



關聯式代數簡介



關聯式代數

關聯式資料模式上的資料操作部份有：關聯式代數(Relational Algebra)與關聯式計算(Relational Calculus)兩種。

聯集運算(Union)

- 聯集運算(Union):此運算將兩個關聯表 R1 與 R2 聯集成一個新的關聯表，新的關聯表 R3 中的記錄為原來兩關聯表記錄的聯集。

R1				R2				R3		
A	B	C		A	B	C		A	B	C
<u>a1</u>	<u>b1</u>	<u>c1</u>	∪	<u>a1</u>	<u>b1</u>	<u>c1</u>	=	<u>a1</u>	<u>b1</u>	<u>c1</u>
<u>a3</u>	<u>b3</u>	<u>c3</u>		<u>a2</u>	<u>b2</u>	<u>c2</u>		<u>a2</u>	<u>b2</u>	<u>c2</u>
								<u>a3</u>	<u>b3</u>	<u>c3</u>

交集運算(Intersection)

- 交集運算(Intersection):此運算將兩個關聯表 R1 與 R2 交集成一個新的關聯表，新的關聯表 R3 中的記錄為原來兩關聯表記錄的交集。

R1				R2				R3		
A	B	C		A	B	C		A	B	C
<u>a1</u>	<u>b1</u>	<u>c1</u>	\cap	<u>a1</u>	<u>b1</u>	<u>c1</u>	=	<u>a1</u>	<u>b1</u>	<u>c1</u>
<u>a3</u>	<u>b3</u>	<u>c3</u>		<u>a2</u>	<u>b2</u>	<u>c2</u>				

差集運算(Difference)

- 差集運算(Difference):此運算將一個關聯表 R1 中的記錄減去另一個關聯表 R2 的記錄，形成新的關聯表 R3 的記錄。事實上差集的運算相當於將關聯表 R1 中之記錄減去 R1 與 R2 共有的記錄。

R1				R2				R3		
A	B	C		A	B	C		X	Y	C
<u>a1</u>	<u>b1</u>	<u>c1</u>	-	<u>a1</u>	<u>b1</u>	<u>c1</u>	=	<u>a3</u>	<u>b3</u>	<u>c3</u>
<u>a3</u>	<u>b3</u>	<u>c3</u>		<u>a2</u>	<u>b2</u>	<u>c2</u>				

乘積運算(Cartesian Product)

- 乘積運算(Cartesian Product):此運算是將兩關聯表 R1 與 R2 的記錄利用集合運算中的乘積運算形成新的關聯表 R3，假如 R1 有 r_1 個屬性， m 筆記錄，R2 有 r_2 個屬性， n 筆記錄，則 R3 會有 (r_1+r_2) 個屬性， $(m \times n)$ 筆記錄。

R1			R2				R3				
X	Y		A	B	C		X	Y	A	B	C
<u>x₁</u>	<u>y₁</u>	×	<u>a₁</u>	<u>b₁</u>	<u>c₁</u>	=	<u>x₁</u>	<u>y₁</u>	<u>a₁</u>	<u>b₁</u>	<u>c₁</u>
<u>x₂</u>	<u>y₂</u>		<u>a₂</u>	<u>b₂</u>	<u>c₂</u>		<u>x₂</u>	<u>y₂</u>	<u>a₁</u>	<u>b₁</u>	<u>c₁</u>
							<u>x₁</u>	<u>y₁</u>	<u>a₂</u>	<u>b₂</u>	<u>c₂</u>
							<u>x₂</u>	<u>y₂</u>	<u>a₂</u>	<u>b₂</u>	<u>c₂</u>

選擇運算(Select)

此運算是由關聯表 R 中選取符合條件的記錄。假設 P 為選取的條件，以 $\sigma_p(R)$ 代表此運算。其結果為原關聯表 R 記錄的子集合。例如： $\sigma_{sal>1600}(\text{Emp})$ 為從關聯表 Emp 中選取符合屬性 Sal 大於 1600 的記錄，如下圖陰影部分所示：

