

- ★期中考：
1. ER (20pt.)
 2. Branch view 產生之資料表 (20pt.)
 3. 正規化 (40pt.)
 4. 參考性限制 (20pt.)

Ch4

(邏輯)資料庫設計與實體關聯模型 (Logical) database design and ER model

★ Entity Relationship.

實體設計

SQL Server 資料放在 data 裡, 名 .mdf \ .ldf
[要把檔案設到家中, 要先卸載
→ 回到家按附加

查詢客
戶資料

查詢訂
單資料

查詢供
應商資
料

客戶訂
單明細
報表

.....

APP訂
單管理
系統

系統分析設計

SASD

(Schema)

客戶編號
客戶名稱

產品編號
產品名稱

供應商編號
供應商名稱

.....

找出各項關係

客戶

訂單

產品

供應
商

(ER圖)

實體關係圖

第一次小考

mapping

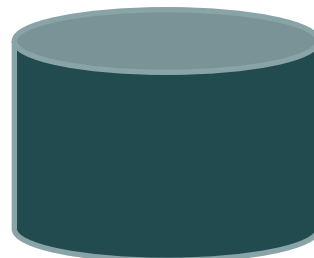
客戶(客戶編號.....)
訂單(訂單編號.....)
產品(產品編號.....)
訂單明細(.....)
供應商(.....)

第二次
小考

(Normalization)

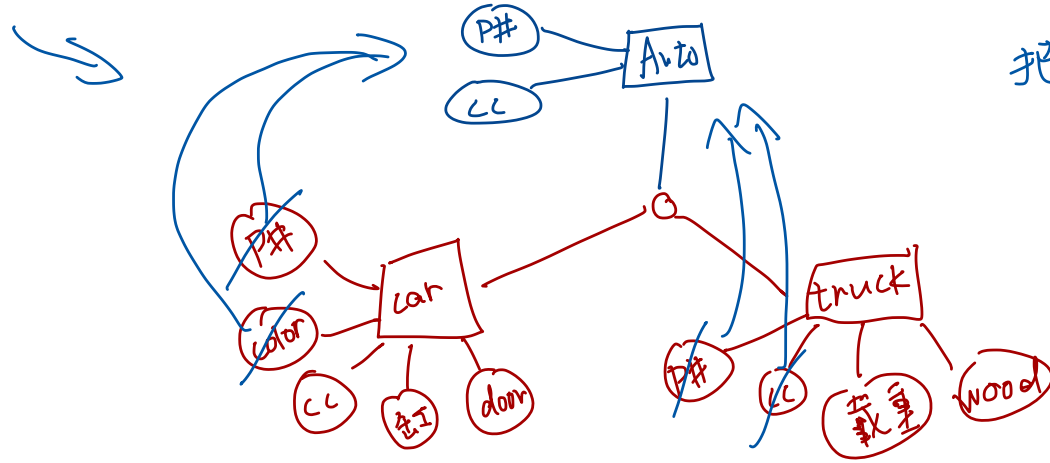
正規化

客戶(客戶編號.....)
訂單(訂單編號.....)
產品(產品編號.....)
訂單明細(.....)
供應商(.....)



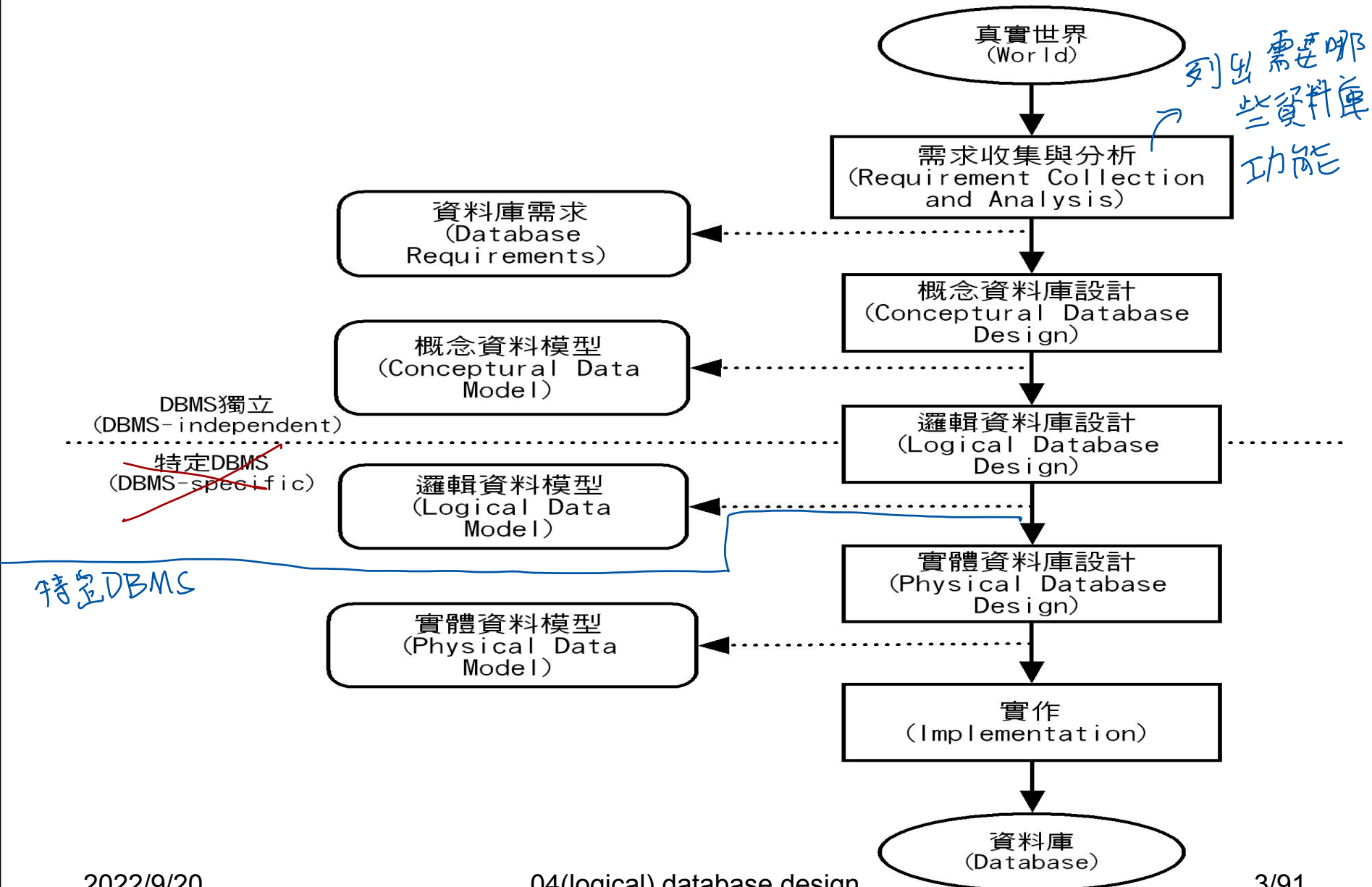
ER: Entity Relationship ^{mapping} 1條規則

EER: Extended Entity Relationship 把物件導向概念加入 — 2條規則



把重複的設上去, 成為複實體

資料庫設計方法論



資料庫設計方法論

Database design methodology

- 使用程序、技術、工具、使用手冊等結構性的方式，來幫助資料庫設計的整個流程稱為資料庫設計方法
- 兩個主要階段
 - 邏輯資料庫設計階段(**Logical** database design)
 - 實體資料庫設計階段(**Physical** database design)

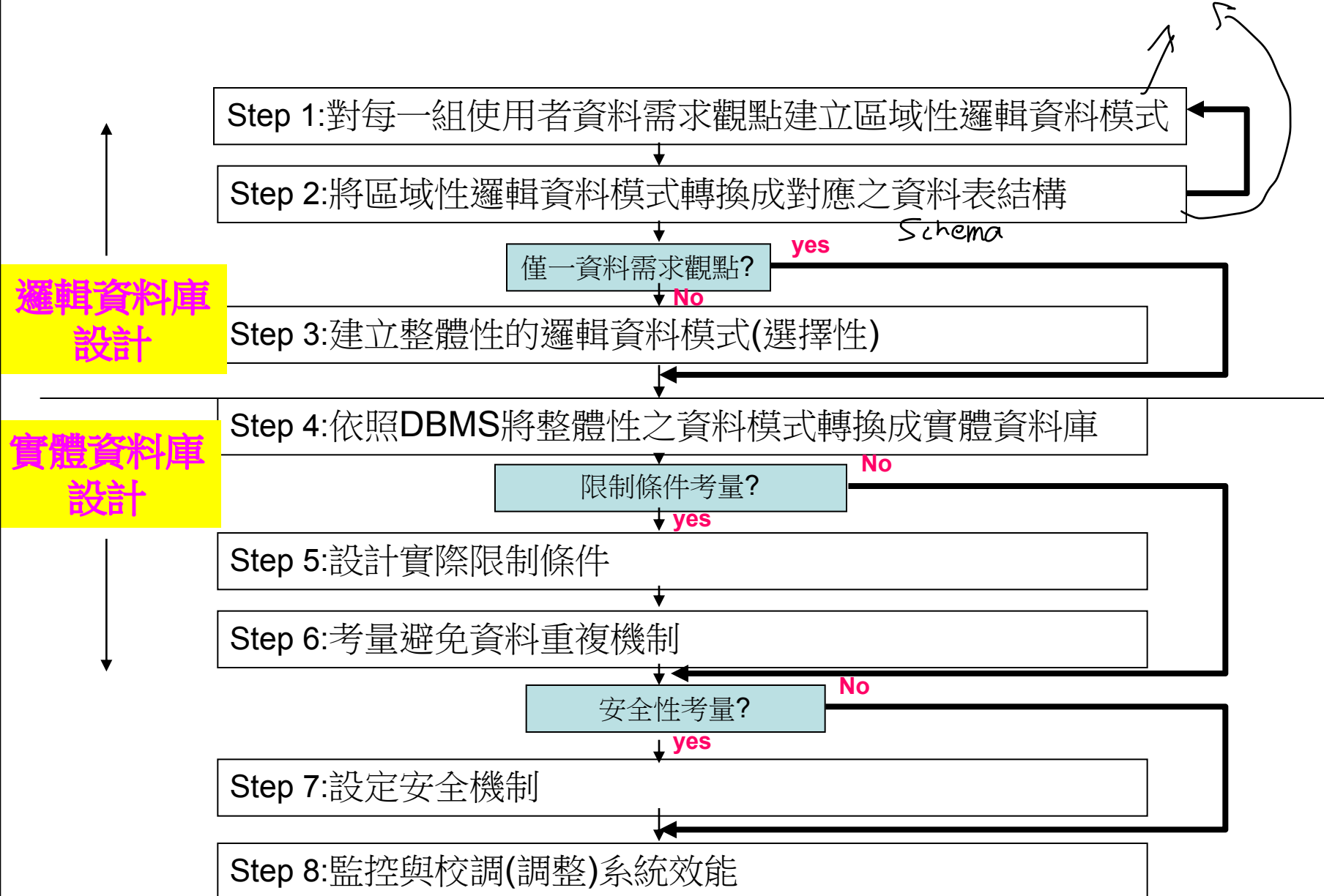
建圖

邏輯與實體資料庫設計

- 邏輯資料庫設計 (ER圖)
 - 使用某種資料模式(工具)以建構組織運作所需的資料模式
 - 此階段和使用哪種DBMS及限制條件無關
 - 此階段應完成重要實體、實體間之關係，並依照轉換規則轉換成相關資料表(tables)
 - 強調資料庫結構的正確性
- 實體資料庫設計
 - 將邏輯資料庫設計產生的資料庫結構，依照某DBMS建構在硬碟上
 - 包括建構資料表、檔案結構、方便取存的索引、整合性限制與安全性考量
 - 強調資料庫的效能

瀏覽整個資料庫開發流程

期中20分



Step 1：對每一使用者需求觀點建立 區域性邏輯資料模式

- 為每一組使用者資料需求觀點建立區域性邏輯資料模式(local logical data model)，包括：
 - 實體(Entities)
 - 關係(Relationships)
 - 屬性(Attributes)、屬性範圍(定義域)(attribute domains)
 - 候選鍵(Candidate keys)、主索引(primary keys)、次索引(alternative keys)
 - 整合性限制(Integrity constraints)

step 1的工作

- Step 1.1 找出相關實體
- Step 1.2 (E/R model) 找出實體間的關係
- Step 1.3 找出各實體或關係的屬性(屬性(欄位))
- Step 1.4 決定屬性範圍(定義域) (有效範圍·決定像那個部門可用)
- Step 1.5 決定候選鍵及主索引 *primary key*
- Step 1.6 將資料實體一般化或特殊化(EER範圍, 省略)
- Step 1.7 檢查產生的資料模式是否滿足使用者交易需求

[Go to step 2](#)

Step 1.1 找出相關實體

- 找出符合使用者需求觀點的主要物件(實體)
 - 找出符合使用者資料需求的屬性，並將相關的屬性群集(擺在同一資料實體)
 - 通常以名詞或名詞片語出現的屬性，如 (staff_no, staff name, catalog_no, title, daily rental rate, purchase price)
 - 找出主要物件，如人、事、時、地、物
 - 分析蠻主觀的，雖然可能設計不同，但效果應會相同(殊途同歸)

StayHome 個案描述

- 1982在西雅圖建立第一家公司，目前全美有許多分公司，成功的原因是提供很好的服務給客戶，並且擁有完整豐富的影帶供租賃
- 目前約有 2000 員工，分布在 100個分公司，新員工加入時，並須進行員工登錄，每一分公司有一位經理及數位主任
- 每一部影帶，有目錄編號(catalog number)，相同錄影帶目錄編號相同，但每一部錄影帶都有唯一的錄影帶編號(video number)
- 客戶必需加入會員才可以租賃影帶，客戶可以在不同分公司同時登錄為會員，一次最多租10部
- 目前約有 100,000 會員；訂單由總公司的採購向供應商下訂單採購
- StayHome公司不斷的成長，目前他的瓶頸是處理每天公司產生的大量資料，所以，急需開發一資料庫來解決資料管理的問題

StayHome 實體

- 從 StayHome 的個案裡，可以找出下列實體
 - Branch(分公司)、Staff(員工)、Video(錄影帶類別)、VideoForRent(錄影帶)、Member(會員)、RentalAgreement(租賃明細)、**Actor(演員)**、**Director(導演)**
- 要去想使用者找片子會找什麼

Entity name 實體	Description	Aliases 未來如果兩個實體 要合併用	Occurrence 筆數
Branch	工作地方	Outlet、branch outlet	美國所有重要城市 均有1到數個分公司
Staff	簡單描述StayHome 公司員工資料	Employee	每一位員工均隸屬 某一分公司
.....			Return to step 1 menu

