



## Chương 12

---

# Thiết kế vật lý database



# Nội dung

---

1. Khái niệm
2. Quá trình thiết kế vật lý cơ sở dữ liệu
3. Thiết kế các vùng tin
4. Thiết kế các bản ghi vật lý
5. Thiết kế tập tin vật lý

# Khái niệm

---

- Cơ sở dữ liệu (database):
  - Tập hợp thông tin có cấu trúc
  - Tập hợp các dữ liệu được lưu trữ trên một thiết bị lưu trữ
  - Tập hợp các tập tin được quản trị bởi một hệ quản trị dữ liệu
  - Là kho chứa dữ liệu
- Mô hình cơ sở dữ liệu (database model):
  - Bản thiết kế kho dữ liệu
  - Mô hình mô tả cách thức lưu trữ dữ liệu

# Khái niệm

---

- ❑ Thiết kế dữ liệu là tiến trình:
  - Tạo ra các định nghĩa dữ liệu cho hệ thống;
  - Thiết lập cấu trúc các tệp dữ liệu chính trong hệ thống
- ❑ Thiết kế cơ sở dữ liệu:
  - Là tiến trình xây dựng cơ sở dữ liệu cho hệ thống với kết quả là cơ sở dữ liệu trên giấy (không có trên thực tế!)
  - “Thiết kế cơ sở dữ liệu chiếm phần nhỏ chi phí so với xây dựng hệ thống, nhưng giúp kiểm soát hệ thống trước khi làm hỏng nó trong quá trình xây dựng”.

# Thiết kế database

---

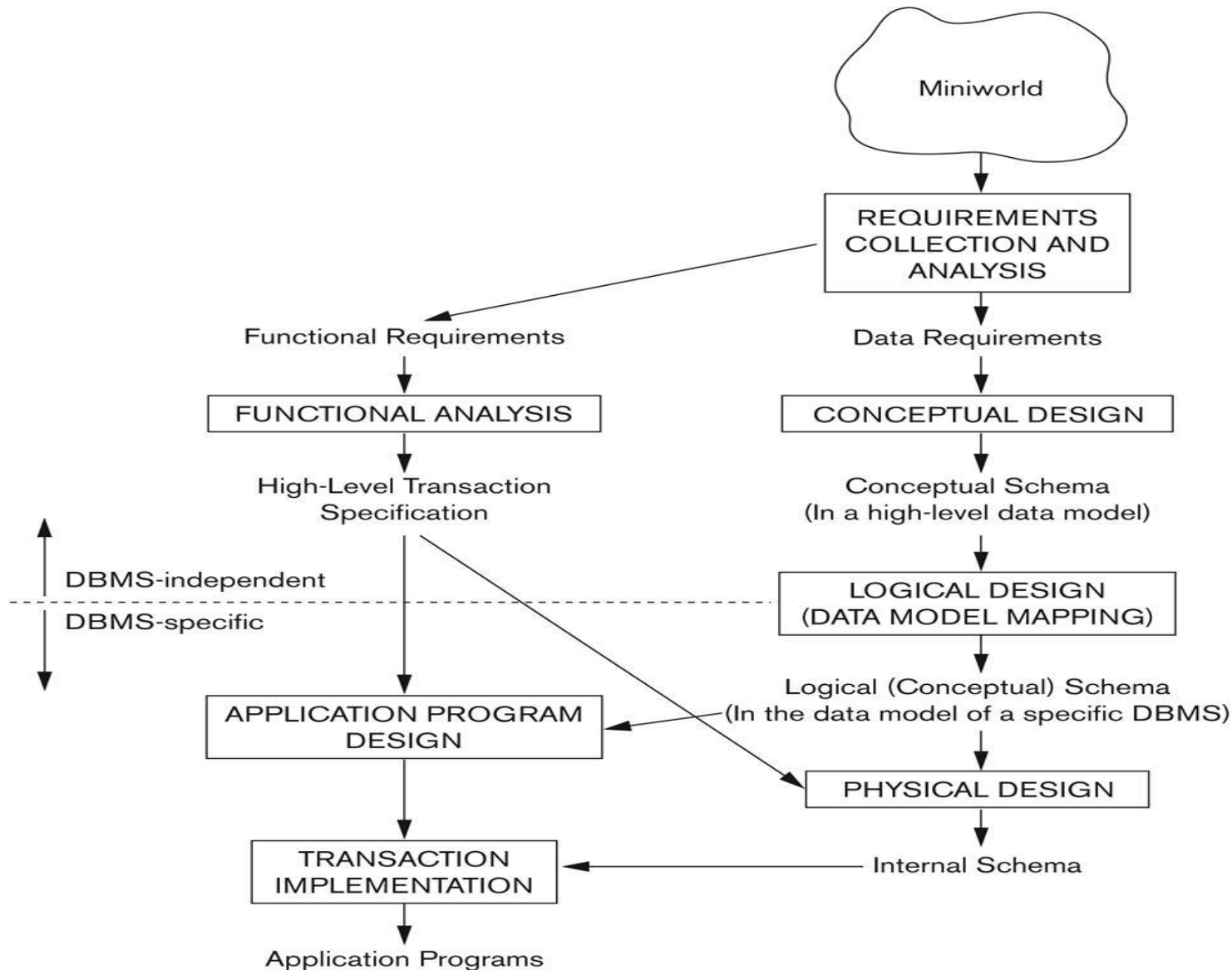
- Thiết kế cơ sở dữ liệu vật lý là quá trình chuyển các đặc tả dữ liệu logic thành các đặc tả kỹ thuật để lưu trữ dữ liệu. Gồm 2 nội dung sau:
  - Lựa chọn công nghệ lưu trữ (Hệ điều hành, HQTCSDL, các công cụ truy nhập dữ liệu).
  - Chuyển các quan hệ của mô hình logic thành các thiết kế vật lý.
- Yêu 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

# Overview of Database Design Process



# Phương pháp thiết kế CSDL

---

1. **Phân tích các yêu cầu:** Thu thập dữ liệu thực, các yêu cầu đặc biệt, đầu ra mong muốn. Thông tin sử dụng để phân tích phải đầy đủ và chính xác.
2. **Thiết kế mức logic:** Thiết lập các trường dữ liệu (fields), bảng dữ liệu (tables) và mối quan hệ (relationships) giữa các bảng. Các bảng dữ liệu ở dạng chuẩn. Xây dựng các câu lệnh tạo các bảng dữ liệu.
3. **Thiết kế mức vật lý:** Xây dựng các bảng vật lý phù hợp với hệ quản trị cơ sở dữ liệu được chọn.
4. **Bước hoàn thiện:** Lập chỉ mục, xem xét lại các dạng chuẩn, yêu cầu bảo mật và các vấn đề khác.



# Phân tích yêu cầu

---

## □ Mục tiêu khách quan:

- Những thông tin gì sẽ có trong CSDL?
- Kết quả mong đợi là gì?
- Xây dựng hệ thống mới hay sửa chữa hệ thống cũ?

## □ Hoạt động trong công ty:

- Những công việc thực hiện theo quy trình cũ?
- Cách thức xử lý trên máy tính?

## □ Quy tắc nghiệp vụ:

- Mô tả các vấn đề đã được phân tích;
- Mô tả danh sách các bảng dữ liệu;
- Mô tả các quan hệ cơ sở giữa các bảng.

# Phân tích yêu cầu

---

## □ Mô hình dữ liệu:

- 1) Mô hình thực thể liên kết,
- 2) Mô hình dữ liệu quan hệ,
- 3) Sơ đồ dòng dữ liệu hệ thống và
- 4) Mô hình lịch sử đòi thực thể.

## ■ Thông tin về những ràng buộc thực hiện:

1. phần cứng, phần mềm được sử dụng,
2. thời gian đáp ứng các yêu cầu,
3. điều kiện kiểm soát,
4. điều kiện an toàn của hệ thống.

# Phân tích yêu cầu

---

- 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
  - Các mô tả về công nghệ được sử dụng để triển khai file và CSDL (thiết bị lưu trữ, hệ điều hành, HQTCSDL...)

# Lưu trữ thông tin

---

- ❑ Đơn vị lưu trữ cơ sở là bản ghi, trong đó mỗi bản ghi được cấu tạo từ các trường.
- ❑ Nếu quan hệ được tổ chức lưu trữ như là một tệp dữ liệu thì mỗi dòng trong quan hệ sẽ là một bản ghi trong tệp dữ liệu.
- ❑ Nếu mỗi kiểu thực thể được tổ chức lưu trữ như một tệp dữ liệu thì mỗi thực thể thành một bản ghi.

