

Thiết kế



Thiết kế hệ thống quản lý thông tin trong Công ty

❖ Các thực thể:

- NhanVien (mã NV, giới tính, ngày sinh, lương, số điện thoại, họ tên)
- PhongBan (mã phòng, tên phòng, địa điểm)
- NguoiPhuThuoc (họ tên, giới tính, ngày sinh)
- DuAn (Mã DA, tên DA, địa điểm)

❖ Các ràng buộc

- 1 nhân viên sẽ bị một NV quản lý giám sát, 1 người quản lý có thể giám sát nhiều nhân viên khác
- 1 NV phải thuộc về 1 phòng ban, 1 phòng ban có thể có 0, 1 hoặc nhiều nhân viên
- 1 phòng ban sẽ có 1 nhân viên là trưởng phòng, 1 nhân viên chỉ được là trưởng phòng của tối đa 1 phòng ban, cần lưu lại thông tin ngày bắt đầu làm trưởng phòng
- 1 Phòng ban có nhiều dự án, mỗi dự án chỉ thuộc về 1 phòng ban
- 1 nhân viên có thể làm nhiều dự án khác nhau, mỗi dự án có thể được làm bởi nhiều nhân viên, cần lưu lại số giờ làm việc của mỗi nhân viên trong mỗi dự án
- 1 nhân viên có nhiều người phụ thuộc. Thông tin người phụ thuộc **của mỗi NV** được xác định bằng HoTen

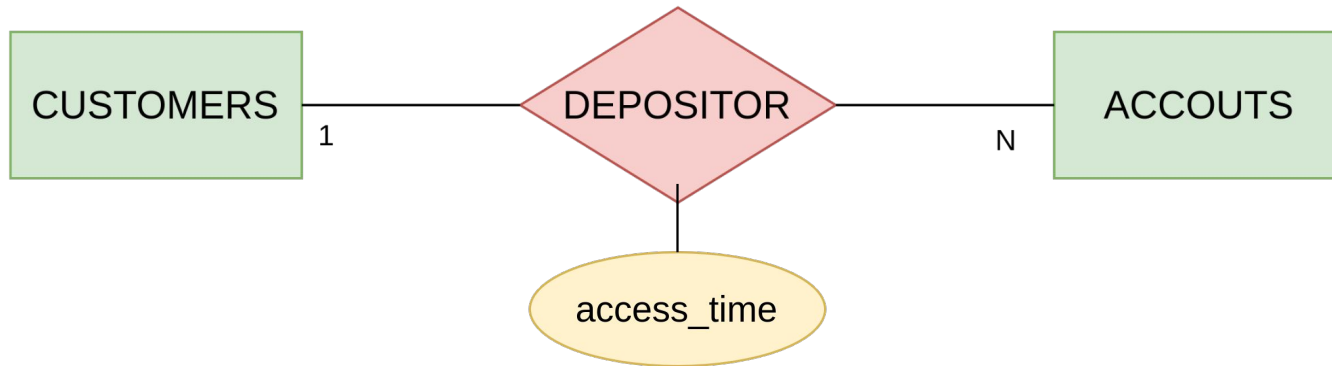
MỘT SỐ VẤN ĐỀ CẦN QUAN TÂM KHI THIẾT KẾ LƯỚI ĐỒ E-R



Đặt vị trí cho các thuộc tính của quan hệ

Đối với Quan hệ 1-N

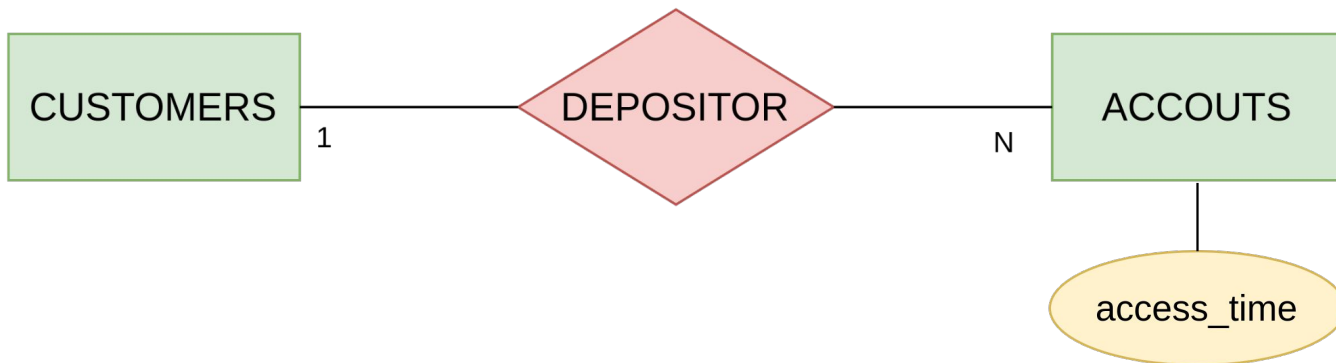
Kịch bản: 1 khách hàng có nhiều tài khoản, và mỗi tài khoản chỉ thuộc về 1



Ảnh hưởng của ánh xạ lực lượng liên kết lên các khóa

Đối với Quan hệ 1-N

Kịch bản: 1 khách hàng có nhiều tài khoản, và mỗi tài khoản chỉ thuộc về 1



Ảnh hưởng của ánh xạ lực lượng liên kết lên các khóa

Đối với Quan hệ N-N

Khóa chính của quan hệ sẽ gồm khóa chính của 2 tập thực thể tham gia vào

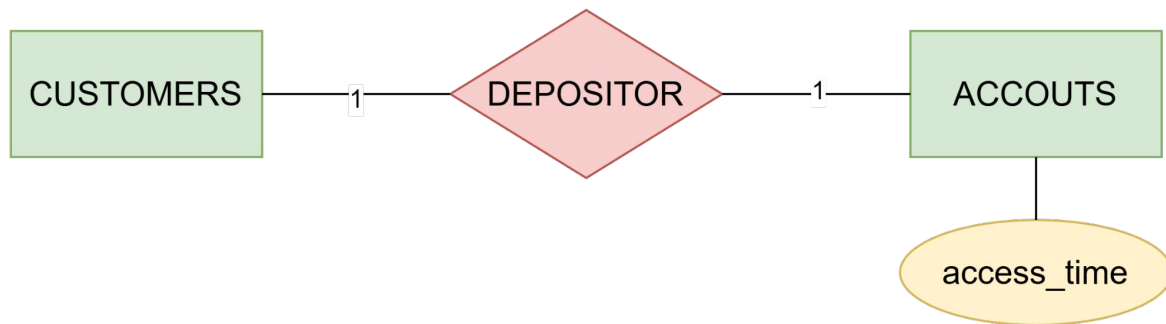


Khóa chính của Registrations là sự **kết hợp khóa chính** của Students và Subjects
Registrations (student id, subject id, ...)

