		是	NULL		
5	address	char(50)	utf8_unicode_ci		是	NULL		

其特性說明如下：

1. publishers 資料表共有 5 個欄位，分別是：Pub_ID，Pub_name，contact，tel 與 address。
2. Pub_ID 欄位是主鍵，是系統自動產生的整數，也是個筆資料之間的唯一識別值 (不允許有兩筆以上的紀錄，NO 欄位內容相同)。
3. Pub_ID 與 Pub_name 欄位不允許空值，一定必須填入內容，其他三個欄位允許空值。

(C) 建立 Guests 資料表

建立 Guests 命令如下如下：(選擇 **temple_library** 資料庫)

```
Create Table Guests (
  Guest_ID CHAR(20) NOT NULL,
  name CHAR(20) NOT NULL,
  sex CHAR(20),
  tel CHAR(20),
  address CHAR(50),
  primary key (Guest_ID)
)
```

如果設定完成之後，可觀察 Guests 資料表結構，如下：

設計

#	名稱	型態	編碼與排序	屬性	空值	預設值	備註
1	Guest_ID 	char(20)	utf8_unicode_ci		否	無	
2	name	char(20)	utf8_unicode_ci		否	無	
3	sex	char(20)	utf8_unicode_ci		是	NULL	
4	tel	char(20)	utf8_unicode_ci		是	NULL	
5	address	char(50)	utf8_unicode_ci		是	NULL	

其特性說明如下：

1. Guests 資料表共有 5 個欄位，分別是：Guest_ID，name，sex，tel 與 address。
2. Guest_ID 欄位是主鍵，是系統自動產生的整數，也是個筆資料之間的唯一識別值(不允許有兩筆以上的紀錄，NO 欄位內容相同)。
3. Guest_ID 與 name 欄位不允許空值，一定必須填入內容，其他三個欄位允許空值。

(D) 建立 Records 資料表

建立 Records 命令如下如下：(選擇 **temple_library** 資料庫)

```
Create Table Records (
  Guest_ID CHAR(20) NOT NULL,
  Book_ID int NOT NULL,
  date DATE,
  primary key (Guest_ID, Book_ID)
)
```

如果設定完成之後，可觀察 Records 資料表結構如下：

#	名稱	型態	編碼與排序	屬性	空值	預設值	備註	額外資訊
1	Guest_ID 	char(20)	utf8_unicode_ci		否	無		
2	Book_ID 	int(11)			否	無		
3	date	date			是	NULL		

其特性說明如下：

1. Records 資料表共有 3 個欄位，分別是：Guest_ID，Book_ID 與 date。

設計

2. Guest_ID 與 Book_ID 兩欄位組合成主鍵。

(D) 觀察 Temple_Library 資料庫結構

建立完成四個資料表之後，可以觀察 Temple_library 資料庫的結構如何，如下：

**(E) 建立資料表之間關連**

我們依照圖 11-8 資料庫關聯圖內，可以觀察到各個資料表之間的關聯，如下：

1. Books 資料表的 Pub_ID 是外來鍵(Foreign Key)，是參考到 Publishers 資料表的 Pub_ID 欄位。
2. Records 資料表的 Book_ID 是外來鍵，是參考到 Books 資料表的 Book_ID 欄位。
3. Records 資料表的 Guest_ID 是外來鍵，是參考到 Guests 資料表的 Guest_ID 欄位。

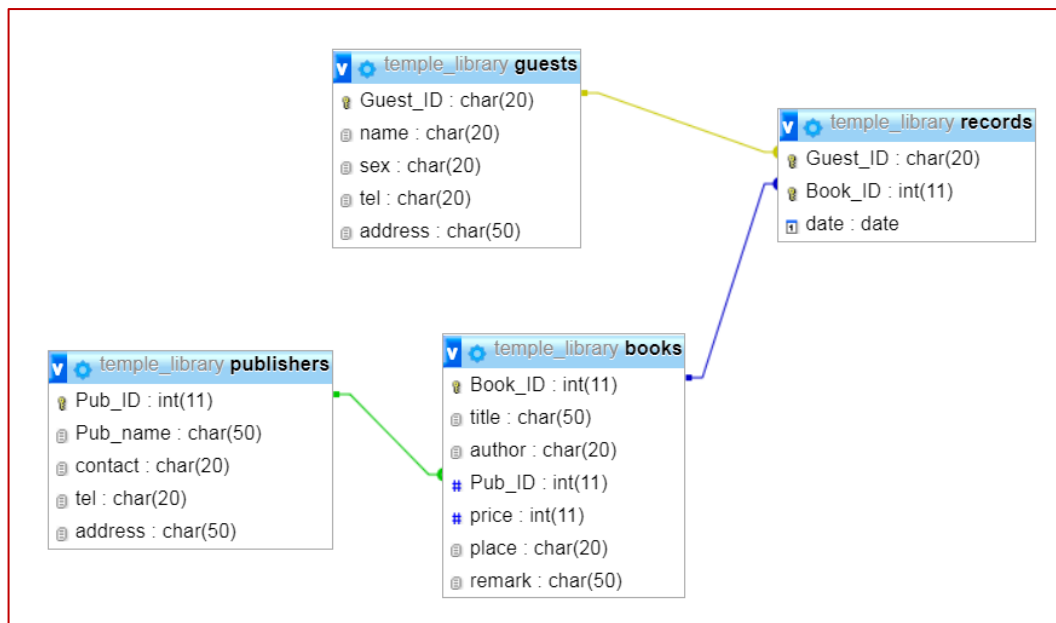
建立外來鍵命令如下：

```
Alter Table Books
    ADD Foreign key (Pub_ID) References publishers(Pub_ID);
Alter Table Records
    ADD Foreign key (Book_ID) References Books(Book_ID);
Alter Table Records
    ADD Foreign key (Guest_ID) References Guests(Guest_ID);
```

執行成功後，點選 AppServ 的『設計』選單，可以觀察到 Temple_Library 資料內各個

設計

資料表之間的關聯，如下：

**11-2-6 建立測試資料**

資料庫與資料表建立完成之後，需產生相關資料來驗證其運作是否能滿足需求。

建立測試資料的步驟如下：

1. 依照資料表內個欄位的屬性，利用 Excel 相關資料，注意：Excel 欄位的順序要與資料表欄位順相同。
2. 將 Excel 建立資料後，再『另存新檔』，成『csv(逗點分隔)』格式。
3. 利用『記事本』將 csv 檔開啟後，再另存新檔，編碼選擇『UFT 8』格式。
4. 開啟資料庫，再匯入相關資料表內。

(A) 建立 Books 資料並匯出 CSV 檔

首先開啟 Excel 並輸入相關資料(取名 Books，第一行為欄位名稱)，如下圖所示：
(可到教學網站下載)

