

INT1313- Cơ Sở Dữ Liệu

Mô Hình Hóa Dữ Liệu Khái niệm và Thiết kế CSDL

Giảng viên: Lê Hà Thanh

lehathanh@ptithcm.edu.vn

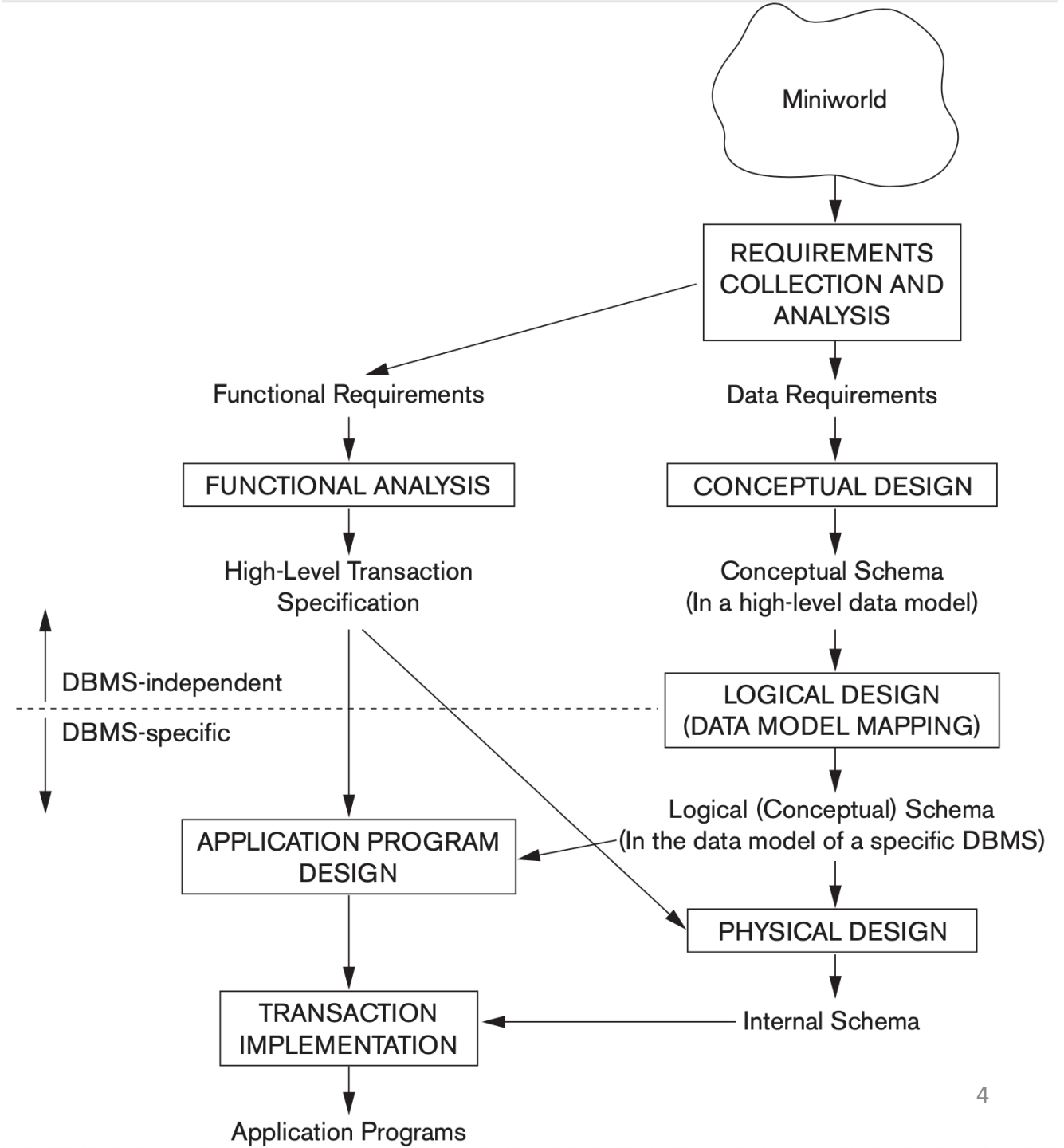
Nội dung

- Mô hình hóa Dữ liệu bằng mô hình Thực thể – Quan hệ (Entity - Relationship, ER)
 - Cấu trúc dữ liệu
 - Các ràng buộc
 - Lược đồ khái niệm
 - Các sơ đồ ER
 - UML và các sơ đồ lớp
- Thiết kế CSDL

Thiết kế CSDL bằng các Mô hình Dữ liệu Khái niệm cấp cao

Tiến trình Thiết kế CSDL

1. Yêu cầu chức năng: các hoạt động hoặc giao dịch tác động lên CSDL
2. Lược đồ khái niệm: mô tả chính xác các yêu cầu dữ liệu, gồm kiểu thực thể, các mối quan hệ, các ràng buộc
 - Mô hình ER
 - Dễ hiểu, ngay cả với người dùng không chuyên môn
 - Đảm bảo tất cả yêu cầu đều được giải quyết và không bị xung đột
 - Đảm bảo các yêu cầu truy vấn và hoạt động chức năng
3. Thiết kế logic/thiết lập ánh xạ mô hình dữ liệu: cài đặt CSDL vào DBMS
4. Thiết kế vật lý: thiết lập các cấu trúc lưu trữ, tổ chức tập tin, lập chỉ mục (index), các đường dẫn truy cập, các bộ tham số



COMPANY: Ứng dụng CSDL mẫu – Mô tả và Yêu cầu (1/3)

- CSDL **COMPANY** theo dõi
 - Nhân viên (**employees**),
 - Phòng ban (**departments**),
 - Các dự án của công ty (**projects**).
- Giả sử đã hoàn tất việc khảo sát, thu thập và phân tích yêu cầu

Chú ý: Các phân tích sau đây dùng trong ứng dụng mẫu để minh họa các trình bày trong phần lý thuyết này

- *Các mô tả chi tiết sẽ được trình bày trong tiếng Việt.*
- *Tuy nhiên, các tên thực thể, trường, loại dữ liệu, v.v. sẽ được trình bày bằng tiếng Anh, với mục đích dễ cài đặt, và tương hợp với khái niệm đang trình bày ở đây*

COMPANY: Ứng dụng CSDL mẫu – Mô tả và Yêu cầu (2/3)

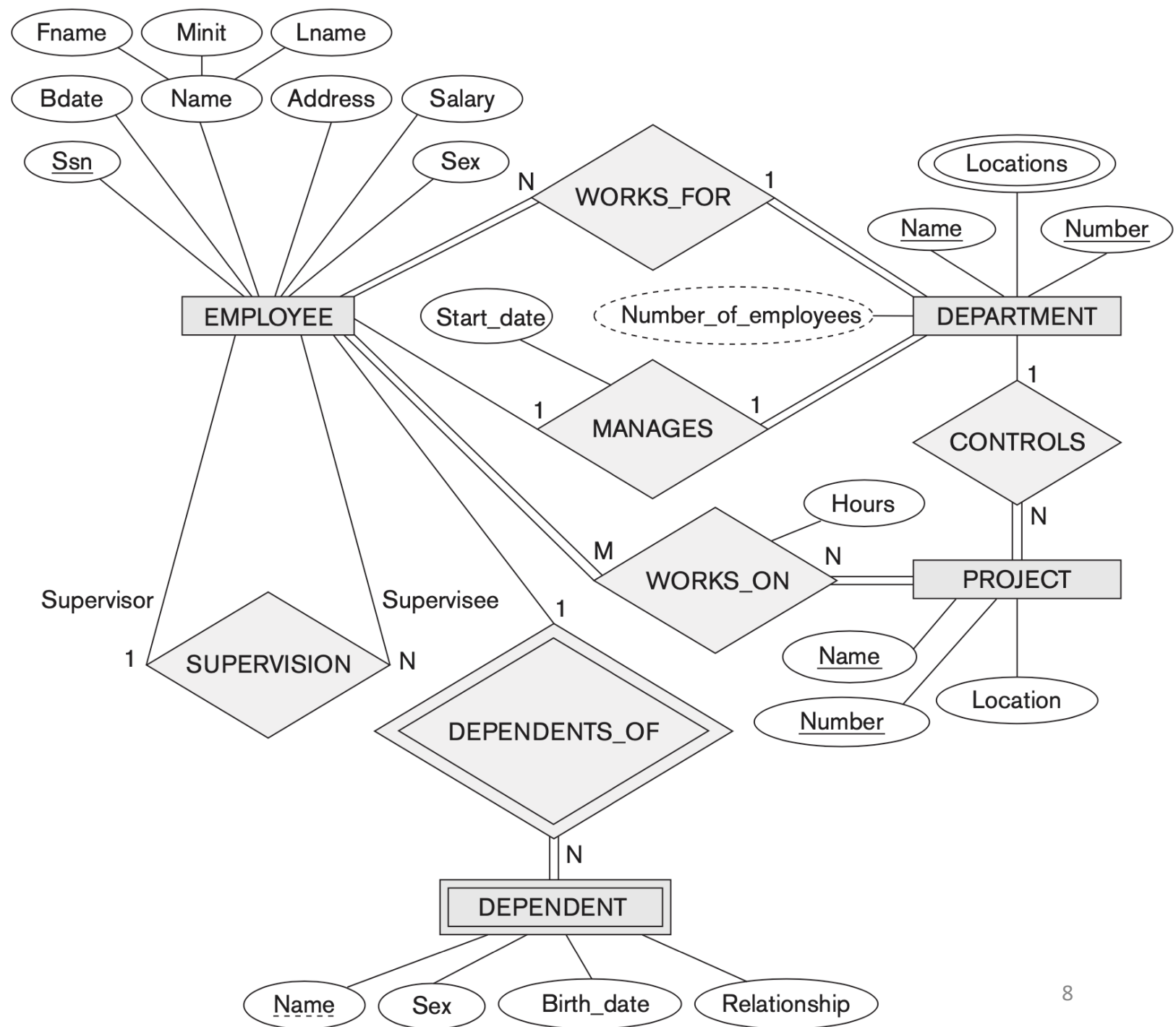
- Công ty được tổ chức thành nhiều phòng ban. Mỗi phòng ban có một tên duy nhất, một mã (số) duy nhất, và có một nhân viên chịu trách nhiệm quản lý.
 - Theo dõi ngày bắt đầu hoạt động của phòng ban, khi nhân viên quản lý bắt đầu nhận việc
 - Một phòng ban có thể có nhiều địa điểm (**location**) hoạt động
- Một phòng ban kiểm soát một số dự án nhất định. Mỗi dự án có một tên duy nhất, một mã số duy nhất, và được triển khai tại một địa điểm nhất định

COMPANY: Ứng dụng CSDL mẫu – Mô tả và Yêu cầu (3/3)

- Với mỗi nhân viên, CSDL sẽ lưu lại:
 - Tên (**name**)
 - Số an sinh xã hội (**Social Security number - SSN**). (Ở Việt Nam, ta lấy số CCCD hay số bảo hiểm)
 - Địa chỉ (**address**)
 - Lương (**salary**)
 - Giới tính (**sex – gender**)
 - Ngày sinh (**birthdate**)
- Một nhân viên được phân công (*assign*) làm ở một phòng ban, nhưng có thể tham gia nhiều dự án, và các dự án này không nhất thiết phải chịu sự kiểm soát của phòng ban nói trên.
- Phải theo dõi được số giờ làm trong tuần hiện tại của mỗi nhân viên khi tham gia một dự án, cũng như biết được người quản lý trực tiếp (**supervisor**) của nhân viên này ở trong dự án đang tham gia
- CSDL sẽ theo dõi những người phụ thuộc của mỗi nhân viên cho mục đích bảo hiểm. Các thông tin của người phụ thuộc bao gồm: họ (**first name**), giới tính, ngày sinh, và mối quan hệ (**relationship**) với nhân viên

COMPANY:

Sơ đồ Lược
đồ ER



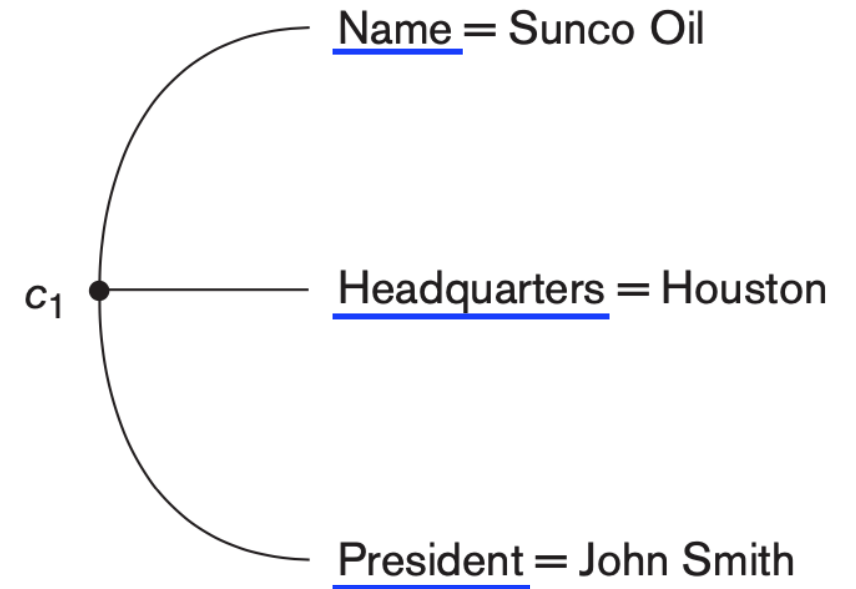
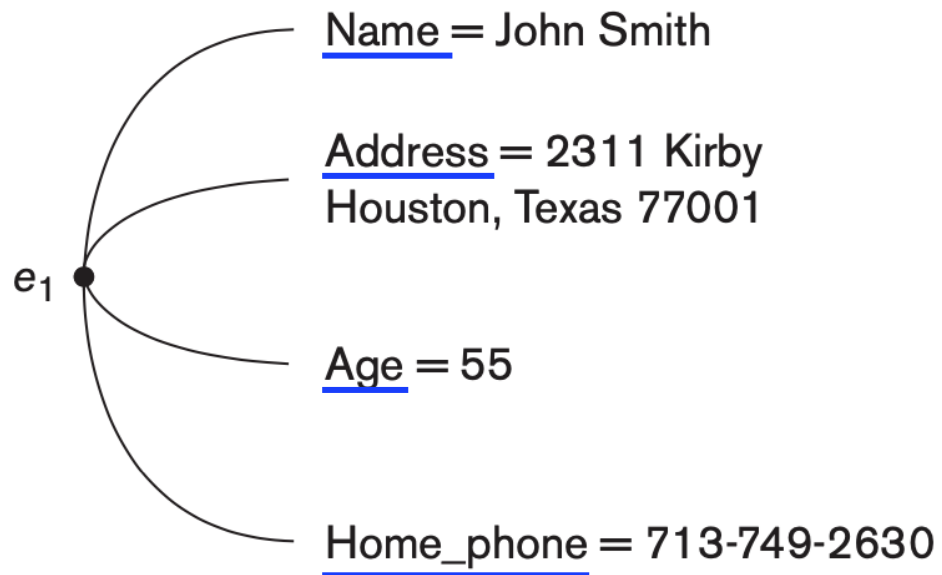
Thực thể và Thuộc tính

- Thực thể (**entity**): sự vật hoặc đối tượng trong thế giới thực, tồn tại độc lập
 - Đối tượng tồn tại dạng vật lý. Ví dụ: người, xe, nhà, nhân viên.
 - Đối tượng tồn tại dạng khái niệm. Ví dụ: công ty, công việc, khóa học.
- Mỗi thực thể có các thuộc tính (**attributes**) là các tính chất mô tả thực thể
- Mỗi thực thể cụ thể sẽ có giá trị cho từng thuộc tính của nó
- Các giá trị thuộc tính mô tả từng thực thể là phần chính được lưu trữ trong CSDL

Thực thể và Thuộc tính – Ví dụ từ COMPANY

Xét các thực thể

- **EMPLOYEE** e1
- **COMPANY** c1

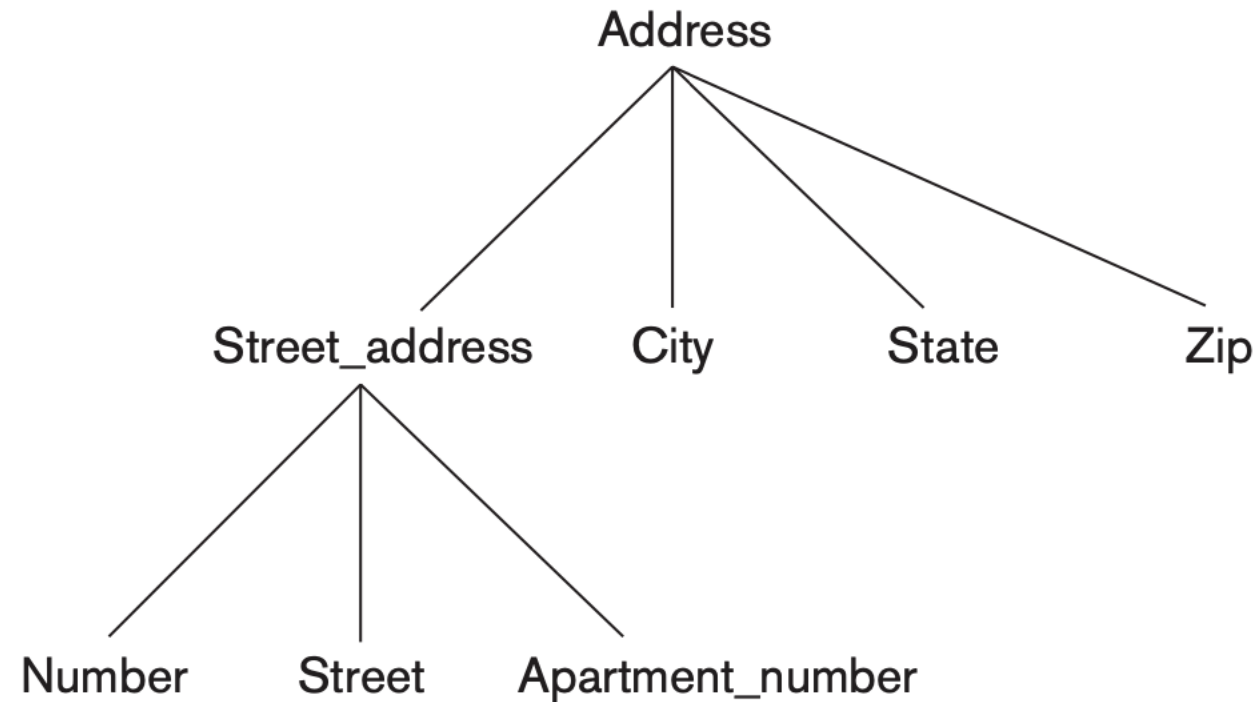


Thuộc tính: Phân loại

- Loại đơn giản
- Loại hợp thành
- Loại đơn trị
- Loại đa trị
- Loại thuộc tính được lưu trữ
- Thuộc tính được suy dẫn ra
- Giá trị **NULL**

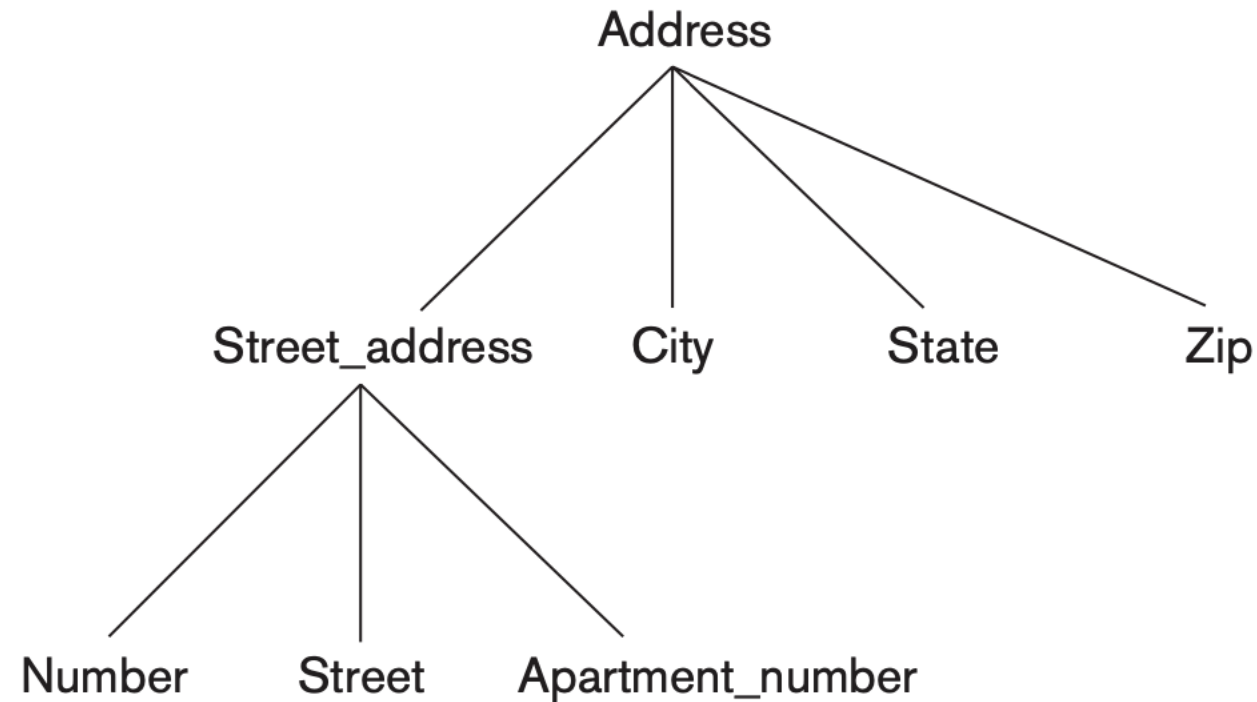
Thuộc tính Đơn giản (Nguyên tố) và Thuộc tính Hợp thành

- Thuộc tính hợp thành bao gồm nhiều thuộc tính con.
- Mỗi thuộc tính con là thuộc tính cơ sở, mang ý nghĩa độc lập
- Thuộc tính không thể chia nhỏ gọi là thuộc tính đơn giản hay thuộc tính nguyên tố
- Các thuộc tính hợp thành được sắp theo kiến trúc phân lớp
- Giá trị của một thuộc tính hợp thành là sự nối tiếp các giá trị của thuộc tính đơn giản là thành phần của nó



Thuộc tính Hợp thành: Dùng khi nào?

- Dữ liệu hợp thành và dữ liệu thành phần của nó được dùng tùy chọn theo trường hợp sử dụng khác nhau
- Thuộc tính hợp thành chỉ có ý nghĩa khi sử dụng toàn thể dữ liệu, không phân tách
→ không cần tách thành phần
- Các thuộc tính thành phần được sử dụng trong các mục đích khác, ví dụ: tìm kiếm, phân loại
→ thuộc tính hợp thành = {<thuộc tính thành phần>}



Thuộc tính Đơn trị và Thuộc tính Đa trị

Với một thực thể cụ thể,

- Thuộc tính đơn trị: Thuộc tính chỉ nhận một giá trị
 - Ví dụ: tuổi (age) của một người
- Thuộc tính đa trị: Thuộc tính nhận một bộ gồm nhiều giá trị
 - Ví dụ: bằng cấp sau phổ thông mà một người đạt được
A không có, B có 2 bằng đại học, C có một bằng đại học và một bằng cao học
- Thuộc tính đa trị cần được xác định số lượng giá trị nó có thể nhận
 - Ví dụ: số màu sơn vỏ xe hơi từ 1-2, nếu ta quy định một chiếc xe được phép sơn tối đa 2 màu vỏ

Thuộc tính được Lưu trữ và Thuộc tính Suy dẫn

- Một số thuộc tính có mối quan hệ.
- Thuộc tính suy dẫn được suy ra từ (các) thuộc tính được lưu trữ

Ví dụ:

- Ngày sinh (**Birth_date**) và tuổi (**Age**):
$$\text{Age} = \text{current_time} - \text{Birth_date}$$
- Số nhân viên
$$\text{DEPARTMENT.Number_of_employees} = \text{count}(\text{employee})$$

Giá trị NULL

- Trong một số trường hợp, một thực thể cụ thể có thể không có giá trị áp dụng được cho một thuộc tính
→ = NULL
Ví dụ: **Apartment_number** = **NULL**
- Không biết được giá trị một thuộc tính
→ = NULL
 - Không nhớ giá trị cụ thể của thuộc tính.
Ví dụ: chiều cao của một người **Height** = **NULL**
 - Không chắc thuộc tính có tồn tại giá trị.
Ví dụ: Điện thoại ở nhà của một người **Home_phone** = **NULL**

Thuộc tính Phức hợp

- Dùng () để gom các thành phần của thuộc tính hợp thành
- Dùng {} để trình bày các thuộc tính đa trị

- Ví dụ

```
{Address_phone( {Phone(Area_code,Phone_number)},Address(Street_address  
(Number,Street,Apartment_number),City,State,Zip) )}
```

Kiểu Thực thể, Tập Thực thể

- CSDL chứa tập hợp các thực thể tương tự nhau.
- Các thực thể chia sẻ cùng tập các thuộc tính, nhưng mỗi thực thể cụ thể có giá trị riêng cho từng thuộc tính
- Kiểu thực thể (entity type) định nghĩa tập hợp (collection/set) các thực thể có chung các thuộc tính.
 - Mỗi kiểu thực thể được mô tả bằng **tên** và **các thuộc tính** của nó.
- Tập thực thể: Tập hợp tất cả các thực thể của một kiểu thực thể tại một thời điểm

Entity Type Name:

EMPLOYEE

Name, Age, Salary

COMPANY

Name, Headquarters, President

Entity Set:
(Extension)

e_1 •

(John Smith, 55, 80k)

e_2 •

(Fred Brown, 40, 30K)

e_3 •

(Judy Clark, 25, 20K)

