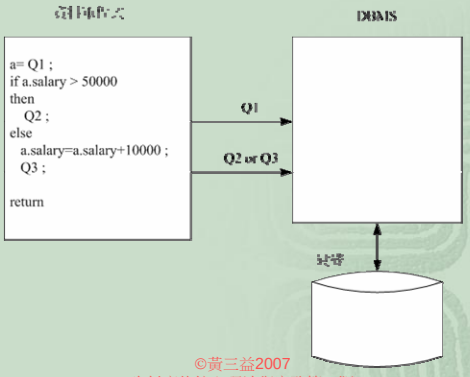
第十章基本的查詢處理與最佳化

- 資料庫程式的執行
- SQL敘述的處理流程
- SQL查詢樹
- 基本關聯代數運算子的處理
  - SELECT的處理方式和成本
  - 外部排序的處理方式和成本

資料庫程式的執行

- 通常SQL的敘述都是由程式執行所產生，但交由DBMS來處理
- DBMS看到的是一串SQL敘述



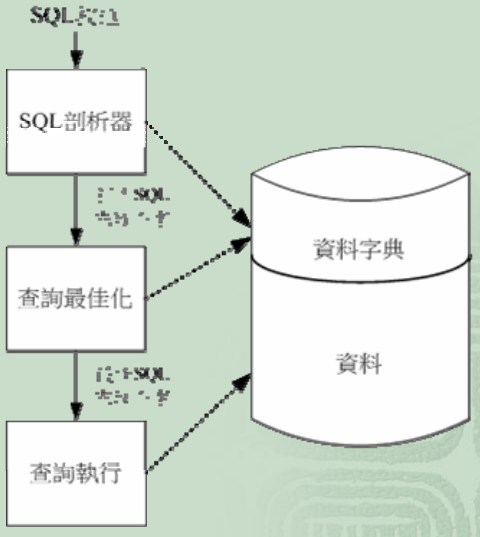
資料庫程式的部分程式碼

```
'建立資料庫連結物件
1. set conn =
   Server.CreateObject("ADODB.Connection")
'   開啟資料庫連結
2. conn.Open "onlinedb"
3. query = "SELECT * FROM product"
4. Set rs = conn.Execute(query)
5. while not rs.EOF
```

下達查詢並取得結果

一筆一筆取出結果

SQL敘述的處理流程



練習10-1

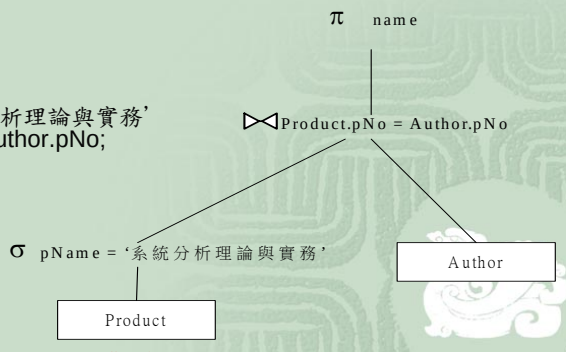
- 考慮圖10-2，如果第4行的SQL指令在檢查時發現錯誤，會有什麼後果？
- Ans:

  此時該SQL指令便不會執行，也因此rs裡不會有值。所以不會執行WHILE迴圈

SQL查詢樹

- 一顆SQL查詢樹是用來表達一種執行方案
  - 每一葉節點記錄查詢所用到的每一資料表
  - 每一中間節點記載處理的動作。標準的處理動作如關聯代數裡的運算子

```
SELECT name
FROM  Product, Author
WHERE pName =  '系統分析理論與實務'
    AND  Product.pNo = Author.pNo;
```



SQL查詢樹（Cont.)