設計

A	B	C	D	E	F	G	
Book_ID	title	author	Pub_ID	price	place	remark	
1	三國演義	劉貫中		1	500	A-1-1	張友德先生贈
2	紅樓夢	曹雪芹		2	500	A-1-2	張友德先生贈
3	論語	論語		1	100	B-1-2	張友德先生贈
4	墨子	墨子		1	500	B-1-2	張友德先生贈
5	孫子兵法	孫子		3	500	B-1-3	張友德先生贈
6	孟子	孟子		3	500	B-1-4	張友德先生贈
7	逍遙遊	莊子		5	500	B-1-5	張友德先生贈
8	呂氏春秋	呂不書		4	300	B-1-6	張友德先生贈
9	電腦網路與連結技術	粘添壽		4	600	C-1-2	採購
10	Unix/Linux 作業系統	粘添壽		6	500	C-1-3	採購
11	資訊與網路安全技術	粘添壽		3	500	C-1-4	採購
12	Java 程式設計(一)	粘添壽		2	500	C-1-5	採購
13	網際網路原理與運用	粘添壽		1	500	C-1-6	粘添壽先生贈
14	Java 程式設計(二)	粘添壽		5	200	C-1-7	粘添壽先生贈
15	資料庫系統概論	粘添壽		2	400	C-1-8	粘添壽先生贈
16	資料庫程式設計	粘添壽		2	200	C-1-9	粘添壽先生贈
17	伺服器系統管理 - CentOS	粘添壽		2	500	C-1-10	粘添壽先生贈
18	電子商務安全	粘添壽		2	500	C-1-11	粘添壽先生贈

轉換成 CSV 格式如下：

```

Books - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Book_ID,title,author,Pub_ID,price,place,remark
1,三國演義,劉貫中,1,500,A-1-1,張友德先生贈
2,紅樓夢,曹雪芹,2,500,A-1-2,張友德先生贈
3,論語,論語,1,100,B-1-2,張友德先生贈
4,墨子,墨子,1,500,B-1-2,張友德先生贈
5,孫子兵法,孫子,3,500,B-1-3,張友德先生贈
6,孟子,孟子,3,500,B-1-4,張友德先生贈
7,逍遙遊,莊子,5,500,B-1-5,張友德先生贈
8,呂氏春秋,呂不書,4,300,B-1-6,張友德先生贈
9,電腦網路與連結技術,粘添壽,4,600,C-1-2,採購
10,Unix/Linux 作業系統,粘添壽,6,500,C-1-3,採購
11,資訊與網路安全技術,粘添壽,3,500,C-1-4,採購
12,Java 程式設計(一),粘添壽,2,500,C-1-5,採購
13,網際網路原理與運用,粘添壽,1,500,C-1-6,粘添壽先生贈
14,Java 程式設計(二),粘添壽,5,200,C-1-7,粘添壽先生贈
15,資料庫系統概論,粘添壽,2,400,C-1-8,粘添壽先生贈
16,資料庫程式設計,粘添壽,2,200,C-1-9,粘添壽先生贈
17,伺服器系統管理 - CentOS,粘添壽,2,500,C-1-10,粘添壽先生贈
18,電子商務安全,粘添壽,2,500,C-1-11,粘添壽先生贈

```

(B) 建立 Publishers 資料並匯出 CSV 檔

利用 Excel 建立 Publishers 檔案內容如下：

設計

A	B	C	D	E
Pub_ID	Pub_name	contact	tel	address
1	大台北圖書公司	劉大德	02-2345671	台北市中山區中山路 200 號
2	真美出版公司	張萬千	02-2345672	台北市大同區德民路 201 號
3	翻轉工作室	粘添壽	07-7310606	高雄市鳥松區澄清路 806 號
4	旗標圖書出版公司	林萬億	07-2345674	高雄市岡山區青山路 807 號
5	學冠行銷公司	劉得住	02-2345675	新北市萬芳區中正路 204 號
6	千葉出版公司	李銘順	02-2345676	台北市文山區清翠路 205 號

轉換成 CSV 檔內容如下：

Publishers - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

```
Pub_ID, Pub_name, contact, tel, address
1, 大台北圖書公司, 劉大德, 02-2345671, 台北市中山區中山路 200 號
2, 真美出版公司, 張萬千, 02-2345672, 台北市大同區德民路路 201 號
3, 翻轉工作室, 粘添壽, 07-7310606, 高雄市鳥松區澄清路 806 號
4, 旗標圖書出版公司, 林萬億, 07-2345674, 高雄市岡山區青山路 807 號
5, 學冠行銷公司, 劉得住, 02-2345675, 新北市萬芳區中正路 204 號
6, 千葉出版公司, 李銘順, 02-2345676, 台北市文山區清翠路 205 號
```

(C) 建立 Guests 資料並匯出 CSV 檔

利用 Excel 建立 Guests 檔案內容如下：

	A	B	C	D	E
1	Guest_ID	name	sex	tel	address
2	A001	郭大豪	男	07-7310634	高雄市鳥松區澄清路 800 號
3	A002	張彥宏	男	07-7310635	台南市鳥松區澄清路 801 號
4	A003	林郁評	男	07-7310636	高雄市鳥松區澄清路 802 號
5	A004	劉嫻君	女	07-7310637	新北市鳥松區澄清路 803 號
6	A005	陸美女	女	07-7310638	高雄市鳥松區澄清路 804 號
7	A006	郭大璋	男	07-7310639	台中市鳥松區澄清路 805 號
8	A007	李真樺	男	07-7310640	高雄市鳥松區澄清路 806 號
9	A008	陳大志	男	07-7310641	高雄市鳥松區澄清路 807 號
10	A009	劉飛翔	男	07-7310642	高雄市鳥松區澄清路 808 號
11	A010	郭珊瑚	女	07-7310643	高雄市鳥松區澄清路 809 號
12	A011	許真人	男	07-7310644	高雄市鳥松區澄清路 810 號
13	A012	孫美麗	女	07-7310645	高雄市鳥松區澄清路 811 號
14	A013	鄭明明	女	07-7310646	台南市鳥松區澄清路 812 號
15	A014	林惠容	女	07-7310647	高雄市鳥松區澄清路 813 號
16	A015	林秀氣	女	07-7310648	高雄市鳥松區澄清路 814 號
17	A016	陳立人	男	07-7310649	高雄市鳥松區澄清路 815 號
18	A017	張婉瑜	女	07-7310650	高雄市鳥松區澄清路 816 號
19	A018	郭鴻偉	女	07-7310651	高雄市鳥松區澄清路 817 號
20	A019	林克強	男	07-7310652	高雄市鳥松區澄清路 818 號

