

THIẾT KẾ CƠ SỞ DỮ LIỆU

Giảng viên: Nguyễn Kim Sao
saonkoliver@gmail.com
0905.883.993

Các khái niệm mở đầu

- CSDL quan hệ
- Mô hình dữ liệu
- Các công cụ

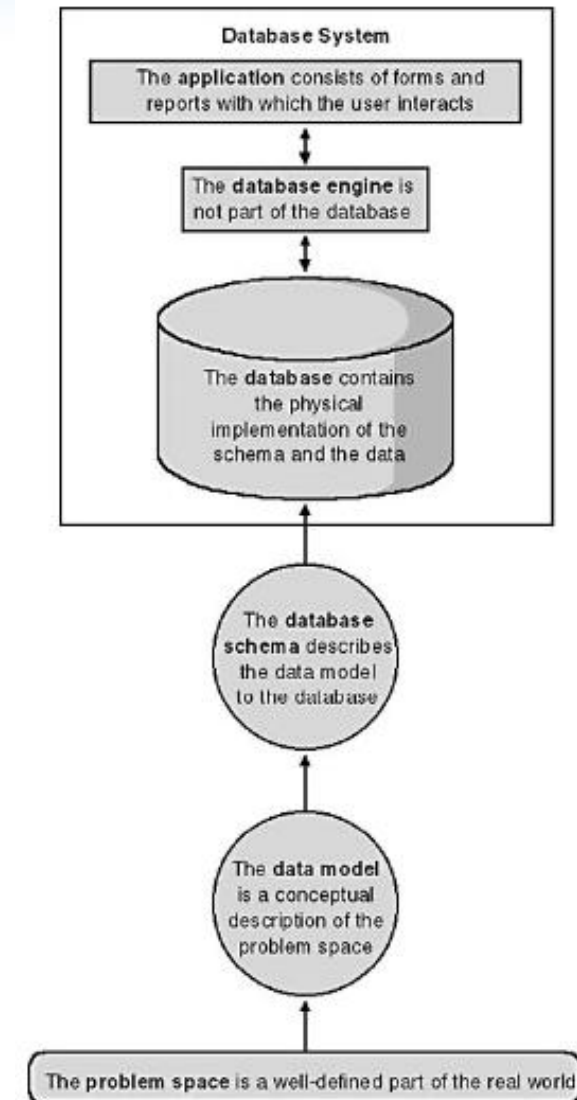


Cơ sở dữ liệu

- Khái niệm: tập hợp dữ liệu của một tổ chức, được lưu trữ trong máy tính, được nhiều người sử dụng và được tổ chức theo một mô hình.
- Một CSDL cần đáp ứng các tiêu chuẩn:
 - Phản ánh được trung thực thế giới thực
 - Tối thiểu sự dư thừa thông tin
 - Có sự độc lập giữa các chương trình và dữ liệu
 - Có hiệu suất sử dụng cao
 - Đảm bảo an toàn, bí mật của dữ liệu.
- CSDL quan hệ mô hình hóa một phần của thế giới thực - không gian bài toán (problem space)

Hệ thống CSDL

- Hệ thống CSDL quan hệ có các đặc điểm sau:
 - Dữ liệu được biểu diễn theo một trật tự hàng và cột, được gọi là quan hệ
 - Tất cả các giá trị là vô hướng. Nghĩa là mỗi giá trị của một vị trí hàng/cột là duy nhất
 - Mọi toán tử thực hiện trên một quan hệ thực thể có kết quả là một quan hệ thực thể (đóng kín)



Mô hình quan hệ

- Dựa trên lý thuyết tập hợp & logic vị từ
- Định nghĩa: Mô hình quan hệ:
 - cách thức dữ liệu được thể hiện (cấu trúc dữ liệu),
 - cách thức dữ liệu được bảo vệ (toàn vẹn dữ liệu) và
 - các thao tác thực hiện trên dữ liệu (thao tác dữ liệu).
- Đặc tính của một CSDL quan hệ:
 - Tất cả dữ liệu được thể hiện trong các hàng và cột (quan hệ)
 - Tất cả các giá trị là đơn trị
 - Tất cả các thao tác được thực hiện trên toàn bộ quan hệ và kết quả nhận được cũng là quan hệ -> khái niệm đóng của quan hệ

Mô hình dữ liệu

- Thực thể
 - Bất kỳ thông tin gì hệ thống cần lưu trữ
 - Nhận biết qua *danh từ* & *động từ*
 - Là mô hình hóa của các đối tượng, sự kiện
- Thuộc tính
 - Là các yếu tố của thực thể
 - Dựa vào ý nghĩa, cách thức sử dụng của dữ liệu
 - Dựa vào không gian bài toán để quyết định là thuộc tính hay thực thể.

Mô hình dữ liệu (tt)

- Mô hình dữ liệu là sự biểu diễn các định nghĩa về: thực thể, thuộc tính, miền xác định, quan hệ
- Miền: Kiểu dữ liệu + Các luật hợp lệ dữ liệu
- Mỗi quan hệ
 - Sự liên kết giữa các thực thể
 - Thành phần: các thực thể tham gia trong liên kết
 - Bậc của mỗi quan hệ: số các thành phần
 - Kiểu liên kết: 1-1, 1-n, n-n
- Sơ đồ thực thể - mối quan hệ
 - *hình chữ nhật* để biểu diễn *thực thể*
 - *hình thoi* để biểu diễn *mối quan hệ*,
 - *hình elip* để biểu diễn *thuộc tính*
 - Các kiểu liên kết



Toàn vẹn dữ liệu

- Toàn vẹn miền: các luật định nghĩa các giá trị hợp lệ
 - Xác định kiểu dữ liệu logic cho miền
 - Xác định phạm vi, độ chính xác đối với kiểu số, độ dài cực đại đối với kiểu xâu ký tự
 - Xác định xem miền có cho phép chứa giá trị chưa biết (unknown value) và giá trị không tồn tại (nonexistent value) hay không?
 - Đặc tả tập các giá trị thuộc miền càng cụ thể càng tốt

Toàn vẹn dữ liệu (tt)

- Toàn vẹn dịch chuyển
 - Định nghĩa các trạng thái mà một bộ dữ liệu (tuple) trong quan hệ có thể được dịch chuyển qua.
 - Là các luật quy định về tính hợp lệ của sự dịch chuyển dữ liệu
- Toàn vẹn giao tác
 - Thủ tục khống chế các cách thức thao tác với CSDL
 - Trong SQL Server là các Trigger

Toàn vẹn dữ liệu (tt)

- Toàn vẹn thực thể
 - Yêu cầu tồn tại khóa chính của quan hệ
 - Thừa kế các luật ràng buộc trong toàn vẹn miền
 - Các luật ràng buộc giữa các thuộc tính của quan hệ
 - Các luật ràng buộc giữa các thực thể
- Toàn vẹn CSDL
 - Hình thức toàn vẹn tổng quát nhất
 - Các luật về nội dung CSDL được cài đặt dưới dạng các thủ tục
 - Trong SQL Server, đó là các Trigger

Toàn vẹn dữ liệu (tt)

- Toàn vẹn tham chiếu
 - Các luật đảm bảo duy trì và bảo vệ các liên kết giữa các quan hệ
 - Toàn vẹn tham chiếu yêu cầu:
 - Hai thuộc tính liên kết của hai quan hệ phải tương thích với nhau về kiểu dữ liệu.
 - Không được phép nhập các giá trị vào khoá ngoại của quan hệ mà giá trị đó không thuộc khoá chính của quan hệ có liên kết với quan hệ đó
 - Trong SQL Server, các yêu cầu trên có thể được cài đặt bởi Trigger

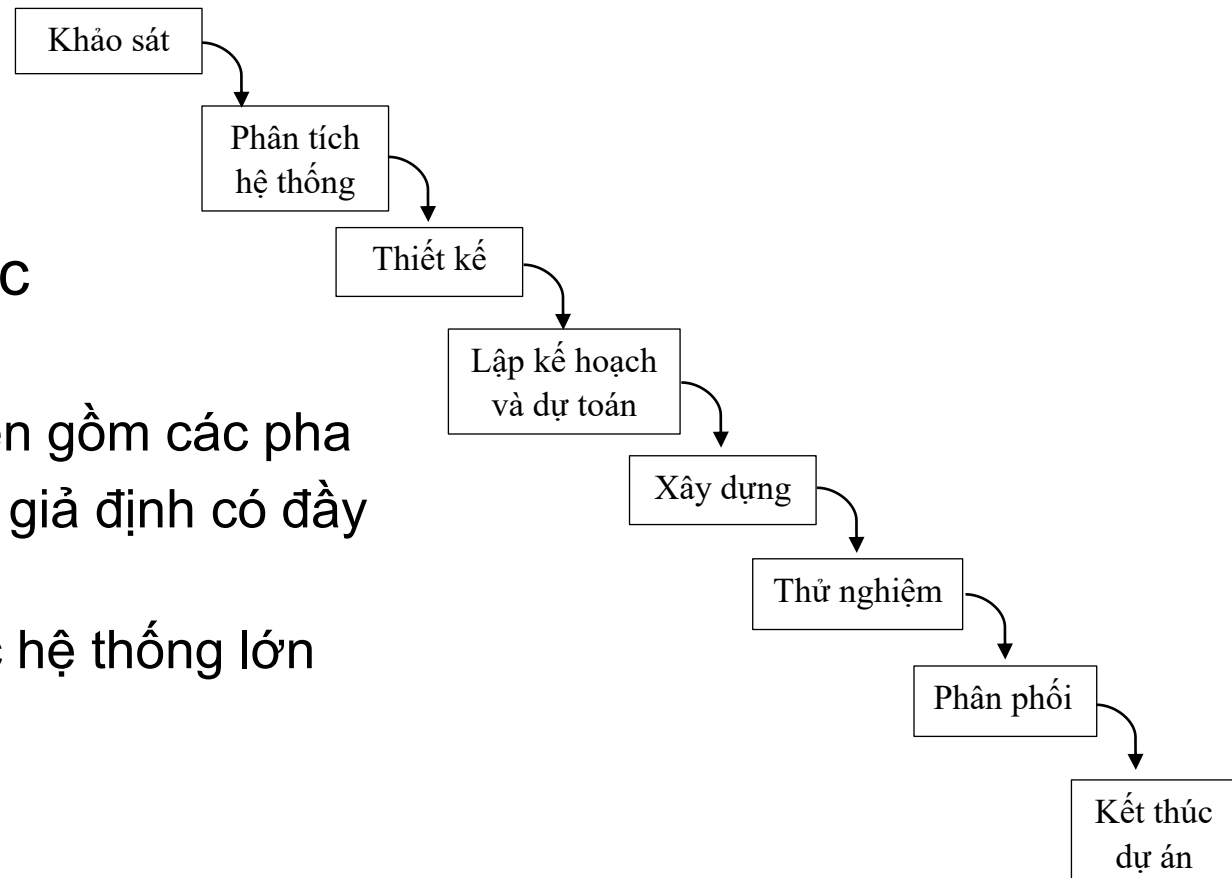
Quy trình thiết kế tổng thể

- Lựa chọn mô hình vòng đời phát triển hệ thống
- Định nghĩa các tham số của hệ thống
- Định nghĩa các tiến trình
- Xây dựng mô hình dữ liệu khái niệm
- Chuẩn bị lược đồ CSDL
- Thiết kế giao diện người dùng

