關聯式代數特性

關聯式代數特性

- 例如：前面介紹的除法運算，可以利用等價轉換而成為一連串的投影運算、乘積運算與差集運算，這種關聯式代數的轉換特性主要可以分成結合律 (Associativity Rule)、交換律 (Commutativity Rule) 與分配律 (Distributivity Rule)。

結合律(Associativity Rule)

下列的二元運算子具有結合律：('≡' 爲等價轉換)

1. $(A \cup B) \cup C \equiv A \cup (B \cup C)$, 假設 A、B 與 C 滿足「型態相容」條件

2. $(A \times B) \times C \equiv A \times (B \times C)$

3. $(A \cap B) \cap C \equiv A \cap (B \cap C)$, 假設 A、B 與 C 滿足「型態相容」條件

4. $(A \bowtie_p B) \bowtie_q C \equiv A \bowtie_p (B \bowtie_q C)$

交換律(Commutativity Rule)

以下的二元運算子具備交換律：

1. $A \cup B \equiv B \cup A$, 假設 A 與 B 滿足「型態相容」條件

2. $A \cap B \equiv B \cap A$, 假設 A 與 B 滿足「型態相容」條件

3. $A \times B \equiv B \times A$

4. $\sigma_{C1}(\sigma_{C2}(R)) \equiv \sigma_{C2}(\sigma_{C1}(R))$, C1 與 C2 爲條件

例：$\sigma_{EMPNO=7566}(\sigma_{SAL>1200}(R)) \equiv \sigma_{SAL>1200}(\sigma_{EMPNO=7566}(R))$

交換律(cont.)

5. $\pi_{C1,C2,\ldots,Cm}(\sigma_P(R)) \equiv \sigma_P(\pi_{C1,C2\ldots,Cm}(R))$, where $P \in \{C1, C2, \ldots, Cm\}$

例：$\pi_{ENAME,DEPTNO}(\sigma_{DEPTNO=20}(EMP)) \equiv \sigma_{DEPTNO=20}(\pi_{ENAME,\ DEPTNO}(EMP))$

6. $A \bowtie_p B \equiv B \bowtie_p A$

7. $\sigma_p(A \cup B) \equiv \sigma_p(B) \cup \sigma_p(A)$, 假設 A 與 B 滿足「型態相容」條件

例：$\sigma_{SAL>2400}(ACCOUNT \cup RESEARCH)$

$\equiv (\sigma_{SAL>2400}(ACCOUNT)) \cup (\sigma_{SAL>2400}(RESEARCH))$

交換律(cont.)

- 8. $\sigma_p(A \cap B) \equiv \sigma_p(B) \cap \sigma_p(A)$, 假設 A 與 B 滿足「型態相容」條件

  例：$\sigma_{SAL>2400}$ (ACCOUNT $\cap$ RESEARCH)

  $\equiv (\sigma_{SAL>2400}(\text{ACCOUNT})) \cap (\sigma_{SAL>2400}(\text{RESEARCH}))$

  9. $\pi_{C1}(A \cup B) \equiv \pi_{C1}(B) \cup \pi_{C1}(A)$, 假設 A 與 B 滿足「型態相容」條件

  例：$\pi_{SAL>2400}$ (ACCOUNT $\cup$ RESEARCH)

  $\equiv (\pi_{SAL>2400}(\text{RESEARCH})) \cup (\pi_{SAL>2400}(\text{ACCOUNT}))$

交換律(cont.)

但是值得注意差集運算並不具備交換律：

$$A-B \neq B-A$$

10. $\pi_{C1}(A \cap B) \equiv \pi_{C1}(B) \cap \pi_{C1}(A)$, 假設 A 與 B 滿足「型態相容」條件

例：$\pi_{SAL>2400}$ (ACCOUNT $\cap$ RESEARCH)

$\equiv (\pi_{SAL>2400}(\text{RESEARCH})) \cap (\pi_{SAL>2400}(\text{ACCOUNT}))$

11. $\pi_{A1,A2\ldots,Am,B1,B2\ldots,Bn}(R1 \times R2) \equiv \pi_{A1,A2\ldots,Am}(R1) \times \pi_{B1,B2\ldots,Bn}(R2)$，A1, A2…, Am 為 R1 的屬性，B1, B2…,Bn 為 R2 的屬性。