轉換成 CSV 檔內容如下：

設計

Guest_ID	name	sex	tel	address	800	號
A001	郭大豪	男	07-7310634	高雄市鳥松區澄清路	800	號
A002	張彥宏	男	07-7310635	台南市鳥松區澄清路	801	號
A003	林郁評	男	07-7310636	高雄市鳥松區澄清路	802	號
A004	劉佩君	女	07-7310637	新北市鳥松區澄清路	803	號
A005	陸美女	女	07-7310638	高雄市鳥松區澄清路	804	號
A006	郭大瑋	男	07-7310639	台中市鳥松區澄清路	805	號
A007	李真樺	男	07-7310640	高雄市鳥松區澄清路	806	號
A008	陳大志	男	07-7310641	高雄市鳥松區澄清路	807	號
A009	劉飛翔	男	07-7310642	高雄市鳥松區澄清路	808	號
A010	郭珊瑚	女	07-7310643	高雄市鳥松區澄清路	809	號
A011	許真人	男	07-7310644	高雄市鳥松區澄清路	810	號
A012	孫美麗	女	07-7310645	高雄市鳥松區澄清路	811	號
A013	鄭明明	女	07-7310646	台南市鳥松區澄清路	812	號
A014	林惠容	女	07-7310647	高雄市鳥松區澄清路	813	號
A015	林秀氣	女	07-7310648	高雄市鳥松區澄清路	814	號
A016	陳立人	男	07-7310649	高雄市鳥松區澄清路	815	號
A017	張婉瑜	女	07-7310650	高雄市鳥松區澄清路	816	號
A018	郭鴻偉	女	07-7310651	高雄市鳥松區澄清路	817	號
A019	林克強	男	07-7310652	高雄市鳥松區澄清路	818	號

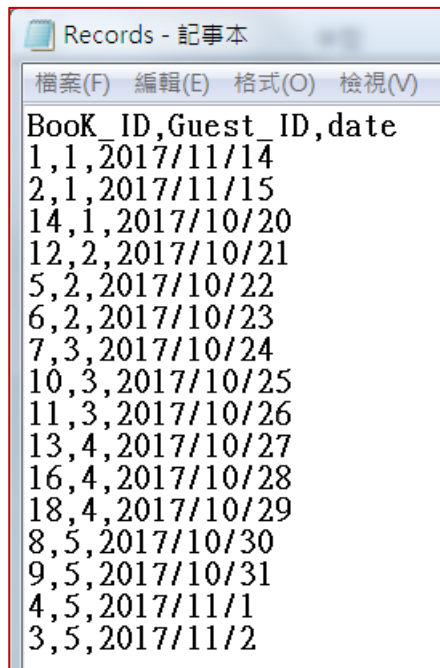
(D) 建立 Records 資料並匯出 CSV 檔

利用 Excel 建立 Records 檔案如下：

A	B	C
Book_ID	Guest_ID	date
1	1	2017/11/14
2	1	2017/11/15
14	1	2017/10/20
12	2	2017/10/21
5	2	2017/10/22
6	2	2017/10/23
7	3	2017/10/24
10	3	2017/10/25
11	3	2017/10/26
13	4	2017/10/27
16	4	2017/10/28
18	4	2017/10/29
8	5	2017/10/30
9	5	2017/10/31
4	5	2017/11/1
3	5	2017/11/2

轉換成 CSV 檔內容如下：

設計



11-2-7 匯入測試資料

匯入資料時必須考慮到資料表的外來鍵是否存在，不然會產生錯誤。由圖 11-8 Temple_library 資料庫關聯看出來，匯入資料的順序是『**Publishers -> Books -> Guests -> Records**』。

(A) 匯入 Publishers 的 CSV 檔

由 Temple_Library 資料庫選擇 Publishers 資料表並匯入 Publishers CSV 檔，結果如下：

Pub_ID	Pub_name	contact	tel	address
1	大台北圖書公司	劉大德	02-2345671	台北市中山區中山路 200 號
2	真美出版公司	張萬千	02-2345672	台北市大同區德民路路 201 號
3	翻轉工作室	粘添壽	07-7310606	高雄市鳥松區澄清路 806 號
4	旗標圖書出版公司	林萬億	07-2345674	高雄市岡山區青山路 807 號
5	學冠行銷公司	劉得住	02-2345675	新北市萬芳區中正路 204 號
6	千葉出版公司	李銘順	02-2345676	台北市文山區清翠路 205 號

(B) 匯入 Books 的 CSV 檔

由 Temple_Library 資料庫選擇 Books 資料表並匯入 Books CSV 檔，結果如下：

設計

Book_ID	title	author	Pub_ID	price	place	remark
1	三國演義	劉貫中	1	500	A-1-1	張友德先生贈
2	紅樓夢	曹雪芹	2	500	A-1-2	張友德先生贈
3	論語	論語	1	100	B-1-2	張友德先生贈
4	墨子	墨子	1	500	B-1-2	張友德先生贈
5	孫子兵法	孫子	3	500	B-1-3	張友德先生贈
6	孟子	孟子	3	500	B-1-4	張友德先生贈
7	逍遙遊	莊子	5	500	B-1-5	張友德先生贈
8	呂氏春秋	呂不書	4	300	B-1-6	張友德先生贈
9	電腦編碼與連結技術	粘添壽	4	600	C-1-2	採購

(C) 匯入 Guests 的 CSV 檔

由 Temple_Library 資料庫選擇 Guests 資料表並匯入 Guests CSV 檔，結果如下：

Guest_ID	name	sex	tel	address
A001	郭大豪	男	07-7310634	高雄市鳥松區澄清路 800 號
A002	張彥宏	男	07-7310635	台南市鳥松區澄清路 801 號
A003	林郁評	男	07-7310636	高雄市鳥松區澄清路 802 號
A004	劉佩君	女	07-7310637	新北市鳥松區澄清路 803 號
A005	陸美女	女	07-7310638	高雄市鳥松區澄清路 804 號
A006	郭大璋	男	07-7310639	台中市鳥松區澄清路 805 號
A007	李真樺	男	07-7310640	高雄市鳥松區澄清路 806 號
A008	陳大志	男	07-7310641	高雄市鳥松區澄清路 807 號
A009	劉飛翔	男	07-7310642	高雄市鳥松區澄清路 808 號