# Quá trình thiết kế database

1. Chọn kiểu dữ liệu cho mỗi thuộc tính có mặt trong mô hình dữ liệu luận lý: kiểu dữ liệu ít tốn bộ nhớ mà vẫn bảo đảm tính toàn vẹn dữ liệu
2. Nhóm các thuộc tính từ mô hình dữ liệu luận lý vào các bản ghi vật lý (physical record)
3. 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 mau chóng.
  - Cần quan tâm đến việc bảo vệ và khôi phục dữ liệu khi có lỗi
4. Chọn cấu trúc lưu trữ và kết nối các file để việc truy xuất dữ liệu hiệu quả hơn.
5. Chuẩn bị các chiến lược để quản lý các truy vấn sao cho các truy vấn được tối ưu khi thực thi

# Quy trình thiết kế

---

- Chuyển mô hình mô tả thông tin trong bước phân tích hệ thống thành các cấu trúc dữ liệu phù hợp cho việc lưu trữ, khai thác trong các tiến trình hệ thống.
  1. Thiết kế mô hình dữ liệu logic
  2. Chuẩn hóa dữ liệu (tối thiểu sự dư thừa)
  3. Phân tích lịch sử vòng đời thực thể. Phân tích dữ liệu vào –ra.
  4. Thiết kế dữ liệu vật lý (thiết kế các tệp bảng dữ liệu và hệ quản trị dữ liệu)
  5. Kiểm tra thiết kế: tính toàn vẹn, tính an toàn. Phân tích các nguy cơ. Thử nghiệm.

# Sản phẩm thiết kế

---

1. Mô hình dữ liệu logic
2. Mô hình lịch sử vòng đời thực thể
3. Danh sách mô tả các bản ghi, tệp.
4. Từ điển dữ liệu

# Ý tưởng thiết kế

---

- Phân định công việc người-máy (xử lý tự động trên máy tính hay xử lý thủ công bởi con người, hoặc xử lý bán thủ công, hoặc kết hợp cả hai phương thức xử lý).
- Xem xét phương thức xử lý: như xử lý tương tác (interactive processing), xử lý giao dịch (transaction processing), xử lý trực tuyến (on-line processing), xử lý theo lô (batch processing), xử lý phân tán (distributed processing) và xử lý thời gian thực (real-time processing).





# Lựa chọn mô hình

---

- Mô hình dữ liệu quan hệ (Relational Database Model),
- Mô hình thực thể liên kết (Entity-Relationship Model),
- Thiết kế hướng đối tượng (Object-Oriented Design),
- Kiến trúc khách-chủ (client/server architecture).

# Chất lượng thiết kế

---

- Dữ liệu được tổ chức hợp lý, nghĩa là phải đủ sử dụng, không dư thừa.
- Dữ liệu được tổ chức thuận lợi cho việc truy cập (tìm kiếm, cập nhật, bổ sung và loại bỏ).
- Để xem xét tính tối ưu của một cơ sở dữ liệu thường phải xem xét ở mức lược đồ khái niệm (logic). Ở mức vật lý (tệp, bảng dữ liệu) việc xem xét tính tối ưu là khó khăn.

# Ví dụ hệ thống quản lý thư viện

---

## Quản lý bạn đọc

1. Sau khi nhập học, sinh viên được yêu cầu làm thẻ bạn đọc. Thông tin về bạn đọc - sinh viên được lưu trữ trong kho dữ liệu nhằm phục vụ cho công tác quản lý.
2. Sau mỗi năm học, ban quản lý xét gia hạn sử dụng thẻ bạn đọc, trên cơ sở xem xét việc sinh viên phải thực hiện đúng các quy định của thư viện là trả sách đúng hạn, nếu làm hỏng hoặc mất sách thì đền sách theo quy định. Trong trường hợp bạn đọc không yêu cầu gia hạn thì huỷ thẻ bạn đọc hoặc vi phạm quy định của thư viện sẽ bị phạt trong 1 năm không được xét cấp thẻ.
3. Việc tra cứu, tìm kiếm thông tin bạn đọc phục vụ cho ban quản lý, thủ thư và bạn đọc.

# Ví dụ hệ thống quản lý thư viện

---

## Quản lý sách

1. Hàng năm thư viện phải lập kế hoạch bổ sung sách mới, dựa trên yêu cầu của các khoa và thống kê những sách có nhiều bạn đọc yêu cầu. Sách mới bổ sung được thư viện được phân loại, ghi số cá biệt, từ khoá, tên sách, tên tác giả, nhà xuất bản, tóm tắt nội dung và được cập nhật vào kho thông tin.
2. Thống kê, tìm kiếm sách theo số cá biệt, tên sách hoặc theo tên tác giả, theo từ khoá để phục vụ bạn đọc mượn sách và trả sách theo dõi sách.
3. Cập nhật thông tin về xử lý sách hỏng, sách cũ cần thanh lý, sách bị mất.
4. Hỗ trợ bạn đọc tra cứu, tìm kiếm thông tin về sách trước khi đến làm thủ tục mượn.

# Ví dụ hệ thống quản lý thư viện

---

## Theo dõi hoạt động mượn trả

### Mượn sách:

1. Bạn đọc điền vào phiếu mượn tên sách, số cá biệt, tác giả những quyển sách cần mượn, số thẻ, ngày mượn và nộp phiếu mượn cho thủ thư. Bạn đọc có thể đặt mượn qua mạng.
2. Sau khi thủ thư kiểm tra thông tin về bạn đọc (đúng chủ số thẻ, còn hạn sử dụng, số lượng sách đang mượn ít hơn số cho phép, không có sách mượn quá hạn) thì kiểm tra tiếp thông tin về sách mà bạn đọc yêu cầu. Nếu mọi yêu cầu đều được đáp ứng bạn đọc sẽ được thông báo đợi để mượn sách.
3. Tại kho, căn cứ vào phiếu mượn được thủ thư chuyển xuống, bộ phận quản lý kho giao sách cho bạn đọc và xác nhận thông tin mượn vào *Sổ theo dõi bạn đọc* và thay đổi trạng thái sách trong kho sách.

# Ví dụ hệ thống quản lý thư viện

---

## Theo dõi hoạt động mượn trả

### Trả sách:

Thủ thư ghi ngày trả, xác nhận vào *Sổ theo dõi bạn đọc* nếu sách không bị hư hại. Trong trường hợp sách bị hư hỏng hoặc mất thì làm thủ tục phạt hoặc đền sách.

Hàng tháng, thư viện xem xét những quyển sách bị mượn quá hạn và lập phiếu đòi sách quá hạn. Những quyển nào đã đòi quá ba lần thì được coi như đã mất.

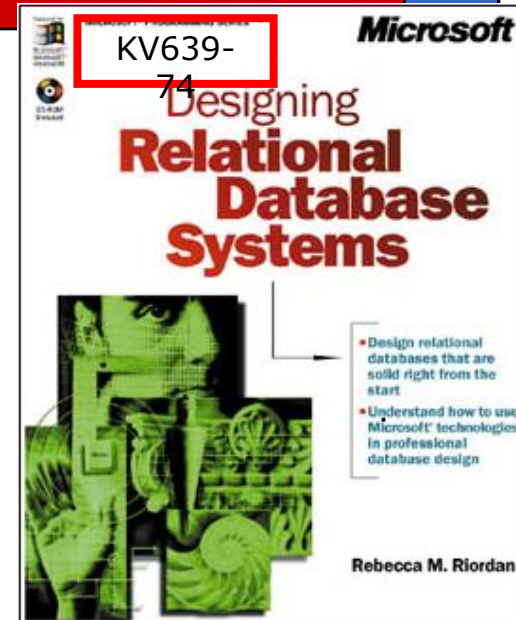
### Thống kê, tìm kiếm phục vụ thủ thư và bạn đọc:

- Tìm bạn đọc mượn sách lần sau cùng hoặc đã từng mượn quyển sách;
- Tìm những sách quá hạn chưa trả và in phiếu báo đòi;
- Thống kê những sách có nhiều người mượn.

Thống kê những sách đang được mượn.

