

第九章 資料儲存結構

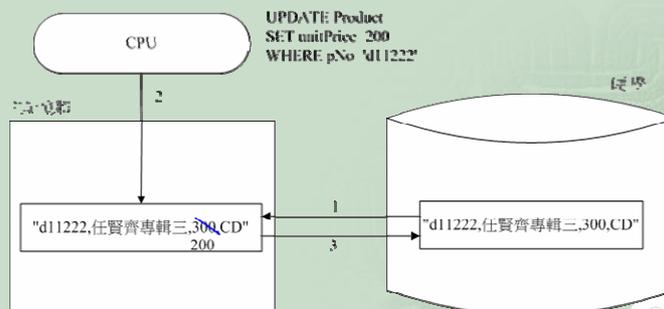
- 資料庫裡資料的儲存特性
- 資料表的資料結構
- B⁺-tree的索引結構
 - 二元樹
 - B⁺-tree的索引結構
 - B⁺-tree的搜尋
 - B⁺-tree的索引結構大小
 - 記錄增刪
- Suffix tree

©黃三益2007
資料庫的核心理論與實務第三版

9-1

資料庫裡資料的儲存特性

- DBMS所管理的資料量一般相當的龐大，必須存在硬碟
- 硬碟的存取單位是硬碟頁
- 硬碟的存取方式如下：



©黃三益2007
資料庫的核心理論與實務第三版

9-2

練習9-1：

- 請問上頁圖中，(1)、(2)，和 (3) 那個動作最快？

- Ans:

由於(1)和(3)都是主記憶體與硬碟交換資料，速度較慢。(2)則是CPU處理主記憶體裡的資料，速度最快

資料表的資料結構

- 一個資料表是由數個資料頁所構成
- 一個資料頁又包括數筆記錄
- 邏輯結構如右圖所示

	Product	
第1筆記錄	'b30999,資料庫理論與實務,500, Book'	第1頁
第2筆記錄	'd11222,任賢齊專輯三,300, CD'	
第3筆記錄	'b20666,OLAP進階, 500, Book'	
第4筆記錄	'b10234, 管理資訊系統概論, 600, Book'	
第5筆記錄	'b40555,系統分析理論與實務,550, Book'	第2頁
第6筆記錄	'd20777,蔡依林專輯二,350, CD'	
第7筆記錄	'v01888,哈利波特：混血王子的奇航,450, DVD'	
第8筆記錄	'd03333,5566專輯,450, CD'	
第9筆記錄	'b51111,電子商務理論與實務,700, Book'	第3頁
第10筆記錄	'v00111,英雄,400, DVD'	

資料表的資料結構 (Cont.)

- 在硬碟裡，同一個資料表的資料頁在硬碟裡不一定連續
- 資料頁的順序關係是用鍊結 (Linked list) 來表達
- 資料頁裡的記錄也不一定連續儲存
- DBMS系統裡也記載每一個資料表的第一個資料頁的頁數和各個屬性的順序和型態，稱為資料字典