(D) 匯入 Records 的 CSV 檔

由 Temple_Library 資料庫選擇 Records 資料表並匯入 Records CSV 檔，結果如下：

設計

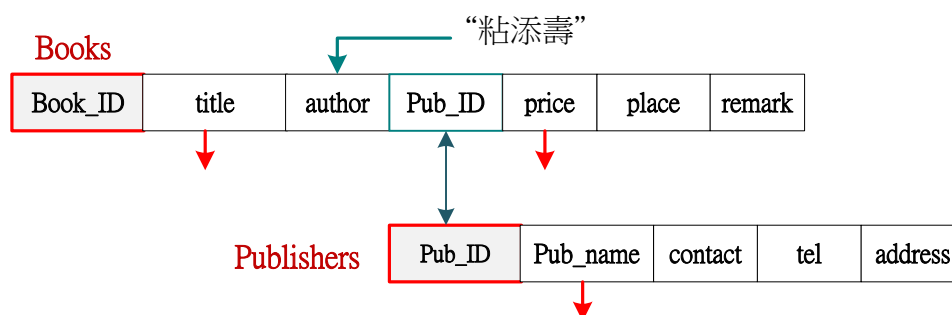
Guest_ID	Book_ID	date
A001	1	2017-11-14
A001	10	2017-10-25
A002	2	2017-11-15
A002	4	2017-11-01
A002	11	2017-10-26
A003	3	2017-11-02

11-2-8 測試資料庫功能

資料庫建構完成後，我們依照幾個系統需求，來驗證其功能是否滿足，如下：

(A) 驗證 1：查詢某位作者有哪些藏書

查詢圖書館有收藏作者粘添壽那些書籍，請書籍名稱、出版社與價格，SQL 命令如下：



```
select A.title, B.pub_name, A.price
from Books as A, publishers as b
where A.author = "粘添壽"
And A.Pub_ID = B.Pub_ID;
```

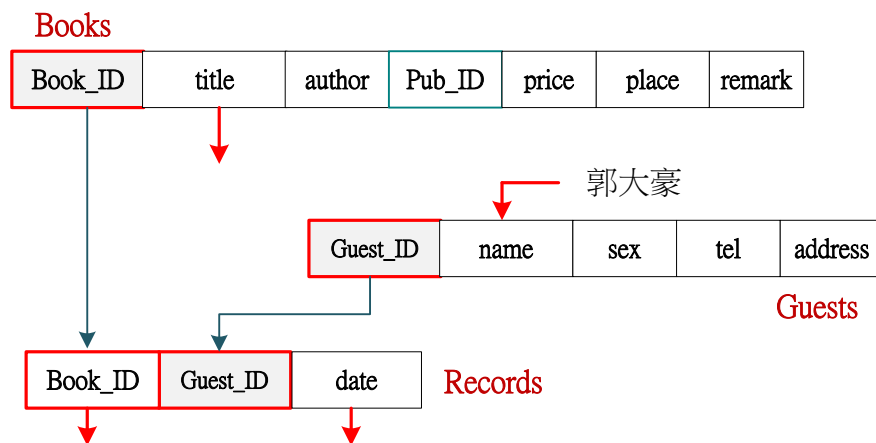
執行結果：

設計

title	pub_name	price
電腦網路與連結技術	旗標圖書出版公司	600
Unix/Linux 作業系統	千葉出版公司	500
資訊與網路安全技術	翻轉工作室	500
Java 程式設計(一)	真美出版公司	500
網際網路原理與運用	大台北圖書公司	500
Java 程式設計(二)	學冠行銷公司	200
資料庫系統概論	真美出版公司	400
資料庫程式設計	真美出版公司	200
伺服器系統管理 - CentOS	真美出版公司	500
電子商務安全	真美出版公司	500

(B) 驗證 2：查詢村民借用那些書，以及借書日期

查詢郭大偉先生借了那些書籍，以及借用日期，SQL 命令如下：



```
select A.title, B.date
from books as A, (select Book_ID, date
                  from records
                  where Guest_ID = (select guest_ID
                                    from guests
                                    where name = "郭大偉")) as B
where A.Book_ID = B.Book_ID;
```

執行結果：

title	date
三國演義	2017-11-14
Unix/Linux 作業系統	2017-10-25

(C) 驗證 3：由圖書公司查詢有哪些書

查詢圖書館有那些翻轉工作室出版的書籍，請列印出書名、作者與價格，SQL 命

設計

令如下：

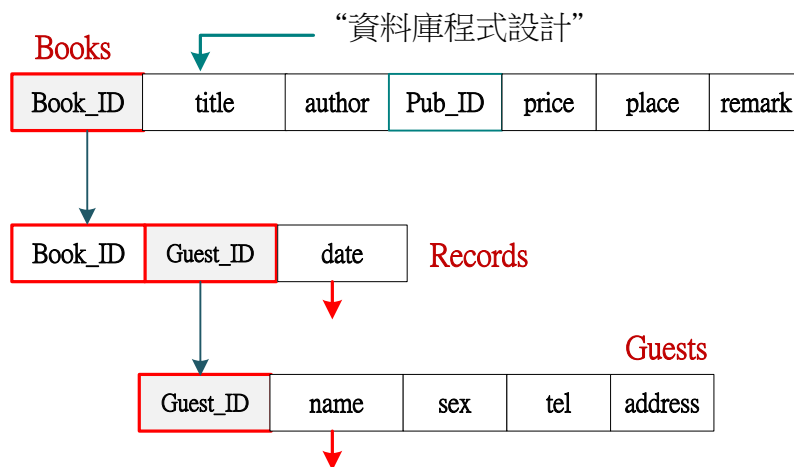
```
select title, author, price
from books
where pub_ID = (select Pub_ID
                from publishers
                where pub_name = "翻轉工作室");
```

執行結果：

title	author	price
孫子兵法	孫子	500
孟子	孟子	500
資訊與網路安全技術	粘添壽	500

(D) 驗證 4：查詢書本出借狀況

查詢『資料庫程式設計』這本書出借的情況，如已出借請列印出借書人與日期。



```
select A.name, B.date
from guests as A, (select Guest_ID, date
                  from records
                  where Book_ID = (select Book_ID
                                  from books
                                  where title = "資料庫程式設計")) as B
where A.guest_ID = B.guest_ID;
```

執行結果：

設計

name	date
孫美麗	2017-10-28

(D) 驗證 5：更新 Records 資料表

測試當村民編號 Guest_ID = A013 借用書籍編號 Book_ID = 12 時，更新 Records 資料表，SQL 命令如下：

```
Insert into records
  values("A013", "12", "");
update records
set date = (select date(now()))
where Guest_ID = "A013"
and Book_ID = "12";
```

接著查詢 Records 資料表是否有增列此筆資料，SQL 命令如下：

```
select *
from records
where Guest_ID = "A013";
```