- 查詢樹的執行次序是由下而上、由左至右
  - 上例的執行方式如下

    Temp1=$\sigma_{pName='系統分析理論與實務'}$ (Product);

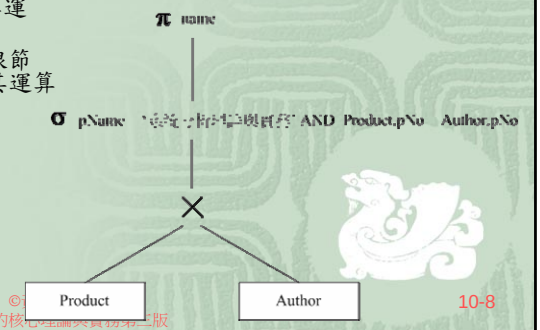
    Temp2= Temp1 $\bowtie_{Temp1.pNo=Author.pNo}$ Author;

    Result= π_{name}(Temp2);
- SQL剖析器所產生的查詢樹是最簡單的查詢樹，稱為初始查詢樹
- 查詢最佳化模組會將初始查詢樹轉換成較有效率的查詢樹

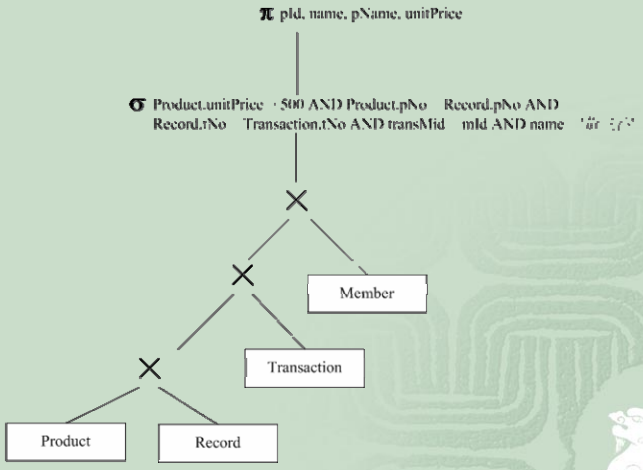
SQL查詢樹（Cont.)

- 從SQL查詢句建立初始查詢樹
  - **FROM**子句裡的每一個資料表是一個葉節點。
  - 葉節點用集合乘法（卡迪森乘積）當中間節點兩兩串連起來。
  - 加上一個**SELECT(σ)**的中間節點，以**WHERE**子句當作其運算。
  - 加上一個**PROJECT(π)**的根節點，以**SELECT**子句當作其運算

```
SELECT name
FROM  Product, Author
WHERE pName = '系統分析理論與實務'
  AND  Product.pNo = Author.pNo;
```

