Chương 4: Thiết kế mức vật lý

Nội dung

- Thiết kế CSDL
- Thiết kế các vùng tin (Field design)
- Thiết kế các bản ghi vật lý
- Thiế kế tệp tin vật lý
- Sử dụng và cách chọn chỉ mục

4.1. Thiết kế database

- **Yêu cầu:**
- Thận trọng trong thiết kế vì những quyết định được làm trong giai đoạn này sẽ ảnh hưởng đến khả năng truy xuất dữ liệu, thời gian đáp ứng, tính bảo mật, tính thân thiện với người dùng.
- **Phạm vi thiết kế:**
- Chỉ thiết kế database tập trung (centralized DB), không phân tán

Mục tiêu thiết kế database

- Tập trung vào tính hiệu quả xử lý dữ liệu (data processing efficiency).
- Chi phí máy tính ngày nay giảm đáng kể, việc thiết kế chỉ cần tập trung vào việc giảm nhỏ thời gian xử lý làm thế nào xử lý database và các file vật lý hiệu quả, không quan tâm nhiều đến không gian lưu trữ

Chuẩn bị trước khi thiết kế

- Cần thu thập thông tin liên quan đến hệ thống sẽ thiết kế:
  - Các quan hệ đã chuẩn hoá, kể cả việc ước lượng khối lượng thông tin
  - Các định nghĩa về các thuộc tính
  - Các mô tả về nơi nào và khi nào dữ liệu được dùng:
  - Thêm, truy xuất, xóa, cập nhật
- Các mong muốn và yêu cầu về thời gian đáp ứng, độ bảo mật dữ liệu, sao lưu phục hồi dữ liệu, tính toàn vẹn dữ liệu
- Mô tả về công nghệ, DBMS sẽ dùng để thực thi DB

Quá trình thiết kế database

- Chọn kiểu dữ liệu cho mỗi thuộc tính có mặt trong mô hình dữ liệu: kiểu dữ liệu ít tốn bộ nhớ mà vẫn bảo đảm tính toàn vẹn dữ liệu
- Nhóm các thuộc tính từ mô hình dữ liệu vào các bản ghi vật lý (physical record)
- Sắp xếp các bản ghi có cấu trúc tương tự vào bộ nhớ phụ (đĩa cứng) sao cho việc truy xuất các bản ghi này nhanh chóng.
- Cần quan tâm đến việc bảo vệ và khôi phục dữ liệu khi có lỗi
- Chọn cấu trúc lưu trữ và kết nối các file để việc truy xuất dữ liệu hiệu quả hơn.
- Tối ưu hóa xử lý các câu truy vấn.

Khối lượng dữ liệu & tần suất sử dụng (Data volume and usage frequency)

- Đánh giá khối lượng dữ liệu và tần số sử dụng dữ liệu là bước cuối của quá trình thiết kế CSDL hay là bước đầu tiên của quá trình thiết kế vật lý CSDL
- Để thống kê, thêm các ghi chú (notation) vào sơ đồ ER biểu diễn các quan hệ được chuẩn hóa cuối cùng

4.2 Thiết kế các vùng tin(Field design)

- Field là đơn vị nhỏ nhất của dữ liệu mà phần mềm hệ thống hay DBMS có thể nhận biết được.
- Field tương ứng với 1 thuộc tính (attribute) trong mô hình dữ liệu
- Quyết định cần làm khi thiết kế là phải chọn kiểu dữ liệu cho field, kiểm soát tính toàn vẹn dữ liệu và DBMS sẽ quản lý các giá trị bị thiếu cho field như thế nào??

4.2.1 Chọn kiểu dữ liệu

- 1. Tối thiểu hoá không gian lưu trữ
- 2. Diễn tả được tất cả các giá trị có thể có của dữ liệu
- 3. Cải thiện được tính toàn vẹn dữ liệu
- 4. Hỗ trợ được tất cả phép thay đổi dữ liệu

4.2.2 Kỹ thuật mã hoá và nén dữ liệu

- Một số thuộc tính có tập giá trị thưa hay có trị quá lớn chiếm nhiều không gian lưu trữ.
- Một trường có số ít giá trị nên mã hoá để chiếm ít không gian hơn.

Ví dụ

MaSV	Tên SV	MaKhoa
2021001	Nguyễn Văn A	CNTT
2021001	Nguyễn Văn B	DTVT

MaNV	NhómNV	TênNV
PU001	Quản lý	Ngô hồng sơn
PU002	Quản lý	…..

