

Kiểm tra giữa kỳ

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu (503004)** - Ngành: **Khoa Học Máy Tính – HK1 – 2012-2013**

Thời gian làm bài: **60 phút**

(Bài kiểm tra gồm 20 câu hỏi. Sinh viên được **tham khảo ghi chú** trong 2 tờ giấy A4.)

Sinh viên chọn 1 câu trả lời đúng nhất. Nếu chọn câu (e) thì sinh viên cần trình bày đáp án khác so với đáp án ở các câu (a), (b), (c), và (d) và/hoặc giải thích lựa chọn (e) của mình.

Câu 1. *Hệ quản trị cơ sở dữ liệu* (database management system, DBMS) là gì?

- a. DBMS là một phần mềm hệ thống dùng giúp các chương trình ứng dụng được thực thi hiệu quả hơn.
- b. DBMS là một phần mềm chuyên dụng dùng giúp quản lý dữ liệu của các chương trình ứng dụng hiệu quả hơn.
- c. DBMS là một phần mềm chuyên dụng dùng giúp tạo và quản lý một lượng lớn dữ liệu hiệu quả và bền vững an toàn theo thời gian. *
- d. DBMS là một phần mềm ứng dụng dùng giúp tận dụng sự hỗ trợ của hệ điều hành để tạo và quản lý lượng lớn dữ liệu và các chương trình ứng dụng hiệu quả theo thời gian.
- e. Ý kiến khác.

Câu 2. Cho kiến trúc của một DBMS trong **Hình 1**. Cho câu lệnh “**CREATE INDEX composite_index ON Nhân_Viên (Họ_Tên, Lương);**” được gửi đến DBMS; trong đó, Nhân_Viên là bảng dữ liệu với chi tiết được trình bày trong **Bảng 1**. Chọn phát biểu **ĐÚNG** về quá trình xử lý câu lệnh tạo chỉ mục phức hợp này.

- a. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên mô-đun DDL compiler thông báo lỗi khi phân tích cú pháp và kiểm tra ngữ nghĩa của câu lệnh tạo chỉ mục này. *
- b. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên thành phần Query Preprocessor của mô-đun Query compiler thông báo lỗi khi truy vấn trên bảng Nhân_Viên được thực hiện.
- c. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên mô-đun Execution engine thông báo lỗi khi thực hiện câu lệnh tạo chỉ mục này.
- d. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên mô-đun Buffer manager thông báo lỗi khi tìm kiếm siêu dữ liệu (metadata) về bảng Nhân_Viên.
- e. Ý kiến khác.

Câu 3. Cây chỉ mục **B** (*B-tree*) và cây chỉ mục **B+** (*B+-tree*) có những đặc điểm gì sau đây **khác nhau**?

- a. B-tree dùng làm chỉ mục đa mức tĩnh và B+-tree dùng làm chỉ mục đa mức động.
- b. B-tree dùng làm chỉ mục sơ cấp và B+-tree dùng làm chỉ mục thứ cấp.
- c. B-tree chỉ hỗ trợ toán tử =, <, và >; trong khi đó, B+-tree hỗ trợ toán tử =, <, <=, >, >=, và between.
- d. Cấu trúc của node nội và node lá trong B-tree giống nhau; trong khi đó, cấu trúc của node nội và node lá trong B+-tree khác nhau và do đó, khả năng chỉ mục của B+-tree thường hiệu quả hơn. *
- e. Ý kiến khác.

Câu 4. Cho tập tin dữ liệu Nhân_Viên gồm 7 khối dữ liệu (7 blocks) trong **Bảng 1**. Mỗi khối chứa 2 bản ghi (record). Vùng tin ID là vùng tin khóa (key field) của tập tin; các vùng tin còn lại là vùng tin phi khóa (non-key field). Ngoài ra, các bản ghi (record) trong tập tin này được sắp thứ tự vật lý theo các giá trị tăng dần của vùng tin ID. Xác định **số truy đạt khối ít nhất** (the minimum number of block accesses) có thể để thực hiện việc tìm kiếm bản ghi của nhân viên có mã ID = 29 trên tập tin dữ liệu Nhân_Viên.

- a. 7
- b. 4
- c. 3
- d. 2 *
- e. Ý kiến khác.

Câu 5. Cho tập tin dữ liệu Nhân_Viên gồm 7 khối dữ liệu (7 blocks) trong **Bảng 1**. Mỗi khối chứa 2 bản ghi (record). Vùng tin ID là vùng tin khóa (key field) của tập tin; các vùng tin còn lại là vùng tin phi khóa (non-key field). Ngoài ra, các bản ghi trong tập tin này được sắp thứ tự vật lý theo các giá trị tăng dần của vùng tin ID. Xác định **số truy đạt khối ít nhất** (the minimum number of block accesses) có thể để thực hiện việc tìm kiếm bản ghi của nhân viên có Tên = Việt trên tập tin dữ liệu Nhân_Viên.

- a. 13
- b. 8
- c. 7 *
- d. 4
- e. Ý kiến khác.

Câu 6. Công nghệ RAID nào sau đây sử dụng cơ chế rải dữ liệu (data striping) ở mức khối (block-level) để cải thiện hiệu suất ghi/đọc đĩa với song song hóa truy đạt đĩa (parallelizing disk access)?

- a. Rải dữ liệu chỉ được thực hiện ở mức bit (bit-level) nên không có RAID nào sử dụng cơ chế rải dữ liệu ở mức khối.
- b. RAID 1
- c. RAID 3
- d. RAID 5 *
- e. Ý kiến khác.

Câu 7. Cho **Hình 2** là cây chỉ mục B có bậc $p = 3$ trên vùng tin ID (khóa (key), sắp thứ tự (ordering field)) của tập tin dữ liệu Nhân_Viên ở **Bảng 1**. Cây chỉ mục **B** sau khi xóa (delete) giá trị tìm kiếm **28** sẽ có **đặc điểm** gì?

- a. Chiều cao cây giảm 1, số node lá giảm 1, và số node nội cũng giảm 1
- b. Chiều cao cây giảm 1, số node lá giảm 1, và số node nội giảm 2 *
- c. Chiều cao cây không đổi, số node lá giảm 1, và số node nội cũng không đổi
- d. Chiều cao cây không đổi, số node lá cũng không đổi, và số node nội giảm 1
- e. Ý kiến khác.

Câu 8. Cho cây chỉ mục B+ có bậc $p = 4$ và $p_{leaf} = 3$ ở **Hình 3**. Xác định **cây chỉ mục B+ kết quả** đạt được sau khi thêm giá trị tìm kiếm **63** vào cây chỉ mục B+ này.

- a. Hình 4
- b. Hình 5

- c. **Hình 6 ***
- d. **Hình 7**
- e. Ý kiến khác.

Câu 9. Cho cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở **Hình 3**. Xác định *số khối dữ liệu* (*number of blocks*) cần thiết để lưu trữ cây chỉ mục B+ này.

- a. 16
- b. $8 *$
- c. 5
- d. 3
- e. Ý kiến khác.

Câu 10. Cho cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở **Hình 3**. Xác định *số truy đạt khối* (*number of block accesses*) cần được thực hiện để đạt được khối dữ liệu chứa bản ghi có giá trị tìm kiếm là **37**.

- a. $4 *$
- b. 3
- c. 2
- d. 1
- e. Ý kiến khác.

Câu 11. Kỹ thuật bộ đệm đôi (*double buffering*) được sử dụng hiệu quả khi các khối dữ liệu cần được đọc/ghi được *tổ chức* dạng nào?

- a. Liên tục (*contiguous*) *
- b. Liên kết (*linked*)
- c. Cụm (*clustered*)
- d. Chỉ mục (*indexed*)
- e. Ý kiến khác.

Câu 12. Cho tập tin dữ liệu Nhân_Viên ở **Bảng 1** được lưu trữ vật lý trên bộ đĩa từ có số vòng quay trong 1 phút là $p = 5600$ rpm, có thời gian tìm kiếm trung bình (*average seek time*) là $s = 20$ msec. Kích thước một khối dữ liệu (*block size*) là $B = 512$ bytes. Kích thước một rãnh dữ liệu (*track size*) là $T = 102400$ bytes. Nếu các khối dữ liệu của tập tin dữ liệu này được tổ chức liên tục (*contiguous*) ở trên cùng một rãnh (*track*) thì tổng thời gian thực hiện chuyển các khối dữ liệu vào vùng nhớ chính khi đọc hết dữ liệu theo thứ tự của các giá trị ở vùng tin ID là *bao nhiêu msec*? Giả sử kết quả của các phép toán được làm tròn với 2 chữ số thập phân.

- a. 177.87 msec
- b. 57.87 msec
- c. 25.71 msec *
- d. 25.41 msec
- e. Ý kiến khác.

Câu 13. Cho tập tin dữ liệu Student gồm 20000 bản ghi ($r = 20000$ records) có chiều dài cố định (*fixed length*). Mỗi bản ghi gồm các vùng tin như sau: NAME (30 bytes), SSN (9 bytes), ADDRESS (40 bytes), PHONE (9 bytes), BIRTHDATE (8 bytes), SEX (1 byte), MAJORDEPTCODE (4 bytes), MINORDEPTCODE (4 bytes), CLASSCODE (4 bytes),

DEGREEPROGRAM (3 bytes), và DELETION MARKER (1 byte). Xác định **hệ số phân khối** (*blocking factor, bfr*) của tập tin dữ liệu Student này khi được lưu trữ trên đĩa từ có kích thước khối dữ liệu (*block size, B*) là **B** = 512 bytes theo cách phân khối không phủ (*unspanned blocking*).

- a. 5000
- b. 5
- c. 4.5
- d. 4 *
- e. Ý kiến khác.

Câu 14. Trong **Hình 8**, cho tập tin dữ liệu được tổ chức dưới dạng băm động (*dynamic hashing*) dùng băm khả mở (*extendible hashing*) với các bit dẫn đầu (*leading bits*) được sử dụng và hàm băm (*hash function*) là $h(K) = K \bmod 128$. Biết rằng: mỗi thùng (*bucket*) chứa 1 khối dữ liệu (*block*), mỗi khối dữ liệu chứa được 2 bản ghi (*record*). Xác định **độ sâu toàn cục** (*global depth, d*) của toàn tập tin và **độ sâu cục bộ** (*local depth, d'*) của thùng chứa bản ghi **3943** sau khi thêm bản ghi này vào tập tin. Cho trước $h(3943) = 3943 \bmod 128 = 103_{10} = 1100111_2$.

- a. $d = 2$ và $d' = 1$
- b. $d = 2$ và $d' = 2 *$
- c. $d = 3$ và $d' = 2$
- d. $d = 3$ và $d' = 3$
- e. Ý kiến khác.

Câu 15. Giả sử thư mục (*directory*) của tập tin băm (*hashed file*) ở **Hình 8** này được chứa trọn trong 1 khối dữ liệu. Xác định **số truy đạt khối** (*number of block accesses*) được thực hiện để đạt được bản ghi **3760**.

- a. 4
- b. 3
- c. 2 *
- d. 1
- e. Ý kiến khác.

Câu 16. Cho kích thước vùng đệm (*buffer*) là $bf = 5$ khối (*blocks*). Tập tin dữ liệu Employee gồm 16 khối dữ liệu ($b_E = 16$ blocks). Tập tin dữ liệu Department gồm 7 khối dữ liệu ($b_D = 7$ blocks). Thực hiện Employee $\bowtie_{DNO=DNUMBER}$ Department (phép kết với điều kiện kết bằng giữa thuộc tính DNO của Employee và DNUMBER của Department). Tập tin dữ liệu nào nên được duyệt ở vòng lặp ngoài (*outer loop*) nếu phép kết này được xử lý bằng phương pháp hai vòng lặp lồng (*nested-loop join*)?

- a. Department do tổng số block accesses trên các tập tin dữ liệu là 112 (nhiều nhất).
- b. Employee do tổng số block accesses trên các tập tin dữ liệu là 58 (nhiều nhất).
- c. Department do tổng số block accesses trên các tập tin dữ liệu là 55 (ít nhất). *
- d. Employee do tổng số block accesses trên các tập tin dữ liệu là 32 (ít nhất).
- e. Ý kiến khác.

Câu 17. Cho câu truy vấn SQL dùng để lấy ra các cặp mã nhân viên của những nhân viên ở cùng phòng và có cùng lương từ tập tin dữ liệu Nhân_Viên ở **Bảng 1** như sau:

```
SELECT n1.ID, n2.ID
FROM Nhân_Viên n1, Nhân_Viên n2
WHERE n1.Phòng = n2.Phòng AND n1.Lương = n2.Lương;
```

Xác định **biểu thức đại số quan hệ** (*relational algebra expression*) tương ứng của câu truy vấn này.

- $\pi_{n1.ID, n2.ID}(\sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}((n1 \leftarrow \text{Nhân_Viên})) \times \sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}((n2 \leftarrow \text{Nhân_Viên})))$
- $\pi_{n1.ID, n2.ID}(\sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}(((n1 \leftarrow \text{Nhân_Viên}) \times (n2 \leftarrow \text{Nhân_Viên})))$
- $\pi_{n1.ID, n2.ID}((n1 \leftarrow \text{Nhân_Viên}) \times \sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương} \pi_{n1.ID, n2.ID} (n2 \leftarrow \text{Nhân_Viên}))$
- Không thể xác định được biểu thức đại số quan hệ do còn thiếu mô tả về các vùng tin khác và các dạng chỉ mục trên tập tin Nhân_Viên này.
- Ý kiến khác.

Câu 18. Cho câu truy vấn SQL dùng để lấy ra các bản ghi của những nhân viên có lương lớn hơn 2000 từ tập tin dữ liệu Nhân_Viên ở **Bảng 1** như sau:

SELECT *

FROM Nhân_Viên

WHERE Lương > 2000;

Giả sử hiện tại tập tin dữ liệu Nhân_Viên là tập tin có thứ tự (*ordered file*) với các bản ghi được sắp thứ tự theo các giá trị của vùng tin khóa ID. Một chỉ mục B được định nghĩa trên vùng tin khóa ID. Xác định **phương thức tìm kiếm** (*search method*) có thể được dùng để trả lời câu truy vấn SQL này.

- Tìm kiếm tuần tự (*linear search*) *
- Tìm kiếm nhị phân (*binary search*)
- Tìm kiếm dùng chỉ mục sơ cấp (*primary index*)
- Tìm kiếm dùng chỉ mục thứ cấp (*secondary index*)
- Ý kiến khác.

Câu 19. Cho 2 quan hệ Nhân_Viên và Phòng_Ban có lược đồ như sau.

Nhân_Viên (ID, Tên, Lương, Phòng, Địa_Chỉ);

Phòng_Ban (ID_Phòng, Tên_Phòng, ID_Quản_Lý)

Trong đó, ID là khóa chính của quan hệ Nhân_Viên và Phòng là khóa ngoại của quan hệ Nhân_Viên tham chiếu đến ID_Phòng của quan hệ Phòng_Ban; ID_Phòng là khóa chính của quan hệ Phòng_Ban và Tên_Phòng là khóa dự tuyển của quan hệ Phòng_Ban.

Câu lệnh SQL sau dùng để truy vấn tên của tất cả nhân viên làm việc ở phòng 'RESEARCH' có lương lớn hơn 2000:

SELECT Tên
FROM Nhân_Viên, Phòng_Ban
WHERE Phòng = ID_Phòng
AND Tên_Phòng = 'RESEARCH'
AND Lương > 2000;

Hãy cho biết **cây truy vấn** (*query tree*) nào sau đây **KHÔNG** là dạng biểu diễn của câu lệnh SQL trên?

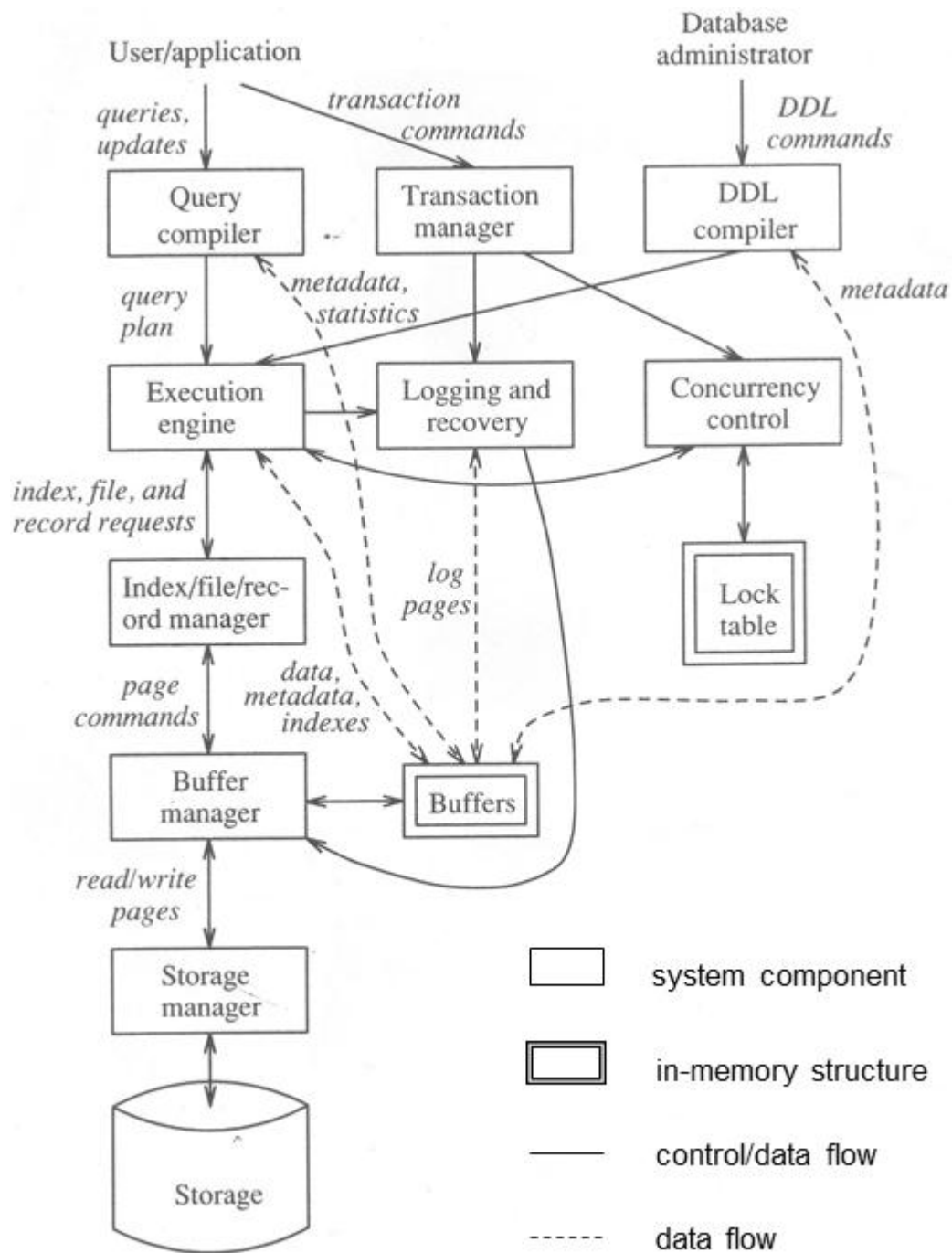
- Hình 9
- Hình 10 *
- Hình 11
- Hình 12
- Ý kiến khác.

Câu 20. Cho đoạn mã chương trình trong **Hình 13**. Xác định *tác vụ* được hiện thực trên hai tập tin (quan hệ) R và S bởi đoạn mã chương trình này.

