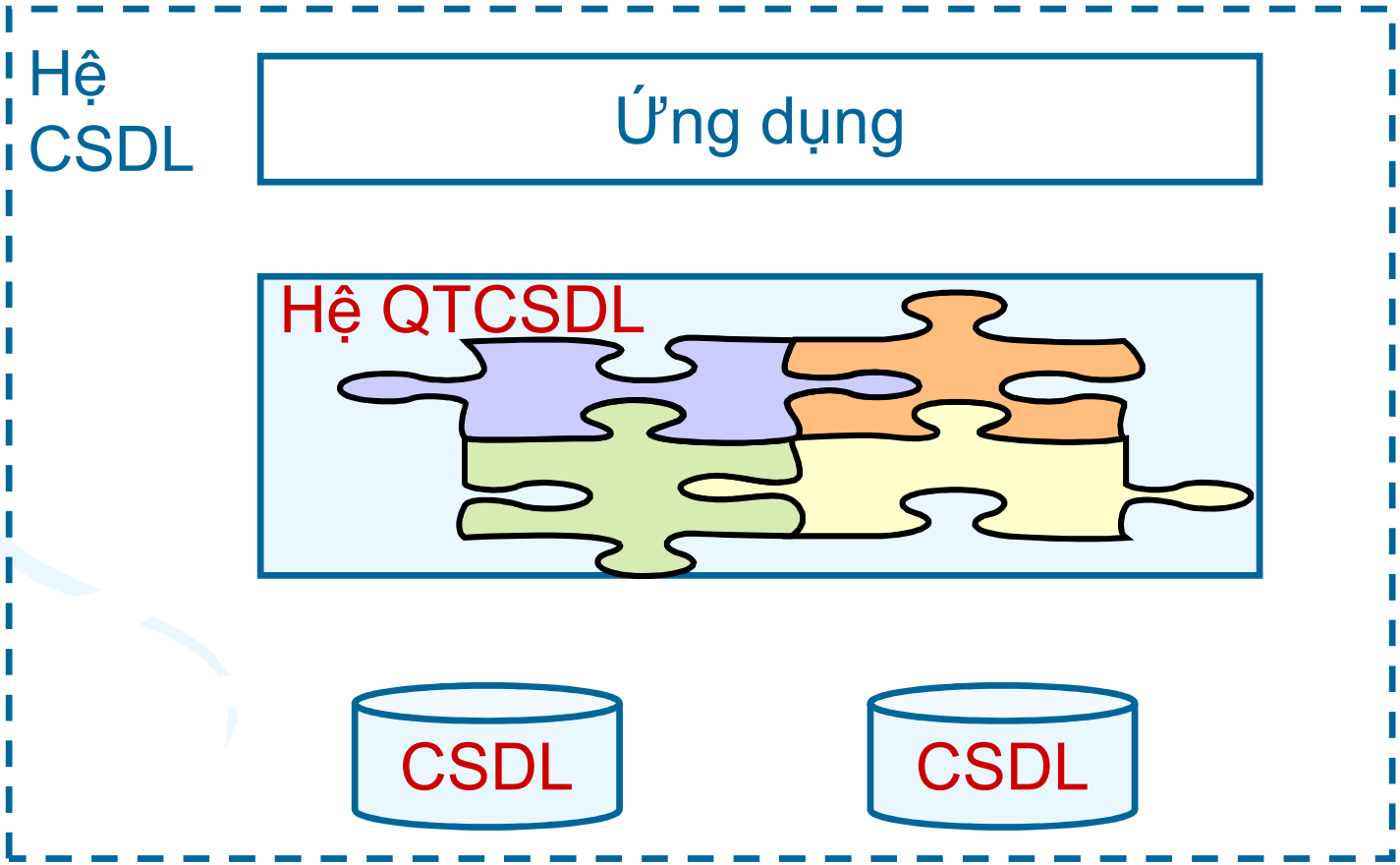
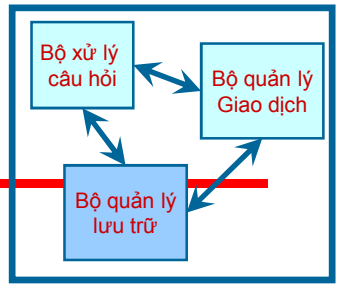
Tổ chức dữ liệu vật lý

