

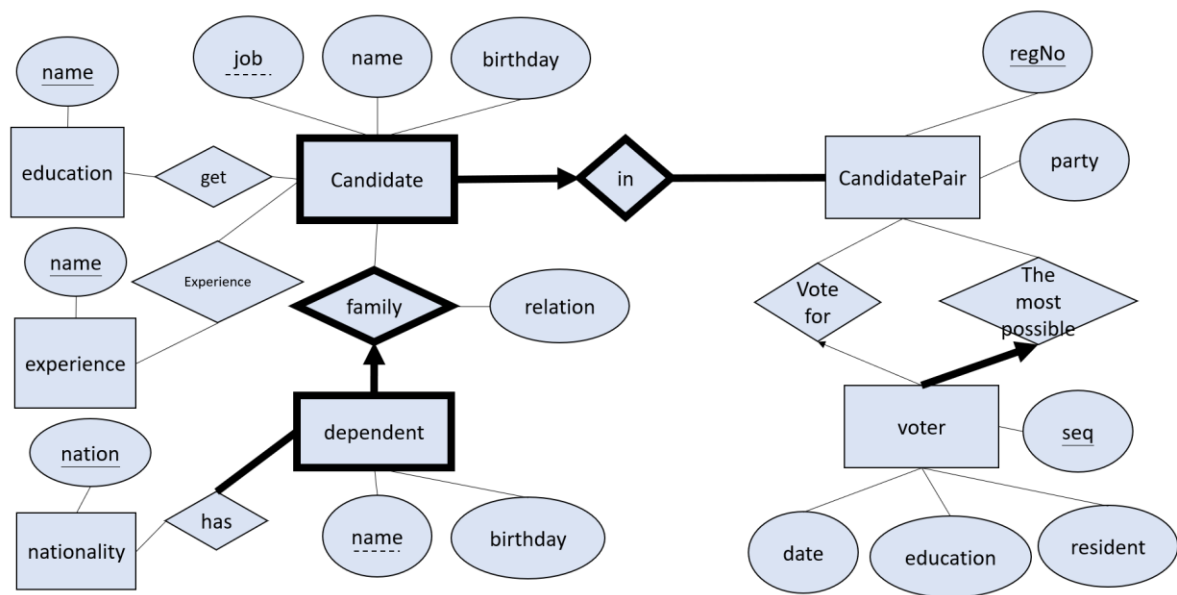
# 資料庫系統作業二

409410050 王謙靜

## 1. Part1

ER-diagram

### PART1



假設：

1. 候選組只會有一個代表政黨
2. 每位選人會參加恰好一個候選組
3. 並不是每位選民都會去投票，但保證他們只會投給一組候選組
4. 當選者只會有一組候選組
5. 每個眷屬只會有一個候選人家人

解釋：

1. 由題目敘述之「候選組的登記號 (regNo) 和代表政黨 (party) 需記載」，可知需要一個名為 candidatePair 的 entity，並有兩個 attribute: regNo 和 party，其中因登記號是流水號，因此確定他是唯一且不會重複的 attribute，故我們可以以 regNo 當 key。因另外假設每位候選組只會有一個代表政黨，故能確定其非 set-valued，因此用 attribute 存。
2. 由題目敘述之「侯選人的候選職位、姓名 (name)、出生年月日 (birthday)、學歷 (degrees) 與經歷 (experience) 要記載。其中學歷與經歷可能有多項資料。」可以知道我們需要另一個名叫 candidate 的 entity，內有三個 attribute: 職位、姓名、生日。而學歷與經歷可能有多項，故另外設兩個 entity，並且分別

用一個 relationship 將他們和 candidate 相連，因為不一定每位候選人都會將自己的學歷以及經歷公開，且有人可能有兩個以上的學經歷，故候選人與 get、experience 間沒有 total participation 和 key constraint，也有可能同一個學經歷被不同的候選人同時擁有(學長和學弟的學歷會相同)，且不保證每個學經歷都會有候選人擁有，所以學歷與經歷對於 get、experience 沒有 key constraint，也沒有 total participation。而在 candidate 中因沒有任何一個 attribute 是唯一的，所以它是一個 weak entity，並由「候選人的候選職位和其『候選組』合起來為唯一。」可知有一名為 in 的 identity relationship 把 candidatePair 跟 candidate 合起來，並且職業是 candidate 這個 weak entity 的 partial key。