⋮

c_1 •

(Sunco Oil, Houston, John Smith)

c_2 •

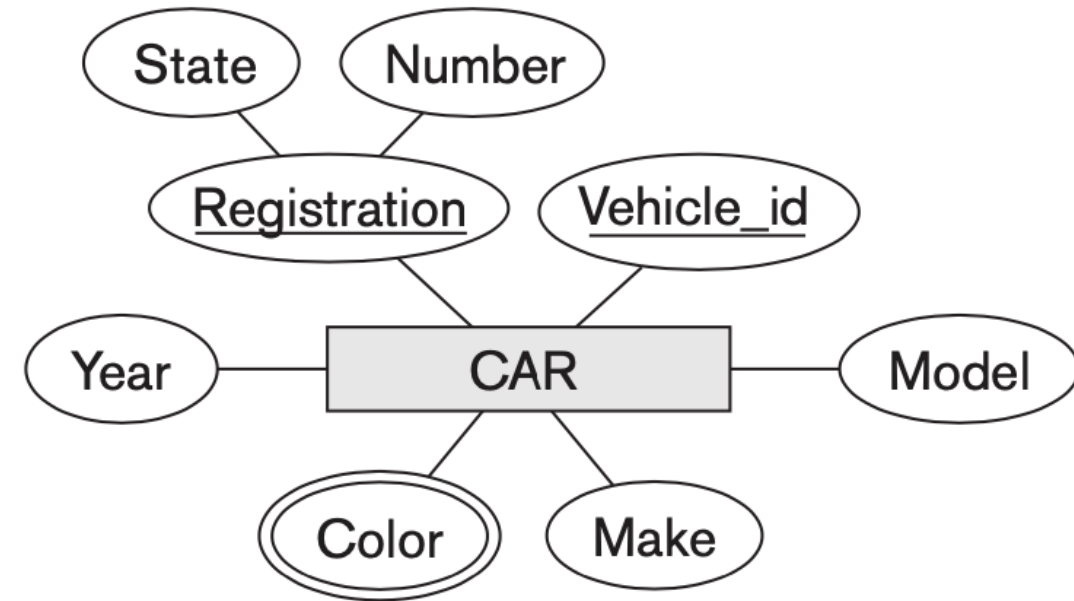
(Fast Computer, Dallas, Bob King)

⋮

Kiểu Thực thể, Tập Thực thể

Ký hiệu trong sơ đồ ER:

- Kiểu thực thể
- thuộc tính thuộc tính hợp thành
- thuộc tính đa trị
- Gắn kết thuộc tính – thực thể ———
- Gạch dưới đánh dấu thuộc tính “Khóa”



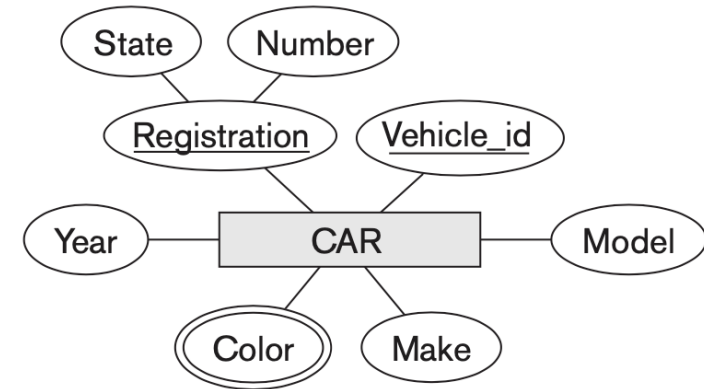
Thuộc tính Khóa

- Ràng buộc khóa hay ràng buộc duy nhất đối với các thuộc tính
 - (Bộ) Thuộc tính khóa (key): Gồm một hoặc nhiều thuộc tính có giá trị phân biệt giữa các thực thể riêng biệt trong tập thực thể
 - Giá trị thuộc tính khóa xác định duy nhất từng thực thể
- Ví dụ:
- Thuộc tính Tên (Name) là thuộc tính khóa (gọi tắt là khóa) của kiểu COMPANY
 - Thuộc tính số an sinh xã hội (Snn) là khóa của kiểu thực thể EMPLOYEE

Thuộc tính Khóa – Một số chú ý

- Một thực thể có thể có nhiều hơn một thuộc tính khóa
- Thuộc tính hợp thành là khóa nhưng từng thuộc tính thành phần lại không phải là khóa
- Một kiểu thực thể có thể không có khóa
→ kiểu thực thể yếu
- Trong sơ đồ, nếu hai thuộc tính được gạch chân riêng biệt, thì mỗi thuộc tính là một khóa riêng
Ví dụ: **Registration** và **Vehicle_id**

(a)



(b)

CAR
Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

CAR₁
((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR₂
((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR₃
((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

⋮

Tập Giá trị (miền) của các Thuộc tính

- Mỗi thuộc tính A của kiểu thực thể E được liên kết với một tập giá trị V (còn gọi là miền giá trị)
- Mỗi giá trị $A(e)$ được gán vào thuộc tính A dành cho một thực thể e
 - Ví dụ: tập giá trị tuổi của người đi làm **EMPLOYEE** là các giá trị số nguyên trong đoạn giá trị [16, 70]
 - Tập giá trị của **Name** được quy định bởi tập các chuỗi ký tự alphabet, ngăn cách nhau bằng các ký tự trống (" ")
 - NULL
- Các giá trị thuộc kiểu dữ liệu cơ sở (integer, string, Boolean, float, kiểu liệt kê – enumerate, miền con, v.v.) và các kiểu định nghĩa trong CSDL (date, time, v.v.)
- Không được trình bày trong các sơ đồ ER

Tập Giá trị (miền) của các Thuộc tính – Định nghĩa toán học

Định nghĩa: Mỗi thuộc tính A của kiểu thực thể E là hàm số:

$$A : E \rightarrow P(V)$$

trong đó,

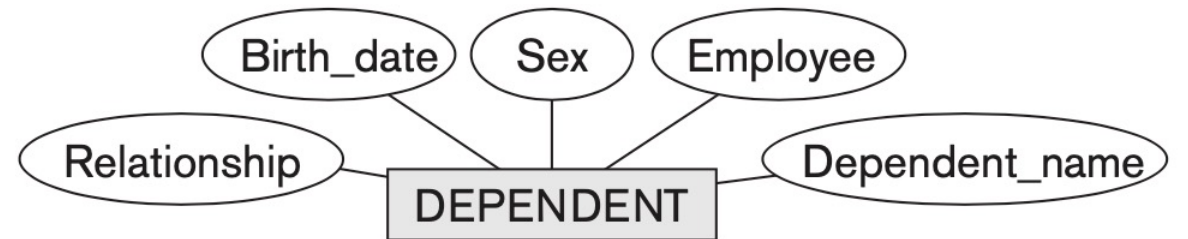
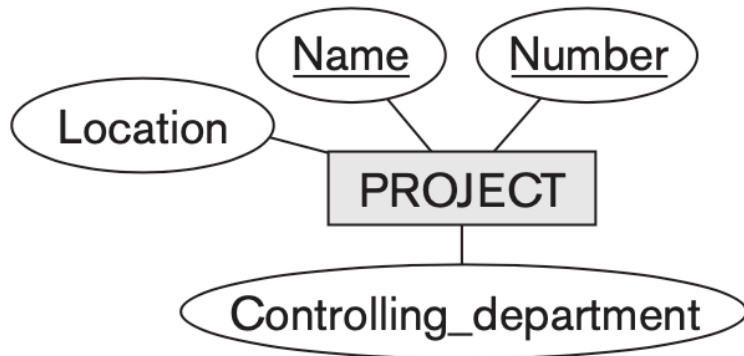
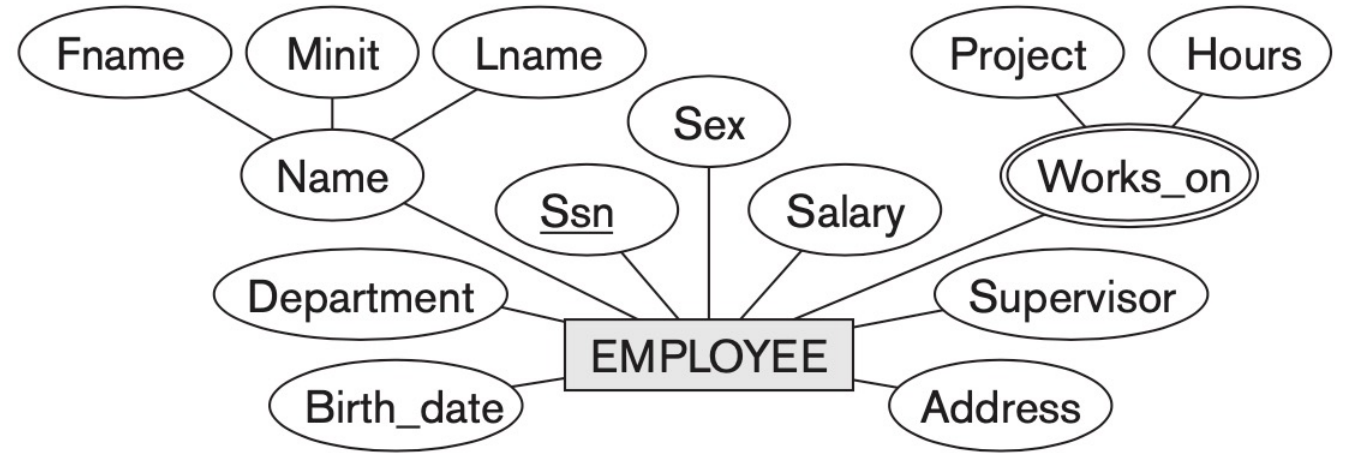
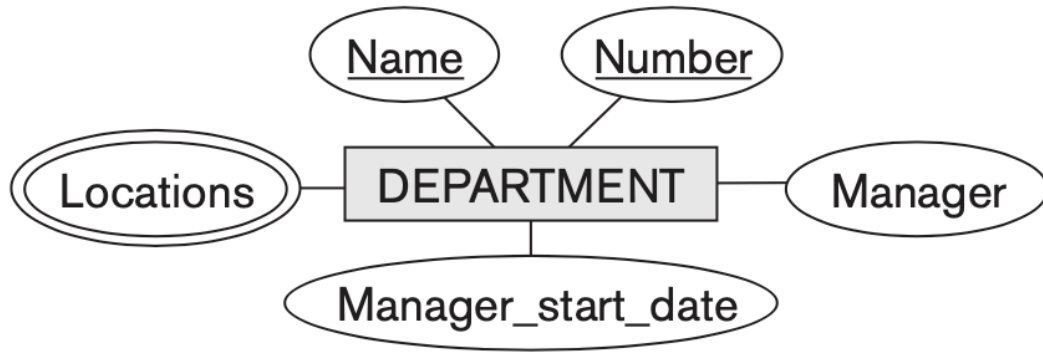
- V : tập giá trị của A
- $P(V)$: tập lũy thừa của V , là tập hợp gồm các tập con của V .
- Áp dụng cho các thuộc tính đơn trị, đa trị và thuộc tính được gán **NULL**
- Gọi giá trị của thuộc tính A cho một thực thể e là $A(e)$. Thế thì,
 - Giá trị NULL được biểu diễn bằng tập rỗng
 - Với thuộc tính đơn trị, $A(e)$ giới hạn là một tập có duy nhất một phần tử (một giá trị) cho mỗi e trong E .
 - Với thuộc tính đa trị, $A(e)$ là tập giá trị không có giới hạn
- Đối với một thuộc tính hợp thành A , tập giá trị V là tập lũy thừa của tích Descartes của $P(V_1), P(V_2), \dots, P(V_n)$, trong đó V_1, V_2, \dots, V_n là các tập giá trị của các thuộc tính thành phần đơn giản tạo thành A :
$$V = P(P(V_1) \times P(V_2) \times \dots \times P(V_n))$$
- Tập các giá trị cho biết tất cả các giá trị có thể sử dụng. Tuy nhiên, ở một thời điểm chỉ một phần nhỏ các giá trị này được dùng trong CSDL, phụ thuộc vào sự tồn tại của dữ liệu.

