

## 第五章關聯模式的運算

- 目的
- 基本關聯代數運算子
  - SELECT
  - PROJECT
- 集合運算子
  - ☞ 交集、聯集和差集
  - ☞ 卡迪森乘積
- 進階關聯代數運算子
  - ☞ JOIN運算子
  - ☞ 除法運算子
- 新的關聯代數運算子
  - ☞ OUTERJOIN運算子
  - ☞ 關聯的分群和彙總函數
- 範例
- 關聯邏輯計算法

©黃三益2007  
資料庫的核心理論與實務第三版

5-1

## 目的

- 知名的運算方式包括關聯代數 (Relational Algebra) 和關聯邏輯計算法 (Relational Calculus)
- 關聯代數裡定義了一些運算子，用來操作關聯
- 關聯代數運算子是作用在關聯上，而其運算的結果也是一個關聯
  - ☞ 就如同整數的加法 (+) 和減法 (-)
- 關聯代數裡的運算子可分成兩類
  - ☞ 集合運算子
  - ☞ 關聯代數專屬運算子
- 關聯代數和關聯邏輯計算法是下一章要介紹的SQL之理論基礎，一旦學會活用關聯代數，學習SQL的指令就變的相當簡單了

©黃三益2007  
資料庫的核心理論與實務第三版

5-2

## 基本關聯代數運算子 (SELECT運算子, $\sigma$ )

- 從一個關聯的數個序列值中選出滿足某個條件的序列值
- 一般式： $\text{Result} = \sigma_{\langle \text{選擇條件} \rangle}(\text{R})$ 
  - ☞ 關聯網目名稱為Result，屬性則同於R的屬性
  - ☞ 序列值滿足 $\langle \text{選擇條件} \rangle$ 的R的序列值
- 範例：
  - ☞  $\text{Result} = \sigma_{\text{unitPrice} > 500}(\text{Product})$
  - ☞  $\text{Result} = \sigma_{\text{unitPrice} > 500 \text{ AND } \text{catalog} = \text{'Book'}}(\text{Product})$

©黃三益2007  
資料庫的核心理論與實務第三版

5-3

## 練習5-1

- 參考圖4-4的資料庫，請列出以下運算式的執行結果：

$$\text{Result} = \sigma_{\text{catalog} = \text{'CD'}}(\sigma_{\text{unitPrice} > 300}(\text{Product}))$$

商品 Product	商品編號 pNo	商品名稱 pName	定價 unitPrice	種類 catalog
	b30999	資料庫理論與實務	500	Book
	d11222	任賢齊專輯二	300	CD
	b20666	OLAP進階	500	Book
	b10234	管理資訊系統概論	600	Book
	b40555	系統分析理論與實務	550	Book
	d20777	蔡依林專輯二	350	CD
	v01888	哈林合唱：蔡依林玉子的音樂	450	DVD
	d03333	5566專輯	450	CD
	b51111	電子商務理論與實務	700	Book
	v00111	英雄	400	DVD

Result

商品編號 pNo	商品名稱 pName	定價 unitPrice	種類 catalog
d20777	蔡依林專輯二	350	CD
d03333	5566專輯	450	CD

©黃三益2007  
資料庫的核心理論與實務第三版

5-4

## 基本關聯代數運算子 (PROJECT運算子, $\pi$ )

- 從一個關聯的屬性中選出某些屬性
- 一般式： $\text{Result} = \pi_{\langle \text{屬性串列} \rangle}(\text{R})$ 
  - ⊗ 關聯網目：名稱為Result，屬性則是R中 $\langle \text{屬性串列} \rangle$ 裡的屬性
  - ⊗ 序列值：R的序列值中屬於 $\langle \text{屬性串列} \rangle$ 裡的屬性值所組成的序列值，此外，重複的序列值只會保留一筆
- 範例
  - ⊗  $\text{Result} = \pi_{\text{catalog, unitPrice}}(\text{Product})$

©黃三益2007  
資料庫的核心理論與實務第三版

5-5

## 基本關聯代數運算子 (PROJECT運算子, $\pi$ )(Cont.)

- PROJECT運算子常與SELECT運算子搭配使用
- 範例：列出所有單價超過500元的商品之商品名稱和定價
  - $\text{Result} = \pi_{\text{name, unitPrice}}(\sigma_{\text{unitPrice} > 500}(\text{Product}))$
  - 或
  - $\text{ExpProd} = \sigma_{\text{unitPrice} > 500}(\text{Product})$
  - $\text{Result} = \pi_{\text{name, unitPrice}}(\text{ExpProd})$

©黃三益2007  
資料庫的核心理論與實務第三版

5-6

## 練習5-2

- 參考圖4-4的資料庫，請列出以下運算式的執行結果：

Result =  $\pi_{pName, unitPrice} (\sigma_{catalog='CD'} (\sigma_{unitPrice > 300} (Product)))$