Step 1.2 找出實體間的關係

- 一般關係在收集的資料中呈現動詞關係

- Branch **Has** Staff

- Branch **IsAllocated** VideoForRent

- VideoForRent **IsPartOf** RentalAgreement

~~~~~  
step 1.1 為名詞關係

- 使用E/R (Entity Relationship)模式表示
- 決定結構性限制(structural constraints or multiplicity) ，包括：
  - 參與比率(基數比)(cardinality ratio)
  - 參與限制(participation constraint)

ER圖

結構性限制條件

Structural constrain.

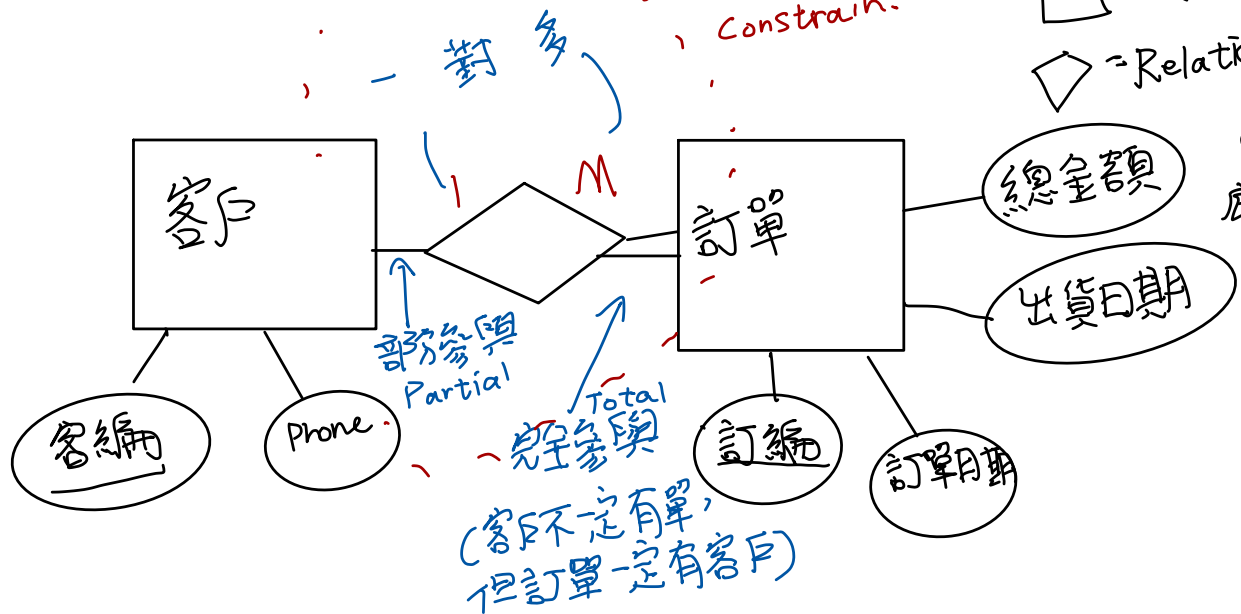
□ = 實體 Entity

◇ = Relationship

○ = 欄位

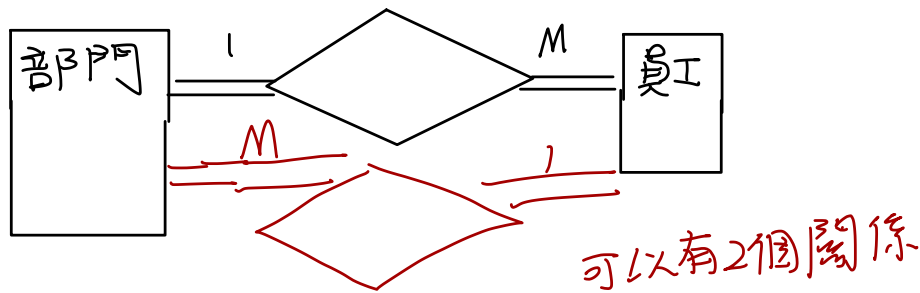
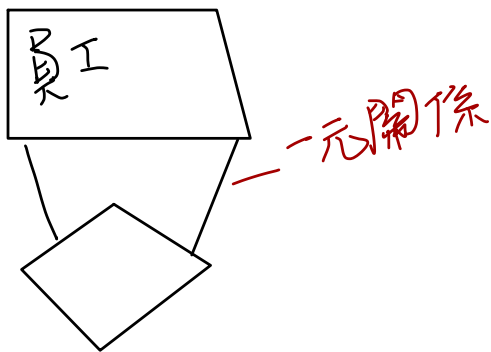
底線 = Primary key  
- 實體可多個 Primary key

作候選 key



# StayHome個案的所有關係

| Entity          | Relationships         | Entity                |
|-----------------|-----------------------|-----------------------|
| Branch          | Has<br>IsAllocated    | Staff<br>VideoForRent |
| Branch, Staff * | Registers             | Member                |
| Staff           | Manages<br>Supervises | Branch<br>Staff       |
| Video           | Is                    | VideoForRent          |
| VideoForRent    | IsPartOf              | RentalAgreement       |
| Member          | Request               | RentalAgreement       |
| Actor           | PlaysIn               | Video                 |
| Director        | Directs               | Video                 |



二元關係  
Binary Relationship

若 Entity = 2, 則稱多元關係

# E/R 模式(Entity Relationship model)

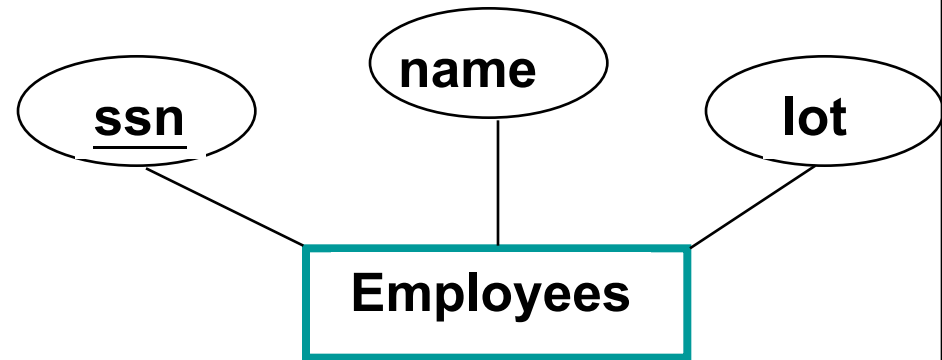
- **E/R model** 是在1976年由 Peter Chen 提出，主要包括**實體**、**關係**、**屬性**、**結構性限制條件**  
  
- 實體是真實世界存在的一個事實，它可能是**真實存在的一個具體物件**，如某特定人(公司員工)、車；也可能只是存在的**某種概念**，如公司、工作、大學課程
- 每一個實體均具有自己的性質來描述他本身，稱為**屬性(屬性(欄位))(attribute)**  
例如：員工(姓名，年齡、地址、薪資，職稱)

# 簡易資料庫設計方法

## YSL's theory

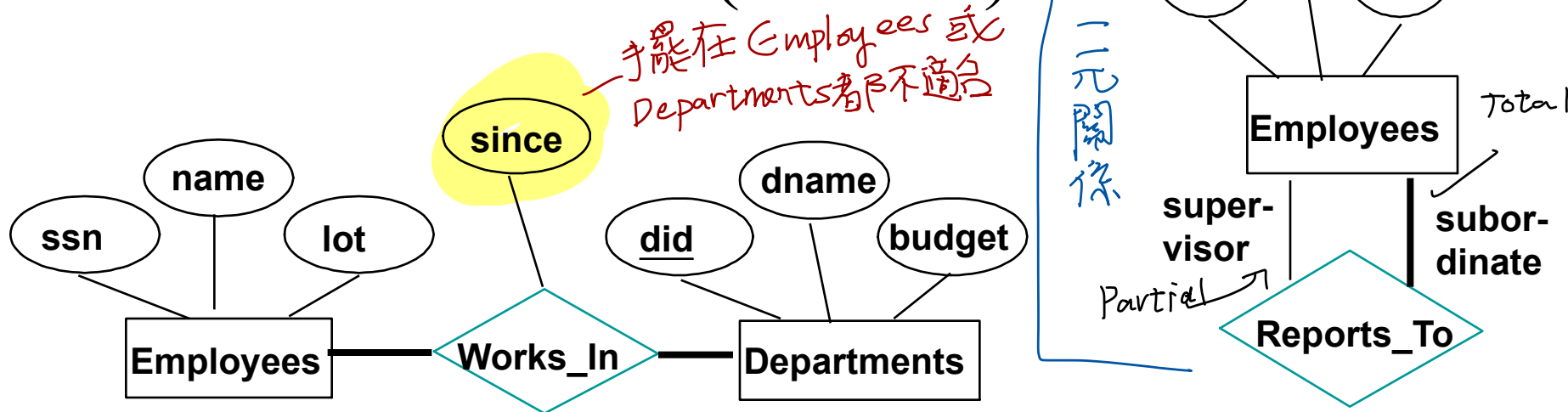
- ❖ 以資料及資料間之關係呈現使用者需求
  - 收集所有資料需求
  - 收集資料屬性(欄位)以滿足上述需求
  - 將相同資料屬性(欄位)放同一群組
  - 建立群組間的關係
  - 畫出ER圖
  - 將ER圖轉換為資料庫結構
- ❖ 正規化
- ❖ 決定特殊限制
- ❖ 合併各子系統(若有一個以上資料需求觀點)

# ER Model Basics



- **實體(Entity):** 真實世界的一個物件，包含許多描述實體的屬性(屬性(欄位)) (attributes )
- **實體集合(Entity Set):** 相同實體的集合E.g., all employees

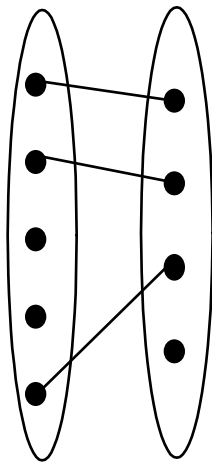
# ER Model Basics (Contd.)



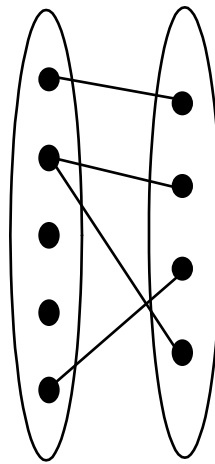
- **關係(Relationship):** 實體之間的關連性，例如 Attishoo works in Pharmacy department
- **關係集合(Relationship Set):** 相同關係的集合(Collection of similar relationships)

# (關係)參與比率(或基數比)

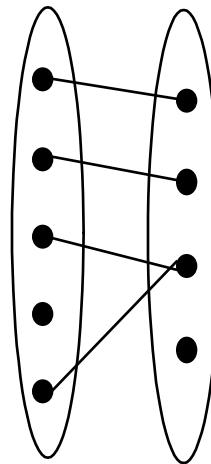
## Cardinality ratio



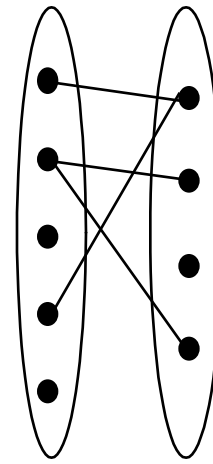
1-to-1



1-to Many



Many-to-1



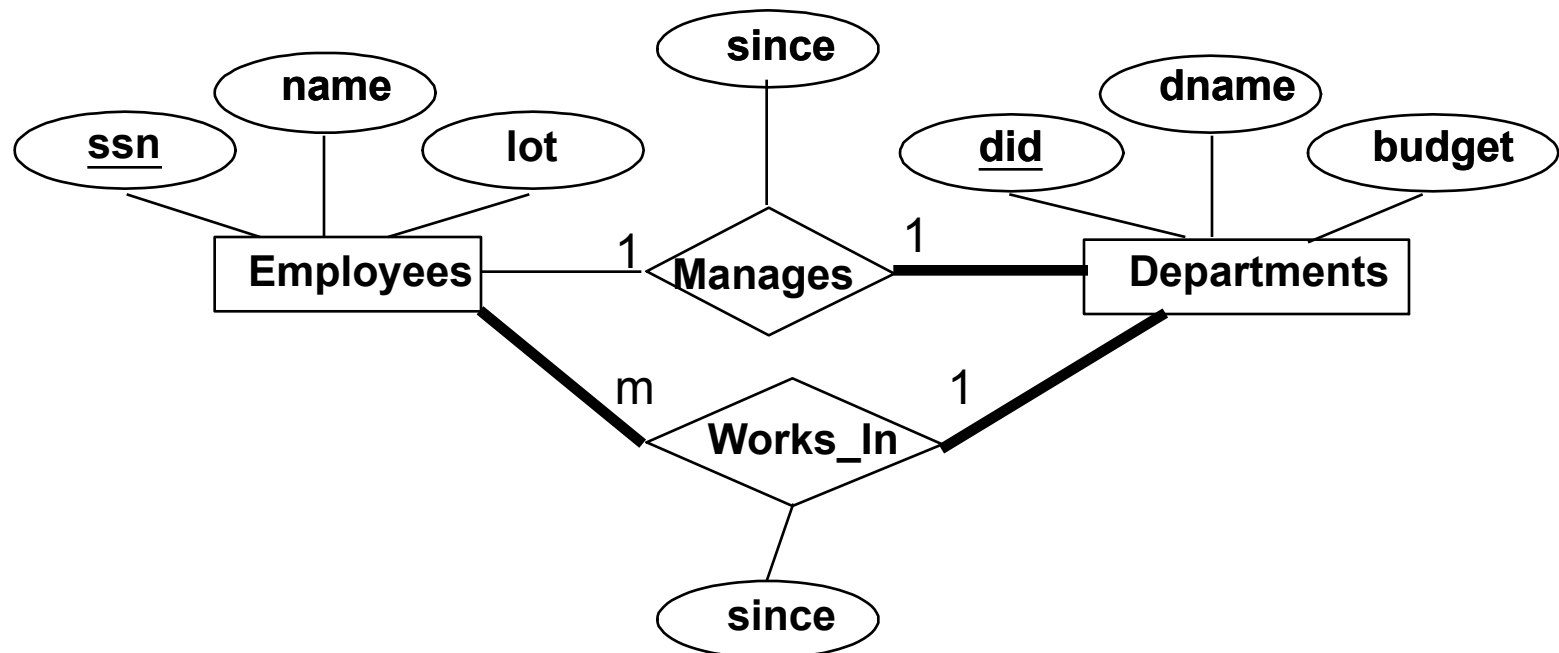
Many-to-Many

如訂單與產品關係

# 參與限制

## Participation Constraints

- 完全參與(*total participation*)
- 部分參與(*partial participation*)

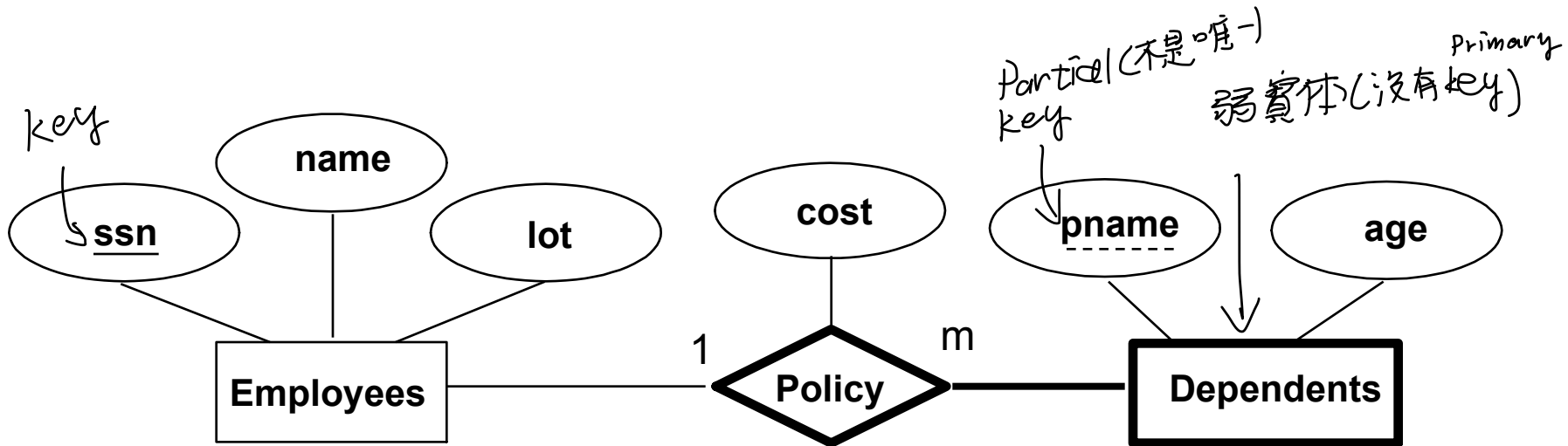


# 弱實體

實體有2種：  
一般實體、弱實體

## Weak Entities

- 弱實體(*weak entity*)無任何索引屬性(欄位)的實體；必須結合擁有者實體(*owner entity*)主鍵才能唯一識別
  - 弱實體基本上，依附於擁有者實體之存在才存在



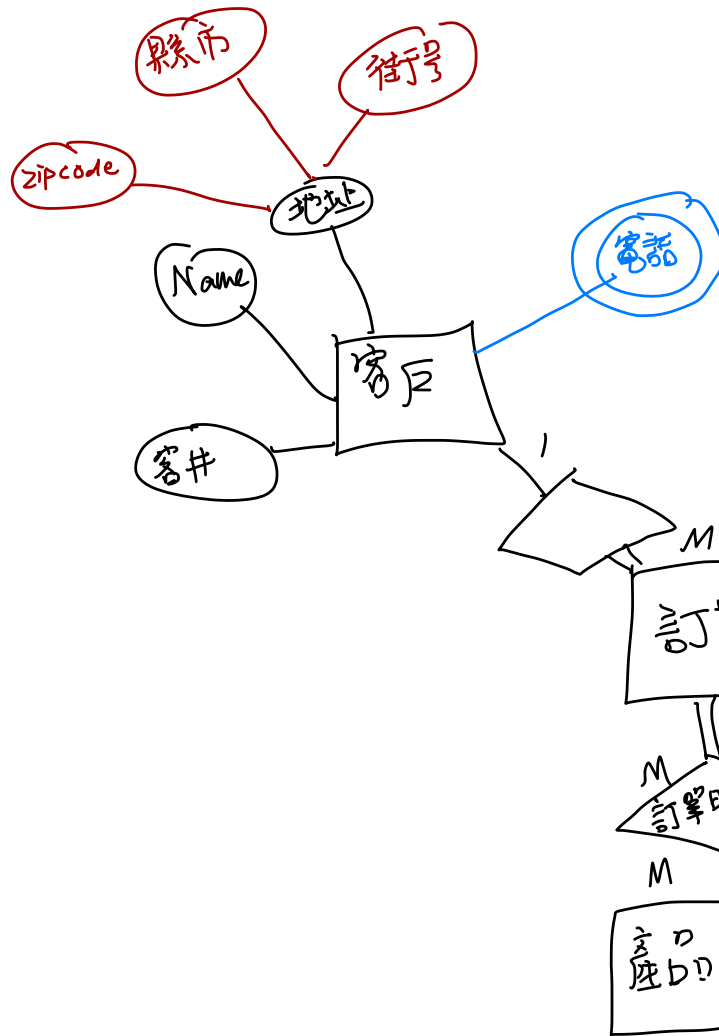
# E/R 圖之(符號)彙總

# 實體種類集合

- 一般(強)實體(集合)(Regular or strong entity set)
- 弱實體(集合)(Weak entity set)
  - 弱實體 與 擁有者實體(用來辨認擁有者)  
*Weak entity VS Owner entity (Identifying owner)*

# 屬性(attribute)種類

- 簡單屬性(欄位)(Simple or atomic) vs. 複合式屬性(欄位)(composite)
- 單值屬性(欄位)(Single-valued )vs. 多(重)值屬性(欄位)(multi-valued)
- 內儲屬性(欄位)(Stored) vs. 衍生式屬性(欄位)(derived)
- 空值(虛值)(NULL value) , 鍵值(Key attribute)
- 定義域(Domain): 屬性(欄位)值的範圍



多 = 複合式屬性

多 = 多(重)值屬性

多行衍生性屬性(虛線)

Total Price

↑  
單價 × 數量  
是衍生出來的

# ER圖必要元素

- 實體(Entities)
- 關係(Relationship)
- 屬性 (Attributes)
- 結構性限制(Structural Constraints)
  - 參與比率(或基數比)(Cardinality Ratio) : 1:1, 1:N, , N:1, M:N
  - 參與限制(Participation Constraint)
    - 完全參與(Total participation) : 每一筆資料在對應的關係實體上一定有對應的資料
    - 部分參與(Partial participation) : 每一筆資料在對應的關係實體上不一定有對應的資料
  - UML以multiplicity表示結構性限制條件

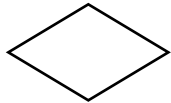
# E/R關係圖的符號



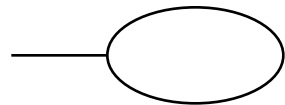
一般實體



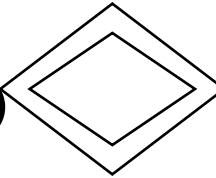
弱實體



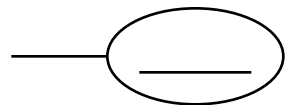
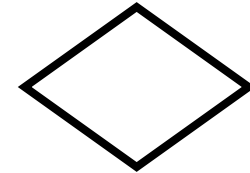
關係



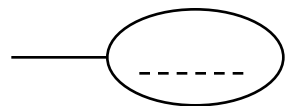
簡單屬性(Simple)



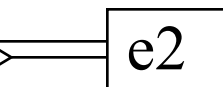
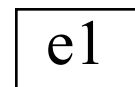
識別關係



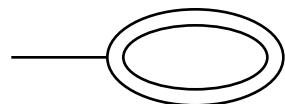
索引屬性(Key) *不是 Primary key*



部分鍵(partial key)

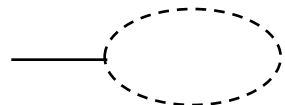


e2完全參與 關係R



多重值屬性(multi-valued)

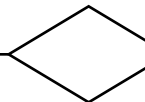
or



衍生式屬性(derived)



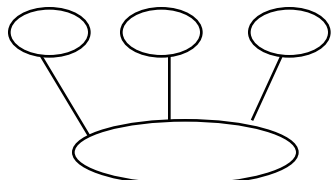
1



N



參與比率1:N(Cardinality ratio)



複合屬性(欄位)(Composite)

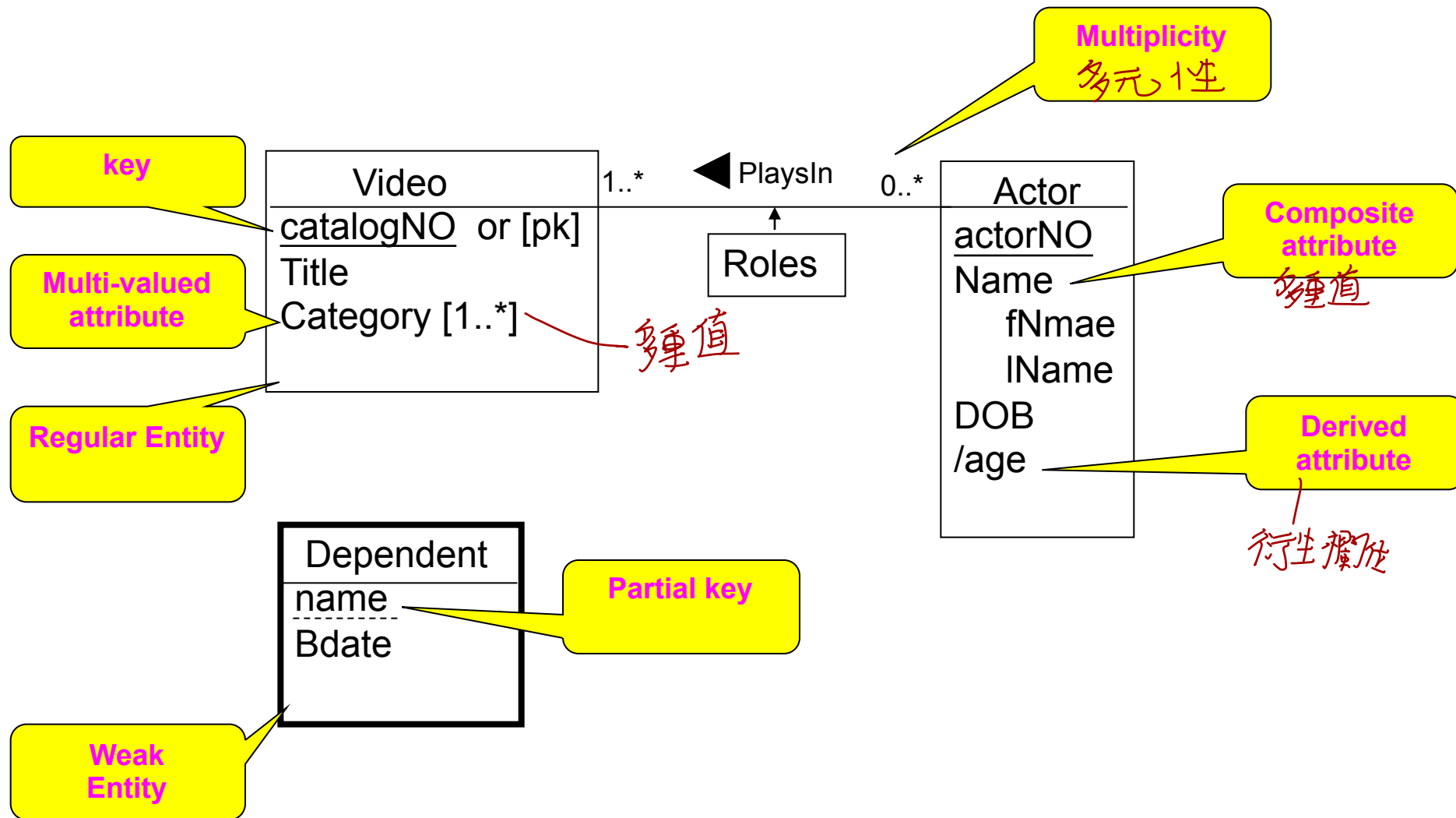


(min,max)

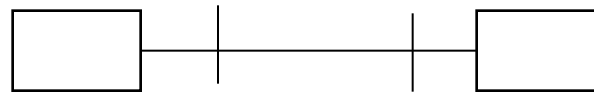
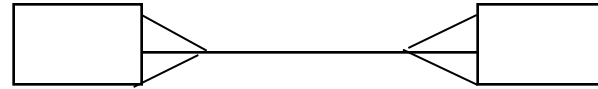
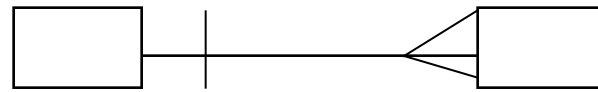
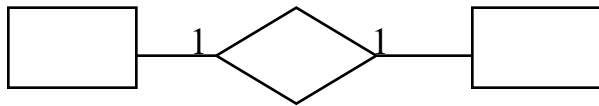
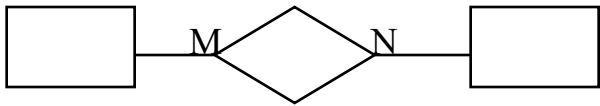
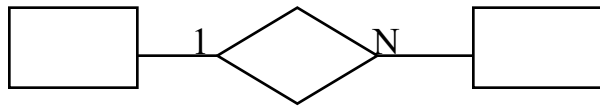
結構性限制



# E/R關係圖的符號—UML

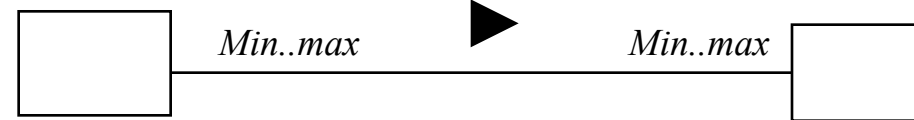
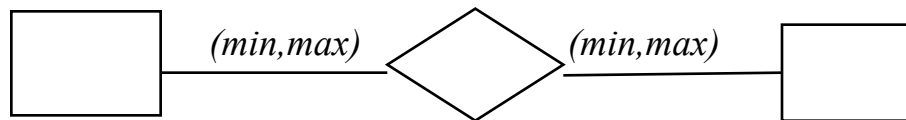


# 不同的參與比率表示法

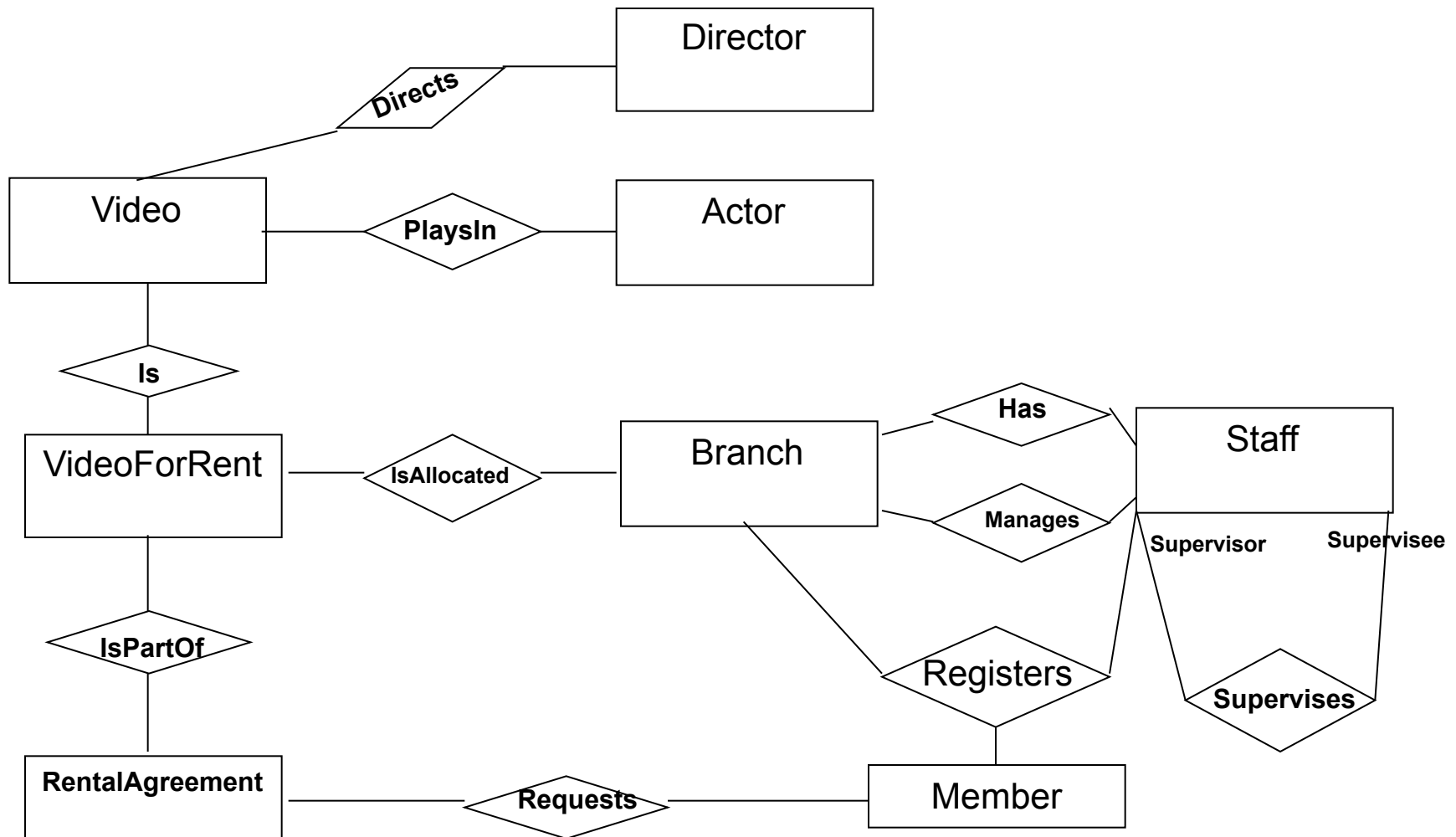


or

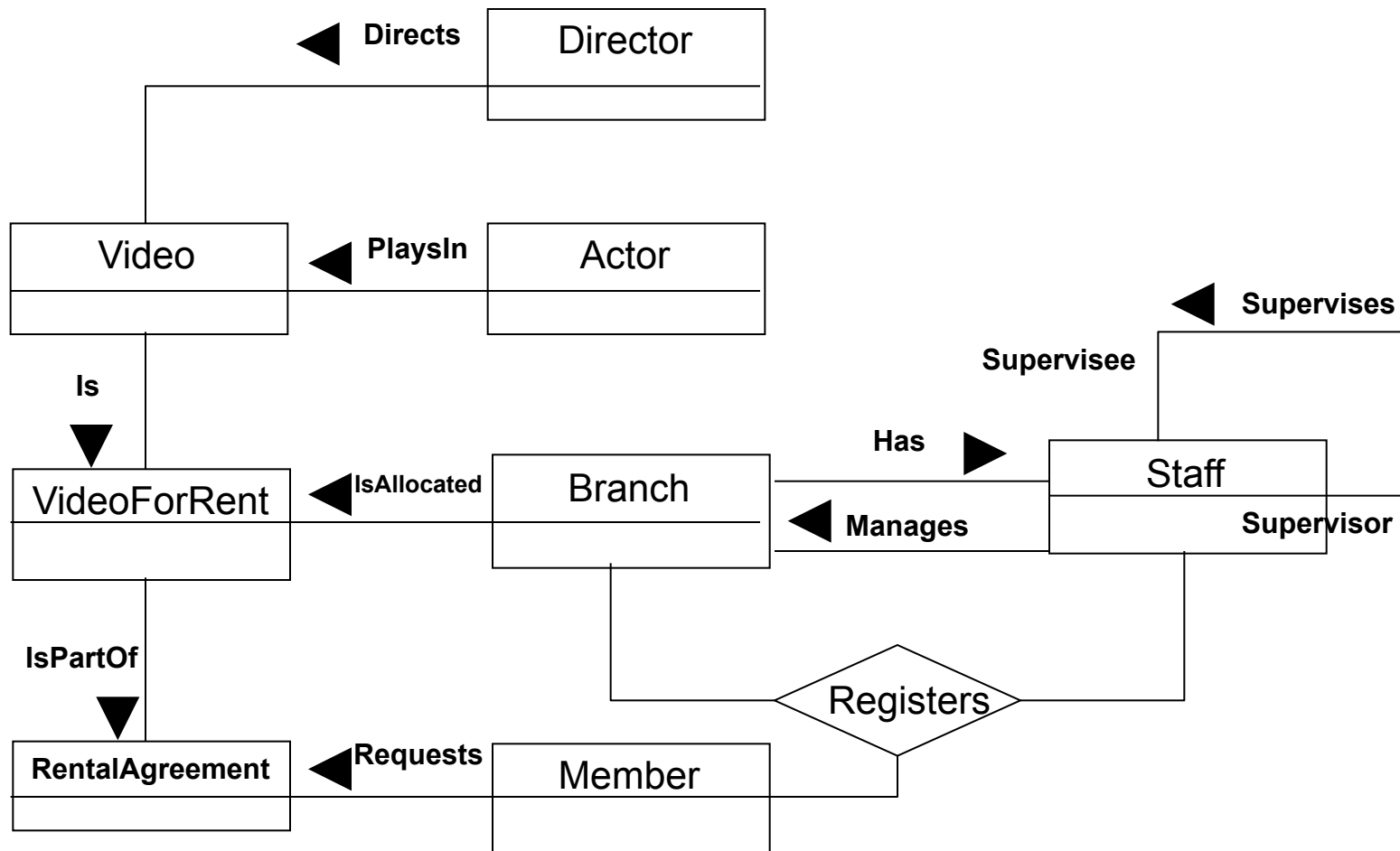
FOR UML



# StayHome 個案E/R 圖的初稿



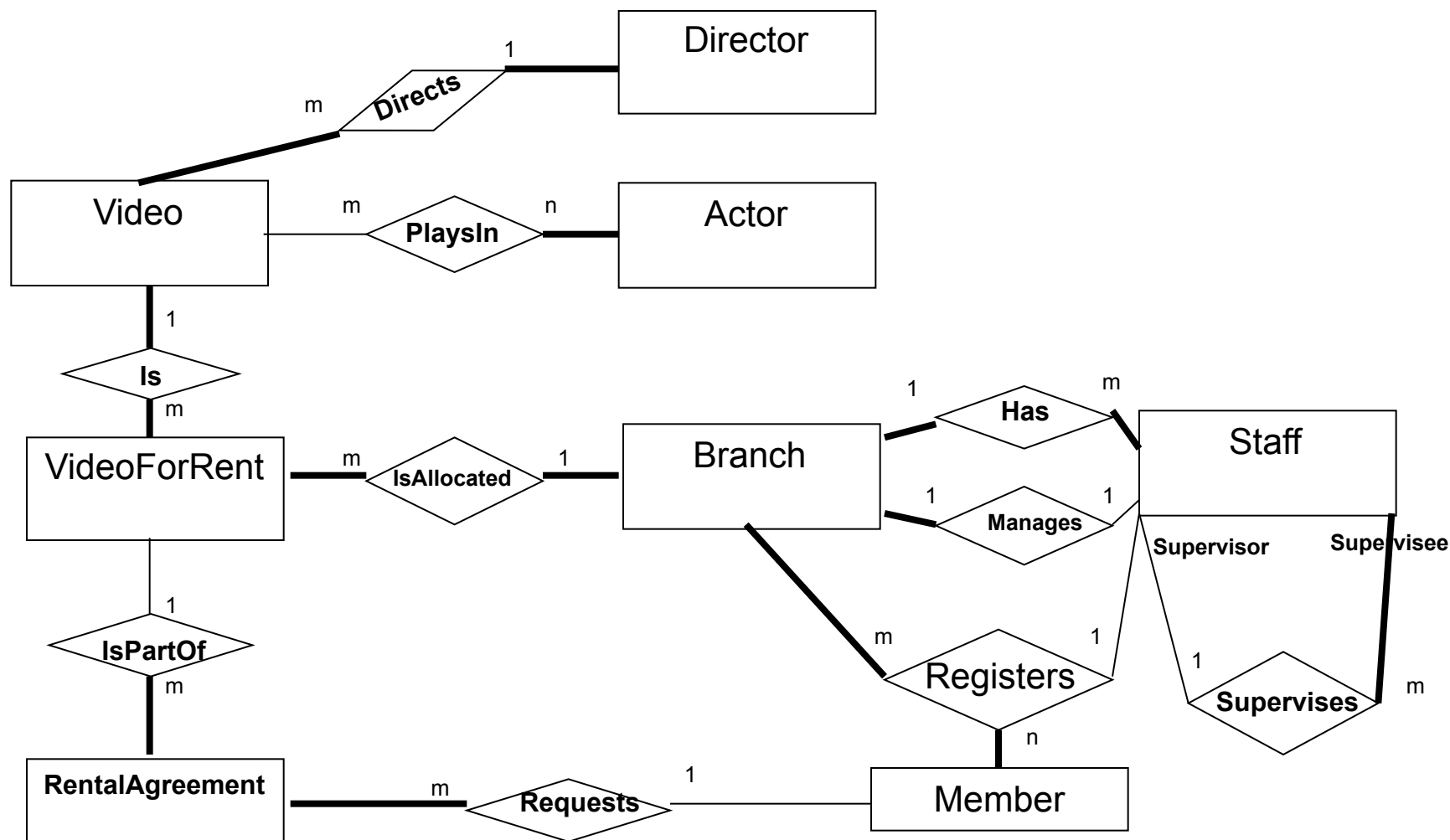
# StayHome 個案E/R 圖的初稿—UML



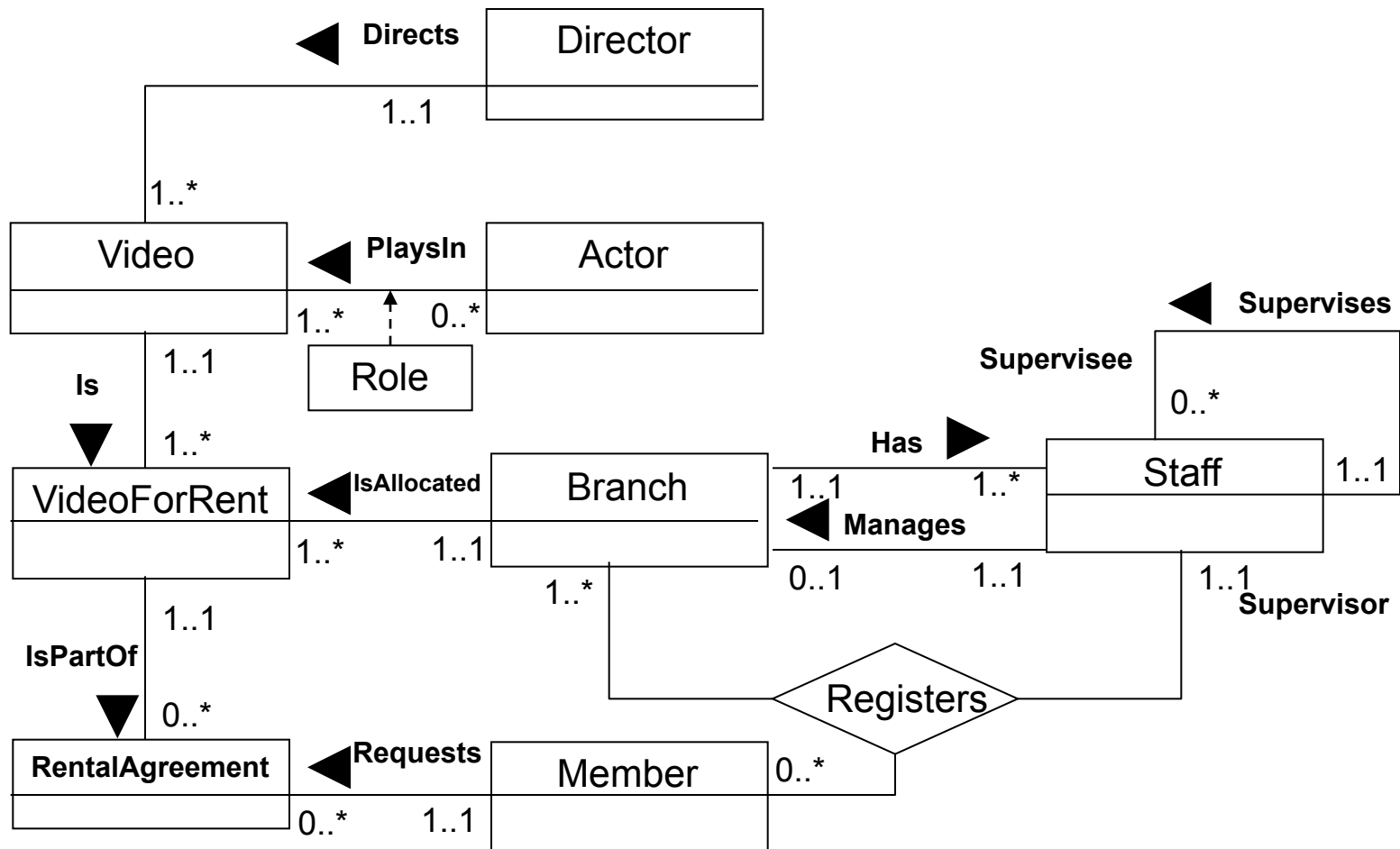
# StayHome個案的結構性限制條件 (structural constraints or Multiplicity)

| Entity          | Multiplicity | Relationship | Entity          | Multiplicity |
|-----------------|--------------|--------------|-----------------|--------------|
| Branch          | 1..*         | Has          | Staff           | 1..1         |
|                 | 1..*         | IsAllocated  | VideoForRent    | 1..1         |
| Branch, Staff * | 1..*, 1..1   | Registers    | Member          | 0..*         |
| Staff           | 0..1         | Manages      | Branch          | 1..1         |
|                 | 0..*         | Supervises   | Staff           | 1..1         |
| Video           | 1..*         | Is           | VideoForRent    | 1..1         |
| VideoForRent    | 0..*         | IsPartOf     | RentalAgreement | 1..1         |
| Member          | 0..*         | Request      | RentalAgreement | 1..1         |
| Actor           | 1..*         | PlaysIn      | Video           | 0..*         |