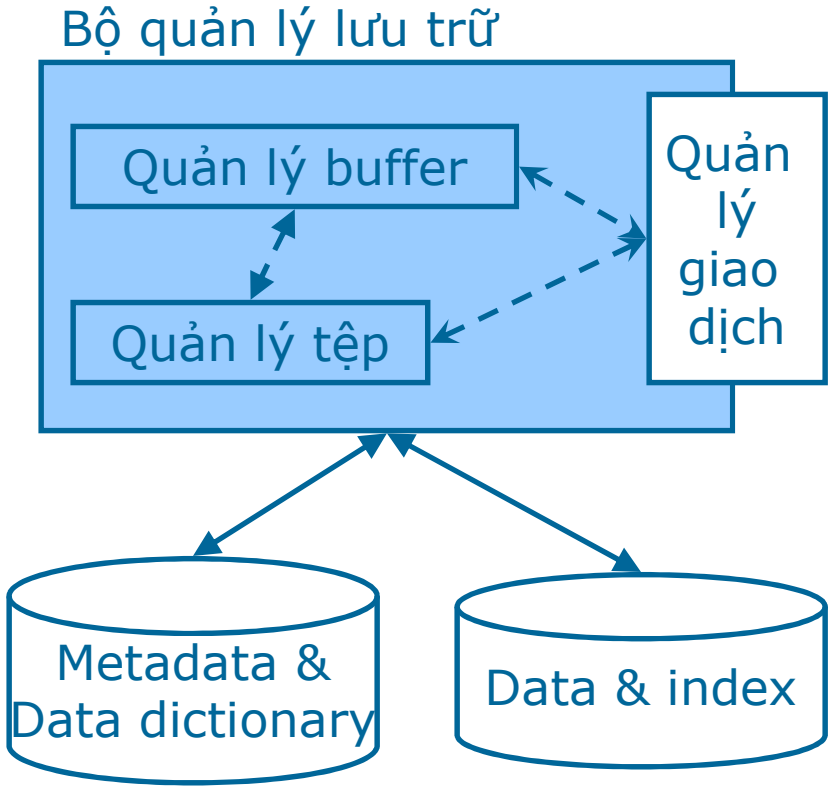
Giới thiệu



Quản lý lưu trữ



- Tổ chức tệp: sắp xếp các bản ghi trên thiết bị nhớ ngoài
  - RID (*record id):* xác định địa chỉ vật lý của các bản ghi
  - chỉ dẫn (index): cấu trúc dữ liệu xác định sự tương ứng giữa RID của bản ghi và giá trị của trường (khoá)
- Vùng nhớ đệm: trung gian giữa thiết bị nhớ ngoài và bộ nhớ trong (có thể sử dụng cho cả DL và chỉ số)



Tổ chức bộ nhớ ngoài

- Mục đích: giảm thiểu truy xuất đến dữ liệu không cần thiết trên thiết bị nhớ ngoài

- Các vấn đề cần quan tâm
  - Cấu trúc lưu trữ
  - Các phép toán (thêm, xoá, sửa, tìm kiếm)

Các thiết bị nhớ ngoài

- Đĩa từ, băng từ, trống từ, ...

- Đĩa từ: được tổ chức thành từng trang
  - Chí phí truy nhập đến các trang bất kỳ là tương đương
  - Chí phí đọc nhiều trang liền nhau < chí phí đọc các trang đó theo thứ tự bất kỳ
- Băng từ:
  - chỉ có thể đọc được các trang liền nhau
  - rẻ hơn đĩa từ nhưng chi phí truy nhập thương lớn hơn
- ...

Đĩa từ vs. bộ nhớ trong

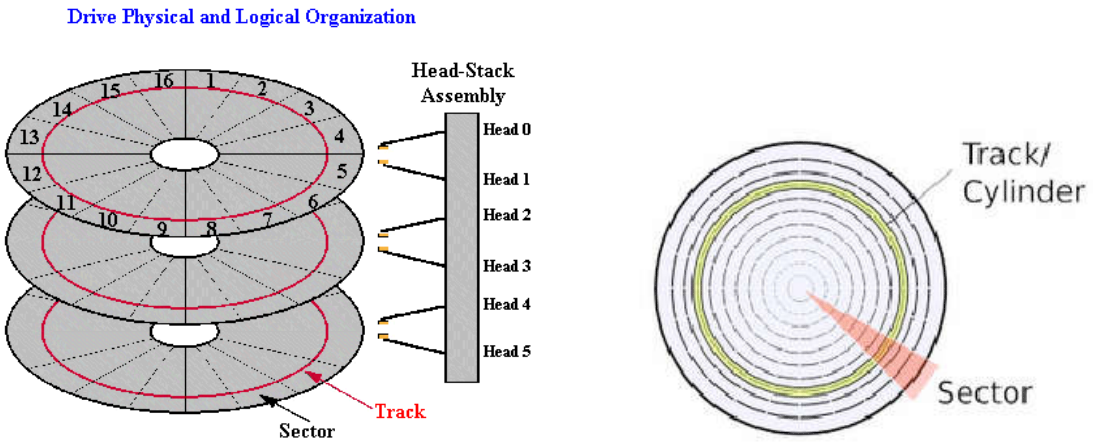
- Tốc độ truy nhập bộ
  ms vs. ns (~1000 lần)
- Kích thước
  GB vs. 10x MB (~ 100 lần với cùng chi phí)
- Lưu trữ
  ổn định (kể cả khi mất điện) vs. tạm thời
- Phân chia block
  4KB vs. 1Byte

Nội dung

- 1. Mô hình tổ chức bộ nhớ ngoài
- 2. Tổ chức tệp đống
- 3. Tổ chức tệp băm
- 4. Tổ chức tệp chỉ dẫn

1. Mô hình tổ chức bộ nhớ ngoài

- Bộ nhớ ngoài (bộ nhớ thứ cấp): đĩa từ, băng từ,...



- Đĩa được chia thành các khối vật lý có kích cỡ như nhau - 512 bytes đến 4096 bytes được đánh địa chỉ khối là một địa chỉ tuyệt đối
- Mỗi tệp dữ liệu chiếm 1 hoặc nhiều khối
- Mỗi khối chứa 1 hoặc nhiều bản ghi

1. Mô hình tổ chức bộ nhớ ngoài

- Thao tác với dữ liệu của tệp thông qua tên tệp-địa chỉ tuyệt đối của khối đầu tiên.
- Các bản ghi đều có địa chỉ bản ghi:
  - địa chỉ tuyệt đối của byte đầu tiên
  - địa chỉ khối và số byte tính từ đầu khối đến vị trí đầu bản ghi
- Địa chỉ của các bản ghi/khối được lưu ở 1 tệp => sử dụng con trỏ (pointer) để truy cập dữ liệu của tệp.