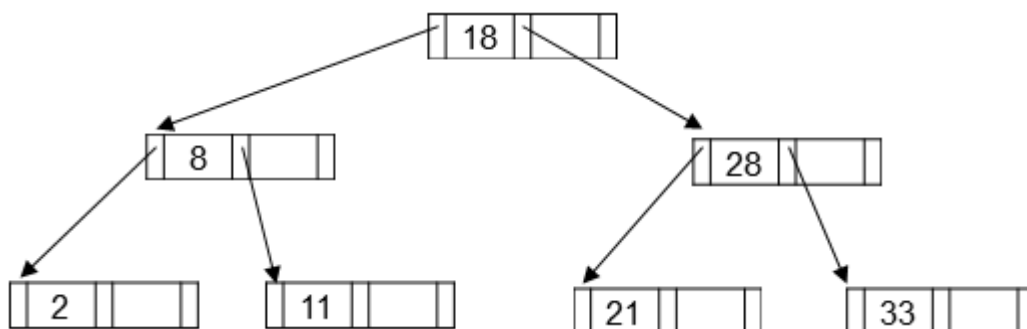
- Kết bằng (*equi join*) *
- Hội (*union*)
- Giao (*intersection*)
- Hiệu (*difference*)
- Ý kiến khác.

Bảng 1 - Tập tin dữ liệu Nhân_Viên tương ứng với bảng Nhân_Viên trong cơ sở dữ liệu

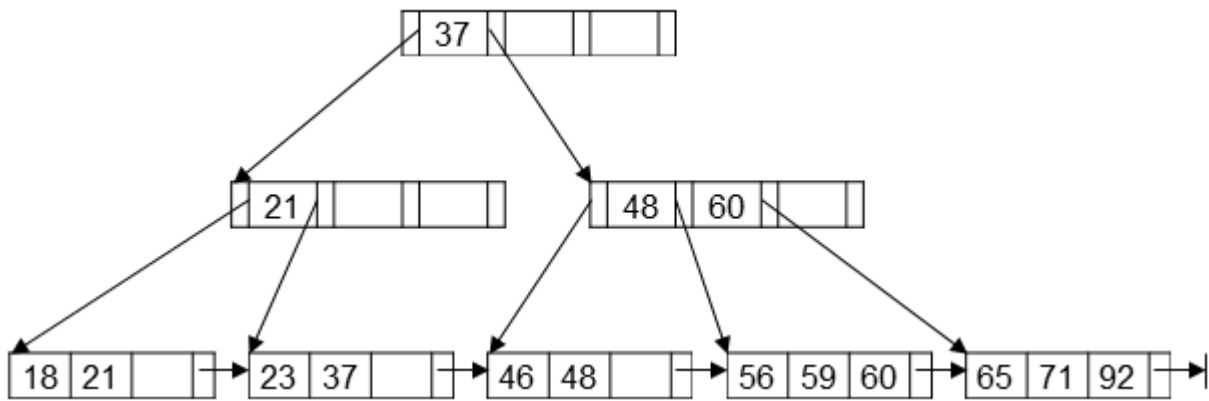
ID	Tên	Lương	Phòng	...
2	An	2000
5	Đông	1000
8	Hạ	2500
10	Phúc	1500
11	Hoa	1800
15	Tuấn	2000
18	Cường	1900
19	Minh	2100
21	Mai	2500
23	Đào	2300
28	Xuân	1600
30	Dũng	2000
33	Thu	2300



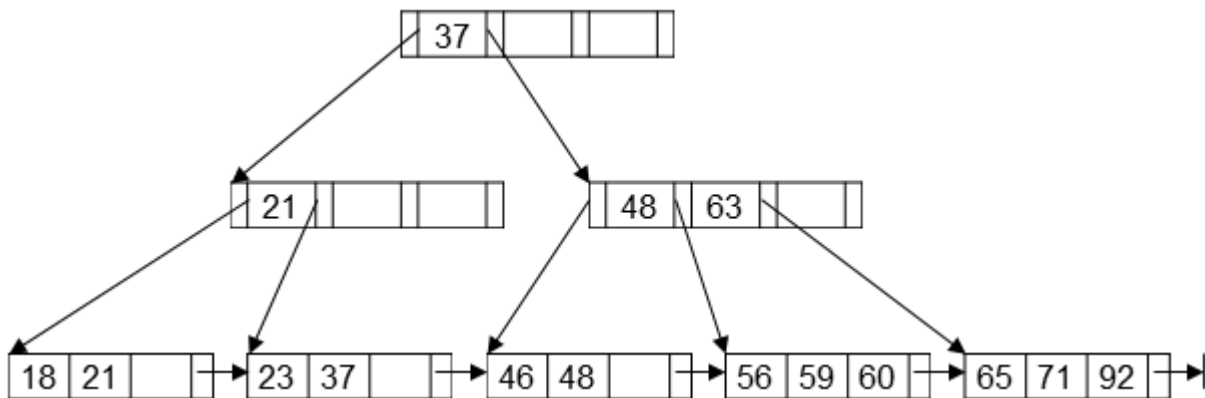
Hình 1 – Kiến trúc hệ quản trị cơ sở dữ liệu (architecture of a database management system) (Câu 2)



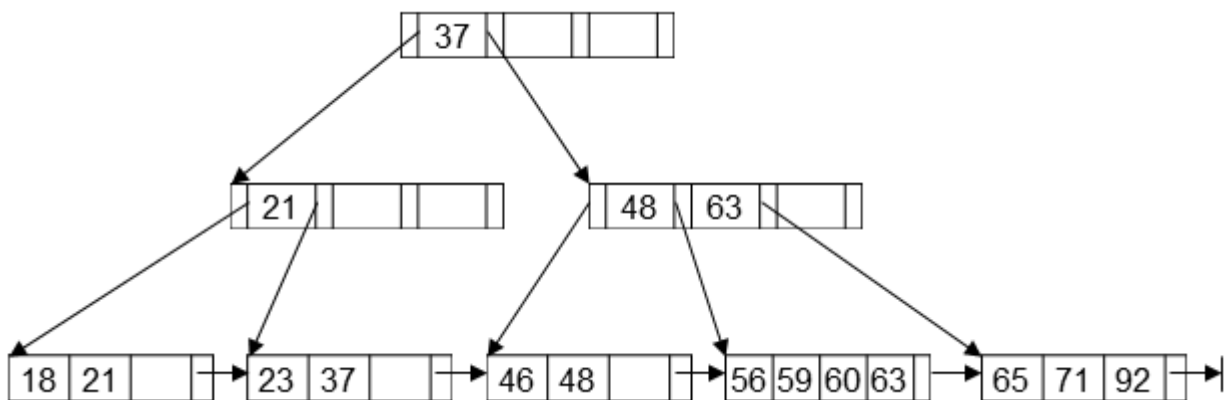
Hình 2 – Cây chỉ mục B trên vùng tin ID (khóa (key), sắp thứ tự (ordering field)) của tập tin dữ liệu Nhân_Viên trong Bảng 1 trước khi xóa 28 (Câu 7)



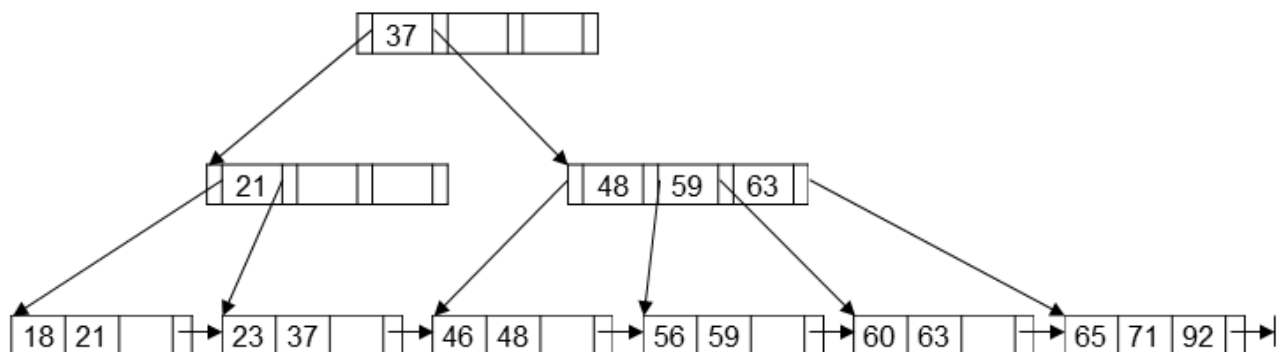
Hình 3 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{leaf} = 3$ trước khi thêm giá trị tìm kiếm 63 (Câu 8, 9, 10)



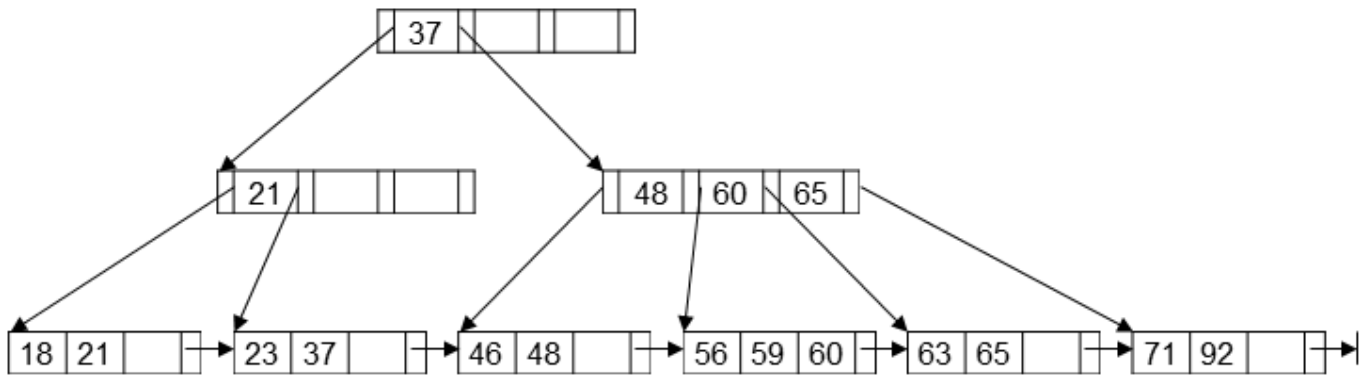
Hình 4 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{leaf} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. a)



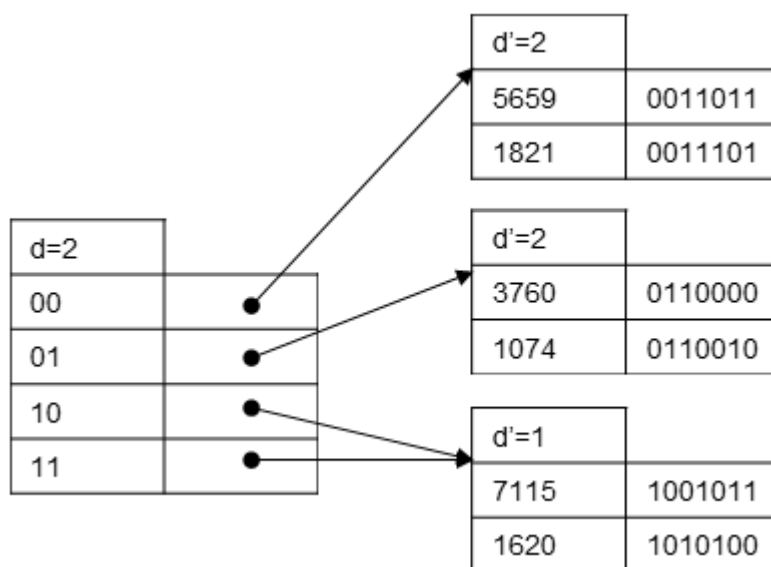
Hình 5 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{leaf} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. b)



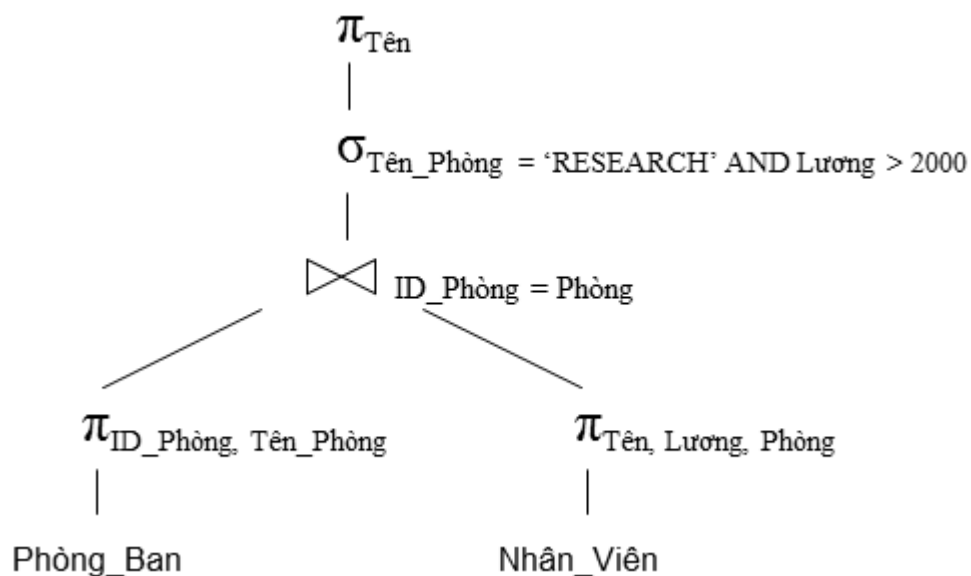
Hình 6 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. c)



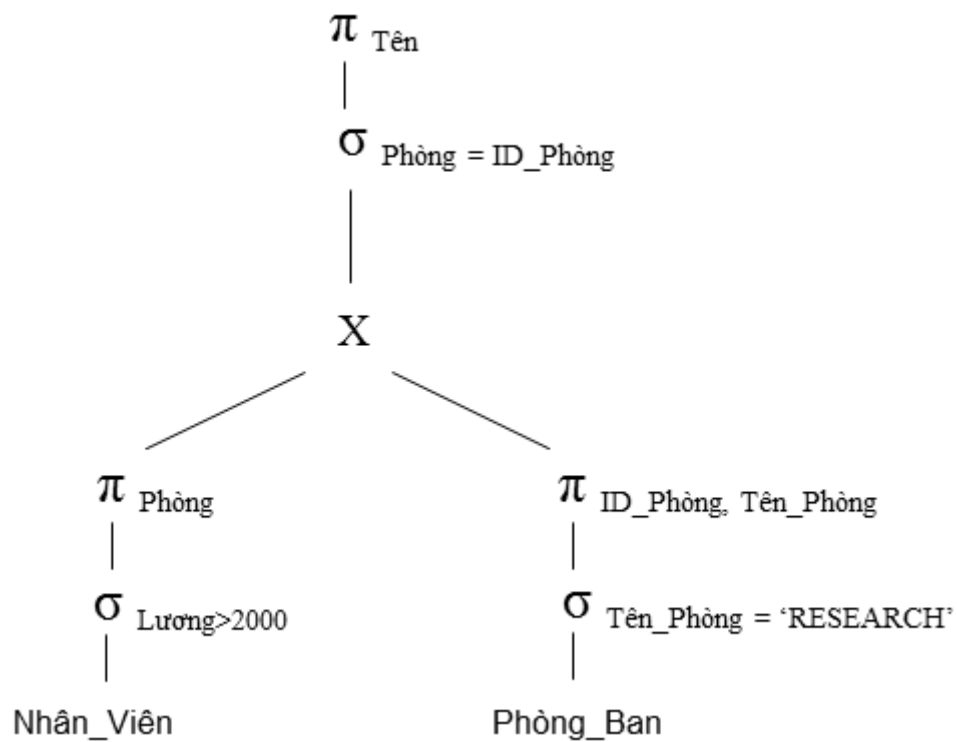
Hình 7 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. d)



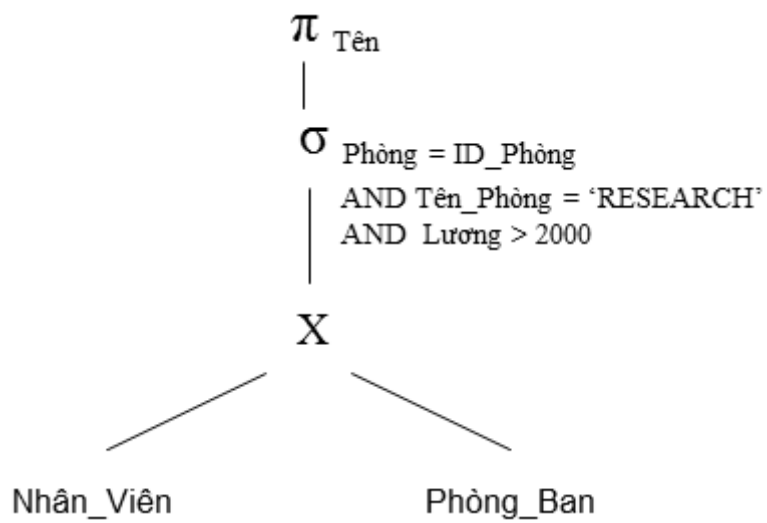
Hình 8 – Tập tin dữ liệu được tổ chức dạng băm động dùng extendible hashing với hàm băm $h(K) = K \bmod 128$ (Câu 14, 15)



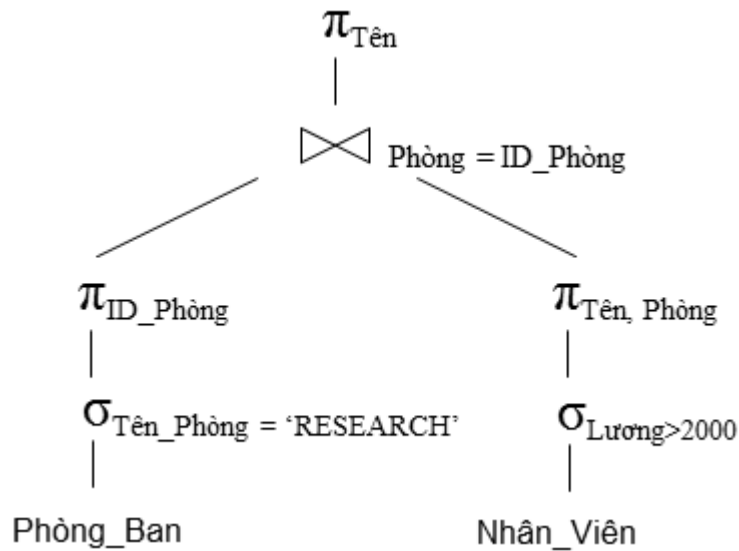
Hình 9 – Cây truy vấn kết quả (Câu 19. a)



Hình 10 – Cây truy vấn kết quả (Câu 19. b)



Hình 11 – Cây truy vấn kết quả (Câu 19. c)



Hình 12 – Cây truy vấn kết quả (Câu 19. d)

sort the tuples in R on attribute A; /* assume R has n tuples */

sort the tuples in S on attribute B; /* assume S has m tuples */

set $i \leftarrow 1, j \leftarrow 1$;

while ($i \leq n$) and ($j \leq m$)

do { if $R(i)[A] > S(j)[B]$

then set $j \leftarrow j + 1$

elseif $R(i)[A] < S(j)[B]$

then set $i \leftarrow i + 1$

else { /* output a matched tuple */

output the combined tuple $\langle R(i), S(j) \rangle$ to T;

/* output other tuples that match R(i), if any */

set $l \leftarrow j + 1$;

while ($l \leq m$) and ($R(i)[A] = S(l)[B]$)

do { output the combined tuple $\langle R(i), S(l) \rangle$ to T;

set $l \leftarrow l + 1$

}

/* output other tuples that match S(j), if any */

set $k \leftarrow i+1$

while ($k \leq n$) and ($R(k)[A] = S(j)[B]$)

do { output the combined tuple $\langle R(k), S(j) \rangle$ to T;

set $k \leftarrow k + 1$

}

set $i \leftarrow i+1, j \leftarrow j+1$;

}

}

Hình 13 - Đoạn mã chương trình hiện thực tác vụ trên 2 tập tin (quan hệ) R và S (Câu 20)

Họ - Tên:

Mã Số Sinh Viên:

Phần trả lời

Câu hỏi	a	b	c	d	e (Ý kiến khác)
1			X		
2	X				
3				X	
4				X	
5			X		
6				X	
7		X			
8				X	
9		X			
10	X				
11	X				
12			X		
13				X	
14		X			
15			X		
16			X		
17		X			
18	X				
19		X			
20	X				

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu** (503004)

Học kỳ 1 - 2012-2013

Ngày thi: 17/10/2012

Phòng thi: 503 C4

Mã đề: **01**

Họ - Tên:

Mã Số Sinh Viên:

Phần giải thích:

Câu 4. Số truy đạt khối ít nhất để thực hiện việc tìm kiếm bản ghi của nhân viên có mã ID = 29 trên tập tin dữ liệu Nhân_Viên là **2**.