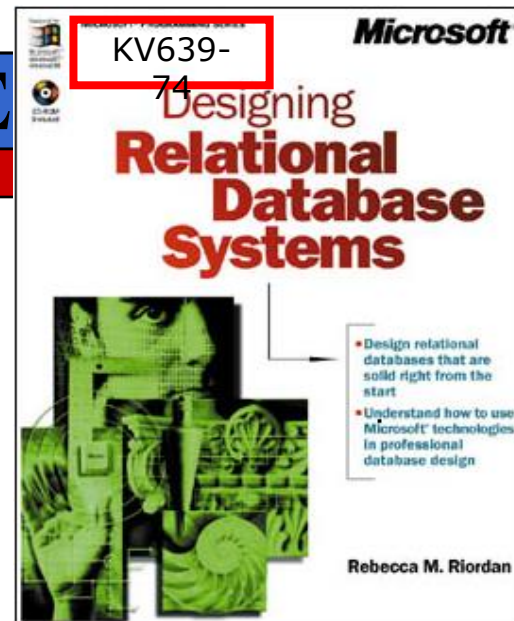
# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Quản lý sách - Mô hình 1



ID	Tên sách	Tác giả	Nhà xuất bản
KV639-74	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-75	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-76	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-77	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft

# HỆ THỐNG QUẢN LÝ THƯ VIỆN



## Hoạt động mượn trả - Mô hình 1



ID	Tên sách	Tác giả	Nhà xuất bản
KV639-74	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-75	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-76	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-77	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft

Bạn đọc tra cứu thông tin về sách và điền phiếu mượn

→Phiếu mượn cần lưu những dữ liệu nào, ai là người mượn sách



# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Hoạt động mượn trả - Mô hình 1



Nộp phiếu mượn và đợi  
thông tin phản hồi

Các tình huống không  
được mượn:

1. Số thẻ không đúng;
2. Thẻ hết hạn sử dụng;
3. Nợ sách quá hạn chưa trả;
4. Không còn sách trong kho;

# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Hoạt động mượn trả - Mô hình 1



Nộp phiếu mượn và đợi  
thông tin phản hồi



Nếu được mượn, bạn đọc  
làm thủ tục mượn tại kho

# HỆ THỐNG QUẢN LÝ THƯ VIỆN

Xét mô hình quản lý sách 1 và  
hoạt động mượn trả 1

**BẠN ĐỌC  
(BD)**

Số thẻ  
Họ tên  
Đơn vị

.....  
Hạn SD (sử dụng)

**SÁCH**

Số CB (cá biệt)  
Tên sách  
Tên TG (tác giả)  
Năm XB (xuất bản)  
Nhà XB  
TTND (tóm tắt nội dung)  
Từ khoá  
CK (có/không)

**MƯỢN/TRẢ**

Số thẻ, Số CB  
Ngày mượn  
Ngày HT (hẹn trả)  
Ngày trả  
Tình trạng

# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Ví dụ 1 bản ghi SÁCH, BẠN ĐỌC và MƯỢN/TRẢ

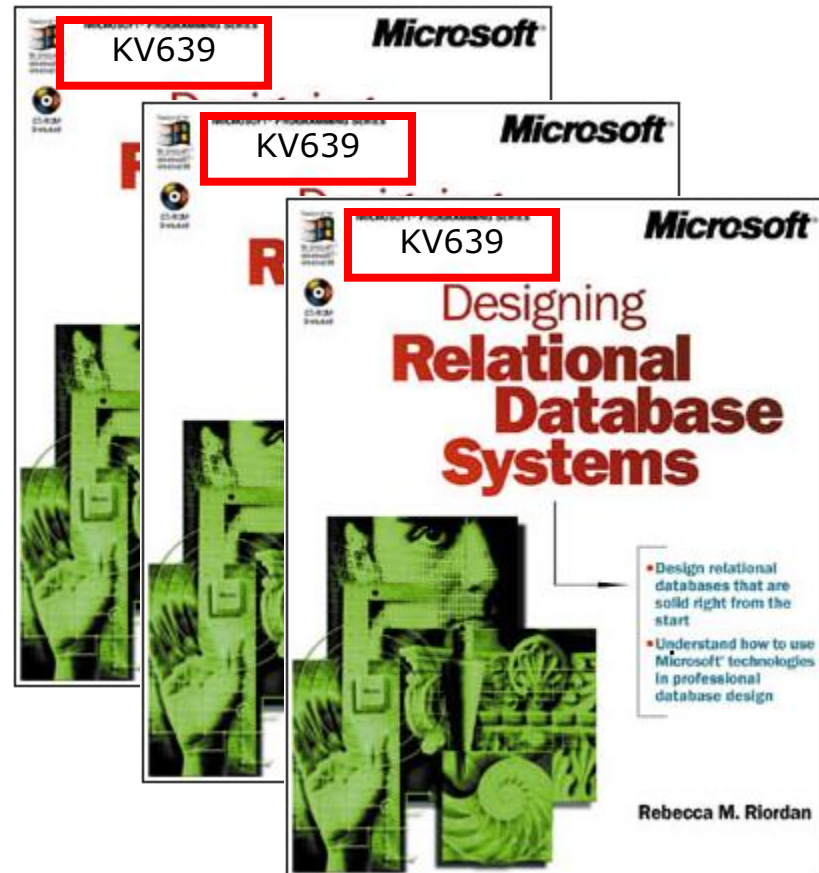
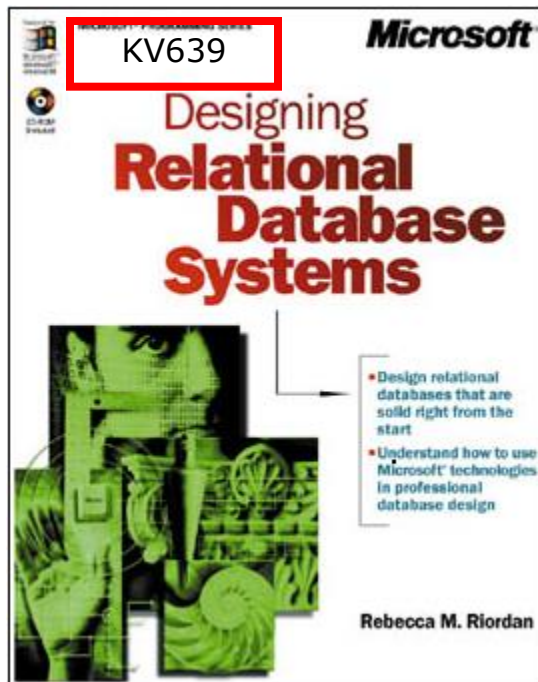
Số CB	<b>KV639-74</b>
Tên sách	Designing Relational Database Systems
Tên TG	Rebecca M. Riordan
Năm XB	1999
Nhà XB	Microsoft
TTND	The book is divided into three sections. Part 1, Relational Database Theory, covers the fundamental principles of the relational model. This is where the really ugly, theoretical stuff is. But don't worry; it will get easier. Part 2, Designing Relational Database Systems, examines the analysis and design process—what you should do to get from the real world to a reliable database system design. Part 3 discusses the most important aspect of a database system from a user's point of view: the user interface.
Từ khoá	relational databases, database systems, database designing, relational model, design process, user interface, normal form
CK	1

Số thẻ	<b>12_321</b>
Họ tên	Hoàng Văn Trọng
Đơn vị	TH_7A
Hạn SD	31/12/2012

Số thẻ	<b>12_321</b>
Số CB	<b>KV639-74</b>
Ngày mượn	23/08/2010
Ngày HT	23/09/2010
Ngày trả	15/09/2010
Tình trạng	Tốt

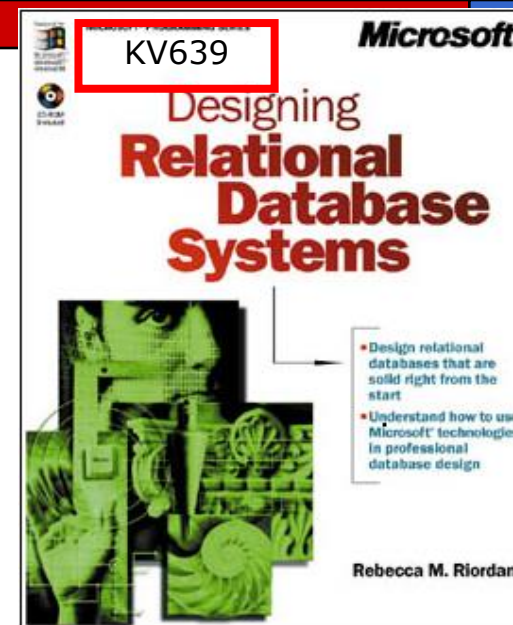
# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Quản lý sách - Mô hình 2



# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Quản lý sách - Mô hình 2



ID	Tên sách	Tác giả	Nhà xuất bản	Số lượng
KV639	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft	10
KV640	Introduction to Relational Database Design	Fernando Lozano	Internet Edition	2
KV641	An Introduction to Database Systems	Date C.J.	Addison-Wesley Publishing Company	5
KV642	Systems Analysis and Design	Hawryszkiewycz I.T.	Univ. of Technology Sydney, Prentice Hall Astralia	1

# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Hoạt động mượn trả - Mô hình 2

ID	Tên sách	Tác giả	Nhà xuất bản
KV639-74	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-75	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-76	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft
KV639-77	Designing Relational Database Systems	Rebecca M. Riordan	Microsoft

Bạn đọc vào  
kho, tự tìm  
sách cần  
thiết



# HỆ THỐNG QUẢN LÝ THƯ VIỆN

## Hoạt động mượn trả - Mô hình 2

Sau đó làm  
thủ tục mượn





# HỆ THỐNG QUẢN LÝ THƯ VIỆN



## BẠN ĐỌC (BD)

Số thẻ

Họ tên

Đơn vị

.....

Hạn SD (sử dụng)

## SÁCH

Số CB (cá biệt)

Tên sách

Tên TG (tác giả)

Năm XB (xuất bản)

Nhà XB

TTND (tóm tắt nội dung)

Từ khoá

CK (có/không)

## MƯỢN/TRẢ

Số thẻ, Số CB

Ngày mượn

Ngày HT (hẹn trả)

Ngày trả

Tình trạng

Đủ lưu trữ thông tin  
cho các tình huống  
xảy ra hay chưa?

