

例如：下面例子中：

S (Supplier)				SP (Supplier-parts)		
S#	SNAME	STATUS	CITY	S#	P#	QTY
S1	趙一	10	台北	S1	P1	100
S2	錢二	20	新竹	S1	P2	200
S3	孫三	30	台中	S1	P3	300
S4	李四	20	台中	S1	P4	200
S5	王五	10	台南	S1	P5	400
				S1	P6	100
				S2	P2	200
				S2	P3	300
				S3	P2	300
				S4	P2	200
				S4	P4	100
				S4	P5	100

P (Parts)				
P#	PNAME	COLOR	WEIGHT	CITY
P1	CPU	Blue	15	台北
P2	CRT	Red	16	新竹
P3	MB	Red	12	台中
P4	MONITOR	Red	14	台南
P5	CD	Green	17	台中
P6	HDD	Blue	19	台南

S ⋈ P 的結果如下：

S#	SNAME	STATUS	CITY	P#	PNAME	COLOR	WEIGHT
S1	趙一	10	台北	P1	CPU	Blue	15
S2	錢二	20	新竹	P2	CRT	Red	16
S3	孫三	30	台中	P3	MB	Red	12
S3	孫三	30	台中	P5	CD	Green	17
S4	李四	20	台中	P3	MB	Red	12
S4	李四	20	台中	P5	CD	Green	17
S5	王五	10	台南	P4	MONITOR	Red	14
S5	王五	10	台南	P6	HDD	Blue	19

(=) θ 合併 (θ -join)：若已給 A、B 兩個乘積相容的關聯式，則所謂 θ 合併 $A \bowtie_{\theta Y} B = \sigma_{\theta Y}(AXB)$ ，其中 θ 可以是任何條件運算，如 $>, =, <, \dots$ 同時若 θ 為 $=$ ，則稱為相等合併 (Equijoin)。

例如前面例子中，若將 P 中的 CITY 改名為 PCITY，

則 $A \bowtie_{CITY=PCITY} B$ 的結果如下：

S#	SNAME	STATUS	CITY	P#	PNAME	COLOR	WEIGHT	PCITY
S1	趙一	10	台北	P1	CPU	Blue	15	台北
S2	錢二	20	新竹	P2	CRT	Red	16	新竹
S3	孫三	30	台中	P3	MB	Red	12	台中
S3	孫三	30	台中	P5	CD	Green	17	台中
S4	李四	20	台中	P3	MB	Red	12	台中
S4	李四	20	台中	P5	CD	Green	17	台中
S5	王五	10	台南	P4	MONITOR	Red	14	台南
S5	王五	10	台南	P6	HDD	Blue	19	台南

九、除法 (Division) → 被除式中具有除式所有值的剩餘屬性

⊙ 定義：若已給 $A:(X_1, \dots, X_m, Y_1, \dots, Y_n)$ ， $B:(Y_1, \dots, Y_n)$ ，則所謂 $A \text{ DIVIDEBY } B$ (或寫作 $A \div B$)，其結果包含其對應的Y值在B中都能找到的由A取出的X值，如下例：

DEND		DNO	DEND DIVIDEBY DNO
S#	P#	P#	S#
S1	P1	P1	S1
S1	P2	P3	S2
S1	P3		
S1	P4		
S1	P5		
S1	P6		
.....		
S2	P1	P1	S1
S2	P3	P3	
S2	P4	P4	
S3	P1	P5	
S4	P1	P6	
S5	P4		

十、基本運算子 (Primitive operator)

(-) 定義：不能由其他運算子導出的運算子。

(=) 最初的關聯運算子中的基本運算子：

1. Restrict
2. Project
3. Union

4. Product

5. Difference

也有將此五個運算子所成的集合稱為完整集合 (Complete set)。

(三) 非基本運算子如何由基本運算子導出：

1. Join： $R \bowtie_{\text{條件}} S \equiv \sigma_{\text{條件}}(R \times S)$

2. Intersection：

$$R \cap S \equiv (R \cup S) - ((R - S) \cup (S - R))$$

$$\equiv R - (R - S)$$

$$\equiv S - (S - R)$$

3. Division： $R \div S \equiv \pi_Y(R) - \pi_Y((S \times \pi_Y(R)) - R)$

(四) 其他運算子：

1. 更名 (RENAME) 運算子表示成 ρ ，用來重新命名關聯名稱或屬性名稱。

例如： $\rho_{S(B_1, B_2, \dots, B_n)}(R)$ 表示關聯名稱變更為 S ，屬性名稱變更為 B_1, B_1, \dots, B_n 。