Đặt vị trí cho các thuộc tính của quan hệ

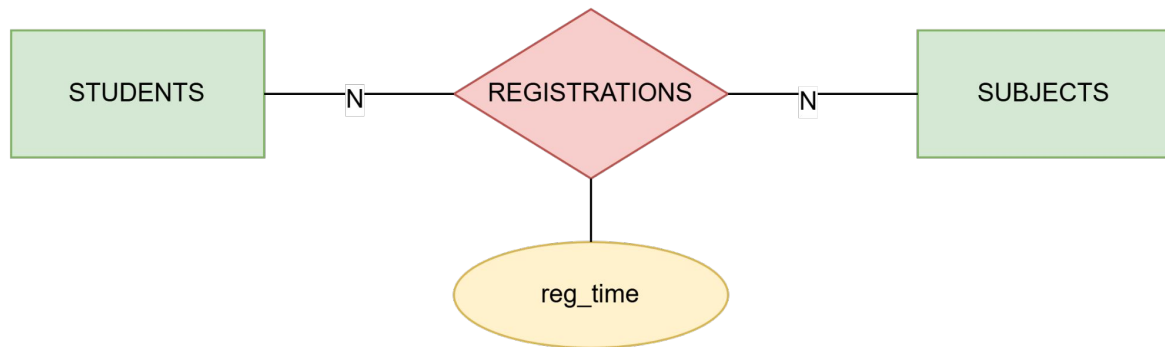
Đối với Quan hệ 1-1

Kịch bản: 1 khách hàng có 1 tài khoản, và mỗi tài khoản chỉ thuộc về 1 Khách hàng.



Đặt vị trí cho các thuộc tính của quan hệ

Đối với Quan hệ N-N

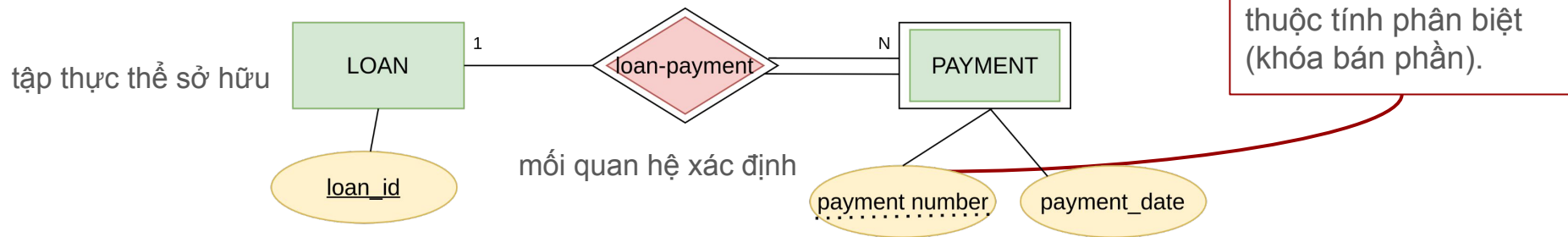


Thuộc tính thời gian đăng ký (reg_time) phải nằm ở Registration chứ không thể nằm ở Students hay Subjects. Nếu nằm ở Students thì sẽ không biết reg_time là thời gian đăng ký môn học nào. Nếu nằm ở Subjects thì sẽ không biết đó thời gian đăng ký của sinh viên nào.

Registrations (student_id, subject_id, reg_time)

Tập thực thể mạnh và Tập thực thể yếu

Tập thực thể không đủ các thuộc tính để hình thành một khóa chính gọi là **tập thực thể yếu**. Tập thực thể có khóa chính được gọi là tập **thực thể mạnh**.



PAYMENT(payment-number, payment date, payment amount).

payment-number là các số liên tiếp từ 1,2,... **riêng cho mỗi khoản nợ**

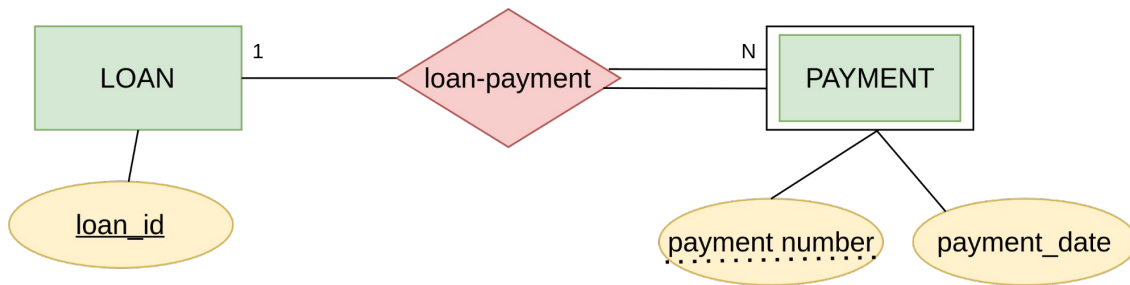
payment-number có thể phân biệt được nếu **trong cùng 1 khoản nợ**. Nhưng nếu đặt trong ngữ cảnh nhiều khoản nợ khác nhau thì payment-number không đủ để phân biệt các lần trả tiền.

⇒ tập PAYMENT không có khóa chính và chỉ là một tập thực thể yếu.

Tập thực thể mạnh và Tập thực thể yếu

Khóa chính của một tập thực thể yếu được cấu thành bởi khóa chính của tập thực thể xác định và thuộc tính phân biệt của tập thực thể yếu.

Khóa chính của PAYMENT sẽ là (**loan-id, payment-number**),

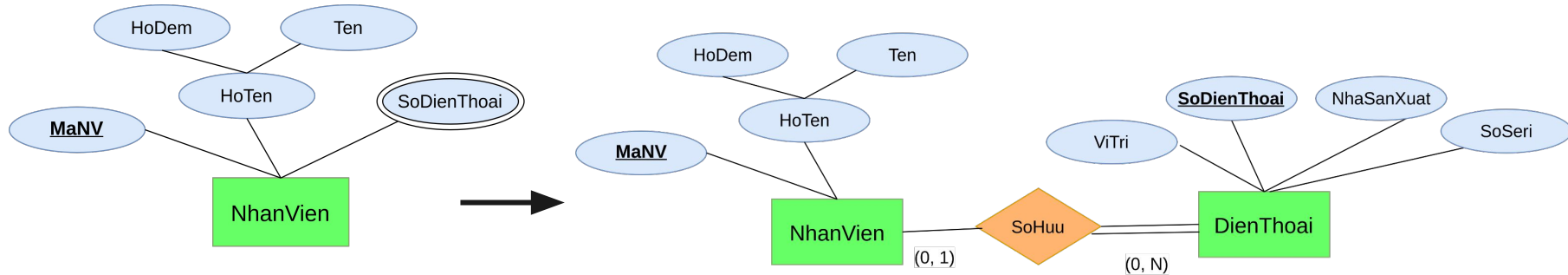


Tập thực thể hay Tập thuộc tính

NhanVien(MaNV, HoDem, Ten, SoDienThoai)