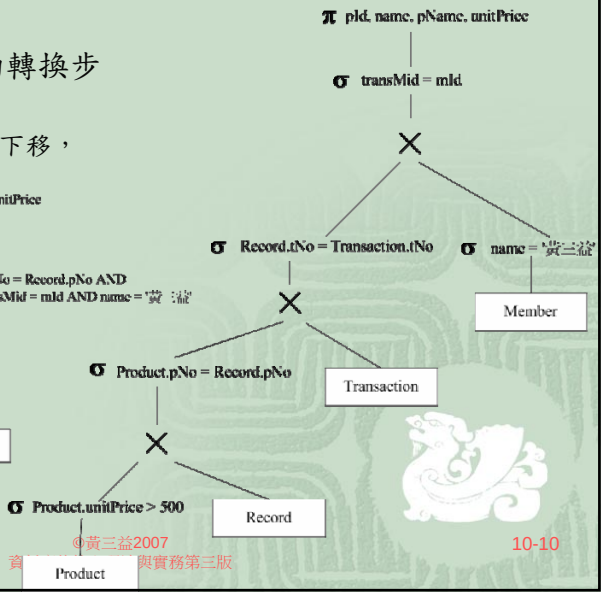


初始查詢樹



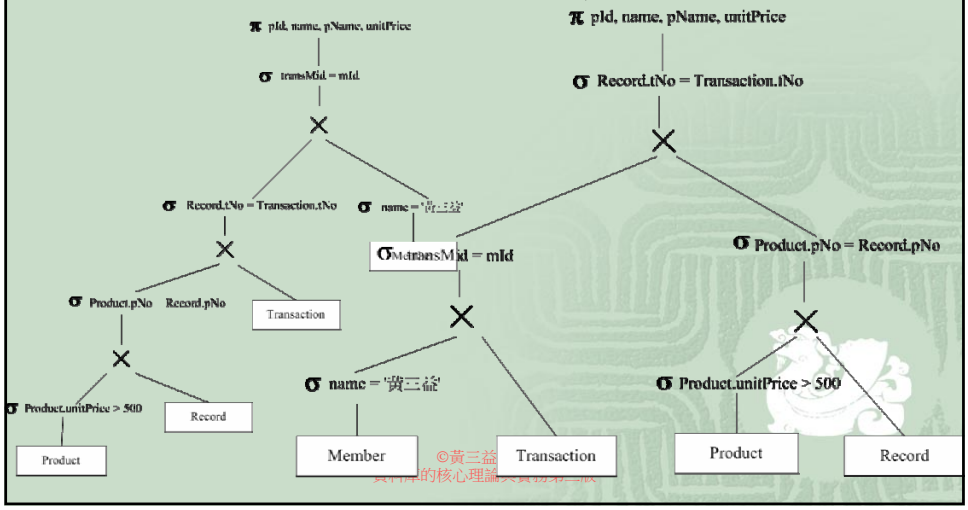
SQL查詢樹（Cont.)