Mô hình tổ chức bộ nhớ ngoài

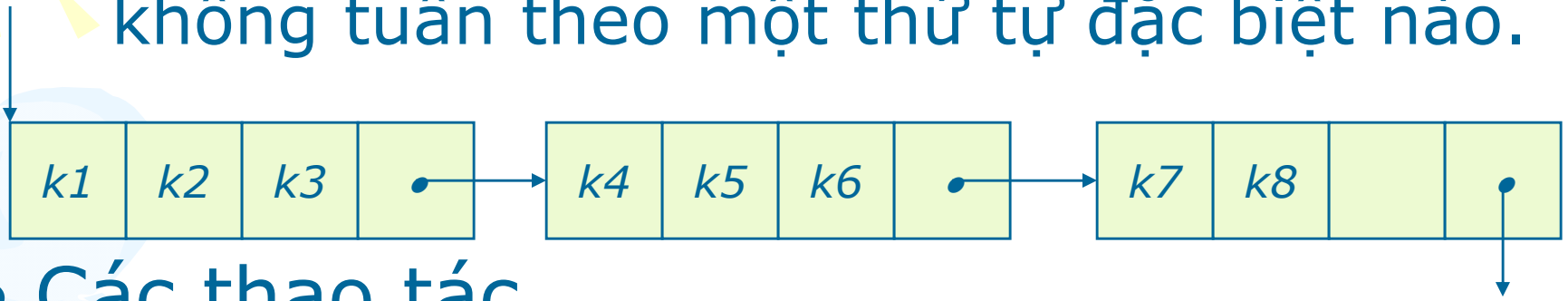
- Quá trình thiết kế CSDL vật lý kéo theo các cách tổ chức dữ liệu cụ thể để phù hợp nhất với các thao tác cần thiết (on SELECT, INSERT, UPDATE, DELETE).
- Dữ liệu được lưu trữ trên đĩa từ được tổ chức như các tệp chứa các bản ghi:
  - Các tổ chức tệp chính: xác định cách sắp đặt các bản ghi trên đĩa để có thể truy cập.
  - Tổ chức thứ cấp: cấu trúc truy cập phụ cho phép truy cập hiệu quả đến các bản ghi theo các trường xác định.

Các tổ chức tệp chính

- Tệp các bản ghi không được sắp(Heap Files)
- Tệp các bản ghi được sắp (Sorted Files)
- Tệp băm

2. Tổ chức tệp đống (Heap file)

- Tổ chức dữ liệu
  - Các bản ghi lưu trữ kế tiếp trong các khối, không tuân theo một thứ tự đặc biệt nào.



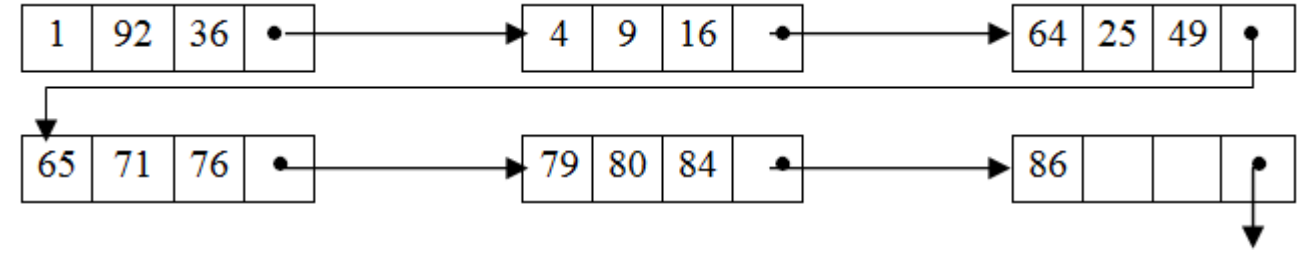
- Các thao tác
  - Tìm kiếm một bản ghi: tìm kiếm một bản ghi có giá trị khóa cho trước => quét toàn bộ tệp.
  - Thêm một bản ghi: thêm bản ghi mới vào sau bản ghi cuối cùng

2. Tổ chức tệp đống (Heap file)

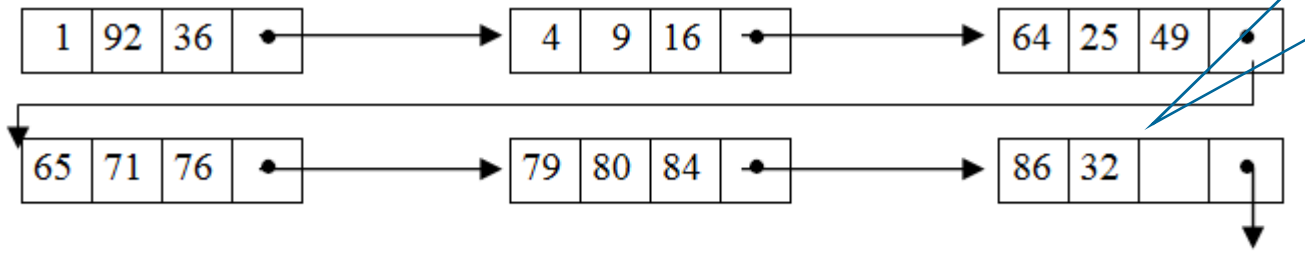
- Các thao tác (tiếp)
  - Xóa một bản ghi: thao tác xóa bao hàm thao tác tìm kiếm. Nếu có bản ghi cần xóa thì nó sẽ được đánh dấu là xóa => hệ thống cần tổ chức lại đĩa định kỳ.
  - Sửa một bản ghi: tìm bản ghi rồi sửa một hay nhiều trường.