COMPANY: Thiết kế Khái niệm Ban đầu

Có 04 kiểu thực thể:

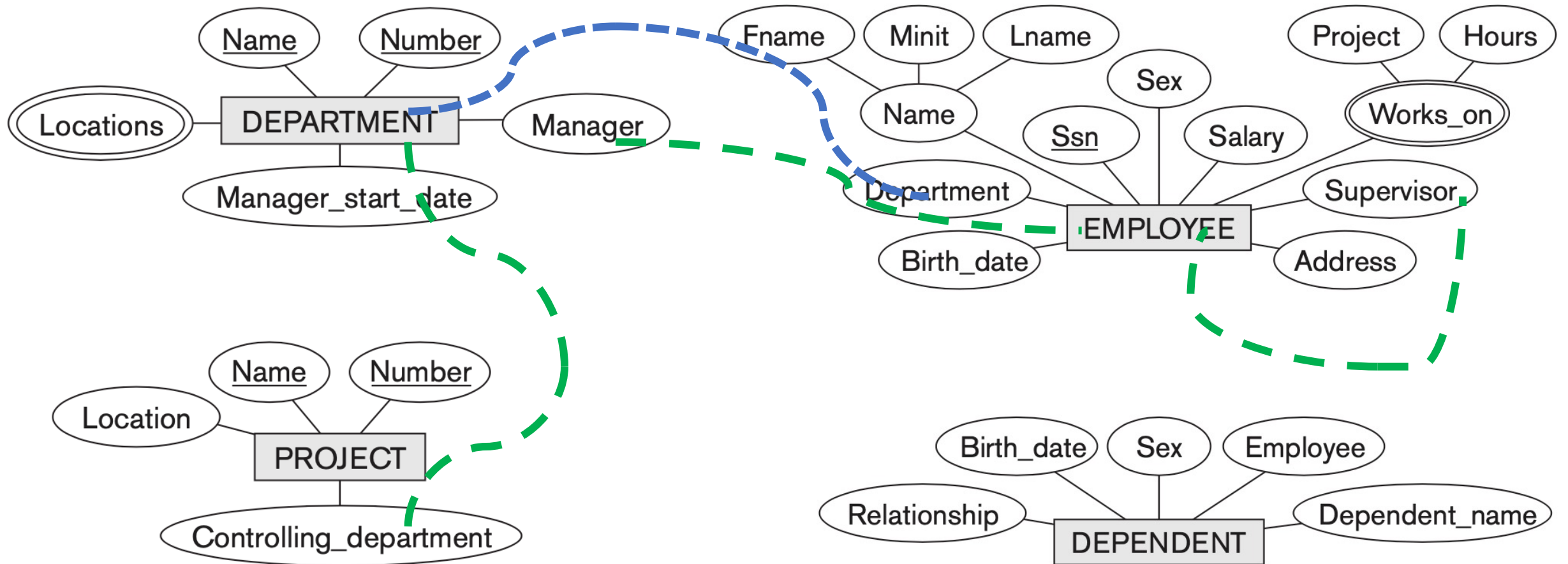
1. **DEPARTMENT**(Name, Number, Locations, Manager, Manager_start_date)
2. **PROJECT**(Name, Number, Location, Controlling_department)
3. **EMPLOYEE**(Name, Ssn, Sex, Address, Salary, Birth_date, Department, Supervisor)
 - Name = (First_name, Middle_name, Last_name)
 - Address: thuộc tính đơn giản, là văn bản (text) mô tả địa chỉ
 - + Có thể xem xét bao gồm các thành phần số nhà (Number), đường (Street), phường (Ward), thành phố/tỉnh (City/Province)
4. **DEPENDENT**(Employee, Dependent_name, Sex, Birth_date, Relationship)
 - Relationship: mô tả mối quan hệ giữa người phụ thuộc với nhân viên

COMPANY: Thiết kế Khái niệm Ban đầu



Quan hệ và Ràng buộc Cấu trúc

COMPANY: Các quan hệ (relationships)



Các kiểu Quan hệ, Tập và các Thể hiện

- Kiểu quan hệ R giữa n kiểu thực thể E_1, E_2, \dots, E_n định nghĩa một tập các mối liên kết – tập quan hệ - giữa các thực thể thuộc kiểu thực thể này.
 - R : ký hiệu dùng chung cho khái niệm kiểu quan hệ và tập quan hệ thuộc kiểu
- Tập quan hệ R là tập các thể hiện quan hệ r_i , trong đó r_i liên kết n thực thể riêng biệt (e_1, e_2, \dots, e_n) , mỗi thực thể e_j trong r_i là một thành viên trong tập thực thể E_j , $1 \leq j \leq n$.
- Một tập quan hệ là liên hệ về mặt toán học trên E_1, E_2, \dots, E_n , được định nghĩa là tập con của tích Decartes của các tập thực thể
$$E_1 \times E_2 \times \dots \times E_n$$
- Theo đó, các kiểu thực thể E_i được mô tả là tham gia vào mỗi quan hệ R ; Các thực thể (e_1, e_2, \dots, e_n) tham gia vào thể hiện quan hệ $r_i = (e_1, e_2, \dots, e_n)$

Quan hệ WORKS_FOR

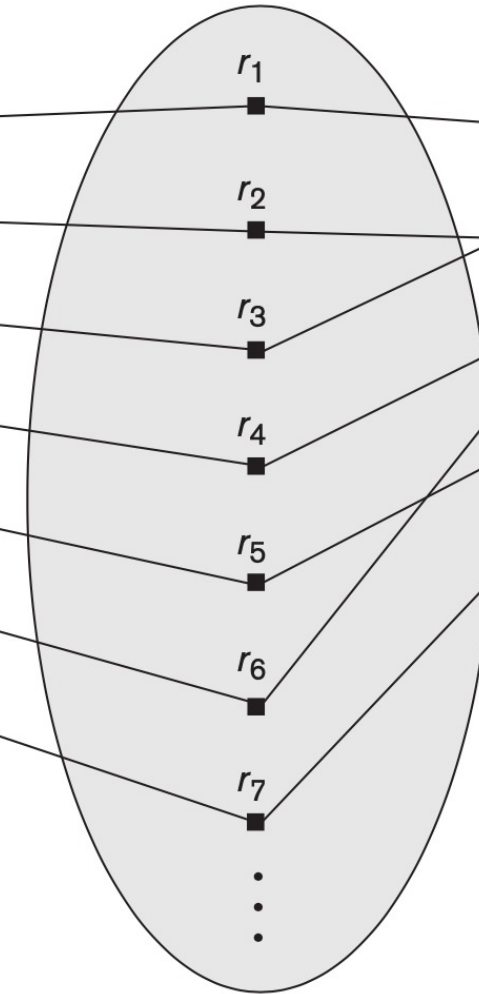
Trong sơ đồ ER, kiểu quan hệ được thể hiện bằng các hộp hình thoi.

Đường thẳng nối tới hộp chữ nhật thể hiện các kiểu thực thể tham gia vào quan hệ này.

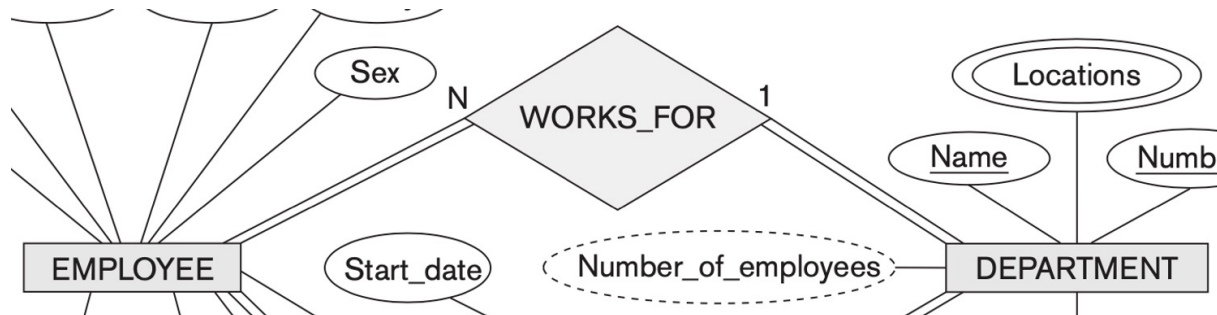
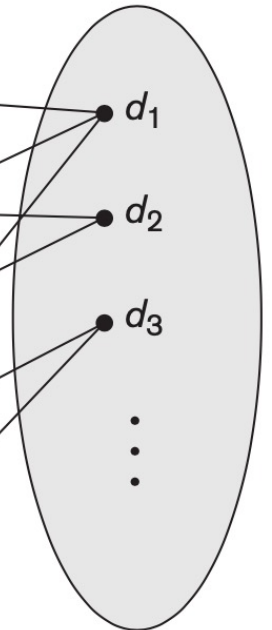
EMPLOYEE



WORKS_FOR

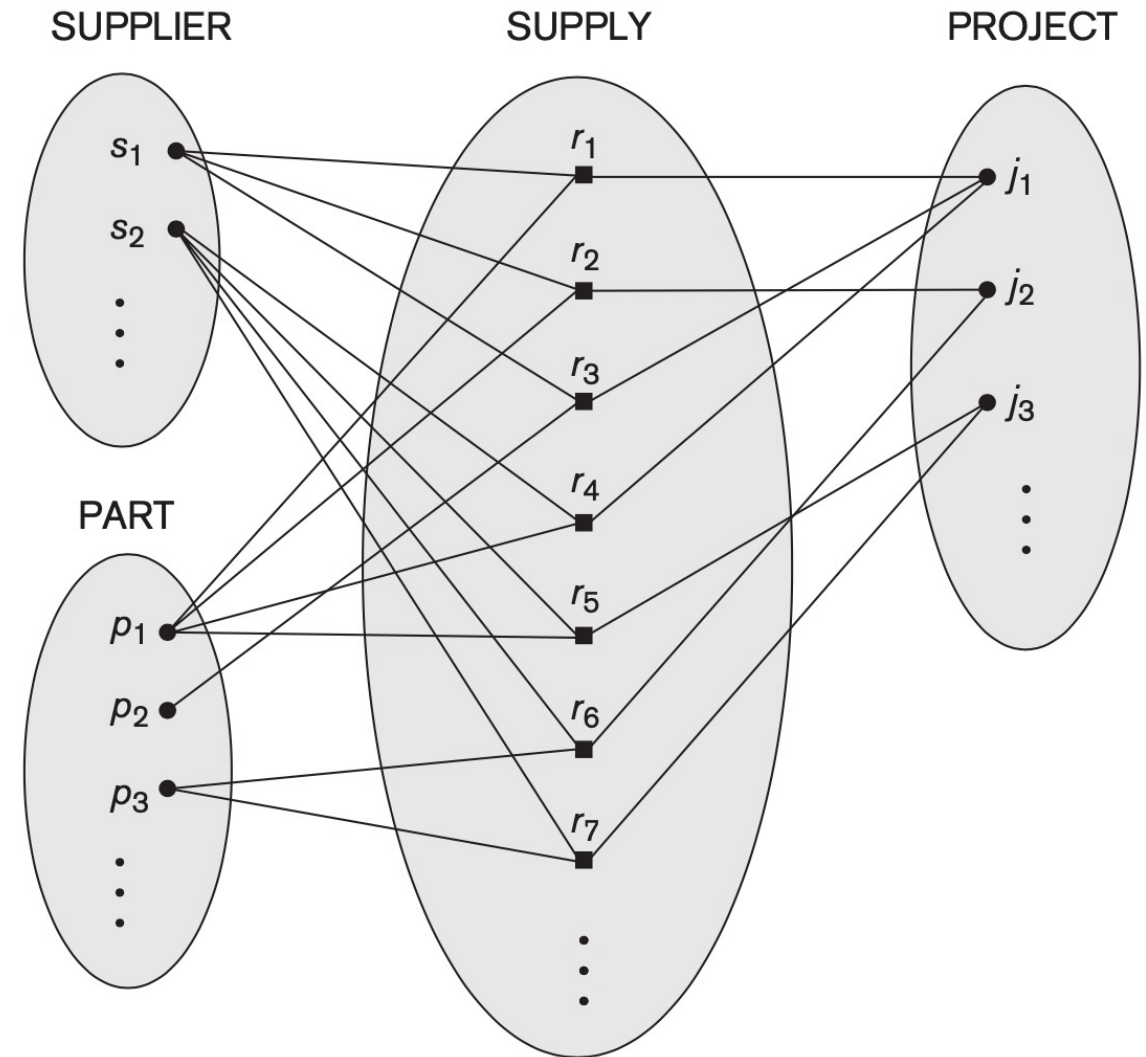


DEPARTMENT



Bậc Quan hệ

- Bậc (cấp độ) quan hệ: số các thực thể tham gia mỗi quan hệ
- Quan hệ bậc hai: quan hệ nhị phân (binary)
Ví dụ: WORKS_FOR
- Quan hệ bậc ba: quan hệ tam phân (ternary)
Ví dụ: SUPPLY
- Quan hệ được xem như là thuộc tính của thực thể