Giải thích: Vùng tin ID là vùng tin khóa, sắp thứ tự các bản ghi của tập tin dữ liệu Nhân_Viên. Do đó, việc tìm kiếm bản ghi của nhân viên có mã ID = 29 chính là tìm kiếm với điều kiện “=” trên vùng tin ID và được thực hiện bằng phương pháp tìm kiếm nhị phân (*binary search*).

Truy đạt khối đầu tiên được thực hiện với khối dữ liệu (18, 19) ở chính giữa. Do $19 < 29$ nên việc tìm kiếm được thực hiện cho phần nửa lớn hơn.

Truy đạt khối thứ hai được thực hiện với khối dữ liệu (28, 30) ở chính giữa của phần nửa lớn hơn. Do $28 < 29 < 30$ nên kết quả tìm kiếm trả về là không tìm thấy bản ghi của nhân viên có mã ID = 29.

Tổng kết lại số truy đạt khối dữ liệu đã được thực hiện là 2.

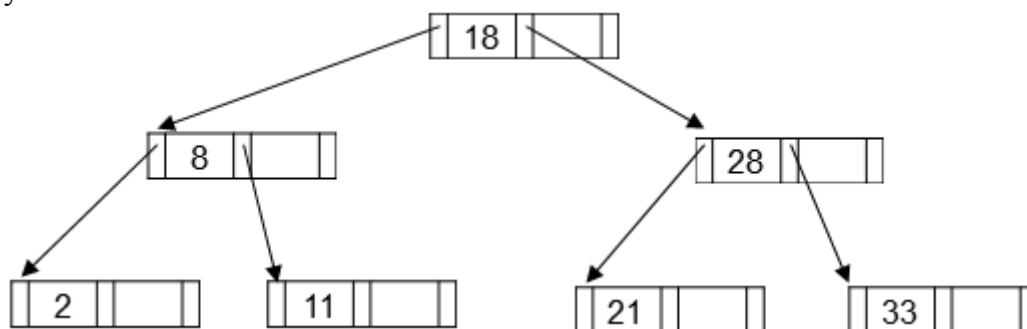
Câu 5. Số truy đạt khối ít nhất để thực hiện việc tìm kiếm bản ghi của nhân viên có Tên = Việt trên tập tin dữ liệu Nhân_Viên là **7**.

Giải thích: Vùng tin Tên là vùng tin không khóa, không sắp thứ tự các bản ghi của tập tin dữ liệu Nhân_Viên. Do đó, việc tìm kiếm bản ghi của nhân viên có Tên = Việt chính là tìm kiếm với điều kiện “=” trên vùng tin không khóa, không sắp thứ tự và được thực hiện tuần tự (*linear search*) từng đầu đến cuối tập tin và dừng lại khi tìm thấy bản ghi nếu tồn tại.

Do vùng tin Tên không có giá trị “Việt” nên số khối truy đạt là 7 khi việc tìm kiếm được thực hiện lần lượt từ khối dữ liệu đầu tiên đến khối dữ liệu cuối cùng.

Câu 7. Cây chỉ mục **B** ở **Hình 2** sau khi xóa giá trị tìm kiếm **28** sẽ có chiều cao giảm 1, số node lá giảm 1, và số node nội giảm 2.

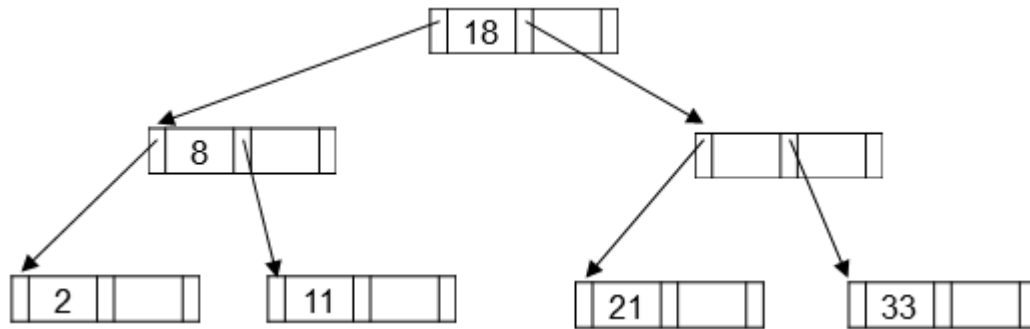
Cây B ban đầu:



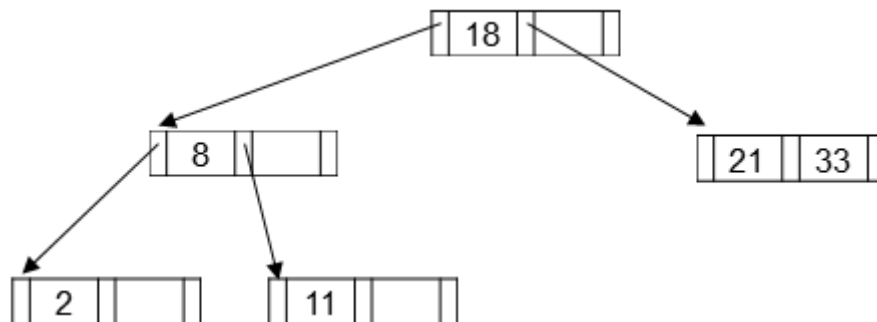
Định vị và xóa giá trị tìm kiếm 28 làm node nội này bị tràn dưới (*underflow*):

Họ - Tên:

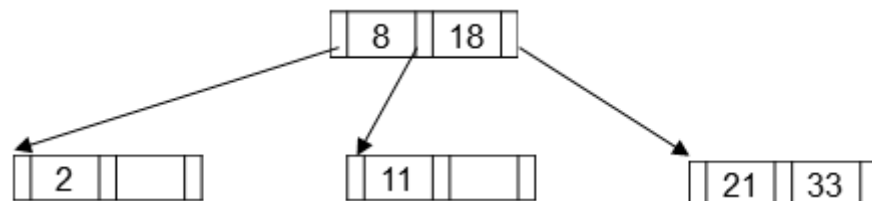
Mã Số Sinh Viên:



Trộn các node con tương ứng thành 1 node mới nên làm giảm số node nội ban đầu và cũng làm giảm số node lá sau khi trộn:



Thực hiện cân bằng cây để tất cả các node lá đều ở cùng mức nên làm giảm chiều cao của cây và đồng thời giảm số node nội.



Kết luận: Cây kết quả đạt được có chiều cao giảm 1, số node lá giảm 1, và số node nội giảm 2 so với cây B ban đầu.

Câu 8. Cây chỉ mục B+ kết quả đạt được sau khi thêm giá trị tìm kiếm 63 vào cây chỉ mục B+ này ở Hình 7.

Giải thích:

- Định vị vị trí của 63 trong cây B+
 - Khối đầu tiên (root) của B+-tree được truy đạt và kiểm tra với 63. Do $37 < 63$ nên đi về cây con bên phải của 37.
 - Khối đầu tiên (root) của cây con bên phải của 37 được truy đạt và kiểm tra với 63. Do $60 < 63$ nên đi về cây con bên phải của 60.
 - Khối đầu tiên (root) của cây con bên phải của 60 được truy đạt và kiểm tra với 63. Khối này là node lá nên đây chính là khối mà 63 sẽ được chèn vào.
- Chèn 63 vào node lá

Họ - Tên:

Mã Số Sinh Viên:

- Node lá (65, 71, 92) đã đầy nên việc chèn 63 vào gây nên việc tràn trên (overflow). Do đó, tách node lá này thành 2 node: (63, 65) và (71, 92). Hai con trỏ cây của trị tìm kiếm 65 sẽ giữ 2 node lá mới.
- Trị tìm kiếm 65 được chèn vào node nội mà trước đó chứa con trỏ cây đến node lá mới vừa được tách. Do đó, 65 được chèn vào node (48, 60) thành node (48, 60, 65).

Tổng kết, việc chèn 63 vào B+-tree này làm cho số node lá tăng lên 1; nhưng số node nội và chiều cao của cây không bị ảnh hưởng.

Câu 9. Số khối dữ liệu cần thiết để lưu trữ cây chỉ mục B+ này chính là số node của cây = 8. Lý do: mỗi node của cây chỉ mục được chứa trọn trong một khối dữ liệu.

Câu 10. Xác định số truy đạt khối cần thực hiện để đọc được bản ghi có giá trị vùng tin lập chỉ mục là 37.

- 1 block access vào nút gốc (37) của cây để đi về nhánh trái
- 1 block access vào nút nội (21) để đi về nhánh phải
- 1 block access vào nút lá (23, 37) để lấy con trỏ dữ liệu đến bản ghi có giá trị vùng tin lập chỉ mục là 37
- 1 block access vào khối chứa bản ghi 37

Kết luận: 4 block accesses cần thực hiện để đọc được bản ghi có giá trị vùng tin lập chỉ mục là 37 = số mức của cây B+-tree (kể cả mức của nút gốc) + 1.

Câu 12. Nếu các khối dữ liệu của tập tin dữ liệu này được tổ chức liên tục (*contiguous*) ở trên cùng một rãnh (*track*) thì tổng thời gian thực hiện chuyển các khối dữ liệu vào vùng nhớ chính khi đọc hết dữ liệu theo thứ tự của các giá trị ở vùng tin ID là ***bao nhiêu msec?***

Tổng số khối dữ liệu cần đọc = $b = 7$ blocks.

Do các khối được tổ chức liên tục (*contiguous*) trên cùng 1 rãnh (*track*) nên tổng thời gian thực hiện chuyển các khối vào vùng nhớ chính là $s + rd + b \cdot btt$; trong đó, s là seek time, rd là rotational delay, và btt là block transfer time.

Seek time = $s = 20$ msec

Rotational delay = $rd = (1/(2 \cdot p)) \cdot 60 \cdot 1000$ msec = $(1/(2 \cdot 5600)) \cdot 60 \cdot 1000 = 5.36$ msec

Transfer rate = $tr = \text{track size/time per rotation} = T/((1/p) \cdot 60 \cdot 1000) = 102400 \cdot 5600 / (60 \cdot 1000) = 9557.33$ bytes/msec

Block transfer time = $btt = \text{block size/transfer rate} = B/tr = 512/9557.33 = 0.05$ msec

Tổng thời gian = $s + rd + b \cdot btt = 20 + 5.36 + 7 \cdot 0.05 = \mathbf{25.71}$ msec

Câu 13. Hệ số phân khối của tập tin dữ liệu Student này theo cách phân khối không phủ (*unspanned blocking*) = $\lfloor B/R \rfloor$; trong đó, B là block size và R là record size.

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu** (503004)

Học kỳ 1 - 2012-2013

Ngày thi: 17/10/2012

Phòng thi: 503 C4

Mã đề: **01**

Họ - Tên:

Mã Số Sinh Viên:

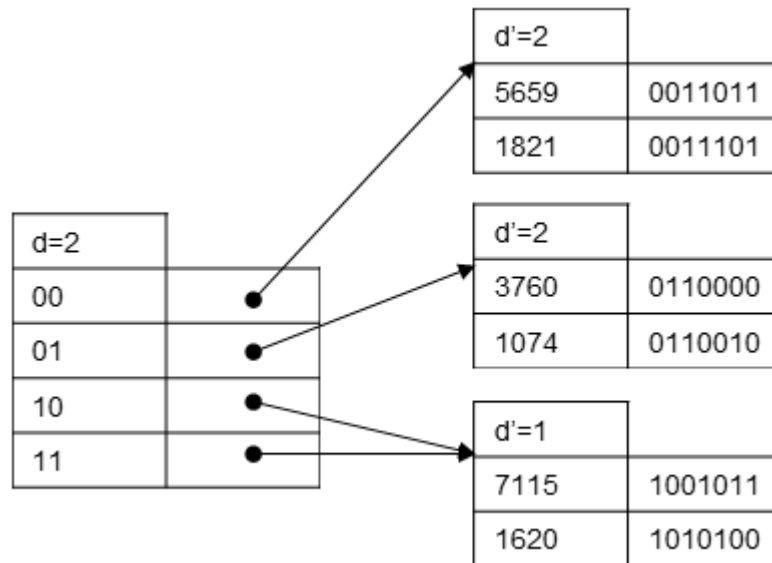
Block size = B = 512 bytes

Record size = R = 30 + 9 + 40 + 9 + 8 + 1 + 4 + 4 + 4 + 3 + 1 = 113 bytes

Blocking factor = bfr = $\lfloor B/R \rfloor = \lfloor 512/113 \rfloor = 4 \text{ records/block}$.

Câu 14. Độ sâu toàn cục (global depth, d) của toàn tập tin và **độ sâu cục bộ (local depth, d')** của thùng chứa bản ghi **3943** sau khi thêm bản ghi này vào tập tin

Tập tin được tổ chức dạng extendible hashing trước khi thêm bản ghi 3943:



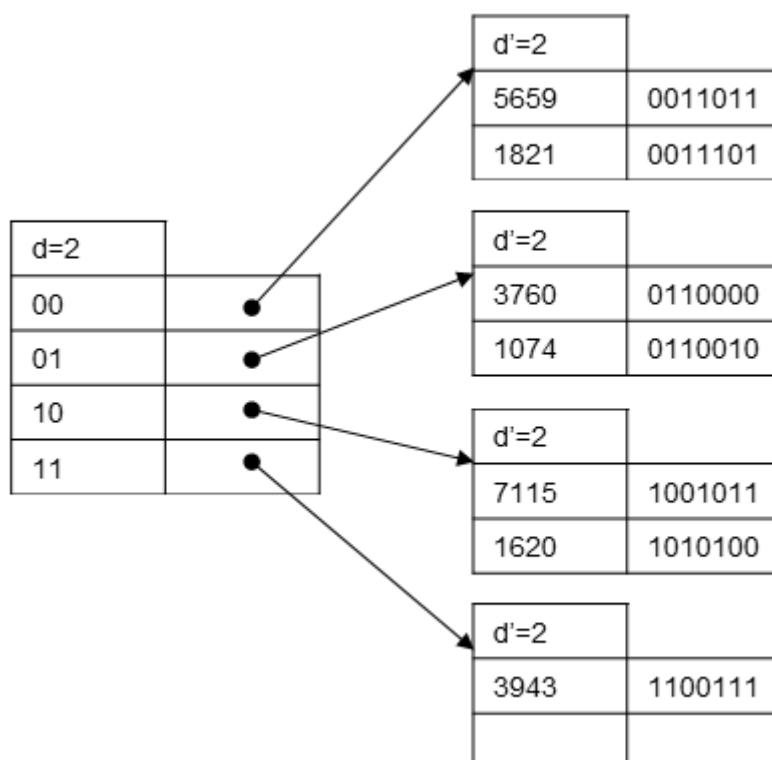
Định vị thùng sẽ chứa bản ghi 3943: $h(3943) = 3943 \bmod 128 = 103_{10} = 1100111_2 \rightarrow$ thùng sẽ chứa bản ghi 3943 là thùng có bit dẫn đầu là “1” tương ứng $d' = 1$. Việc thêm bản ghi vào thùng này làm cho thùng bị tràn (overflow) nên việc tách thùng được tiến hành.

Do $d' = 1 < d = 2$ nên chỉ mỗi thùng này được tách đôi và thư mục được giữ nguyên; nghĩa là d' trở thành d'_{10} và d'_{11} có giá trị là 2 và $d = 2$ được giữ nguyên. Các bản ghi hiện tại 7115 và 1620 được phân bổ lại trong hai thùng “10” và “11” dùng 2 bit đầu của kết quả băm lấy trị nhị phân. Sau đó, phân bổ bản ghi mới 3943 vào thùng “11” do 2 bit đầu tương ứng là “11”.

Kết quả sau khi thêm bản ghi 3943 là:

Họ - Tên:

Mã Số Sinh Viên:



Câu 15. Số truy đạt khối (block access) cần thực hiện để đạt được bản ghi 3760 là tổng số khối truy đạt vào thư mục và sau đó truy đạt vào thùng/khối chứa bản ghi 3760 này.

Do thư mục (directory) được chứa trọn trong 1 khối nên cần 1 truy đạt khối đến thư mục này.

Do độ sâu toàn cục $d = 2$ và $h(3760) = 3760 \bmod 128 = 48_{10} = 0110000_2$ nên sử dụng 2 bit đầu **01** để xác định được thùng chứa bản ghi 3760. Do đó, cần 1 truy đạt khối đến thùng chứa bản ghi 3760 cũng chính là đến khối (block) chứa bản ghi 3760 (vì 1 thùng chứa 1 khối dữ liệu, 1 khối dữ liệu chứa 2 bản ghi).

Tổng kết: số truy đạt khối cần thực hiện là **2 block accesses**.

Ghi chú: nếu trước đó, thư mục đã được đưa vào vùng nhớ chính thì khi này, 1 truy đạt khối đến thư mục không cần được thực hiện nữa và do đó, chỉ tốn 1 truy đạt khối từ thư mục đến thùng chứa bản ghi 3760.

Câu 16. Department nên được duyệt ở vòng lặp ngoài (outer loop) do tổng số block accesses trên các tập tin dữ liệu nhập khi đó là 53 (ít nhất).

Giải thích:

. Department ở vòng lặp ngoài (outer loop)

$$b_D + \lceil b_D / (b_f - 2) \rceil * b_E = 7 + 3 * 16 = 55 \text{ block accesses}$$

. Employee ở vòng lặp ngoài (outer loop)

$$b_E + \lceil b_E / (b_f - 2) \rceil * b_D = 16 + 6 * 7 = 58 \text{ block accesses}$$

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu (503004)**

Học kỳ 1 - 2012-2013

Ngày thi: 17/10/2012

Phòng thi: 503 C4

Mã đề: **01**

Kiểm tra giữa kỳ

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu (503004)** - Ngành: **Khoa Học Máy Tính – HK1 – 2012-2013**

Thời gian làm bài: **60** phút

(Bài kiểm tra gồm 20 câu hỏi. Sinh viên được **tham khảo ghi chú** trong 2 tờ giấy A4.)

Sinh viên chọn 1 câu trả lời đúng nhất. Nếu chọn câu (e) thì sinh viên cần trình bày đáp án khác so với đáp án ở các câu (a), (b), (c), và (d) và/hoặc giải thích lựa chọn (e) của mình.