2. 聚合函數 (aggregate function) 運算子 \mathcal{F} ：

$\mathcal{F}_{\text{MAX Salary}}(\text{EMPLOYEE})$ 擷取出Employee關聯中薪資的最大值

$\mathcal{F}_{\text{MIN Salary}}(\text{EMPLOYEE})$ 擷取出Employee關聯中薪資的最小值

$\mathcal{F}_{\text{SUM Salary}}(\text{EMPLOYEE})$ 擷取出Employee關聯中Salary的總和

$\mathcal{F}_{\text{COUNT SSN}}(\text{EMPLOYEE})$ 計算員工人數

$\mathcal{F}_{\text{AVERAGE Salary}}(\text{EMPLOYEE})$ 計算員工的平均薪資

十一、關聯式代數的目的

(一) 提供一種將使用者需要符號化的高階表示法，以達成下列目的：

1. 定義查詢範圍。
2. 定義修改範圍。
3. 定義有命名的虛擬關聯式。
4. 定義快照 (Snapshot)。
5. 定義安全規則。
6. 定義穩定性需求。

7. 定義整合規則。

(二) 提供一個符號化的轉換規則：如前述非基本運算子的轉換。

(三) 作為最佳化的基礎。

十二、關聯式代數設計方法

(一) 找出目標屬性與關聯式。

(二) 找出限制條件、相關屬性與所屬關聯式。

(三) 關聯式是否相同？

1. 關聯式相同：採用Project與Select。

2. 關聯式不同：必須再考慮是否以主鍵Join？

(1) 以主鍵Join：採用直接Join。

(2) 不以主鍵Join：採用間接Join（必須找出中間關聯式）。

(四) 特殊類型：

1. 有「包含」的語意：採用除法。

2. 有「不」的語意：採用差集。

十三、關聯式代數查詢舉例（以P1中的三個關聯式為例）

(一) 找出供應零件P2的供應商姓名：

$\pi_{SNAME} \sigma_{P\#='P2'}(S \bowtie SP)$ 或是寫成

$((S \text{ JOIN } SP) \text{ WHERE } P\#='P2') [SNAME]$

(二) 找出至少供應一種綠色零件的供應商姓名：

$\pi_{SNAME}(\pi_{S\#}(\sigma_{COLOR='GREEN'}P) \bowtie SP) \bowtie S$ 或是寫成

$((P \text{ WHERE } COLOR='GREEN') \text{ JOIN } SP)[S\#] \text{ JOIN } S [SNAME]$

(三) 找出至少供應供應商S1所供應所有零件的供應商姓名：

$\pi_{SNAME}(((\pi_{S\#,P\#}SP) \div (\pi_{P\#}(\sigma_{S\#='S1'}SP))) \bowtie S)$ 或是寫成

$((SP[S\#,P\#] \text{ DIVIDEBY } P[P\#]) \text{ JOIN } S) [SNAME]$

(四) 找出供應所有零件的供應商姓名：

$\pi_{SNAME}(((\pi_{S\#,P\#}SP) \div (\pi_{P\#}P)) \bowtie S)$ 或是寫成

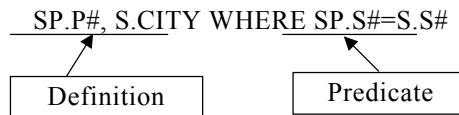
$((SP[S\#,P\#] \text{ DIVIDEBY } (SP \text{ WHERE } S\#='S1')[P\#])) \text{ JOIN } S [SNAME]$

(五) 找出不供應零件P1的供應商姓名：

$\pi_{SNAME}(((\pi_{S\#}S)-(\pi_{S\#}(\sigma_{P\#='P1'}SP)))) \bowtie S$ 或是寫成
 $((S[S\#] \text{ MINUS } (SP \text{ WHERE } P\#='P1'))[S\#]) \text{ JOIN } S$ [SNAME]

十四、關聯式計算法則簡介

(一)是一種表示符號：用一些已給的關聯式來表示一些新關聯式的定義。例如：



化成SQL就是：

```
SELECT SP.P#, S.CITY
FROM    S, SP
WHERE   SP.S# = S.S#
```

(二)Relationally complete：若在一種關聯式查詢語言L所能表示的所有查詢均能用關聯式計算法則表示，則稱此種語言為Relationally complete。

(三)關聯式計算法則的形式：

1. 值組導向的關聯式計算法則 (Tuple-oriented relational calculus)：其變數範圍是以值組（橫列）為主。
 2. 定義域導向的關聯式計算法則 (Domain-oriented relational calculus)：其變數範圍是以定義域中屬性的值（直行）為主。
- 以上兩者均基於First-order predicate calculus。

十五、值組關聯式計算法則的語法

範圍變數定義：

:: = RANGE OF 變數 IS 範圍項目列（以逗號隔開）；

範圍項目：

:: = 關聯式 | 表示式

表示式：

:: = (目標項目列(以逗號隔開)) [WHERE wff]