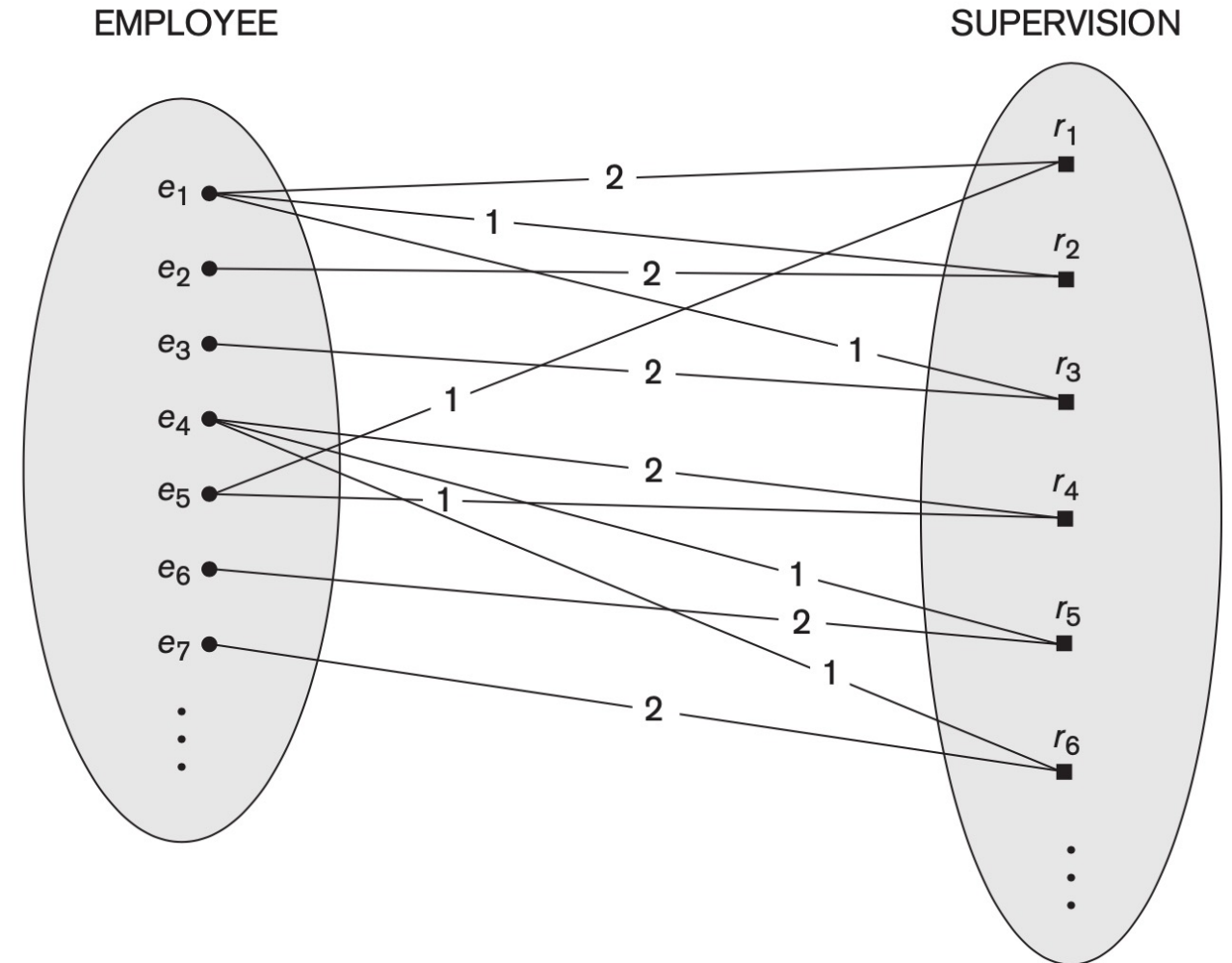
Vai trò (Role)

- Mỗi kiểu thực thể tham gia trong kiểu quan hệ có vai trò (role) nhất định trong mối quan hệ
- Đặt tên vai trò biểu thị vai trò mà thực thể tham gia, và giải thích ý nghĩa mối quan hệ
Ví dụ: Trong WORKS_FOR, EMPLOYEE đóng vai trò là người làm (employee, worker) và DEPARTMENT đóng vai trò là đơn vị chủ quản (department, employer)

Quan hệ Đế quy

- Một kiểu thực thể tham gia trong mỗi quan hệ với vai trò khác nhau
→ quan hệ đệ quy (recursive relationship) hoặc quan hệ tự tham chiếu (self-referencing relationship)

Ví dụ: Mỗi quan hệ đệ quy SUPERVISION giữa EMPLOYEE trong vai trò (1) giám sát (supervisor role) và EMPLOYEE trong vai trò (2) cấp dưới (subordinate role)



Các Ràng buộc trong kiểu Quan hệ Nhị phân

- Các ràng buộc áp đặt giới hạn lên các kết hợp khả dĩ giữa các thực thể tham gia mỗi quan hệ tương ứng.
- Ràng buộc được lấy từ tình huống thực tế mà mỗi quan hệ đang biểu diễn
- Hai loại:
 - Ràng buộc tỷ lệ số lượng (Cardinality Ratio)
 - Ràng buộc tham gia và sự tồn tại (Participation & Dependency)

Quan hệ Nhị phân: Ràng buộc Tỷ lệ Số lượng

- Xác định số lượng tối đa các thể hiện (instance) của quan hệ mà một thực thể có thể tham gia.

Ví dụ: trong **WORKS_FOR**, tỷ lệ **DEPARTMENT:EMPLOYEE** là **1:N**

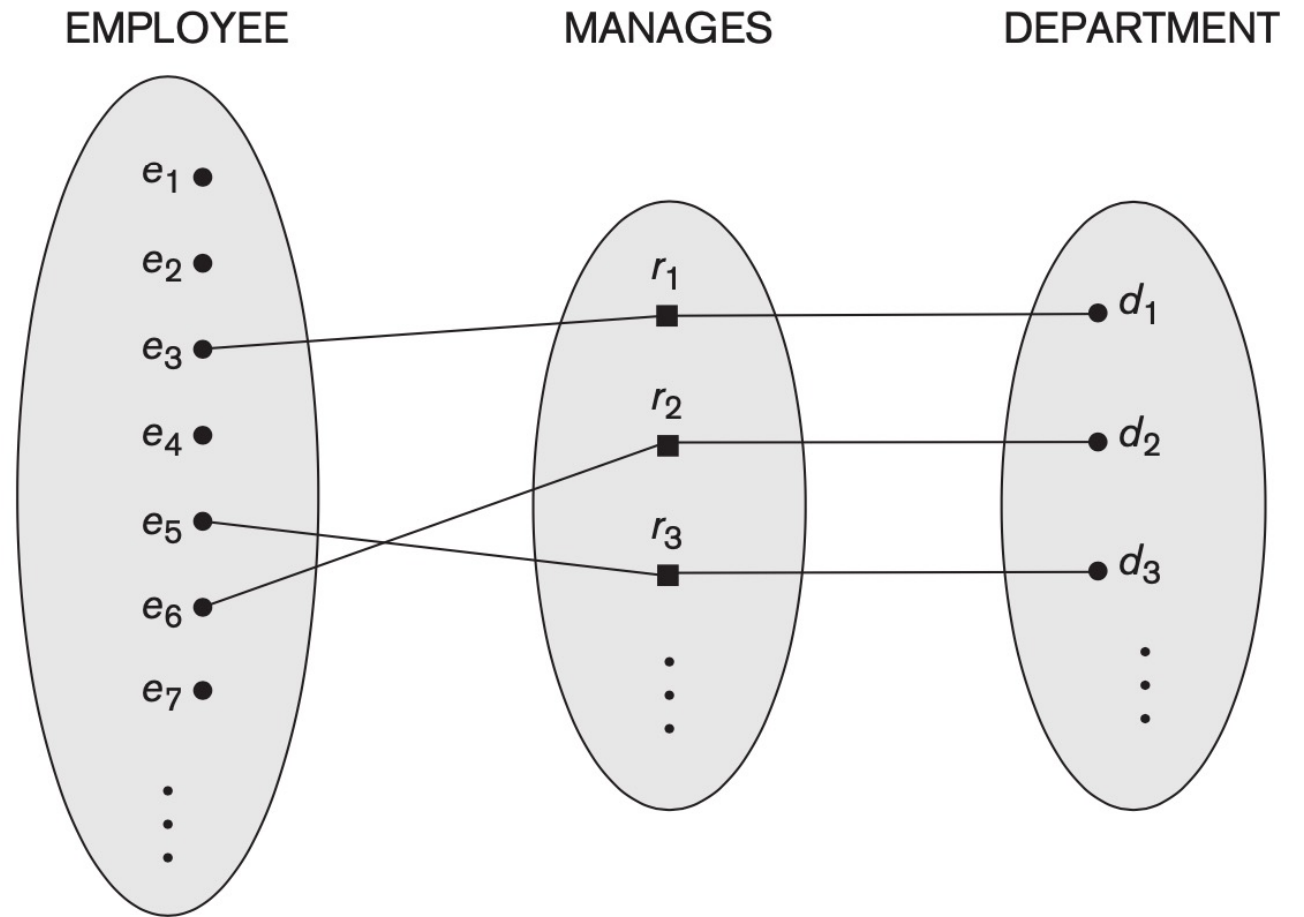
được hiểu: mỗi phòng ban có thể liên quan tới (~ tuyển dụng) số nhân viên bất kỳ (tối thiểu 0, tối đa **N**),

ngược lại, mỗi nhân viên chỉ có thể liên quan tới (~ làm việc cho) tối đa 1 phòng ban (**1**)

- Các tỷ lệ có thể tồn tại là:
 - **1:1** : one – one, một – một
 - **1:N** : one – many, một – nhiều
 - **N:1** : many – one, nhiều – một
 - **M:N** : many – many, nhiều – nhiều
- Biểu diễn: N hoặc *

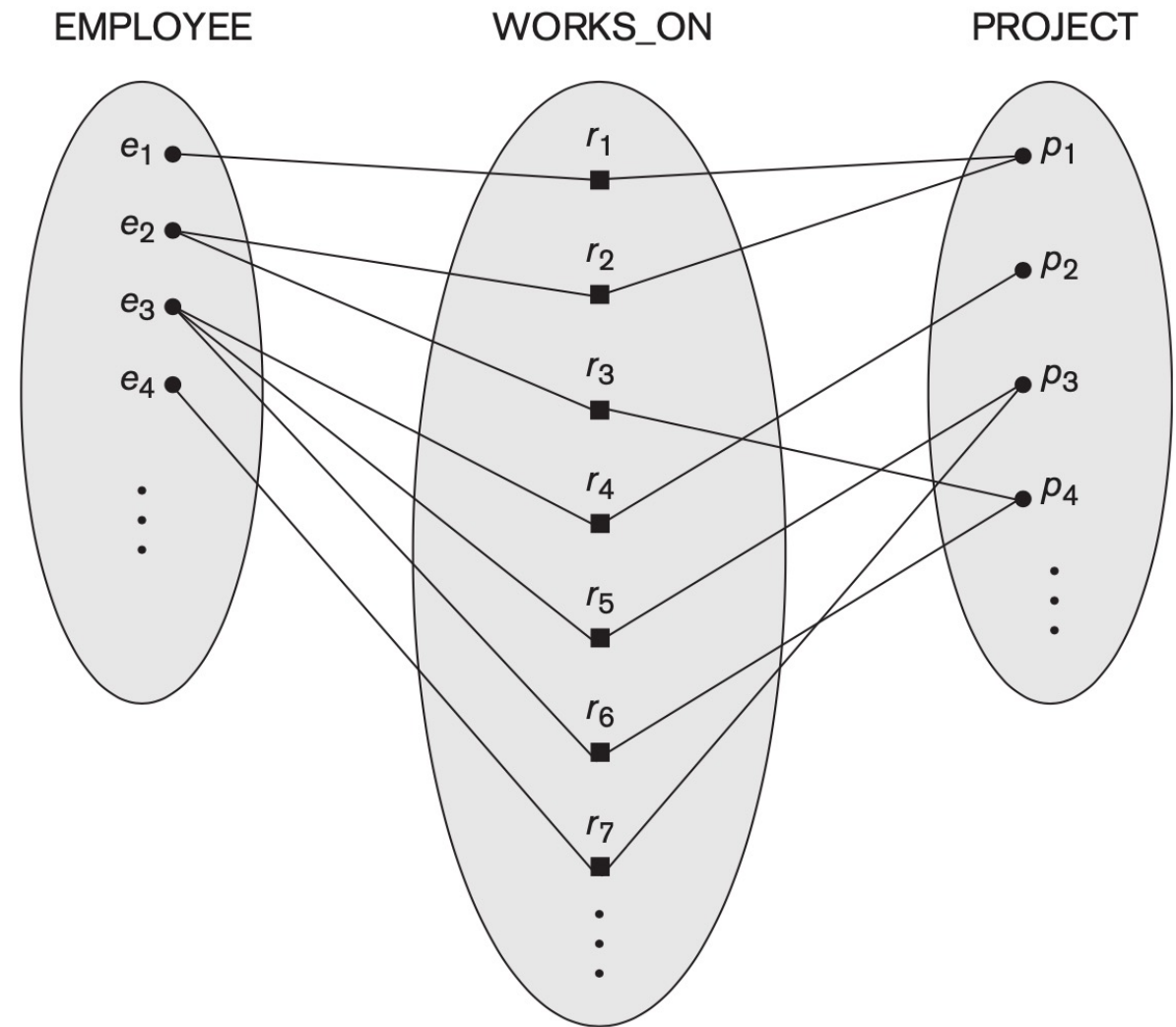
Quan hệ Nhị phân: Ràng buộc Tỷ lệ Số lượng 1:1