1. Vòng đời phát triển hệ thống

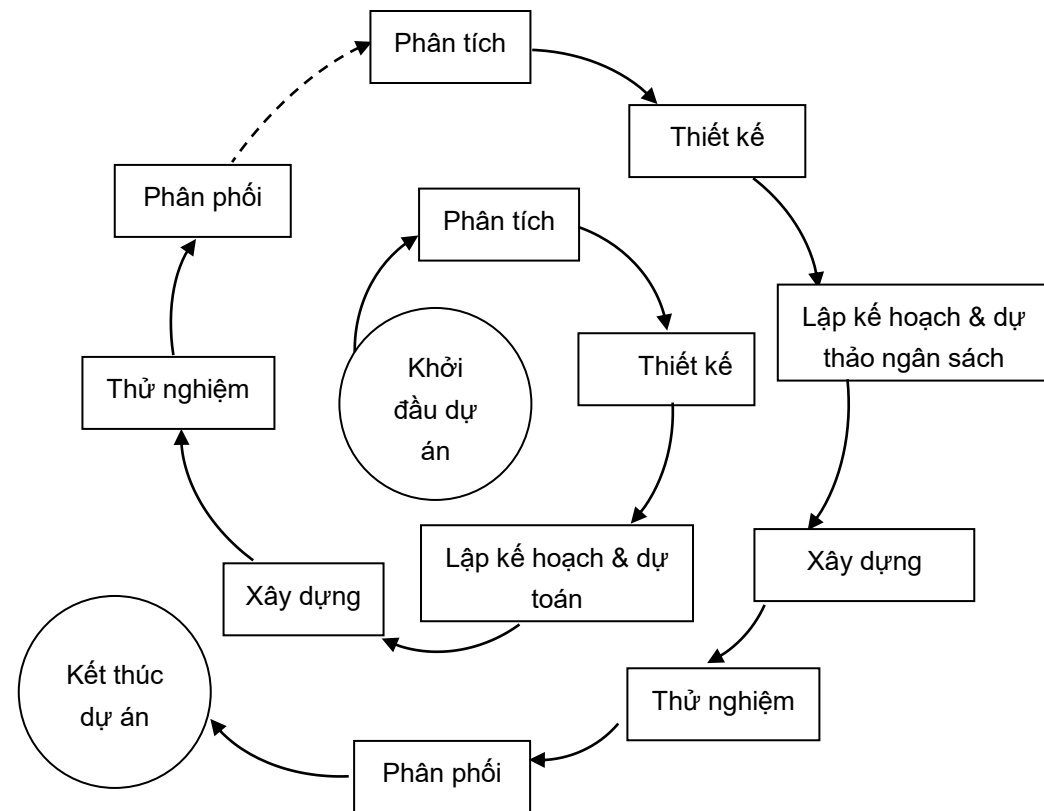
- Mô hình thác nước

- Cổ điển
- Quá trình phát triển gồm các pha
- Tại mỗi pha được giả định có đầy đủ thông tin
- Thích hợp với các hệ thống lớn



1. Vòng đời phát triển hệ thống (tt)

- Mô hình xoáy ốc
 - Dự án được phát triển như 1 đường xoáy ốc
 - Các pha của quá trình phát triển được lặp lại theo chu kỳ là một vòng xoáy ốc
 - Thích hợp với các hệ thống vừa & nhỏ



1. Vòng đời phát triển hệ thống (tt)

- Mô hình làm mẫu

- Xuất phát từ một dự án phần mềm đã được xây dựng hoàn thiện, ta phát triển các dự án phần mềm có yêu cầu tương tự ở những khía cạnh nào đó
- Cho khách hàng sử dụng phần mềm đã có để thu thập yêu cầu của dự án mới
- Thích hợp với các dự án đủ nhỏ

2. Các tham số của hệ thống

- Ở pha này, người thiết kế cần:
 - Xác định được mục tiêu của toàn bộ hệ thống
 - Thiết lập các tiêu chuẩn của hệ thống. Đó là các tiêu chuẩn dùng để đánh giá trong suốt quá trình thiết kế, vận hành của dự án, đánh giá sự thành công hay thất bại của dự án.
 - Xác định phạm vi của hệ thống. Đó là các yêu cầu phải đạt được của dự án.

2.1 Xác định mục tiêu của hệ thống

- Là yếu tố quan trọng đối với việc thiết lập các tiêu chuẩn và xác định phạm vi của hệ thống
- Là câu trả lời của câu hỏi “Tại sao cần phải tự động hóa hệ thống hiện tại?”
 - Có phải để nâng cao tốc độ xử lý công việc?
 - Để tăng độ chính xác? để giảm chi phí?
 - Để cải thiện vị trí của công ty trên thị trường? để hỗ trợ người quản lý trong việc tìm kiếm, xử lý thông tin, lập kế hoạch?, ...

2.1 Xác định mục tiêu của hệ thống (tt)

- Cần định lượng các mục tiêu, không nên đưa ra các mục tiêu chung chung.
- Không phải bất kỳ mục tiêu nào cũng có thể định lượng được -> xây dựng các tiêu chuẩn phù hợp để đánh giá.
- Các mục tiêu phải ổn định để thiết lập các tiêu chuẩn đánh giá và sẵn sàng cho các công đoạn tiếp theo của quá trình thiết kế.

2.2 Thiết lập các tiêu chuẩn

- Mỗi tiêu chuẩn cần tương ứng với một hoặc một số mục tiêu
- Nếu mục tiêu nào không gắn được với một tiêu chuẩn với nó thì xem như người thiết kế chưa hiểu đủ các yêu cầu của khách hàng
- Các tiêu chuẩn là những đích nhỏ cần đạt được cho một mục tiêu lớn hơn

2.2 Thiết lập các tiêu chuẩn (tt)

- Có các tiêu chuẩn về:
 - Các yêu cầu được định lượng rõ ràng
 - Môi trường
 - Các chiến lược thiết kế tổng thể
- Các tiêu chuẩn cần được định lượng rõ ràng
- Khi thiết kế đạt được tiêu chuẩn thì hệ thống được xem là hoàn thành, không cần làm thêm điều gì nữa

2.3 Xác định phạm vi của hệ thống

- Xác định các chức năng trong phạm vi dự án
- Dự án phần mềm sẽ thực hiện những chức năng nhất định không phải là mọi thứ
- Đánh giá được tỷ lệ chi phí / lợi nhuận của mỗi chức năng -> có kế hoạch thiết kế phù hợp

3. Định nghĩa tiến trình

- Tiến trình là một tập của một hoặc nhiều các tác vụ (task) rời rạc cùng thực hiện một hoạt động có ý nghĩa của tổ chức.
- Một tác vụ (task) là một hành động rời rạc, là một bước thực hiện của tiến trình.
- Ví dụ:
 - Tiến trình “Xử lý đơn đặt hàng của khách” bao gồm các tác vụ:
 - Nhận đơn đặt hàng
 - Kiểm tra thẻ thanh toán của khách
 - Kiểm tra kho hàng
 - Giao hàng cho khách
 - Tiến trình “tìm số phone của khách hàng” chỉ bao gồm một tác vụ là chính tiến trình đó.
 - Phân biệt giữa một tác vụ và một hoạt động đôi khi rất khó, danh giới giữa chúng không rõ ràng. Để quyết định là tác vụ hay là hoạt động phải dựa vào không gian bài toán

3. Định nghĩa tiến trình (tt)

- Xác định các tiến trình hiện tại của hệ thống
 - Thu thập yêu cầu từ khách hàng
 - Định danh tác vụ
- Phân tích các tiến trình

3.1 Thu thập yêu cầu từ khách hàng

- Khảo sát hệ thống bằng phỏng vấn KH. Mục tiêu là thu được *quy trình nghiệp vụ* và các *hồ sơ*
- Hướng KH tập trung vào nghiệp vụ hệ thống
- Phải có sự chuẩn bị câu hỏi trước
- Câu hỏi dạng mở, kết hợp với câu hỏi dạng đóng khi cần
- Cần chuẩn bị các thiết bị hỗ trợ: máy ghi âm, camera, ...
- Cuối buổi phỏng vấn cần xác nhận lại các câu trả lời của KH
- Phải giúp khách hàng thấy được các yêu cầu hiện tại hệ thống cần đáp ứng và các yêu cầu tiềm năng trong tương lai

Công việc sau khảo sát

- Dữ liệu thu được trong khảo sát còn ở dạng thô, tản mạn
- Các công việc sau khảo sát:
 - Xử lý sơ bộ kết quả: xem xét, hoàn thiện tài liệu:
 - Phân loại, trích rút, tổng hợp -> dữ liệu đầy đủ, chính xác, gọn gàng, dễ kiểm tra và theo dõi
 - Phát hiện chỗ thiếu để bổ sung, chỗ sai không logic để sửa
 - Xây dựng các bảng mô tả chi tiết tài liệu
 - Quá trình lặp
 - Tổng hợp kết quả: tổng hợp theo xử lý & tổng hợp theo dữ liệu