執行結果：

Guest_ID	Book_ID	date
A013	12	2017-11-15
A013	18	2017-10-29

11-3 進階研討 – 外來鍵 Cascade**11-3-1 級聯 (Cascade)**

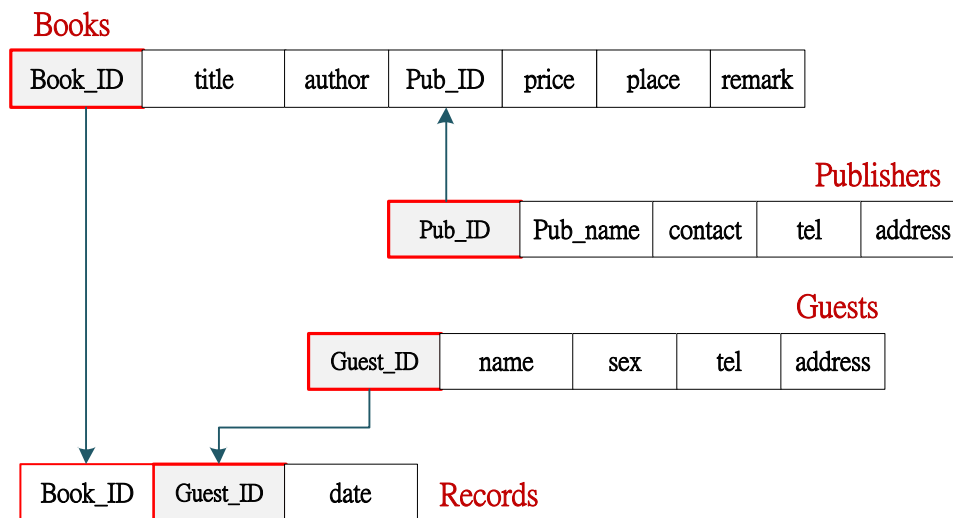
『外來鍵』(Foreign Key) 表示資料表某一欄位參考到另一資料表的某一欄位，但該欄位大多是被參考資料表的『主鍵』(Primary Key)。亦是，外來鍵是跟隨另一資料表的主鍵而生，當被參考的主鍵內的『值』被刪除或更新時，外來鍵相對的該『值』，是否隨著被刪除或更新，這就是『Cascade』的問題。但發生 Cascade 狀況並非僅兩個資料表之間，也許會發生在多個資料表之間。吾人以『Temple_Library』資料庫為例，下圖是它的 E-R 關聯圖，從中可以看出，Records 的 Book_ID 是主鍵也是外來鍵，它參

設計

考到 Books 的 Book_ID 欄位，又 Books 的 Pub_ID 欄位是拽來鍵，參考到 Publishers 表的 Pub_ID 欄位。

Pub_ID (Publishers) -> Pub_ID (Books) -> Book_ID (Books) -> Book_ID (Records)

當 publishers 表中的『翻轉工作室』出版社被更新或刪除時，是否 Books 表中有『翻轉工作室』的 Pub_ID 的紀錄是否要隨著被更新或刪除，又當 Books 表的相關紀錄被更新或刪除時，Records 的相關紀錄是否又隨之更改，因此稱之為『**級聯**』(Cascade)。



11-3-2 InnoDB 與 Cascade 語法

MySQL 資料庫的儲存引擎有兩種型態：MyISAM 與 InnoDB 兩種型態，前者是早期版本，目前較少用。對於多人使用、安全性要求較高的環境大多使用 InnoDB 型態，才具有 Cascade 之功能，目前在 MySQL 建立表格時，預設型態都指定 InnoDB 型態(功能比較請自行參考其他資料)，如下圖所示

資料表	動作	資料列數	類型	編碼與排
books	★ 瀏覽 結構 搜尋 新增 清空 刪除	18	InnoDB	utf8mb4_
guests	★ 瀏覽 結構 搜尋 新增 清空 刪除	19	InnoDB	utf8mb4_
publishers	★ 瀏覽 結構 搜尋 新增 清空 刪除	6	InnoDB	utf8mb4_
records	★ 瀏覽 結構 搜尋 新增 清空 刪除	16	InnoDB	utf8mb4_
4 張資料表 總計			59	InnoDB

宣告 Cascade 語法如下：(在 Foreign Key 語句增加)

```
Create Table Table_name (
欄位宣告,
```

設計

```

.....
Foreign Key .. References .. On Delete { Cascade | Set Null | Not Action| Restrict }
Foreign Key .. References .. On Update { Cascade | Set Null | Not Action| Restrict }
) Engine InnoDB;

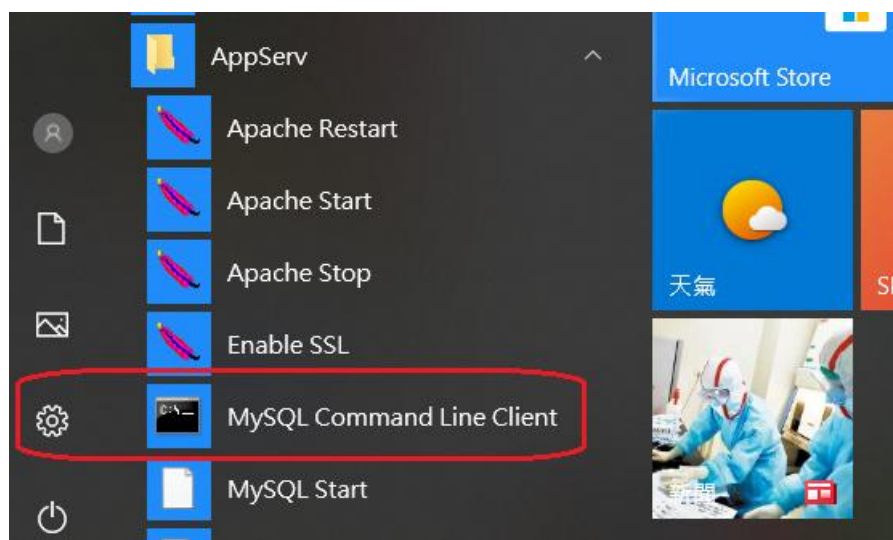
```

選項敘述如下：

- **On Delete** 或 **On Update**：當發生刪除或更新動作時，依後面宣告處理。
- **Cascade**：所有關聯的紀錄也會跟隨刪除或更新。
- **Set Null**：將所有關聯的紀錄設定成 Null。(對應欄位不可以 Not Null)
- **No Action**：即是不處理。(預設值)(系統會禁止刪除或更新動作)
- **Restrict**：與 No Action 相同。

11-3-3 刪除外來鍵設定

原來 temp_library 資料庫中表格並沒有規劃 Cascade 功能，吾人需要將原來設定的外來鍵刪除後，再重新建立。MySQL 對於刪除外來鍵有所限制，須找出所該外來鍵的 "Foreign Key id"，方法是：由桌面 => 開始 => 由選單中找出 AppServ => MySQL Command Line，如下圖所示。



進入交談式視窗後，輸入密碼 (12345678)，接著如下操作，則能查出 Books 與 Records 表外來鍵的識別碼。

```

Enter password: *****)
Welcome to the MySQL monitor.  Commands end with ; or \g.