3. 由題目敘述之「眷屬的姓名 (name)、出生年月日 (birthday)、國籍 (nationalities) 要記載。其中國籍可能有一個以上。沒有任何一個屬性是唯一」可以知道有一個名為 dependent 的 entity，其有姓名、生日兩個 attribute，因國籍有一個以上，故另外設成 entity，並讓他和 dependent 有一個 relationship(has)，且因每個人必定要有國籍，故在 dependent 與 has 之間有 total participation；因為一個國籍可能被不同眷屬擁有，也有可能不是每個國籍都能找到一個眷屬，所以國籍與 has 之間沒有 total participation 和 key constraint。由於在沒有任何一個屬性是唯一的，所以 dependent 是一個 weak entity，而「每一候選人之眷屬姓名必然不同」中我們可以知道在 dependent 和 candidate 中有一 identity relationship(family)，並且姓名是 partial key
4. 由題目敘述「選民的教育程度 (education)、民調日期 (date)、設籍縣市 (residence) 要記載。由於沒有任何屬性是唯一，可加一流水號屬性 (seq) 成關鍵屬性。」可知有一個名為 voter 的 entity 有四個 attribute：education、date、residence 和 seq，可以透過 seq 唯一找到一個選民，所以把 seq 當 key。
5. 由於需要紀錄候選人與候選組的關係，故設了一個名為 in 的 relationship，因候選人可以透過 in 和自己的 partial key 去找到唯一一筆資料，所以 in 是 identity relationship。每個候選人皆可以透過 in 找到一個唯一的候選組，且每人都要參加候選組，因此 candidate 對 in 是 total participation 以及具有 key constraint。而每一候選組皆至少有一候選人，所以 candidatePair 和 in 之間中具有 total participation。
6. 由於需要紀錄候選人和眷屬的關係，故另外設了一個 relationship：family，family 將候選人與眷屬的關係綁在一起，透過 family 以及眷屬中的 name 我們可以找到唯一一個眷屬資料，所以 family 是 dependent 的 identity relationship。因每個眷屬必須要有一個候選人家人，故有 total participation，且藉由 family 可以找到一個唯一的候選人，故有 key constraint，而在 family 上有一個 attribute 用來記錄候選人與眷屬的關係(父女、母子等等)，此關係也可以綁在眷屬的 attribute 裡，但因父女、母子等關係

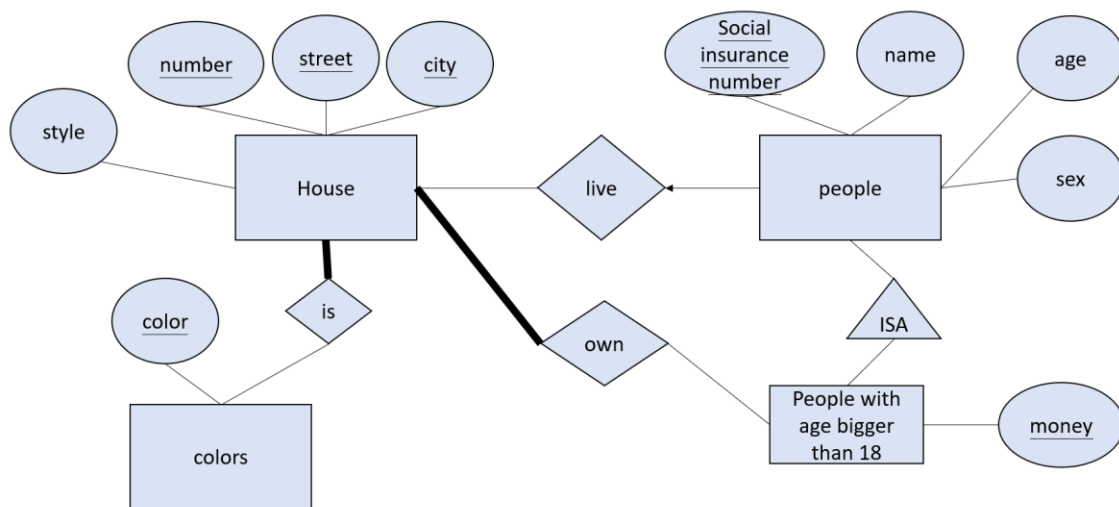
在把兩個人綁在一起時，才比較有意義，故把它放在 family 的 attribute 中。

7. 需要紀錄選民與候選組的兩個關係。第一個是 vote for，由於並不是每個選民在民調時都會表態，故沒有 total participation，但因最多只會投給一個候選組對，故有 key constraint；另一個是 the most possible，因是選民心裡認為的，故每個人都會有答案，故有 total participation，而總統當選只會有一對，故也有 key constraint。