Câu 1. *Hệ quản trị cơ sở dữ liệu (database management system, DBMS) là gì?*

- a. DBMS là một phần mềm hệ thống dùng giúp các chương trình ứng dụng được thực thi hiệu quả hơn.
- b. DBMS là một phần mềm chuyên dụng dùng giúp quản lý dữ liệu của các chương trình ứng dụng hiệu quả hơn.
- c. DBMS là một phần mềm ứng dụng dùng giúp tận dụng sự hỗ trợ của hệ điều hành để tạo và quản lý lượng lớn dữ liệu và các chương trình ứng dụng hiệu quả theo thời gian.
- d. DBMS là một phần mềm chuyên dụng dùng giúp tạo và quản lý một lượng lớn dữ liệu hiệu quả và bền vững an toàn theo thời gian. *
- e. Ý kiến khác.

Câu 2. Cho kiến trúc của một DBMS trong **Hình 1**. Cho câu lệnh “**CREATE INDEX** composite_index **ON** Nhân_Viên (Họ_Tên, Lương);” được gửi đến DBMS; trong đó, Nhân_Viên là bảng dữ liệu với chi tiết được trình bày trong **Bảng 1**. Chọn phát biểu **ĐÚNG** về quá trình xử lý câu lệnh tạo chỉ mục phức hợp này.

- a. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên thành phần Query Preprocessor của mô-đun Query compiler thông báo lỗi khi truy vấn trên bảng Nhân_Viên được thực hiện.
- b. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên mô-đun DDL compiler thông báo lỗi khi phân tích cú pháp và kiểm tra ngữ nghĩa của câu lệnh tạo chỉ mục này. *
- c. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên mô-đun Execution engine thông báo lỗi khi thực hiện câu lệnh tạo chỉ mục này.
- d. Do vùng tin Họ_Tên không đúng là vùng tin Tên của bảng Nhân_Viên nên mô-đun Buffer manager thông báo lỗi khi tìm kiếm siêu dữ liệu (metadata) về bảng Nhân_Viên.
- e. Ý kiến khác.

Câu 3. Cây chỉ mục **B** (*B-tree*) và cây chỉ mục **B+** (*B+-tree*) có những đặc điểm gì sau đây **khác nhau**?

- a. B-tree dùng làm chỉ mục sơ cấp và B+-tree dùng làm chỉ mục thứ cấp.
- b. B-tree chỉ hỗ trợ toán tử =, <, và >; trong khi đó, B+-tree hỗ trợ toán tử =, <, <=, >, >=, và between.
- c. B-tree dùng làm chỉ mục đa mức tĩnh và B+-tree dùng làm chỉ mục đa mức động.
- d. Cấu trúc của node nội và node lá trong B-tree giống nhau; trong khi đó, cấu trúc của node nội và node lá trong B+-tree khác nhau và do đó, khả năng chỉ mục của B+-tree thường hiệu quả hơn. *
- e. Ý kiến khác.

Câu 4. Cho tập tin dữ liệu Nhân_Viên gồm 7 khối dữ liệu (7 blocks) trong **Bảng 1**. Mỗi khối chứa 2 bản ghi (record). Vùng tin ID là vùng tin khóa (key field) của tập tin; các vùng tin còn lại là vùng tin phi khóa (non-key field). Ngoài ra, các bản ghi (record) trong tập tin này được sắp thứ tự vật lý theo các giá trị tăng

dẫn của vùng tin ID. Xác định **số truy đạt khối ít nhất** (*the minimum number of block accesses*) có thể để thực hiện việc tìm kiếm bản ghi của nhân viên có mã ID = 29 trên tập tin dữ liệu Nhân_Viên.

- a. 2 *
- b. 3
- c. 4
- d. 7
- e. Ý kiến khác.

Câu 5. Cho tập tin dữ liệu Nhân_Viên gồm 7 khối dữ liệu (7 blocks) trong **Bảng 1**. Mỗi khối chứa 2 bản ghi (*record*). Vùng tin ID là vùng tin khóa (*key field*) của tập tin; các vùng tin còn lại là vùng tin phi khóa (*non-key field*). Ngoài ra, các bản ghi trong tập tin này được sắp thứ tự vật lý theo các giá trị tăng dần của vùng tin ID. Xác định **số truy đạt khối ít nhất** (*the minimum number of block accesses*) có thể để thực hiện việc tìm kiếm bản ghi của nhân viên có Tên = Phong trên tập tin dữ liệu Nhân_Viên.

- a. 4
- b. 7 *
- c. 8
- d. 13
- e. Ý kiến khác.

Câu 6. Công nghệ RAID nào sau đây sử dụng cơ chế rải dữ liệu (*data striping*) ở mức khối (*block-level*) để cải thiện hiệu suất ghi/đọc đĩa với song song hóa truy đạt đĩa (*parallelizing disk access*)?

- a. Rải dữ liệu chỉ được thực hiện ở mức bit (*bit-level*) nên không có RAID nào sử dụng cơ chế rải dữ liệu ở mức khối.
- b. RAID 1
- c. RAID 3
- d. RAID 5 *
- e. Ý kiến khác.

Câu 7. Cho **Hình 2** là cây chỉ mục B có bậc $p = 3$ trên vùng tin ID (khóa (*key*), sắp thứ tự (*ordering field*)) của tập tin dữ liệu Nhân_Viên ở **Bảng 1**. Cây chỉ mục **B** sau khi xóa (*delete*) giá trị tìm kiếm **8** sẽ có **đặc điểm** gì?

- a. Chiều cao cây giảm 1, số node lá giảm 1, và số node nội cũng giảm 1
- b. Chiều cao cây giảm 1, số node lá giảm 1, và số node nội giảm 2 *
- c. Chiều cao cây không đổi, số node lá giảm 1, và số node nội cũng không đổi
- d. Chiều cao cây không đổi, số node lá cũng không đổi, và số node nội giảm 1
- e. Ý kiến khác.

Câu 8. Cho cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở **Hình 3**. Xác định **cây chỉ mục B+ kết quả** đạt được sau khi thêm giá trị tìm kiếm **63** vào cây chỉ mục B+ này.

- a. **Hình 4**
- b. **Hình 5**
- c. **Hình 6** *
- d. **Hình 7**
- e. Ý kiến khác.

Câu 9. Cho cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở **Hình 3**. Xác định *số khối dữ liệu* (number of blocks) cần thiết để lưu trữ cây chỉ mục B+ này.

- a. 3
- b. 5
- c. 8 *
- d. 16
- e. Ý kiến khác.

Câu 10. Cho cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở **Hình 3**. Xác định *số truy đạt khối* (number of block accesses) cần được thực hiện để đạt được khối dữ liệu chứa bản ghi có giá trị tìm kiếm là **21**.

- a. 1
- b. 2
- c. 3
- d. 4 *
- e. Ý kiến khác.

Câu 11. Kỹ thuật bộ đệm đôi (double buffering) được sử dụng hiệu quả khi các khối dữ liệu cần được đọc/ghi được *tổ chức* dạng nào?

- a. Liên tục (contiguous) *
- b. Liên kết (linked)
- c. Cụm (clustered)
- d. Chỉ mục (indexed)
- e. Ý kiến khác.

Câu 12. Cho tập tin dữ liệu Nhân_Viên ở **Bảng 1** được lưu trữ vật lý trên bộ đĩa từ có số vòng quay trong 1 phút là $p = 5600$ rpm, có thời gian tìm kiếm trung bình (average seek time) là $s = 20$ msec. Kích thước một khối dữ liệu (block size) là $B = 512$ bytes. Kích thước một rãnh dữ liệu (track size) là $T = 102400$ bytes. Nếu các khối dữ liệu của tập tin dữ liệu này được tổ chức liên tục (contiguous) ở trên cùng một rãnh (track) thì tổng thời gian thực hiện chuyển các khối dữ liệu vào vùng nhớ chính khi đọc hết dữ liệu theo thứ tự của các giá trị ở vùng tin ID là *bao nhiêu msec*? Giả sử kết quả của các phép toán được làm tròn với 2 chữ số thập phân.

- a. 25.41 msec
- b. 25.71 msec *
- c. 57.87 msec
- d. 177.87 msec
- e. Ý kiến khác.

Câu 13. Cho tập tin dữ liệu Student gồm 20000 bản ghi ($r = 20000$ records) có chiều dài cố định (fixed length). Mỗi bản ghi gồm các vùng tin như sau: NAME (30 bytes), SSN (9 bytes), ADDRESS (40 bytes), PHONE (9 bytes), BIRTHDATE (8 bytes), SEX (1 byte), MAJORDEPTCODE (4 bytes), MINORDEPTCODE (4 bytes), CLASSCODE (4 bytes), DEGREEPROGRAM (3 bytes), và DELETION MARKER (1 byte). Xác định *hệ số phân khối* (blocking factor, **bfr**) của tập tin dữ liệu Student này khi được lưu trữ trên đĩa từ có kích thước khối dữ liệu (block size, **B**) là $B = 512$ bytes theo cách phân khối không phủ (unspanned blocking).

- a. 4 *
- b. 4.5

- c. 5
- d. 5000
- e. Ý kiến khác.

Câu 14. Trong **Hình 8**, cho tập tin dữ liệu được tổ chức dưới dạng băm động (*dynamic hashing*) dùng băm khả mở (*extendible hashing*) với các bit dẫn đầu (*leading bits*) được sử dụng và hàm băm (hash function) là $h(K) = K \bmod 128$. Biết rằng: mỗi thùng (*bucket*) chứa 1 khối dữ liệu (block), mỗi khối dữ liệu chứa được 2 bản ghi (*record*). Xác định **độ sâu toàn cục** (*global depth, d*) của toàn tập tin và **độ sâu cục bộ** (*local depth, d'*) của thùng chứa bản ghi **3943** sau khi thêm bản ghi này vào tập tin. Cho trước $h(3943) = 3943 \bmod 128 = 103_{10} = 1100111_2$.

- a. $d = 3$ và $d' = 2$
- b. $d = 3$ và $d' = 3$
- c. $d = 2$ và $d' = 1$
- d. $d = 2$ và $d' = 2$
- e. Ý kiến khác.

Câu 15. Giả sử thư mục (*directory*) của tập tin băm (*hashed file*) ở **Hình 8** này được chứa trọn trong 1 khối dữ liệu. Xác định **số truy đạt khối** (*number of block accesses*) được thực hiện để đạt được bản ghi **3760**.

- a. 1
- b. 2
- c. 3
- d. 4
- e. Ý kiến khác.

Câu 16. Cho kích thước vùng đệm (*buffer*) là $bf = 5$ khối (*blocks*). Tập tin dữ liệu Employee gồm 16 khối dữ liệu ($b_E = 16$ blocks). Tập tin dữ liệu Department gồm 7 khối dữ liệu ($b_D = 7$ blocks). Thực hiện Employee $\bowtie_{DNO=DNUMBER}$ Department (phép kết với điều kiện kết bằng giữa thuộc tính DNO của Employee và DNUMBER của Department). Tập tin dữ liệu nào nên được duyệt ở vòng lặp ngoài (*outer loop*) nếu phép kết này được xử lý bằng phương pháp hai vòng lặp lồng (*nested-loop join*)?

- a. Employee do tổng số block accesses trên các tập tin dữ liệu là 58 (nhiều nhất).
- b. Department do tổng số block accesses trên các tập tin dữ liệu là 112 (nhiều nhất).
- c. Department do tổng số block accesses trên các tập tin dữ liệu là 55 (ít nhất).
- d. Employee do tổng số block accesses trên các tập tin dữ liệu là 32 (ít nhất).
- e. Ý kiến khác.

Câu 17. Cho câu truy vấn SQL dùng để lấy ra các cặp mã nhân viên của những nhân viên ở cùng phòng và có cùng lương từ tập tin dữ liệu Nhân_Viên ở **Bảng 1** như sau:

SELECT n1.ID, n2.ID

FROM Nhân_Viên n1, Nhân_Viên n2

WHERE n1.Phòng = n2.Phòng **AND** n1.Lương = n2.Lương;

Xác định **biểu thức đại số quan hệ** (*relational algebra expression*) tương ứng của câu truy vấn này.

- a. $\pi_{n1.ID, n2.ID}(\sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}((n1 \leftarrow \text{Nhân_Viên})) \times \sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}((n2 \leftarrow \text{Nhân_Viên})))$
- b. $\pi_{n1.ID, n2.ID}((n1 \leftarrow \text{Nhân_Viên}) \times \sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương} \pi_{n1.ID, n2.ID} (n2 \leftarrow \text{Nhân_Viên}))$
- c. $\pi_{n1.ID, n2.ID}(\sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}((n1 \leftarrow \text{Nhân_Viên}) \times (n2 \leftarrow \text{Nhân_Viên})))$

- d. Không thể xác định được biểu thức đại số quan hệ do còn thiếu mô tả về các vùng tin khác và các dạng chỉ mục trên tập tin Nhân_Viên này.
- e. Ý kiến khác.

Câu 18. Cho câu truy vấn SQL dùng để lấy ra các bản ghi của những nhân viên có lương lớn hơn 2000 từ tập tin dữ liệu Nhân_Viên ở **Bảng 1** như sau:

```
SELECT *  
FROM Nhân_Viên  
WHERE Lương > 2000;
```

Giả sử hiện tại tập tin dữ liệu Nhân_Viên là tập tin có thứ tự (*ordered file*) với các bản ghi được sắp thứ tự theo các giá trị của vùng tin khóa ID. Một chỉ mục B được định nghĩa trên vùng tin khóa ID.

Xác định **phương thức tìm kiếm** (*search method*) có thể được dùng để trả lời câu truy vấn SQL này.

- a. Tìm kiếm tuần tự (*linear search*) *
- b. Tìm kiếm nhị phân (*binary search*)
- c. Tìm kiếm dùng chỉ mục sơ cấp (*primary index*)
- d. Tìm kiếm dùng chỉ mục thứ cấp (*secondary index*)
- e. Ý kiến khác.

Câu 19. Cho 2 quan hệ Nhân_Viên và Phòng_Ban có lược đồ như sau.

Nhân_Viên (ID, Tên, Lương, Phòng, Địa_Chỉ);

Phòng_Ban (ID_Phòng, Tên_Phòng, ID_Quản_Lý)

Trong đó, ID là khóa chính của quan hệ Nhân_Viên và Phòng là khóa ngoại của quan hệ Nhân_Viên tham chiếu đến ID_Phòng của quan hệ Phòng_Ban; ID_Phòng là khóa chính của quan hệ Phòng_Ban và Tên_Phòng là khóa dự tuyển của quan hệ Phòng_Ban.

Câu lệnh SQL sau dùng để truy vấn tên của tất cả nhân viên làm việc ở phòng 'RESEARCH' có lương lớn hơn 2000:

```
SELECT Tên  
FROM Nhân_Viên, Phòng_Ban  
WHERE Phòng = ID_Phòng  
AND Tên_Phòng = 'RESEARCH'  
AND Lương > 2000;
```

Hãy cho biết **cây truy vấn** (*query tree*) nào sau đây **KHÔNG** là dạng biểu diễn của câu lệnh SQL trên?

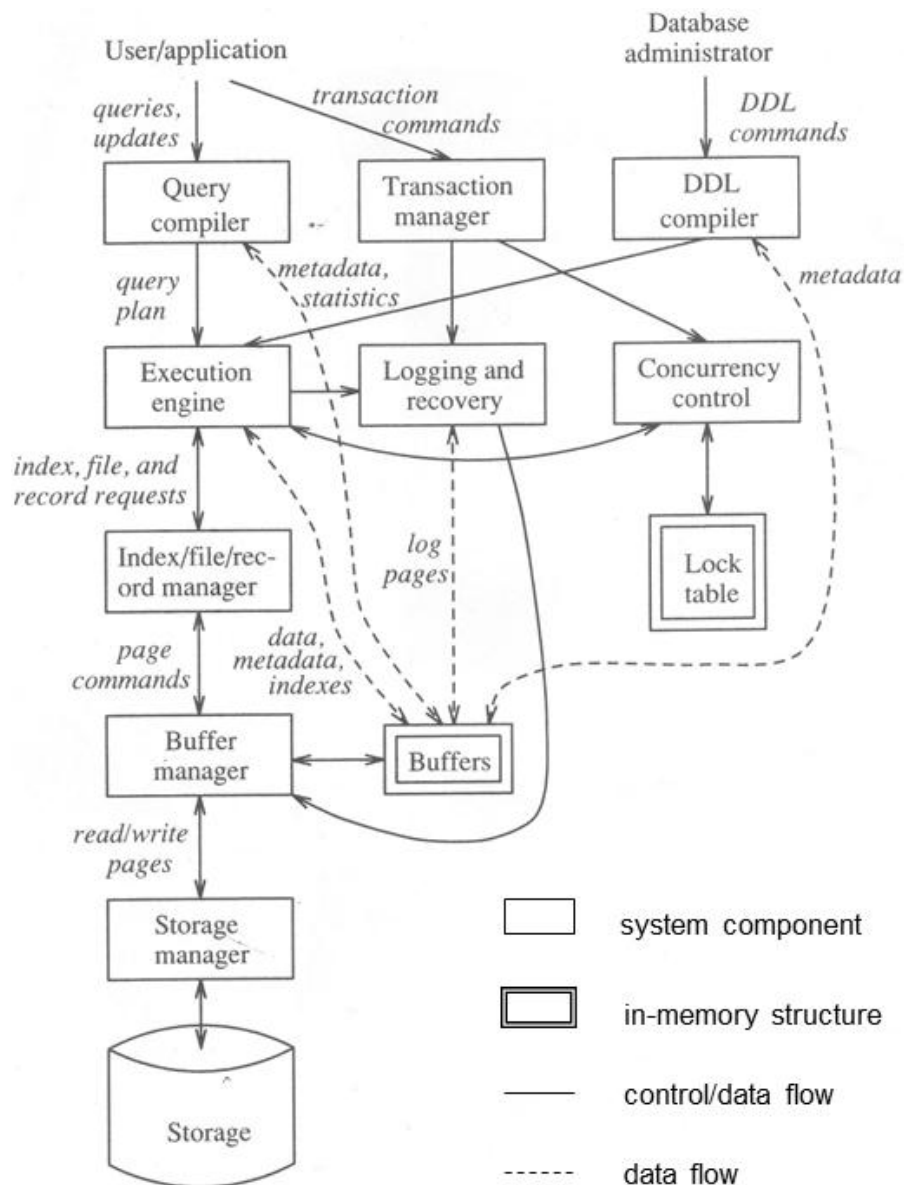
- a. **Hình 9**
- b. **Hình 10** *
- c. **Hình 11**
- d. **Hình 12**
- e. Ý kiến khác.

Câu 20. Cho đoạn mã chương trình trong **Hình 13**. Xác định **tác vụ** được hiện thực trên hai tập tin (quan hệ) R và S bởi đoạn mã chương trình này.

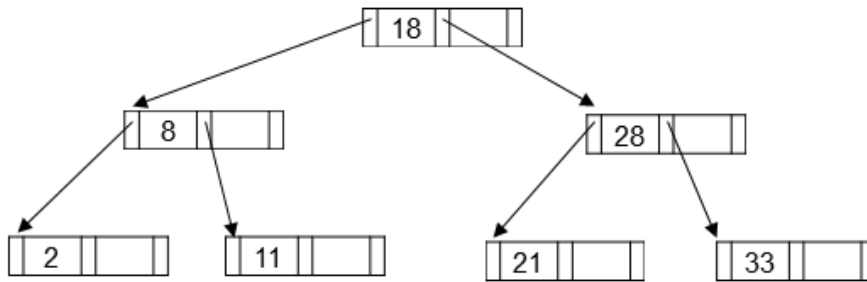
- a. Kết bằng (*equi join*) *
- b. Hội (*union*)
- c. Giao (*intersection*)
- d. Hiệu (*difference*)
- e. Ý kiến khác.