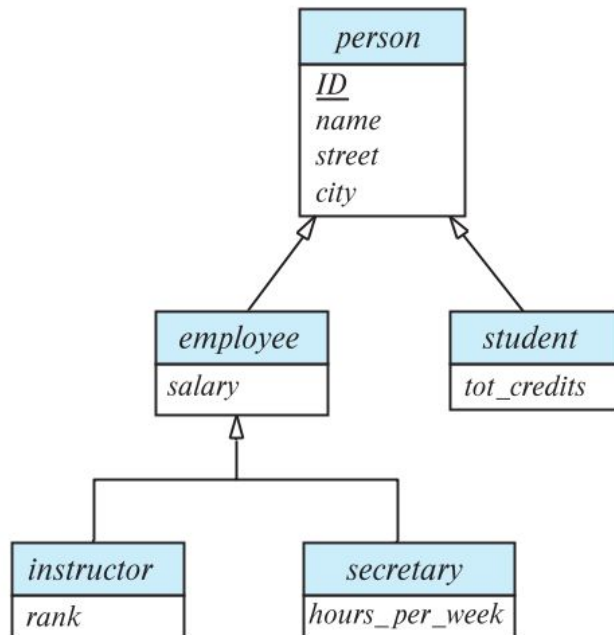
Nếu điện thoại có thể được coi là một thực thể

DienThoai(SoDienThoai, ViTri, NhaSanXuat, SoSeri)

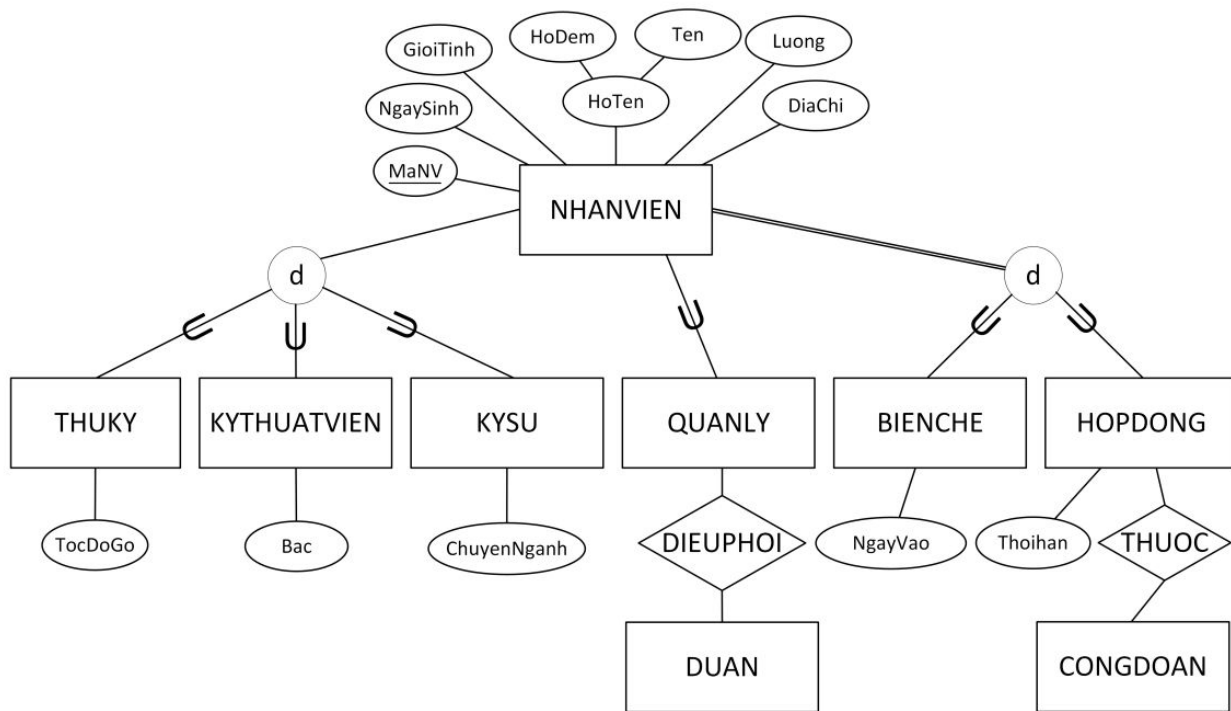


Lớp cha và lớp con

VD: person là 1 lớp cha, employee và student là lớp con.



Lớp cha và lớp con



Lớp cha và lớp con



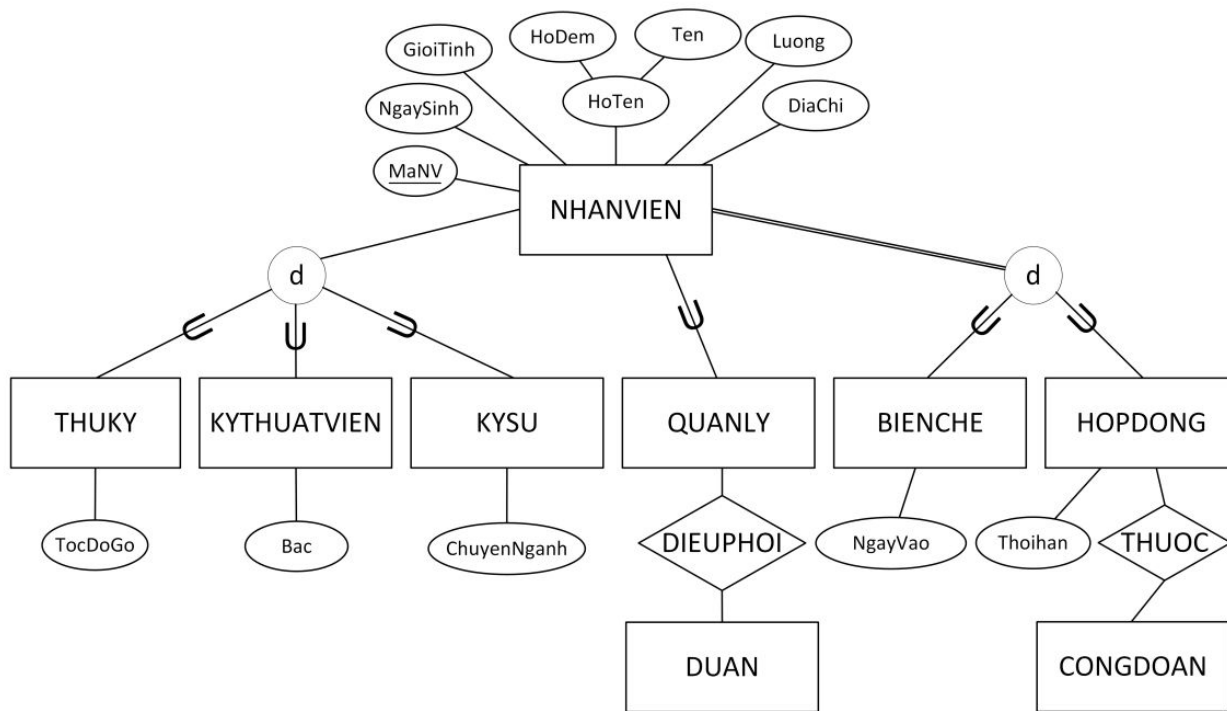
- ❖ Trong nhiều trường hợp một kiểu thực thể có nhiều kiểu con (subtype) hoặc lớp con (subclass) cần phải biểu diễn rõ ràng
- ❖ Ví dụ: các thực thể trong kiểu thực thể NHANVIEN có thể được chia thành các nhóm nhỏ hơn như THUKY, KYSU, KYTHUATVIEN; BIENCHE, HOPDONG...
 - Một thực thể có thể thuộc một hoặc nhiều lớp con, ví dụ: một kỹ sư có thể thuộc hai lớp con KYSU và HOPDONG
 - Một thực thể không thể tồn tại ở dạng một lớp con độc lập, phải có lớp cha
- ❖ Kiểu liên kết lớp cha/lớp con có tỷ số lực lượng 1:1

Lớp cha và lớp con



- ❖ Một thực thể ở lớp con là một thực thể như trong mô hình E-R
- ❖ Một thực thể thuộc một lớp con **thừa kế tất cả các thuộc tính** của lớp cha và tham gia vào liên kết của lớp cha
- ❖ Tập thực thể mức thấp hơn cũng **kế thừa các mối quan hệ** thuộc về tập thực thể mức cao hơn định nghĩa nó.