Công việc sau khảo sát

- Tổng hợp theo xử lý
 - Tổng hợp theo lĩnh vực hoạt động: nhóm các hoạt động có sự gắn kết chặt chẽ với nhau vào 1 nhóm
- Tổng hợp theo dữ liệu
 - Liệt kê được các tài liệu liên quan đến tổ chức
 - Sàng lọc để thu được dữ liệu đầy đủ, chính xác và gán tên gọi cho chúng.
 - Kết quả: bảng tổng hợp các hồ sơ và bảng từ điển dữ liệu
- Hợp thức hóa kết quả khảo sát
 - Hiểu và thể hiện thông tin khảo sát ở những dạng khác nhau được người sử dụng và đại diện tổ chức xác nhận là đúng đắn và đầy đủ
 - Nhằm đảm bảo sự chính xác hóa của thông tin, yêu cầu của hệ thống, đảm bảo tính pháp lý cho việc sử dụng sau

3.2 Định danh tác vụ

- Sắp xếp thông tin thành 1 tập tác vụ
- Một tác vụ (hành động rời rạc):
 - Phải có điểm bắt đầu và kết thúc được xác định rõ ràng
 - Tất cả các luật về nghiệp vụ hệ thống phải hợp lệ trước khi tác vụ bắt đầu và sau khi tác vụ hoàn tất. Tuy nhiên, có thể bị phá vỡ trong quá trình tác vụ thi hành
- Mục tiêu là định rõ được các tác vụ trong mỗi tiến trình
- Ví dụ: danh sách các tác vụ xảy ra trong quá trình bán hàng

3.3 Phân tích tiến trình

- Phân tích sự phụ thuộc giữa các tác vụ của tiến trình
- Phân tích sự phụ thuộc dữ liệu giữa các tác vụ
- -> Sắp xếp các tác vụ theo trật tự nhất định

4. Mô hình dữ liệu quan niệm

- Gồm các mô tả về thực thể, thuộc tính của thực thể và mối quan hệ giữa các thực thể
- Sản phẩm của quá trình là *sơ đồ thực thể - mối quan hệ*
- Các bước xây dựng sơ đồ thực thể - mối quan hệ:
 - Liệt kê, chính xác hóa và lựa chọn thông tin cơ sở
 - Xác định thực thể, thuộc tính, thuộc tính định danh cho mỗi thực thể
 - Xác định mối quan hệ & các thuộc tính riêng của nó
 - Vẽ sơ đồ mô hình E-R và xác định bản số cho các thực thể của mối quan hệ
 - Chuẩn hóa & thu gọn sơ đồ

Liệt kê,... lựa chọn thông tin cơ sở

- Xây dựng từ điển dữ liệu gồm các hồ sơ và các thuộc tính trong nội dung hồ sơ
- Quy tắc: Ghi tên hồ sơ và các thuộc tính của nó ở dưới
- Không bỏ sót bất kỳ thông tin nào
- Chính xác hóa thông tin:
 - Thêm từ cần thiết vào tên gọi cho các thuộc tính -> rõ nghĩa hơn
 - Hai thuộc tính tên gọi khác nhau phải chỉ ra các đối tượng khác nhau
 - Duyệt từ trên xuống dưới và giữ lại các thuộc tính đảm bảo:
 - Mỗi thuộc tính phải đặc trưng cho 1 lớp hồ sơ được xét
 - Một thuộc tính chỉ được duyệt 1 lần
 - Mỗi thuộc tính phải là sơ cấp (nếu được suy ra từ các thuộc tính khác thì bỏ)

Xác định thực thể, thuộc tính...

- Duyệt từ trên xuống dưới các thuộc tính đã chọn, chưa bị loại để tìm thuộc tính “*tên gọi*”
- Mỗi thuộc tính *tên gọi* sẽ có tương ứng 1 thực thể
- Với mỗi thực thể, tìm trong các thuộc tính còn lại để ghi các thuộc tính thực sự của nó
- Xét lần lượt các thực thể, chọn các thuộc tính định danh cho nó

Xác định mối quan hệ ...

- Trong các thuộc tính còn lại của bảng danh sách, tìm các động từ -> mối quan hệ
- Mỗi động từ, trả lời câu hỏi cho mỗi động từ: **ai?**, **cái gì?**, **ở đâu?**, **bằng cách nào?**, **như thế nào?**, **bao nhiêu?** để tìm các thực thể tham gia vào mối quan hệ và tìm các thuộc tính trong danh sách là thuộc tính của mối quan hệ
- Khi không còn tìm thấy mối quan hệ và danh sách các thuộc tính đã hết -> kết thúc

Vẽ sơ đồ E-R

- Sử dụng các ký pháp biểu diễn thực thể, thuộc tính, mối quan hệ
- Vẽ theo thứ tự: thực thể, mối quan hệ, thuộc tính
- Thuộc tính định danh đặt ở phía trên, bên trái thực thể.
- Xác định các bản số

Chuẩn hóa, thu gọn lược đồ

- Chuẩn hóa lược đồ nếu có thuộc tính lặp, phụ thuộc thời gian -> thực thể, thuộc tính đơn
- Thu gọn lược đồ khi:
 - Thực thể treo (tham gia vào mỗi quan hệ và chỉ có 1 thuộc tính) -> loại bỏ thực thể này, đưa thuộc tính vào thực thể liên kết với nó
 - Mỗi quan hệ là bậc 2 và không có thuộc tính riêng -> loại thực thể này.

5. Lược đồ CSDL

- Xây dựng cấu trúc vật lý của dữ liệu dưới dạng trừu tượng
- Chuyển các yếu tố của mô hình dữ liệu quan niệm thành các yếu tố tương ứng trong lược đồ quan hệ:
 - Thực thể -> bảng
 - Mối quan hệ có thuộc tính -> bảng
 - Chuẩn hóa các bảng về các dạng chuẩn 3NF, BCNF

5. Lược đồ CSDL (tt)

- Chuyển các ràng buộc của mô hình dữ liệu quan niệm thành các ràng buộc tương ứng trong lược đồ quan hệ:
 - Ràng buộc miền và thuộc tính -> ràng buộc ở mức trường dữ liệu
 - Ràng buộc thực thể -> ràng buộc ở mức bảng, các ràng buộc này được thể hiện như: bảng phải có khóa chính, các luật hợp lệ dữ liệu giữa các trường dữ liệu của bảng...
 - Ràng buộc trong mô hình dữ liệu quan niệm không cài đặt được bằng các công cụ có sẵn -> được cài đặt bằng các trigger.

Phân tích, thiết kế CSDL

- Phân tích yêu cầu
- Thiết kế CSDL mức khái niệm
- Thiết kế CSDL mức logic
- Tinh chỉnh CSDL và các dạng chuẩn
- Thiết kế CSDL mức vật lý
- Thiết kế an toàn bảo mật

1. Phân tích yêu cầu

- Xác định dữ liệu cần lưu trữ
 - Tên dữ liệu; Định nghĩa dữ liệu; Kiểu dữ liệu; Loại dữ liệu; định lượng; ghi chú thêm
- Xác định ứng dụng sẽ được cài đặt trên CSDL
- Xác định các thao tác thường xuyên được thực hiện
- Một số kỹ thuật phân tích yêu cầu
 - Kỹ thuật phỏng vấn; kỹ thuật sử dụng phiếu khảo sát; Kỹ thuật phân tích tài liệu

2. Thiết kế CSDL mức khái niệm

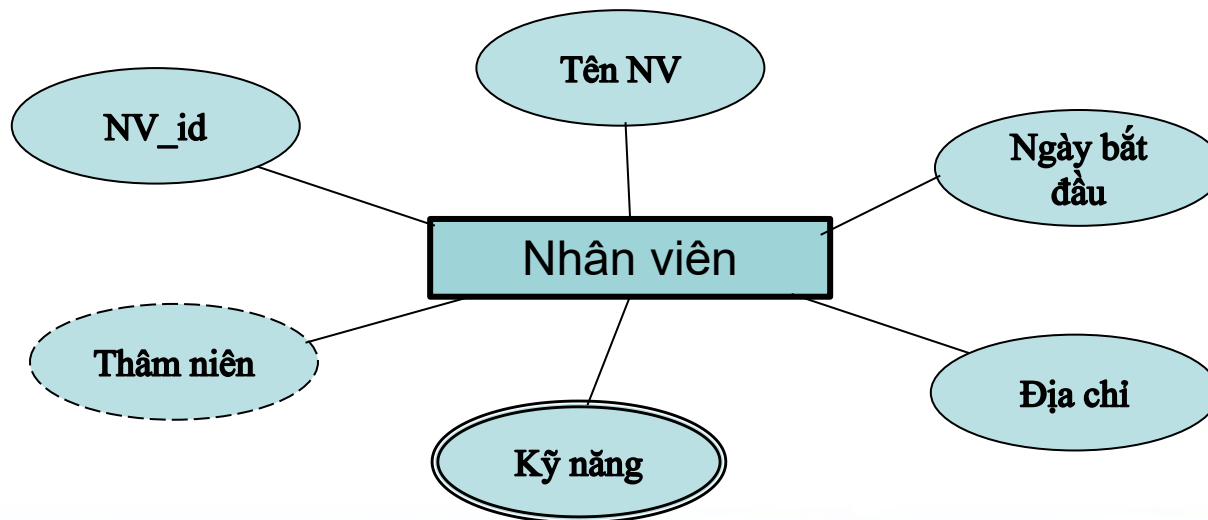
- Lựa chọn kiểu thực thể mạnh và kiểu thực thể yếu
- Lựa chọn giữa các thuộc tính
- Lựa chọn giữa các loại liên kết
- Lựa chọn giữa kiểu thực thể và thuộc tính
- Lựa chọn giữa kiểu thực thể và liên kết
- Các phân cấp “là – một”
- Sơ đồ thực thể liên kết
- Mô tả các ràng buộc trên dữ liệu

Lựa chọn kiểu thực thể mạnh và kiểu thực thể yếu

- Thực thể mạnh: kiểu thực thể tồn tại độc lập với các kiểu thực thể khác
- Thực thể yếu: phụ thuộc vào sự tồn tại của thực thể khác, chứa định danh của thực thể nó phụ thuộc