2. Tổ chức tệp đống (Heap file)

- Ví dụ:

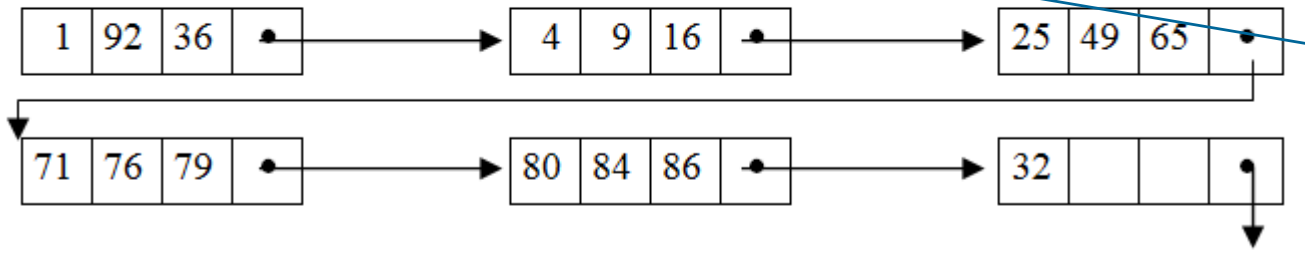


(a)



Thêm bản ghi có giá trị khóa là 32

(b)



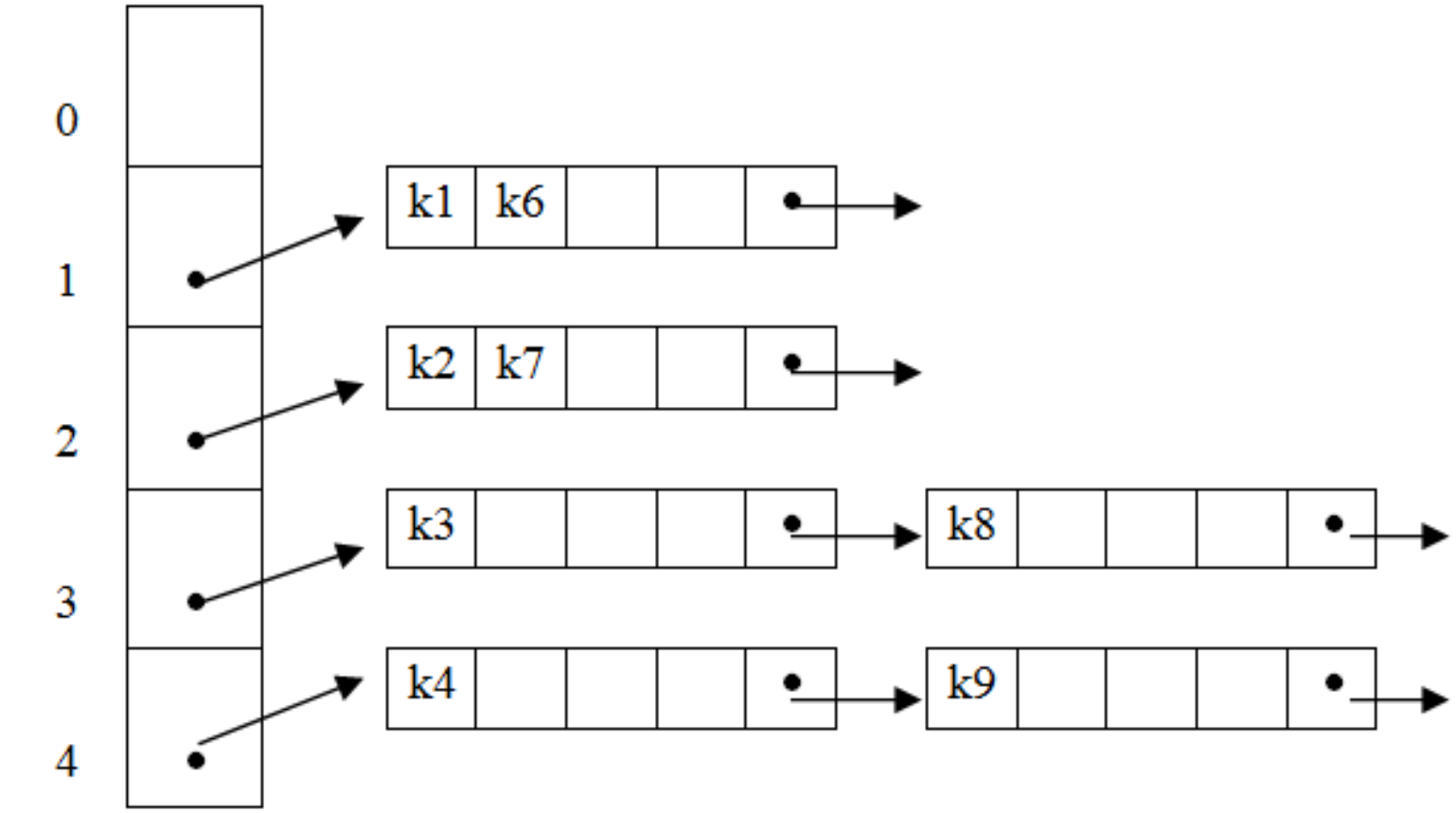
Xóa bản ghi có giá trị khóa là 64

(c)