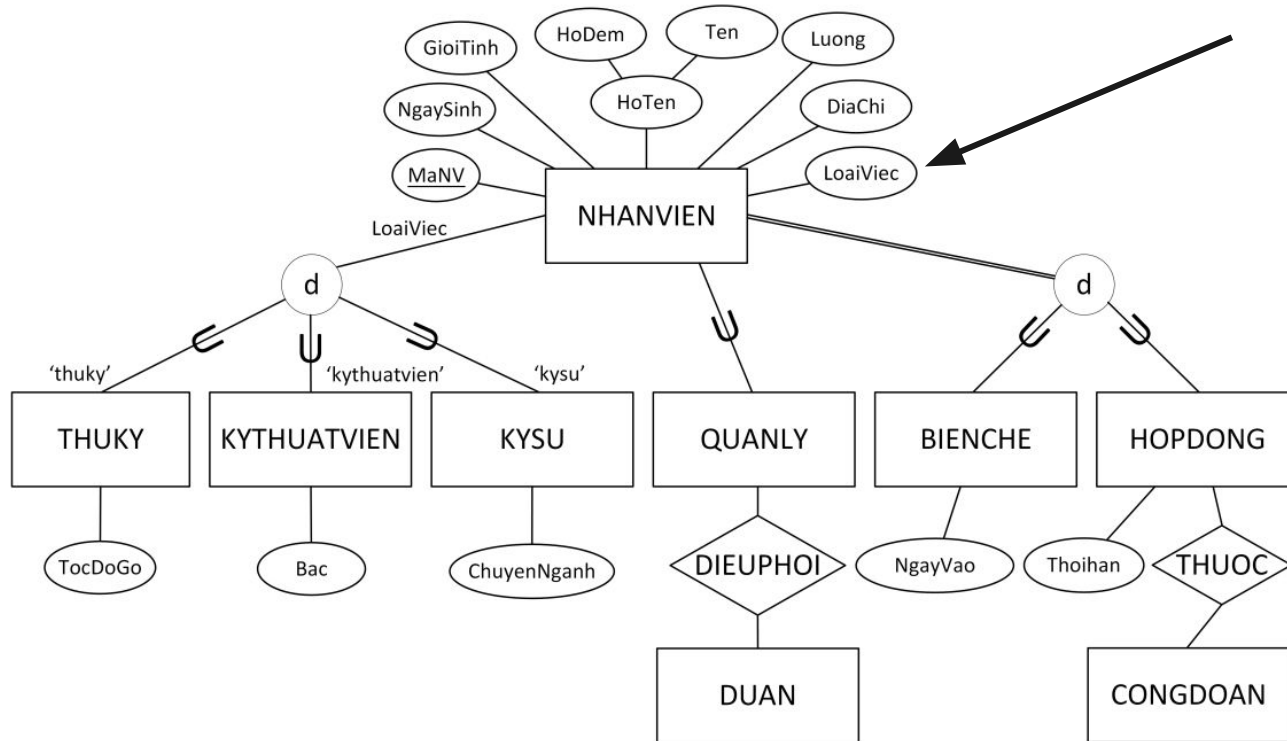
Chuyên biệt hóa (Specialization)



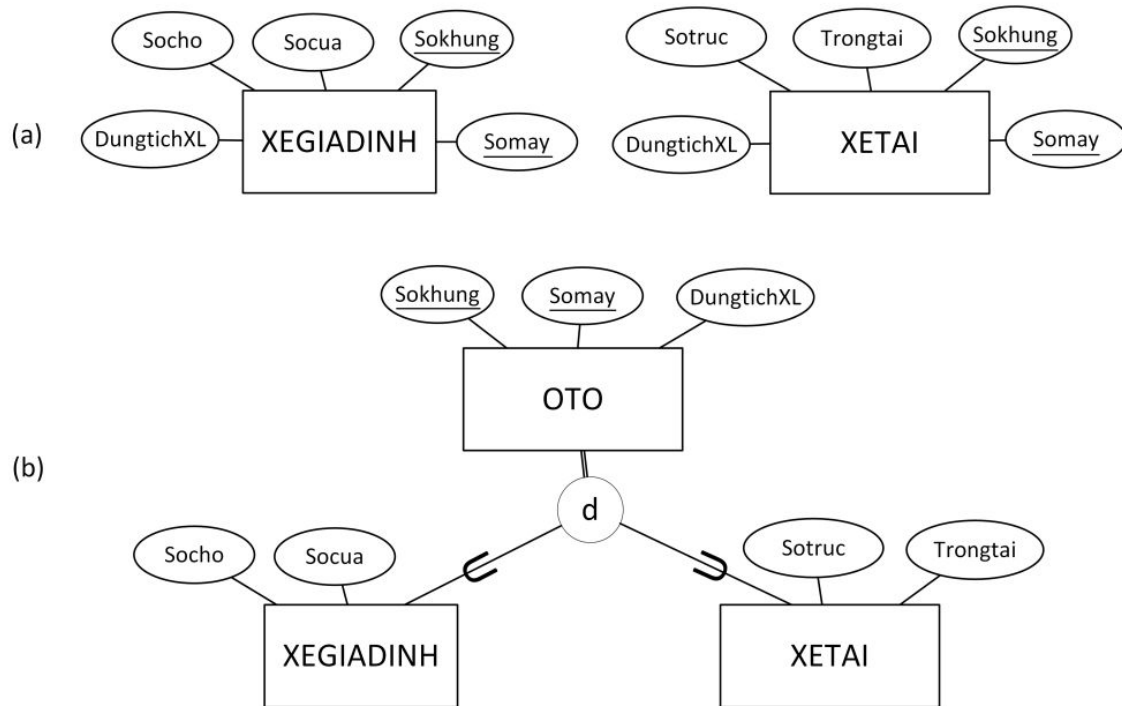
Chuyên biệt hóa (Specialization)