分配律(Distributivity Rule)

1. $\sigma_{C1 \text{ and } C2}(R) \equiv \sigma_{C1}(R) \cap \sigma_{C2}(R) \equiv \sigma_{C1}(\sigma_{C2}(R))$

   例：$\sigma_{EMPNO=7566 \text{ AND } SAL>1200}(EMP) \equiv \sigma_{EMPNO=7566}(\sigma_{SAL>1200}(EMP))$

2. $\sigma_{C1 \text{ OR } C2}(R) \equiv \sigma_{C1}(R) \cup \sigma_{C2}(R)$

   例：$\sigma_{EMPNO=7566 \text{ OR } SAL>1200}(EMP)$

   $= (\sigma_{EMPNO=7566}(EMP)) \cup (\sigma_{SAL>1200}(EMP))$

分配律(cont.)

3. $\sigma_{NOT(C)}(R) \equiv R-\sigma_C(R)$

4. $(A \cup B) \bowtie_P (C \cup D) \equiv (A \bowtie_P C) \cup (A \bowtie_P D) \cup (B \bowtie_P C) \cup (B \bowtie_P D)$

5. 假設 R1 的屬性有 (X, Y) 而 R2 的屬性有 (Y, Z) 則

$$\sigma_{X\theta C1 \text{ AND } Z\theta C2}(R1 \bowtie_q R2) \equiv (\sigma_{X\theta C1}(R1)) \bowtie_q (\sigma_{Z\theta C2}(R2))$$

6. $\sigma_p(A-B) \equiv \sigma_p(A)-\sigma_p(B)$, 假設 A 與 B 滿足「型態相容」條件

關聯式代數的最佳化

- 通常最佳化查詢的原則是將二元運算，例如：關聯運算、除法運算等，延遲到越後面做越好，儘量讓選擇運算或投影運算先過濾掉不需要的值組，以便執行二元運算。

關聯式代數的最佳化(cont.)

- (1) $\sigma_{C1 \text{ and } C2}(R) \equiv \sigma_{C1}(\sigma_{C2}(R))$

說明 先利用能過濾較多值組的條件C2 對 R 做篩選($\sigma_{C2}(R)$)，去掉不合乎條件 C2 的大部份值組，因而得到較少資料量的 R_1，使得再做 $\sigma_{C1}(R_1)$ 時，需要篩選的資料量變少了，因而會有較好的效能。

關聯式代數的最佳化(cont.)

- (2) $\sigma_P(\pi_{C1,C2\ldots,Cm}(R)) \equiv \pi_{C1,C2\ldots,Cm}(\sigma_P(R))$

說明 利用上面的等價轉換使得 $\sigma_P(R)$ 先做篩選，去掉不合乎條件 P 的值組，如此可以得到較少資料量的中間結果，這樣對後面的查詢操作會有較好的效能。

關聯式代數的最佳化(cont.)

(3) $\sigma_{X\theta C1 \text{ AND } Z\theta C2}(R1 \bowtie_q R2) \equiv (\sigma_{X\theta C1}(R1)) \bowtie_q (\sigma_{Z\theta C2}(R2))$, 假設 R1 的屬性有 (X, Y) 而 R2 的屬性有 (Y, Z)

說明 利用上面的等價轉換對 R1 與 R2 先做篩選，去掉不合乎條件的值組，如此可以個別得到較少資料量的中間結果，這樣再做關聯會有較好的效能。

關聯式代數的最佳化(cont.)

- (4) $\sigma_p(A-B) \equiv \sigma_p(A)-\sigma_p(B)$, 假設 A 與 B 滿足「型態相容」條件

說明 利用上面的等價轉換對 A 與 B 先做篩選，去掉不合乎條件的值組，如此可以個別得到較少資料量的中間結果，這樣再做差集會有較好的效能。

關聯式代數的最佳化(cont.)