## 2. PART2

ER-diagram

PART2



假設：

1. 每個人有獨一無二的社會保險號碼
2. 每個房子都至少有一種顏色

解釋：

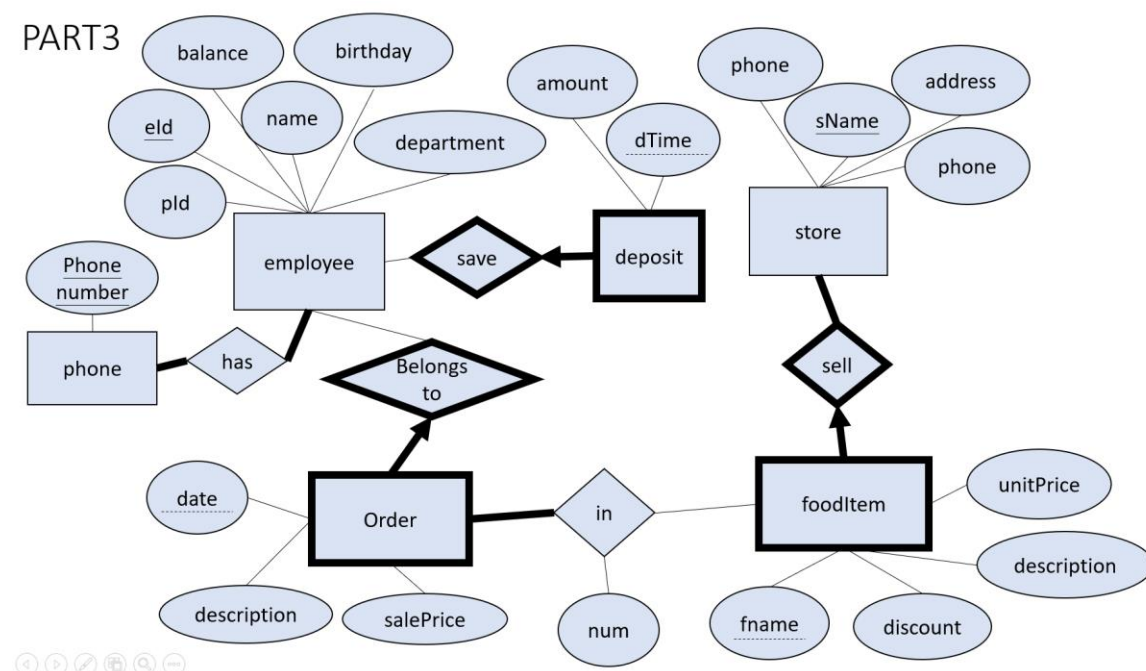
1. 由第一段敘述「A house is identified by a three-part address consisting of a number, street, and city. Each house also has a style (e.g., bungalow 帶走廊的平房) and a set of colors.」可以知道有一個名為 house 的 entity，其有三個 attribute 分別是 number、street、city 以及 style，因 color 有多種可能，故另外設成 entity，讓他用 relationship(is)和 house 和在一起，因每個房子都必須要有顏色，故 is 和 house 之間有 total participation，因為一個顏色可能被多個房子擁有，且並不是每個顏色都會被用在房子上，所以 color 和 is 之間沒有 total participation 和 key constraint。而房子的地址需要三個和在一起才有唯一的房子，故將 number、street 和 city 綁在一起當 key。
2. 由第二段敘述「A person is identified by a social insurance number. For

each person we record his/her name, age, and sex. 」可以知道需要一個名為 people 的 entity，並且有 attribute: social insurance number、name、age、sex，因為有提到人可以用社會保險號碼分別出來，故我們把 social insurance number 當 key。

- 由第三段: 「Persons who are at least 18 years old may own zero or more houses, and every house is owned by at least one person. 」可知有一個名為 own 的 relationship，紀錄 house 與 18 歲以上的人的關係，因為每個房子都必須至少被一個人擁有，所以 own 與 house 之間有 total participation，因可能被多個人擁有，所以 own 與 house 之間沒有 key constraint。而為了強調是 18 歲以上的人，使用了 ISA，從 people 中找出 18 歲以上的人將他們繼承給 people with age bigger than 18 的 entity 中。由於一個人可能有多間房子，也可能沒有房子，所以 own 與 people with age bigger than 18 間沒有 key constraint 和 total participation。
- 由第四段: 「Any person (regardless of age) lives in at most one house as his/her principal residence, and a house has zero or more persons living there. 」可知 house 與 people 有另外一個名為 live 的 relationship，由於每個人只能住一間房子，所以 people 到 live 具有 key constraint，由於有可能有人沒有住在房子裡(無家者)，故 people 到 live 沒有 total participation；由於房子可能沒有人住，也可能很多人一起住，故沒有 total participation 與 key constraint。