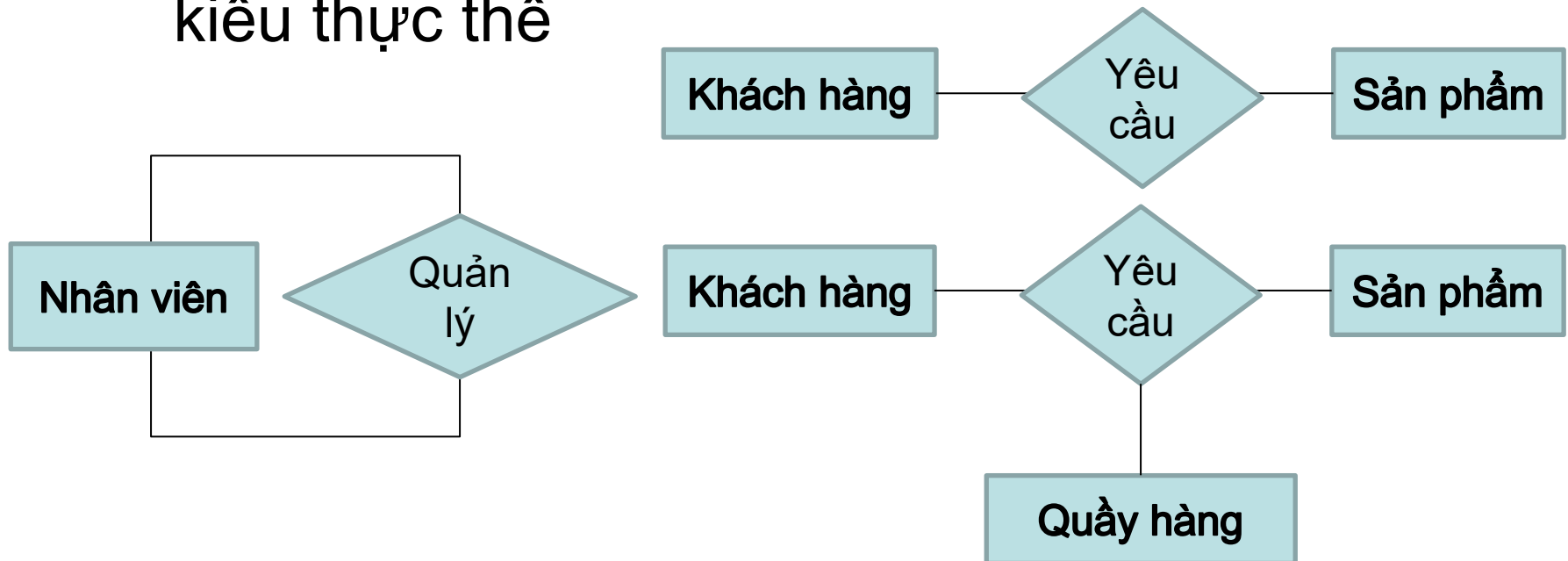
Lựa chọn giữa các thuộc tính

- Thuộc tính đơn và gộp
- Thuộc tính đơn trị và thuộc tính đa trị
- Thuộc tính lưu trữ và thuộc tính dẫn xuất
- Thuộc tính khóa hay thuộc tính định danh



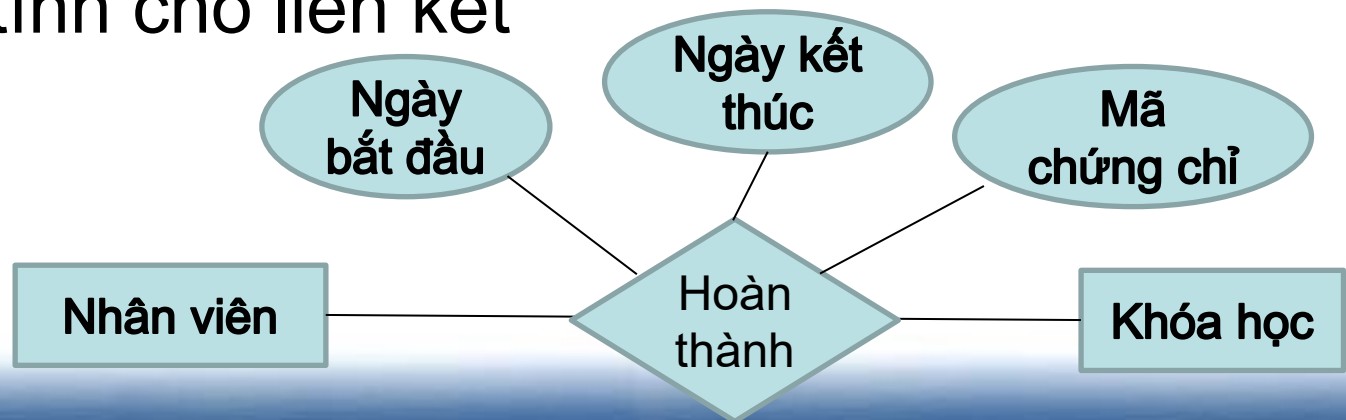
Lựa chọn giữa các loại liên kết

- Khái niệm cơ bản về liên kết
 - Cấp độ liên kết: liên kết cùng một kiểu thực thể, liên kết giữa hai thực thể, liên kết nhiều kiểu thực thể



Lựa chọn giữa các loại liên kết

- Khái niệm cơ bản về liên kết (tt)
 - Lực lượng của liên kết: liên kết một – một; liên kết một – nhiều, liên kết nhiều – nhiều
 - Ràng buộc tham gia của liên kết: tính bắt buộc hay không bắt buộc của thực thể có quan hệ khi tham gia vào liên kết
 - Thuộc tính cho liên kết

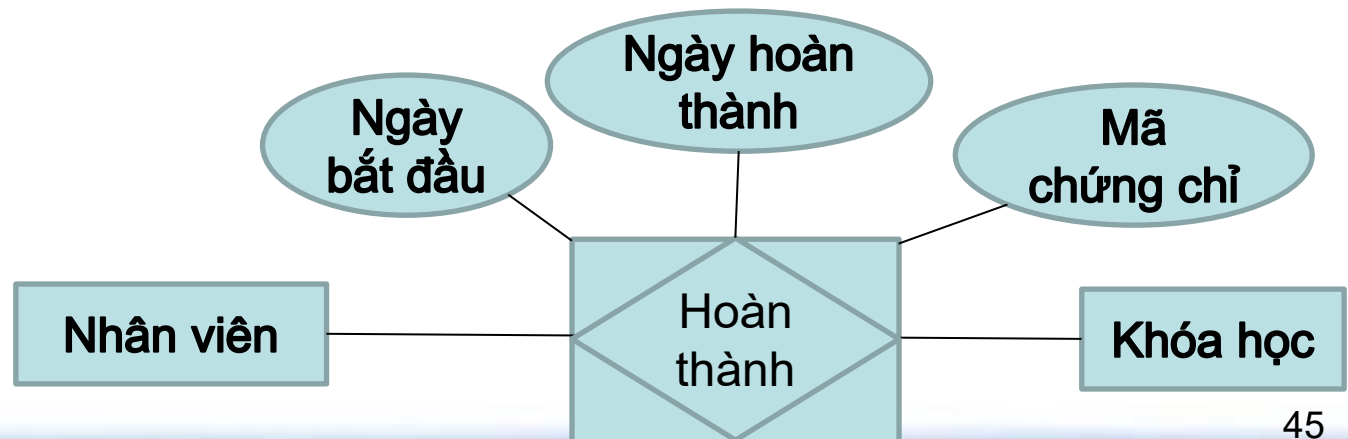


Lựa chọn giữa kiểu thực thể và thuộc tính

- Xem xét dựa trên việc có quản lý chặt chẽ khái niệm đó trước sự thay đổi của thời gian không
- VD: Khi xem xét lương của cá nhân
 - Nếu chỉ quan tâm đến mức lương cuối cùng: Lương là thuộc tính (ngân hàng lưu thông tin lương khách hàng khi làm thẻ)
 - Nếu quan tâm đến quá trình tăng lương của người lao động thì lương có thể lập thành thực thể với các thuộc tính: hệ số lương, ngày tăng lương, (ngày kết thúc),... (công ty bảo hiểm)
- Các VD khác: Địa chỉ,...

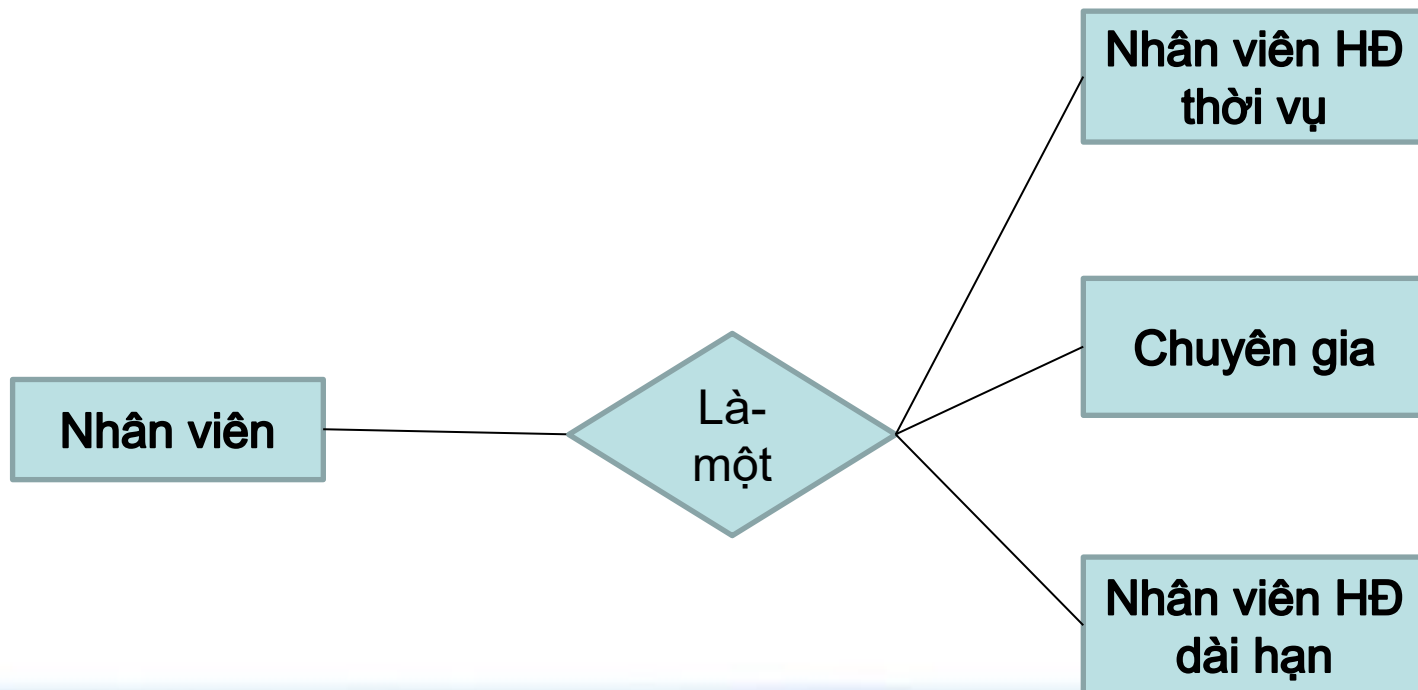
Lựa chọn giữa kiểu thực thể và liên kết

- Chuyển đổi một liên kết thành một thực thể:
 - Liên kết giữa các kiểu thực thể nhiều nhiều
 - Kiểu thực thể kết nối có định danh đơn
 - Liên kết có những thuộc tính mô tả
 - Khi hình thành kiểu thực thể kiểu kết nối thì thực thể này độc lập với các kiểu thực thể được nối bởi liên kết đó