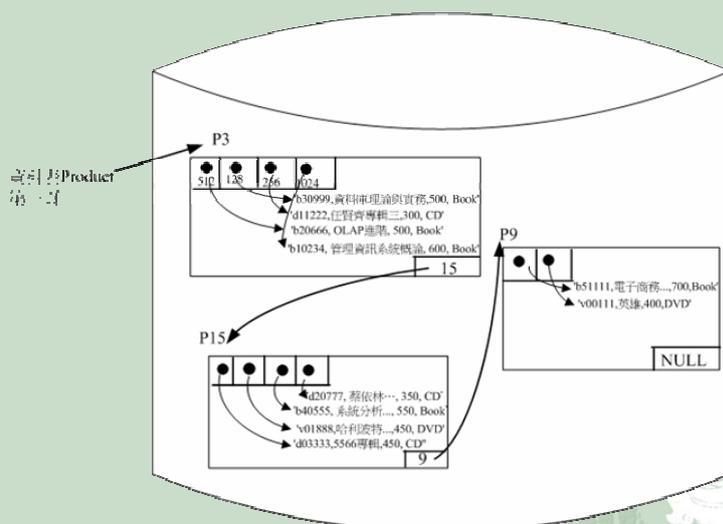
☞ 資料字典也可存成資料表

SystemTable			SystemAttribute				
tableName	tableFirstPage	tableConstraint	tableName	attrName	attrOrder	attrType	attribute Constraint
Product	3	Product	pNo	1	CHAR(6)
Member	215	Product	pName	2	VARCHAR(30)
Cart	30	Product	unitPrice	3
Transaction	108	Product	catalog	4
Order	12	Member	mId	1	CHAR(8)
Browse	1028	Member	pId	2	CHAR(10)
Record	29
Author	224
		

©黃三益2007
資料庫的核心理論與實務第三版

9-5

資料表的資料結構 (Cont.)



©黃三益2007
資料庫的核心理論與實務第三版

9-6

練習9-2

- 假設資料字典已被載入主記憶體，但Product的資料頁還沒被載入。上例中若想取得'v01888'的產品資料，請描述其處理動作。此時共需載入幾個資料頁？
- Ans:

檢查資料字典後，先載入Product資料表的第一頁（P3），再載入第二頁（P15），即發現所要的資料。所以共載入二個資料頁



B+-tree索引結構

- 要找出滿足某個條件的所有記錄，可以對相關資料表的所有資料頁一個一個的順序找尋
 - ☞效率可能很差
- 索引結構是用來將某些屬性的屬性值組織起來，以便快速找出滿足這些屬性的條件之記錄
- 最普遍的索引結構為B⁺-tree
- B⁺-tree的基本概念是由二元樹而來



傳統二元樹

- 節點

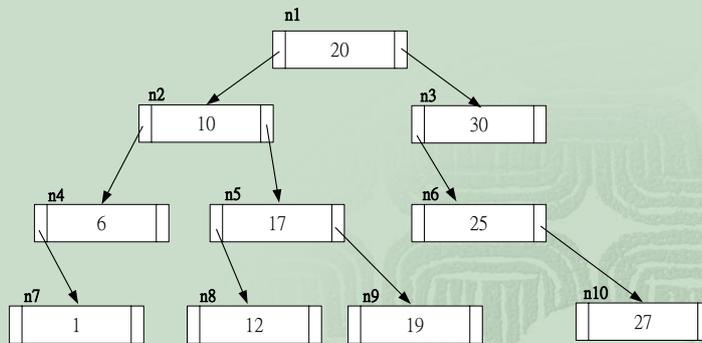
- ⊗ 根節點

- ⊗ 葉節點

- 子樹

- ⊗ 左子樹

- ⊗ 右子樹



©黃三益2007
資料庫的核心理論與實務第三版

9-9

傳統二元樹 (Cont.)

