HỆ QUẢN TRỊ CSDL

ĐẠI HỌC SƯ PHẠM TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

Phiên bản 2019

Nội dung học phần

Chương 1. TỔNG QUAN

Chương 2. TỔ CHỨC LƯU TRỮ

Chương 3. TỐI ƯU TRUY VẤN

Chương 4. LẬP TRÌNH VỚI CURSORS

Chương 5. XỬ LÝ TRUY XUẤT ĐỒNG THỜI

HỆ QUẢN TRỊ CSDL

CHƯƠNG 2: TỔ CHỨC LƯU TRỮ

Chương 2. TỔ CHỨC LƯU TRỮ

1. Tổ chức CSDL

2. Chỉ mục (Index)

3. Triggers

1. Tổ chức CSDL

1.1 Tổ chức tập tin

1.2 Tổ chức mẩu tin

1.3 Tổ chức tập tin gồm các mẩu tin không sắp thứ tự (Heap File)

1.4 Tổ chức tập tin gồm các mẩu tin có sắp thứ tự (Sorted File)

1.5 SAN (Storage Area Network)

1.6 RAID (Redundant Arrays of Independent Disks)

- Những dạng lưu trữ:
  - Primary storage
  - Secondary storage

❑ Primary storage

- Là dạng lưu trữ mà CPU có thể thao tác trực tiếp được.
- Ví dụ:
  - *bộ nhớ chính của máy tính,*
  - *bộ nhớ được sử dụng cho cache*
- Tốc độ truy cập nhanh nhưng có giới hạn về khả năng lưu trữ, giá thành cao

❑ Secondary storage

Là dạng lưu trữ mà CPU không thể thao tác trực tiếp được (dữ liệu phải được chuyển vào primary storage)

- Ví dụ: *đĩa từ, đĩa quang, băng từ*
- Tốc độ truy cập chậm hơn so với primary storage
- Khả năng lưu trữ cao hơn, giá thành thấp hơn

Các dạng tổ chức bộ nhớ - Primary storage

static RAM (Random Access Memory)

- Bộ nhớ truy cập ngẫu nhiên (thời gian để đọc/ghi các ô nhớ là như nhau)
- Bộ nhớ cho phép đọc ghi (các dữ liệu bị thay đổi hay đang sử dụng)
- Dữ liệu trên RAM sẽ bị mất khi mất điện.

Các dạng tổ chức bộ nhớ - Primary storage

Cache memory

- Chính là RAM nhưng lưu dữ liệu của những lần đọc trước đó
- Khi chương trình cần đọc dữ liệu thì có thể đọc trong cache => việc thực thi chương trình sẽ nhanh.

Các dạng tổ chức bộ nhớ - Primary storage

DRAM (dynamic RAM)

- Là vùng làm việc chính cho CPU (main memory)
- Lưu trữ các chương trình và dữ liệu
- Tốc độ truy cập chậm hon so với RAM nhưng giá thành lại rẻ hon

Các dạng tổ chức đĩa (Secondary storage)

Gồm các loại

- CD-ROM (Compact Disk Read Only)
- Đĩa quang (optical disk)
- Đĩa từ (magnetic disk)
- Băng từ (magnetic tape)

1.2 Tổ chức mẩu tin

- Người thiết kế, cài đặt và quản trị:
  - Phải nắm được các kỹ thuật tổ chức lưu trữ
  - Biết được ưu và khuyết điểm của các kỹ thuật này
- CSDL được tổ chức vật lý
  - Là các tập tin chứa các mẩu tin (files of records)
  - Mỗi mẩu tin được xem là một thực thể
  - Ví dụ: mỗi mẫu tin là một sinh viên, có các thuộc tính như mã số, họ tên, địa chỉ…

1.2 Tổ chức mẩu tin

- Mẩu tin và kiểu mẩu tin là gì ?
  - Mẩu tin là 1 thực thể và có các trường dữ liệu (field)
  - Mỗi trường đều có kiểu dữ liệu
  - Các kiểu dữ liệu cơ sở như chuỗi, số, ngày, luận lý
  - Các kiểu dữ liệu đặc biệt như hình ảnh, âm thanh, phim, …
  - Tập hợp tất cả các tên trường cùng với kiểu dữ liệu của nó được gọi là kiểu mẩu tin.

Ví dụ một mẩu tin

sinh viên Trần Sơn Hải	
Mã số	A001
Họ và Tên	Trần Sơn Hải
Ngày sinh	09/04/1981
Giới tính	Nam
Địa chỉ	46 Tân Hải
Số diện thoại	0903080808
Học bổng	8.400.000

1.2 Tổ chức mẩu tin

Ví dụ một kiểu mẩu tin

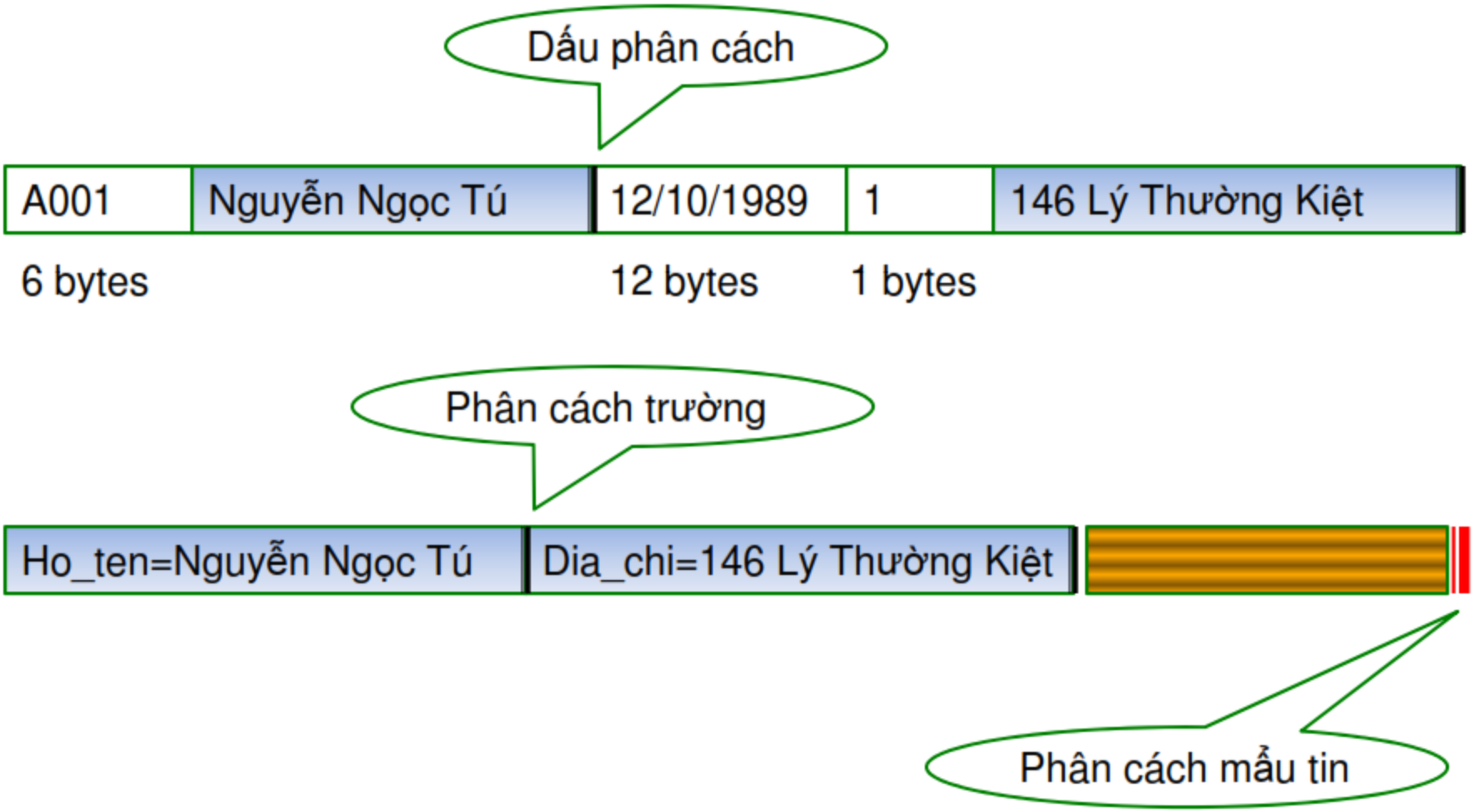
```
structure SINH_VIEN
{
    string  ma_so;
    string  ho_ten;
    date    ngay_sinh;
    boolean gioi_tinh;
    date    ngay_sinh;
    string  dia_chi;
    string  dien_thoai;
    real    hoc_bong;
}
```