Các phân cấp “là – một”

- A là một tập con của thực thể B. A chứa các thuộc tính của A song lại có những tính chất riêng. A “là-một” B



Mô tả các ràng buộc trên dữ liệu

- Kiểu dữ liệu
- Độ dài
- Khoảng giá trị
- Hỗ trợ giá trị null
- Các định dạng dữ liệu

3. Thiết kế CSDL mức logic

- Ràng buộc toàn vẹn
 - Sau khi ràng buộc được định nghĩa, bất kỳ một câu lệnh theo tác dữ liệu trên bảng không thỏa mãn ràng buộc đó thì câu lệnh sẽ không được thực hiện thành công và trả một lỗi
 - Các ràng buộc toàn vẹn được định nghĩa trên các bảng dữ liệu và được lưu như một phần của định nghĩa bảng trong từ điển dữ liệu, do vậy các ứng dụng thao tác trên bảng dữ liệu đều phải tuân thủ các ràng buộc toàn vẹn đó. Khi thay đổi ràng buộc, ta chỉ cần thay đổi ở mức CSDL
 - Nếu một ràng buộc toàn vẹn được tạo ra cho một bảng mà trong bảng đó đã tồn tại một số dữ liệu không thỏa mãn ràng buộc đó thì có nghĩa ràng buộc đó không thể có hiệu lực

Thiết kế CSDL mức logic (tt)

- Ràng buộc toàn vẹn
 - Ràng buộc miền: ràng buộc về kiểu dữ liệu
 - Ràng buộc khóa: not null, unique key, primary key, foreign key, check
 - Các kích hoạt: trigger

Lựa chọn hệ quản trị CSDL

- SQL server
- SQL lite
- Oracle
- My SQL

Chuyển đổi các kiểu thực thể sang quan hệ

- Chuyển đổi các kiểu thực thể thông thường

```
CREATE TABLE CUSTOMER (  
    Cust_id          Char(20), NOT NULL  
    Cust_name        Char(30), NOT NULL  
    Address           Char(150)  
    PRIMARY KEY (Cust_id))
```

Chuyển đổi các kiểu thực thể sang quan hệ (tt)

- Chuyển đổi kiểu thực thể yếu
 - Chuyển các thuộc tính thực thể thành thuộc tính của quan hệ
 - Khóa chính trong quan hệ sở hữu là khóa ngoại lai của quan hệ
 - Khóa chính của quan hệ là hợp các khóa ngoại lai và các thuộc tính tham gia định danh bộ phận của thực thể yếu

Chuyển đổi các kiểu thực thể sang quan hệ (tt)

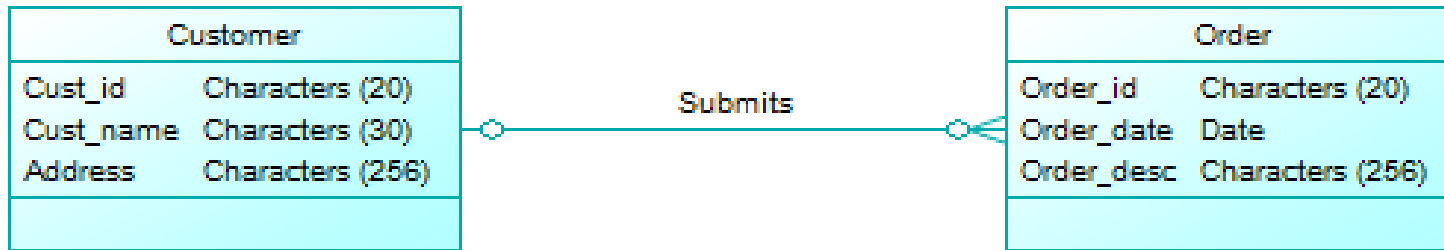
```
CREATE TABLE EMPLOYEE (Emp_id char(20)
                        Emp_name char(30) NOT NULL
                        Address char(150)
                        PRIMARY KEY(Emp_id))
```

```
CREATE TABLE CHILDREN (First_name Char(10)
                        Middle_name char(10)
                        Last_name char(10)
                        Emp_id char(20)
                        Birthday date
                        Gender char(1))
```

```
PRIMARY KEY (First_name, Middle_name, Last_name, Emp_id)
FOREIGN KEY (Emp_id) REFERENCES EMPLOYEE)
```

Chuyển đổi các kiểu thực thể sang quan hệ (tt)

- Chuyển đổi các liên kết giữa hai kiểu thực thể
 - Chuyển đổi liên kết một – nhiều
 - Đưa thuộc tính khóa chính của bên một thành thuộc tính ngoại lai của bên nhiều
 - Nếu đầu liên kết nhiều là ràng buộc bắt buộc thì khóa ngoại lai sẽ tham gia vào khóa chính của quan hệ



```
CREATE TABLE CUSTOMER (Cust_id char(20)
```

```
    Cust_name char(30) not nul
```

```
    Address char(150)
```

```
    PRIMARY KEY (Cust_id))
```

```
CREATE TABLE ORDER (Oder_id char(20)
```

```
    Cust_id char(20) NOT NULL
```

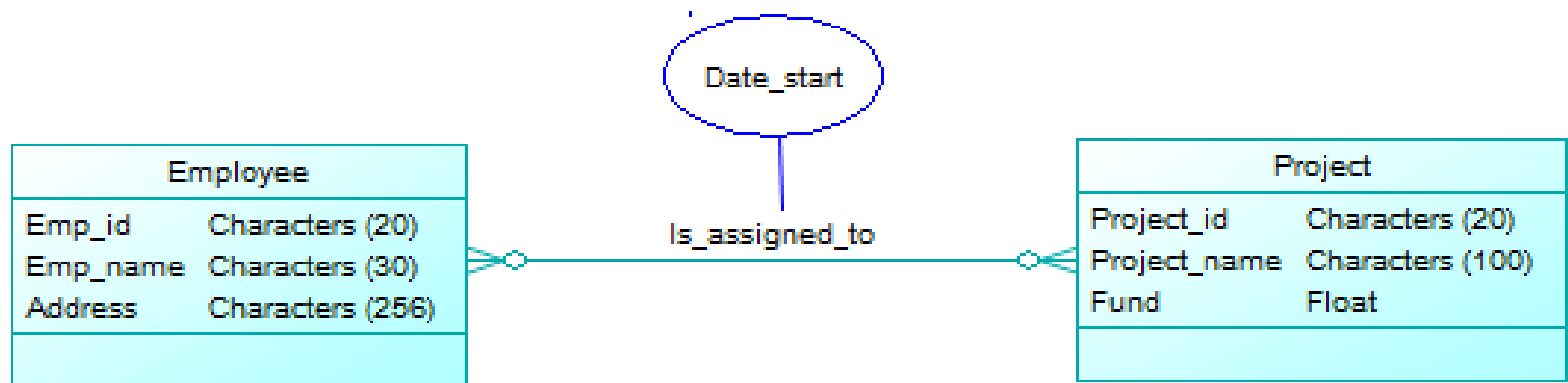
```
    Order_desc char(200)
```

```
    PRIMARY KEY (Order_id),
```

```
    FOREIGN KEY (Cust_id) REFERECES CUSTOMER)
```

Chuyển đổi các kiểu thực thể sang quan hệ (tt)

- Chuyển đổi liên kết nhiều – nhiều
 - Tạo thêm một quan hệ mới: khóa chính là khóa chính của hai quan hệ, các thuộc tính mô tả liên kết là thuộc tính của quan hệ



Chuyển đổi các kiểu thực thể sang quan hệ (tt)

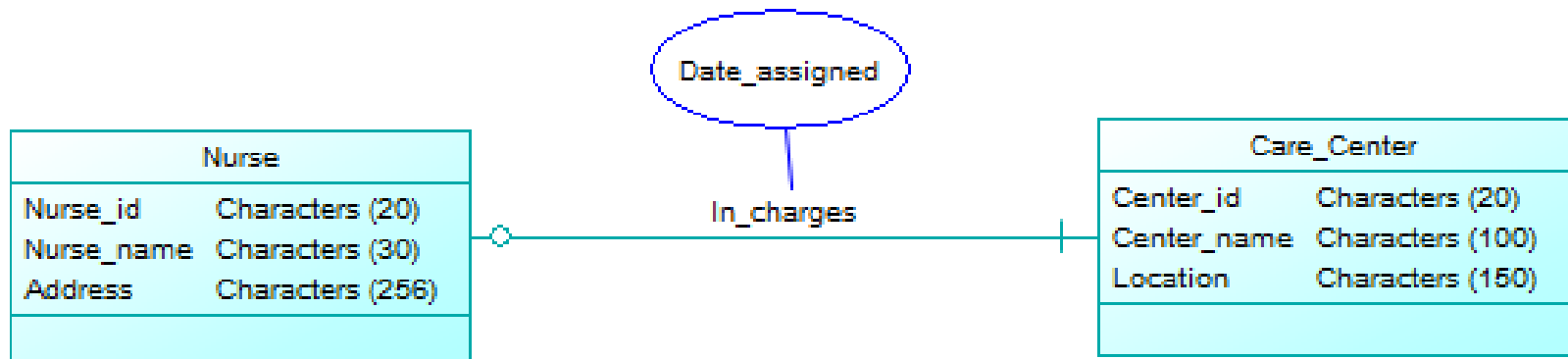
```
CREATE TABLE EMPLOYEE (Emp_id      char(20),
                        Emp_name     char(30) NOT NULL,
                        Address       char(150),
                        PRIMARY KEY(Emp_id))

CREATE TABLE PROJECT (Project_id   char(20),
                        Project_name  char(100) NOT NULL,
                        Fund          number(10),
                        PRIMARY KEY(Project_id))

CREATE TABLE EMP_PROJECT (Emp_id   char(20),
                            Project_id char(20),
                            Date_start date,
                            PRIMARY KEY(Emp_id, Project_id),
                            FOREIGN KEY (Emp_id) REFERENCES EMPLOYEE,
                            FOREIGN KEY (Project_id) REFERENCES PROJECT)
```