產品 Product	產品編號 pNo	產品名稱 pName	定價 unitPrice	種類 catalog
	b39999	資料庫理論與實務	500	Book
	d11222	蔡依林專輯二	300	CD
	b29666	OLAP進階	500	Book
	b19214	管理資訊系統概論	600	Book
	b99255	系統分析理論與實務	550	Book
	d29777	蔡依林專輯二	350	CD
	v91888	哈林張學友：混血兒的音樂	450	DVD
	d03333	5566專輯	450	CD
	b51111	電子商務理論與實務	700	Book
	v69111	英雄	400	DVD

Result

Name	unitPrice
蔡依林專輯二	350
5566專輯	450



©黃三益2007  
資料庫的心理理論與實務第三版

5-7

## 集合運算子

- 常見的集合運算子包括
  - ☞ 交集 (Intersection)，以符號  $\cap$  表示
  - ☞ 聯集 (Union)，以符號  $\cup$  表示
  - ☞ 差集 (Difference)，以符號  $-$  表示
- 但這些關聯的綱目必須相同（也就是說所有的屬性和其型態必須相同），這種性質稱為聯集相容性
- 聯集的結果裡，重複的序列值也只保留一筆
- 範例如 [下頁圖](#)



©黃三益2007  
資料庫的心理理論與實務第三版

5-8

(R) 學生 Student		(S) 助教 Assistant		Result	
學生ID sId	學生姓名 sName	助教ID aId	助教姓名 aName	學生ID sId	學生姓名 sName
89666	Jack	89333	Eva	89666	Jack
91888	Jim	89666	Jack	90111	Jennifer
91333	Lucy	90111	Jennifer	91123	Tony
90555	Eric	90100	Jenny		
90111	Jennifer	91123	Tony		
90222	Amy				
90777	David				
91123	Tony				

(a)

Result		Result		Result	
學生ID sId	學生姓名 sName	學生ID sId	學生姓名 sName	助教ID aId	助教姓名 aName
89666	Jack	91888	Jim	89333	Eva
91888	Jim	91333	Lucy	90100	Jenny
91333	Lucy	90555	Eric		
90555	Eric	90222	Amy		
90111	Jennifer	90777	David		
90222	Amy				
90777	David				
91123	Tony				
89333	Eva				
90100	Jenny				

(c)

(b)

(d)

(e)

(b) **Result (sId, sName) = Student ∩ Assistant**

(c) **Result (sId, sName) = Student ∪ Assistant**

(d) **Result (sId, sName) = Student - Assistant**

(e) **Result (aId, aName) = Assistant - Student**

5-9

## 集合運算子 (卡迪森乘積)

- 兩個集合也可以做相乘 (又稱為卡迪森乘積, Cartesian Product), 此時其結果的元素就是兩個集合裡的元素兩兩匹配
- 兩個關聯R和S相乘, 表示成 **Result = R × S**
  - 關聯網目: 名稱為Result, 屬性為R的屬性和S的屬性合起來
  - 序列值: R的序列值和S的序列值兩兩結合起來
- 關聯相乘後的序列值數目常常很龐大, 假設R有p個屬性和n筆序列值, S有t個屬性和m筆序列值。則R × S會有p+t個屬性和m × n筆序列值

- 考慮以下的Transaction和Cart資料表和其卡迪森乘積

交易Transaction

交易編號 tNo	交易會員 transMid	交易方式 method	交易時間 transTime	銀行代號 bankId	銀行名稱 bank Name	信用卡 種類 cardType	信用卡號 cardId	到期日 dueDate
91100	a0911234	cart	2005-02-02:18:30:00	010	tb	visa	987654321	2008-01-01
92666	c0927777	cart	2005-10-10:22:10:30	020	fb	visa	123456789	2006-12-31
90111	b0905555	cart	2005-05-05:12:30:30	020	fb	master	444455555	2006-01-01
92555	b0922468	cart	2005-11-11:09:10:00	010	tb	visa	333300000	2007-01-01

購物車Cart

會員編號 mId	購物車產生時間 cartTime	交易編號 tNo
c0927777	2005-10-10:22:00:00	92666
b0905555	2005-05-05:12:00:00	90111
a0911234	2005-02-02:18:00:30	91100

(a)

Transaction×Cart會有9+3個屬性和4× 3筆序列值

©黃三益2007  
資料庫的心理理論與實務第三版

5-11

## 練習5-3

- 參考圖4-4的資料庫，請列出以下運算式的執行結果：  
Result =  $\sigma_{\text{Product.pNo=Author.pNo}}$  (Product × Author)