- Trong một tập tin, nếu kích thước tất cả mẩu tin đều như nhau thì gọi là mẩu tin có chiều dài cố định.
- Ngược lại thì gọi là mẩu tin có chiều dài thay đổi

❑ Ví dụ mẩu tin có chiều dài cố định

A001	Nguyễn Ngọc Tú	12/10/1989	1	146 Lý Thường Kiệt
6 bytes	30 bytes	12 bytes	1 bytes	100 bytes

- Ví dụ mẩu tin có chiều dài thay dổi



LƯU TRỮ CÁC MẨU TIN TRONG CÁC KHỐI

❑ Khối (Block)

- Là đơn vị dữ liệu được sử dụng để chuyển đổi dữ liệu giữa đĩa và bộ nhớ. Ví dụ: khối có kích thuớc 8 Kbyte
- Các mẩu tin trong tập tin sẽ được luu trữ trong các khối của đĩa (disk block)
- Một khối có thể lưu trữ được nhiều mẩu tin (khi kích thuớc của khối lớn hơn kích thuớc của mẩu tin)

LƯU TRỮ CÁC MẨU TIN TRONG CÁC KHỐI CÓ PHÂN CHIA

- Các mẩu tin có thể được lưu trên nhiều khối
- Nếu B là kích thước khối (theo byte) và R là kích thước mẩu tin (chiều dài mẩu tin cố định). Và B>=R thì:
- Khối cho phép chứa đến B/R mẩu tin. Nếu B/R có phần dư khác 0 thì phải cấp thêm không gian sử dụng cho phần du này (cấp thêm khối mới)
- Nếu chiều dài mẩu tin thay đổi thì có thể lưu trữ phân chia hoặc không phân chia.

LƯU TRỮ CÁC MẨU TIN TRONG CÁC KHỐI CÓ PHÂN CHIA

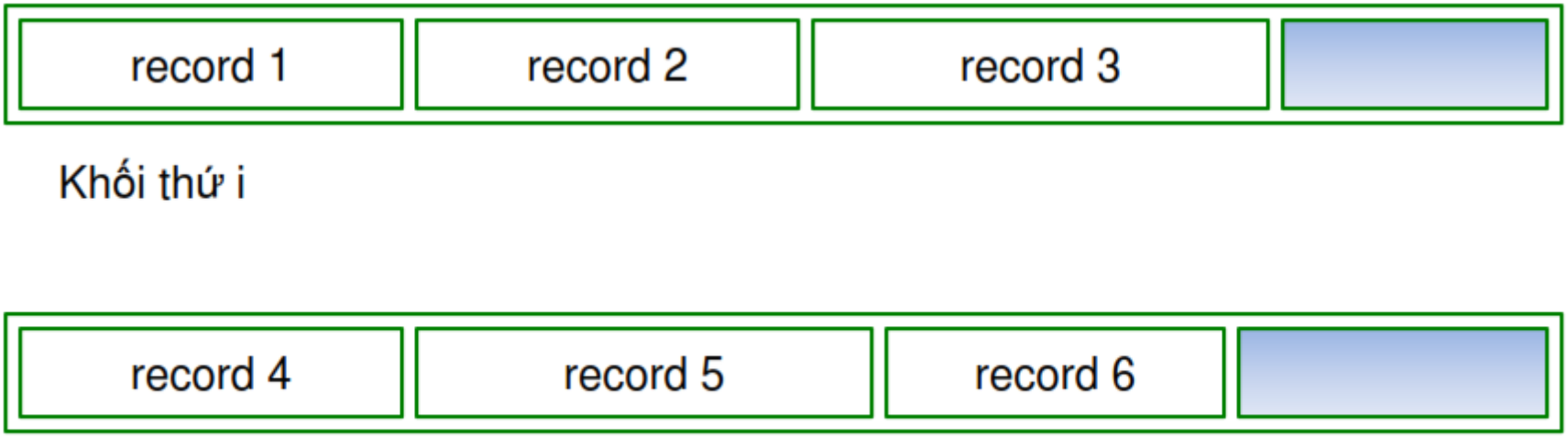
- Mẩu tin sẽ lưu trên một khối và phần còn lại được lưu trên một khối khác
- Cuối khối thứ nhất sẽ có một con trỏ (pointer) trỏ đến địa chỉ của khối tiếp theo
- Nếu kích thước của các mẩu tin lớn thì việc lưu trữ có phân chia sẽ tiết kiệm được các không gian sử dụng của khối.

LƯU TRỮ CÁC MẨU TIN TRONG CÁC KHỐI KHÔNG PHÂN CHIA

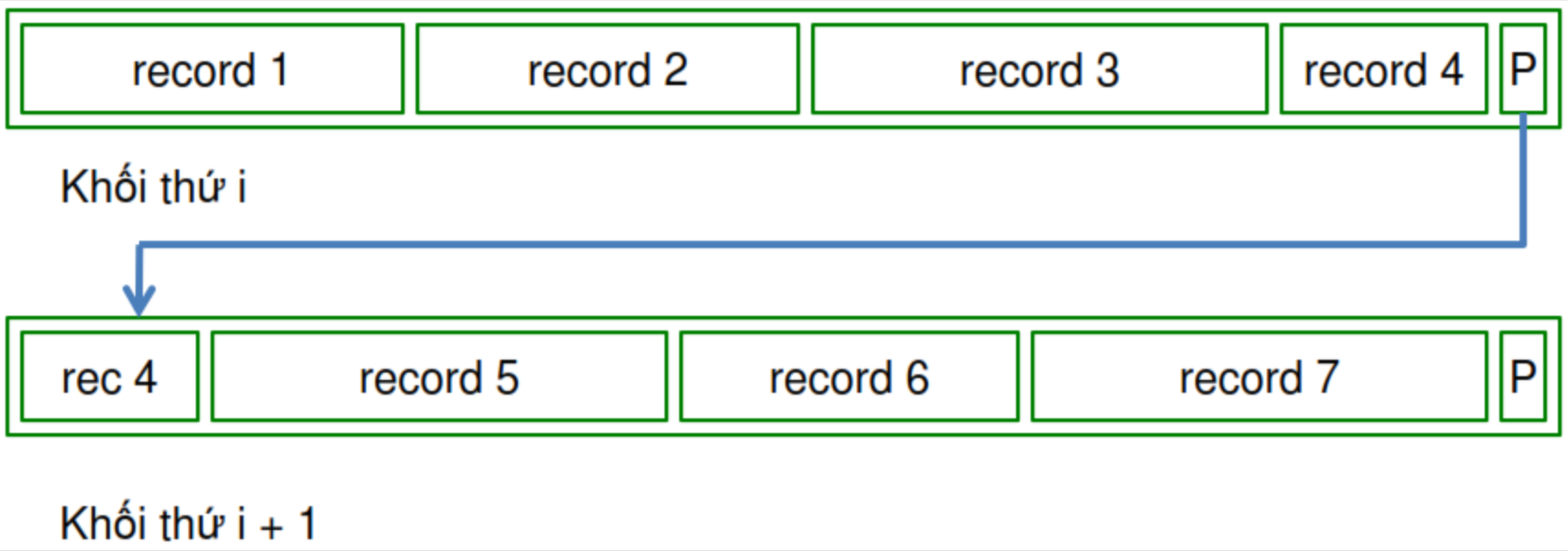
- Khi các mẩu tin không cho phép phân chia trên các khối
- Nếu chiều dài mẩu tin cố định và B>R thì các mẩu tin sẽ được lưu trong một khối và có một địa chỉ bắt đầu.
- Địa chỉ bắt dầu này được tính như sau:

Gọi B là kích thuớc của khối
Gọi R là kích thuớc của mẩu tin
Vậy N = B/R là số mẩu tin chứa trong khối
Nếu F là tổng số mẩu tin thì số khối cần M = F/N
Các khối dánh số thứ tự từ 0 dến M-1
Mẩu tin thứ I sẽ có dịa chỉ: I div N + I mod N

Ví dụ lưu trữ các mẩu tin trong khối không phân chia



Ví dụ lưu trữ các mẩu tin trong các khối có phân chia



❑ Các dạng cấp phát khối

Cấp phát liên tục

- Các khối trên đĩa sẽ được cấp phát liền kề nhau
- Ưu diểm: khi đọc toàn bộ tập tin sẽ đọc rất nhanh
- Khuyết điểm: gặp khó khăn trong việc tăng kích thước tập tin

❑ Các dạng cấp phát khối

Cấp phát liên kết

- Mỗi khối sẽ có một con trỏ, trỏ đến khối tiếp theo
- Ưu diểm: dễ dàng trong việc tăng trưởng kích thước tập tin
- Khuyết diểm: khi đọc toàn bộ tập tin sẽ chậm.