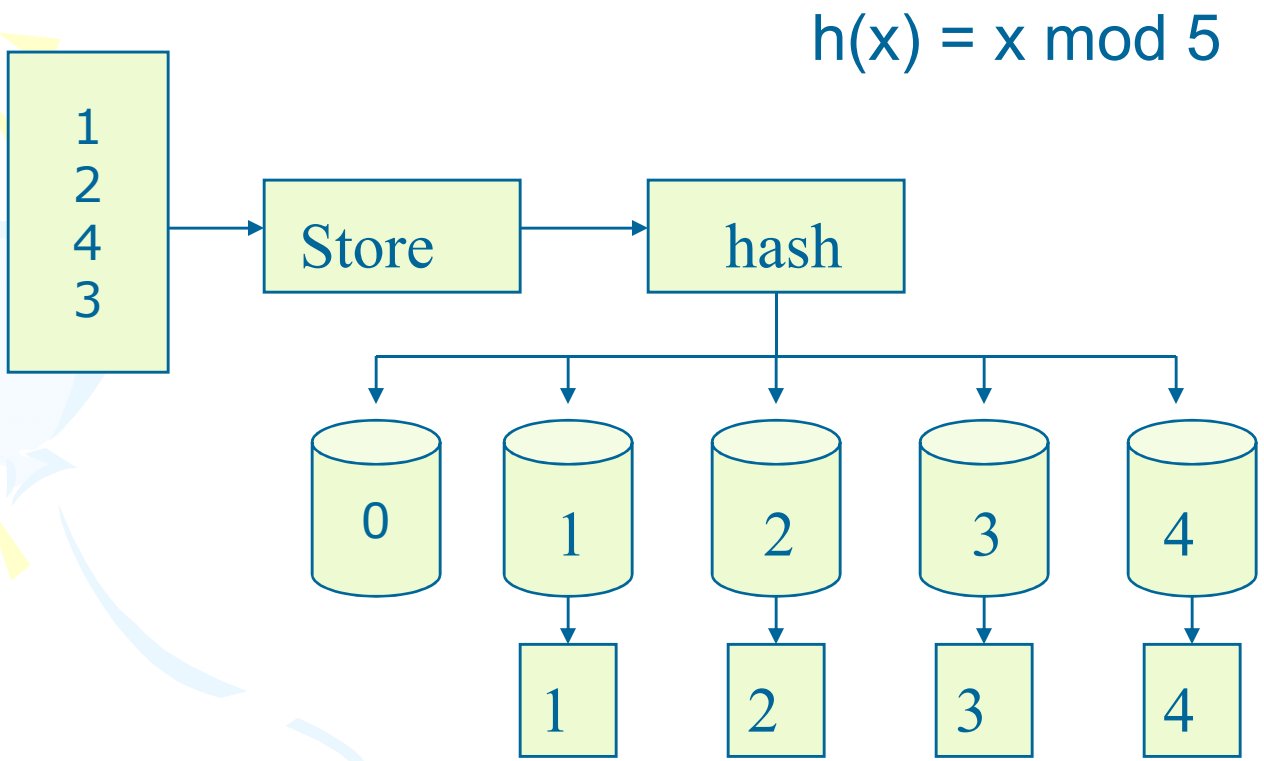
3. Tổ chức tệp băm (Hashed files)

- Hàm băm: h(x) nhận một giá trị trong đoạn [0,k], ví dụ: h(x)=x mod k
- Tổ chức tệp dữ liệu
  - Phân chia các bản ghi vào các cụm.
  - Mỗi cụm gồm một hoặc nhiều khối.
  - Mỗi khối chứa số lượng bản ghi cố định.
  - Tổ chức lưu trữ dữ liệu trong mỗi cụm áp dụng theo tổ chức đống
- Tiêu chí chọn hàm băm: phân bố các bản ghi tương đối đồng đều theo các cụm.

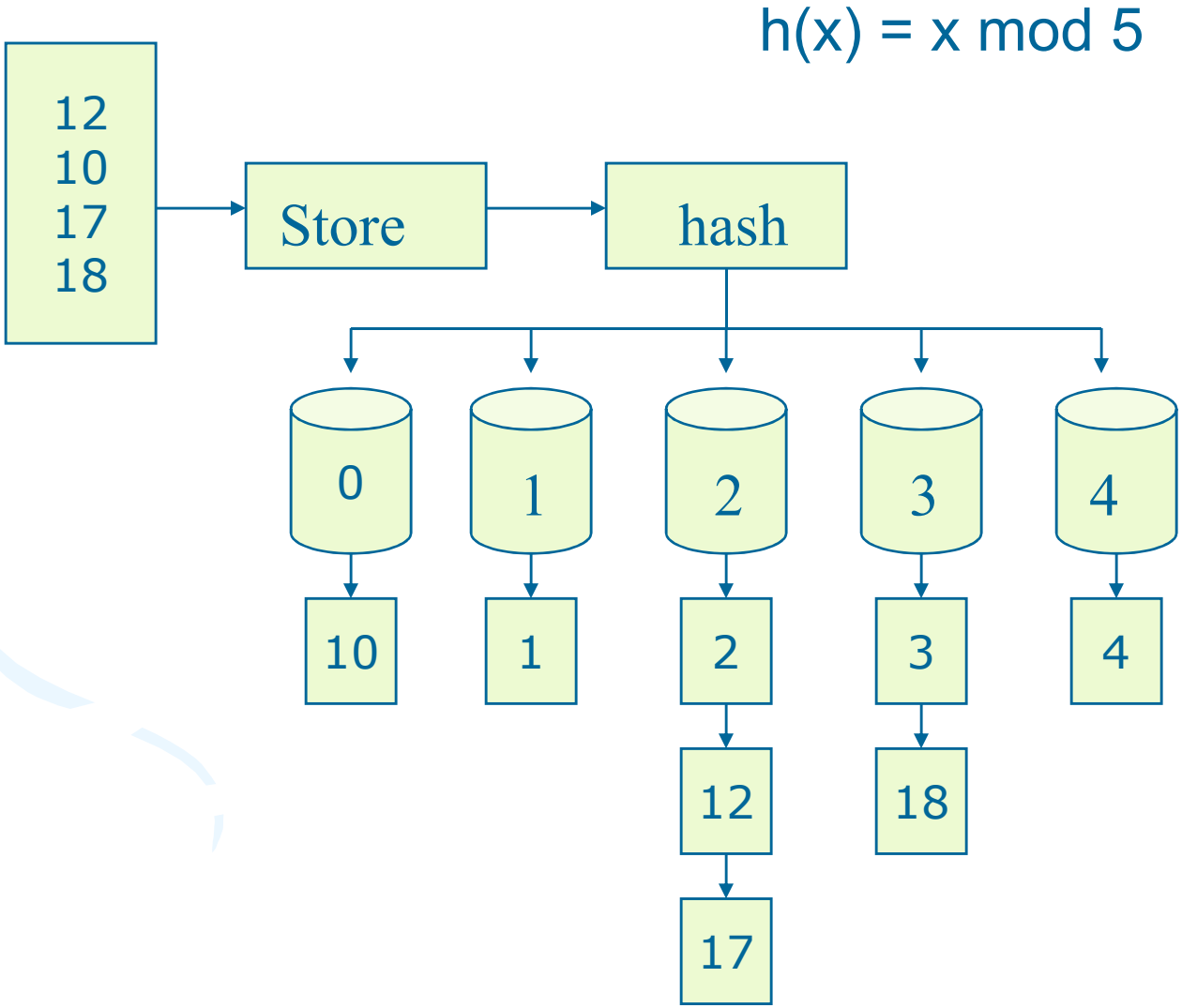
3. Tổ chức tệp băm (Hashed files)