商品 Product	商品編號 pNo	商品名稱 pName	定價 unitPrice	種類 catalog	作者 Author	商品編號 pNo	作者名稱 name	作者編號 Author_pNo
	b30999	系統分析理論與實務	500	Book		b30999	Huang	b30999
	d11222	任賢齊專輯三	300	CD		b10234	Lin	b10234
	b20666	蔡依林專輯專輯三	350	CD		d11222	William	d11222
	d20777	蔡依林專輯二	350	CD		b20666	Sandra	b20666
	v01888	哈利波特：混血王子的 奇蹟	450	DVD		b40555	Wu	b40555
	b51111	張子高管理理論與實務	700	Book		d20777	JoIn	d20777
	b00111	英雄	400	DVD		v01888	J.K.	v01888
	d03333	5566專輯	450	CD		b51111	Lai	b51111
	d03333	5566專輯	450	CD		b51111	Huang	b51111
	d03333	5566專輯	450	CD		b51111	Lin	b51111
	b51111	張子高管理理論與實務	700	Book		d03333	Jackey	d03333
	b51111	張子高管理理論與實務	700	Book		d03333	David	d03333
	b51111	張子高管理理論與實務	700	Book		d03333	Tom	d03333
	v00111	英雄	400	DVD		d03333	Eric	d03333
	v00111	英雄	400	DVD		v00111	Lee	v00111

5-12

## JOIN運算子 (⋈)

- 將兩個關聯JOIN的意義就如同將兩個關聯相乘後再用**SELECT**運算子找出所需的序列值

- 一般式：**Result = R ⋈ <配合條件> S**

⋈ 關聯網目：名稱為Result，屬性為R的屬性和S的屬性合起來

⋈ 序列值：滿足<配合條件>的R序列值和S序列值兩兩結合起來

⋈ 範例：

**Result = Transaction ⋈ Transaction.tNo=Cart.tNo Cart**

- **R ⋈ S=R×S**



©黃三益2007  
資料庫的核心理論與實務第三版

5-13

## JOIN運算子 (⋈) (Cont.)

**Result = Transaction ⋈ Transaction.tNo=Cart.tNo Cart**

交易 Transaction	交易編號 tNo	交易會員 transMid	交易方式 method	交易時間 transTime	銀行代號 bankId	銀行名稱 bank Name	信用卡種 類 cardType	信用卡號 cardId	到期日 dueDate	會員編號 mId	購物車產生 時間 cartTime	購物車之 交易編號 Cart_tNo
Transaction												
Cart												
Result	交易之交易編號 Transaction_tNo	交易會員 transMid	交易方式 method	交易時間 transTime	銀行代號 bankId	銀行名稱 bank Name	信用卡種類 cardType	信用卡號 cardId	到期日 dueDate	會員編號 mId	購物車產生時間 cartTime	購物車之交易編號 Cart_tNo
	91100	a0911234	cart	2005-02-02:18:30:00	010	tb	visa	987654321	2008-01-01	a0911234	2005-02-02:18:00:30	91100
	92666	c0927777	cart	2005-10-10:22:10:30	020	fb	visa	123456789	2006-12-31	c0927777	2005-10-10:22:00:00	92666
購物車 Cart	會員編號 mId	購物車產生時間 cartTime		交易編號 tNo								
	c0927777	2005-10-10:22:00:00		92666			master	444555555	2006-01-01	b0905555	2005-05-05:12:00:00	90111
	b0905555	2005-05-05:12:00:00		90111								
	a0911234	2005-02-02:18:00:30		91100								
	b0922468	2005-11-11:09:00:30		92555								
	a0910001	2005-09-09:10:00:10		null								

# JOIN運算子 (Cont.)

範例：

Result = Cart ⋈<sub>Cart.mld=Order.mld AND Cart.cartTime=Order.cartTime</sub> Order

購物車 Cart				Order				Cart		
會員編號 mld	購物車產生時間 cartTime	交易編號 tNo	訂購之會員編號 Order_mld	訂購之購物車產生時間 Order_cartTime	商品編號 pNo	訂購數量 amount	購物車之會員編號 Cart_mld	購物車之產生時間 Cart_cartTime	交易編號 tNo	
c0927777	2005-10-10:22:00:00	92	a0911234	2005-02-02:18:00:30	b30999	1	a0911234	2005-02-02:18:00:30	91100	
b0905555	2005-05-05:12:00:00	90	b0905555	2005-05-05:12:00:00	v01888	3	b0905555	2005-05-05:12:00:00	90111	
a0911234	2005-02-02:18:00:30	91	c0927777	2005-10-10:22:00:00	d11222	1	c0927777	2005-10-10:22:00:00	92666	
b0922468	2005-11-11:09:00:30	92	c0927777	2005-10-10:22:00:00	d20777	1	c0927777	2005-10-10:22:00:00	92666	
a0910001	2005-09-09:10:00:10	n	c0927777	2005-10-10:22:00:00	v00111	2	c0927777	2005-10-10:22:00:00	92666	
			b0922468	2005-11-11:09:00:30	b10234	5	b0922468	2005-11-11:09:00:30	92555	
			c0927777	2005-10-10:22:00:00	d11222	10	b0922468	2005-11-11:09:00:30	92555	
			c0927777	2005-10-10:22:00:00	d20777	1	a0910001	2005-09-09:10:00:10	92666	
			c0927777	2005-10-10:22:00:00	v00111	1	a0910001	2005-09-09:10:00:10	92666	
			b0922468	2005-11-11:09:00:30	b10234	5				
			b0922468	2005-11-11:09:00:30	b40555	10				
			a0910001	2005-09-09:10:00:10	d11222	1				