❑ Các dạng cấp phát khối

Cấp phát xâu (cluster)

- Là một xâu các khối, còn dược gọi là file segment hoặc file extend
- Các khối cấp phát có thể liền kề nhau hoặc liên kết nhau

Ví dụ: Tổ chức lưu trữ - SQL Server

- Bộ phận quản lý tập tin
  - .mdf : meta data file
  - .ldf : log data file
  - .bak : bakup data file
- Bộ phận quản lý đĩa
- Bộ phận quản lý dữ liệu vật lý

Ví dụ: Tổ Chức Vật Lý - SQL Server

- Dữ liệu trong CSDL được tổ chức thành các thành phần (Component) logic cho user sử dụng như: Table, View….
- Dữ liệu vật lý có thể lưu trên nhiều file hay nhiều ổ đĩa.
- Người dùng (trừ các DBA) chỉ làm việc trên các thành phần logic của SQL Server.

File header:

Là thông tin mô tả cho tập tin. Bao gồm:

- Địa chỉ các khối lưu trữ trên đĩa.
- Mô tả định các định dạng bản ghi như kích thước các trường, thứ tự các trường.

1.2 Tổ chức mẩu tin

- File header:

Khi tìm kiếm mẩu tin:

- Một hoặc nhiều khối được đọc vào bộ đệm của bộ nhớ chính
- Chương trình thực hiện việc tìm kiếm trên bộ nhớ chính
- Nếu mẩu tin vẫn chưa tìm thấy các khối tiếp tục được đọc vào bộ nhớ chính
- Việc tìm kiếm sẽ chấm dứt khi đã tìm thấy mẩu tin hoặc không tìm thấy mẩu tin nào trong tập tin
- Tập tin càng lớn thì thời gian tìm kiếm sẽ càng lâu

1.2 Tổ chức mẩu tin

- Một quan hệ là tập các bản ghi. Với một tập các bản ghi, vấn đề là làm sao để tổ chức chúng trong 1 file.
  - Tổ chức Heap File: Một bản ghi được lưu bất kỳ trong file. Không có thứ tự của các bản ghi. Thông thường một file lưu một quan hệ.
  - Tổ chức Sorted File: Các bản ghi được tổ chức tuần tự theo giá trị của một khóa tìm kiếm (search key) trong mỗi bản ghi.
  - Tổ chức tệp tin băm: Sử dụng hàm băm tính toán trên một số thuộc tính của bản ghi và dùng kết quả đó để xác định khối đĩa mà bản ghi được lưu trữ.

1.3 TC tập tin gồm các mẩu tin không sắp thứ tự

❑ (Heap File):

- Là kiểu tổ chức file đơn giản nhất
- Các bản ghi không được sắp xếp thứ tự.
- Việc thêm mới một bản ghi là đơn giản:
  - Tìm khối đĩa cuối cùng của file, sao chép khối đĩa vào bộ đệm, thêm mới bản ghi rồi ghi lại vào đĩa.
  - Địa chỉ của khối đĩa cuối cùng được cập nhật lại vào file header.

1.3 TC tập tin gồm các mẩu tin không sắp thứ tự

- (Heap File):
  - Sửa mẩu tin:
    - Xóa mẩu tin cũ và thêm mẩu tin mới.
  - Xóa mẩu tin:
    - Định vị khối chứa mẩu tin muốn xóa, đọc vào bộ đệm, xóa mẩu tin này trong khối, ghi khối trở lại đĩa.

    Nếu xóa nhiều mẩu tin thì có thể bị phân mảnh

=> giải pháp là chỉ đánh dấu xóa

1.3 TC tập tin gồm các mẩu tin không sắp thứ tự

❑ (Heap File)

- Do các bản ghi trong file không được sắp xếp theo thứ tự nên khi tìm kiếm phải sử dụng phương pháp tìm kiếm tuần tự: Đọc lần lượt từng khối đĩa vào bộ nhớ chính rồi tiến hành tìm kiếm các bản ghi.

↳ Như vậy nếu file gồm b khối đĩa thì thời gian tìm kiếm trung bình sẽ là n/2.

1.4 Tổ chức tập tin gồm các mẩu tin có sắp thứ tự

❑ (Sorted File)

- Các mẫu tin được sắp thứ tự theo giá trị của một hoặc nhiều trường.
- Nếu trường sắp thứ tự là trường khóa thì gọi là khóa sắp thứ tự (giá trị phải duy nhất)
- Tổ chức tập tin có sắp thứ tự còn gọi là tập tin tuần tự

1.4 Tổ chức tập tin gồm các mẩu tin có sắp thứ tự

❑ (Sorted File)

- Thuận lợi về tìm kiếm, có thể tìm mẩu tin nhanh bằng phương pháp tìm nhị phân.
- Khi thêm mẫu tin:
  - Phải xác định vị trí thích hợp cho mẩu tin mới.
  - Cách 1: khai báo dư vùng không gian trống
  - Cách 2: thêm mẩu tin mới vào tập tin phụ

1.4 Tổ chức tập tin gồm các mẩu tin có sắp thứ tự

❑ (Sorted File)

- Khi sửa mẫu tin:
  - Nếu sửa các giá trị của trường khóa, gồm hai thao tác: xóa mẩu tin cũ và thêm mẩu tin mới

  => hạn chế việc sửa các giá trị của trường khóa

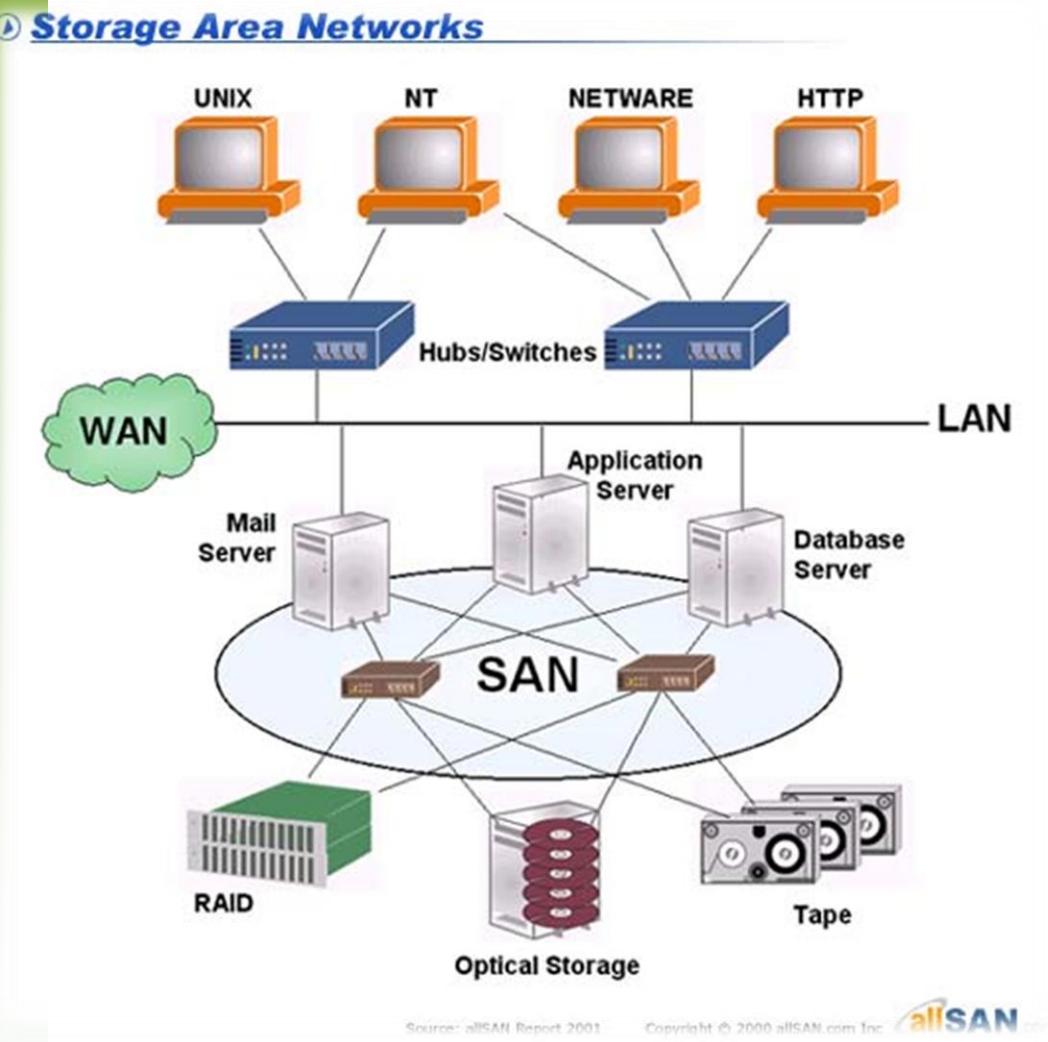
- Khi xóa mẫu tin:
  - Nếu xóa nhiều mẩu tin có thể dẫn đến việc phân mảnh