MaKhoa	TenKhoa	TruongKhoa
CNTT	Công nghệ thông tin	PU001
DTVT	Điện tử viễn thông	PU002

Kỹ thuật mã hoá và nén dữ liệu

- **Kỹ thuật nén tin** (data compression technique) tìm các mẫu (pattern) và mã hoá các mẫu xuất hiện thường xuyên với số bit ít hơn
- **Kỹ thuật mã hoá** (encryption technique): dùng để chuyển 1 trường sang dạng bảo mật
- **Kỹ thuật nén tin hay mã hoá** được dùng với 1 số DBMSs. Để người dùng đọc được giá trị thực sự của các trường, phần mềm cần phải biết quá trình dịch ngược lại

4.2.3 Kiểm soát tính toàn vẹn dữ liệu

- Việc kiểm tra tính toàn vẹn dữ liệu được xây dựng thành cấu trúc vật lý của các trường và được DBMS quản lý tự động.
- Kiểu dữ liệu là 1 dạng của tính toàn vẹn dữ liệu??
- Các kiểm tra toàn vẹn dữ liệu khác mà DBMS có thể hỗ trợ:
  - Default value
  - Range control
  - Null value control
  - Referential integrity

Giá trị mặc định (Default value)

- Là giá trị mà 1 trường luôn thừa nhận ngoại trừ người dùng đưa vào 1 giá trị tường minh khác để thay thế.
- Giảm thời gian nhập liệu
- Giảm những sai sót khi nhập liệu

Kiểm soát miền giá trị (Range control)

- Giới hạn 1 tập các giá trị cho phép mà 1 trường có thể nhận được.
- Miền giá trị có thể là 1 cận dưới và cận trên dạng số hay là 1 tập các giá trị cụ thể
  - Sự cố năm 2000
- Nên để DBMS thực hiện việc kiểm soát miền giá trị thay cho chương trình

Kiểm tra giá trị rỗng (Null value control)

- Một khoá chính thường bị cấm không được có giá trị null
- Các trường khác cũng có thể cần kiểm tra giá trị null tuỳ theo yêu cầu của tổ chức.
  - VD: Một trường đại học có thể cấm không chấp nhận bất kỳ course nào thiếu tiêu đề

Bảo toàn tham chiếu (Referential integrity)

- Là 1 dạng của kiểm tra miền trong đó giá trị của 1 trường có thể tồn tại như giá trị trường của 1 hàng nào đó trong cùng bảng hay của 1 bảng khác.

4.2.4 Xử lý dữ liệu bị thiếu (missing data)

- Dữ liệu bị thiếu khi không có dữ liệu để nhập vào 1 trường và trường cho phép có giá trị null
- Để tránh giá trị bị thiếu:
  - Dùng giá trị default
  - Không cho phép giá trị bị thiếu khi nhập liệu
  - Thay trị bị thiếu bằng 1 giá trị phỏng đoán
  - Theo dõi những giá trị bị thiếu, tổng kết thành báo cáo để buộc người dùng có liên quan đến phải nhanh chóng giải quyết các giá trị chưa biết.
  - Dùng phương pháp thử để xác định trị bị thiếu có ảnh hưởng đến kết quả tính toán hay không?

4.3 Thiết kế các bản ghi vật lý

- Bản ghi vật lý (physical record): là 1 nhóm các trường được lưu trữ trong những vị trí bộ nhớ cạnh nhau và được truy xuất như 1 đơn vị.
- Bản ghi logic (logical record) nhóm các thuộc tính vào cùng một quan hệ căn cứ vào việc các thuộc tính ấy phụ thuộc hàm vào cùng 1 khóa chính.
- Thiết kế bản ghi vật lý liên quan đến việc chọn sắp xếp các trường vào vị trí kề cận nhau sao cho đảm bảo 2 mục tiêu:
- – Sử dụng hiệu quả không gian lưu trữ
- – Tốc độ truy xuất dữ liệu

Sử dụng hiệu quả bộ nhớ phụ

- Hai yếu tố ảnh hưởng:
  - Kích thước của bản ghi vật lý
  - Cấu trúc của bộ nhớ phụ
- Hệ điều hành thường đọc/ghi dữ liệu từ đĩa cứng theo từng page, không theo bản ghi vật lý
- Page: là lượng dữ liệu được đọc/ghi vào bộ nhớ trong 1 thao tác xuất/nhập của bộ nhớ phụ.
- Kích thước trang do người thiết kế HĐH quyết định.
- Nếu chiều dài trang không chia hết cho kích cỡ của 1 bản ghi:
  - Sẽ có khoảng trống không dùng cuối mỗi trang
- Blocking factor: số bản ghi vật lý trên 1 trang

4.3.1 Trường dữ liệu

- Trường có độ dài cố định
  - Nếu các trường có chiều dài cố định, các trường sẽ đặt liền kề nhau, việc quản lý bộ nhớ sẽ dễ dàng hơn
- Trường có độ dài thay đổi
  - Vị trí của 1 trường thuộc 1 bản ghi nào đó thường không theo quy luật.
  - Cách chung để quản lý các trường độ dài thay đổi là chia quan hệ thành 1 bản ghi vật lý chứa toàn bộ các trường có chiều dài cố định và 1 hay nhiều bản ghi vật lý chứa các trường có chiều dài thay đổi