3. Tổ chức tệp băm (Hashed files)



3. Tổ chức tệp băm (Hashed files)



3. Tổ chức tệp băm (Hashed files)

- Các thao tác
  - Tìm kiếm một bản ghi: để tìm bản ghi có khóa x, tính h(x) sẽ được cụm chứa bản ghi, sau đó tìm kiếm theo tổ chức đống.
  - Thêm một bản ghi: thêm 1 bản ghi có giá trị khóa là x.
    - nếu trong tệp đã có một bản ghi có trùng khóa x =>bản ghi mới sai (vì khóa là duy nhất!)
    - nếu không có bản ghi trùng khóa, bản ghi được thêm vào khối còn chỗ trống đầu tiên trong cụm, nếu hết chỗ thì tạo khối mới.

3. Tổ chức tệp băm (Hashed files)

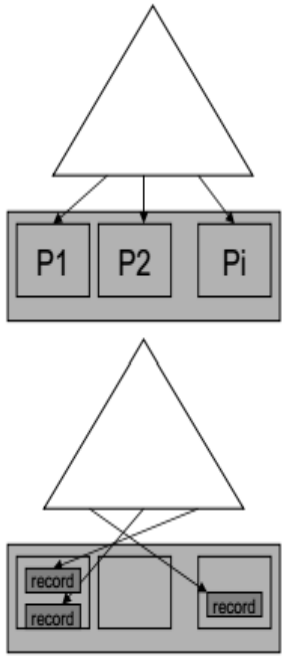
- Xóa một bản ghi: tìm kiếm bản ghi rồi xóa
- Sửa đổi một bản ghi:
  - nếu trường cần sửa có tham gia vào trong khóa thì việc sửa sẽ là loại bỏ bản ghi này và thêm mới 1 bản ghi (bản ghi có thể thuộc vào 1 cụm khác)
  - nếu trường cần sửa không thuộc khóa: tìm kiếm rồi sửa. Nếu bản ghi không tồn tại thì xem như có lỗi.

4. Tổ chức tệp chỉ dẫn(Indexed Files)

- Tệp chỉ dẫn là một tổ chức thứ cấp hay một cấu trúc truy cập phụ cho phép truy cập hiệu quả đến các bản ghi theo các trường xác định.
- Các loại chỉ dẫn:
  - Chỉ dẫn thưa đ/v chỉ dẫn đặc
  - Chỉ dẫn nhóm đ/v không nhóm
  - B-cây

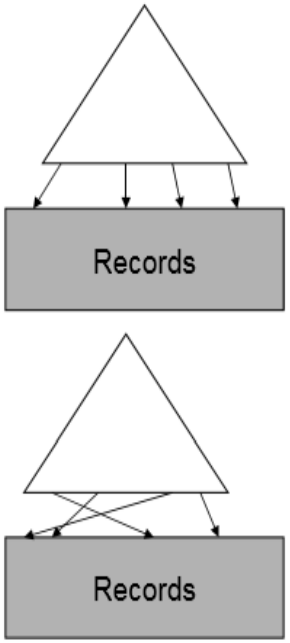
Chỉ dẫn thưa đ/v chỉ dẫn đặc

- Chỉ dẫn thưa
  - Các con trỏ trỏ tới các khối trên đĩa từ
  - Số con trỏ ít hơn nhiều số bản ghi
- Chỉ dẫn đặc
  - Các con trỏ trỏ tới các bản ghi
  - Số con trỏ bằng số bản ghi
  - Các khóa chỉ dẫn lặp lại lưu trữ chỉ 1 lần