  => giải pháp là chỉ đánh dấu xóa

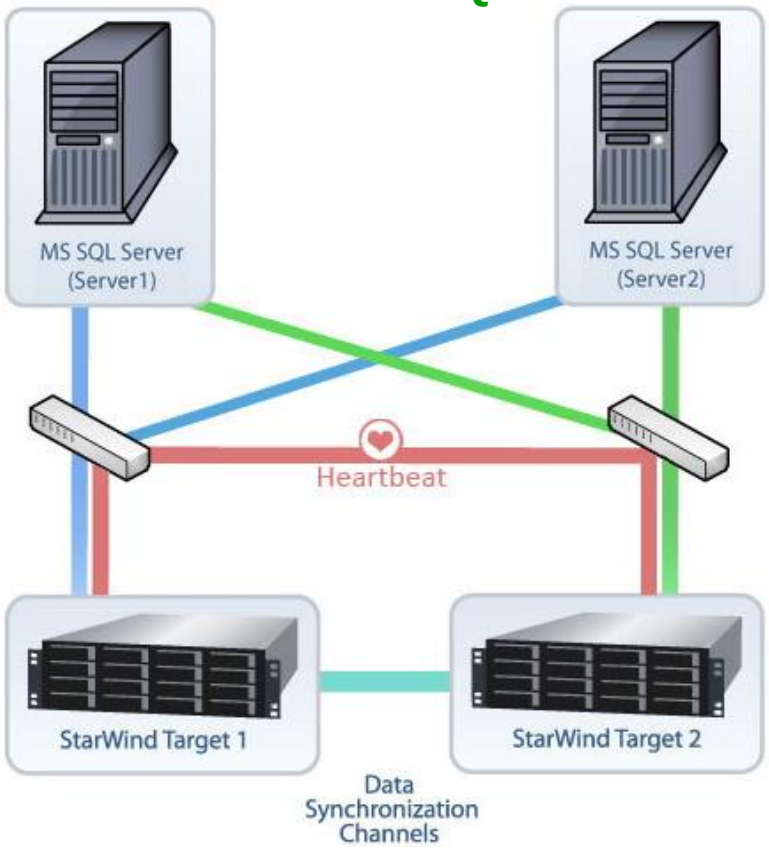
1.5 SAN (Storage Area Network)

- SANs Là một mạng được thiết kế cho việc thêm các thiết bị lưu trữ cho máy chủ một cách dễ dàng như: Disk Array Controllers, hay Tape Libraries.
- SANs được thiết kế dễ dàng cho tận dụng các tính năng lưu trữ, cho phép nhiều máy chủ cùng chia sẻ một thiết bị lưu trữ.

1.5 SAN (Storage Area Network)



Ví dụ:
StarWind iSCSI SAN Solution for Microsoft SQL Server



1.5 SAN (Storage Area Network)

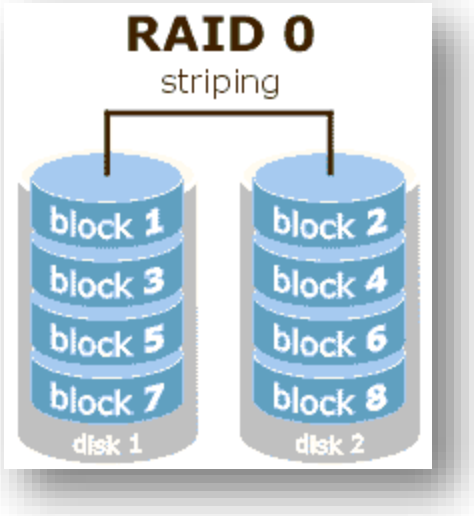
❑ Lợi ích khi sử dụng SANs

- Dễ dàng quản lý, chia sẻ, mở rộng khả thông qua quá trình thêm các thiết bị lưu trữ vào mạng mà không phải thay đổi máy chủ hay các thiết bị lưu trữ.
- SANs cho phép nhiều máy chủ cùng chia sẻ một thiết bị lưu trữ.
- SANs cho phép thay các máy chủ bị khi đang sử dụng không hề ảnh hưởng dữ liệu.
- SANs cung cấp giải pháp khôi phục dữ liệu một cách nhanh chóng.

1.6 RAID (Redundant Arrays of Independent Disks)

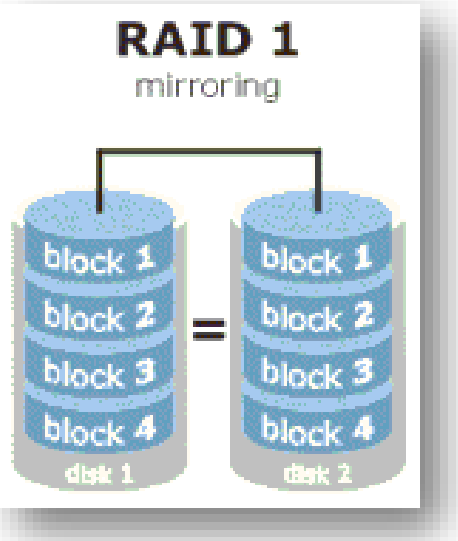
- Ban đầu, RAID được sử dụng như một giải pháp phòng hộ:
  - nó cho phép ghi dữ liệu lên nhiều đĩa cứng cùng lúc.
- Về sau, RAID đã có nhiều biến thể:
  - đảm bảo an toàn dữ liệu
  - giúp gia tăng đáng kể tốc độ truy xuất dữ liệu từ đĩa cứng.