4.4 Thiết kế tệp tin vật lý

- Tệp tin vật lý (physical file) là một thành phần của bộ nhớ phụ (bộ nhớ ngoài) được cấp phát để lưu trữ các bản ghi vật lý
- Có 2 dạng:
  - Lưu trữ tuần tự (sequential storage)
  - Con trỏ (pointer)

4.4.1 Các phương pháp truy xuất (access method)

- Phương pháp truy xuất tương đối (relative access): truy xuất dựa vào khoảng cách (offset) giữa dữ liệu đó và dữ liệu vừa truy xuất trong bộ nhớ ngoài.
- Phương pháp truy xuất tuần tự là trường hợp đặc biệt của phương pháp truy xuất tương đối.
- Phương pháp truy xuất trực tiếp(direct access): dùng một cách tính để tính ra địa chỉ của bản ghi cần truy xuất.

4.4.2 Các kiểu tổ chức tệp tin

- Một kiểu tổ chức tệp tin (file organization) là một kỹ thuật sắp xếp về mặt vật lý các bản ghi của một tệp tin trên bộ nhớ ngoài.
- Đối với các HQTCSDL hiện đại ta không phải thiết kế tổ chức tệp tin, mà chỉ lựa một kiểu tổ chức và những thông số của cách tổ chức này đối với một tệp tin vật lý
- Các yếu tố liên quan đến việc lựa chọn:
- 1. Truy xuất dữ liệu nhanh
- 2. Xử lý việc nhập dữ liệu và các giao tác với hiệu năng cao.
- 3. Sử dựng chỗ bộ nhớ hữu hiệu
- 4. Bảo vệ để không bị hỏng hóc hay mất dữ liệu
- 5. Tối thiểu hóa việc tái tổ chức tệp tin
- 6. Thích ứng với yêu cầu phát triển mở rộng
- 7. Tránh khỏi những sự truy xuất dữ liệu không có thẩm quyền.

- **Tổ chức tệp tin tuần tự (sequential file organization)**
  - Các bản ghi được lưu trữ một cách tuần tự theo giá trị khóa chính.
  - Tìm kiếm: duyệt từ điểm khở đầu cho đến khi tìm thấy
- **Tổ chức tệp tin với chỉ mục (indexed file organization)**
  - Các bản ghi được lưu trữ một cách tuần tự hay không tuần tự và một chỉ mục (Index) được tạo ra cho phép phần mềm ứng dụng có thể định trị được các bản ghi riêng lẻ.
  - Chỉ mục là một bảng được dùng để định trị các hàng trong một tệp tin mà thỏa một điều kiện nào đó
  - Mỗi phần tử trong chỉ mục sẽ liên kết một khóa với một hay nhiều bản ghi

4.5 Sử dụng và cách chọn chỉ mục

- Tạo một chỉ mục khóa chính
- Tạo một chỉ mục thứ cấp
- Khi nào nên dùng chỉ mục:
  - CSDL cần nhiều thao tác truy xuất thì nên tạo nhiều chỉ mục
- CSDL cần nhiều thao tác cập nhật thì hạn chế chỉ mục

Chỉ mục - Indixes

- Mục đích: cải thiện việc truy tìm dữ liệu.
- Ý tưởng: tương tự như index của sách.
- Cho phép tìm nhanh 1 hàng mà không phải duyệt tuần tự từng hàng của bảng dữ liệu giảm thời gian thực thi truy vấn.
- Index chứa 1 tập hợp các index entry + cơ chế dò tìm entry dựa vào giá trị dò tìm (search key)
- Các cơ chế tìm kiếm
  - Các index entry được xếp theo search key
  - Heap file, index file
  - B-tree (balance tree)
  - Hash index

Chỉ mục - Indixes

- Cơ chế xếp theo search key: có 2 dạng
  - Các index entries được tích hợp (intergrated) vào cùng file dữ liệu
  - Index entries được lưu trữ vào 1 file khác.

