

Weekly review 1

用 DBMS 比直接使用 File 管理資料的好處

可減少資料重複 ... ((請參考投影片)

資料架構可分成哪幾種 level

- ☒ logical level
- ☒ view level
- ☒ physical level

續上題, 哪一個 level 用來 "describes data stored in database, and the relationships among the data."

- ☐ physical level
- ☒ logical level
- ☐ view level

一個資料庫的 logical structure 稱為什麼?

- ☒ schema
- ☐ instance

續上題, 資料庫中定義好的 schema 會存放在哪裡?

Data dictionary

Weekly review 2

Relational database 中的一個 Table 可以有幾個 superkey?

- ☐ 1 個
- ☒ 可有多個

若已知一個 relational database 中 Table R 的 attribute 有 (A, B, C), 而且已知 A 是 R 的 superkey. B 跟 C 合起來, 也是 R 的 superkey, 而且單獨 B 不是 R 的 superkey, 單獨 C 也不是 R 的 superkey. 以下敘述何者正確?

- ☒ {A, B} 合起來是 R 的一個 superkey
- ☒ {A, C} 合起來是 R 的一個 superkey
- ☒ {A, B, C} 合起來是 R 的一個 superkey

續上題, 以下敘述何者正確?

- ☒ {A} 是 R 的一個 Candidate key
- ☐ {B} 是 R 的一個 Candidate key
- ☒ {B, C} 是 R 的一個 Candidate key
- ☐ {A, B} 是 R 的一個 Candidate key

Relational database 中一個 Table 的 superkey 數目跟 candidate key 數目, 那個大?

- ☒ # superkey \geq # candidate key
- ☐ # superkey \leq # candidate key

續前題, 表 R 的 primary key 可設為什麼?

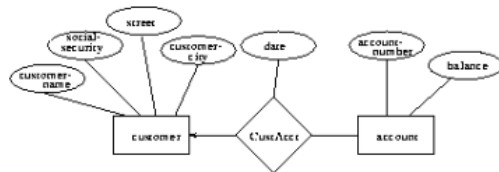
{A} 或 {B, C}

ER-diagram 中是用來在資料庫設計階段, 分析所需儲存的資料. 主要包括 entity 及 entity 之間的 relationship. entity 的描述必須訂出那些?

- ☐ entity name
- ☐ attributes
- ☐ superkey
- ☒ 以上皆是

如以下ER-diagram圖所示, 表示的意義何者正確?

- ☐ 一個customer 最多可有一個account
- ☒ 一個customer 可有多個account
- ☒ 一個account 最多可屬於一個costomer
- ☐ 一個account 最多可屬於多個costomer



如以上ER-diagram圖所示, 根據課本介紹的表示法, 若要表示一個account一定要對應到最少一個 customer, 要在跟哪的entity 的連接邊畫雙線?

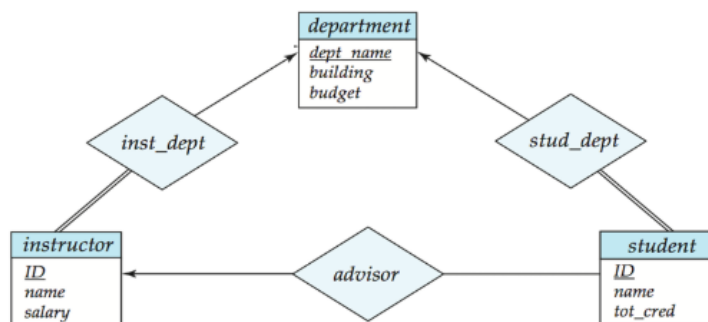
- ☐ customer
- ☒ account

Weekly review 3

從設計好的 ER-diagram 可採用以下轉換方式, 轉成對應的schema. 首先從轉換 strong entity 開始. 以下圖一為例, strong entity set department 轉成一個Table 叫 department, 它的欄位有哪些?

- ☒ dept_name
- ☒ building
- ☒ budget
- ☐ s_ID
- ☐ i_ID

圖一



續上題, strong entity set "instructor" 轉成一個表叫 instructor, 其對應欄位有哪些?

- ☒ ID
- ☒ name
- ☒ salary
- ☐ student
- ☒ dept_name

續上題, 其中 instructor 表中加入了一個原先在strong entity 中不存在的屬性 dept_name, 是用來記錄 er-diagram 中的哪個 relationship?

inst_dept

續上題, 課本將 relationship advisor 對應建一個表 叫做advisor, 其中的屬性是 instructor 的primary key ID, 及student 的primary key ID, 因為同名所以另外命名為 sID 跟 iID. 請問此表的 primary key 該設成甚麼?