| Director        | 1..*         | Directs      | Video           | 1..1         |

# 加入結構性限制條件到E/R圖



# 加入結構性限制條件到E/R圖—UML



[Return to step 1 menu](#)

## Step 1.3 找出相關實體或關係之屬性

|                 |                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------|
| Branch          | BranchNo, address ( <b>composite</b> : street, city, state, zipCode), tel( <b>multi-valued</b> ) |
| Staff           | staffNo, name, position, salary                                                                  |
| Video           | catalogNo, title, category, dailyRental, price                                                   |
| Director        | DirectorNo, DirectorName                                                                         |
| Actor           | ActorNo, ActorName                                                                               |
| Member          | memberNo, name ( <b>composite</b> : fName, lName), address                                       |
| RentalAgreement | rentalNo, dateOut, DateReturn                                                                    |
| VideoForRent    | VideoNo, available                                                                               |

[Return to step 1 menu](#)

# Step 1.4 決定屬性(屬性(欄位))定義域(有效範圍)

| Entity | Attributes     | Description | Data type and length   | Nulls | 多重值<br>Multivalued | ... |
|--------|----------------|-------------|------------------------|-------|--------------------|-----|
| Branch | branchNo       | 唯一值         | 4 fixed characters     | No    | No                 |     |
|        | Address:       | 分公司地址       |                        |       |                    |     |
|        | Street         | .....       | 30 variable characters | ..... | .....              |     |
|        | City...<br>tel | .....       | .....                  | ..... | .....              |     |
| Staff  | staffNo        | 唯一值         | .....                  |       |                    |     |
|        | Name           | 員工姓名        | .....                  |       |                    |     |

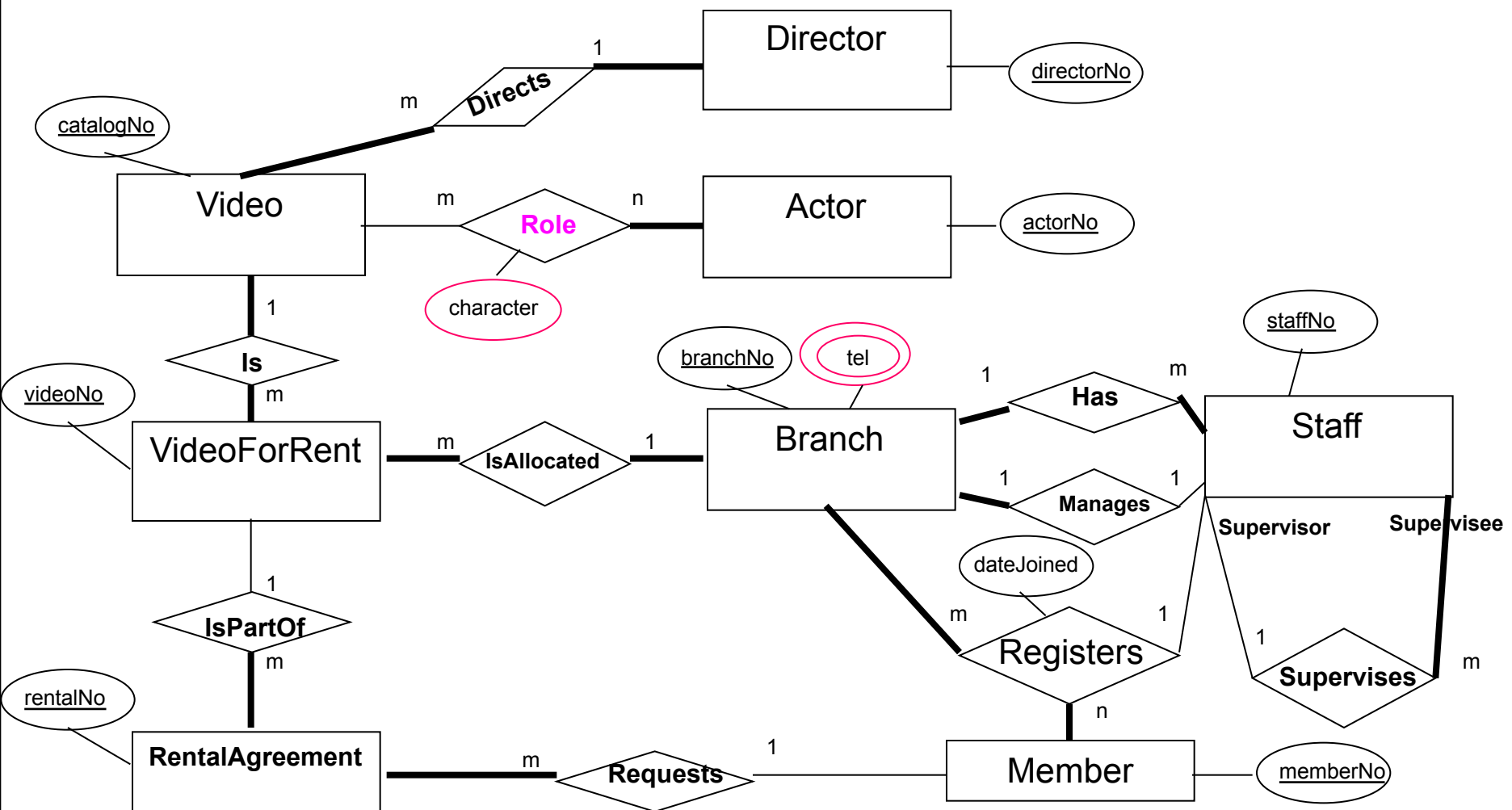
[Return to step 1 menu](#)

# Step 1.5 決定候選鍵(candidate key)與主索引(primary key)

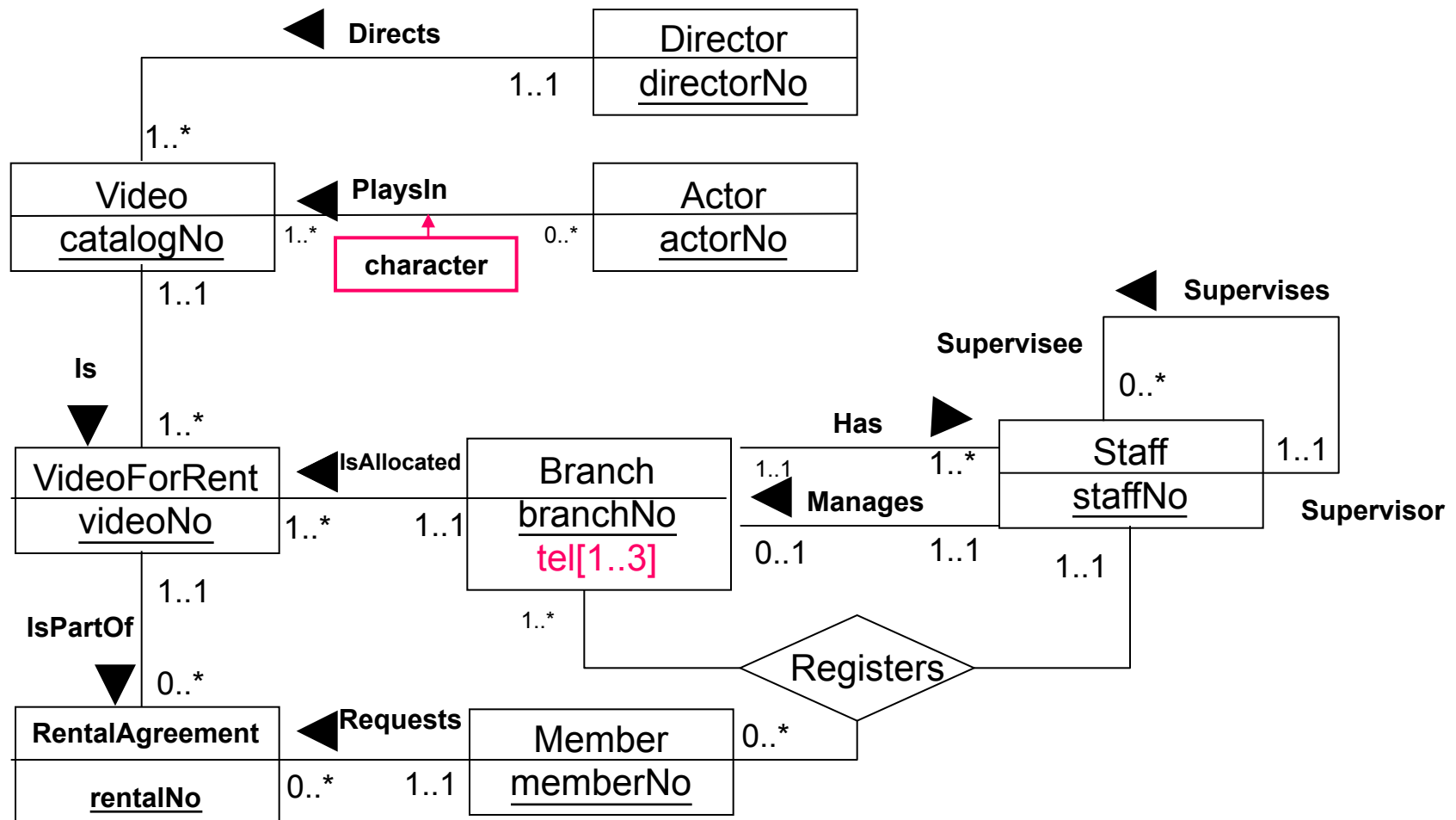
- 決定候選鍵的一般原則
  - 候選鍵應是屬性(欄位)數的最小集合( minimal set of attributes)
  - 候選鍵的值不常變動
  - 候選鍵的唯一性應該不會變
  - 文字型的候選鍵應找字數較少的屬性(欄位)
  - 數字型的候選鍵應找最大值較小的屬性(欄位)
  - 應從使用者角度找最容易使用的屬性(欄位)當候選鍵.
- 再從眾多候選鍵中選一個當作主索引( primary key)
  - 若在E/R圖中某實體可以找出鍵(候選鍵或主索引)，稱此實體為一般(強)實體( regular or strong Entity)反之，若無法找出候選鍵或主索引，稱此實體為弱實體(weak Entity)

2種實體

# StayHome個案之ER 圖中加入Key-1



# StayHome個案之ER 圖中加入Key—UML



[Return to step 1 menu](#)

# Step 1.7 檢查資料模式是否滿足使用者交易處理

- 要保證資料模式可以滿足使用者交易處理
- 可以使用資料模式與資料辭典(data dictionary)，進行人工檢驗
  - 若都可以解決使用者交易處理需求，表示產生的資料模式可以滿足使用者交易處理
  - 若無法解決某些使用者交易處理需求，表示產生的資料模式尚無法滿足使用者交易處理，資料模式必須修正

## Step2 對每一邏輯資料模式建構相對應的資料庫結構(綱要)(database schema)

- [Step 2.1](#) 使用E/R對應規則產生對應之資料表
- [Step 2.2](#) 使用正規化規則檢查資料表結構
- [Step 2.3](#) 檢查資料表是否滿足交易處理需求
- [Step 2.4](#) 定義整合性限制(integrity constraints)
- [Step 2.5](#) 與使用者一起檢查完成的資料模式是否滿足需求(正確)

[Go to step 3](#)

考試記:

ER:

1. RE (Regular Entity) ] Entity

2. WE (Weak Entity) ]

3. 1-1 ] Relationship

4. 1-M ]

5. M-N ]

6. M V (Multiple Value)

7. MR (Multiple Relationship)

# 對應規則(1/2)

- **Rule 1**：對每一個一般實體建立關聯(資料表)，包含所有簡單屬性(欄位)，選擇主索引
- **Rule 2**：對每一個弱實體建立關聯(資料表)，包含所有簡單屬性(欄位)，並包含擁有者實體(owner entity type)之主索引，形成外來鍵(foreign key)，並建立主索引(由部分鍵及上述外來鍵組成)
- **Rule 3**：對每一1:1關係，一般模擬成1:N關係解決；或
  - 若參與此關係之二實體均為完全參與，則可合併之
- **Rule 4**：每一1:N關係，將1方實體之主索引納入多方實體內，形成外來鍵(**關係上之屬性，亦跟隨放入多方的資料表中**)

example

## 對應規則(2/2)

- **Rule 5**：為每一N:M關係建立一資料表，包含本身之簡單屬性(欄位)，並將參與實體之主索引納入資料表中，形成外來鍵，並合併為主索引
- **Rule 6**：為多重值建一資料表，包含此多重值與原實體之主索引(形成外來鍵)，並合併為主索引
- **Rule 7**：為多元關係建資料表，包含所有參與實體之主索引，(皆為外來鍵)，結合形成主索引
  - 或無解，必須結合程式，避免不符合條件資料納入
  - 若參與此關係之實體為1，則此外來鍵可以不納入主索引的一部分

Rule 1: 學生(學號, 姓氏, 名字, 通訊郵遞區號, 通訊縣市, 通訊街道, 通訊電話, 永久郵遞區號, 永久通訊縣市, 永久通訊街道, 永久通訊電話, 性別, 年級, 主修代號, 副修代號, 科系(科系代號, 科系名稱, 系編號, 院別, 電話, 科目(科目編號, 科目名稱, 科目介紹, 時數, 科系代號))

key

FK

FK

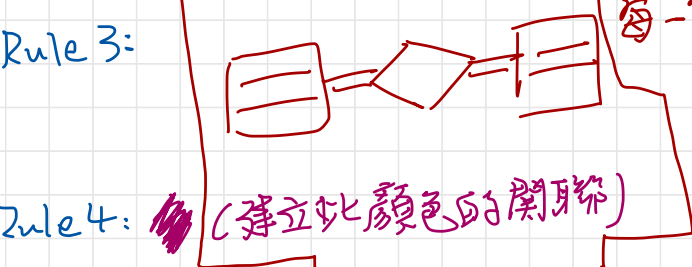
FK

Rule 2: 課程(老師, 科目編號, 年度, 學期, 課號)

Foreign Key

外來

每一筆都有對應, 乾脆合併



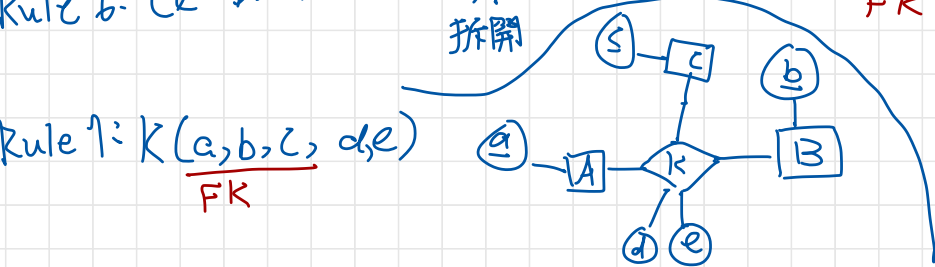
Rule 5: 成績(學號, 科目編號, 數值成績, 文字成績)

FK

FK

Rule 6: (如以科系電話為例) 科系電話(科系代號, 電話)

FK



# StayHome 公司依Branch view 產生的 資料表

|                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Actor ( <a href="#">actorNo</a> , actorname)                                                                                                                                                                                                                                       | Branch ( <a href="#">branchNo</a> , street, city, state, zipCode, <a href="#">mgrStaffNo</a> )<br><a href="#">Alternate key zipCode</a><br>Foreign key mgrStaffNo references Staff(staffNo)                                                                                   |
| Director ( <a href="#">directorNo</a> , directorName)                                                                                                                                                                                                                              | Member( <a href="#">memberNo</a> , fName, lName, address)                                                                                                                                                                                                                     |
| Registration( <a href="#">branchNo</a> , <a href="#">memberNo</a> , <a href="#">staffNo</a> , dataJoined)<br><a href="#">Foreign key</a> branchNo references Branch(branchNo)<br>Foreign key memberNo references Member(memberNo)<br>Foreign key staffNo references Staff(staffNo) | RentalAgreement ( <a href="#">rentalNo</a> , dateOut, dateReturn, <a href="#">memberNo</a> , <a href="#">videoNo</a> )<br>Alternate key (memberNo, videoNo , dateOut)<br>Foreign key memberNo references Member(memberNo)<br>Foreign key videoNo references videoFor(videoNo) |