- Tuy nhiên trong DBMSs
  - Đa số các DBMS đều tạo chỉ mục tự động cho các trường primary key bảng chỉ mục được tích hợp vào bảng dữ liệu. Các chỉ mục trên các trường khác được lưu vào bảng chỉ mục

Bất lợi của Indexs

- Chiếm không gian đĩa
- Nếu index lớn thì các trang index cần được đọc vào bộ nhớ gây ra chi phí cho thao tác vào ra
- Index cần được bảo trì (maintenance) các chỉ mục phải được sửa đổi cùng với sự thay đổi của dữ liệu

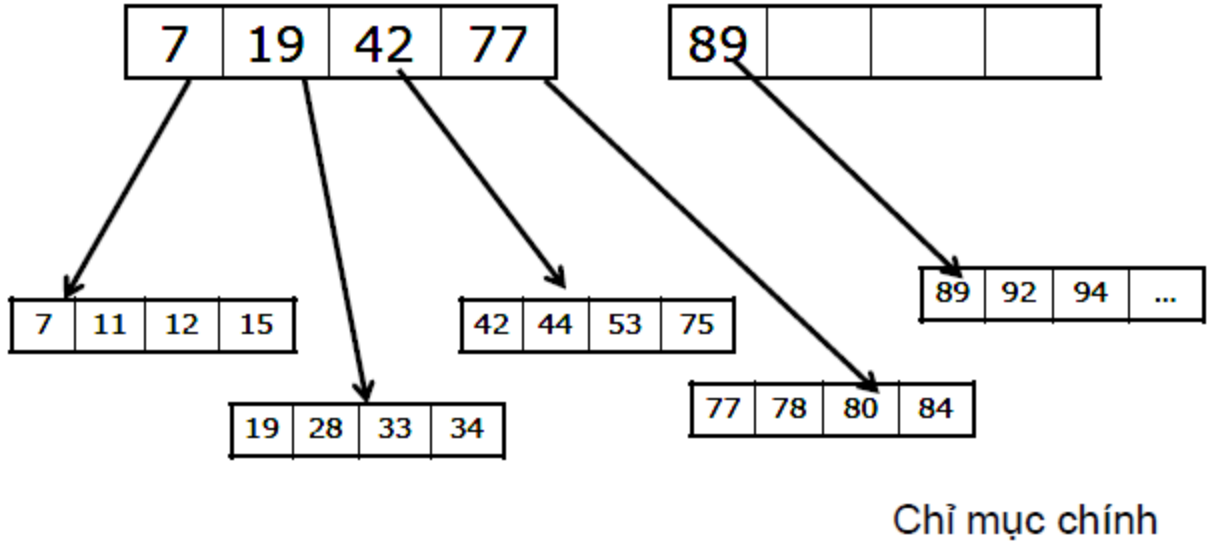
Phân loại chỉ mục

- Chỉ mục clustered còn được gọi là chỉ mục sơ cấp (primary index) hay main index
- Unclustered index thường được gọi là secondary index
- Thường thì với mỗi bảng chỉ có 1 clustered index và có thể có nhiều unclustered index
- **Chỉ mục thưa và dày (Sparse versus dense index)**
- Chỉ mục dày là chỉ mục mà các phần tử của nó tương ứng 1-1 với các bản ghi trong file dữ liệu
  - Chỉ mục dày thường là unclustered index
- Chỉ mục thưa trên 1 file đã sắp xếp là chỉ mục mà trong đó chỉ có 1 phần tử trong chỉ mục duy nhất cho mỗi trang dữ liệu và ngược lại.
  - Chỉ mục thưa là clustered index

Lựa chọn chỉ mục

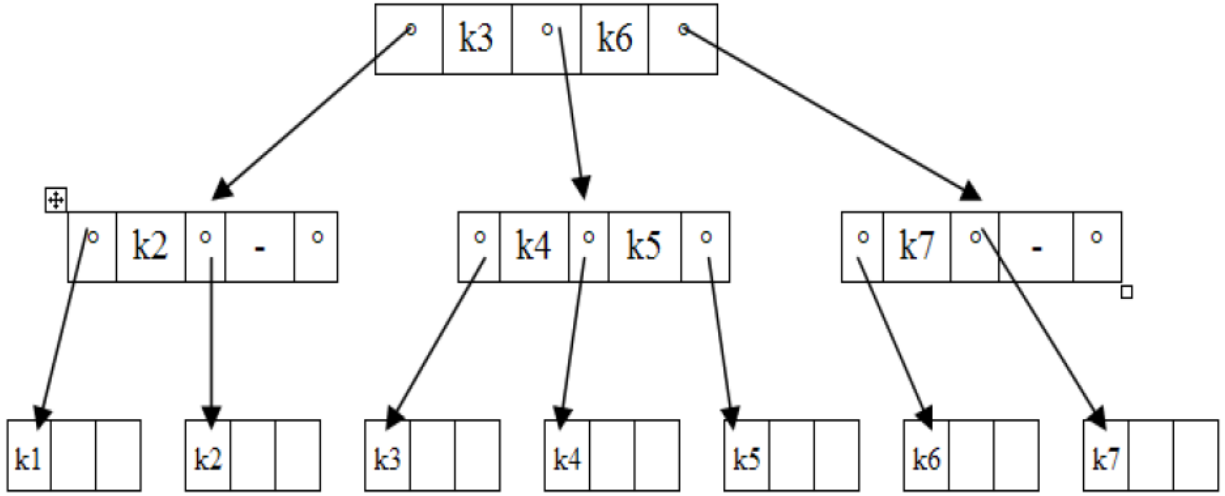
- 1. Chỉ mục rất có ích đối với những bảng lớn
- 2. Đặc tả chỉ mục là UNIQUE đối với chỉ mục là khóa chính
- 3. Chỉ mục có ích cho những cột xuất phát trong mệnh đề WHERE