Bảng 1 - Tập tin dữ liệu Nhân_Viên tương ứng với bảng Nhân_Viên trong cơ sở dữ liệu

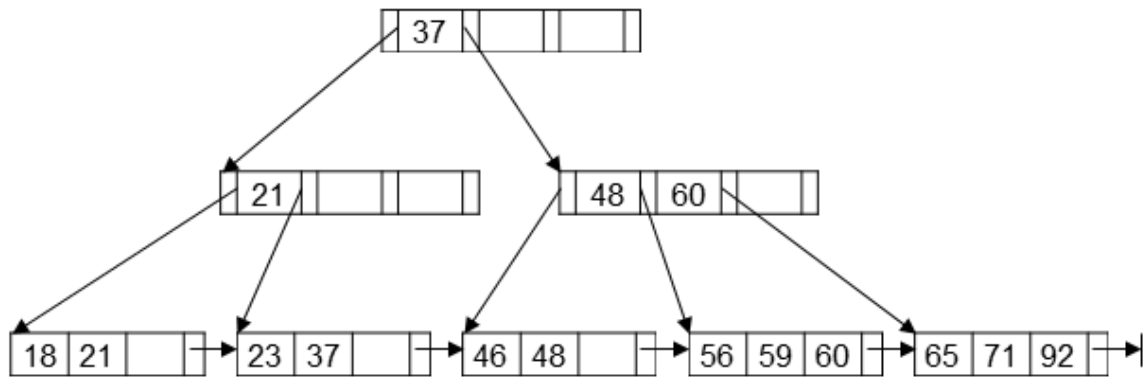
ID	Tên	Lương	Phòng	...
2	An	2000
5	Đông	1000
8	Hạ	2500
10	Phúc	1500
11	Hoa	1800
15	Tuân	2000
18	Cường	1900
19	Minh	2100
21	Mai	2500
23	Đào	2300
28	Xuân	1600
30	Dũng	2000
33	Thu	2300



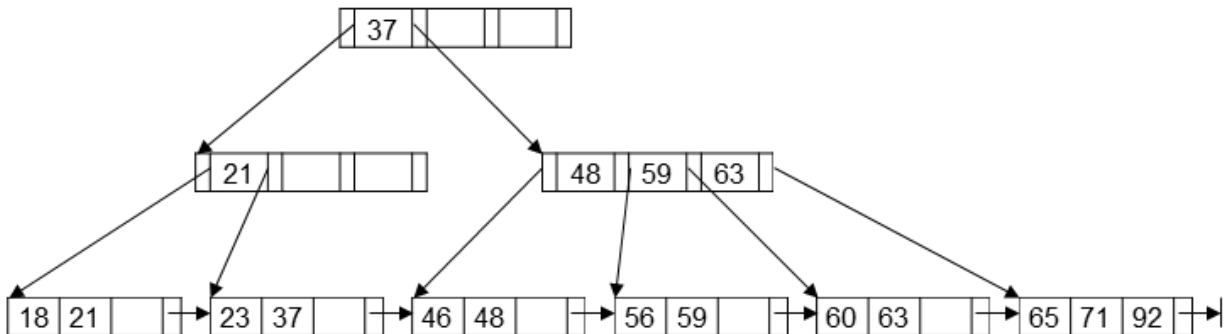
Hình 1 – Kiến trúc hệ quản trị cơ sở dữ liệu (architecture of a database management system) (Câu 2)



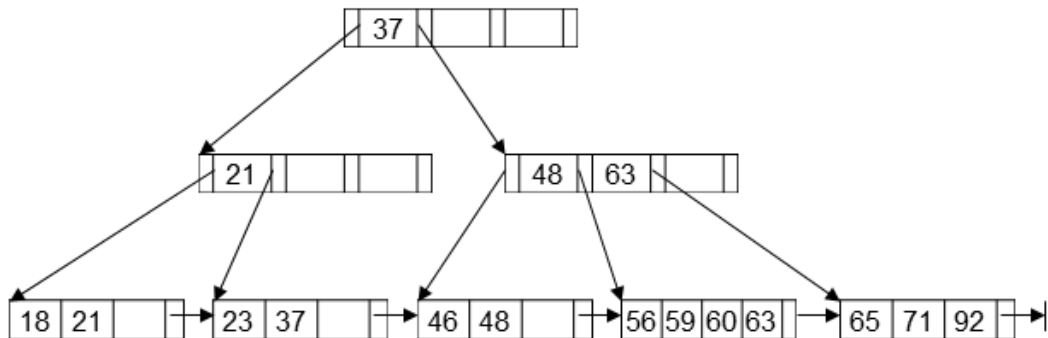
Hình 2 – Cây chỉ mục B trên vùng tin ID (khóa (key), sắp thứ tự (ordering field)) của tập tin dữ liệu Nhân_Viên trong Bảng 1 trước khi xóa 8 (Câu 7)



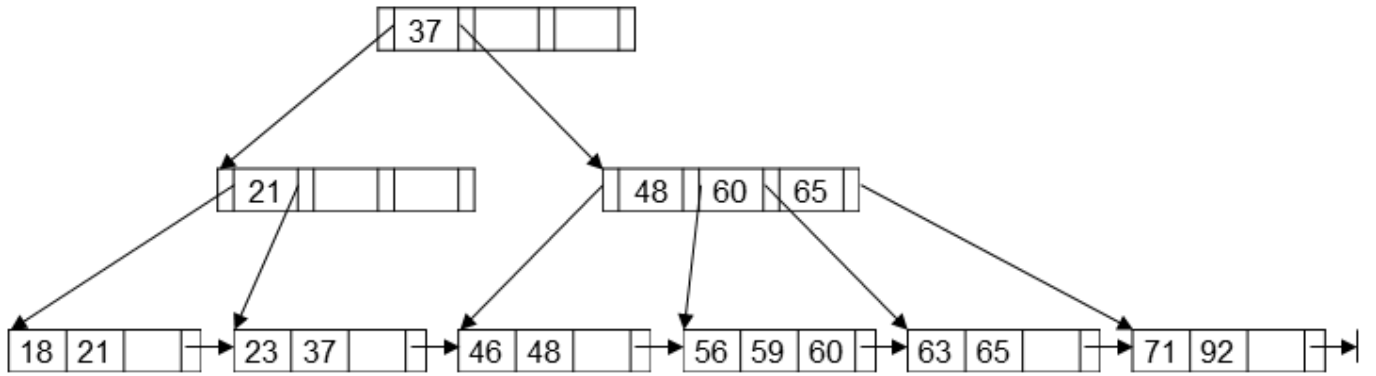
Hình 3 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{leaf} = 3$ trước khi thêm giá trị tìm kiếm 63 (Câu 8, 9, 10)



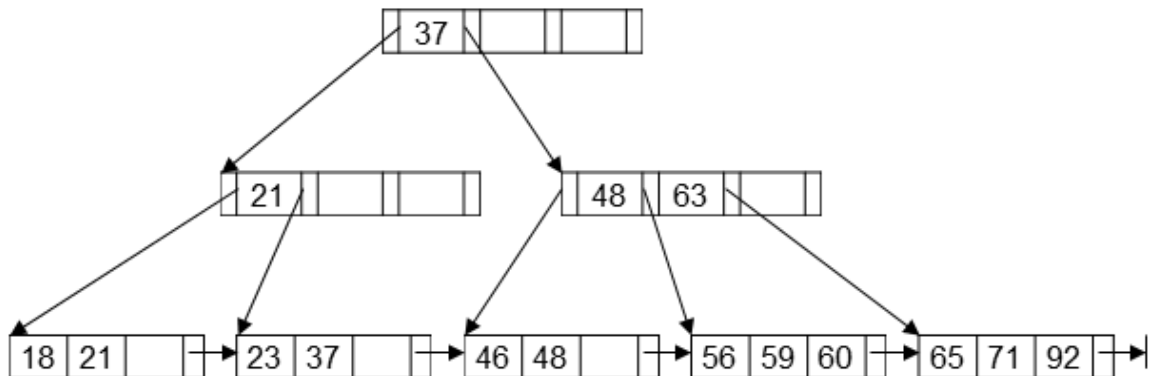
Hình 4 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{leaf} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. a)



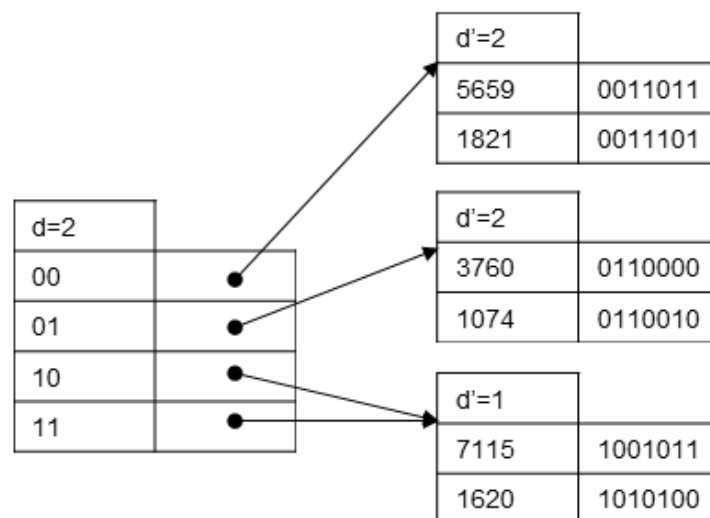
Hình 5 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{leaf} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. b)



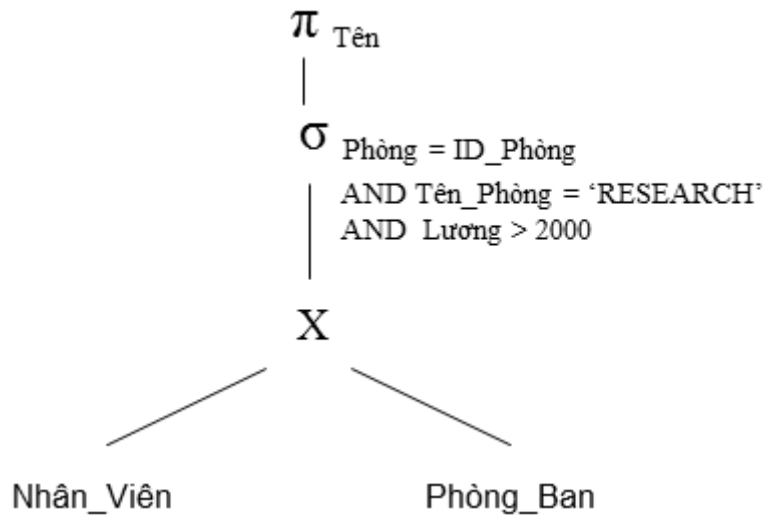
Hình 6 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. c)



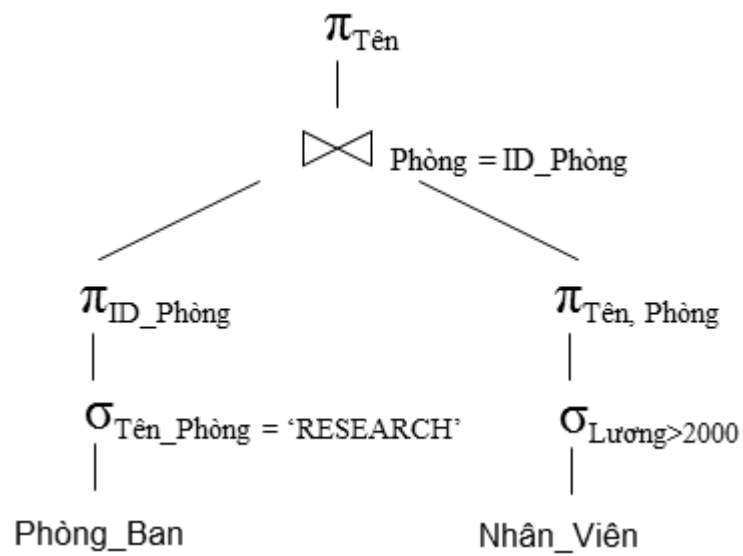
Hình 7 - Cây chỉ mục B+ có bậc $p = 4$ và $p_{\text{leaf}} = 3$ ở Hình 3 sau khi thêm 63 (Câu 8. d)



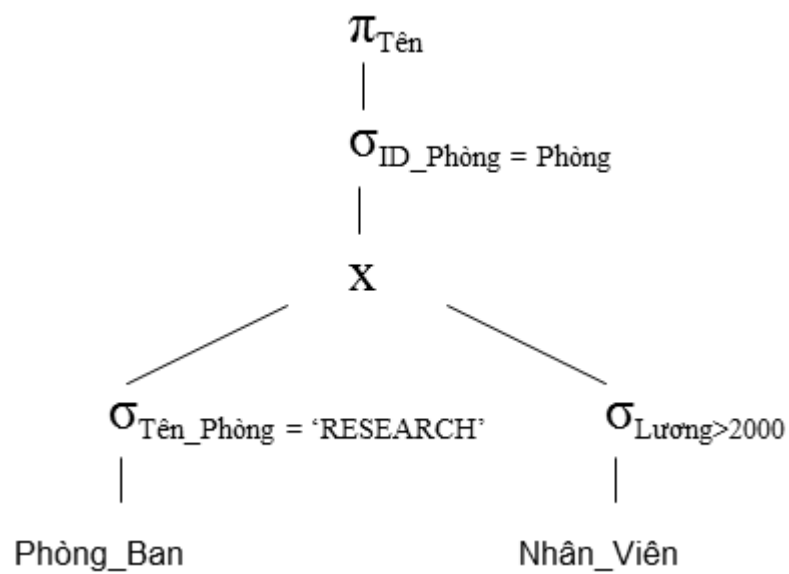
Hình 8 – Tập tin dữ liệu được tổ chức dạng băm động dùng extendible hashing với hàm băm $h(K) = K \bmod 128$ (Câu 14, 15)



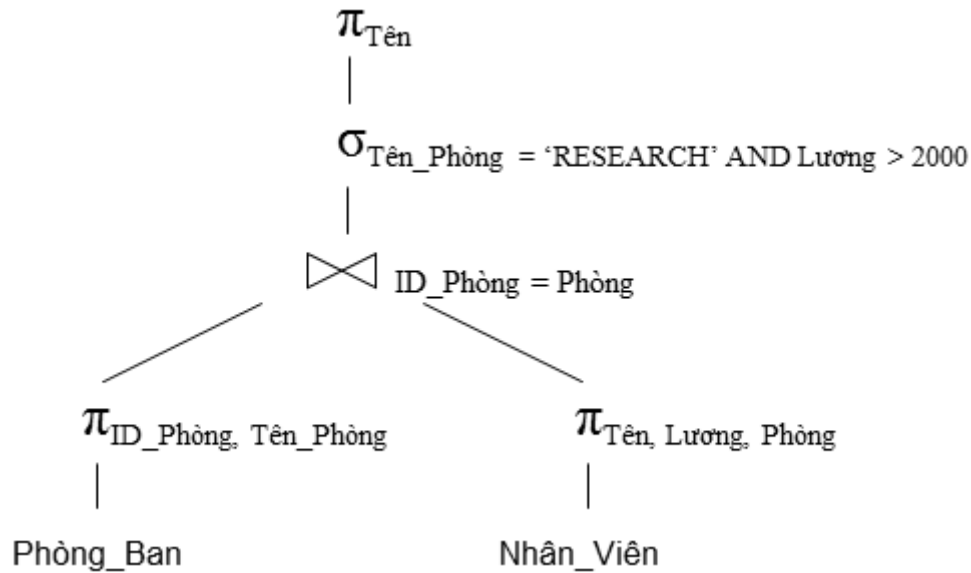
Hình 9 – Cây truy vấn kết quả (Câu 19. a)



Hình 10 – Cây truy vấn kết quả (Câu 19. b)



Hình 11 – Cây truy vấn kết quả (Câu 19. c)



Hình 12 – Cây truy vấn kết quả (Câu 19. d)

```

sort the tuples in R on attribute A; /* assume R has n tuples */
sort the tuples in S on attribute B; /* assume S has m tuples */
set i ← 1, j ← 1;
while (i ≤ n) and (j ≤ m)
do { if R(i)[A] > S(j)[B]
    then set j ← j + 1
    elseif R(i)[A] < S(j)[B]
    then set i ← i + 1
    else { /* output a matched tuple */
        output the combined tupe <R(i), S(j)> to T;
        /* output other tuples that match R(i), if any */
        set l ← j + 1 ;
        while ( l ≤ m) and (R(i)[A] = S(l)[B])
        do { output the combined tuple <R(i), S(l)> to T;
            set l ← l + 1
        }
        /* output other tuples that match S(j), if any */
        set k ← i+1
        while ( k ≤ n) and (R(k)[A] = S(j)[B])
        do { output the combined tuple <R(k), S(j)> to T;
            set k ← k + 1
        }
        set i ← i+1, j ← j+1;
    }
}
  
```

Hình 13 - Đoạn mã chương trình hiện thực tác vụ trên 2 tập tin (quan hệ) R và S (Câu 20)

Họ - Tên:

Mã Số Sinh Viên:

Phần trả lời

Câu hỏi	a	b	c	d	e (Ý kiến khác)
1				X	
2		X			
3				X	
4	X				
5		X			
6				X	
7		X			
8			X		
9			X		
10				X	
11	X				
12		X			
13	X				
14				X	
15		X			
16			X		
17			X		
18	X				
19					X: không có cây truy vấn nào không là cây truy vấn biểu diễn của câu lệnh đã cho.
20	X				

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu (503004)**

Học kỳ 1 - 2012-2013

Ngày thi: 17/10/2012

Phòng thi: 503 C4

Mã đề: **11**

Họ - Tên:

Mã Số Sinh Viên:

Phần giải thích:

Câu 4. Số truy đạt khối ít nhất để thực hiện việc tìm kiếm bản ghi của nhân viên có mã ID = 29 trên tập tin dữ liệu Nhân_Viên là **2**.

Giải thích: Vùng tin ID là vùng tin khóa, sắp thứ tự các bản ghi của tập tin dữ liệu Nhân_Viên. Do đó, việc tìm kiếm bản ghi của nhân viên có mã ID = 29 chính là tìm kiếm với điều kiện “=” trên vùng tin ID và được thực hiện bằng phương pháp tìm kiếm nhị phân (*binary search*).

Truy đạt khối đầu tiên được thực hiện với khối dữ liệu (18, 19) ở chính giữa. Do $19 < 29$ nên việc tìm kiếm được thực hiện cho phần nửa lớn hơn.

Truy đạt khối thứ hai được thực hiện với khối dữ liệu (28, 30) ở chính giữa của phần nửa lớn hơn. Do $28 < 29 < 30$ nên kết quả tìm kiếm trả về là không tìm thấy bản ghi của nhân viên có mã ID = 29.

Tổng kết lại số truy đạt khối dữ liệu đã được thực hiện là 2.