- (5) $\sigma_p(A \cup B) \equiv \sigma_p(B) \cup \sigma_p(A)$, 假設 A 與 B 滿足「型態相容」條件

說明 利用上面的等價轉換對 A 與 B 先做篩選，去掉不合乎條件的值組，如此可以個別得到較少資料量的中間結果，這樣再做聯集會有較好的效能。

(6) $\sigma_{C1}(R) \cup \sigma_{C2}(R) \equiv \sigma_{C1 \text{ OR } C2}(R)$

說明 如果能使用 OR 達成查詢，就儘量不要用 UNION。

增進效能的方法

1. 避免在查詢中使用函數，特別是包含索引列，因爲函數的存在將會關閉索引。
2. 重覆看看在 WHERE 子句中的每一列以決定是否適當的用了索引，如果一個小型的表格或者 SELECT 指令接收到大量的資料時，這時使用索引並不是一個明智的做法，而且你甚至應該停止索引或把它們結束掉。

增進效能的方法(cont.)

3. 設法避免在 LIKE 函數時使用萬用字元 % 符號或是任一字元 _，例如： ENAME LIKE ‘M%’。因爲 LIKE 函數會造成完全的表格掃描。儘量用較小且較明確的範圍做查詢，例如：ENAME BETWEEN ‘Mad’ AND ‘Max’。再者別在索引列中使用計算式，因爲這將會關閉掉索引的使用。

增進效能的方法(cont.)

4. 不要使用非必須的子查詢。
5. 不要對非必要的欄位發出要求。例如：使用“ * ”符號以選擇所有的欄位，這通常不是好的做法。
6. 不要對你用不到的列發出要求，所以請謹慎使用 WHERE 子句做篩選。同樣的，只有在你必須要做群組總結時，才使用 ORACLE 群組函數中提供的功能，例如：AVG 或是 MAX。

增進效能的方法(cont.)

7. 善用狄摩根定律(DeMorgan’s Law)，例如：NOT (SAL > 1200 AND JOB = ‘SALESMAN’) 可以轉換成 (SAL <= 1200 AND JOB<> ‘SALESMAN’)。
8. 把條件中 “…> ALL(…)” 改成“…> MAX(…)”，例如：SAL > ALL (SELECT SAL FROM …) 可改成 SAL > MAX (SELECT SAL FROM …)。

增進效能的方法(cont.)

9. 建議可以選擇利用 ”IN 子查詢” 來代替 “exists 子查詢”。
10. 建議可以將 “IN 條件範圍”轉成多個利用 OR 的個別條件，例如：” DEPTNO IN (10, 20, 30)” 可改成 “DEPTNO = 10 OR DEPTNO = 20 OR DEPTNO = 30”。

關聯式代數運算後的主鍵

1. $\pi_{X1,\ X2,\ X3....Xn}(R)$：若投影運算後的結果有包含原 R 中的主鍵 Xi 則 $\pi_{X1,\ X2,\ X3.....Xn}(R)$ 的主鍵仍為 Xi。否則，主鍵由投影出的屬性 (X1, X2, X3......Xn)中選出。

2. $\sigma_{p}(R)$：主鍵仍為原 R 中的主鍵，我們以 PK_R 表示。

3. $A{\times}B$：假設 PK_A 為 A 的主鍵，PK_B 為 B 的主鍵，則 $A{\times}B$ 的主鍵為 (PK_A, PK_B) 組成。

關聯式代數運算後的主鍵(cont.)

4. $A \cap B$：因爲 A 與 B 爲「型態相容」，所以 PK_A 與 PK_B 相同，因此主鍵爲 PK_A 或 PK_B。

5. $A \cup B$：主鍵爲 PK_A 或 PK_B (PK_A 與 PK_B 相同)。

6. A-B：主鍵爲原關聯表 A 的主鍵 PK_A。

聯式代數運算後的主鍵(cont.)

7. $A \bowtie_p B$：假設 PK_A 為 A 的主鍵，PK_B 為 B 的主鍵，則 $A \bowtie_p B$ 的主鍵為 (PK_A, PK_B) 所組成。

8. $A \div B$：主鍵為所有的屬性所構成。