# JOIN運算子 (Cont.)

- 假設R有p個屬性和n筆序列值，S有t個屬性和m筆序列值。則R⋈S會有p+t個屬性，至於其序列值的筆數最少有0筆，最多則有m×n筆
- JOIN運算子也可以串連起來形成較複雜的JOIN運算式：

TransCart = Transaction ⋈<sub>Transaction.tNo=Cart.tNo</sub> Cart

Result1 = TransCart ⋈<sub>TransCart.mld=Order.mld AND TransCart.cartTime=Order.cartTime</sub> Order

其結果等同於

CartOrder = Cart ⋈<sub>Cart.mld=Order.mld AND Cart.cartTime=Order.cartTime</sub> Order

Result2 = Transaction ⋈<sub>Transaction.tNo=CartOrder.tNo</sub> CartOrder

## JOIN運算子 (Cont.)

- 等同匹配運算式(EQUIJOIN)：配合條件用的比較運算子都是等號(=)
- 等同匹配運算式的結果會有屬性值重複的現象
- **NATURAL JOIN (\*)** 去除重複屬性。一般式為 **Result = R\*S**
  - ☞ 關聯網目：名稱為Result，屬性為R的屬性和S的屬性裡的非重複屬性
  - ☞ 序列值：假設C為R和S的共同屬性（即C裡的每一個屬性名稱同時出現在R和S），則將C屬性值一樣的R序列值和S序列值兩兩結合起來，並去除重複屬性值

©黃三益2007  
資料庫的核心理論與實務第三版

5-17

## JOIN運算子 (Cont.)

- 範例：  
**Result = Transaction \* Cart**  
(結果有幾個屬性?)

交易編號 tNo	會員編號 transMid	交易方式 method	交易時間 transTime	銀行代號 bankId	銀行名稱 bankName	信用卡 種類 cardType	信用卡號 cardId	到期日 dueDate	購物車 會員編號 mId	購物車 產生時間 cartTime
91100	a0911234	cart	2005-02-02 18:30:00	010	tb	visa	987654321	2008-01-01	a0911234	2005-02-02 18:00:30
92666	c0927777	cart	2005-10-10 22:10:30	020	fb	visa	123456789	2006-12-31	c0927777	2005-10-10 22:00:00
90111	b0905555	cart	2005-05-05 12:30:30	020	fb	master	444455555	2006-01-01	b0905555	2005-05-05 12:00:00
92555	b0922468	cart	2005-11-11 09:10:00	010	tb	visa	333300000	2007-01-01	b0922468	2005-11-11 09:00:30

©黃三益2007  
資料庫的核心理論與實務第三版

5-18

## JOIN運算子 (Cont.)

Result = Cart\*Order

(結果有幾個屬性?)

Result	訂購會員編號 mId	購物車產生時間 cartTime	商品編號 pNo	訂購數量 amount	交易編號 tNo
	a0911234	2005-02-02 18:00:30	b30999	1	91100
	b0905555	2005-05-05 12:00:00	v01888	3	90111
	c0927777	2005-10-10 22:00:00	d11222	1	92666
	c0927777	2005-10-10 22:00:00	d20777	1	92666
	c0927777	2005-10-10 22:00:00	v00111	2	92666
	b0922468	2005-11-11 09:00:30	b10234	5	92555
	b0922468	2005-11-11 09:00:30	b40555	10	92555
	a0910001	2005-09-09 10:00:00	d11222	1	null

©黃三益2007  
資料庫的核心理論與實務第三版

5-19

## 除法運算子 (÷)

- 假設R的屬性集合為 $K_R$ , S的屬性集合為 $K_S$ , 且 $K_S \subset K_R$ , 才可以將R除以S。一般式為 **Result = R ÷ S**

☞ 關聯網目：名稱為Result, 屬性為R的屬性去除S的屬性 (也就是說  $K_{Result} = K_R - K_S$ )

☞ 序列值：將R按照 $K_{Result}$ 的屬性值分成數群, 對於每一群 (假設其屬性值為 $t$ ), 若其包括所有S的序列值, 則 $t$ 為Result裡的序列值

☞ 範例：有哪些會員曾經購買過「任賢齊專輯三」和「蔡依林專輯二」這兩張 CD

$R(mId, pNo) = \pi_{transMid, pNo} (Record * Transaction);$

$S = \pi_{pNo} (\sigma_{pName = '任賢齊專輯三' \text{ OR } pName = '蔡依林專輯二'} (Product));$