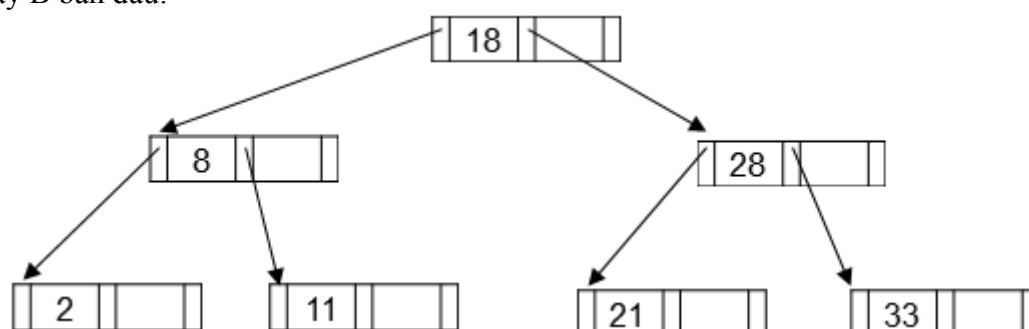
Câu 5. Số truy đạt khối ít nhất để thực hiện việc tìm kiếm bản ghi của nhân viên có Tên = Việt trên tập tin dữ liệu Nhân_Viên là **7**.

Giải thích: Vùng tin Tên là vùng tin không khóa, không sắp thứ tự các bản ghi của tập tin dữ liệu Nhân_Viên. Do đó, việc tìm kiếm bản ghi của nhân viên có Tên = Việt chính là tìm kiếm với điều kiện “=” trên vùng tin không khóa, không sắp thứ tự và được thực hiện tuần tự (*linear search*) từng đầu đến cuối tập tin và dừng lại khi tìm thấy bản ghi nếu tồn tại.

Do vùng tin Tên không có giá trị “Việt” nên số khối truy đạt là 7 khi việc tìm kiếm được thực hiện lần lượt từ khối dữ liệu đầu tiên đến khối dữ liệu cuối cùng.

Câu 7. Cây chỉ mục **B** ở **Hình 2** sau khi xóa giá trị tìm kiếm **8** sẽ có chiều cao giảm 1, số node lá giảm 1, và số node nội giảm 2.

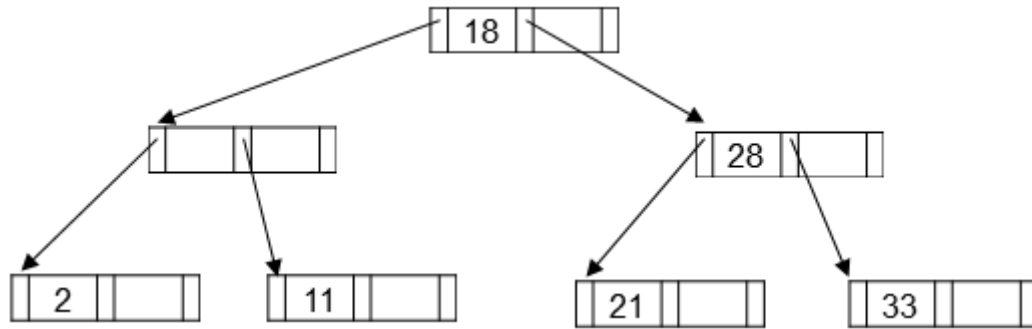
Cây B ban đầu:



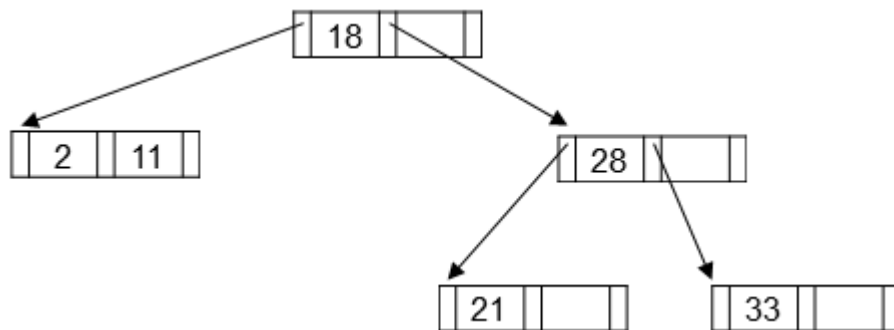
Định vị và xóa giá trị tìm kiếm 8 làm node nội này bị tràn dưới (*underflow*):

Họ - Tên:

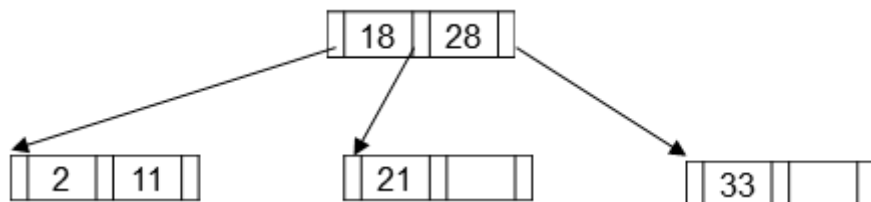
Mã Số Sinh Viên:



Trộn các node con tương ứng thành 1 node mới nên làm giảm số node nội ban đầu và cũng làm giảm số node lá sau khi trộn:



Thực hiện cân bằng cây để tất cả các node lá đều ở cùng mức nên làm giảm chiều cao của cây và đồng thời giảm số node nội.



Kết luận: Cây kết quả đạt được có chiều cao giảm 1, số node lá giảm 1, và số node nội giảm 2 so với cây B ban đầu.

Câu 8. Cây chỉ mục B+ kết quả đạt được sau khi thêm giá trị tìm kiếm 63 vào cây chỉ mục B+ này ở Hình 7.

Giải thích:

- Định vị vị trí của 63 trong cây B+
 - Khối đầu tiên (root) của B+-tree được truy đạt và kiểm tra với 63. Do $37 < 63$ nên đi về cây con bên phải của 37.
 - Khối đầu tiên (root) của cây con bên phải của 37 được truy đạt và kiểm tra với 63. Do $60 < 63$ nên đi về cây con bên phải của 60.
 - Khối đầu tiên (root) của cây con bên phải của 60 được truy đạt và kiểm tra với 63. Khối này là node lá nên đây chính là khối mà 63 sẽ được chèn vào.
- Chèn 63 vào node lá

Họ - Tên:

Mã Số Sinh Viên:

- Node lá (65, 71, 92) đã đầy nên việc chèn 63 vào gây nên việc tràn trên (overflow). Do đó, tách node lá này thành 2 node: (63, 65) và (71, 92). Hai con trỏ cây của trị tìm kiếm 65 sẽ giữ 2 node lá mới.
- Trị tìm kiếm 65 được chèn vào node nội mà trước đó chứa con trỏ cây đến node lá mới vừa được tách. Do đó, 65 được chèn vào node (48, 60) thành node (48, 60, 65).

Tổng kết, việc chèn 63 vào B+-tree này làm cho số node lá tăng lên 1; nhưng số node nội và chiều cao của cây không bị ảnh hưởng.

Câu 9. Số khối dữ liệu cần thiết để lưu trữ cây chỉ mục B+ này chính là số node của cây = 8. Lý do: mỗi node của cây chỉ mục được chứa trọn trong một khối dữ liệu.

Câu 10. Xác định số truy đạt khối cần thực hiện để đọc được bản ghi có giá trị vùng tin lập chỉ mục là 21.

- 1 block access vào nút gốc (37) của cây để đi về nhánh trái
- 1 block access vào nút nội (21) để đi về nhánh trái
- 1 block access vào nút lá (18, 21) để lấy con trỏ dữ liệu đến bản ghi có giá trị vùng tin lập chỉ mục là 21
- 1 block access vào khối chứa bản ghi 21

Kết luận: 4 block accesses cần thực hiện để đọc được bản ghi có giá trị vùng tin lập chỉ mục là 21 = số mức của cây B+-tree (kể cả mức của nút gốc) + 1.

Câu 12. Nếu các khối dữ liệu của tập tin dữ liệu này được tổ chức liên tục (*contiguous*) ở trên cùng một rãnh (*track*) thì tổng thời gian thực hiện chuyển các khối dữ liệu vào vùng nhớ chính khi đọc hết dữ liệu theo thứ tự của các giá trị ở vùng tin ID là ***bao nhiêu msec?***

Tổng số khối dữ liệu cần đọc = $b = 7$ blocks.

Do các khối được tổ chức liên tục (*contiguous*) trên cùng 1 rãnh (*track*) nên tổng thời gian thực hiện chuyển các khối vào vùng nhớ chính là $s + rd + b \cdot btt$; trong đó, s là seek time, rd là rotational delay, và btt là block transfer time.

Seek time = $s = 20$ msec

Rotational delay = $rd = (1/(2 \cdot p)) \cdot 60 \cdot 1000$ msec = $(1/(2 \cdot 5600)) \cdot 60 \cdot 1000 = 5.36$ msec

Transfer rate = $tr = \text{track size/time per rotation} = T/((1/p) \cdot 60 \cdot 1000) = 102400 \cdot 5600 / (60 \cdot 1000) = 9557.33$ bytes/msec

Block transfer time = $btt = \text{block size/transfer rate} = B/tr = 512/9557.33 = 0.05$ msec

Tổng thời gian = $s + rd + b \cdot btt = 20 + 5.36 + 7 \cdot 0.05 = \mathbf{25.71}$ msec

Câu 13. Hệ số phân khối của tập tin dữ liệu Student này theo cách phân khối không phủ (*unspanned blocking*) = $\lfloor B/R \rfloor$; trong đó, B là block size và R là record size.

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu** (503004)

Học kỳ 1 - 2012-2013

Ngày thi: 17/10/2012

Phòng thi: 503 C4

Mã đề: **11**

Họ - Tên:

Mã Số Sinh Viên:

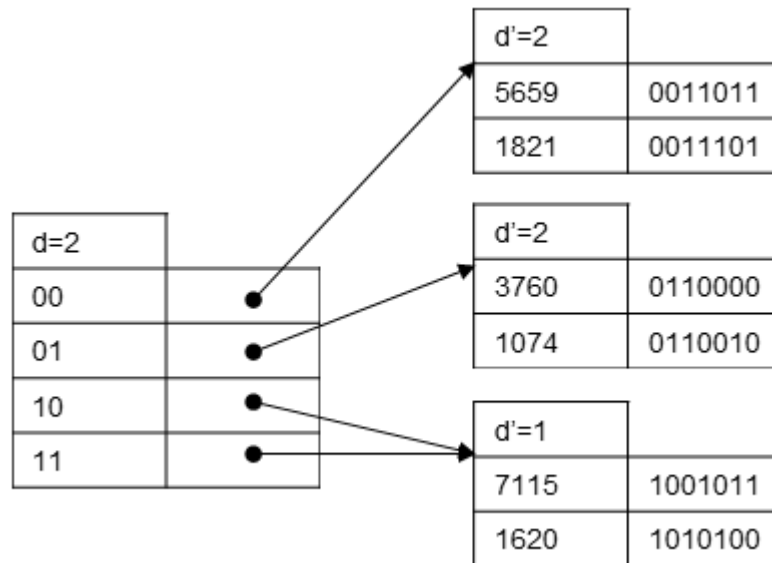
Block size = B = 512 bytes

Record size = R = 30 + 9 + 40 + 9 + 8 + 1 + 4 + 4 + 4 + 3 + 1 = 113 bytes

Blocking factor = bfr = $\lfloor B/R \rfloor = \lfloor 512/113 \rfloor = 4 \text{ records/block}$.

Câu 14. Độ sâu toàn cục (global depth, d) của toàn tập tin và **độ sâu cục bộ (local depth, d')** của thùng chứa bản ghi **3943** sau khi thêm bản ghi này vào tập tin

Tập tin được tổ chức dạng extendible hashing trước khi thêm bản ghi 3943:



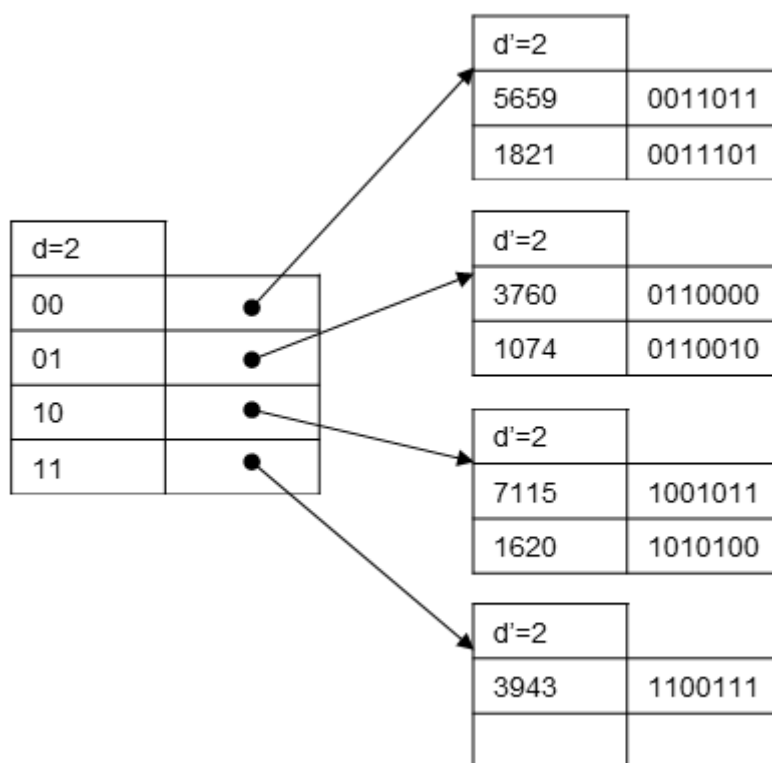
Định vị thùng sẽ chứa bản ghi 3943: $h(3943) = 3943 \bmod 128 = 103_{10} = 1100111_2 \rightarrow$ thùng sẽ chứa bản ghi 3943 là thùng có bit dẫn đầu là “1” tương ứng $d' = 1$. Việc thêm bản ghi vào thùng này làm cho thùng bị tràn (overflow) nên việc tách thùng được tiến hành.

Do $d' = 1 < d = 2$ nên chỉ mỗi thùng này được tách đôi và thư mục được giữ nguyên; nghĩa là d' trở thành d'_{10} và d'_{11} có giá trị là 2 và $d = 2$ được giữ nguyên. Các bản ghi hiện tại 7115 và 1620 được phân bổ lại trong hai thùng “10” và “11” dùng 2 bit đầu của kết quả băm lấy trị nhị phân. Sau đó, phân bổ bản ghi mới 3943 vào thùng “11” do 2 bit đầu tương ứng là “11”.

Kết quả sau khi thêm bản ghi 3943 là:

Họ - Tên:

Mã Số Sinh Viên:



Câu 15. Số truy đạt khối (block access) cần thực hiện để đạt được bản ghi 3760 là tổng số khối truy đạt vào thư mục và sau đó truy đạt vào thùng/khối chứa bản ghi 3760 này.

Do thư mục (directory) được chứa trọn trong 1 khối nên cần 1 truy đạt khối đến thư mục này.

Do độ sâu toàn cục $d = 2$ và $h(3760) = 3760 \bmod 128 = 48_{10} = 0110000_2$ nên sử dụng 2 bit đầu **01** để xác định được thùng chứa bản ghi 3760. Do đó, cần 1 truy đạt khối đến thùng chứa bản ghi 3760 cũng chính là đến khối (block) chứa bản ghi 3760 (vì 1 thùng chứa 1 khối dữ liệu, 1 khối dữ liệu chứa 2 bản ghi).

Tổng kết: số truy đạt khối cần thực hiện là **2 block accesses**.

Ghi chú: nếu trước đó, thư mục đã được đưa vào vùng nhớ chính thì khi này, 1 truy đạt khối đến thư mục không cần được thực hiện nữa và do đó, chỉ tốn 1 truy đạt khối từ thư mục đến thùng chứa bản ghi 3760.

Câu 16. Department nên được duyệt ở vòng lặp ngoài (outer loop) do tổng số block accesses trên các tập tin dữ liệu nhập khi đó là 53 (ít nhất).

Giải thích:

. Department ở vòng lặp ngoài (outer loop)

$$b_D + \lceil b_D / (b_f - 2) \rceil * b_E = 7 + 3 * 16 = 55 \text{ block accesses}$$

. Employee ở vòng lặp ngoài (outer loop)

$$b_E + \lceil b_E / (b_f - 2) \rceil * b_D = 16 + 6 * 7 = 58 \text{ block accesses}$$

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu** (503004)

Học kỳ 1 - 2012-2013

Ngày thi: 17/10/2012

Phòng thi: 503 C4

Mã đề: **11**

Họ - Tên:

Mã Số Sinh Viên:

Do 55 block accesses < 58 block accesses nên Department nên được duyệt ở vòng lặp ngoài.

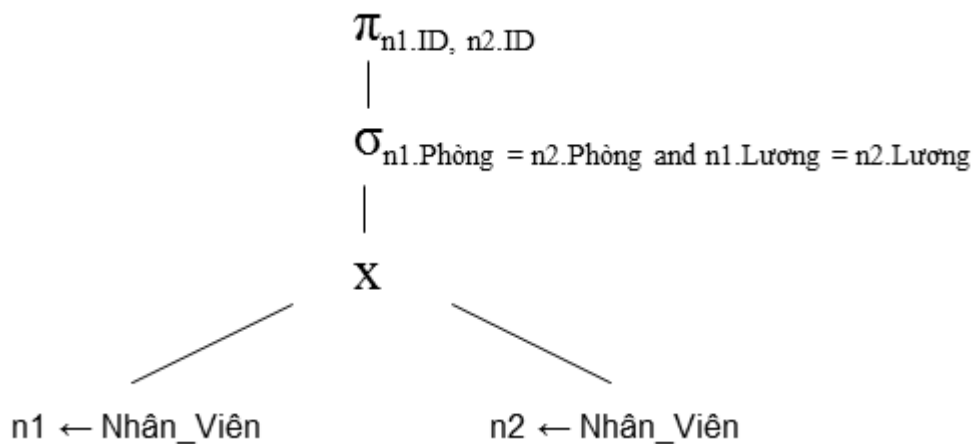
Câu 17. Xác định *biểu thức đại số quan hệ* (*relational algebra expression*) tương ứng của câu truy vấn:

SELECT n1.ID, n2.ID

FROM Nhân_Viên n1, Nhân_Viên n2

WHERE n1.Phòng = n2.Phòng **AND** n1.Lương = n2.Lương;

- Không là biểu thức đúng do phép chọn $\sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}$ trên $(n1 \leftarrow \text{Nhân_Viên})$ và phép chọn $\sigma_{n1.Phòng = n2.Phòng \text{ AND } n1.Lương = n2.Lương}$ trên $(n2 \leftarrow \text{Nhân_Viên})$ không đúng ở phần điều kiện chọn.
- Không là biểu thức đúng do phép tích đề-cạc (Cartesian product) đúng sẽ không có phần điều kiện.
- Biểu thức đúng và có cây truy vấn tương ứng như sau:**



- Không đúng do biểu thức đại số quan hệ tương ứng của câu trúc vẫn không phụ thuộc vào đặc điểm và các chỉ mục trên các vùng tin.

Câu 19. Do có sự nhầm lẫn trong việc copy-paste hình ảnh nên cây truy vấn KHÔNG là dạng biểu diễn của câu lệnh SQL đã cho không có trong các hình vẽ: Hình 9-12.

Đáp án đúng: e.

Họ - Tên:

Mã Số Sinh Viên:

Kiểm tra giữa kỳ #1

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu (503004)** – Ngành: Khoa Học Máy Tính

Thời gian làm bài: **30 phút**

(Bài kiểm tra gồm 8 câu hỏi. Sinh viên được tham khảo tài liệu.)

