

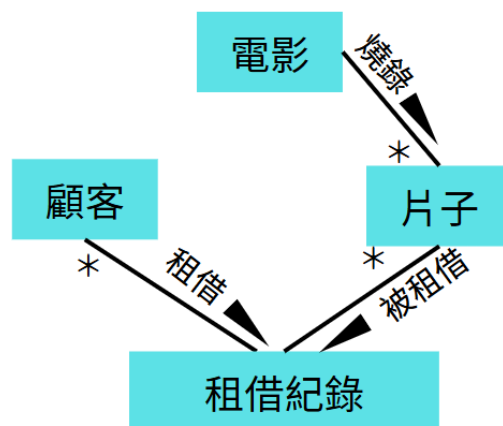
1. Candidate Classes :

片子
電影
顧客
計價
租片紀錄
租借日期
顧客積點

2. Define Classes :

片子：儲存影片數據的實體，可能是光碟的形式，可以用商品編號區別
電影：這裡指的是電影的種類、內容、劇情等，可以用電影的編號區別
顧客：前往租片的顧客，可以用顧客編號區別
租片紀錄：每個顧客過去租過的片、總價格

3. Establish Associations :



4. Expand Many-to-Many Associations :

目前沒看到

5. Attributes :

電影：
種類（新、舊、兒童）
* 電影編號
電影名稱

劇情

演員

片子：

*商品編號

電影編號

租借價格

租借日期

顧客：

*顧客編號

顧客名稱

顧客積點

租借紀錄：

*編號

顧客編號

影片編號

租借天數

租借價格

6. Normalization

電影：

* 電影編號

電影名稱

種類（新、舊、兒童）

片子：

*商品編號

電影

顧客：

*顧客編號

顧客名稱

顧客積點

租借紀錄：

*編號

顧客編號

影片編號

租借天數

租借價格

7. Operations

電影：

getMovieInfo():

Input: :

Output :

電影種類

Algorithm :

回傳 Movie 物件中的”種類”Attribute

片子：

getPrice(Days) :

Input :

Days : 租借天數

Output :

價格

Algorithm :

利用自己的”電影編號” Attribute 查詢電影的種類

依據種類以及天數算出價格並回傳

getPoints():

Input :

Output :

該片子能夠提供的顧客積點

Algorithm :

利用影片編號查詢電影種類後回傳能提供的顧客積點

顧客：

addPoints(float points)

Input :

顧客新增的點數

Output :

Algorithm :

為顧客添加點數

getPoints(float points)

Input :

取得顧客的總點數

Output :

顧客的總點數

Algorithm :

回傳顧客的點數

租片紀錄：

rentVideo(Customer, Video, Days):

Input :

顧客物件、片子物件

Output :

Algorithm :

登入顧客、片子、天數

從 Video 計算價格、顧客積點

為顧客添加顧客積點

登入價格

printStatement (Customer):

Input :

顧客物件

Output :

過去租過的片子、租片的總消費金額、顧客積點總數

Algorithm :

從”租片紀錄”查詢租過的片子以及計算總金額

回傳它們和顧客積點總數