$Result = R \div S;$

©黃三益2007  
資料庫的核心理論與實務第三版

5-20

## 除法運算子 (Cont.)

R

會員編號	商品編號
mld	pNo
a0911234	b30999
b0905555	v01888
b0922468	b10234
b0922468	b40555
a0910001	b40555
a0910001	d03333
c0927777	d11222
c0927777	d20777
c0927777	v00111
c0927777	b51111

(a)

S

商品編號
pNo
d11222
d20777

(b)

Result

會員編號
mld
c0927777

(c)

©黃三益2007  
資料庫的核心理論與實務第三版

5-21

## 完整的關聯代數運算子

- 任何DBMS只要能表達之前介紹的運算子所構成的所有運算式就稱為是具**關聯完整性** (Relational complete) 的DBMS
- 有些運算子可也用另一些運算子來取代
  - 交集 ( $\cap$ ) 可以用聯集 ( $\cup$ ) 和差集 ( $-$ ) 來表達：因為  $R \cap S = (R \cup S) - (R - S) - (S - R)$
  - JOIN 可以用關聯乘法 ( $\times$ ) 和 SELECT 來表達：因為  $R \bowtie_{\langle \text{配合條件} \rangle} S = \sigma_{\langle \text{配合條件} \rangle} (R \times S)$
  - 除法 ( $\div$ ) 可以用關聯乘法 ( $\times$ )、差集 ( $-$ ) 和 SELECT 來表達。考慮  $T(Y) = R(Z) \div S(X)$ ，其中  $Y=Z-X$ ， $X \subseteq Z$ 。我們可用以下運算式得到同樣的 T：
    - $T1 = \pi_Y(R)$
    - $T2 = \pi_Y((S \times T1) - R)$
    - $T = T1 - T2$
- 因此， $\{\sigma, \pi, \cup, -, \times\}$  是一組完整的關聯代數運算子

©黃三益2007  
資料庫的核心理論與實務第三版

5-22

## 關聯代數運算式範例一

- 列出「系統分析理論與實務」的作者姓名

系統分析理論與實務

Product (商品編號pNo, 商品名稱pName, 定價unitPrice, 種類catalog)

Author (商品編號pNo, 創作者名稱name)

Temp =  $\pi_{pNo} (\sigma_{pName='系統分析理論與實務'} (Product))$

Result =  $\pi_{name} (Temp * Author)$



©黃三益2007  
資料庫的核心理論與實務第三版

5-23

## 關聯代數運算式範例一

- 列出所有購買過”系統分析理論與實務”的會員之會員編號和會員姓名

系統分析理論與實務

Product (商品編號pNo, 商品名稱pName, 定價unitPrice, 種類catalog)

Record (交易編號tNo, 商品編號pNo, 交易數量amount, 售價salePrice)

Transaction (交易編號tNo, 會員編號transMId, 交易方式method, ……)

Member (會員編號mId, 姓名name, 會員ID pld, ……)

Temp\_book =  $\pi_{pNo} (\sigma_{pName='系統分析理論與實務'} (Product))$

Temp\_trans =  $\pi_{transMId} (Temp\_book * Record * Transaction)$

Result =  $\pi_{mId, name} (Temp\_trans \bowtie_{transMId=mId} Member)$



©黃三益2007  
資料庫的核心理論與實務第三版

5-24

## 關聯代數運算式範例一

- 對於會員編號為 'b0905555' 的會員，列出其姓名以及所（直接）介紹的會員之會員編號、姓名

b0905555

Member (會員編號mld, 會員姓名name, 會員ID pld, ……., 介紹人introducer)

Member (會員編號mld, 會員姓名name, 會員ID pld, ……., 介紹人introducer)

$Mem5555 = \pi_{mld, name}(\sigma_{mld='b0905555'}(Member))$

$Result = \pi_{Mem5555.name, Member.mld, Member.name}$   
 $(Mem5555 \bowtie_{Mem5555.mld = introducer} (Member))$



©黃三益2007  
資料庫的核心理論與實務第三版

5-25

## 關聯代數運算式範例一

- 列出所有瀏覽過或購買過「系統分析理論與實務」的會員之會員編號和會員姓名
  - 這些會員包括：瀏覽過「系統分析理論與實務」的會員，以及購買過「系統分析理論與實務」的會員

Product (商品編號pNo, 商品名稱pName, 定價unitPrice, 種類catalog)

Browse (會員編號mld, 瀏覽時間browseTime, 商品編號pNo)

Record (交易編號tNo, 商品編號pNo, 交易數量amount, 售價salePrice)

Transaction (交易編號tNo, 會員編號transMid, 交易方式method, …….)

Member (會員編號mld, 姓名name, 會員ID pld, …….)