# Thiết kế trường

## (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 luận lý
- ❑ 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??

# Thiết kế trường (Field design)

## BẢNG MÔ TẢ CÁC TRƯỜNG

Loại đặc tả	Mô tả nội dung
Tên trường (field name)	Theo quy định về cách đặt tên trường của HQTCSDL.
Kiểu trường (data type)	Chọn kiểu dữ liệu mà HQTCSDL đó hỗ trợ
Kích cỡ (size)	Là kích thước tối đa dùng để lưu trữ dữ liệu của trường đó
Mã hóa (Coding)	Cách viết tắt giá trị của trường. Ví dụ, mỗi nước được biểu diễn bằng hai ký tự

# Thiết kế trường (Field design)

## BẢNG MÔ TẢ CÁC TRƯỜNG

Loại đặc tả	Mô tả nội dung
Các quy tắc toàn vẹn dữ liệu (data integrity rules)	Các đặc tả về các hạn chế đặt lên giá trị của trường
Các kiểm soát bảo trì (maintenance controls)	Chỉ ra những giá trị nào được phép thay đổi
Công thức (Formular)	Là kích thước tối đa Mô tả công thức tính toán giá trị với những trường số cần tính toán.
Toàn vẹn tham chiếu (references integrity)	Đặc tả giá trị của trường có liên quan đến giá trị của trường khác
Sở hữu (Ownership)	Ai là người sở hữu trường đó (có quyền đổi <sup>3B7</sup> với dữ liệu)



# Bốn mục tiêu để chọn kiểu dữ liệu

---

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

# Bốn mục tiêu để chọn kiểu dữ liệu

## Chọn kiểu và cách biểu diễn dữ liệu

- Các kiểu dữ liệu mà HQTCSDL SQL hỗ trợ và ý nghĩa của nó

DECIMAL(m,n)	Số thập phân có độ dài là m chữ số và n số thập phân
INTEGER	Số nguyên lớn (độ dài tối đa là 11 chữ số)
SMALLINT	Số nguyên nhỏ (độ dài tối đa là 6 chữ số)
FLOAT(m,n)	Số thực có độ dài là m chữ số và n số thập phân
CHAR	Xâu ký tự có độ dài là m ký tự
DATE	Kiểu dữ liệu thời gian và có rất nhiều cách biểu diễn
LOGICAL	Giá trị logic (đúng/sai)

# Bốn mục tiêu để chọn kiểu dữ liệu

---

## Các trường tính toán

- Khi giá trị của một trường là giá trị nhận được từ các giá trị của trường khác thì trường đó gọi là trường tính toán.

- Có các loại tính toán sau:

Tính toán số học:  $\text{Lương} = \text{Hệ số lương} * 210$ .

Tính toán logic: Tiền trợ cấp = 50.000 đ nếu cán bộ là nữ.  
0 nếu cán bộ là nam.

- Tính toán hỗn hợp:

$\text{Tiền điện} = \text{Số điện} * 500 \text{ đ}$  nếu số điện < 100.

$\text{Số điện} * 500 + (\text{Số điện} - 100) * 750$  nếu số điện > 100.



# 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.

# Các kỹ thuật mã hoá và nén dữ liệu

- ❑ **Mã hóa phân cấp:** để mô tả các dữ liệu phân cấp người ta dùng nhiều nhóm, mỗi nhóm đại diện cho cấp và các nhóm được sắp xếp lần lượt từ trái sang phải.
- ❑ Ví dụ: Hệ thống phân loại sách trong thư viện:

Các cấp			Mã số	Tên tài khoản
1	2	3		
500			500	Khoa học tự nhiên
	1		5001	Toán học
	2		5002	Vật lý
	1	1	50011	Toán giải tích
	1	2	50012	Toán rời rạc

# Các kỹ thuật mã hoá và nén dữ liệu

---

- ❑ **Mã liên tiếp:** Mã này được tạo ra theo quy tắc một dãy liên tục, như 1, 2, 3 ...A, B, C.... Mã loại này dùng cho những dữ liệu là danh sách như danh sách sinh viên. Nó đơn giản, dễ tự động hóa, không nhầm lẫn. Tuy nhiên nó không gợi nhớ về đối tượng được mã hóa và không cho phép chèn thêm vào giữa.
- ❑ **Mã gợi nhớ:** Căn cứ vào đối tượng được mã hóa để cấu tạo mã. Ví dụ: VND (Đồng Việt Nam), TL001 (Thủy lợi 001)...Loại này giúp ta nhận ra đối tượng được mã hóa, có thể nói rộng hoặc thu hẹp số lượng mã. Tuy nhiên khó tổng hợp và phân tích.

# Các kỹ thuật mã hoá và nén dữ liệu

- ❑ **Mã thành phần ngữ nghĩa:** Theo phương pháp này, mã được chia làm nhiều thành phần, mỗi phần mô tả một đặc trưng nhất định của đối tượng như phân loại, địa danh... Những phần này có thể sử dụng các nhóm ký tự khác nhau.
- ❑ Mã loại này rất thông dụng và được sử dụng nhiều trong công nghiệp cũng như giao tiếp quốc tế.
- ❑ Ví dụ: Địa chỉ miền trên internet có dạng:  
    <Tên tổ chức>.<Loại tổ chức>.<Tên nước>
- ❑ Ví dụ : **hui.edu.vn**: Đại học Công Nghiệp Tp.HCM, Tổ chức giáo dục, Tên nước
- ❑ Mã loại này công kênh, và cần chọn các thành phần sao cho ổn định, nếu không việc sử dụng mã sẽ gặp nhiều khó khăn.

# Các kỹ thuật mã hoá và nén dữ liệu

---

- ❑ Mã hoá bằng cách tạo 1 bảng tra cứu, sao cho mỗi giá trị của trường được thay thế bằng 1 mã
- ❑ Ví dụ: trường Finish của bảng Product chỉ có 1 ít giá trị là Birch, Maple và Oak.
  - giảm không gian lưu trữ cho trường Finish
  - Thêm không gian phụ cho bảng FINISH
  - Không có lợi khi Finish ít dùng hay số sản phẩm quá lớn
  - Bảng mã FINISH không xuất hiện trong mô hình nhận thức, là 1 thiết kế vật lý để cải thiện việc xử lý dữ liệu

# Các kỹ thuật mã hoá và nén dữ liệu

Product No	Description		
B100	Chair	C	
B120	Desk	A	
M128	Table	C	
T100	Bookcase	B	
...			

Bảng Product

Code	Value
A	Birch
B	Maple
C	Oak

Bảng tra cứu FINISH

# 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

# 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 - Giá trị ngầm định
  - Picture control - Kiểm tra khuôn dạng
  - Range control - Kiểm tra giới hạn
  - Null value control - Kiểm tra giá trị rỗng
  - Referential integrity - Tính toàn vẹn tham chiếu



# Giá trị mặc định

(Default value)

---

- Là giá trị được gán sẵn cho một trường nào đó khi bản ghi mới được nhập vào.
  - Giảm thời gian nhập liệu
  - Giảm những sai sót khi nhập liệu

Ví dụ: Trong hóa đơn bán hàng, trường ngày bán được mặc định là ngày hiện tại.

# Kiểm tra khuôn dạng

(picture control)

---

- Là mẫu định dạng bao gồm độ rộng, các giá trị có thể trong từng vị trí.
- Ví dụ: TLA006, \$999,999.99.

# 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
- ❑ Ví dụ: Điểm môn học được giới hạn là các số và được giới hạn từ 0..10.

# Kiểm tra giá trị rỗng

(Null value control)

---

- Một khoá chính thường bị cấm không được có giá trị null.
  - Ví dụ: Các trường là khóa chính (MASV, MAMH,...)
- Các trường khác cũng có thể cần kiểm tra giá trị null tùy 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 giá trị của thuộc tính đã cho có thể bị hạn chế bởi giá trị của những thuộc tính khác.
- Ví dụ: Trong mỗi quan hệ 1\_N, nếu giá trị của bảng bên 1 chưa có thì sẽ không được có bên N.

SVIEN

<u>MASV</u>	TEN	MALOP
TCTH01	Sơn	TCTHA
TCTH02	Bảo	TCTHB
TCTH03	Trang	TCTHA

LOP		
<u>MALOP</u>	TENLOP	SISO
TCTHA	TCTH32A	80
TCTHB	TCTH32B	65
TCTHC	TCTH32C	82



# 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?

# Thiết kế các bản ghi vật lý

## BẢNG MÔ TẢ CÁC BẢN GHI

Các trường (fields)	Danh sách các trường trong một bản ghi
Dữ liệu có cấu trúc (Structure Data)	Định nghĩa cấu trúc dữ liệu dùng để lưu trữ bản ghi (Thứ tự các trường, khóa chính, khóa ngoại...)
Sự lưu trữ lại (retention)	Đặc tả những bản ghi nào đó được giữ lại trong file bao lâu (dữ liệu về sinh viên không được lưu trữ quá 10 năm sau khi ra trường).