### 3. PART3

ER-diagram



假設：

1. 員工必至少有一個電話
2. 每個存款必恰好屬於一個員工
3. 每間商店至少有一個商品
4. 每個餐點必定會被一家餐廳賣出
5. 電話裡只會存員工電話
6. 不是每個員工都要訂飯
7. 每個訂單中必定至少有一個餐點
8. 存在餐廳多個的餐點

解釋：

1. 由第一段關於員工的描述可以知道有一個 entity 名叫 employee，他有很多 attribute，分別為：pId, eId, name, birthday, department, balance，其中 pId 是每個人獨一無二的，因此可以當 key(也有考慮使用身分證編號，但想到員工可能是外國人，故採用公司編號能處理較多的 case)。因為一個人的電話可能有多組，故另外設了一個名為 phone 的 entity，其有一個 attribute：phone number，且以 phone number 當 key，他和員工之間有一個 relationship：has 來紀錄每個員工有哪一些電話，因為員工必至少有一個電話，故員工與 has 之間有 total participation，因為保證每個電話皆有主人，且電話可能被多個人擁有，所以 phone 和 has 之間有 total participation，但是沒有 key constraint。
2. 由第二段的敘述可以知道有一個 entity 叫 store，他有很多 attribute，分別是：sName, address, phone, description，因題目保證 sName 是獨一無二的，因此我們可以把 sName 當作 store 的 key。
3. 由第三段可知，有一個 entity 名為 foodItem，他有 attribute：fname, unitPrice, description, discount，因為未保證其中一個 attribute 是獨一無二的，故 foodItem 是一個 weak entity，由於每一間餐廳只會有一個同名字的商品，因此我們可以用 store 和 fname 去唯一決定一個 foodItem，故他和餐廳之間的 relationship：sell 是 identity relation，並且使用 fname 當作 partial key。在 sell 與 store 中，因為每間餐廳都至少會有一個餐點，且有許多餐廳都有多個餐點，所以 sell 與 store 之間有 total participation，但沒有 key constraint；在關於 foodItem，由於每個餐點會恰好被一間餐廳賣出，故 sell 和 foodItem 間有 total participation 以及 key constraint。
4. 由第四段可以知道，有一個 entity 名為 order，它有 attribute：date, description, sellPrice，因為未能保證其中一個 attribute 是獨一無二的，故 order 也是一個 weak entity，但因為每個人每天最多只會有一筆訂單，因此我們可以用 employee 和 order 的 date 去找到一個唯一的 order，故 order 和 employee 之間的 relationship：belongs to 是 identity relation，並且以 date 為 partial key。在 belongs to 和 employee 中，由於一個人可以在不

同天擁有不同的訂單，並且不一定每個人都有訂飯，所以是一條簡單的細線(無 total participation 和 key constraint)；在 belongs to 和 order 中，由於一筆訂單只能屬於一個人，且所有訂單必定都能找到一個擁有者，故它有 total participation 和 key constraint。另外訂單與餐點之間有另外一個 relationship 名為 in，代表餐點在訂單之中，in 有 description attribute : num，代表該項餐點在訂單中訂了幾個。其中因為訂單中必定有一個餐點，所以訂單與 in 之間有 total participation，但因訂單可以有多項餐點所以沒有 key constraint；因為餐點不會只在一個訂單中出現，且有可能有餐點不曾被點過，所以 foodItem 與 in 之間沒有 total participation，也沒有 key constraint。