- Quan hệ **1:1**
- Ví dụ: MANAGES



Quan hệ Nhị phân: Ràng buộc Tỷ lệ Số lượng **M:N**

- Quan hệ **M:N**
- Ví dụ: MANAGES



Ràng buộc Tham gia và Phụ thuộc Tồn tại

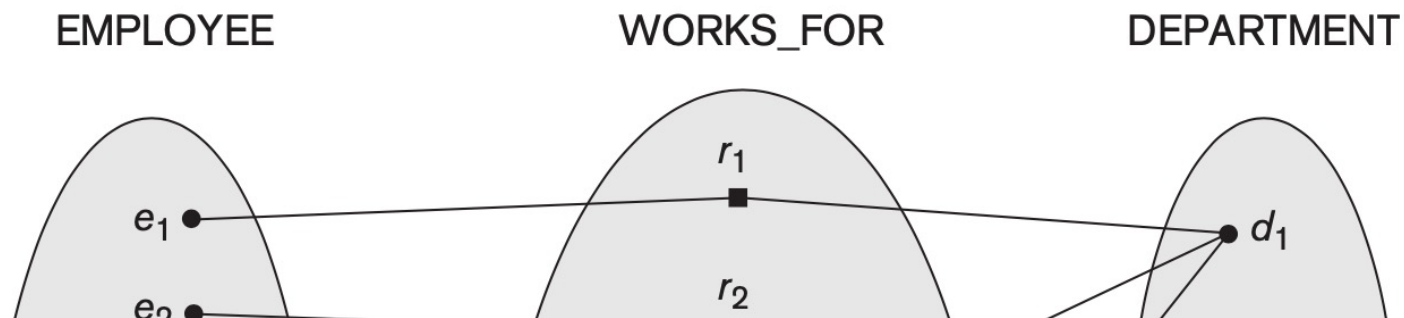
- Ràng buộc tham gia xác định liệu sự tồn tại của một thực thể có phụ thuộc vào thực thể khác có liên quan trong mối quan hệ hay không.
- Xác định số thể hiện quan hệ tối thiểu mà mỗi thực thể có thể tham gia, còn gọi là ràng buộc số lượng tối thiểu
- Hai loại: ràng buộc toàn phần và ràng buộc từng phần
 - Kiểu ràng buộc tham gia được quy định cụ thể căn cứ mô tả thực thể

Ràng buộc Tham gia và Phụ thuộc Tồn tại

Ví dụ: Chính sách công ty quy định “*mọi nhân viên phải làm việc cho một phòng ban*”

→ thực thể nhân viên chỉ có thể tồn tại nếu thực thể nhân viên này tham gia vào ít nhất một thể hiện quan hệ WORKS_FOR

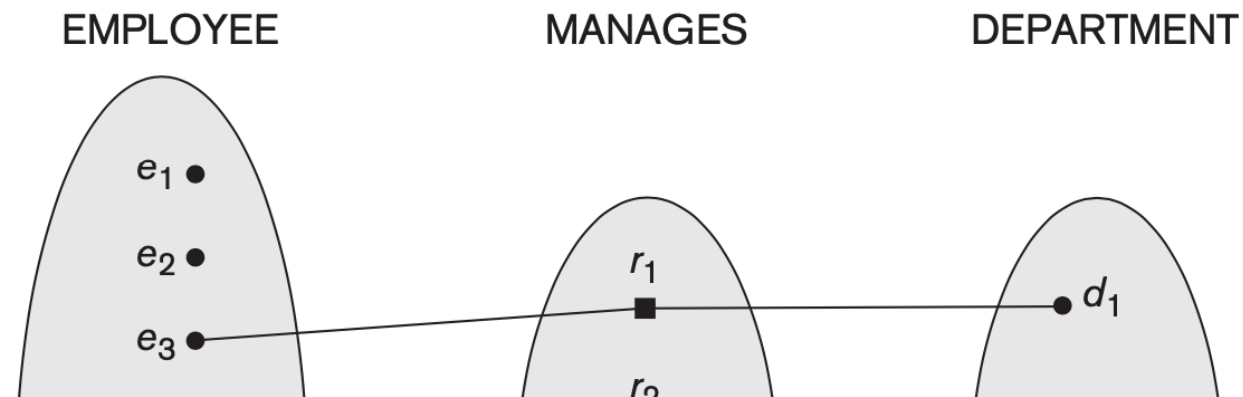
- **Phụ thuộc toàn phần**, còn gọi tham gia tổng thể: mọi thực thể đều tham gia vào mối quan hệ
- Ký hiệu trong ER: đường thẳng đôi liền nét



Ràng buộc Tham gia và Phụ thuộc Tồn tại

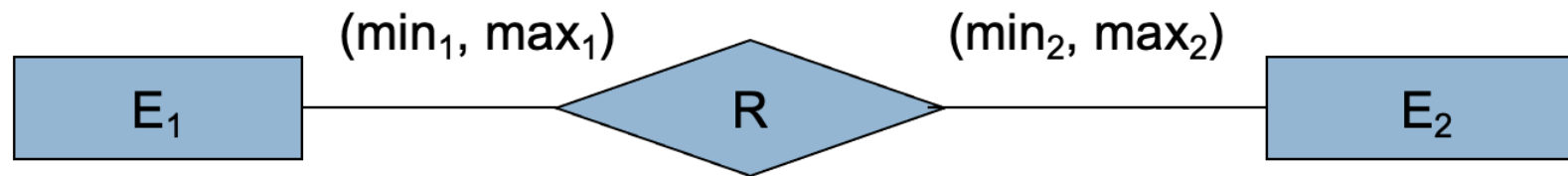
- **Phụ thuộc từng phần:** chỉ một vài phần của tập thực thể có tham gia vào mối quan hệ
- Ký hiệu trong ER: Đường thẳng đơn liền nét

Ví dụ: quan hệ MANAGES



Ký hiệu (min, max) cho các ràng buộc cấu trúc trên mỗi quan hệ

- Chỉ định mỗi thực thể $e \in E$ tham gia ít nhất min và nhiều nhất max thể hiện quan hệ trong R



- Ý nghĩa:
Mỗi thực thể $e_1 \in E_1$ phải tham gia trong tập mỗi quan hệ R ít nhất \min_1 và nhiều nhất là \max_1 lần (tương tự cho $e_2 \in E_2$)

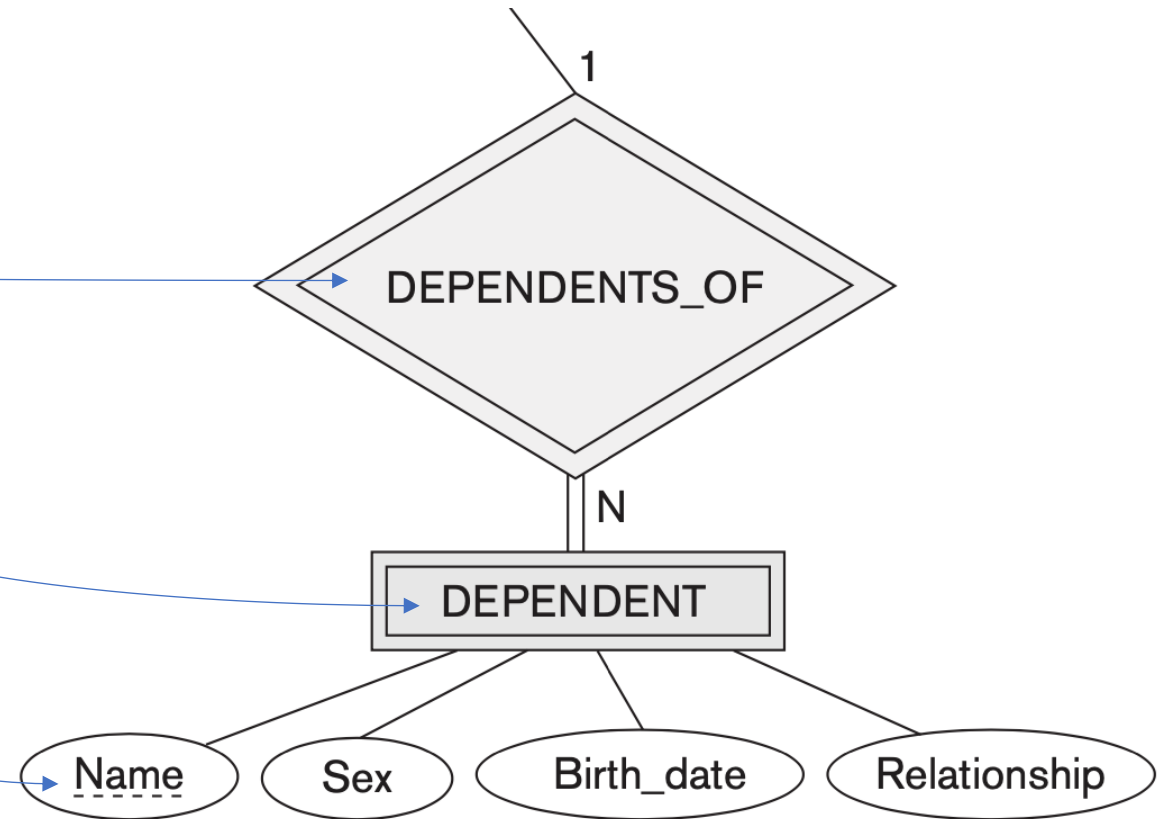
Loại Thực thể Yếu (Weak Entity Type)

- Loại thực thể **không** có thuộc tính khóa
- Phải tham gia trong một loại mối quan hệ xác định (identifying relationship type) trong đó có một loại thực thể chủ hay loại thực thể xác định (owner entity type)
- Các thực thể được xác định bằng sự kết hợp của:
 - Khóa riêng phần (partial key) của loại thực thể yếu
 - Loại thực thể mà chúng có quan hệ với trong loại thực thể xác định
- Loại thực thể yếu luôn có ràng buộc tham gia toàn phần (phụ thuộc tồn tại) trong mỗi kết hợp xác định tương ứng, bởi vì thực thể yếu không thể xác định mà không có thực thể chủ tương ứng

Loại Thực thể Yếu (Weak Entity Type)

Biểu diễn trong sơ đồ ER

- Quan hệ xác định:
- Kiểu thực thể yếu:
- Thuộc tính khóa riêng phần:



Thiết kế Khái niệm ER

- Có thể định nghĩa thực thể và mối quan hệ theo các cách khác nhau với cùng một mô hình
- Thiết kế lược đồ là một việc lặp đi lặp lại và có thể cần có sự tinh chế
 - Các thuộc tính của loại mối kết hợp 1:1 có thể được ghép vào một trong các thực thể tham gia
 - Các thuộc tính của loại mối kết hợp 1:N (or N:1) có thể được ghép vào loại thực thể của nhánh N
 - Các thuộc tính của loại mối kết hợp N:M không thể được ghép chung


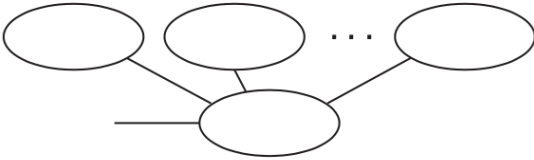
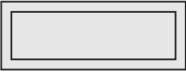

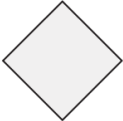
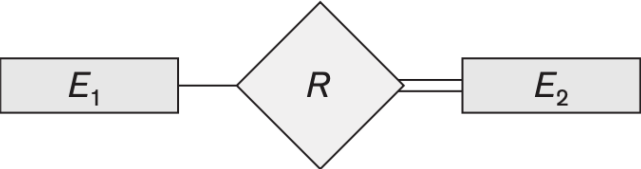

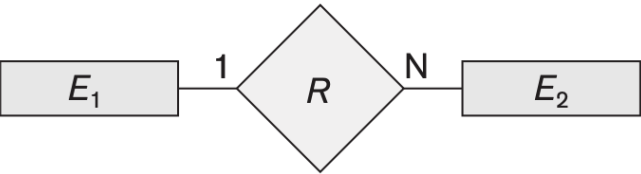

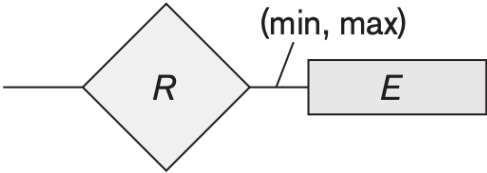


Các bước thiết kế ER

1. Xác định các loại thực thể (loại thực thể hay thuộc tính)
2. Xác định loại mối kết hợp
3. Xác định và gán thuộc tính với loại thực thể và loại mối kết hợp
4. Quyết định miền giá trị của thuộc tính
5. Quyết định các thuộc tính khóa cho loại thực thể
6. Gán (tinh chế) cardinality ratio(s) vào loại mối kết hợp
7. Thiết kế phân cấp chuyên biệt hóa/tổng quát hóa (specialization/generalization) trong các ràng buộc

