

BỘ MÔN HỆ THỐNG THÔNG TIN
KHOA CÔNG NGHỆ THÔNG TIN – ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP HCM

CƠ SỞ DỮ LIỆU NÂNG CAO

Chương 09:

THIẾT KẾ CƠ SỞ DỮ LIỆU MỨC VẬT LÝ (PHYSICAL DATABASE DESIGN)

Giảng viên: TS. Nguyễn Trần Minh Thư



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



Objectives

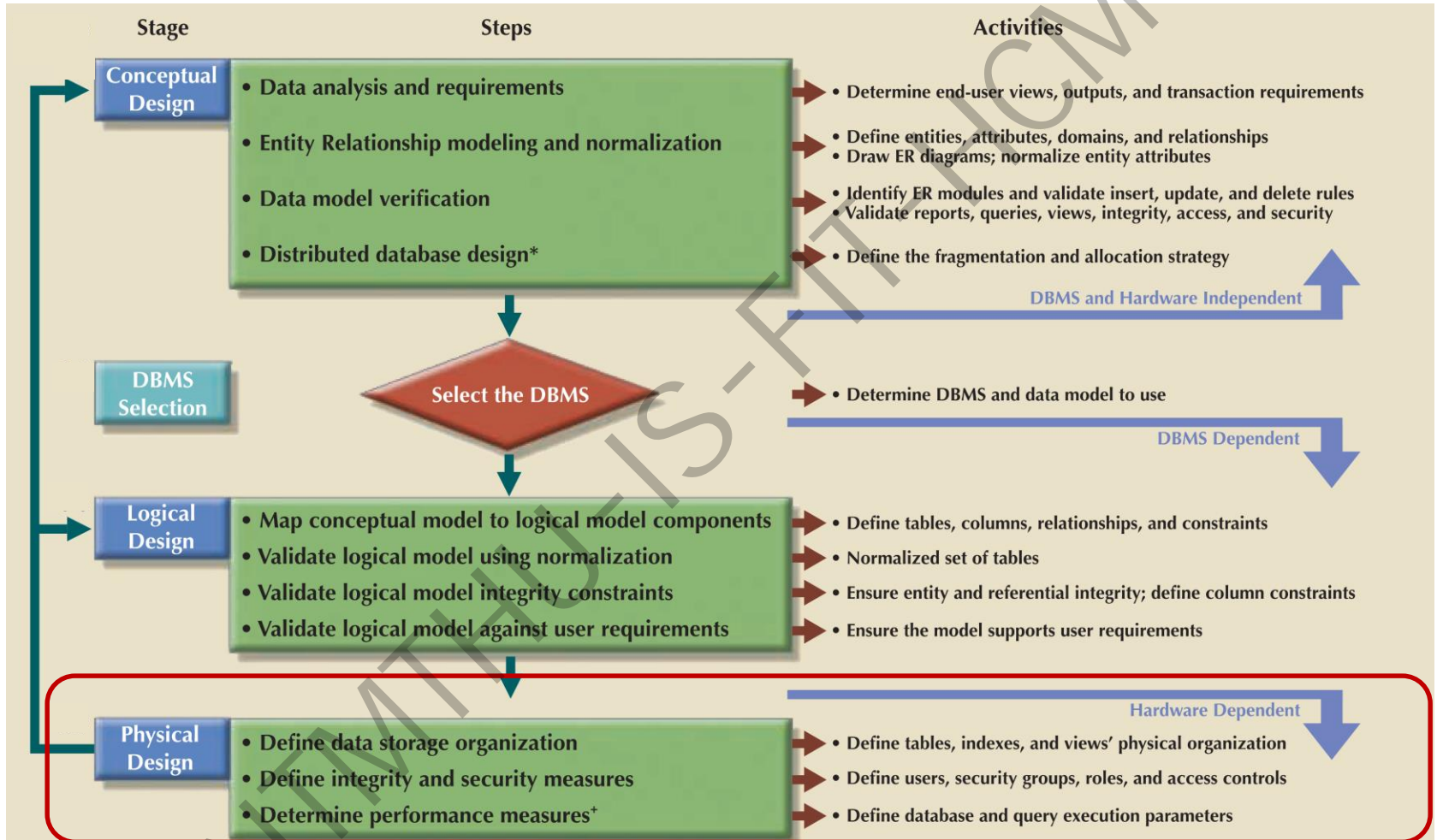
- Define terms
- Describe the physical database design process
- Choose storage formats for attributes
- Select appropriate file organizations
- Describe three types of file organization
- Describe indexes and their appropriate use
- Translate a database model into efficient structures
- Know when and how to use denormalization



Physical Database Design

- **Purpose**—translate the logical description of data into the technical specifications for storing and retrieving data
- **Goal**—create a design for storing data that will provide adequate performance and ensure database integrity, security, and recoverability

Database Design Process



Thiết kế CSDL mức vật lý là gì?

Thiết kế CSDL mức vật lý là một qui trình đưa ra việc mô tả việc thực thi của CSDL trên bộ nhớ phụ bao gồm: mô tả các quan hệ (bảng dữ liệu); cách tổ chức tập tin (file organization), chỉ mục được sử dụng để tăng hiệu quả việc truy cập dữ liệu, các ràng buộc toàn vẹn, phạm vi bảo mật dữ liệu.

Thiết kế CSDL mức vật lý không phải một hoạt động độc lập, mà nó có liên quan đến mức logic và thiết kế ứng dụng.

Yêu cầu đối với mức thiết kế dữ liệu vật lý:

- Biết rõ về hệ thống máy tính đang cài đặt HQT CSDL (HĐH)
- Biết rõ về các chức năng cung cấp của HQT CSDL
- Ý đồ khai thác dữ liệu

Các bước thiết kế vật lý

- **Bước 1**: Chuyển mô hình dữ liệu ở mức logic sang một HQT CSDL đích cụ thể.
- **Bước 2**: Lựa chọn tổ chức tập tin và chỉ mục
- **Bước 3**: Thiết kế các khung nhìn người dùng
- **Bước 4**: Thiết kế các kỹ thuật bảo mật
- **Bước 5**: Xem xét việc quản lý dư thừa
- **Bước 6**: Giám sát và tinh chỉnh hệ thống xử lý



Các bước thiết kế CSDL mức vật lý

Bước 1: Chuyển mô hình dữ liệu ở mức logic sang một HQT CSDL đích cụ thể

- 1.1 Thiết kế các quan hệ cơ sở (bảng dữ liệu cơ sở)
- 1.2 Thiết kế hình thức của dữ liệu suy diễn
- 1.3 Thiết kế các ràng buộc của doanh nghiệp

Bước 2: Thiết kế thể hiện vật lý

- 2.1 Phân tích truy vấn/giao tác
- 2.2 Chọn tổ chức lưu trữ tập tin (File Organization)
- 2.3 Lựa chọn chỉ mục
- 2.4 Ước lượng yêu cầu không gian đĩa

Bước 3: Thiết kế các khung nhìn người dùng

Bước 4: Thiết kế kỹ thuật bảo mật

Bước 5: Xem xét việc quản lý dư thừa

Bước 6: Giám sát và tinh chỉnh hệ thống xử lý



4.0

Bước 1: Chuyển đổi mô hình dữ liệu logic

- **Bước 1: Chuyển đổi mô hình dữ liệu ở mức logic cho một DBMS đích cụ thể.**
Mục tiêu: Xây dựng một lược đồ cơ sở dữ liệu quan hệ từ mô hình dữ liệu thiết kế được ở mức logic để có thể cài đặt được trong HQT CSDL đích đã chọn.
- **Bước 1.1: Thiết kế các quan hệ cơ sở**
Mục tiêu: Quyết định làm thế nào để biểu diễn lại các quan hệ đã xác định được trong mô hình dữ liệu ở mức vật lý vào HQT CSDL đích.
- **Bước 1.2: Thiết kế hình thức của dữ liệu suy diễn**
Mục tiêu: Quyết định làm thế nào để biểu diễn các thể hiện của dữ liệu suy diễn trong mô hình dữ liệu mức vật lý vào HQT CSDL đích
- **Bước 1.3: Thiết kế ràng buộc của doanh nghiệp**
Mục tiêu: Thiết kế các ràng buộc doanh nghiệp vào HQT CSDL đích



4.0

Bước 1.1 Thiết kế các quan hệ cơ sở

- **Mô tả thông tin cho các quan hệ cơ sở:**
 - Dùng ngôn ngữ định nghĩa dữ liệu để định nghĩa quan hệ cơ sở
 - Dùng từ điển dữ liệu để mô tả phạm vi (domain) của thuộc tính
- **Khai báo/thực thi các quan hệ cơ sở:**
 - Cách thức khai báo phụ thuộc vào HQT CSDL
 - Dùng SQL chuẩn để mô tả (MS SQL, Oracle, MS Access)
- **Viết tài liệu cho việc thiết kế các quan hệ cơ sở**
 - Ghi nhận giải pháp thiết kế đã lựa chọn đề xuất

Bước 1.1 Thiết kế các quan hệ cơ sở

- **Mô tả thông tin cho các quan hệ cơ sở:**
 - Dùng ngôn ngữ định nghĩa dữ liệu để định nghĩa quan hệ cơ sở:
 - Tên của quan hệ
 - Danh sách các thuộc tính
 - Khóa chính, khóa ngoại, khóa thay thế
 - Danh sách các thuộc tính suy diễn và cách chúng được tính toán.
 - Xác định ràng buộc tham chiếu cho bất kỳ khóa ngoại nào được xác định.



Bước 1.1 Thiết kế các quan hệ cơ sở

- **Mô tả thông tin cho các quan hệ cơ sở:**
 - Mô tả phạm vi của từng thuộc tính trong từ điển dữ liệu:
 - Kiểu dữ liệu
 - Chiều dài cho phép
 - Ràng buộc cho thuộc tính (Null, default, unique,...)



4.0

Ví dụ: Mô tả thông tin cho quan hệ cơ sở

The physical design of the Branch table using an extended DBDL.

domain Branch_Numbers	fixed length character string length 4
domain Street_Names	variable length character string maximum length 30
domain City_Names	variable length character string maximum length 20
domain State_Codes	fixed length character string length 2
domain Zip_Codes	fixed length character string length 5
domain Staff_Numbers	fixed length character string length 5

Branch(branchNo	Branch_Numbers	NOT NULL,
	street	Street_Names	NOT NULL,
	city	City_Names	NOT NULL,
	state	State_Names	NOT NULL,
	zipCode	Zip_Codes	NOT NULL,
	mgrStaffNo	Staff_Numbers	NOT NULL)
	Primary Key branchNo		
	Alternate Key zipCode		
	Foreign Key mgrStaffNo References Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION		

**Khai báo vào DBMS
bằng cách nào?**



4.0

Bước 1.1 Ví dụ

THIẾT KẾ VẬT LÝ CHO BẢNG BRANCH

```
CREATE TABLE BRANCH (  
  branchNo CHAR(4) NOT NULL CONSTRAINT PK_Branch PRIMARY KEY,  
  street VARCHAR(30) NOT NULL,  
  city VARCHAR(20) NOT NULL,  
  state CHAR(2) NOT NULL,  
  zipCode CHAR(5) NOT NULL CONSTRAINT U_Branch UNIQUE ,  
  mgrStaffNo CHAR(5) NOT NULL,  
  CONSTRAINT FK_BmgrStaffNo FOREIGN KEY REFERENCES STAFF(staffNo) ON DELETE  
  NO ACTION ON UPDATE CASCADE  
)
```

[Source:http://msdn.microsoft.com/en-us/library/ms174979.aspx](http://msdn.microsoft.com/en-us/library/ms174979.aspx)

Bước 1.2 Quyết định dữ liệu suy diễn

- Thuộc tính suy diễn là giá trị của thuộc tính được tính toán từ giá trị của các thuộc tính khác.
- Ví dụ:
 - Số lượng nhân viên cho mỗi phòng ban
 - Tổng lương chi trả hằng tháng cho tất cả nhân viên.
 - Tổng tiền thanh toán của hóa đơn.
 - Số dư của tài khoản khách hàng.
- Xây dựng danh sách các thuộc tính suy diễn và mô tả vào trong từ điển dữ liệu.
- Cần quyết định thuộc tính suy diễn cần lưu trữ thật sự vào CSDL hay sẽ được tính toán vào thời điểm truy xuất.