- 初始查詢樹最佳化的轉換步驟
  - 將**SELECT**的動作往下移，盡量接近葉節點



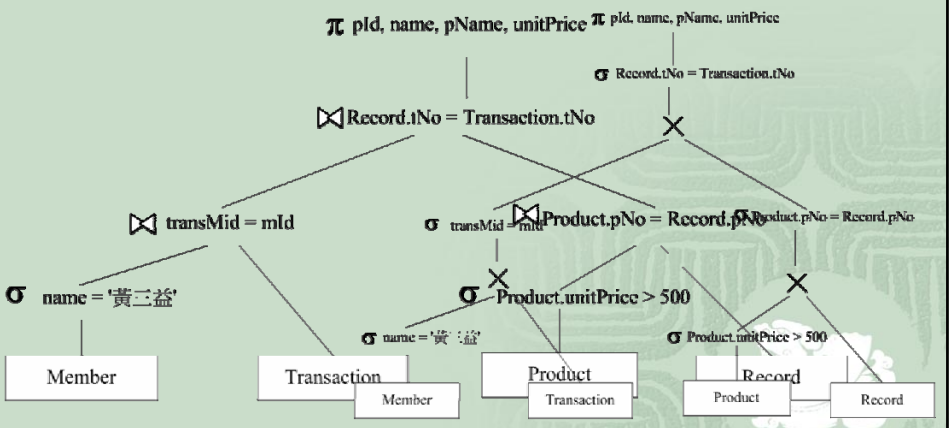
SQL查詢樹（Cont.）

- 將條件較嚴格的SELECT中間節點盡量往左邊移



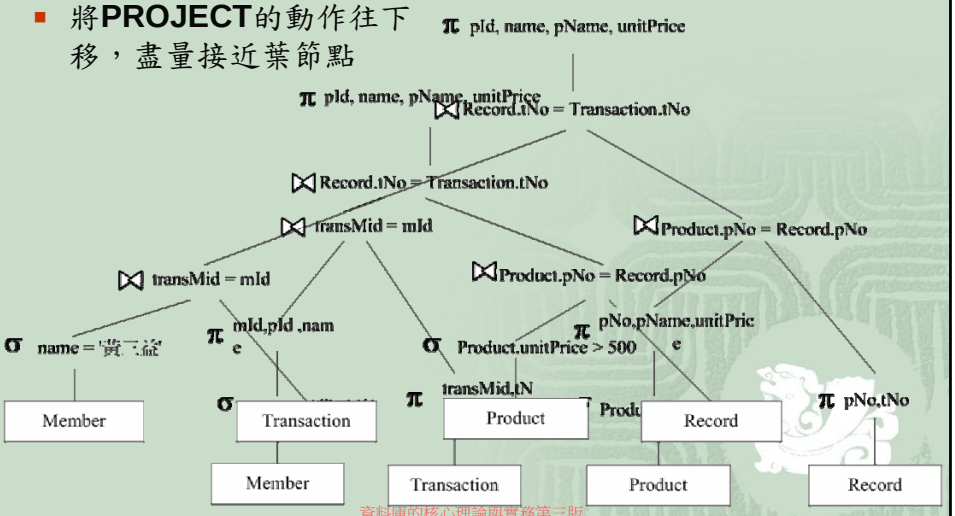
SQL查詢樹（Cont.）

- 將相鄰的×中間節點 和 σ 中間節點合併成一個 ⋈中間節點



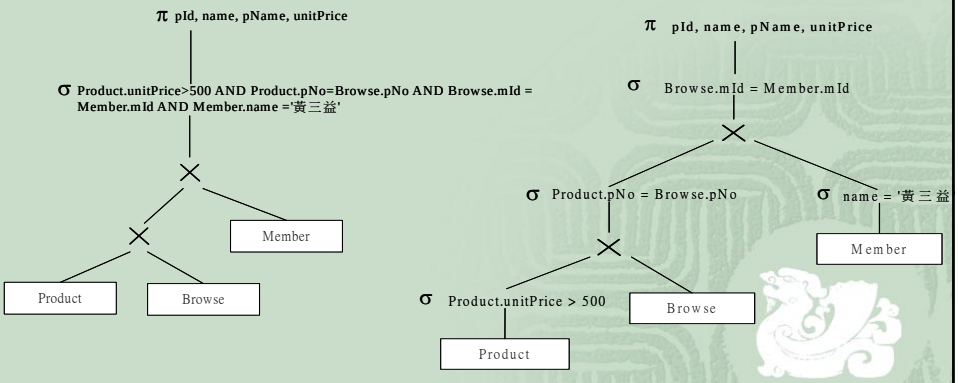
SQL查詢樹（Cont.)