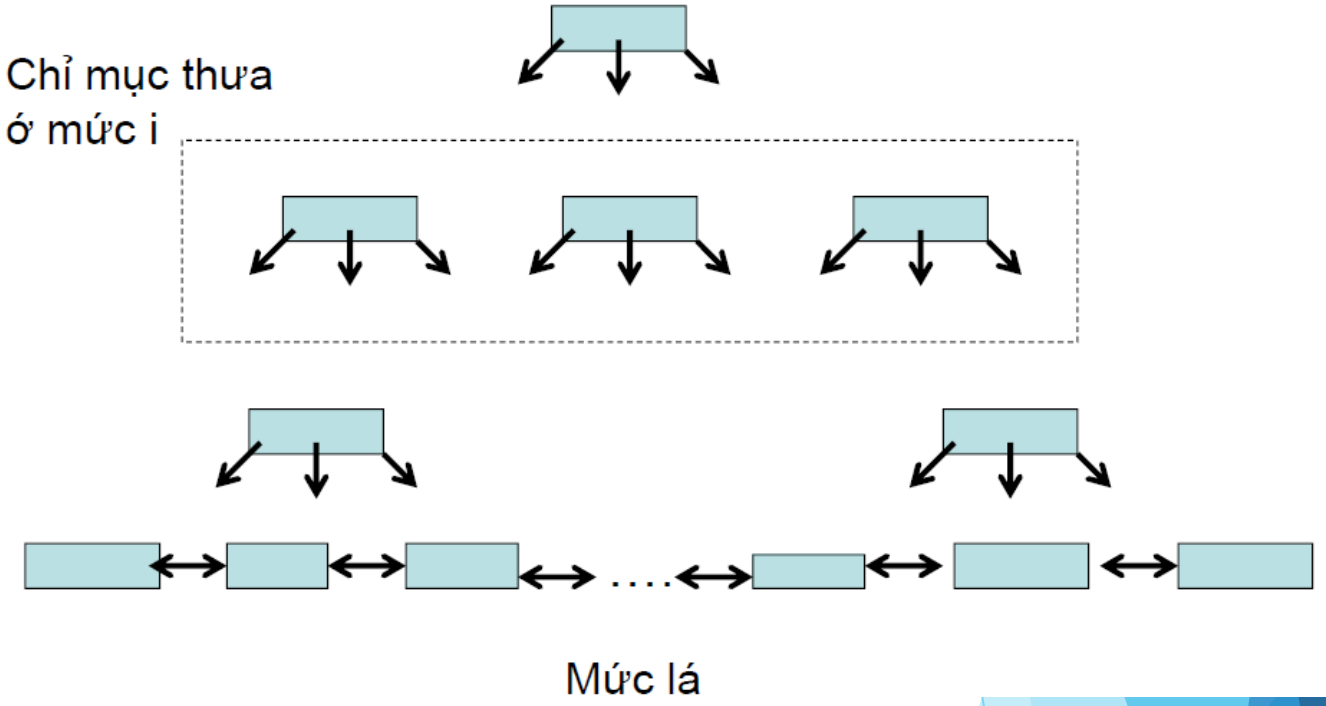
- Ví dụ cây chỉ mục 2 mức



B-Trees

- B-Trees là cấu trúc index thông dụng nhất.
- B-Trees cũng là chỉ mục nhiều mức và hỗ trợ việc tìm kiếm khóa các dạng miền, một phần khóa hay cả khóa.
- B-tree cân bằng 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.
- Có 2 dạng B-Trees:
  - Các lá chỉ mục có thể chứa các record dữ liệu B-Trees sẽ hoạt động không chỉ như 1 index mà còn như 1 cấu trúc lưu trữ của file dữ liệu B-Trees là chỉ mục chính (main index)
  - Cây được lưu trữ trong file chỉ mục mà các lá của nó chỉ đến các record của file dữ liệu có thể là chỉ mục chính (main index) hoặc thứ cấp (secondary index).


