

**Exercises**  
**(Course: Database Management Systems)**  
**Chapter 2**  
**Indexing Structures for Files**

1. Exercise 18.18 in the text book (“Fundamentals of Database Systems- 6th Edition”, Elmasri et al.)

Consider a disk with block size  $B = 512$  bytes. A block pointer is  $P = 6$  bytes long, and a record pointer is  $PR = 7$  bytes long. A file has  $r = 30,000$  **EMPLOYEE** records of *fixed length*. Each record has the following fields: **Name** (30 bytes), **Ssn** (9 bytes), **Department\_code** (9 bytes), **Address** (40 bytes), **Phone** (10 bytes), **Birth\_date** (8 bytes), **Sex** (1 byte), **Job\_code** (4 bytes), and **Salary** (4 bytes, real number). An additional byte is used as a deletion marker.

- Calculate the record size  $R$  in bytes.
- Calculate the blocking factor  $bfr$  and the number of file blocks  $b$ , assuming an unspanned organization.
- Suppose that the file is *ordered* by the key field **Ssn** and we want to construct a *primary index* on **Ssn**. Calculate (i) the index blocking factor  $bfr_i$  (which is also the index fan-out  $fo$ ); (ii) the number of first-level index entries and the number of first-level index blocks; (iii) the number of levels needed if we make it into a multilevel index; (iv) the total number of blocks required by the multilevel index; and (v) the number of block accesses needed to search for and retrieve a record from the file—given its **Ssn** value—using the primary index.
- Suppose that the file is *not ordered* by the key field **Ssn** and we want to construct a *secondary index* on **Ssn**. Repeat the previous exercise (part c) for the secondary index and compare with the primary index.

**Answer:**

- a) Tính kích thước của mỗi bản ghi.

**Ans:**  $R = 30 + 9 + 9 + 40 + 10 + 8 + 1 + 4 + 4 + 1 = 116$  byte.

- b) Tính hệ số phân khối  $bfr$  và số khối (block)  $b$  của tập tin dữ liệu EMPLOYEE.

**Ans:**  $bfr = \lfloor B/R \rfloor = \lfloor 512/116 \rfloor = 4$  bản ghi mỗi block

$$b = \lceil 30000/bfr \rceil = \lceil 30000/4 \rceil = 7500 \text{ blocks}$$

- c) Giả sử tập tin dữ liệu sắp thứ tự theo thuộc tính khóa SSN và chúng ta muốn lập **chỉ mục sơ cấp** theo SSN.

- Tính (i) hệ số phân khối của chỉ mục sơ cấp  $bfr_i$

**Ans:** Kích thước của mỗi phần tử (entry) trong chỉ mục là  $R_i = 9 + 6 = 15$

Hệ số phân khối của chỉ mục là:  $bfr_i = \lfloor B/R_i \rfloor = \lfloor 512/15 \rfloor = 34$  phần tử mỗi block.

Lưu ý hệ số phân khối của chỉ mục cũng là độ phân nhánh (fan-out) của chỉ mục đa mức.

- Tính (ii) số phần tử (entry) của chỉ mục ở mức thứ nhất và số khối của chỉ mục mức thứ nhất.

**Ans:** Số phần tử của chỉ mục ở mức thứ nhất bằng với số khối của tập tin dữ liệu, tức là 7500.

$$\text{Số khối của chỉ mục ở mức thứ nhất} = \lceil 7500/bfr_i \rceil = \lceil 7500/34 \rceil = 221$$

- Tính (iii) số mức cần thiết nếu chúng ta tổ chức chỉ mục thành một cấu trúc chỉ mục đa mức.

**Ans:** Ta đã biết số khối của chỉ mục ở mức thứ nhất là 221, với độ phân nhánh là 34, số khối ở mức thứ hai của chỉ mục đa mức sẽ là  $\lceil 221/34 \rceil = 7$ .

Suy ra số khối ở mức kế tiếp của chỉ mục đa mức là  $\lceil 7/34 \rceil = 1$  khối. Do đó mức ba là mức cao nhất của chỉ mục đa cấp và số mức cần thiết của cấu trúc chỉ mục đa mức  $x = 3$ .

$$r1/(f0)^t \geq 1 \rightarrow t = \lceil \log_{f0}(r1) \rceil \rightarrow t = \log_{34}(7500) \rightarrow t = 2.5302 \rightarrow t = 3$$

- Tính (iv) số khối cần thiết của cấu trúc chỉ mục đa mức.

**Ans:** Mức ba chứa 1 khối gồm 34 phần tử mà sẽ có 34 block pointer, mỗi block pointer chỉ đến một block ở mức hai. Như vậy mức hai sẽ có 34 block.