$Temp\_book = \pi_{pNo}(\sigma_{pName='系統分析理論與實務'}(Product))$

$Browse\_member = \pi_{mld}(Temp\_book * Browse)$

$Trans\_member(mld) = \pi_{transMid}((Temp\_book * Record * Transaction)$

$Result = \pi_{mld, name}((Browse\_member \cup Trans\_member) * Member)$



©黃三益2007  
資料庫的核心理論與實務第三版

5-26

## 關聯代數運算式範例一

- 列出購買所有「Jackey」所創作商品的會員之會員編號和會員姓名
  - ☞ 先找出「Jackey」創作的商品，以及每一個會員所購買的所有商品，然後做關聯的除法 (÷)

**Jacky\_product(pNo) =  $\pi_{pNo}(\sigma_{name='Jackey'}(Author))$**

**Member\_product (mId, pNo) =  $\pi_{transMid, pNo}(Transaction * Record)$**

**Jacky\_member = Member\_product ÷ Jacky\_product**

**Result =  $\pi_{mId, name}(Jacky\_member * Member)$**



©黃三益2007  
資料庫的核心理論與實務第三版

5-27

## OUTER JOIN運算子

- 一般的JOIN (又稱為INNER JOIN)會去除兩個關聯裡沒有被匹配到的序列值
  - ☞ 考慮以下查詢：  
**M =  $\pi_{mId, name, birthday, introducer}(Member)$**   
**I(sMId, sName) =  $\pi_{mId, name}(Member)$**   
**Result =  $\pi_{mId, name, birthday, sName}(M \bowtie_{introducer=sMId} I)$** 
    - ☞ 沒有介紹人的會員會在Result(mId)裡嗎？
    - ☞ 沒有當介紹人的會員會在Result(sName)裡嗎？
- 關聯代數裡定義了**OUTER JOIN**，包括三種運算子
  - ☞ **LEFT OUTER JOIN**
  - ☞ **RIGHT OUTER JOIN**
  - ☞ **FULL OUTER JOIN**



©黃三益2007  
資料庫的核心理論與實務第三版

5-28

## LEFT OUTER JOIN 運算子 (⋈<sub>L</sub>)

- 保留左邊關聯，其一般式為  $\text{Result} = R \text{ ⋈}_L \langle \text{配合條件} \rangle S$
- 關聯網目：同於  $R \text{ ⋈}_L \langle \text{配合條件} \rangle S$
- 序列值： $R \text{ ⋈}_L \langle \text{配合條件} \rangle S$  的序列值再加上沒有與  $S$  配對的  $R$  的序列值加上空值（在  $S$  的屬性）
- 範例：**Transaction** ⋈<sub>L</sub> **Transaction.tNo=Cart.tNo** **Cart**
- (Transaction 和 Cart 的 [範例資料庫](#))

Result_left	交易之 交易編號 Transaction tNo	交易會員 transMid	交易方式 method	交易時間 transTime	銀行代號 bankId	銀行 名稱 bank Name	信用卡 種類 cardType	信用卡號 cardId	到期日 dueDate	會員編號 mId	購物車產生時間 cartTime	購物車之 交易編號 Cart_tNo
	91100	a0911234	cart	2005-02-02:18:30:00	010	tb	visa	987654321	2008-01-01	a0911234	2005-02-02:18:00:30	91100
	92666	c0927777	cart	2005-10-10:22:10:30	020	fb	visa	123456789	2006-12-31	c0927777	2005-10-10:22:00:00	92666
	91888	a0910001	fax	2005-09-10:10:10:00	040	cb	master	777788888	2007-10-10	null	null	null
	92333	c0927777	email	2005-10-15:09:00:00	070	sb	visa	111122222	2007-12-31	null	null	null
	90111	b0905555	cart	2005-05-05:12:30:30	020	fb	master	444455555	2006-01-01	b0905555	2005-05-05:12:00:00	90111
	92555	b0922468	cart	2005-11-11:09:10:00	010	tb	visa	333300000	2007-01-01	b0922468	2005-11-11:09:00:30	92555

©黃三益2007  
資料庫的核心理論與實務第三版

5-29

## RIGHT OUTER JOIN 運算子 (⋈<sub>R</sub>)

- 保留右邊關聯，其一般式為  $\text{Result} = R \text{ ⋈}_R \langle \text{配合條件} \rangle S$
- 關聯網目：同於  $R \text{ ⋈}_R \langle \text{配合條件} \rangle S$
- 序列值： $R \text{ ⋈}_R \langle \text{配合條件} \rangle S$  的序列值再加上沒有與  $R$  配對的  $S$  的序列值加上空值（在  $R$  的屬性）
- 範例：**Transaction** ⋈<sub>R</sub> **Transaction.tno=Cart.tno** **Cart**
- (Transaction 和 Cart 的 [範例資料庫](#))