Chuyển đổi các kiểu thực thể sang quan hệ (tt)

- Chuyển đổi các liên kết một – một
 - Khóa chính của một quan hệ sẽ được ánh xạ thành khóa ngoại lai của của quan hệ còn lại
 - Phía bên bắt buộc trở thành quan hệ có khóa ngoại lai lấy từ khóa chính phía quan hệ tùy chọn



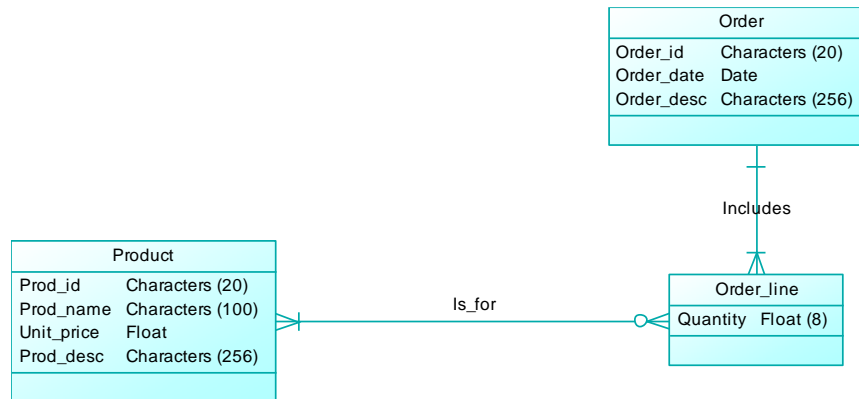
Chuyển đổi các kiểu thực thể sang quan hệ (tt)

```
CREATE TABLE NURSE (Nurse_id char(20),  
    Nurse_name char(30) NOT NULL,  
    Address char(150),  
    PRIMARY KEY(Nurse_id))
```

```
CREATE TABLE CARE_CENTER (Center_id char(20),  
    Center_name char(100) NOT NULL,  
    Nurse_id char(20) NOT NULL,  
    Location char(200),  
    Date_assigned date,  
    PRIMARY KEY(Center_id),  
    FOREIGN KEY (Nurse_id) REFERENCES NURSE)
```

Chuyển đổi các kiểu thực thể sang quan hệ (tt)

- Chuyển đổi các kiểu thực thể kết nối
 - Trường hợp thuộc tính định danh không xác định: khóa chính của quan hệ này gồm các thuộc tính khóa chính của hai quan hệ còn lại

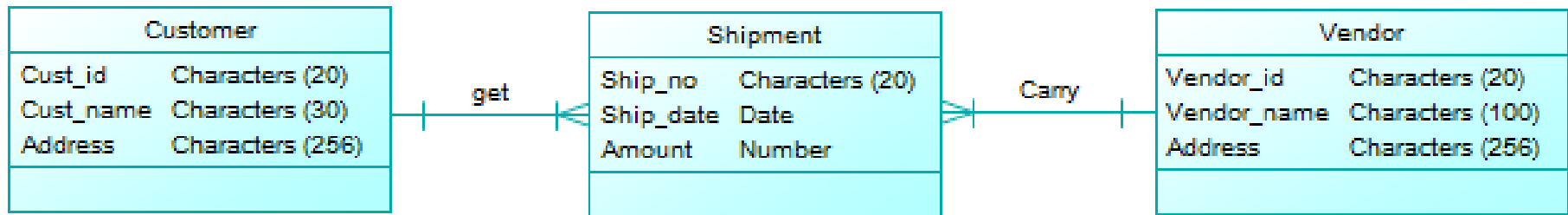


Chuyển đổi các kiểu thực thể sang quan hệ (tt)

```
CREATE TABLE ORDER (Oder_id      char(20)
                      Cust_id char(20) NOT NULL
                      Order_desc   char(200)
                      PRIMARY KEY (Order_id),
CREATE TABLE PRODUCT (Prod_id   char(20),
                      Prod_name    char(100) NOT NULL,
                      Unit_price   number(10) NOT NULL,
                      PRIMARY KEY(Prod_id))
CREATE TABLE ORDER_LINE (Oder_id      char(20),
                      Prod_id char(20),
                      Quantity        number(10) NOT NULL,
                      PRIMARY KEY(Oder_id, Prod_id),
                      FOREIGN KEY (Oder_id) REFERENCES ORDER,
                      FOREIGN KEY (Prod_id) REFERENCES PRODUCT
```

Chuyển đổi các kiểu thực thể sang quan hệ (tt)

- Chuyển đổi các kiểu thực thể kết nối (tt)
 - Trường hợp thuộc tính định danh đã xác định
 - Khi định danh dễ dàng xác định phù hợp với bài toán thực tế
 - Khi định danh xác định bằng kết hợp định danh của hai kiểu quan hệ liên kết không thể xác định duy nhất
 - Khóa chính là thuộc tính định danh, các khóa chính của hai quan hệ liên kết là khóa ngoại lai



```

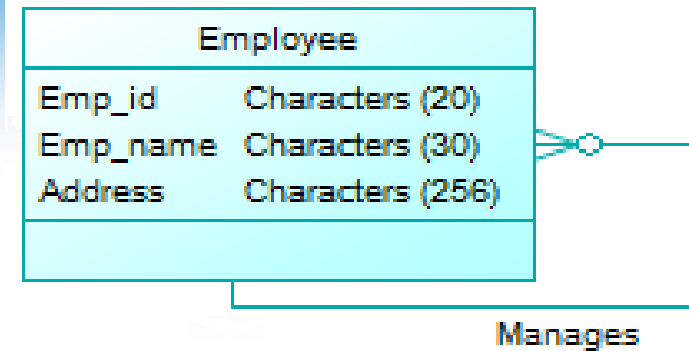
CREATE TABLE CUSTOMER ( Cust_id          Char(20) NOT NULL,
                          Cust_name      Char(30) NOT NULL, Address  Char(150)
                          PRIMARY KEY (Cust_id))
    
```

```

CREATE TABLE VENDOR (Vendor_id  char(20),
                      Vendor_name char(100) NOT NULL, Address  Char(150),
                      PRIMARY KEY(Vendor_id))
    
```

```

CREATE TABLE SHIPMENT (Ship_no  char(20),
                        Cust_id char(20) NOT NULL, Vendor_id char(20) NOT NULL,
                        Amount number(10) NOT NULL, Ship_date      date
                        PRIMARY KEY(Ship_no),
                        FOREIGN KEY (Cust_id) REFERENCES CUSTOMER,
                        FOREIGN KEY (Vendor_id) REFERENCES VENDOR)
    
```



- Chuyển đổi các liên kết đơn

- Chuyển đổi liên kết đơn một – nhiều

- Một thuộc tính đóng vai trò khóa ngoại lai sẽ được đưa thêm vào quan hệ và liên kết với khóa chính (khóa ngoại lai đệ quy)

```
CREATE TABLE EMPLOYEE (Emp_id char(20)
```

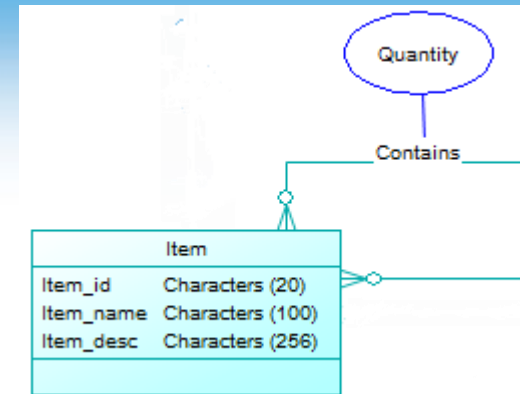
```
    Emp_name      char(30)
```

```
    Manager_id    char(20)
```

```
    Address       char(150)
```

```
PRIMARY KEY (Emp_id)
```

```
FOREIGN KEY (Manager_id) REFERENCES EMPLOYEE
```

- Chuyển đổi các liên kết đơn nhiều – nhiều
 - Tạo ra hai quan hệ: từ kiểu thực thể và từ mối liên kết nhiều - nhiều
 - Khóa chính tạo nên từ 2 thuộc tính đều là khóa của quan hệ được tạo nên từ thực thể