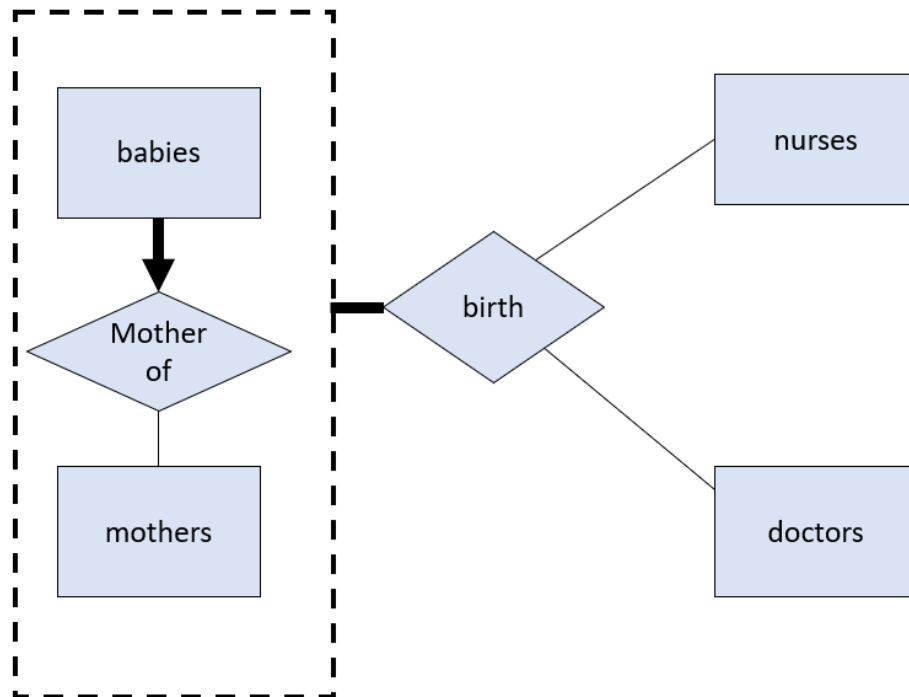
5. 由第五段可以知道，有一個 entity 名為 deposit，它有 attribute : dTime, amount，由於時間和金額都不是唯一，所以 deposit 是 weak entity。一個人在同一時間只能存款一次，所以我們可以透過 employee 和 deposit 的 dTime 去唯一決定一筆存款紀錄。依此規定我們可以找到 identity relationship 是 save，也就是將員工和存款連結的 relationship，而 partial key 是 dTime。一個 employee 可以有多筆存款，也可以都沒有存款(像是直接不參加訂飯系統，或者沒訂過飯)，故 employee 與 save 之間，既沒有 key constraint，也沒有 total participation；但一個存款紀錄只能有一個擁有者，以及所有存款都必須要有擁有者，故 deposit 和 save 之間是 total participation 和 key constraint 都有。

## 4. PART4

### PART4-A

#### ER-diagram

## PART4-A



假設：

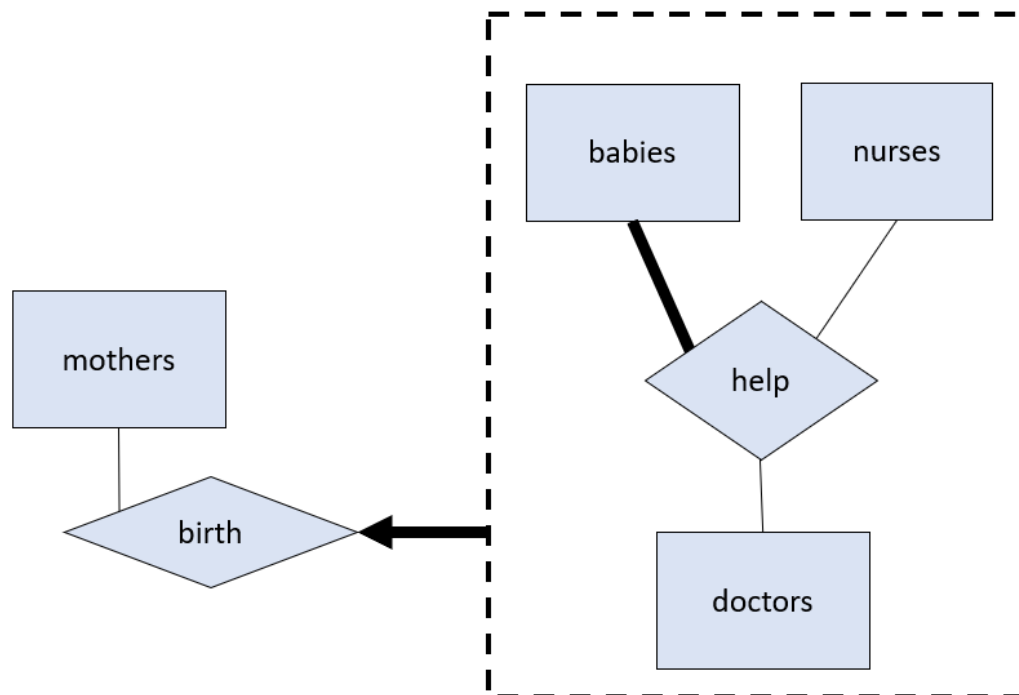
1. 每個嬰兒都有參與 birth 關係(每個嬰兒的出生都有紀錄)
2. 不是每個醫生、護士都有參與 birth