| Role ( <a href="#">catalogNo</a> , <a href="#">actorNo</a> , character)<br>Foreign key catalogNo references Video(catalogNo)<br>Foreign key actorNo references Actor(actorNo)                                                                                                      | Staff ( <a href="#">staffNo</a> , name, position, salary, <a href="#">branchNo</a> , <a href="#">supervisorStaffNo</a> )<br>Foreign key branchNo references Branch(branchNo)<br>Foreign key supervisorStaffNo references Staff(staffNo)                                       |
| Tel ( <a href="#">telNo</a> , <a href="#">branchNo</a> )<br>Foreign key branchNo references Branch(branchNo)                                                                                                                                                                       | Video ( <a href="#">catalogNo</a> , title, category, dailyRental, price, <a href="#">directorNo</a> )<br>Foreign key directorNo references Director(directorNo)                                                                                                               |
| VideoForRent ( <a href="#">videoNo</a> , available, <a href="#">catalogNo</a> , <a href="#">branchNo</a> )<br>Foreign key catalogNo references Video(catalogNo)<br>Foreign key branchNo references Branch(branchNo)                                                                | <a href="#">Return to step 2 menu</a>                                                                                                                                                                                                                                         |

## Step 2.2 使用正規化檢查每一資料表結構

- 必須保證 step 2.1 所產生的資料表至少應該 3NF；若無法保證 3NF，可能 E/R 轉換或正規化有錯誤，必須修正
  - 我們將在 **schema refinement module** 介紹正規化 (第五章)

Third Normal Form  
三階正規化

[Return to step 2 menu](#)

## Step 2.3 檢查資料表是否滿足使用者 交易需求

- 要保證資料表滿足使用者交易需求

# StayHome 公司維護(新增、修改、刪除)處理 所需之資料表

| 交易                                                                                                                                | 所需資料表                        |
|-----------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| <ul style="list-style-type: none"><li>•Enter the details of a new branch</li><li>•Update/delete the details of a branch</li></ul> | Branch, Tel                  |
| <ul style="list-style-type: none"><li>•Enter, Update/delete Staff</li></ul>                                                       | Staff                        |
| <ul style="list-style-type: none"><li>•Enter, Update/delete Video</li></ul>                                                       | Video, Director, Role, Actor |
| <ul style="list-style-type: none"><li>•Enter, Update/delete Copies of video</li></ul>                                             | VideoForRent                 |
| <ul style="list-style-type: none"><li>•Enter, Update/delete Member</li></ul>                                                      | Member, Registration         |
| <ul style="list-style-type: none"><li>•Enter, Update/delete Rental agreement</li></ul>                                            | RentalAgreement              |

# StayHome 公司查詢處理所需之資料表

| 交易                                                                                | 所需資料表                                          |
|-----------------------------------------------------------------------------------|------------------------------------------------|
| •List the details of branches in a given city                                     | Branch, Tel                                    |
| •List name, position, and salary of staff at a given branch ordered by staff name | Staff                                          |
| •.....                                                                            | .....                                          |
| •List the titles of all videos in a specified category, ordered by title          | Video                                          |
| •List the total cost of the videos at all branches                                | Video, videoForrent                            |
| •.....                                                                            | .....<br><a href="#">Return to step 2 menu</a> |

# Step 2.4 找出整合性限制

- 必須要有資料(Required data)
  - 某些屬性(欄位)一定要填入資料不可有空值(虛值)(null)或且必須限制某些資料，如職稱 job position (manager, supervisor, assistant, buyer)
- 屬性(欄位)定義域(domain constraints) (如字串長度限制)
  - 某一個屬性(欄位)可以填入的值
- 實體(主索引)限制(Entity integrity constraint) ☆ - 一個檔只有一個主索引  
只有clustering key可多重  
叢集索引
  - 主索引不可以為空值(虛值)(null)
- 參考性完整限制(Referential integrity constraints)
  - 資料表間建立關聯後，父子檔間形成參考性完整限制條件，子檔資料相依於父檔該筆資料是否存在而存在
- 企業規則
  - 例如：每次最多租10部

# StayHome 個案的參考性限制

☆參考03

|                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Branch</b><br>Foreign key mgrStaffNo references Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION                                                                                                                                                                                        |
| <b>Registration</b><br>Foreign key branchNo references Branch(branchNo) ON UPDATE CASCADE ON DELETE NO ACTION<br>Foreign key memberNo references Member(memberNo) ON UPDATE CASCADE ON DELETE NO ACTION<br>Foreign key staffNo references Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION |
| <b>RentalAgreement</b><br>Foreign key memberNo references Member(memberNo) ON UPDATE CASCADE ON DELETE NO ACTION<br>Foreign key videoNo references videoFor(videoNo) ON UPDATE CASCADE ON DELETE NO ACTION                                                                                     |
| <b>Role</b><br>Foreign key catalogNo references Video(catalogNo) ON UPDATE CASCADE ON DELETE CASCADE<br>Foreign key actorNo references Actor(actorNo) ON UPDATE CASCADE ON DELETE NO ACTION                                                                                                    |
