



# 第五章 關聯式代數與關聯式 計算

資料庫系統設計理論  
李紹綸 著



# 本章內容

- 關聯式代數的基本運算子
  - 選擇運算 (Select)
  - 投射運算 (Project)
  - 聯集運算 (Union)
  - 差集運算 (Difference)
  - 卡笛生乘積運算 (Cartesian Product)
- 關聯式代數的非基本運算子
  - 聯結運算 (Join)
  - 交集運算 (Intersection)
  - 除法運算 (Divide)
- 關聯式代數的聚合函數
- 關聯式代數的更新運算
  - 新增運算 (Insert)
  - 修改運算 (Update)
  - 刪除運算 (Delete)
- 關聯式計算
  - 命題邏輯與賓辭邏輯
  - 列錄計算
  - 定義域計算



# 關聯式代數 V.S. 關聯式計算

- 關聯式代數 (Relational Algebra)
  - 一種程序式查詢語言 (Procedural Query Language)
  - 屬於低階資料操作語言 (Low-Level DML)
  - 使用者不但需指出要尋找那些 (What) 資料，同時更要詳細定義出如何 (How) 去尋找這些資料
- 關聯式計算 (Relational Calculus)
  - 一種非程序式查詢語言 (Non-Procedural Query Language)
  - 屬於高階資料操作語言 (High-Level DML)
  - 使用者通常僅須定義要尋找那些資料，而不需定義要如何去尋找這些資料



# 關聯式代數(Relational Algebra)

- 傳統的集合運算子
  - 聯集運算子 (Union)，符號為  $\cup$
  - 交集運算子 (Intersection)，符號為  $\cap$
  - 差集運算子 (Difference)，符號為  $-$
  - 卡笛生乘積運算子 (Cartesian Product)，符號為  $\times$
- 特殊的集合運算子
  - 選擇運算子 (Select)，符號為  $\sigma$
  - 投射運算子 (Project)，符號為  $\pi$
  - 聯結運算子 (Join)，符號為  $\bowtie$
  - 除法運算子 (Divide)，符號為  $\div$



# 聯集相容和封閉性

- 聯集相容 (Union Compatible)
  - 兩個關聯表的維度 (Degree) 必須相同
  - 兩個關聯表中，其相互對映的屬性之定義域也必須一樣
- 封閉性 (Closure)
  - 關聯表經過關聯式代數運算子運算過後的結果仍然還是一個關聯表



# 選擇運算 (Select)

- 選擇運算是從一個指定的關聯表中選取出符合選擇條件的列錄出來
- 需輸入兩個參數：一個是選擇條件 P (Predicate)，另一個是關聯表名稱 R
- 數學定義的寫法： $\sigma_p(R) = \{ t \mid t \in R \wedge P(t) \}$
- 表示式： $\sigma_{\langle \text{選擇條件} \rangle}(\langle \text{關聯表名稱} \rangle)$
- 具有交換性 (Commutative)
  - $\sigma_{\langle \text{選擇條件}1 \rangle}(\sigma_{\langle \text{選擇條件}2 \rangle}(\langle \text{關聯表名稱} \rangle)) = \sigma_{\langle \text{選擇條件}2 \rangle}(\sigma_{\langle \text{選擇條件}1 \rangle}(\langle \text{關聯表名稱} \rangle))$
  - $\sigma_{\langle \text{選擇條件}1 \rangle}(\sigma_{\langle \text{選擇條件}2 \rangle}(\cdots(\sigma_{\langle \text{選擇條件}n \rangle}(\langle \text{關聯表名稱} \rangle))\cdots)) = \sigma_{\langle \text{選擇條件}1 \rangle \text{ AND } \langle \text{選擇條件}2 \rangle \text{ AND } \cdots \text{ AND } \langle \text{選擇條件}n \rangle}(\langle \text{關聯表名稱} \rangle)$




# 公司資料庫的關聯表

員工	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2

員工電話	身分證號碼	電話
	A123456789	89511111
	A123456789	89512222
	F212121212	77388888
	F232323232	27041234

部門	部門代號	部門名稱	經理身分證號碼	到職日
	1	資訊	E123456789	89/08/01
	2	研發	F232323232	92/02/01

員工眷屬	身分證號碼	眷屬姓名	關係
	A123456789	李嘉誠	父子
	E123456789	李嘉誠	父子
	E123456789	李筱嵐	父女
	F123123123	簡哲維	父子
	F232323232	李筱彤	母女

參加	身分證號碼	計劃代號	工作時數
	E123456789	1	10
	E123456789	2	8
	E123456789	3	12
	F123123123	2	8
	F212121212	1	12
	F212121212	4	7
	F232323232	2	5
	F232323232	3	11
	F232323232	5	6

計劃	計劃代號	計劃名稱	執行地點	委託單位	控制部門
	1	P <sub>1</sub>	台北	政府機關	1
	2	P <sub>2</sub>	台中	私人企業	1
	3	P <sub>3</sub>	台中	私人企業	2
	4	P <sub>4</sub>	台南	私人企業	2
	5	P <sub>5</sub>	高雄	私人企業	1



## 【範例 5.1】

- 查詢有哪些員工的薪水是高於40000 元
- $\sigma_{\text{薪水} > 40000}(\text{員工})$

員工	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2





## 【範例 5.2】

- 查詢有哪些員工的薪水是高於40000 元且生日是在民國60 年以前出生
- $\sigma_{\text{薪水} > 40000 \text{ AND 生日} < '60/01/01'}(\text{員工})$
- $\sigma_{\text{薪水} > 40000}(\sigma_{\text{生日} < '60/01/01'}(\text{員工}))$