- ❖ **Chuyên biệt hóa (specialization)** là quá trình xác định một tập các lớp con của một kiểu thực thể E. Và E được gọi là lớp cha của chuyên biệt hóa
 - Ví dụ: **THUKY, KYSU, KYTHUATVIEN** là chuyên biệt hóa của lớp cha **NHANVIEN** căn cứ vào loại công việc
 - **BIENCHE, HOPDONG** là một chuyên biệt hóa khác của **NHANVIEN** căn cứ vào cách trả lương
- ❖ Quá trình chuyên biệt hóa cho phép:
 - Xác định một tập các lớp con của một kiểu thực thể
 - Thiết lập các thuộc tính riêng của mỗi lớp con
 - Thiết lập các kiểu liên kết riêng của mỗi lớp con với các kiểu thực thể
 - hoặc các lớp con khác

Chuyên biệt hóa xác định bằng thuộc tính



Tổng quát hóa (Generalization)



Tổng quát hóa (Generalization)



- ❖ **Tổng quát hóa** (generalization) là quá trình ngược với **Chuyên biệt hóa** Xác định các đặc điểm chung của một số kiểu thực thể đã tồn tại E_1, E_2, \dots, E_n và tổng quát hóa chúng thành một lớp cha (mới) E
- ❖ E có các lớp con là E_1, E_2, \dots, E_n

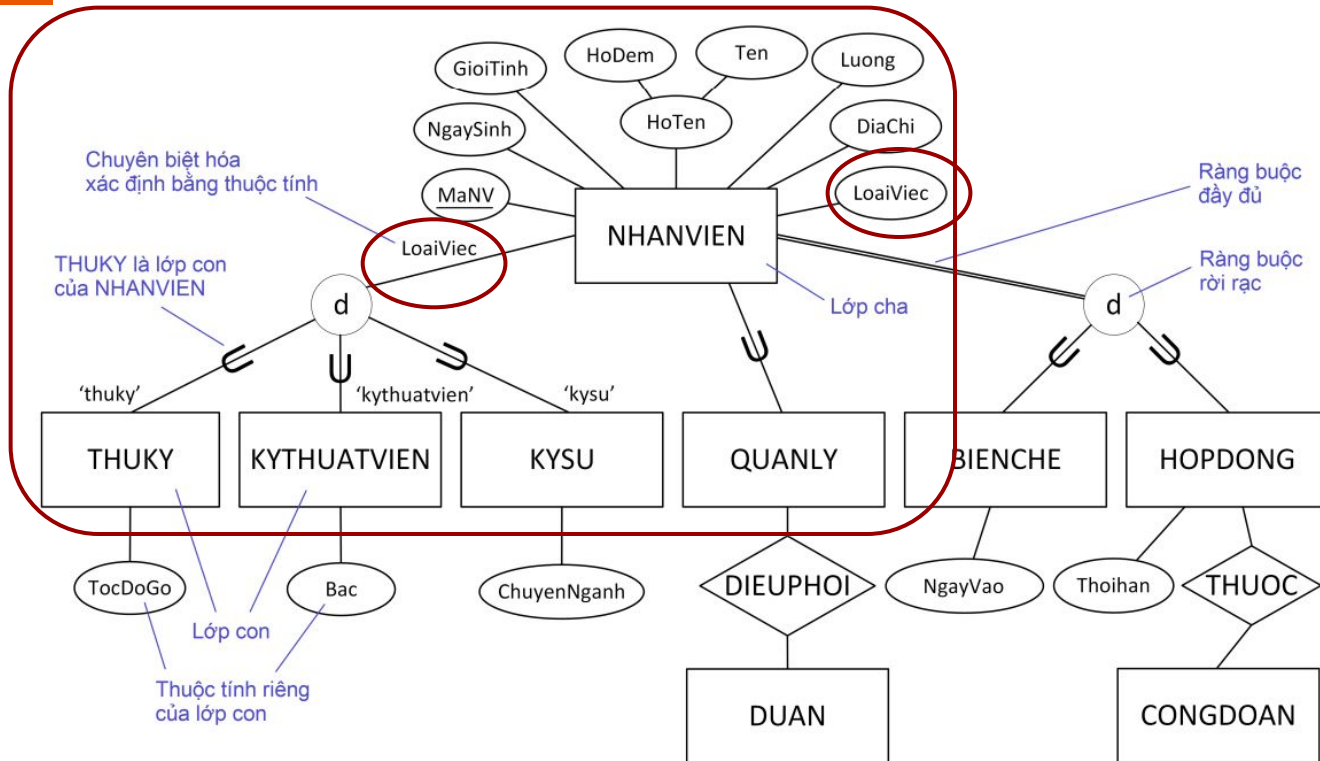
Các ràng buộc của chuyên biệt hóa/tổng quát hóa



Ràng buộc chọn thành viên: Xác định thực thể nào có thể là thành viên của tập thực thể mức thấp hơn.

1. **Mệnh đề xác định:** Thành viên được đánh giá trên cơ sở xác định xem thực thể có thỏa mãn một mệnh đề (điều kiện) tương minh nào đó không.
2. **Người dùng xác định:** Tập các thực thể mức thấp do người dùng xác định *không bị ràng buộc* bởi các điều kiện thành viên

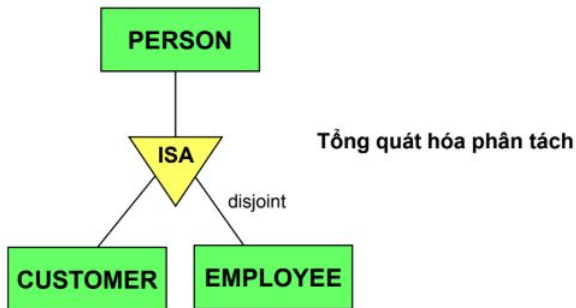
Các ký hiệu sử dụng



Các ràng buộc của chuyên biệt hóa/tổng quát hóa

Ràng buộc rời rạc: Ràng buộc này được thỏa mãn khi mỗi thực thể chỉ là thành viên của nhiều nhất một lớp con, ngược lại mỗi thực thể có thể thuộc vào hai lớp con trở lên

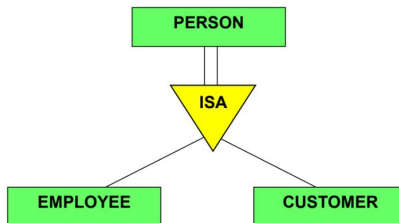
1. **Overlapping (Giao nhau)** – employee và student
2. **Disjoint (Không giao nhau)** – instructor và secretary



Các ràng buộc của chuyên biệt hóa/tổng quát hóa

Ràng buộc đầy đủ: Ràng buộc này được thỏa mãn khi mỗi thực thể của lớp cha phải thuộc vào một lớp con nào đó, ngược lại sẽ có một số thực thể của lớp cha không thuộc bất kỳ lớp con nào

1. **toàn bộ:** một thực thể phải thuộc về một trong các tập thực thể cấp thấp hơn
(VD: sinh viên bắt buộc chia thành 'đã tốt nghiệp' và 'chưa tốt nghiệp')
2. **một phần:** một thực thể không cần phải thuộc về một trong các tập thực thể cấp thấp hơn