# 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 luận lý (logical record)** dùng để nhóm các thuộc tính được xác định bởi cùng 1 khóa chính, thứ tự các thuộc tính không quan trọng
- ❑ 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

# Trường có chiều 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
- Để tìm vị trí của trường thứ  $m$  trong bản ghi thứ  $n$  của tập tin CSDL

$$\begin{aligned} & \text{Địa chỉ bắt đầu của file} \\ & + (n-1) \text{ chiều dài bản ghi} \\ & + \text{sum}(\text{length}_i) \quad \text{với } i=1 \text{ đến } m-1 \end{aligned}$$

---

= Địa chỉ bắt đầu của trường thứ  $m$

$\text{Length}_i$  : chiều dài của trường thứ  $i$

# Trường có chiều 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

# Phi Chuẩn

---

- ❑ Việc phi chuẩn hóa các quan hệ đã chuẩn hóa trong nhiều trường hợp là cần thiết để tận dụng dung lượng trang của máy.
  - BENHNNHAN(MaBN, TenBN, Diachi\_BN, Ngay\_nhap, Giuong\_phong, Khoa, Tinh\_trang, Ngayra, ThanhToan)
- ❑ Ta có thể phân chia nó thành 2 quan hệ mới để có độ dài gần với dung lượng trang:
  - BENHNNH1(MaBN, TenBN, Diachi\_BN, Khoa)
  - BENHNNH2(MaBN, Ngay\_nhap, Giuong\_phong, Tinh\_trang, Ngayra, ThanhToan)

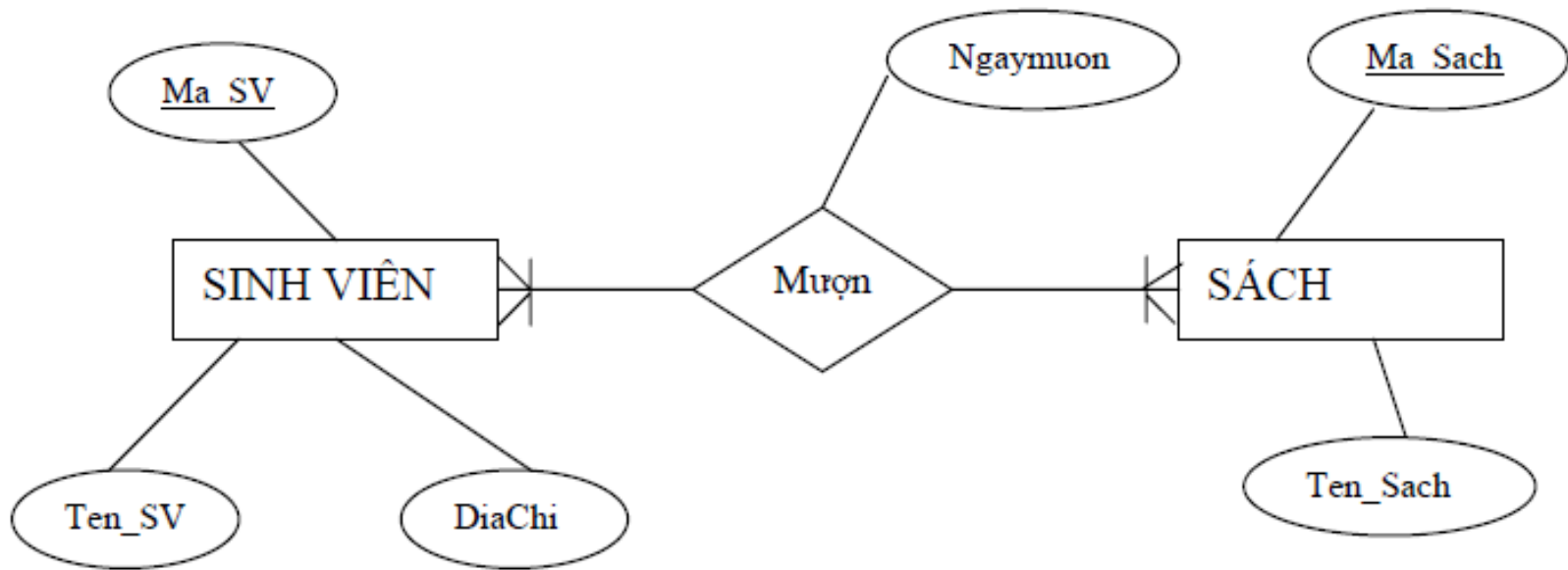
# Phi Chuẩn

- ❑ Có một số dạng phi chuẩn hóa, nhưng không có một quy tắc chặt chẽ nào. Rodger đã thảo luận đến một số trường hợp chung có thể xét phi chuẩn:
- ❑ Hai thực thể có quan hệ một – một.
- ❑ Ví dụ: Có 2 quan hệ có mối liên kết 1\_1 như sau:
  - SINHVIEN(MaSV, TenSV, MaThe)
  - THEDOC(MaThe, DiaChi, NgayCap, MaSV)
- ❑ Phi chuẩn hóa ta có quan hệ sau:
  - SINHVIEN(MaSV, TenSV, MaThe, DiaChi, NgayCap)
- ❑ Và trong trường hợp này MaThe, DiaChi, NgayCap có thể bỏ trống đối với những SV không có thể.

# Phi Chuẩn

- ❑ Hai thực thể có mối quan hệ M\_N trong đó liên kết có thuộc tính riêng.

Ví dụ: Có quan hệ như sau:



# Phi Chuẩn

---

- ❑ Sau khi chuẩn hóa, ta nhận được 3 quan hệ sau:
  - SINHVIÊN(Ma\_SV, Ten\_SV, DiaChi)
  - SÁCH(Ma\_Sach, Ten\_Sach)
  - MUỖN(Ma\_SV, Ma\_Sach, Ngay\_muon)
- ❑ Phi chuẩn hóa ta được:
  - SINHVIÊN(Ma\_SV, Ten\_SV, DiaChi)
  - MUỖN(Ma\_SV, Ma\_Sach, TenSach, NgayMuon)

# Phi Chuẩn

---

- ❑ Dữ liệu tham chiếu: Trong quan hệ 1\_N nếu bảng ở bên 1 không tham gia vào một quan hệ nào khác thì ta có thể hợp nhất 2 thực thể này thành 1.
- ❑ Ví dụ:
  - KHO (Ma\_Kho, Ten\_Kho, Loai\_Kho)
  - HANG(Ma\_Hang, Ten\_Hang)
- ❑ Phi chuẩn hóa ta được:
  - HANG(Ma\_Hang, Ten\_Hang, Ma\_Kho, Ten\_Kho, Loai\_Kho)



# Thiết kế file vật lý

---

- ❑ File vật lý là một phần nhỏ của bộ nhớ thứ cấp (đĩa cứng, băng từ...) lưu các bản ghi vật lý một cách độc lập.
- ❑ Mỗi bản ghi (record) đều có 1 mã nhận dạng duy nhất (unique identifier), gọi tắt là **rid**.
- ❑ Tất cả các bản ghi được lưu trữ theo thứ tự ngẫu nhiên (random order) vào file.
- ❑ File không xếp thứ tự (unordered file) được gọi là **heap file**. Các bản ghi sẽ được lưu trữ trong các trang (page) có cùng kích cỡ.

# Thiết kế file vật lý

---

1. **Các loại file:** Một hệ thống thông tin có thể cần đến 6 loại file:
  - ❑ **File dữ liệu (Data file- master file):** là file chứa dữ liệu liên quan với mô hình dữ liệu vật lý và logic. File này luôn tồn tại, nhưng nội dung thay đổi.
  - ❑ **File lấy từ bảng (look up table file):** Là danh sách các dữ liệu tham chiếu lấy từ một hay một số file khác theo một yêu cầu nào đó.
  - ❑ **File giao dịch (Transaction file):** là file dữ liệu tạm thời phục vụ các hoạt động hàng ngày của một tổ chức. File này thường được thiết kế để phục vụ các yêu cầu xử lý nhanh.

# Thiết kế file vật lý

---

- ❑ **File làm việc (Work file):** Là file tạm thời dùng để lưu kết quả trung gian, file này sẽ tự động xóa đi mỗi khi không cần thiết.
- ❑ **File bảo vệ (Protection file):** là file được thiết kế để khắc phục những sai sót trong quá trình hệ thống hoạt động. Các file này cho hình ảnh của file dữ liệu trước và sau những hoạt động nhất định (cập nhật, sửa đổi, xử lý...) của hệ thống.
- ❑ **File lịch sử (History file):** File này ghi lại quá trình hoạt động của hệ thống, cũng có thể là các dữ liệu cũ hiện không cần sử dụng.