1.6 RAID (Redundant Arrays of Independent Disks)



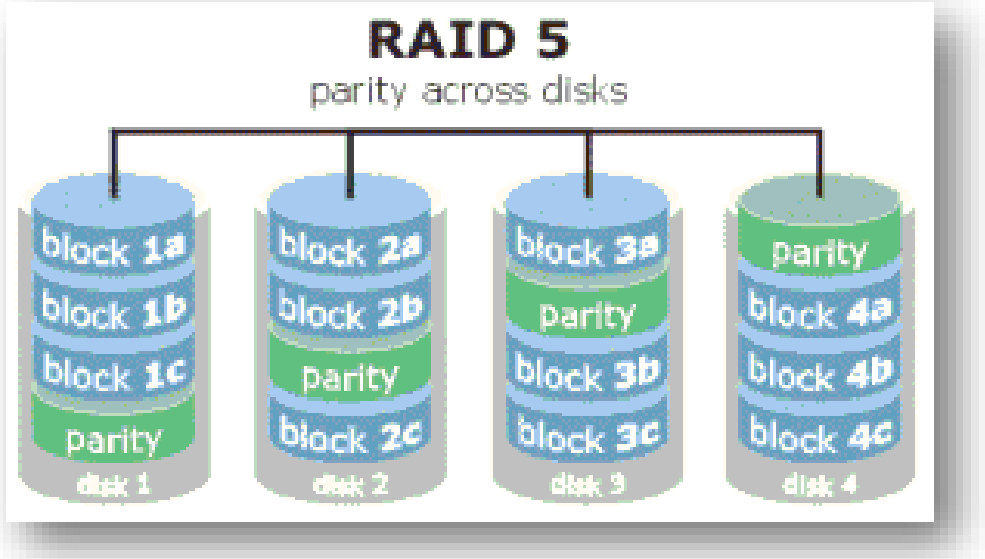
- Cần ít nhất hai ổ cứng.
- Dữ liệu được ghi thành nhiều phần trên nhiều ổ đĩa (Striping)
- Ưu điểm: tăng tốc độ đọc/ghi.
- Khuyết điểm: kém an toàn, nếu có một đĩa bị hư thì không phục hồi lại được dữ liệu.

1.6 RAID (Redundant Arrays of Independent Disks)



- Cần ít nhất 2 đĩa cứng
- Dữ liệu được ghi giống như nhau trên cả 2 đĩa (mirroring)
- Ưu điểm: độ an toàn cao, nếu có 1 đĩa bị hư thì dữ liệu vẫn tồn tại ở đĩa còn lại
- Khuyết điểm: hiệu năng tốc độ không phải là yếu tố hàng đầu

1.6 RAID (Redundant Arrays of Independent Disks)



- Cần ít nhất 3 ổ cứng trở lên
- Dữ liệu và bản sao lưu được chia lên tất cả các ổ cứng.
- Dữ liệu đươc ghi thành các phần trên các đĩa (striping) (-1 đĩa ghi parity)
- RAID 5 vừa đảm bảo cải thiện tốc độ, vừa giữ được độ an toàn.

2. Chỉ mục (Index)

- **Chỉ mục (index) là gì ?**
  - Tổ chức các cặp, gồm khóa và địa chỉ của dữ liệu.
  - Địa chỉ trỏ tới một khối đĩa hoặc một mẩu tin
- **Phân loại chỉ mục**
  - Chỉ mục một cấp (single level index) là chỉ mục của tập tin có sắp thứ tự.
  - Chỉ mục đa cấp (multi level index) là chỉ mục của dữ liệu có trúc cây

2. Chỉ mục (Index)

- **Tại sao tạo Index**
  - Tăng tốc độ truy xuất dữ liệu
  - Không bắt buộc tính liên tục trên các dòng
- **Khi nào không nên tạo Index?**
  - Tốn bộ nhớ trên đĩa để lưu trữ Index. Khi user cập nhật dữ liệu trên cột Index, SQL Server cũng cập nhật index
  - Việc quản lý Index sẽ tốn thời gian và tài nguyên nên nếu Index không thường sử dụng thì không cần tạo.

2. Chỉ mục (Index)

- **Primary/Secondary**
  - Ví dụ: Sắp tăng theo tên, cùng tên thì sắp theo tuổi

    tên là primary và tuổi là secondary.
- **Clustered/Non clustered**
  - Clustered: thứ tự các record lưu trữ vật lý sắp thứ tự của index
  - Non clustered: thứ tự các record lưu trữ vật lý không sắp thứ tự của index
- **Dense/sparse**
  - Dense = Đánh chỉ mục cho tất cả các records
  - Sparse = Chỉ đánh chỉ một số records

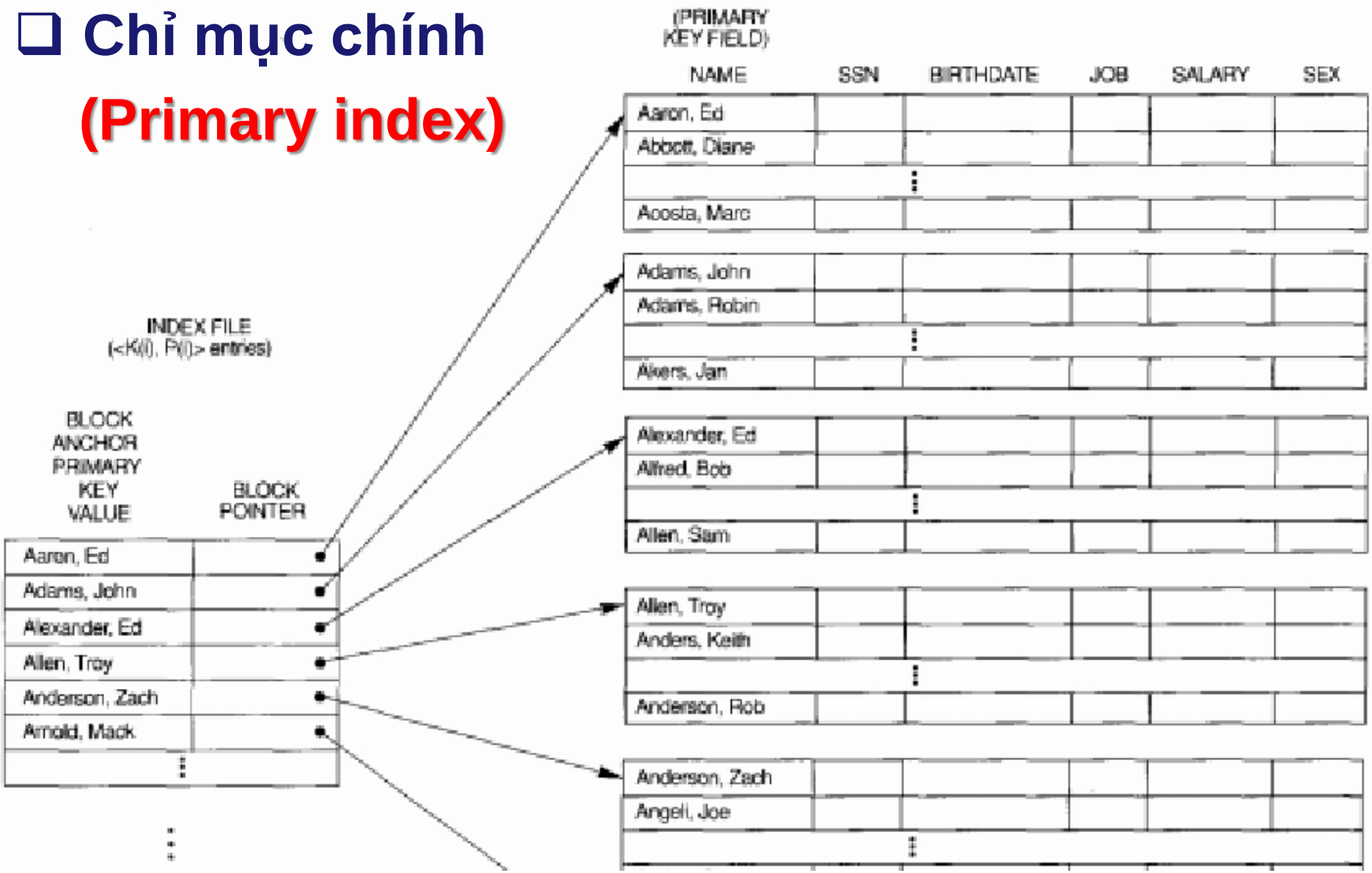
2. Chỉ mục (Index)

❑ Chỉ mục chính (Primary index)

- Khóa của chỉ mục là khóa của tập tin có sắp thứ tự, các giá trị khóa phải là duy nhất.
- File chỉ mục là một file có thứ tự gồm hai trường với các bản ghi có độ dài cố định.
- Một bản ghi trong file chỉ mục, gồm 2 phần **[Ki, Pi]** trong đó:
  - **Ki** là giá trị ứng với giá trị trên trường khóa đã sắp thứ tự
  - **Pi** chứa địa chỉ khối của bản ghi có khóa **Ki** trên file dữ liệu.
- Mỗi index entry tương ứng với mẩu tin đầu tiên của khối.