員工	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1



# 投射運算 (Project)

- 投射運算是從一個指定的關聯表所有的屬性名單中選取出使用者所指定的屬性，而形成一個新關聯表
- 需輸入兩個參數：一個是欲投射出來的屬性名單 (Attribute List)  $A_1, A_2, \dots, A_n$ ，另一個是關聯表名稱  $R$
- 數學定義的寫法： $\pi_{A_1, A_2, \dots, A_n}(R) = \{t[A_1, A_2, \dots, A_n] \mid t \in R\}$
- 表示式： $\pi_{\langle \text{屬性名單} \rangle}(\langle \text{關聯表名稱} \rangle)$
- 當所挑選的  $\langle \text{屬性名單1} \rangle$  包含於  $\langle \text{屬性名單2} \rangle$  時
  - $\pi_{\langle \text{屬性名單1} \rangle}(\pi_{\langle \text{屬性名單2} \rangle}(\langle \text{關聯表名稱} \rangle)) = \pi_{\langle \text{屬性名單1} \rangle}(\langle \text{關聯表名稱} \rangle)$




## 【範例 5.3】

- 查詢每一位員工的姓名和其薪水所得
- $\Pi_{\text{姓名, 薪水}}(\text{員工})$

員工	姓名	薪水
	李鴻章	28000
	李紹綸	50000
	簡嘉祐	43000
	賴怡君	Null
	莊雅玫	43000



## 【範例 5.4】

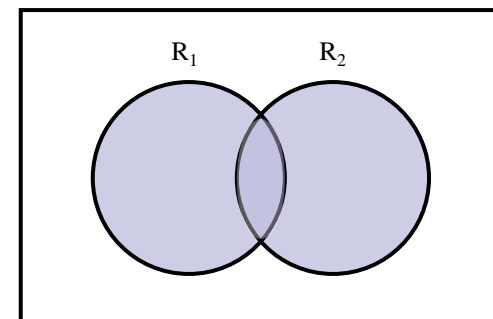
- 查詢每一位員工的薪水所得
- $\Pi_{\text{薪水}}(\text{員工})$

員工	薪水
	28000
	50000
	43000



# 聯集運算 (Union)

- 將兩個滿足聯集相容的關聯表  $R_1$  和  $R_2$ ，聯集成一個新的關聯表
- 需輸入兩個運算元：指定關聯表名稱  $R_1$  和  $R_2$  當成運算元
- 數學定義的寫法： $R_1 \cup R_2 = \{t \mid t \in R_1 \vee t \in R_2\}$
- 表示式： $R_1 \cup R_2$ 
  - 交換律： $R_1 \cup R_2 = R_2 \cup R_1$
  - 結合律： $(R_1 \cup R_2) \cup R_3 = R_1 \cup (R_2 \cup R_3) = R_1 \cup R_2 \cup R_3$
  - 包含性： $R_1 \subseteq R_1 \cup R_2$
  - 吸收律：若  $R_1 \subseteq R_2 \rightarrow (R_1 \cup R_2) = R_2$





## 【範例 5.5】

- 查詢出有參與 1 號計劃或 2 號計劃的員工身分證號碼
- $\pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加})) \cup \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加}))$

–  $R_1 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加}))$

$R_1$	身分證號碼
	E123456789
	F212121212

–  $R_2 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加}))$

$R_2$	身分證號碼
	E123456789
	F123123123
	F232323232

–  $R_1 \cup R_2$

$R_1 \cup R_2$	身分證號碼
	E123456789
	F212121212
	F123123123
	F232323232



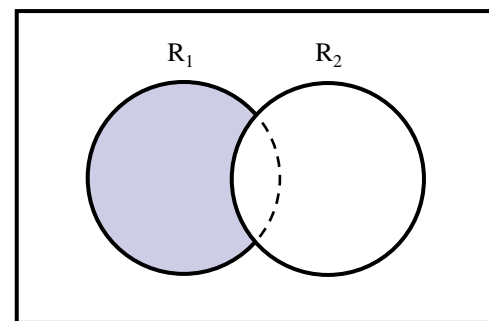
# 指定運算 (Assignment)

- 使用符號  $\leftarrow$  來表示「指定」運算子
- 搭配「重新命名」運算子  $\rho_{\langle \text{暫存關聯表名稱} \rangle}(\langle \text{關聯表名稱} \rangle)$ ，將關聯表重新命名
- 使用時機：為清楚表達出每個查詢的動作，我們可將一個查詢表示式分成數個查詢運算式
- 表示式： $TR(A_1, A_2, \dots, A_n) \leftarrow \text{查詢表示式}$



# 差集運算 (Difference)

- 將兩個滿足聯集相容的關聯表  $R_1$  和  $R_2$ ，以一個關聯表減去另一個關聯表
- 需輸入兩個運算元：指定關聯表名稱  $R_1$  和  $R_2$  當成運算元
- 數學定義的寫法： $R_1 - R_2 = \{t \mid t \in R_1 \wedge t \notin R_2\}$
- 表示式： $R_1 - R_2$
- $R_1 - R_2$  的關聯表是由屬於關聯表  $R_1$  但不屬於  $R_2$  的列錄所組成的集合
  - 不具有交換律： $R_1 - R_2 \neq R_2 - R_1$
  - 不滿足結合律： $(R_1 - R_2) - R_3 \neq R_1 - (R_2 - R_3)$







## 【範例 5.6】

- 查詢出有參與 1 號計劃但沒參與 2 號計劃的員工身分證號碼
- $\pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加})) - \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加}))$

–  $R_1 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加}))$

$R_1$	身分證號碼
	E123456789
	F212121212

–  $R_2 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加}))$

$R_2$	身分證號碼
	E123456789
	F123123123
	F232323232

–  $R_1 - R_2$

$R_1 - R_2$	身分證號碼
	F212121212



## 【範例 5.7】

- 查詢出有參與 2 號計劃但沒參與 1 號計劃的員工身分證號碼
- $\pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加})) - \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加}))$   
 $- R_1 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加}))$   
 $- R_2 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加}))$   
 $- R_2 - R_1$