# Thiết kế file vật lý

---

## Các phương pháp truy cập

- ❑ **Phương pháp truy cập trực tiếp:** sử dụng tính toán để xác định địa chỉ chính xác của một bản ghi và truy nhập trực tiếp đến bản ghi đó.
- ❑ **Phương pháp gián tiếp:** hỗ trợ việc tìm kiếm bản ghi thứ n xuất phát từ một vị trí hiện thời của con trỏ hay điểm bắt đầu của 1 file.

# Thiết kế file vật lý

---

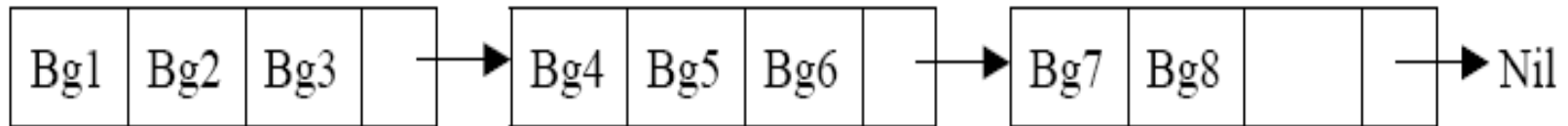
## Tổ chức file

- Cách tổ chức file là kỹ thuật sắp xếp các bản ghi vật lý của một file trên một thiết bị nhớ thứ cấp. Tổ chức một file cụ thể cần tính toán đến các yếu tố sau:
  - Lấy dữ liệu nhanh.
  - Thông lượng các giao dịch xử lý lớn
  - Sử dụng hiệu quả không gian nhớ.
  - Tránh được sai sót khi mất dữ liệu
  - Tối ưu hóa nhu cầu tổ chức file.
  - Đáp ứng được nhu cầu khi tăng dữ liệu
  - Mô hình tổ chức bộ nhớ ngoài.
  - Các phép toán đặc trưng trên tệp dữ liệu (Thêm, Sửa, Xóa, Tìm)

# Thiết kế file vật lý

## Tổ chức file tuần tự (Sequential file)

- Trong tổ chức file tuần tự, các bản ghi được sắp xếp tuần tự.
- Giả sử: Hiện tại tệp cơ 8 bản ghi. 1 khối chứa 3 bản ghi. Con trỏ tệp trỏ đến bản ghi đầu tiên.



# Thiết kế file vật lý

- ❑ **Thêm bản ghi:** lần theo con trỏ đến kiểm tra khối cuối cùng (khối 3) xem có đủ bộ nhớ không. Nếu không đủ thì phải cấp thêm khối mới.
- ❑ Xóa bản ghi (có giá trị khóa =k):
  - Tìm đến bản ghi đó
  - Xóa :
    - ❑ Xóa logic: chỉ đánh dấu xóa
    - ❑ Xóa vật lý: Xóa hẳn bản ghi đó
- ❑ Tìm kiếm bản ghi (có giá trị khóa =k): Tìm lần lượt từ trên xuống dưới cho đến khi gặp khóa k.
- ❑ Sửa đổi giá trị thuộc tính:
  - Tìm đến thuộc tính cần sửa
  - Ghi đè giá trị mới lên giá trị cũ

# Thiết kế file vật lý

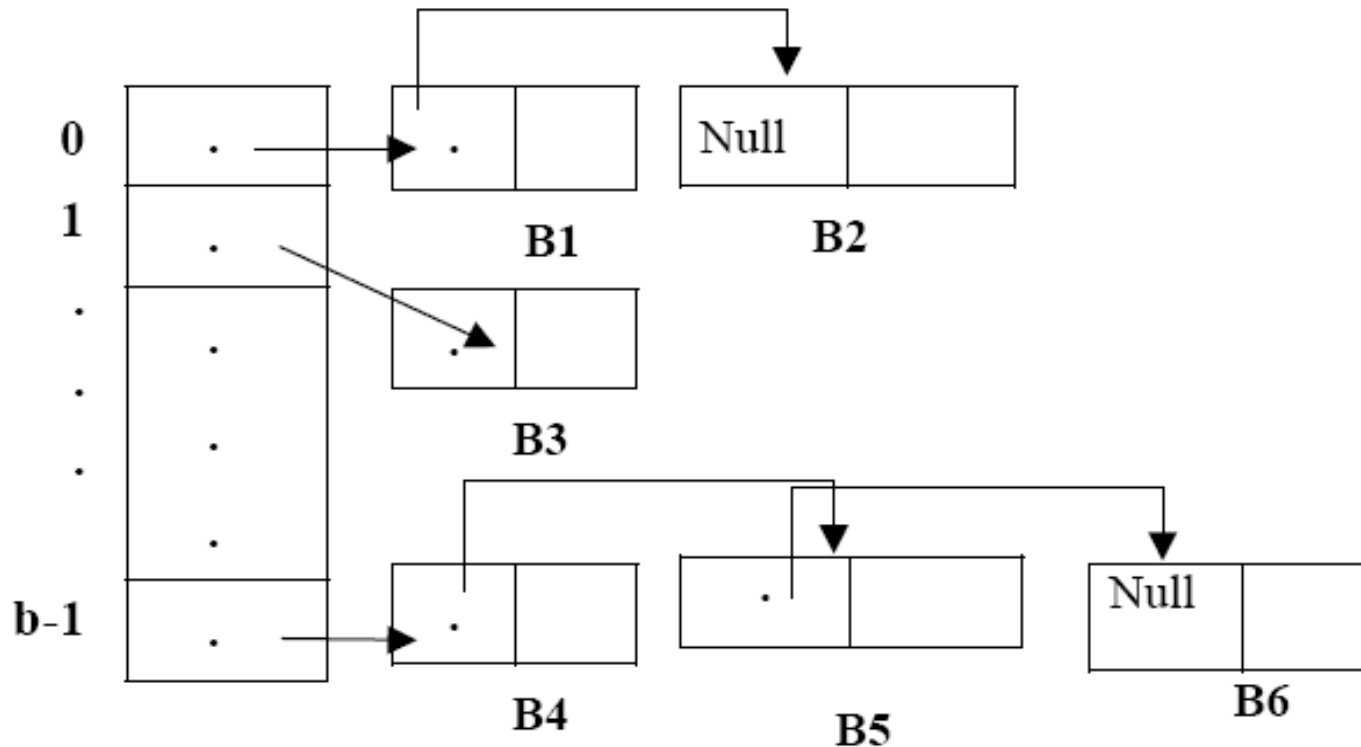
## Tổ chức file băm (Hashed File)

- ❑ Khái niệm hàm băm: Mỗi bản ghi đều có 1 khóa là giá trị số (giả sử là  $k$ )
- ❑ Hàm băm  $H(k)=b$ . Trong đó,  $b$  là số cụm (hàm băm sẽ tác động lên giá trị khóa và trả lại 1 số nguyên là số cụm)
- ❑ Phân chia tập hợp các bản ghi của tệp dữ liệu thành các cụm (Buckets). Mỗi cụm bao gồm một hoặc nhiều khối, mỗi khối chứa một số lượng cố định các bản ghi.
- ❑ Mỗi cụm ứng với một địa chỉ băm được đánh số từ  $0..b-1$ . Ở mỗi đầu của khối đều chứa con trỏ trỏ đến khối tiếp theo trong cụm, khối cuối cùng trong cụm chứa con trỏ rỗng.
- ❑ Có một bảng chỉ dẫn cụm (bucket directory): chứa  $k$  con trỏ, mỗi con trỏ chứa địa chỉ khối đầu tiên của từng cụm.



# Thiết kế file vật lý

## Tổ chức file băm (Hashed File)



# Thiết kế file vật lý

---

- **Thêm bản ghi mới (với giá trị khóa k):**
  - Tính  $H(k)$ : Nếu  $H(k)=b$  thì bổ sung vào nhóm b (Lần theo con trỏ, thêm vào như tổ chức tuần tự).
- **Tìm kiếm bản ghi (với giá trị khóa k):**
  - Tính  $H(k)$ : Nếu  $H(k)=b$  thì b chính là nhóm chứa bản ghi có khóa là k. Sau đó tìm kiếm tuần tự trên nhóm đó.
- **Xóa bản ghi (với giá trị khóa k):**
  - Tìm đến bản ghi có khóa k bằng cách tính  $H(k)$ : Nếu  $H(k)=b$  thì tìm bản ghi đó trong nhóm b.
  - Xóa logic hoặc xóa vật lý.
  - Nếu bản ghi là duy nhất trong khối thì khi xóa bản ghi sẽ đồng thời giải phóng khối khỏi cụm chứa khối.