```

設計

```
mysql> use temp_library;
Database changed
mysql> show create table books;
....
| books | CREATE TABLE `books` (
  `book_ID` int(11) NOT NULL AUTO_INCREMENT,
  `title` char(50) NOT NULL,
  `author` char(20) NOT NULL,
  `Pub_ID` int(11) NOT NULL,
  `price` int(11) DEFAULT NULL,
  `place` char(20) DEFAULT NULL,
  `remark` char(50) DEFAULT NULL,
  PRIMARY KEY (`book_ID`),
  KEY `Pub_ID` (`Pub_ID`),
  CONSTRAINT `books_ibfk_1` FOREIGN KEY (`Pub_ID`) REFERENCES
`publishers` (`pub_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=19 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci |
.....
mysql> show create table records;
....
records | CREATE TABLE `records` (
  `Guest_ID` char(20) NOT NULL,
  `Book_ID` int(11) NOT NULL,
  `date` date DEFAULT NULL,
  PRIMARY KEY (`Guest_ID`,`Book_ID`),
  KEY `Book_ID` (`Book_ID`),
  CONSTRAINT `records_ibfk_1` FOREIGN KEY (`Book_ID`) REFERENCES
`books` (`book_ID`),
  CONSTRAINT `records_ibfk_2` FOREIGN KEY (`Guest_ID`) REFERENCES
`guests` (`Guest_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci |
.....
```

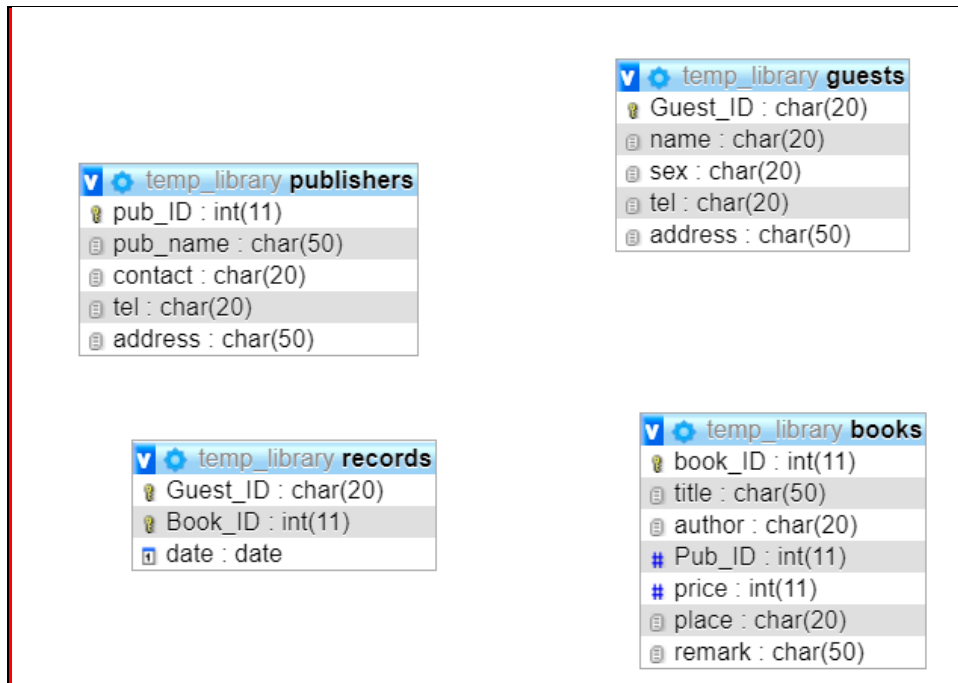
回到 phpMyAdmin 網頁則以 SQL 命令刪除外來鍵如下：

```
Alter Table Books
  Drop Foreign key `books_ibfk_1`;
Alter Table Records
  Drop Foreign key `records_ibfk_1`;
Alter Table Records
  Drop Foreign key `records_ibfk_2`;
```

設計



刪除完外來鍵之後，由 AppServ 功能選單內『設計』可觀察到各資料表之間都沒有聯繫。



11-3-4 驗證外來鍵 Cascade 功能

吾人在『Temp-Library』資料庫具有 Cascade 功能。吾人希望系統具有下類規範：

1. Books 外來鍵(Pub_ID) 沒有宣告 Cascade 功能 (預設值為 **Not Action**)，則表示當 Publishers 表內記錄有被 Books 表參考到時，則不允許刪除或更新，除非先除刪除 Books 記錄，再刪除 Publishers 記錄。(Update/Delete Not Action)
2. Books 表中某筆資料被刪除，則 Records 表內相關紀錄也被刪除。(Delete Cascade)
3. Guests 表中某筆記錄(Guest_ID) 變更時，Records 表內相關紀錄也隨之變更。(Update cascade)
4. Guests 表內某筆記錄被刪除時，則 Records 表內相對的紀錄也被刪除。(Delete Cascade)

依照上述規範，吾人規劃外來鍵具有 Cascade 的命令如下：

設計

```

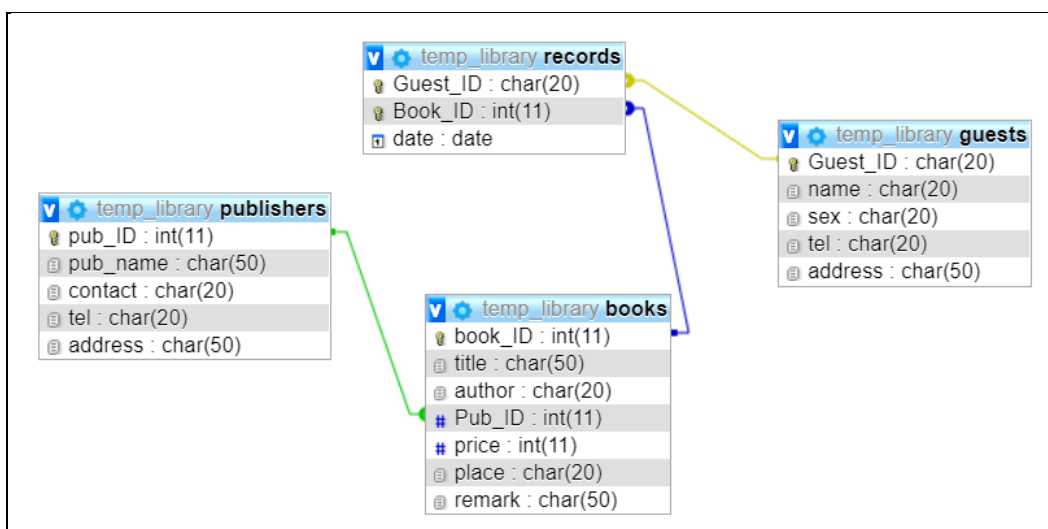
/* 第一條規範 (預設值 Not Action)*/
Alter Table Books ADD Foreign key (Pub_ID) References publishers(Pub_ID);

/* 第二條規範 */
Alter Table Records ADD Foreign key (Book_ID) References Books(Book_ID)
On Delete Cascade;

/* 第三、四條規範 */
Alter Table Records ADD Foreign key (Guest_ID) References Guests(Guest_ID)
On Delete Cascade On Update Cascade;