- 不適合資料庫使用

- ⊗ 存在主記憶體裡

- ⊗ 不是一棵平衡樹

- ⊗ 沒有存記錄的指標值

- 資料庫的索引結構應具有以下特性

- ⊗ 每一個節點就是硬碟裡的一頁

- 一個節點裡要包括多個<索引值, 指標>

- ⊗ 該樹狀結構必須是平衡的。

- ⊗ 空間的利用率不能太低

©黃三益2007
資料庫的核心理論與實務第三版

9-10

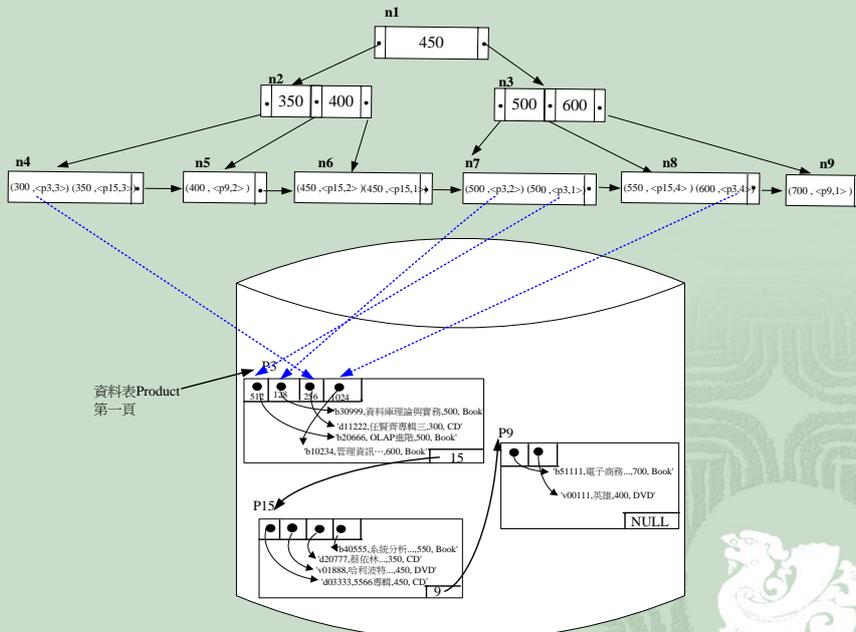
B+-tree索引結構 (Cont.)

- B+-tree 的結構包含兩種節點：
 - ☞ 中間節點：包括多個索引值和節點指標值
 - ☞ 葉節點：包含多個<屬性值, 記錄指標值>, 再加上一個指標值指到下一個葉節點
- 除了根節點外, 每一個節點的空間利用率至少為 50%
- 搜尋方式類似二元樹
- 範例 (結構如下頁)

```
CREATE INDEX I1
ON Product(unitPrice);
```

©黃三益2007
資料庫的核心理論與實務第三版

9-11



©黃三益2007
資料庫的核心理論與實務第三版

9-12

B+-tree的搜尋

- 類似二元樹，但每一個節點可能必須檢視多個索引值
- 範例
SELECT *
FROM Product
WHERE unitPrice=550;
- 按上圖，共檢視了4個硬碟頁
 - ☞ 3個索引頁 (n1, n3, n8)
 - ☞ 1個資料頁 (p15)

©黃三益2007
資料庫的核心理論與實務第三版

9-13

B+-tree的搜尋 (Cont.)

- B+-tree也可用來做範圍條件的搜尋。考慮以下的SQL指令：
SELECT *
FROM Product
WHERE unitPrice BETWEEN 400 AND 550;
- 在圖9-6裡，共需搜尋索引頁有n1, n2, n5, n6, n7, n8共6個，資料頁則有p9, p15, 和p3共3個。所以共需抓取9個硬碟頁

©黃三益2007
資料庫的核心理論與實務第三版

9-14

B+-tree的搜尋 (Cont.)

- 練習9-4：考慮以下SQL查詢句：

```
SELECT *
FROM Product
WHERE unitPrice = 700;
```

- 請問，若系統已有一個索引結構如圖9-6，要執行以上查詢，共需造訪幾個硬碟頁（包括索引頁和資料頁）？

- Ans:

索引頁會造訪n1, n3和n9，資料頁則造訪p9。所以共造訪四個硬碟頁

©黃三益2007
資料庫的核心理論與實務第三版

9-15

B+-tree索引結構的大小

- 假設：
 - ☞ 一個硬碟頁有4KB容量。
 - ☞ 一個索引值（屬性值）佔20B
 - ☞ 一個節點指標佔8B
 - ☞ 一個記錄指標佔10B
- 每一中間節點可容納p個節點指標及p-1個索引值
 - ☞ $(p \times 8) + ((p-1) \times 20) \leq 4K$
 - ☞ $p \leq 147, p \geq 74$
- 每一葉節點可容納 P_{leaf} 個記錄指標加上屬性值，再加上一個節點指標指到下一個鄰接的葉節點
 - ☞ $(P_{leaf} \times (10 + 20)) + 8 \leq 4K$
 - ☞ $P_{leaf} \leq 136, P_{leaf} \geq 68$
- 每一節點的空間利用率至少一半
- 三層B+-tree範例