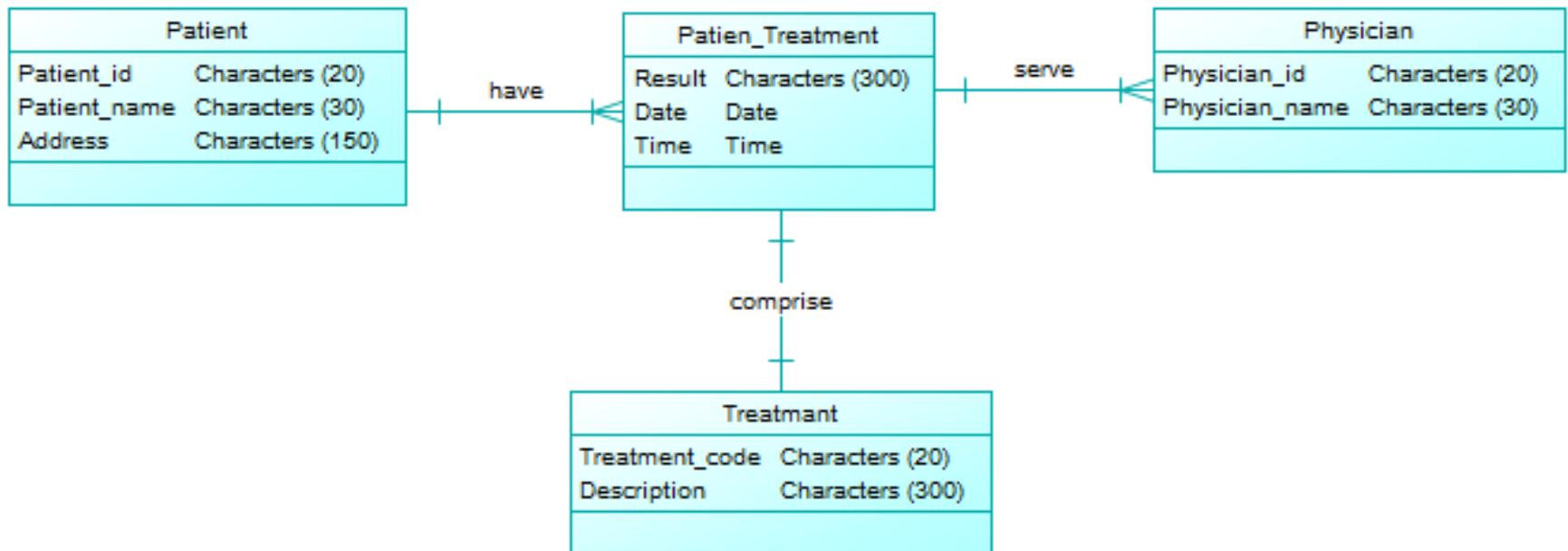
```

CREATE TABLE ITEM (Item_id      char(20),
                    Item_name     char(30) NOT NULL, Unit_cost  number(10),
                    PRIMARY KEY (Item_id))
  
```

```

CREATE TABLE COMPONENT (Item_id      char(20),
                          Component_id char(20), Quantity     number
                          PRIMARY KEY (Item_id, Component_id),
                          FOREIGN KEY (Component_id) REFERENCES COMPONENT)
  
```

Chuyển đổi liên kết giữa nhiều kiểu thực thể

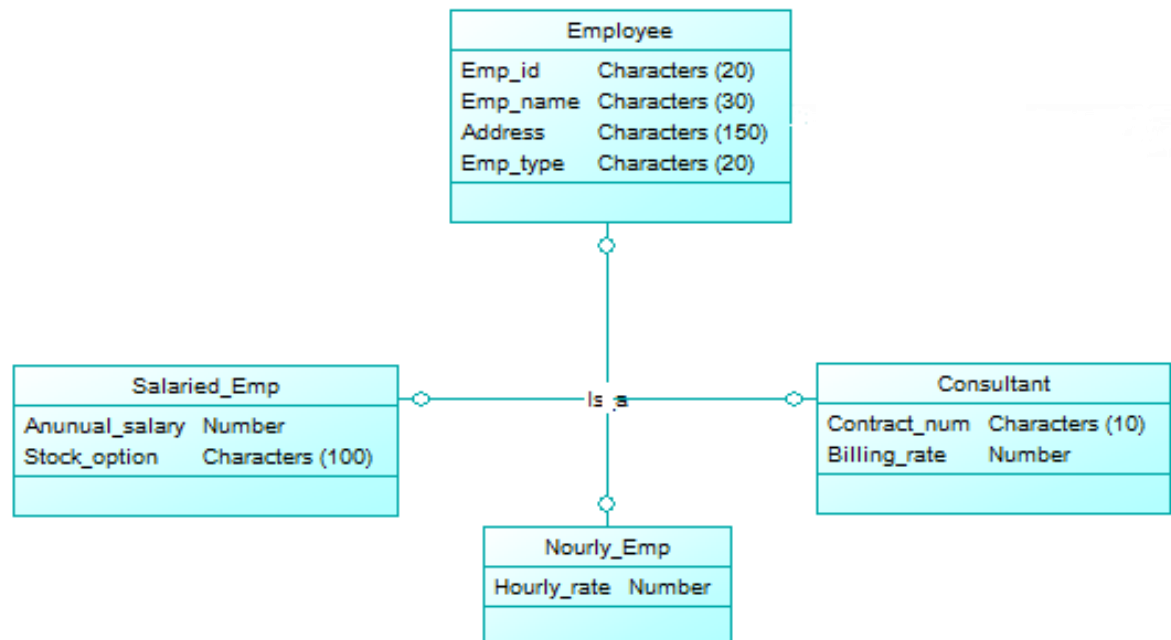


- Mỗi thực thể là một quan hệ
- Thực thể kết nối có khóa là các khóa của các quan hệ có kết nối, có thể bổ sung thêm thuộc tính vào khóa để tạo khóa chính duy nhất

```
CREATE TABLE PATIENT (Patient_id Char(20) NOT NULL,  
    Patient_name Char(30) NOT NULL, Address Char(150)  
    PRIMARY KEY (Patient_id))  
CREATE TABLE PHYSICIAN (Physician_id char(20),  
    Physician_name char(100) NOT NULL,  
    PRIMARY KEY(Physician_id))  
CREATE TABLE TREATMENT (Treatment_code char(20),  
    Description char(300),  
    PRIMARY KEY (Treatment_code))
```

```
CREATE TABLE PATIENT_TREATMENT (Patient_id Char(20),  
    Physician_id char(20), Treatment_code char(20),  
    T_date date, T_Time time,  
    Result char(300) NOT NULL  
PRIMARY KEY (Patient_id, Physician_id, Treatment_code,  
    T_date, T_Time)  
FOREING KEY (Patient_id) REFERENCES PATIENT,  
FOREING KEY (Physician_id) REFERENCES PHYSICIAN  
FOREING KEY (Treatment_code) REFERENCES TREATMENT)
```

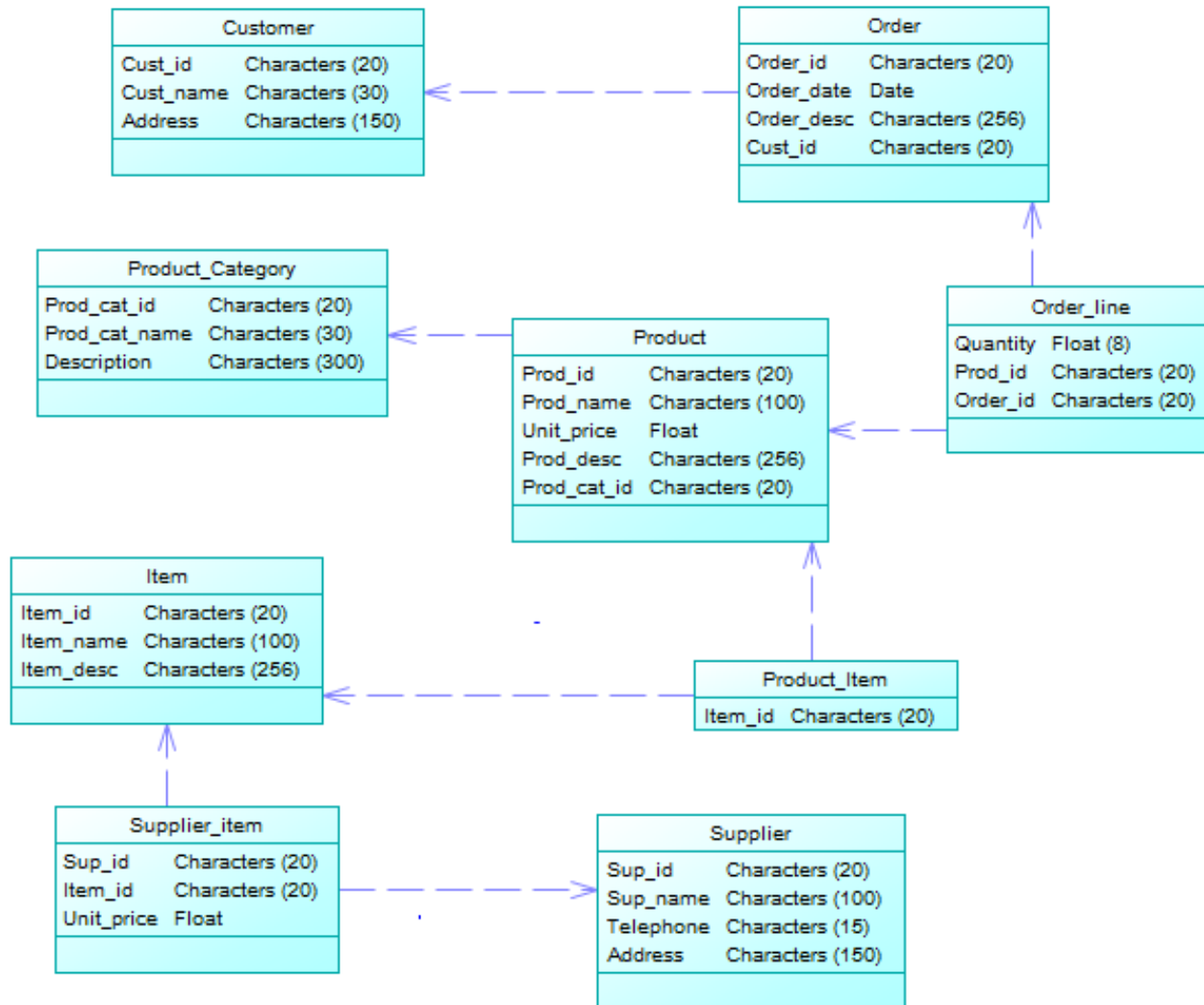
- Chuyển đổi các phân cấp “là-một”
 - Mỗi thực thể là một quan hệ
 - Thuộc tính khóa của quan hệ con lấy từ khóa của quan hệ gốc



```
CREATE TABLE EMPLOYEE (Emp_id char(20)
                        Emp_name char(30) NOT NULL
                        Emp_type char(30) NOT NULL
                        Address char(150)
                        PRIMARY KEY (Emp_id)
CREATE TABLE SALARY_EMPLOYEE (S_Emp_id char(20),
Annual_salary number(7) NOT NULL,
Stock_option char(100),
PRIMARY KEY(S_Emp_id)
FOREING KEY (S_Emp_id) REFERENCES EMPLOYEE
```

```
CREATE TABLE HOURLY_EMPLOYEE (H_emp_id
    char(20),
    Hourly_rate  number(5) NOT NULL
    PRIMARY KEY(H_emp_id),
    FOREIGN KEY (H_emp_id) REFERENCES EMPLOYEE
CREATE TABLE CONSULTANT (C_emp_id char(20),
    Contract_num      char(10) NOT NULL
    Bill_rate         number(8) NOT NULL
    PRIMARY KEY(C_emp_id),
    FOREIGN KEY (C_emp_id) REFERENCES EMPLOYEE
```