Mức hai chứa 34 block sẽ gồm  $34 * 34$  block pointer chỉ đến  $34 * 34$  block ở mức một.

Như vậy toàn chỉ mục đa mức có tổng cộng  $1 + 34 + 34 * 34 = 1 + 34 + 1156 = 1191$  khối.

- *Tính (v) số lần truy đạt khối cần thiết để khi dùng chỉ mục đa mức vừa nêu.*

**Ans:** Vì chỉ mục có ba mức nên cần tất cả 3 lần truy đạt khối.

- Tính (vi) số lần truy đạt khối cần thiết để khi dùng chỉ mục sơ cấp để tìm kiếm và truy xuất một bản ghi từ tập tin dữ liệu khớp với một trị SSN cho trước.

**Ans:** Nếu chỉ mục sơ cấp chỉ có một mức, ta sẽ dùng binary search và tốn  $\lceil \log_2 b_i \rceil$  truy đạt khối với  $b_i$  là số khối của chỉ mục. Ở đây  $b_i = 221$  và  $\lceil \log_2(221) \rceil = 8$  lần truy đạt khối để dò tìm trên chỉ mục sơ cấp. Để tìm kiếm một bản ghi trên tập tin dữ liệu, ta cần thêm một truy đạt khối nữa để tìm kiếm trên tập tin dữ liệu. Như vậy tốn tổng cộng  $8 + 1 = 9$  lần truy đạt khối.

- Nếu chỉ mục sơ cấp là chỉ mục đa mức thì nó gồm 3 mức như ở câu (iii) thì sẽ tốn một lần truy đạt khối ở mỗi mức của chỉ mục và một lần truy đạt khối tại tập tin dữ liệu, tức là ta cần  $x + 1 = 3 + 1 = 4$  lần truy đạt khối.

d) Giả sử tập tin không được sắp thứ tự theo thuộc tính khóa SSN và chúng ta muốn xây dựng một **chỉ mục thứ cấp** theo SSN. Lập lại các tính toán ở phần c) cho chỉ mục thứ cấp và so sánh với chỉ mục sơ cấp.

- Tính (i) hệ số phân khối của chỉ mục thứ cấp  $bfr_i$

**Ans:** Kích thước của mỗi phần tử (entry) trong chỉ mục là  $R_i = 9 + 6 = 15$

Hệ số phân khối của chỉ mục là:  $bfr_i = \lfloor B/R_i \rfloor = \lfloor 512/15 \rfloor = 34$  phần tử mỗi block.

Lưu ý hệ số phân khối của chỉ mục cũng là độ phân nhánh (fan-out) của chỉ mục đa mức.

- Tính (ii) số phần tử (entry) của chỉ mục thứ cấp ở mức thứ nhất và số khối của chỉ mục thứ cấp ở mức thứ nhất.

**Ans:** Chỉ mục thứ cấp này là một chỉ mục dày do lập chỉ mục trên thuộc tính khóa SSN. Do đó số phần tử của chỉ mục này ở mức thứ nhất bằng với số bản ghi của tập tin dữ liệu, tức là 30000.

$$\text{Số khối của chỉ mục ở mức thứ nhất} = \lceil 30000/bfr_i \rceil = \lceil 3000/34 \rceil = 883$$

- Tính (iii) số mức cần thiết nếu chúng ta tổ chức chỉ mục thứ cấp thành một cấu trúc chỉ mục đa mức.

**Ans:** Ta đã biết số khối của chỉ mục ở cấp thứ nhất là 883, với độ phân nhánh là 34, số khối ở mức thứ hai của chỉ mục đa mức sẽ là  $\lceil 883/34 \rceil = 26$ .

Suy ra số khối ở mức kế tiếp của chỉ mục đa mức là  $\lceil 26/34 \rceil = 1$  khối. Do đó mức ba là mức cao nhất của chỉ mục đa cấp và số mức cần thiết của cấu trúc chỉ mục đa mức  $x = 3$ .

$$r1/(f0)^t \geq 1 \rightarrow t = \lceil \log_{f0}(r1) \rceil \rightarrow t = \log_{34}(30000) \rightarrow t = 2.5302 \rightarrow 3$$

- Tính (iv) số khối cần thiết của cấu trúc chỉ mục đa mức.

**Ans:** Mức ba chứa 1 khối gồm 34 phần tử mà sẽ có 34 block pointer, mỗi block pointer chỉ đến một block ở mức hai. Như vậy mức hai sẽ có 34 block.

Mức hai chứa 34 block sẽ gồm  $34 * 34$  block pointer chỉ đến  $34 * 34$  block ở mức một.