解釋：

由於每個 babies 可以找到唯一的母親，所以我們藉由 aggregation 可以把原本有四個 input 的 relationship 以類似降階的概念，變成只有三個 input，新增了一個 relationship：mother of，所有 babies 都可以透過 mother of 可以找到一個唯一的 mother，所以 babies 與 mother of 之間有 total participation 和 key constraint；而一個母親可以有多个小孩，也可不生孩子，所以沒有 key constraint 也沒有 total participation。處理好 mothers 和 babies 的關係後，將它做 aggregation，代表 birth 是有同時考慮 mother 以及 babies 的，只是我們先處理 mother 與 babies，處理好後再與 doctors 與 nurses 有 birth 的關係。aggregation 與 birth 之間因為每個嬰兒都要出生，所以有 total participation，但一個嬰兒與媽媽的組合可能會遇到多個護士與醫生，所以沒有 key constraint。因為可能有醫生、護士幫兩個嬰兒出生，或者沒有負責幫嬰兒出生(可能不是婦產科的醫生)，所以醫生、護士與 birth 之間沒有 total participation 和 key constraint。

## PART4-B

## PART4-B



假設：

1. 每個嬰兒都有參與 birth 關係(每個嬰兒的出生都有紀錄)
2. 不是每個醫生、護士都有參與 birth

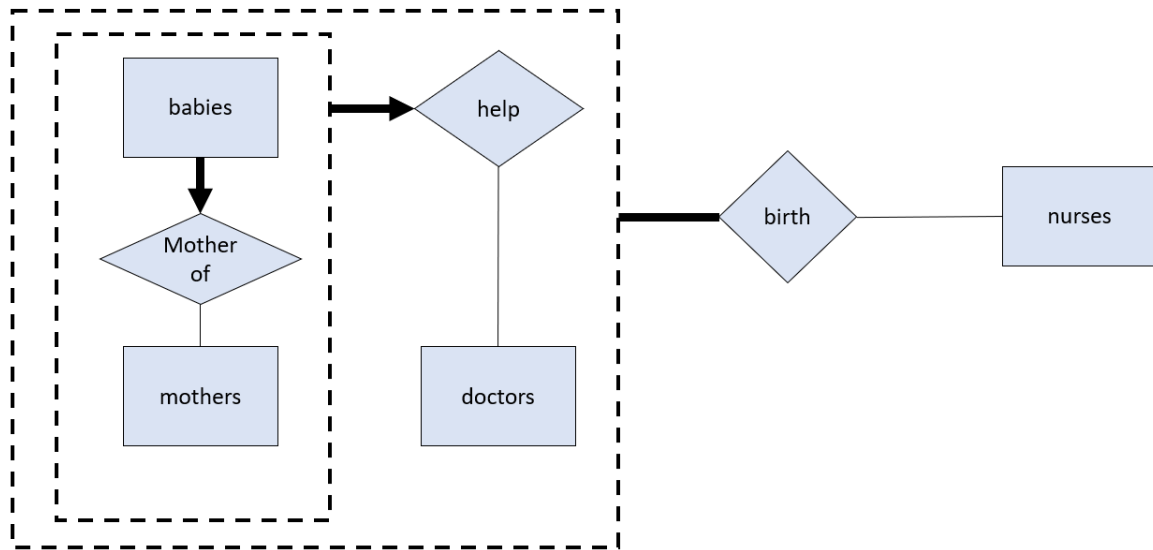
解釋：

由於每個 **babies** 與醫生護士的組合可以找到唯一的母親，所以我們藉由 aggregation 可以把原本有四個 input 的 relationship 以類似降階的概念，將 **birth** 變成只有兩個 input 的 relationship，但在 aggregation 中的 **help** 仍是 ternary，由於每個 **babies** 都要出生，所以他與 **help** 之間有 total participation；有可能有醫生、護士幫兩個嬰兒出生，或者沒有負責幫嬰兒出生(可能不是婦產科的醫生)，所以醫生、護士與 **help** 之間沒有 total participation 和 key constraint。而嬰兒、醫生和護士三人的組合可以恰好找到一個媽媽，所有 aggregation 與 **birth** 之間有 total participation 以及 key constraint；而一個母親可以有 multiple 小孩，也可不生孩子，所以沒有 key constraint 也沒有 total participation。