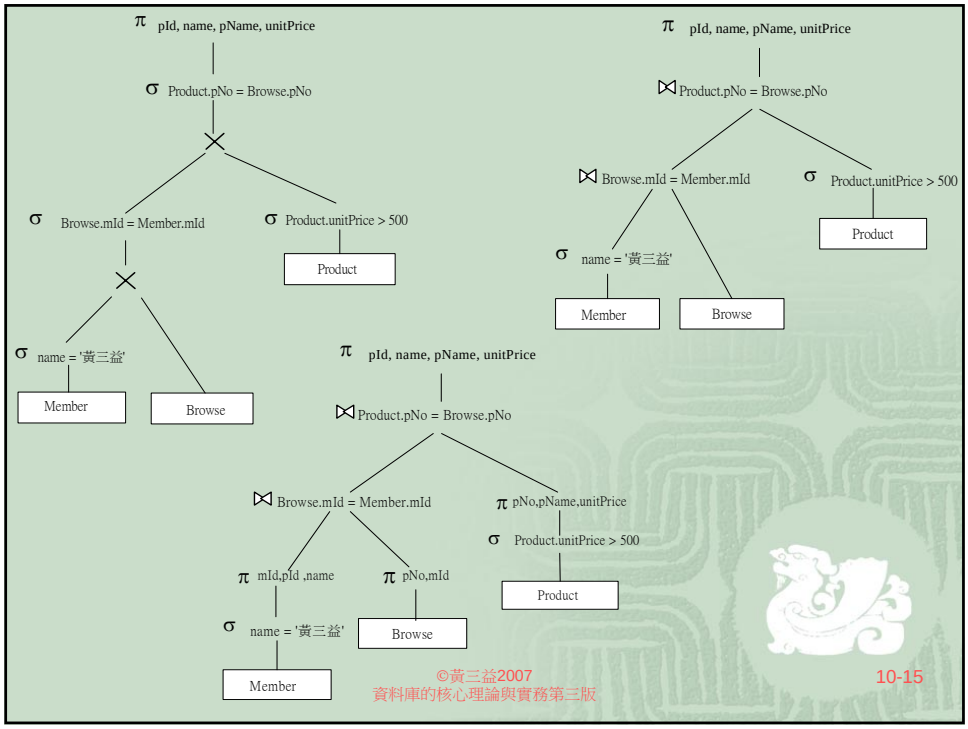
- 將**PROJECT**的動作往下移，盡量接近葉節點



練習10-2

- 找出「黃三益」所瀏覽過單價超過500元的商品資訊的最佳化查詢





查詢成本的預估指標

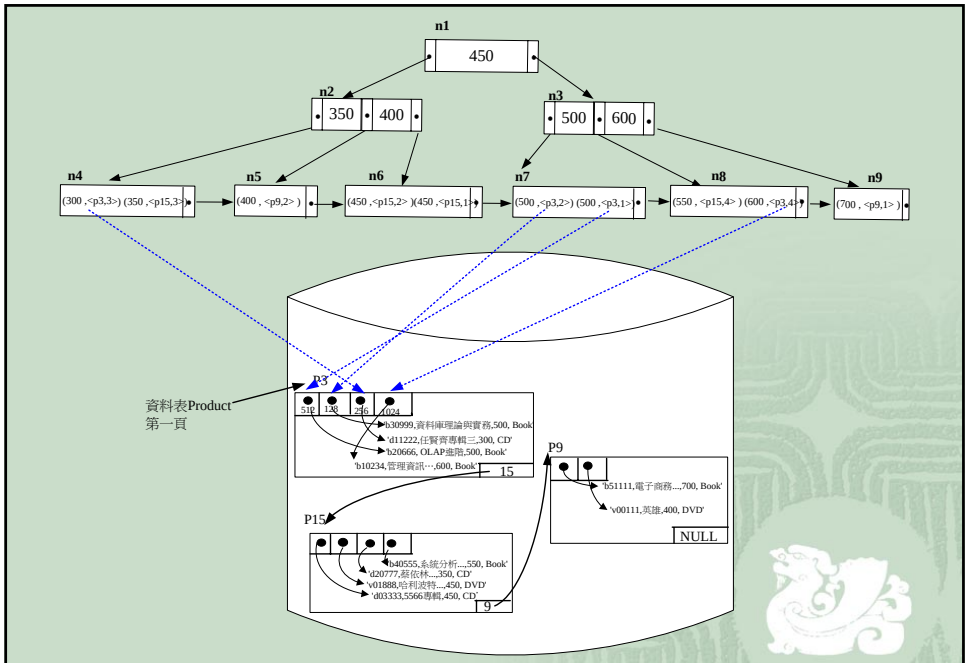
- 查詢樹的每一中間節點，實作的方式可能有數種，到底該採取哪一種？
- 沒有哪一種處理方式必然可以得到比較有效率的運算
- 對於一個查詢句，現代的DBMS於是採用**成本預估**（Cost estimate）的方式來決定該採取哪一種處理方式
- 查詢最佳化模組試圖計算出和比較它們的「成本」

何謂運算成本

- 何謂成本：執行時間
  - 硬碟的存取成本 → 佔大部分時間
    - 大部分DBMS的瓶頸
  - CPU的計算成本
    - 主記憶體DBMS的瓶頸
  - 網路的通訊成本
    - 分散式DBMS的成本

資料表和索引的相關符號

- 資料表的相關數據
  - **r**：資料表裡的記錄筆數
    - $r_{Product} = 100{,}000$
  - **b**：資料表所佔的資料頁數
    - $b_{Product} = 5000$
  - **bfr**：每一資料頁可容納幾筆記錄
    - $r_{Product} / b_{Product} = bfr_{Product} = 20$
- 索引的相關數據
  - **x**：B^+-tree的層數
    - $x_{unitPrice} = 3$
  - **bl1**：B^+-tree裡葉節點的個數
    - $bl1_{unitPrice} = 500$
  - **d**：不同索引值的個數
    - $d_{catalog} = 100$, $d_{SEX} = 2$
- 參考下頁圖9-6