4.0

Bước 1.2 Quyết định dữ liệu suy diễn

- Thiết kế viên cần cân nhắc giữa:
 - Chi phí lưu trữ cho thuộc tính suy diễn của nhất quán dữ liệu
 - Và Chi phí phải tính toán ra giá trị cho cho mỗi thời điểm yêu cầu (phép kết, na sum, count,...)
- Ví dụ: Số lượng nhân viên cho mỗi phòng ban?
 - Số lượng nhân viên cho mỗi phòng ban
 - Tổng lương chi trả hằng tháng cho tất cả nhân viên.
 - Tổng tiền thanh toán của hóa đơn.
 - Số dư của tài khoản khách hàng.
- Viết tài liệu cho quá trình thiết kế dữ liệu suy diễn

SQL bị giới hạn bởi các hàm kết hợp và truy vấn đệ qui

Bước 1.2 Quyết định dữ liệu suy diễn

Ví dụ về thể hiện của dữ liệu suy diễn

RentalAgreement

rentalNo	dateOut	dateReturn	memberNo	videoNo
R753461	4-Feb-03	6-Feb-03	M284354	245456
R753462	4-Feb-03	6-Feb-03	M284354	243431
R668256	5-Feb-03	7-Feb-03	M115656	199004
R668189	2-Feb-03		M115656	178643

Member

memberNo	fName	lName	address	noOfRentals
M250178	Bob	Adams	57 – 11th Avenue, Seattle, WA 98105	0
M166884	Art	Peters	89 Redmond Rd, Portland, OR 97117	0
M115656	Serena	Parker	22 W. Capital Way, Portland, OR 97201	2
M284354	Don	Nelson	123 Suffolk Lane, Seattle, WA 98117	2

Bước 1.2 Quyết định dữ liệu suy diễn

Ví dụ: Cho biết số lượng video 'Don Nelson' thuê

```
SELECT COUNT(*) AS noOfRentals
FROM Member m, RentalAgreement ra
WHERE m.memberNo= ra.memberNo AND m.fName = 'Don'
      AND m.lName = 'Nelson';
```

```
SELECT noOfRentals
FROM Member
WHERE fName = 'Don' AND lName = 'Nelson';
```

TIP

It may also be more appropriate to store derived columns whenever the system's query language cannot easily cope with the algorithm to calculate the derived column. For example, SQL has a limited set of aggregate functions and also cannot easily handle recursive queries.



4.0

Bước 1.3 Thiết kế các ràng buộc của hệ thống

- Việc thiết kế các ràng buộc của hệ thống phụ thuộc vào HQT CSDL đích. Do vậy:
 - Cần kiểm tra RB của hệ thống có được HQT CSDL hỗ trợ khai báo.
 - Nếu HQT không hỗ trợ thì RB sẽ được khai báo ở mức ứng dụng.
 - Cần cân nhắc nơi kiểm tra RB, vì việc này có ảnh hưởng đến hiệu quả hệ thống thực thi.
- Viết tài liệu cho quá trình thiết kế ràng buộc hệ thống



4.0

Ví dụ Bước 1.3

- Ràng buộc: Một người thuê không được thuê quá 10 căn nhà tại một thời điểm.
- Cách khai báo trong SQL?

```
CREATE TRIGGER TR_SLNha ON RENTAL
FOR INSERT, UPDATE
AS
BEGIN
    DECLARE X int,
    SELECT X = COUNT(R.propertyNo)
    FROM INSERTED I, RENTAL R
    WHERE I.clientNo = R.clientNo
    GROUP BY R.clientNo
    IF X >= 10 THEN
        BEGIN
            RAISERROR ('Không thuê quá 10 căn nhà', 0,1)
            ROLLBACK TRANS
        END
    END
```

Các bước thiết kế CSDL mức vật lý

Bước 1: Chuyển mô hình dữ liệu ở mức logic sang một HQT CSDL đích cụ thể

- 1.1 Thiết kế các quan hệ cơ sở (bảng dữ liệu cơ sở)
- 1.2 Thiết kế hình thức của dữ liệu suy diễn
- 1.3 Thiết kế các ràng buộc của doanh nghiệp

Bước 2: Thiết kế thể hiện vật lý

- 2.1 Phân tích truy vấn/giao tác
- 2.2 Chọn tổ chức lưu trữ tập tin (File Organization)
- 2.3 Lựa chọn chỉ mục
- 2.4 Ước lượng yêu cầu không gian đĩa

Bước 3: Thiết kế các khung nhìn người dùng

Bước 4: Thiết kế kỹ thuật bảo mật

Bước 5: Xem xét việc quản lý dư thừa

Bước 6: Giám sát và điều chỉnh hệ thống xử lý

Bước 2: Thiết kế thể hiện vật lý

- Mục tiêu:

Xác định cách tổ chức tập tin tối ưu để lưu trữ các quan hệ cơ sở và yêu cầu chỉ mục được xây dựng để đạt được tốc độ thực thi chấp nhận. Nghĩa là, xây dựng cách thức các quan hệ và thuộc tính sẽ được lưu giữ trên bộ nhớ phụ.

- Cần xem xét:

- Tần suất giao tác/truy vấn
- Thời gian hồi đáp
- Không gian lưu trữ dữ liệu
- Phần cứng:
 - Bộ nhớ chính (main memory)
 - CPU
 - Disk I/O
 - Network

Bước 2: Thiết kế biểu diễn vật lý

- **Bước 2.1: Phân tích các giao tác/truy vấn**

Mục tiêu: Hiểu đặc trưng của giao tác sẽ được thực hiện trong CSDL và phân tích tầm quan trọng của nó.

- **Bước 2.2: Chọn cách tổ chức tập tin**

Mục tiêu: Xác định một cách tổ chức tập tin hiệu quả cho mỗi quan hệ cơ sở

- **Bước 2.3: Lựa chọn chỉ mục**

Mục tiêu: Xác định liệu chỉ mục nào được thêm vào để tăng tốc độ của hệ thống.

- **Bước 2.4: Ước lượng không gian lưu trữ**

Bước 2.1: Phân tích giao tác

- Xác định truy vấn được thực hiện thường xuyên.
- Các xử lý liên quan đến xử lý quy trình nghiệp vụ.
- Tần suất của truy vấn: *Số lần được thực hiện trong ngày/tuần/tháng.*

Bước 2.1: Phân tích giao tác

- Sử dụng ma trận tham chiếu truy vấn/quan hệ để xác định quan hệ cần điều tra nghiên cứu:
 - Sắp xếp các đường dẫn của truy vấn lên các quan hệ.
 - Xác định quan hệ nào truy cập thường xuyên bởi các truy vấn.
 - Phân tích tính hữu dụng của dữ liệu kết quả của các câu truy vấn mà ảnh hưởng đến các quan hệ.

Bước 2.1: Phân tích giao tác

Biểu diễn các đường dẫn truy vấn vào ma trận

Truy vấn/Quan hệ	Truy vấn 1				Truy vấn 2				...				Truy vấn m			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Quan hệ 1		X			X	X	X									
Quan hệ 2						X										
Quan hệ 3		X	X	X					X	X			X	X		
Quan hệ 4																
...																
....																
Quan hệ n																

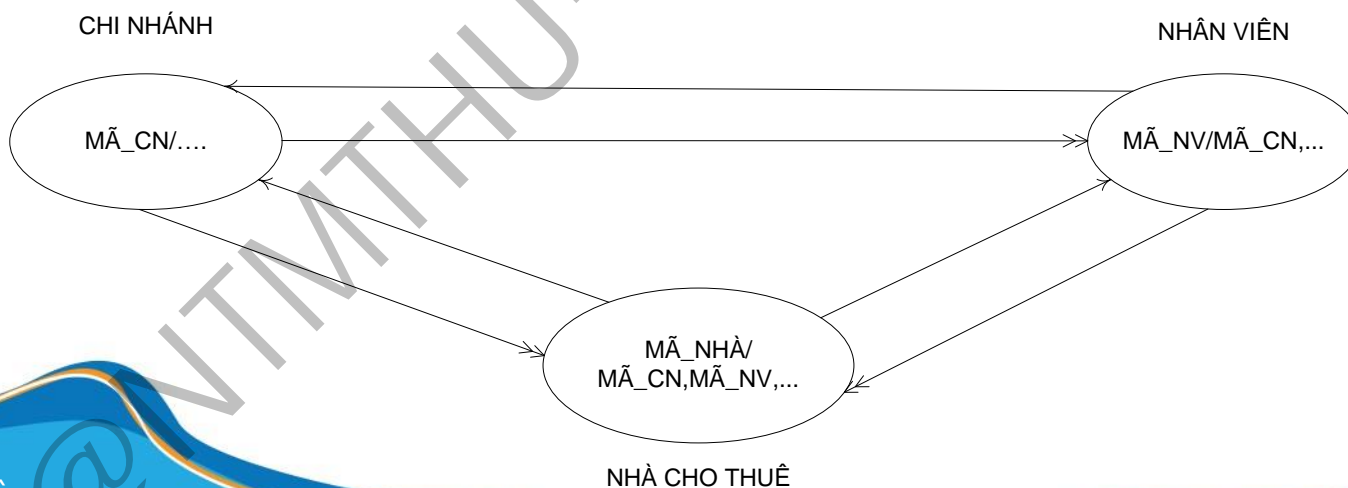
Ma trận tham chiếu truy vấn/quan hệ

Ghi chú: I = Insert; R = Read; U = Update, D = Delete

- **Truy vấn 1:** Xác định tổng số nhân viên có trong của mỗi chi nhánh tại TP HCM (tên chi nhánh, tổng số NV).
- **Truy vấn 2:** Cho biết thông tin của các nhà cho thuê của chi nhánh tại TP HCM, kết quả sắp xếp theo giá thuê.
- **Truy vấn 3:** Cho biết thông tin của các nhà cho thuê và tên nhân viên quản lý.
- **Truy vấn 4:** Cho biết số nhà mà mỗi nhân viên tại chi nhánh ABC quản lý.