# Thiết kế file vật lý

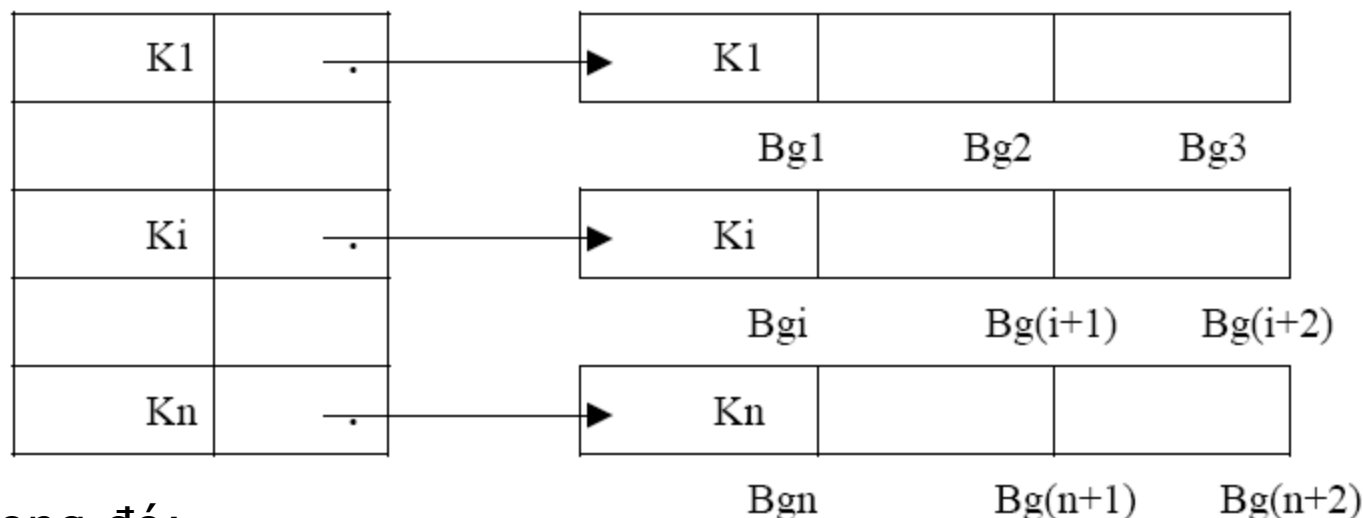
## □ Sửa đổi thuộc tính:

- Sửa đổi thuộc tính không phải là khóa: Ghi đè giá trị mới lên giá trị cũ.
- Sửa đổi thuộc tính khóa: Xóa bản ghi cũ, thêm bản ghi mới vào.
- Tính  $H(k')$ .
  - Nếu  $h(k) = h(k') \rightarrow$  Ghi đè lên
  - Nếu  $h(k) \neq h(k') \rightarrow$  Xóa bản ghi cũ, thêm bản ghi mới vào.

# Thiết kế file vật lý

## Tổ chức file chỉ số

- Một kiểu tổ chức tệp dữ liệu truy nhập khóa rất thường dùng trong CSDL là tệp chỉ số.

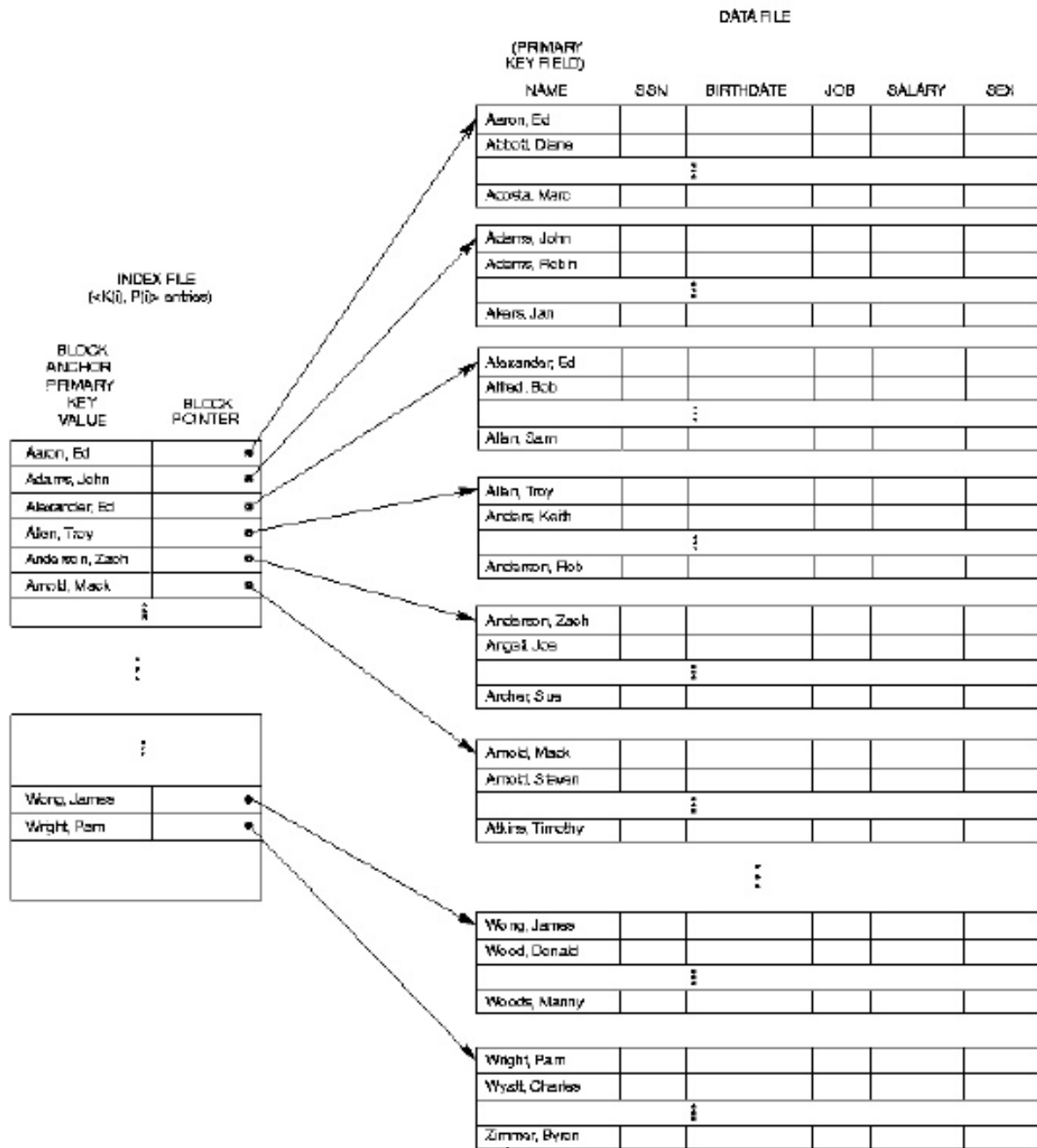


Trong đó:

- K1: Giá trị khóa của bản ghi đầu tiên của khối thứ nhất.
- Tệp chỉ dẫn có 2 trường: khóa và con trỏ
- Tệp chính có n khối, trong khối chứa các bản ghi.



# TỔ



Trần Thị Kim Chi

Hình 7.3. Minh họa tệp chỉ số

# Tổ chức file chỉ số

---

- **Thêm bản ghi (có giá trị khóa k):**
  - Tìm vị trí khối cần thêm ( $H(k)=b$ ).
  - Nếu khối đó còn chỗ trống thì thêm bản ghi vào theo đúng thứ tự sắp xếp và chú ý thay đổi khóa trong bảng chỉ dẫn khối nếu có sự thay đổi.
- **Xóa bản ghi (có giá trị khóa k):**
  - Quá trình giống như thêm một bản ghi. Tuy nhiên, nếu khi xóa tạo ra khối rỗng, khi đó có thể xóa bỏ toàn bộ khối.
  - Tìm kiếm bản ghi (có giá trị khóa k): Có thể tìm kiếm tuần tự hoặc nhị phân.



# Tổ chức file chỉ số

---

- **Sửa đổi thuộc tính:**
  - Thuộc tính khóa: Xóa cũ, thêm mới.
  - Thuộc tính không khóa: Ghi đè lên thuộc tính cũ.