$R_1$	身分證號碼
	E123456789
	F212121212

$R_2$	身分證號碼
	E123456789
	F123123123
	F232323232

$R_2 - R_1$	身分證號碼
	F123123123
	F232323232



# 卡笛生乘積運算 (Cartesian Product)

- 將原有兩個關聯表中的所有列錄一一合併成一個新的關聯表
- 需輸入兩個運算元：指定關聯表名稱  $R_1$  和  $R_2$  當成運算元
- 數學定義的寫法： $R_1 \times R_2 = \{ t \mid t=(a,b), a=(a_1,a_2,\dots,a_n) \in R_1 \wedge b=(b_1,b_2,\dots,b_m) \in R_2 \}$
- 假設關聯表綱目  $R_1(A_1,A_2,\dots,A_n)$  和  $R_2(B_1,B_2,\dots,B_m)$ ，當  $R_1 \times R_2 = Q$  時，則關聯表  $Q$  的綱目是  $Q(A_1,A_2,\dots,A_n, B_1,B_2,\dots,B_m)$
- 表示式： $R_1 \times R_2$ 
  - 參與運算的兩個關聯表並不需要滿足聯集相容的特性
  - 滿足交換律： $R_1 \times R_2 = R_2 \times R_1$
  - 滿足結合律： $(R_1 \times R_2) \times R_3 = R_1 \times (R_2 \times R_3)$

## 【範例 5.8】

- 查詢出所有員工與部門的資訊，並將其結果儲存於員工部門的臨時表格
- 員工部門  $\leftarrow$  員工  $\times$  部門

員工部門	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門	部門代號	部門名稱	經理身分證號碼	到職日
#	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1	1	資訊	E123456789	89/08/01
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1	2	研發	F232323232	92/02/01
#	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	1	資訊	E123456789	89/08/01
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	2	研發	F232323232	92/02/01
#	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1	1	資訊	E123456789	89/08/01
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1	2	研發	F232323232	92/02/01
#	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2	1	資訊	E123456789	89/08/01
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2	2	研發	F232323232	92/02/01
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2	1	資訊	E123456789	89/08/01
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2	2	研發	F232323232	92/02/01



# 正確查詢表示式

- 員工部門  $\leftarrow \sigma_{\text{工作部門} = \text{部門代號}} (\text{員工} \times \text{部門})$

員工部門	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門	部門代號	部門名稱	經理身分證號碼	到職日
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1	1	資訊	E123456789	89/08/01
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	1	資訊	E123456789	89/08/01
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1	1	資訊	E123456789	89/08/01
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2	2	研發	F232323232	92/02/01
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2	2	研發	F232323232	92/02/01



## 【範例 5.9】

- 請將聯結、交集和除法等運算子利用五種基本運算子來表示
- 聯結運算
  - $R_1 \bowtie_{\langle \text{聯結條件} \rangle} R_2 = \sigma_{\langle \text{選擇條件} \rangle} (R_1 \times R_2)$
- 交集運算
  - $R_1 \cap R_2 = (R_1 \cup R_2) - ((R_1 - R_2) \cup (R_2 - R_1)) = R_1 - (R_1 - R_2) = R_2 - (R_2 - R_1)$
- 除法運算
  - 假設兩個關聯表  $R_1(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$  和  $R_2(B_1, B_2, \dots, B_m)$ ，則
  - $R_1 \div R_2 = \pi_{A_1, A_2, \dots, A_n}(R_1) - \pi_{A_1, A_2, \dots, A_n}((\pi_{A_1, A_2, \dots, A_n}(R_1) \times R_2) - R_1)$



# 聯結運算 (Join)

- 將原有兩個關聯表中的所有列錄一一合併成一個新的關聯表
- 需輸入三個參數：一個是聯結條件  $P$  (Predicate)，另外兩個則是指定關聯表名稱  $R_1$  和  $R_2$
- 數學定義的寫法： $R_1 \bowtie_p R_2 = \{ t \mid t=(a,b), a=(a_1,a_2,\dots,a_n) \in R_1 \wedge b=(b_1,b_2,\dots,b_m) \in R_2 \wedge P(t) \}$
- 表示式： $R_1 \bowtie_{\langle \text{聯結條件} \rangle} R_2$



## $\theta$ 聯結 ( $\theta$ Join)

- $R_1 \bowtie_{\langle \text{聯結條件} \rangle} R_2$
- $\langle \text{聯結條件} \rangle$  可表示成： $\langle \text{條件式子} \rangle \text{ AND } \langle \text{條件式子} \rangle \text{ AND } \dots \text{ AND } \langle \text{條件式子} \rangle$
- $\langle \text{條件式子} \rangle$  的格式都為  $A_i \theta B_j$ ，其中， $A_i$  是  $R_1$  中的一個屬性， $B_j$  是  $R_2$  中的一個屬性，並且  $\text{DOM}(A_i) = \text{DOM}(B_j)$
- $\theta$  是一個比較運算子， $\theta \in \{ =, \neq, >, \geq, <, \leq \}$





# 相等聯結 (Equijoin)

- $R_1 \bowtie_{\langle \text{聯結條件} \rangle} R_2$
- 假設  $R_1(A_1, A_2, \dots, A_n)$  和  $R_2(B_1, B_2, \dots, B_m)$ ，當  $R_1 \bowtie_{\langle \text{聯結條件} \rangle} R_2 = Q$  時，則關聯表  $Q$  的綱目是  $Q(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$
- 在關聯表  $Q$  中的每一個列錄  $t$ ，皆是由關聯表  $R_1$  中的一個列錄  $t_1$  和關聯表  $R_2$  中的另一個列錄  $t_2$  所組合而成
  - $t[A_1, A_2, \dots, A_n] = t_1[A_1, A_2, \dots, A_n]$  和  $t[B_1, B_2, \dots, B_m] = t_2[B_1, B_2, \dots, B_m]$
- $R_1 \bowtie_{\langle \text{聯結條件} \rangle} R_2 = \sigma_{\langle \text{選擇條件} \rangle} (R_1 \times R_2)$