Biểu diễn các đường dẫn truy vấn vào ma trận

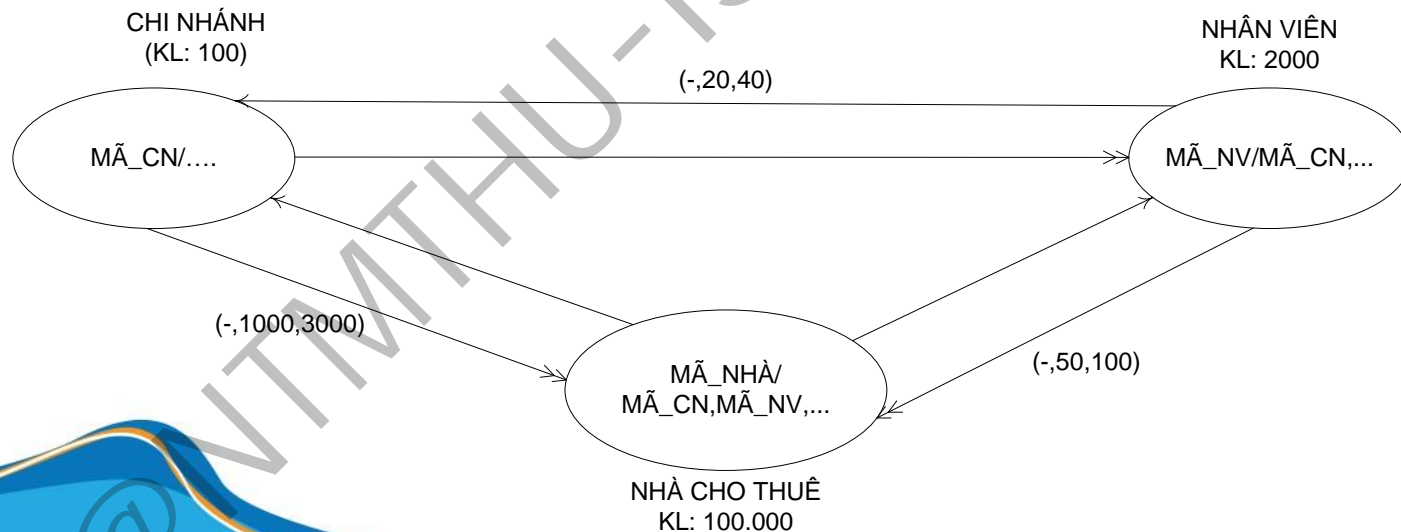
Truy vấn/Quan hệ	Truy vấn 1				Truy vấn 2				Truy vấn 3				Truy vấn 4			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
CHI NHÁNH		X				X								X		
NHÂN VIÊN		X								X				X		
NHÀ CHO THUÊ						X				X				X		



Bước 2.1: Phân tích giao tác

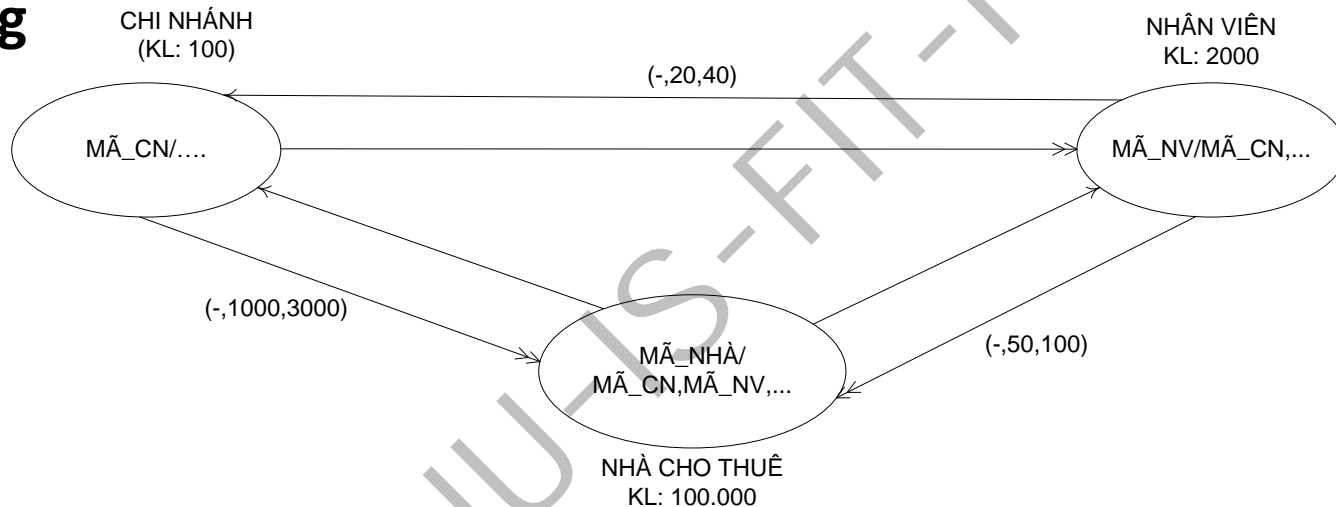
Xác định thông tin truy xuất thường xuyên

Truy vấn/Quan hệ	Truy vấn 1				Truy vấn 2				Truy vấn 3				Truy vấn 4			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
CHI NHÁNH		X				X								X		
NHÂN VIÊN		X								X				X		
NHÀ CHO THUÊ						X				X				X		



Bước 2.1: Phân tích giao tác

- Xem xét khối lượng dữ liệu để xác định bảng dữ liệu quan trọng



- Xem xét tần suất truy vấn để xác định truy vấn quan trọng:
 - Các truy vấn chạy với tần suất truy bình ở là như nhau các thời điểm, nhưng Truy vấn 2 chạy với tần suất cao vào 14h-16h vào thứ 5, 6 hằng tuần.



4.0

Bước 2.1: Phân tích giao tác

- Tiếp tục phân tích chi tiết từng truy vấn:
 - Các quan hệ và các thuộc tính mà câu truy vấn truy cập, và mỗi loại truy cập đó là gì? (I, U, D, R).
 - Các thuộc tính được sử dụng trong điều kiện tìm kiếm.
 - Với truy vấn, các thuộc tính được gắn vào phép kết từ 2 quan hệ hoặc nhiều hơn.
 - Tần suất của mỗi truy vấn.
- ⇒ Các thuộc tính xác định được này có độ ưu tiên cao, và thường sẽ là thuộc tính ứng viên trong cấu trúc truy cập.

**MẪU PHÂN TÍCH TRUY VẤN****16/02/2012****Truy vấn:** (2) Cho biết thông tin của các nhà cho thuê của chi nhánh tại TP HCM, kết quả sắp xếp theo giá thuê**Tần suất TV:** Trung bình: 50 lần/giờ,

Cao điểm: 100 lần/giờ (từ 17h-19h, Thứ 2 đến Thứ 7)

SELECT

FROM CHINHANH c INNER JOIN NHACHOTHUE n ON c.MACN = c.MACN

WHERE c.DIACHI LIKE '%TP HCM%'

ORDER BY n.GIATHUE

Điều kiện: c.DIACHI LIKE '%TP HCM%'.

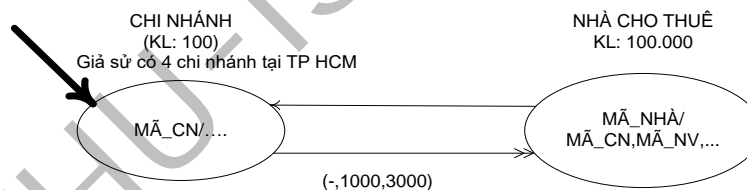
Thuộc tính kết: c.MACN = c.MACN

Thuộc tính sx: GIATHUE

Thuộc tính gom nhóm: không

Các hàm xây dựng: không

Thuộc tính cập nhật: không

Đồ thị con đường truy xuất dữ liệu:

Truy xuất	Quan hệ	Loại truy xuất	Thông số về truy xuất		
			Trên truy vấn	Trung bình/h	Cao điểm/h
1	CHINHANH	R	100	5.000	10000
2	NHACHOTHUE	R	4.000 - 12000	200000 - 600000	400000 - 12000000
Tổng cộng truy xuất			4100 - 12100	205000 - 605000	410000 - 1210000

Cơ sở dữ liệu nâng cao



4.0

Bước 2.2 Tổ chức tập tin

- Các khái niệm cơ bản:
 - Tập tin dữ liệu (data files)
 - Trang (page) /Khối (block)
 - Bộ (record)
 - Số bộ được lưu trên một khối (Blocking factor): $bfr = B/R$
 - Unspanned records: không cho phép lưu một record trên nhiều blocks
 - Spanned Records: cho phép lưu một record trên nhiều blocks.

Bước 2.2 Tổ chức tập tin

- Minh họa các khái niệm cơ bản:

MÃNV	TÊN	CHỨC VỤ	MÃCN
SL21	White	Manager	B005
SG37	Beech	Assistant	B003
SG14	Ford	Supervisor	B003
SA9	Howe	Assistant	B007
SG5	Brand	Manager	B003
SL41	Lee	Assistant	B003

Block 1

Block 2

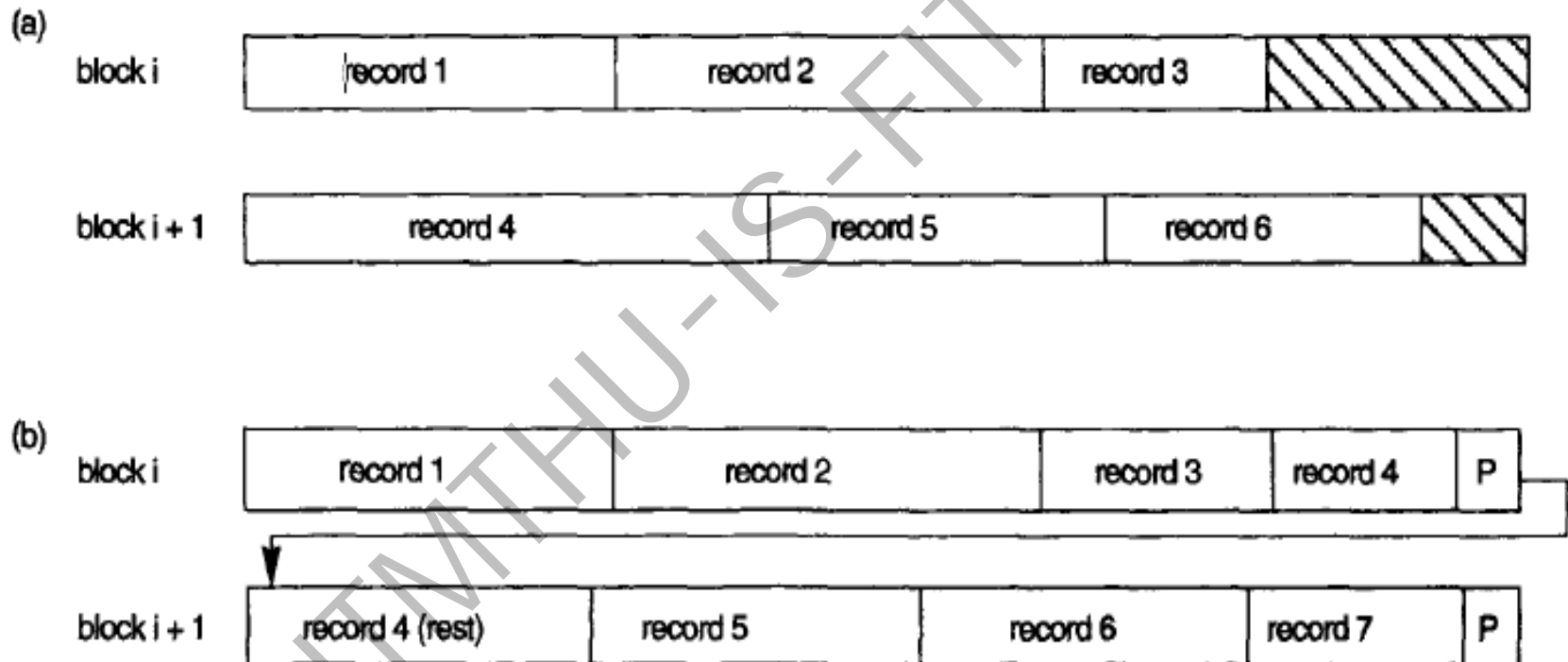
Record 6



4.0

Bước 2.2 Tổ chức tập tin

- Minh họa spanned and unspanned record:

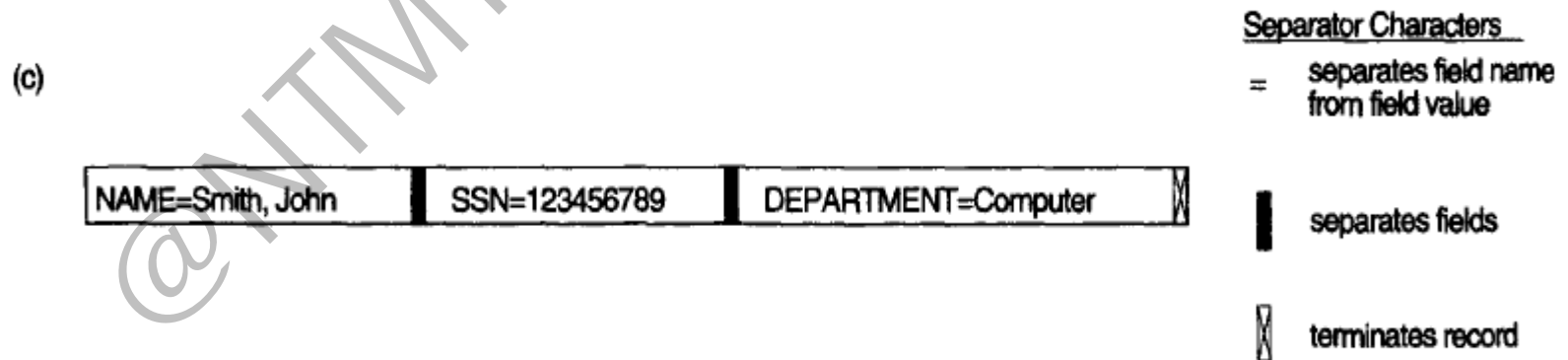
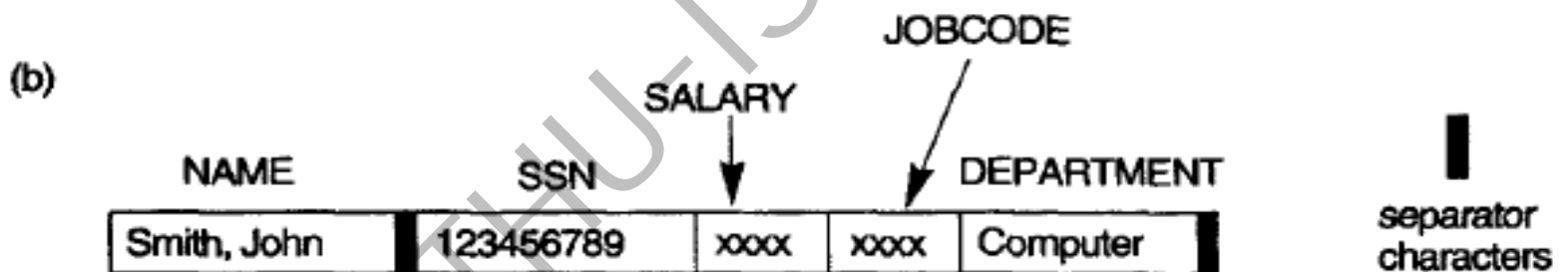
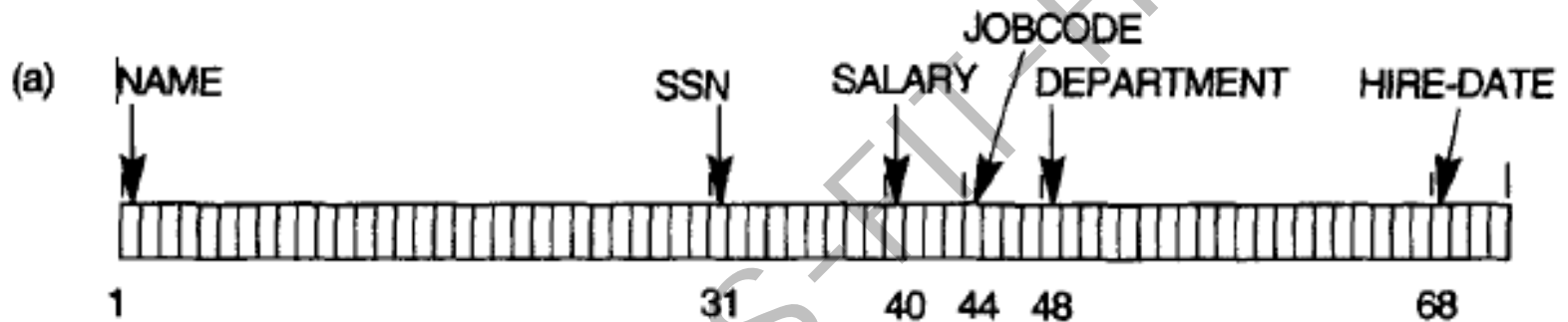




4.0

Bước 2.2 Tổ chức tập tin

- Minh họa record có chiều dài cố định và biến động:





4.0

Bước 2.2 Tổ chức tập tin

- **Mục tiêu:** xác định một cách tổ chức file hiệu quả cho mỗi quan hệ **nếu DBMS cho phép điều này.**
- Tổ chức tập tin là cách sắp xếp vật lý dữ liệu các tập tin gồm các records và blocks trong bộ nhớ thứ cấp (secondary storage).
- Có các loại tổ chức tập tin:
 - Heap (unordered) Files.
 - Sequential (ordered) Files.
 - Hash Files
 - B+-Tree
 - Clusters



4.0

Bước 2.3 Lựa chọn chỉ mục

- Chỉ mục là gì?
- Các loại chỉ mục đơn cấp (single – level) có thứ tự
 - Primary Indexes
 - Clustering Indexes
 - Secondary Indexes
- Chỉ mục đa cấp (Multilevel Indexes)
- Chỉ mục đa cấp động (Dynamic Multilevel Indexes) sử dụng B-Trees và B++Trees
- Chỉ mục trên nhiều khóa (Indexes on Multiple Keys)



4.0

Chỉ mục là gì?

- Chỉ mục được gọi là các đường dẫn truy cập trên file (access paths)
- Chỉ mục đơn cấp là một file phụ (auxiliary) hỗ trợ tìm kiếm hiệu quả một record trong file dữ liệu.
- Chỉ mục thường được chỉ định trên một thuộc tính của file (mặc dù nó có thể chỉ định trên nhiều thuộc tính)
- Cấu trúc của một index trên file chứa các entries gồm <**giá trị thuộc tính (field value), con trỏ trỏ đến nơi chứa dữ liệu (pointer to record/block)**>, với giá trị thuộc tính được sắp theo thứ tự.
- File chỉ mục thường chiếm số lượng blocks trên đĩa ít hơn file dữ liệu, bởi vì số lượng entries của nó nhỏ hơn.



4.0

Chỉ mục là gì?

- Áp dụng tìm kiếm nhị phân trên file chỉ mục để tìm đến dữ liệu trong file dữ liệu.
- Chỉ mục cũng có đặc tính là: chỉ mục dày (dense) hoặc chỉ mục thưa (sparse).
 - Chỉ mục dày là một mục chỉ mục (index entry) có một giá trị khóa tìm kiếm (mỗi record) trong file dữ liệu.
 - Chỉ mục thưa là có nhiều mục chỉ mục cho một vài giá trị tìm kiếm.

Chỉ mục là gì?

Cho một file dữ liệu: EMPLOYEE(NAME, SSN, ADDRESS, JOB, SAL, ...)

Giả định rằng:

record size $R = 150$ bytes

block size $B = 512$ bytes

$r = 30000$ records

Các records được lưu trữ theo kỹ thuật unspanned.

Chúng ta có:

Số lượng records trên một Block: $Bfr = B \div R = 512 \div 150 = 3$ records/block

Số lượng block cần sử dụng: $b = (r/Bfr) = (30000/3) = 10000$ blocks

Chi phí trung bình tìm kiếm một record:

- Tuyến tính (linear search): $= b/2 = 10.000/2 = 5000$ (blocks)
- Tìm nhị phân (binary search): $\log_2 b = \log_2 10.000 = 14$ (blocks)



4.0

Chỉ mục là gì?

Cho một index trên thuộc tính SSN, giả định rằng kích thước thuộc tính là $V_{SSN}=9$ bytes, và kích thước con trỏ là $P_R=7$ bytes. Ta có:

Kích thước của một index entry: $R_I = (V_{SSN} + P_R) = (9 + 7) = 16$ bytes

Số index entries trên một block:

$$Bfr_I = B \div R_I = 512 \div 16 = 32 \text{ (entries/block)}$$

Số index blocks: $b = (r / Bfr_I) = (30000 / 32) = 938$ blocks

Thời gian tìm kiếm một record:

- Tuyến tính: $= b/2 = 938/2 = 469$ (blocks).
- Nhị phân: $\log_2 b = \log_2 938 = 10$ (blocks)



4.0

Các loại chỉ mục đơn cấp

- **Chỉ mục chính (Primary Index)**

- Được định nghĩa trên một file dữ liệu có thứ tự.
- File dữ liệu được sắp xếp trên thuộc tính khóa.
- Một index entry tương ứng với một block trên file dữ liệu, index entry có giá trị thuộc tính khóa cho record đầu tiên trong block, được gọi là block anchor.
- Một lược đồ tương tự có thể sử dụng cho record cuối cùng trong block.
- **Primary index là chỉ mục thưa (sparse)**, bởi vì mỗi một entry tương ứng với mỗi block đĩa của file dữ liệu và giá trị khóa của anchor record.