| <b>Staff</b><br>Foreign key branchNo references Branch(branchNo) ON UPDATE CASCADE ON DELETE NO ACTION<br>Foreign key supervisorStaffNo references Staff(staffNo) ON UPDATE CASCADE ON DELETE SET NULL                                                                                         |
| <b>Tel</b><br>Foreign key branchNo references Branch(branchNo) ON UPDATE CASCADE ON DELETE CASCADE                                                                                                                                                                                             |
| <b>Video</b><br>Foreign key directorNo references Director(directorNo) ON UPDATE CASCADE ON DELETE NO ACTION                                                                                                                                                                                   |
| <b>VideoForRent</b><br>Foreign key catalogNo references Video(catalogNo) ON UPDATE CASCADE ON DELETE NO ACTION<br>Foreign key branchNo references Branch(branchNo) ON UPDATE CASCADE ON DELETE NO ACTION                                                                                       |

No Action = 限制

[Return to step 2 menu](#)

## Step 2.5 與使用者一起討論所設計之 邏輯資料模式

- 要保證所設計之邏輯資料模式即為使用者之需求觀點

[Return to step 2 menu](#)

## Step 3 建立整體性邏輯資料模式 (global logical data model)

- 當系統較複雜時可以先將部分資料需求觀點(Manager、Supervisor、Assistant)等使用者觀點整合為 **Branch view** (已完成)，並即將Director、Buyer等使用者觀點整合為 **Business view**
  - 接著才進行整體性邏輯資料模式的建構

# StayHome個案的 **Business view** 的使用者需求規格

- 資料方面的需求 (Data requirements)
  - 描述 Branch view所需資料的情形
- 交易處理方面的需求(Transaction requirements)
  - 描述Business view 交易處理所需的功能、資料

# 資料需求

## Data requirements

- Branch (branch\_no, address, tel)
  - Branch( 1 Manager, 1 or more Supervisors, a number of staff)
- Staff (staff\_no , name, position, salary)
- Video (video\_no, catalog\_no, title, category (action, adult, children, thriller, horror, sci-fi), daily rental rate, purchase price)
- Supplier (supplier\_no, name, address, tel, status)
- Order (order\_no, supplier\_no, supplier\_address, video\_catalog\_no, video\_title, video\_purchase\_price, quantity, date\_order\_placed, date\_order\_received, address of branch receiving the order)
- Member (member\_no, name, address, register\_date,)
- Video\_rented (rental\_no, member's full name, member\_no, video\_no, title, daily rental rate, date\_rented, date\_return)

# 交易處理功能需求

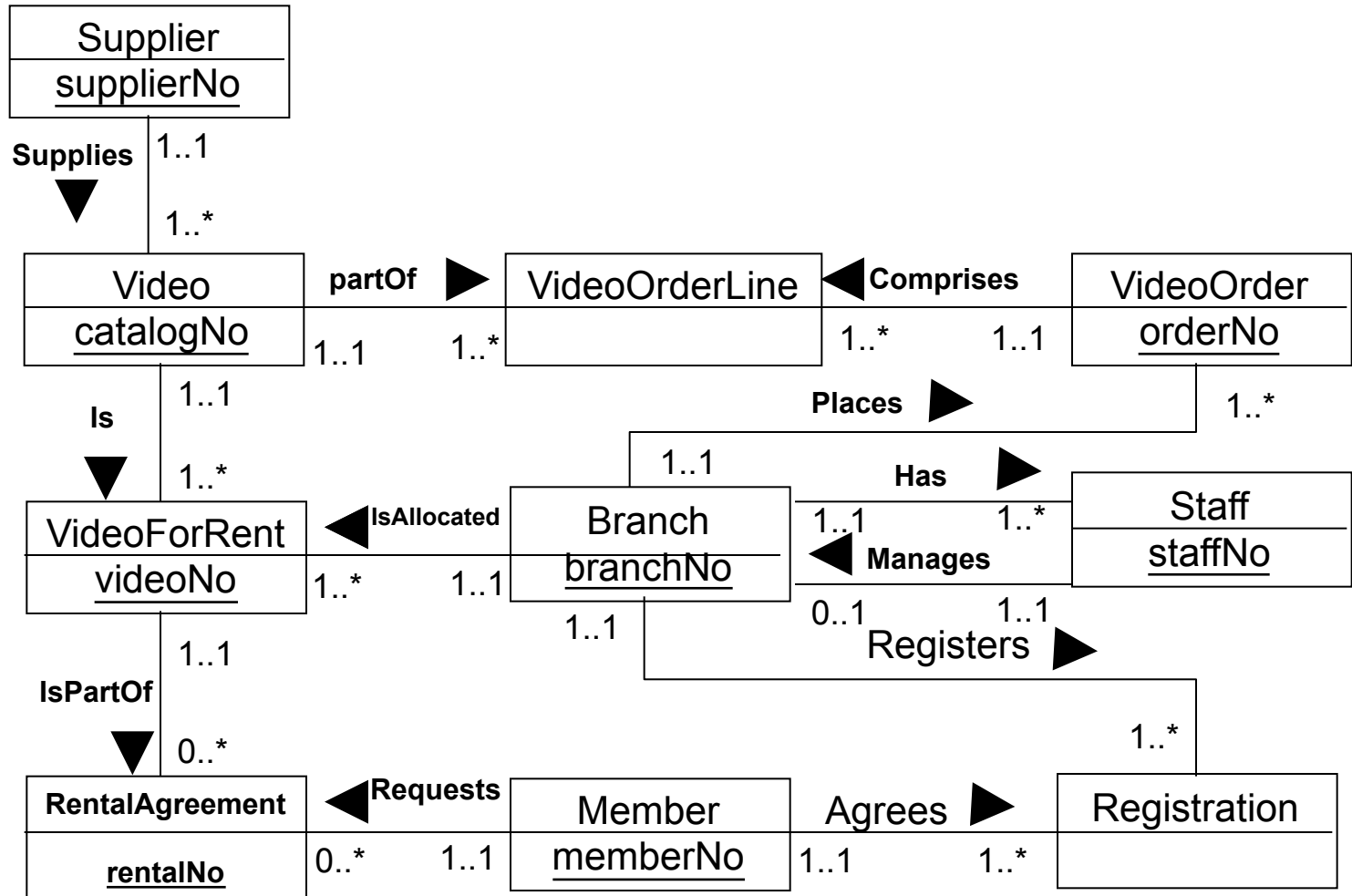
## Transaction requirements

- 資料**輸入**(Data entry)
  - 輸入新錄影帶、新供應商、新訂單資料
- 資料**更新與刪除**(Data update/deletion)
  - 更新與刪除錄影帶、供應商、訂單資料
- 資料**查詢**(Data queries)
  - 找出某城市所有分公司的明細資料
  - 找出所有訂單明細資料
  - 找出所有供應商名資料
  - .....
- **報表**

# StayHome個案Business view的區域 性邏輯資料模式

- E/R 模式(E/R model)
- 對應的資料表

# StayHome個案公司 Business view 的 E/R圖



Quiz: Schema for Business view

# StayHome個案公司 Business view所 對應之資料表

|                                                                                                                                                                               |                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Branch</b> ( <u>branchNo</u> , address, tel, mgrStaffNo)<br>Alternate key tel<br>Foreign key mgrStaffNo references Staff(staffNo)                                          | <b>Member</b> ( <u>memberNo</u> , name, address)                                                                                                                                                                                     |
| <b>Registration</b> ( <u>branchNo</u> , <u>memberNo</u> , dataJoined)<br>Foreign key branchNo references Branch(branchNo)<br>Foreign key memberNo references Member(memberNo) | <b>RentalAgreement</b> ( <u>rentalNo</u> , dateOut, dateReturn, memberNo, videoNo)<br>Alternate key memberNo, videoNo , dateOut<br>Foreign key memberNo references Member(memberNo)<br>Foreign key videoNo references Video(videoNo) |
| <b>Staff</b> ( <u>staffNo</u> , name, position, salary, branchNo)<br>Foreign key branchNo references Branch(branchNo)                                                         | <b>Supplier</b> ( <u>supplierNo</u> , sName, sAddress, sTel, status)<br>Alternate key sTel                                                                                                                                           |
| <b>Video</b> ( <u>catalogNo</u> , title, category, dailyRental, price, supplierNo)<br>Foreign key supplierNo references Supplier(supplierNo)                                  | <b>VideoForRent</b> ( <u>videoNo</u> , available, catalogNo, branchNo)<br>Foreign key catalogNo references Video(catalogNo)<br>Foreign key branchNo references Branch(branchNo)                                                      |
| <b>VideoOrder</b> ( <u>orderNo</u> , dateOrdered, dateReceived, branchNo)<br>Foreign key branchNo references Branch(branchNo)                                                 | <b>VideoOrderLine</b> ( <u>orderNo</u> , <u>catalogNo</u> , quantity)<br>Foreign key orderNo references VideoOrder(orderNo)<br>Foreign key catalogNo references Video(catalogNo)                                                     |

# Step 3 建構整體性(global)邏輯資料模式

- 將各區域性邏輯資料模式整合為整體性(global)邏輯資料模式
  - [Step 3.1](#)  
將各區域性邏輯資料模式合併為整體性邏輯資料模式
  - [Step 3.2](#)檢查整體性邏輯資料模式
  - [Step 3.3](#)檢查系統是否滿足未來的成長
  - [Step 3.4](#)檢查整體性邏輯資料模式是否滿足使用者需求

[Next is physical design](#)

# Step 3.1 將各區域性邏輯資料模式合併為整體性邏輯資料模式

- 合併各區域性邏輯資料模式，並解決模式間的不一致性
  - 檢查各實體(資料表)的名稱及主索引
  - 檢查各關係的名稱
  - 合併各資料模式的資料表
  - 包含在某資料模式單獨出現的實體
  - 合併各資料模式的關係
  - 包含在某資料模式單獨出現的關係
  - 檢查是否尚有遺漏的實體/資料表、關係
  - 檢查外來鍵
  - 檢查整合性限制、型態、定義域、限制等衝突
  - 繪出整體性邏輯資料模式圖
  - Update the documentation更新文件

[Return to step 3](#)

# 檢查各實體(資料表)的名稱及主索引

- 是否有同樣名稱但代表不同意義的實體(同名異義，**homonyms**)
  - 同樣是Part，但卻是**電腦零件** VS **家具零件**
- 是否有不同名稱但代表相同意義的實體(異名同義，**synonyms**)
  - Customer vs. Client