## 【範例 5.10】

- 查詢出所有員工與部門的資訊，並將其結果儲存於員工部門的臨時表格
- 員工部門  $\leftarrow \sigma_{\text{工作部門} = \text{部門代號}} (\text{員工} \times \text{部門})$
- 員工部門  $\leftarrow \text{員工} \bowtie_{\text{工作部門} = \text{部門代號}} \text{部門}$

員工部門	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門	部門代號	部門名稱	經理身分證號碼	到職日
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1	1	資訊	E123456789	89/08/01
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	1	資訊	E123456789	89/08/01
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1	1	資訊	E123456789	89/08/01
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2	2	研發	F232323232	92/02/01
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2	2	研發	F232323232	92/02/01



# 自然聯結 (Natural Join)

- $R_1 *_{\langle \text{聯結條件} \rangle} R_2$
- 也稱為「內部聯結」(Inner Join)
- 會將一個關聯表的主鍵和另一個關聯表的外鍵，其利用屬性值相等時為條件來執行聯結運算
- 如果  $\langle \text{條件式子} \rangle$  中的兩個聯結屬性的名稱是不相同時，則此  $\langle \text{條件式子} \rangle$  不可以被省略掉
- 如果  $\langle \text{條件式子} \rangle$  中的兩個聯結屬性的名稱是完全一樣時，則此  $\langle \text{條件式子} \rangle$  可以被省略掉



## 【範例 5.11】

- 查詢出所有員工與部門的資訊，並將其結果儲存於員工部門的臨時表格
- 會將在每一個 <條件式子> 中的第二個具有完全相同屬性值的屬性去除掉
- 員工部門  $\leftarrow$  員工 \*<sub>工作部門 = 部門代號</sub> 部門

員工部門	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門	部門名稱	經理身分證號碼	到職日
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1	資訊	E123456789	89/08/01
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	資訊	E123456789	89/08/01
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1	資訊	E123456789	89/08/01
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2	研發	F232323232	92/02/01
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2	研發	F232323232	92/02/01



## 【範例 5.12】

- 查詢出所有擁有眷屬的員工並列出其員工與員工眷屬的相關資料，並將其結果儲存於員工眷屬的臨時表格
- 員工眷屬  $\leftarrow$  員工 \* 員工眷屬

員工眷屬	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門	眷屬姓名	關係
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1	李嘉誠	父子
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	李嘉誠	父子
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	李筱嵐	父女
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1	簡哲維	父子
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2	李筱彤	母女



## 外部聯結 (Outer Join)

- 會將一個關聯表的主鍵和另一個關聯表的外鍵，利用屬性值相等為條件來執行聯結運算
- 與「內部聯結」不同的地方在於，「外部聯結」除了將兩個關聯表中主鍵與外鍵的屬性值相等的列錄合併成一個新的關聯表之外，對於那些主鍵與外鍵的屬性值不相等的列錄而言，也必須將這些列錄收納到這個新的關聯表之中
- 如果某一邊關聯表的各筆列錄，在另一邊關聯表中找不到相對應的主、外鍵屬性值時，另一邊關聯表的各個屬性值則會以虛值 (Null) 呈現



# 完全外部聯結 (Full Outer Join)

- 使用符號  $\bowtie$  來表示「完全外部聯結」
- 具有下列兩點特性
  - 對於這些主鍵與外鍵的屬性值不相等的列錄而言，該列錄會先以左邊的關聯表為主要表格列出左邊的關聯表中之所有列錄，那麼右邊關聯表的各個屬性值則會以虛值 (Null) 呈現
  - 對於這些主鍵與外鍵的屬性值不相等的列錄而言，該列錄會再以右邊的關聯表為主要表格列出右邊的關聯表中之所有列錄，那麼左邊關聯表的各個屬性值則會以虛值 (Null) 呈現

## 【範例 5.13】

- 查詢出所有員工的姓名以及每位員工擁有的眷屬姓名，且該名員工如果沒有任何眷屬，則該名員工之眷屬姓名屬性須以虛值 (Null) 呈現，並將其結果儲存於員工眷屬的臨時表格
- 員工眷屬  $\leftarrow \pi$  姓名, 眷屬姓名 (員工  $\bowtie$  員工眷屬)

員工眷屬	姓名	眷屬姓名
	李鴻章	李嘉誠
	李紹綸	李嘉誠
	李紹綸	李筱嵐
	簡嘉祐	簡哲維
	賴怡君	Null
	莊雅玫	李筱彤





# 左外部聯結 (Left Outer Join)

- 使用符號  $\bowtie$  來表示「左外部聯結」
- 具有下列特性
  - 對於這些主鍵與外鍵的屬性值不相等的列錄而言，該列錄會以左邊的關聯表為主要表格，列出左邊的關聯表中之所有列錄，那麼右邊關聯表的各個屬性值則會以虛值 (Null) 呈現



## 【範例 5.14】

- 查詢出所有員工的姓名以及每位員工的直屬上司姓名，如果該名員工沒有任何直屬上司，則該名員工之直屬上司姓名屬性需以虛值 (Null) 呈現，並將其結果儲存於員工直屬上司的臨時表格
- 員工直屬上司(姓名, 上司姓名)  $\leftarrow \Pi_{\text{下屬員工.姓名, 上司員工.姓名}} ((\text{下屬員工} \leftarrow \sigma_{\text{員工}}) \bowtie_{\text{下屬員工.直屬上司} = \text{上司員工.身分證號碼}} (\text{上司員工} \leftarrow \sigma_{\text{員工}}))$

員工直屬上司	姓名	上司姓名
	李鴻章	簡嘉祐
	李紹綸	Null
	簡嘉祐	李紹綸
	賴怡君	莊雅玫
	莊雅玫	Null



# 右外部聯結 (Right Outer Join)

- 使用符號  $\bowtie$  來表示「右外部聯結」
- 具有下列特性
  - 對於這些主鍵與外鍵的屬性值不相等的列錄而言，該列錄會以右邊的關聯表為主要表格，列出右邊的關聯表中之所有列錄，那麼左邊關聯表的各個屬性值則會以虛值 (Null) 呈現