Anchor record

Index entry

DATA FILE

(PRIMARY KEY FIELD)

NAME

SSN

BIRTHDATE

JOB

SALARY

SEX

Aaron, Ed					
Abbott, Diane					
⋮					
Acosta, Marc					
⋮					
Adams, John					
Adams, Robin					
⋮					
Akers, Jan					
⋮					
Alexander, Ed					
Alfred, Bob					
⋮					
Allen, Sam					
⋮					
Allen, Troy					
Anders, Keith					
⋮					
Anderson, Rob					
⋮					
Anderson, Zach					
Angeli, Joe					
⋮					
Archer, Sue					
⋮					
Arnold, Mack					
Arnold, Steven					
⋮					
Atkins, Timothy					
⋮					
Wong, James					
Wood, Donald					
⋮					
Woods, Manny					
⋮					
Wright, Pam					
Wyatt, Charles					
⋮					
Zimmer, Byron					

INDEX FILE
($\langle K(i), P(i) \rangle$ entries)

BLOCK
ANCHOR
PRIMARY
KEY
VALUE

BLOCK
POINTER

Aaron, Ed	•
Adams, John	•
Alexander, Ed	•
Allen, Troy	•
Anderson, Zach	•
Arnold, Mack	•
⋮	
⋮	
Wong, James	•
Wright, Pam	•
⋮	



4.0

Ví dụ - Primary Index

Giả định rằng:

record size $R = 100$ bytes, block size $B = 1024$ bytes, $r = 30000$ records.

Sử dụng chỉ mục primary index với $V_{\text{INDEX FIELD}} = 9$ bytes, Pointer size $P = 6$.

Các records trong file dữ liệu được lưu trữ theo kỹ thuật unspanned.

a) Hãy tính số block trung bình phải truy xuất khi tìm một record bất kỳ khi không sử dụng index (áp dụng cả hai phương pháp tìm tuyến tính và nhị phân).

b) Thực hiện câu a) khi sử dụng primary index



4.0

Ví dụ - Primary Index

a) Hãy tính số block trung bình phải truy xuất khi tìm một record bất kỳ khi không sử dụng index (áp dụng cả hai phương pháp tìm tuyến tính và nhị phân).

- Số record trên 1 block: $bfr = 1024/100 = 10$
- Số block trên data file: $b = 30000/10 = 3000$ (blocks)
- Tuyến tính: $b/2 = 1500$ (block).
- Nhị phân: $\log_2 b = \log_2 3000 = 12$ (blocks)

b) Thực hiện câu a) khi sử dụng primary index

- Kích thước một index entry: $R_i = 9 + 6 = 15$ (bytes)
- Số entry trong một block: $bfr_i = B/R_i = 1024/15 = 68$ (entries)
- Số entries trên index file = số block trên data file: $r_i = 3000$ (entries)
- Số block trên index file: $b_i = r_i / bfr_i = 3000/68 = 45$ (blocks)
- Tuyến tính: $b_i / 2 + 1 = 24$ (block).
- Nhị phân: $\log_2 b_i = \log_2 45 + 1 = 7$ (blocks)



4.0

Các loại chỉ mục đơn cấp

- **Chỉ mục gom cụm (Clustering Index)**
 - Được định nghĩa trên file dữ liệu có thứ tự.
 - File dữ liệu được sắp xếp trên thuộc tính không khóa và các giá trị có thể trùng lặp (không giống như primary index yêu cầu không có giá trị trùng trên các records).
 - Một index entry chứa giá trị khóa tìm kiếm và con trỏ trỏ đến block dữ liệu đầu tiên chứa các records liên quan đến giá trị của thuộc tính index. Cấu trúc có thể minh họa: (**clustering field**, **pointer**)
 - **Chỉ mục gom cụm cũng là một loại chỉ mục thừa**, khi thêm hoặc xóa thì ảnh hưởng đến chỉ mục gom cụm.

DATA FILE

(CLUSTERING
FIELD)

DEPTNUMBER NAME SSN JOB BIRTHDATE SALARY

1					
1					
1					
2					

2					
3					
3					
3					

3					
3					
4					
4					

5					
5					
5					
5					

6					
6					
6					
6					

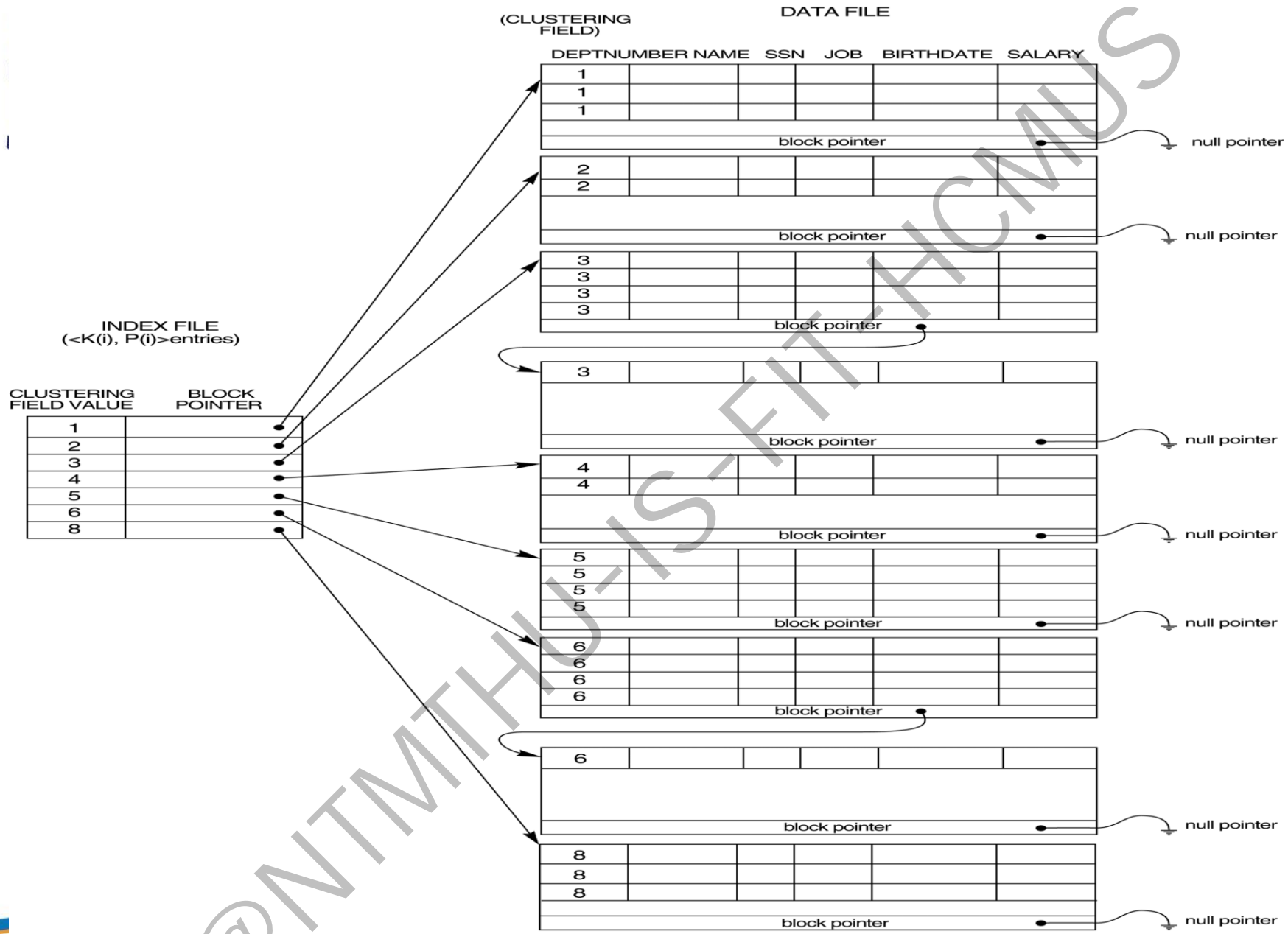
6					
8					
8					
8					

INDEX FILE
(<K(i), P(i)> entries)

CLUSTERING
FIELD VALUE

BLOCK
POINTER

1	
2	
3	
4	
5	
6	
8	





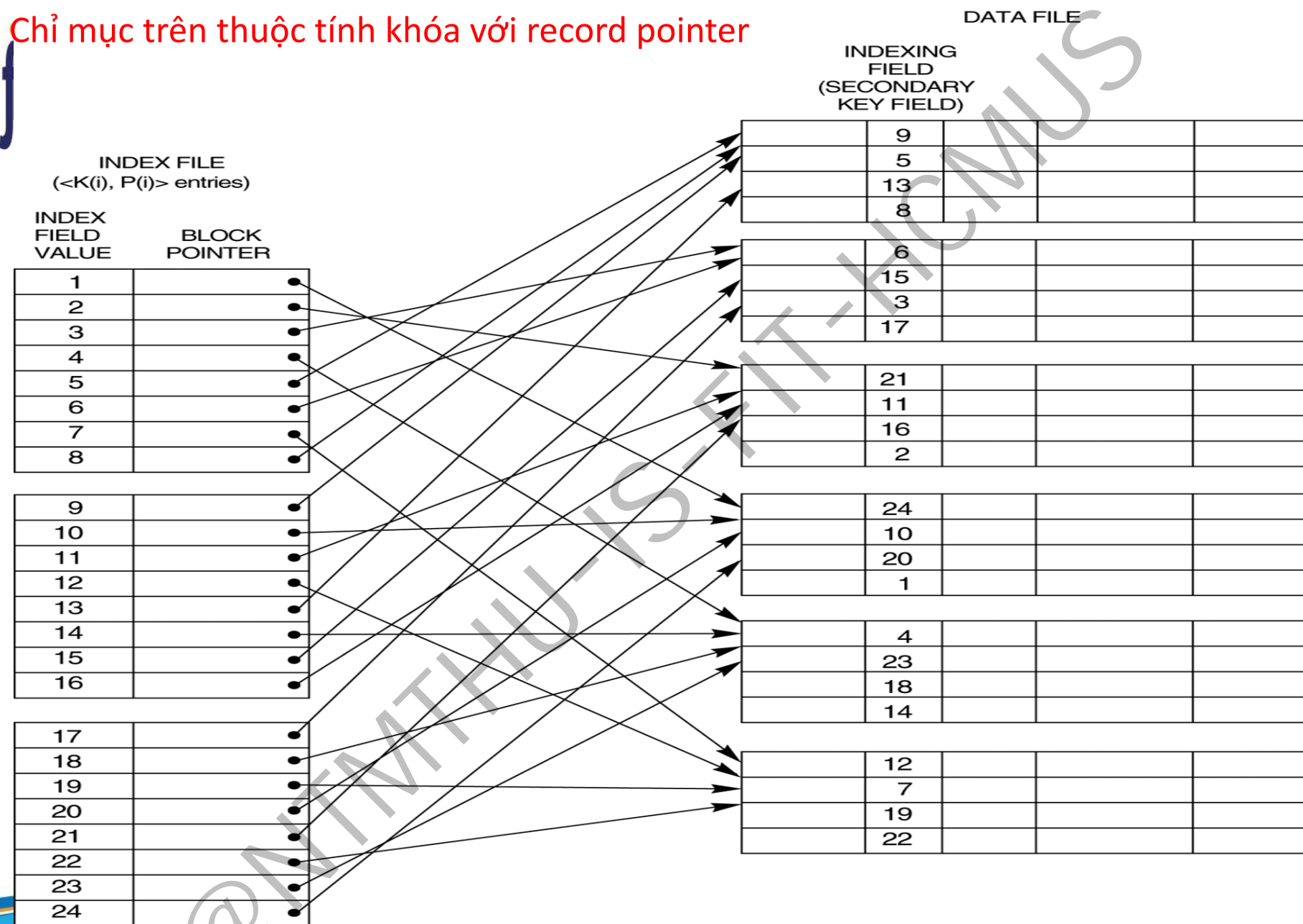
4.0

Các loại chỉ mục đơn cấp

- **Chỉ mục thứ cấp/phụ (Secondary Index)**

- Chỉ mục thứ cấp cung cấp một phương tiện phụ để truy cập đến file với những truy cập chính đã tồn tại.
- Chỉ mục thứ cấp có thể dựa trên thuộc tính khóa ứng viên và có giá trị duy nhất trên mỗi record, hoặc là thuộc tính không khóa mà có chứa các giá trị trùng.
- Chỉ mục là một file được sắp xếp thứ tự theo 2 thuộc tính: thuộc tính chỉ mục và con trỏ block hoặc là một con trỏ record.
- Có thể có nhiều chỉ mục thứ cấp cho cùng một file.
- Mỗi entry chứa một record trên file dữ liệu do vậy đây là loại chỉ mục dày (*dense index*)

Chỉ mục trên thuộc tính khóa với record pointer





4.0

Ví dụ - Secondary Index

Giả định rằng:

record size $R = 100$ bytes, block size $B = 1024$ bytes, $r = 30000$ records.