# 比較Branch views及Business views 實體(資料表)的名稱及主索引

| Branch view     |                           | Business view         |                                    |
|-----------------|---------------------------|-----------------------|------------------------------------|
| Table           | Primary key               | Table                 | Primary key                        |
| Branch          | branchNo                  | Branch                | branchNo                           |
| Staff           | staffNo                   | Staff                 | staffNo                            |
| <b>Tel</b>      | <b>Tel</b>                |                       |                                    |
| Video           | catalogNo                 | Video                 | catalogNo                          |
| VideoForrent    | videoNo                   | VideoForrent          | videoNo                            |
|                 |                           | <b>Supplier</b>       | <b>supplierNo</b>                  |
|                 |                           | <b>VideoOrder</b>     | <b>orderNo</b>                     |
|                 |                           | <b>VideoOrderLine</b> | <b>orderNo, catalogNo</b>          |
| RentalAgreement | rentalNo                  | RentalAgreement       | rentalNo                           |
| Member          | memberNo                  | Member                | memberNo                           |
| Registration    | branchNo, memberNo        | Registration          | branchNo, memberNo                 |
| <b>Actor</b>    | <b>actorNo</b>            |                       |                                    |
| <b>Role</b>     | <b>catalogNo, actorNo</b> |                       |                                    |
| <b>Director</b> | <b>directorNo</b>         |                       | <a href="#">Return to step 3.1</a> |

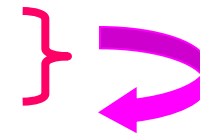
# 檢查各關係的名稱

| Branch view                                 |                                                    |                                               | Business view                                         |                                                       |                                                           |
|---------------------------------------------|----------------------------------------------------|-----------------------------------------------|-------------------------------------------------------|-------------------------------------------------------|-----------------------------------------------------------|
| Entity                                      | Relationship                                       | Entity                                        | Entity                                                | Relationship                                          | Entity                                                    |
| Branch<br>Branch<br><b>Branch</b><br>Branch | Has<br>IsAllocated<br><b>Provides</b><br>Registers | Staff<br>VideoForRent<br><b>Tel</b><br>Member | Branch<br>Branch<br><br>Branch                        | Has<br>IsAllocated<br><br>Registers                   | Staff<br>VideoForRent<br><br>Member                       |
|                                             |                                                    |                                               | <b>Branch</b><br><b>VideoOrder</b><br><b>Supplier</b> | <b>Places</b><br><b>Comprises</b><br><b>Suppliers</b> | <b>VideoOrder</b><br><b>VideoOrdeLine</b><br><b>Video</b> |
| Staff<br><b>Staff</b><br><b>Staff</b>       | Manages<br><b>Supervises</b><br><b>Processes</b>   | Branch<br><b>Staff</b><br><b>Registration</b> | Staff                                                 | Manages                                               | Branch                                                    |
| Video<br><b>Video</b>                       | Is<br><b>Features</b>                              | VideoForRent<br><b>Role</b>                   | Video<br><b>Video</b>                                 | Is<br><b>PartOf</b>                                   | VideoForRent<br><b>VideoOrdeLine</b>                      |
| VideoForRent                                | IsPartOf                                           | RentalAgreement                               | VideoForRent                                          | IsPartOf                                              | RentalAgreement                                           |
| Member<br>Member                            | Agrees<br>Request                                  | Registration<br>RentalAgreement               | Member<br>Member                                      | Agrees<br>Request                                     | Registration<br>RentalAgreement                           |
| <b>Actor</b>                                | <b>Plays</b>                                       | <b>Role</b>                                   |                                                       |                                                       |                                                           |
| <b>Director</b>                             | <b>Directs</b>                                     | <b>Video</b>                                  |                                                       |                                                       |                                                           |

[Return to step 3.1](#)

# 合併各區域資料模式的資料表

- 將名稱相同且主索引相同的實體(資料表)合併
  - 合併的實體(資料表)必須包含兩實體(資料表)所有的屬性(屬性(欄位))，相同屬性(欄位)只保留一份
  - Member( memberNo, fName, lName, address)  
Member( memberNo, name, address)
  - Member( memberNo, fName, lName, address)
- 將名稱相同但主索引不同的實體(資料表)合併
  - 選擇其中一個主索引當合併後的主索引，另一個當候選鍵(或次主索引)
- 將名稱不同，主索引相同或不同的實體(資料表)合併
  - 必須確定此二實體(資料表)代表相同意義的實體(資料表)
  - 利用其內容，尤其是主索引，以確定此二實體(資料表)代表相同意義的實體(資料表)
  - 利用其關係，以確定此二實體(資料表)代表相同意義的實體(資料表)



**Global logical  
data model**

[Return to step 3.1](#)

# 包含在某資料模式單獨出現的實體

- Branch view : Tel, Role, Actor, and Director