## 【範例 5.15】

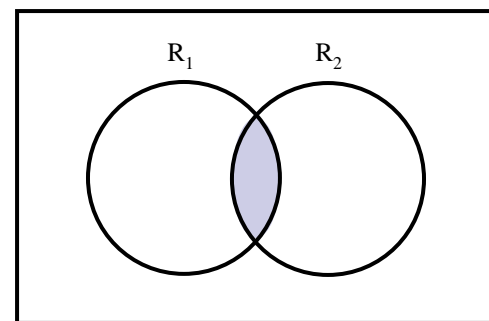
- 查詢出所有員工的姓名以及每位員工的直屬下屬姓名，如果該名員工沒有任何直屬下屬，則該名員工之直屬下屬姓名屬性需以虛值 (Null) 呈現，並將其結果儲存於員工直屬下屬的臨時表格
- 員工直屬下屬(姓名, 下屬姓名)  $\leftarrow \pi_{\text{上司員工.姓名, 下屬員工.姓名}}((\text{下屬員工} \leftarrow \sigma_{\text{員工}}) \bowtie_{\text{下屬員工.直屬上司} = \text{上司員工.身分證號碼}} (\text{上司員工} \leftarrow \sigma_{\text{員工}}))$

員工直屬下屬	姓名	下屬姓名
	李鴻章	Null
	李紹綸	簡嘉祐
	簡嘉祐	李鴻章
	賴怡君	Null
	莊雅玫	賴怡君



# 交集運算 (Intersection)

- 將兩個滿足聯集相容的關聯表  $R_1$  和  $R_2$ ，交集成一個新的關聯表
- 需輸入兩個運算元：指定關聯表名稱  $R_1$  和  $R_2$  當成運算元
- 數學定義的寫法： $R_1 \cap R_2 = \{t \mid t \in R_1 \wedge t \in R_2\}$
- 表示式： $R_1 \cap R_2$ 
  - 交換律： $R_1 \cap R_2 = R_2 \cap R_1$
  - 結合律： $(R_1 \cap R_2) \cap R_3 = R_1 \cap (R_2 \cap R_3) = R_1 \cap R_2 \cap R_3$
  - 包含性： $R_1 \cap R_2 \subseteq R_1$
  - 吸收律：若  $R_1 \subseteq R_2 \rightarrow (R_1 \cap R_2) = R_1$





## 【範例 5.16】

- 查詢出有參與 1 號和計劃 2 號計劃的員工身分證號碼
- $\pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加})) \cap \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加}))$ 
  - $R_1 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=1}(\text{參加}))$
  - $R_2 \leftarrow \pi_{\text{身分證號碼}}(\sigma_{\text{計劃代號}=2}(\text{參加}))$
  - $R_1 \cap R_2$

$R_1$	身分證號碼
	E123456789
	F212121212

$R_2$	身分證號碼
	E123456789
	F123123123
	F232323232

$R_1 \cap R_2$	身分證號碼
	E123456789



## 除法運算 (Divide)

- 使用符號  $\div$  來表示「除法」
- 需輸入兩個運算元：指定「被除」關聯表名稱  $R_1$  和「除」關聯表名稱  $R_2$  當成運算元
- 數學定義的寫法： $R_1 \div R_2 = \{ t \mid (t \times R_2) \in R_1, t \in \pi_{A_1, A_2, \dots, A_n}(R_1) \}$
- 表示式： $R_1 \div R_2$
- 假設  $R_1(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$  和  $R_2(B_1, B_2, \dots, B_m)$ ，當  $R_1 \div R_2 = R$  時，那麼  $R$  的關聯表綱目為  $R(A_1, A_2, \dots, A_n)$



## 【範例 5.17】

- 查詢在  $R_1$  關聯表的  $X$ 、 $Z$  屬性中，有哪些列錄的  $X$ 、 $Z$  屬性值是在  $R_2$  關聯表  $Y$  屬性中「全部」的屬性值  $y_1$  與  $y_2$  均有出現在  $R_1$  關聯表的  $Y$  屬性值中，意即找出  $R(X,Z)$ ，且使得  $R(X,Z) = R_1(X,Y,Z) \div R_2(Y)$
- $R_1(X,Y,Z) \div R_2(Y) = \pi_{X,Z}(R_1) - \pi_{X,Z}((\pi_{X,Z}(R_1) \times R_2) - R_1)$

$R_1$	X	Y	Z
	$x_1$	$y_1$	$z_1$
	$x_1$	$y_2$	$z_1$
	$x_1$	$y_3$	$z_1$
	$x_1$	$y_1$	$z_2$
	$x_2$	$y_1$	$z_3$
	$x_2$	$y_2$	$z_3$

$R_2$	Y
	$y_1$
	$y_2$

R	X	Z
	$x_1$	$z_1$
	$x_2$	$z_3$





# 拆成三個步驟

- $T_1 \leftarrow \pi_{X,Z}(R_1)$
- $T_2 \leftarrow \pi_{X,Z}((T_1 \times R_2) - R_1)$   
 $- T_1 \times R_2$   
 $- (T_1 \times R_2) - R_1$   
 $- T_2 \leftarrow \pi_{X,Z}((T_1 \times R_2) - R_1)$
- $R \leftarrow T_1 - T_2$

$T_1$	X	Z
	$x_1$	$z_1$
	$x_1$	$z_2$
	$x_2$	$z_3$

	X	Z	Y
	$x_1$	$z_1$	$y_1$
	$x_1$	$z_1$	$y_2$
	$x_1$	$z_2$	$y_1$
	$x_1$	$z_2$	$y_2$
	$x_2$	$z_3$	$y_1$
	$x_2$	$z_3$	$y_2$