k3
k6
k2
-
k4
k5
k7
-
k1
k2
k3
k4
k5
k6
k7


Chỉ mục thưa
ở mức i
....
Mức lá

Cấu trúc của B-tree

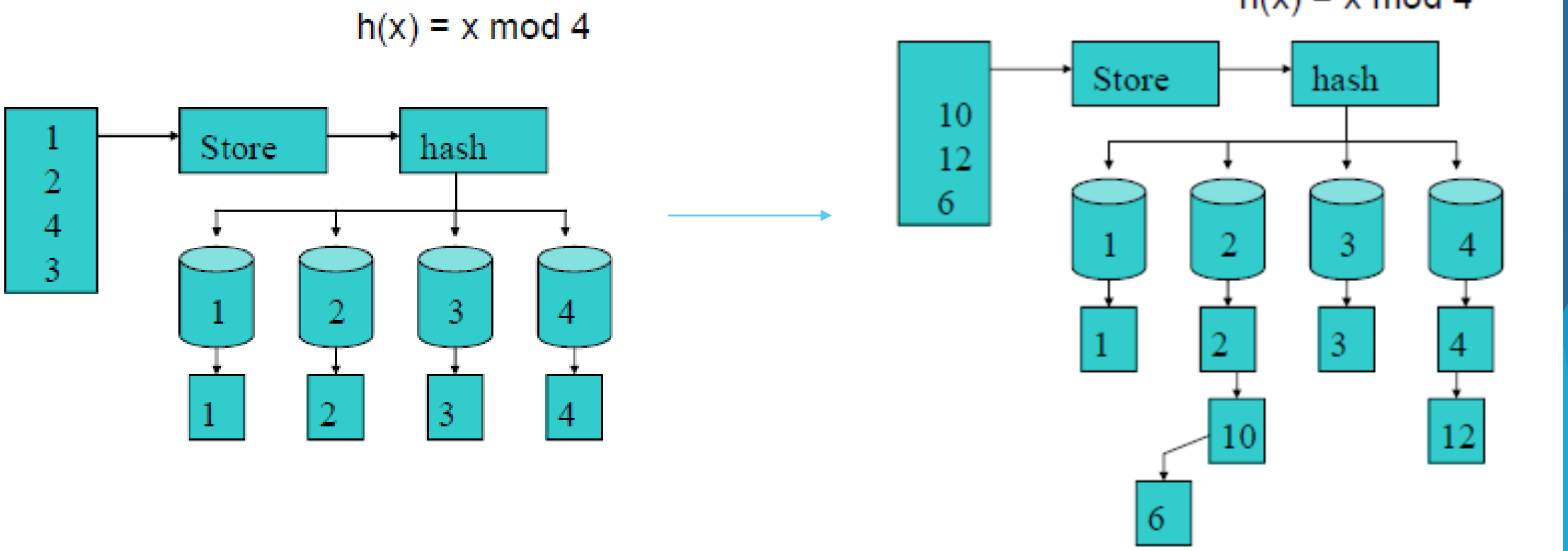
- Cấu trúc của mỗi nút trong B-tree (p_0, k_1, p_1, k_2,...,k_n, p_n)
- p_i (i=l..n) là con trỏ trỏ tới khối i của nút có k_i là khoá đầu tiên của khối đó.
- Các khoá k trong một nút được sắp xếp theo thứ tự tăng dần.
- Mọi khoá trong cây con, trỏ bởi p_i đều nhỏ hơn k_i+1
- Mọi khoá trong cây con, trỏ bởi p_n đều lớn hơn k_n.

Các phép toán

- Tìm kiếm 1 bản ghi
- Thêm 1 bản ghi
- Xoá 1 bản ghi
- Sửa đổi một bản ghi

Chỉ mục hash - Hash index

- Hashing là 1 giải thuật tìm kiếm quan trọng trong nhiều ứng dụng máy tính.
- Mục đích
  - Sử dụng chỉ số để hạn chế số lượng phép truy xuất đĩa bằng các phân nhóm các bản ghi (giả thiết n nhóm)
  - *Mapping* giá trị khoá với vị trí của (nhóm) bản ghi tương ứng
- Dựa trên bảng băm (*hash table*)
  - Hàm băm (*hash function*)
  - Cụm (*bucket*)
- Có 2 loại hashing:
  - Static hashing: kích thước của bảng hash không thay đổi, thường được dùng cho các quan hệ ít thay đổi
  - Dynamic hashing: kích thước của bảng hash có thể mở rộng ra hay thu nhỏ lại, thích hợp cho các quan hệ thường xuyên phải thêm mới hay xóa bớt bản ghi


h(x) = x mod 4
1
2
4
3
Store
hash
1
2
3
4
1
2
3
4
h(x) = x mod 4
10
12
6
Store
hash
1
2
3
4
1
2
3
4
10
12
6