Sử dụng chỉ chỉ mục secondary index với $V_{\text{INDEX FIELD}} = 9$ bytes, Pointer size $P = 6$.

Các records trong file dữ liệu được lưu trữ theo kỹ thuật unspanned.

- Hãy tính số block trung bình phải truy xuất khi tìm một record bất kỳ khi không sử dụng index theo phương pháp tìm tuyến tính?
- Khi sử dụng secondary index với record pointer thì số block trung bình truy xuất khi tìm tuyến tính và nhị phân là bao nhiêu?



4.0

Ví dụ - Secondary Index

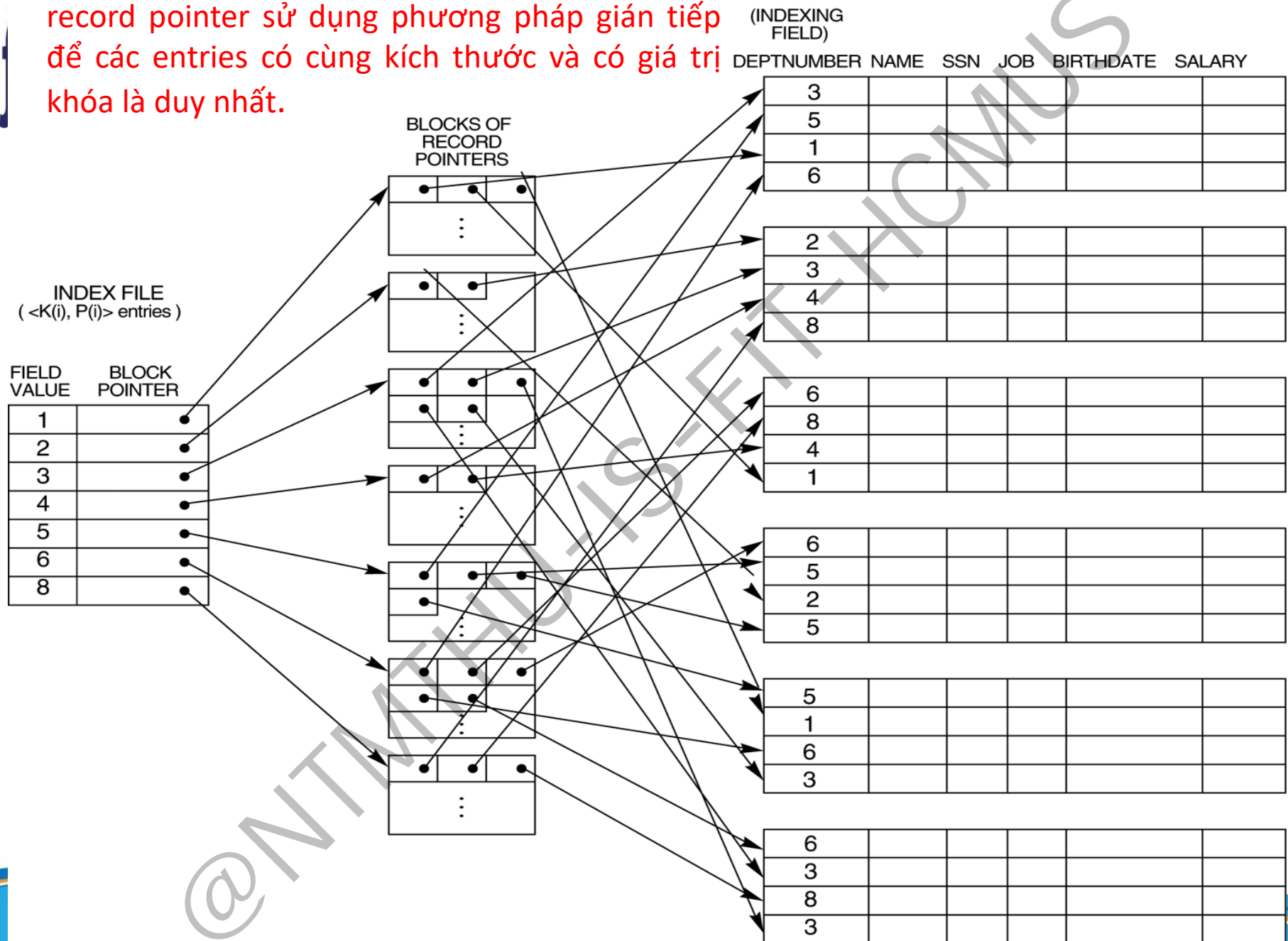
a) Hãy tính số block trung bình phải truy xuất khi tìm một record bất kỳ khi không sử dụng index.

- Số record trên 1 block: $bfr = 1024/100 = 10$
- Số block trên data file: $b = 30000/10 = 3000$ (blocks)
- Tuyển tính: $b/2 = 1500$ (block).

b) Thực hiện câu a) khi sử dụng secondary index (chỉ mục dày)

- Kích thước một index entry: $R_i = 9 + 6 = 15$ (bytes)
- Số entry trong một block: $bfr_i = B/R_i = 1024/15 = 68$ (entries)
- Số entries trên index file = số record trên data file: $r_i = 30000$ (entries)
- Số block trên index file: $b_i = r_i / bfr_i = 30000/68 = 442$ (blocks)
- Tuyển tính: $b_i / 2 + 1 = 222$ (block).
- Nhị phân: $\log_2 b_i = \log_2 442 + 1 = 9$ (blocks) > 7 (blocks) (primary index)

Chỉ mục trên thuộc tính không khóa với
record pointer sử dụng phương pháp gián tiếp
để các entries có cùng kích thước và có giá trị
khóa là duy nhất.



ĐẶC TÍNH CHO CÁC LOẠI INDEX

TABLE 14.2 PROPERTIES OF INDEX TYPES

TYPE OF INDEX	NUMBER OF (FIRST-LEVEL) INDEX ENTRIES	DENSE OR NONDENSE	BLOCK ANCHORING ON THE DATA FILE
Primary	Number of blocks in data file	Nondense	Yes
Clustering	Number of distinct index field values	Nondense	Yes/no ^a
Secondary (key)	Number of records in data file	Dense	No
Secondary (nonkey)	Number of records ^b or Number of distinct index field values ^c	Dense or Nondense	No

^aYes if every distinct value of the ordering field starts a new block; no otherwise.

^bFor option 1.

^cFor options 2 and 3.



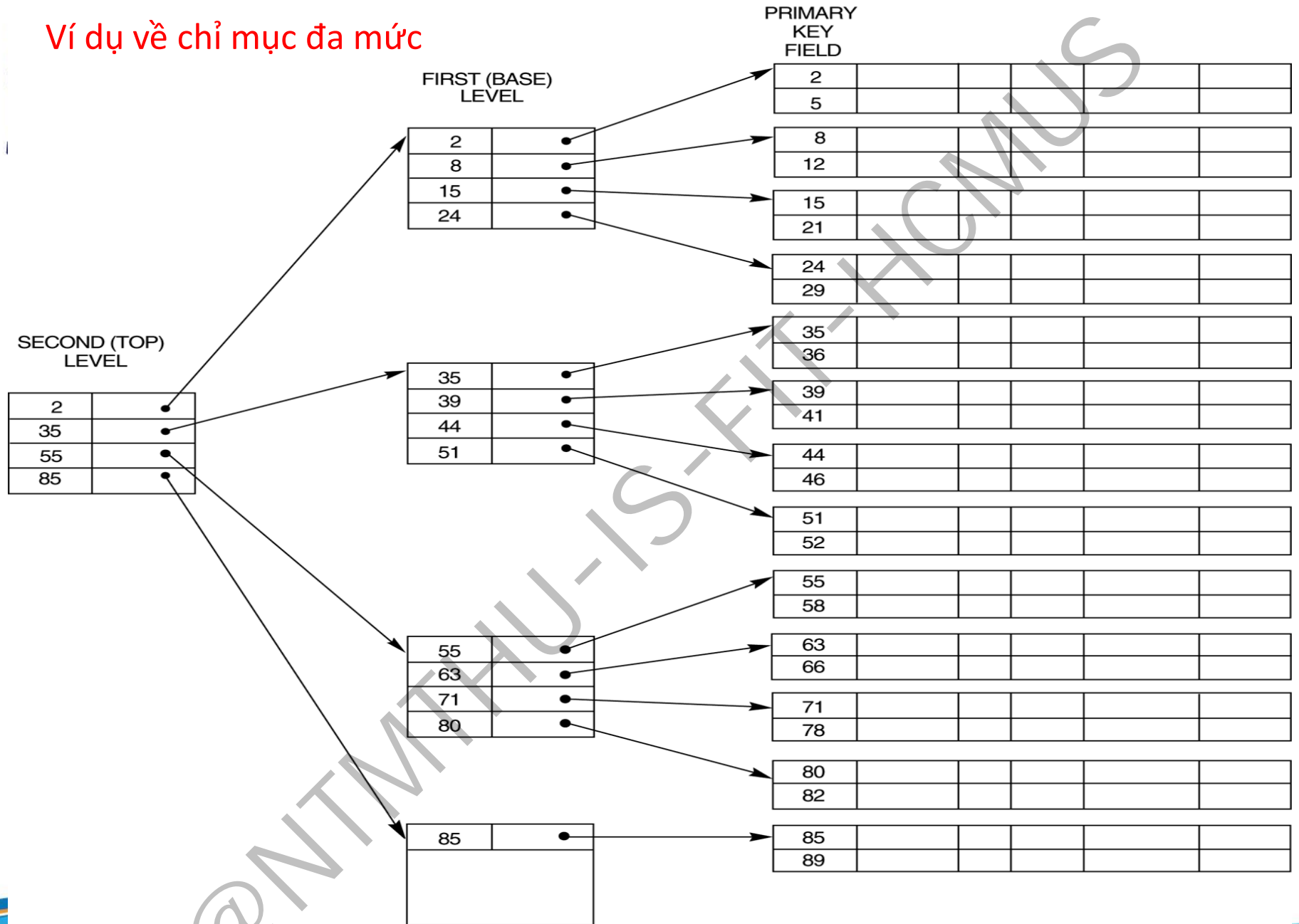
4.0

CHỈ MỤC ĐA CẤP

- Chỉ mục một mức là một file có thứ tự, chúng ta có thể tạo một primary index đến chính chỉ mục đó, lúc này file chỉ mục nguyên thủy được gọi là chỉ mục mức 1, và chỉ mục đến file chỉ mục được gọi chỉ mục thứ 2,...
- Chúng ta có thể lặp lại xử lý trên, nghĩa là tạo chỉ mục thứ 3, thứ 4,... đến mức cao nhất cho đến khi tất cả các entry đều nằm trong 1 block ở mức cao nhất.
- Chỉ mục đa mức có thể được tạo trên bất kỳ loại chỉ mục nào ở mức đầu tiên (primary, secondary, clustering) chỉ cần mức index đầu tiên phải có nhiều hơn 1 block đĩa.
- Kích thước của mỗi entry index ở các cấp là như nhau, do vậy số entry trong một block ở các chỉ mục cũng như nhau và $bfr_i = B/R_i$ và được gọi là f_0 .