	X	Z	Y
	$x_1$	$z_2$	$y_2$

$T_2$	X	Z
	$x_1$	$z_2$

R	X	Z
	$x_1$	$z_1$
	$x_2$	$z_3$



## 【範例 5.18】

- 查詢哪些員工有全部參與每一個執行地點在台中的計劃，請列出這些員工的姓名，並將其結果儲存於參與所有台中計劃員工的臨時表格
- 參與所有台中計劃員工  $\leftarrow \pi_{\text{姓名}}(\text{員工} (\pi_{\text{身分證號碼}, \text{計劃代號}}(\text{參加}) \div \pi_{\text{計劃代號}}(\sigma_{\text{執行地點} = \text{'台中'}} \text{計劃})))$

參與所有台中計劃員工	姓名
	李紹綸
	莊雅玫



# 拆成三個步驟

- 台中計劃  $\leftarrow \pi_{\text{計劃代號}}(\sigma_{\text{執行地點} = \text{'台中'}} \text{計劃})$
- 全部參與  $\leftarrow \pi_{\text{身分證號碼, 計劃代號}}(\text{參加}) \div \text{台中計劃}$ 
  - $\pi_{\text{身分證號碼, 計劃代號}}(\text{參加})$

台中計劃	計劃代號
	2
	3

參加	身分證號碼	計劃代號
	E123456789	1
	E123456789	2
	E123456789	3
	F123123123	2
	F212121212	1
	F212121212	4
	F232323232	2
	F232323232	3
	F232323232	5

- 全部參與  $\leftarrow \pi_{\text{身分證號碼, 計劃代號}}(\text{參加}) \div \text{台中計劃}$

全部參與	身分證號碼
	E123456789
	F232323232

- 參與所有台中計劃員工  $\leftarrow \pi_{\text{姓名}}(\text{員工} * \text{全部參與})$

參與所有台中計劃員工	姓名
	李紹綸
	莊雅玫



# 聚合函數 (Aggregate Functions)

- 針對某一群體資料的部分屬性值作一些簡單的數學計算工作
- 表示式： $\langle \text{群組屬性名單} \rangle \mathcal{F} \langle \text{聚合函數名單} \rangle (\langle \text{關聯表名稱} \rangle)$
- 符號名稱和所代表的意義分別為
  - Count：計算某一個關聯表或某屬性的列錄數
  - Avg：計算某屬性的平均值
  - Max：找出某屬性中的最大值
  - Min：找出某屬性中的最小值
  - Sum：計算某屬性的總合

## 【範例 5.19】

- 查詢員工的總人數、最高薪水、最低薪水和平均薪水
- 員工薪水(總人數, 最高薪水, 最低薪水, 平均薪水) ←  
Count(身分證號碼), Max(薪水), Min(薪水), Avg(薪水) 員工

員工薪水	總人數	最高薪水	最低薪水	平均薪水
	5	50000	28000	41000

## 【範例 5.20】

- 查詢每一位員工的身分證號碼和該員工參加計劃個數與總工作時數
- 員工計劃(身分證號碼, 參加計劃個數, 總工作時數) ←  
身分證號碼  $\curvearrowright$  身分證號碼, Count(身分證號碼), Sum(工作時數) 參加

員工計劃	身分證號碼	參加計劃個數	總工作時數
	E123456789	3	30
	F123123123	1	8
	F212121212	2	19
	F232323232	3	22



# 新增運算 (Insert)

- 在關聯表  $R$  中新增一筆列錄  $t$
- 使用聯集運算 ( $\cup$ ) 和指定運算 ( $\leftarrow$ ) 來完成「新增」運算
- 表示式： $R \leftarrow R \cup t$

## 【範例 5.21】

- 在員工關聯表中新增一筆 < K111111111, 郭信義, 64/07/30, 台北市忠孝路186號, 31500, F123123123, 1 > 資料
- 員工  $\leftarrow$  員工  $\cup \{('K111111111', '郭信義', '64/07/30', '台北市忠孝路186號', 31500, 'F123123123', 1)\}$

員工	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2
	K111111111	郭信義	64/07/30	台北市忠孝路186號	31500	F123123123	1





## 修改運算 (Update)

- 在關聯表  $R$  中將某一筆列錄  $t$  修改成為新的列錄  $t_1$ ，新的列錄  $t_1$  可以只更新部分的屬性值
- 使用差集運算 ( $-$ )、聯集運算 ( $\cup$ ) 和指定運算 ( $\leftarrow$ ) 來完成「修改」運算
- 表示式： $R \leftarrow (R - t) \cup t_1$



## 【範例 5.22】

- 員工關聯表中，將「身分證號碼」為K111111111的員工「地址」由“台北市忠孝路186號”改為“台北市忠孝路286號”
- 員工  $\leftarrow (\text{員工} - \sigma_{\text{身分證號碼} = 'K111111111'}(\text{員工})) \cup \{('K111111111', '郭信義', '64/07/30', '台北市忠孝路286號', 31500, 'F123123123', 1)\}$

員工	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2
	K111111111	郭信義	64/07/30	台北市忠孝路286號	31500	F123123123	1



# 刪除運算 (Delete)

- 在關聯表  $R$  中刪除一筆列錄  $t$
- 使用差集運算 ( $-$ ) 和指定運算 ( $\leftarrow$ ) 來完成「刪除」運算
- 表示式： $R \leftarrow R - t$



## 【範例 5.23】

- 在員工關聯表中，將「身分證號碼」為K1111111111 的員工資料予以刪除
- 員工  $\leftarrow$  員工  $- \sigma_{\text{身分證號碼} = 'K1111111111'}$  (員工)

員工	身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門
	A123456789	李鴻章	61/09/01	板橋市忠孝路100號	28000	F123123123	1
	E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1
	F123123123	簡嘉祐	61/07/11	台北市仁愛路168號	43000	E123456789	1
	F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2
	F232323232	莊雅玫	62/12/04	台北市信義路100號	43000	Null	2