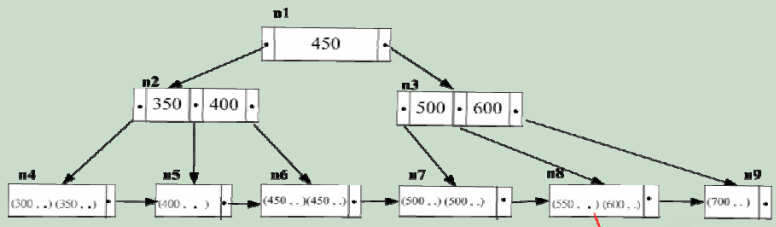
基本SELECT的處理方式

- **SELECT**條件裡只有單一屬性
  - $\sigma_{catalog='Book'}$ Product
  - $\sigma_{unitPrice>500}$ Product
  - $\sigma_{pNo='b30999'}$ Product
- 三種處理方式
  - （SL）資料頁循序搜尋
  - （SI）利用索引結構（參考圖9-6）
  - （SIC）利用群聚索引結構

- 假設有unitPrice的群聚索引，參考下頁圖10-7

```
CREATE INDEX  I3
ON  Product(unitPrice)
CLUSTER;
```

- 利用該索引結構來處理$\sigma_{unitPrice>500}$ Product 非常有效率
  - 利用該索引結構找到包含unitPrice>500 的資料頁指標
  - 到該資料頁和以下資料頁找出所有記錄



Product

'd11222, 任賢齊專輯三, 300, CD'
'd20777, 蔡依林專輯二, 350, CD'
'v00111, 英雄,400, DVD'
'v01888, 哈利波特：混血王子的背叛, 450, DVD'

p 1

'd03333, 5566專輯, 450, CD'
'b30999, 資料庫理論與實務, 500, Book'
'b20666, OLAP進階, 500, Book'
'b40555, 系統分析理論與實務, 550, Book'

p 2

'b10234, 管理資訊系統, 600, Book'
'b51111,電子商務理論與實務, 700, Book'

p 3

SELECT的選擇幅度

- SELECT的選擇幅度即是查詢結果的預估資料筆數，以s表示
- $\sigma_{catalog='Book'}Product$
  - $d_{catalog}$= 100，$r_{Product}$ = 100,000
  - $s=r_{Product} / d_{catalog}$ = 1000
- $\sigma_{pNo='b30999'}Product$
  - s=1
- $\sigma_{unitPrice > 500}Product$
  - $s=r_{Product}/2$ = 50,000

練習10-3

- 考慮圖9-6的索引結構，要列出所有unitPrice>500的Product記錄，請問需造訪哪些索引頁？會造訪哪些資料頁？
- Ans:
  - 索引頁：n1, n3, n8, n9
  - 資料頁：p15, p3, p9

練習10-4

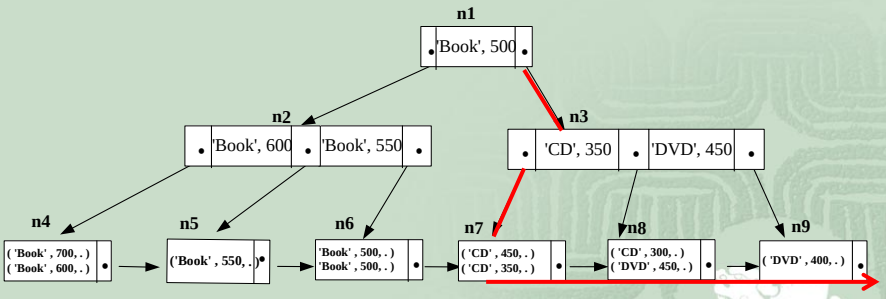
- 考慮圖10-7的索引結構，要列出所有unitPrice>500的Product記錄，請問需造訪哪些索引頁？會造訪哪些資料頁？
- Ans:
  - 索引頁：n1, n3, n8
  - 資料頁：p2, p3