- Business view : Supplier, VideoOrder, VideoOrderLine

[Return to step 3.1](#)

# 合併各資料模式的關係

- 檢查所有資料模式中關係的名稱與意義，解決一些不一致性的問題，例如結構性限制條件
  - 將相同實體(資料表)間相同關係名稱與相同意義的關係納入ER中，再納入不同關係名稱但相同意義的關係
    - Branch Has Staff 此關係在兩個資料模式均出現

[Return to step 3.1](#)

# 包含在某資料模式單獨出現的關係

- 單獨出現在 Branch view的關係有：  
Branch **Provides** Tel, Staff **Supervises** Staff,  
Staff **Processes** Registration, Video **Features** role,  
Actor **Plays** Role, Director **directs** Video
- 單獨出現在 Business view的關係有：  
Branch **Places** VideoOrder, Video **Comprises**  
VideoOrderLine, Supplier **Supplies** Video, and  
Video **PartOf** VideoOrderLine

[Return to step 3.1](#)

# 檢查外來鍵

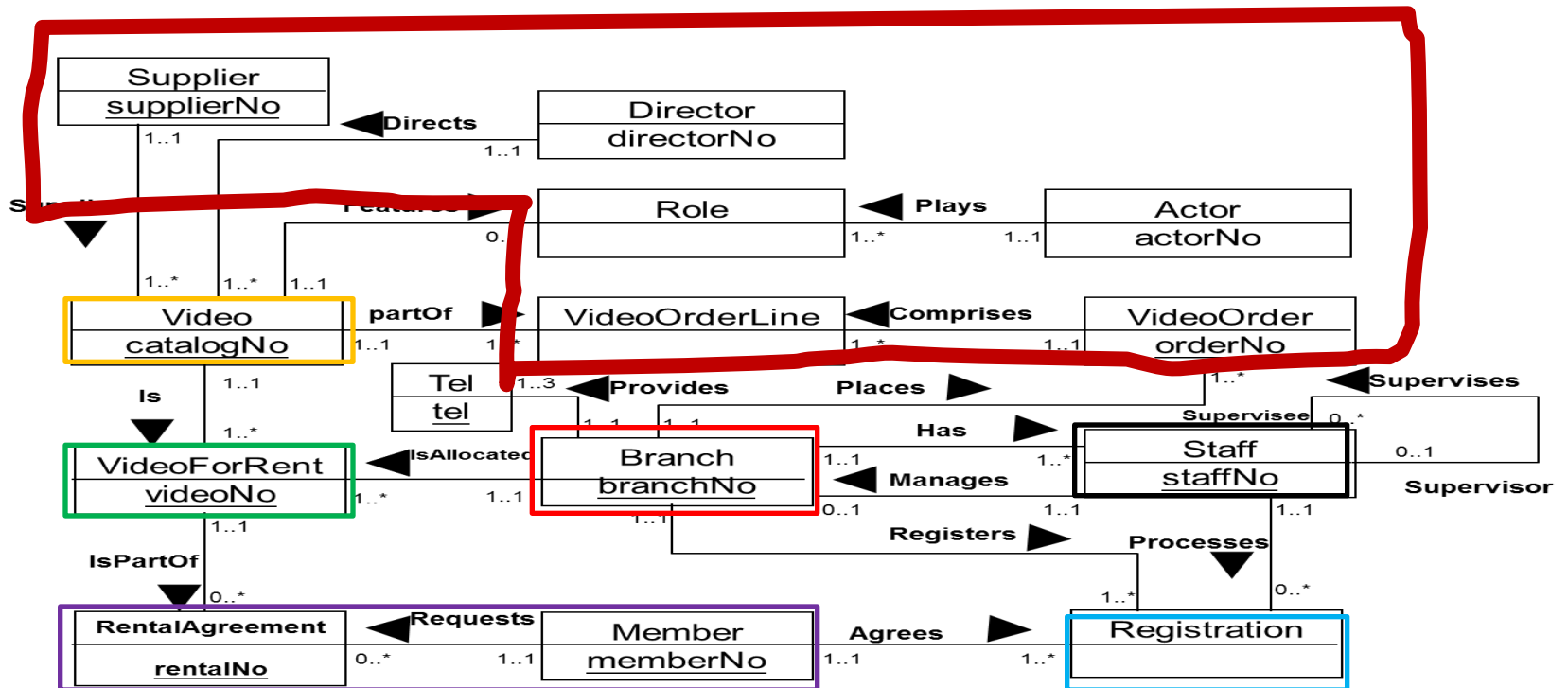
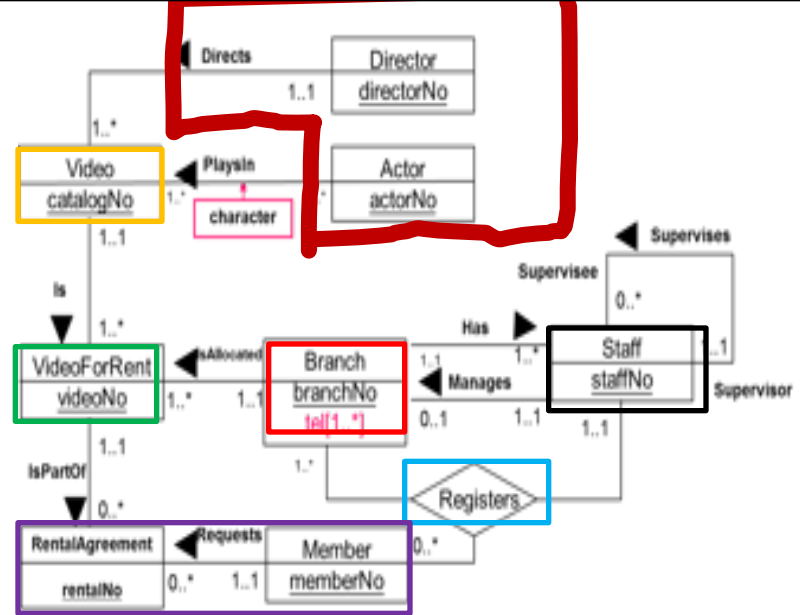
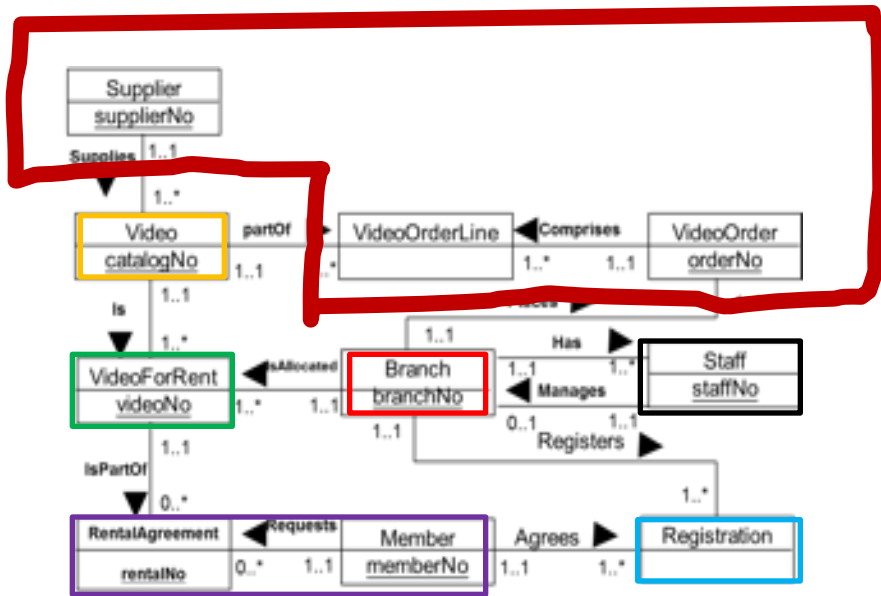
- 此階段由於可能改變實體(資料表)、及其間之關係，所以，參考性完整限制也可能跟著改變，所以，必須進行必要的修正

[Return to step 3.1](#)

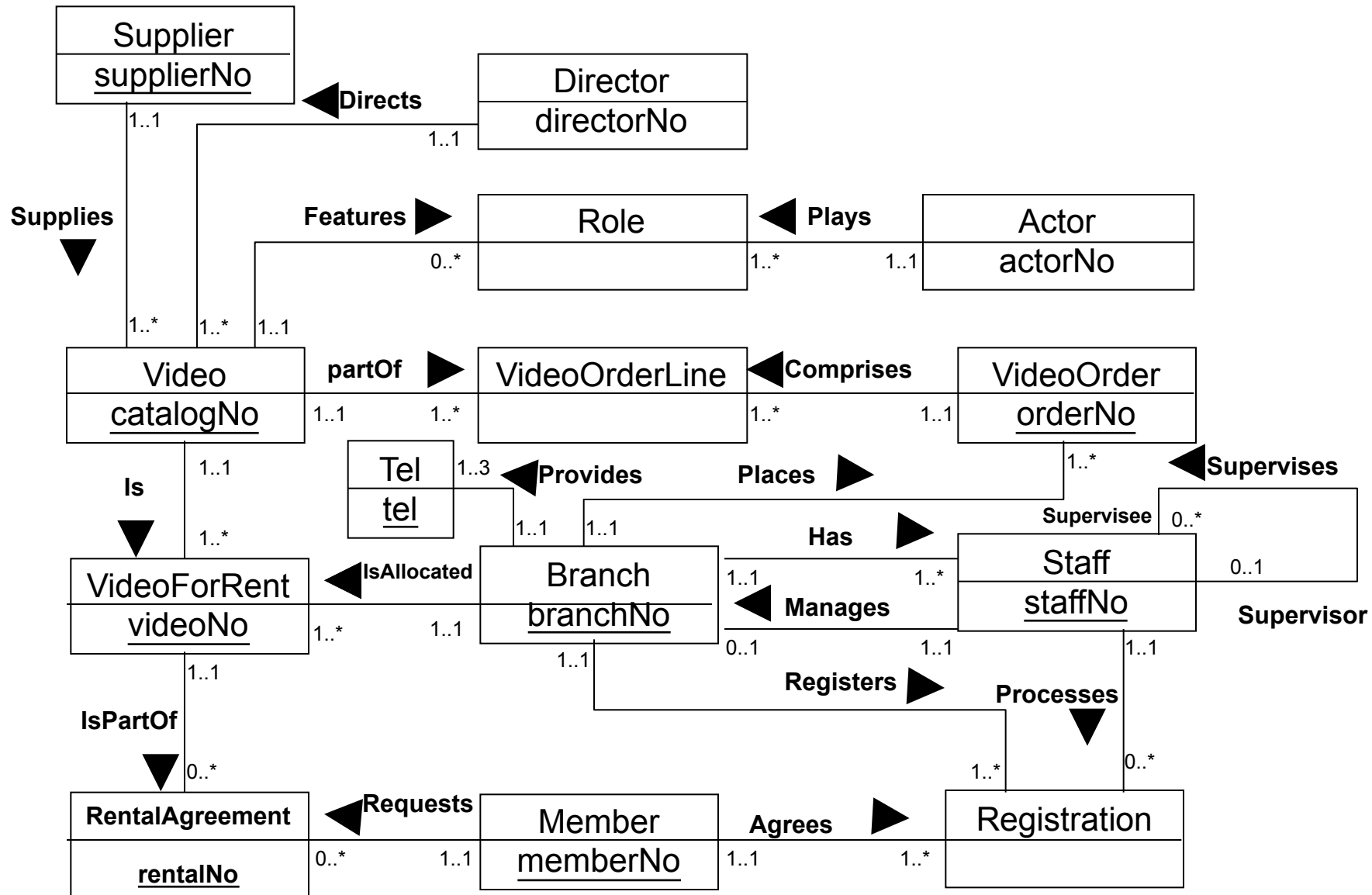
# 檢查整合性限制

## 型態衝突、定義域、限制衝突

- 檢查整體資料模式的**整合性限制**，不能和原始各區域性資料模式相衝突，若發生衝突，則必須和使用者溝通解決
- 型態衝突
  - DEPARTMENT可能是實體 VS 屬性
- 定義域衝突
  - SSN：整數、字串；Weight：磅、公斤
- 參予限制衝突
  - TEACHES在某綱要1:N(一門課一位老師)；在另一綱要M:N(一門課超過一位老師) [Return to step 3.1](#)



# 繪出整體性邏輯資料模式圖



HW: schema for Global view

[Return to step 3.1](#)

# Step 3.2 檢查整體性邏輯資料模式

- 檢查由整體性邏輯資料模式建構的關聯資料表符合正規劃並滿足使用者交易功能需求

|                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Actor</b> ( <u>actorNo</u> , actorName)                                                                                                                                                                                                                    | <b>Director</b> ( <u>directorNo</u> , directorName)                                                                                                                                                                                  |
| <b>Role</b> ( <u>catalogNo</u> , <u>actorNo</u> , character)<br><b>Foreign key catalogNo references Video(catalogNo)</b><br><b>Foreign key actorNo references Actor(actorNo)</b>                                                                              | <b>Tel</b> ( <u>telNo</u> , branchNo)<br><b>Foreign key branchNo references Branch(branchNo)</b>                                                                                                                                     |
| <b>Supplier</b> ( <u>supplierNo</u> , sName, sAddress, tel, status)<br>Alternate key tel                                                                                                                                                                      | <b>Member</b> ( <u>memberNo</u> , fName, lName, address)                                                                                                                                                                             |
| <b>VideoOrder</b> ( <u>orderNo</u> , dateOrdered, dateReceived, branchNo)<br>Foreign key branchNo references Branch(branchNo)                                                                                                                                 | <b>RentalAgreement</b> ( <u>rentalNo</u> , dateOut, dateReturn, memberNo, videoNo)<br>Alternate key memberNo, videoNo , dateOut<br>Foreign key memberNo references Member(memberNo)<br>Foreign key videoNo references Video(videoNo) |
| <b>Registration</b> ( <u>branchNo</u> , <u>memberNo</u> , <u>staffNo</u> , dataJoined)<br><b>Foreign key</b> branchNo references Branch(branchNo)<br>Foreign key memberNo references Member(memberNo)<br><b>Foreign key staffNo references Staff(staffNo)</b> | <b>Video</b> ( <u>catalogNo</u> , title, category, dailyRental, price, <b>DirectorNo</b> , supplierNo)<br>Foreign key supplierNo references Supplier(supplierNo)<br><b>Foreign key directorNo references Director(directorNo)</b>    |
| <b>Staff</b> ( <u>staffNo</u> , name, position, salary, branchNo, <b>supervisorStaffNo</b> )<br>Foreign key branchNo references Branch(branchNo)<br><b>Foreign key supervisorStaffNo references Staff(staffNo)</b>                                            | <b>VideoForRent</b> ( <u>videoNo</u> , available, catalogNo, branchNo)<br>Foreign key catalogNo references Video(catalogNo)<br>Foreign key branchNo references Branch(branchNo)                                                      |
| <b>Branch</b> ( <u>branchNo</u> , <b>street</b> , <b>city</b> , <b>state</b> , <b>zipCode</b> , mgrStaffNo)<br><b>Alternate key zipCode</b><br>Foreign key mgrStaffNo references Staff(staffNo)                                                               | <b>VideoOrderLine</b> ( <u>orderNo</u> , <u>catalogNo</u> , quantity)<br>Foreign key orderNo references VideoOrder(orderNo)<br>Foreign key catalogNo references Video(catalogNo)                                                     |

[Return To step 3](#)

## Step 3.3 檢查系統是否滿足未來的成長

- 預估未來可能會發生的重大需求改變，並檢查系統是否滿足未來的成長

[Return To step 3](#)

## Step 3.4 檢查整體性邏輯資料模式是否滿足使用者需求

- 必須與使用者一起討論建構的模式與相關文件，確實滿足使用者需求

[Return to step 3](#)

# 實體資料庫設計

## Physical database design

# 資料庫設計的整體觀

- 邏輯資料庫設計 Logical database design
  - Step 1: 對每一觀點建構區域性邏輯資料模式
  - Step 2: 將區域性邏輯資料模式轉換成對應之資料表結構(schema)
  - Step 3: 建立整體性的邏輯資料模式(選擇性)
- 實體資料庫設計 Physical database design
  - Step 4: 依照使用之DBMS將整體性之資料模式轉換成實體資料庫
  - Step 5: 設計實際限制條件
  - Step 6: 考量避免資料重複機制
  - Step 7: 設定安全機制
  - Step 8: 監控與校調(調整)系統效能

# 實體資料庫設計

## Physical database design

- 實體資料庫設計的流程包括設計資料庫在硬碟中存放的結構，及基本資料表、檔案結構及索引、整合性限制(integrity constraints)、及安全機制(securities)

# 實體資料庫設計方法

## Physical database design methodology

- Step 4:  
將整體性邏輯資料模式依照所使用的DBMS轉換
- Step 5: 設計實體結構
- Step 6: 注意避免產生重複(多餘)資料
- Step 7: 設計安全機制
- Step 8: 監控並校調(調整)系統

Final is an example

# Step 4:將整體性邏輯資料模式依照所使用的DBMS轉換

- Step 4.1 依照所使用的DBMS設計基本資料表
  - 對每一個資料表設計名稱、屬性(欄位)、主索引、候選鍵、外來鍵及參考性整合限制條件
  - 對每一屬性(欄位)設定其定義域(資料型態、長度、限制條件)、預設值、是否允許空值(虛值)(null)
  - 紀錄基本資料表的設計
- Step 4.2依照所使用的DBMS設計企業規則
  - 觸發程式(Trigger)
  - 驗證規則(Validation rule)

# 觸發程式範例

## Trigger example

- 此觸發程式限制會員一次租片不能超過10部