2. Chỉ mục (Index)

- **Chỉ mục chính**
  (Primary index)



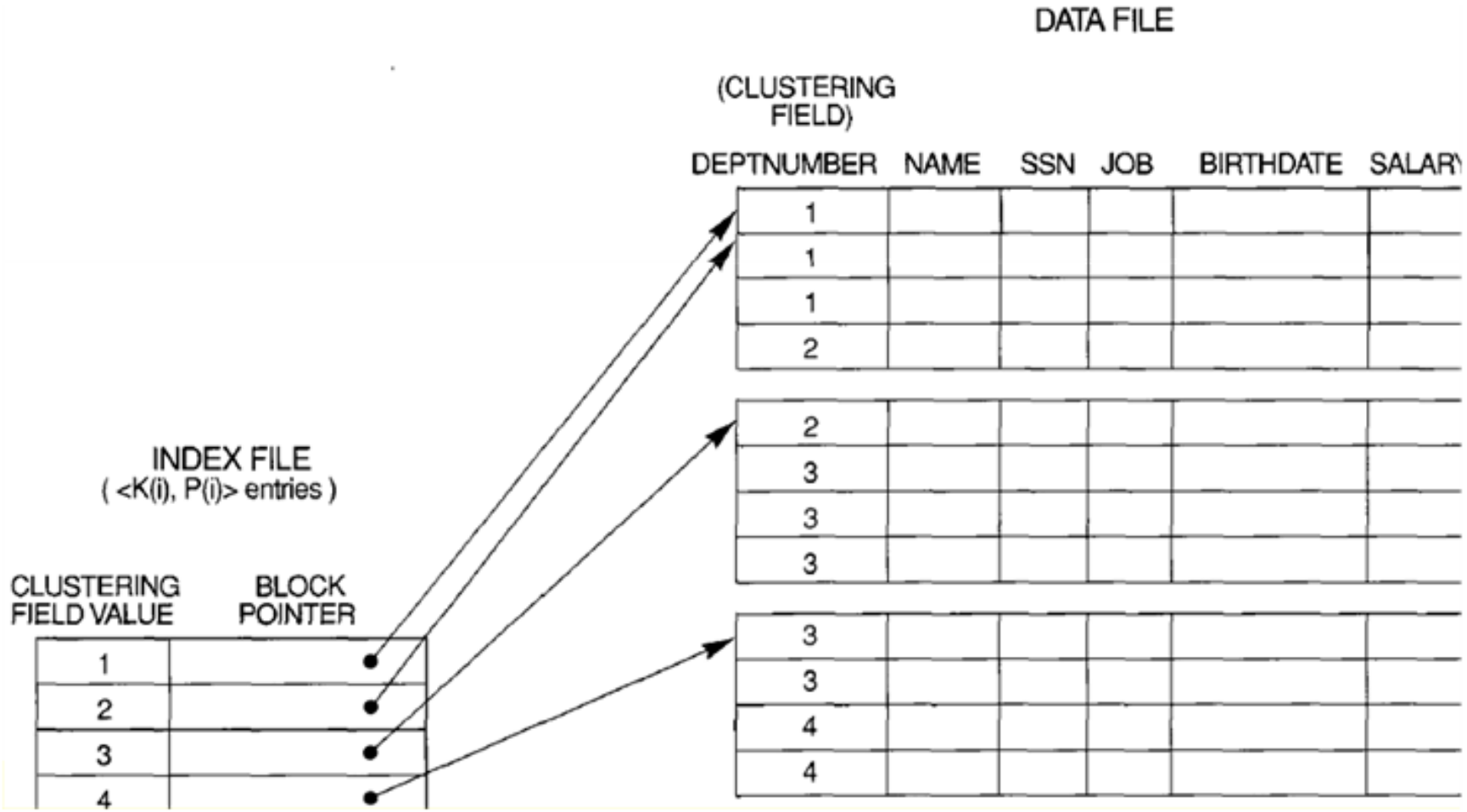
2. Chỉ mục (Index)

❑ Chỉ mục Clustered

- Khóa của chỉ mục không là khóa của tập tin có sắp thứ tự, các giá trị khóa có thể lặp lại.
- Trong Index entry:
  - phần tử thứ I là khóa (chỉ lấy giá trị đại diện)
  - phần tử thứ II là địa chỉ của khối đĩa.
- Mỗi index entry tương ứng với mẫu tin đầu tiên của khối.