- ☐ {iID}
- ☒ {sID}
- ☐ {iID, sID}

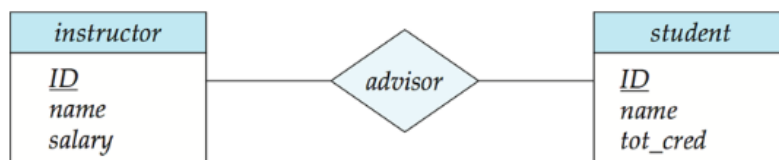
若對 多對一的 relationship advisor 不對應建一個表, 而是在student 的表中增加一個屬性 advisor_id 記錄指導老師 ID, 可能有什麼問題

- ☒ 當一個學生沒有指導老師時, advisor_id 為空值
- ☐ 當一個學生有多個指導老師時, advisor_id 無法儲存 (根據圖一顯示的意義判斷)
- ☐ 當一個老師指導多個學生時, 無法記錄

若 relationship advisor 改成如以下圖二的多對多關係, 轉成表後可以如何表示?

- ☐ 在 student 表中加一個屬性iID
- ☐ 在 instructor 中加一個屬性sID
- ☒ 必須另建一個表advisor, 其中的屬性是 iID, 及sID,

圖二



續上題, 對應到多對多關係 advisor 的表之 primary key 應該為

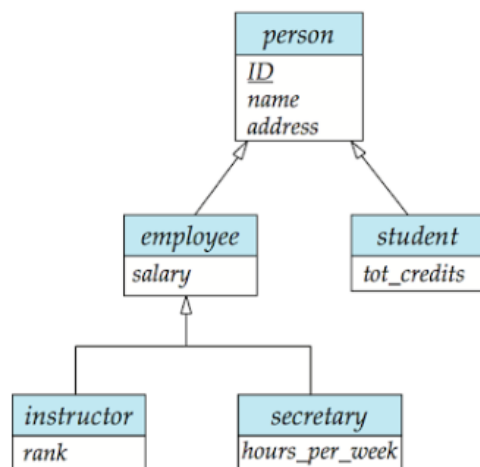
- ☐ {iID}
- ☐ {sID}
- ☒ {iID, sID}

Weekly review 4

由以下圖一 ER diagram 中 specialization 的設計, 轉成表格時, instructor 對應的表中需存那些欄位 (可多選)

- ☒ ID, rank
- ☒ ID, name, address, salary, rank
- ☐ rank

圖一



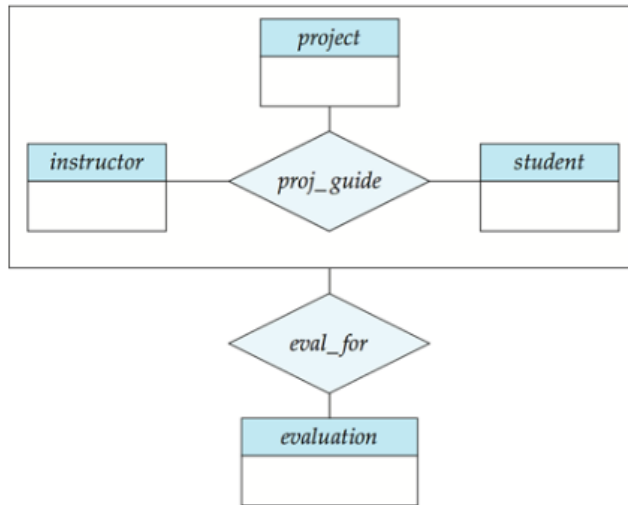
續上題, 若 instructor 中只記錄 ID 及 rank, 其缺點為何?

需要其他欄位如 name, address, salary 需要 join 其他表

續上題, 若 instructor 中記錄 ID, name, address, salary, rank, 其缺點為何?

不同身分之人之基本資料(ID, name, address)分散存在各表中, 需要分別查詢後union起來

圖二



如以上圖二，該設計將instructor, student, project 的三元關係，視為一種entity，稱為"Aggregation"，再跟evaluation 建立一個relationship，所以eval_for 轉成表格後，其中的屬性應該有哪些？

- ☒ i_ID
- ☒ s_ID
- ☒ p_ID
- ☒ e_ID

如以下圖三的三元關係，如果要以二元關係表示，需要哪幾個關係？

- ☒ A跟B間的二元關係R1
- ☒ B跟C之間的二元關係R2
- ☒ A跟C之間的二元關係R3

圖三