Chỉ dẫn nhóm đ/v chỉ dẫn không nhóm

- Chỉ dẫn nhóm trên thuộc tính X
  - Chỉ dẫn này điều khiển việc đặt các bản ghi trên đĩa
  - Chỉ có 1 chỉ dẫn nhóm trên 1 bảng
  - Chỉ dẫn có thể đặc hay thưa
- Chỉ dẫn không nhóm trên thuộc tính
  - Không có ràng buộc đ/v tổ chức tệp
  - Có thể có nhiều chỉ dẫn không nhóm trên bảng
  - luôn là chỉ dẫn đặc

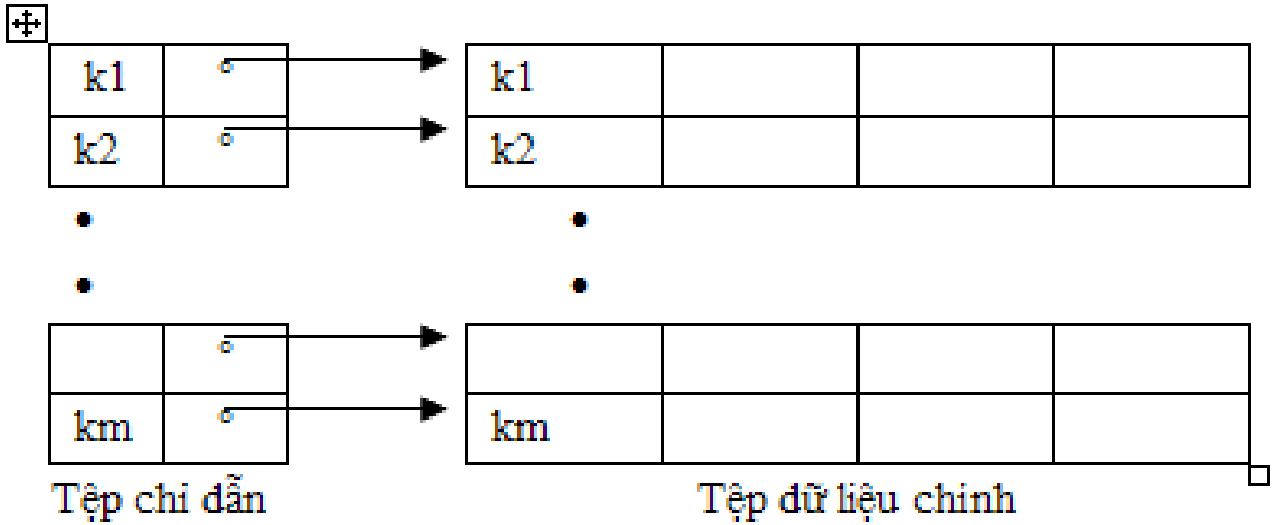


Các cấu trúc dữ liệu chỉ dẫn

- Các chỉ dẫn có thể được cài đặt với các cấu trúc dữ liệu khác nhau:
  - B+-tree index
  - R-tree: index for special data (points, lines, shapes)
  - quadtree: recursively partition a 2D plane into four quadrants
  - main memory indexes: T-tree, binary search tree

Tổ chức tệp chỉ dẫn thưa

- Giả sử giá trị các khóa của các bản ghi được sắp xếp tăng dần(tệp được sắp)
- Tệp chỉ dẫn được tạo bằng cách chọn các giá trị khóa trong các bản ghi
- Tệp chỉ dẫn bao gồm các cặp (k,d), trong đó k là giá trị khoá của bản ghi đầu tiên, d là địa chỉ của khối (hay con trỏ khối).



Tổ chức tệp chỉ dẫn thưa

- Tìm kiếm trên tệp chỉ dẫn
  - Cho một giá trị khóa k_i, tìm một bản ghi (k_m,d) trong tệp chỉ dẫn sao cho $k_m <= k_i$ và:
    - hoặc (k_m,d) là bản ghi cuối cùng trong tệp chỉ dẫn
    - hoặc bản ghi tiếp theo (k_{m+1},d') thỏa mãn $k_i < k_{m+1}$
  - Khi đó, chúng ta nói k_m phủ k_i
  - Tìm kiếm này có thể là:
    - tuần tự
    - nhị phân

Tổ chức tệp chỉ dẫn thưa

- Các thao tác
  - Tìm kiếm một bản ghi
  - Thêm một bản ghi: xác định khối i sẽ chứa bản ghi đó
    - nếu trong khối i còn chỗ thì đặt bản ghi này vào đúng chỗ theo thứ tự sắp xếp của khóa, dồn toa các bản ghi đằng sau nó.
    - nếu khối i hết chỗ thì việc thêm này sẽ đẩy bản ghi cuối cùng trong khối sang làm bản ghi đầu tiên của khối tiếp theo i+1 => sửa bản ghi chỉ dẫn tương ứng
    - nếu bản ghi mới này có giá trị khóa lớn hơn tất cả mọi khóa trong tệp dữ liệu chính và không còn chỗ thì tạo thêm một khối mới.