B+-tree是一顆非常扁平的樹

第一層中間節點	1	74 節點指標
第二層中間節點	74	74x74=5476 節點指標
第三層葉節點	5476	5476x68=372,368 記錄指標

©黃三益2007
資料庫的核心理論與實務第三版

9-16

練習9-3

- 有些研究已經證明B⁺-tree的每一節點平均利用率為69%，請據此計算在以上範例裡，一個三層的B⁺-tree平均可容納幾個記錄指標

- Ans:

每一個中間節點平均有 $147 \times 0.69 = 101$ 個節點指標

每一個葉節點平均有 $136 \times 0.69 = 93$ 個記錄指標。

對於三層的B⁺-tree，我們可以計算如下：

	總節點數	總指標數
第一層中間節點	1	101 節點指標
第二層中間節點	101	$101 \times 101 = 10201$ 節點指標
第三層葉節點	10201	$10201 \times 93 = 948,693$ 記錄指標

©黃三益2007
資料庫的核心理論與實務第三版

9-17

多屬性值索引的B⁺-tree

- B⁺-tree也可用於多屬性索引的建立

CREATE INDEX I2

ON Product(catalog ASC, unitPrice DESC);

☞ 葉節點裡是按照catalog欄位值由小排到大，而同一catalog欄位值的記錄則又按其unitPrice欄位值由大排到小

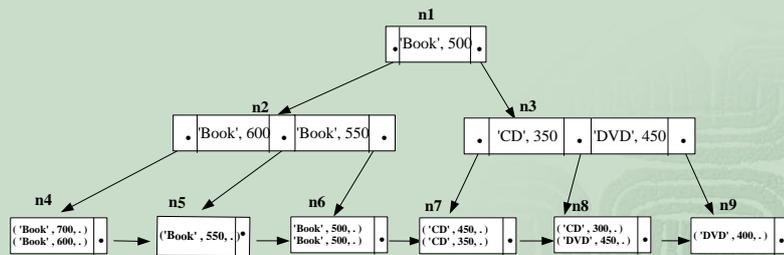
☞ 中間節點裡的索引值也是按照這樣的次序排列

☞ 範例結構如下頁圖

©黃三益2007
資料庫的核心理論與實務第三版

9-18

多屬性值索引的B⁺-tree (Cont.)



©黃三益2007
資料庫的核心理論與實務第三版

9-19

練習9-6

- 在圖9-7的索引裡，如果要搜尋所有250元的書，請問會經過哪些節點？
- Ans:

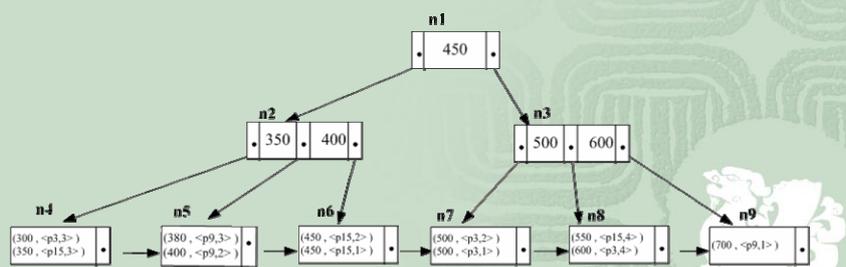
要搜尋('Book' , 250)，先找n1，由於該索引裡 unitPrice是由大排到小，而 $250 < 500$ ，所以接下來找n3，由於pName是由小排到大且' Book' < 'CD'，所以接下來找n7。至此，可以發現沒有('Book' , 250)這筆記錄

©黃三益2007
資料庫的核心理論與實務第三版

9-20

B+-tree的記錄增刪

- B+-tree是一棵平衡樹，且每一節點具有至少50%的空間利用率
- 記錄的增刪必須有適當的處理
- 圖9-6中，增加一筆記錄(假設是存在p9的第三筆記錄)：
('b40333' , '測試專用書1' , 380, 'Book')

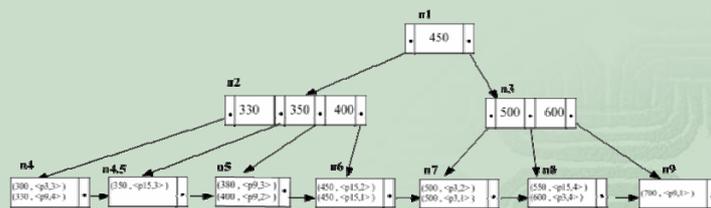


©黃三益2007
資料庫的核心理論與實務第三版

9-21

B+-tree的記錄增刪 (Cont.)