## PART4-C



## PART4-C



假設：

1. 每個嬰兒都恰好有一個媽媽
2. 每個嬰兒都有參與 birth 關係(每個嬰兒的出生都有紀錄)
3. 不是每個醫生、護士都有參與 birth

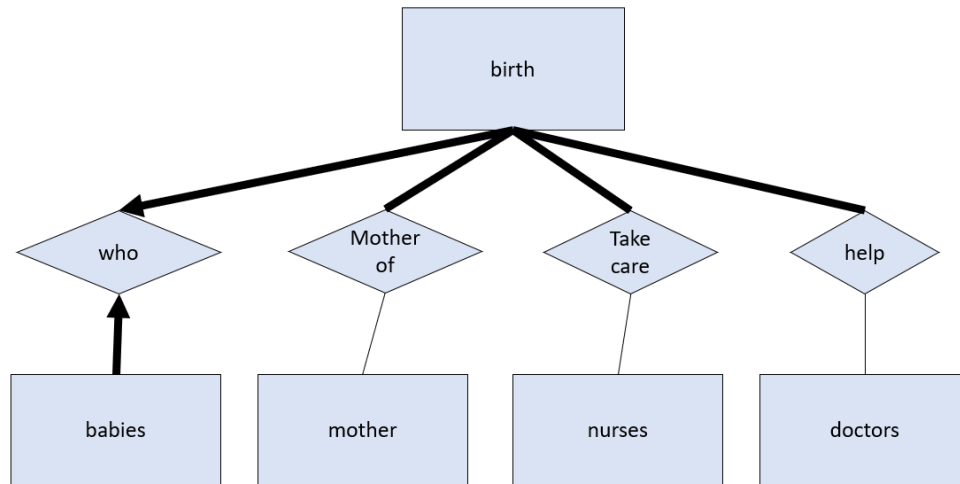
解釋：

由於每個嬰兒都恰好有一個媽媽，所以所有嬰兒可以藉由 mother of 來找到唯一一個母親，所以 babies 與 mother of 是有 total participation 和 key constraint，而母親可能有 0 或多個孩子，媽媽與 mother of 則是 total participation 和 key constraint 都沒有。一個嬰兒和媽媽的組合可以找到一個唯一的醫生，所以另外設了一個 relation 叫 help，為了呈現是嬰兒與媽媽的組合，所以透過 aggregation 將嬰兒與母親綁在一起，透過 help 可以讓有 aggregation 的 babies 和 mother 找到恰好一個醫生，所以 aggregation(mothers 和 babies 的)與 help 之間有 total participation 和 key constraint。由於每個嬰兒都要參加 birth 的 relationship，透過 birth 我們可以知道嬰兒出生時需要哪一些護士，因為已經處理好嬰兒、媽媽以及醫生三方的關係，所以我們可以再做一次 aggregation，再利用 aggregation 來跟護士建立 birth 關係，由於所有嬰兒都要參加，且可能有多個護士協助出生所以 birth 與媽媽、嬰兒、醫生的 aggregation 有之間 total participation 但沒有 key constraint，而有可能有護士幫兩個嬰兒出生，或者沒有負責幫嬰兒出生，所以護士與 help 之間沒有 total participation 和 key constraint。

## 5. PART5

### PART5-A

## PART5-A



假設：

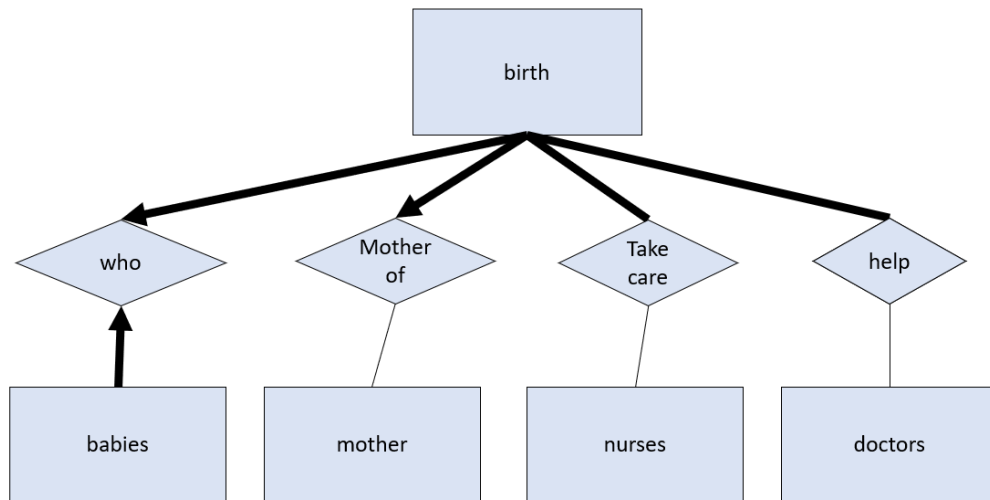
1. 每個嬰兒都要出生
2. 所有 Birth 都需要嬰兒、媽媽、護士和醫生四方參與