Chuyển đổi Lược đồ ER thành các bảng dữ liệu

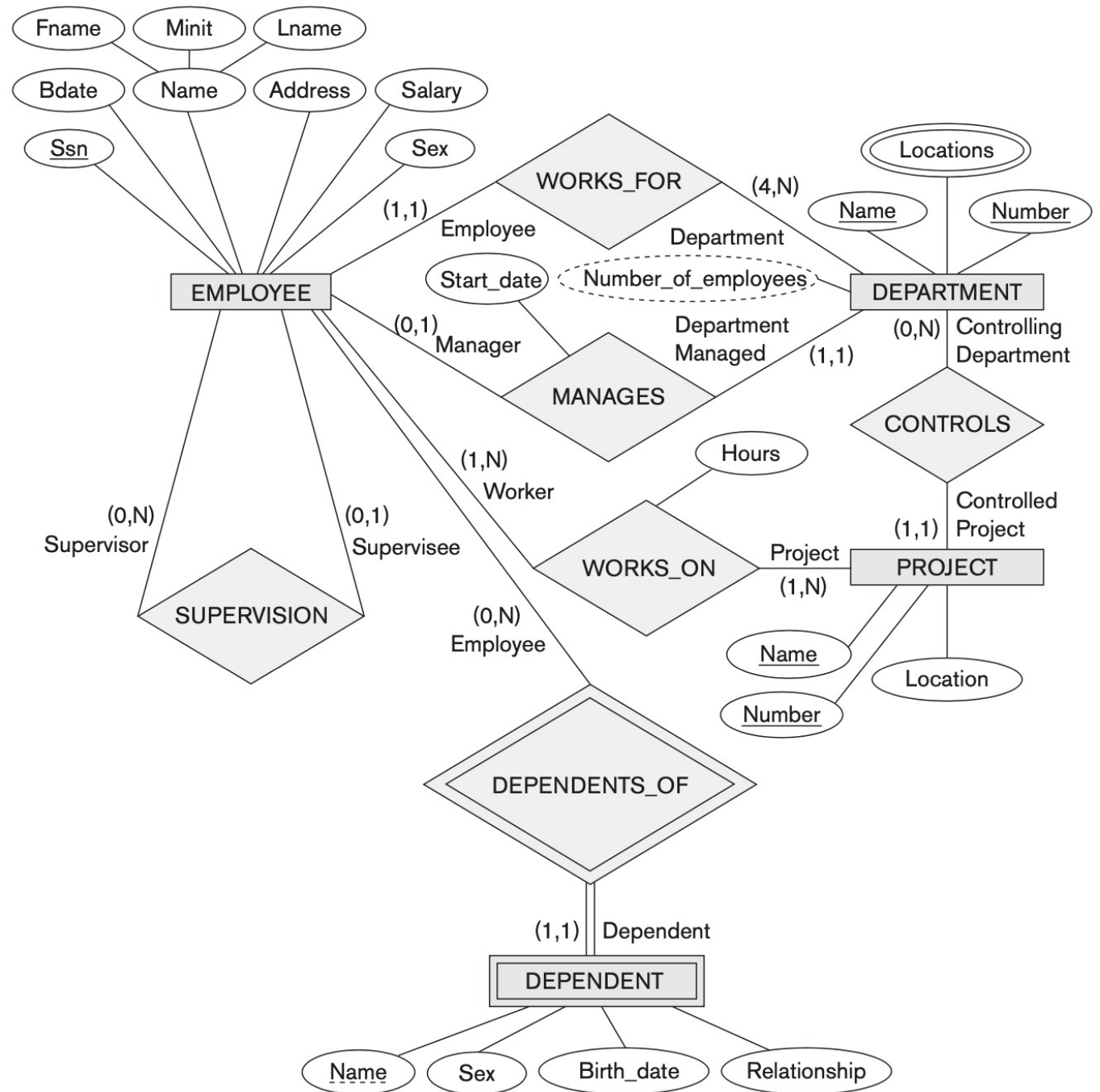
- Một lược đồ ER có thể được thể hiện dưới dạng tập hợp các bảng thể hiện nội dung của CSDL (instance)
- Các khóa chính cho phép các loại thực thể và loại mối kết hợp được diễn đạt dưới dạng bảng
- Đối với mỗi loại thực thể và loại mối kết hợp, một bảng duy nhất có thể được phát sinh được gán cho một tên tương ứng với tên của loại thực thể hay loại mối kết hợp
- Mỗi bảng có một số cột tương ứng với các thuộc tính (atomic) và có một tên duy nhất. Một thuộc tính của một bảng có thể có cùng miền giá trị với thuộc tính trong lược đồ ER
- Việc chuyển đổi một lược đồ ER thành một tập hợp các bảng là cơ bản để phát sinh một lược đồ quan hệ từ một lược đồ ER

Các Ký hiệu trong Mô hình ER

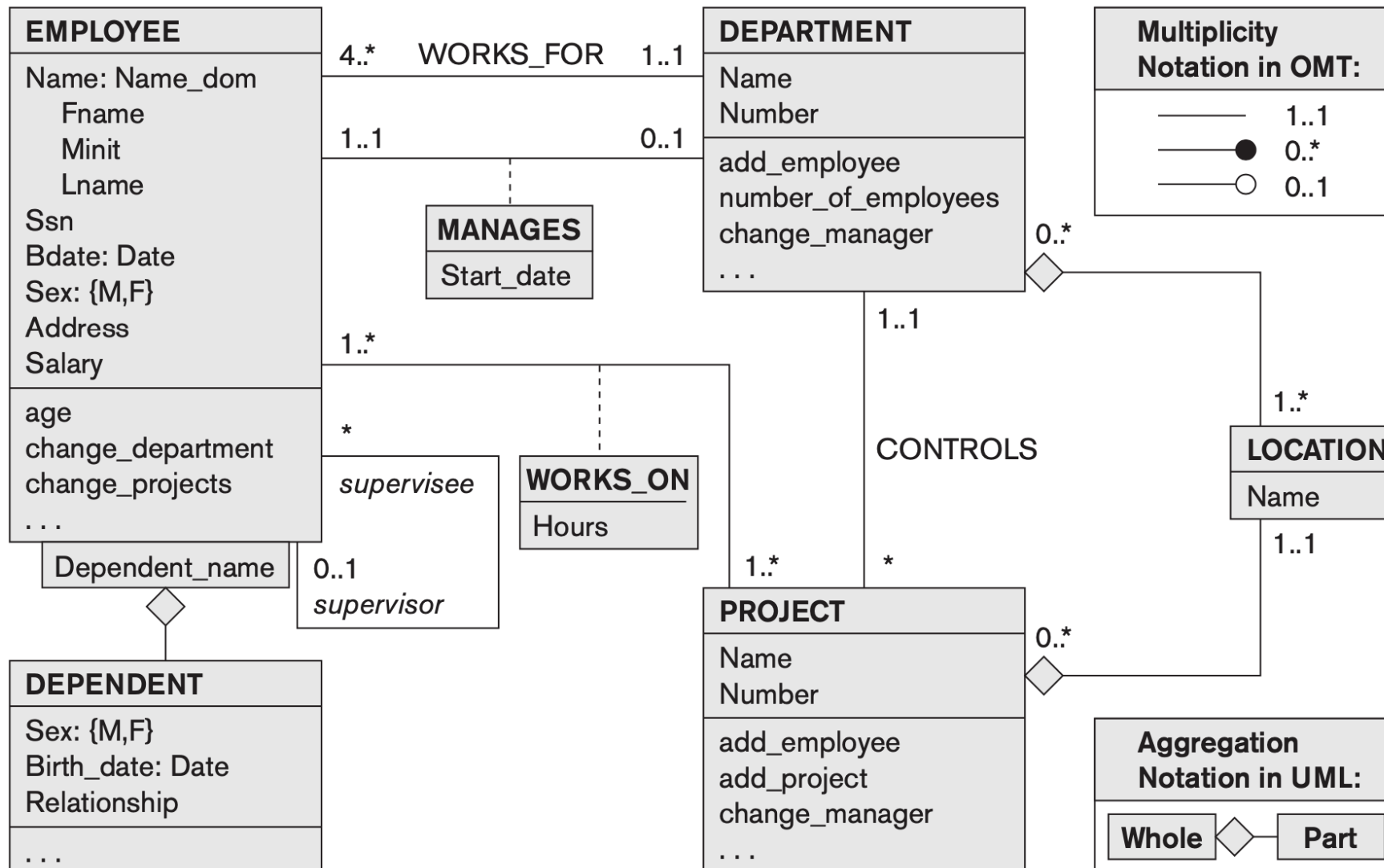
Symbol	Meaning		
	Entity		Composite Attribute
	Weak Entity		Derived Attribute
	Relationship		Total Participation of E_2 in R
	Identifying Relationship		Cardinality Ratio 1 : N for $E_1 : E_2$ in R
	Attribute		Structural Constraint (min, max) on Participation of E in R
	Key Attribute		
	Multivalued Attribute		

COMPANY:

mô hình ER



COMPANY: Lược đồ khái niệm UML



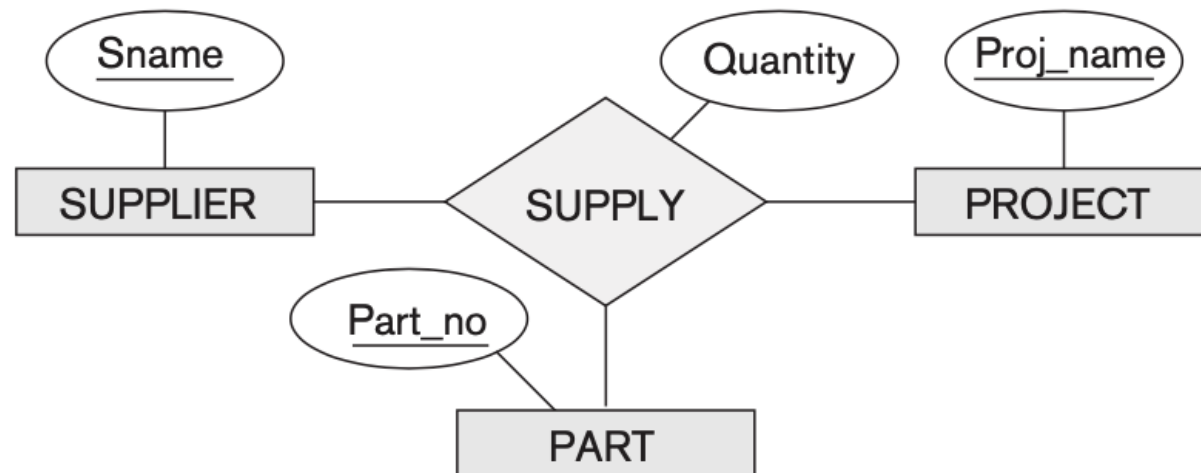
Chọn dùng Quan hệ Bậc hai hay bậc cao hơn (≥ 3)

Trong sơ đồ ER, quan hệ **R** có bậc **n** sẽ có **n** cạnh nối từ **R** tới các kiểu thực thể tham gia vào quan hệ.

Ví dụ: SUPPLY mô tả một quan hệ bậc 3.

Tập quan hệ gồm các thể hiện bộ 3 (**s**, **j**, **p**)

Ý nghĩa: **s** là nhà cung cấp **SUPPLIER** đang cung cấp phần **p**, là thiết bị, dụng cụ, nguyên vật liệu, chương trình v.v. được quy định bởi **PART**, cho dự án **j** mô tả bởi **PROJECT**

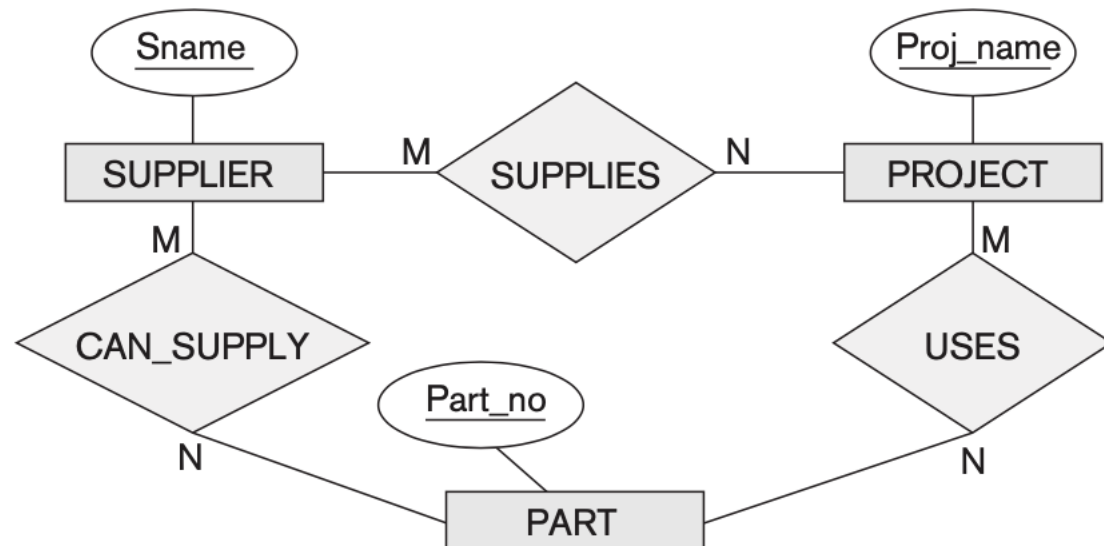


Chọn dùng Quan hệ Bậc hai hay bậc cao hơn (≥ 3)

Dùng các quan hệ bậc 2.

- **CAN_SUPPLY** gồm các thể hiện (s, p) mô tả s có thể cung cấp p và không phụ thuộc dự án cụ thể.
- **USES** gồm các thể hiện (j, p) mô tả việc dự án j đang sử dụng p .
- **SUPPLIES** gồm các thể hiện (s, j) mô tả s đang là nhà cung cấp cho dự án j .