Result_right	交易之 交易編號 Transaction tNo	會員編號 transMid	交易方式 method	交易時間 transTime	銀行代號 bankId	銀行 名稱 bank Name	信用卡 種類 cardType	信用卡號 cardId	到期日 dueDate	會員編號 mId	購物車產生時間 cartTime	購物車之 交易編號 Cart_tNo
	91100	a0911234	cart	2005-02-02:18:30:00	010	tb	visa	987654321	2008-01-01	a0911234	2005-02-02:18:00:30	91100
	92666	c0927777	cart	2005-10-10:22:10:30	020	fb	visa	123456789	2006-12-31	c0927777	2005-10-10:22:00:00	92666
	90111	b0905555	cart	2005-05-05:12:30:30	020	fb	master	444455555	2006-01-01	b0905555	2005-05-05:12:00:00	90111
	92555	b0922468	cart	2005-11-11:09:10:00	010	tb	visa	333300000	2007-01-01	b0922468	2005-11-11:09:00:30	92555
	null	null	null	null	null	null	null	null	null	a0910001	2005-09-09:10:00:10	null

©黃三益2007  
資料庫的核心理論與實務第三版

5-30

## FULL OUTER JOIN 運算子 (⋈)

- 保留左右兩邊關聯，其一般式為  $Result = R \bowtie_{<配合條件>S}$
- 關聯網目：同於  $R \bowtie_{<配合條件>S}$
- 序列值：R 的序列值再加上沒有與 S 配對的 R 的序列值加上空值（在 S 的屬性）以及沒有與 R 配對的 S 的序列值加上空值（在 R 的屬性）
- 範例： $Transaction \bowtie_{Transaction.tno=Cart.tno} Cart$
- (Transaction 和 Cart 的範例資料庫)

交易之 Transaction tNo	交易會員 transMid	交易方式 method	交易時間 transTime	銀行代號 bankId	銀行 名稱 bank Name	信用卡種 類 cardType	信用卡號 cardId	到期日 dueDate	會員編號 mId	購物車產生時間 cartTime	購物車之 交易編號 Cart_tNo
91100	a0911234	cart	2005-02-02:18:30:00	010	fb	visa	987654321	2008-01-01	a0911234	2005-02-02:18:00:30	91100
92666	e0927777	cart	2005-10-10:22:10:30	020	fb	visa	123456789	2006-12-31	e0927777	2005-10-10:22:00:00	92666
91888	a0910001	fax	2005-09-10:10:10:00	040	eb	master	77788888	2007-10-10	null	null	null
92333	e0927777	email	2005-10-15:09:00:00	070	sb	visa	111122222	2007-12-31	null	null	null
90111	b0905555	cart	2005-05-05:12:30:30	020	fb	master	444455555	2006-01-01	b0905555	2005-05-05:12:00:00	90111
92555	b0922468	cart	2005-11-11:09:10:00	010	fb	visa	533300000	2007-01-01	b0922468	2005-11-11:09:00:30	92555
null	null	null	null	null	null	null	null	null	a0910001	2005-09-09:10:00:10	null

©黃三益2007  
資料庫的心理理論與實務第三版

5-31

## 關聯代數運算式範例二

- 列出會員之會員編號和姓名，以及其所瀏覽過的商品名稱（如果有的話）
- 表示一會員若沒有瀏覽過任何商品，她的會員編號和姓名也要列出。本題應該用 **LEFT OUTER JOIN**

$BrowseProduct = Browse \times Product$

$MemberBrowse = Member \bowtie_{Member.mId=BrowseProduct.mId} BrowseProduct$

$Result = \pi_{Member\_mId, name, pName} MemberBrowse$

©黃三益2007  
資料庫的心理理論與實務第三版

5-32

## 關聯的分群和彙總函數

- 為滿足統計報表的功能，關聯代數提供運算子來計算彙總，一般式如下：

$$\text{Result} = \langle \text{分群屬性} \rangle \mathcal{F} \langle \text{彙總函數列} \rangle (\text{R})$$

- 關聯網目：名稱為Result，屬性為<分群屬性>加上<彙總函數列>
- 序列值：將R依<分群屬性>的屬性值分群，每一群序列值再依<彙總函數列>裡的函數運算成一序列值



©黃三益2007  
資料庫的核心理論與實務第三版

5-33

## 關聯的分群和彙總函數 (Cont.)

- 範例：

$$\text{Result}(\text{transNo}, \text{totalAmount}) = \text{tNo} \mathcal{F} \text{SUM salePrice} (\text{Record})$$

Result	transNo	totalAmount
	91100	450
	90111	1350
	92555	8000
	91888	2500
	92666	1450
	92333	700

$$\text{Result}(\text{totalProduct}, \text{averagePrice}) = \mathcal{F} \text{COUNT pNo, AVERAGE unitPrice} (\text{Product})$$