- Kích thước của mỗi entry index ở các cấp là như nhau, do vậy số entry trong một block ở các chỉ mục cũng như nhau và $bfr_i = B/R_i$ và được gọi là f_0 .
- Vì thế:
 - Số blocks ở mức thứ 1: $b_1 = r_1/f_0$ và đây cũng chính là số entry cần cho chỉ mục mức thứ 2.
 - Số blocks ở mức thứ 2: $b_2 = b_1/f_0 = r_1/(f_0)^2$
 - ...
 - Số blocks ở mức thứ t : $b_t = r_1/(f_0)^t$
 - Do vậy $t = \log_{f_0}(r_1)$

Ví dụ về chỉ mục đa mức





4.0

VÍ DỤ CHỈ MỤC ĐA MỨC

Giả định rằng:

record size $R = 100$ bytes, block size $B = 1024$ bytes, $r = 30000$ records.

Sử dụng chỉ mục primary index với $V_{\text{INDEX FIELD}} = 9$ bytes, Pointer size $P = 6$.

Các records trong file dữ liệu được lưu trữ theo kỹ thuật unspanned và sử dụng kỹ thuật primary index trên các mức.

- Tính số mức t sẽ dùng?
- Số block trung bình truy xuất khi tìm một mục dữ liệu bất kỳ là bao nhiêu?

VÍ DỤ CHỈ MỤC ĐA MỨC

- Chỉ mục ở mức đầu tiên:

- Kích thước một index entry: $R_i = 9 + 6 = 15(\text{bytes})$
- Số entry trong một block: $f_0 = bfr_i = B/R_i = 1024/15 = 68 (\text{entries})$
- Số entries trên index file = số block trên data file: $r_1 = 3000 (\text{entries})$
- Số block trên index file: $b_1 = r_1 / f_0 = 3000/68 = 44 (\text{blocks})$

- Chỉ mục ở mức thứ 2: $b_2 = b_1 / f_0 = 44/68 = 1 (\text{blocks})$.

⇒ Chỉ mục đa mức dừng lại ở mức $t = 2$

⇒ Số block trung bình để tìm kiếm một giá trị bất kỳ dựa trên chỉ mục đa cấp là: $t + 1 = 2 + 1 = 3 (\text{blocks})$.

- Chỉ mục đa mức là hình thức của việc tìm kiếm trên cây, trong đó mỗi node sẽ chứa thông tin giá trị của thuộc tính được chỉ mục và dùng nó để tìm đến bộ dữ liệu cụ thể
- Tuy nhiên, khi thêm và xóa các index entries sẽ gây ra một số vấn đề cần tổ chức lại chỉ mục, vì mỗi mức chỉ mục đều là một file có thứ tự.
- Cải tiến: thiết kế lại chỉ mục đa cấp sao cho dành thêm không gian trống trên mỗi blocks để thêm các entries mới và được gọi là chỉ mục đa mức động (dynamic multilevel index).
- Cấu trúc dữ liệu được sử dụng cho chỉ mục đa mức động là B-tree và B⁺ - tree



4.0

BƯỚC 2.3 LỰA CHỌN CHỈ MỤC

- **Chi phí cho chỉ mục:**
 - Thêm một bộ vào data files → thêm một entry vào index file.
 - Cập nhật một bộ ở data files → cập nhật lại ở index file.
 - Cần không gian để lưu trữ index file
 - Có thể giảm hiệu quả thực thi trong quá trình tối ưu truy vấn.



4.0

BƯỚC 2.3 LỰA CHỌN CHỈ MỤC

- Hướng dẫn lựa chọn chỉ mục ('wish-list' of indexes):
 - 1) Không chỉ mục trên các quan hệ nhỏ.
 - 2) Chỉ mục cho khóa chính.
 - 3) Thêm secondary index trên khóa ngoại nếu nó được truy cập thường xuyên.
 - 4) Thêm secondary index cho bất kỳ thuộc tính nào thường xuyên truy cập.
 - 5) Thêm secondary index cho các thuộc tính:
 - a) Sử dụng cho phép kết
 - b) ORDER BY
 - c) GROUP BY
 - d) Các xử lý gắn liền với sự sắp xếp (Union, Distinct)



4.0

BƯỚC 2.3 LỰA CHỌN CHỈ MỤC

- 6) Thêm chỉ mục Secondary index cho các thuộc tính gắn với hàm kết hợp và sử dụng xây dựng hàm kết hợp.

Ví dụ: Select MÃCN, AVG(LƯƠNG)
 From NHÂNVIÊN
 Group by MÃCN

- 7) Thêm chỉ mục trên tập thuộc tính mà có thể ảnh hưởng đến kết quả của một kế hoạch.
- 8) Tránh chỉ mục trên các quan hệ/thuộc tính thường xuyên cập nhật.
- 9) Tránh chỉ mục trên thuộc tính nếu truy vấn trả về nhiều hơn 25% số bộ của quan hệ.
- 10) Tránh chỉ mục trên thuộc tính có kiểu chuỗi dài

Bước 2.3: Lựa chọn chỉ mục

- Các lưu ý cho kỹ thuật chỉ mục:

Hạn chế sử dụng chỉ mục cho hệ thống giao tác (transaction systems).

Sử dụng nhiều chỉ mục để giảm thời gian hồi đáp dữ liệu trong các hệ thống hỗ trợ ra quyết định.

Trong mỗi bảng dữ liệu, nên tạo chỉ mục duy nhất (unique index) cho tập thuộc tính khóa chính.

Trong mỗi bảng dữ liệu, nên tạo chỉ mục cho thuộc tính khóa ngoại để tăng hiệu quả phép kết (join).

Tạo chỉ mục cho các thuộc tính mà được sử dụng thường xuyên trong gom nhóm (grouping), sắp xếp (sorting) và điều kiện tìm kiếm (criteria).

Bước 2.3: Lựa chọn chỉ mục

- Các lưu ý cho kỹ thuật dùng chỉ mục:

Không chỉ mục trên các quan hệ nhỏ

Nên thêm chỉ mục cho bất kỳ thuộc tính nào thường xuyên truy cập

Tránh chỉ mục trên các quan hệ/thuộc tính thường xuyên cập nhật.

Tránh chỉ mục trên thuộc tính nếu truy vấn trả về nhiều hơn 25% số bộ của quan hệ.

Tránh chỉ mục trên thuộc tính có kiểu chuỗi dài

Bước 2.4 Ước lượng không gian đĩa

- Mục đích: Ước lượng kích thước không gian đĩa yêu cầu để lưu trữ CSDL
- Cần xem xét:
 - Kích thước một block
 - Kích thước record
 - Số lượng thuộc tính
 - Kiểu dữ liệu thuộc tính và chiều dài khai báo
 - ...
 - Số lượng record được lưu trên block
 - Số lượng record phải lưu trên data file
 -

Bước 2.4: Ước lượng không gian đĩa

Bảng VIII - 9. Ước lượng kích thước cho bảng NHÀ_THUÊ

Thuộc tính	Kích thước trung bình
Mã_Nhà	8
ĐC_Nhà	24
Giá_thuê	4
Mã_CN	8
Mã_loại_nhà	5
Kích thước 1 bộ dữ liệu (record size)	49
Chi phí	30%
Tổng kích thước cho 1 bộ dữ liệu	63.7
Số bộ dữ liệu ban đầu trong bảng	50.000
Kích thước khởi tạo cho bảng	3.185.000
Tỉ lệ bộ tăng trưởng/1 tháng	1000 (bộ)/tháng
Kích thước của bảng dữ liệu sau 3 năm	5.478.200



4.0

Các bước thiết kế CSDL mức vật lý

Bước 1: Chuyển mô hình dữ liệu ở mức logic sang một HQT CSDL đích cụ thể

- 1.1 Thiết kế các quan hệ cơ sở (bảng dữ liệu cơ sở)
- 1.2 Thiết kế hình thức của dữ liệu suy diễn
- 1.3 Thiết kế các ràng buộc của doanh nghiệp

Bước 2: Thiết kế thể hiện vật lý

- 2.1 Phân tích truy vấn/giao tác
- 2.2 Chọn tổ chức lưu trữ tập tin (File Organization)
- 2.3 Lựa chọn chỉ mục
- 2.4 Ước lượng yêu cầu không gian đĩa

Bước 3: Thiết kế các khung nhìn người dùng

Bước 4: Thiết kế kỹ thuật bảo mật

Bước 5: Xem xét việc quản lý dư thừa

Bước 6: Giám sát và điều chỉnh hệ thống xử lý

Bước 3 Thiết kế khung nhìn

Objective

To design the user views that were identified during the requirements collection and analysis stage of the database application lifecycle.

```
CREATE VIEW Staff1_View  
AS SELECT staffNo, name, position  
FROM Staff  
WHERE branchNo = 'B001';
```

Staff1_View

staffNo	name	position
S1500	Tom Daniels	Manager
S0003	Sally Adams	Assistant



4.0

Các bước thiết kế CSDL mức vật lý

Bước 1: Chuyển mô hình dữ liệu ở mức logic sang một HQT CSDL đích cụ thể

- 1.1 Thiết kế các quan hệ cơ sở (bảng dữ liệu cơ sở)
- 1.2 Thiết kế hình thức của dữ liệu suy diễn
- 1.3 Thiết kế các ràng buộc của doanh nghiệp

Bước 2: Thiết kế thể hiện vật lý

- 2.1 Phân tích truy vấn/giao tác
- 2.2 Chọn tổ chức lưu trữ tập tin (File Organization)
- 2.3 Lựa chọn chỉ mục
- 2.4 Ước lượng yêu cầu không gian đĩa