複合SELECT的處理方式

- **SELECT**條件裡包括一個以上的基本子條件，這些子條件用**AND**或**OR**連結起來
  - $\sigma_{catalog='Book' \text{ AND } unitPrice<500}$ Product
  - $\sigma_{catalog='Book' \text{ OR } unitPrice<500}$ Product
- 有以下的四種作法
  - （SL）：資料頁循序搜尋
  - （SSI）：單一索引結構搜尋（參考圖9-6）
  - （SMI）：多索引結構搜尋
  - （SCI）：複合索引結構搜尋
    - 假設有一**多屬性值索引**：（catalog, unitPrice）
    - 參考下頁圖9-7

複合SELECT的處理方式(Cont.)

CREATE INDEX I2
ON Product(catalog **ASC**, unitPrice **DESC**);



$\sigma_{\text{catalog='Book' AND unitPrice<500}}$ **Product**

練習10-5：

- 考慮圖9-7的（catalog, unitPrice）索引結構，要列出所有catalog = 'Book' AND unitPrice=500的Product記錄，請問需造訪哪些索引頁？
- Ans:

  n1, n2, n6

SELECT處理方式的成本推估

- $\sigma_{A=a}R$
  - （SL）資料頁循序搜尋
    - case 1: A為R的關聯鍵：（$1+b_R$）/2
    - case 2: A不為R的關聯鍵：b_R
  - （SI）利用索引結構搜尋，A上建有一B^+-tree
    - case 1: A為R的關聯鍵：x_A+1
    - case 2: A不為R的關聯鍵：$x_A-1+\left\lceil bI1_A \cdot \frac{s_A}{r_R} \right\rceil + s_A$
  - （SIC）利用群聚索引結構搜尋，A上有建置一群聚索引
    - $x_A+\left\lceil \frac{s_A}{bfr_R} \right\rceil$

範例一：

- 假設$r_{Product}$ = 100,000，$b_{Product}$ = 5000，$bfr_{Product}$ = 20，x_{pNo}=3，計算$\sigma_{pNo='b30999'}$ Product的成本如下（請注意pNo為Product關聯的主鍵）
  - (SL): $(1+b_{Product})/2 \approx 2500$
  - (SI): $x_{pNo}+1=3+1=4$
  - (SIC): $x_{pNo}+1=3+1=4$

範例二

- 假設$r_{Product}$ = 100,000，$b_{Product}$ = 5000（因此$bfr_{Product}$ = 20)，$x_{catalog}$=2，$d_{catalog}$=100（因此$s_{catalog}$ = 1000），$bI1_{catalog}$=100，計算$\sigma_{catalog='Book'}$Product的成本如下：
  - (SL): $b_{Product}$=5000
  - (SI): $x_{catalog} - 1 + \left\lceil bI1_{catalog} \cdot \frac{s_{catalog}}{r_{Product}} \right\rceil + s_{catalog}$

    $=2 - 1 + 1 + 1000$

    $=1002$
  - (SIC): $x_{catalog} + \left\lceil \frac{s_{catalog}}{bfr_{Product}} \right\rceil = 2+50=52$

  （假設catalog為群聚索引）

SELECT處理方式的成本推估(Cont.)

- $\sigma_{A>a}R$
  - （SL）資料頁循序搜尋

    b_R
  - （SI）利用索引結構搜尋

    $x_A - 1 + \lceil \frac{bI1_A}{2} \rceil + \frac{r_R}{2}$
  - （SIC）利用群聚索引結構，A上有建置一群聚索引

    $x_A + \frac{b_R}{2}$

範例三

- 假設$r_{Product}$ = 100,000，$b_{Product}$ = 5000（因此$bfr_{Product}$ = 20），$x_{unitPrice}$=3，$bI1_{unitPrice}$=1000，計算$\sigma_{unitPrice > 500}$Product的成本如下
  - (SL): $b_{Product}$=5000
  - (SI): $x_{unitPrice} - 1 + \frac{bI1_{unitPrice}}{2} + \frac{r_{Product}}{2} = 2 + 500 + 50000 = 50502$
  - (SIC): $x_{unitPrice} + \frac{b_{Product}}{2} = 3 + 2500 = 2503$（假設unitPrice為群聚索引）