Result	totalProduct	averagePrice
	10	470

©黃三益2007  
資料庫的核心理論與實務第三版

5-34

## 關聯的分群和彙總函數

- 五個標準的彙總函數：
  - ☞ **SUM**：加總，其意義如上例所示。
  - ☞ **AVERAGE**：取數個屬性值的平均值。比如  
 $\text{catalog} \mathcal{F} \text{ AVERAGE unitPrice (Product)}$
  - ☞ **COUNT**：取屬性值的總個數。比如  
 $\text{catalog} \mathcal{F} \text{ COUNT pNo (Product)}$
  - ☞ **MAX**：取數個屬性值的最大值。比如  
 $\text{catalog} \mathcal{F} \text{ MAX unitPrice (Product)}$
  - ☞ **MIN**：取數個屬性值的最小值。比如  
 $\text{catalog} \mathcal{F} \text{ MIN unitPrice (Product)}$

©黃三益2007  
資料庫的核心理論與實務第三版

5-35

## 關聯代數運算式範例二

- 列出所有包括兩項以上商品的交易之交易編號和會員姓名

$T1(tNo, mId, pNo) = \pi_{tNo, transMid, pNo}(\text{Transaction} * \text{Record})$

$T2(tNo, mId, total) = \pi_{tNo, mId} \mathcal{F} \text{ COUNT pNo (T1)}$

$T3 = \sigma_{total > 2} (T2)$

$\text{Result} = \pi_{tNo, name} (T3 * \text{Member})$

©黃三益2007  
資料庫的核心理論與實務第三版

5-36

## 關聯代數運算式範例

- 列出每一位會員的會員編號、姓名，以及所瀏覽過的商品總數

**MemberBrowse(mld, name, pNo)=**

$\pi_{\text{Member\_mld, name, pNo}}(\text{Member} \bowtie_{\text{Member.mld=Browse.mld}} \text{Browse})$

**Result=** mld, name  $\sum_{\text{COUNT pNo}}(\text{MemberBrowse})$



## 關聯邏輯計算

- 關聯邏輯計算法 (Relational calculus) 是用邏輯計算方式來處理關聯
- 兩種形式：
  - ☞ 序列值邏輯計算 (tuple calculus)
    - 本章所採用
  - ☞ 定義域邏輯計算 (domain calculus)
- 邏輯計算方式與代數計算最大的不同點是邏輯計算式裡頭的子運算式是沒有次序性的。
  - ☞ 關聯代數是一種程序式的語言
  - ☞ 關聯邏輯計算法是一種宣告式的語言



## 關聯邏輯計算

- 序列值變數 (tuple variable) 是代表某個關聯裡的序列值。比如：

☞  $\{m \mid \text{Member}(m)\}$

- 代表Member關聯裡所有序列值所成的集合。

☞  $\{p \mid \text{Product}(p), p.\text{unitPrice} > 500\}$

- 代表Product關聯裡所有定價超過500的商品序列值所成的集合。

☞  $\{p.pNo, p.pName \mid \text{Product}(p), p.\text{unitPrice} > 500\}$

- 代表Product關聯裡所有定價超過500的商品之商品編號和名稱合成的序列值所成的集合。

☞  $\{t.tNo, t.mId, c.cartTime \mid \text{Transaction}(t), \text{Cart}(c), t.tNo = c.tNo\}$

- 代表有購物車的所有交易之交易編號，交易會員，和購物車時間組成序列值所成的集合。



## 關聯邏輯計算 (Cont.)

- 關聯邏輯計算式裡可以使用有兩個邏輯限制子：所有 (universal quantifier, 符號為 $\forall$ )和存在 (existential quantifier, 符號為 $\exists$ )

☞  $\forall(t)(P(t))$  表示所有對於任意一個序列值 $t$ ，條件式 $P(t)$ 必然成立。

☞  $\exists(t)(P(t))$  表示存在一個序列值 $t$ ，使得條件式 $P(t)$ 必然成立。

- 沒有被這兩個邏輯限制子所限制的變數稱為自由變數，需出現在關聯邏輯計算式的結果裡。



## 關聯邏輯計算 (Cont.)

- 找出所有有購物車的交易之交易編號和會員編號
  - ☞  $\{t.No, t.mId \mid Transaction(t), \exists (c)((Cart(c), c.tNo=t.tNo))\}$
- 找出所有沒有購物車的交易之交易編號和會員編號
  - ☞  $\{t.No, t.mId \mid Transaction(t), \forall (c)(\neg Cart(c) \text{ or } c.tNo \neq t.tNo)\}$   
或是
  - ☞  $\{t.tNo, t.transMid \mid Transaction(t), \neg (\exists c)( Cart(c) \text{ and } c.tNo=t.tNo)\}$
- 找出所有沒被編號為 'a0910001' 的會員所瀏覽過或購買過的商品之商品編號和名稱
  - ☞  $\{p.pNo, p.pName \mid Product(p), \neg((\exists b)(Browse(b), b.pNo=p.pNo, b.mId= 'a0910001' )), \neg(\exists t)(\exists r)(Transaction(t), Record(r), t.transMid = 'a0910001' , t.tNo=r.tNo, r.pNo=p.pNo)\}$