Sự tồn tại ba bộ (s, p) , (j, p) và (s, j) **KHÔNG** có ý nghĩa tương đương với sự tồn tại của bộ (s, j, p)



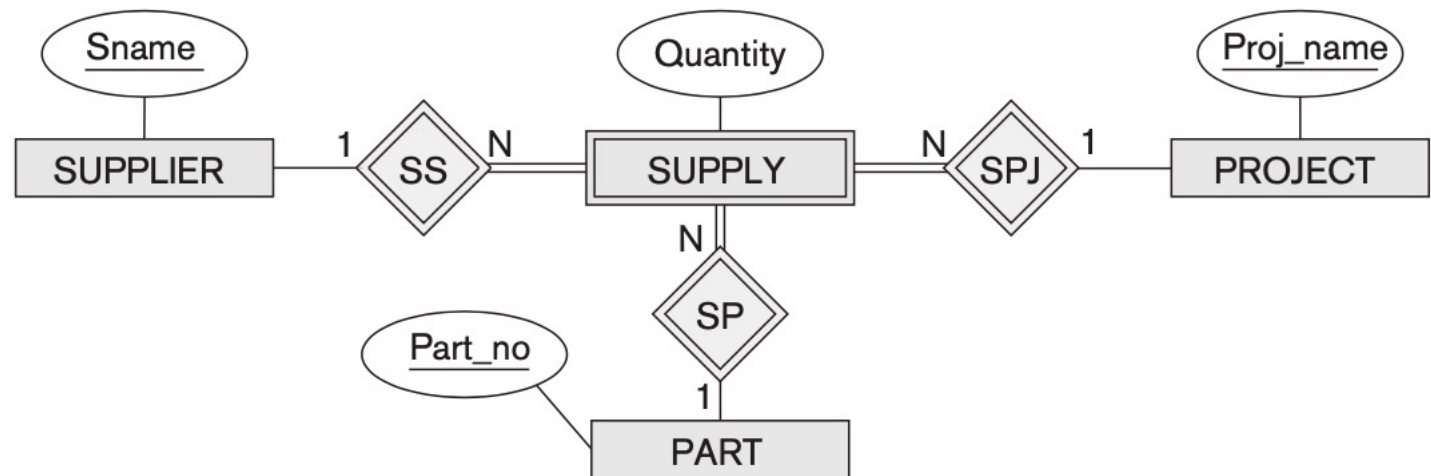
Chọn dùng Quan hệ Bậc hai hay bậc cao hơn (≥ 3)

- Tùy thuộc ngữ nghĩa và ý nghĩa thực tế của tình huống sử dụng để quyết định chọn cách thể hiện một quan hệ bậc n hay chia thành các quan hệ có bậc thấp hơn.
- Có thể kết hợp quan hệ bậc ba với một hoặc nhiều quan hệ bậc hai nếu ý nghĩa sử dụng khác nhau và ứng dụng có các chức năng cần đến các mối quan hệ đó.

Sử dụng kiểu Thực thể yếu

Nếu bắt buộc phải chia quan hệ bậc ba thành các quan hệ bậc hai

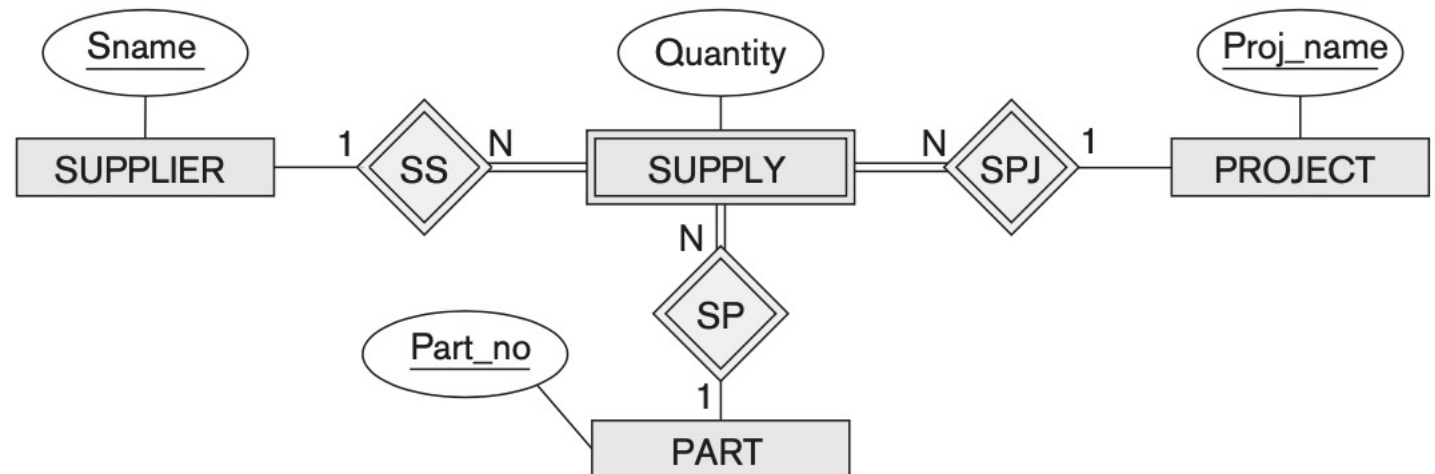
→ dùng kiểu thực thể yếu.



Sử dụng kiểu Thực thể yếu

Câu hỏi: Thử chuyển kiểu thực thể yếu SUPPLY thành kiểu thực thể thông thường, nghĩa là có thuộc tính khóa riêng?

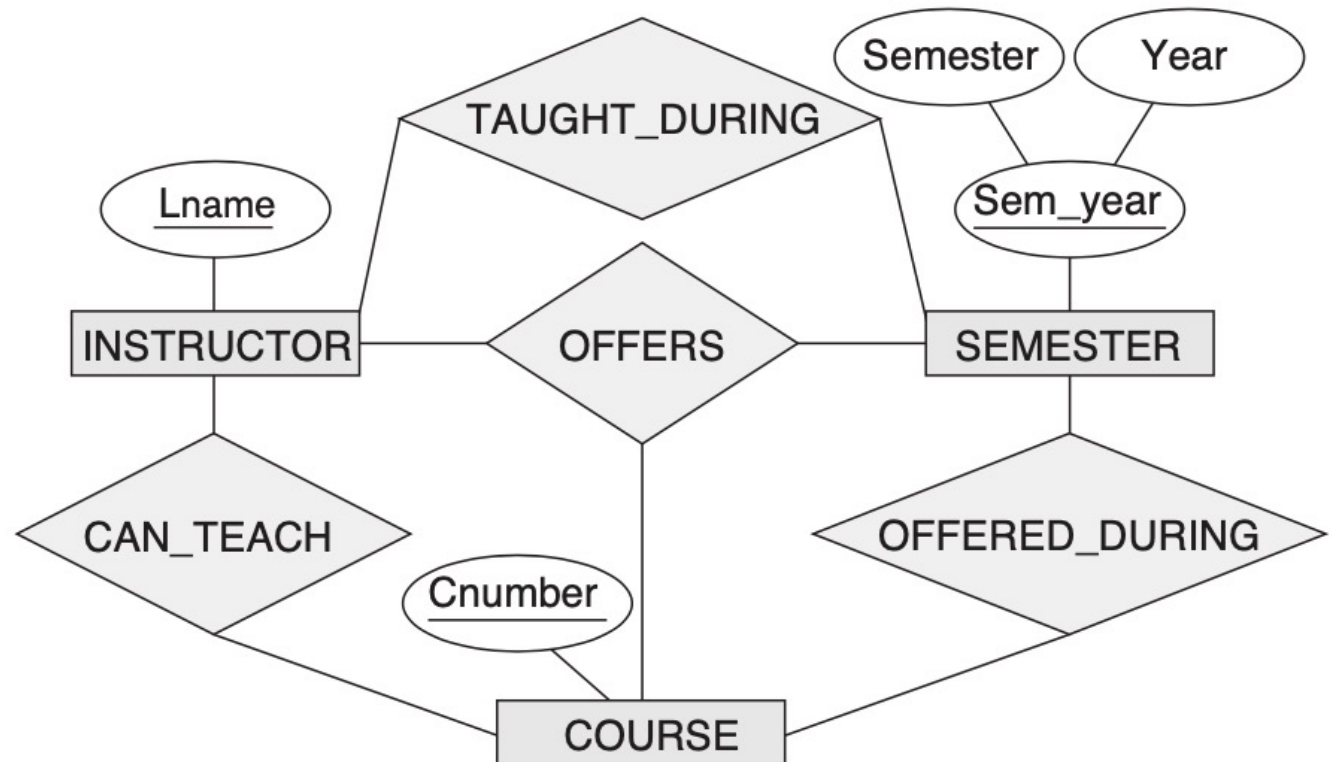
Trả lời:



Ví dụ: Tổ chức môn học

Câu hỏi: Hãy diễn giải sơ đồ này?

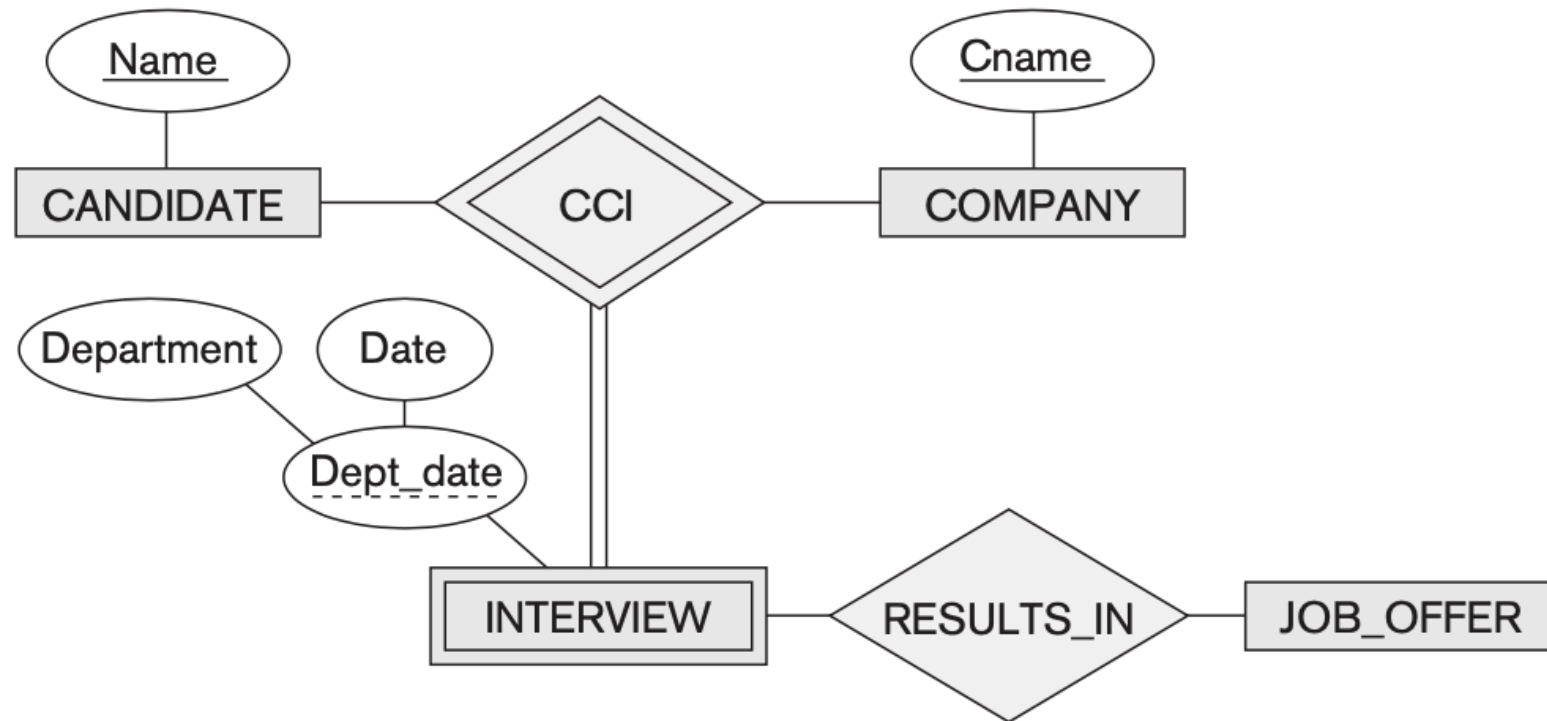
Trả lời:



Ví dụ: Theo dõi ứng viên dự phỏng vấn xin việc

Câu hỏi: Hãy diễn giải sơ đồ này?

Trả lời:



(CSDL môi giới việc làm)

Các Ràng buộc trên Quan hệ bậc cao ($n \geq 3$)

Có hai loại ràng buộc:

- Ràng buộc về tỷ lệ số lượng, tương tự như trong quan hệ bậc hai (quan hệ nhị phân) (xem [trang 52](#))
- Ràng buộc quy định giới hạn số lượng (min, max) (xem [trang 47](#))

Hết phần này