# Tổ chức file chỉ số

**Bảng so sánh các cách tổ chức file khác nhau**

Yếu tố	Cách tổ chức file		
	Tuần tự	Chỉ số	Băm
Không gian lưu trữ	Không lãng phí	Cần nhiều không gian hơn vì phải lưu file chỉ số	Cần nhiều không gian khi thêm và xóa bản ghi (phải tính $H(k)$ )
Truy nhập tuần tự theo khóa chính	Rất nhanh	Trung bình	Không thực tế
Truy nhập ngẫu nhiên theo khóa chính	Không thực tế	Trung bình	Rất nhanh
Xóa bản ghi	Tạo ra khoảng trống và có thể yêu cầu tổ chức lại	Nếu không gian nhớ được bố trí động thì dễ dàng, nhưng yêu cầu phải bảo trì các chỉ số.	Rất dễ dàng
Thêm bản ghi	Yêu cầu sắp xếp lại file sau khi thêm	Nt	Nt
Sửa bản ghi	Yêu cầu sắp xếp lại file sau khi sửa	Dễ dàng, nhưng yêu cầu phải bảo trì các chỉ số	Rất dễ dàng



**Bảng so sánh các cách tổ chức file khác nhau**

Yếu tố	Cách tổ chức file		
	Tuần tự	Chỉ số	Băm
Không gian lưu trữ	Không lãng phí	Cần nhiều không gian hơn vì phải lưu file chỉ số	Cần nhiều không gian khi thêm và xóa bản ghi (phải tính $H(k)$ )
Truy nhập tuần tự theo khóa chính	Rất nhanh	Trung bình	Không thực tế
Truy nhập ngẫu nhiên theo khóa chính	Không thực tế	Trung bình	Rất nhanh
Xóa bản ghi	Tạo ra khoảng trống và có thể yêu cầu tổ chức lại	Nếu không gian nhớ được bố trí động thì dễ dàng, nhưng yêu cầu phải bảo trì các chỉ số.	Rất dễ dàng
Thêm bản ghi	Yêu cầu sắp xếp lại file sau khi thêm	Nt	Nt
Sửa bản ghi	Yêu cầu sắp xếp lại file sau khi sửa	Dễ dàng, nhưng yêu cầu phải bảo trì các chỉ số	Rất dễ dàng

# Ví dụ về thiết kế file

- ❑ Xét Ví dụ sau: Có 2 chứng từ.

ĐƠN ĐẶT HÀNG				
				Số hóa đơn: xxxxxx
Người đặt hàng: (30 ký tự.....)				
Địa chỉ: (50 ký tự.....)				
Ngày đặt: dd/mm/yyyy				
Số tt	Tên hàng	Mô tả hàng	Đơn vị tính	Số lượng
Xx	Char(15)	Char(50)	Char(10)	xxxxx
...	....	....	....	....

# Ví dụ về thiết kế file

- ❑ Xét Ví dụ sau: Có 2 chứng từ.

PHIẾU GIAO HÀNG

Số phiếu: xxxxxx

Tên khách hàng: (30 ký tự.....)

Địa chỉ: (50 ký tự.....)

Nơi giao hàng: (50 ký tự.....)

Ngày giao: dd/mm/yyyy

Số tt	Tên hàng	Đơn vị tính	Đơn giá	Số lượng	Thành tiền
Xx	Char(15)	Char(10)	Number	Number	Number
...	....	....	....	....	....

# Ví dụ về thiết kế file

---

- Sau quá trình thiết kế logic ta được các quan hệ sau:
  - KHACH(Ma\_Khach, Ten\_kh, DiaChi\_kh)
  - HANG(Ma\_hang, Ten\_hang, Mota\_hang, Donvi)
  - DONHANG(So\_Don, Ma\_Khach, NgayDon)
  - PHIEUGIAO(So\_Phieu, Ma\_Khach , Noi\_Giao, Ngay\_Giao)
  - DONGDON(So\_Don, Ma\_Hang, So\_luongD)
  - DONGPHIEUGIAO(So\_Phieu, MaHang, So\_luonggi, DonGia)
- Khi tiến hành thiết kế vật lý, ta được các file sau:

# Ví dụ về thiết kế file

## 1. KHACH

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>Ma_khach</u>	Ký tự	6	Chữ hoa +số
Ten_kh	Ký tự	30	Chữ đầu viết hoa
Diachi_kh	Ký tự	30	Chữ đầu viết hoa

## 2. HANG

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>Ma_hang</u>	Ký tự	6	Chữ hoa+số
Ten_hang	Ký tự	15	Chữ đầu viết hoa
Mota_hang	Ký tự	30	Chữ đầu viết hoa
Don_vi	Ký tự	10	Chữ thường

# Ví dụ về thiết kế file

## 3. DONHANG

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_don</u>	Ký tự	6	Chữ hoa +số
<u>Ma_khach</u>	Ký tự	15	Chữ hoa +số
Ngay_don	Ngày	8	Dd/mm/yy

## 4. DONGDON

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_don</u>	Ký tự	6	Chữ hoa +số
<u>Ma_hang</u>	Ký tự	6	Chữ hoa +số
So_luongd	Số	7	Số nguyên

# Ví dụ về thiết kế file

## 5. PHIEUGIAO

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_phieu</u>	Ký tự	6	Chữ hoa + số
<u>Ma_khach</u>	Ký tự	6	Chữ hoa + số
Ngay_giao	Ngày	30	Dd/mm/yy
Noi_giao	Ký tự	30	
Tong_tien	Số	9	Số thực

## 6. DONGPHIEU

Tên thuộc tính	Kiểu dữ liệu	Kích cỡ	Khuôn dạng
<u>So_phieu</u>	Ký tự	6	Chữ hoa+ số
<u>Ma_hang</u>	Ký tự	6	Chữ hoa + số
Don_gia	Số	6	Số thực
So_luonggi	Số	7	Số thực

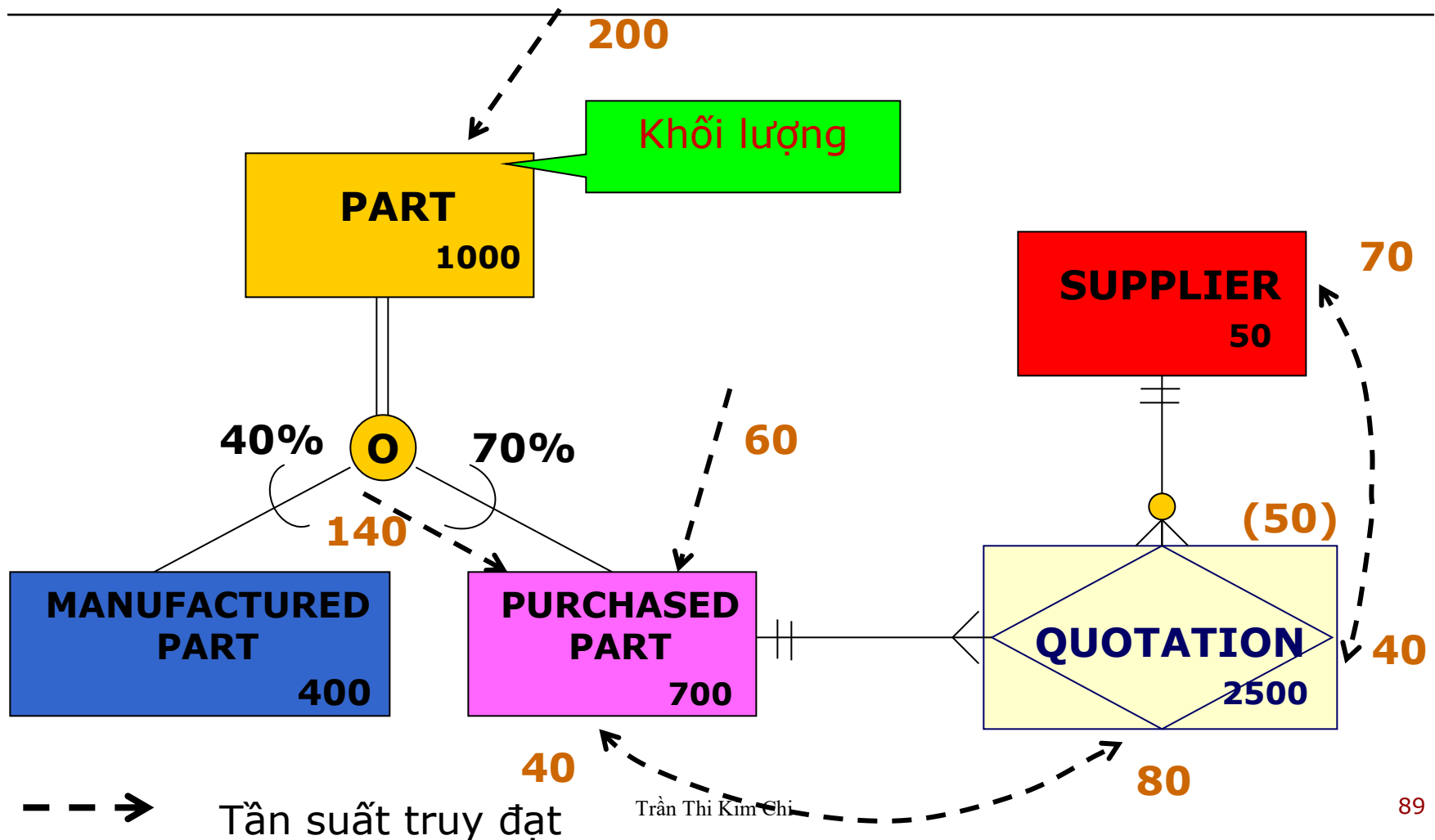
# Khối lượng dữ liệu & tần suất sử dụng (Data volume and usage frequency)

---

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



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



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

---

- ❑ Việc thống kê khối lượng và tần suất được thực hiện trong giai đoạn phân tích hệ thống bởi phân tích viên hệ thống (system analyst)
- ❑ Việc thống kê không đòi hỏi chính xác tuyệt đối mà chỉ dùng làm cơ sở cho bước thiết kế tiếp theo.