Sơ đồ CSDL mức logic

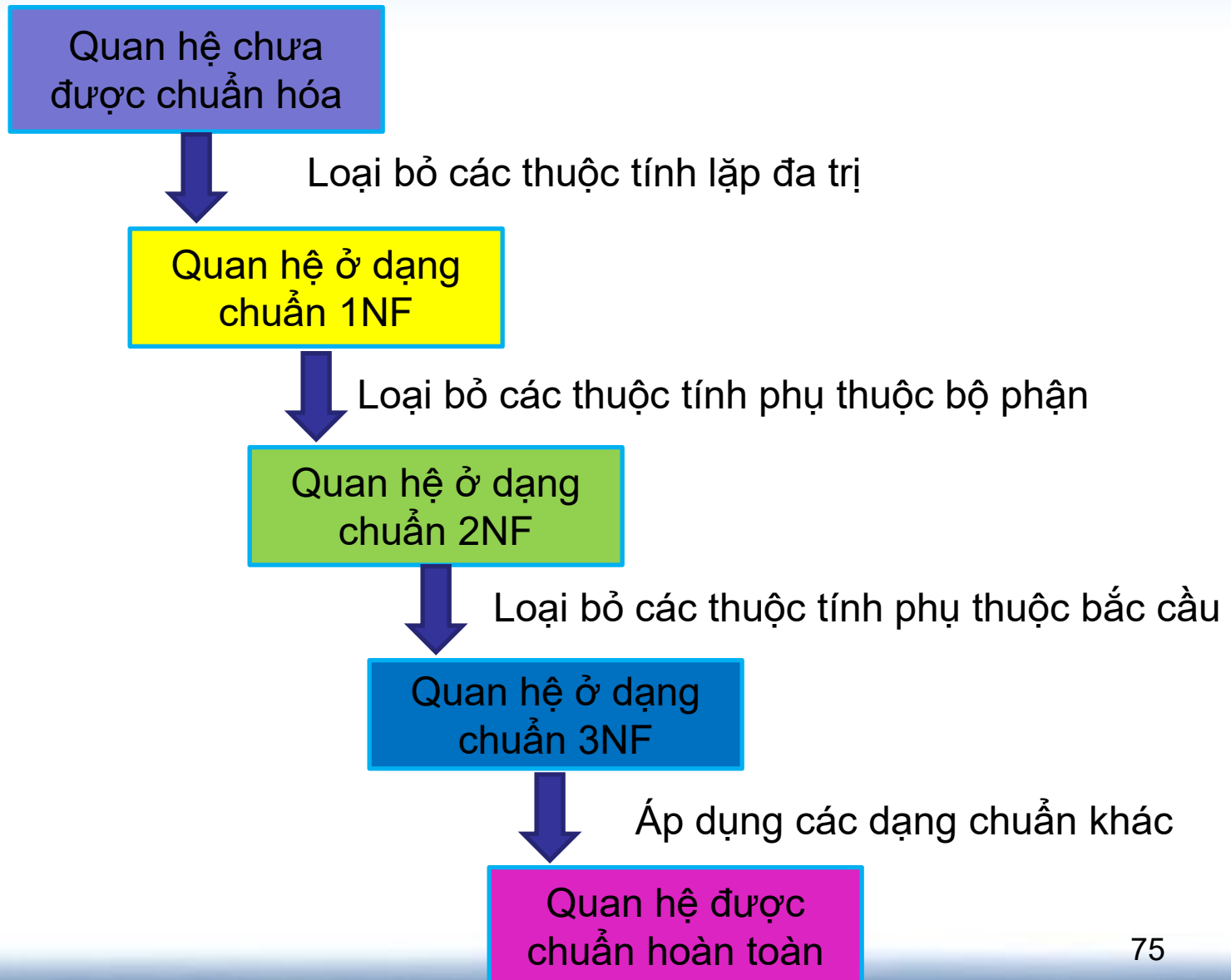


4. Tinh chỉnh dữ liệu và chuẩn hóa

- Không toàn vẹn dẫn đến:
 - Dị thường cập nhật: khi dữ liệu được lưu trữ nhiều nơi, khi sửa phải sửa tất cả, nếu bỏ sót dẫn đến không nhất quán dữ liệu
 - Dị thường khi thêm mới: đôi khi việc thêm mới không thực hiện được do phải thêm những dữ liệu khác tại thời điểm thêm chưa có
 - Dị thường xóa bỏ: khi xóa bỏ một số dữ liệu có thể đánh mất dữ liệu khác không chủ ý dẫn đến mất mát dữ liệu

Chuẩn hóa

- 1NF
- 2NF
- 3NF
- BCNF



Phi chuẩn

- Gộp thành một quan hệ
 - Hai kiểu thực thể liên kết 1-1 và truy suất quan hệ thường xuyên
 - Quan hệ 1-n và kiểu thực thể đầu 1 không tham gia vào bất cứ liên kết nào khác
- Phân mảnh: tách quan hệ thành nhiều bảng riêng
 - Chiều ngang: Chia quan hệ nhiều vùng khác nhau có cùng cấu trúc trường
 - Chiều dọc: Chia các cột của hệ thành những tập khác nhau, thuộc tính tham gia khóa chính có mặt ở tất cả các tập
- Lưu trữ dữ liệu dư thừa trong các bảng
- Lưu trữ dữ liệu tóm tắt trong các bảng
- Ưu điểm, nhược điểm

Tổ chức CSDL mức vật lý

- Tổ chức tệp tuần tự
- Tổ chức tệp chỉ mục
- Tổ chức tệp băm
- Tổ chức các tệp được phân cụm

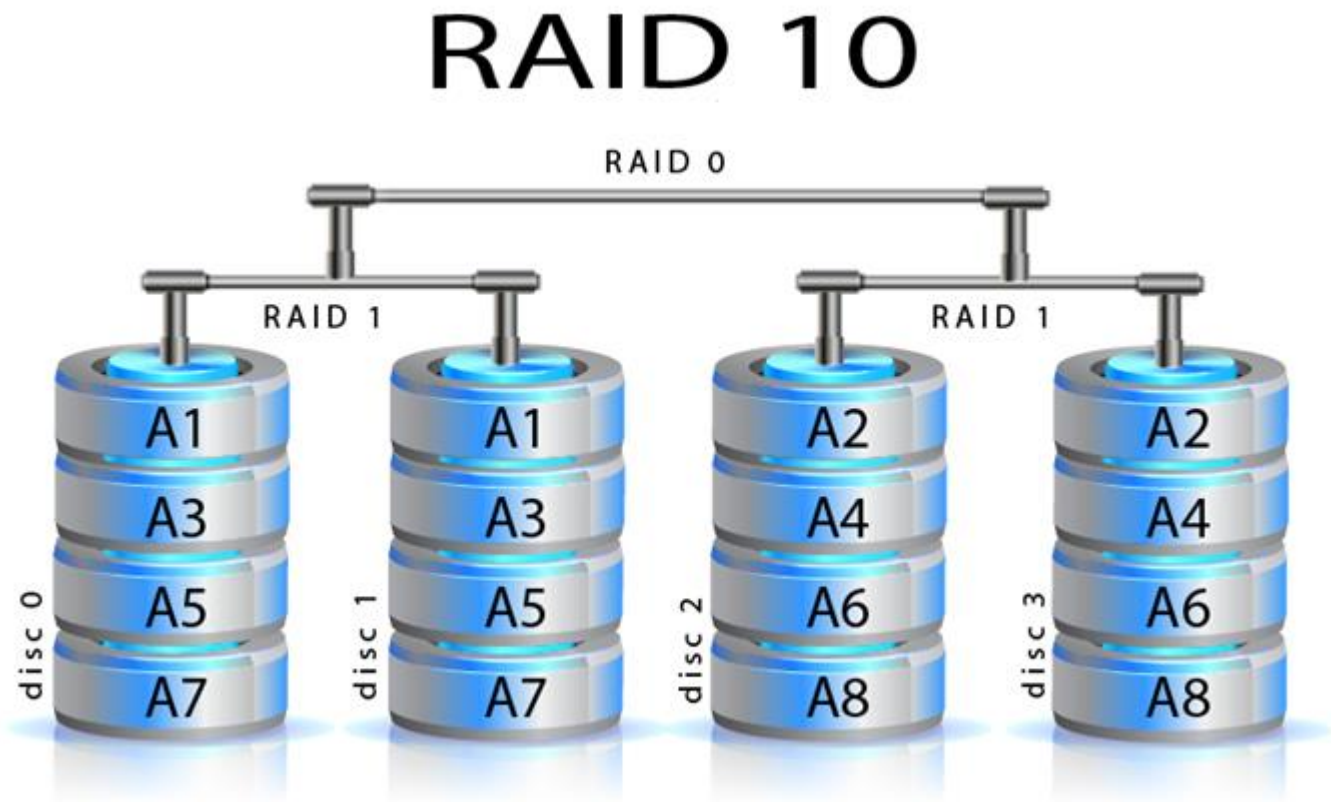
Lựa chọn và sử dụng các loại chỉ mục

- Khi nào xây dựng chỉ mục
- Lựa chọn khóa tìm kiếm
- Khóa tìm kiếm đa thuộc tính
- Khi nào sử dụng phân nhóm
- Lựa chọn chỉ mục băm hay cây cân bằng
- Cân đối chi phí và bảo trì chỉ mục

RAID

Redundant Arrays of Inexpensive Disks hoặc Redundant Arrays of Independent Disks

- RAID 0
- RAID 1
- RAID 2
- RAID 3
- RAID 4
- RAID 5
- RAID 6
- RAID 10



Thiết kế an toàn bảo mật

- Xác định nhóm sử dụng và quyền hạn
- Xác định quyền hạn thao tác trên dữ liệu
- Mã hóa dữ liệu
- Vai trò người quản trị

**Cảm ơn các em
đã chú ý lắng nghe**

**Những em chưa chú ý vẫn được cảm ơn
bình thường**