

Appendix

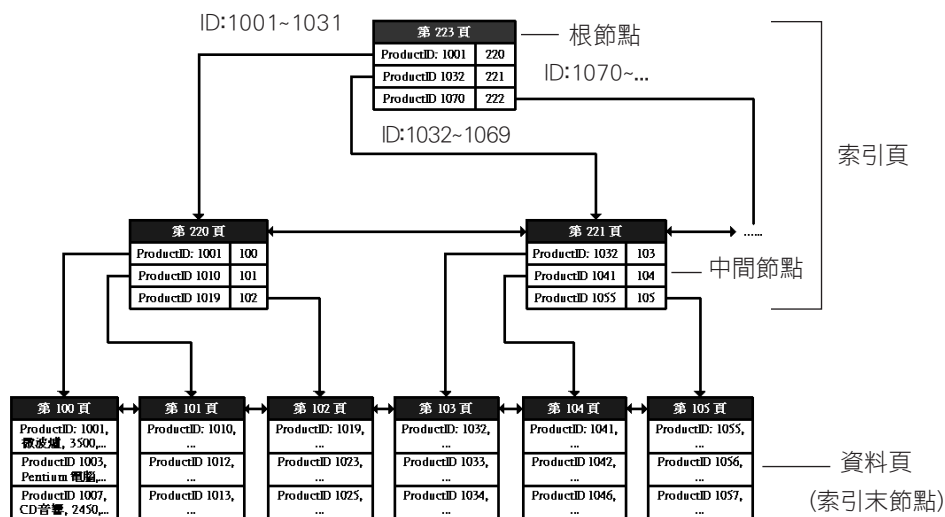
E

叢集索引與非叢集索引的結構

叢集索引與非叢集索引都是使用 B-tree 的結構來建立索引, 而且會使用到索引頁與資料頁, 其中索引頁是用來存放索引, 以及指到下一層的指標, 資料頁則是實際儲存記錄的地方(只有叢集索引才會包含資料頁)。由於叢集索引會將記錄依照索引欄位的順序來排列, 因此其結構要比非叢集索引的結構簡單一些, 以下分別做介紹。

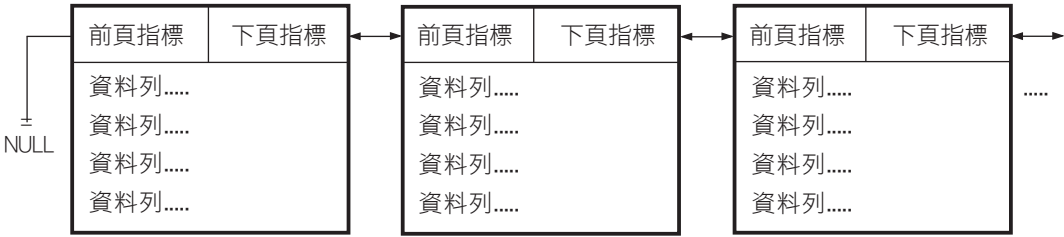
E-1 叢集索引的結構

叢集索引的 B-tree 是由下而上建構而成，一個資料頁（也是索引的末節點）中包含數筆記錄，再由數個資料頁而連結到一個中間節點的索引頁。接著由數個中間節點的索引頁再連結到更上層的索引頁，最後匯合而成根節點的索引頁：



在叢集索引的資料頁中，記錄是已經依照順序排列好的，因此當搜尋記錄時，即可從根節點處，一層一層地往下尋找。我們以上圖為例來說明，假設要搜尋 ProductID 等於 1025 的記錄，則先由根節點的第 223 號索引頁查到 1025 是排在第 220 號索引頁中，然後再查到第 102 號資料頁，最後再由其中找到 1025 的記錄。

如果是要循序讀取資料，例如要查詢某個範圍內的資料（像是 WHERE id >120 之類的條件），那麼由於資料頁中的記錄已循序排列，而且資料頁之間也是循序排列(資料頁間還有橫向的連結指標)，因此可以很快地循序讀取記錄。底下我們來看看資料頁的細部結構：