Cho tập tin dữ liệu Nhân_Viên gồm 7 khối dữ liệu (7 *blocks*) trong Bảng 1. Mỗi khối chứa 2 bản ghi (*record*). Vùng tin ID là vùng tin khóa (*key field*) của tập tin; các vùng tin còn lại là vùng tin phi khóa (*non-key field*). Ngoài ra, các bản ghi trong tập tin này được sắp thứ tự vật lý theo các giá trị tăng dần của vùng tin ID.

1. Nếu thực hiện tìm kiếm bản ghi về nhân viên có ID = 21 thì số truy đạt khối trung bình (*the average number of block accesses*) cần được thực hiện trên tập tin này là bao nhiêu? (0.5 điểm)
2. Nếu thực hiện tìm kiếm bản ghi về nhân viên có lương = 2000 thì số truy đạt khối trung bình cần được thực hiện trên tập tin này là bao nhiêu? (0.5 điểm)
3. Nếu một chỉ mục (*index*) được định nghĩa trên vùng tin ID thì chỉ mục này là chỉ mục gì: sơ cấp (*primary*), cụm (*clustering*), hay thứ cấp (*secondary*)? (0.5 điểm)
4. Nếu một chỉ mục được định nghĩa trên vùng tin Lương thì chỉ mục này là chỉ mục gì: sơ cấp, cụm, hay thứ cấp? (0.5 điểm)
5. Xây dựng cây chỉ mục B (*B-tree*) có bậc $p = 3$ trên vùng tin ID (2.5 điểm). Giả sử thứ tự đưa các giá trị tìm kiếm (*search values*) vào chỉ mục B này được thực hiện tuần tự theo đúng đặc điểm chỉ mục trên vùng tin ID và theo đúng thứ tự xuất hiện từ trên xuống dưới của các giá trị ở vùng tin ID. Xác định số khối (*number of blocks*) cần thiết để lưu trữ cây chỉ mục B này. (0.5 điểm)
6. Tập tin dữ liệu này được lưu trữ vật lý trên bộ đĩa từ có số vòng quay trong 1 phút là $p = 5600$ rpm, có dung lượng là $4GB = 4 \cdot 10^6 \cdot 1024$ bytes, gồm một chồng đĩa có 100 đĩa 1 mặt (*single-sided*), mỗi mặt đĩa gồm 400 rãnh (*track*). Kích thước một khối dữ liệu (*block size*) là $B = 512$ bytes. Thời gian tìm kiếm trung bình (*average seek time*) là $s = 20$ msec. Tính tổng thời gian thực hiện chuyển các khối dữ liệu vào vùng nhớ chính để tìm kiếm bản ghi ở câu (1) và câu (2). (2.5 điểm)

Họ - Tên:

Mã Số Sinh Viên:

7. Nếu các khối dữ liệu của tập tin dữ liệu này ở trên cùng một rãnh thì tổng thời gian thực hiện chuyển các khối dữ liệu vào vùng nhớ chính khi đọc hết dữ liệu theo thứ tự của các giá trị ở vùng tin ID là bao nhiêu msec? (1 điểm)
8. Nếu các bản ghi của tập tin dữ liệu này không được sắp thứ tự theo vùng tin ID mà được tổ chức vào 4 thùng (4 *buckets*), mỗi thùng gồm 2 khối dữ liệu, và dùng hàm băm (*hashing function*) trên vùng tin ID là $f(v_{ID}) = v_{ID} \bmod 4$. Vấn đề đụng độ (*collision*) ở các thùng được xử lý bằng xâu riêng (*chaining*). Khi tìm kiếm bản ghi về nhân viên có ID = 21, tính tổng thời gian thực hiện chuyển các khối dữ liệu tương ứng vào vùng nhớ chính. (1.5 điểm)

Bảng 1 – Tập tin dữ liệu Nhân_Viên

ID	Tên	Lương	...
2	An	2000	...
5	Đồng	1000	...
8	Hạ	2500	...
10	Phúc	1500	...
11	Hoa	1800	...
15	Tuấn	2000	...
18	Cường	1900	...
19	Minh	2100	...
21	Mai	2500	...
23	Đào	2300	...
28	Xuân	1600	...
30	Dũng	2000	...
33	Thu	2300	...

Phần trả lời

Câu 1: $\lceil \log_2 b \rceil = \lceil \log_2 7 \rceil = 3$ block accesses

Câu 2: $\lceil b/2 \rceil = \lceil 7/2 \rceil = 4$ block accesses

Câu 3: Chỉ mục trên vùng tin ID (khóa, sắp thứ tự) là chỉ mục sơ cấp.

Câu 4: Chỉ mục trên vùng tin Lương (không khóa, không sắp thứ tự) là chỉ mục thứ cấp.

Họ - Tên:

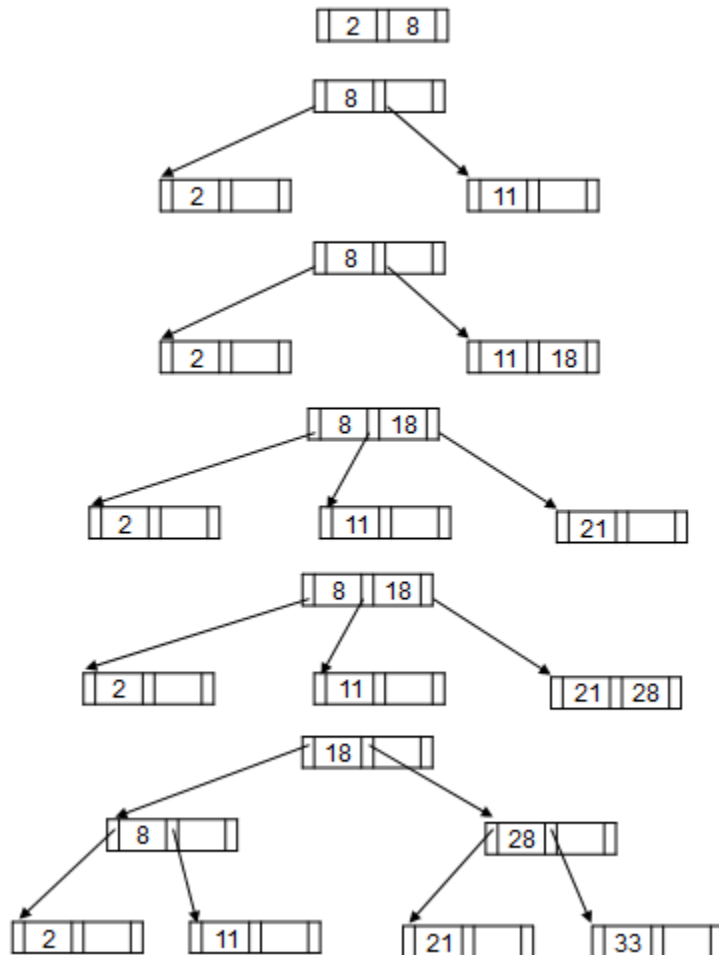
Mã Số Sinh Viên:

Câu 5: Đưa các giá trị tìm kiếm theo thứ tự sau vào cây B có bậc $p = 3$, chỉ mục sơ cấp trên vùng tin ID: 2, 8, 11, 18, 21, 28, 33. Đây là dạng chỉ mục thừa, dùng block anchors.

Cây B có bậc $p = 3 \rightarrow$ số tree pointers tối đa ở mỗi node là 3 và số giá trị tìm kiếm tối đa là 2, số tree pointers tối thiểu ở mỗi node là $\lceil p/2 \rceil = 2$ và số giá trị tìm kiếm tối thiểu là 1.

Kết quả đạt được ở **Hình 1**.

Số khối cần thiết để lưu trữ cây chỉ mục B này chính là số nodes của cây $B = 7$ blocks.



Hình 1 – Cây chỉ mục B kết quả đạt được

Câu 6:

Seek time, $s = 20$ msec

Rotational delay, $rd = (1/(2 \cdot p)) \cdot 60 \cdot 1000 = (1/(2 \cdot 5600)) \cdot 60 \cdot 1000 = 5.36$ msec

Họ - Tên:

Mã Số Sinh Viên:

$$\text{Số tracks} = \text{số đĩa} * \text{số track/đĩa} = 100 * 400 = 40000 \text{ tracks}$$

$$\text{Track size} = \text{tổng dung lượng/số tracks} = 4\text{GB}/40000 \text{ tracks} = 4*10^6*1024/40000 = 102400 \text{ bytes}$$

$$\text{Transfer rate} = \text{track size/time per rotation} = \text{track size} / ((1/p)*60*1000) = 102400*5600/(60*1000) = 9557.33 \text{ bytes/msec}$$

$$\text{Block transfer time} = \text{btt} = \text{block size/transfer rate} = 512/9557.33 = 0.05 \text{ msec}$$

$$\text{Tổng thời gian chuyển các khối ở câu (1)} = 3 * (s + \text{rd} + \text{btt}) = 3 * (20 + 5.36 + 0.05) = 76.23 \text{ msec}$$

$$\text{Tổng thời gian chuyển các khối ở câu (2)} = 4 * (s + \text{rd} + \text{btt}) = 4 * (20 + 5.36 + 0.05) = 101.64 \text{ msec}$$

$$\text{Câu 7: Tổng thời gian} = s + b*(\text{rd} + \text{btt}) = 20 + 7*(5.36 + 0.05) = 57.87 \text{ msec}$$

Câu 8:

Bucket 0 chứa các records có ID = 8, 28

Bucket 1 chứa các records có ID = 5, 21, 33

Bucket 2 chứa các records có ID = 2, 10, 18, 30

Bucket 3 chứa các records có ID = 11, 15, 19, 23

$$\text{Số block accesses} = 1$$

$$\text{Tổng thời gian} = 1*(s + \text{rd} + \text{btt}) = 20 + 5.36 + 0.05 = 25.41 \text{ msec.}$$

Họ - Tên:

Mã Số Sinh Viên:

Cách tính điểm

Câu	Nội dung	Điểm tối đa
Câu 1	Binary search trên ordering field	0.5
Câu 2	Linear search trên non-ordering field	0.5
Câu 3	Primary index	0.5
Câu 4	Secondary index	0.5
Câu 5.1	Search values	0.5
Câu 5.2	Insert 2, 8, 11, 18	0.5
Câu 5.3	Insert 21	0.5
Câu 5.4	Insert 28	0.25
Câu 5.5	Insert 33	0.75
Câu 5.6	Số blocks cho B-tree	0.5
Câu 6.1	Rotational delay (rd), số tracks	0.5
Câu 6.2	Track size	0.5
Câu 6.3	Transfer rate (tr)	0.5
Câu 6.4	Block transfer time (btt)	0.5
Câu 6.5	Time cho câu (1), (2)	0.5
Câu 7.1	Công thức	0.5
Câu 7.2	Tính thời gian	0.5
Câu 8.1	Xác định bucket và block tương ứng của record ID = 21	0.5
Câu 8.2	Số blocks	0.5
Câu 8.3	Tính thời gian	0.5

Họ - Tên:

Mã Số Sinh Viên:

Kiểm tra giữa kỳ #2

Môn: **Hệ Quản Trị Cơ Sở Dữ Liệu (503004)** – Ngành: Khoa Học Máy Tính

Thời gian làm bài: **45 phút**

(Bài kiểm tra gồm **4** câu hỏi. Sinh viên được tham khảo tài liệu.)

Câu 1 (2 điểm). Cho 2 quan hệ Nhân_Viên và Phòng_Ban có lược đồ như sau.

Nhân_Viên (ID, Tên, Lương, Phòng, Địa_Chỉ);

Phòng_Ban (ID_Phòng, Tên_Phòng, ID_Quản_Lý)

Trong đó, ID là khóa chính của quan hệ Nhân_Viên và Phòng là khóa ngoại của quan hệ Nhân_Viên tham chiếu đến ID_Phòng của quan hệ Phòng_Ban; ID_Phòng là khóa chính của quan hệ Phòng_Ban và Tên_Phòng là khóa dự tuyển của quan hệ Phòng_Ban.

Câu lệnh SQL sau dùng để truy vấn tên của tất cả nhân viên làm việc ở phòng 'RESEARCH' có lương lớn hơn hay bằng 2000:

```
SELECT Tên
FROM Nhân_Viên, Phòng_Ban
WHERE Phòng = ID_Phòng
AND Tên_Phòng = 'RESEARCH'
AND Lương >= 2000;
```

Hãy trình bày các bước thực hiện để có được *cây truy vấn tối ưu (optimized query tree)* biểu diễn câu lệnh SQL trên *hiệu quả nhất* dựa trên các qui tắc biến đổi tối ưu (*optimization transformation rule*).

Câu 2 (4 điểm). Hiện tại, tập tin dữ liệu của quan hệ Nhân_Viên được sắp thứ tự theo vùng tin khóa ID, gồm 5000 bản ghi ($r_{NV} = 5000$ records), trong 500 khối ($b_{NV} = 500$ blocks) với hệ số phân khối là 10 bản ghi/khối ($bfr_{NV} = 10$ records/block). Trên vùng tin khóa ID, một chỉ mục sơ cấp đa mức B+ được định nghĩa có 4 mức ($x_{ID} = 4$) và số lượng khối chỉ mục ở tầng lá là 20 ($bl_{ID} = 20$ blocks).

Tập tin dữ liệu của quan hệ Phòng_Ban được sắp thứ tự theo vùng tin khóa ID_Phòng, gồm 100 bản ghi ($r_{PB} = 100$ records), trong 10 khối ($b_{PB} = 10$ blocks) với hệ số phân khối là 10 bản ghi/khối ($bfr_{PB} = 10$ records/block). Trên vùng tin không khóa ID_Quản_Lý, một chỉ mục thứ cấp đa mức B+ được định nghĩa có 2 mức ($x_{ID_Quản_Lý} = 2$) và số lượng khối chỉ mục ở tầng lá là 3 ($bl_{ID_Quản_Lý} = 3$ blocks). Biết rằng, mỗi nhân viên chỉ quản lý 1 phòng ban và mỗi phòng ban chỉ được quản lý bởi 1 nhân viên.

Họ - Tên:

Mã Số Sinh Viên:

Thực hiện phép kết bằng: Nhân_Viên $\bowtie_{ID=ID_Quản_Lý}$ Phòng_Ban. Cho trước độ chọn lọc kết (*join selectivity*) là $js = 1/r_{NV} = 1/5000$ và hệ số phân khối của tập tin kết quả kết là $bfr_{NV_PB} = 5$ records/block.

2.1 (1.5 điểm). Giả sử kích thước vùng đệm là 10 khối ($b_{buffer}=10$ blocks). Xác định chi phí của phép kết bằng giữa quan hệ Nhân_Viên và quan hệ Phòng_Ban ở trên với phương thức hiện thực kết sắp thứ tự-trộn (*sort-merge join*).

2.2 (2.5 điểm). Xác định chi phí của phép kết bằng giữa quan hệ Nhân_Viên và quan hệ Phòng_Ban với phương thức hiện thực kết 1 vòng lặp (*single-loop join*). Trong trường hợp hiện thực phép kết này, tập tin dữ liệu nào nên được duyệt trên vòng lặp?

Câu 3 (3 điểm). Cho các giao tác T1 và T2 như sau:

Giao tác T1: $r_1(X); w_1(X); r_1(Y); w_1(Y); c_1$.

Giao tác T2: $r_2(Y); r_2(X); w_2(X); w_2(Y); c_2$.

3.1 (0.5 điểm). T1 commit tại c_1 . Điều này có nghĩa là gì ?

3.2 (0.75 điểm). Trong số các trình tự thực thi các tác vụ của các giao tác T1 và T2 này, xác định các trình tự thực thi đúng là một lịch biểu (*schedule*). Giải thích.

a. Trình tự S1 = $r_1(X); r_2(Y); w_2(Y); w_1(X); r_1(Y); r_2(X); w_2(X); c_2; w_1(Y); c_1$.

b. Trình tự S2 = $r_1(X); w_1(X); r_2(Y); r_2(X); r_1(Y); w_1(Y); w_2(X); w_2(Y); c_2; c_1$.

c. Trình tự S3 = $r_2(Y); r_2(X); r_1(X); r_1(Y); w_1(X); w_1(Y); c_1; w_2(X); w_2(Y); c_2$.

3.3 (1 điểm). Cho lịch biểu S4 của việc thực thi các tác vụ của các giao tác T1 và T2 như sau :

$S4 = r_1(X); w_1(X); r_2(Y); r_1(Y); w_1(Y); c_1; r_2(X); w_2(X); w_2(Y); c_2$.

Xác định đặc điểm khả phục hồi (*recoverability*) của S4 và xác định nội dung system log theo trình tự thực thi S4 này.

3.4 (0.75 điểm). Xác định đồ thị tuần tự hóa (*serialization graph*) của lịch biểu S4 và từ đó, xác định đặc điểm khả tuần tự hóa (*serializability*) của S4.

Câu 4 (1 điểm). Cho $X = 5$ và $Y = 10$ đã được bàn giao trong cơ sở dữ liệu. Hai giao tác T1 và T2 thực thi đồng thời trong môi trường đa người dùng nhưng điều khiển tương tranh không được thực hiện như sau.

Hãy xác định giá trị hiện tại của X và Y sau khi T1 và T2 đã commit.

Các giá trị này đúng hay không? Nếu không đúng thì hãy cho biết vấn đề điều khiển tương tranh nào sau đây đã xảy ra với việc thực thi đồng thời của T1 và T2?