Các phép toán

- Tìm kiếm 1 bản ghi
- Thêm 1 bản ghi
- Xoá 1 bản ghi
- Sửa đổi một bản ghi

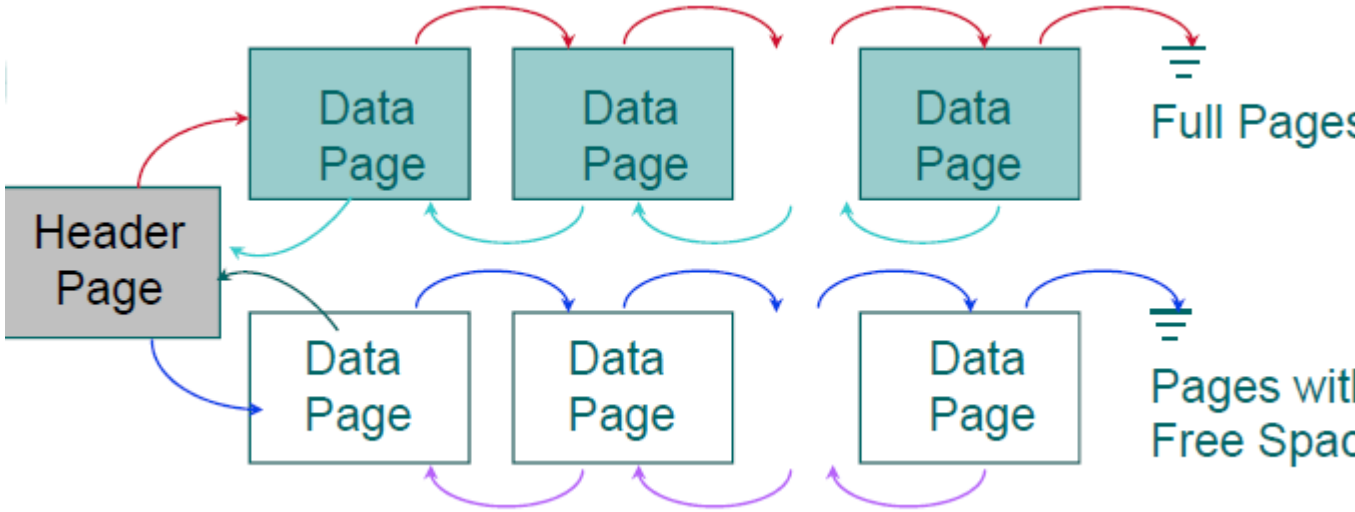
Tiêu chí chọn hàm băm

- Phân bố các bản ghi tương đối đồng đều (theo các cụm)
- Hạn chế việc sử dụng nhiều trang bộ nhớ cho 1 cụm

Tổ chức tệp đống (Heap File)

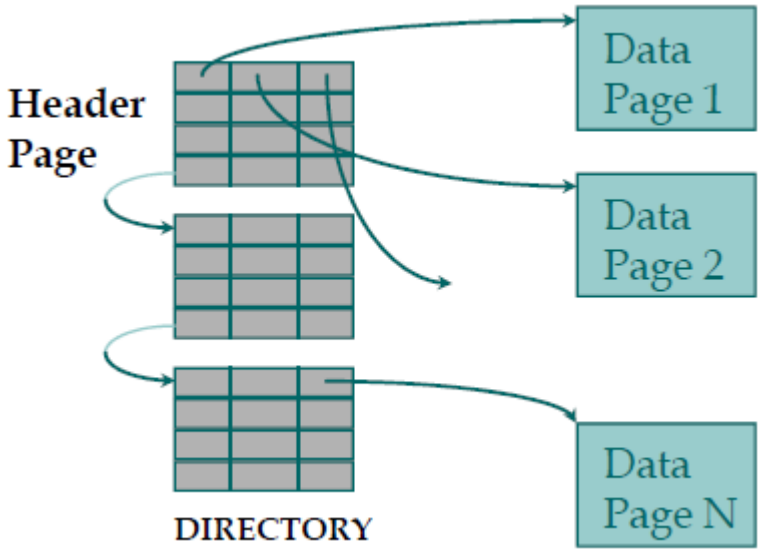
- Lưu trữ kế tiếp các bản ghi trong các trang không tuân theo một thứ tự đặc biệt nào
- Để thực hiện các phép toán, cần:
  - Ghi nhớ số trang trong 1 tệp
  - Ghi nhớ không gian trống trên các trang
  - Ghi nhớ các bản ghi trên các trang
- Có các con trỏ trỏ tới tất cả các trang của tệp và các con trỏ này được lưu trữ ở bộ nhớ trong.