- 再增加一筆記錄：
('b40334' , '測試專用書2' , 330, 'Book')

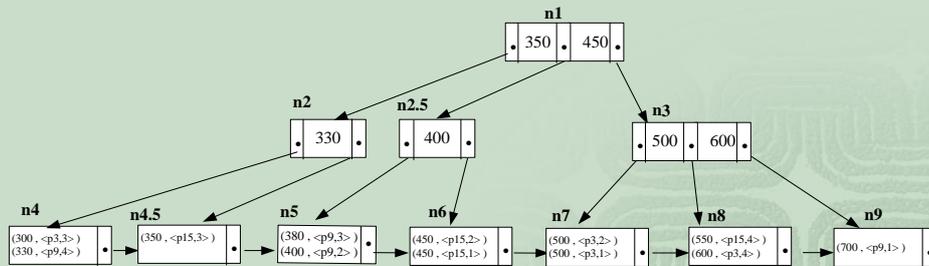


假設一個中間節點只能存2個索引值，以上中間節點n2裡存了過多的索引值，必須切割，如下頁圖

©黃三益2007
資料庫的核心理論與實務第三版

9-22

B+-tree的記錄增刪 (Cont.)

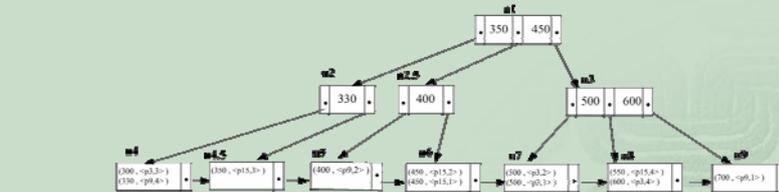


©黃三益2007
資料庫的核心理論與實務第三版

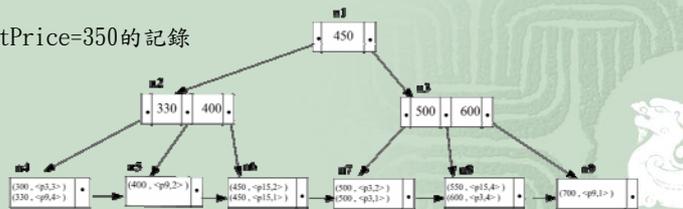
9-23

B+-tree的記錄增刪 (Cont.)

- 刪除記錄
('b40333' , '測試專用書1' , 380 , 'Book')



刪除unitPrice=350的記錄



©黃三益2007
資料庫的核心理論與實務第三版

9-24

Suffix tree

- 前述B+-tree適用於搜尋整個屬性值的條件
 - ☞ 相等值的查詢
 - ☞ 屬性值位於一定範圍的查詢
- SQL敘述裡對於字串欄位常用**LIKE**來搜尋
- Suffix tree，可用來加快具有文字欄位匹配條件的查詢句之處理

☞ 比如以下的查詢句：

```
SELECT *  
FROM Member  
WHERE address LIKE '%中華路%' ;
```

©黃三益2007
資料庫的核心理論與實務第三版

9-25



Suffix tree (Cont.)