## 【範例 5.24】

- 查詢出參與計劃名稱  $P_1$  的員工姓名，並將其結果儲存於參加計劃  $P_1$  的員工的臨時表格
- 參加計劃  $P_1$  的員工  $\leftarrow \pi_{\text{姓名}}(\text{員工} \bowtie_{\text{員工.身分證號碼} = \text{參加.身分證號碼}} (\text{參加} \bowtie_{\text{參加.計劃代號} = \text{計劃.計劃代號}} (\sigma_{\text{計劃名稱} = 'P1'} \text{計劃})))$

參加計劃 $P_1$ 的員工	姓名
	李紹綸
	賴怡君



# 分為下列四個步驟

- $\sigma$  計劃名稱 = 'P1' 計劃

計劃	計劃代號	計劃名稱	執行地點	委託單位	控制部門
	1	P <sub>1</sub>	台北	政府機關	1

- 參加  $\bowtie$  參加.計劃代號 = 計劃.計劃代號 ( $\sigma$  計劃名稱 = 'P1' 計劃)

身分證號碼	計劃代號	工作時數	計劃名稱	執行地點	委託單位	控制部門
E123456789	1	10	P <sub>1</sub>	台北	政府機關	1
F212121212	1	12	P <sub>1</sub>	台北	政府機關	1

- 員工  $\bowtie$  員工.身分證號碼 = 參加.身分證號碼 (參加  $\bowtie$  參加.計劃代號 = 計劃.計劃代號 ( $\sigma$  計劃名稱 = 'P1' 計劃)))

身分證號碼	姓名	生日	地址	薪水	直屬上司	工作部門	計劃代號	工作時數	計劃名稱	執行地點	委託單位	控制部門
E123456789	李紹綸	58/11/01	台北市信義路200號	50000	Null	1	1	10	P <sub>1</sub>	台北	政府機關	1
F212121212	賴怡君	63/07/03	板橋市四川路120號	Null	F232323232	2	1	12	P <sub>1</sub>	台北	政府機關	1

- 參加計劃 P<sub>1</sub> 的員工  $\leftarrow \pi$  姓名 (員工  $\bowtie$  員工.身分證號碼 = 參加.身分證號碼 (參加  $\bowtie$  參加.計劃代號 = 計劃.計劃代號 ( $\sigma$  計劃名稱 = 'P1' 計劃))))

參加計劃P <sub>1</sub> 的員工	姓名
	李紹綸
	賴怡君



# 關聯式計算表示法

- 查詢出參與計劃名稱  $P_1$  的員工姓名，並將其結果儲存於參加計劃  $P_1$  的員工的臨時表格
- $\{(t_1.姓名) \mid (\exists t_1) (\exists t_2) (\exists t_3) (t_1 \in 員工 \wedge t_2 \in 參加 \wedge t_3 \in 計劃 \wedge t_1.身分證號碼 = t_2.身分證號碼 \wedge t_2.計劃代號 = t_3.計劃代號 \wedge t_3.計劃名稱 = 'P_1')\}$
- $\{(t_1.姓名) \mid (\forall t_1 \in 員工) (\forall t_2 \in 參加) (\forall t_3 \in 計劃) (t_1.身分證號碼 = t_2.身分證號碼 \wedge t_2.計劃代號 = t_3.計劃代號 \wedge t_3.計劃名稱 = 'P_1')\}$

參加計劃 $P_1$ 的員工	姓名
	李紹綸
	賴怡君



# 關聯式計算(Relational Calculus)

- 基於「第一階賓辭計算」(The First Order Predicate Calculus) 所發展出來的
- 主要可以分成下列兩種形式
  - 列錄計算 (Tuple Calculus)
    - 格式如： $\{ t \mid P(t) \}$ ，意指關聯表的列錄  $t$  滿足  $P(t)$  的特性描述
  - 定義域計算 (Domain Calculus)
    - 格式如： $\{ \langle A_1, A_2, \dots, A_n \rangle \mid P(\langle A_1, A_2, \dots, A_n \rangle) \}$ ，意指關聯表的定義域  $\langle A_1, A_2, \dots, A_n \rangle$  滿足  $P(\langle A_1, A_2, \dots, A_n \rangle)$  的特性描述





# 命題邏輯 (Propositional Logic)

- 三個簡單的例子來說明何謂「命題」
  - 李紹綸的歌聲好聽嗎？
  - 張學友唱歌非常好聽。
  - 我覺得他的歌聲很好聽耶。



## 基本命題V.S.複合命題

- 命題邏輯 (Propositional Logic) 使用邏輯運算子 (Logical Operators) 聯結所有的「基本命題」，進而產生新的命題
- 邏輯運算子包括：且「 $\wedge$ 」(And)、或「 $\vee$ 」(Or)、否定「 $\neg$ 」(Not)、蘊含「 $\rightarrow$ 」(Implicate) 和等價「 $\leftrightarrow$ 」(Equivalences)



# 真值表

- “天下雨”是一個「基本命題」，馬路濕”也是一個「基本命題」，加上蘊含「 $\rightarrow$ 」邏輯運算子，得到一個全新的「複合命題」：天下雨 $\rightarrow$ 馬路濕
  - 天下雨“蘊含”馬路濕
  - 馬路濕的“前提”是天下雨
  - 天下雨的“結果”是馬路濕
  - 若天下雨則馬路濕

天下雨	馬路濕	天下雨 $\rightarrow$ 馬路濕
真 (True)	真 (True)	真 (True)
真 (True)	假 (False)	假 (False)
假 (False)	真 (True)	真 (True)
假 (False)	假 (False)	真 (True)



# 賓辭邏輯 (Predicate Logic)

- 除了重視如何利用基本命題和邏輯運算子產生一個新的命題之外，也探討一個基本命題的內部結構
  - “所有的鳥類身上都有羽毛”是一個「基本命題」，“鴿子屬於鳥類”也是一個「基本命題」
  - 我們可以輕易的得知“鴿子身上有羽毛”
  - 可以將「基本命題」參數化，將「基本命題」分解成個體 (Individuals) 和賓辭 (Predicate)，使用賓辭來描述個體的性質與特徵