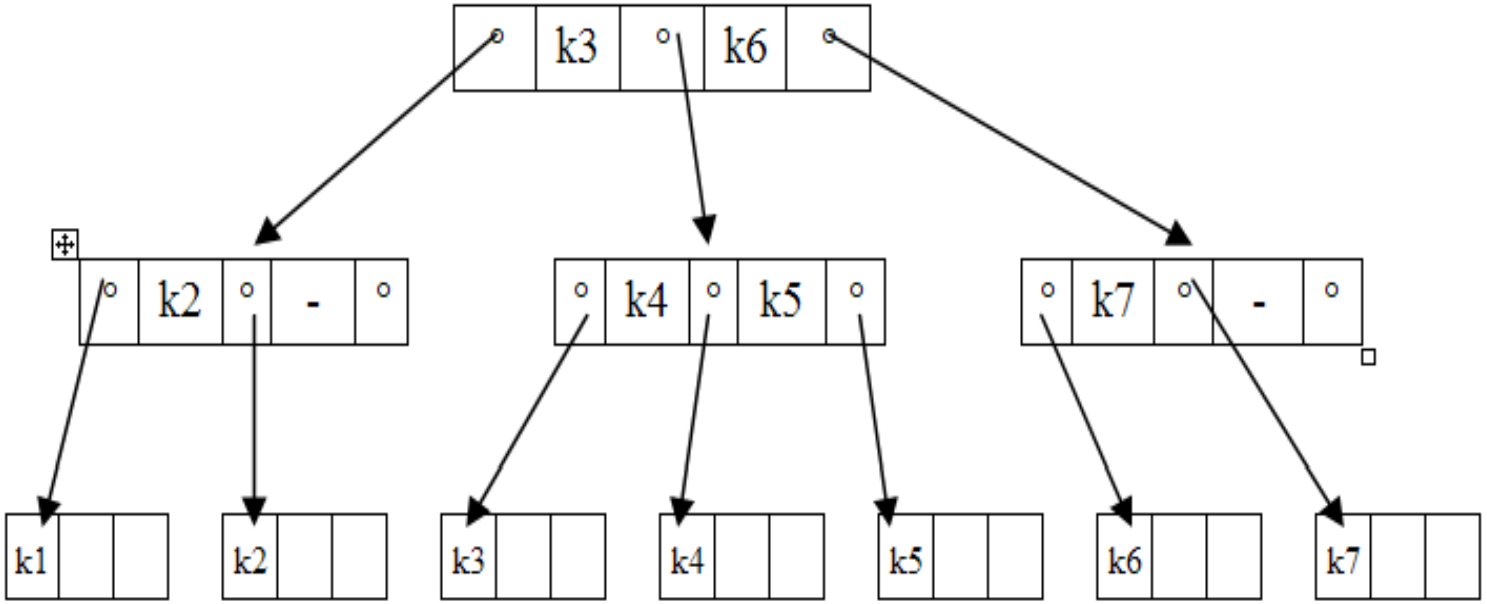
Tổ chức tệp chỉ dẫn thưa

- Xóa một bản ghi: giống như thêm một bản ghi, nếu xóa mà tạo thành 1 khối rỗng, khi đó có thể loại bỏ cả khối đó.
- Sửa một bản ghi:
  - Sử dụng thủ tục tìm kiếm để xác định bản ghi cần sửa
  - nếu các trường cần sửa không phải là khóa thì sửa bình thường
  - nếu các trường cần sửa tham gia vào khóa thì quá trình sửa sẽ là quá trình thêm và xóa 1 bản ghi.

Cây cân bằng(Balanced-trees)

- B-tree được tổ chức theo cấp m, có các tính chất sau đây:
  - Gốc của cây hoặc là một nút lá hoặc ít nhất có hai con.
  - Mỗi nút (trừ nút gốc và nút lá) có từ [m/2] đến m con.
  - Mỗi đường đi từ nút gốc đến bất kỳ nút lá nào đều có độ dài như nhau.

Cây cân bằng(Balanced-trees)



- Cấu trúc của mỗi nút trong B-cây có dạng $(p_0, k_1, p_1, k_2,...,k_n, p_n)$ với p_i (i=1..n) là con trỏ trỏ tới khối i của nút có k_i là khoá đầu tiên của khối đó. Các khoá k trong một nút được sắp xếp theo thứ tự tăng dần.

Cây cân bằng(Balanced-trees)

- Mọi khoá trong cây con, trỏ bởi con trỏ p_0 đều nhỏ hơn k_1;
- Mọi khoá trong cây con, trỏ bởi con trỏ p_i đều nhỏ hơn k_{i+1}.
- Mọi khoá trong cây con, trỏ bởi con trỏ p_n đều lớn hơn k_n.

Cây cân bằng(Balanced-trees)

- Các thao tác
  - Tìm kiếm một bản ghi: xác định đường dẫn từ nút gốc tới nút lá chứa bản ghi này
  - Thêm một bản ghi:
    - Xác định vị trí nút lá sẽ chứa bản ghi này (như tìm kiếm)
    - Nếu còn chỗ thì thêm bình thường
    - Nếu hết chỗ thì phải tạo thêm nút lá mới, chuyển nửa dữ liệu cuối của nút lá hiện tại sang nút mới, sau đó thêm bản ghi mới này vào vị trí phù hợp nút lá hiện tại hoặc nút mới tạo
    - Rất có khả năng "lan truyền" đến nút cha,….nút gốc.

Cây cân bằng(Balanced-trees)

- Loại bỏ 1 bản ghi:
  - Dùng thủ tục tìm kiếm một bản ghi để xác định nút L có thể chứa bản ghi đó.
  - Rất có khả năng "lan truyền" đến nút cha,….nút gốc.

Kết luận

- Tổ chức tệp chỉ dẫn:
  - được áp dụng phổ biến
  - Với các ứng dụng yêu cầu cả xử lý tuần tự và truy nhập trực tiếp đến các bản ghi
  - Hiệu năng sẽ giảm khi kích thước tệp tăng =>chỉ dẫn B-cây
- Tổ chức băm:
  - Dựa trên 1 hàm băm, cho phép tìm thấy địa chỉ khoản mục dữ liệu một cách trực tiếp
  - Hàm băm tốt? Phân bố các bản ghi đồng đều trong các cụm