- Suffix tree是用來儲存一些字串的後段字串 (Suffix)
- “台北市中華路一段100號”的其後段字串包括
 - ☞ 台北市中華路一段100號
 - ☞ 北市中華路一段100號
 - ☞ 市中華路一段100號
 - ☞ 中華路一段100號
 - ☞ 華路一段100號
 - ☞ 路一段100號
 - ☞ 一段100號
 - ☞ 段100號
 - ☞ 100號
 - ☞ 00號
 - ☞ 0號
 - ☞ 號

©黃三益2007
資料庫的核心理論與實務第三版

9-26



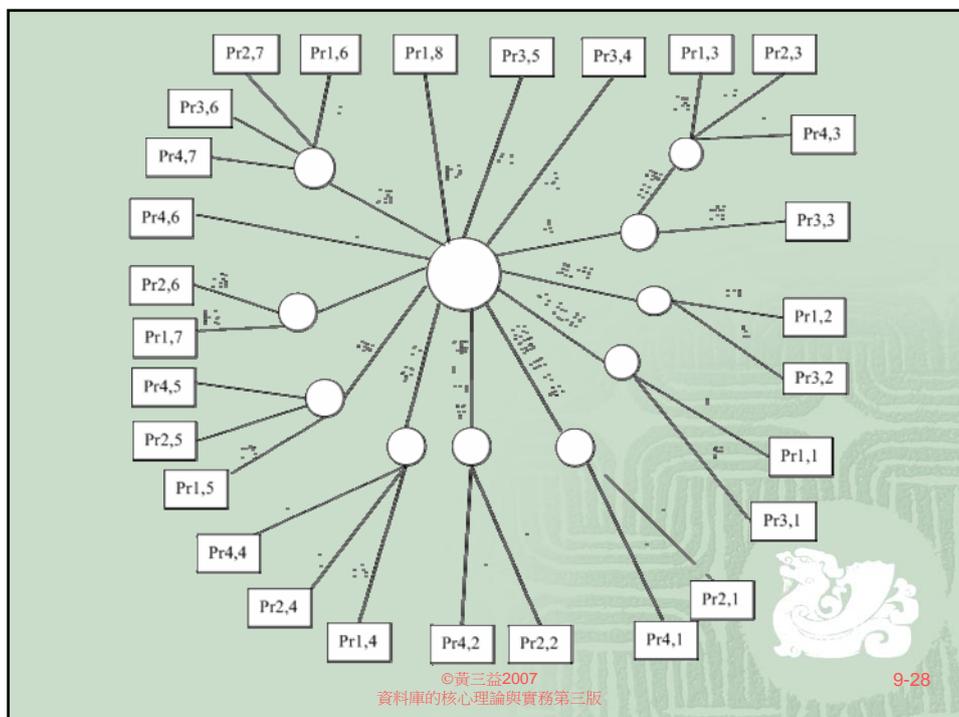
Suffix tree (Cont.)

- Suffix tree的葉節點裡存的是一個後端字串所屬的記錄指標以及其開始位置
- 假設我們有四筆Member記錄，其記錄指標值分別為pr1, pr2, pr3, pr4，且其address屬性值分別為：
 - ☞ pr1: 台北市中華路三段
 - ☞ pr2: 高雄市中華三路
 - ☞ pr3: 台北市南昌路
 - ☞ pr4: 高雄市中華二路
- Suffix tree如 [下圖](#)



©黃三益2007
資料庫的核心理論與實務第三版

9-27



9-28

Suffix tree (Cont.)

- Suffix tree也可用來輔助搜尋較複雜的LIKE條件。考慮以下的查詢：

SELECT *

FROM Member

WHERE address LIKE ‘%中華_路%’ ;

❧ 找” 中華” 會找到 (Pr1, 4), (Pr2, 4), (Pr4, 4)

❧ 找” 路” 也可找到 (Pr1, 6), (Pr2, 7), (Pr3, 6), (Pr4, 7)

❧ ’ 中華’ 和’ 路’ 的開始位置必須差3個字元

- Pr2, Pr4

©黃三益2007
資料庫的核心理論與實務第三版

9-29

練習9-7

- 請問如何利用圖9-12的Suffix tree來處理以下查詢：

SELECT *

FROM Member

WHERE address LIKE ‘台北市%中華%’ ;

- Ans:

先找’ 台北市’，會找到 (Pr1, 1) 和 (Pr3, 1)，再找’ 中華’，會找到 (Pr1, 4), (Pr2, 4), (Pr4, 4)，由於本題不要求確切距離，因此只找到Pr1合乎條件

©黃三益2007
資料庫的核心理論與實務第三版

9-30