2. Chỉ mục (Index)

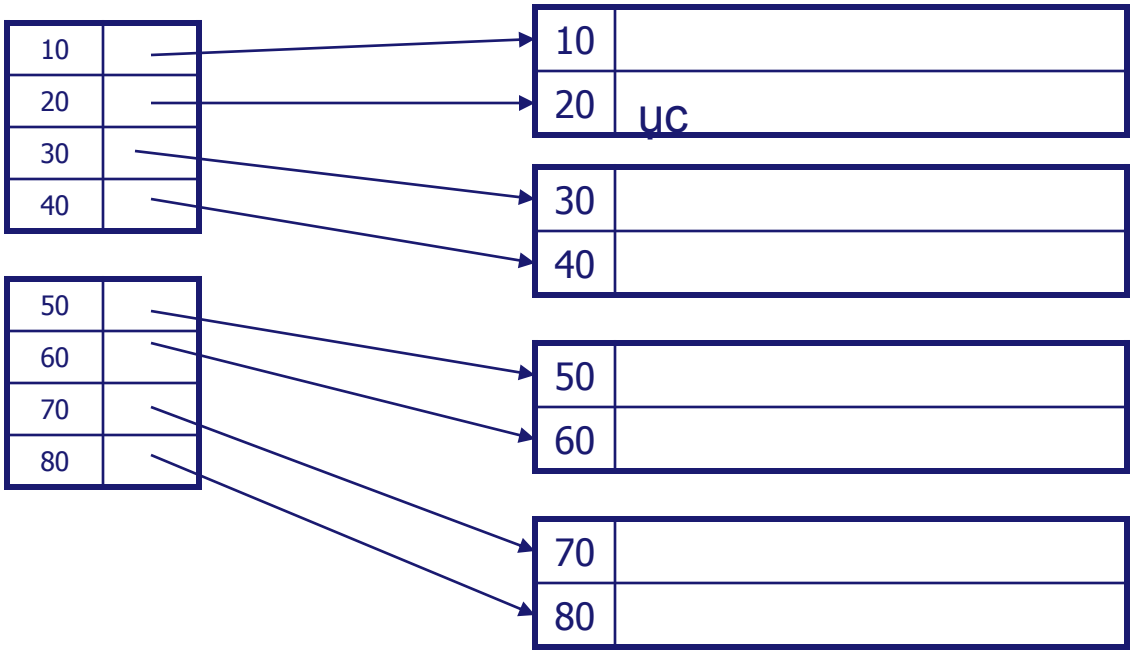
❑ Clustered Index



2. Chỉ mục (Index)

❑ Clustered Index trong SQL

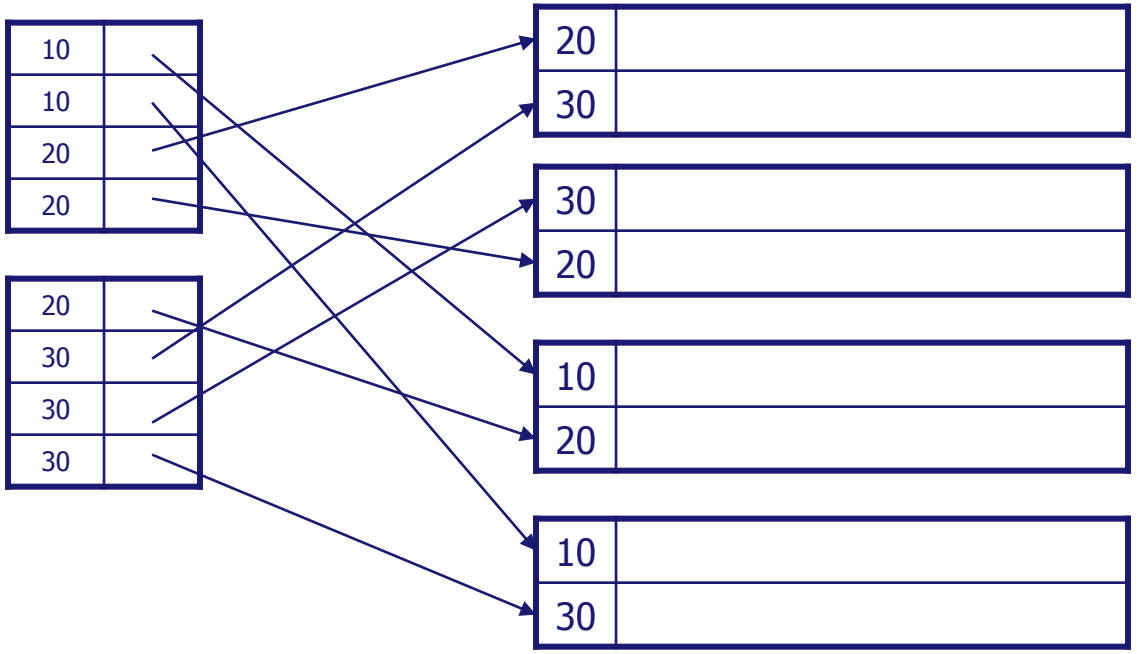
Dùng để sắp chỉ mục các thuộc tính là khóa chính