Như vậy toàn chỉ mục đa mức có tổng cộng  $1 + 34 + 34 * 34 = 1 + 34 + 1156 = 1191$  khối. (Cũng bằng với trường hợp chỉ mục sơ cấp là đa mức)

- Tính (v) số lần truy đạt khối cần thiết để khi dùng chỉ mục đa mức vừa nêu.

**Ans:** Vì chỉ mục có ba mức nên cần tất cả 3 lần truy đạt khối. (Cũng bằng với trường hợp chỉ mục sơ cấp là đa mức)

- Tính (vi) số lần truy đạt khối cần thiết để khi dùng chỉ mục sơ cấp để tìm kiếm và truy xuất một bản ghi từ tập tin dữ liệu khớp với một trị SSN cho trước.

**Ans:** Nếu chỉ mục thứ cấp chỉ có một mức, ta sẽ tốn  $\lceil \log_2 bi \rceil$  với bi là số khối của chỉ mục. Ở đây bi = 883 và  $\lceil \log_2(883) \rceil = 10$  lần truy đạt khối. Và thêm một lần truy đạt khối để tìm kiếm bản ghi tương ứng trên tập tin dữ liệu. Như vậy tổng cộng tốn  $10 + 1 = 11$  lần truy đạt khối.

(Như vậy so với chỉ có 9 lần truy đạt khối nếu có chỉ mục sơ cấp trên SSN, thì sẽ tốn hơn 2 lần truy đạt khối)

Nếu chỉ mục thứ cấp là chỉ mục đa mức thì nó gồm 3 mức như ở câu (iii) thì sẽ tốn một lần truy đạt khối ở mỗi mức của chỉ mục và một lần truy đạt khối tại tập tin dữ liệu, tức là ta cần tốn cộng  $x + 1 = 3 + 1 = 4$  lần truy đạt khối.

(Như vậy số lần truy đạt khối ở đây bằng với số lần truy đạt khối khi sử dụng chỉ mục sơ cấp đa mức).

2. Exercise 18.19 in the text book (“Fundamentals of Database Systems- 6th Edition”, Elmasri et al.)

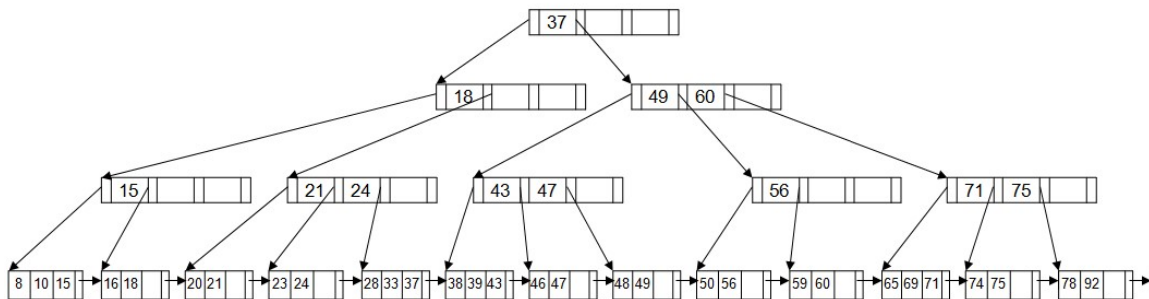
PARTS file with Part# as the key field includes records with the following Part# values: 23, 65, 37, 60, 46, 92, 48, 71, 56, 59, 18, 21, 10, 74, 78, 15, 16, 20, 24, 28, 39, 43, 47, 50, 69, 75, 8, 49, 33, 38. Suppose that the search field values are inserted in the given order in a B<sup>+</sup>-tree of order p = 4 and p<sub>leaf</sub> = 3.

Show how the tree will expand and what the final tree will look like.

**Answer:**

23, 65, 37, 60, 46, 92, 48, 71, 56, 59, 18, 21, 10, 74, 78, 15, 16, 20, 24, 28, 39, 43, 47, 50, 69, 75, 8, 49, 33, 38

B<sup>+</sup>-tree with order p = 4, p<sub>leaf</sub> = 3, half full → 50%\*p = 50%\*4 = 2 → at least 2 tree pointers;  
50%\*p<sub>leaf</sub> = 50%\*3 = 1.5 → at least 2 data pointers



3. Exercise 18.21 in the text book (“Fundamentals of Database Systems- 6th Edition”, Elmasri et al.)

Suppose that the following search field values are deleted, in the given order, from the B<sup>+</sup>-tree of Exercise 18.19. The deleted values are 65, 75, 43, 18, 20, 92, 59, 37.

Show how the tree will shrink and show the final tree.

**Answer:**

From the B<sup>+</sup>-tree of Exercise 14.15, delete: 65, 75, 43, 18, 20, 92, 59, 37