```
CREATE TRIGGER member_not_renting_too_many
BEFORE INSERT OR UPDATE ON rentalAgreement
FOR EACH ROW
DECLARE
    x    NUMBER;
BEGIN
    SELECT COUNT(*) INTO x
        FROM rentalAgreement r
        WHERE r.memberNo = :new.memberNo;
    IF x >= 10 THEN
        raise_application_error (-20000, ('Member' || :new.memberNo || 'already renting 10
        videos'));
    END IF;
END
```

[Return physical DB design menu](#)

# Step 5:設計實體結構

- Step 5.1 分析交易處理
  - 找出一般執行週期
  - 找出資料使用量
- Step 5.2 決定檔案結構
  - Heap, hash, ISAM, B+-tree
- Step 5.3 選擇索引
  - 可以有許多次索引
  - 主索引 or
  - 叢集索引(Clustering index)

# 實體資料庫設計決策

- 建立索引策略
- 為加速查詢而去正規化

# 建立索引策略

- 是否對某個屬性屬性(欄位)建索引
  - 此屬性屬性(欄位)為鍵值、或此屬性屬性(欄位)在查詢或合併時會用到
- 建立多重屬性屬性(欄位)索引
  - 此多重屬性屬性(欄位)一起被查詢時使用 (Garment\_style\_#, Color)
- 是否建叢集索引：每個資料表僅能建一主索引或叢集索引，所以，如果是鍵值則建主索引，否則建叢集索引
- 是否要建雜湊索引：一般RDBMS以B+tree建索引，有些系統亦提供ISAM及雜湊索引，雜湊索引適合合併時找出符合條件的紀錄(比較相等條件)
- 檔案是否使用動態雜湊：適合檔案不斷成長或縮減(但多數RDBMS不提供)

# 為加速查詢而去正規化

- 為加速查詢以更低階(1NF, 2NF)正規化來表現(不用BCNF或4NF)
- 避免因合併而產生多餘資料(多出多筆資料、多出屬性(欄位))
- 避免喪失某些功能性相依

# 為 StayHome建構(其他)索引

| Table           | Column       | Reason             |
|-----------------|--------------|--------------------|
| Branch          | City         | Search condition   |
| Staff           | Name         | ordering           |
| Video           | Category     | ordering           |
|                 |              | Search condition   |
|                 |              | grouping           |
|                 | Title        | ordering           |
|                 |              | Search condition   |
| Actor           | actorName    | Search condition   |
|                 |              | Grouping, ordering |
| Director        | directorName | Search condition   |
| Member          | fName/lName  | Search condition   |
| RentalAgreement | dateReturn   | Search condition   |
| Registration    | dateJoined   | Search condition   |

[Return physical DB design menu](#)

## Step 6: 注意避免產生重複(多餘)資料

- Step 6.1 注意衍生性屬性(欄位)
- Step 6.2 注意重複屬性(欄位)或合併的資料表

[Return physical DB design menu](#)

# Step 7:設計安全機制

- Step 7.1 設計虛擬資料表(view)
- Step 7.2 設計取存規則
  - 給使用者適當的取存權限(勿給予超過其工作權限的權利)

[Return physical DB design menu](#)

# Step 8: 監控並校調(調整)系統——目標

- 資料庫校調(調整)目標使應用程式執行加快、減少交易回應時間、改善交易整體處理量(throughput)
- 校調(調整)前應先收集資料
  - 資料表、索引、緩衝區大小、I/O效能統計資料、查詢/交易時間資料、鎖定/日誌統計資料、索引統計資料(索引層數、分頁數)

# Step 8: 監控並校調(調整)系統——校調(調整)策略(1/2)

- 校調(調整)索引
  - 增加索引：使某類型查詢加速
  - 刪除某些極少使用的索引
  - 對建立在經常變動的屬性(欄位)上之索引，考慮刪除索引(含叢集索引)、重整索引(改建成叢集索引)
- 校調(調整)資料庫設計
  - 多資料表中某些屬性(欄位)需要經常合併在一起，所以現存資料表可能會合併(去正規化)，由BCNF降至3NF、2NF甚至1NF

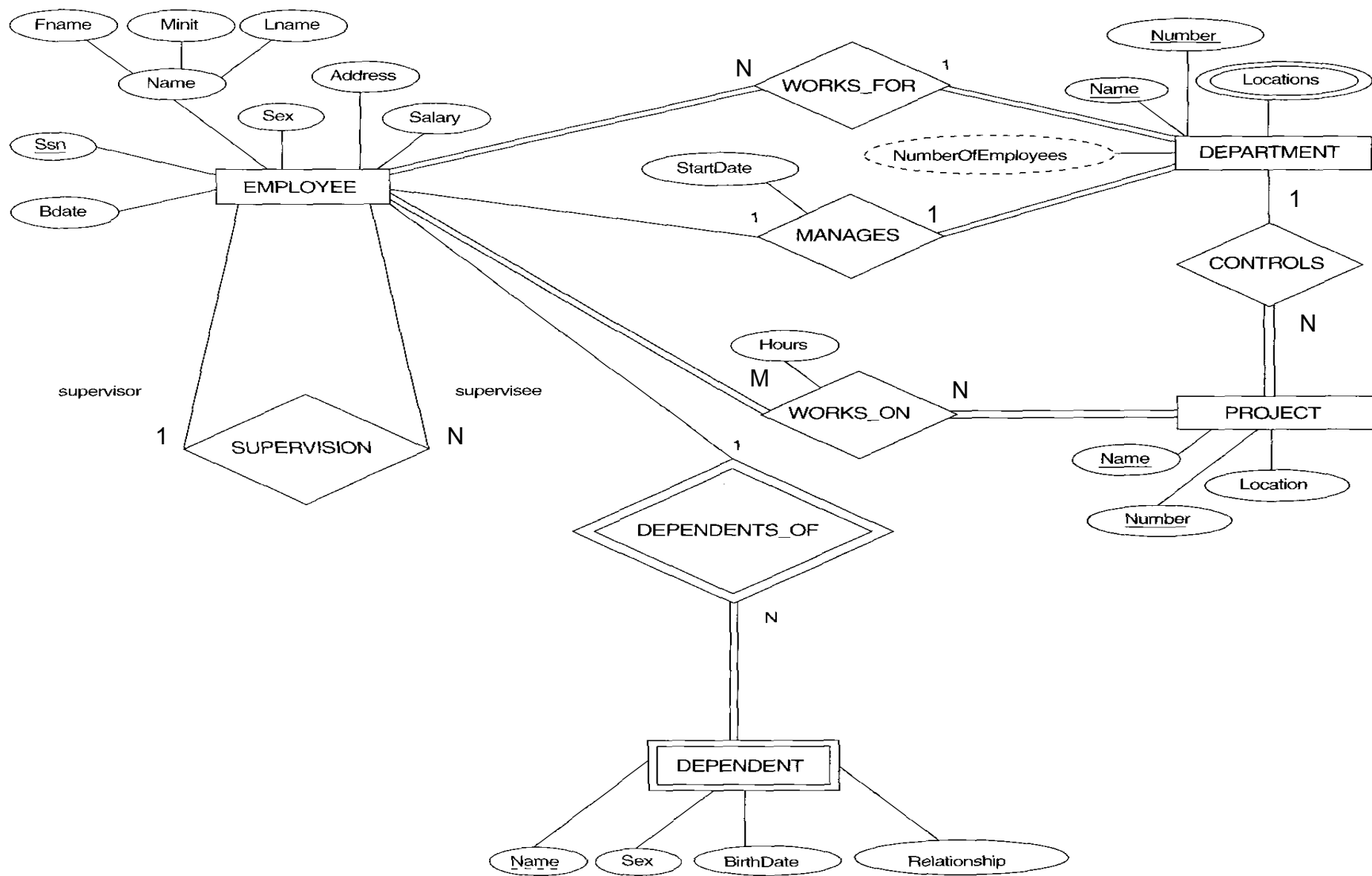
# Step 8: 監控並校調(調整)系統——校調(調整)策略(2/2)

- 對某一組資料表採取替代方案，此替代方案可達3NF或BCNF
- 若原始資料表量大，而某些欄類查詢完全沒交集(如電話和薪資)，則垂直分割(vertical partition)
  - Employee(SSN, Name, Phone, Grade, Salary) → Emp1(SSN, Name, Phone), Emp2(SSN, Grade, Salary)
- 有些屬性(欄位)可能在不同資料表中重複出現(如：跟隨外來鍵的屬性(欄位))，必須有一主檔保證是最新的資料
  - Part\_name 跟隨Part#出現；主檔Master(Part#, Part\_name, .....
- 若查詢時僅某部分資料則可水平分割(屬性(欄位)一樣)
  - 依照產品線分割成10個資料表
  - 查詢時若要全部資料則執行合併指令

[Return physical DB design menu](#)

# Example

- 某公司之每一部門有唯一的名稱，唯一的編號，及一個部門經理，並記錄此經理開始掌管部門的時間，一個部門可能有許多地點。
- 每一部門控制許多專案，每一專案有唯一名稱，唯一編號和單一地點，專案可能有不同部門的員工。
- 每個員工紀錄姓名，身分證字號，地址，薪資，性別，及生日，每一員工隸屬單一部門，可能同時參加許多專案，紀錄每一員工在專案工作時數，及其直屬上司。
- 為了健保，並紀錄每一員工之眷屬，包括眷屬姓名,性別,生日及和員工之關係。



**Figure 3.2** ER schema diagram for the COMPANY database.

# The mapping Schema

- Employee(ssn, Fname, Minit, Lname, sex, address, salary, Bdate, Mgrssn(fk), Dept\_no(fk))
- Department(Dept no, Dept\_name, No\_of\_employee, Mgrssn(fk), start\_date)
- Project(Proj no, Proj\_name, Proj\_location, Dept\_no(fk))
- Dependent(ssn(fk), name, sex, Birthdate, Relationship)
- Works\_on(ssn(fk), proj no(fk), hours)
- Dept\_location(dept no(fk), location)

[Return](#)