範例四

- 假設$r_{Product}$ = 100,000，$b_{Product}$ = 5000（因此$bfr_{Product}$ = 20），有三個索引：catalog, unitPrice，和（catalog, unitPrice）。$x_{catalog}$=2，$d_{catalog}$=100（因此$s_{catalog}$ = 1000），$bI1_{catalog}$=100，$x_{unitPrice}$=3， $bI1_{unitPrice}$=1000，$x_{catalog,unitPrice}$=4，$bI1_{catalog,unitPrice}$=2000，這三個索引皆非群聚索引。計算$\sigma_{catalog='Book' \text{ AND } unitPrice > 500}$Product的成本
  - (SL): $b_{Product}$=5000
  - (SSI): 這裡有兩個索引可以使用
    - 使用catalog索引： 1002（參考範例二）
    - 使用unitPrice索引：50502（參考範例三）

範例四(Cont.)

- (SMI): 根據catalog索引取得記錄指標共需花費成本

$$x_{catalog} - 1 + \left\lfloor bI1_{catalog} \cdot \frac{s_{catalog}}{r_{Product}} \right\rfloor = 2$$

- 根據unitPrice索引取得記錄指標共需花費成本

$$x_{unitPrice} - 1 + \frac{bI1_{unitPrice}}{2} = 2+500=502$$

- 取這些記錄指標的交集
  - 可在上一步驟裡一併處理，故成本為0
- 選擇幅度為$s_{catalog,unitPrice} = \frac{r_{Product}}{d_{catalog}} \cdot \frac{1}{2} = 500$
- 總成本為

2+502+0+500 = 1004

範例四(Cont.)

- (SCI): 由上述（SMI）的推論，我們知道 $s_{catalog,unitPrice}=50$，所以成本為

$$x_{catalog,unitPrice} - 1 + \left\lceil bI1_{catalog,unitPrice} \cdot \frac{s_{catalog,unitPrice}}{r_{Product}} \right\rceil + s_{catalog,unitPrice}$$
$$= 4 - 1 + 10 + 500 = 513$$

練習11-1

- 在範例四的（SSI）中，利用catalog索引比利用unitPrice索引的成本低許多，你可以因此得到什麼結論嗎？
- Ans:
  - 選擇幅度小的屬性之索引成本較低。以本例來說，catalog='Book'的選擇幅度為1000，而unitPrice>500的選擇幅度為50,000。

外部排序的運算

- 有些查詢句要求結果要排序

```
SELECT *
FROM Product
ORDER BY unitPrice;
```

- 排序也有助於某些查詢句的處理（如下一章的JOIN）
- 資料表裡的記錄是位於次記憶體（硬碟）
- 我們需要的是能處理位於次記憶體裡記錄的排序演算法，稱為**外部排序法**

外部排序的運算（Cont)

- 類似Merge Sort
  - 先將所有的資料切成數塊，每一塊都可放入主記憶體裡分別作**排序**
  - 再將這些排序好的數塊資料作適當的**合併**
- 成本
  - 假設主記憶體裡最多可容納n_B個讀入的資料頁
  - 資料被切成$\lceil b_R/n_B \rceil$個資料塊
  - 假設$n_B \geq \lceil b_R/n_B \rceil$
    - 資料塊排序總成本：$2b_R$
    - 資料塊合併總成本： $2b_R$
    - 總成本：$4b_R$

外部排序的運算（Cont)

- 成本
  - 假設$n_B < \lceil b_R/n_B \rceil$
    - 資料塊排序總成本：$2b_R$
    - 資料塊合併總成本： $2b_R \log_{n_B} b_R$
    - 總成本：$2b_R + 2b_R \log_{n_B} b_R$
- 成本通常簡化成 $K \cdot b_R \cdot \log_2 b_R$