- Cài đặt tệp đống bằng danh sách
  - Cần lưu trữ HeaderPage và tên của tệp
  - Mỗi trang gồm dữ liệu và 2 con trỏ
- Các phép toán
  - Tìm kiếm 1 bản ghi
  - Thêm 1 bản ghi
  - Xoá 1 bản ghi
  - Sửa đổi một bản ghi



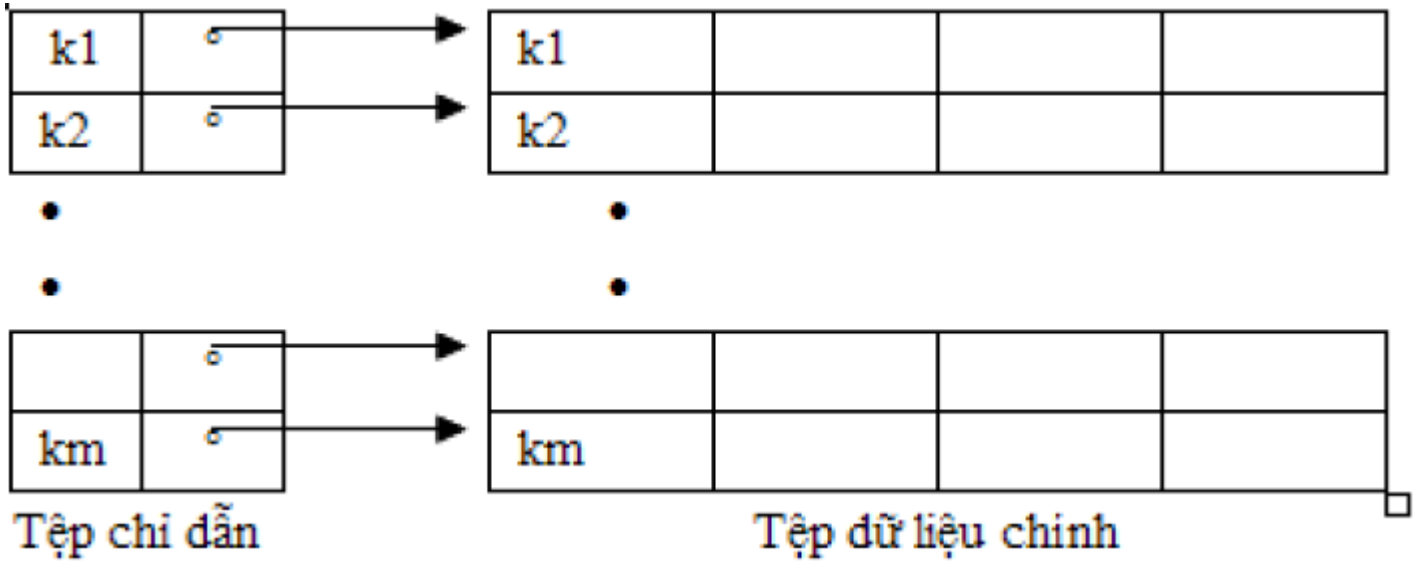
Ví dụ:

- Sử dụng trang danh bạ
  - Lưu thông tin về số byte còn trống trên trang đó
  - Danh bạ là 1 tập các trang



Tổ chức tệp chỉ dẫn (*Index File*)

- Tệp chỉ dẫn theo khoá được chọn trong bản ghi
- Tệp chỉ dẫn bao gồm các cặp (k,d), trong đó k là giá trị của khoá của bản ghi đầu tiên, d là địa chỉ của khối (hay con trỏ khối).
- Tệp chỉ dẫn được sắp xếp theo giá trị của khoá.



Các phép toán

- Tìm kiếm 1 bản ghi
- Thêm 1 bản ghi
- Xoá 1 bản ghi
- Sửa đổi một bản ghi

Tìm kiếm 1 bản ghi

- Tìm kiếm tuần tự
  - Duyệt tệp chỉ dẫn từ bản ghi đầu tiên đến khi tìm thấybản ghi có khoá k cần tìm
  - Nhận xét
    - Chậm đối với các tệp chỉ dẫn nói chung.
    - Thích hợp với các tệp chỉ dẫn nhỏ đủ để lưu ở bộ nhớ trong
- Tìm kiếm nhị phân
  - Chia đôi tệp chỉ dẫn đã sắp xếp để hạn chế số bản ghi cần duyệt
  - Tại mỗi lần chia hạn chế được ½ số bản ghi cần xem xét

Kết luận

- Truy cập đến CSDL thường liên quan đến một phần n nhỏ các bản ghi trong một tệp dữ liệu hay một vài trường (đặc biệt là các trường khoá) của các bản ghi dữ liệu.
  - Xác định các yêu cầu này cho phép thiết kế dữ liệu vật lý hiệu quả thông qua việc sử dụng các tổ chức lưu trữ đặc biệt
- Tệp chỉ dẫn được tạo lập trên khoá tìm kiếm để tăng hiệu quả của lưu trữ dữ liệu
  - Hiệu quả của các cấu trúc chỉ dẫn khác nhau phụ thuộc vào điều kiện áp dụng chúng