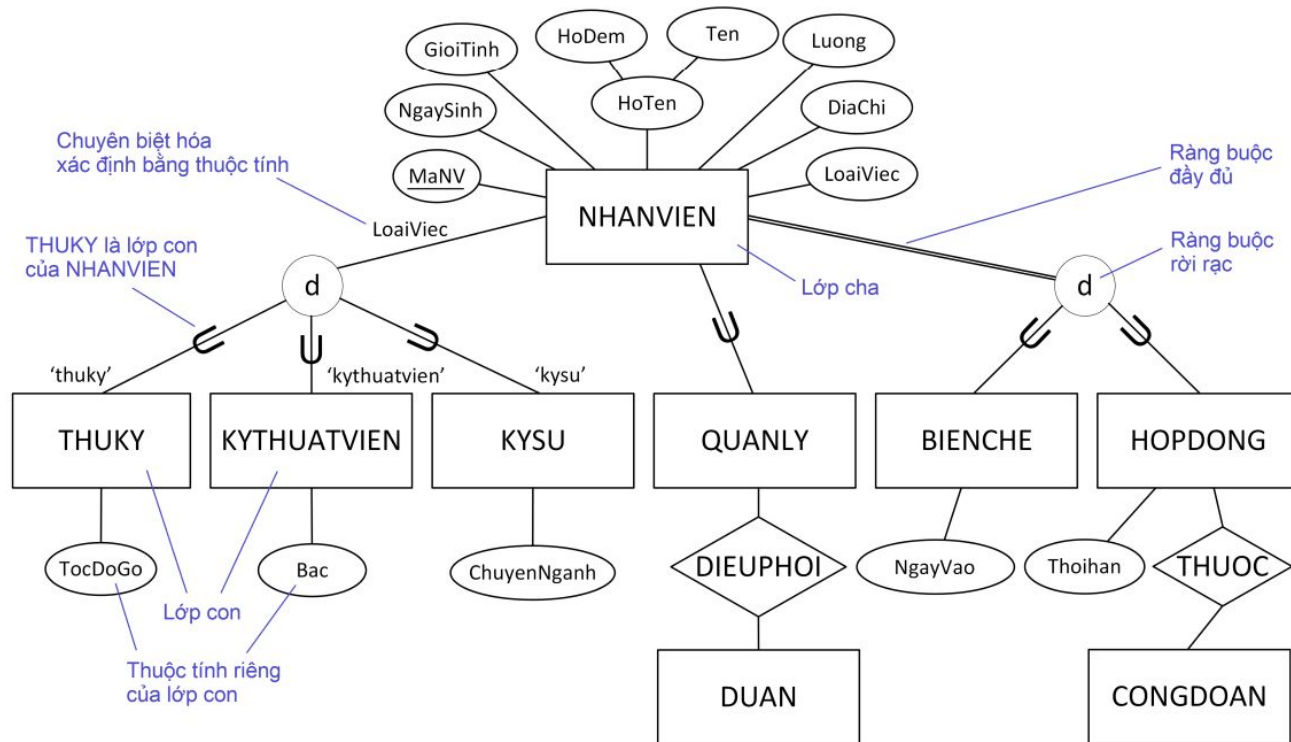


Tổng quát hóa (Specialization)



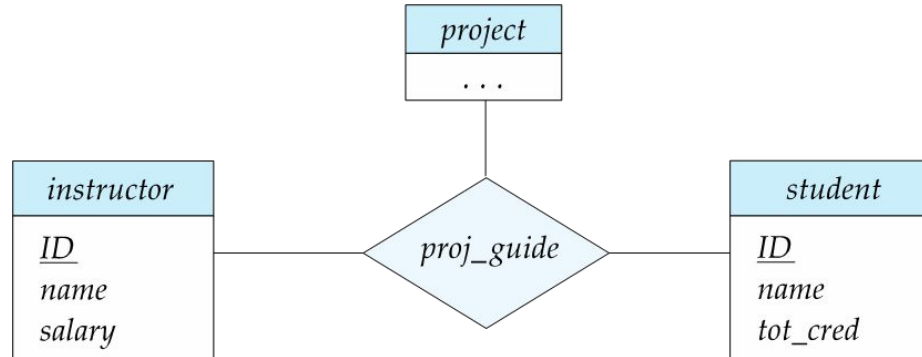
- ❖ Trong sơ đồ E-R, không có sự khác biệt giữa **Tổng quát hóa** và **Chuyên biệt hóa** mà chỉ khác nhau ở cách xem sơ đồ từ dưới lên hay từ trên xuống.

Các ký hiệu sử dụng



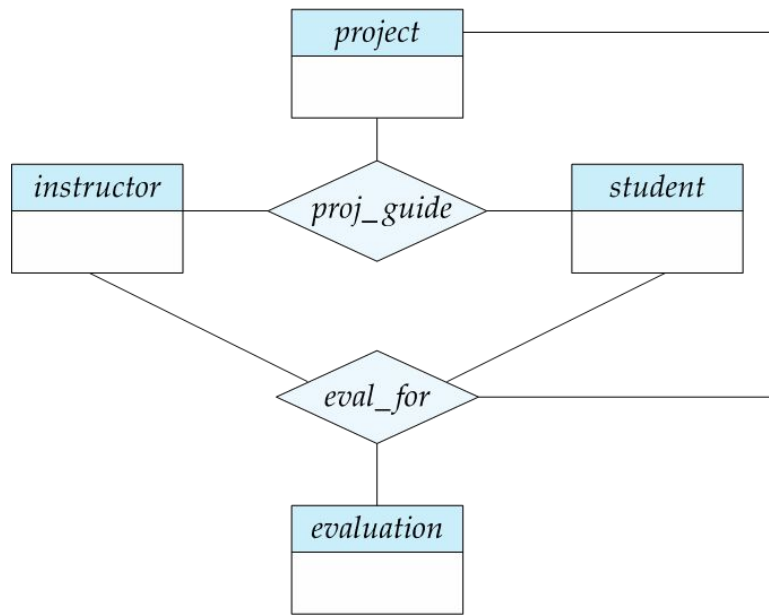
Kết hợp

- ❖ Tập thực thể: project: Bài tập lớn, student: Sinh viên thực hiện, instructor: Giảng viên hướng dẫn
- ❖ Kịch bản 1: Đăng ký bài tập lớn trong trường đại học, cần biểu diễn mối quan hệ giữa 3 thực thể trên \Rightarrow Có mối quan hệ proj_guide



Kết hợp

- ❖ Kịch bản 2: Có thêm evaluation: Hội đồng đánh giá
- ❖ Nếu thiết kế như phía dưới, proj_guide và eval_for sẽ bị trùng lặp dữ liệu