解釋：

由於一個嬰兒只會恰好出生一次，所以嬰兒到與 birth 之間的 relationship 有 total participation 和 key constraint，因為每個 birth 都需要媽媽、醫生、護士、嬰兒參與，所以 birth 與他們之間的 relationship 都有 total participation，由於題目中保證一個 birth 只會有一個嬰兒，所以也有 key constraint。每個嬰兒只會出生恰好一次，所以嬰兒和 who 之間的關係，也具有 total participation 和 key constraint。一位母親可能有 0 或多個孩子，因此媽媽與 mother of 之間是 total participation 和 key constraint 都沒有。有可能有醫生、護士幫兩個嬰兒出生，或者沒有負責幫嬰兒出生，所以醫生、護士與 help、take care 之間沒有 total participation 和 key constraint。

## PART5-B

## PART5-B



假設:與 part5-A 相同

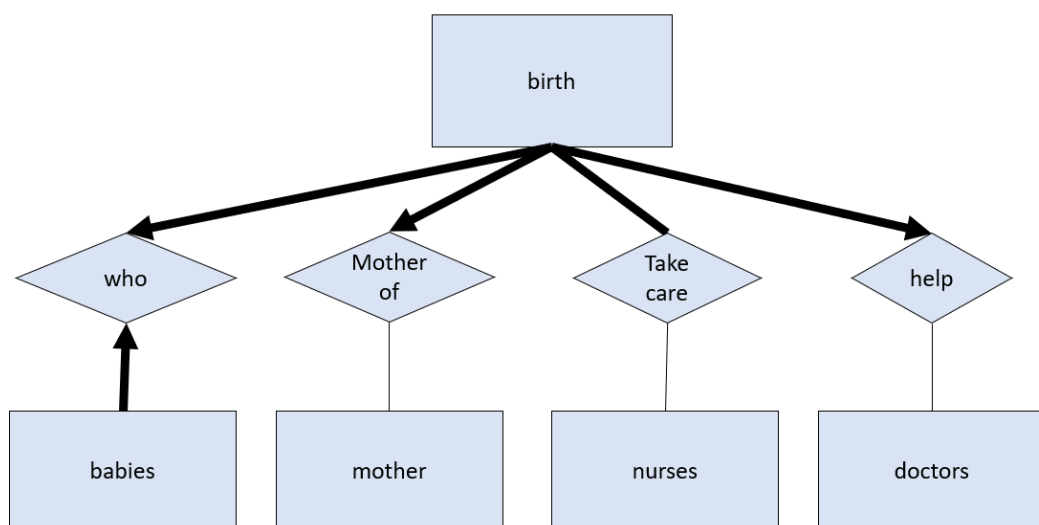
解釋:

大部分內容於 A 部分相同，差別在於 birth 往 mother 的 relationship 上多加了一個箭頭，因為每一個嬰兒可以找到唯一一個媽媽，由加上 A 部分的條件，所以嬰兒可以透過 birth 來找到媽媽，因為保證媽媽是唯一的，所以 birth 對他與 mother 之間的 relationship 具有 key constraint。

## PART5-C

ER-diagram

### PART5-C



假設與 part5-A 相同

解釋：

此圖與 B 圖的差異在於在 birth 與 doctor 之間 birth 加了一個箭頭(key constraint) 代表每一個出生，都恰好有一位醫生參與。

## PART5-D

在 part4 的圖中我們不能夠有一種情況：有一個有醫生、媽媽和嬰兒的關係沒有護士，一旦有一個 birth 那他就必定要有四個 entity 參與，缺一不可。而 part5 的圖則可能發生，一個 birth 可以沒有參加與護士一起的 relationship，故兩圖是不同的。