```

重新建立之後，可觀察到各資料表之間的關係圖，如下：



(B) 驗證：Not Action (第一條規範)

由 **Publishers** 表中刪除某一出版社，觀察是否允許刪除，如不行，則先刪除 **Books** 有參考到記錄，再刪除 **Publishers** 中記錄。

觀察 **Books** 是否有『翻轉工作室』(Pub_ID = 3)的相關紀錄，再刪除 **Publishers** 表『真美出版公司』(Pub_ID = 3)，記錄是否允許。

- 首先 **Books** 表內是否有『真美出版公司』的相關紀錄。如有，則刪除 **publishers** 表內『真美出版公司』記錄是否可行。

```

select *
from books
where Pub_id = (select Pub_ID
                from publishers
                where pub_name = "翻轉工作室");

```


設計

Book_ID	title	author	Pub_ID	price	place	remark
5	孫子兵法	孫子	3	500	B-1-3	張友德先生贈
6	孟子	孟子	3	500	B-1-4	張友德先生贈
11	資訊與網路安全技術	粘添壽	3	500	C-1-4	採購

```
Delete
from publishers
where pub_name = "翻轉工作室";
```

錯誤**SQL 查詢：**

```
DELETE
from publishers
where pub_name = "翻轉工作室"
```

MySQL 回應：

```
#1451 - Cannot delete or update a parent row: a foreign key constraint fails:
FOREIGN KEY (`Pub_ID`) REFERENCES `publishers` (`Pub_ID`))
```

(無法刪除，表示該筆記錄有被其它表格參考到。 **Not Action 功能**)

- 吾人必須先刪除掉 Books 有參考到 publishers 表『翻轉工作室』的所有記錄，才允許刪除 publishers 表內的『翻轉工作室』記錄。

```
Delete
from Books
where pub_id = (select Pub_ID
                from publishers
                where pub_name = "翻轉工作室");

select *
from Books
where pub_id = (select Pub_ID
                from publishers
                where pub_name = "翻轉工作室");
```

Book_ID	title	author	Pub_ID	price	place	remark
---------	-------	--------	--------	-------	-------	--------

傳回空值，表示 Books 表內相關 "翻轉工作室" 記錄被刪除

```
Delete
from publishers
where pub_name = "翻轉工作室";
```

設計

```
select *
from publishers
where pub_name = "翻轉工作室";
```

Pub_ID	Pub_name	contact	tel	address

傳回空值，表示 Publishers 表內 "翻轉工作室" 記錄被刪除

(C) 驗證：刪除某一書籍，Records 表是否隨之改變。(第二條規範)

由 Books 表中刪除『紅樓夢』書籍，觀察 Records 是否還有人租用該本書。

- 首先查詢『三國演義』的 Book_id 為何。

```
select Book_ID
from Books
where title = "三國演義";
```

Book_ID
1

- 再由 Books 表內刪除 Book_id = 1 的記錄，之後觀察 Records 記錄中是否還有借用該本書的記錄，如果沒有則表示 Records 跟隨著刪除相關紀錄，符合 Delete Cascade 之功能。

```
delete
from Books
where Book_id = "1";
```

```
select *
from Records
where Book_id = "1";
```

Guest_ID	Book_ID	date

傳回空值，Delete Cascade 正確

(D) 驗證：變更某 Guest_ID，Records 表是否隨之改變。(第三條規範)

將『張彥宏』先生的 Guest_ID 改為 B002，觀察 Records 表內相關紀錄是否被更改。

設計

- 首先查詢『張彥宏』的 Guest_id 為何，並觀察他的借書紀錄(Records)。

```
select guest_ID
from Guests
where name= "張彥宏";
```

guest_ID
A002

```
select *
from records
where guest_id = (select guest_ID
                  from Guests
                  where name= "張彥宏");
```

Guest_ID	Book_ID	date
A002	4	2017-11-01
A002	11	2017-10-26

- 再由 Guests 表將『張彥宏』的 Guest_id 改為 B002，再觀察他的借書紀錄是否有所改變。之後觀察 Records 記錄中『張彥宏』借書記錄是否有改變，如果沒有則表示符合 Update Cascade 之功能。

```
update Guests
set guest_id = "B002"
where name= "張彥宏";
```

```
select *
from records
where guest_id = (select guest_ID
                  from Guests
                  where name= "張彥宏");
```

Guest_ID	Book_ID	date
B002	2	2017-11-15
B002	4	2017-11-01

Guest_ID 被變更，Update Cascade 正確

(E) 驗證：刪除某 Guest_ID，Records 表紀錄是否跟隨刪除。(第四條規範)

由 Guests 表中將『張彥宏』(Guest_ID=B002)先生記錄刪除，觀察 Records 表內是否還有它的借書紀錄。

- 首先查詢『張彥宏』的 Guest_id 為何，並觀察他的借書紀錄(Records)。

設計

```
select guest_ID
from Guests
where name= "張彥宏";
```

guest_ID
B002

```
select *
from records
where guest_id = (select guest_ID
                  from Guests
                  where name= "張彥宏");
```

Guest_ID	Book_ID	date
B002	2	2017-11-15
B002	4	2017-11-01

- 再由 Guests 表中刪除『張彥宏』記錄，再觀察 Records 記錄中『張彥宏』(Guest_id = "B002")是否還存在，如果沒有則表示符合 Delete Cascade 之功能。

```
delete
from Guests
where Guest_id = "B002";
```

```
select *
from records
where guest_id = " B002";
```

Guest_ID	Book_ID	date

傳回空值，Delete Cascade 正確

11-4 自我挑戰

11-4-1 人事管理系統

(A) 系統需求與規格

『大石精密公司』接受委託製造各種生產機械，目前大約有 300 位員工，老闆期望建立一套人事管理系統，希望有下列功能：

設計

- 可登錄、刪除或更新員工資料。
- 可查詢每位員工服務部門、底薪、職務加級、加班時數。
- 每月可計算員工薪資表。
- 可查詢各部門負責人、員工名單，並計算各部門薪資結構如何。
- 全公司內員工的薪點、加級與津貼都有統一計算標準。