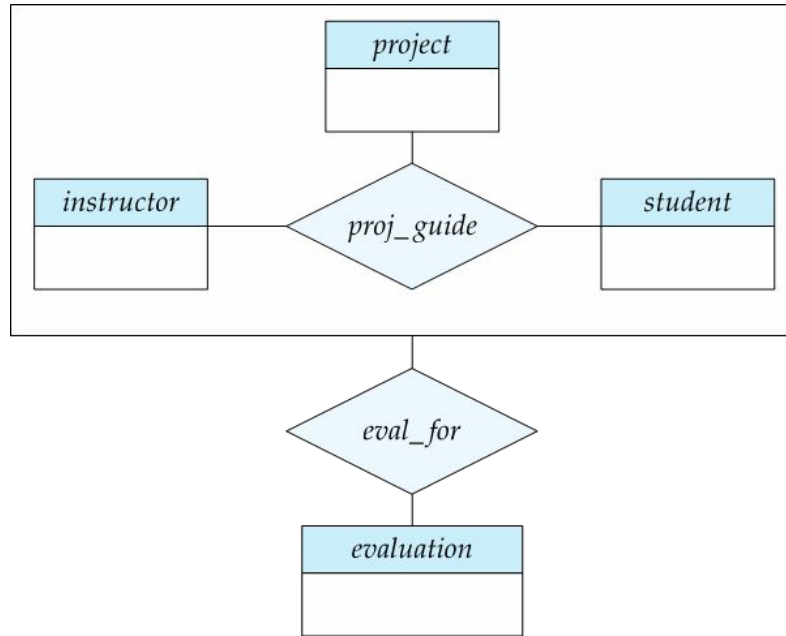


Kết hợp

- ❖ **Hướng xử lý 1:** Tạo mối quan hệ 2 ngôi giữa **evaluation** và **student**
⇒ Mối quan hệ 2 ngôi này sẽ không thể hiện được hội đồng đang đánh giá project nào, của instructor nào hướng dẫn sinh viên
- ❖ **Hướng xử lý 2:** Kết hợp **proj_guide** và **eval_for** gộp thành 1 quan hệ duy nhất
⇒ Không thể thực hiện kết hợp vì một bộ ba (student, project, instructor) có thể không thuộc hội đồng nào

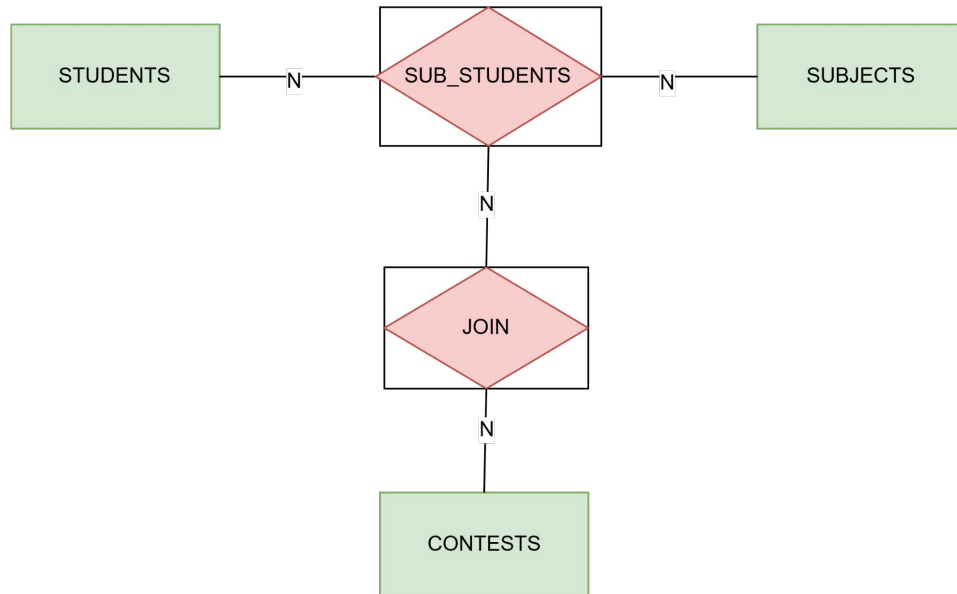
Kết hợp

⇒ Cách tốt nhất để mô hình hóa là sử dụng **kết hợp**. Coi proj_guide là 1 thực thể mức cao. Tức là, coi các mối quan hệ như là các thực thể ở mức cao.



Kết hợp

Ví dụ khác về **Kết hợp**



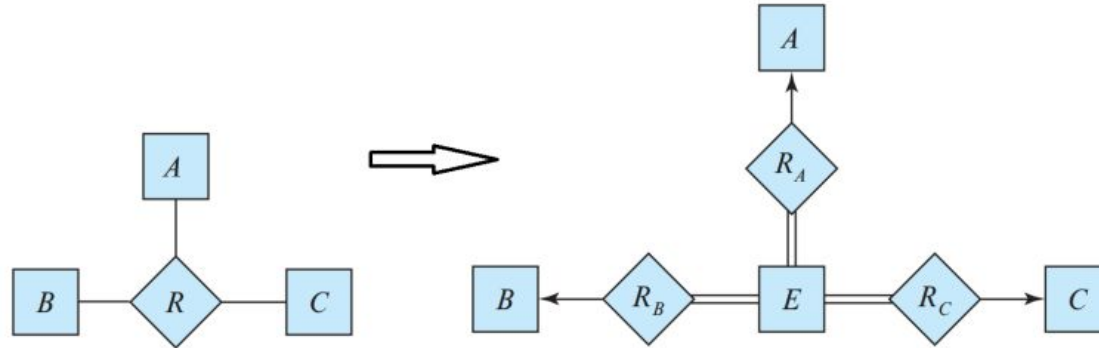
Kết hợp



Ký hiệu "chân chim" chỉ mô tả các liên kết cấp 2, tuy nhiên:

- ❖ Trong thực tế, hầu hết các liên kết đều là **cấp 2**, và
- ❖ Luôn có thể thay thế một liên kết cấp $n \geq 3$ bằng các liên kết **cấp 2**

Chuyển đổi liên kết cấp 3 sang các liên kết cấp 2

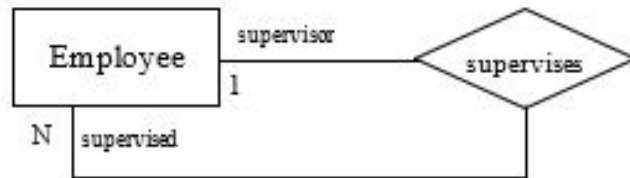
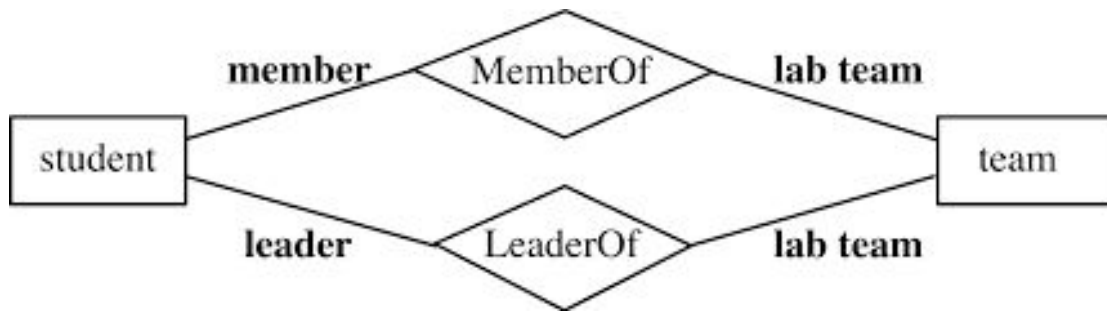