<u>Empno</u>	<u>Ename</u>	...	Sal	<u>Deptno</u>
7369	SMITH	...	800	20
7499	FORD	...	1800	10
...
7698	BLAKE	...	2850	40
7566	JONES	...	2975	10
7839	KING	...	3975	40

R1 ↓
(員工基本資料表格 EMP)

投影運算(Project)

此運算是由關聯表中選出所需的關聯表屬性。假設關聯表 R 中欲選出的屬性列表為 $A_1, A_2, A_3, \dots, A_n$ ，則以 $\pi_{A_1, A_2, A_3, \dots, A_n}(R)$ 表示此投影運算。例如： $\pi_{Empno, Ename, Sal}(Emp)$ 則為由關聯表 Emp 中的 Empno, Ename 與 Sal 屬性中選出這些屬性的記錄，如下圖陰影部份所示：

<u>Empno</u>	<u>Ename</u>	...	Sal	<u>Deptno</u>
7369	SMITH	...	800	20
7499	FORD	...	1800	10
...
7698	BLADE	...	2850	40
7566	JONES	...	2975	10
7839	KING	...	3975	40

R1

(員工基本資料表格 EMP)

數位教材-資料庫系統設計與應用

關聯運算(Join)

此運算將兩個關聯表 R1 與 R2 依關聯條件合併成一個新關聯表 R3，假設 P 為關聯條件，以 $R1 \bowtie_P R2$ 表示此關聯運算。而關聯的條件 P 的型式為： $A1 \theta A2$ ，其中 A1 為 R1 的屬性，A2 為 R2 的屬性， θ 為比較運算子，例如： $<$ 、 $>$ 、 $=$ 、 \leq 、 \geq 、 \neq 等等。關聯運算會因為 θ 不同而產生不同的結果。一般我們稱此為「 θ -關聯」(θ -join)。以下面兩個關聯表 Emp 與 Dept 為例：

<u>Empno</u>	<u>Ename</u>	...	<u>Deptno</u>
7369	SMITH	...	20
7499	ALLEN	...	10
7521	WARD	...	30
7566	JONES	...	40

R1 (員工基本資料表格 EMP)

<u>Deptno</u>	<u>Dname</u>	<u>Loc</u>
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

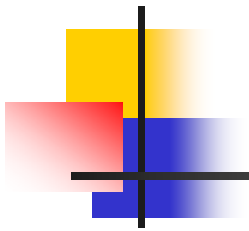
R2 (部門基本資料表格 DEPT)

則 ↵

Emp ⋈ Emp.deptno=Dept.deptno Dept ↵

的結果如下： ↵

<u>Empno</u>	<u>Ename</u>	...	<u>Emp.deptno</u>	<u>Dept.deptno</u>	<u>Dname</u>	<u>Loc</u>
7369	SMITH	...	20	20	RESEARCH	DALLAS
7499	ALLEN	...	10	10	ACCOUNTING	NEW YORK
7521	WARD	...	30	30	SALES	CHICAGO
7566	JONES	...	40	40	OPERATIONS	BOSTON



上面例子的關聯條件是利用等號“=”，所以一般也稱為「等於關聯」(Equijoin)。由於「等於關聯」通常會造成新的關聯表中有兩個一模一樣的欄位資料。但是在實務應用上是多餘的，因此可以將其中的一個欄位刪除，而此種關聯運算稱為「自然關聯」(Natural Join)，其結果如下：

<u>Empno</u>	<u>Ename</u>	...	<u>Deptno</u>	<u>Dname</u>	<u>Loc</u>
7369	SMITH	...	20	RESEARHD	DALLAS
7499	ALLEN	...	10	ACCOUNTING	NEW YORK
7521	WARD	...	30	SALES	CHICAGO
7566	JONES	...	40	OPERATIONS	BOSTON



除法運算(Divide)

- 此種運算如同數學上的除法一般，有二個運算元：第一個關聯表 R1 當作「被除表格」，第二個關聯表 R2 當作「除表格」。其中「被除表格」的屬性必須比「除表格」多，而且「除表格」中的任何屬性的值域都要與「被除表格」中的某屬性之值域相符合。