2. Chỉ mục (Index)

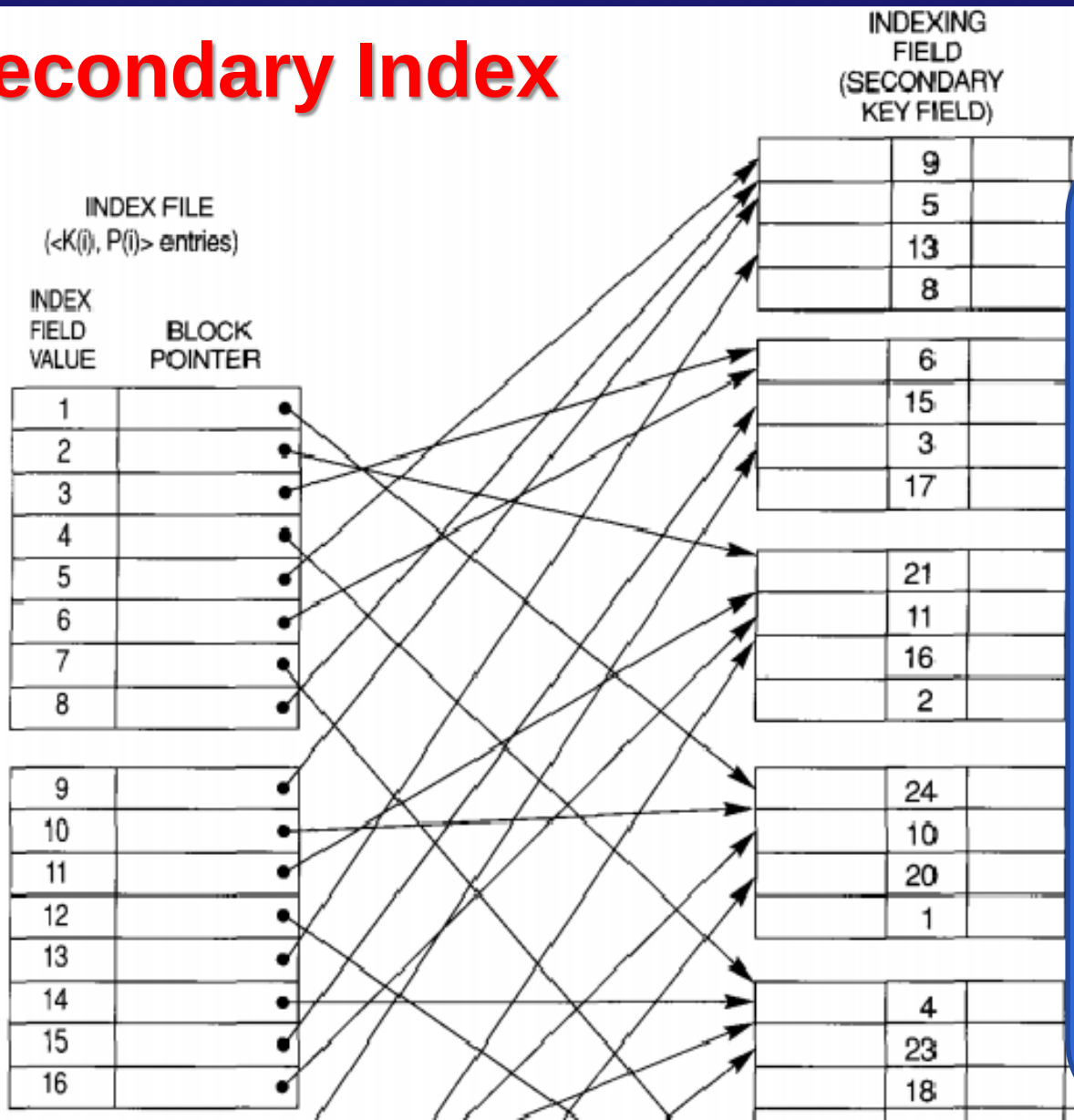
❑ Non clustered Index trong SQL

Dùng để sắp chỉ mục các thuộc tính không là khóa chính



2. Chỉ mục (Index)

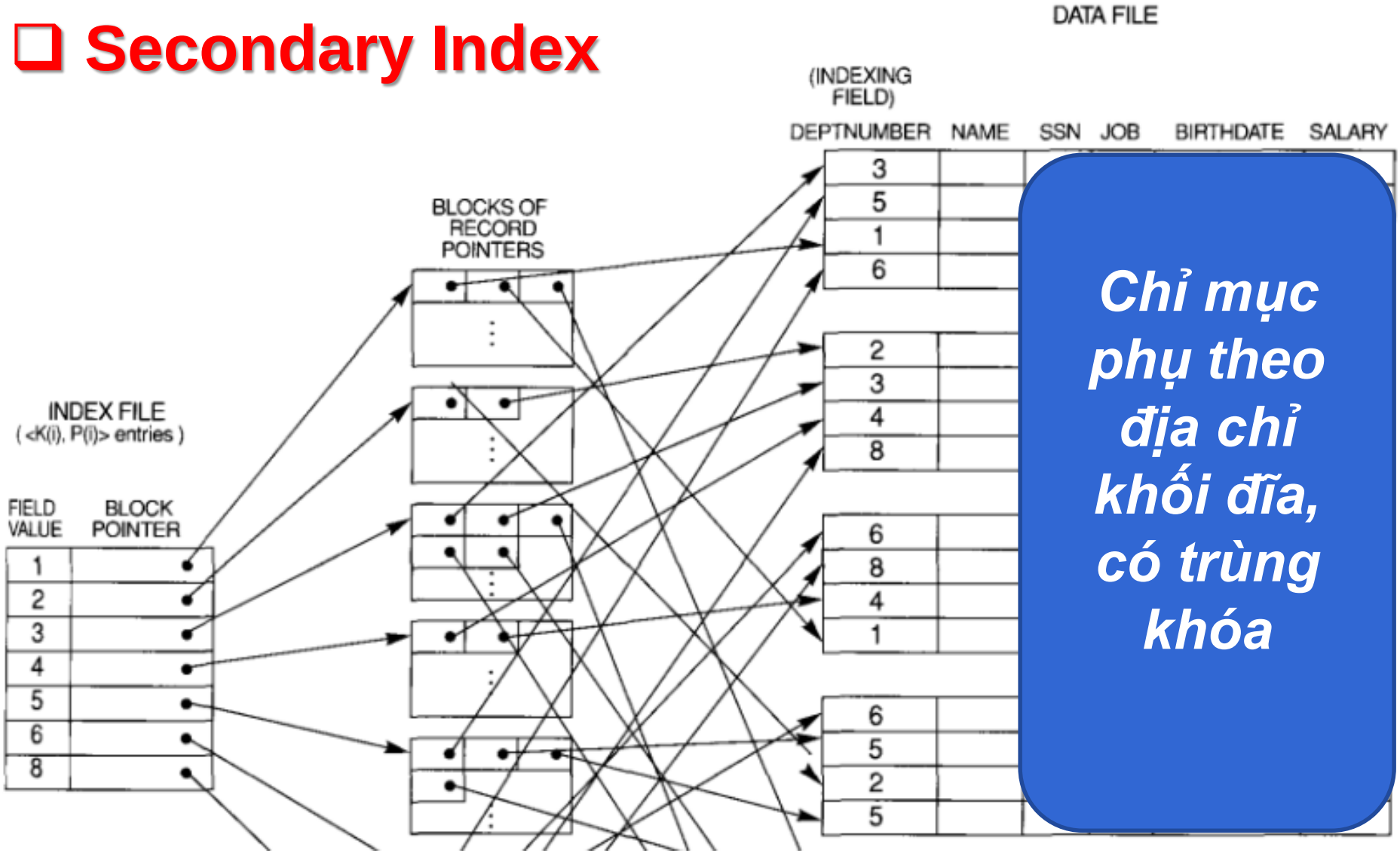
❑ Secondary Index



Chỉ mục phụ theo địa chỉ mẫu tin, không trùng khóa

2. Chỉ mục (Index)

❑ Secondary Index



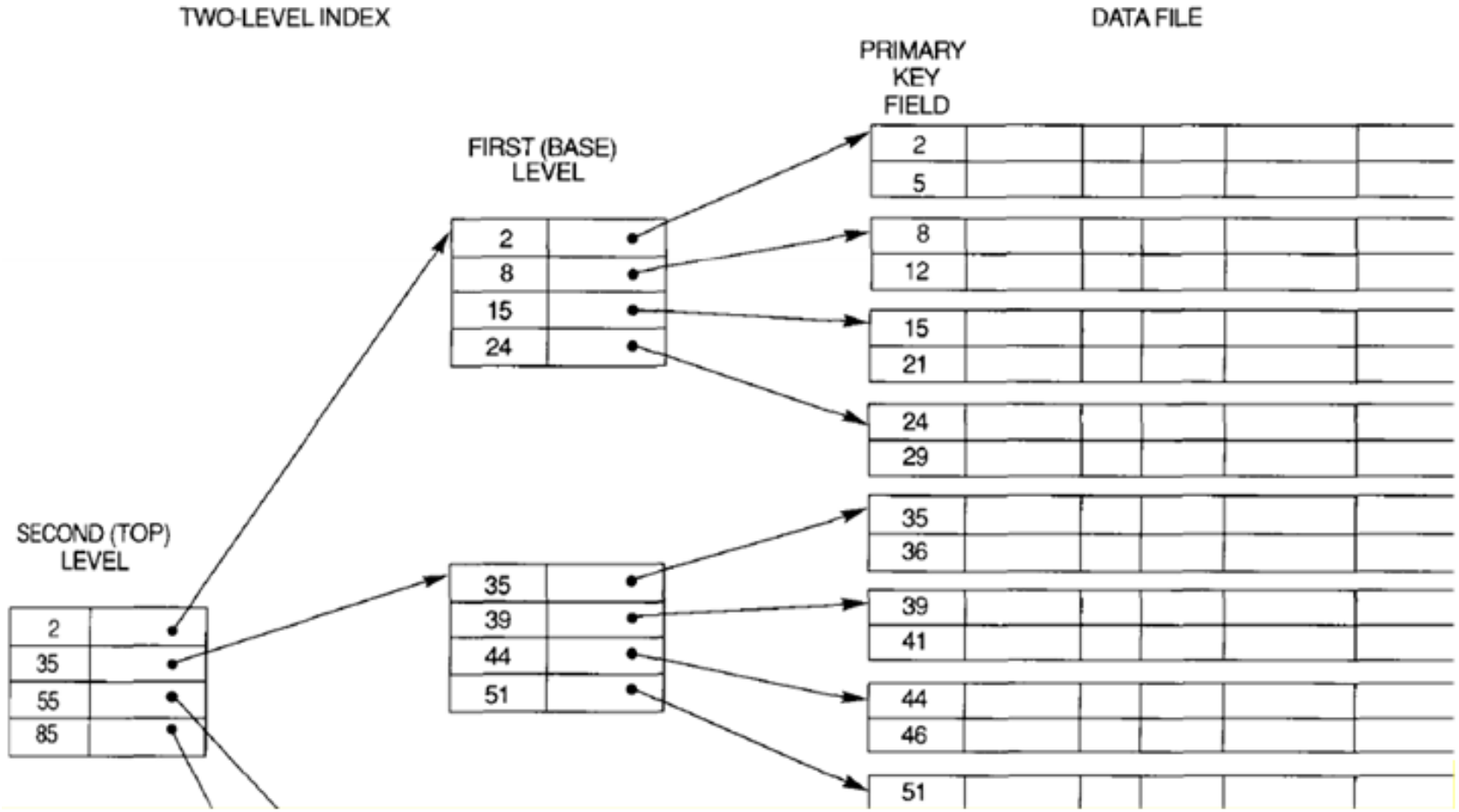
Chỉ mục phụ theo địa chỉ khối đĩa, có trùng khóa

2. Chỉ mục (Index)

- **Chỉ mục đa cấp (multi level index)**
  - Tổ chức đa cấp nhằm cải thiện thêm tốc độ tìm kiếm
  - Chỉ mục cấp một (cơ sở) là các index entry có các giá trị khóa duy nhất. Chỉ mục này xem như tập tin chỉ mục
  - Trên tập tin chỉ mục này tạo chỉ mục chính (Primary index): chỉ mục cấp hai
  - Chỉ mục cấp hai có các index entry là mẩu tin đầu tiên của khối đĩa

2. Chỉ mục (Index)

❑ Multi level index



Tạo Index trên SQL Server

- Trên SQL hỗ trợ 2 loại Index:
  - Cluster Index
  - Non Cluster Index

Tạo Index trên SQL Server

- **Cluster Index:** Chỉ có thể tạo một cluster index duy nhất cho một bảng dữ liệu.
  - Mặc định khóa chính sẽ thành cluster index
  - Dữ liệu của bảng sắp xếp theo thứ tự của cluster index

Tạo Index trên SQL Server

- **Non Cluster Index:** có thể tạo 249 non-cluster index cho một bảng dữ liệu.
  - Dữ liệu của bảng không sắp theo thứ tự của non-cluster index.
  - Thường tạo index cho các cột dữ liệu dùng để join hay trong điều kiện where hoặc giá trị cột này thường xuyên thay đổi.

Tạo Index trên SQL Server

- Cú pháp tạo Index:
  - CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name
    ON table_name
    (column_name[,column_name]…)

Tạo Index trên SQL Server

- Cú pháp tạo Index:
  - ***CREATE NONCLUSTERED INDEX***

    idxExternalCandidate

    ON ExternalCandidate(cAgencyCode)
  - ***CREATE CLUSTERED INDEX***

    idxRecruitment

    ON RecruitmentAgencies(cAgencyCode)

Bài tập tạo Index trên SQL Server

Giả sử CSDL của bạn có 1 bảng sau:

SinhVien(MaSV, TenSV, TuoiSV, DiaChi)

- Trong đó MaSV là khóa chính, thường dùng để join các bảng khác; tên (TenSV) thường xuất hiện trong điều kiện WHERE trong các câu truy vấn thông tin.
- Yêu cầu: Xác định Cluster và non cluster index cho bảng SinhVien. Viết câu lệnh SQL tạo bảng và tạo các câu Index tương ứng.

Tạo Index trên SQL Server

SinhVien(MaSV, TenSV, TuoiSV, DiaChi)

Trong đó MaSV là khóa chính, thường dùng để join các bảng khác; tên (TenSV) thường xuất hiện trong điều kiện WHERE trong các câu truy vấn thông tin.

- CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX

  index_name

  ON table_name (column_name[,column_name]…)

- CREATE CLUSTERED INDEX idx_MASV

  ON SINHVIEN(MASV)

- CREATE NONCLUSTERED INDEX idx_TENSV

  ON SINHVIEN(TENSV)

Index trên SQL Server

- Khóa chính → Cluster Index
- Các cột hay truy xuất nên tạo non cluster index.→ tăng tốc độ truy xuất CSDL.