- ❖ Tạo một thực thể mới *E*
- ❖ Tạo liên kết *R_A* có tỷ số lực lượng 1 : N giữa *A* và *E*
- ❖ Tạo liên kết *R_B* có tỷ số lực lượng 1 : N giữa *B* và *E*
- ❖ Tạo liên kết *R_C* có tỷ số lực lượng 1 : N giữa *C* và *E*
- ❖ *E* tham gia toàn bộ vào *R_A*, *R_B*, *R_C*

Vai trò

Các tập thực thể tham gia vào Mối quan hệ *không nhất thiết phải khác nhau*

⇒ Mỗi tập thực thể chỉ cần đóng 1 **vai trò** trong mối quan hệ

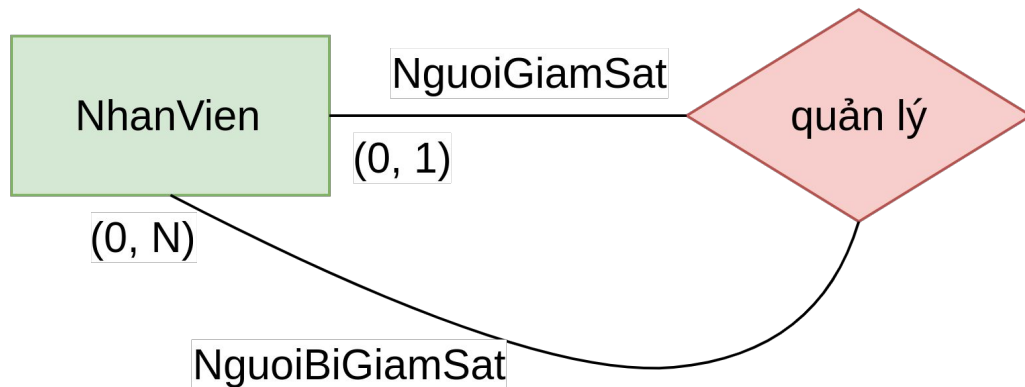


Đệ quy

Điều này xảy ra khi một thực thể (entity) có mối quan hệ với chính nó.

VD: Một nhân viên có thể là **cấp trên** của một nhân viên khác. Thực thể **NhanVien**, trong đó:

- Một nhân viên có thể **quản lý** một hoặc nhiều nhân viên khác
- Một nhân viên có thể **được quản lý bởi** một nhân viên khác



Ràng buộc toàn vẹn tham chiếu

ROLL_NO	NAME	ADDRESS	PHONE	AGE	BRANCH_CODE
1	RAM	DELHI	9455123451	18	CS
2	RAMESH	GURGAON	9652431543	18	CS
3	SUJIT	ROHTAK	9156253131	20	ECE
4	SURESH	DELHI		18	IT

Employee

BRANCH_CODE	BRANCH_NAME
CS	COMPUTER SCIENCE
IT	INFORMATION TECHNOLOGY
ECE	ELECTRONICS AND COMMUNICATION ENGINEERING
CV	CIVIL ENGINEERING

Branch

Không thể INSERT một hàng trong Employee nếu branch_code không có trong danh sách branch_code của bảng Branch. VD: Không thể thêm 1 sinh viên có branch_code là "HELLO", chỉ có thể là 1 trong [CS, IT, ECE, CV]

Thiết kế



Bài 1. Thiết kế cơ bản hệ thống quản lý đăng ký tín chỉ tại 1 trường đại học

Tập thực thể

1. user (admins, instructors, students, employees)
2. departments: Khoa của giảng viên và sinh viên
3. subjects: Môn học
4. courses: Khóa học
5. registrations: Sinh viên đăng ký khóa học
6. sessions (start_time, end_time): Lịch đăng ký dạy học của giảng viên
7. assignments: Bài tập của khóa học
8. buildings: Tòa nhà dạy học
9. rooms: Phòng học
10. terms: Kỳ học

Thiết kế



Bài 1. Thiết kế cơ bản hệ thống quản lý đăng ký tín chỉ tại 1 trường đại học

- ❖ users có các subclass là admins, instructors, students, employees
- ❖ Mỗi sinh viên và giảng viên bắt buộc phải thuộc 1 khoa, 1 khoa không nhất thiết phải có sv và gv
- ❖ 1 môn học có nhiều khóa học, 1 khóa học chỉ thuộc về 1 môn học.
- ❖ 1 kỳ học có nhiều khóa học, 1 khóa học chỉ thuộc về 1 kỳ học.
- ❖ 1 môn học có thể có nhiều môn học tiên quyết (prerequisite)
- ❖ 1 sinh viên có thể đăng ký nhiều khóa học, 1 khóa học có thể được đăng ký bởi nhiều sinh viên, thời gian đăng ký phải được lưu lại
- ❖ 1 giảng viên có thể tham gia dạy nhiều khóa học, 1 khóa học có thể được dạy bởi nhiều giảng viên. Mỗi giảng viên khi tham gia dạy, cần đăng ký danh sách thời gian dạy cho mỗi khóa học (sessions). Mỗi sessions gồm thời gian bắt đầu, thời gian kết thúc, phòng học.
- ❖ 1 khóa học có nhiều bài tập, 1 bài tập chỉ thuộc về 1 khóa học
- ❖ 1 sinh viên khi tham gia vào khóa học có thể nộp nhiều bài tập trong khóa học. Mỗi bài tập trong khóa học có thể được nộp bởi nhiều sinh viên khi tham gia khóa học. Điểm của mỗi lần nộp bài cần được lưu lại

Thiết kế



Bài 2. Hệ thống đặt phòng khách sạn

Các thực thể:

1. User (admin, customers, staffs)
2. Hotel: Khách sạn
3. Room: Phòng khách sạn
4. Service: Các dịch vụ đã được sử dụng trong mỗi phòng sau khi checkout (VD: nước ngọt, đồ ăn, ...)
5. Bill: Hóa đơn đặt phòng

Ràng buộc:

- ❖ Trong hóa đơn đặt phòng cần có thông tin của 1 đại diện khách hàng, 1 nhân viên khách sạn, và **danh sách phòng** được đặt.
- ❖ Cần thiết kế để truy vết được lịch sử đặt phòng. Đặc biệt, **cần lưu lại giá lúc đặt của từng phòng**

Thiết kế

Bài 3. Hệ thống Quản lý rạp chiếu phim

Thiết kế



Bài 4. Hệ thống Quản lý thư viện

Tập thực thể

1. Người mượn
2. Kho sách (chứa nhiều sách)
3. Thủ thư (Người quản lý, 1 thủ thư có thể quản lý nhiều kho sách, 1 kho sách chỉ được quản lý bởi 1 thủ thư)
4. Sách
5. Nhà xuất bản
6. Phiếu mượn (cần có ngày mượn, ngày trả)
7. Phiếu phạt (nếu trả sách muộn sẽ bị nhận Phiếu Phạt)

⇒ Cần tìm cách đặt đúng liên kết của Phiếu phạt với các thực thể khác

Gợi ý



Bài 1. Thiết kế cơ bản hệ thống quản lý đăng ký tín chỉ tại 1 trường đại học

- ❖ sessions là chính là quan hệ giữa instructor và course
- ❖ Tập thực thể liên kết với assignments không đơn giản là students