Họ - Tên:

Mã Số Sinh Viên:

- a. Vấn đề lost update
- b. Vấn đề temporary update/dirty read
- c. Vấn đề incorrect summary
- d. Vấn đề unrepeatable read
- e. Vấn đề phantom

T1	T2
Read_item(X);	
X := X-5;	
	Read_item(X);
	X := X + 3;
Write_item(X);	
Read_item(Y);	
Y := Y * 2;	
	Write_item(X);
Write_item(Y);	
Commit;	
	Commit;

Phần trả lời

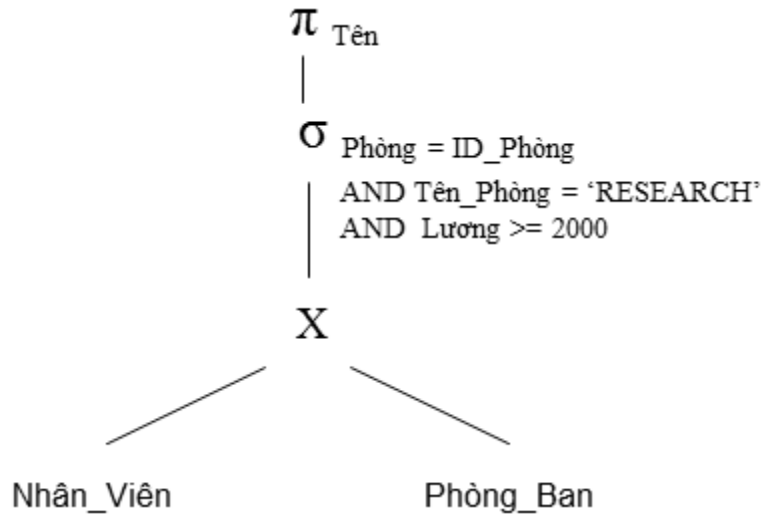
Câu 1.

```
SELECT Tên
FROM Nhân_Viên, Phòng_Ban
WHERE Phòng = ID_Phòng
AND Tên_Phòng = 'RESEARCH'
AND Lương >= 2000;
```

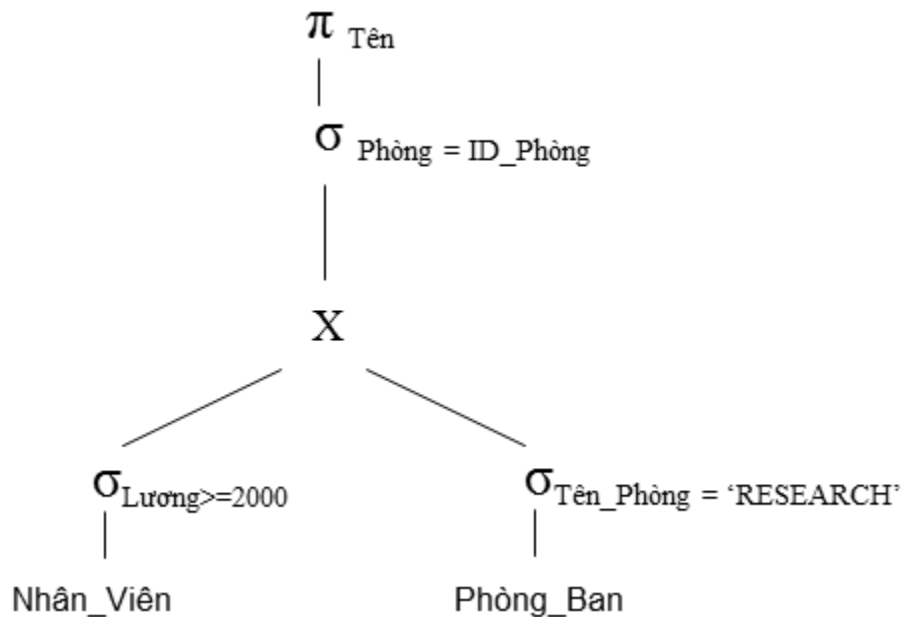
Cây truy vấn ban đầu:

Họ - Tên:

Mã Số Sinh Viên:



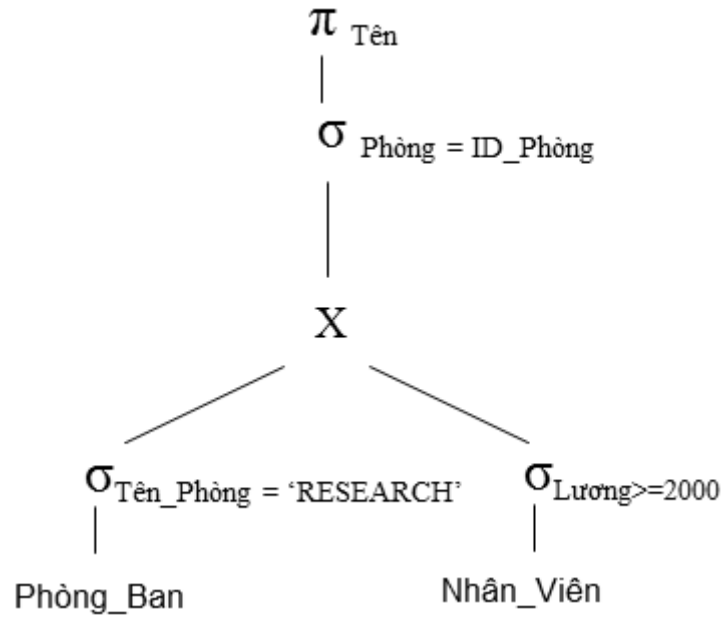
Dùng qui tắc 1 (cascade of selection) và qui tắc 6 (commuting selection with cartesian product x) để đưa các tác vụ chọn xuống dưới cùng:



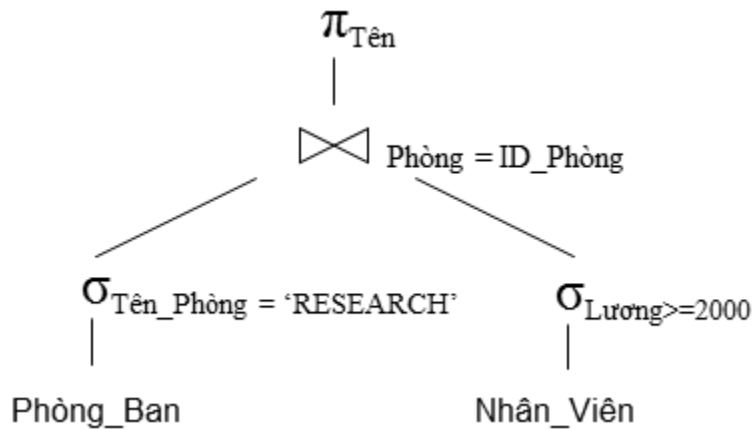
Dùng qui tắc 5 (commutativity of cartesian product x) để thực hiện tác vụ chọn trả về kết quả ít, phép chọn trên Phòng_Ban trả về kết quả ít hơn so với phép chọn trên Nhân_Viên do Tên_Phòng là khóa dự tuyển của Phòng_Ban và Lương là thuộc tính không khóa của Nhân_Viên và việc thực thi theo thứ tự duyệt cây LNR:

Họ - Tên:

Mã Số Sinh Viên:



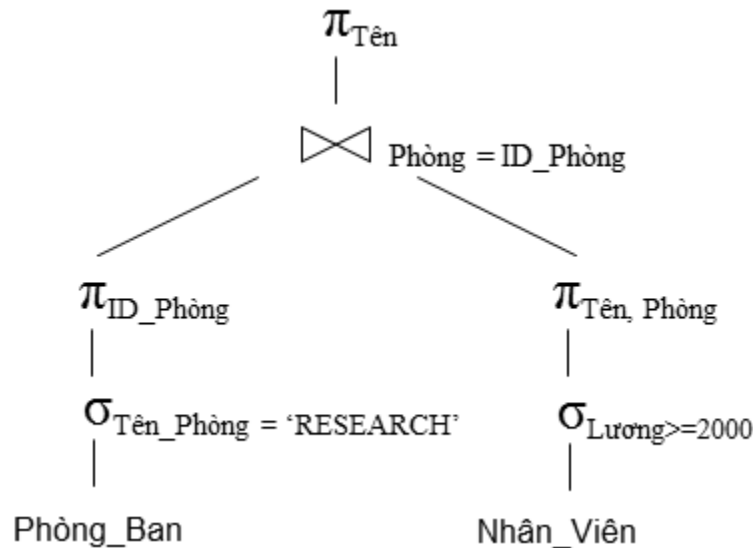
Dùng qui tắc 12 (converting a (σ, \times) sequence into \bowtie) để chuyển chuỗi phép chọn và phép tích cartesian thành phép kết:



Dùng qui tắc 7 (commuting projection with join) để giảm kết quả trung gian trước khi thực hiện tác vụ kết và cây truy vấn tối ưu cũng đạt được sau khi áp dụng qui tắc này:

Họ - Tên:

Mã Số Sinh Viên:



Câu 2.

2.1. Xác định chi phí của phép kết bằng giữa quan hệ Nhân_Viên và quan hệ Phòng_Ban ở trên với phương thức hiện thực kết sắp thứ tự-trộn (sort-merge join).

$$\text{Chi phí} = C_s + b_{NV} + b_{PB} + (js * r_{NV} * r_{PB}) / bfr_{NV_PB}$$

$$C_s = C_s \text{ trên Nhân_Viên} + C_s \text{ trên Phòng_Ban} = 0 + C_s \text{ trên Phòng_Ban}$$

Truy đặt thứ tự trên Phòng_Ban theo các giá trị của vùng tin ID_Quản_Lý có thể dùng chỉ mục thứ cấp trên vùng tin này hoặc có thể dùng sắp thứ tự ngoại Phòng_Ban theo các giá trị của vùng tin này.

Nếu dùng chỉ mục thứ cấp thì chi phí là:

$$C_s \text{ trên Phòng_Ban} = x_{ID_Quản_Lý} + b1_{ID_Quản_Lý} + r_{PB} = 2 + 3 + 100 = 105 \text{ block accesses}$$

Trong đó, $x_{ID_Quản_Lý} + b1_{ID_Quản_Lý}$ là chi phí truy đặt trên chỉ mục và r_{PB} là chi phí truy đặt từng bản ghi một trên tập tin Phòng_Ban.

Nếu dùng sắp thứ tự ngoại thì chi phí là:

$$C_s \text{ trên Phòng_Ban} = 2b_{PB} + 2b_{PB} * \log_{\text{buffer-1}}(n_{Run}) = 2 * 10 + 2 * 10 * \log_9 1 = 20 \text{ block accesses}$$

$$C_s \text{ trên Phòng_Ban} = \min \{ 105, 20 \} = 20 \text{ block accesses}$$

$$C_s = 20 \text{ block accesses}$$

$$b_{NV} = 500; b_{PB} = 10; js = 1/r_{NV} = 1/5000; r_{NV} = 5000; r_{PB} = 100; bfr_{NV_PB} = 5$$

Do đó, chi phí cho phép kết dùng sort-merge join là:

$$C = C_s + b_{NV} + b_{PB} + (js * r_{NV} * r_{PB}) / bfr_{NV_PB} = 20 + 500 + 10 + ((1/5000) * 5000 * 100) / 5$$

Họ - Tên:

Mã Số Sinh Viên:

$C = 550$ block accesses

2.2. Xác định chi phí của phép kết bằng giữa quan hệ Nhân_Viên và quan hệ Phòng_Ban với phương thức hiện thực kết 1 vòng lặp (single-loop join).

* Khi Nhân_Viên được duyệt trên vòng lặp và sử dụng chỉ mục thứ cấp trên vùng tin ID_Quản_Lý của Phòng_Ban, chi phí của phép kết dùng single-loop join là:

$$C = b_{NV} + r_{NV} * (x_{ID_Quản_Lý} + s_{ID_Quản_Lý}) + (js * r_{NV} * r_{PB}) / bfr_{NV_PB}$$

$$C = 500 + 5000 * (2 + 1) + ((1/5000) * 5000 * 100) / 5 = 15520 \text{ block accesses}$$

* Khi Phòng_Ban được duyệt trên vòng lặp và sử dụng chỉ mục sơ cấp trên vùng tin ID của Nhân_Viên, chi phí của phép kết dùng single-loop join là:

$$C = b_{PB} + r_{PB} * (x_{ID} + 1) + (js * r_{NV} * r_{PB}) / bfr_{NV_PB}$$

$$C = 10 + 100 * (4 + 1) + ((1/5000) * 5000 * 100) / 5 = 530 \text{ block accesses}$$

Do đó, chi phí của phép kết dùng single-loop join là $C = \min \{15520, 530\} = 530$ block accesses

Trong trường hợp hiện thực phép kết này, tập tin dữ liệu Phòng_Ban nên được duyệt trên vòng lặp.

Câu 3.

3.1. T1 commit tại c_1 . Điều này có nghĩa là: Tại điểm c_1 , T1 đã hoàn thành thành công tất cả các thao tác xử lý trên cơ sở dữ liệu và những tác động của các thao tác đó đều đã được ghi nhận trong system log.

3.2. Xác định các trình tự thực thi đúng là một lịch biểu (schedule). Giải thích.

a. Trình tự $S1 = r_1(X); r_2(Y); w_2(Y); w_1(X); r_1(Y); r_2(X); w_2(X); c_2; w_1(Y); c_1$.

→ S1 không là một lịch biểu do trình tự thực thi của các thao tác của giao tác T2 trong S1 là $r_2(Y); w_2(Y); r_2(X); w_2(X); c_2$ không đúng với trình tự thực thi của các thao tác của giao tác T2 khi không thực hiện đồng thời với giao tác T1.

b. Trình tự $S2 = r_1(X); w_1(X); r_2(Y); r_2(X); r_1(Y); w_1(Y); w_2(X); w_2(Y); c_2; c_1$.

→ S2 là một lịch biểu do trình tự thực thi của các thao tác của giao tác T1 và giao tác T2 đều đúng với trình tự thực thi của các thao tác của các giao tác này khi không thực hiện đồng thời với nhau.

c. Trình tự $S3 = r_2(Y); r_2(X); r_1(X); r_1(Y); w_1(X); w_1(Y); c_1; w_2(X); w_2(Y); c_2$.

Họ - Tên:

Mã Số Sinh Viên:

→ S3 không là một lịch biểu do trình tự thực thi của các thao tác của giao tác T1 trong S3 là $r_1(X); r_1(Y); w_1(X); w_1(Y); c_1$ không đúng với trình tự thực thi của các thao tác của giao tác T1 khi không thực hiện đồng thời với giao tác T2.

3.3. Xác định đặc điểm khả phục hồi (recoverability) của S4 và xác định nội dung system log theo trình tự thực thi S4 này.

$S4 = r_1(X); w_1(X); r_2(Y); r_1(Y); w_1(Y); c_1; r_2(X); w_2(X); w_2(Y); c_2.$

* Kiểm tra liệu S4 có là một lịch biểu nghiêm cách (strict schedule): Cho giao tác T1, tất cả các tác vụ đọc và ghi của T1 đều thực hiện trên dữ liệu trực tiếp từ cơ sở dữ liệu; nghĩa là, dữ liệu do T1 đọc và ghi đều được cập nhật lần cuối bởi các giao tác nào đó đã commit. Cho giao tác T2, tác vụ đọc Y của giao tác T2 được thực hiện trên dữ liệu Y từ cơ sở dữ liệu, nghĩa là dữ liệu Y này đã được cập nhật lần cuối bởi giao tác nào đó đã commit. Khi giao tác T2 đọc và ghi X, giá trị dữ liệu mới được cập nhật ghi bởi giao tác T1 ; nhưng khi này, giao tác T1 vừa commit trước đó nên việc đọc/ghi X của T2 được thực hiện trên dữ liệu X đã được cập nhật bởi giao tác T1 đã commit. Tương tự cho thao tác ghi Y của T2, khi này giao tác T1 đã ghi dữ liệu Y và cũng đã commit trước khi T2 thực hiện ghi Y. Điều này có nghĩa là: dữ liệu do T1 đọc và ghi đều được cập nhật lần cuối bởi các giao tác nào đó và giao tác T1 đã commit.

Do đó, S4 là một lịch biểu nghiêm cách (strict), có nghĩa là S4 cũng là một lịch biểu không dất dây (cascadeless schedule) và cũng là một lịch biểu khả phục hồi (recoverable schedule).

Sổ ghi hệ thống system log cho phần trình tự thực thi của S4 là:

[start_transaction, T1] ;

[write_item, T1, X, old_value];

[start_transaction, T2];

[write_item, T1, Y, old_value];

[commit, T1];

[write_item, T2, X, old_value];

[write_item, T2, Y, old_value];

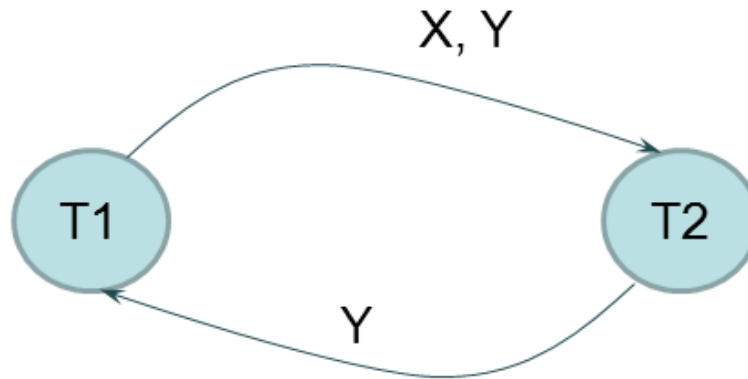
[commit, T2];

3.4. Xác định đồ thị tuần tự hóa (serialization graph) của lịch biểu S4 và từ đó, xác định đặc điểm khả tuần tự hóa (serializability) của S4.

Đồ thị tuần tự hóa (serialization graph) của lịch biểu S4 như sau:

Họ - Tên:

Mã Số Sinh Viên:



Do đồ thị tuần tự hóa (serialization graph) ở trên có chu trình (Y(T1, T2), Y(T2, T1)) nên lịch biểu S4 không là một lịch biểu khả tuần tự hóa xung đột (not a conflict-serliazable schedule).

Câu 4. Hãy xác định giá trị hiện tại của X và Y sau khi T1 và T2 đã commit.

Các giá trị này đúng hay không? Nếu không đúng thì hãy cho biết vấn đề điều khiển tương tranh nào sau đây đã xảy ra với việc thực thi đồng thời của T1 và T2?

T1	T2	X(DB)	Y(DB)	X(buffer – T1)	X(buffer – T2)	Y(buffer – T1)
		5	10			
Read_item(X);				5		
X := X-5;				0		
	Read_item(X);				5	
	X := X + 3;				8	
Write_item(X);		0				
Read_item(Y);						10
Y := Y * 2;						20
	Write_item(X);	8				
Write_item(Y);			20			
Commit;						
		8	20			
	Commit;					
		8	20			

Các giá trị hiện tại của X và Y sau khi T1 và T2 đã commit là: X = 8 và Y = 20.

Các giá trị này không đúng do vấn đề lost update đã xảy ra đối với T1 khi việc thực thi đồng thời của T1 và T2 diễn ra nhưng không có điều khiển tương tranh.

Cụ thể là trong trường hợp không có việc thực thi đồng thời của T1 và T2:

Họ - Tên:

Mã Số Sinh Viên:

- Giả sử T1 diễn ra trước T2 : $X = 3$ và $Y = 20$ là các giá trị đúng.
- Giả sử T2 diễn ra trước T1 : $X = 3$ và $Y = 20$ là các giá trị đúng.

Cách tính điểm

Câu	Nội dung	Điểm tối đa
Câu 1	Tối ưu hóa truy vấn dựa trên qui tắc (rule-based optimization)	2
1.1	Cây truy vấn ban đầu	0.5
1.2	Dùng qui tắc 1 và 6	0.5
1.3	Dùng qui tắc 5	0.25
1.4	Dùng qui tắc 12	0.5
1.5	Dùng qui tắc 7	0.25
Câu 2	Tối ưu hóa truy vấn dựa trên chi phí (cost-based optimization)	4
2.1	Chi phí cho sort-merge join	1.5
2.1.1	Chi phí sắp thứ tự các tập tin	1
2.1.2	Chi phí tổng cho phép kết	0.5
2.2	Chi phí cho single-loop join	2.5
2.2.1	Chi phí với tập tin Nhân_Viên trên vòng lặp	0.75
2.2.2	Chi phí với tập tin Phòng_Ban trên vòng lặp	0.75
2.2.3	Chi phí tổng cho phép kết và kết luận lựa chọn đúng	0.5
Câu 3	Xử lý giao tác và các đặc điểm của lịch biểu	3
3.1	Commit point	0.5
3.1.1	Commit point của giao tác T1	0.5
3.2	Xác định các lịch biểu hợp lệ	0.75
3.2.1	Xác định S1 có là lịch biểu hay không	0.25
3.2.2	Xác định S2 có là lịch biểu hay không	0.25
3.2.3	Xác định S3 có là lịch biểu hay không	0.25
3.3	Đặc điểm khả phục hồi của lịch biểu và nội dung system log tùy vào đặc điểm khả phục hồi này	1
3.3.1	Đặc điểm khả phục hồi của lịch biểu S4	0.5
3.3.2	Nội dung system log	0.5
3.4	Đặc điểm tuần tự hóa của lịch biểu và cách xác định	0.75
3.4.1	Đồ thị tuần tự hóa	0.5
3.4.2	Xác định đặc điểm khả tuần tự hóa của lịch biểu từ đồ thị tuần tự hóa của lịch biểu	0.25
Câu 4	Vấn đề tại sao cần có điều khiển tương tranh trong môi trường đa người dùng đồng thời	1
4.1	Tính đúng giá trị X và Y	0.5
4.2	Xác định đúng vấn đề về điều khiển tương tranh	0.5
Tổng điểm		10.0