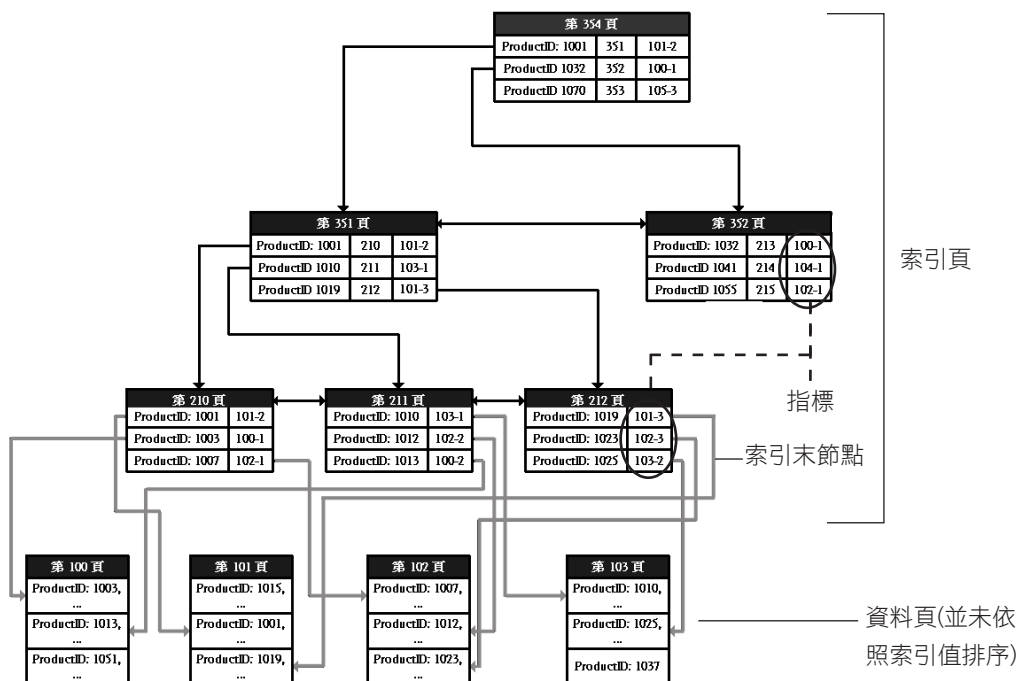


其中的**前頁指標**、**下頁指標**指向前一個及下一個資料頁，而資料頁中則是存放實際的記錄（資料列）。另外，索引頁的結構也和資料頁差不多，只不過在索引頁的資料列中，是存放著已排序好的索引值（即索引欄的值），以及指向下一層索引頁的指標。

新增或刪除記錄時，則有可能會影響到每一個索引頁所能夠容納的索引數量，而造成需要將索引頁分割或合併，此時 B-tree 的架構與中間節點數量及深度就有可能會改變。

E-2 非叢集索引的結構

非叢集索引的結構與叢集索引相當類似，但差別在於**非叢集索引**中不包含資料頁，所以在每一個索引頁的各資料列中都有一個指標，指到某一個資料頁的第某筆記錄：



在非叢集索引的結構中，因記錄並沒有依照順序排列好，所以到了末節點索引頁時，並無法直接找到要搜尋的記錄，此時就需要靠索引頁中的另一個指標來指向真正的資料頁位置。

例如從上圖中要搜尋 ProductID 等於 1025 的記錄，則從根節點 (第 354 號的索引頁) 開始，查出 1025 是排在第 351 號索引頁中，然後再查到第 212 號索引頁，最後找到 1025 所對應到的第 103 頁的第 2 筆記錄。

以上是當資料表沒有設定叢集索引的情況，此時資料頁中的記錄並未依照特定的欄位值排序，所以非叢集索引中的每一個指標都會直接指向對應資料頁中的記錄。

TIP

注意，資料頁與非叢集索引是分開存放的。

若一個資料表中同時有**叢集索引**與**非叢集索引**時，則在**非叢集索引**的索引頁中，其指標實際上是儲存著**叢集索引**的索引值（即 Primary key），系統必須依此索引值再到**叢集索引**中找出對應的記錄來。例如下面一筆記錄：

編號	姓名	地址
100	楊小雄	台北市....

其中以『編號』欄為**叢集索引**，以『姓名』欄為**非叢集索引**，那麼當我們要查詢“楊小雄”的記錄時，會先在**非叢集索引**中找到『編號』100，然後再用此值到**叢集索引**中找出實際的記錄。這樣做的好處是，當新增或更改一筆記錄而導致資料頁有多筆記錄更動位置時，**非叢集索引**中仍然只需加入該筆新增或修改記錄的索引，其他索引不需更改其指標值。

由於這樣的特性，當同時使用這二種索引時，最好以單一欄位做**叢集索引**，而且欄位長度越短越好，以加快查詢效率並節省儲存空間。此外，由於這二種索引的關係相當密切，若是更換或刪除**叢集索引**的索引欄位，則必須重建**非叢集索引**的結構。

在**叢集索引**結構中，索引頁的末節點與資料頁是合在一起的；而**非叢集索引**中只儲存索引頁，並利用指標去讀取資料頁中的實際記錄