(B) 資料收集

系統分析師到現場收集到商品資料如下：

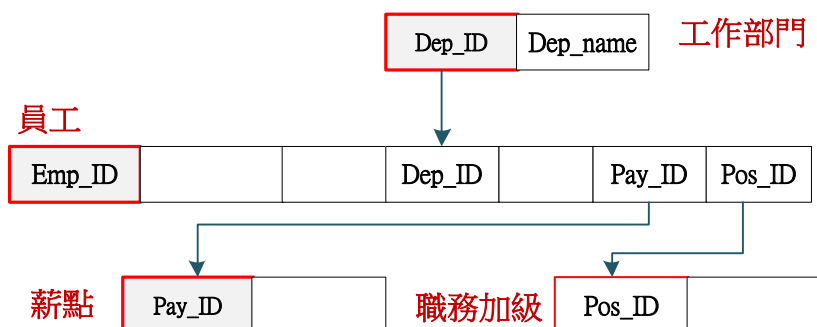
- 員工資料有：員工編號、姓名、地址、電話、性別、出生日期、服務部門、薪點、與職務。
- 薪資類別：薪點、薪點金額、加級金額、加級基點。
- 服務部門：部門名稱、部門地區(高雄、台北、上海、...)。

(C) 完成事項

- (1) 資料庫邏輯設計。
- (2) E-R 關係圖轉換資料表與正規化。
- (3) 實體建置。
- (4) 匯入測試資料。
- (5) 測試資料庫功能。
- (6) 資料庫效能分析。

(C) 提示

以下是提示資料庫關聯圖，並非完整設計。



設計

圖 11-16

11-4-2 員工休閒活動管理系統

(A) 系統需求與規格

『研技半導體設計公司』是接受委託設計多項功能半導體的公司，目前大約有 500 位員工，每天透過電腦設計半導體結構，公司有感於員工過於忙碌，需要有適當休閒活動以調適心身，才能發揮最大工作效益。在公司裡成立許多休閒活動讓同仁參加，但已希望了解每位同仁參與程度，才能給予適當輔導。因此，委託建立一套員工休閒活動管理系統，希望有下列功能：

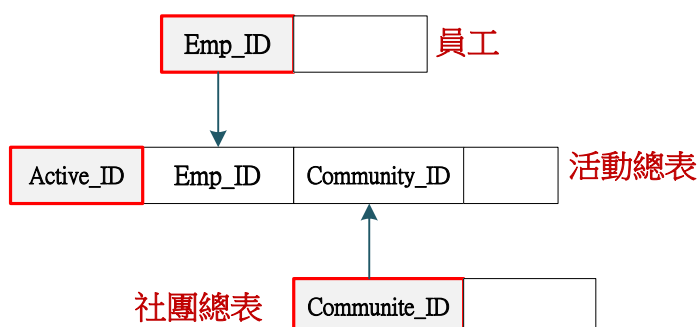
- 員工可以到各項活動社團報名參加。
- 可查詢員工參加了那些社團，並可量化員工身心調適量多寡。
- 可查詢社團裡有哪些員工參加，並可計算社團花費成本多寡。
- 可查詢調適量過低或過高的員工。
- 可查詢花費成本過高或過低的社團。

(B) 資料收集

系統分析師到現場收集到員工與社團資料如下：

- 員工資料有：員工編號、姓名、地址、電話、性別、出生日期。
- 社團資料：社團名稱、每位參與者費用、身心調適量。

(C) 提示



設計圖 11-18(D) 完成事項

- (1) 資料庫邏輯設計。
- (2) E-R 關係圖轉換資料表與正規化。
- (3) 實體建置。
- (4) 匯入測試資料。
- (5) 測試資料庫功能。
- (6) 資料庫效能分析。

(D) 提示：

將關係轉換為資料表。

11-4-3 餐廳管理系統(A) 系統需求與規格

『真善美健康美食餐廳』是一家提供健康美食的專屬餐廳，期望建立一套餐廳管理系統，希望有下列功能：

- 按照桌次點菜，點菜完畢後除了顯示金額外，還顯示卡路里的總和。
- 可隨時查詢桌次菜單。
- 會計部門可依照桌次收費。
- 可統計分析各樣菜被點用的狀況。

(B) 資料收集

系統分析師到現場收集到該餐廳的資料如下：

- 菜單資料：菜單名稱、單價(大、中、小)、卡路里(大、中、小)。
- 桌次：1 ~ 10 桌。

(C) 完成事項

設計

- (1) 資料庫邏輯設計。
- (2) E-R 關係圖轉換資料表與正規化。
- (3) 實體建置。
- (4) 匯入測試資料。
- (5) 測試資料庫功能。
- (6) 資料庫效能分析。

11-4-4 高爾夫會員管理系統

(A) 系統需求與規格

『長青高爾夫球場』期望建立一套會員管理系統，紀錄與分析會員繳費與消費情況如何，希望有下列功能：

- 可登錄會員資料與查詢繳費狀況。
- 可查詢會員消費狀況。
- 可查詢消費最高與最低會員，消費較高會員可以升級會員等級，享受不同待遇。

(B) 資料收集

系統分析師到現場收集到的資料如下：

- 會員資料有：姓名、地址、電話、性別、地址、出生日期。
- 會員等級：金卡(年費 2 萬元)、白金卡(年費 1.5 萬元)、貴賓卡(年費 1 萬元)。
- 臨場消費：假日：1200 元、非假日：1000 元。

(C) 提示

設計

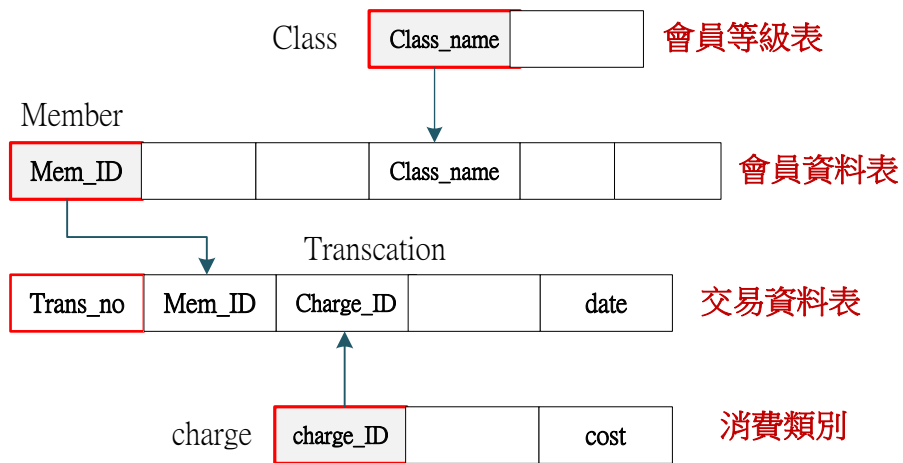


圖 11-19

(D) 完成事項

- (1) 資料庫邏輯設計。
- (2) E-R 關係圖轉換資料表與正規化。
- (3) 實體建置。
- (4) 匯入測試資料。
- (5) 測試資料庫功能。
- (6) 資料庫效能分析。