## 賓辭邏輯

- 基本命題  $p$  是“鴿子屬於鳥類”，基本命題  $q$  是“麻雀屬於鳥類”
- 可以分為個體“鴿子”和“麻雀”，賓辭是“鳥類”，用以描述個體的性質是鳥類
  - $p = \text{鳥類 (鴿子)}$
  - $q = \text{鳥類 (麻雀)}$
- 可以使用變數  $x$  代表個體“鴿子”和“麻雀”，變數  $x$  則代表所有的個體
  - $p = \text{鳥類}(x) \wedge x = \text{鴿子}$
  - $q = \text{鳥類}(x) \wedge x = \text{麻雀}$



# 列錄計算

- 在使用列錄計算之前，通常要宣告各個列錄變數的定義域範圍
- 列錄變數  $t$  的定義域範圍是某一個關聯表  $R$ ，則變數  $t$  的值必須是這個關聯表  $R$  中的符合條件  $P(t)$  的各個列錄
- 可以表示成： $\{t \mid P(t)\}$
- $P(t)$  由範圍運算式、列錄變數、屬性和常數等單元值 (Atoms) 組成的第一階賓辭計算 (FOPC) 運算式，亦可稱為公式 (Formula)



## 查詢條件 $P(t)$

- 可以用 BNF (Backus-Normal Form) 來表示
- BNF 是一種「完構公式」(Well-Formed Formula, 簡稱wff), 其格式如下:

```
<wff>  $\equiv$  <condition>  
| NOT <wff>  
| <condition> AND <wff>  
| <condition> OR <wff>  
| IF <condition> THEN <wff>  
| EXISTS t (<wff>)  
| FORALL t (<wff>)  
| (<wff>)  
|  
;
```



# 存在限量詞(Existential Quantifier)

- 存在限量詞 EXISTS
  - 使用符號  $\exists$  來表示
  - $\exists t \in R(<wff>)$  表示存在一個列錄  $t$  會滿足關聯表  $R$  的條件  $<wff>$
  - 必須要有一個列錄需符合  $<wff>$  條件，如此才能使得  $\text{EXISTS } t (<wff>)$  的結果為「真」，並且傳回符合條件  $<wff>$  的各個列錄  $t$





# 總體限量詞 (Universal Quantifier)

- 總體限量詞 FORALL
  - 使用符號  $\forall$  來表示
  - $\forall t \in R(<wff>)$  表示所有的列錄  $t$  會滿足關聯表  $R$  的條件  $<wff>$
  - 必須所有的列錄需符合  $<wff>$  條件，如此才能使得  $\text{FORALL } t (<wff>)$  的結果為「真」，並且傳回符合條件  $<wff>$  的各個列錄  $t$



# 束縛變數 V.S. 自由變數

- 束縛變數(Bounded Variable)
  - 如果該變數是以限量詞 EXISTS 或 FORALL 所引出稱之
  - 其作用範圍僅限於由限量詞所涵蓋的範圍，如同區域變數一般
- 自由變數(Free Variable)
  - 如果該變數不是以限量詞 EXISTS 或 FORALL 所引出稱之
  - 其作用範圍如同整體變數一般



## 【範例 5.25】

- 使用列錄計算查詢出所有在資訊部門上班的員工姓名
- $\{(t_1.姓名) \mid (\exists t_1) (\exists t_2) (t_1 \in 員工 \wedge t_2 \in 部門 \wedge t_1.工作部門 = t_2.部門代號 \wedge t_2.部門名稱 = '資訊'))\}$
- $\{(t_1.姓名) \mid (\forall t_1 \in 員工) (\forall t_2 \in 部門) (t_1.工作部門 = t_2.部門代號 \wedge t_2.部門名稱 = '資訊'))\}$



## 【範例 5.26】

- 使用列錄計算查詢有哪些員工的薪水是高於40000 元且生日是在民國 60 年以前出生
- $\{(t_1.\text{姓名}) \mid (t_1 \in \text{員工}) (t_1.\text{薪水} > 40000 \wedge t_1.\text{生日} < '60/01/01 ')\}$



## 定義域計算

- 以定義域變數取代列錄變數
- 採用「成員條件」，這種條件一般是以“屬性名稱：定義域變數”或“屬性名稱：常數值”成雙成對來表示
- 定義域變數  $t$  的範圍是某一個屬性的定義域，變數  $t$  的值則代表某個屬性的值
- 可以表示成： $\{ \langle A_1, A_2, \dots, A_n \rangle \mid P(\langle A_1, A_2, \dots, A_n \rangle) \}$
- $A_1, A_2, \dots, A_n$  是定義域變數， $P(\langle A_1, A_2, \dots, A_n \rangle)$  則是一個由定義域變數所建立的公式
- $P(\langle A_1, A_2, \dots, A_n \rangle)$  通常也可以用「完構公式」 $\langle wff \rangle$  來表示，只不過寫法方式有所不同而已



## 【範例 5.27】

- 使用定義域計算查詢出所有在資訊部門上班的員工姓名
- $\{ \langle \text{ename} \rangle \mid \exists \text{dno} \langle \text{姓名:ename, 工作部門:dno} \rangle \in \text{員工} \wedge \langle \text{部門代號:dno, 部門名稱: dname} \rangle \in \text{部門} \wedge \text{dname} = \text{'資訊'} \}$



## 【範例 5.28】

- 使用定義域計算查詢有哪些員工的薪水是高於40000 元且生日是在民國 60 年以前出生
- $\{ \langle \text{ename} \rangle \mid \langle \text{姓名:ename, 薪水:salary, 生日:bdate} \rangle \in \text{員工} \wedge \text{salary} > 40000 \wedge \text{bdate} < '60/01/01' \}$