Bước 3: Thiết kế các khung nhìn người dùng

Bước 4: Thiết kế kỹ thuật bảo mật

Bước 5: Xem xét việc quản lý dư thừa

Bước 6: Giám sát và điều chỉnh hệ thống xử lý



Bước 4 Thiết kế các kỹ thuật bảo mật

Objective

To design the security measures for the database as specified by the users during the requirements collection and analysis stage of the database application lifecycle.

```
GRANT privilege_name  
ON object_name  
TO {user_name |PUBLIC |role_name}  
[WITH GRANT OPTION];
```

```
REVOKE privilege_name  
ON object_name  
FROM {user_name |PUBLIC  
      |role_name}
```

```
GRANT ALL PRIVILEGES  
ON Staff  
TO Manager WITH GRANT OPTION;
```

```
GRANT EXECUTE  
ON Staff  
TO Admin;
```

```
GRANT SELECT  
ON Staff1_View  
TO Admin;
```

```
CREATE ROLE testing  
[IDENTIFIED BY pwd];
```

```
GRANT CREATE TABLE TO testing;
```

```
REVOKE CREATE TABLE FROM testing;
```



4.0

Các bước thiết kế CSDL mức vật lý

Bước 1: Chuyển mô hình dữ liệu ở mức logic sang một HQT CSDL đích cụ thể

- 1.1 Thiết kế các quan hệ cơ sở (bảng dữ liệu cơ sở)
- 1.2 Thiết kế hình thức của dữ liệu suy diễn
- 1.3 Thiết kế các ràng buộc của doanh nghiệp

Bước 2: Thiết kế thể hiện vật lý

- 2.1 Phân tích truy vấn/giao tác
- 2.2 Chọn tổ chức lưu trữ tập tin (File Organization)
- 2.3 Lựa chọn chỉ mục
- 2.4 Ước lượng yêu cầu không gian đĩa

Bước 3: Thiết kế các khung nhìn người dùng

Bước 4: Thiết kế kỹ thuật bảo mật

Bước 5: Xem xét việc quản lý dư thừa

Bước 6: Giám sát và điều chỉnh hệ thống xử lý

Bước 5: Quản lý sự thừa dữ liệu

DẠNG CHUẨN 0

Có bảng dữ liệu nào có các thuộc tính bị lặp? Có bộ dữ liệu nào trong bảng có số thuộc tính khác với các bộ khác?

Có: Xóa các trường lặp. Thêm một bảng mới chứa các thuộc tính bị lặp.

Không: Lược đồ đạt DC1

DẠNG CHUẨN 1

Khóa chính của bảng dữ liệu bao gồm nhiều thuộc tính? Nếu có, có thuộc tính không khóa nào của bảng chỉ phụ thuộc vào một phần của khóa chính?

Có: Xóa phụ thuộc thành phần vào khóa. Thêm một bảng mới chứa các thuộc tính tạo ra phụ thuộc một phần vào khóa chính.

Không: Lược đồ đạt DC2

DẠNG CHUẨN 2

Có thuộc tính không khóa nào của bảng dữ liệu phụ thuộc lẫn nhau?

Có: Xóa các phụ thuộc giữa các thuộc tính không khóa. Thêm một bảng mới chứa các thuộc tính tạo ra phụ thuộc này.

Không: Lược đồ đạt DC3

DẠNG CHUẨN 3



4.0

Bước 5: Quản lý sự dư thừa dữ liệu

- Mục tiêu: Xác định liệu có nên lưu trữ dư thừa (bỏ qua tiêu chí dạng chuẩn) để tăng hiệu quả xử lý của hệ thống.
- Các công việc gồm có:
 - 5.1) Gộp 2 bảng có mối quan hệ 1-1
 - 5.2) Lưu trữ trùng thuộc tính không khóa trong mối quan hệ 1-n (liên quan đến giảm phép kết)
 - 5.3) Lưu trữ trùng thuộc tính khóa trong mối quan hệ 1-n (liên quan đến giảm phép kết)
 - 5.4) Lưu trữ trùng mỗi quan hệ n-n (mục tiêu giảm phép kết)
 - 5.5) Nhóm lặp
 - 5.6) Kết nối các bảng tìm kiếm dựa trên các quan hệ cơ sở
 - 5.7) Tạo các bản rút trích



Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- Lược đồ CSDL cho thuê nhà
 - 1) Client (clientNo, fname, lname, telNo, maxRent)
 - 2) Interview (clientNo, staffNo, dateInterview, comment)
 - 3) Viewing (clientNo, propertyNo, viewDate, comment)
 - 4) PropertyForRent (propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)
 - 5) Owner (ownerNo, fname, lname, address, telNo)
 - 6) Branch (branchNo, street, city, postcode)
 - 7) TelephoneBranch (telNo, branchNo)
 - 8) PropertyType (type, description)



Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- ❑ Lưu trữ trùng lặp thuộc tính trên mỗi kết hợp 1-1
- ❑ Ví dụ: gộp 2 quan hệ Client và Interview thành một quan hệ ClientInterview
 - 1) Client(clientNo, fname, lname, telNo, maxRent)
 - 2) Interview(clientNo, staffNo, dateInterview, comment)

ClientInterview (clientNo, fname, lname, telNo, maxRent, staffNo, dateInterview, comment) • •

Lưu ý giá trị NULL



Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- ❑ Lưu trữ trùng lặp thuộc tính không khóa trên mỗi kết hợp 1-n
- Ví dụ: Cho biết thông tin nhà cho thuê và tên ông chủ nhà
 - 1) PropertyForRent(propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)
 - 2) Owner(ownerNo, fname, lname, address, telNo)

PropertyForRent

<u>propertyNo</u>	OwnerNo	lName	staffNo	branchNo
-------------------	-----	-----	-----	-----	---------	-------	---------	----------



Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- ❑ Lưu trữ trùng lặp thuộc tính khóa trên mỗi kết hợp 1-n
- Ví dụ: Cho biết thông tin tất cả các chủ nhà liên quan đến chi nhánh B003
 - 1) PropertyForRent(propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)
 - 2) Owner(ownerNo, fname, lname, address, telNo)

Owner

<u>OwnerNo</u>	branchNo
----------------	-----	-----	-----	-----	----------

Chủ nhà
thuộc về
nhiều chi
nhánh?



Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- Lưu trữ trùng lặp thuộc tính trên mỗi kết hợp **n-n**
- Ví dụ: Cho biết thông tin khách xem nhà (mã người thuê, mã căn nhà, địa chỉ căn nhà)
 - 1) Client(clientNo, fname, lname, telNo, maxRent)
 - 2) Viewing(clientNo, propertyNo, viewDate, comment)
 - 3) PropertyForRent(propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo)

Viewing

<u>clientNo</u>	<u>propertyNo</u>	street	viewDate	comment
-----------------	-------------------	--------	----------	---------



Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

- Lưu trữ theo nhóm lặp
- Ví dụ: Giả sử mỗi chi nhánh có tối đa 3 điện thoại \Rightarrow gộp bảng điện thoại vào chi nhánh
 - 1) Branch(branchNo, street, city, postcode)
 - 2) TelephoneBranch(telNo, branchNo)

Branch

branchNo	street	city	postcode	telNo1	telNo2	telNo1
----------	--------	------	----------	--------	--------	--------



Tối ưu hóa về tốc độ truy cập: Phi chuẩn hóa

□ Gộp quan hệ

- Ví dụ: Giả sử có truy xuất dữ liệu thường xuyên truy xuất trên thông tin nhà và mô tả loại nhà với tần xuất rất lớn
⇒ gộp bảng nhà cho thuê và loại nhà

1) PropertyForRent (propertyNo, street, city, postcode, rooms, rent, ownerNo, staffNo, branchNo, type)

2) PropertyType (type, description)

PropertyForRent

propertyNo	type	discription	OwnerNo	lName	staffNo	branchNo
------------	-----	-----	------	-------------	---------	-------	---------	----------



4.0

Bước 5: Quản lý sự dư thừa dữ liệu

Tạo các bảng rút trích

- **Mục tiêu:** Phục vụ cho việc tạo các báo biểu ở các thời điểm với tần suất cao trong ngày. Các báo biểu này truy cập đến dữ liệu suy diễn và thực thi trên sự kết nối nhiều quan hệ.

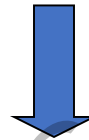


4.0

Bước 5: Quản lý sự thừa dữ liệu

•Tách dọc:

KHÁCH_HANG(MÃ_KH, TÊN_KH, DCHI, DT, TAIKHOAN, MASOTHUE, PTTTOAN, MUC_NO, CONG_NO)



KH1(MA_KH, TEN_KH, DCHI)

KH2(MÃ_KH,DT, TAIKHOAN, MASOTHUE, PTTTOAN, MUC_NO, CONG_NO)

Cấu trúc truy cập thường xuyên

Cấu trúc truy cập không thường xuyên

Bước 5: Quản lý sự dư thừa dữ liệu

84

- Tách ngang:

HOA_DON

KL: ~10.000.000/năm

Các xử lý truy cập dữ liệu

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyyy	2.000.000
....			
Hd15000	1/1/05	Zxzxzzxzx	1.400.000
Hd15001	2/1/05	Qqqqqqqq	2.100.000
...			
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdsdsds	1.000.000

Mã số	Tên xử lý	Tần suất
O1	Tìm hóa đơn	100/ngày
O2	Tính doanh thu tháng	1/tháng
O3	Tính doanh thu theo khách hàng	100/tháng
O4	Tổng hợp doanh số năm	1/năm
O5	Lập biểu đồ so sánh doanh số theo các năm	1/năm

Bước 5: Quản lý sự dư thừa dữ liệu

• Ví dụ tách ngang:

KL: ~10.000.000/năm

HOA_DON

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyyy	2.000.000
....			
Hd15000	1/1/05	Zxzxzzxzx	1.400.000
Hd15001	2/1/05	Qqqqqqqq	2.100.000
...			
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdsdsds	1.000.000

Tách table HOA_DON theo năm
Các phép toán o1, o2, o4 hiệu quả hơn

HD004

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd00001	1/1/04	Xxxxxxx	1.000.000
Hd00002	2/1/04	Yyyyyyy	2.000.000

HD005

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd15000	1/1/05	Zxzxzzxzx	1.400.000
Hd15001	2/1/05	Qqqqqqqq	2.100.000

HD006

Số_HD	Ngày_HD	Diễn_giải	Trị giá
Hd30000	2/1/06	Asasasas	12.000.000
Hd30001	2/1/06	Dsdsdsds	1.000.000

Các bước thiết kế CSDL mức vật lý

Bước 1: Chuyển mô hình dữ liệu ở mức logic sang một HQT CSDL đích cụ thể

- 1.1 Thiết kế các quan hệ cơ sở (bảng dữ liệu cơ sở)
- 1.2 Thiết kế hình thức của dữ liệu suy diễn
- 1.3 Thiết kế các ràng buộc của doanh nghiệp

Bước 2: Thiết kế thể hiện vật lý

- 2.1 Phân tích truy vấn/giao tác
- 2.2 Chọn tổ chức lưu trữ tập tin (File Organization)
- 2.3 Lựa chọn chỉ mục
- 2.4 Ước lượng yêu cầu không gian đĩa

Bước 3: Thiết kế các khung nhìn người dùng

Bước 4: Thiết kế kỹ thuật bảo mật

Bước 5: Xem xét việc quản lý dư thừa

Bước 6: Giám sát và điều chỉnh hệ thống xử lý



Bước 6: Giám sát và điều chỉnh hệ thống xử lý

- Mục tiêu:
 - Giám sát hệ thống xử lý và cải tiến hiệu quả thực thi của hệ thống.
 - Chỉnh sửa các quyết định thiết kế không phù hợp hoặc do thay đổi của yêu cầu nghiệp vụ.

